Hitachi 16-Bit Single-Chip Microcomputer

H8S/2678Series, H8S/2678R Series

H8S/2676 F-ZTAT™
HD64F2676

H8S/2676
HD6432676

H8S/2675
HD6432675

H8S/2674R
HD6412674R

H8S/2673
HD6432673

H8S/2670
HD6412670

Hardware Manual

HITACHI

ADE-602-242A

Rev. 2.0
04/05/02
Hitachi, Ltd.

Shaping intelligence
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 Handling of Product

1. Treatment of NC Pins

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

2. Treatment of Unused Input Pins

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

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

Note: Access to undefined or reserved addresses is prohibited.
The undefined or reserved addresses may be used to expand functions, or test registers may have been 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. General Precautions on Handling of Product
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

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

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

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

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

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

11. Index

Preface

The H8S/2678 Series and H8S/2678R Series are microcomputers (MCU) made up of the H8S/2600 CPU employing Hitachi's original architecture as their cores, and the peripheral functions required to configure a system.

The H8S/2600 CPU has an internal 32-bit configuration, sixteen 16-bit general registers, and a simple and optimized instruction set for high-speed operation. The H8S/2600 CPU can handle a 16-Mbyte linear address space.

This LSI is equipped with direct memory access controller (DMAC and EXDMAC) and data transfer controller (DTC) bus masters, ROM and RAM memory, a 16-bit timer pulse unit (TPU), a programmable pulse generator (PPG), an 8-bit timer (TMR), a watchdog timer (WDT), a serial communication interface (SCI and IrDA), a 10-bit A/D converter, an 8-bit D/A converter, and I/O ports as on-chip peripheral modules required for system configuration

A high functionality bus controller is also provided, enabling fast and easy connection of DRAM, SDRAM, and other kinds of memory.

A single-power flash memory (F-ZTAT™*) version and masked ROM version are available for this LSI's ROM. The F-ZTAT version provides flexibility as it can be reprogrammed in no time to cope with all situations from the early stages of mass production to full-scale mass production. This is particularly applicable to application devices with specifications that will most probably change.

This manual describes this LSI's hardware.

Note: * F-ZTAT™ is a trademark of Hitachi, Ltd.

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

Objective: This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users.
Refer to the H8S/2600 Series, H8S/2000 Series Programming Manual for a detailed description of the instruction set.

Notes on reading this manual:

In order to understand the overall functions of the chip

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

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

Read the H8S/2600 Series, H8S/2000 Series Programming Manual.

In order to understand the details of a register when its name is known

Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 23, List of Registers.

Examples:	Register name:	The following notation is used for cases when the same or a similar function, e.g. 16-bit timer pulse unit or serial communication, is implemented on more than one channel: XXX_N (XXX is the register name and N is the channel number)
	Bit order:	The MSB is on the left and the LSB is on the right.
	Number notation:	Binary is B'xxxx, hexadecimal is H'xxxx, decimal is xxxx.
	Signal notation:	An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

Related Manuals: The latest versions of all related manuals are available from our web site. Please ensure you have the latest versions of all documents you require. http://www.hitachisemiconductor.com/

H8S/2678 Series and H8S/2678R Series manuals:

Manual Title	ADE No.
H8S/2678 Series,H8S/2678R Series Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Programming Manual	ADE-602-083

User's manuals for development tools:

Manual Title	ADE No.
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	ADE-702-247
H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-702-282
H8S, H8/300 Series Hitachi Embedded Workshop, Hitachi Debugging Interface Tutorial	ADE-702-231
Hitachi Embedded Workshop User's Manual	ADE-702-201

Contents

Figures

Section 6 Bus Controller (BSC)

Section 7 DMA Controller (DMAC)

Section 8 EXDMA Controller

Section 9 Data Transfer Controller (DTC)

Section 11 16-Bit Timer Pulse Unit (TPU)

Section 24 Electrical Characteristics

Appendix

Tables

Section 1 Overview

1.1 Features

- High-speed H8S/2600 central processing unit with an internal 16-bit architecture
  Upward-compatible with H8/300 and H8/300H CPUs on an object level
  Sixteen 16-bit general registers
  69 basic instructions
- Various peripheral functions
  DMA controller (DMAC)
  EXDMA controller (EXDMAC)
  Data transfer controller (DTC)
  16-bit timer-pulse unit (TPU)
  Programmable pulse generator (PPG)
  8-bit timer (TMR)
  Watchdog timer (WDT)
  Asynchronous or clocked synchronous serial communication interface (SCI)
  10-bit A/D converter
  8-bit D/A converter
  Clock pulse generator
- On-chip memory

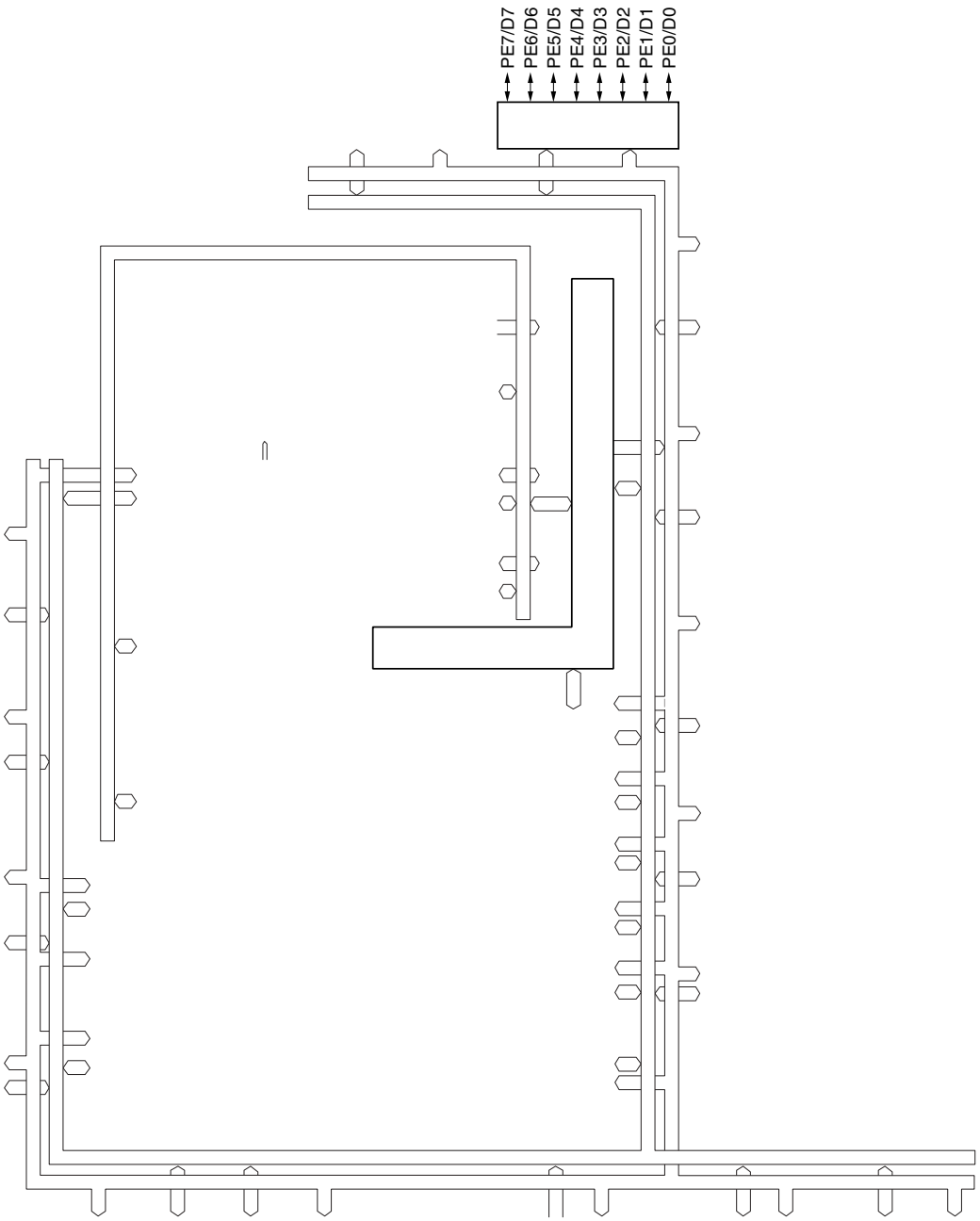
ROM Type	Model	ROM	RAM	Remarks
Flash memory Version	HD64F2676	256 kbytes	8 kbytes	
Masked ROM version	HD6432676	256 kbytes	8 kbytes	
	HD6432675	128 kbytes	8 kbytes	
	HD6432673	64 kbytes	8 kbytes	
ROMless version	HD6412674R	—	32 kbytes	
	HD6412670	—	8 kbytes	

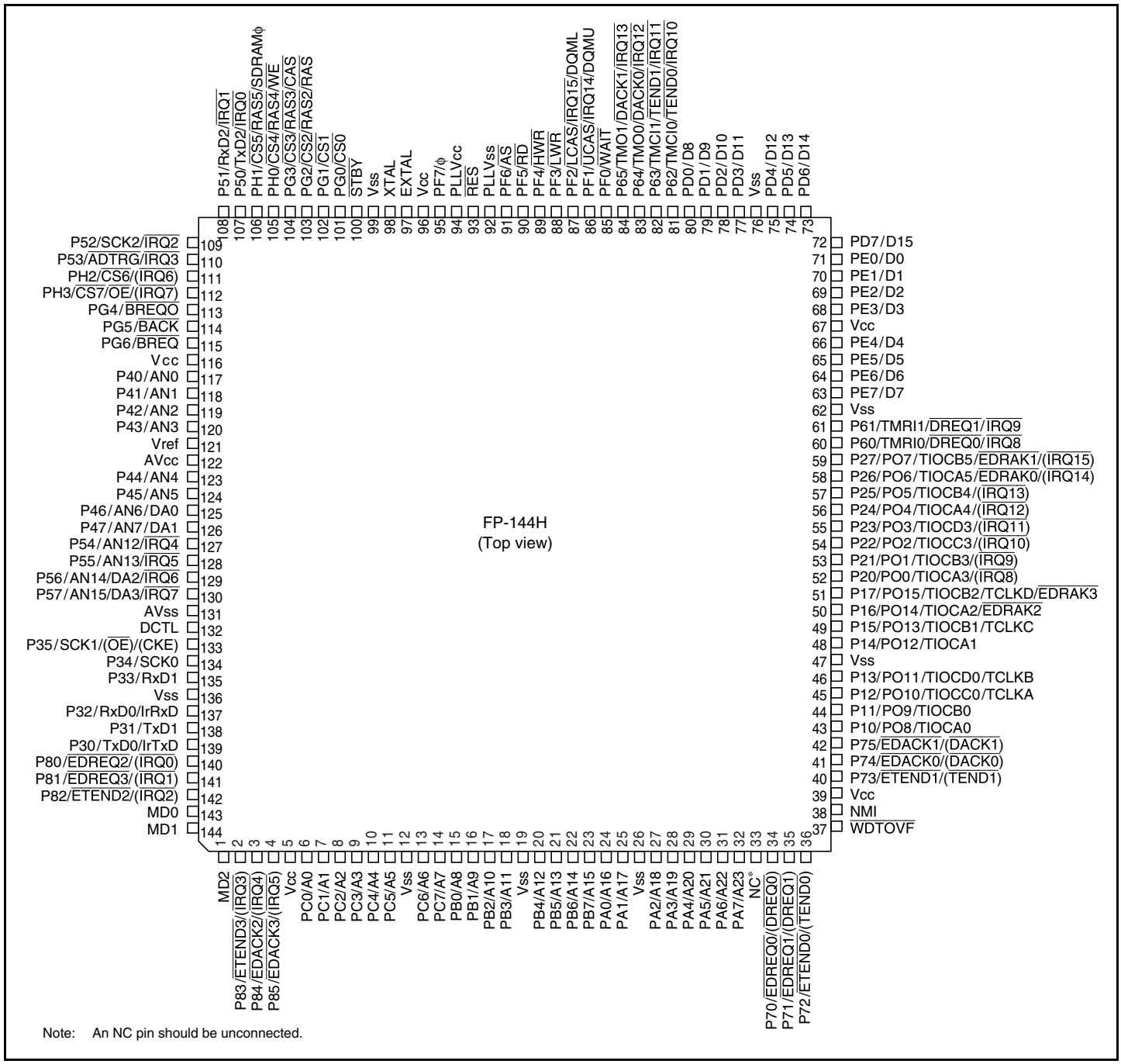
- General I/O ports
  I/O pins: 103
  Input-only pins: 12
- Supports various power-down states
- Compact package

Product	Package	(Code)	Mounting Height	Body Size	Pin Pitch
H8S/2678 Series	QFP-144	FP-144G	3.05 mm (Max.)	22.0 × 22.0 mm	0.5 mm
H8S/2678R Series	LQFP-144	FP-144H	1.70 mm (Max.)	22.0 × 22.0 mm	0.5 mm

P10/PO8/TIOCA0
P11/PO9/TIOCB0
P12/PO10/TIOCC0/TCLKA
P13/PO11/TIOCD0/TCLKB
P14/PO12/TIOCA1
P15/PO13/TIOCB1/TCLKC
P16/PO14/TIOCA2/EDRAK2
P17/PO15/TIOCB2/TCLKD/EDRAK3
P20/PO0/TIOCA3/(IRQ8)
P21/PO1/TIOCB3/(IRQ9)
P22/PO2/TIOCC3/(IRQ10)
P23/PO3/TIOCD3/(IRQ11)
P24/PO4/TIOCA4/(IRQ12)
P25/PO5/TIOCB4/(IRQ13)
P26/PO6/TIOCA5/EDRAK0/(IRQ14)
P27/PO7/TIOCB5/EDRAK1/(IRQ15)
Vref
P47/AN7/DA
P46/AN6/D
P45/A
P44/A
P43/
P42
P41
P40

HITACHI


PE7/D7
PE6/D6
PE5/D5
PE4/D4
PE3/D3
PE2/D2
PE1/D1
PE0/D0



Note: An NC pin should be unconnected.

Figure 1.4 H8S/2678R Series Pin Arrangement

1.3.2 Pin Arrangement in Each Operating Mode

Table 1.1 Pin Arrangement in Each Operating Mode

Pin No.	Pin Name					
				Mode 7		Flash Memory Programmer Mode
	Modes 1 and 5	Modes 2 and 6	Mode 4	EXPE = 1	EXPE = 0	
1	MD2	MD2	MD2	MD2	MD2	Vss
2	P83/ETEND3/ (IRQ3)	P83/ETEND3/ (IRQ3)	P83/ETEND3/ (IRQ3)	P83/ETEND3/ (IRQ3)	P83/(IRQ3)	NC
3	P84/EDACK2/ (IRQ4)	P84/EDACK2/ (IRQ4)	P84/EDACK2/ (IRQ4)	P84/EDACK2/ (IRQ4)	P84/(IRQ4)	NC
4	P85/EDACK3/ (IRQ5)	P85/EDACK3/ (IRQ5)	P85/EDACK3/ (IRQ5)	P85/EDACK3/ (IRQ5)	P85/(IRQ5)	NC
5	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
6	A0	A0	PC0/A0	PC0/A0	PC0	A0
7	A1	A1	PC1/A1	PC1/A1	PC1	A1
8	A2	A2	PC2/A2	PC2/A2	PC2	A2
9	A3	A3	PC3/A3	PC3/A3	PC3	A3
10	A4	A4	PC4/A4	PC4/A4	PC4	A4
11	A5	A5	PC5/A5	PC5/A5	PC5	A5
12	Vss	Vss	Vss	Vss	Vss	Vss
13	A6	A6	PC6/A6	PC6/A6	PC6	A6
14	A7	A7	PC7/A7	PC7/A7	PC7	A7
15	A8	A8	PB0/A8	PB0/A8	PB0	A8
16	A9	A9	PB1/A9	PB1/A9	PB1	A9
17	A10	A10	PB2/A10	PB2/A10	PB2	A10
18	A11	A11	PB3/A11	PB3/A11	PB3	A11
19	Vss	Vss	Vss	Vss	Vss	Vss
20	A12	A12	PB4/A12	PB4/A12	PB4	A12
21	A13	A13	PB5/A13	PB5/A13	PB5	A13
22	A14	A14	PB6/A14	PB6/A14	PB6	A14
23	A15	A15	PB7/A15	PB7/A15	PB7	A15
24	A16	A16	PA0/A16	PA0/A16	PA0	A16
25	A17	A17	PA1/A17	PA1/A17	PA1	A17
26	Vss	Vss	Vss	Vss	Vss	Vss
27	A18	A18	PA2/A18	PA2/A18	PA2	A18

Pin No.	Pin Name					
				Mode 7		Flash Memory Programmer Mode
	Modes 1 and 5	Modes 2 and 6	Mode 4	EXPE = 1	EXPE = 0	
28	A19	A19	PA3/A19	PA3/A19	PA3	NC
29	A20	A20	PA4/A20	PA4/A20	PA4	NC
30	PA5/A21	PA5/A21	PA5/A21	PA5/A21	PA5	NC
31	PA6/A22	PA6/A22	PA6/A22	PA6/A22	PA6	NC
32	PA7/A23	PA7/A23	PA7/A23	PA7/A23	PA7	NC
33	NC	NC	NC	NC	NC	NC
34	P70/EDREQ0/ (DREQ0)	P70/EDREQ0/ (DREQ0)	P70/EDREQ0/ (DREQ0)	P70/EDREQ0/ (DREQ0)	P70/(DREQ0)	NC
35	P71/EDREQ1/ (DREQ1)	P71/EDREQ1/ (DREQ1)	P71/EDREQ1/ (DREQ1)	P71/EDREQ1/ (DREQ1)	P71/(DREQ1)	NC
36	P72/ETEND0/ (TEND0)	P72/ETEND0/ (TEND0)	P72/ETEND0/ (TEND0)	P72/ETEND0/ (TEND0)	P72/(TEND0)	NC
37	WDTOVF	WDTOVF	WDTOVF	WDTOVF	WDTOVF	NC
38	NMI	NMI	NMI	NMI	NMI	Vcc
39	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
40	P73/ETEND1/ (TEND1)	P73/ETEND1/ (TEND1)	P73/ETEND1/ (TEND1)	P73/ETEND1/ (TEND1)	P73/(TEND1)	NC
41	P74/EDACK0/ (DACK0)	P74/EDACK0/ (DACK0)	P74/EDACK0/ (DACK0)	P74/EDACK0/ (DACK0)	P74/(DACK0)	NC
42	P75/EDACK1/ (DACK1)	P75/EDACK1/ (DACK1)	P75/EDACK1/ (DACK1)	P75/EDACK1/ (DACK1)	P75/(DACK1)	NC
43	P10/PO8/TIOCA0	P10/PO8/TIOCA0	P10/PO8/TIOCA0	P10/PO8/TIOCA0	P10/PO8/TIOCA0	NC
44	P11/PO9/TIOCB0	P11/PO9/TIOCB0	P11/PO9/TIOCB0	P11/PO9/TIOCB0	P11/PO9/TIOCB0	NC
45	P12/PO10/ TIOCC0/TCLKA	P12/PO10/ TIOCC0/TCLKA	P12/PO10/ TIOCC0/TCLKA	P12/PO10/ TIOCC0/TCLKA	P12/PO10/ TIOCC0/TCLKA	NC
46	P13/PO11/ TIOCD0/TCLKB	P13/PO11/ TIOCD0/TCLKB	P13/PO11/ TIOCD0/TCLKB	P13/PO11/ TIOCD0/TCLKB	P13/PO11/ TIOCD0/TCLKB	NC
47	Vss	Vss	Vss	Vss	Vss	Vss
48	P14/PO12/ TIOCA1	P14/PO12/ TIOCA1	P14/PO12/ TIOCA1	P14/PO12/ TIOCA1	P14/PO12/ TIOCA1	NC
49	P15/PO13/ TIOCB1/TCLKC	P15/PO13/ TIOCB1/TCLKC	P15/PO13/ TIOCB1/TCLKC	P15/PO13/ TIOCB1/TCLKC	P15/PO13/ TIOCB1/TCLKC	NC
50	P16/PO14/ TIOCA2/EDRAK2	P16/PO14/ TIOCA2/EDRAK2	P16/PO14/ TIOCA2/EDRAK2	P16/PO14/ TIOCA2/EDRAK2	P16/PO14/ TIOCA2	NC

Pin No.	Pin Name					
				Mode 7		Flash Memory Programmer Mode
	Modes 1 and 5	Modes 2 and 6	Mode 4	EXPE = 1	EXPE = 0	
51	P17/PO15/ TIOCB2/TCLKD/ EDRAK3	P17/PO15/ TIOCB2/TCLKD/ EDRAK3	P17/PO15/ TIOCB2/TCLKD/ EDRAK3	P17/PO15/ TIOCB2/TCLKD/ EDRAK3	P17/PO15/ TIOCB2/TCLKD	NC
52	P20/PO0/ TIOCA3/(IRQ8)	P20/PO0/ TIOCA3/(IRQ8)	P20/PO0/ TIOCA3/(IRQ8)	P20/PO0/ TIOCA3/(IRQ8)	P20/PO0/ TIOCA3/(IRQ8)	NC
53	P21/PO1/ TIOCB3/(IRQ9)	P21/PO1/ TIOCB3/(IRQ9)	P21/PO1/ TIOCB3/(IRQ9)	P21/PO1/ TIOCB3/(IRQ9)	P21/PO1/ TIOCB3/(IRQ9)	NC
54	P22/PO2/ TIOCC3/(IRQ10)	P22/PO2/ TIOCC3/(IRQ10)	P22/PO2/ TIOCC3/(IRQ10)	P22/PO2/ TIOCC3/(IRQ10)	P22/PO2/ TIOCC3/(IRQ10)	OE
55	P23/PO3/ TIOCD3/(IRQ11)	P23/PO3/ TIOCD3/(IRQ11)	P23/PO3/ TIOCD3/(IRQ11)	P23/PO3/ TIOCD3/(IRQ11)	P23/PO3/ TIOCD3/(IRQ11)	CE
56	P24/PO4/ TIOCA4/(IRQ12)	P24/PO4/ TIOCA4/(IRQ12)	P24/PO4/ TIOCA4/(IRQ12)	P24/PO4/ TIOCA4/(IRQ12)	P24/PO4/ TIOCA4/(IRQ12)	WE
57	P25/PO5/ TIOCB4/(IRQ13)	P25/PO5/ TIOCB4/(IRQ13)	P25/PO5/ TIOCB4/(IRQ13)	P25/PO5/ TIOCB4/(IRQ13)	P25/PO5/ TIOCB4/(IRQ13)	Vss
58	P26/PO6/ TIOCA5/ EDRAK0/(IRQ14)	P26/PO6/ TIOCA5/ EDRAK0/(IRQ14)	P26/PO6/ TIOCA5/ EDRAK0/(IRQ14)	P26/PO6/ TIOCA5/ EDRAK0/(IRQ14)	P26/PO6/ TIOCA5/(IRQ14)	NC
59	P27/PO7/ TIOCB5/ EDRAK1/(IRQ15)	P27/PO7/ TIOCB5/ EDRAK1/(IRQ15)	P27/PO7/ TIOCB5/ EDRAK1/(IRQ15)	P27/PO7/ TIOCB5/ EDRAK1/(IRQ15)	P27/PO7/ TIOCB5/(IRQ15)	NC
60	P60/TMRI0/ DREQ0/IRQ8	P60/TMRI0/ DREQ0/IRQ8	P60/TMRI0/ DREQ0/IRQ8	P60/TMRI0/ DREQ0/IRQ8	P60/TMRI0/ DREQ0/IRQ8	NC
61	P61/TMRI1/ DREQ1/IRQ9	P61/TMRI1/ DREQ1/IRQ9	P61/TMRI1/ DREQ1/IRQ9	P61/TMRI1/ DREQ1/IRQ9	P61/TMRI1/ DREQ1/IRQ9	NC
62	FWE*[1] Vss*[2]	FWE*[1] Vss*[2]	FWE*[1] Vss*[2]	FWE*[1] Vss*[2]	FWE*[1] Vss*[2]	FWE*[1] Vss*[2]
63	D7	PE7/D7	PE7/D7	PE7/D7	PE7	NC
64	D6	PE6/D6	PE6/D6	PE6/D6	PE6	NC
65	D5	PE5/D5	PE5/D5	PE5/D5	PE5	NC
66	D4	PE4/D4	PE4/D4	PE4/D4	PE4	NC
67	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
68	D3	PE3/D3	PE3/D3	PE3/D3	PE3	NC
69	D2	PE2/D2	PE2/D2	PE2/D2	PE2	NC
70	D1	PE1/D1	PE1/D1	PE1/D1	PE1	NC
71	D0	PE0/D0	PE0/D0	PE0/D0	PE0	NC
72	D15	D15	D15	D15	PD7	I/O7

Pin No.	Pin Name					
				Mode 7		Flash Memory Programmer Mode
	Modes 1 and 5	Modes 2 and 6	Mode 4	EXPE = 1	EXPE = 0	
73	D14	D14	D14	D14	PD6	I/O6
74	D13	D13	D13	D13	PD5	I/O5
75	D12	D12	D12	D12	PD4	I/O4
76	Vss	Vss	Vss	Vss	Vss	Vss
77	D11	D11	D11	D11	PD3	I/O3
78	D10	D10	D10	D10	PD2	I/O2
79	D9	D9	D9	D9	PD1	I/O1
80	D8	D8	D8	D8	PD0	I/O0
81	P62/TMCI0/ TEND0/IRQ10	P62/TMCI0/ TEND0/IRQ10	P62/TMCI0/ TEND0/IRQ10	P62/TMCI0/ TEND0/IRQ10	P62/TMCI0/ TEND0/IRQ10	NC
82	P63/TMCI1/ TEND1/IRQ11	P63/TMCI1/ TEND1/IRQ11	P63/TMCI1/ TEND1/IRQ11	P63/TMCI1/ TEND1/IRQ11	P63/TMCI1/ TEND1/IRQ11	NC
83	P64/TMO0/ DACK0/IRQ12	P64/TMO0/ DACK0/IRQ12	P64/TMO0/ DACK0/IRQ12	P64/TMO0/ DACK0/IRQ12	P64/TMO0/ DACK0/IRQ12	NC
84	P65/TMO1/ DACK1/IRQ13	P65/TMO1/ DACK1/IRQ13	P65/TMO1/ DACK1/IRQ13	P65/TMO1/ DACK1/IRQ13	P65/TMO1/ DACK1/IRQ13	NC
85	PF0/WAIT	PF0/WAIT	PF0/WAIT	PF0/WAIT	PF0	NC
86	PF1/UCAS/ IRQ14/DQMU*[2]	PF1/UCAS/ IRQ14/DQMU*[2]	PF1/UCAS/ IRQ14/DQMU*[2]	PF1/UCAS/ IRQ14/DQMU*[2]	PF1/IRQ14	NC
87	PF2/LCAS/ IRQ15/DQML*[2]	PF2/LCAS/ IRQ15/DQML*[2]	PF2/LCAS/ IRQ15/DQML*[2]	PF2/LCAS/ IRQ15/DQML*[2]	PF2/IRQ15	NC
88	PF3/LWR	PF3/LWR	PF3/LWR	PF3/LWR	PF3	NC
89	HWR	HWR	HWR	HWR	PF4	NC
90	RD	RD	RD	RD	PF5	NC
91	PF6/AS	PF6/AS	PF6/AS	PF6/AS	PF6	NC
92	PLLVss	PLLVss	PLLVss	PLLVss	PLLVss	Vss
93	RES	RES	RES	RES	RES	RES
94	PLLVcc	PLLVcc	PLLVcc	PLLVcc	PLLVcc	Vcc
95	PF7/ ϕ	PF7/ ϕ	PF7/ ϕ	PF7/ ϕ	PF7/ ϕ	NC
96	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
97	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL
98	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL
99	Vss	Vss	Vss	Vss	Vss	Vss
100	STBY	STBY	STBY	STBY	STBY	Vcc

Pin No.	Pin Name					
				Mode 7		Flash Memory Programmer Mode
	Modes 1 and 5	Modes 2 and 6	Mode 4	EXPE = 1	EXPE = 0	
101	PG0/CS0	PG0/CS0	PG0/CS0	PG0/CS0	PG0	NC
102	PG1/CS1	PG1/CS1	PG1/CS1	PG1/CS1	PG1	NC
103	PG2/CS2/ RAS2*2/RAS*2	PG2/CS2/ RAS2*2/RAS*2	PG2/CS2/ RAS2*2/RAS*2	PG2/CS2/ RAS2*2/RAS*2	PG2	NC
104	PG3/CS3/ RAS3*2/CAS*2	PG3/CS3/ RAS3*2/CAS*2	PG3/CS3/ RAS3*2/CAS*2	PG3/CS3/ RAS3*2/CAS*2	PG3	NC
105	PH0/CS4/ RAS4*2/WE*2	PH0/CS4/ RAS4*2/WE*2	PH0/CS4/ RAS4*2/WE*2	PH0/CS4/ RAS4*2/WE*2	PH0	NC
106	PH1/CS5/ RAS5*2/ SDRAMϕ*2	PH1/CS5/ RAS5*2/ SDRAMϕ*2	PH1/CS5/ RAS5*2/ SDRAMϕ*2	PH1/CS5/ RAS5*2/ SDRAMϕ*2	PH1	NC
107	P50/TxD2/IRQ0	P50/TxD2/IRQ0	P50/TxD2/IRQ0	P50/TxD2/IRQ0	P50/TxD2/IRQ0	Vss
108	P51/RxD2/IRQ1	P51/RxD2/IRQ1	P51/RxD2/IRQ1	P51/RxD2/IRQ1	P51/RxD2/IRQ1	Vss
109	P52/SCK2/IRQ2	P52/SCK2/IRQ2	P52/SCK2/IRQ2	P52/SCK2/IRQ2	P52/SCK2/IRQ2	Vcc
110	P53/ADTRG/ IRQ3	P53/ADTRG/ IRQ3	P53/ADTRG/ IRQ3	P53/ADTRG/ IRQ3	P53/ADTRG/ IRQ3	NC
111	PH2/CS6/(IRQ6)	PH2/CS6/(IRQ6)	PH2/CS6/(IRQ6)	PH2/CS6/(IRQ6)	PH2/(IRQ6•	NC
112	PH3/CS7/OE/ (IRQ7)/CKE*2	PH3/CS7/OE/ (IRQ7)/CKE*2	PH3/CS7/OE/ (IRQ7)/CKE*2	PH3/CS7/OE/ (IRQ7)/CKE*2	PH3/(IRQ7)	NC
113	PG4/BREQO	PG4/BREQO	PG4/BREQO	PG4/BREQO	PG4	NC
114	PG5/BACK	PG5/BACK	PG5/BACK	PG5/BACK	PG5	NC
115	PG6/BREQ	PG6/BREQ	PG6/BREQ	PG6/BREQ	PG6	NC
116	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
117	P40/AN0	P40/AN0	P40/AN0	P40/AN0	P40/AN0	NC
118	P41/AN1	P41/AN1	P41/AN1	P41/AN1	P41/AN1	NC
119	P42/AN2	P42/AN2	P42/AN2	P42/AN2	P42/AN2	NC
120	P43/AN3	P43/AN3	P43/AN3	P43/AN3	P43/AN3	NC
121	Vref	Vref	Vref	Vref	Vref	NC
122	AVcc	AVcc	AVcc	AVcc	AVcc	Vcc
123	P44/AN4	P44/AN4	P44/AN4	P44/AN4	P44/AN4	NC
124	P45/AN5	P45/AN5	P45/AN5	P45/AN5	P45/AN5	NC
125	P46/AN6/DA0	P46/AN6/DA0	P46/AN6/DA0	P46/AN6/DA0	P46/AN6/DA0	NC
126	P47/AN7/DA1	P47/AN7/DA1	P47/AN7/DA1	P47/AN7/DA1	P47/AN7/DA1	NC
127	P54/AN12/IRQ4	P54/AN12/IRQ4	P54/AN12/IRQ4	P54/AN12/IRQ4	P54/AN12/IRQ4	NC

Pin No.	Pin Name					
				Mode 7		Flash Memory Programmer Mode
	Modes 1 and 5	Modes 2 and 6	Mode 4	EXPE = 1	EXPE = 0	
128	P55/AN13/IRQ5	P55/AN13/IRQ5	P55/AN13/IRQ5	P55/AN13/IRQ5	P55/AN13/IRQ5	NC
129	P56/AN14/DA2/ IRQ6	P56/AN14/DA2/ IRQ6	P56/AN14/DA2/ IRQ6	P56/AN14/DA2/ IRQ6	P56/AN14/DA2/ IRQ6	NC
130	P57/AN15/DA3/ IRQ7	P57/AN15/DA3/ IRQ7	P57/AN15/DA3/ IRQ7	P57/AN15/DA3/ IRQ7	P57/AN15/DA3/ IRQ7	NC
131	AVss	AVss	AVss	AVss	AVss	Vss
132	NC*3 DCTL*2	NC*3 DCTL*2	NC*3 DCTL*2	NC*3 DCTL*2	NC*3 DCTL*2	NC*3 Vss *2
133	P35/SCK1/(OE)/ (CKE)*2	P35/SCK1/(OE)/ (CKE)*2	P35/SCK1/(OE)/ (CKE)*2	P35/SCK1/(OE)/ (CKE)*2	P35/SCK1	NC
134	P34/SCK0	P34/SCK0	P34/SCK0	P34/SCK0	P34/SCK0	NC
135	P33/RxD1	P33/RxD1	P33/RxD1	P33/RxD1	P33/RxD1	NC
136	Vss	Vss	Vss	Vss	Vss	Vss
137	P32/RxD0/IrRxD	P32/RxD0/IrRxD	P32/RxD0/IrRxD	P32/RxD0/IrRxD	P32/RxD0/IrRxD	Vcc
138	P31/TxD1	P31/TxD1	P31/TxD1	P31/TxD1	P31/TxD1	NC
139	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD	NC
140	P80/EDREQ2/ (IRQ0)	P80/EDREQ2/ (IRQ0)	P80/EDREQ2/ (IRQ0)	P80/EDREQ2/ (IRQ0)	P80/(IRQ0)	NC
141	P81/EDREQ3/ (IRQ1)	P81/EDREQ3/ (IRQ1)	P81/EDREQ3/ (IRQ1)	P81/EDREQ3/ (IRQ1)	P81/(IRQ1)	NC
142	P82/ETEND2/ (IRQ2)	P82/ETEND2/ (IRQ2)	P82/ETEND2/ (IRQ2)	P82/ETEND2/ (IRQ2)	P82/(IRQ2)	NC
143	MD0	MD0	MD0	MD0	MD0	Vss
144	MD1	MD1	MD1	MD1	MD1	Vss

Notes: *1 The FWE pin is used only in the flash memory version of the H8S/2678 Series. In the masked ROM and ROMless versions of the H8S/2678 Series, this is an NC pin.

*2 Only for the H8S/2678R Series.

*3 Only for the H8S/2678 Series.

1.3.3 Pin Functions

Table 1.2 Pin Functions

Type	Symbol	Pin No. FP-144G (H8S/2678 Series)	Pin No. FP-144H (H8S/2678R Series)	I/O	Function
Power	V_{CC}	5, 39, 67, 96, 116	5, 39, 67, 96, 116	Input	For connection to the power supply. All V_{CC} pins should be connected to the system power supply.
	V_{SS}	12, 19, 26, 47, 76, 99, 136	12, 19, 26, 47, 76, 99, 136	Input	For connection to ground. All V_{SS} pins should be connected to the system power supply (0 V).
	$PLLV_{CC}$	94	94	Input	Power supply pin for the on-chip PLL oscillator.
	$PLLV_{SS}$	92	92	Input	Ground pin for the on-chip PLL oscillator.
Clock	XTAL	98	98	Input	For connection to a crystal oscillator. See section 21, Clock Pulse Generator for typical connection diagrams for a crystal oscillator and external clock input.
	EXTAL	97	97	Input	For connection to a crystal oscillator. The EXTAL pin can also input an external clock. See section 21, Clock Pulse Generator for typical connection diagrams for a crystal oscillator and external clock input.
	ϕ	95	95	Output	Supplies the system clock to external devices.
	SDRAMϕ	—	106	Output	When a synchronous DRAM is connected, this pin is connected to the CLK pin of the synchronous DRAM. For details, refer to section 6, Bus Controller.
Operating mode control	MD2 MD1 MD0	1, 144, 143	1, 144, 143	Input	These pins set the operating mode. These pins should not be changed while the MCU is operating.

Type	Symbol	Pin No. FP-144G (H8S/2678 Series)	Pin No. FP-144H (H8S/2678R Series)	I/O	Function
Operating mode control	DCTL	—	132	Input	When this pin is driven high, SDRAMϕ dedicated to the synchronous DRAM is output. When not using the synchronous DRAM interface, drive this pin low. The level of this pin must not be changed during operation.
System control	RES	93	93	Input	When this pin is driven low, the chip is reset.
	STBY	100	100	Input	When this pin is driven low, a transition is made to hardware standby mode.
	BREQ	115	115	Input	Requests chip to release the bus to an external bus master.
	BREQO	113	113	Output	External bus request signal used when an internal bus master accesses external space when the external bus is released.
	BACK	114	114	Output	Indicates that the bus has been released to an external bus master.
	FWE	62	—	Input	Enables/disables flash memory. This pin is only used in the flash memory version.

Type	Symbol	Pin No. FP-144G (H8S/2678 Series)	Pin No. FP-144H (H8S/2678R Series)	I/O	Function
Bus control	WAIT	85	85	Input	Requests insertion of a wait state in the bus cycle when accessing external 3-state address space.
	OE (OE)	112, 133	112, 133	Output	Output enable signal for DRAM interface space. The output pins of OE and (OE) are selected by the port function control register 2 (PFCR2) of port 3.
	CKE (CKE)	—	112, 133	Output	Clock enable signal of the synchronous DRAM interface space. The output pins of CKE and (CKE) are selected by the port function control register 2 (PFCR2) of port 3.
Interrupt signals	NMI	38	38	Input	Nonmaskable interrupt request pin. Fix high when not used.
	IRQ15 to IRQ0	87, 86, 84 to 81, 61, 60, 130 to 127, 110 to 107	87, 86, 84 to 81, 61, 60, 130 to 127, 110 to 107	Input	These pins request a maskable interrupt. The input pins of DREQn and (DREQn) are selected by the IRQ pin select register (ITSR) of the interrupt controller. (n = 0 to 15)
	(IRQ15) to (IRQ0)	59 to 52, 112, 111, 4 to 2, 142 to 140	59 to 52, 112, 111, 4 to 2, 142 to 140		
DMA controller (DMAC)	DREQ1 DREQ0 (DREQ1) (DREQ0)	61, 60, 35, 34	61, 60, 35, 34	Input	These signals request DMAC activation. The input pins of DREQn and (DREQn) are selected by the IRQ pin select register (ITSR) of the interrupt controller. (n = 0 to 15)
	TEND1 TEND0 (TEND1) (TEND0)	82, 81, 40, 36	82, 81, 40, 36	Output	These signals indicate the end of DMAC data transfer. (n = 0 to 15)

/REQ i1(15))]TJET3 TD0

Type	Symbol	Pin No. FP-144G (H8S/2678 Series)	Pin No. FP-144H (H8S/2678R Series)	I/O	Function
DMA controller (DMAC)	DACK1 DACK0 (DACK1) (DACK0)	84, 83, 42, 41	84, 83, 42, 41	Output	DMAC single address transfer acknowledge signals. The input pins of DACKn and (DACKn) are selected by the port function control register 2 (PFCR2) of port 3. (n = 1, 0)
EXDMA controller (EXDMAC)	EDREQ3 to EDREQ0	141, 140, 35, 34	141, 140, 35, 34	Input	These signals request EXDMAC activation.
	ETEND3 to ETEND0	2, 142, 40, 36	2, 142, 40, 36	Output	These signals indicate the end of EXDMAC data transfer.
	EDACK3 to EDACK0	4, 3, 42, 41	4, 3, 42, 41	Output	EXDMAC single address transfer acknowledge signals.
	EDRAK3 to EDRAK0	51, 50, 59, 58	51, 50, 59, 58	Output	These signals notify an external device of acceptance and start of execution of a DMA transfer request.
16-bit timer pulse unit (TPU)	TCLKA TCLKB TCLKC TCLKD	45, 46, 49, 51	45, 46, 49, 51	Input	External clock input pins.
	TIOCA0 TIOCB0 TIOCC0 TIOCD0	43, 44, 45, 46	43, 44, 45, 46	Input/ output	TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
	TIOCA1 TIOCB1	48, 49	48, 49	Input/ output	TGRA_1 and TGRB_1 input capture input/output compare output/PWM output pins.
	TIOCA2 TIOCB2	50, 51	50, 51	Input/ output	TGRA_2 and TGRB_2 input capture input/output compare output/PWM output pins.
	TIOCA3 TIOCB3 TIOCC3 TIOCD3	52, 53, 54, 55	52, 53, 54, 55	Input/ output	TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.
	TIOCA4 TIOCB4	56, 57	56, 57	Input/ output	TGRA_4 and TGRB_4 input capture input/output compare output/PWM output pins.

Type	Symbol	Pin No. FP-144G (H8S/2678 Series)	Pin No. FP-144H (H8S/2678R Series)	I/O	Function
16-bit timer pulse unit (TPU)	TIOCA5, TIOCB5	58, 59	58, 59	Input/ output	TGRA_5 and TGRB_5 input capture input/output compare output/PWM output pins.
Programmable pulse generator (PPG)	PO15 to PO0	51 to 48, 46 to 43, 59 to 52	51 to 48, 46 to 43, 59 to 52	Output	Pulse output pins.
8-bit timer	TMO0 TMO1	83, 84	83, 84	Output	Waveform output pins with output compare function.
	TMCI0 TMCI1	81, 82	81, 82	Input	External event input pins.
	TMRI0 TMRI1	60, 61	60, 61	Input	Counter reset input pins.
Watchdog timer (WDT)	WDTOVF	37	37	Output	Counter overflow signal output pin in watchdog timer mode.
Serial communication interface (SCI)/smart card interface (SCI_0 with IrDA function)	TxD2 TxD1 TxD0/IrTxD	107, 138, 139	107, 138, 139	Output	Data output pins.
	RxD2 RxD1 RxD0/ IrRxD	108, 135, 137	108, 135, 137	Input	Data input pins.
	SCK2 SCK1 SCK0	109, 133, 134	109, 133, 134	Input/ output	Clock input/output pins.
A/D converter	AN15 to AN12, AN7 to AN0	130 to 127, 126 to 123, 120 to 117	130 to 127, 126 to 123, 120 to 117	Input	Analog input pins for the A/D converter.
	ADTRG	110	110	Input	Pin for input of an external trigger to start A/D conversion.
D/A converter	DA3 to DA0	130, 129, 126, 125	130, 129, 126, 125	Output	Analog input pins for the D/A converter.
A/D converter, D/A converter	AV_{CC}	122	122	Input	The analog power-supply pin for the A/D converter and D/A converter. When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+3 V).

Type	Symbol	Pin No. FP-144G (H8S/2678 Series)	Pin No. FP-144H (H8S/2678R Series)	I/O	Function
A/D converter, D/A converter	AV_{SS}	131	131	Input	The ground pin for the A/D converter and D/A converter. This pin should be connected to the system power supply (0 V).
	Vref	121	121	Input	The reference voltage input pin for the A/D converter and D/A converter. When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+3 V).
I/O ports	P17 to P10	51 to 48, 46 to 43	51 to 48, 46 to 43	Input/ output	Eight input/output pins.
	P27 to P20	59 to 52	59 to 52	Input/ output	Eight input/output pins.
	P35 to P30	133 to 135, 137 to 139	133 to 135, 137 to 139	Input/ output	Six input/output pins.
	P47 to P40	126 to 123, 120 to 117	126 to 123, 120 to 117	Input	Eight input pins.
	P57 to P54	130 to 127	130 to 127	Input	Four input pins.
	P53 to P50	110 to 107	110 to 107	Input/ output	Four input/output pins.
	P65 to P60	84 to 81, 61, 60	84 to 81, 61, 60	Input/ output	Six input/output pins.
	P75 to P70	42 to 40, 36 to 34	42 to 40, 36 to 34	Input/ output	Six input/output pins.
	P85 to P80	4 to 2, 142 to 140	4 to 2, 142 to 140	Input/ output	Six input/output pins.
	PA7 to PA0	32 to 27, 25, 24	32 to 27, 25, 24	Input/ output	Eight input/output pins.
	PB7 to PB0	23 to 20, 18 to 15	23 to 20, 18 to 15	Input/ output	Eight input/output pins.
	PC7 to PC0	14, 13, 11 to 6	14, 13, 11 to 6	Input/ output	Eight input/output pins.

Type	Symbol	Pin No. FP-144G (H8S/2678 Series)	Pin No. FP-144H (H8S/2678R Series)	I/O	Function
I/O ports	PD7 to PD0	72 to 75, 77 to 80	72 to 75, 77 to 80	Input/ output	Eight input/output pins.
	PE7 to PE0	63 to 66, 68 to 71	63 to 66, 68 to 71	Input/ output	Eight input/output pins.
	PF7 to PF0	95, 91 to 85	95, 91 to 85	Input/ output	Eight input/output pins.
	PG6 to PG0	115 to 113, 104 to 101	115 to 113, 104 to 101	Input/ output	Seven input/output pins.
	PH3 to PH0	112, 111, 106, 105	112, 111, 106, 105	Input/ output	Four input/output pins.

Section 2 CPU

The H8S/2600 CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 and H8/300H CPUs. The H8S/2600 CPU has sixteen 16-bit general registers, can address a 16-Mbyte linear address space, and is ideal for realtime control. This section describes the H8S/2600 CPU. The usable modes and address spaces differ depending on the product. For details on each product, refer to section 3, MCU Operating Modes.

2.1 Features

- Upward-compatible with H8/300 and H8/300H CPUs
  Can execute H8/300 and H8/300H object programs
- General-register architecture
  Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-bit registers
- Sixty-nine basic instructions
  8/16/32-bit arithmetic and logic instructions
  Multiply and divide instructions
  Powerful bit-manipulation instructions
  Multiply-and-accumulate instruction
- Eight addressing modes
  Register direct [Rn]
  Register indirect [@ERn]
  Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  Immediate [#xx:8, #xx:16, or #xx:32]
  Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  Memory indirect [@@aa:8]
- 16-Mbyte address space
  Program: 16 Mbytes
  Data: 16 Mbytes
- High-speed operation
  All frequently-used instructions execute in one or two states
  8/16/32-bit register-register add/subtract: 1 state
  8 × 8-bit register-register multiply: 3 states
  16 ÷ 8-bit register-register divide: 12 states
  16 × 16-bit register-register multiply: 4 states
  32 ÷ 16-bit register-register divide: 20 states

- Two CPU operating modes
  Normal mode*
  Advanced mode
- Power-down state
  Transition to power-down state by SLEEP instruction
  CPU clock speed selection

Note: Normal mode is not available in this LSI.

2.1.1 Differences between H8S/2600 CPU and H8S/2000 CPU

The differences between the H8S/2600 CPU and the H8S/2000 CPU are as shown below.

- Register configuration
  The MAC register is supported only by the H8S/2600 CPU.
- Basic instructions
  The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported only by the H8S/2600 CPU.
- The number of execution states of the MULXU and MULXS instructions

		Execution States	
Instruction	**Mnemonic**	**H8S/2600**	**H8S/2000**
MULXU	MULXU.B Rs, Rd	3	12
	MULXU.W Rs, ERd	4	20
MULXS	MULXS.B Rs, Rd	4	13
	MULXS.W Rs, ERd	5	21

In addition, there are differences in address space, CCR and EXR register functions, power-down modes, etc., depending on the model.

2.1.2 Differences from H8/300 CPU

In comparison to the H8/300 CPU, the H8S/2600 CPU has the following enhancements.

- More general registers and control registers
  Eight 16-bit expanded registers, and one 8-bit and two 32-bit control registers, have been added.
- Expanded address space
  Normal mode supports the same 64-kbyte address space as the H8/300 CPU.
  Advanced mode supports a maximum 16-Mbyte address space.

- Enhanced addressing
  The addressing modes have been enhanced to make effective use of the 16-Mbyte address space.
- Enhanced instructions
  Addressing modes of bit-manipulation instructions have been enhanced.
  Signed multiply and divide instructions have been added.
  A multiply-and-accumulate instruction has been added.
  Two-bit shift and rotate instructions have been added.
  Instructions for saving and restoring multiple registers have been added.
  A test and set instruction has been added.
- Higher speed
  Basic instructions execute twice as fast.

Note: Normal mode is not available in this LSI.

2.1.3 Differences from H8/300H CPU

In comparison to the H8/300H CPU, the H8S/2600 CPU has the following enhancements.

- Additional control register
  One 8-bit and two 32-bit control registers have been added.
- Enhanced instructions
  Addressing modes of bit-manipulation instructions have been enhanced.
  A multiply-and-accumulate instruction has been added.
  Two-bit shift and rotate instructions have been added.
  Instructions for saving and restoring multiple registers have been added.
  A test and set instruction has been added.
- Higher speed
  Basic instructions execute twice as fast.

2.2 CPU Operating Modes

The H8S/2600 CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64-kbyte address space. Advanced mode supports a maximum 16-Mbyte total address space. The mode is selected by the mode pins.

2.2.1 Normal Mode

The exception vector table and stack have the same structure as in the H8/300 CPU.

- Address Space

  The H8S/2600 CPU provides linear access to a maximum 64-kbyte address space.
- Extended Registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers.

  When En is used as a 16-bit register it can contain any value, even when the corresponding general register (Rn) is used as an address register. If the general register is referenced in the register indirect addressing mode with pre-decrement (@–Rn) or post-increment (@Rn+) and a carry or borrow occurs, however, the value in the corresponding extended register (En) will be affected.
- Instruction Set

  All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid.
- Exception Vector Table and Memory Indirect Branch Addresses

  In normal mode the top area starting at H'0000 is allocated to the exception vector table. One branch address is stored per 16 bits. The exception vector table in normal mode is shown in figure 2.1. For details of the exception vector table, see section 4, Exception Handling.

  The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In normal mode the operand is a 16-bit word operand, providing a 16-bit branch address. Branch addresses can be stored in the top area from H'0000 to H'00FF. Note that this area is also used for the exception vector table.
- Stack Structure

  When the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.2. EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 4, Exception Handling.

Note: Normal mode is not available in this LSI.

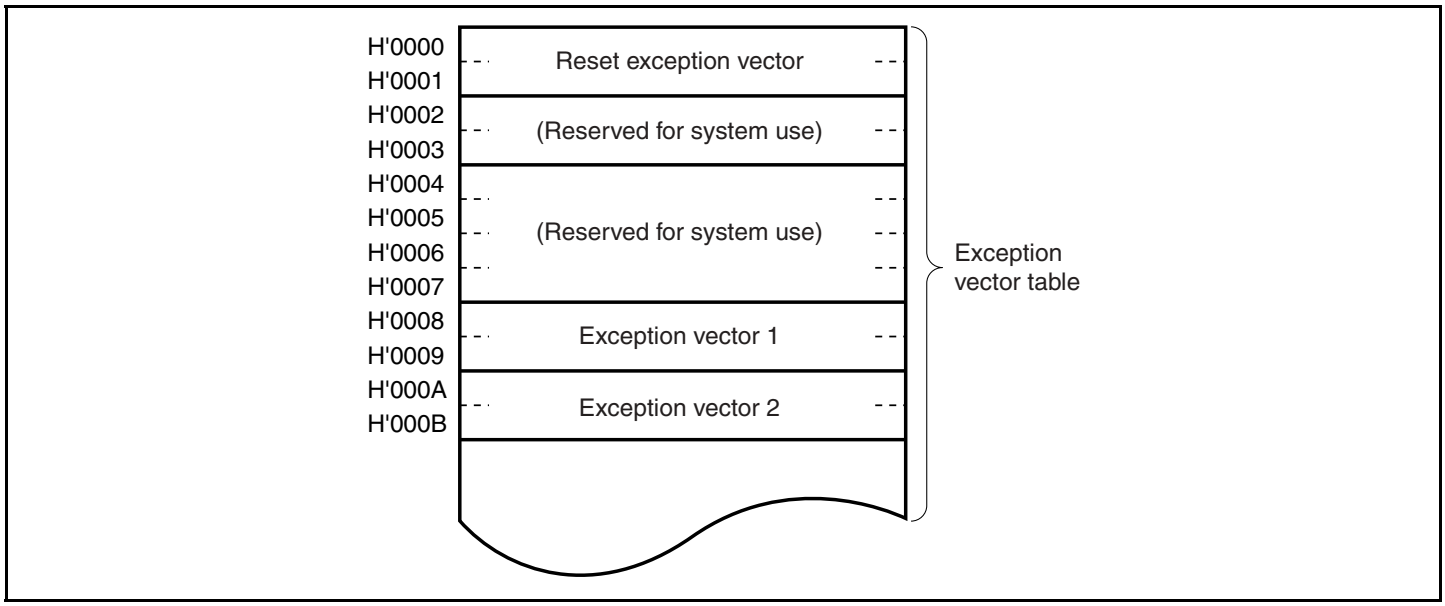


Figure 2.1 Exception Vector Table (Normal Mode)

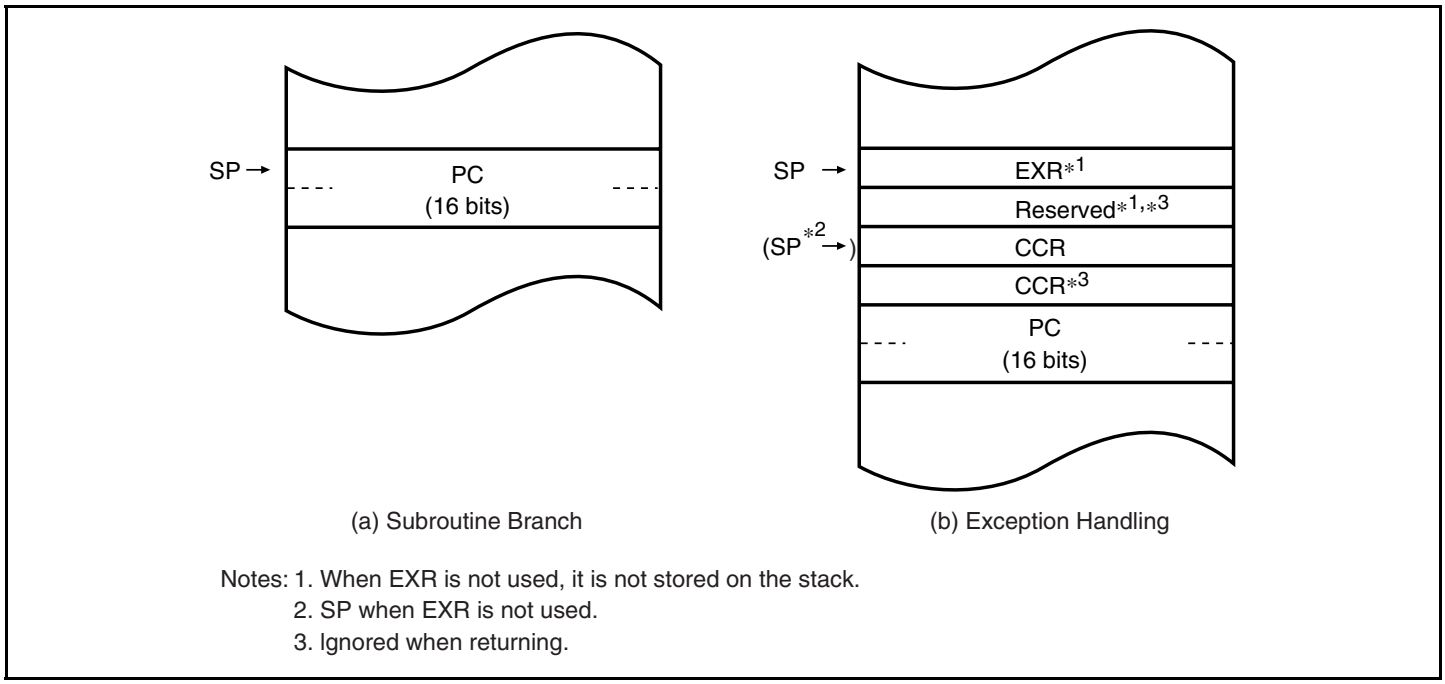


Figure 2.2 Stack Structure in Normal Mode

2.2.2 Advanced Mode

- Address Space

  Linear access is provided to a 16-Mbyte maximum address space.
- Extended Registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers or address registers.
- Instruction Set

  All instructions and addressing modes can be used.

- Exception Vector Table and Memory Indirect Branch Addresses

  In advanced mode the top area starting at H'00000000 is allocated to the exception vector table in units of 32 bits. In each 32 bits, the upper 8 bits are ignored and a branch address is stored in the lower 24 bits (figure 2.3). For details of the exception vector table, see section 4, Exception Handling.

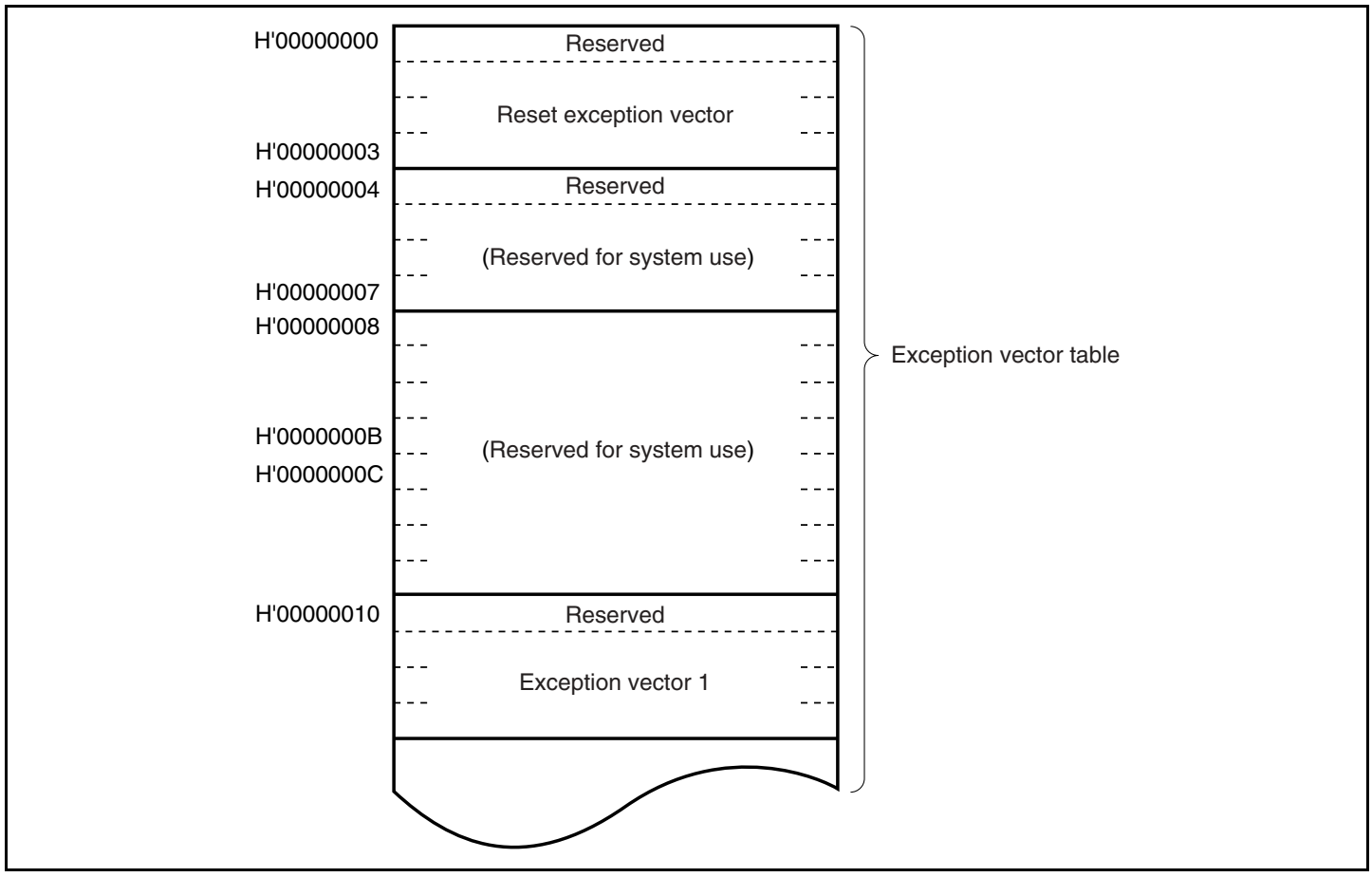


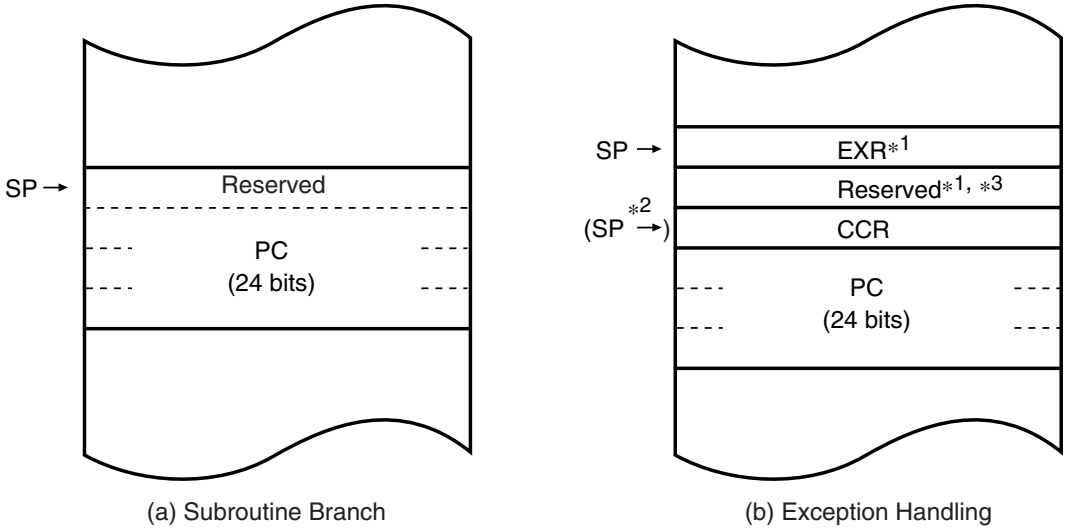
Figure 2.3 Exception Vector Table (Advanced Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address.

In advanced mode the operand is a 32-bit longword operand, providing a 32-bit branch address. The upper 8 bits of these 32 bits are a reserved area that is regarded as H'00. Branch addresses can be stored in the area from H'00000000 to H'000000FF. Note that the first part of this range is also used for the exception vector table.

- Stack Structure

  In advanced mode, when the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.4. EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 4, Exception Handling.



(a) Subroutine Branch

(b) Exception Handling

Notes: 1. When EXR is not used, it is not stored on the stack.
2. SP when EXR is not used.
3. Ignored when returning.

Figure 2.4 Stack Structure in Advanced Mode

2.3 Address Space

Figure 2.5 shows a memory map of the H8S/2600 CPU. The H8S/2600 CPU provides linear access to a maximum 64-kbyte address space in normal mode, and a maximum 16-Mbyte (architecturally 4-Gbyte) address space in advanced mode. The usable modes and address spaces differ depending on the product. For details on each product, refer to section 3, MCU Operating Modes.

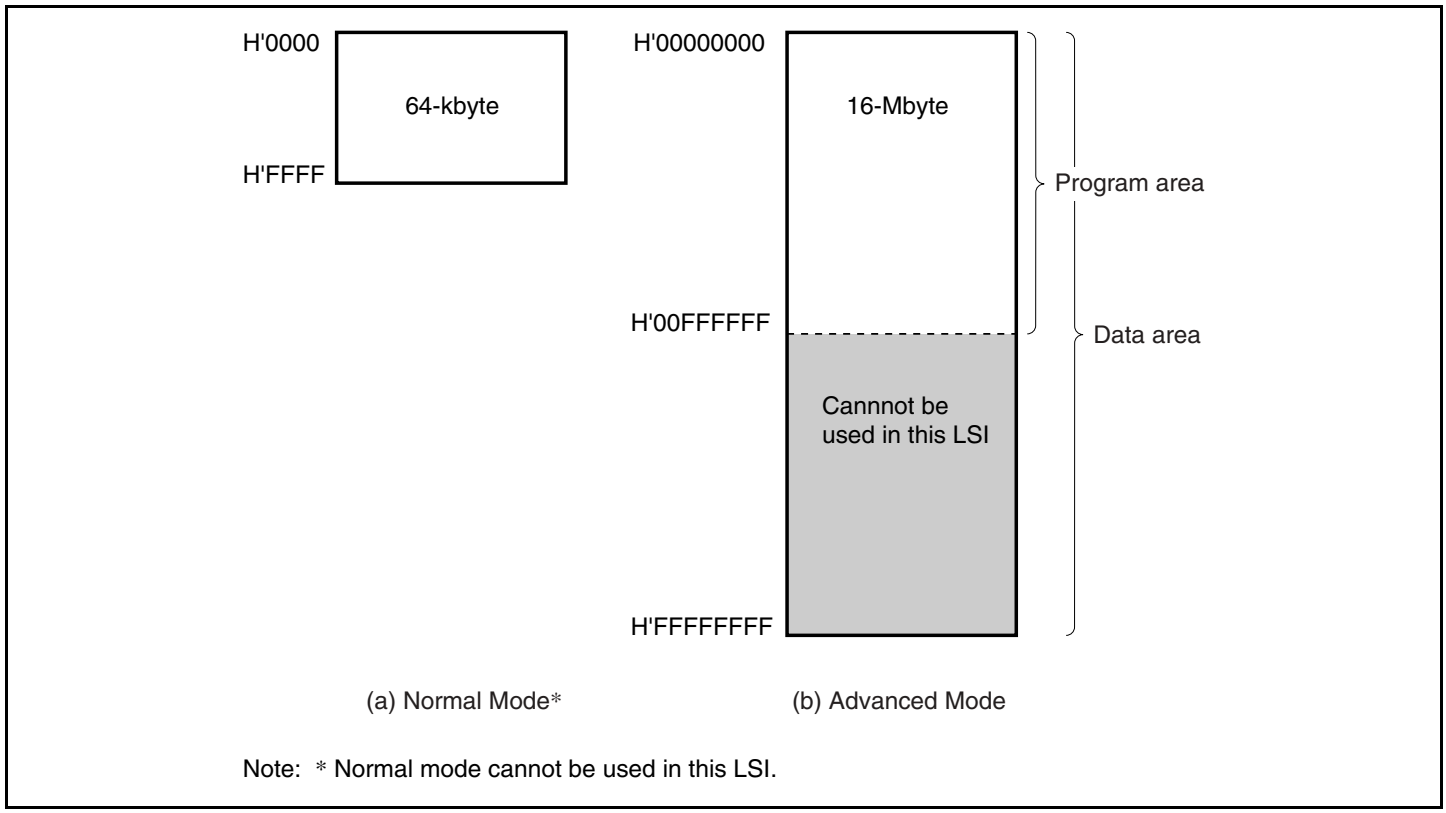


Figure 2.5 Memory Map

Note: Normal mode is not available in this LSI.

2.4 Registers

The H8S/2600 CPU has the internal registers shown in figure 2.6. There are two types of registers: general registers and control registers. Control registers are a 24-bit program counter (PC), an 8-bit extended register (EXR), an 8-bit condition code register (CCR), and a 64-bit multiply-accumulate register (MAC).

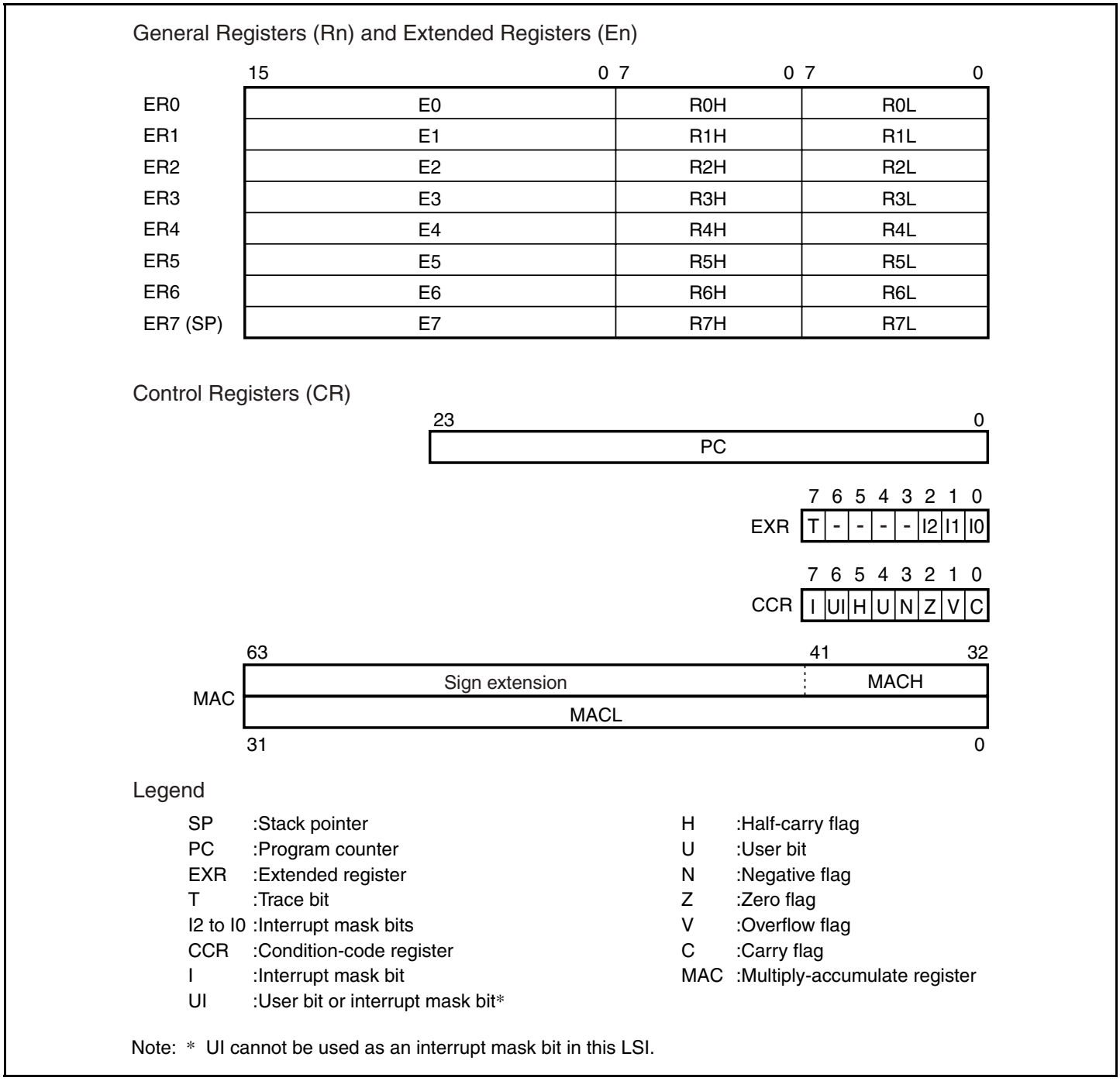


Figure 2.6 CPU Registers

2.4.1 General Registers

The H8S/2600 CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. Figure 2.7 illustrates the usage of the general registers. When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

The ER registers divide into 16-bit general registers designated by the letters E (E0 to E7) and R (R0 to R7). These registers are functionally equivalent, providing a maximum sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

The R registers divide into 8-bit general registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum sixteen 8-bit registers.

The usage of each register can be selected independently.

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.8 shows the stack.

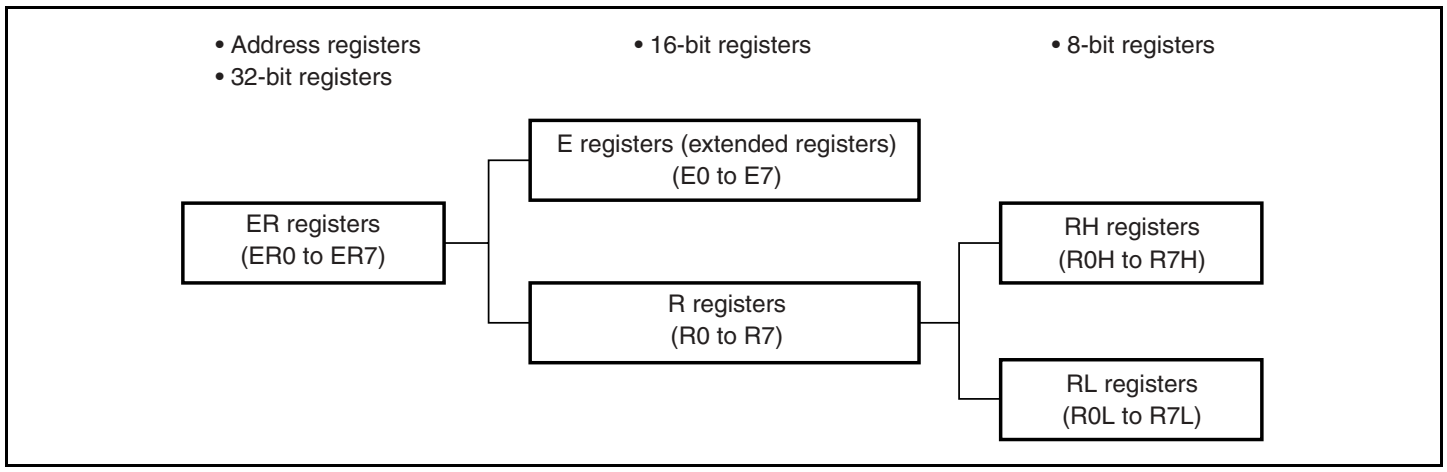


Figure 2.7 Usage of General Registers

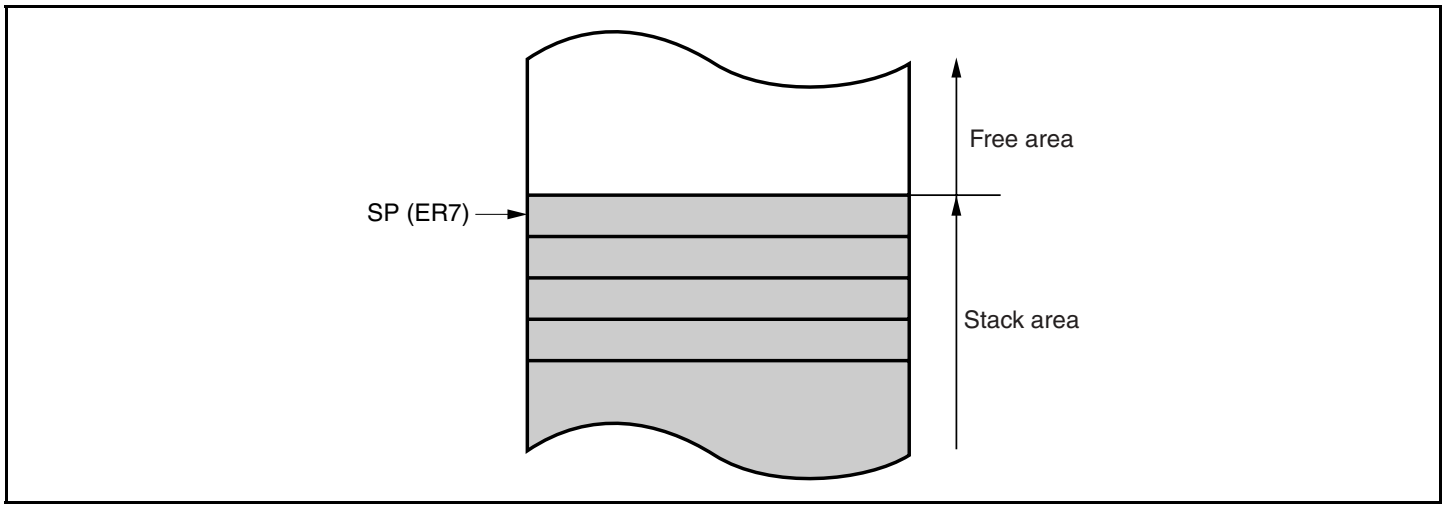


Figure 2.8 Stack

2.4.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (When an instruction is fetched, the least significant PC bit is regarded as 0.)

2.4.3 Extended Register (EXR)

EXR is an 8-bit register that can be manipulated by the LDC, STC, ANDC, ORC, and XORC instructions. When these instructions except for the STC instruction is executed, all interrupts including NMI will be masked for three states after execution is completed.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit When this bit is set to 1, a trace exception is started each time an instruction is executed. When this bit is cleared to 0, instructions are executed in sequence.
6 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	I2	1	R/W	These bits designate the interrupt mask level (0 to 7). For details, refer to section 5, Interrupt Controller.
1	I1	1	R/W	
0	I0	1	R/W	

2.4.4 Condition-Code Register (CCR)

This 8-bit register contains internal CPU status information, including an interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags.

Bit	Bit Name	Initial Value	R/W	Description
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, and cleared to 0 otherwise.
0	C	Undefined	R/W	Carry Flag Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by: • Add instructions, to indicate a carry • Subtract instructions, to indicate a borrow • Shift and rotate instructions, to indicate a carry The carry flag is also used as a bit accumulator by bit manipulation instructions.

2.4.5 Multiply-Accumulate Register (MAC)

This 64-bit register stores the results of multiply-and-accumulate operations. It consists of two 32-bit registers denoted MACH and MACL. The lower 10 bits of MACH are valid; the upper bits are a sign extension.

2.4.6 Initial Values of CPU Internal Registers

When the reset exception handling loads the start address from the vector address, PC is initialized, the T bit in EXR is cleared to 0, and the I bits in EXR and CCR are set to 1. However, the general registers and the other CCR bits are not initialized. The initial value of SP (ER7) is undefined. SP should therefore be initialized by using the MOV.L instruction immediately after a reset.

2.5 Data Formats

The H8S/2600 CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n (n = 0, 1, 2, …, 7) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

2.5.1 General Register Data Formats

Figure 2.9 shows the data formats in general registers.

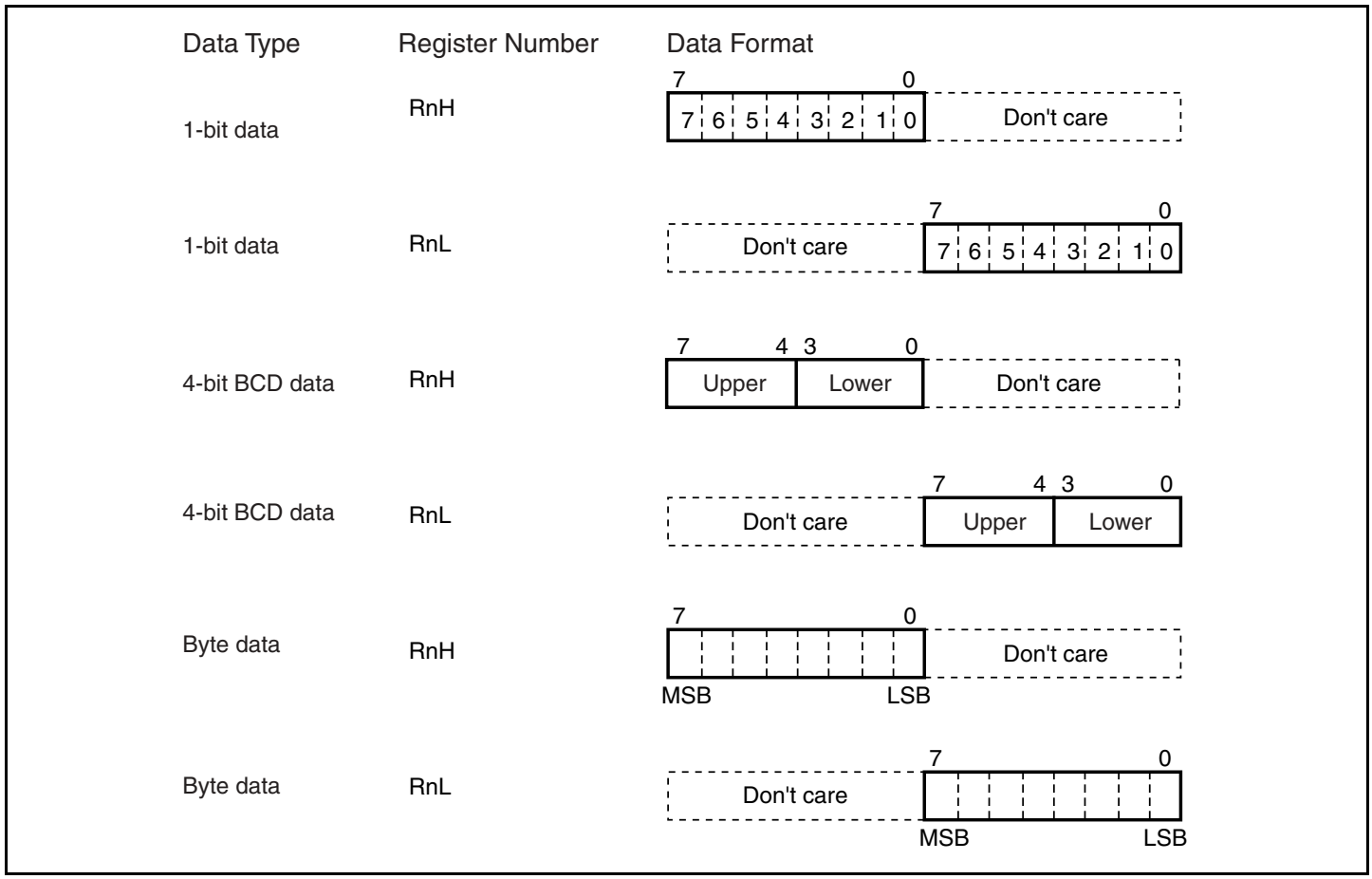


Figure 2.9 General Register Data Formats (1)

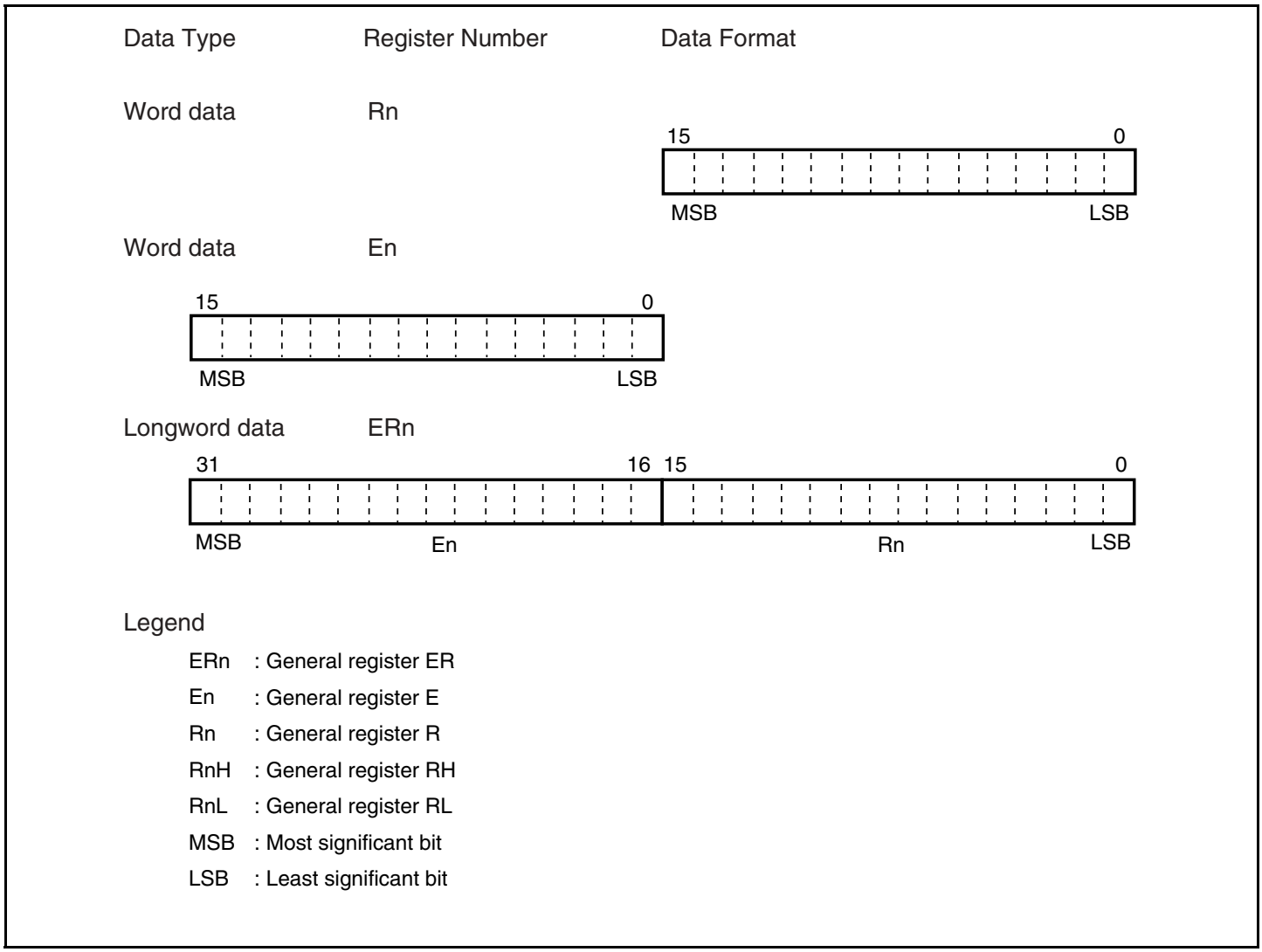


Figure 2.9 General Register Data Formats (2)

2.5.2 Memory Data Formats

Figure 2.10 shows the data formats in memory. The H8S/2600 CPU can access word data and longword data in memory, but word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

When ER7 is used as an address register to access the stack, the operand size should be word size or longword size.

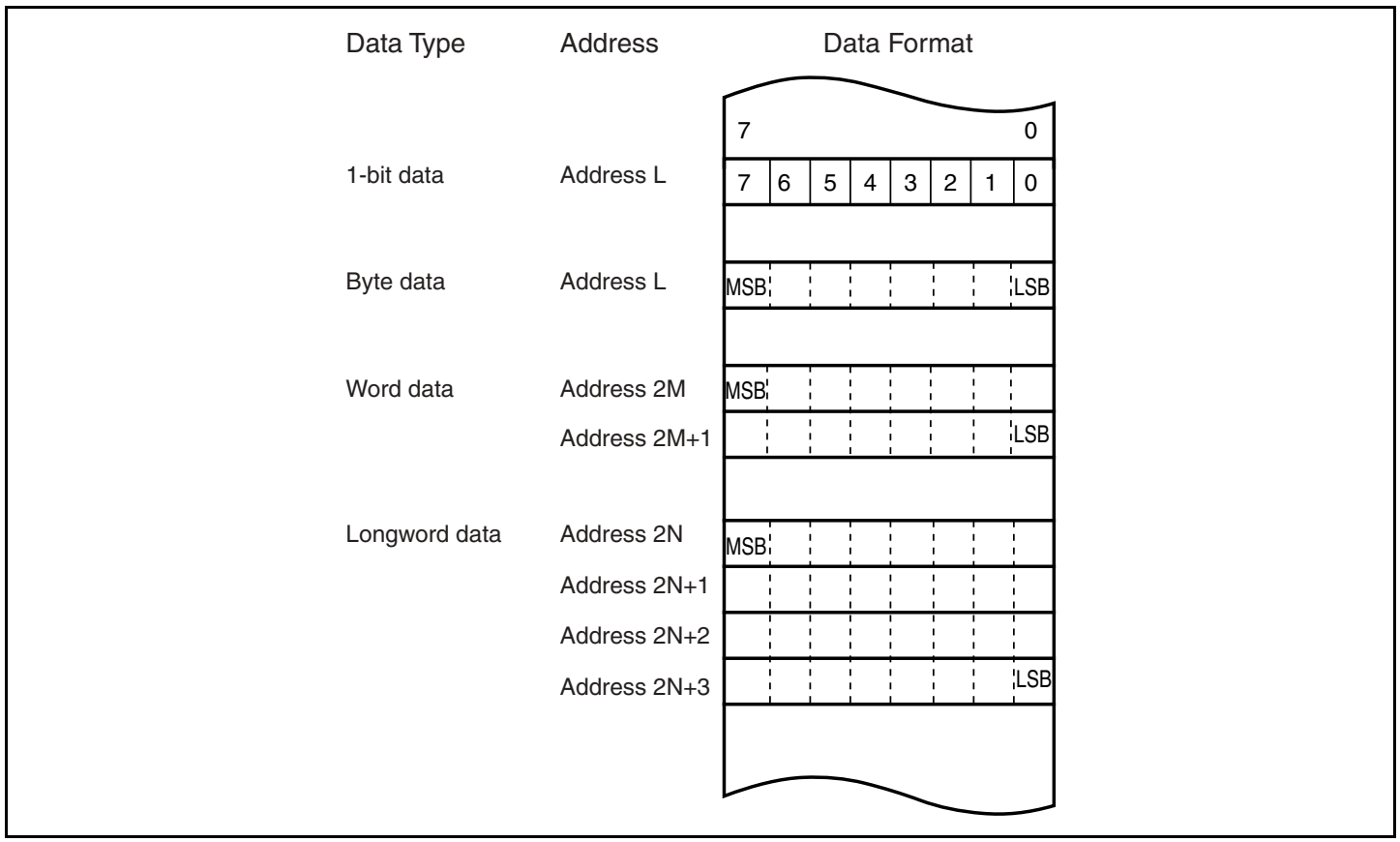


Figure 2.10 Memory Data Formats

2.6 Instruction Set

The H8S/2600 CPU has 69 types of instructions. The instructions are classified by function in table 2.1.

Table 2.1 Instruction Classification

Function	Instructions	Size	Types
Data transfer	MOV	B/W/L	5
	POP*[1], PUSH*[1]	W/L	
	LDM, STM	L	
	MOVFPE*[3], MOVTPE*[3]	B	
Arithmetic operations	ADD, SUB, CMP, NEG	B/W/L	23
	ADDX, SUBX, DAA, DAS	B	
	INC, DEC	B/W/L	
	ADDS, SUBS	L	
	MULXU, DIVXU, MULXS, DIVXS	B/W	
	EXTU, EXTS	W/L	
	TAS*[4]	B	
	MAC, LDMAC, STMAC, CLRMAC	—	
Logic operations	AND, OR, XOR, NOT	B/W/L	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	B/W/L	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	Bcc*[2], JMP, BSR, JSR, RTS	—	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—	9
Block data transfer	EEPMOV	—	1

Total: 69

Notes: B: byte size; W: word size; L: longword size.

1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W Rn, @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.
2. Bcc is the general name for conditional branch instructions.
3. Cannot be used in this LSI.
4. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

2.6.1 Table of Instructions Classified by Function

Tables 2.3 to 2.10 summarize the instructions in each functional category. The notation used in tables 2.3 to 2.10 is defined below.

Table 2.2 Operation Notation

Symbol	Description
Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
MAC	Multiply-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Move
¬	NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

Table 2.3 Data Transfer Instructions

Instruction	Size*	Function
MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
MOVFPE	B	Cannot be used in this LSI.
MOVTPE	B	Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn.
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn, @–SP.
LDM	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
STM	L	Rn (register list) → @–SP Pushes two or more general registers onto the stack.

Note: Size refers to the operand size.
B: Byte
W: Word
L: Longword

Table 2.4 Arithmetic Operations Instructions (1)

Instruction	Size*	Function
ADD SUB	B/W/L	Rd ± Rs → Rd, Rd ± #IMM → Rd Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register. (Immediate byte data cannot be subtracted from byte data in a general register. Use the SUBX or ADD instruction.)
ADDX SUBX	B	Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd Performs addition or subtraction with carry or borrow on byte data in two general registers, or on immediate data and data in a general register.
INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.)
ADDS SUBS	L	Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → Rd Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register.
DAA DAS	B	Rd (decimal adjust) → Rd Decimal-adjusts an addition or subtraction result in a general register by referring to the CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.

Note: Size refers to the operand size.
B: Byte
W: Word
L: Longword

Table 2.4 Arithmetic Operations Instructions (2)

Instruction	Size*[1]	Function
DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
CMP	B/W/L	Rd – Rs, Rd – #IMM Compares data in a general register with data in another general register or with immediate data, and sets CCR bits according to the result.
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data in a general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by padding with zeros on the left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by extending the sign bit.
TAS*[2]	B	@ERd – 0, 1 → (<bit 7> of @ERd) Tests memory contents, and sets the most significant bit (bit 7) to 1.
MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory contents and adds the result to the multiply-accumulate register. The following operations can be performed: 16 bits × 16 bits + 32 bits → 32 bits, saturating 16 bits × 16 bits + 42 bits → 42 bits, non-saturating
CLRMAC	—	0 → MAC Clears the multiply-accumulate register to zero.
LDMAC STMAC	L	Rs → MAC, MAC → Rd Transfers data between a general register and a multiply-accumulate register.

Note: 1. Size refers to the operand size.
B: Byte
W: Word
L: Longword
2. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

Table 2.5 Logic Operations Instructions

Instruction	Size*	Function
AND	B/W/L	Rd ∧ Rs → Rd, Rd ∧ #IMM → Rd Performs a logical AND operation on a general register and another general register or immediate data.
OR	B/W/L	Rd ∨ Rs → Rd, Rd ∨ #IMM → Rd Performs a logical OR operation on a general register and another general register or immediate data.
XOR	B/W/L	Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd Performs a logical exclusive OR operation on a general register and another general register or immediate data.
NOT	B/W/L	¬ (Rd) → (Rd) Takes the one's complement (logical complement) of general register contents.

Note: Size refers to the operand size.
B: Byte
W: Word
L: Longword

Table 2.6 Shift Instructions

Instruction	Size*	Function
SHAL SHAR	B/W/L	Rd (shift) → Rd Performs an arithmetic shift on general register contents. 1-bit or 2-bit shift is possible.
SHLL SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register contents. 1-bit or 2-bit shift is possible.
ROTL ROTR	B/W/L	Rd (rotate) → Rd Rotates general register contents. 1-bit or 2-bit rotation is possible.
ROTXL ROTXR	B/W/L	Rd (rotate) → Rd Rotates general register contents through the carry flag. 1-bit or 2-bit rotation is possible.

Note: Size refers to the operand size.
B: Byte
W: Word
L: Longword

Table 2.7 Bit Manipulation Instructions (1)

Instruction	Size*	Function
BSET	B	1 → (<bit-No.> of <EAd>) Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BCLR	B	0 → (<bit-No.> of <EAd>) Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BNOT	B	¬ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIAND	B	C ∧ ¬ (<bit-No.> of <EAd>) → C ANDs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIOR	B	C ∨ ¬ (<bit-No.> of <EAd>) → C ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

Note: Size refers to the operand size.
B: Byte

Table 2.7 Bit Manipulation Instructions (2)

Instruction	Size*[1]	Function
BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C Exclusive-ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIXOR	B	C ⊕ ¬ (<bit-No.> of <EAd>) → C Exclusive-ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BLD	B	(<bit-No.> of <EAd>) → C Transfers a specified bit in a general register or memory operand to the carry flag.
BILD	B	¬ (<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in a general register or memory operand to the carry flag. The bit number is specified by 3-bit immediate data.
BST	B	C → (<bit-No.> of <EAd>) Transfers the carry flag value to a specified bit in a general register or memory operand.
BIST	B	¬ C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data.

Note: Size refers to the operand size.
B: Byte

Table 2.8 Branch Instructions

Instruction	Size	Function
Bcc	—	Branches to a specified address if a specified condition is true. The branching conditions are listed below.

Mnemonic	Description	Condition
BRA (BT)	Always (true)	Always
BRN (BF)	Never (false)	Never
BHI	High	$C \vee Z = 0$
BLS	Low or same	$C \vee Z = 1$
BCC (BHS)	Carry clear (high or same)	$C = 0$
BCS (BLO)	Carry set (low)	$C = 1$
BNE	Not equal	$Z = 0$
BEQ	Equal	$Z = 1$
BVC	Overflow clear	$V = 0$
BVS	Overflow set	$V = 1$
BPL	Plus	$N = 0$
BMI	Minus	$N = 1$
BGE	Greater or equal	$N \oplus V = 0$
BLT	Less than	$N \oplus V = 1$
BGT	Greater than	$Z \vee (N \oplus V) = 0$
BLE	Less or equal	$Z \vee (N \oplus V) = 1$

JMP	—	Branches unconditionally to a specified address.
BSR	—	Branches to a subroutine at a specified address.
JSR	—	Branches to a subroutine at a specified address.
RTS	—	Returns from a subroutine

Table 2.9 System Control Instructions

Instruction	Size*	Function
TRAPA	—	Starts trap-instruction exception handling.
RTE	—	Returns from an exception-handling routine.
SLEEP	—	Causes a transition to a power-down state.
LDC	B/W	(EAs) → CCR, (EAs) → EXR Moves the contents of a general register or memory, or immediate data to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memory. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically exclusive-ORs the CCR or EXR contents with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

Note: Size refers to the operand size.
B: Byte
W: Word

Table 2.10 Block Data Transfer Instructions

Instruction	Size	Function
EEPMOV.B	—	if R4L ≠ 0 then Repeat @ER5+ → @ER6+ R4L–1 → R4L Until R4L = 0 else next;
EEPMOV.W	—	if R4 ≠ 0 then Repeat @ER5+ → @ER6+ R4–1 → R4 Until R4 = 0 else next;
		Transfers a data block. Starting from the address set in ER5, transfers data for the number of bytes set in R4L or R4 to the address location set

- Operation Field

  Indicates the function of the instruction, the addressing mode, and the operation to be carried out on the operand. The operation field always includes the first four bits of the instruction. Some instructions have two operation fields.
- Register Field

  Specifies a general register. Address registers are specified by 3 bits, data registers by 3 bits or 4 bits. Some instructions have two register fields. Some have no register field.
- Effective Address Extension

  8, 16, or 32 bits specifying immediate data, an absolute address, or a displacement.
- Condition Field

  Specifies the branching condition of Bcc instructions.

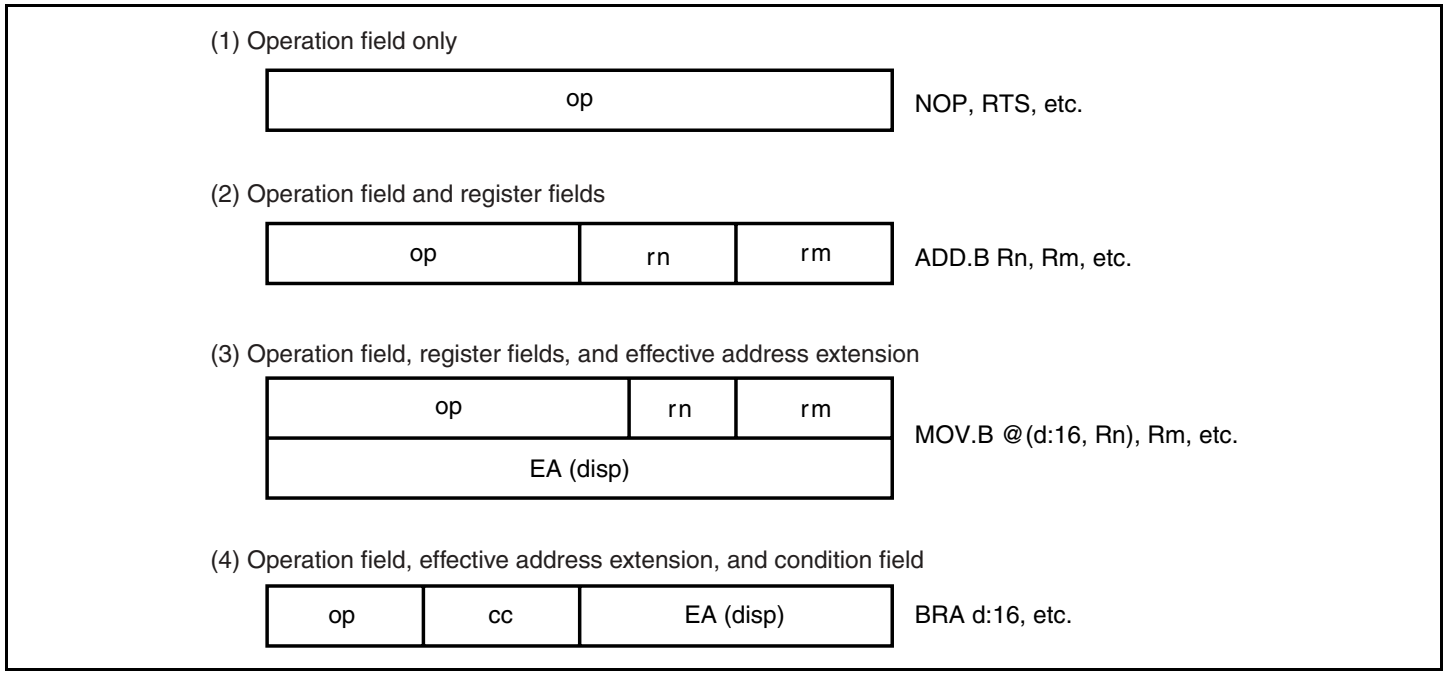


Figure 2.11 Instruction Formats (Examples)

2.7 Addressing Modes and Effective Address Calculation

The H8S/2600 CPU supports the eight addressing modes listed in table 2.11. The usable address modes are different in each instruction.

Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

Table 2.11 Addressing Modes

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:16,ERn)/@(d:32,ERn)
4	Register indirect with post-increment Register indirect with pre-decrement	@ERn+ @–ERn
5	Absolute address	@aa:8/@aa:16/@aa:24/@aa:32
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

2.7.1 Register Direct—Rn

The register field of the instruction code specifies an 8-, 16-, or 32-bit general register containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

2.7.2 Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn) which contains the address of the operand on memory. If the address is a program instruction address, the lower 24 bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

2.7.3 Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn)

A 16-bit or 32-bit displacement contained in the instruction is added to an address register (ERn) specified by the register field of the instruction code, and the sum gives the address of a memory operand. A 16-bit displacement is sign-extended when added.

2.7.4 Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn

Register indirect with post-increment—@ERn+: The register field of the instruction code specifies an address register (ERn) which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is stored in the address register. The value added is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer instruction. For word or longword transfer instruction, the register value should be even.

Register indirect with pre-decrement—@-ERn: The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the result becomes the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer instruction. For word or longword transfer instruction, the register value should be even.

2.7.5 Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32

The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24), or 32 bits long (@aa:32). Table 2.12 indicates the accessible absolute address ranges.

To access data, the absolute address should be 8 bits (@aa:8), 16 bits (@aa:16), or 32 bits (@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address, the upper 16 bits are a sign extension. A 32-bit absolute address can access the entire address space.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The upper 8 bits are all assumed to be 0 (H'00).

Table 2.12 Absolute Address Access Ranges

Absolute Address		Normal Mode*	Advanced Mode
Data address	8 bits (@aa:8)	H'FF00 to H'FFFF	H'FFFF00 to H'FFFFFF
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF
	32 bits (@aa:32)		H'000000 to H'FFFFFF
Program instruction address	24 bits (@aa:24)		

Note: Not available in this LSI.

2.7.6 Immediate—#xx:8, #xx:16, or #xx:32

The instruction code contains 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data as an operand.

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some bit manipulation instructions contain 3-bit immediate data in the instruction code, specifying a bit number. The TRAPA instruction contains 2-bit immediate data in its instruction code, specifying a vector address.

2.7.7 Program-Counter Relative—@(d:8, PC) or @(d:16, PC)

This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction code is sign-extended and added to the 24-bit PC contents to generate a branch address. Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00). The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

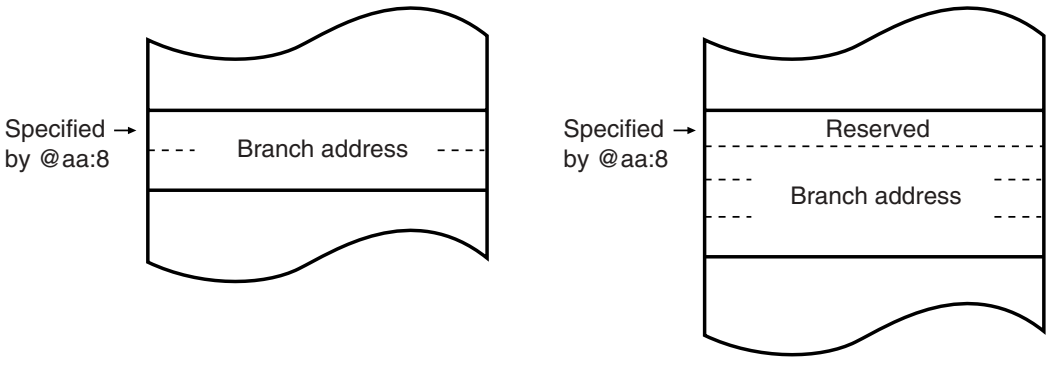
2.7.8 Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF in normal mode, H'000000 to H'0000FF in advanced mode).

In normal mode the memory operand is a word operand and the branch address is 16 bits long. In advanced mode the memory operand is a longword operand, the first byte of which is assumed to be all 0 (H'00). Note that the first part of the address range is also the exception vector area. For further details, refer to section 4, Exception Handling.

If an odd address is specified in word or longword memory access, or as a branch address, the least significant bit is regarded as 0, causing data to be accessed or instruction code to be fetched at the address preceding the specified address. (For further information, see section 2.5.2, Memory Data Formats.)

Note: Normal mode is not available in this LSI.



2.7.9 Effective Address Calculation

Table 2.13 indicates how effective addresses are calculated in each addressing mode. In normal mode the upper 8 bits of the effective address are ignored in order to generate a 16-bit address.

Note: Normal mode is not available in this LSI.

Table 2.13 Effective Address Calculation (1)

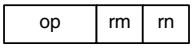
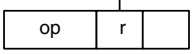
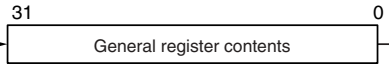
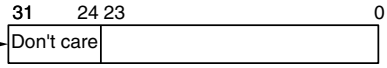
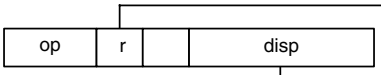
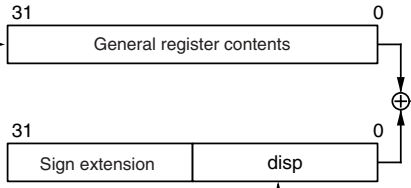
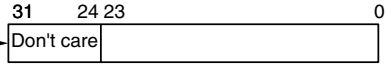
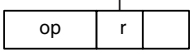
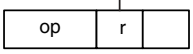
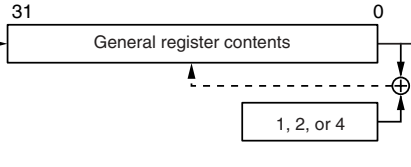
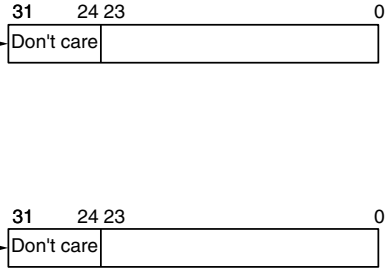
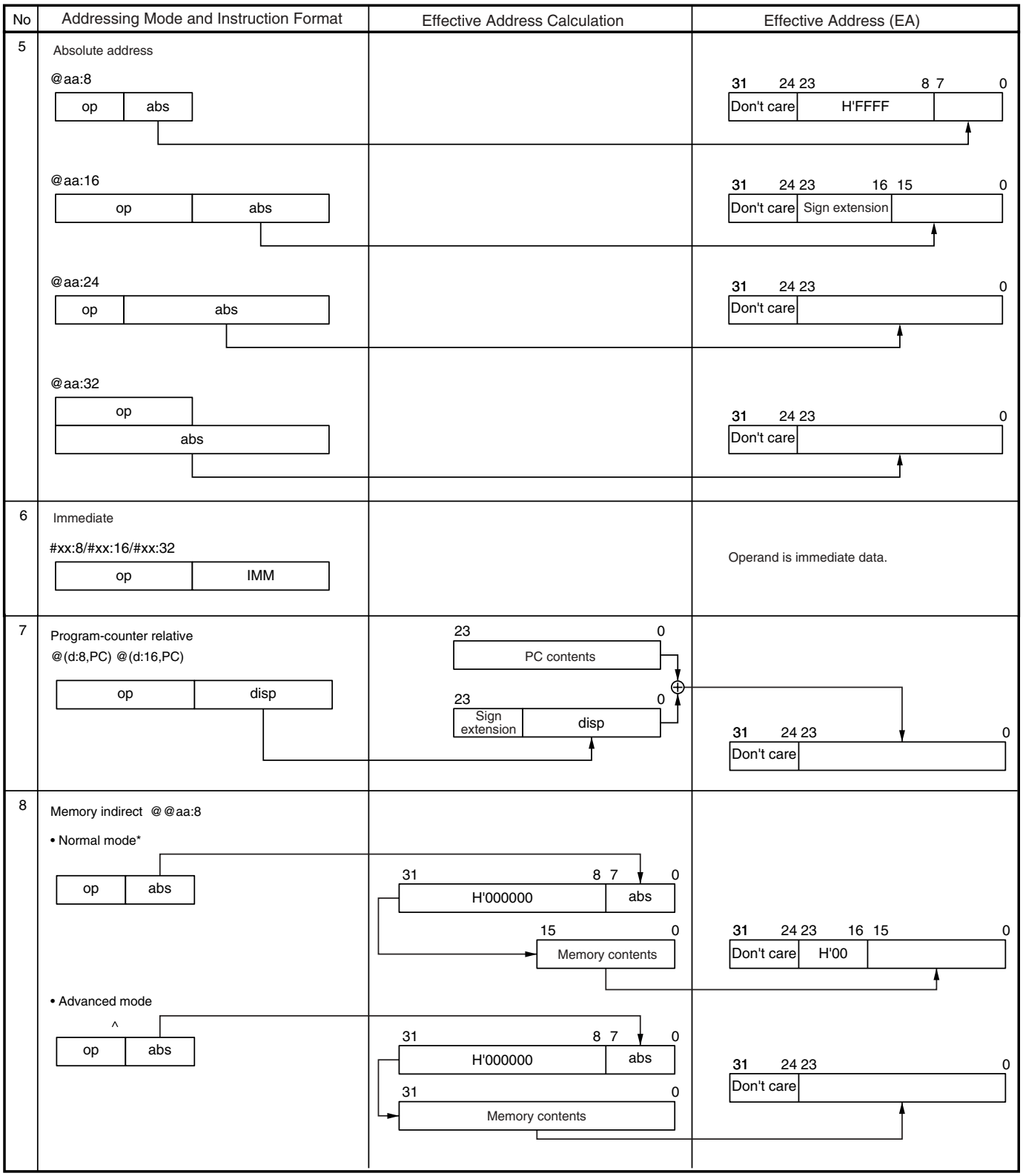
No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct (Rn) 		Operand is general register contents.
2	Register indirect (@ERn) 		
3	Register indirect with displacement @(d:16,ERn) or @(d:32,ERn) 		
4	Register indirect with post-increment or pre-decrement •Register indirect with post-increment @ERn+  •Register indirect with pre-decrement @-ERn 	 Operand Size / Offset Byte / 1 Word / 2 Longword / 4	

Table 2.13 Effective Address Calculation (2)



Note: * Normal mode is not available in this LSI.

2.8 Processing States

The H8S/2600 CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and program stop state. Figure 2.13 indicates the state transitions.

- Reset State

  The CPU and on-chip peripheral modules are all initialized and stop. When the RES input goes low, all current processing stops and the CPU enters the reset state. All interrupts are masked in the reset state. Reset exception handling starts when the RES signal changes from low to high. For details, refer to section 4, Exception Handling.

  The reset state can also be entered by a watchdog timer overflow.

- Exception-Handling State

  The exception-handling state is a transient state that occurs when the CPU alters the normal processing flow due to an exception source, such as, a reset, trace, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception vector table and branches to that address. For further details, refer to section 4, Exception Handling.

- Program Execution State

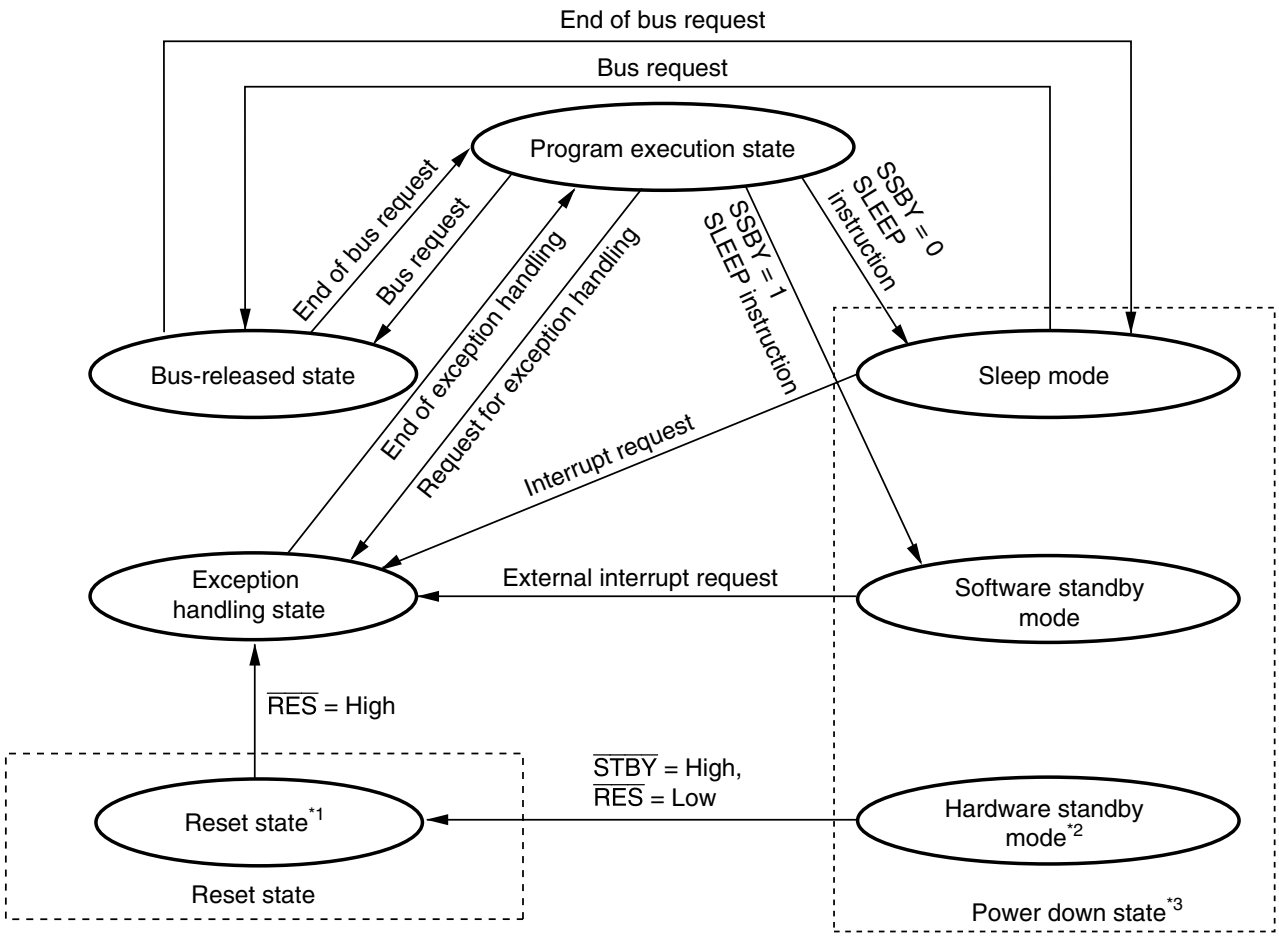
  In this state the CPU executes program instructions in sequence.

- Bus-Released State

  In a product which has a bus master other than the CPU, such as a direct memory access controller (DMAC) and a data transfer controller (DTC), the bus-released state occurs when the bus has been released in response to a bus request from a bus master other than the CPU. While the bus is released, the CPU halts operations.

- Program stop state

  This is a power-down state in which the CPU stops operating. The program stop state occurs when a SLEEP instruction is executed or the CPU enters hardware standby mode. For further details, refer to section 22, Power-Down Modes.



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{RES}$ goes low. A transition can also be made to the reset state when the watchdog timer overflows.
2. In every state, when the STBY pin becomes low, the hardware standby mode is entered.
3. For details, refer to section 22, Power-Down Modes.

Figure 2.13 State Transitions

2.9 Usage Note

2.9.1 Usage Notes on Bit-wise Operation Instructions

The BSET, BCLR, BNOT, BST, and BIST instructions are used to read data in byte-wise, operate the data in bit-wise, and write the result of the bit-wise operation in bit-wise again. Therefore, special care is necessary to use these instructions for the registers and the ports that include write-only bit.

The BCLR instruction can be used to clear the flags in the internal I/O registers to 0. In this time, if it is obvious that the flag has been set to 1 in the interrupt handler, there is no need to read the flag beforehand.

Section 3 MCU Operating Modes

3.1 Operating Mode Selection

The H8S/2678 Series has twelve operating modes (modes 1, 2, 4 to 7, and 10 to 15). All operating modes are available for the flash memory version. Modes 1, 2, and 4 to 7 are available in the masked ROM version. Modes 1 and 2 are available in the ROMless version.

The H8S/2678R Series has seven operating modes (modes 1 to 7). All operating modes are available for the flash memory version. Modes 1 and 2 are available in the ROMless version.

These modes are determined by the mode pin (MD2 to MD0) setting.

Modes 1, 2, and 4 to 6 are externally expanded modes in which the CPU can access an external memory and peripheral devices. In the externally expanded mode, each area can be switched to 8-bit or 16-bit address space by the bus controller. If one of areas is set to 16-bit address space, the bus mode is 16 bits. If all areas are set to 8-bit address space, the bus mode is 8 bits.

Mode 7 is a single-chip activation externally expanded mode in which the CPU can switch to access an external memory and peripheral devices at the beginning of a program execution.

Modes 3, 10, and 11 are boot modes in which the flash memory can be accessed.

Modes 12 to 15 are user program modes in which the flash memory can be accessed.

For details, refer to section 19, Flash Memory.

Do not change the FWE and MD2 to MD0 pin settings during operation.

Table 3.1 MCU Operating Mode Selection

MCU Operating Mode*[1]	FWE*[2]	MD2	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM	External Data Bus	
								Initial Width	Max. Value
1	0	0	0	1	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
2	0	0	1	0	Advanced	Expanded mode with on-chip ROM disabled	Disabled	8 bits	16 bits
3	—	0	1	1	Advanced	Boot mode	Enabled	—	16 bits
4	0	1	0	0	Advanced	Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
5	0	1	0	1	Advanced	Expanded mode with on-chip ROM enabled	Enabled	16 bits	16 bits
6	0	1	1	0	Advanced	Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7	0	1	1	1	Advanced	Single-chip mode	Enabled	—	16 bits
10	1	0	1	0	Advanced	Boot mode	Enabled	8 bits	16 bits
11	1	0	1	1	Advanced	Boot mode	Enabled	—	16 bits
12	1	1	0	0	Advanced	User program mode	Enabled	8 bits	16 bits
13	1	1	0	1	Advanced	User program mode	Enabled	16 bits	16 bits
14	1	1	1	0	Advanced	User program mode	Enabled	8 bits	16 bits
15	1	1	1	1	Advanced	User program mode	Enabled	—	16 bits

Notes: 1. Modes 1, 2, 4 to 7, and 10 to 15 are supported in the H8S/2678 Series.
Modes 1 to 7 are supported in the H8S/2678R Series.
2. The FWE pin setting is available only in the H8S/2678 Series. The FWE pin is not available in the H8S/2678R Series.

3.2 Register Descriptions

The following registers are related to the operating mode.

- Mode control register (MDCR)
- System control register (SYSCR)

3.2.1 Mode Control Register (MDCR)

MDCR monitors the current operating mode of the H8S/2678 Series chip.

Bit	Bit Name	Initial Value	R/W	Descriptions
7 to 3	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
2	MDS2	—*	R	Mode Select 2 to 0 These bits indicate the input levels at pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to MD2 to MD0. MDS2 to MDS0 are read-only bits and they cannot be written to. The mode pin (MD2 to MD0) input levels are latched into these bits when MDCR is read. These latches are canceled by a reset.
1	MDS1	—*	R	
0	MDS0	—*	R	

Note: Determined by pins MD2 to MD0.

3.2.2 System Control Register (SYSCR)

SYSCR selects saturating or non-saturating calculation for the MAC instruction, controls CPU access to the flash memory control registers (FLMCR1, FLMCR2, EBR1, and EBR2), sets external bus mode, and enables or disables on-chip RAM.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	—	1	R/W	Reserved
6	—	1	R/W	The initial value should not be modified.
5	MACS	0	R/W	MAC Saturation Selects either saturating or non-saturating calculation for the MAC instruction. 0: Non-saturating calculation for MAC instruction 1: Saturating calculation for MAC instruction
4	—	0	R/W	Reserved The initial value should not be modified.
3	FLSHE	0	R/W	Flash Memory Control Register Enable Controls CPU access to the flash memory control registers (FLMCR1, FLMCR2, EBR1, and EBR2). If this bit is set to 1, the flash memory control registers can be read/written to. If this bit is cleared to 0, the flash memory control registers are not selected. At this time, the contents of the flash memory control registers are maintained. This bit should be written to 0 other than flash memory version. 0: Flash memory control registers are not selected for area H'FFFFC8 to H'FFFFCB 1: Flash memory control registers are selected for area H'FFFFC8 to H'FFFFCB
2	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
1	EXPE	—	R/W	External Bus Mode Enable Sets external bus mode. In modes 1, 2, and 4 to 6, this bit is fixed at 1 and cannot be modified. In mode 3* and 7, this bit has an initial value of 0, and can be read and written. Writing of 0 to EXPE when its value is 1 should only be carried out when an external bus cycle is not being executed. 0: External bus disabled 1: External bus enabled
0	RAME	1	R/W	RAM Enable Enables or disables the on-chip RAM. The RAME bit is initialized when the reset status is released. 0: On-chip RAM is disabled 1: On-chip RAM is enabled

Note: Mode 3 is available only in the F-ZTAT version of H8S/2678R Series.

3.3 Operating Mode Descriptions

3.3.1 Mode 1

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled. Ports A, B, and C function as an address bus, ports D and E function as a data bus, and parts of ports F, G, and H carry bus control signals.

The initial bus mode after a reset is 16 bits, with 16-bit access to all areas. However, if 8-bit access is designated for all areas by the bus controller, the bus mode switches to 8 bits.

3.3.2 Mode 2

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled. Ports A, B, and C function as an address bus, ports D and E function as a data bus, and parts of ports F, G, and H carry bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, if 16-bit access is designated for all areas by the bus controller, the bus mode switches to 16 bits and port E functions as a data bus.

3.3.3 Mode 3

This mode is a boot mode of the flash memory. This mode is the same as mode 7, except for accessing to the flash memory. Mode 3 is available only in the flash memory version of the H8S/2678R Series.

3.3.4 Mode 4

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. The program in the on-chip ROM connected to the first half of area 0 is executed.

Ports A, B, and C function as input ports immediately after a reset, but can be set to function as an address bus. For details, see section 10, I/O Ports. Ports D and E function as a data bus, and parts of ports F, G, and H carry bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, if 16-bit access is designated for any area by the bus controller, the bus mode switches to 16 bits and port E functions as a data bus. In the flash memory version, user program mode is entered by setting 1 to the SWE bit of FLMCR1.

3.3.5 Mode 5

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. The program in an external ROM connected to the first half of area 0 is executed.

Ports A, B, and C function as an address bus, ports D and E function as a data bus, and parts of ports F, G and H carry bus control signals.

The initial bus mode after a reset is 16 bits, with 16-bit access to all areas. However, if 8-bit access is designated for any area by the bus controller, the bus mode switches to 8 bits.

In the flash memory version, user program mode is entered by setting 1 to the SWE bit of FLMCR1.

3.3.6 Mode 6

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. The program in an external ROM connected to the first half of area 0 is executed.

Ports A, B, and C function as an address bus, ports D and E function as a data bus, and parts of ports F, G, and H carry bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, if 16-bit access is designated for any area by the bus controller, the bus mode switches to 16 bits and port E functions as a data bus.

In the flash memory version, user program mode is entered by setting 1 to the SWE bit of FLMCR1.

3.3.7 Mode 7

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled, and the chip starts up in single-chip mode. External addresses cannot be used in single-chip mode.

The initial mode after a reset is single-chip mode, with all I/O ports available for use as input/output ports. However, the mode can be switched to externally expanded mode by setting 1 to the EXPE bit of SYSCR and then the external address space is enabled. When externally expanded mode is selected, all areas are initially designated as 16-bit access space. The function of pins in ports A to H is the same as in externally expanded mode with on-chip ROM enabled.

In the flash memory version, user program mode is entered by setting 1 to the SWE bit of FLMCR1.

Table 3.2 Pin Functions in Each Operating Mode

Port		Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7	Mode 10	Mode 11	Mode 12	Mode 13	Mode 14	Mode 15
Port A	PA7 to PA5	P*/A	P*/A	P*/A	P*/A	P*/A	P*/A	P*/A	P*/A	P*/A	P*/A	P*/A	P*/A	P*/A
	PA4 to PA0	A	A			A	A				A		A	
Port B		A	A	P*/A	P*/A	A	A	P*/A	P*/A	P*/A	P*/A	A	A	P*/A
Port C		A	A	P*/A	P*/A	A	A	P*/A	P*/A	P*/A	P*/A	A	A	P*/A
Port D		D	D	P*/D	P*/D	D	D	P*/D	D	P*/D	D	D	D	P*/D
Port E		P/D*	P*/D	P*/D	P*/D	P/D*	P*/D	P*/D	P*/D	P*/D	P*/D	P/D*	P*/D	P*/D
Port F	PF7, PF6	P/C*	P*/C	P*/C	P/C*	P/C*	P/C*	P*/C	P*/C	P*/C	P/C*	P/C*	P/C*	P*/C
	PF5, PF4	C	C		C	C	C		C		C	C	C	
	PF3	P/C*	P/C*		P/C*	P/C*	P/C*		P/C*		P/C*	P/C*	P/C*	
	PF2 to PF0	P*/C	P*/C		P*/C	P*/C	P*/C		P*/C		P*/C	P*/C	P*/C	
Port G	PG6 to PG1	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C
	PG0	P/C*	P/C*		P*/C	P/C*	P/C*		P/C*		P*/C	P/C*	P/C*	
Port H		P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C

Legend: P: I/O port

A: Address bus output

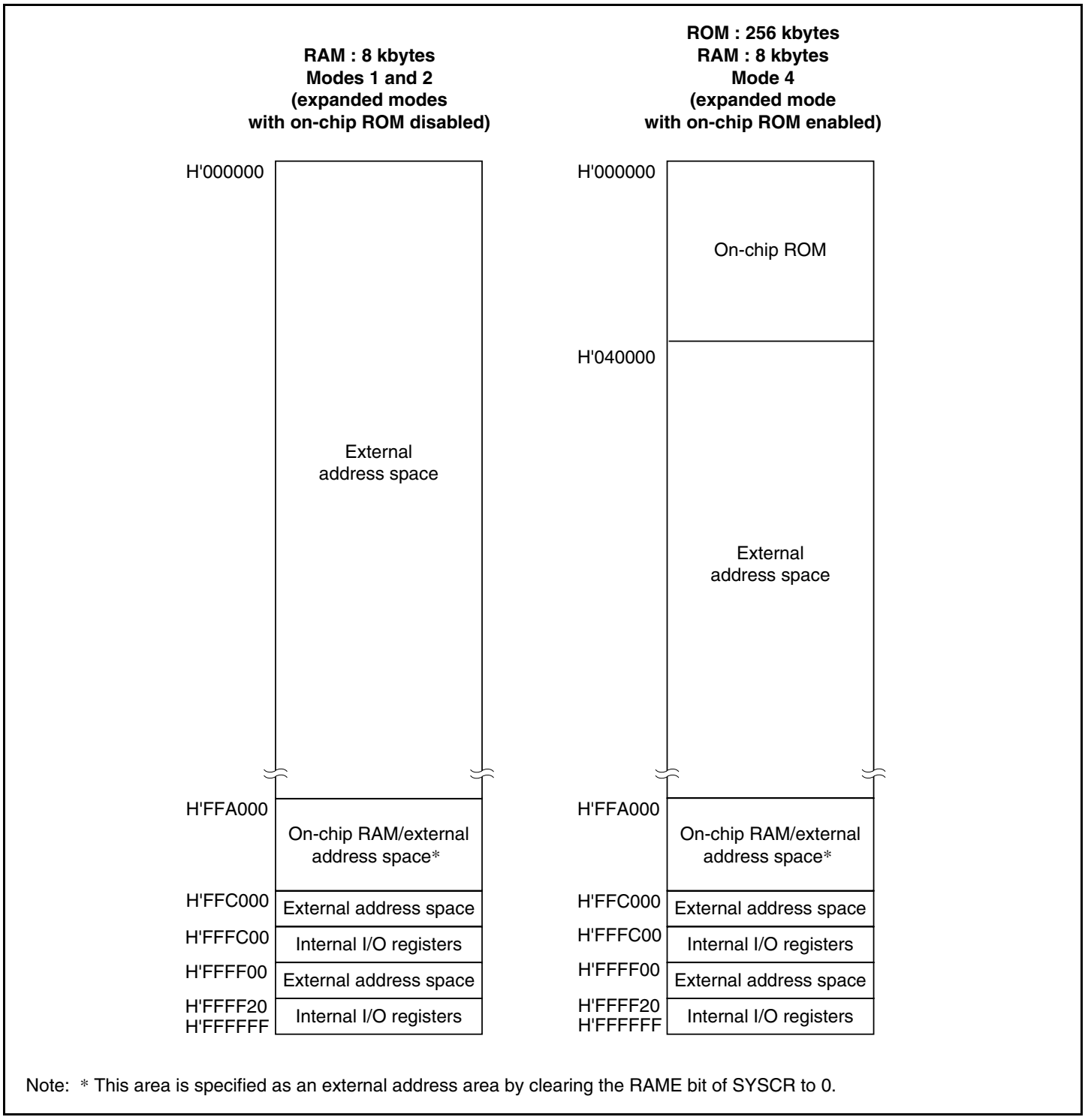
D: Data bus input/output

C: Control signals, clock input/output

Note: After reset

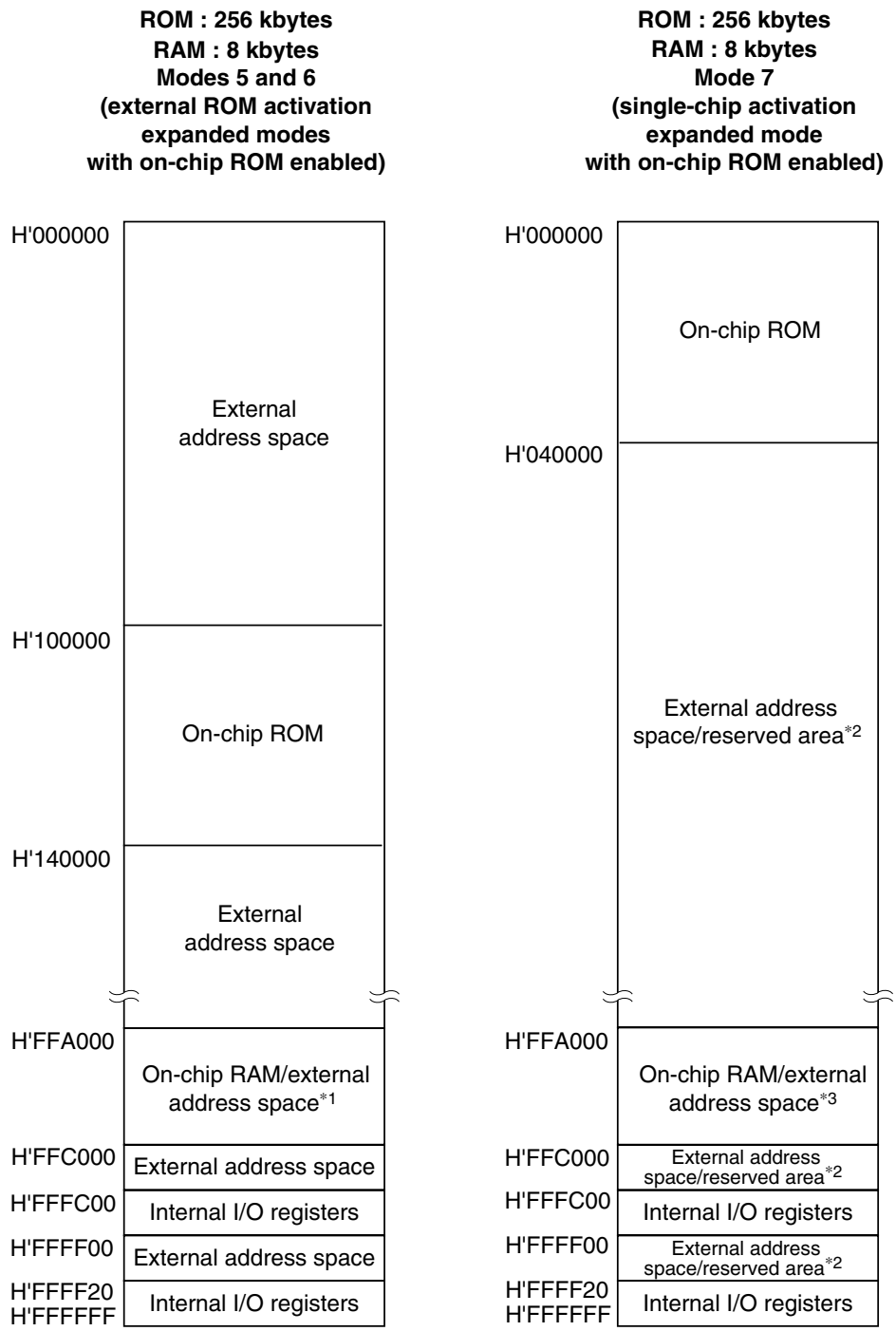
3.4 Memory Map in Each Operating Mode

Figures 3.1 to 3.6 show memory maps for each product.



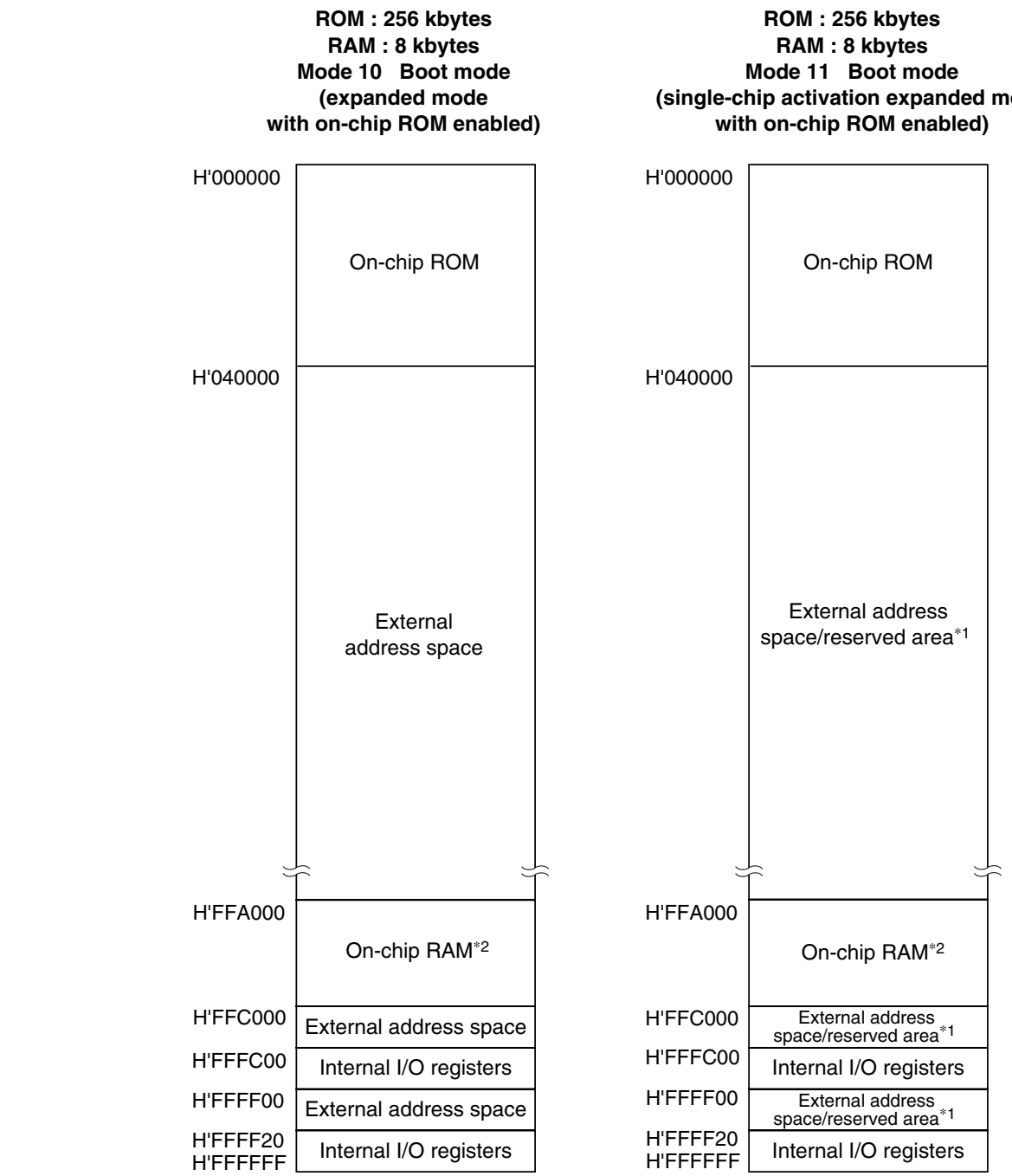
Note: * This area is specified as an external address area by clearing the RAME bit of SYSCR to 0.

Figure 3.1 H8S/2676 Memory Map (1)



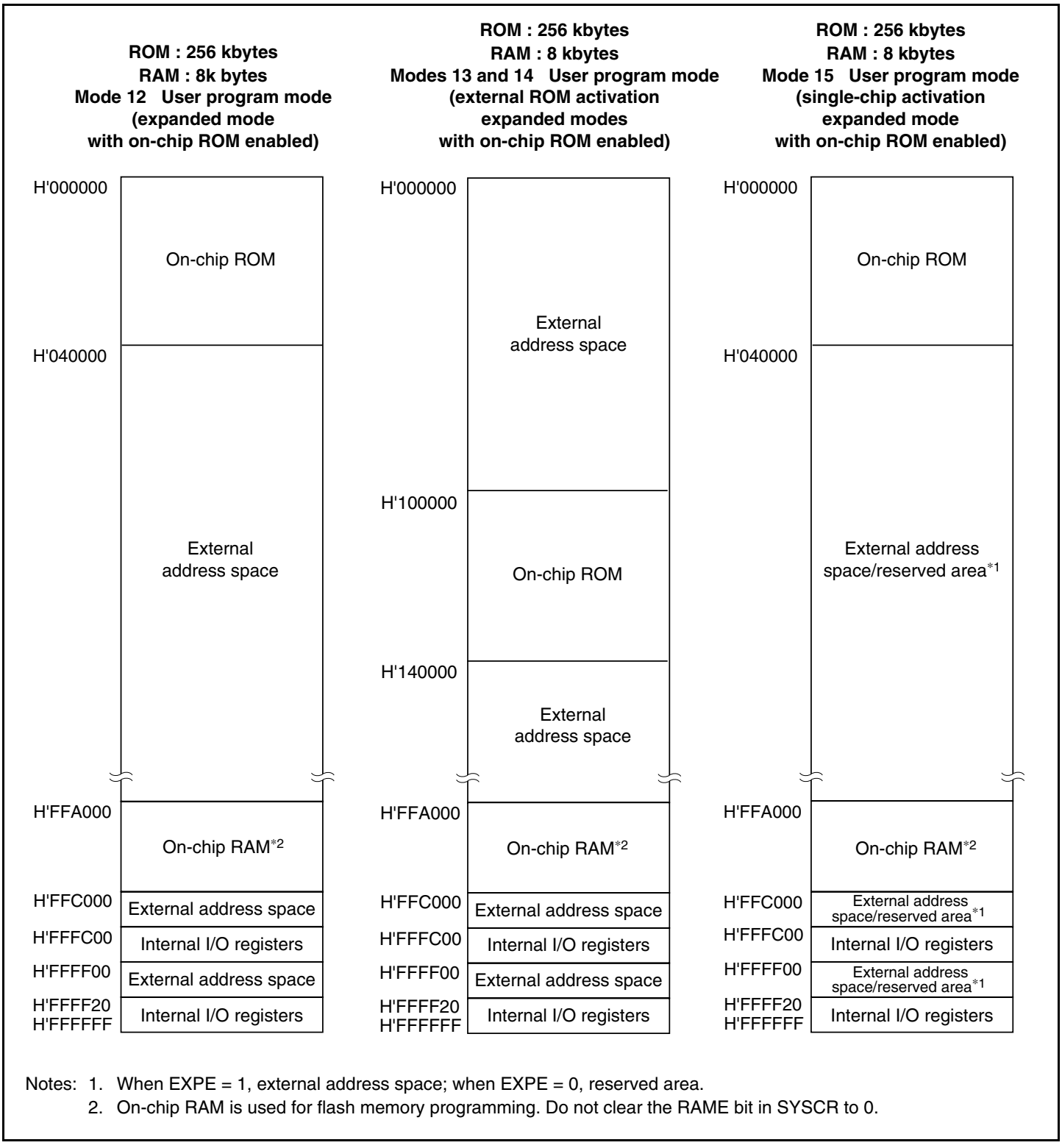
Notes: 1. This area is specified as an external address area by clearing the RAME bit of SYSCR to 0.
2. When EXPE = 1, external address space; when EXPE = 0, reserved area.
3. When EXPE = 1, external address space when RAME = 0, on-chip RAM when RAME = 1.
When EXPE = 0, on-chip RAM area.

Figure 3.1 H8S/2676 Memory Map (2)



Notes: 1. When EXPE = 1, external address space; when EXPE = 0, reserved area.
2. On-chip RAM is used for flash memory programming. Do not clear the RAME bit in SYSCR to 0.

Figure 3.1 H8S/2676 Memory Map (3)



Notes: 1. When EXPE = 1, external address space; when EXPE = 0, reserved area.
2. On-chip RAM is used for flash memory programming. Do not clear the RAME bit in SYSCR to 0.

Figure 3.1 H8S/2676 Memory Map (4)

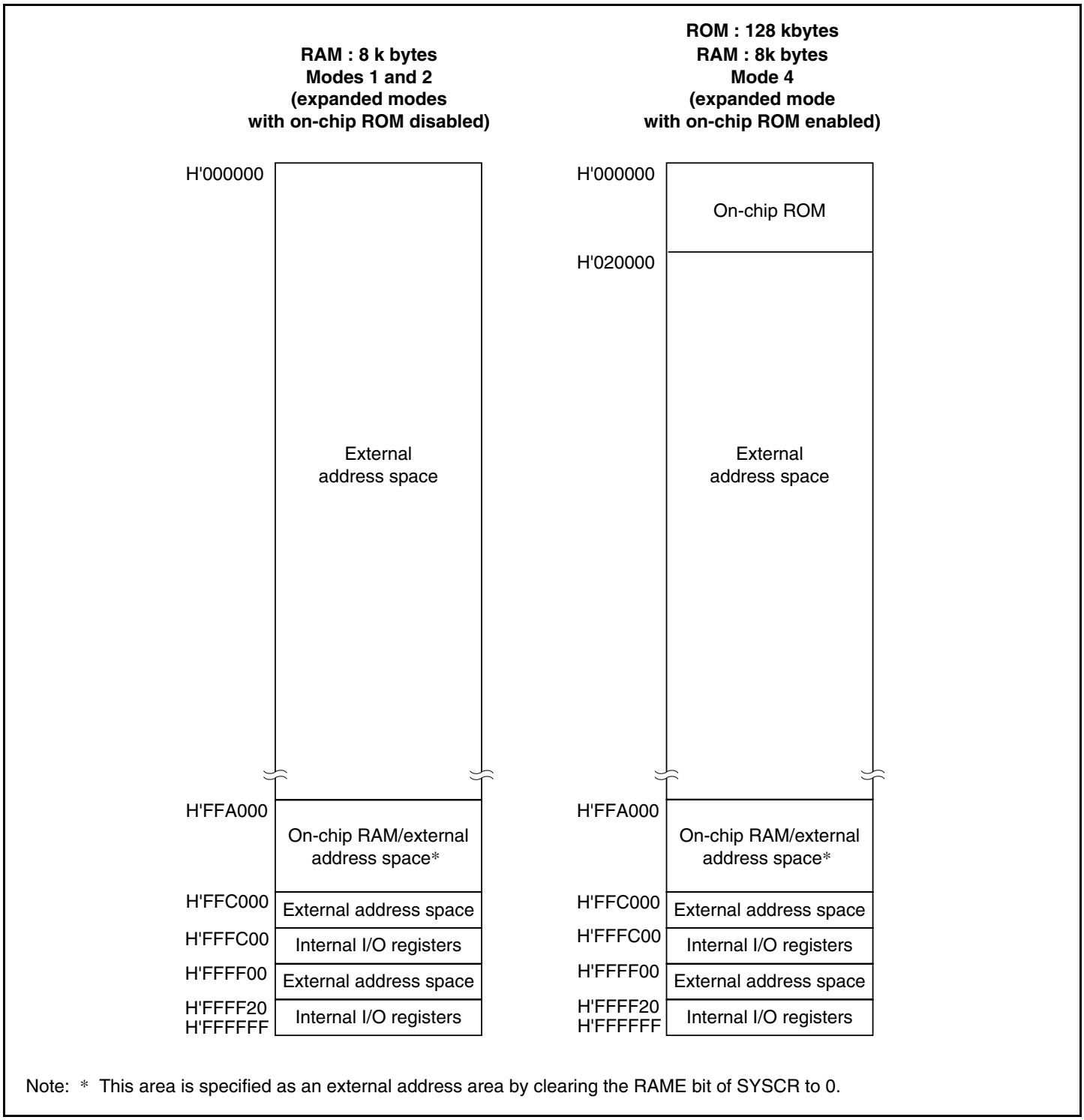
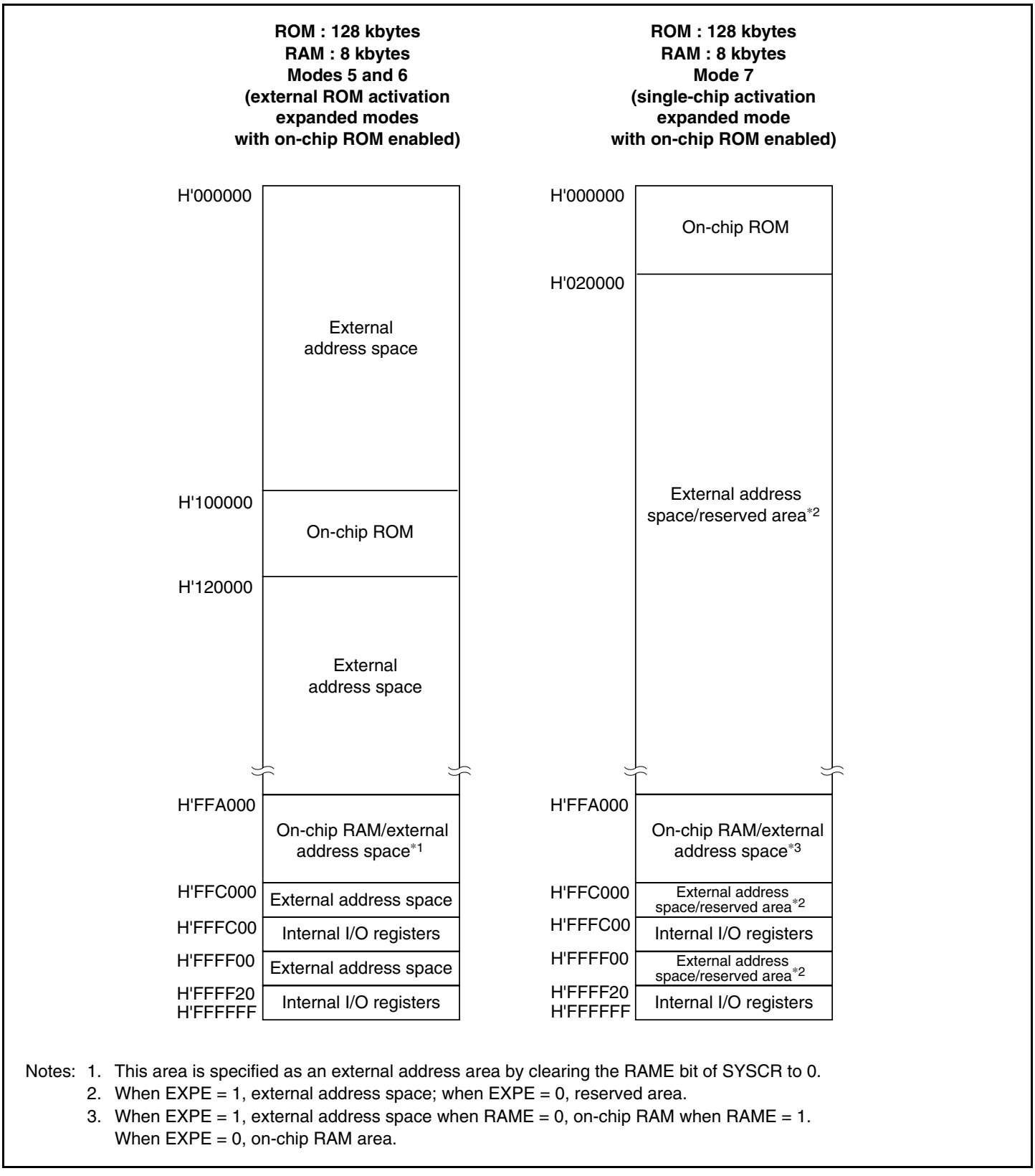


Figure 3.2 H8S/2675 Memory Map (1)



Notes: 1. This area is specified as an external address area by clearing the RAME bit of SYSCR to 0.
2. When EXPE = 1, external address space; when EXPE = 0, reserved area.
3. When EXPE = 1, external address space when RAME = 0, on-chip RAM when RAME = 1.
When EXPE = 0, on-chip RAM area.

Figure 3.2 H8S/2675 Memory Map (2)

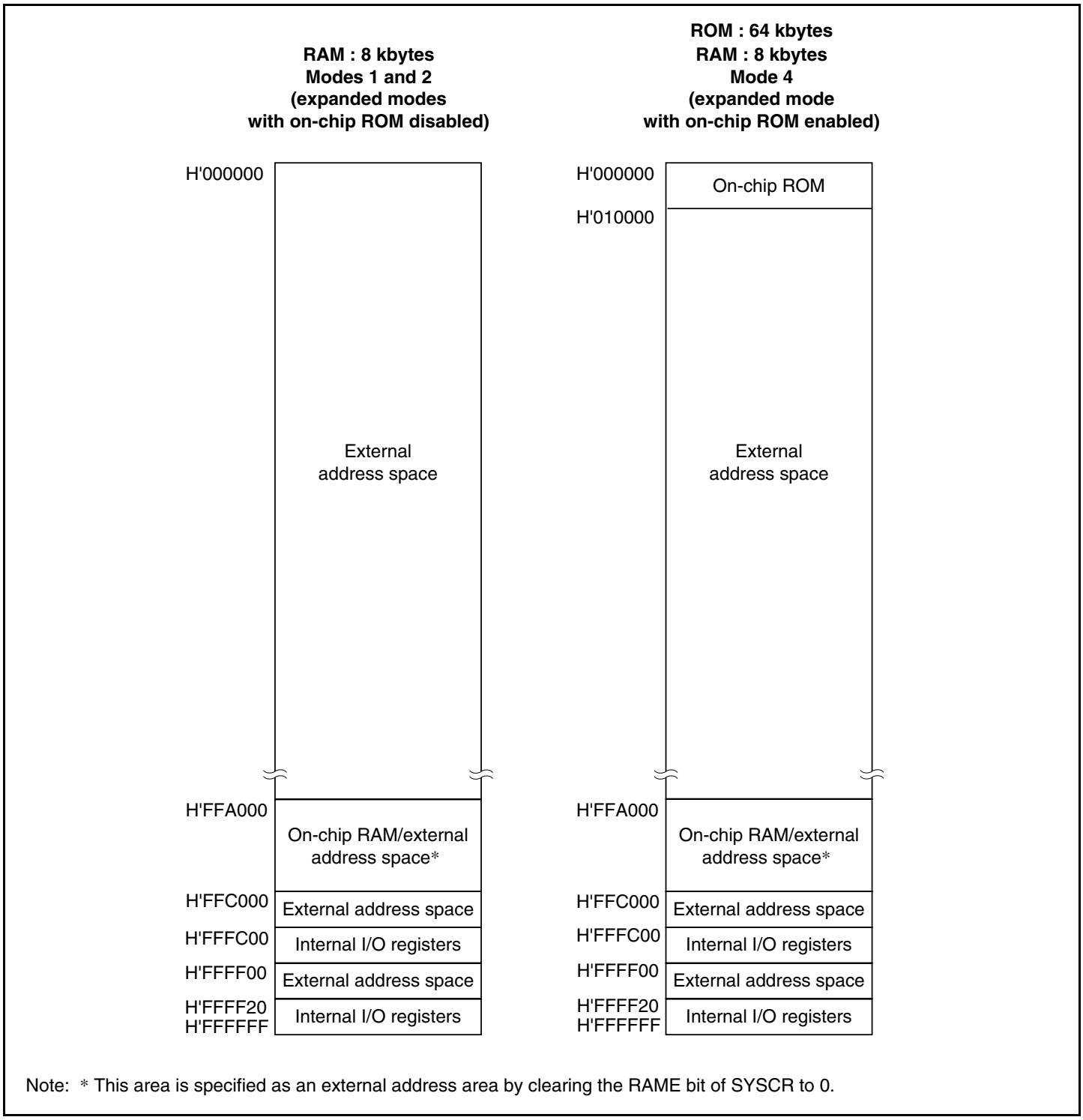
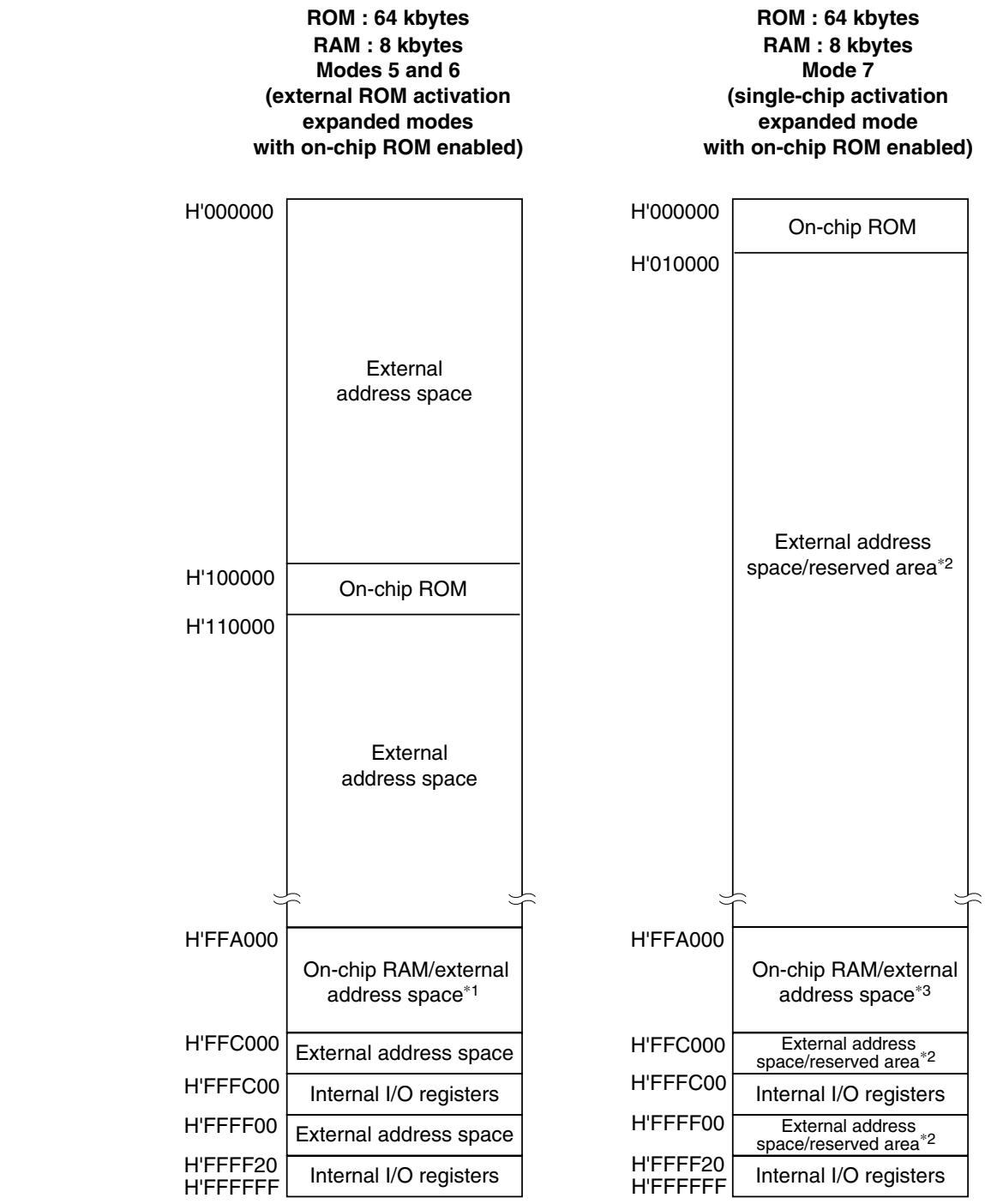


Figure 3.3 H8S/2673 Memory Map (1)



Notes: 1. This area is specified as an external address area by clearing the RAME bit of SYSCR to 0.
2. When EXPE = 1, external address space; when EXPE = 0, reserved area.
3. When EXPE = 1, external address space when RAME = 0, on-chip RAM when RAME = 1.
When EXPE = 0, on-chip RAM area.

Figure 3.3 H8S/2673 Memory Map (2)

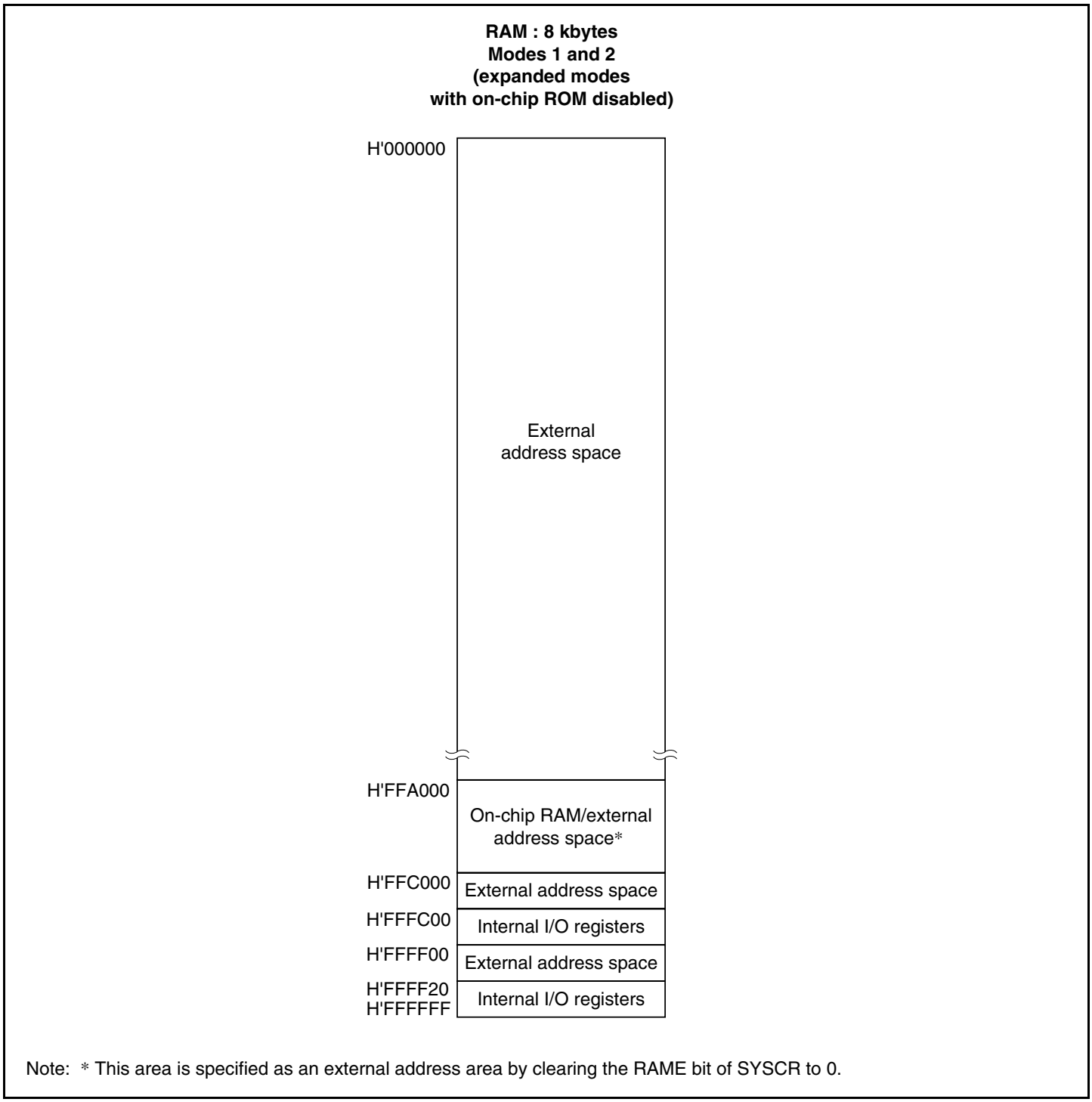
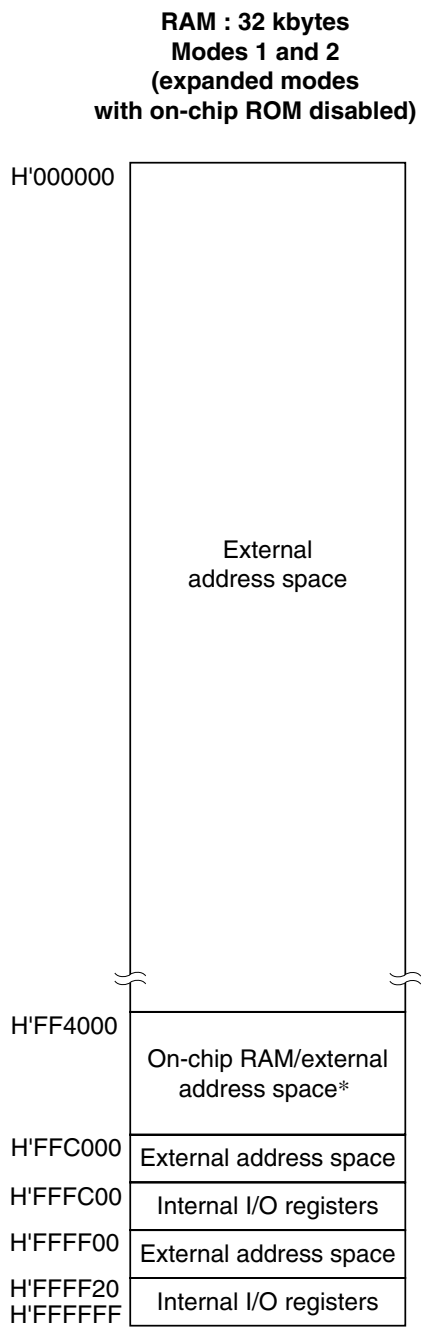


Figure 3.4 H8S/2670 Memory Map



Note: * This area is specified as an external address area by clearing the RAME bit of SYSCR to 0.

Figure 3.5 H8S/2674R Memory Map

Section 4 Exception Handling

4.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, trace, interrupt, or trap instruction. Exception handling is prioritized as shown in table 4.1. If two or more exceptions occur simultaneously, they are accepted and processed in order of priority. Exception sources, the stack structure, and operation of the CPU vary depending on the interrupt control mode. For details on the interrupt control mode, refer to section 5, Interrupt Controller.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High	Reset	Starts immediately after a low-to-high transition at the RES pin, or when the watchdog timer overflows. The CPU enters the reset state when the RES pin is low.
	Trace*[1]	Starts when execution of the current instruction or exception handling ends, if the trace (T) bit in the EXR is set to 1.
	Direct transition*[2]	Starts when the direct transition occurs by execution of the SLEEP instruction.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued.*[3]
Low	Trap instruction *[4]	Started by execution of a trap instruction (TRAPA)

Notes: 1. Traces are enabled only in interrupt control mode 2. Trace exception handling is not executed after execution of an RTE instruction.
2. Not available in this LSI.
3. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.
4. Trap instruction exception handling requests are accepted at all times in program execution state.

4.2 Exception Sources and Exception Vector Table

Different vector addresses are assigned to different exception sources. Table 4.2 lists the exception sources and their vector addresses. Since the usable modes differ depending on the product, for details on each product, refer to section 3, MCU Operating Modes.

Table 4.2 Exception Handling Vector Table

Exception Source		Vector Number	Vector Address*[1] Normal Mode*[2]	Vector Address*[1] Advanced Mode
Power-on reset		0	H'0000 to H'0001	H'0000 to H'0003
Manual reset *[2]		1	H'0002 to H'0003	H'0004 to H'0007
Reserved for system use		2	H'0004 to H'0005	H'0008 to H'000B
		3	H'0006 to H'0007	H'000C to H'000F
		4	H'0008 to H'0019	H'0010 to H'0013
Trace		5	H'000A to H'000B	H'0014 to H'0017
Interrupt (direct transition)*[2]		6	H'000C to H'000D	H'0018 to H'001B
Interrupt (NMI)		7	H'000E to H'000F	H'001C to H'001F
Trap instruction (#0)		8	H'0010 to H'0011	H'0020 to H'0023
(#1)		9	H'0012 to H'0013	H'0024 to H'0027
(#2)		10	H'0014 to H'0015	H'0028 to H'002B
(#3)		11	H'0016 to H'0017	H'002C to H'002F
Reserved for system use		12	H'0018 to H'0019	H'0030 to H'0033
		13	H'001A to H'001B	H'0034 to H'0037
		14	H'001C to H'001D	H'0038 to H'003B
		15	H'001E to H'001F	H'003C to H'003F
External interrupt	IRQ0	16	H'0020 to H'0021	H'0040 to H'0043
	IRQ1	17	H'0022 to H'0023	H'0044 to H'0047
	IRQ2	18	H'0024 to H'0025	H'0048 to H'004B
	IRQ3	19	H'0026 to H'0027	H'004C to H'004F
	IRQ4	20	H'0028 to H'0029	H'0050 to H'0053
	IRQ5	21	H'002A to H'002B	H'0054 to H'0057
	IRQ6	22	H'002C to H'002D	H'0058 to H'005B
	IRQ7	23	H'002E to H'002F	H'005C to H'005F
	IRQ8	24	H'0030 to H'0031	H'0060 to H'0063
	IRQ9	25	H'0032 to H'0033	H'0064 to H'0067
	IRQ10	26	H'0034 to H'0035	H'0068 to H'006B
	IRQ11	27	H'0036 to H'0037	H'006C to H'006F
	IRQ12	28	H'0038 to H'0039	H'0070 to H'0073
	IRQ13	29	H'003A to H'003B	H'0074 to H'0077
	IRQ14	30	H'003C to H'003D	H'0078 to H'007B
	IRQ15	31	H'003E to H'003F	H'007C to H'007F

Exception Source	Vector Number	Vector Address*1	
		Normal Mode*2	Advanced Mode
Internal interrupt*3	32	H'0040 to H'0041	H'0080 to H'0083
	|	|	|
	99	H'00C6 to H'00C7	H'018C to H'018F

Notes: 1. Lower 16 bits of the address.
2. Not available in this LSI.
3. For details of internal interrupt vectors, see section 5.5, Interrupt Exception Handling Vector Table.

4.3 Reset

A reset has the highest exception priority. When the RES pin goes low, all processing halts and this LSI enters the reset. To ensure that this LSI is reset, hold the RES pin low for at least 20 ms at power-up. To reset the chip during operation, hold the RES pin low for at least 20 states. A reset initializes the internal state of the CPU and the registers of on-chip peripheral modules.

The chip can also be reset by overflow of the watchdog timer. For details see section 14, Watchdog Timer.

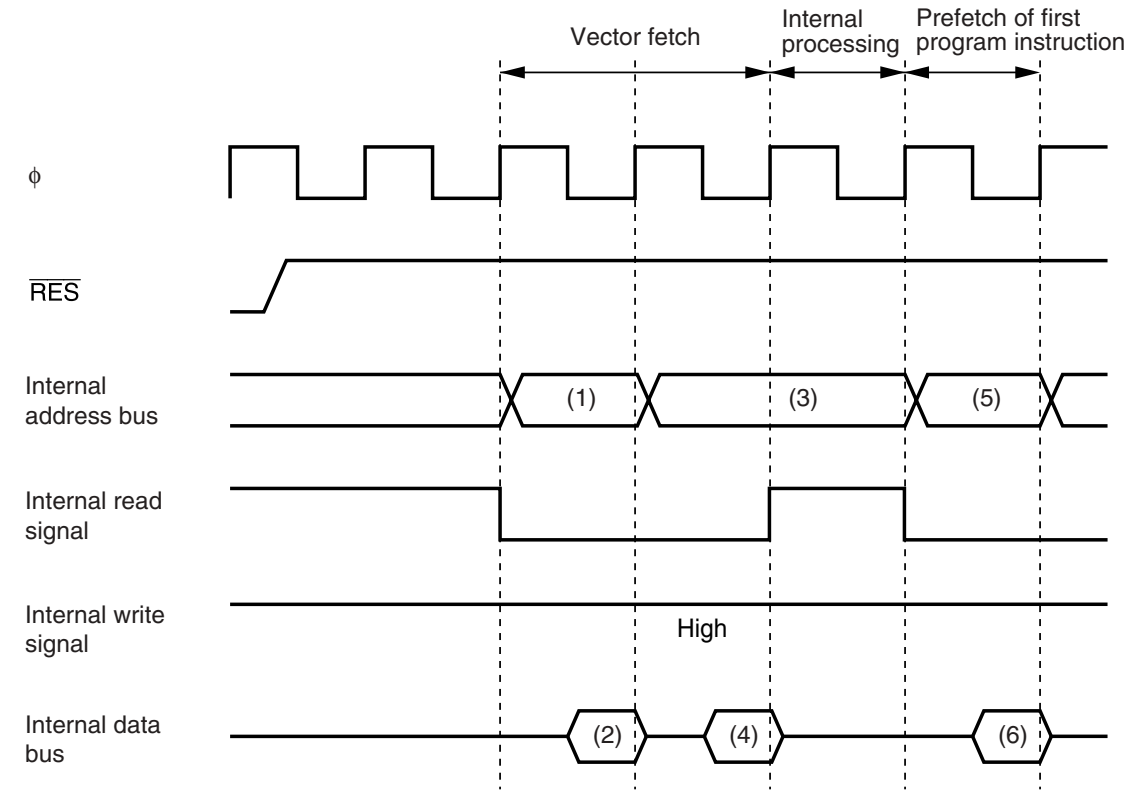
The interrupt control mode is 0 immediately after reset.

4.3.1 Reset exception handling

When the RES pin goes high after being held low for the necessary time, this LSI starts reset exception handling as follows:

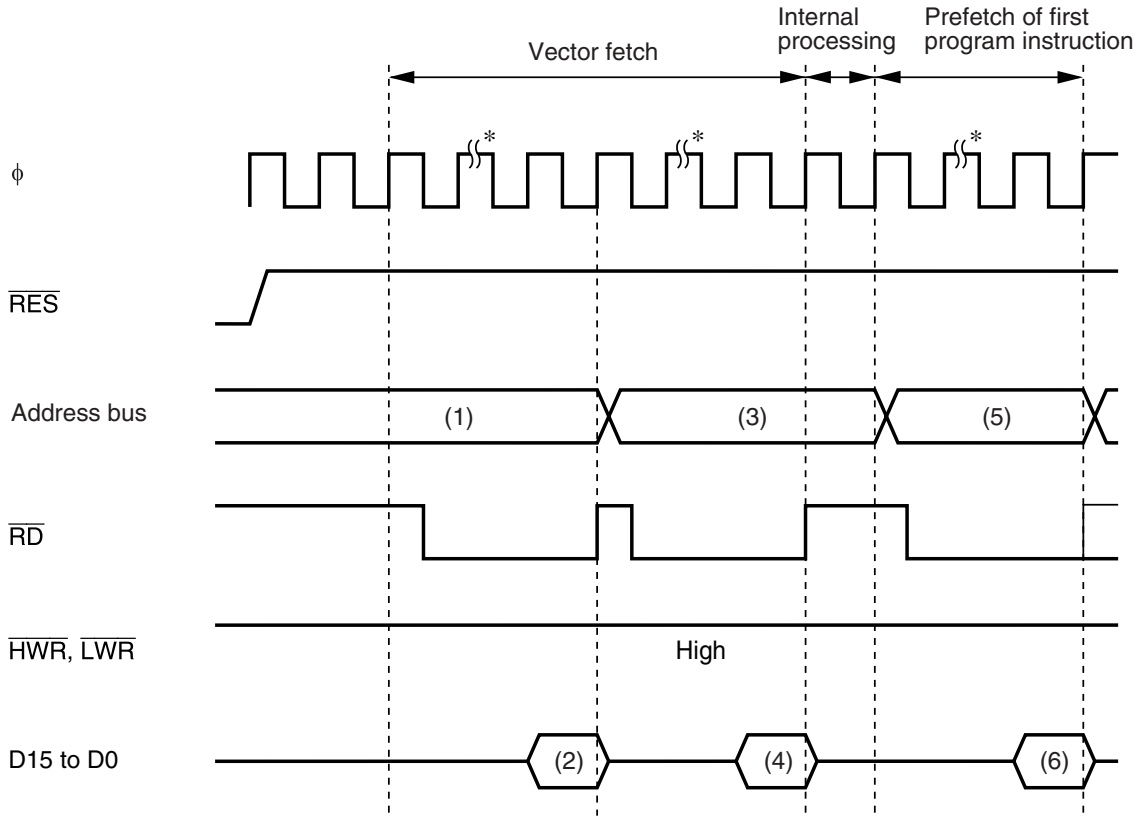
1. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized, the T bit is cleared to 0 in EXR, and the I bit is set to 1 in EXR and CCR.
2. The reset exception handling vector address is read and transferred to the PC, and program execution starts from the address indicated by the PC.

Figures 4.1 and 4.2 show examples of the reset sequence.



(1)(3) Reset exception handling vector address (when reset, (1)=H'000000, (3)=H'000002)
(2)(4) Start address (contents of reset exception handling vector address)
(5) Start address ((5)=(2)(4))
(6) First program instruction

Figure 4.1 Reset Sequence (Advanced Mode with On-chip ROM Enabled)



(1)(3) Reset exception handling vector address (when reset, (1)=H'000000, (3)=H'000002)
(2)(4) Start address (contents of reset exception handling vector address)
(5) Start address ((5)=(2)(4))
(6) First program instruction

Note: * Seven program wait states are inserted.

Figure 4.2 Reset Sequence (Advanced Mode with On-chip ROM Disabled)

4.3.2 Interrupts after Reset

If an interrupt is accepted after a reset but before the stack pointer (SP) is initialized, the PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset. Since the first instruction of a program is always executed immediately after the reset state ends, make sure that this instruction initializes the stack pointer (example: MOV.L #xx: 32, SP).

4.3.3 On-Chip Peripheral Functions after Reset Release

After reset release, MSTPCR is initialized to H'0FFF and all modules except the DMAC, EXDMAC and the DTC enter module stop mode.

Consequently, on-chip peripheral module registers cannot be read or written to. Register reading and writing is enabled when module stop mode is exited.

4.4 Traces

Traces are enabled in interrupt control mode 2. Trace mode is not activated in interrupt control mode 0, irrespective of the state of the T bit. For details on interrupt control modes, see section 5, Interrupt Controller.

If the T bit in EXR is set to 1, trace mode is activated. In trace mode, a trace exception occurs on completion of each instruction. Trace mode is not affected by interrupt masking. Table 4.3 shows the state of CCR and EXR after execution of trace exception handling. Trace mode is canceled by clearing the T bit in EXR to 0. The T bit saved on the stack retains its value of 1, and when control is returned from the trace exception handling routine by the RTE instruction, trace mode resumes. Trace exception handling is not carried out after execution of the RTE instruction.

Interrupts are accepted even within the trace exception handling routine.

Table 4.3 Status of CCR and EXR after Trace Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**I2 to I0**	**T**
0	Trace exception handling cannot be used.			
2	1	—	—	0

Legend:

1: Set to 1

0: Cleared to 0

—: Retains value prior to execution.

4.5 Interrupts

Interrupts are controlled by the interrupt controller. The interrupt controller has two interrupt control modes and can assign interrupts other than NMI to eight priority/mask levels to enable multiplexed interrupt control. The source to start interrupt exception handling and the vector address differ depending on the product. For details, refer to section 5, Interrupt Controller.

The interrupt exception handling is as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended register (EXR) are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution starts from that address.

4.6 Trap Instruction

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap instruction exception handling can be executed at all times in the program execution state.

The trap instruction exception handling is as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended register (EXR) are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution starts from that address.

The TRAPA instruction fetches a start address from a vector table entry corresponding to a vector number from 0 to 3, as specified in the instruction code.

Table 4.4 shows the status of CCR and EXR after execution of trap instruction exception handling.

Table 4.4 Status of CCR and EXR after Trap Instruction Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**I2 to I0**	**T**
0	1	—	—	—
2	1	—	—	0

Legend:

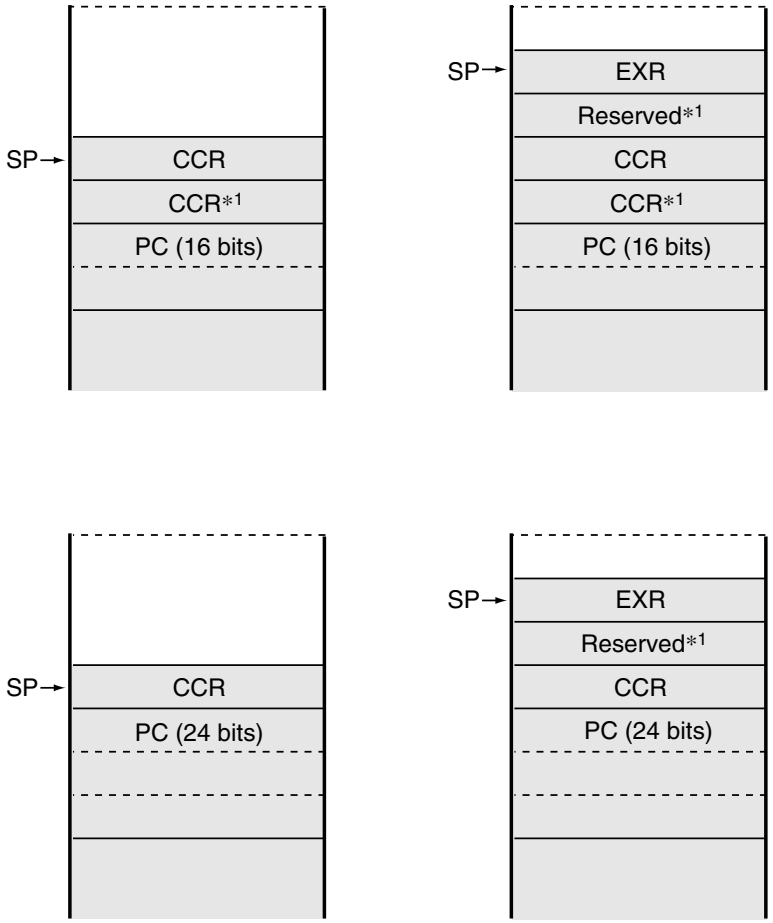
1: Set to 1

0: Cleared to 0

—: Retains value prior to execution.

4.7 Stack Status after Exception Handling

Figure 4.3 shows the stack after completion of trap instruction exception handling and interrupt exception handling.



4.8 Usage Note

When accessing word data or longword data, this LSI assumes that the lowest address bit is 0. The stack should always be accessed by word transfer instruction or longword transfer instruction, and the value of the stack pointer (SP, ER7) should always be kept even. Use the following instructions to save registers:

```
PUSH.W   Rn    (or MOV.W Rn, @-SP)
PUSH.L   ERn   (or MOV.L ERn, @-SP)
```

Use the following instructions to restore registers:

```
POP.W    Rn    (or MOV.W @SP+, Rn)
POP.L    ERn   (or MOV.L @SP+, ERn)
```

Setting SP to an odd value may lead to a malfunction. Figure 4.4 shows an example of operation when the SP value is odd.

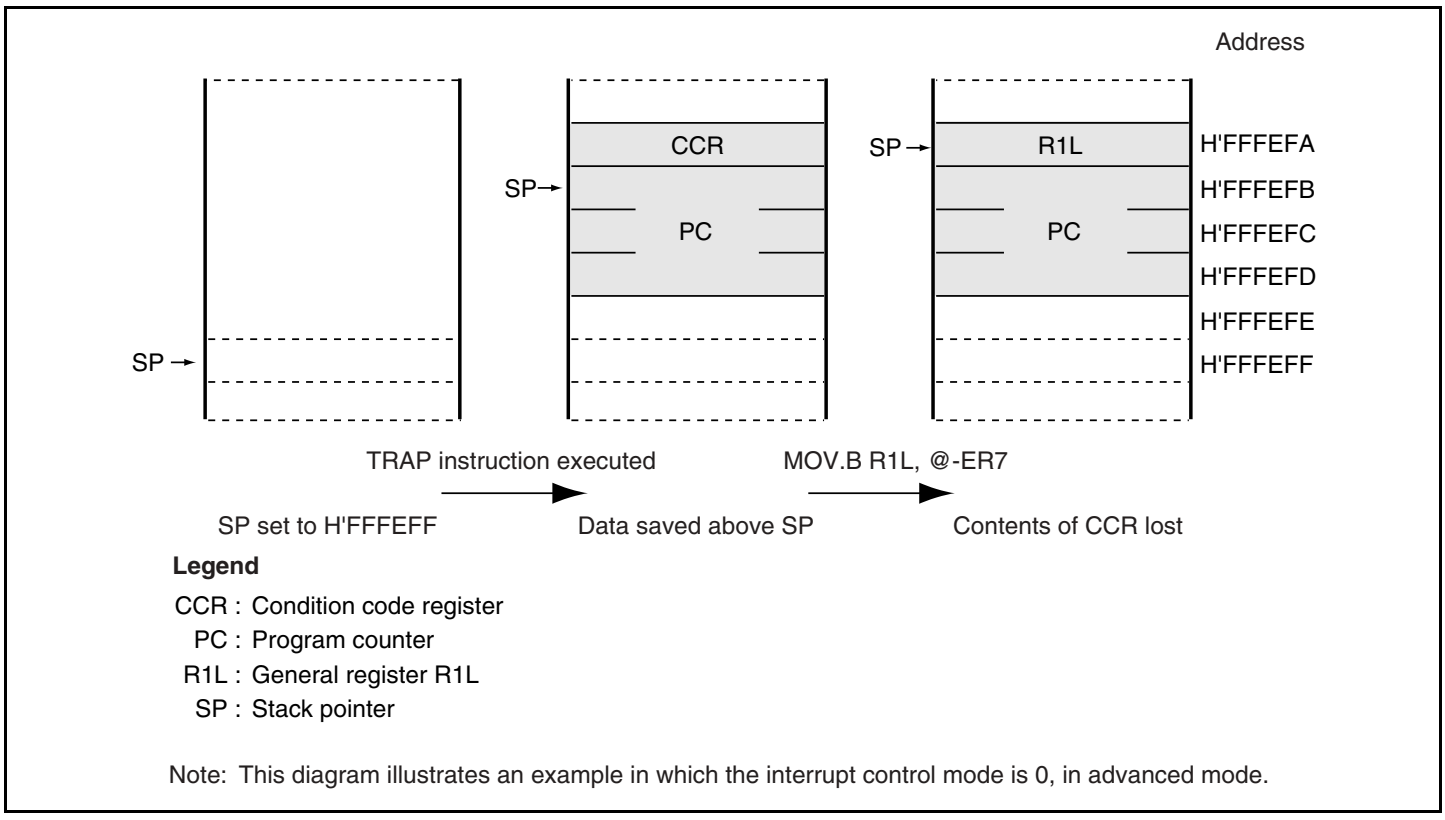


Figure 4.4 Operation when SP Value Is Odd

Section 5 Interrupt Controller

5.1 Features

- Two interrupt control modes

  Any of two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the interrupt control register (INTCR).
- Priorities settable with IPR

  An interrupt priority register (IPR) is provided for setting interrupt priorities. Eight priority levels can be set for each module for all interrupts except NMI. NMI is assigned the highest priority level of 8, and can be accepted at all times.
- Independent vector addresses

  All interrupt sources are assigned independent vector addresses, making it unnecessary for the source to be identified in the interrupt handling routine.
- Seventeen external interrupts

  NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge can be selected for NMI. Falling edge, rising edge, or both edge detection, or level sensing, can be selected for IRQ15 to IRQ0.
- DTC and DMAC control

  DTC and DMAC activations are performed by means of interrupts.

A block diagram of the interrupt controller is shown in figure 5.1.

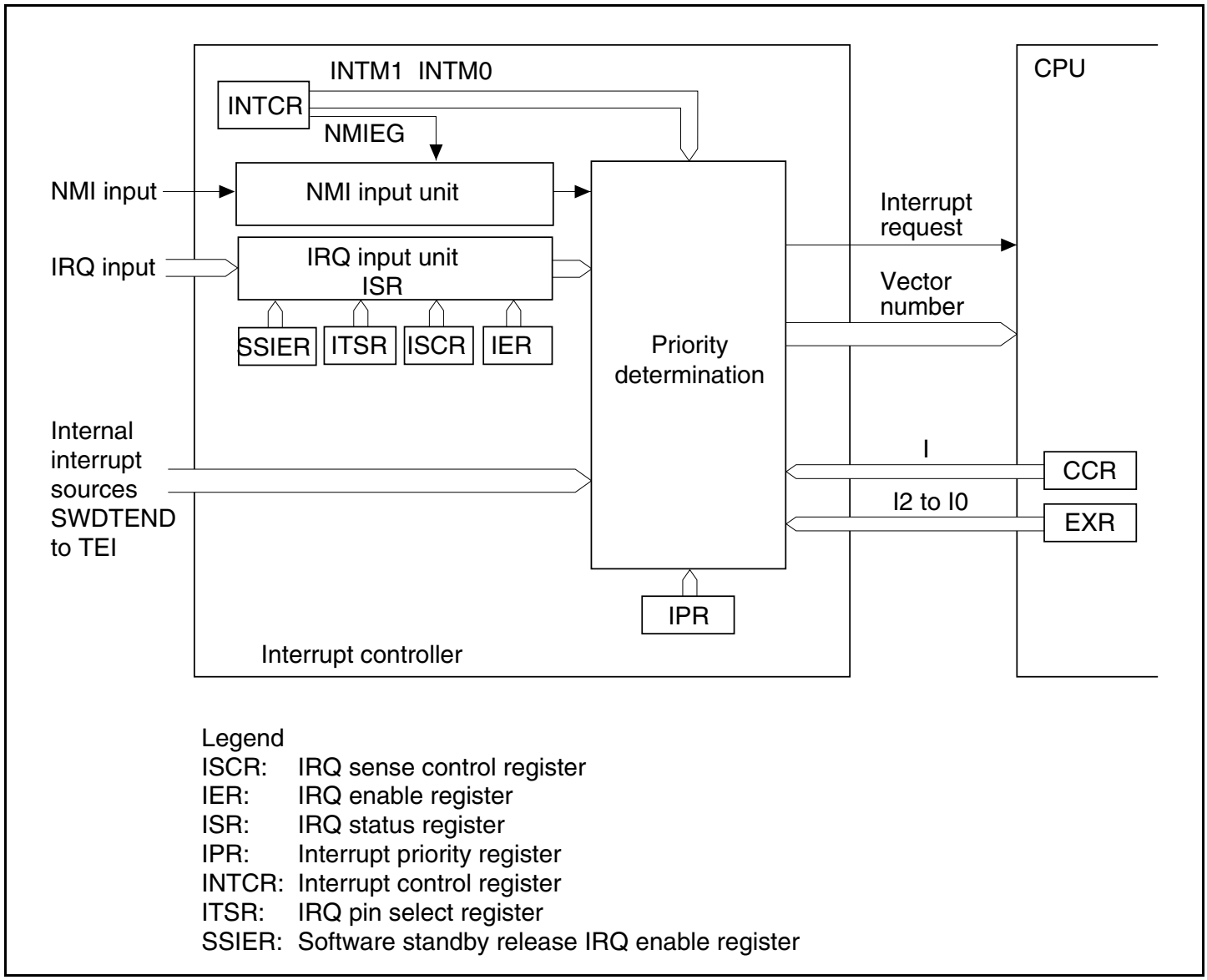


Figure 5.1 Block Diagram of Interrupt Controller

5.2 Input/Output Pins

Table 5.1 shows the pin configuration of the interrupt controller.

Table 5.1 Pin Configuration

Name	I/O	Function
NMI	Input	Nonmaskable external interrupt Rising or falling edge can be selected.
IRQ15 to IRQ0	Input	Maskable external interrupts Rising, falling, or both edges, or level sensing, can be selected.

5.3 Register Descriptions

The interrupt controller has the following registers.

- Interrupt control register (INTCR)
- IRQ sense control register H (ISCRH)
- IRQ sense control register L (ISCRL)
- IRQ enable register (IER)
- IRQ status register (ISR)
- IRQ pin select register (ITSR)
- Software standby release IRQ enable register (SSIER)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)
- Interrupt priority register H (IPRH)
- Interrupt priority register I (IPRI)
- Interrupt priority register J (IPRJ)
- Interrupt priority register K (IPRK)

5.3.1 Interrupt Control Register (INTCR)

INTCR selects the interrupt control mode, and the detected edge for NMI.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved
6	—	0	R/W	These bits can be read from or written to. However, the write value should always be 0.
5	INTM1	0	R/W	Interrupt Control Select Mode 1 and 0
4	INTM0	0	R/W	These bits select either of two interrupt control modes for the interrupt controller. 00: Interrupt control mode 0 Interrupts are controlled by I bit. 01: Setting prohibited. 10: Interrupt control mode 2 Interrupts are controlled by bits I2 to I0, and IPR. 11: Setting prohibited.
3	NMIEG	0	R/W	NMI Edge Select Selects the input edge for the NMI pin. 0: Interrupt request generated at falling edge of NMI input 1: Interrupt request generated at rising edge of NMI input
2 to 0	—	All 0	R/W	Reserved These bits can be read from or written to. However, the write value should always be 0.

5.3.2 Interrupt Priority Registers A to K (IPRA to IPRK)

IPR are eleven 16-bit readable/writable registers that set priorities (levels 7 to 0) for interrupts other than NMI.

The correspondence between interrupt sources and IPR settings is shown in table 5.2 (Interrupt Sources, Vector Addresses, and Interrupt Priorities). Setting a value in the range from H'0 to H'7 in the 3-bit groups of bits 14 to 12, 10 to 8, 6 to 4, and 2 to 0 sets the priority of the corresponding interrupt. IPR should be read in word size.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
14 13 12	IPR14 IPR13 IPR12	1 1 1	R/W R/W R/W	Sets the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)
11	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
10 9 8	IPR10 IPR9 IPR8	1 1 1	R/W R/W R/W	Sets the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)
7	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
6 5 4	IPR6 IPR5 IPR4	1 1 1	R/W R/W R/W	Sets the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
2 1 0	IPR2 IPR1 IPR0	1 1 1	R/W R/W R/W	Sets the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)

5.3.3 IRQ Enable Register (IER)

IER controls enabling and disabling of interrupt requests IRQ15 to IRQ0.

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ15E	0	R/W	IRQ15 Enable The IRQ15 interrupt request is enabled when this bit is 1.
14	IRQ14E	0	R/W	IRQ14 Enable The IRQ14 interrupt request is enabled when this bit is 1.
13	IRQ13E	0	R/W	IRQ13 Enable The IRQ13 interrupt request is enabled when this bit is 1.
12	IRQ12E	0	R/W	IRQ12 Enable The IRQ12 interrupt request is enabled when this bit is 1.
11	IRQ11E	0	R/W	IRQ11 Enable The IRQ11 interrupt request is enabled when this bit is 1.
10	IRQ10E	0	R/W	IRQ10 Enable The IRQ10 interrupt request is enabled when this bit is 1.

Bit	Bit Name	Initial Value	R/W	Description
9	IRQ9E	0	R/W	IRQ9 Enable The IRQ9 interrupt request is enabled when this bit is 1.
8	IRQ8E	0	R/W	IRQ8 Enable The IRQ8 interrupt request is enabled when this bit is 1.
7	IRQ7E	0	R/W	IRQ7 Enable The IRQ7 interrupt request is enabled when this bit is 1.
6	IRQ6E	0	R/W	IRQ6 Enable The IRQ6 interrupt request is enabled when this bit is 1.
5	IRQ5E	0	R/W	IRQ5 Enable The IRQ5 interrupt request is enabled when this bit is 1.
4	IRQ4E	0	R/W	IRQ4 Enable The IRQ4 interrupt request is enabled when this bit is 1.
3	IRQ3E	0	R/W	IRQ3 Enable The IRQ3 interrupt request is enabled when this bit is 1.
2	IRQ2E	0	R/W	IRQ2 Enable The IRQ2 interrupt request is enabled when this bit is 1.
1	IRQ1E	0	R/W	IRQ1 Enable The IRQ1 interrupt request is enabled when this bit is 1.
0	IRQ0E	0	R/W	IRQ0 Enable The IRQ0 interrupt request is enabled when this bit is 1.

5.3.4 IRQ Sense Control Registers H and L (ISCRH, ISCRL)

ISCR select the source that generates an interrupt request at pins IRQ15 to IRQ0.

- ISCRH

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ15SCB	0	R/W	IRQ15 Sense Control B
14	IRQ15SCA	0	R/W	IRQ15 Sense Control A 00: Interrupt request generated at IRQ15 input low level 01: Interrupt request generated at falling edge of IRQ15 input 10: Interrupt request generated at rising edge of IRQ15 input 11: Interrupt request generated at both falling and rising edges of IRQ15 input
13	IRQ14SCB	0	R/W	IRQ14 Sense Control B
12	IRQ14SCA	0	R/W	IRQ14 Sense Control A 00: Interrupt request generated at IRQ14 input low level 01: Interrupt request generated at falling edge of IRQ14 input 10: Interrupt request generated at rising edge of IRQ14 input 11: Interrupt request generated at both falling and rising edges of IRQ14 input
11	IRQ13SCB	0	R/W	IRQ13 Sense Control B
10	IRQ13SCA	0	R/W	IRQ13 Sense Control A 00: Interrupt request generated at IRQ13 input low level 01: Interrupt request generated at falling edge of IRQ13 input 10: Interrupt request generated at rising edge of IRQ13 input 11: Interrupt request generated at both falling and rising edges of IRQ13 input

Bit	Bit Name	Initial Value	R/W	Description
9	IRQ12SCB	0	R/W	IRQ12 Sense Control B
8	IRQ12SCA	0	R/W	IRQ12 Sense Control A 00: Interrupt request generated at IRQ12 input low level 01: Interrupt request generated at falling edge of IRQ12 input 10: Interrupt request generated at rising edge of IRQ12 input 11: Interrupt request generated at both falling and rising edges of IRQ12 input
7	IRQ11SCB	0	R/W	IRQ11 Sense Control B
6	IRQ11SCA	0	R/W	IRQ11 Sense Control A 00: Interrupt request generated at IRQ11 input low level 01: Interrupt request generated at falling edge of IRQ11 input 10: Interrupt request generated at rising edge of IRQ11 input 11: Interrupt request generated at both falling and rising edges of IRQ11 input
5	IRQ10SCB	0	R/W	IRQ10 Sense Control B
4	IRQ10SCA	0	R/W	IRQ10 Sense Control A 00: Interrupt request generated at IRQ10 input low level 01: Interrupt request generated at falling edge of IRQ10 input 10: Interrupt request generated at rising edge of IRQ10 input 11: Interrupt request generated at both falling and rising edges of IRQ10 input
3	IRQ9SCB	0	R/W	IRQ9 Sense Control B
2	IRQ9SCA	0	R/W	IRQ9 Sense Control A 00: Interrupt request generated at IRQ9 input low level 01: Interrupt request generated at falling edge of IRQ9 input 10: Interrupt request generated at rising edge of IRQ9 input 11: Interrupt request generated at both falling and rising edges of IRQ9 input

Bit	Bit Name	Initial Value	R/W	Description
1	IRQ8SCB	0	R/W	IRQ8 Sense Control B
0	IRQ8SCA	0	R/W	IRQ8 Sense Control A 00: Interrupt request generated at IRQ8 input low level 01: Interrupt request generated at falling edge of IRQ8 input 10: Interrupt request generated at rising edge of IRQ8 input 11: Interrupt request generated at both falling and rising edges of IRQ8 input

- ISCRL

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ7SCB	0	R/W	IRQ7 Sense Control B
14	IRQ7SCA	0	R/W	IRQ7 Sense Control A 00: Interrupt request generated at IRQ7 input low level 01: Interrupt request generated at falling edge of IRQ7 input 10: Interrupt request generated at rising edge of IRQ7 input 11: Interrupt request generated at both falling and rising edges of IRQ7 input
13	IRQ6SCB	0	R/W	IRQ6 Sense Control B
12	IRQ6SCA	0	R/W	IRQ6 Sense Control A 00: Interrupt request generated at IRQ6 input low level 01: Interrupt request generated at falling edge of IRQ6 input 10: Interrupt request generated at rising edge of IRQ6 input 11: Interrupt request generated at both falling and rising edges of IRQ6 input

Bit	Bit Name	Initial Value	R/W	Description
11	IRQ5SCB	0	R/W	IRQ5 Sense Control B
10	IRQ5SCA	0	R/W	IRQ5 Sense Control A 00: Interrupt request generated at IRQ5 input low level 01: Interrupt request generated at falling edge of IRQ5 input 10: Interrupt request generated at rising edge of IRQ5 input 11: Interrupt request generated at both falling and rising edges of IRQ5 input
9	IRQ4SCB	0	R/W	IRQ4 Sense Control B
8	IRQ4SCA	0	R/W	IRQ4 Sense Control A 00: Interrupt request generated at IRQ4 input low level 01: Interrupt request generated at falling edge of IRQ4 input 10: Interrupt request generated at rising edge of IRQ4 input 11: Interrupt request generated at both falling and rising edges of IRQ4 input
7	IRQ3SCB	0	R/W	IRQ3 Sense Control B
6	IRQ3SCA	0	R/W	IRQ3 Sense Control A 00: Interrupt request generated at IRQ3 input low level 01: Interrupt request generated at falling edge of IRQ3 input 10: Interrupt request generated at rising edge of IRQ3 input 11: Interrupt request generated at both falling and rising edges of IRQ3 input
5	IRQ2SCB	0	R/W	IRQ2 Sense Control B
4	IRQ2SCA	0	R/W	IRQ2 Sense Control A 00: Interrupt request generated at IRQ2 input low level 01: Interrupt request generated at falling edge of IRQ2 input 10: Interrupt request generated at rising edge of IRQ2 input 11: Interrupt request generated at both falling and rising edges of IRQ2 input

Bit	Bit Name	Initial Value	R/W	Description
3	IRQ1SCB	0	R/W	IRQ1 Sense Control B
2	IRQ1SCA	0	R/W	IRQ1 Sense Control A 00: Interrupt request generated at IRQ1 input low level 01: Interrupt request generated at falling edge of IRQ1 input 10: Interrupt request generated at rising edge of IRQ1 input 11: Interrupt request generated at both falling and rising edges of IRQ1 input
1	IRQ0SCB	0	R/W	IRQ0 Sense Control B
0	IRQ0SCA	0	R/W	IRQ0 Sense Control A 00: Interrupt request generated at IRQ0 input low level 01: Interrupt request generated at falling edge of IRQ0 input 10: Interrupt request generated at rising edge of IRQ0 input 11: Interrupt request generated at both falling and rising edges of IRQ0 input

5.3.5 IRQ Status Register (ISR)

ISR is an IRQ15 to IRQ0 interrupt request flag register.

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ15F	0	R/(W)*	[Setting conditions] When the interrupt source selected by ISCR occurs [Clearing conditions] • Cleared by reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag • When interrupt exception handling is executed when low-level detection is set and IRQn input is high • When IRQn interrupt exception handling is executed when falling, rising, or both-edge detection is set • When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRB of the DTC is cleared to 0 (n=15 to 0)
14	IRQ14F	0	R/(W)*	
13	IRQ13F	0	R/(W)*	
12	IRQ12F	0	R/(W)*	
11	IRQ11F	0	R/(W)*	
10	IRQ10F	0	R/(W)*	
9	IRQ9F	0	R/(W)*	
8	IRQ8F	0	R/(W)*	
7	IRQ7F	0	R/(W)*	
6	IRQ6F	0	R/(W)*	
5	IRQ5F	0	R/(W)*	
4	IRQ4F	0	R/(W)*	
3	IRQ3F	0	R/(W)*	
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	
0	IRQ0F	0	R/(W)*	

Note: Only 0 can be written, to clear the flag.

5.3.6 IRQ Pin Select Register (ITSR)

ITSR selects input pins IRQ15 to IRQ0.

Bit	Bit Name	Initial Value	R/W	Description
15	ITS15	0	R/W	Selects IRQ15 input pin. 0: PF2 1: P27
14	ITS14	0	R/W	Selects IRQ14 input pin. 0: PF1 1: P26
13	ITS13	0	R/W	Selects IRQ13 input pin. 0: P65 1: P25
12	ITS12	0	R/W	Selects IRQ12 input pin. 0: P64 1: P24
11	ITS11	0	R/W	Selects IRQ11 input pin. 0: P63 1: P23
10	ITS10	0	R/W	Selects IRQ10 input pin. 0: P62 1: P22
9	ITS9	0	R/W	Selects IRQ9 input pin. 0: P61 1: P21
8	ITS8	0	R/W	Selects IRQ8 input pin. 0: P60 1: P20
7	ITS7	0	R/W	Selects IRQ7 input pin. 0: P57 1: PH3
6	ITS6	0	R/W	Selects IRQ6 input pin. 0: P56 1: PH2

Bit	Bit Name	Initial Value	R/W	Description
5	ITS5	0	R/W	Selects IRQ5 input pin. 0: P55 1: P85
4	ITS4	0	R/W	Selects IRQ4 input pin. 0: P54 1: P84
3	ITS3	0	R/W	Selects IRQ3 input pin. 0: P53 1: P83
2	ITS2	0	R/W	Selects IRQ2 input pin. 0: P52 1: P82
1	ITS1	0	R/W	Selects IRQ1 input pin. 0: P51 1: P81
0	ITS0	0	R/W	Selects IRQ0 input pin. 0: P50 1: P80

5.3.7 Software Standby Release IRQ Enable Register (SSIER)

SSIER selects the IRQ pins used to recover from the software standby state.

Bit	Bit Name	Initial Value	R/W	Description
15	SSI15	0	R/W	Software Standby Release IRQ Setting These bits select the IRQn pins used to recover from the software standby state. 0: IRQn requests are not sampled in the software standby state (Initial value when n = 15 to 3) 1: When an IRQn request occurs in the software standby state, the chip recovers from the software standby state after the elapse of the oscillation settling time (Initial value when n = 2 to 0) (n = 15 to 0)
14	SSI14	0	R/W	
13	SSI13	0	R/W	
12	SSI12	0	R/W	
11	SSI11	0	R/W	
10	SSI10	0	R/W	
9	SSI9	0	R/W	
8	SSI8	0	R/W	
7	SSI7	0	R/W	
6	SSI6	0	R/W	
5	SSI5	0	R/W	
4	SSI4	0	R/W	
3	SSI3	0	R/W	
2	SSI2	0	R/W	
1	SSI1	0	R/W	
0	SSI0	0	R/W	

5.4 Interrupt Sources

5.4.1 External Interrupts

There are seventeen external interrupts: NMI and IRQ15 to IRQ0. These interrupts can be used to restore the chip from software standby mode.

NMI Interrupt: Nonmaskable interrupt request (NMI) is the highest-priority interrupt, and is always accepted by the CPU regardless of the interrupt control mode or the status of the CPU interrupt mask bits. The NMIEG bit in INTCR can be used to select whether an interrupt is requested at a rising edge or a falling edge on the NMI pin.

IRQ15 to IRQ0 Interrupts: Interrupts IRQ15 to IRQ0 are requested by an input signal at pins IRQ15 to IRQ0. Interrupts IRQ15 to IRQ0 have the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at pins IRQ15 to IRQ0.
- Enabling or disabling of interrupt requests IRQ15 to IRQ0 can be selected with IER.
- The interrupt priority level can be set with IPR.
- The status of interrupt requests IRQ15 to IRQ0 is indicated in ISR. ISR flags can be cleared to 0 by software.

When IRQ15 to IRQ0 interrupt requests occur at low level of IRQn, the corresponding IRQ should be held low until an interrupt handling starts. Then the corresponding IRQ should be set to high in the interrupt handling routine and clear the IRQnF bit (n = 0 to 15) in ISR to 0. Interrupts may not be executed when the corresponding IRQ is set to high before the interrupt handling starts.

Detection of IRQ15 to IRQ0 interrupts does not depend on whether the relevant pin has been set for input or output. However, when a pin is used as an external interrupt input pin, do not clear the corresponding DDR to 0 and use the pin as an I/O pin for another function.

A block diagram of interrupts IRQ15 to IRQ0 is shown in figure 5.2.

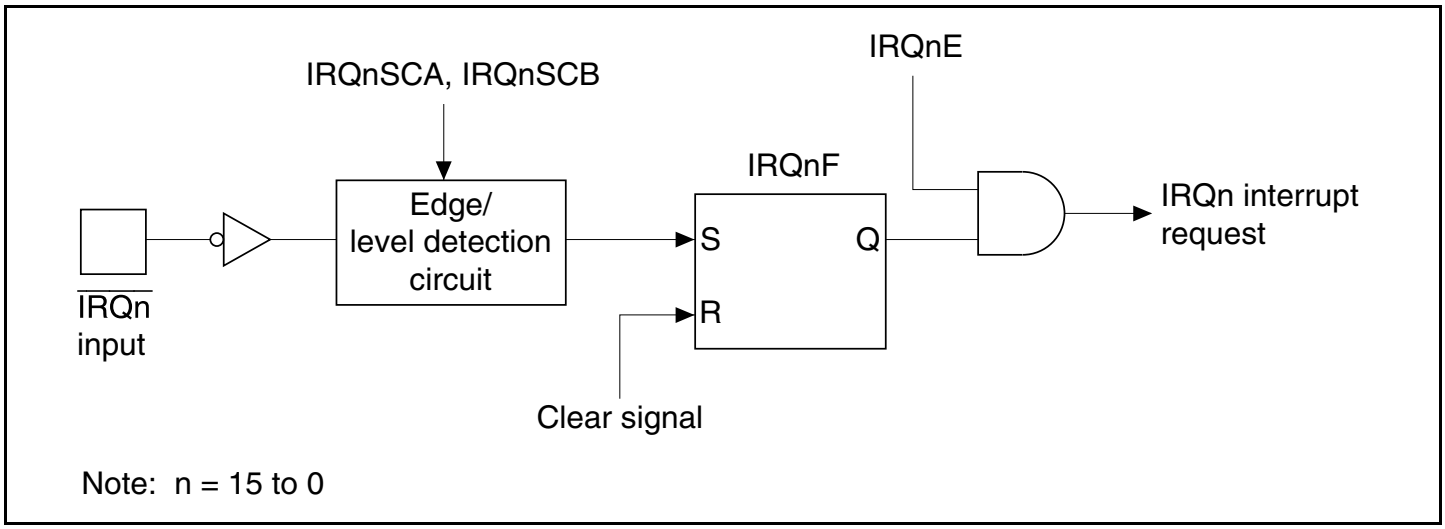


Figure 5.2 Block Diagram of Interrupts IRQ15 to IRQ0

5.4.2 Internal Interrupts

The sources for internal interrupts from on-chip peripheral modules have the following features:

- For each on-chip peripheral module there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. They can be controlled independently. When the enable bit is set to 1, an interrupt request is issued to the interrupt controller.
- The interrupt priority level can be set by means of IPR.
- The DMAC and DTC can be activated by a TPU, SCI, or other interrupt request.
- When the DMAC or DTC is activated by an interrupt request, it is not affected by the interrupt control mode or CPU interrupt mask bit.

5.5 Interrupt Exception Handling Vector Table

Table 5.2 shows interrupt exception handling sources, vector addresses, and interrupt priorities.

For default priorities, the lower the vector number, the higher the priority. When interrupt control mode 2 is set, priorities among modules can be set by means of the IPR. Modules set at the same priority will conform to their default priorities. Priorities within a module are fixed.

Table 5.2 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
External pin	NMI	7	H'001C	—	High	—	—
	IRQ0	16	H'0040	IPRA14 to IPRA12		○	—
	IRQ1	17	H'0044	IPRA10 to IPRA8		○	—
	IRQ2	18	H'0048	IPRA6 to IPRA4		○	—
	IRQ3	19	H'004C	IPRA2 to IPRA0		○	—
	IRQ4	20	H'0050	IPRB14 to IPRB12		○	—
	IRQ5	21	H'0054	IPRB10 to IPRB8		○	—
	IRQ6	22	H'0058	IPRB6 to IPRB4		○	—
	IRQ7	23	H'005C	IPRB2 to IPRB0		○	—
	IRQ8	24	H'0060	IPRC14 to IPRC12		○	—
	IRQ9	25	H'0064	IPRC10 to IPRC8		○	—
	IRQ10	26	H'0068	IPRC6 to IPRC4		○	—
	IRQ11	27	H'006C	IPRC2 to IPRC0		○	—
	IRQ12	28	H'0070	IPRD14 to IPRD12		○	—
	IRQ13	29	H'0074	IPRD10 to IPRD8		○	—
	IRQ14	30	H'0078	IPRD6 to IPRD4		○	—
	IRQ15	31	H'007C	IPRD2 to IPRD0		○	—
DTC	SWDTEND	32	H'0080	IPRE14 to IPRE12		○	—
WDT	WOVI	33	H'0084	IPRE10 to IPRE8		—	—
—	Reserved for system use	34	H'0088	IPRE6 to IPRE4		—	—
Refresh controller	CMI	35	H'008C	IPRE2 to IPRE0		—	—
—	Reserved for system use	36	H'0090	IPRF14 to IPRF12		—	—
		37	H'0094			—	—
A/D	ADI	38	H'0098	IPRF10 to IPRF8		○	○
—	Reserved for system use	39	H'009C			—	—
TPU_0	TGI0A	40	H'00A0	IPRF6 to IPRF4		○	○
	TGI0B	41	H'00A4			○	—
	TGI0C	42	H'00A8		Low	○	—

Interrupt Source	Origin of InterruptSource	Vector Address*

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
—	Reserved for system use	75	H’012C		High	—	—
TMR_1	CMIA1	76	H'0130	IPRH10 to IPRH8		○	—
	CMIB1	77	H'0134			○	—
	OVI1	78	H'0138			—	—
—	Reserved for system use	79	H’013C			—	—

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
—	Reserved for system use	104	H'01A0	IPRJ2 to IPRJ0	High	—	—
		105	H'01A4			—	—
		106	H'01A8			—	—
		107	H'01AC			—	—
		108	H'01B0	IPRK14 to IPRK12		—	—
		109	H'01B4			—	—
		110	H'01B8			—	—
		111	H'01BC			—	—
		112	H'01C0	IPRK10 to IPRK8		—	—
		113	H'01C4			—	—
		114	H'01C8			—	—
		115	H'01CC			—	—
		116	H'01D0	IPRK6 to IPRK4		—	—
		117	H'01D4			—	—
		118	H'01D8			—	—
		119	H'01DC			—	—
		120	H'01E0	IPRK2 to IPRK0		—	—
		121	H'01E4			—	—
		122	H'01E8			—	—
		123	H'01EC			—	—
		124	H'01F0			—	—
		125	H'01F4			—	—
		126	H'01F8			—	—
		127	H'01EC		Low	—	—

Note: Lower 16 bits of the start address.

5.6 Interrupt Control Modes and Interrupt Operation

The interrupt controller has two modes: interrupt control mode 0 and interrupt control mode 2. Interrupt operations differ depending on the interrupt control mode. The interrupt control mode is selected by INTCR. Table 5.3 shows the differences between interrupt control mode 0 and interrupt control mode 2.

Table 5.3 Interrupt Control Modes

Interrupt Control Mode	Priority Setting Registers	Interrupt Mask Bits	Description
0	Default	I	The priorities of interrupt sources are fixed at the default settings. Interrupt sources except for NMI is masked by the I bit.
2	IPR	I2 to I0	8 priority levels except for NMI can be set with IPR. 8-level interrupt mask control is performed by bits I2 to I0.

5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests except for NMI is masked by the I bit of CCR in the CPU. Figure 5.3 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending. If the I bit is cleared, an interrupt request is accepted.
3. Interrupt requests are sent to the interrupt controller, the highest-ranked interrupt according to the priority system is accepted, and other interrupt requests are held pending.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.
7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

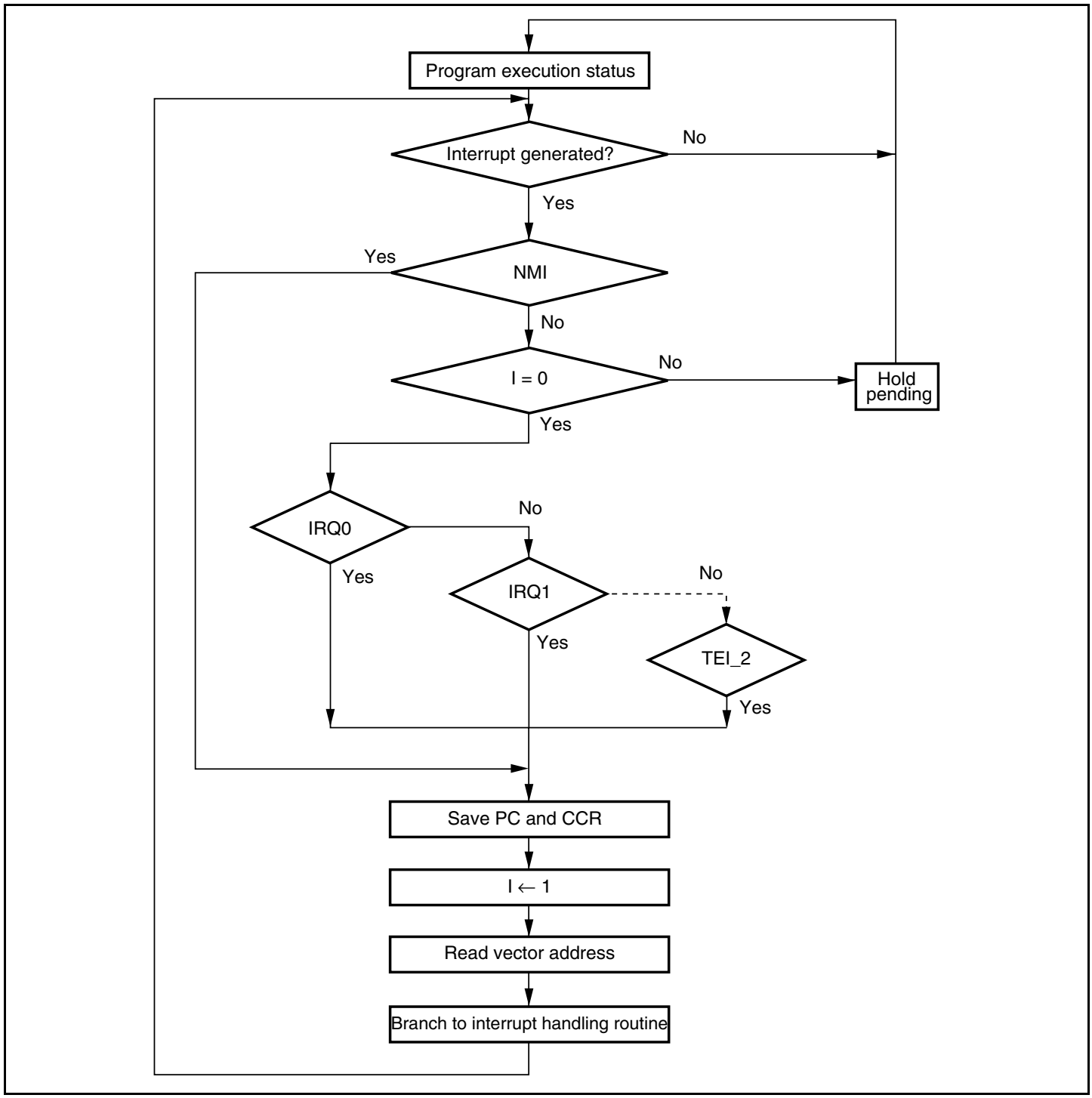


Figure 5.3 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

5.6.2 Interrupt Control Mode 2

In interrupt control mode 2, mask control is done in eight levels for interrupt requests except for NMI by comparing the EXR interrupt mask level (I2 to I0 bits) in the CPU and the IPR setting. Figure 5.4 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. When interrupt requests are sent to the interrupt controller, the interrupt with the highest priority according to the interrupt priority levels set in IPR is selected, and lower-priority interrupt requests are held pending. If a number of interrupt requests with the same priority are generated at the same time, the interrupt request with the highest priority according to the priority system shown in table 5.2 is selected.
3. Next, the priority of the selected interrupt request is compared with the interrupt mask level set in EXR. An interrupt request with a priority no higher than the mask level set at that time is held pending, and only an interrupt request with a priority higher than the interrupt mask level is accepted.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC, CCR, and EXR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority level of the accepted interrupt.

   If the accepted interrupt is NMI, the interrupt mask level is set to H'7.
7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

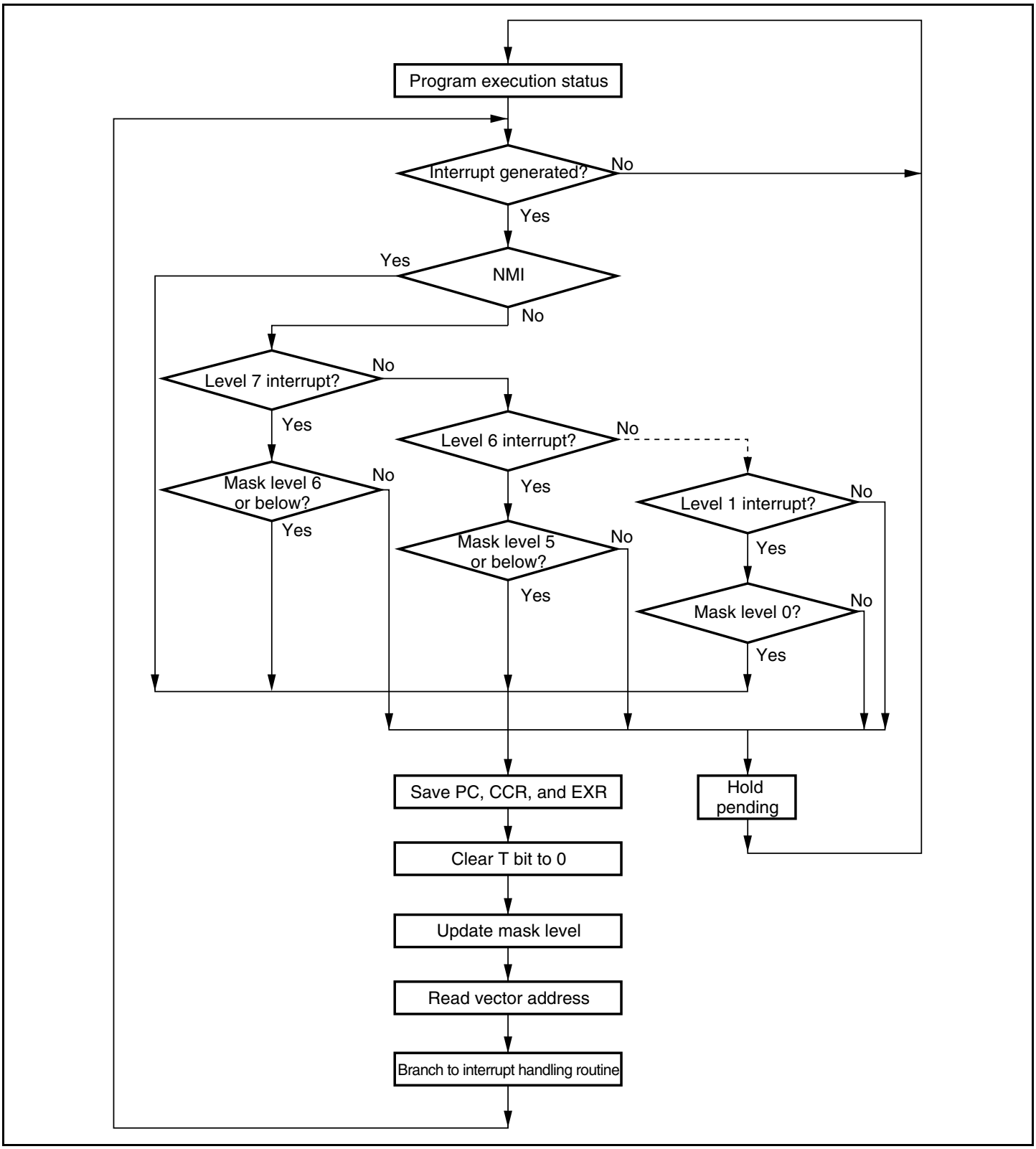


Figure 5.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

5.6.3 Interrupt Exception Handling Sequence

Figure 5.5 shows the interrupt exception handling sequence. The example shown is for the case where interrupt control mode 0 is set in advanced mode, and the program area and stack area are in on-chip memory.

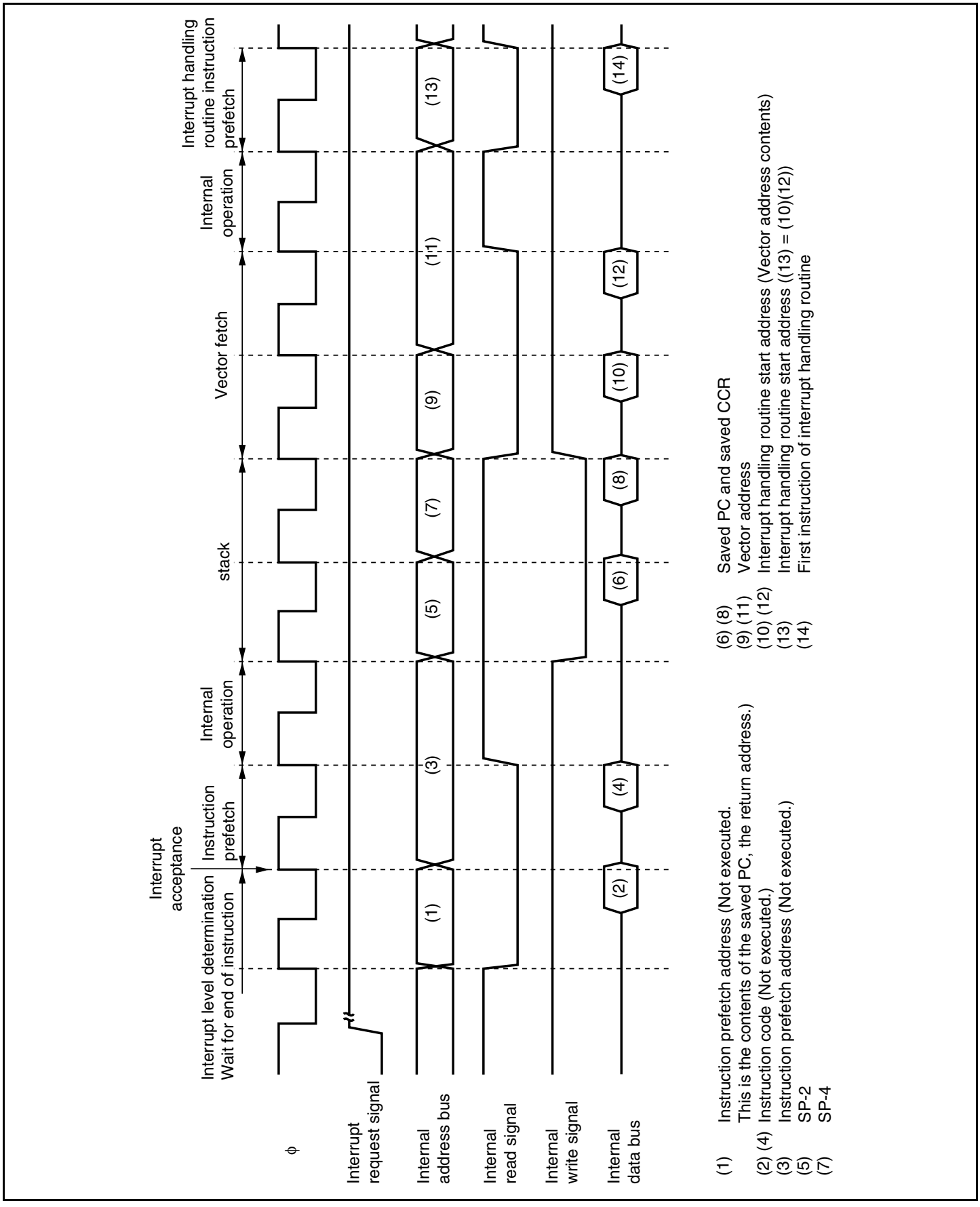


Figure 5.5 Interrupt Exception Handling

5.6.4 Interrupt Response Times

Table 5.4 shows interrupt response times - the interval between generation of an interrupt request and execution of the first instruction in the interrupt handling routine. The execution status symbols used in table 5.4 are explained in table 5.5.

This LSI is capable of fast word transfer to on-chip memory, and have the program area in on-chip ROM and the stack area in on-chip RAM, enabling high-speed processing.

Table 5.4 Interrupt Response Times

		Normal Mode*[5]		Advanced Mode	
No.	Execution Status	Interrupt control mode 0	Interrupt control mode 2	Interrupt control mode 0	Interrupt control mode 2
1	Interrupt priority determination*[1]	3	3	3	3
2	Number of wait states until executing instruction ends*[2]	1 to 19 +2·S_I	1 to 19+2·S_I	1 to 19+2·S_I	1 to 19+2·S_I
3	PC, CCR, EXR stack save	2·S_K	3·S_K	2·S_K	3·S_K
4	Vector fetch	S_I	S_I	2·S_I	2·S_I
5	Instruction fetch*[3]	2·S_I	2·S_I	2·S_I	2·S_I
6	Internal processing*[4]	2	2	2	2
Total (using on-chip memory)		11 to 31	12 to 32	12 to 32	13 to 33

Notes: 1. Two states in case of internal interrupt.
2. Refers to MULXS and DIVXS instructions.
3. Prefetch after interrupt acceptance and interrupt handling routine prefetch.
4. Internal processing after interrupt acceptance and internal processing after vector fetch.
5. Not available in this LSI.

Table 5.5 Number of States in Interrupt Handling Routine Execution Statuses

	Object of Access				
		External Device			
		8 Bit Bus		16 Bit Bus	
Symbol	Internal Memory	2-State Access	3-State Access	2-State Access	3-State Access
Instruction fetch S_I	1	4	6+2m	2	3+m
Branch address read S_J					
Stack manipulation S_K					

Legend:
m: Number of wait states in an external device access.

5.6.5 DTC and DMAC Activation by Interrupt

The DTC and DMAC can be activated by an interrupt. In this case, the following options are available:

- Interrupt request to CPU
- Activation request to DTC
- Activation request to DMAC
- Selection of a number of the above

For details of interrupt requests that can be used to activate the DTC and DMAC, see table 5.2 and section 9, Data Transfer Controller and section 7, DMA Controller.

Figure 5.6 shows a block diagram of the DTC, DMAC, and interrupt controller.

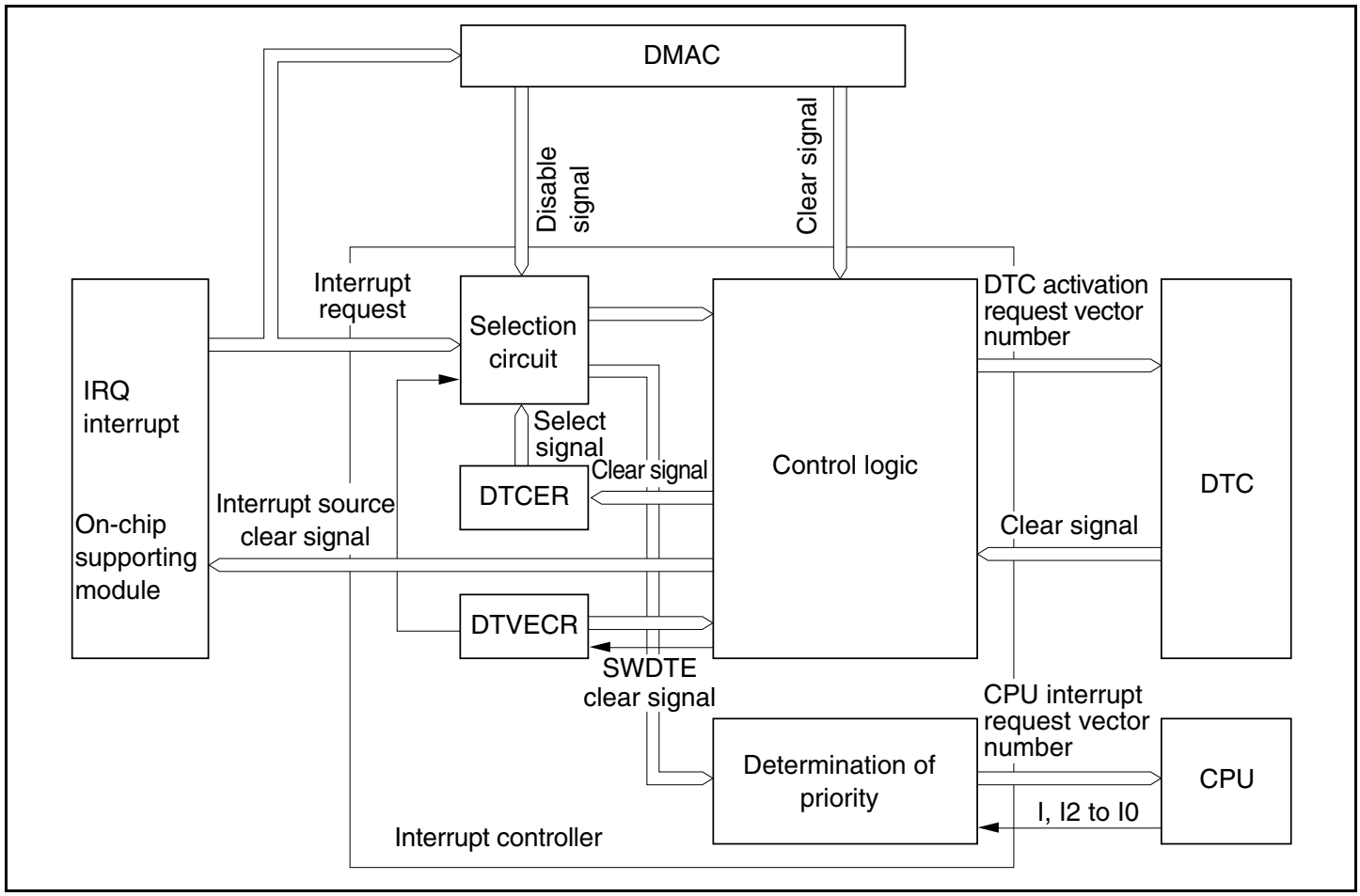


Figure 5.6 DTC, DMAC, and Interrupt Controller

(1) Selection of Interrupt Source: The activation factors for each channel of DMAC are selected by DTF3 to DTF0 bits of DMACR. The DTA bit of DMABCR can be used to select whether the selected activation factors are managed by DMAC. By setting the DTA bit to 1, the interrupt factor which were the activation factor for that DMAC do not act as the DTC activation factor or the CPU interrupt factor.

Interrupt factors other than the interrupts managed by the DMAC are selected as DTC activation source or CPU interrupt source by the DTCE bit of DTCERA to DTCERF of DTC.

By specifying the DISEL bit of the DTC's MRB, it is possible to clear the DTCE bit to 0 after DTC data transfer, and request a CPU interrupt.

If DTC carries out the designate number of data transfers and the transfer counter reads 0, after DTC data transfer, the DTCE bit is also cleared to 0, and an interrupt is requested to the CPU.

(2) Determination of Priority: The DTC activation source is selected in accordance with the default priority order, and is not affected by mask or priority levels. See table 9.1 for the respective priority. DMAC inputs activation factor directly to each channel.

(3) Operation Order: If the same interrupt is selected as a DTC activation source and a CPU interrupt source, the DTC data transfer is performed first, followed by CPU interrupt exception handling.

If the same interrupt is selected as the DMAC activation factor and as the DTC activation factor or CPU interrupt factor, these operate independently.

Table 5.6 shows the interrupt factor clear control and selection of interrupt factors by specification of the DTA bit of DMAC's DMABCR, the DTCE bit of DTC's DTCERA to DTCERH, and the DISEL bit of DTC's MRB.

Table 5.6 Interrupt Source Selection and Clearing Control

Settings

DMAC	DTC		Interrupt Sources Selection/Clearing Control		
DTA	**DTCE**	**DISEL**	**DMAC**	**DTC**	**CPU**
0	0	*	○	X	◎
	1	0	○	◎	X
		1	○	○	◎
1	*	*	◎	X	X

Legend

◎ : The relevant interrupt is used. Interrupt source clearing is performed.
(The CPU should clear the source flag in the interrupt handling routine.)

○ : The relevant interrupt is used. The interrupt source is not cleared.

X : The relevant interrupt cannot be used.

* : Don’t care

Note: The SCI or A/D converter interrupt source is cleared when the DMAC or DTC reads or writes to the prescribed register, and is not dependent upon the DTA bit or DISEL bit.

5.7 Usage Notes

5.7.1 Contention between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to mask interrupts, the masking becomes effective after execution of the instruction.

When an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request of higher priority than that interrupt, interrupt exception handling will be executed for the higher-priority interrupt, and the lower-priority interrupt will be ignored. The same also applies when an interrupt source flag is cleared to 0. Figure 5.7 shows an example in which the TCIEV bit in the TPU's TIER_0 register is cleared to 0. The above contention will not occur if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.

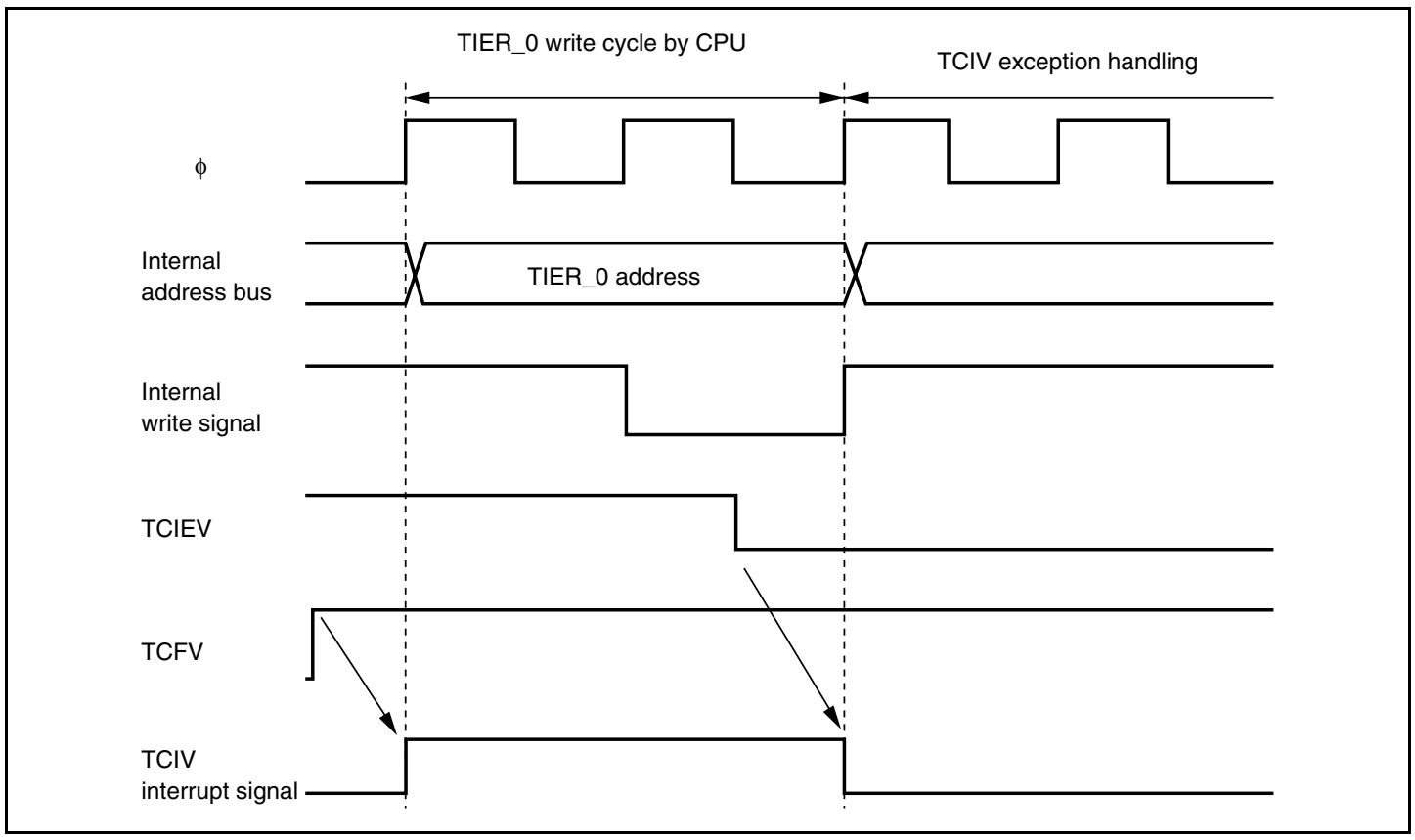


Figure 5.7 Contention between Interrupt Generation and Disabling

5.7.2 Instructions that Disable Interrupts

Instructions that disable interrupts are LDC, ANDC, ORC, and XORC. After any of these instructions is executed, all interrupts including NMI are disabled and the next instruction is always executed. When the I bit is set by one of these instructions, the new value becomes valid two states after execution of the instruction ends.

5.7.3 Times when Interrupts are Disabled

There are times when interrupt acceptance is disabled by the interrupt controller.

The interrupt controller disables interrupt acceptance for a 3-state period after the CPU has updated the mask level with an LDC, ANDC, ORC, or XORC instruction.

5.7.4 Interrupts during Execution of EEPMOV Instruction

Interrupt operation differs between the EEPMOV.B instruction and the EEPMOV.W instruction.

With the EEPMOV.B instruction, an interrupt request (including NMI) issued during the transfer is not accepted until the transfer is completed.

With the EEPMOV.W instruction, if an interrupt request is issued during the transfer, interrupt exception handling starts at a break in the transfer cycle. The PC value saved on the stack in this case is the address of the next instruction. Therefore, if an interrupt is generated during execution of an EEPMOV.W instruction, the following coding should be used.

```
L1:    EEPMOV.W
       MOV.W     R4,R4
       BNE       L1
```

5.7.5 Change of IRQ Pin Select Register (ITSR) Setting

When the ITSR setting is changed, an edge occurs internally and the IRQnF bit (n = 0 to 15) of ISR may be set to 1 at the unintended timing if the selected pin level before the change is different from the selected pin level after the change. If the IRQn interrupt request (n = 0 to 15) is enabled, the interrupt exception handling is executed. To prevent the unintended interrupt, ITSR setting should be changed while the IRQn interrupt request is disabled, then the IRQnF bit should be cleared to 0.

5.7.6 Note on IRQ Status Register (ISR)

Since IRQnF flags may be set to 1 depending on the pin states after a reset, be sure to read from ISR after a reset and then write 0 to clear the IRQnF flags.

Section 6 Bus Controller (BSC)

This LSI has an on-chip bus controller (BSC) that manages the external address space divided into eight areas.

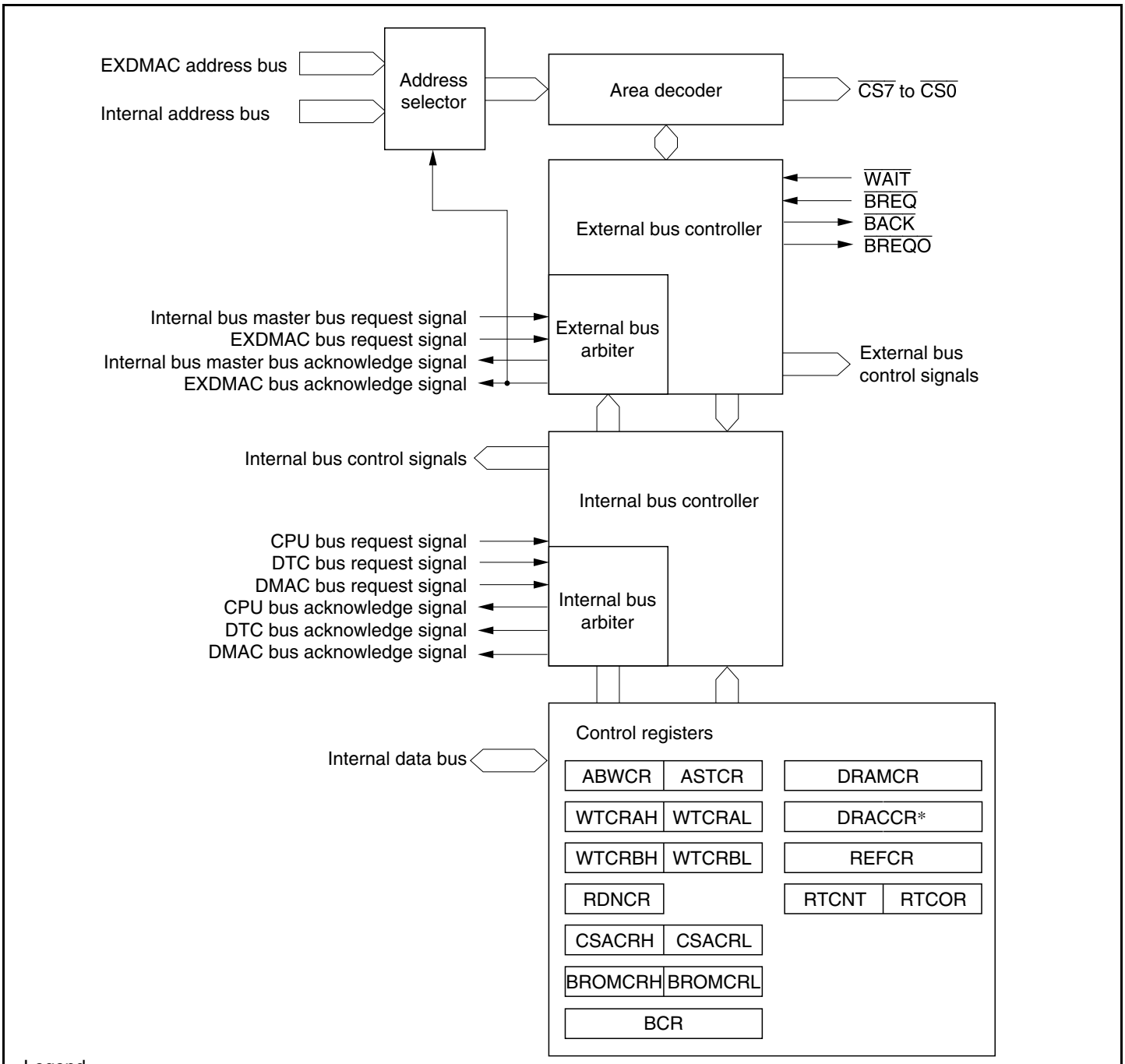
The bus controller also has a bus arbitration function, and controls the operation of the internal bus mastership—the CPU, DMA controller (DMAC), EXDMA controller (EXDMAC), and data transfer controller (DTC).

6.1 Features

- Manages external address space in area units
  Manages the external address space divided into eight areas of 2 Mbytes
  Bus specifications can be set independently for each area
  Burst ROM, DRAM, or synchronous DRAM* interface can be set
- Basic bus interface
  Chip select signals (CS0 to CS7) can be output for areas 0 to 7
  8-bit access or 16-bit access can be selected for each area
  2-state access or 3-state access can be selected for each area
  Program wait states can be inserted for each area
- Burst ROM interface
  Burst ROM interface can be set independently for areas 0 and 1
- DRAM interface
  DRAM interface can be set for areas 2 to 5
- Synchronous DRAM interface
  Continuous synchronous DRAM space can be set for areas 2 to 5
- Bus arbitration function
  Includes a bus arbiter that arbitrates bus mastership between the CPU, DMAC, and DTC

Note: The Synchronous DRAM interface is not supported in the H8S/2678 Series.

A block diagram of the bus controller is shown in figure 6.1.



Legend

ABWCR	: Bus width control register	BROMCRL	: Area 1 burst ROM interface control register
ASTCR	: Access state control register	BCR	: Bus control register
WTCRAH, WTCRAL,		DRAMCR	: DRAM control register
WTCRBH, and WTCRBL	: Wait control registers AH, AL, BH, and BL	DRACCR	: DRAM access control register
RDNCR	: Read strobe timing control register	REFCR	: Refresh control register
CSACRH and CSACRL	: CS assertion period control registers	RTCNT	: Refresh timer counter
BROMCRH	: Area 0 burst ROM interface control register	RTCOR	: Refresh time constant register

Note: * DRACCR is an 8-bit register in the H8S/2678 Series and a 16-bit register in the H8S/2678R Series.

Figure 6.1 Block Diagram of Bus Controller

6.2 Input/Output Pins

Table 6.1 shows the pin configuration of the bus controller.

Table 6.1 Pin Configuration

Name	Symbol	I/O	Function
Address strobe	AS	Output	Strobe signal indicating that basic bus interface space is accessed and address output on address bus is enabled.
Read	RD	Output	Strobe signal indicating that basic bus interface space is being read.
High write/write enable	HWR	Output	Strobe signal indicating that basic bus interface space is written to, and upper half (D15 to D8) of data bus is enabled or DRAM interface space write enable signal.
Low write	LWR	Output	Strobe signal indicating that basic bus interface space is written to, and lower half (D7 to D0) of data bus is enabled.
Chip select 0	CS0	Output	Strobe signal indicating that area 0 is selected.
Chip select 1	CS1	Output	Strobe signal indicating that area 1 is selected
Chip select 2/row address strobe 2/row address strobe*	CS2/ RAS2/* RAS*	Output	Strobe signal indicating that area 2 is selected, DRAM row address strobe signal when area 2 is DRAM interface space or areas 2 to 5 are set as continuous DRAM interface space, or row address strobe signal of the synchronous DRAM when the synchronous DRAM interface is selected.
Chip select 3/row address strobe 3/column address strobe*	CS3/ RAS3/* CAS*	Output	Strobe signal indicating that area 3 is selected, DRAM row address strobe signal when area 3 is DRAM interface space, or column address strobe signal of the synchronous DRAM when the synchronous DRAM interface is selected.

Name	Symbol	I/O	Function
Chip select 4/row address strobe 4/write enable*	CS4/ RAS4/* WE*	Output	Strobe signal indicating that area 4 is selected, DRAM row address strobe signal when area 4 is DRAM interface space, or write enable signal of the synchronous DRAM when the synchronous DRAM interface is selected.
Chip select 5/row address strobe 5/SDRAMϕ*	CS5/ RAS5/* SDRAMϕ*	Output	Strobe signal indicating that area 5 is selected, DRAM row address strobe signal when area 5 is DRAM interface space, or dedicated clock signal for the synchronous DRAM when the synchronous DRAM interface is selected.
Chip select 6	CS6	Output	Strobe signal indicating that area 6 is selected.
Chip select 7	CS7	Output	Strobe signal indicating that area 7 is selected.
Upper column address strobe/upper data mask enable	UCAS/ DQMU*	Output	16-bit DRAM interface space upper column address strobe signal, 8-bit DRAM interface space column address strobe signal, upper data mask signal of 16-bit synchronous DRAM interface space, or data mask signal of 8-bit synchronous DRAM interface space.
Lower column address strobe/ lower data mask enable	LCAS/ DQML*	Output	16-bit DRAM interface space lower column address strobe signal or lower data mask signal for the 16-bit synchronous DRAM interface space.
Output enable/clock enable	OE/CKE*	Output	Output enable signal for the DRAM interface space or clock enable signal for the synchronous DRAM interface space.
Wait	WAIT	Input	Wait request signal when accessing external space.
Bus request	BREQ	Input	Request signal for release of bus to external bus master.
Bus request acknowledge	BACK	Output	Acknowledge signal indicating that bus has been released to external bus master.

Name	Symbol	I/O	Function
Bus request output	BREQO	Output	External bus request signal used when internal bus master accesses external address space when external bus is released.
Data transfer acknowledge 1 (DMAC)	DACK1	Output	Data transfer acknowledge signal for single address transfer by DMAC channel 1.
Data transfer acknowledge 0 (DMAC)	DACK0	DACK0	Data transfer acknowledge signal for single address transfer by DMAC channel 0.
Data transfer acknowledge 3 (EXDMAC)	EDACK3	Output	Data transfer acknowledge signal for single address transfer by EXDMAC channel 3.
Data transfer acknowledge 2 (EXDMAC)	EDACK2	Output	Data transfer acknowledge signal for single address transfer by EXDMAC channel 2.
Data transfer acknowledge 1 (EXDMAC)	EDACK1	Output	Data transfer acknowledge signal for single address transfer by EXDMAC channel 1.
Data transfer acknowledge 0 (EXDMAC)	EDACK0	Output	Data transfer acknowledge signal for single address transfer by EXDMAC channel 0.

Note: These pins are not supported in the H8S/2678 Series.

6.3 Register Descriptions

The bus controller has the following registers.

- Bus width control register (ABWCR)
- Access state control register (ASTCR)
- Wait control register AH (WTCRAH)
- Wait control register AL (WTCRAL)
- Wait control register BH (WTCRBH)
- Wait control register BL (WTCRBL)
- Read strobe timing control register (RDNCR)
- CS assertion period control register H (CSACRH)
- CS assertion period control register L (CSACRL)
- Area 0 burst ROM interface control register (BROMCRH)
- Area 1 burst ROM interface control register (BROMCRL)
- Bus control register (BCR)
- DRAM control register (DRAMCR)
- DRAM access control register (DRACCR)
- Refresh control register (REFCR)
- Refresh timer counter (RTCNT)

- Refresh time constant register (RTCOR)

6.3.1 Bus Width Control Register (ABWCR)

ABWCR designates each area in the external address space as either 8-bit access space or 16-bit access space.

Bit	Bit Name	Initial Value*	R/W	Description
7	ABW7	1/0	R/W	Area 7 to 0 Bus Width Control
6	ABW6	1/0	R/W	These bits select whether the corresponding area is to be designated as 8-bit access space or 16-bit access space.
5	ABW5	1/0	R/W	
4	ABW4	1/0	R/W	
3	ABW3	1/0	R/W	0: Area n is designated as 16-bit access space
2	ABW2	1/0	R/W	
1	ABW1	1/0	R/W	1: Area n is designated as 8-bit access space
0	ABW0	1/0	R/W	(n = 7 to 0)

Note: In modes 2, 4, and 6, ABWCR is initialized to 1. In modes 1, 5, and 7, ABWCR is initialized to 0.

6.3.2 Access State Control Register (ASTCR)

ASTCR designates each area in the external address space as either 2-state access space or 3-state access space.

Bit	Bit Name	Initial Value	R/W	Description
7	AST7	1	R/W	Area 7 to 0 Access State Control
6	AST6	1	R/W	These bits select whether the corresponding area is to be designated as 2-state access space or 3-state access space. Wait state insertion is enabled or disabled at the same time.
5	AST5	1	R/W	
4	AST4	1	R/W	
3	AST3	1	R/W	
2	AST2	1	R/W	
1	AST1	1	R/W	
0	AST0	1	R/W	0: Area n is designated as 2-state access space Wait state insertion in area n access is disabled 1: Area n is designated as 3-state access space Wait state insertion in area n access is enabled (n = 7 to 0)

6.3.3 Wait Control Registers AH, AL, BH, and BL (WTCRAH, WTCRAL, WTCRBH, and WTCRBL)

WTCRA and WTCRB select the number of program wait states for each area in the external address space.

In addition, CAS latency is set when a synchronous DRAM is connected.

- WTCRAH

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
14 13 12	W72 W71 W70	1 1 1	R/W R/W R/W	Area 7 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 7 while AST7 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
11	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
10	W62	1	R/W	Area 6 Wait Control 2 to 0
9	W61	1	R/W	These bits select the number of program wait states when accessing area 6 while AST6 bit in ASTCR = 1.
8	W60	1	R/W	000: Program wait not inserted
				001: 1 program wait state inserted
				010: 2 program wait states inserted
				011: 3 program wait states inserted
				100: 4 program wait states inserted
				101: 5 program wait states inserted
				110: 6 program wait states inserted
				111: 7 program wait states inserted

- WTARAL

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
				This bit is always read as 0 and cannot be modified.
6	W52	1	R/W	Area 5 Wait Control 2 to 0
5	W51	1	R/W	These bits select the number of program wait states when accessing area 5 while AST5 bit in ASTCR = 1.
4	W50	1	R/W	000: Program wait not inserted
				001: 1 program wait state inserted
				010: 2 program wait states inserted
				011: 3 program wait states inserted
				100: 4 program wait states inserted
				101: 5 program wait states inserted
				110: 6 program wait states inserted
				111: 7 program wait states inserted

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
2 1 0	W42 W41 W40	1 1 1	R/W R/W R/W	Area 4 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 4 while AST4 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted

- WTCRBH

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
14 13 12	W32 W31 W30	1 1 1	R/W R/W R/W	Area 3 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 3 while AST3 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
11	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
10	W22	1	R/W	Area 2 Wait Control 2 to 0
9	W21	1	R/W	These bits select the number of program wait states when accessing area 2 while AST2 bit in ASTCR = 1.
8	W20	1	R/W	A CAS latency is set when the synchronous DRAM is connected*. The setting of area 2 is reflected to the setting of areas 2 to 5. A CAS latency can be set regardless of whether or not an ASTCR wait state insertion is enabled.
				000: Program wait not inserted
				001: 1 program wait state inserted
				010: 2 program wait states inserted
				011: 3 program wait states inserted
				100: 4 program wait states inserted
				101: 5 program wait states inserted
				110: 6 program wait states inserted
				111: 7 program wait states inserted
				000: Synchronous DRAM of CAS latency 1 is connected to areas 2 to 5.
				001: Synchronous DRAM of CAS latency 2 is connected to areas 2 to 5.
				010: Synchronous DRAM of CAS latency 3 is connected to areas 2 to 5.
				011: Synchronous DRAM of CAS latency 4 is connected to areas 2 to 5.
				1XXX: Setting prohibited.

Note: The synchronous DRAM interface is not supported in the H8S/2678 Series.

Legend x: Don't care.

- WTCRBL

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
6 5 4	W12 W11 W10	1 1 1	R/W R/W R/W	Area 1 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 1 while AST1 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
3	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
2 1 0	W02 W01 W00	1 1 1	R/W R/W R/W	Area 0 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 0 while AST0 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted

6.3.4 Read Strobe Timing Control Register (RDNCR)

RDNCR selects the read strobe signal (RD) negation timing in a basic bus interface read access.

Bit	Bit Name	Initial Value	R/W	Description
7	RDN7	0	R/W	Read Strobe Timing Control 7 to 0
6	RDN6	0	R/W	These bits set the negation timing of the read strobe in a corresponding area read access.
5	RDN5	0	R/W	
4	RDN4	0	R/W	
3	RDN3	0	R/W	As shown in figure 6.2, the read strobe for an area for which the RDNn bit is set to 1 is negated one half-state earlier than that for an area for which the RDNn bit is cleared to 0. The read data setup and hold time specifications are also one half-state earlier.
2	RDN2	0	R/W	
1	RDN1	0	R/W	
0	RDN0	0	R/W	0: In an area n read access, the RD is negated at the end of the read cycle
				1: In an area n read access, the RD is negated one half-state before the end of the read cycle
				(n = 7 to 0)

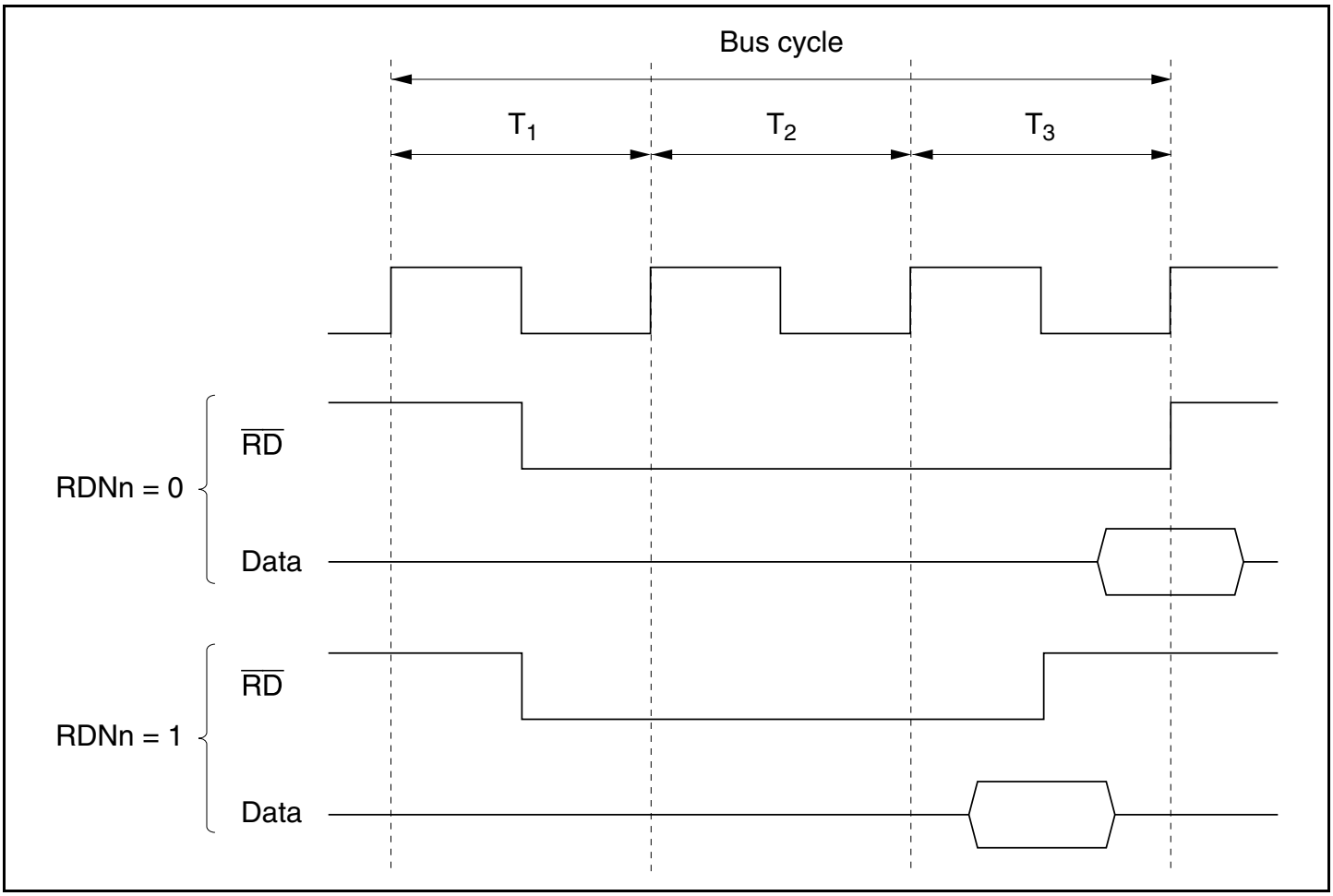


Figure 6.2 Read Strobe Negation Timing (Example of 3-State Access Space)

6.3.5 CS Assertion Period Control Registers H, L (CSACRH, CSACRL)

CSACRH and CSACRL select whether or not the assertion period of the basic bus interface chip select signals (CSn) and address signals is to be extended. Extending the assertion period of the CSn and address signals allows flexible interfacing to external I/O devices.

- CSACRH

Bit	Bit Name	Initial Value	R/W	Description
7	CSXH7	0	R/W	CS and Address Signal Assertion Period Control 1 These bits specify whether or not the T_h cycle is to be inserted (see figure 6.3). When an area for which the CSXHn bit is set to 1 is accessed, a one-state T_h cycle, in which only the CSn and address signals are asserted, is inserted before the normal access cycle. 0: In area n basic bus interface access, the CSn and address assertion period (T_h) is not extended 1: In area n basic bus interface access, the CSn and address assertion period (T_h) is extended (n = 7 to 0)
6	CSXH6	0	R/W	
5	CSXH5	0	R/W	
4	CSXH4	0	R/W	
3	CSXH3	0	R/W	
2	CSXH2	0	R/W	
1	CSXH1	0	R/W	
0	CSXH0	0	R/W	

- CSACRL

Bit	Bit Name	Initial Value	R/W	Description
7	CSXT7	0	R/W	CS and Address Signal Assertion Period Control 2 These bits specify whether or not the T_t cycle shown in figure 6.3 is to be inserted. When an area for which the CSXTn bit is set to 1 is accessed, a one-state T_t cycle, in which only the CSn and address signals are asserted, is inserted before the normal access cycle. 0: In area n basic bus interface access, the CSn and address assertion period (T_t) is not extended 1: In area n basic bus interface access, the CSn and address assertion period (T_t) is extended (n = 7 to 0)
6	CSXT6	0	R/W	
5	CSXT5	0	R/W	
4	CSXT4	0	R/W	
3	CSXT3	0	R/W	
2	CSXT2	0	R/W	
1	CSXT1	0	R/W	
0	CSXT0	0	R/W	

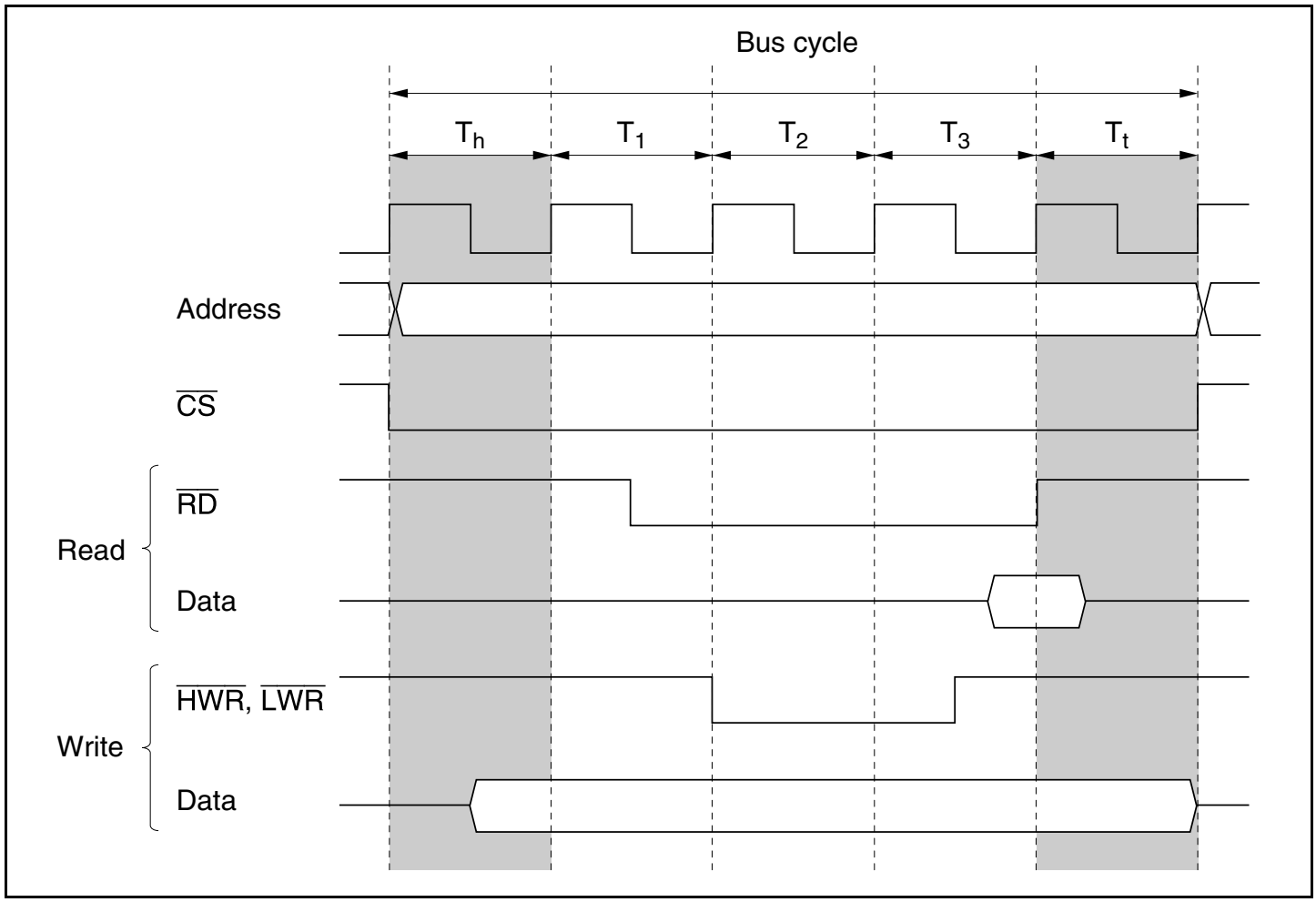


Figure 6.3 CS and Address Assertion Period Extension (Example of 3-State Access Space and RDNn = 0)

6.3.6 Area 0 Burst ROM Interface Control Register (BROMCRH)
Area 1 Burst ROM Interface Control Register (BROMCRL)

BROMCRH and BROMCRL are used to make burst ROM interface settings. Area 0 and area 1 burst ROM interface settings can be made independently in BROMCRH and BROMCRL, respectively.

Bit	Bit Name	Initial Value	R/W	Description
7	BSRMn	0	R/W	Burst ROM Interface Select Selects the basic bus interface or burst ROM interface. 0: Basic bus interface space 1: Burst ROM interface space
6 5 4	BSTSn2 BSTSn1 BSTSn0	0 0 0	R/W R/W R/W	Burst Cycle Select These bits select the number of burst cycle states. 000: 1 state 001: 2 states 010: 3 states 011: 4 states 100: 5 states 101: 6 states 110: 7 states 111: 8 states
3 2	— —	0 0	R/W R/W	Reserved These bits are always read as 0. The initial value should not be changed.
1 0	BSWDn1 BSWDn0	0 0	R/W R/W	Burst Word Number Select These bits select the number of words that can be burst-accessed on the burst ROM interface. 00: Maximum 4 words 01: Maximum 8 words 10: Maximum 16 words 11: Maximum 32 words

(n = 1 or 0)

6.3.7 Bus Control Register (BCR)

BCR is used for idle cycle settings, selection of the external bus released state protocol, enabling or disabling of the write data buffer function, and enabling or disabling of WAIT pin input.

Bit	Bit Name	Initial Value	R/W	Description
15	BRLE	0	R/W	External Bus Release Enable Enables or disables external bus release. 0: External bus release disabled BREQ, BACK, and BREQO pins can be used as I/O ports 1: External bus release enabled
14	BREQOE	0	R/W	BREQO Pin Enable Controls outputting the bus request signal (BREQO) to the external bus master in the external bus released state, when an internal bus master performs an external address space access. 0: BREQO output disabled. I/O port 1: BREQO output enabled. BREQO request is generated./F3 1 Tf3

Bit	Bit Name	Initial Value	R/W	Description
10	ICIS0	1	R/W	Idle Cycle Insert 0 When an external read cycle and external write cycle are performed consecutively, an idle cycle can be inserted between the bus cycles. 0: Idle cycle not inserted 1: Idle cycle inserted
9	WDBE	0	R/W	Write Data Buffer Enable The write data buffer function can be used for an external write cycle or DMAC single address transfer cycle. 0: Write data buffer function not used 1: Write data buffer function used
8	WAITE	0	R/W	WAIT Pin Enable Selects enabling or disabling of wait input by the WAIT pin. 0: Wait input by WAIT pin disabled WAIT pin can be used as I/O port 1: Wait input by WAIT pin enabled
7 to 3	—	0	R/W	Reserved These are readable/writable bits, but the write value should always be 0.
2	ICIS2	0	R/W	Idle Cycle Insert 2 When an external write cycle and external read cycle are performed consecutively, an idle cycle can be inserted between the bus cycles. 0: Idle cycle not inserted 1: Idle cycle inserted Note: Bit 2 is a reserved bit in the H8S/2678 Series. This bit is readable/writable, but the write value should always be 0.
1 to 0	—	0	R/W	Reserved These bits can be read from or written to. However, the write value should always be 0.

6.3.8 DRAM Control Register (DRAMCR)

DRAMCR is used to make DRAM/synchronous DRAM* interface settings.

Note: The synchronous DRAM interface is not supported in the H8S/2678 Series.

Bit	Bit Name	Initial Value	R/W	Description
15	OEE	0	R/W	OE Output Enable The OE signal used when EDO page mode DRAM is connected can be output from the (OE) pin. The OE signal is common to all areas designated as DRAM space. When the synchronous DRAM is connected, the CKE signal can be output from the (OE) pin. The CKE signal is common to the continuous synchronous DRAM space. 0: OE/CKE signal output disabled (OE)/(CKE) pin can be used as I/O port 1: OE/CKE signal output enabled
14	RAST	0	R/W	RAS Assertion Timing Select Selects whether, in DRAM access, the RAS signal is asserted from the start of the T_r cycle (rising edge of ø) or from the falling edge of ø. Figure 6.4 shows the relationship between the RAST bit setting and the RAS assertion timing. The setting of this bit applies to all areas designated as DRAM space. 0: RAS is asserted from ø falling edge in T_r cycle 1: RAS is asserted from start of T_r cycle
13	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12	CAST	0	R/W	Column Address Output Cycle Number Select Selects whether the column address output cycle in DRAM access comprises 3 states or 2 states. The setting of this bit applies to all areas designated as DRAM space. 0: 2 states 1: 3 states
11	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.
10 9 8	RMTS2 RMTS1 RMTS0	0 0 0	R/W R/W R/W	DRAM/Continuous Synchronous DRAM Space Select These bits designate DRAM/continuous synchronous DRAM space for areas 2 to 5. When continuous DRAM space is set, it is possible to connect large-capacity DRAM exceeding 2 Mbytes per area. In this case, the RAS signal is output from the CS2 pin. When continuous synchronous DRAM space is set, it is possible to connect large-capacity synchronous DRAM exceeding 2 Mbytes per area. In this case, the RAS, CAS, and WE signals are output from CS2, CS3, and CS4 pins, respectively. When synchronous DRAM mode is set, the mode registers of the synchronous DRAM can be set. 000: Normal space 001: Normal space in areas 3 to 5 DRAM space in area 2 010: Normal space in areas 4 and 5 DRAM space in areas 2 and 3 011: DRAM space in areas 2 to 5 100: Continuous synchronous DRAM space (setting prohibited in the H8S/2678 Series) 101: Synchronous DRAM mode setting (setting prohibited in the H8S/2678 Series) 110: Setting prohibited 111: Continuous DRAM space in areas 2 to 5

Bit	Bit Name	Initial Value	R/W	Description
7	BE	0	R/W	Burst Access Enable Selects enabling or disabling of burst access to areas designated as DRAM/continuous synchronous DRAM space. DRAM/continuous synchronous DRAM space burst access is performed in fast page mode. When using EDO page mode DRAM, the OE signal must be connected. 0: Full access 1: Access in fast page mode
6	RCDM	0	R/W	RAS Down Mode When access to DRAM space is interrupted by an access to normal bus space, an access to an internal I/O register, etc., this bit selects whether the RAS signal is held low while waiting for the next DRAM access (RAS down mode), or is driven high again (RAS up mode). The setting of this bit is valid only when the BE bit is set to 1. If this bit is cleared to 0 when set to 1 in the RAS down state, the RAS down state is cleared at that point, and RAS goes high. When continuous synchronous DRAM space is set, reading from and writing to this bit is enabled. However, the setting does not affect the operation. 0: RAS up mode selected for DRAM space access 1: RAS down mode selected for DRAM space access

Bit	Bit Name	Initial Value	R/W	Description
5	DDS	0	R/W	DMAC Single Address Transfer Option Selects whether full access is always performed or burst access is enabled when DMAC single address transfer is performed on the DRAM/synchronous DRAM interface. When the BE bit is cleared to 0 in DRAMCR, disabling DRAM/synchronous DRAM burst access, DMAC single address transfer is performed in full access mode regardless of the setting of this bit. This bit has no effect on other bus master external accesses or DMAC dual address transfers. 0: Full access is always executed 1: Burst access is enabled
4	EDDS	0	R/W	EXDMAC Single Address Transfer Option Selects whether full access is always performed or burst access is enabled when EXDMAC single address transfer is performed on the DRAM/synchronous DRAM interface. When the BE bit is cleared to 0 in DRAMCR, disabling DRAM/synchronous DRAM burst access, EXDMAC single address transfer is performed in full access mode regardless of the setting of this bit. This bit has no effect on other bus master external accesses or EXDMAC dual address transfers. 0: Full access is always executed 1: Burst access is enabled
3	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	MXC2	0	R/W	Address Multiplex Select
1	MXC1	0	R/W	These bits select the size of the shift toward the lower half of the row address in row address/column address multiplexing. In burst operation on the DRAM/synchronous DRAM interface, these bits also select the row address bits to be used for comparison. When the MXC2 bit is set to 1 while continuous synchronous DRAM space is set, the address precharge setting command (Precharge-sel) is output to the upper column address. For details, refer to sections 6.6.2 and 6.7.2, Address Multiplexing. DRAM interface 000: 8-bit shift • When 8-bit access space is designated: Row address bits A23 to A8 used for comparison • When 16-bit access space is designated: Row address bits A23 to A9 used for comparison 001: 9-bit shift • When 8-bit access space is designated: Row address bits A23 to A9 used for comparison • When 16-bit access space is designated: Row address bits A23 to A10 used for comparison 010: 10-bit shift • When 8-bit access space is designated: Row address bits A23 to A10 used for comparison • When 16-bit access space is designated: Row address bits A23 to A11 used for comparison
0	MXC0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
				011: 11-bit shift • When 8-bit access space is designated: Row address bits A23 to A11 used for comparison When 16-bit access space is designated: Row address bits A23 to A12 used for comparison Synchronous DRAM interface 100: 8-bit shift • When 8-bit access space is designated: Row address bits A23 to A8 used for comparison • When 16-bit access space is designated: Row address bits A23 to A9 used for comparison The precharge-sel is A15 to A9 of the column address. 101: 9-bit shift • When 8-bit access space is designated: Row address bits A23 to A9 used for comparison • When 16-bit access space is designated: Row address bits A23 to A10 used for comparison The precharge-sel is A15 to A10 of the column address. 110: 10-bit shift • When 8-bit access space is designated: Row address bits A23 to A10 used for comparison • When 16-bit access space is designated: Row address bits A23 to A11 used for comparison The precharge-sel is A15 to A11 of the column address.

Bit	Bit Name	Initial Value	R/W	Description
				111: 11-bit shift • When 8-bit access space is designated: Row address bits A23 to A11 used for comparison • When 16-bit access space is designated: Row address bits A23 to A12 used for comparison The precharge-sel is A15 to A12 of the column address.

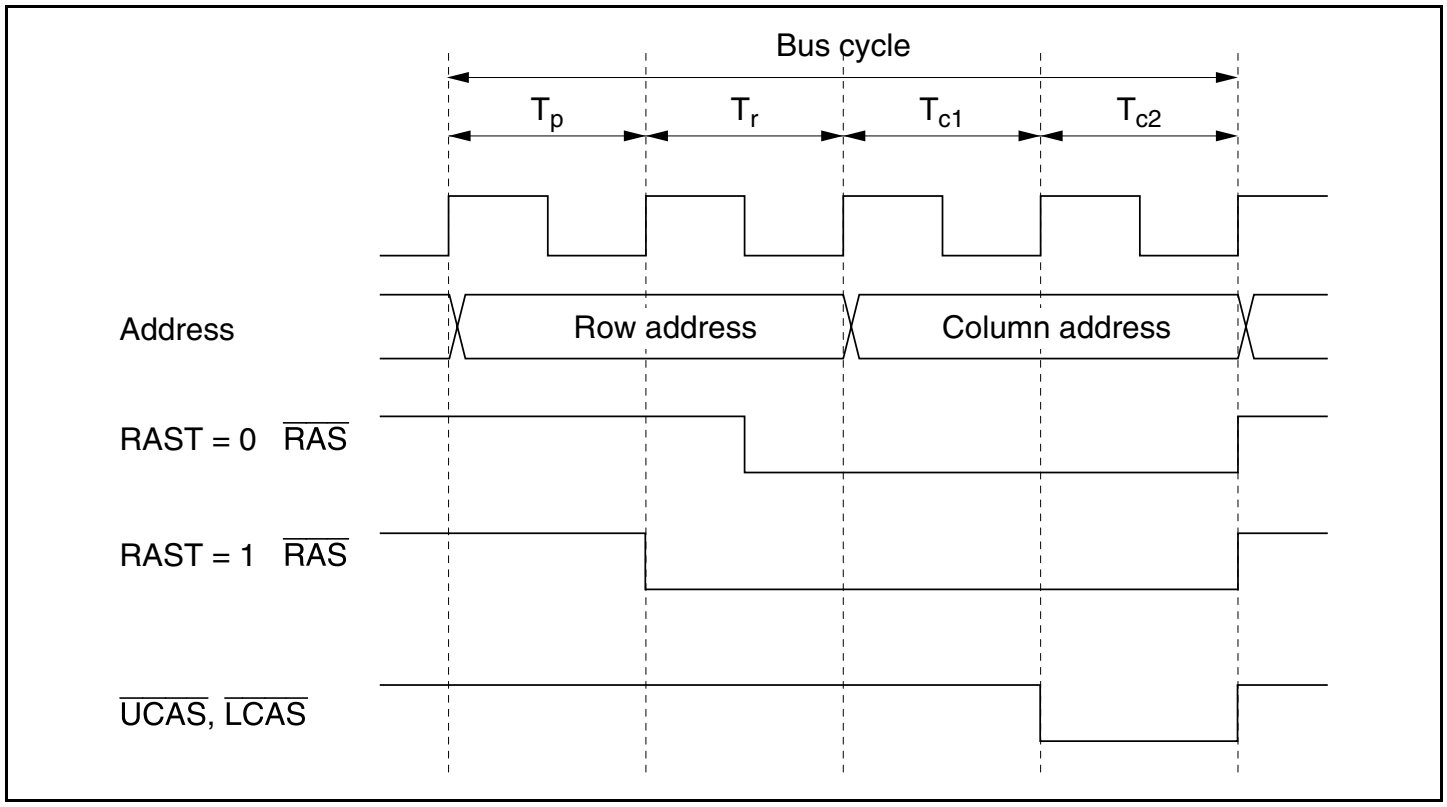


Figure 6.4 $\overline{RAS}$ Signal Assertion Timing (2-State Column Address Output Cycle, Full Access)

6.3.9 DRAM Access Control Register (DRACCR)

DRACCR is used to set the DRAM/synchronous DRAM interface bus specifications.

Note: The synchronous DRAM interface is not supported in the H8S/2678 Series.

- H8S/2678 Series

Bit	Bit Name	Initial Value	R/W	Description
7	DRMI	0	R/W	Idle Cycle Insertion An idle cycle can be inserted after a DRAM read cycle when a continuous normal space access cycle follows a DRAM read cycle. Idle cycle insertion conditions, setting of number of states, etc., comply with settings of bits ICIS1, ICIS0, and IDLC in BCR register 0: Idle cycle not inserted 1: Idle cycle inserted
6	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.
5	TPC1	0	R/W	Precharge State Control
4	TPC0	0	R/W	These bits select the number of states in the RAS precharge cycle in normal access and refreshing. 00: 1 state 01: 2 states 10: 3 states 11: 4 states
3	—	0	R/W	Reserved
2	—	0	R/W	These bits can be read from or written to. However, the write value should always be 0.
1	RCD1	0	R/W	RAS-CAS Wait Control
0	RCD0	0	R/W	These bits select a wait cycle to be inserted between the RAS assert cycle and CAS assert cycle. 00: Wait cycle not inserted 01: 1-state wait cycle inserted 10: 2-state wait cycle inserted 11: 3-state wait cycle inserted

Bit	Bit Name	Initial Value	R/W	Description
15	DRMI	0	R/W	Idle Cycle Insertion An idle cycle can be inserted after a DRAM/synchronous DRAM access cycle when a continuous normal space access cycle follows a DRAM/synchronous DRAM access cycle. Idle cycle insertion conditions, setting of number of states, etc., comply with settings of bits ICIS2, ICIS1, ICIS0, and IDLC in BCR register 0: Idle cycle not inserted 1: Idle cycle inserted
14	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.
13	TPC1	0	R/W	Precharge State Control
12	TPC0	0	R/W	These bits select the number of states in the RAS precharge cycle in normal access and refreshing. 00: 1 state 01: 2 states 10: 3 states 11: 4 states
11	SDWCD	0	R/W	CAS Latency Control Cycle Disabled during Continuous Synchronous DRAM Space Write Access Disables CAS latency control cycle (Tc1) inserted by WTCR settings during synchronous DRAM write access (see figure 6.5). 0: Enables CAS latency control cycle 1: Disables CAS latency control cycle
10	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	RCD1	0	R/W	RAS-CAS Wait Control
8	RCD0	0	R/W	These bits select a wait cycle to be inserted between the RAS assert cycle and CAS assert cycle. A 1- to 4-state wait cycle can be inserted. 00: Wait cycle not inserted 01: 1-state wait cycle inserted 10: 2-state wait cycle inserted 11: 3-state wait cycle inserted
7 to 4	—	0	R/W	Reserved These bits can be read from or written to. However, the write value should always be 0.
3	CKSPE	0	R/W	Clock Suspend Enable Enables clock suspend mode for extend read data during DMAC and EXDMAC single address transfer with the synchronous DRAM interface. 0: Disables clock suspend mode 1: Enables clock suspend mode
2	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.
1	RDXC1	0	R/W	Read Data Extension Cycle Number Selection
0	RDXC0	0	R/W	Selects the number of read data extension cycle (Tsp) insertion state in clock suspend mode. These bits are valid when the CKSPE bit is set to 1. 00: Inserts 1state 01: Inserts 2state 10: Inserts 3state 11: Inserts 4state

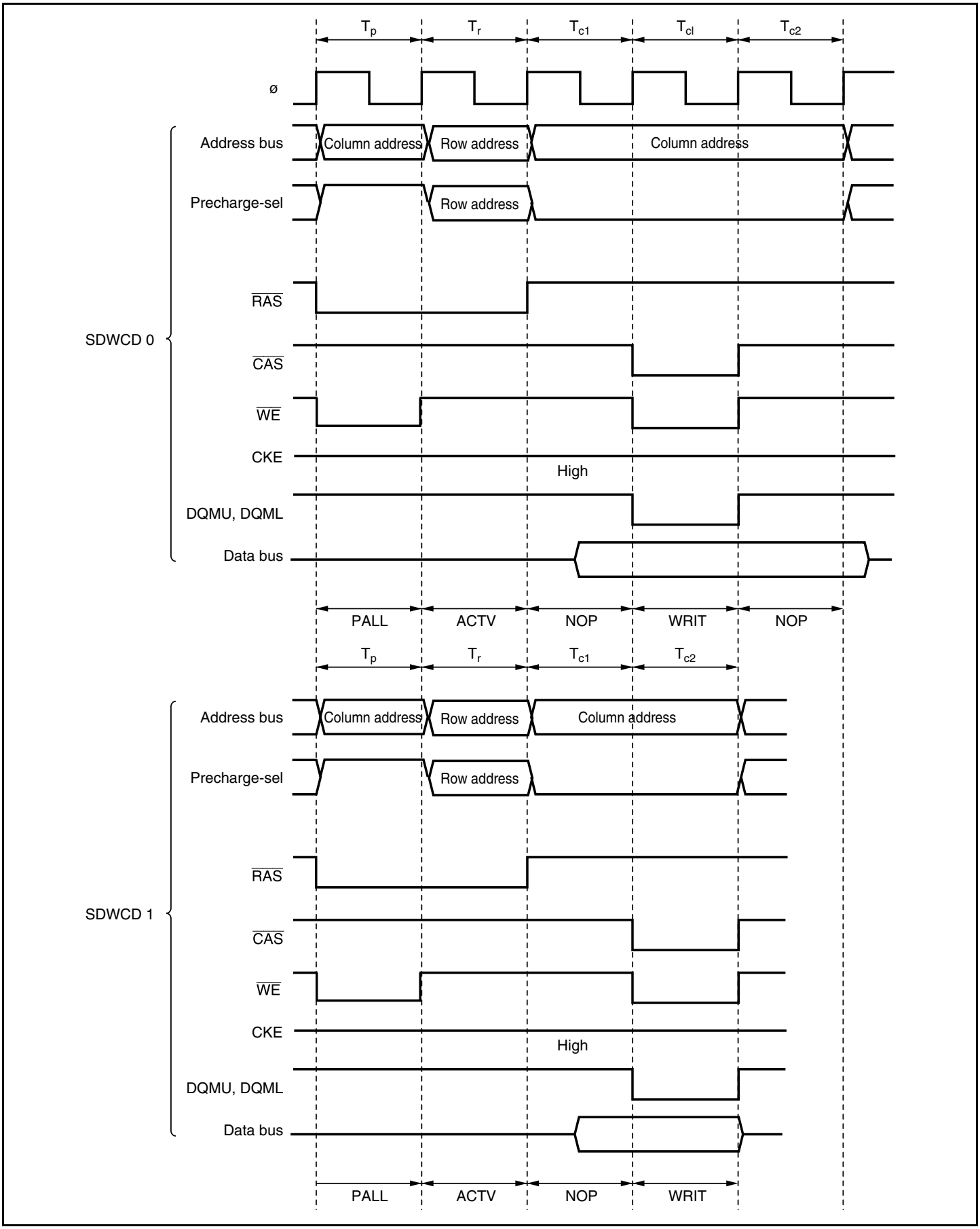


Figure 6.5 CAS Latency Control Cycle Disable Timing during Continuous Synchronous DRAM Space Write Access (for CAS Latency 2)

6.3.10 Refresh Control Register (REFCR)

REFCR specifies DRAM/synchronous DRAM interface refresh control.

Note: The synchronous DRAM interface is not supported in the H8S/2678 Series.

Bit	Bit Name	Initial Value	R/W	Description
15	CMF	0	R/(W)*	Compare Match Flag Status flag that indicates a match between the values of RTCNT and RTCOR. [Clearing conditions] • When 0 is written to CMF after reading CMF = 1 while the RFSHE bit is cleared to 0 • When CBR refreshing is executed while the RFSHE bit is set to 1 [Setting condition] When RTCOR = RTCNT
14	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables interrupt requests (CMI) by the CMF flag when the CMF flag is set to 1. This bit is valid when refresh control is not performed. When the refresh control is performed, this bit is always cleared to 0 and cannot be modified. 0: Interrupt request by CMF flag disabled 1: Interrupt request by CMF flag enabled
13	RCW1	0	R/W	CAS-RAS Wait Control These bits select the number of wait cycles to be inserted between the CAS assert cycle and RAS assert cycle in a DRAM/synchronous DRAM refresh cycle. 00: Wait state not inserted 01: 1 wait state inserted 10: 2 wait states inserted 11: 3 wait states inserted
12	RCW0	0	R/W	

Note: Only 0 can be written, to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
11	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.
10 9 8	RTCK2 RTCK1 RTCK0	0 0 0	R/W R/W R/W	Refresh Counter Clock Select These bits select the clock to be used to increment the refresh counter. When the input clock is selected with bits RTCK2 to RTCK0, the refresh counter begins counting up. 000: Count operation halted 001: Count on ø/2 010: Count on ø/8 011: Count on ø/32 100: Count on ø/128 101: Count on ø/512 110: Count on ø/2048 111: Count on ø/4096
7	RFSHE	0	R/W	Refresh Control Refresh control can be performed. When refresh control is not performed, the refresh timer can be used as an interval timer. 0: Refresh control is not performed 1: Refresh control is performed
6	CBRM	0	R/W	CBR Refresh Control Selects CBR refreshing performed in parallel with other external accesses, or execution of CBR refreshing alone. When the continuous synchronous DRAM space is set, this bit can be read/written, but the setting contents do not affect operations. 0: External access during CAS-before-RAS refreshing is enabled 1: External access during CAS-before-RAS refreshing is disabled

Bit	Bit Name	Initial Value	R/W	Description
5	RLW1	0	R/W	Refresh Cycle Wait Control
4	RLW0	0	R/W	These bits select the number of wait states to be inserted in a DRAM interface CAS-before-RAS refresh cycle/synchronous DRAM interface auto-refresh cycle. This setting applies to all areas designated as DRAM/continuous synchronous DRAM space. 00: No wait state inserted 01: 1 wait state inserted 10: 2 wait states inserted 11: 3 wait states inserted
3	SLFRF	0	R/W	Self-Refresh Enable If this bit is set to 1, DRAM/synchronous DRAM self-refresh mode is selected when a transition is made to the software standby state. This bit is valid when the RFSHE bit is set to 1, enabling refresh operations. It is cleared after recovery from software standby mode. 0: Self-refreshing is disabled 1: Self-refreshing is enabled
2	TPCS2	0	R/W	Self-Refresh Precharge Cycle Control
1	TPCS1	0	R/W	These bits select the number of states in the precharge cycle immediately after self-refreshing. The number of states in the precharge cycle immediately after self-refreshing are added to the number of states set by bits TPC1 and TPC0 in DRACCR. 000: [TPC set value] states 001: [TPC set value + 1] states 010: [TPC set value + 2] states 011: [TPC set value + 3] states 100: [TPC set value + 4] states 101: [TPC set value + 5] states 110: [TPC set value + 6] states 111: [TPC set value + 7] states
0	TPCS0	0	R/W	

6.3.11 Refresh Timer Counter (RTCNT)

RTCNT is an 8-bit readable/writable up-counter. RTCNT counts up using the internal clock selected by bits RTCK2 to RTCK0 in REFCR.

When RTCNT matches RTCOR (compare match), the CMF flag in REFCR is set to 1 and RTCNT is cleared to H'00. If the RFSHE bit in REFCR is set to 1 at this time, a refresh cycle is started. If the RFSHE bit is cleared to 0 and the CMIE bit in REFCR is set to 1, a compare match interrupt (CMI) is generated.

RTCNT is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

6.3.12 Refresh Time Constant Register (RTCOR)

RTCOR is an 8-bit readable/writable register that sets the period for compare match operations with RTCNT.

The values of RTCOR and RTCNT are constantly compared, and if they match, the CMF flag in REFCR is set to 1 and RTCNT is cleared to H'00.

RTCOR is initialized to H'FF by a reset and in hardware standby mode. It is not initialized in software standby mode.

6.4 Bus Control

6.4.1 Area Division

The bus controller divides the 16-Mbyte address space into eight areas, 0 to 7, in 2-Mbyte units, and performs bus control for external address space in area units. Chip select signals (CS0 to CS7) can be output for each area. In normal mode, a part of area 0, 64-kbyte address space, is controlled. Figure 6.6 shows an outline of the memory map.

Note: Normal mode is not available in this LSI.

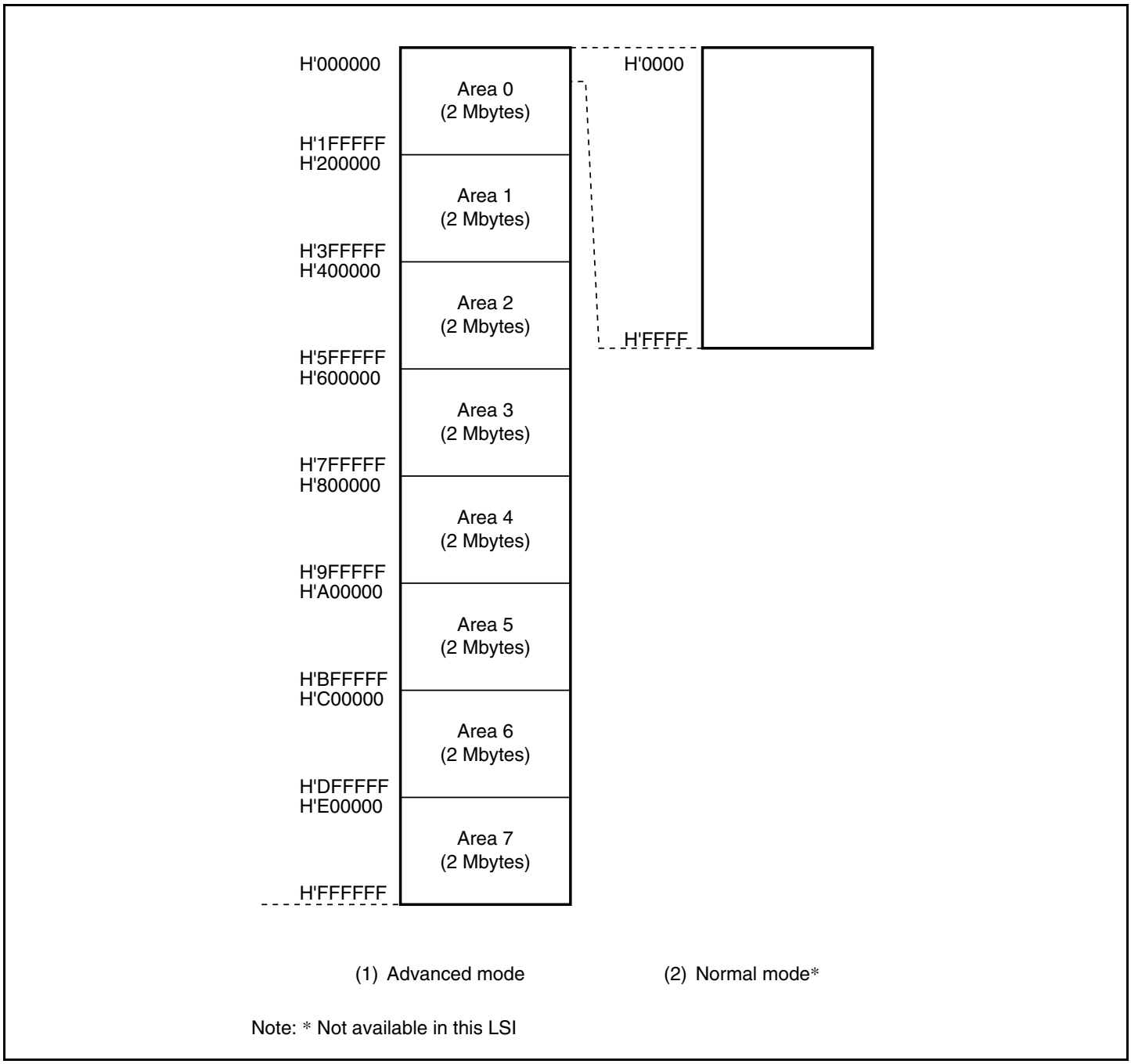


Figure 6.6 Area Divisions

6.4.2 Bus Specifications

The external address space bus specifications consist of five elements: bus width, number of access states, number of program wait states, read strobe timing, and chip select (CS) assertion period extension states. The bus width and number of access states for on-chip memory and internal I/O registers are fixed, and are not affected by the bus controller.

Bus Width: A bus width of 8 or 16 bits can be selected with ABWCR. An area for which an 8-bit bus is selected functions as an 8-bit access space, and an area for which a 16-bit bus is selected functions as a 16-bit access space. If all areas are designated as 8-bit access space, 8-bit bus mode is set; if any area is designated as 16-bit access space, 16-bit bus mode is set.

Number of Access States: Two or three access states can be selected with ASTCR. An area for which 2-state access is selected functions as a 2-state access space, and an area for which 3-state access is selected functions as a 3-state access space. With the DRAM or synchronous DRAM interface and burst ROM interface, the number of access states may be determined without regard to the setting of ASTCR.

When 2-state access space is designated, wait insertion is disabled. When 3-state access space is designated, it is possible to insert program waits by means of the WTCRA and WTCRB, and external waits by means of the WAIT pin.

Note: The synchronous DRAM interface is not supported in the H8S/2678 Series.

Number of Program Wait States: When 3-state access space is designated by ASTCR, the number of program wait states to be inserted automatically is selected with WTCRA and WTCRB. From 0 to 7 program wait states can be selected. Table 6.2 shows the bus specifications (bus width, and number of access states and program wait states) for each basic bus interface area.

Table 6.2 Bus Specifications for Each Area (Basic Bus Interface)

ABWCR	ASTCR	WTCRA, WTCRB			Bus Specifications (Basic Bus Interface)		
ABWn	ASTn	Wn2	Wn1	Wn0	Bus Width	Access States	Program Wait States
0	0	—	—	—	16	2	0
	1	0	0	0		3	0
				1			1
			1	0			2
				1			3
		1	0	0			4
				1			5
			1	0			6
				1			7
1	0	—	—	—	8	2	0
	1	0	0	0		3	0
				1			1
			1	0			2
				1			3
		1	0	0			4
				1			5
			1	0			6
				1			7

(n = 0 to 7)

Read Strobe Timing: RDNCR can be used to select either of two negation timings (at the end of the read cycle or one half-state before the end of the read cycle) for the read strobe (RD) used in the basic bus interface space.

Chip Select (CS) Assertion Period Extension States: Some external I/O devices require a setup time and hold time between address and CS signals and strobe signals such as RD, HWR, and LWR. CSACR can be used to insert states in which only the CS, AS, and address signals are asserted before and after a basic bus space access cycle.

6.4.3 Memory Interfaces

The memory interfaces in this LSI comprise a basic bus interface that allows direct connection of ROM, SRAM, and so on; a DRAM interface that allows direct connection of DRAM; a synchronous DRAM interface* that allows direct connection of synchronous DRAM; and a burst

ROM interface that allows direct connection of burst ROM. The interface can be selected independently for each area.

An area for which the basic bus interface is designated functions as normal space, an area for which the DRAM interface is designated functions as DRAM space, an area for which the synchronous DRAM interface is designated functions as continuous synchronous DRAM space, and an area for which the burst ROM interface is designated functions as burst ROM space.

The initial state of each area is basic bus interface, 3-state access space. The initial bus width is selected according to the operating mode.

Note: The synchronous DRAM interface is not supported in the H8S/2678 Series.

Area 0: Area 0 includes on-chip ROM in expanded mode with on-chip ROM enabled and the space excluding on-chip ROM is external address space, and in expanded mode with on-chip ROM disabled, all of area 0 is external address space.

When area 0 external space is accessed, the CS0 signal can be output.

Either basic bus interface or burst ROM interface can be selected for area 0.

Area 1: In externally expanded mode, all of area 1 is external address space.

When area 1 external address space is accessed, the CS1 signal can be output.

Either basic bus interface or burst ROM interface can be selected for area 1.

Areas 2 to 5: In externally expanded mode, areas 2 to 5 are all external address space.

When area 2 to 5 external space is accessed, signals CS2 to CS5 can be output.

Basic bus interface, DRAM interface, or synchronous DRAM interface can be selected for areas 2 to 5. With the DRAM interface, signals CS2 to CS5 are used as RAS signals.

If areas 2 to 5 are designated as continuous DRAM space, large-capacity (e.g. 64-Mbit) DRAM can be connected. In this case, the CS2 signal is used as the RAS signal for the continuous DRAM space.

If areas 2 to 5 are designated as continuous synchronous DRAM space, large-capacity (e.g. 64-Mbit) synchronous DRAM can be connected. In this case, the CS2, CS3, CS4, and CS5 pins are used as the RAS, CAS, WE, and CLK signals for the continuous synchronous DRAM space. The OE pin is used as the CKE signal.

Area 6: In externally expanded mode, all of area 6 is external space.

When area 6 external space is accessed, the CS6 signal can be output.

Only the basic bus interface can be used for area 6.

Area 7: Area 7 includes the on-chip RAM and internal/O registers. In externally expanded mode, the space excluding the on-chip RAM and internal I/O registers is external address space. The on-chip RAM is enabled when the RAME bit is set to 1 in the system control register (SYSCR); when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding addresses are in external address space.

When area 7 external address space is accessed, the CS7 signal can be output.

Only the basic bus interface can be used for the area 7 memory interface.

6.4.4 Chip Select Signals

This LSI can output chip select signals (CS0 to CS7) for areas 0 to 7. The signal outputs low when the corresponding external space area is accessed. Figure 6.7 shows an example of CS0 to CS7 signals output timing.

Enabling or disabling of CS0 to CS7 signals output is set by the data direction register (DDR) bit for the port corresponding to the CS0 to CS7 pins.

In expanded mode with on-chip ROM disabled, the CS0 pin is placed in the output state after a reset. Pins CS1 to CS7 are placed in the input state after a reset and so the corresponding DDR bits should be set to 1 when outputting signals CS1 to CS7.

In expanded mode with on-chip ROM enabled, pins CS0 to CS7 are all placed in the input state after a reset and so the corresponding DDR bits should be set to 1 when outputting signals CS0 to CS7.

When areas 2 to 5 are designated as DRAM space, outputs CS2 to CS5 are used as RAS signals.

When areas 2 to 5 are designated as continuous synchronous DRAM space in the H8S/2678R Series, outputs CS2, CS3, CS4, and CS5 are used as RAS, CAS, WE, and CLK signals.

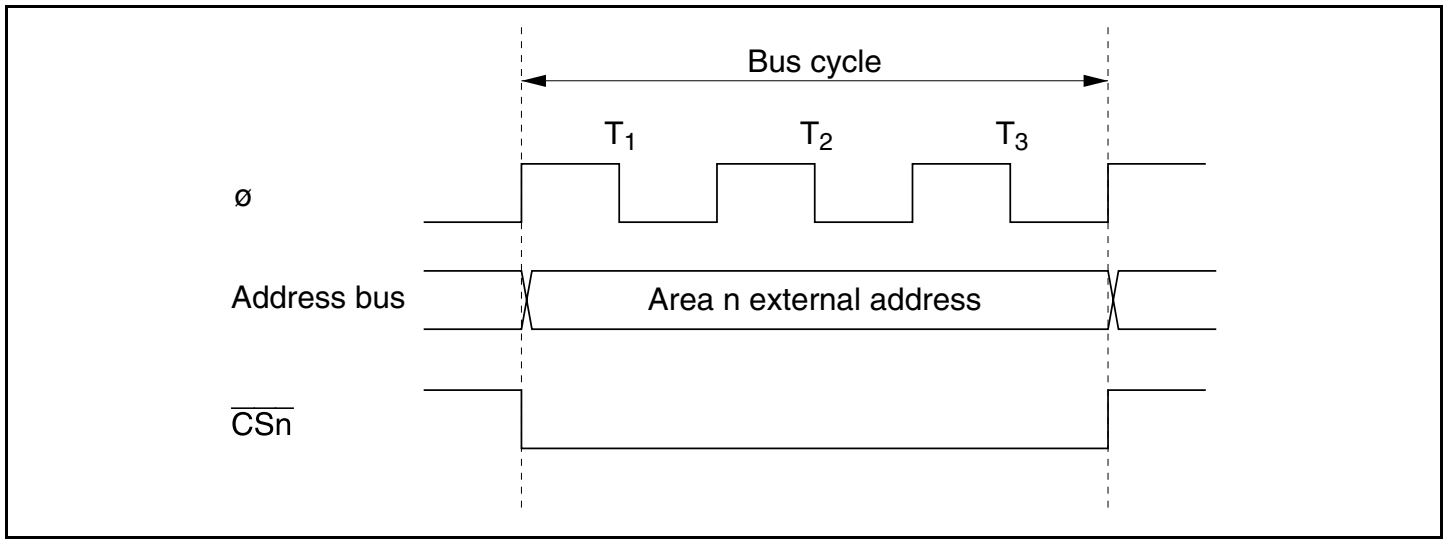


Figure 6.7 CSn Signal Output Timing (n = 0 to 7)

6.5 Basic Bus Interface

The basic bus interface enables direct connection of ROM, SRAM, and so on.

6.5.1 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. The bus controller has a data alignment function, and when accessing external address space, controls whether the upper data bus (D15 to D8) or lower data bus (D7 to D0) is used according to the bus specifications for the area being accessed (8-bit access space or 16-bit access space) and the data size.

8-Bit Access Space: Figure 6.8 illustrates data alignment control for the 8-bit access space. With the 8-bit access space, the upper data bus (D15 to D8) is always used for accesses. The amount of data that can be accessed at one time is one byte: a word access is performed as two byte accesses, and a longword access, as four byte accesses.

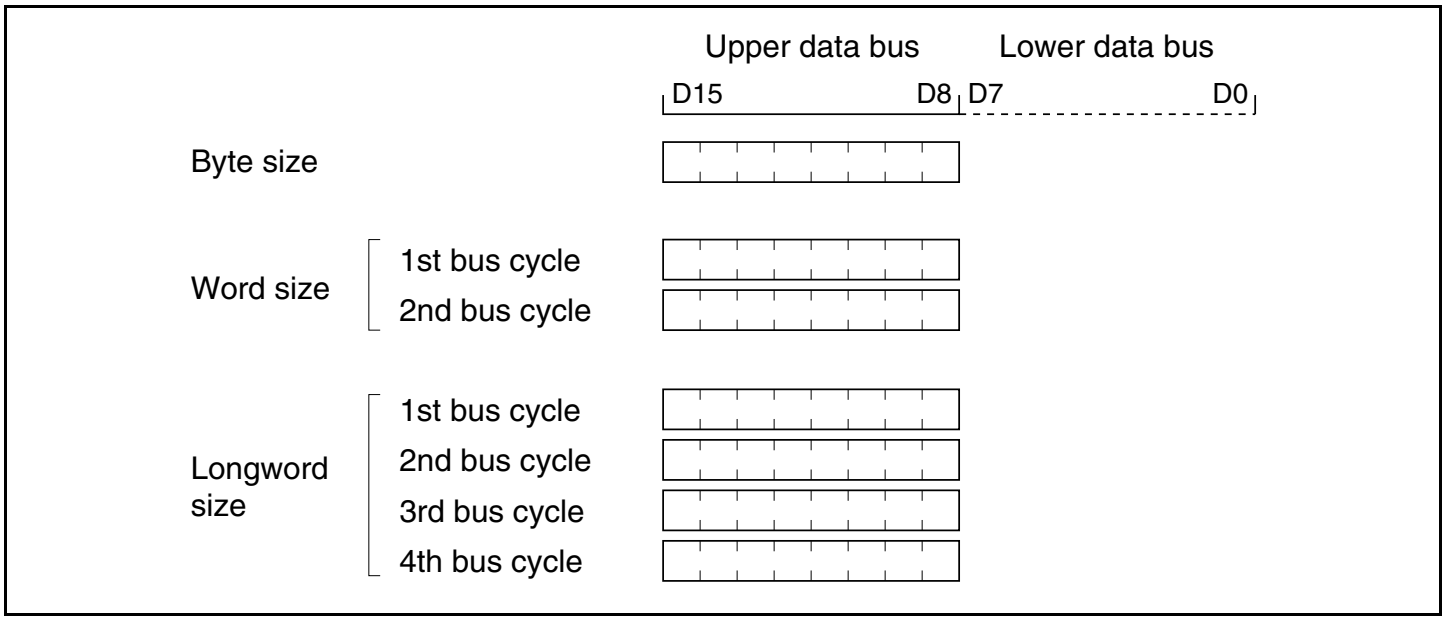


Figure 6.8 Access Sizes and Data Alignment Control (8-Bit Access Space)

16-Bit Access Space: Figure 6.9 illustrates data alignment control for the 16-bit access space. With the 16-bit access space, the upper data bus (D15 to D8) and lower data bus (D7 to D0) are used for accesses. The amount of data that can be accessed at one time is one byte or one word, and a longword access is executed as two word accesses.

In byte access, whether the upper or lower data bus is used is determined by whether the address is even or odd. The upper data bus is used for an even address, and the lower data bus for an odd address.

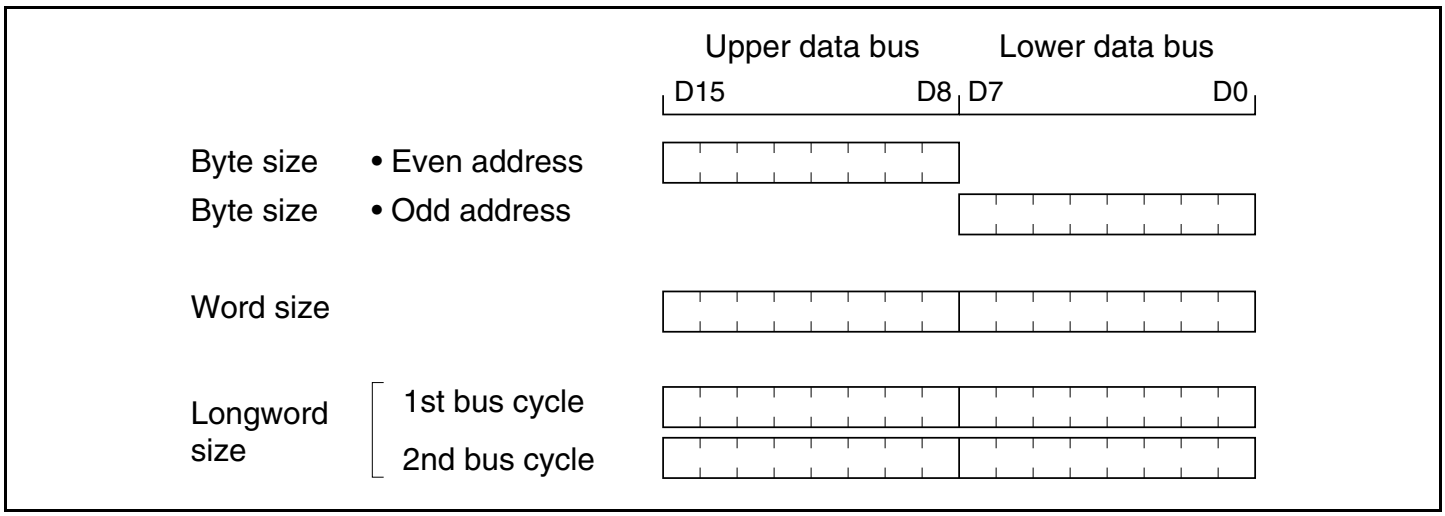


Figure 6.9 Access Sizes and Data Alignment Control (16-bit Access Space)

6.5.2 Valid Strobes

Table 6.3 shows the data buses used and valid strobes for the access spaces.

In a read, the RD signal is valid for both the upper and the lower half of the data bus. In a write, the HWR signal is valid for the upper half of the data bus, and the LWR signal for the lower half.

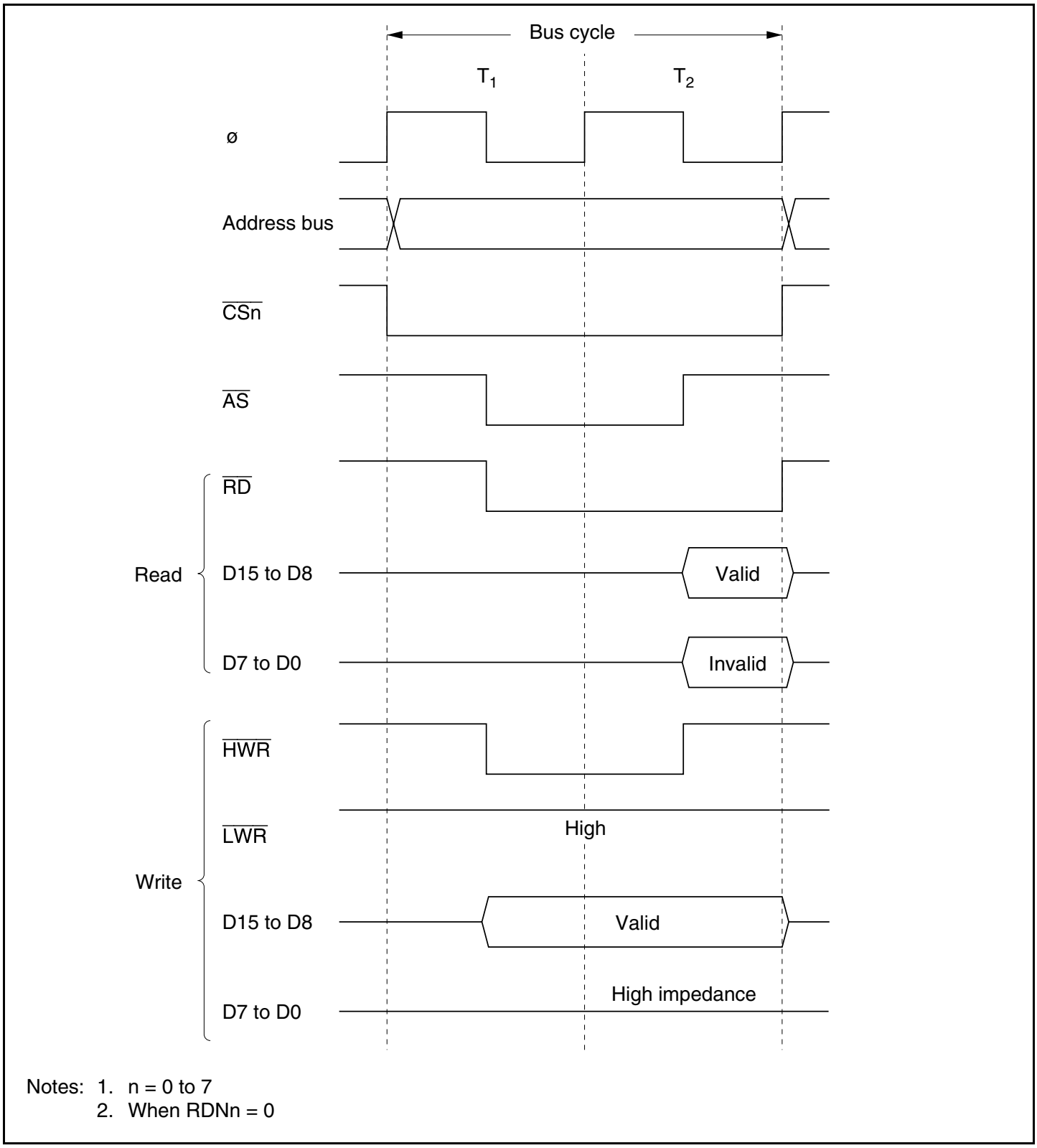
Table 6.3 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D15 to D8)	Lower Data Bus (D7 to D0)
8-bit access space	Byte	Read	—	RD	Valid	Invalid
		Write	—	HWR		Hi-Z
16-bit access space	Byte	Read	Even	RD	Valid	Invalid
			Odd		Invalid	Valid
		Write	Even	HWR	Valid	Hi-Z
			Odd	LWR	Hi-Z	Valid
	Word	Read	—	RD	Valid	Valid
		Write	—	HWR, LWR	Valid	Valid

Note: Hi-Z: High-impedance state
Invalid: Input state; input value is ignored.

6.5.3 Basic Operation Timing

8-Bit, 2-State Access Space: Figure 6.10 shows the bus timing for an 8-bit, 2-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used. The LWR pin is always fixed high. Wait states can be inserted.



Notes: 1. n = 0 to 7
2. When RDNn = 0

Figure 6.10 Bus Timing for 8-Bit, 2-State Access Space

8-Bit, 3-State Access Space: Figure 6.11 shows the bus timing for an 8-bit, 3-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used. The LWR pin is always fixed high. Wait states can be inserted.

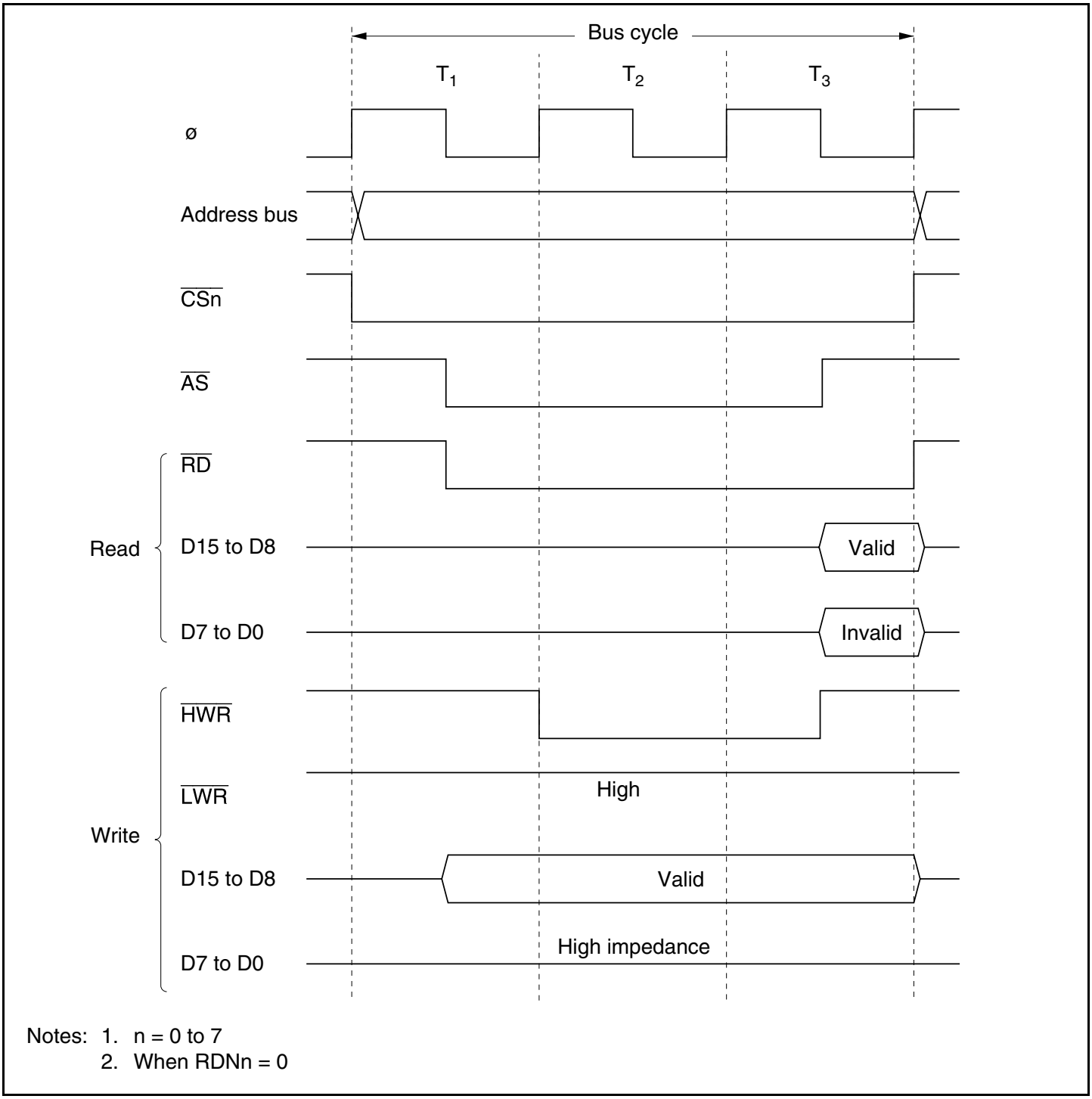
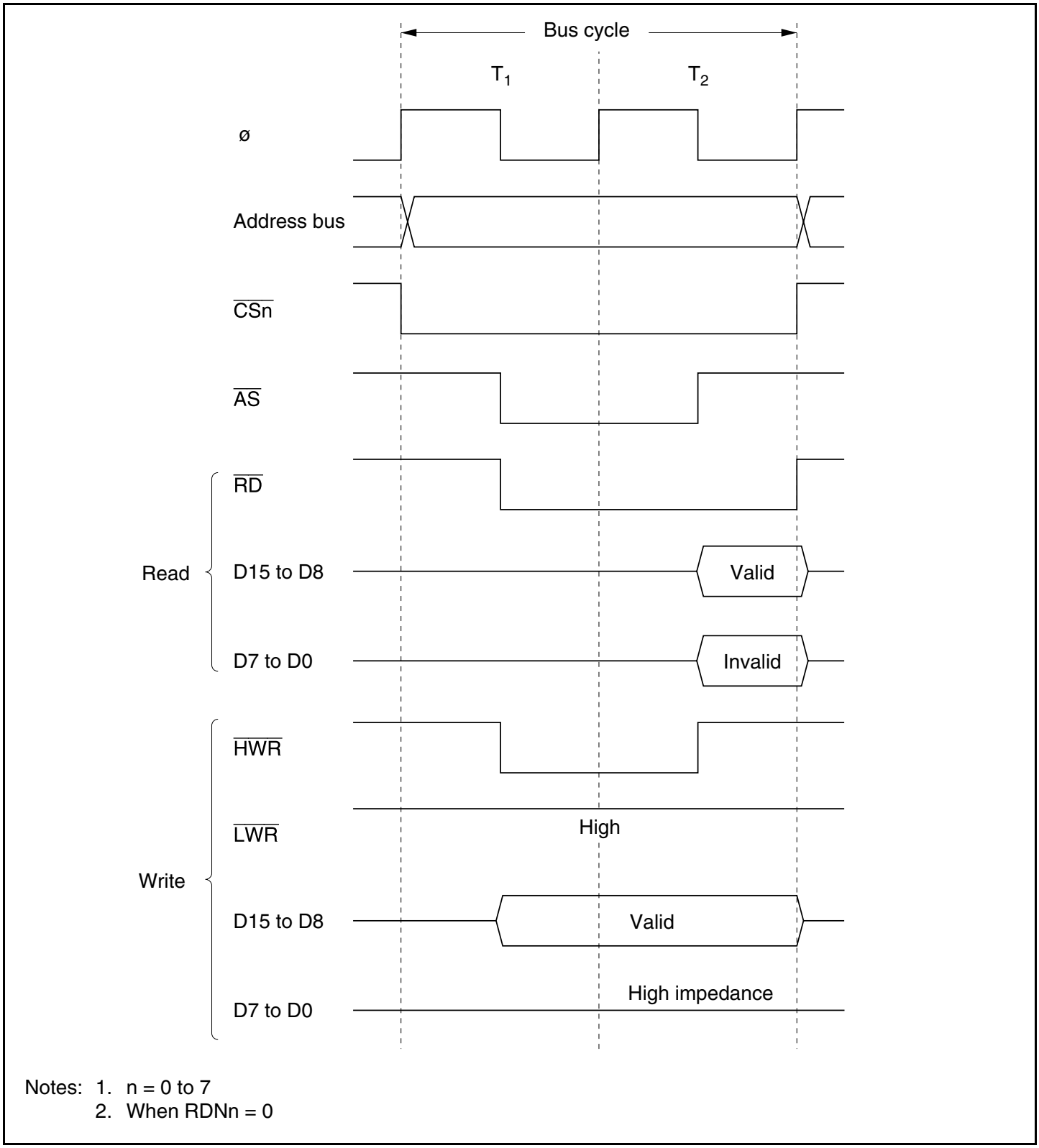


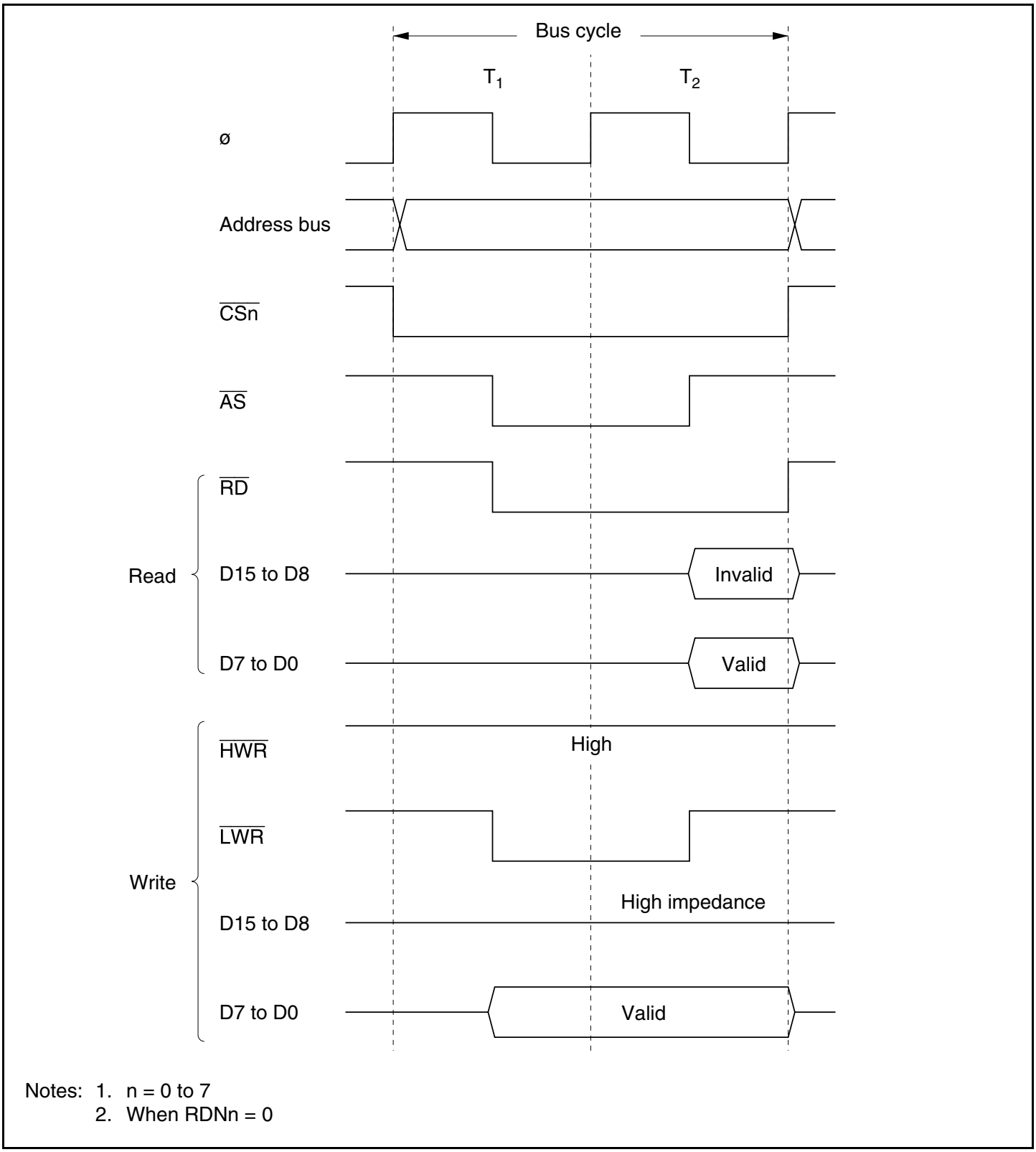
Figure 6.11 Bus Timing for 8-Bit, 3-State Access Space

16-Bit, 2-State Access Space: Figures 6.12 to 6.14 show bus timings for a 16-bit, 2-state access space. When a 16-bit access space is accessed, the upper half (D15 to D8) of the data bus is used for odd addresses, and the lower half (D7 to D0) for even addresses. Wait states cannot be inserted.



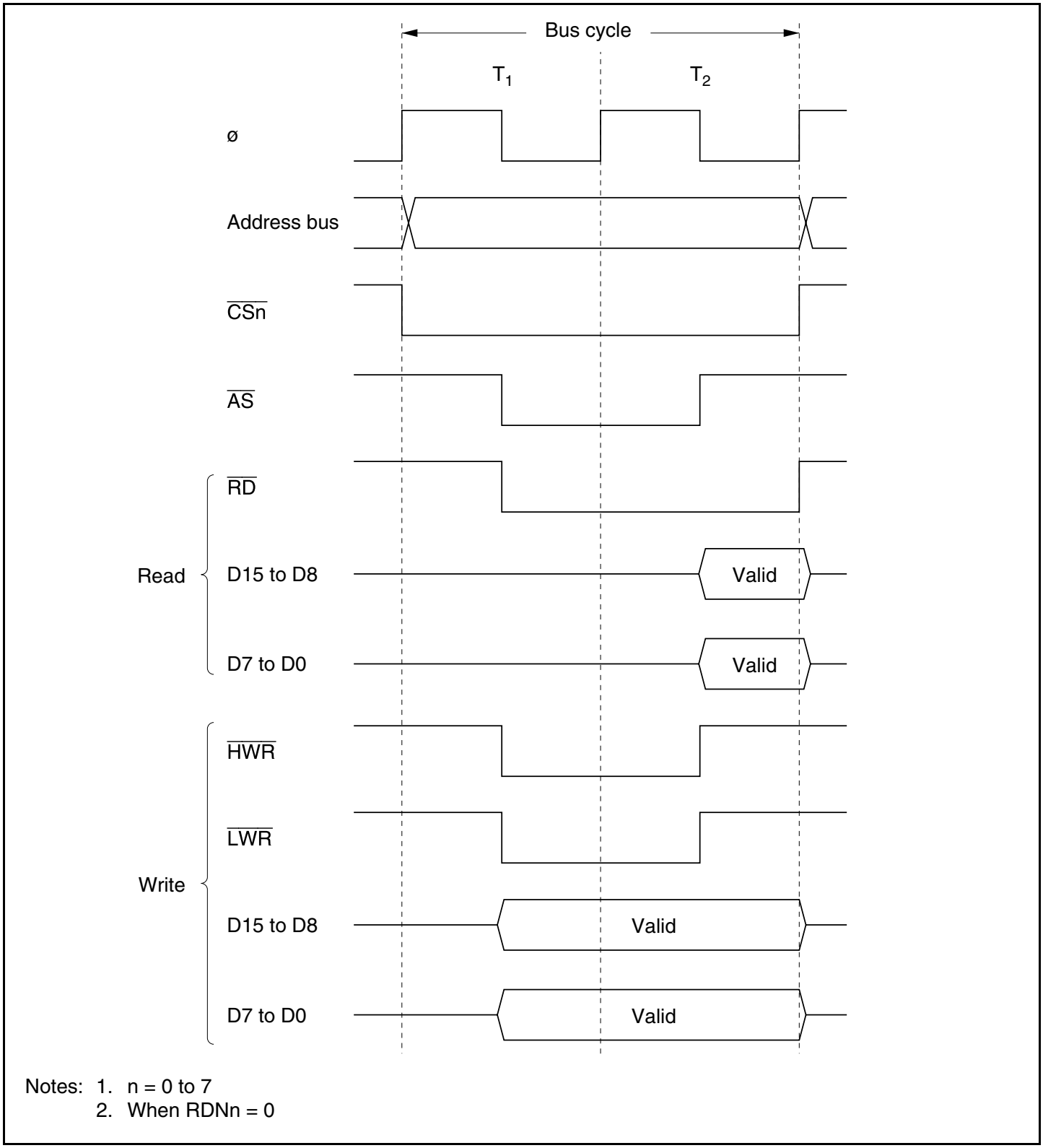
Notes: 1. n = 0 to 7
2. When RDNn = 0

Figure 6.12 Bus Timing for 16-Bit, 2-State Access Space (Even Address Byte Access)



Notes: 1. n = 0 to 7
2. When RDNn = 0

Figure 6.13 Bus Timing for 16-Bit, 2-State Access Space (Odd Address Byte Access)



Notes: 1. n = 0 to 7
2. When RDNn = 0

Figure 6.14 Bus Timing for 16-Bit, 2-State Access Space (Word Access)

16-Bit, 3-State Access Space: Figures 6.15 to 6.17 show bus timings for a 16-bit, 3-state access space. When a 16-bit access space is accessed, the upper half (D15 to D8) of the data bus is used for the even address, and the lower half (D7 to D0) for the odd address. Wait states can be inserted.

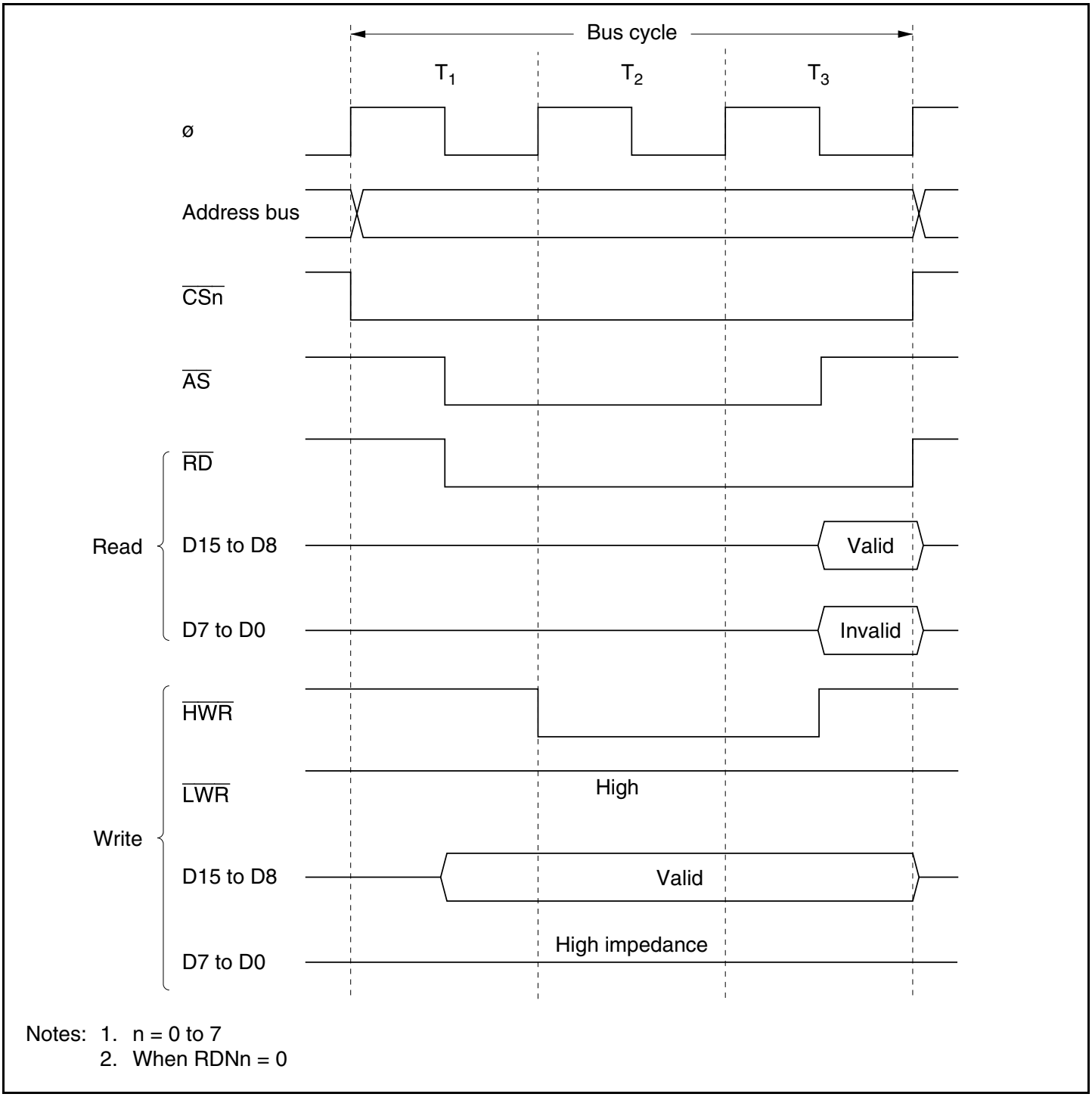
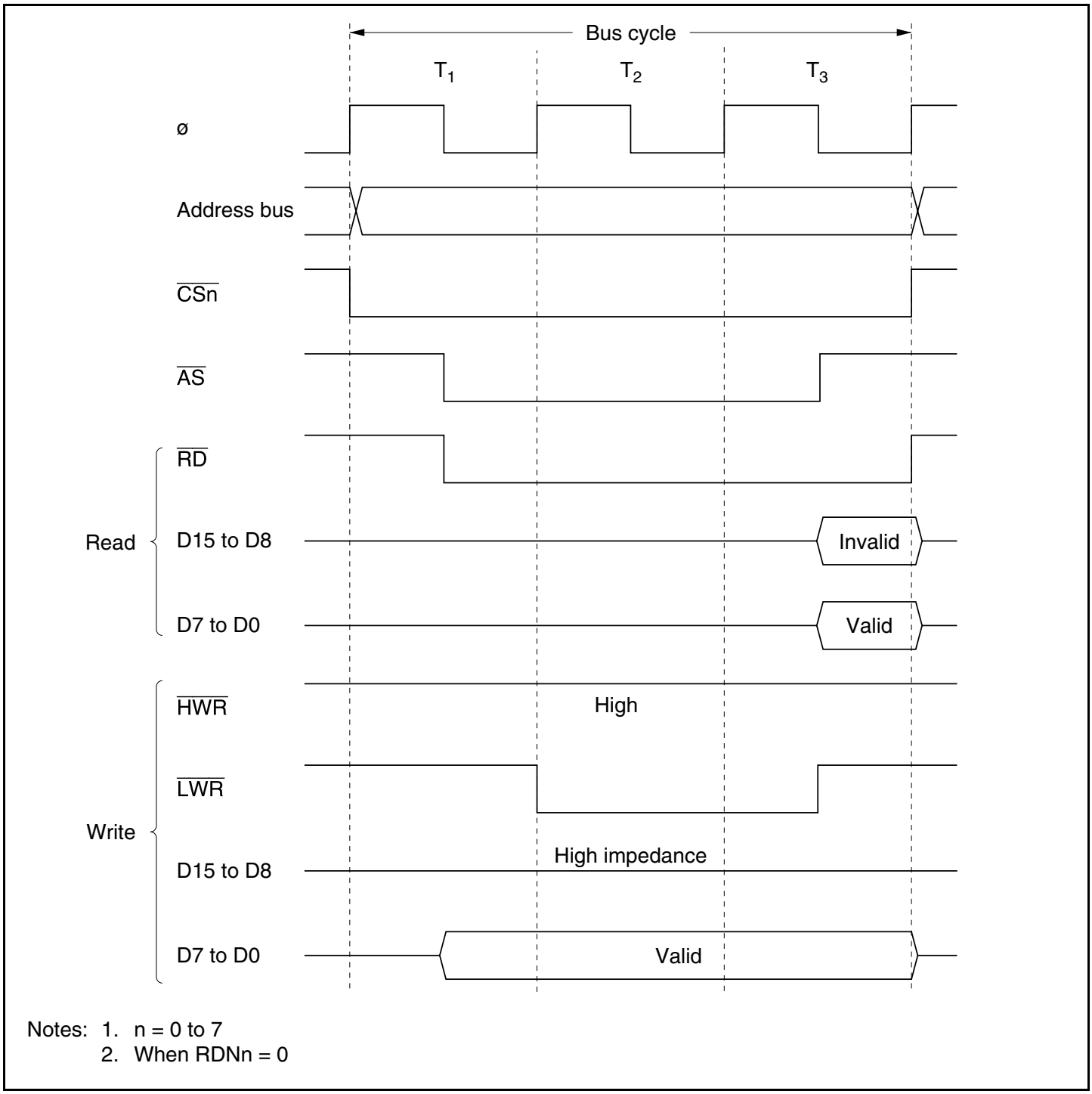


Figure 6.15 Bus Timing for 16-Bit, 3-State Access Space (Even Address Byte Access)



Notes: 1. n = 0 to 7
2. When RDNn = 0

Figure 6.16 Bus Timing for 16-Bit, 3-State Access Space (Odd Address Byte Access)

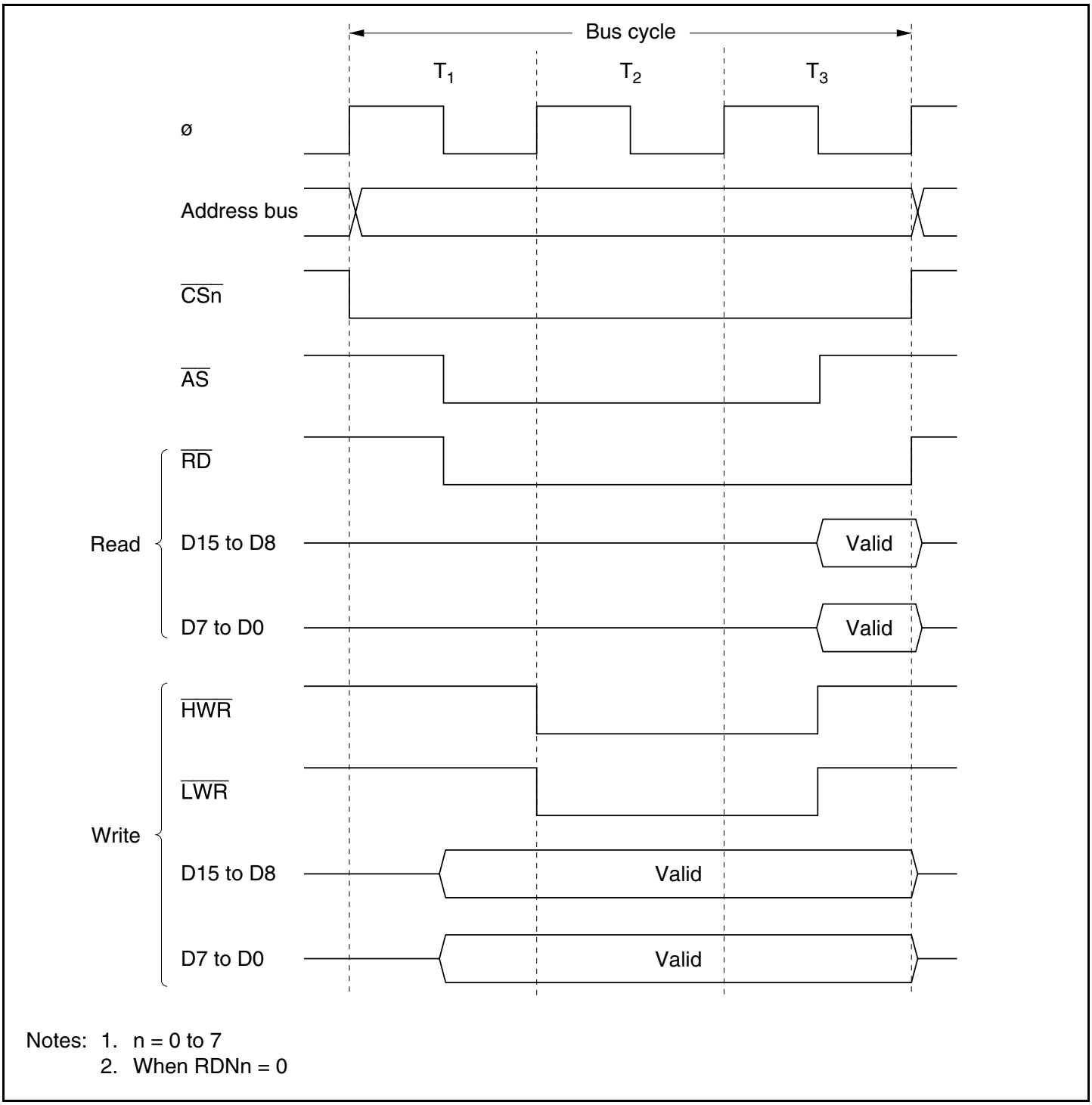


Figure 6.17 Bus Timing for 16-Bit, 3-State Access Space (Word Access)

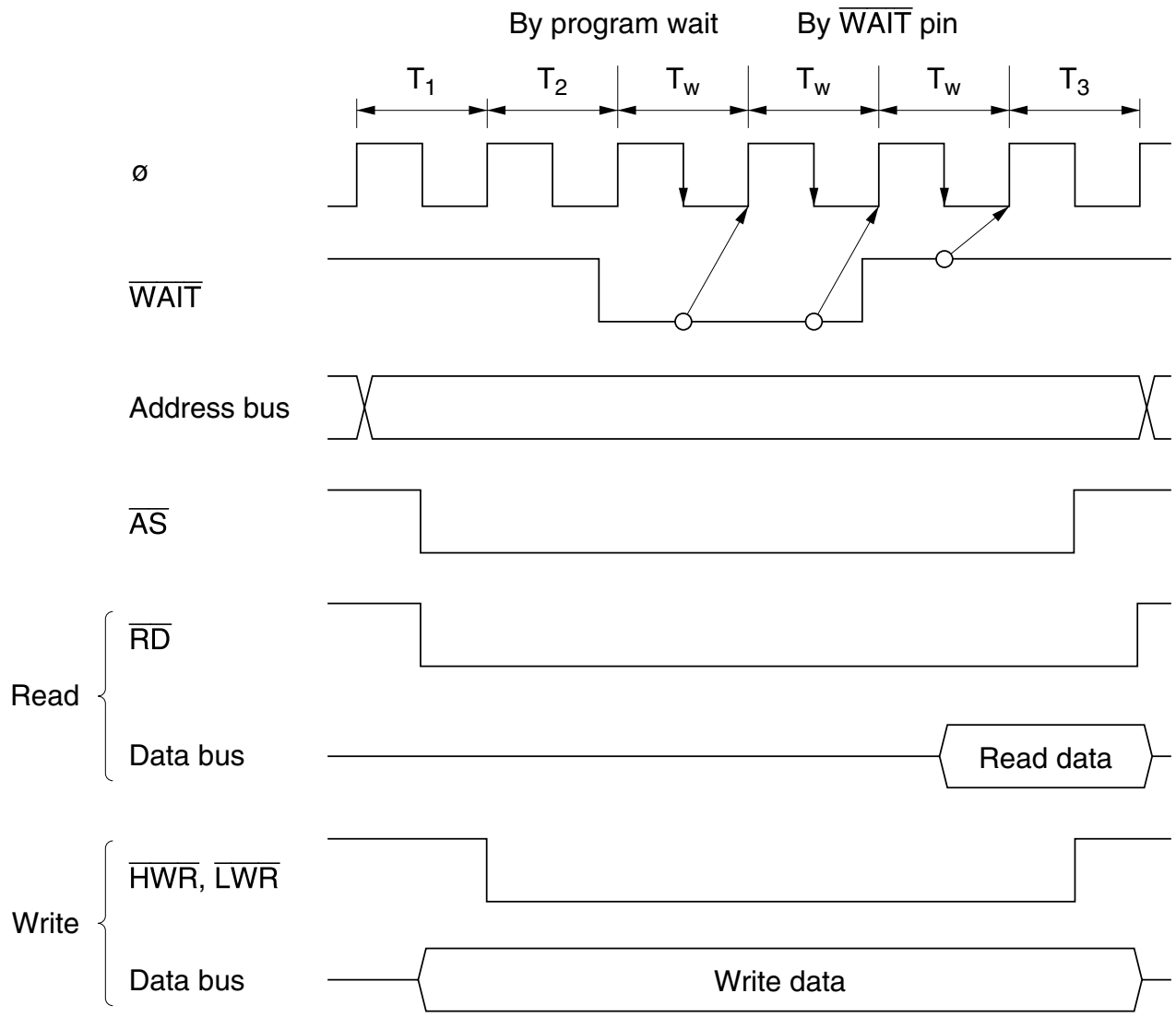
6.5.4 Wait Control

When accessing external space, this LSI can extend the bus cycle by inserting one or more wait states (T_w). There are two ways of inserting wait states: program wait insertion and pin wait insertion using the WAIT pin.

Program Wait Insertion: From 0 to 7 wait states can be inserted automatically between the T_2 state and T_3 state on an individual area basis in 3-state access space, according to the settings in WTCRA and WTCRB.

Pin Wait Insertion: Setting the WAITE bit to 1 in BCR enables wait input by means of the WAIT pin. When external space is accessed in this state, a program wait is first inserted in accordance with the settings in WTCRA and WTCRB. If the WAIT pin is low at the falling edge of ø in the last T_2 or T_w state, another T_w state is inserted. If the WAIT pin is held low, T_w states are inserted until it goes high. This is useful when inserting seven or more T_w states, or when changing the number of T_w states to be inserted for different external devices. The WAITE bit setting applies to all areas. Figure 6.18 shows an example of wait state insertion timing.

The settings after a reset are: 3-state access, insertion of 7 program wait states, and WAIT input disabled.



Notes: 1. Downward arrows indicate the timing of $\overline{WAIT}$ pin sampling.
2. When RDN = 0

Figure 6.18 Example of Wait State Insertion Timing

6.5.5 Read Strobe (RD) Timing

The read strobe (RD) timing can be changed for individual areas by setting bits RDN7 to RDN0 to 1 in RDNCR. Figure 6.19 shows an example of the timing when the read strobe timing is changed in basic bus 3-state access space.

When the DMAC or EXDMAC is used in single address mode, note that if the RD timing is changed by setting RDNn to 1, the RD timing will change relative to the rise of DACK or EDACK.

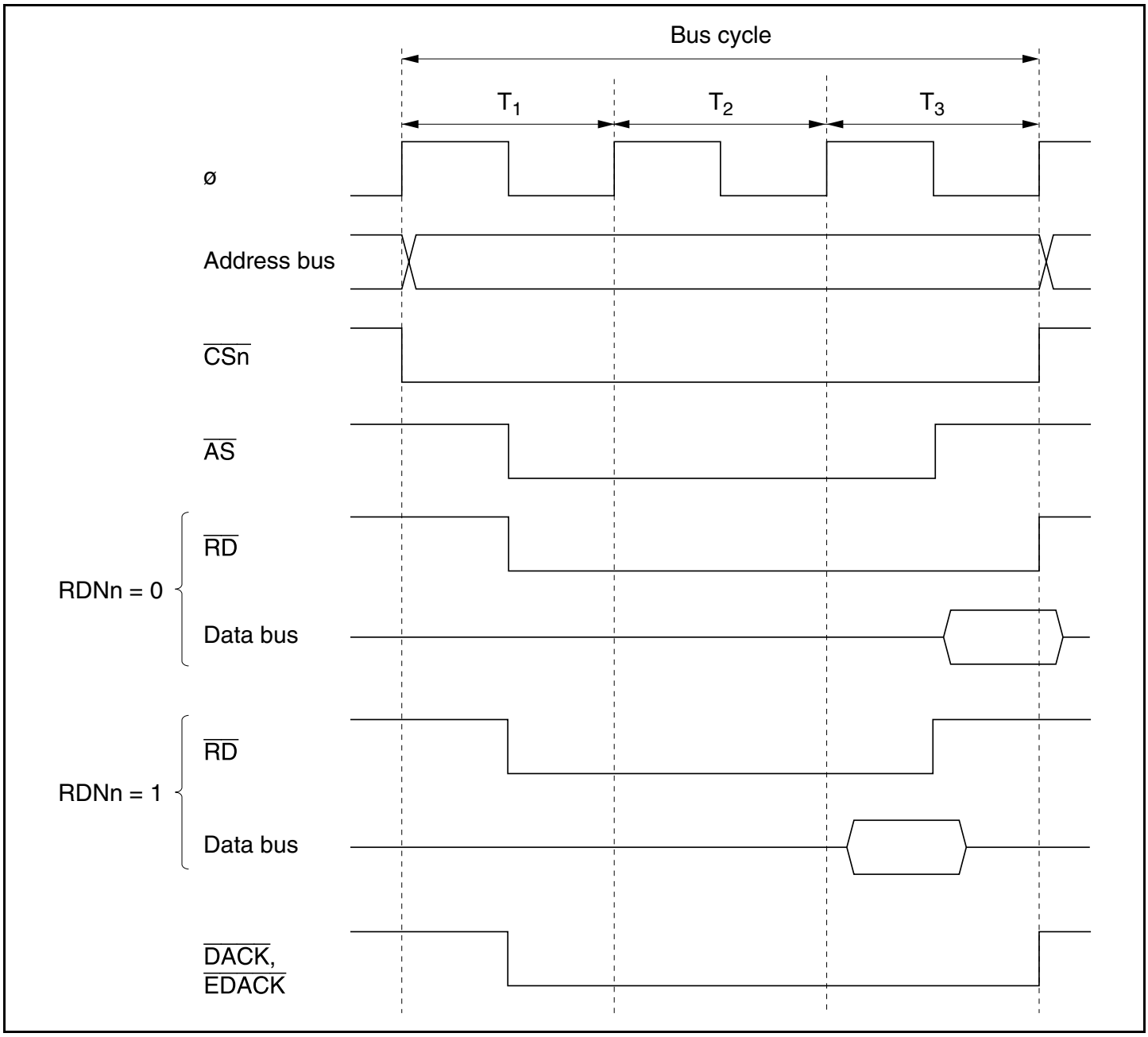


Figure 6.19 Example of Read Strobe Timing

6.5.6 Extension of Chip Select (CS) Assertion Period

Some external I/O devices require a setup time and hold time between address and CS signals and strobe signals such as RD, HWR, and LWR. Settings can be made in the CSACR register to insert states in which only the CS, AS, and address signals are asserted before and after a basic bus space access cycle. Extension of the CS assertion period can be set for individual areas. With the CS assertion extension period in write access, the data setup and hold times are less stringent since the write data is output to the data bus.

Figure 6.20 shows an example of the timing when the CS assertion period is extended in basic bus 3-state access space.

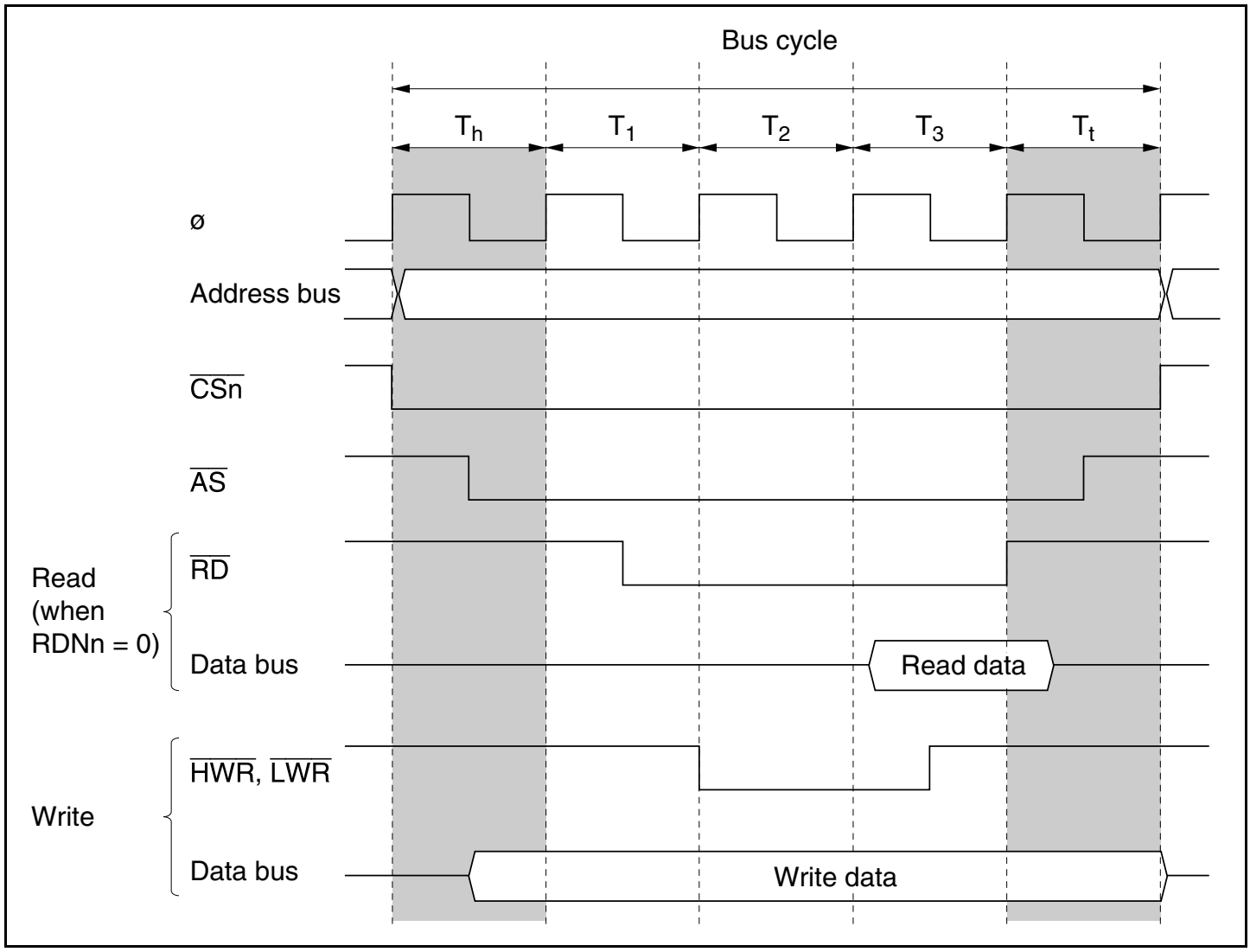


Figure 6.20 Example of Timing when Chip Select Assertion Period is Extended

Both extension state T_h inserted before the basic bus cycle and extension state T_t inserted after the basic bus cycle, or only one of these, can be specified for individual areas. Insertion or non-insertion can be specified for the T_h state with the upper 8 bits (CSXH7 to CSXH0) in the CSACR register, and for the T_t state with the lower 8 bits (CSXT7 to CSXT0).

6.6 DRAM Interface

In this LSI, external space areas 2 to 5 can be designated as DRAM space, and DRAM interfacing performed. The DRAM interface allows DRAM to be directly connected to this LSI. A DRAM space of 2, 4, or 8 Mbytes can be set by means of bits RMTS2 to RMTS0 in DRAMCR. Burst operation is also possible, using fast page mode.

6.6.1 Setting DRAM Space

Areas 2 to 5 are designated as DRAM space by setting bits RMTS2 to RMTS0 in DRAMCR. The relation between the settings of bits RMTS2 to RMTS0 and DRAM space is shown in table 6.4.

Possible DRAM space settings are: one area (area 2), two areas (areas 2 and 3), four areas (areas 2 to 5), and continuous area (areas 2 to 5).

Table 6.4 Relation between Settings of Bits RMTS2 to RMTS0 and DRAM Space

RMTS2	RMTS1	RMTS0	Area 5	Area 4	Area 3	Area 2
0	0	1	Normal space	Normal space	Normal space	DRAM space
	1	0	Normal space	Normal space	DRAM space	DRAM space
		1	DRAM space	DRAM space	DRAM space	DRAM space
1	0	0	Continuous synchronous DRAM space*			
		1	Mode register settings of synchronous DRAM*			
	1	0	Reserved (setting prohibited)			
		1	Continuous DRAM space	Continuous DRAM space	Continuous DRAM space	Continuous DRAM space

Note: Reserved (setting prohibited) in the H8S/2678 Series.

With continuous DRAM space, RAS2 is valid. The bus specifications (bus width, number of wait states, etc.) for continuous DRAM space conform to the settings for area 2.

6.6.2 Address Multiplexing

With DRAM space, the row address and column address are multiplexed. In address multiplexing, the size of the shift of the row address is selected with bits MXC2 to MXC0 in DRAMCR. Table 6.5 shows the relation between the settings of MXC2 to MXC0 and the shift size.

The MXC2 bit should be cleared to 0 when the DRAM interface is used.

Table 6.5 Relation between Settings of Bits MXC2 to MXC0 and Address Multiplexing

	DRAMCR				Address Pins															
	MXC2	MXC1	MXC0	Shift Size	A23 to A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Row address	0	0	0	8 bits	A23 to A16	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8
			1	9 bits	A23 to A16	A15	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9
		1	0	10 bits	A23 to A16	A15	A14	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			1	11 bits	A23 to A16	A15	A14	A13	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
	1	X	X		Reserved (setting prohibited)																
Column address	0*	X	X	—	A23 to A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
	1*	X	X		Reserved (setting prohibited)																

Note: In the H8S/2678 Series, address pins are A23 to A0. x: Don't care.

6.6.3 Data Bus

If a bit in ABWCR corresponding to an area designated as DRAM space is set to 1, that area is designated as 8-bit DRAM space; if the bit is cleared to 0, the area is designated as 16-bit DRAM space. In 16-bit DRAM space, ×16-bit configuration DRAM can be connected directly.

In 8-bit DRAM space the upper half of the data bus, D15 to D8, is enabled, while in 16-bit DRAM space both the upper and lower halves of the data bus, D15 to D0, are enabled.

Access sizes and data alignment are the same as for the basic bus interface: see section 6.5.1, Data Size and Data Alignment.

6.6.4 Pins Used for DRAM Interface

Table 6.6 shows the pins used for DRAM interfacing and their functions. Since the CS2 to CS5 pins are in the input state after a reset, set the corresponding DDR to 1 when RAS2 to RAS5 signals are output.

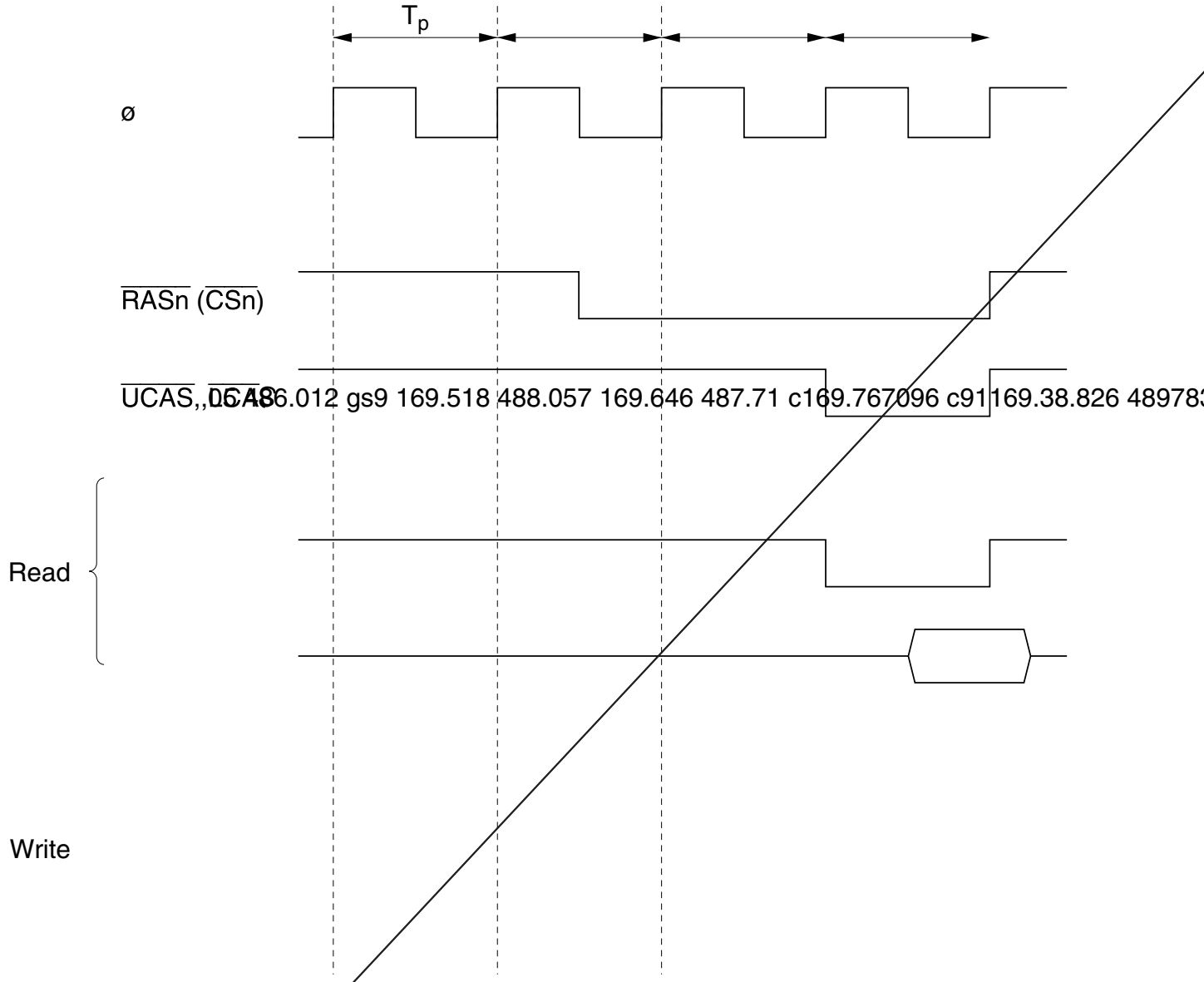
Table 6.6 DRAM Interface Pins

Pin	With DRAM Setting	Name	I/O	Function
HWR	WE	Write enable	Output	Write enable for DRAM space access
CS2	RAS2/RAS	Row address strobe 2/ row address strobe	Output	Row address strobe when area 2 is designated as DRAM space or row address strobe when areas 2 to 5 are designated as continuous DRAM space
CS3	RAS3	Row address strobe 3	Output	Row address strobe when area 3 is designated as DRAM space
CS4	RAS4	Row address strobe 4	Output	Row address strobe when area 4 is designated as DRAM space
CS5	RAS5	Row address strobe 5	Output	Row address strobe when area 5 is designated as DRAM space
UCAS	UCAS	Upper column address strobe	Output	Upper column address strobe for 16-bit DRAM space access or column address strobe for 8-bit DRAM space access
LCAS	LCAS	Lower column address strobe	Output	Lower column address strobe signal for 16-bit DRAM space access
RD, OE	OE	Output enable	Output	Output enable signal for DRAM space access
WAIT	WAIT	Wait	Input	Wait request signal
A15 to A0	A15 to A0	Address pins	Output	Row address/column address multiplexed output
D15 to D0	D15 to D0	Data pins	I/O	Data input/output pins

6.6.5 Basic Timing

Figure 6.21 shows the basic access timing for DRAM space.

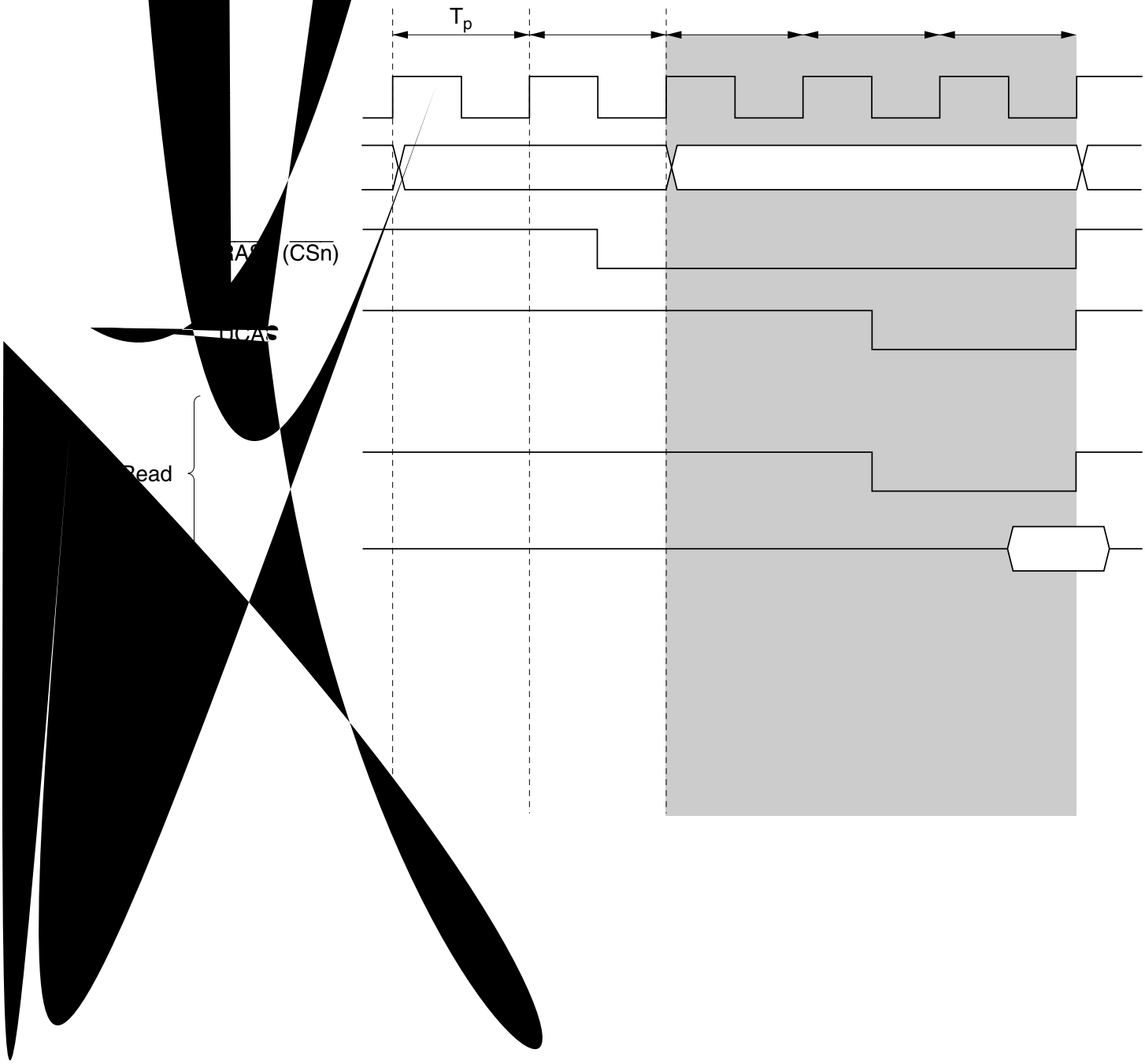
The four states of the basic timing consist of one T_p (precharge cycle) state, one T_r (row address output cycle) state, and the T_{c1} and two T_{c2} (column address output cycle) states.



from the RD pin and the (OE) pin, but in external read cycles for other than DRAM space, the output only from the RD pin.

6.6.4 Column Address Output Cycle Control

The column address output cycle can be changed from 2 states to 3 states by setting the CAST bit to 1 in MCR. Use this setting that gives the optimum specification values (CAS pulse width, etc.) according to the DRAM connected and the operating frequency of this LSI. Figure 6.22 shows an example of the timing when a 3-state column address output cycle is selected.



6.6.7 Row Address Output State Control

If the RAST bit is set to 1 in DRAMCR, the RAS signal goes low from the beginning of the T_r state, and the row address hold time and DRAM read access time are changed relative to the fall of the RAS signal. Use the optimum setting according to the DRAM connected and the operating frequency of this LSI. Figure 6.23 shows an example of the timing when the RAS signal goes low from the beginning of the T_r state.

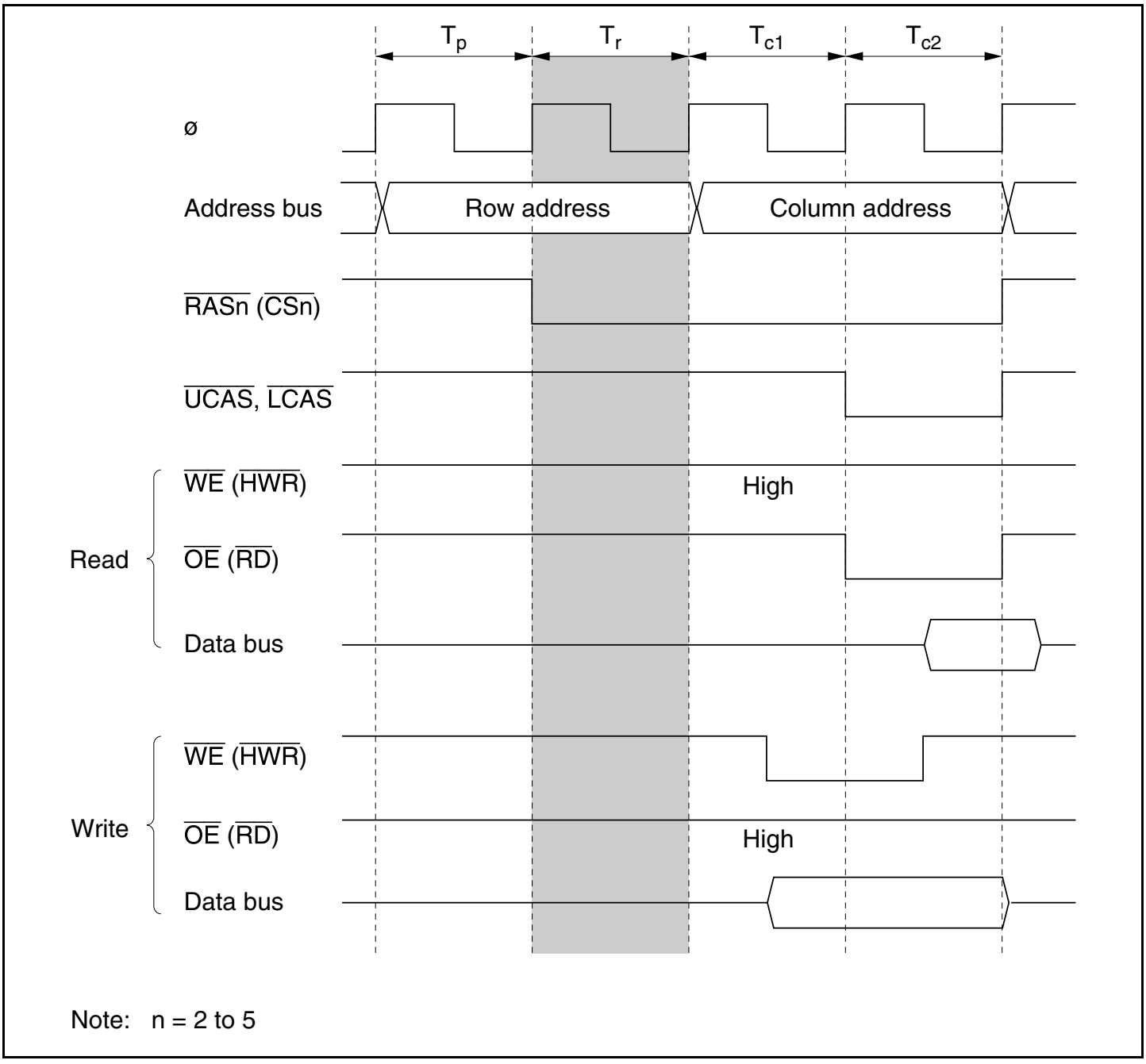


Figure 6.23 Example of Access Timing when RAS Signal Goes Low from Beginning of T_r State (CAST = 0)

If a row address hold time or read access time is necessary, making a setting in bits RCD1 and RCD0 in DRACCR allows from one to three T_{rw} states, in which row address output is maintained, to be inserted between the T_r cycle, in which the RAS signal goes low, and the T_{c1} cycle, in which the column address is output. Use the setting that gives the optimum row address signal hold time relative to the falling edge of the RAS signal according to the DRAM connected and the operating frequency of this LSI. Figure 6.24 shows an example of the timing when one T_{rw} state is set.

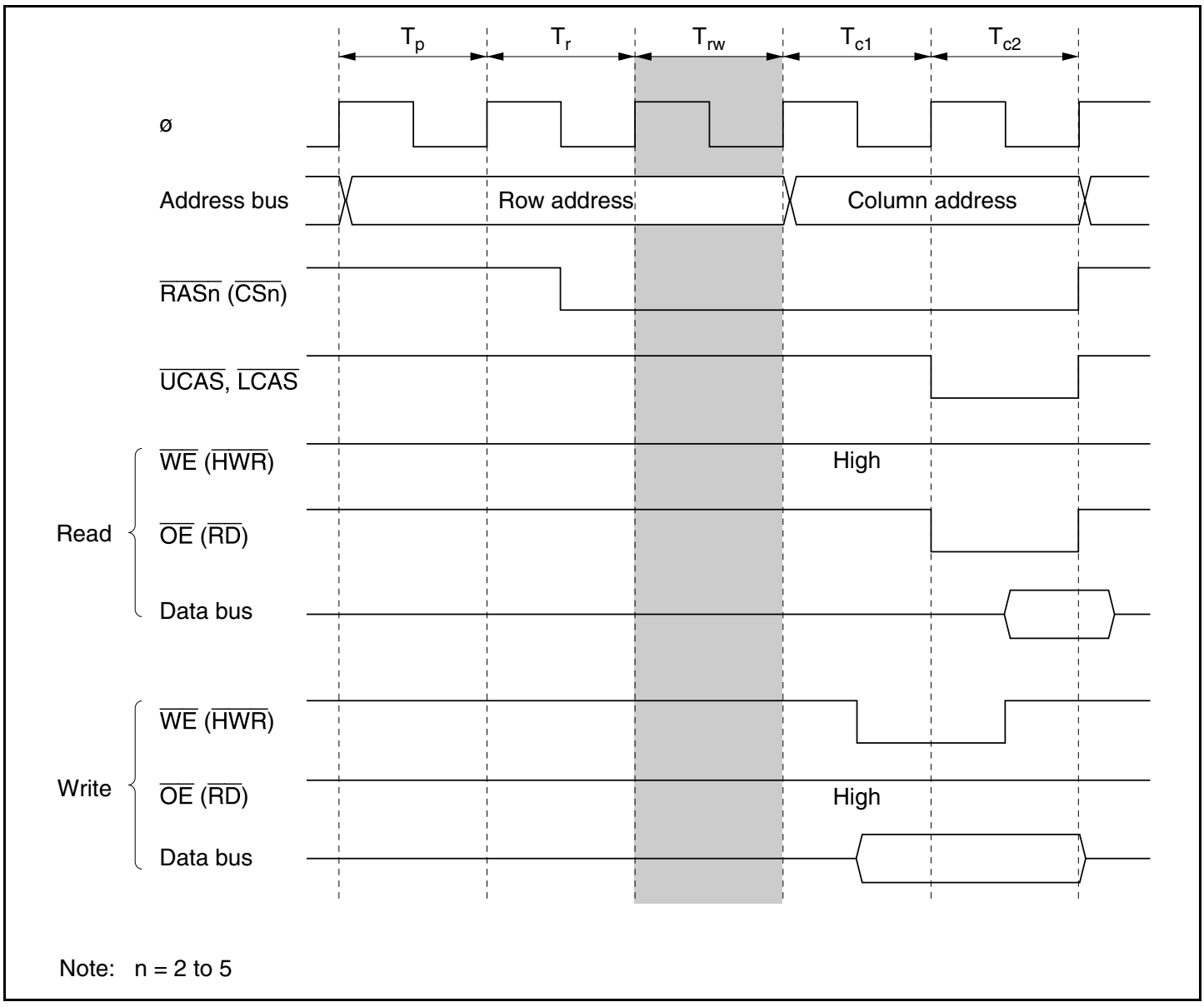


Figure 6.24 Example of Timing with One Row Address Output Maintenance State (RAST = 0, CAST = 0)

6.6.8 Precharge State Control

When DRAM is accessed, a RAS precharge time must be secured. With this LSI, one T_p state is always inserted when DRAM space is accessed. From one to four T_p states can be selected by setting bits TPC1 and TPC0 in DRACCR. Set the optimum number of T_p cycles according to the DRAM connected and the operating frequency of this LSI. Figure 6.25 shows the timing when two T_p states are inserted. The setting of bits TPC1 and TPC0 is also valid for T_p states in refresh cycles.

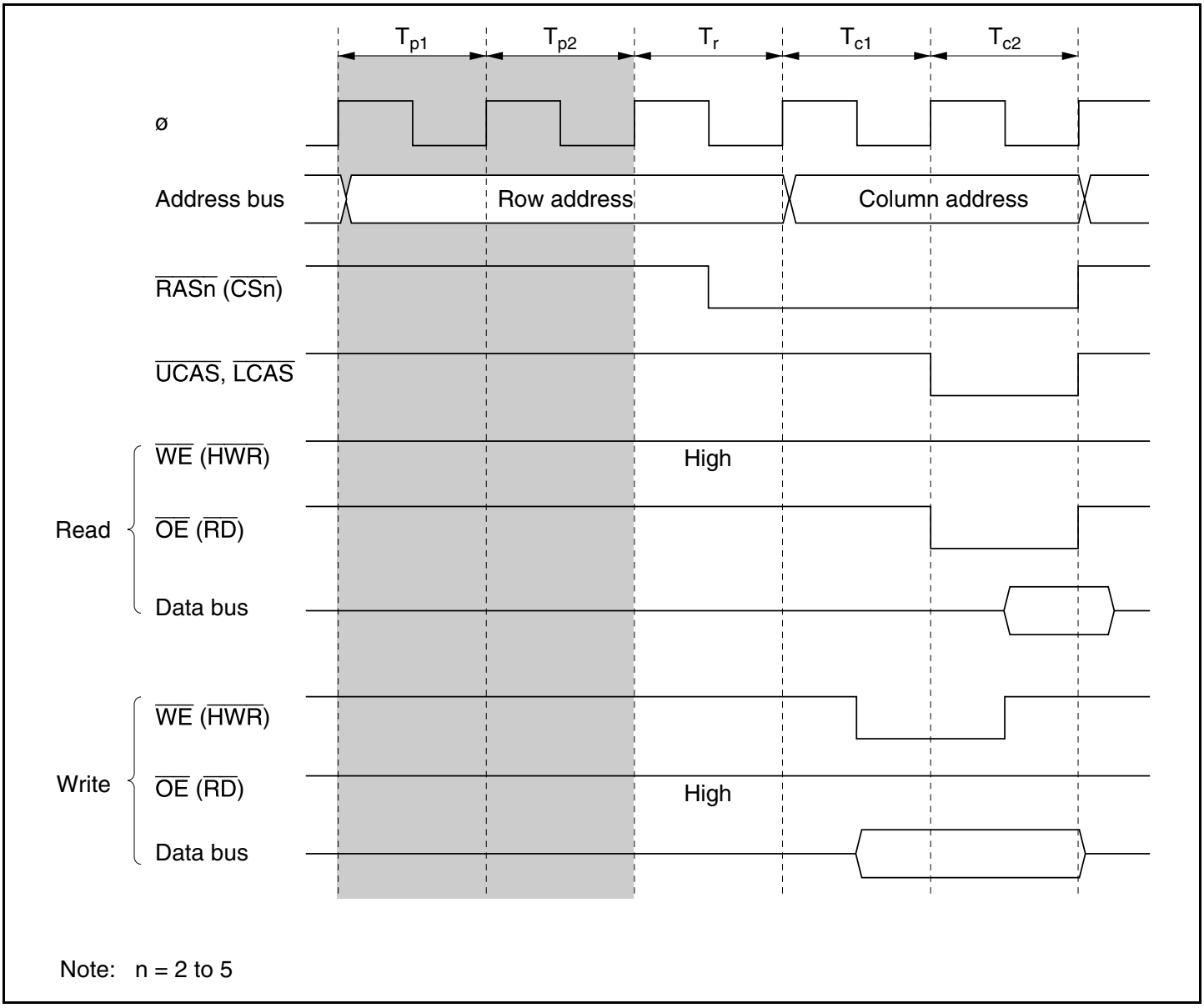


Figure 6.25 Example of Timing with Two-State Precharge Cycle (RAST = 0, CAST = 0)

6.6.9 Wait Control

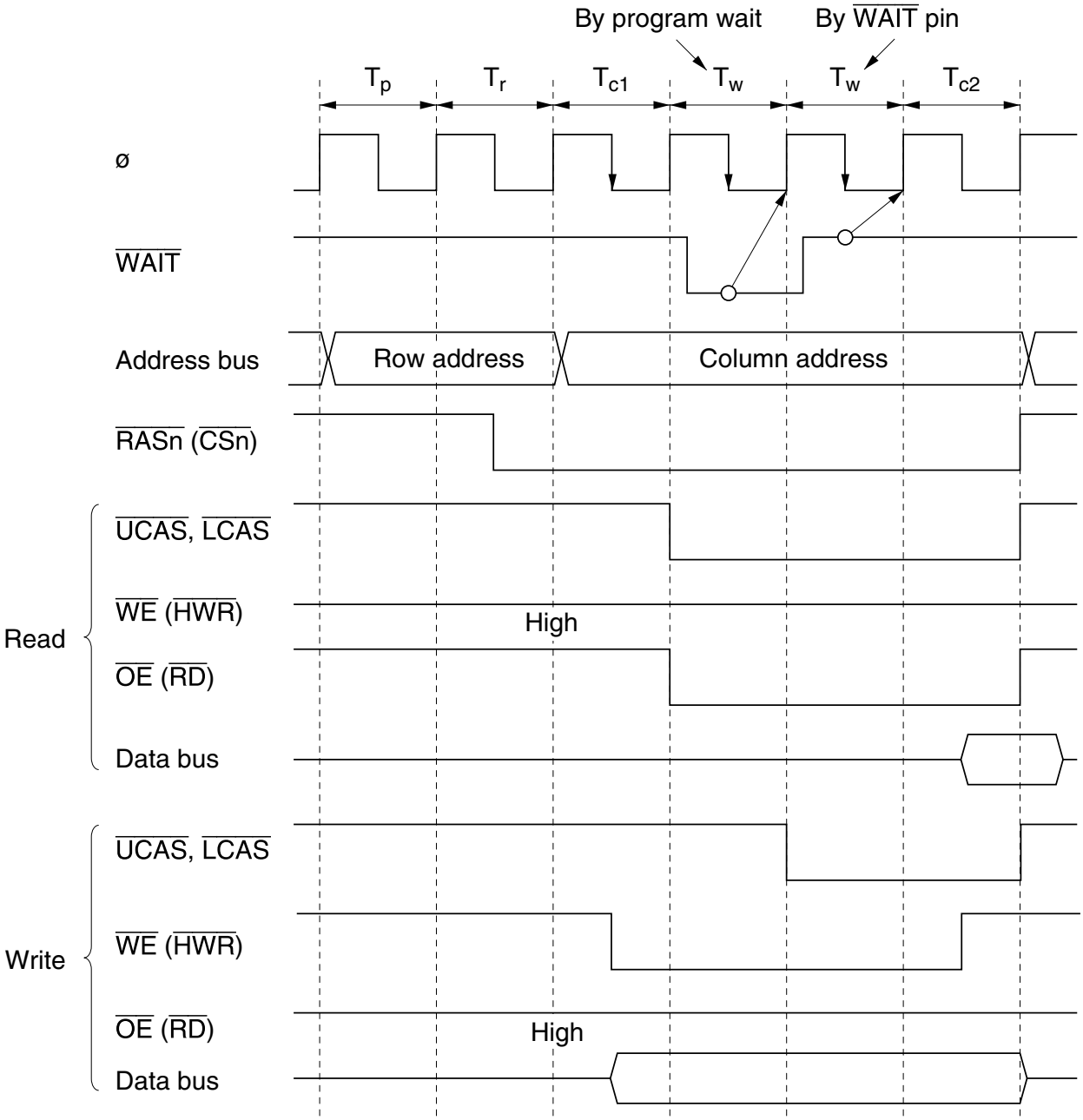
There are two ways of inserting wait states in a DRAM access cycle: program wait insertion and pin wait insertion using the WAIT pin.

Wait states are inserted to extend the CAS assertion period in a read access to DRAM space, and to extend the write data setup time relative to the falling edge of CAS in a write access.

Program Wait Insertion: When the bit in ASTCR corresponding to an area designated as DRAM space is set to 1, from 0 to 7 wait states can be inserted automatically between the T_{c1} state and T_{c2} state, according to the settings in registers WTCRA and WTCRB.

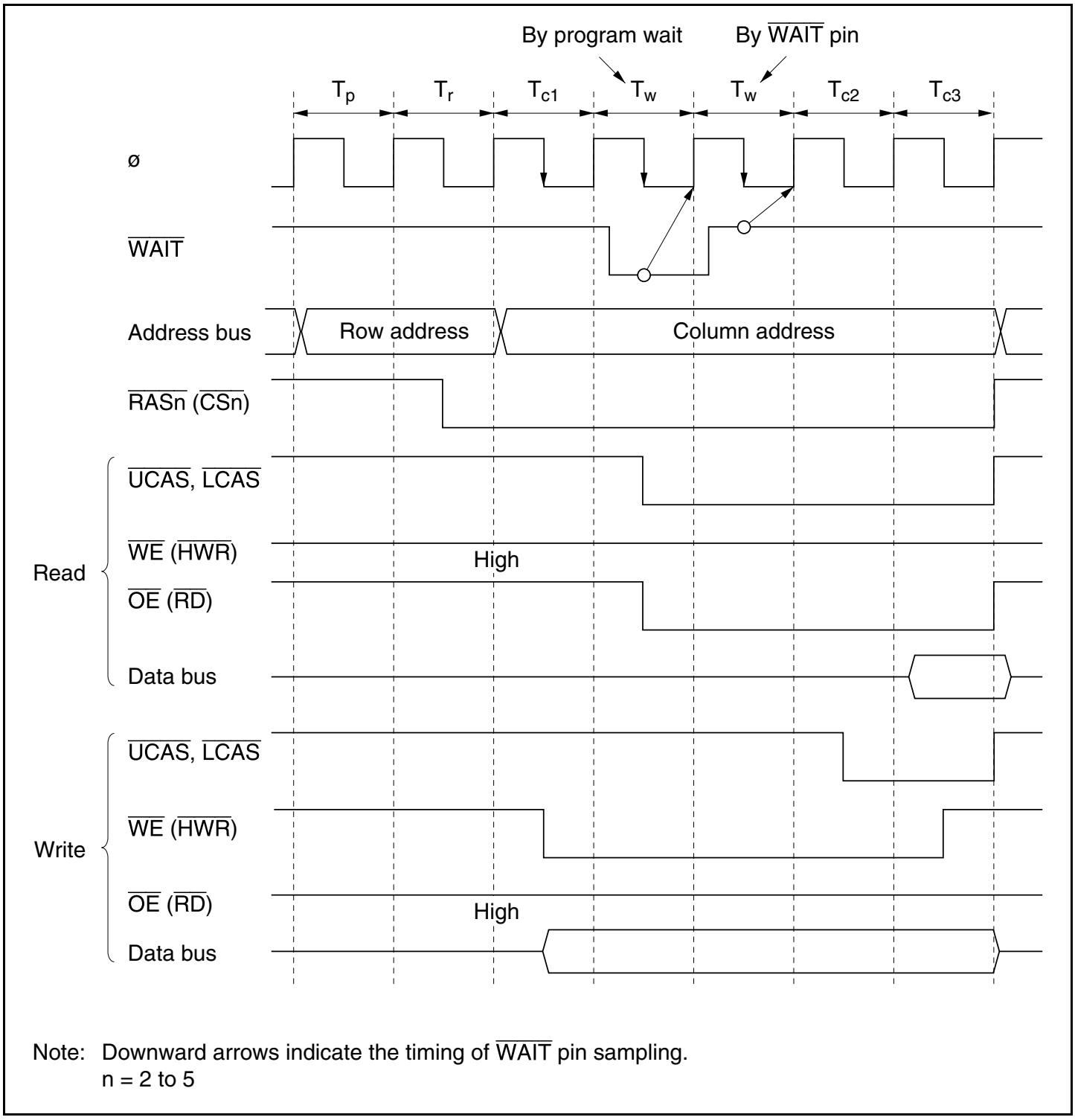
Pin Wait Insertion: When the WAITE bit in BCR is set to 1 and the ASTCR bit is set to 1, wait input by means of the WAIT pin is enabled. When DRAM space is accessed in this state, a program wait (T_w) is first inserted. If the WAIT pin is low at the falling edge of ø in the last T_{c1} or T_w state, another T_w state is inserted. If the WAIT pin is held low, T_w states are inserted until it goes high.

Figures 6.26 and 6.27 show examples of wait cycle insertion timing in the case of 2-state and 3-state column address output cycles.



Note: Downward arrows indicate the timing of WAIT pin sampling.
n = 2 to 5

Figure 6.26 Example of Wait State Insertion Timing (2-State Column Address Output)



Note: Downward arrows indicate the timing of WAIT pin sampling.
n = 2 to 5

Figure 6.27 Example of Wait State Insertion Timing (3-State Column Address Output)

6.6.10 Byte Access Control

When DRAM with a ×16-bit configuration is connected, the 2-CAS access method is used for the control signals needed for byte access. Figure 6.28 shows the control timing for 2-CAS access, and figure 6.29 shows an example of 2-CAS DRAM connection.

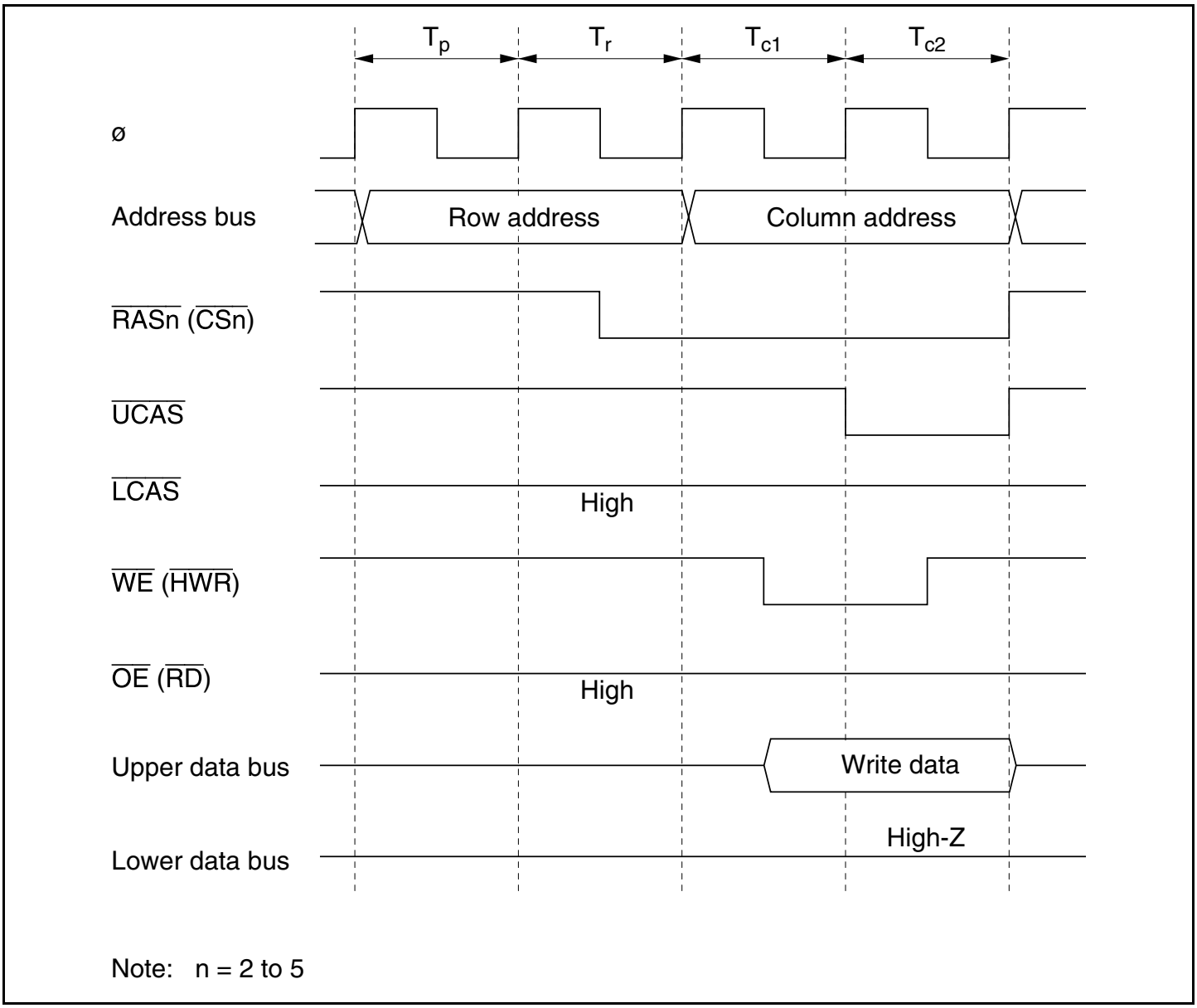


Figure 6.28 2-CAS Control Timing
(Upper Byte Write Access: RAST = 0, CAST = 0)

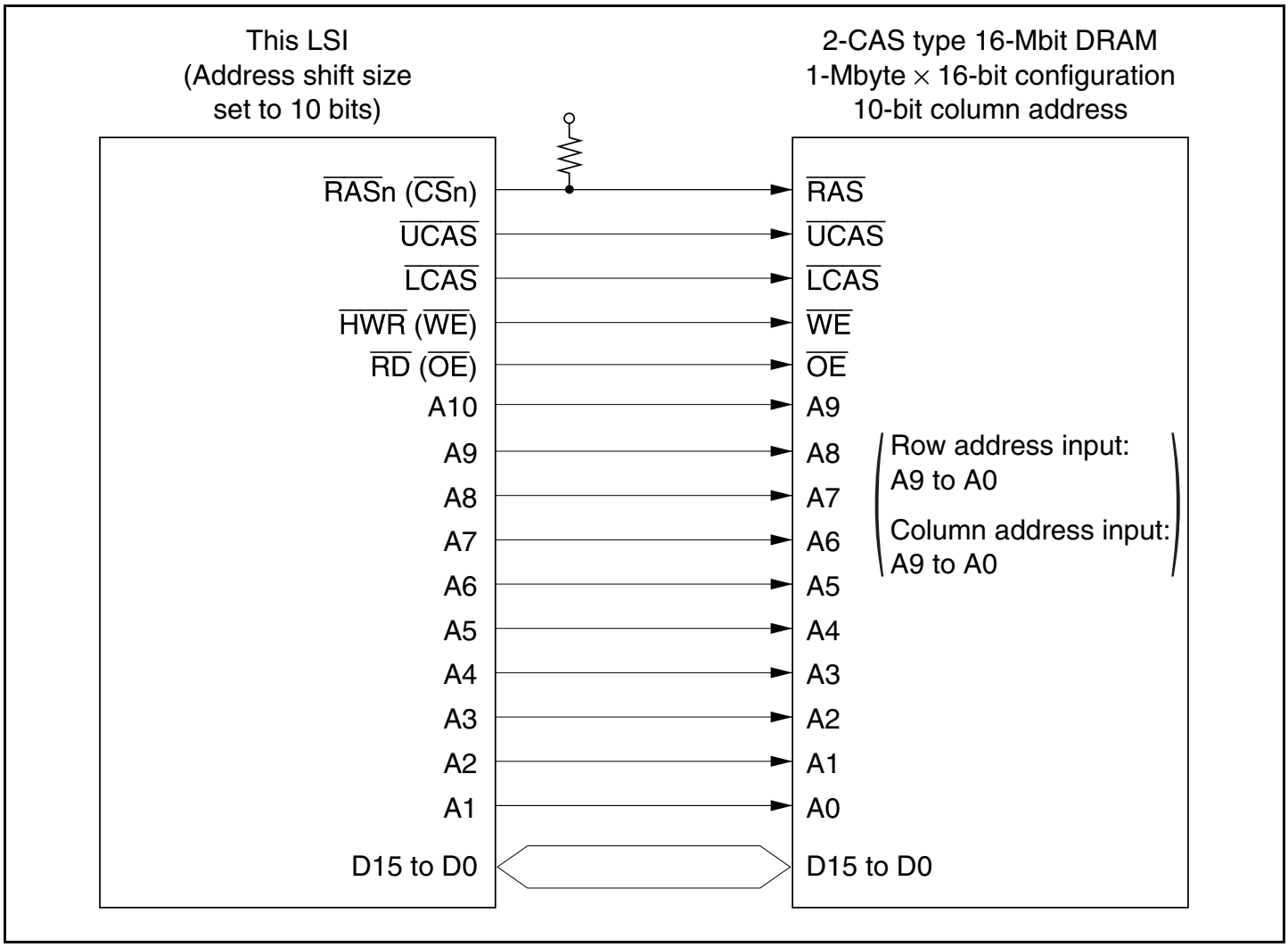


Figure 6.29 Example of 2-CAS DRAM Connection

6.6.11 Burst Operation

With DRAM, in addition to full access (normal access) in which data is accessed by outputting a row address for each access, a fast page mode is also provided which can be used when making consecutive accesses to the same row address. This mode enables fast (burst) access of data by simply changing the column address after the row address has been output. Burst access can be selected by setting the BE bit to 1 in DRAMCR.

Burst Access (Fast Page Mode): Figures 6.30 and 6.31 show the operation timing for burst access. When there are consecutive access cycles for DRAM space, the CAS signal and column address output cycles (two states) continue as long as the row address is the same for consecutive access cycles. The row address used for the comparison is set with bits MXC2 to MXC0 in DRAMCR.

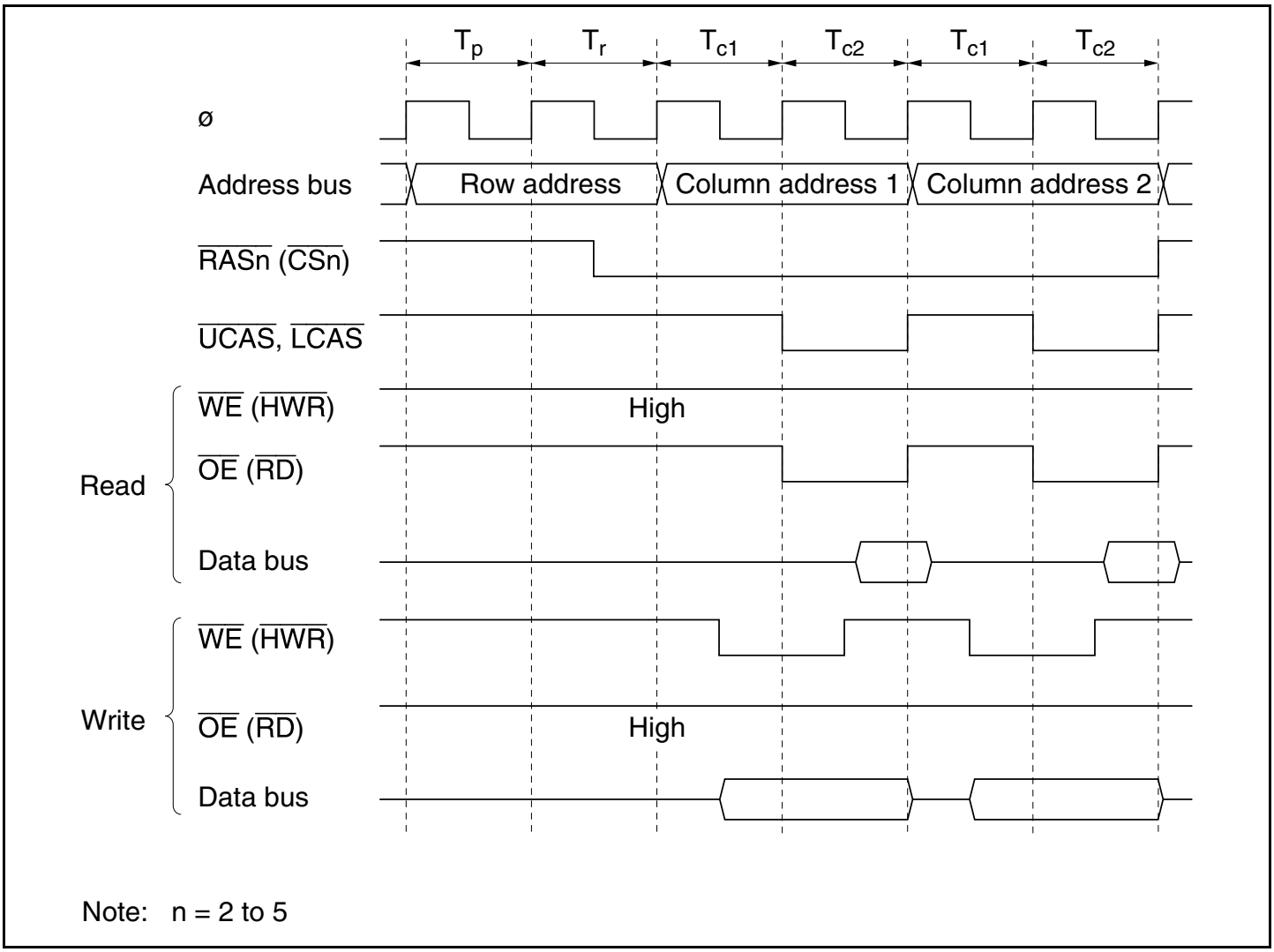


Figure 6.30 Operation Timing in Fast Page Mode (RAST = 0, CAST = 0)

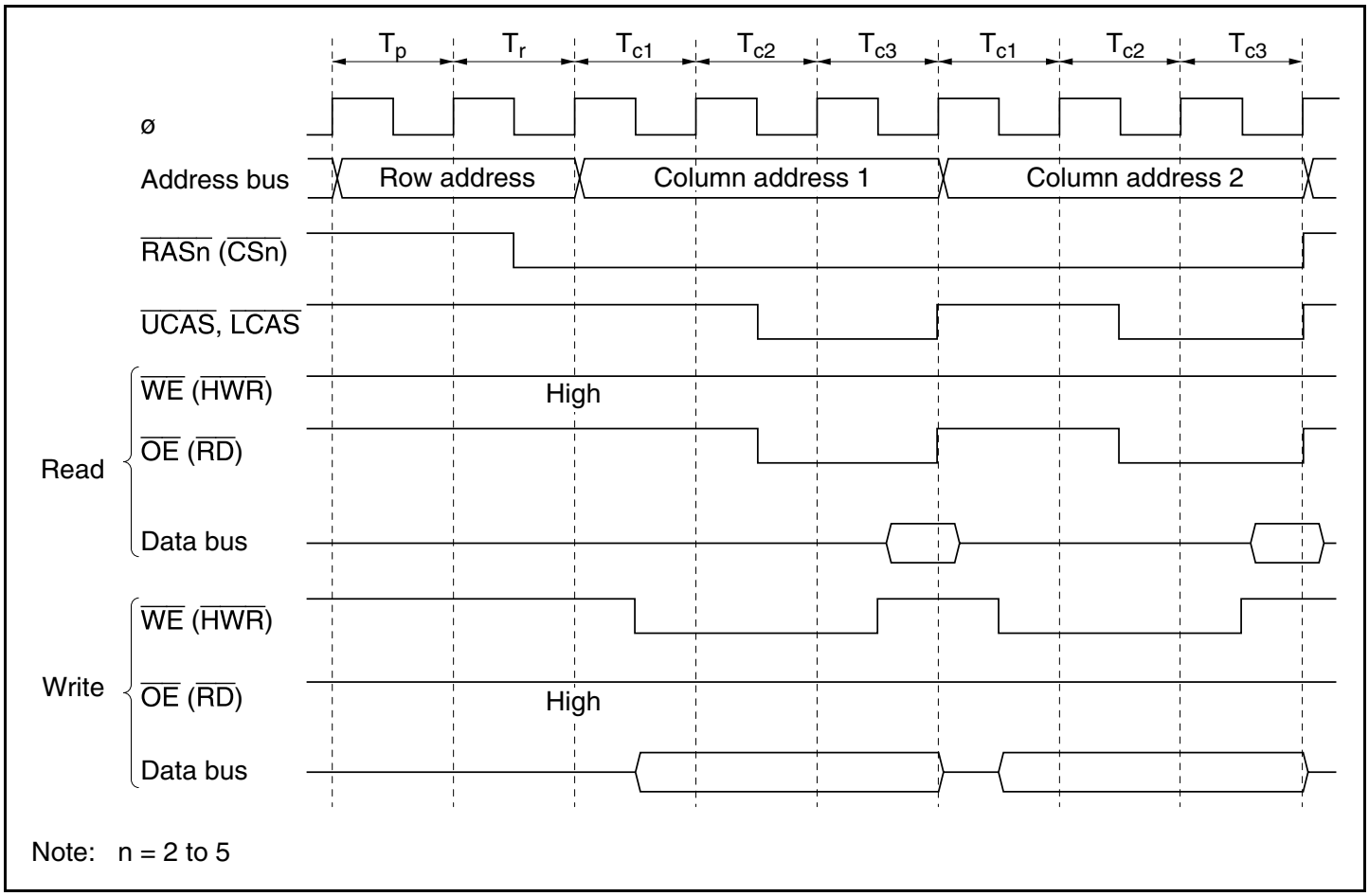


Figure 6.31 Operation Timing in Fast Page Mode (RAST = 0, CAST = 1)

The bus cycle can also be extended in burst access by inserting wait states. The wait state insertion method and timing are the same as for full access. For details see section 6.6.9, Wait Control.

RAS Down Mode and RAS Up Mode: Even when burst operation is selected, it may happen that access to DRAM space is not continuous, but is interrupted by access to another space. In this case, if the RAS signal is held low during the access to the other space, burst operation can be resumed when the same row address in DRAM space is accessed again.

- RAS Down Mode

  To select RAS down mode, set both the RCDM bit and the BE bit to 1 in DRAMCR. If access to DRAM space is interrupted and another space is accessed, the RAS signal is held low during the access to the other space, and burst access is performed when the row address of the next DRAM space access is the same as the row address of the previous DRAM space access. Figure 6.32 shows an example of the timing in RAS down mode.

  Note, however, that the RAS signal will go high if:

  — a refresh operation is initiated in the RAS down state

  — self-refreshing is performed

  — the chip enters software standby mode

  — the external bus is released

— the RCDM bit or BE bit is cleared to 0

If a transition is made to the all-module-clocks-stopped mode in the RAS down state, the clock will stop with RAS low. To enter the all-module-clocks-stopped mode with RAS high, the RCDM bit must be cleared to 0 before executing the SLEEP instruction.

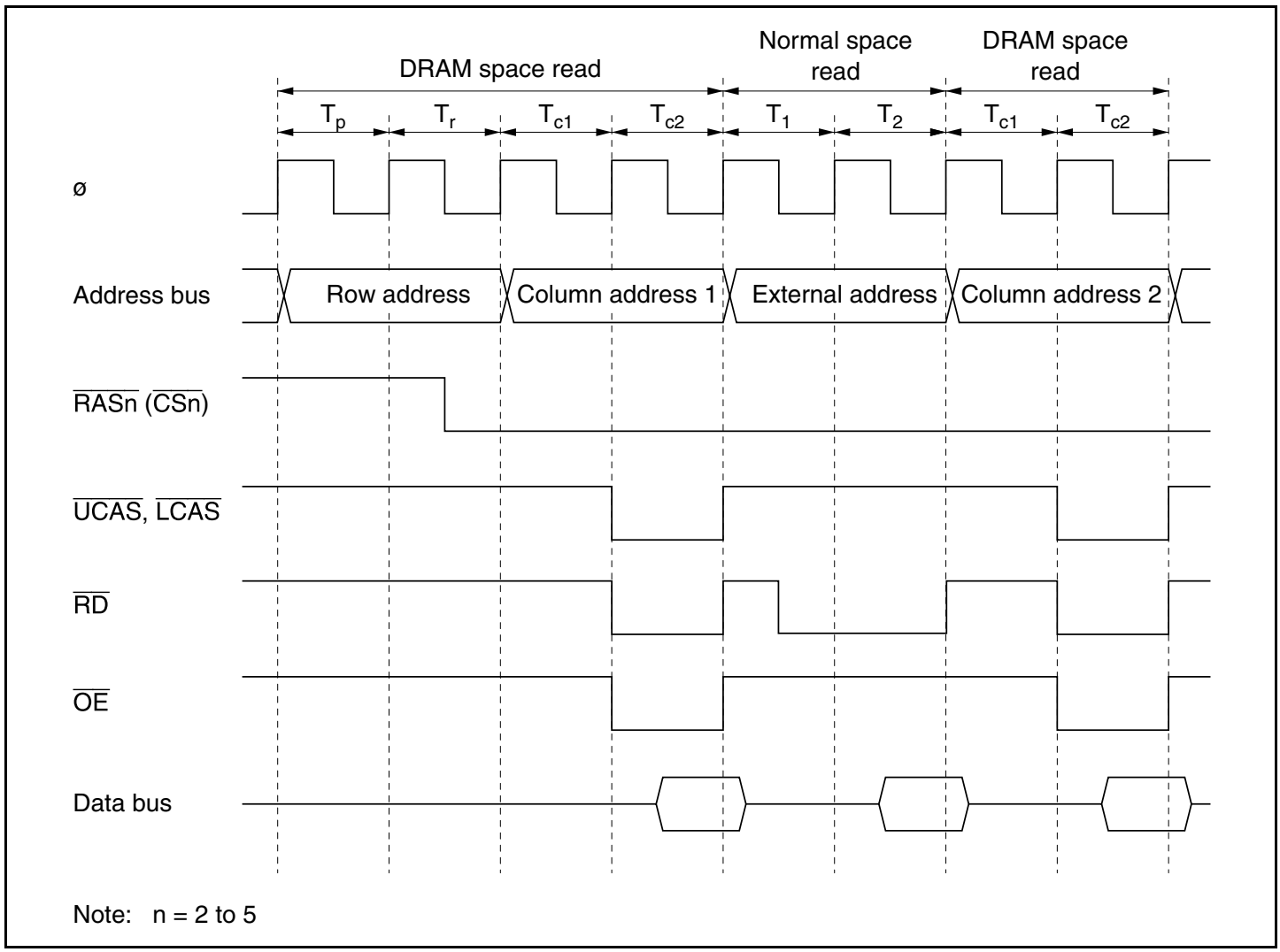


Figure 6.32 Example of Operation Timing in RAS Down Mode (RAST = 0, CAST = 0)

- RAS Up Mode

  To select RAS up mode, clear the RCDM bit to 0 in DRAMCR. Each time access to DRAM space is interrupted and another space is accessed, the RAS signal goes high again. Burst operation is only performed if DRAM space is continuous. Figure 6.33 shows an example of the timing in RAS up mode.

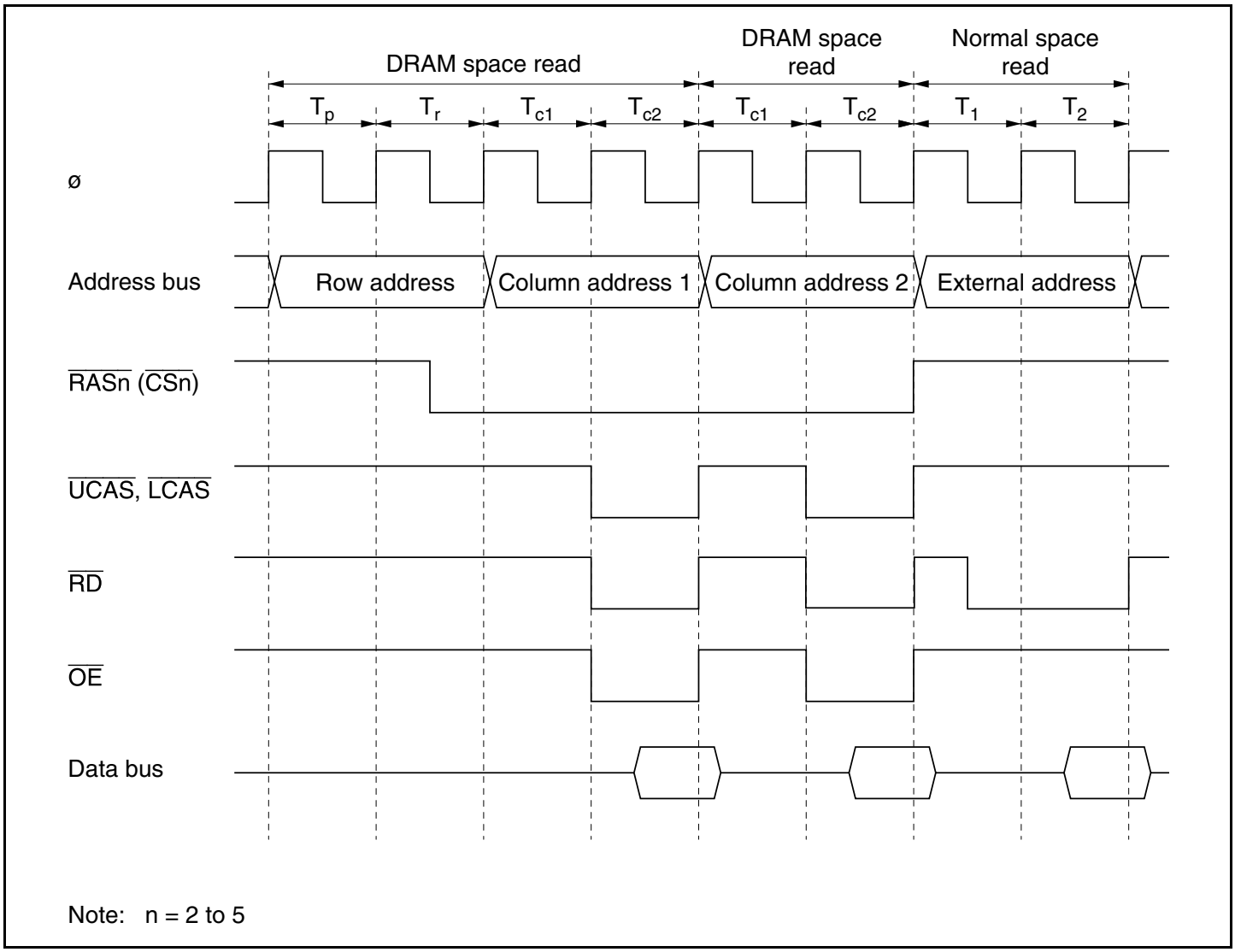


Figure 6.33 Example of Operation Timing in RAS Up Mode (RAST = 0, CAST = 0)

6.6.12 Refresh Control

This LSI is provided with a DRAM refresh control function. CAS-before-RAS (CBR) refreshing is used. In addition, self-refreshing can be executed when the chip enters the software standby state.

Refresh control is enabled when any area is designated as DRAM space in accordance with the setting of bits RMTS2 to RMTS0 in DRAMCR.

CAS-before-RAS (CBR) Refreshing: To select CBR refreshing, set the RFSHE bit to 1 in REFCR.

With CBR refreshing, RTCNT counts up using the input clock selected by bits RTCK2 to RTCK0 in REFCR, and when the count matches the value set in RTCOR (compare match), refresh control is performed. At the same time, RTCNT is reset and starts counting up again from H'00. Refreshing is thus repeated at fixed intervals determined by RTCOR and bits RTCK2 to RTCK0. Set a value in RTCOR and bits RTCK2 to RTCK0 that will meet the refreshing interval specification for the DRAM used.

When bits RTCK2 to RTCK0 in REFCR are set, RTCNT starts counting up. RTCNT and RTCOR settings should therefore be completed before setting bits RTCK2 to RTCK0. RTCNT operation is shown in figure 6.34, compare match timing in figure 6.35, and CBR refresh timing in figure 6.36.

When the CBRM bit in REFCR is cleared to 0, access to external space other than DRAM space is performed in parallel during the CBR refresh period.

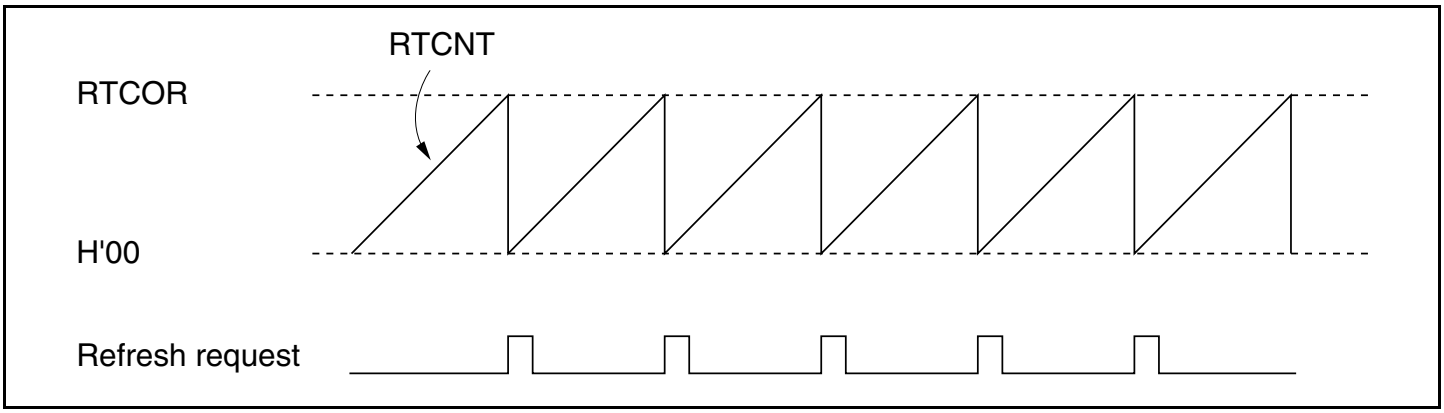


Figure 6.34 RTCNT Operation

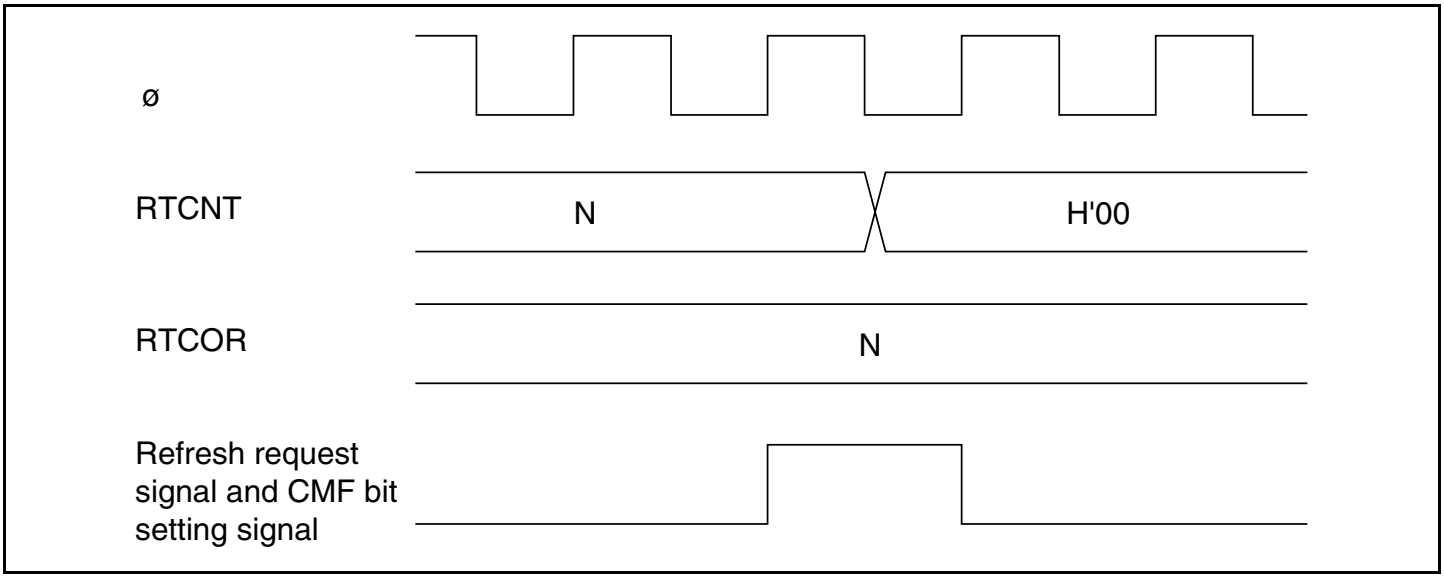


Figure 6.35 Compare Match Timing

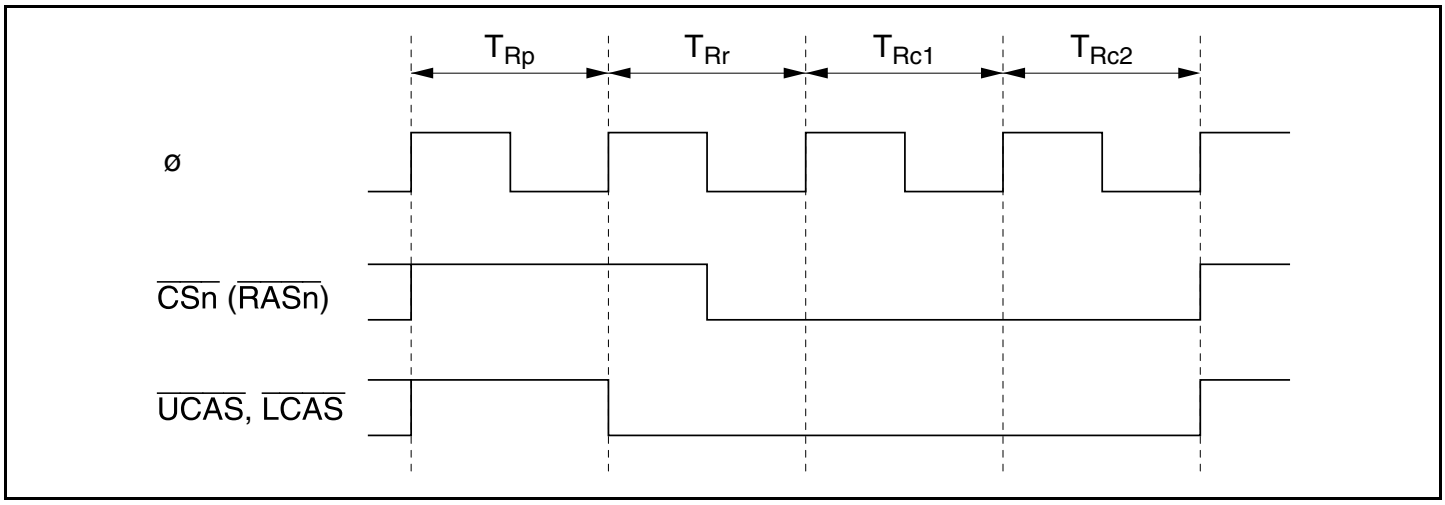


Figure 6.36 CBR Refresh Timing

A setting can be made in bits RCW1 and RCW0 in REFCR to delay RAS signal output by one to three cycles. Use bits RLW1 and RLW0 in REFCR to adjust the width of the RAS signal. The settings of bits RCW1, RCW0, RLW1, and RLW0 are valid only in refresh operations.

Figure 6.37 shows the timing when bits RCW1 and RCW0 are set.

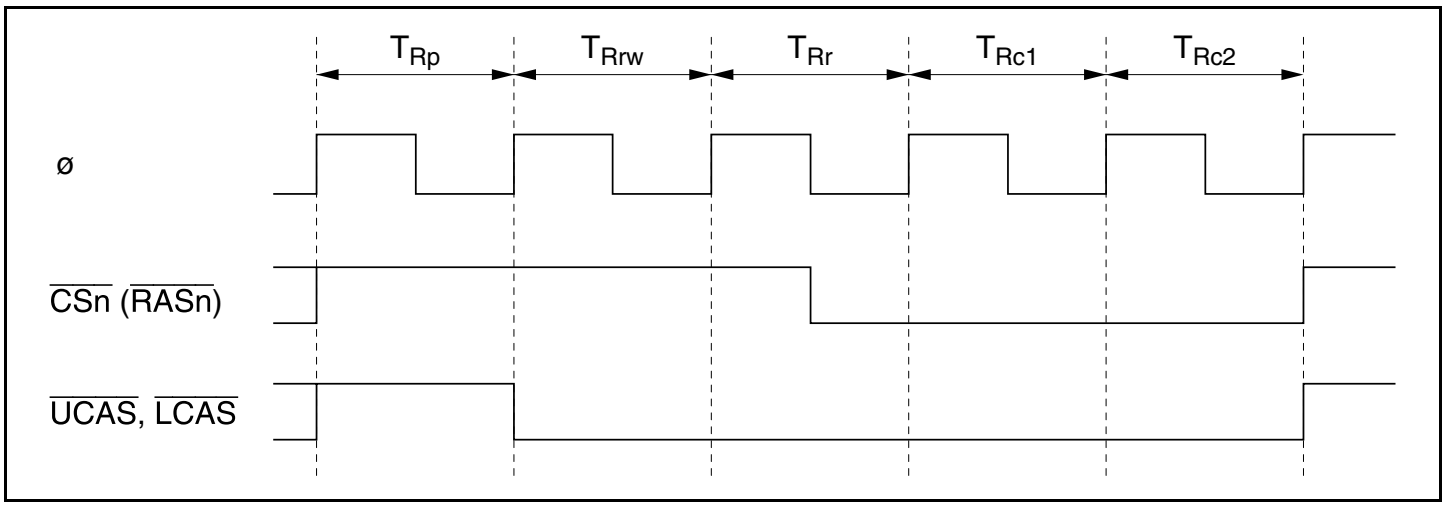


Figure 6.37 CBR Refresh Timing
(RCW1 = 0, RCW0 = 1, RLW1 = 0, RLW0 = 0)

Depending on the DRAM used, modification of the WE signal may not be permitted during the refresh period. In this case, the CBRM bit in REFCR should be set to 1. The bus controller will then insert refresh cycles in appropriate breaks between bus cycles. Figure 6.38 shows an example of the timing when the CBRM bit is set to 1. In this case the CS signal is not controlled, and retains its value prior to the start of the refresh period.

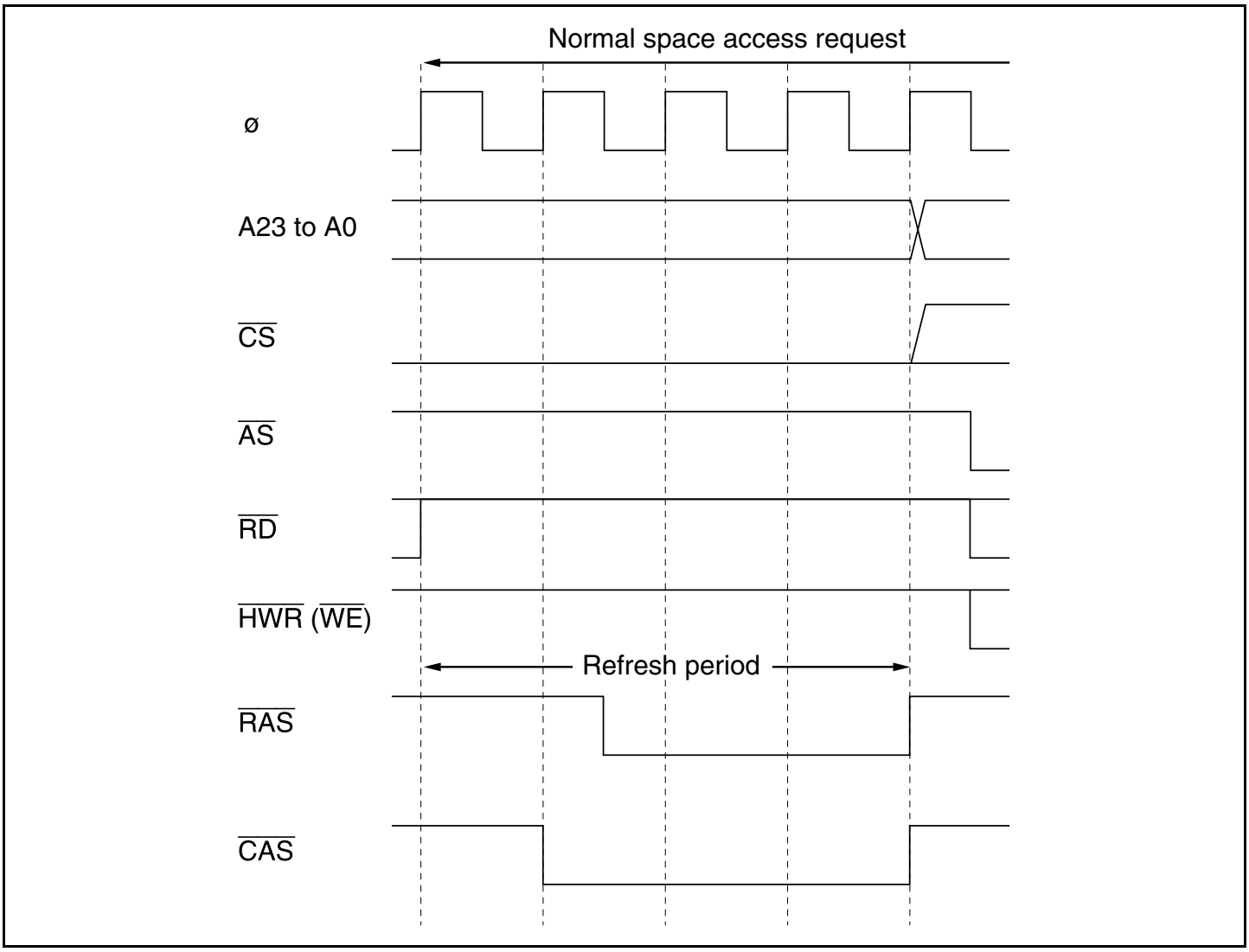


Figure 6.38 Example of CBR Refresh Timing (CBRM = 1)

Self-Refreshing: A self-refresh mode (battery backup mode) is provided for DRAM as a kind of standby mode. In this mode, refresh timing and refresh addresses are generated within the DRAM.

To select self-refreshing, set the RFSHE bit and SLFRF bit to 1 in REFCR. When a SLEEP instruction is executed to enter software standby mode, the CAS and RAS signals are output and DRAM enters self-refresh mode, as shown in figure 6.39.

When software standby mode is exited, the SLFRF bit is cleared to 0 and self-refresh mode is exited automatically. If a CBR refresh request occurs when making a transition to software standby mode, CBR refreshing is executed, then self-refresh mode is entered.

When using self-refresh mode, the OPE bit must not be cleared to 0 in the SBYCR register.

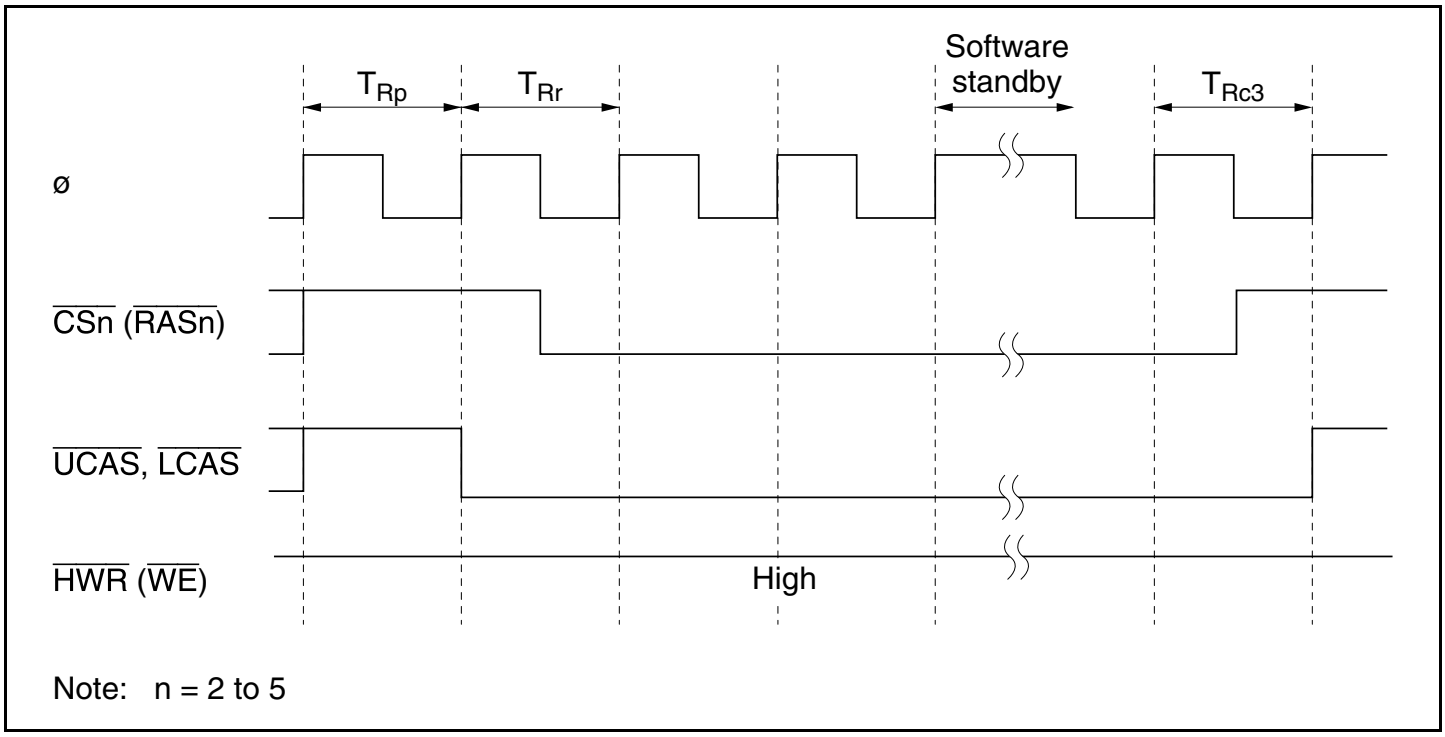


Figure 6.39 Self-Refresh Timing

In some DRAMs provided with a self-refresh mode, the RAS signal precharge time immediately after self-refreshing is longer than the normal precharge time. A setting can be made in bits TPCS2 to TPCS0 in REFCR to make the precharge time immediately after self-refreshing from 1 to 7 states longer than the normal precharge time. In this case, too, normal precharging is performed according to the setting of bits TPC1 and TPC0 in DRACCR, and therefore a setting should be made to give the optimum post-self-refresh precharge time, including this time. Figure 6.40 shows an example of the timing when the precharge time immediately after self-refreshing is extended by 2 states.

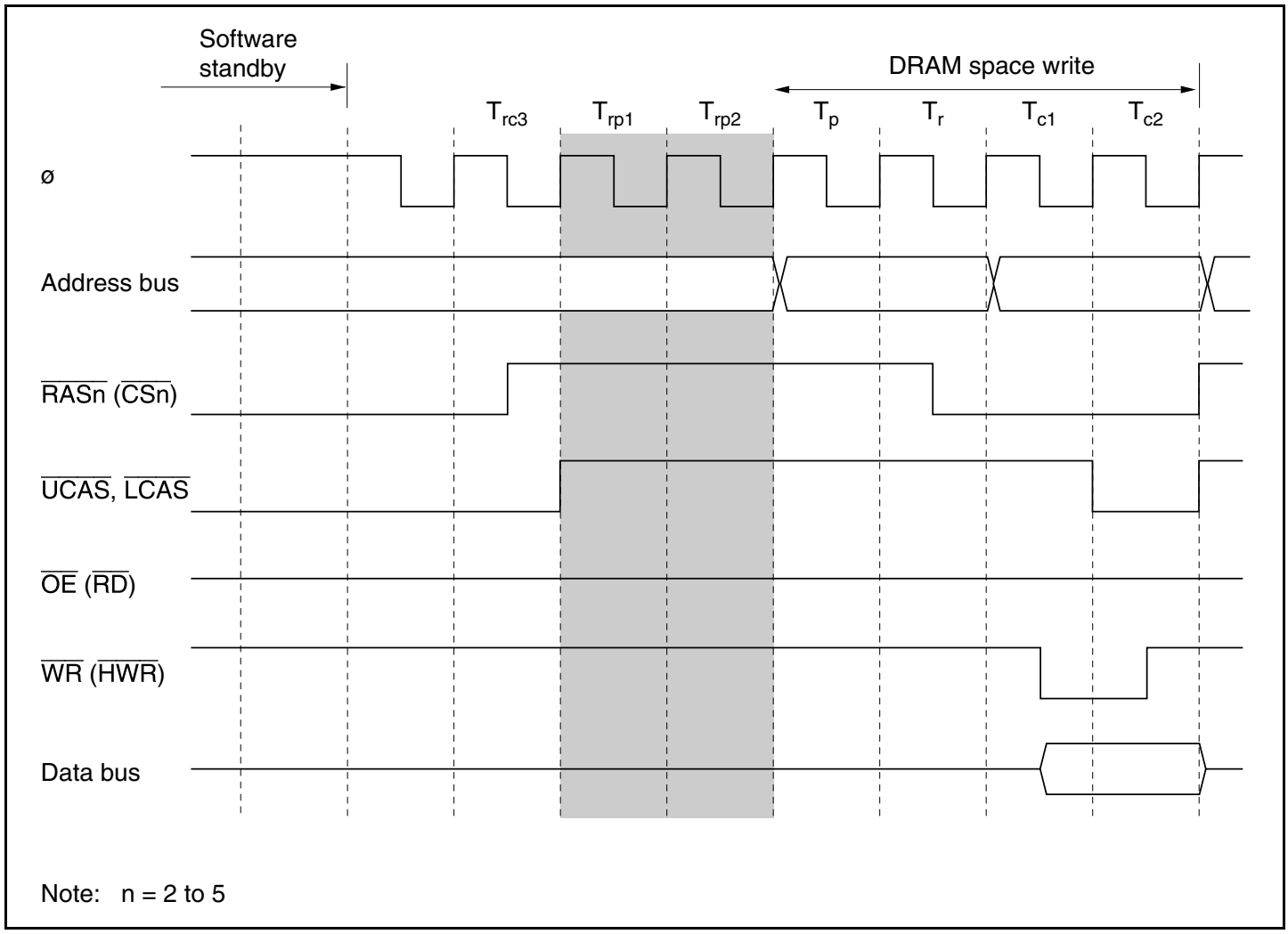


Figure 6.40 Example of Timing when Precharge Time after Self-Refreshing Is Extended by 2 States

Refreshing and All-Module-Clocks-Stopped Mode: In this LSI, if the ACSE bit is set to 1 in MSTPCRH, and then a SLEEP instruction is executed with the setting for all peripheral module clocks to be stopped (MSTPCR = H'FFFF) or for operation of the 8-bit timer module alone (MSTPCR = H'FFFE), and a transition is made to the sleep state, the all-module-clocks-stopped mode is entered, in which the bus controller and I/O port clocks are also stopped. As the bus controller clock is also stopped in this mode, CBR refreshing is not executed. If DRAM is connected externally and DRAM data is to be retained in sleep mode, the ACSE bit must be cleared to 0 in MSTPCRH.

6.6.13 DMAC and EXDMAC Single Address Transfer Mode and DRAM Interface

When burst mode is selected on the DRAM interface, the DACK and EDACK output timing can be selected with the DDS and EDDS bits in DRAMCR. When DRAM space is accessed in DMAC or EXDMAC single address mode at the same time, these bits select whether or not burst access is to be performed.

When DDS = 1 or EDDS = 1: Burst access is performed by determining the address only, irrespective of the bus master. With the DRAM interface, the DACK or EDACK output goes low from the T_{c1} state.

Figure 6.41 shows the DACK or EDACK output timing for the DRAM interface when DDS = 1 or EDDS = 1.

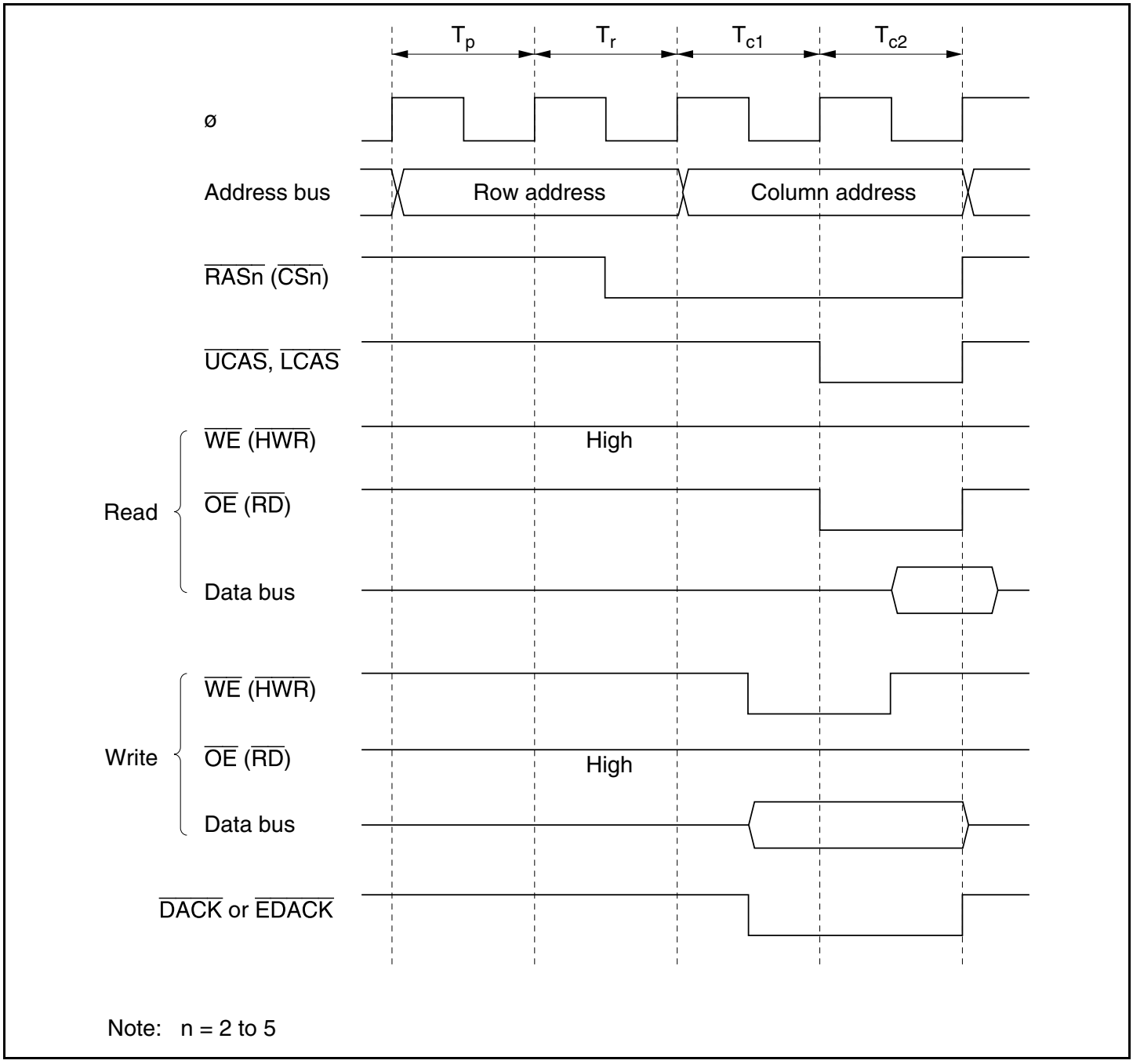


Figure 6.41 Example of DACK/EDACK Output Timing when DDS = 1 or EDDS = 1 (RAST = 0, CAST = 0)

When DDS = 0 or EDDS = 0: When DRAM space is accessed in DMAC or EXDMAC single address transfer mode, full access (normal access) is always performed. With the DRAM interface, the DACK or EDACK output goes low from the T_r state.

In modes other than DMAC or EXDMAC single address transfer mode, burst access can be used when accessing DRAM space.

Figure 6.42 shows the DACK or EDACK output timing for the DRAM interface when DDS = 0 or EDDS = 0.

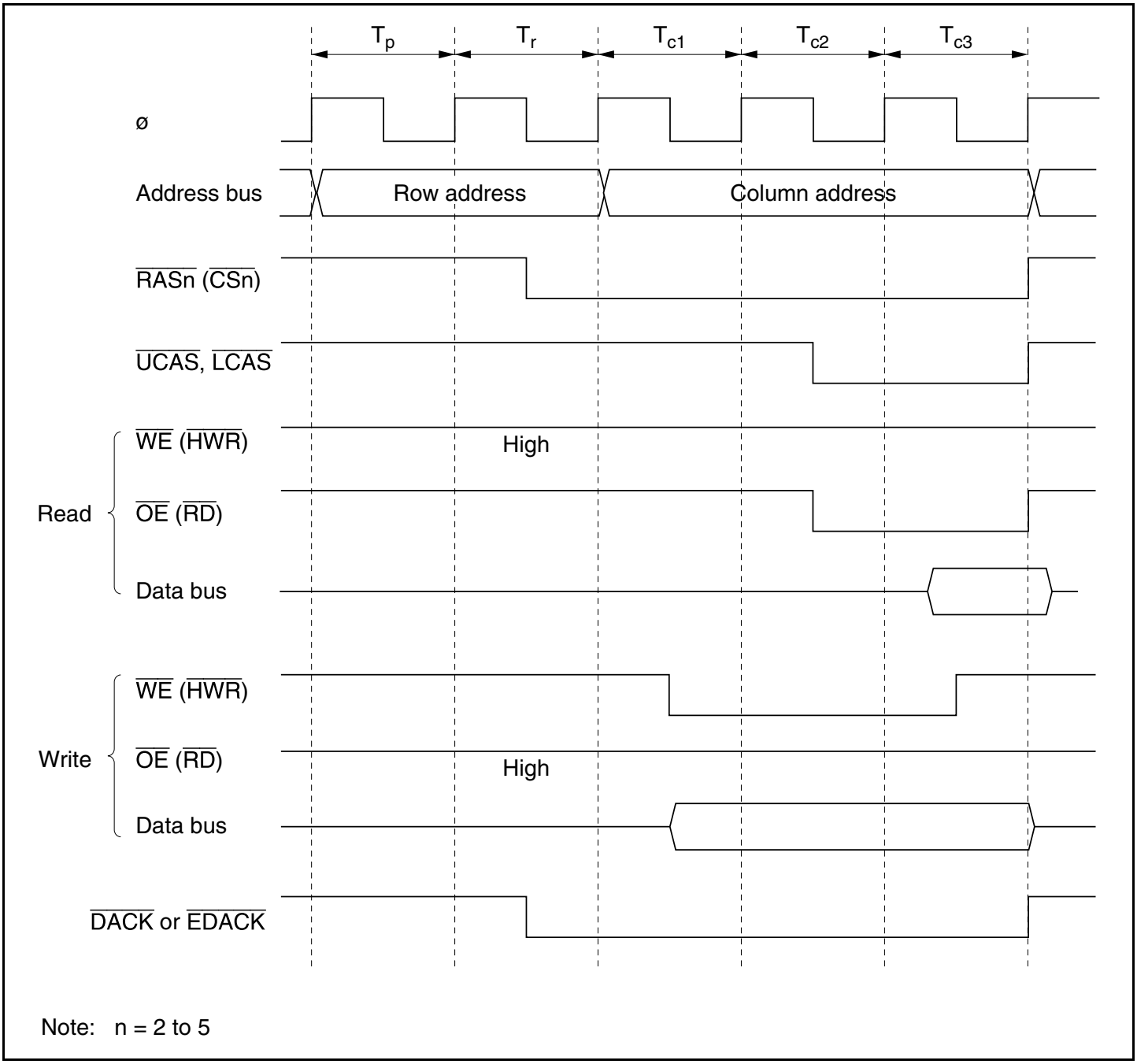


Figure 6.42 Example of DACK/EDACK Output Timing when DDS = 0 or EDDS = 0 (RAST = 0, CAST = 1)

6.7 Synchronous DRAM Interface

In the H8S/2678R Series, external address space areas 2 to 5 can be designated as continuous synchronous DRAM space, and synchronous DRAM interfacing performed. The synchronous DRAM interface allows synchronous DRAM to be directly connected to this LSI. A synchronous DRAM space of up to 8 Mbytes can be set by means of bits RMTS2 to RMTS0 in DRAMCR. Synchronous DRAM of CAS latency 1 to 4 can be connected.

Note: The synchronous DRAM interface is not supported in the H8S/2678 Series.

6.7.1 Setting Continuous Synchronous DRAM Space

Areas 2 to 5 are designated as continuous synchronous DRAM space by setting bits RMTS2 to RMTS0 in DRAMCR. The relation between the settings of bits RMTS2 to RMTS0 and synchronous DRAM space is shown in table 6.7. Possible synchronous DRAM interface settings are and continuous area (areas 2 to 5).

Table 6.7 Relation between Settings of Bits RMTS2 to RMTS0 and Synchronous DRAM Space

RMTS2	RMTS1	RMTS0	Area 5	Area 4	Area 3	Area 2
0	0	1	Normal space	Normal space	Normal space	DRAM space
	1	0	Normal space	Normal space	DRAM space	DRAM space
		1	DRAM space	DRAM space	DRAM space	DRAM space
1	0	0	Continuous synchronous DRAM space*			
		1	Mode settings of synchronous DRAM			
	1	0	Reserved (setting prohibited)			
		1	Continuous DRAM space			

With continuous synchronous DRAM space, CS2, CS3, CS4 pins are used as RAS, CAS, WE signal. The (OE) pin of the synchronous DRAM is used as the CKE signal, and the CS5 pin is used as synchronous DRAM clock (SDRAMϕ). The bus specifications for continuous synchronous DRAM space conform to the settings for area 2. The pin wait and program wait for the continuous synchronous DRAM are invalid.

Commands for the synchronous DRAM can be specified by combining RAS, CAS, WE, and address-precharge-setting command (Prechrge-sel) output on the upper column addresses.

Commands that are supported by this LSI are NOP, auto-refresh (REF), self-refresh (SELF), all bank precharge (PALL), row address strobe bank-active (ACTV), read (READ), write (WRIT), and mode-register write (MRS). Commands for bank control cannot be used.

6.7.2 Address Multiplexing

With continuous synchronous DRAM space, the row address and column address are multiplexed. In address multiplexing, the size of the shift of the row address is selected with bits MXC2 to MXC0 in DRAMCR. The address-precharge-setting command (Prechrge-sel) can be output on the upper column address. Table 6.8 shows the relation between the settings of MXC2 to MXC0 and the shift size. The MXC2 bit should be set to 1 when the synchronous DRAM interface is used.

Table 6.8 Relation between Settings of Bits MXC2 to MXC0 and Address Multiplexing

	DRAMCR				Address Pins															
	MXC2	MXC1	MXC0	Shift Size	A23 to A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Row address	0	x	x		Reserved (setting prohibited)																
	1	0	0	8 bits	A23 to A16	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8
			1	9 bits	A23 to A16	A15	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9
		1	0	10 bits	A23 to A16	A15	A14	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			1	11 bits	A23 to A16	A15	A14	A13	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
Column address	0	x	x		Reserved (setting prohibited)																
	1	0	0	—	A23 to A16	P	P	P	P	P	P	P	A8	A7	A6	A5	A4	A3	A2	A1	A0
			1	—	A23 to A16	P	P	P	P	P	P	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		1	0	—	A23 to A16	P	P	P	P	P	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
			1	—	A23 to A16	P	P	P	P	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

X: Don't care.

P: Precharge-sel

6.7.3 Data Bus

If the ABW2 bit in ABWCR corresponding to an area designated as continuous synchronous DRAM space is set to 1, area 2 to 5 are designated as 8-bit continuous synchronous DRAM space; if the bit is cleared to 0, the areas are designated as 16-bit continuous synchronous DRAM space. In 16-bit continuous synchronous DRAM space, ×16-bit configuration synchronous DRAM can be connected directly.

In 8-bit continuous synchronous DRAM space the upper half of the data bus, D15 to D8, is enabled, while in 16-bit continuous synchronous DRAM space both the upper and lower halves of the data bus, D15 to D0, are enabled.

Access sizes and data alignment are the same as for the basic bus interface: see section 6.5.1, Data Size and Data Alignment.

6.7.4 Pins Used for Synchronous DRAM Interface

Table 6.9 shows pins used for the synchronous DRAM interface and their functions. To enable the synchronous DRAM interface, fix the DCTL pin to 1. Do not vary the DCTL pin during operation.

Since the CS2 to CS4 pins are in the input state after a reset, set DDR to 1 when RAS, CAS, and WE signals are output. For details, see section 10, I/O Ports. Set the OEE bit of the DRAMCR register to 1 when the CKE signal is output.

Table 6.9 Synchronous DRAM Interface Pins

Pin	With Synchronous DRAM Setting	Name	I/O	Function
CS2	RAS	Row address strobe	Output	Row address strobe when areas 2 to 5 are designated as continuous synchronous DRAM space
CS3	CAS	Column address strobe	Output	Column address strobe when areas 2 to 5 are designated as continuous synchronous DRAM space
CS4	WE	Write enable	Output	Write enable strobe when areas 2 to 5 are designated as continuous synchronous DRAM space
CS5	SDRAMϕ	Clock	Output	Clock only for synchronous DRAM
(OE)	(CKE)	Clock enable	Output	Clock enable signal when areas 2 to 5 are designated as continuous synchronous DRAM space
UCAS	DQMU	Upper data mask enable	Output	Upper data mask enable for 16-bit continuous synchronous DRAM space access/data mask enable for 8-bit continuous synchronous DRAM space access
LCAS	DQML	Lower data mask enable	Output	Lower data mask enable signal for 16-bit continuous synchronous DRAM space access
A15 to A0	A15 to A0	Address pins	Output	Row address/column address multiplexed output pins
D15 to D0	D15 to D0	Data pins	I/O	Data input/output pins
DCTL	DCTL	Device control pin	Input	Output enable pin for SDRAMϕ

6.7.5 Synchronous DRAM Clock

When the DCTL pin is fixed to 1, synchronous clock (SDRAMϕ) is output from the CS5 pin. When the frequency multiplication factor of the PLL circuit of this LSI is set to ×1 or ×2, SDRAMϕ is 90° phase shift from ϕ. Therefore, a stable margin is ensured for the synchronous DRAM that operates at the rising edge of clocks. Figure 6.43 shows the relationship between ϕ and SDRAMϕ. When the frequency multiplication factor of the PLL circuit is ×4, the phase of SDRAMϕ and that of ϕ are the same.

When the CLK pin of the synchronous DRAM is directly connected to SDRAMϕ of this LSI, it is recommended to set the frequency multiplication factor of the PLL circuit to ×1 or ×2.

Note: SDRAMϕ output timing is shown when the frequency multiplication factor of the PLL circuit is ×1 or ×2.

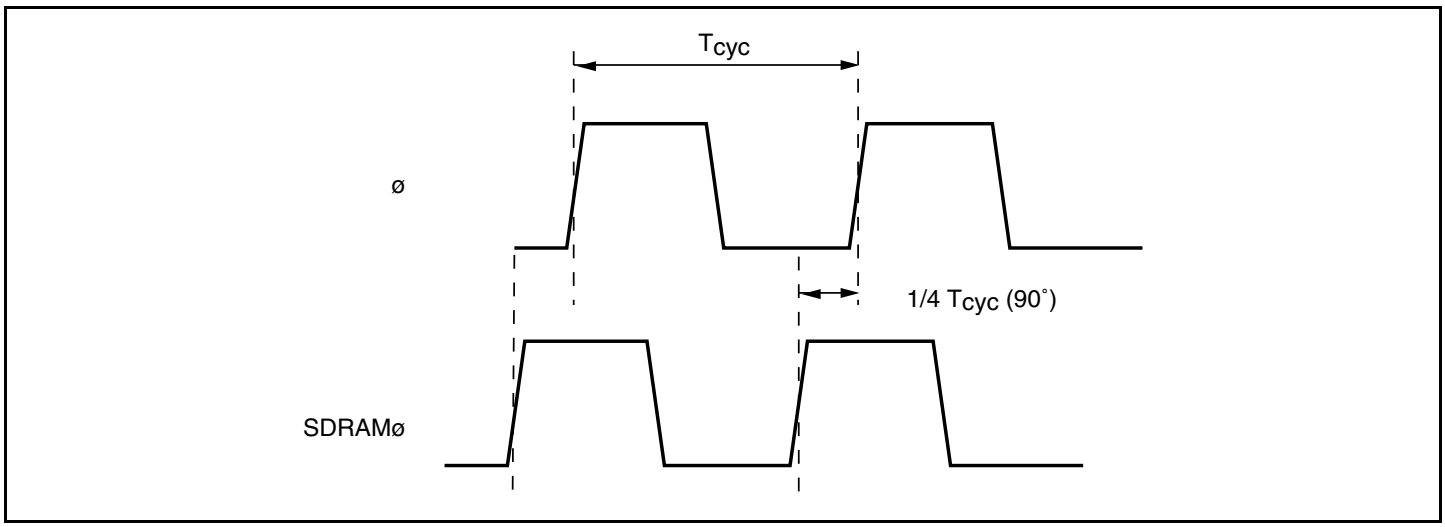


Figure 6.43 Relationship between ϕ and SDRAMϕ (when PLL frequency multiplication factor is ×1 or ×2)

6.7.6 Basic Operation Timing

The four states of the basic timing consist of one T_p (precharge cycle) state, one T_r (row address output cycle) state, and the T_{c1} and two T_{c2} (column address output cycle) states.

When areas 2 to 5 are set for the continuous synchronous DRAM space, settings of the WAITE bit of BCR, RAST, CAST, RCDM bits of DRAMCR, and the CBRM bit of REFCR are ignored.

Figure 6.44 shows the basic timing for synchronous DRAM.

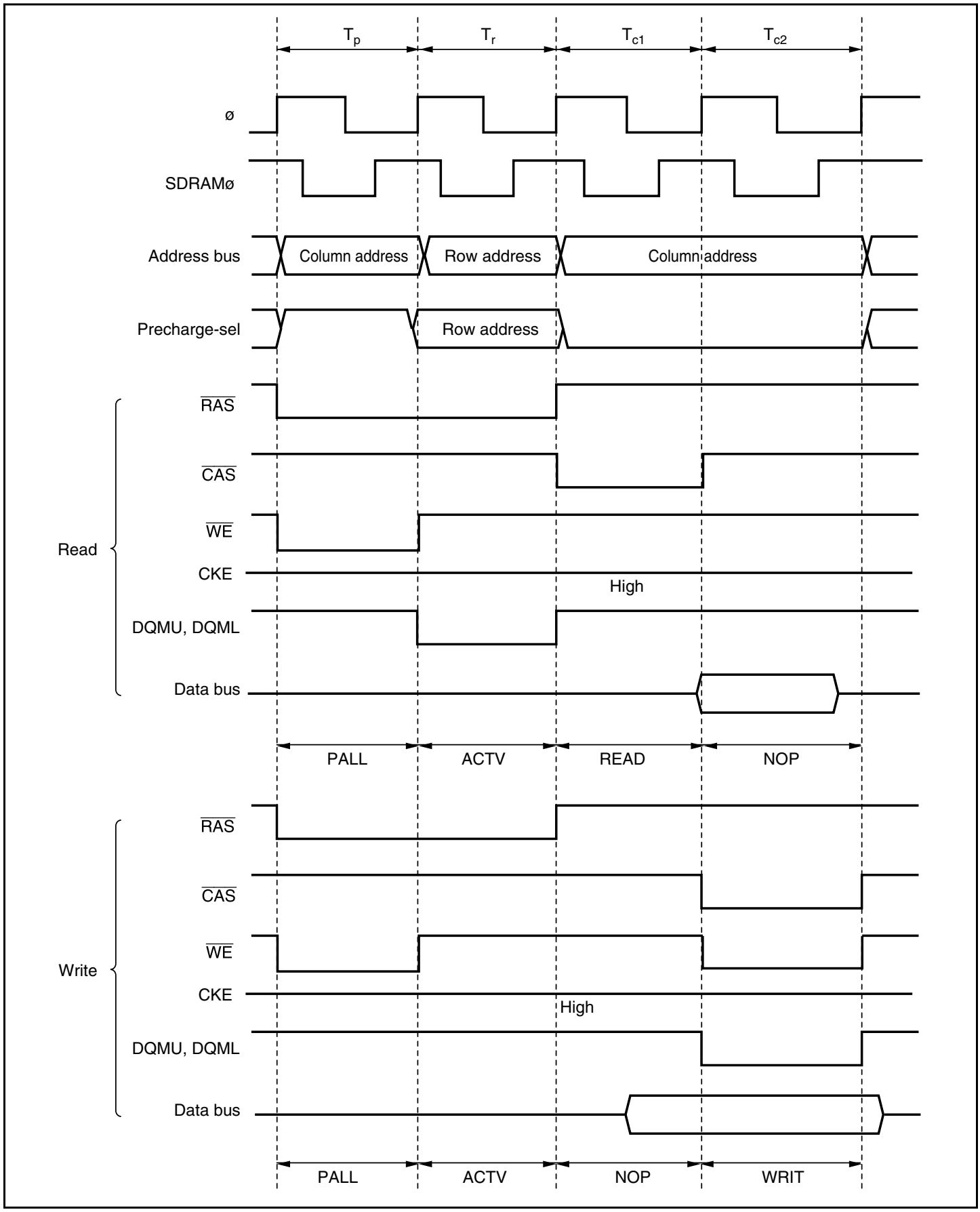


Figure 6.44 Basic Access Timing of Synchronous DRAM (CAS Latency 1)

6.7.7 CAS Latency Control

CAS latency is controlled by settings of the W22 to W20 bits of WTCRB. Set the CAS latency count, as shown in table 6.10, by the setting of synchronous DRAM. Depending on the setting, the CAS latency control cycle (T_{cl}) is inserted. WTCRB can be set regardless of the setting of the AST2 bit of ASTCR. Figure 6.45 shows the CAS latency control timing when synchronous DRAM of CAS latency 3 is connected.

The initial value of W22 to W20 is H'7. Set the register according to the CAS latency of synchronous DRAM to be connected.

Table 6.10 Setting CAS Latency

W22	W21	W20	Description	CAS Latency Control Cycle Inserted
0	0	0	Connect synchronous DRAM of CAS latency 1	0 state
		1	Connect synchronous DRAM of CAS latency 2	1 state
	1	0	Connect synchronous DRAM of CAS latency 3	2 states
		1	Connect synchronous DRAM of CAS latency 4	3 states
1	0	0	Reserved (must not used)	—
		1	Reserved (must not used)	—
	1	0	Reserved (must not used)	—
		1	Reserved (must not used)	—

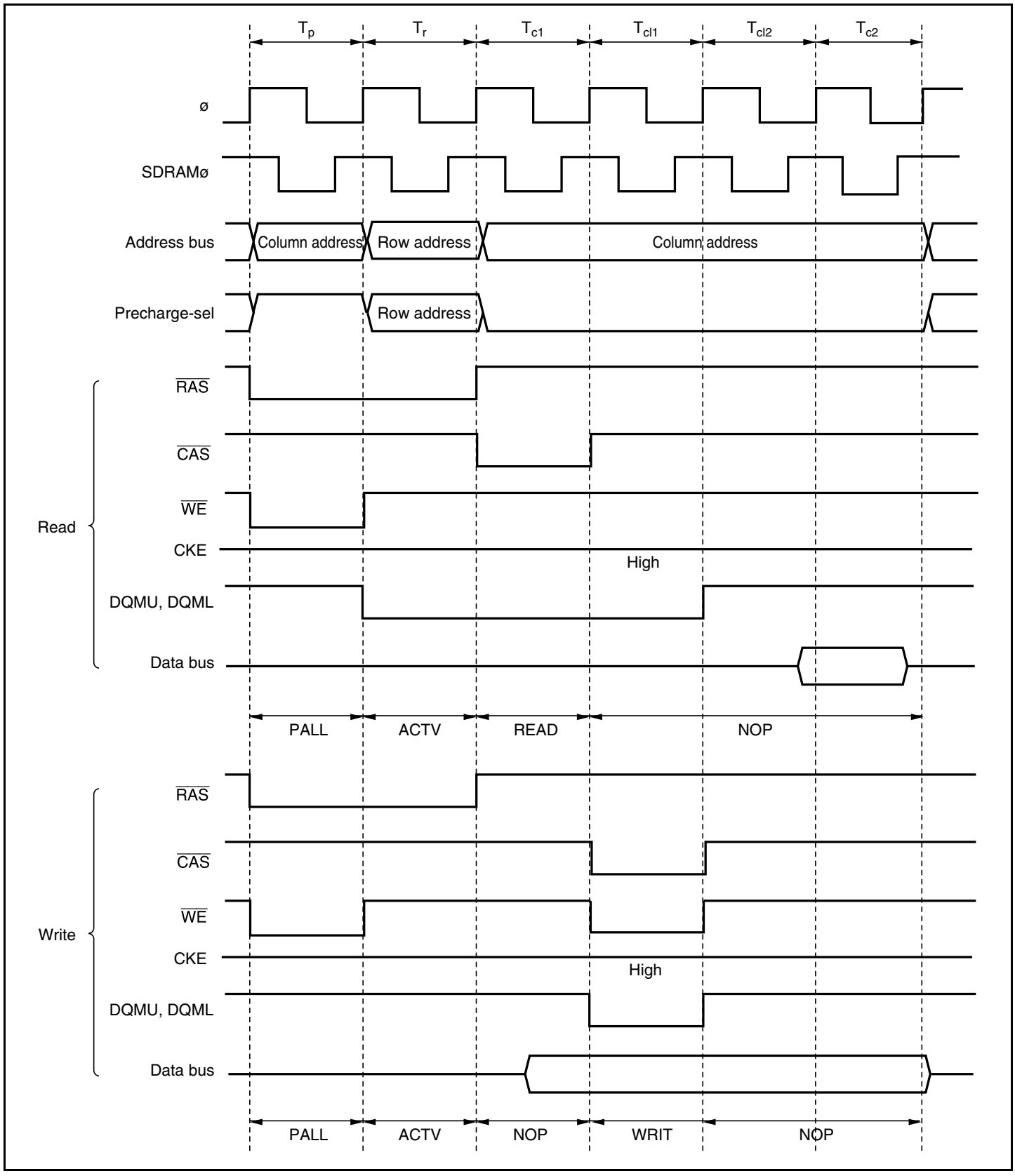


Figure 6.45 CAS Latency Control Timing (SDWCD = 0, CAS Latency 3)

6.7.8 Row Address Output State Control

When the command interval specification from the ACTV command to the next READ/WRIT command cannot be satisfied, 1 to 3 states (Trw) that output the NOP command can be inserted between the Tr cycle that outputs the ACTV command and the Tc1 cycle that outputs the column address by setting the RCD1 and RCD0 bits of DRACCR. Use the optimum setting for the wait time according to the synchronous DRAM connected and the operating frequency of this LSI. Figure 6.46 shows an example of the timing when the one Trw state is set.

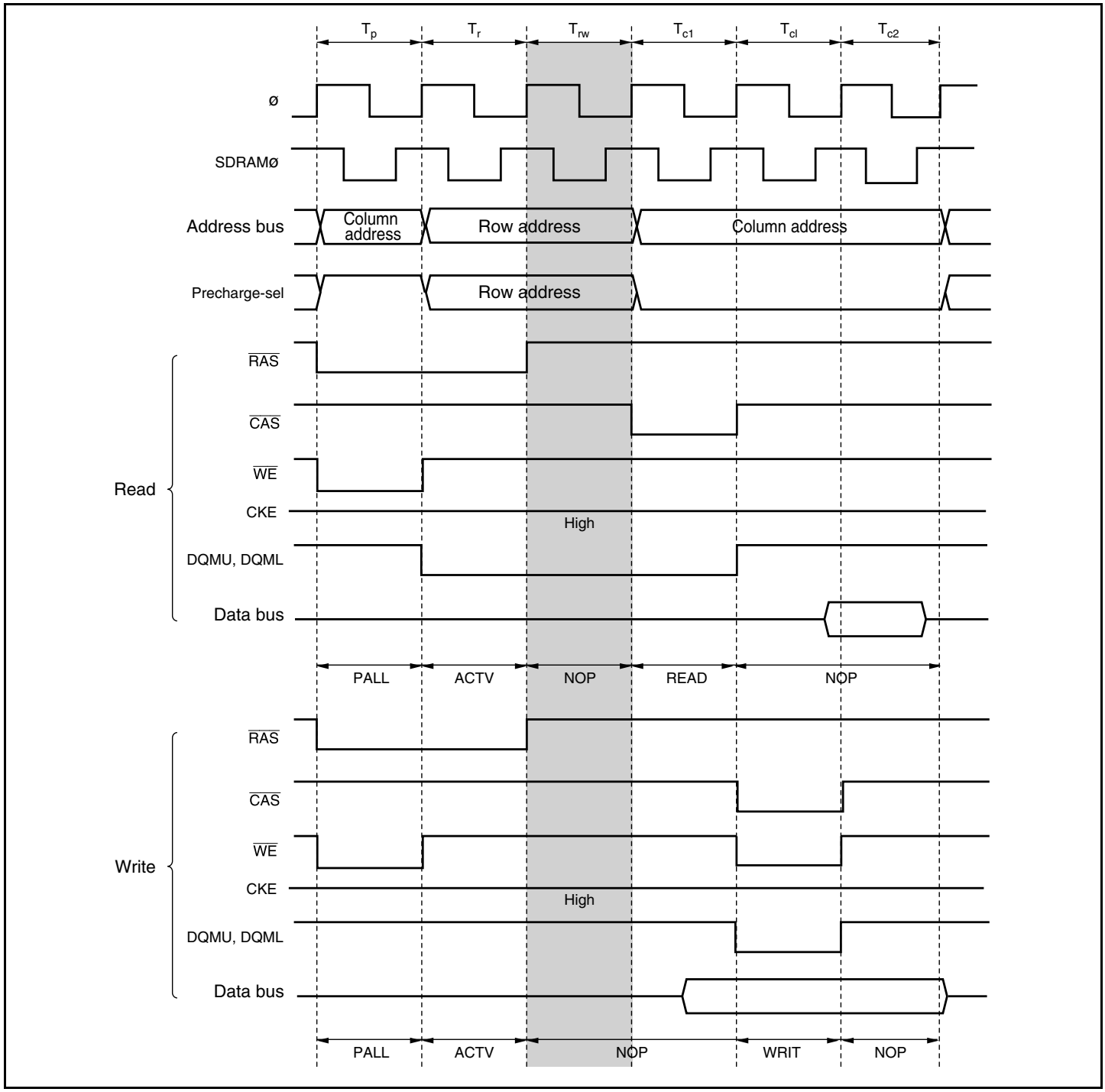


Figure 6.46 Example of Access Timing when Row Address Output Hold State is 1 State (RCD1 = 0, RCD0 = 1, SDWCD = 0, CAS Latency 2)

6.7.9 Precharge State Count

When the interval specification from the PALL command to the next ACTV/REF command cannot be satisfied, from one to four T_p states can be selected by setting bits TPC1 and TPC0 in DRACCR. Set the optimum number of T_p cycles according to the synchronous DRAM connected and the operating frequency of this LSI. Figure 6.47 shows the timing when two Tp states are inserted.

The setting of bits TPC1 and TPC0 is also valid for T_p states in refresh cycles.

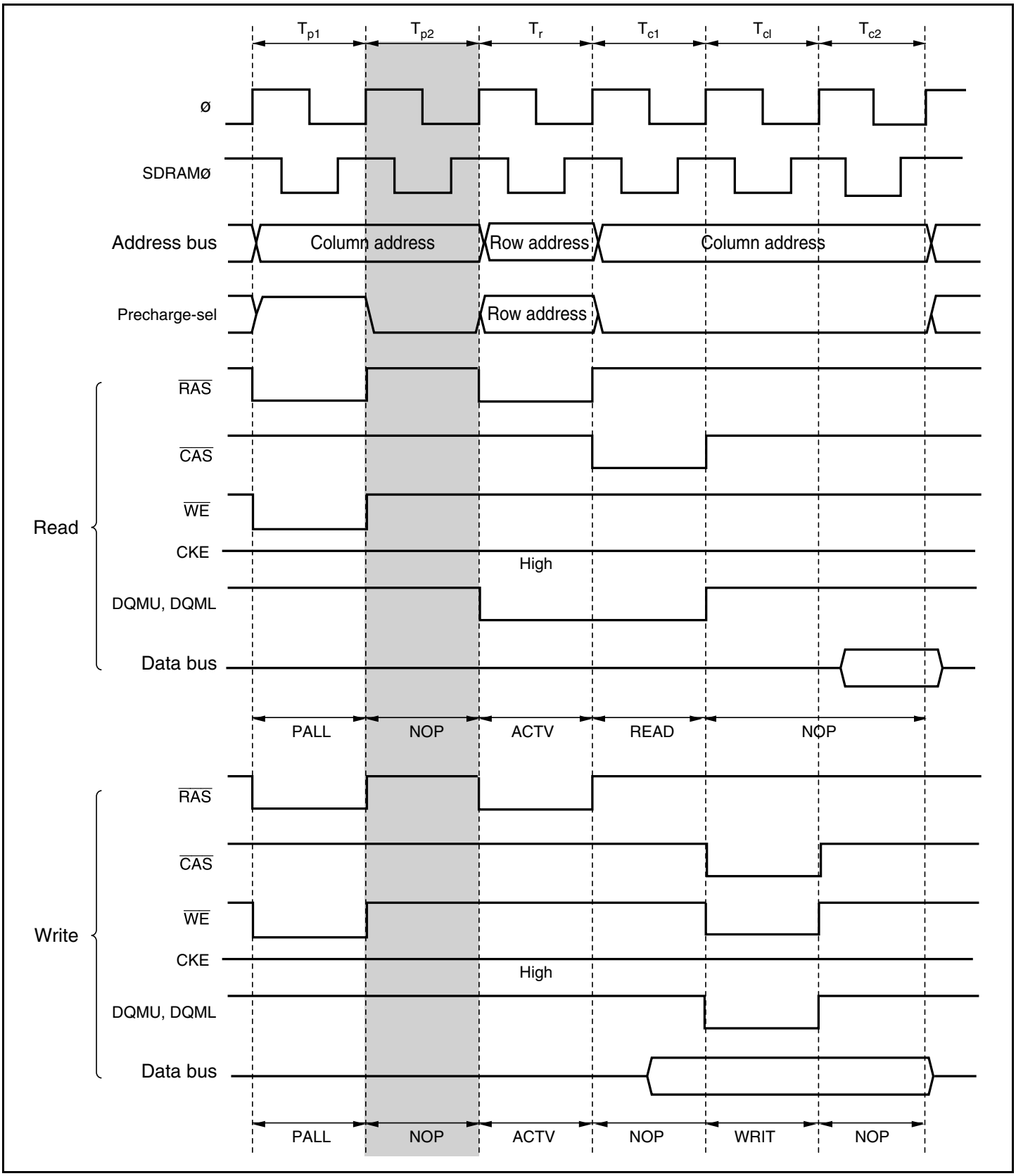


Figure 6.47 Example of Timing with Two-State Precharge Cycle (TPC1 = 0, TPC0 = 1, SDWCD = 0, CAS Latency 2)

6.7.10 Bus Cycle Control in Write Cycle

By setting the SDWCD bit of the DRACCR to 1, the CAS latency control cycle (Tc1) that is inserted by the WTCRB register in the write access of the synchronous DRAM can be disabled. Disabling the CAS latency control cycle can reduce the write-access cycle count as compared to synchronous DRAM read access. Figure 6.48 shows the write access timing when the CAS latency control cycle is disabled.

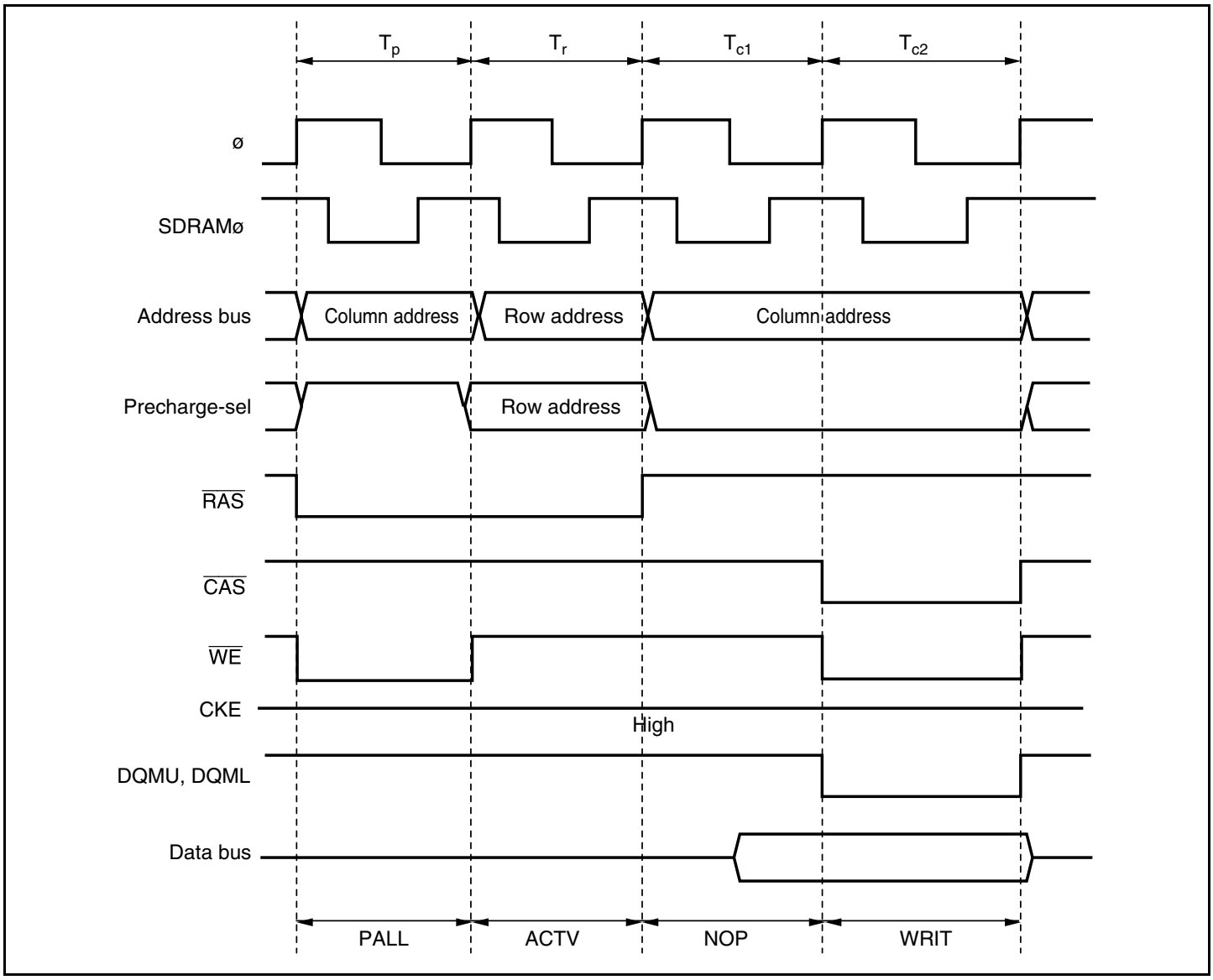


Figure 6.48 Example of Write Access Timing when CAS Latency Control Cycle is Disabled (SDWCD = 1)

6.7.11 Byte Access Control

When synchronous DRAM with a ×16-bit configuration is connected, DQMU and DQML are used for the control signals needed for byte access.

Figures 6.49 and 6.50 show the control timing for DQM, and figure 6.51 shows an example of connection of byte control by DQMU and DQML.

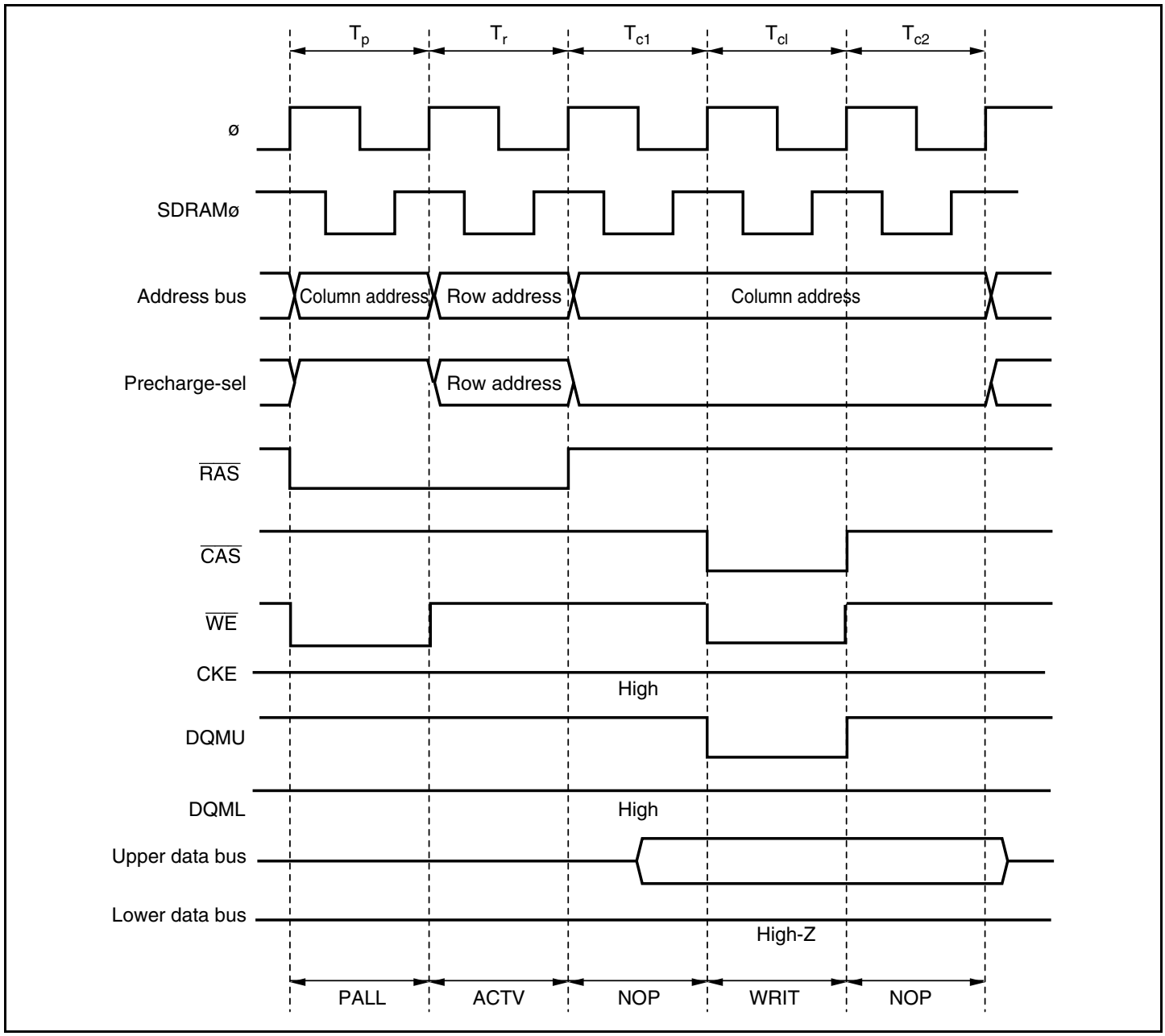


Figure 6.49 DQMU and DQML Control Timing (Upper Byte Write Access: SDWCD = 0, CAS Latency 2)

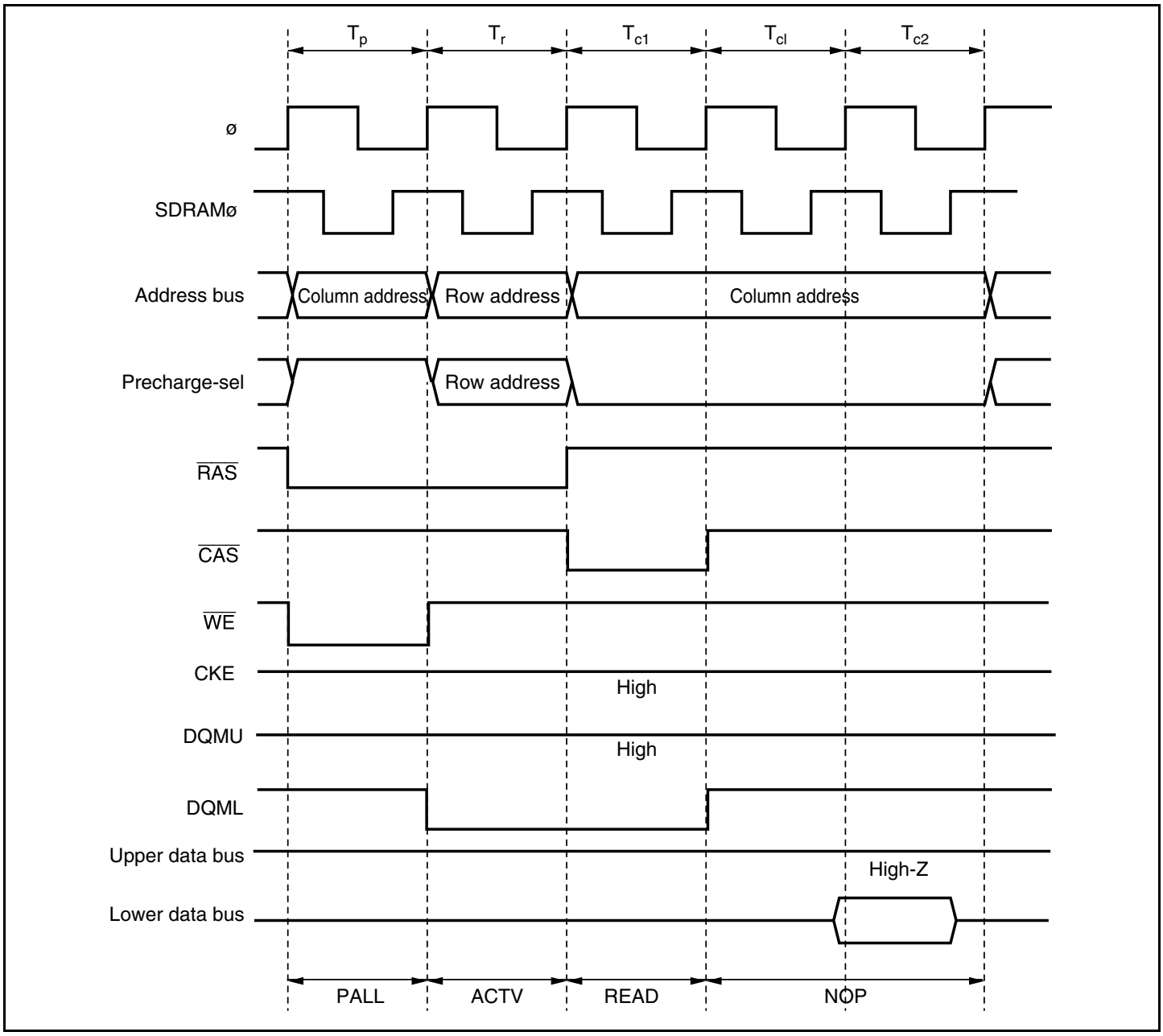


Figure 6.50 DQMU and DQML Control Timing (Lower Byte Read Access: CAS Latency 2)

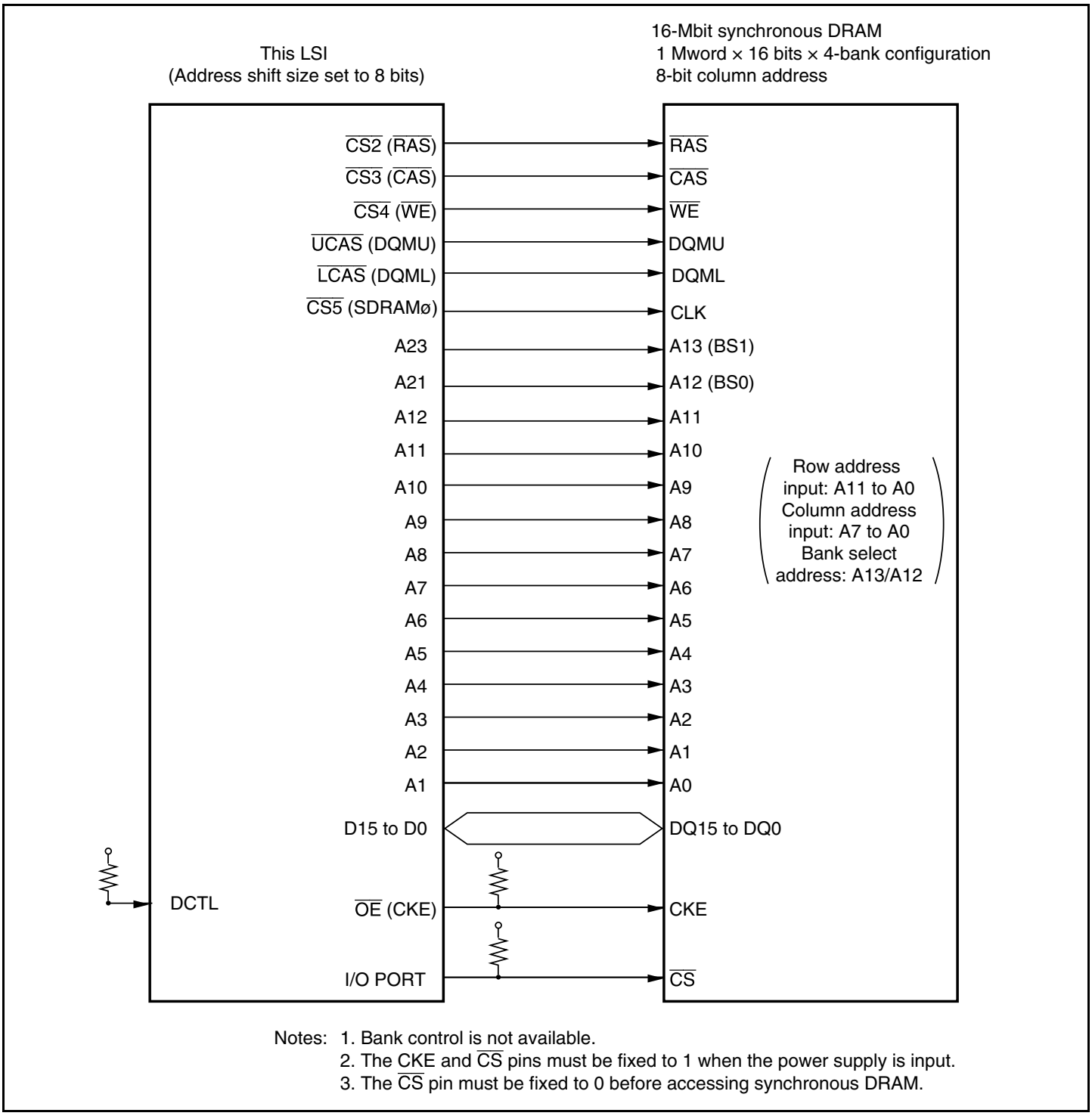


Figure 6.51 Example of DQMU and DQML Byte Control

6.7.12 Burst Operation

With synchronous DRAM, in addition to full access (normal access) in which data is accessed by outputting a row address for each access, burst access is also provided which can be used when making consecutive accesses to the same row address. This access enables fast access of data by simply changing the column address after the row address has been output. Burst access can be selected by setting the BE bit to 1 in DRAMCR.

DQM has the 2-cycle latency when synchronous DRAM is read. Therefore, the DQM signal cannot be specified to the Tc2 cycle data output if Tc1 cycle is performed for second or following column address when the CAS latency is set to 1 to issue the READ command. Do not set the BE bit to 1 when synchronous DRAM of CAS latency 1 is connected.

Burst Access Operation Timing: Figure 6.52 shows the operation timing for burst access. When there are consecutive access cycles for continuous synchronous DRAM space, the column address output cycles continue as long as the row address is the same for consecutive access cycles. The row address used for the comparison is set with bits MXC2 to MXC0 in DRAMCR.

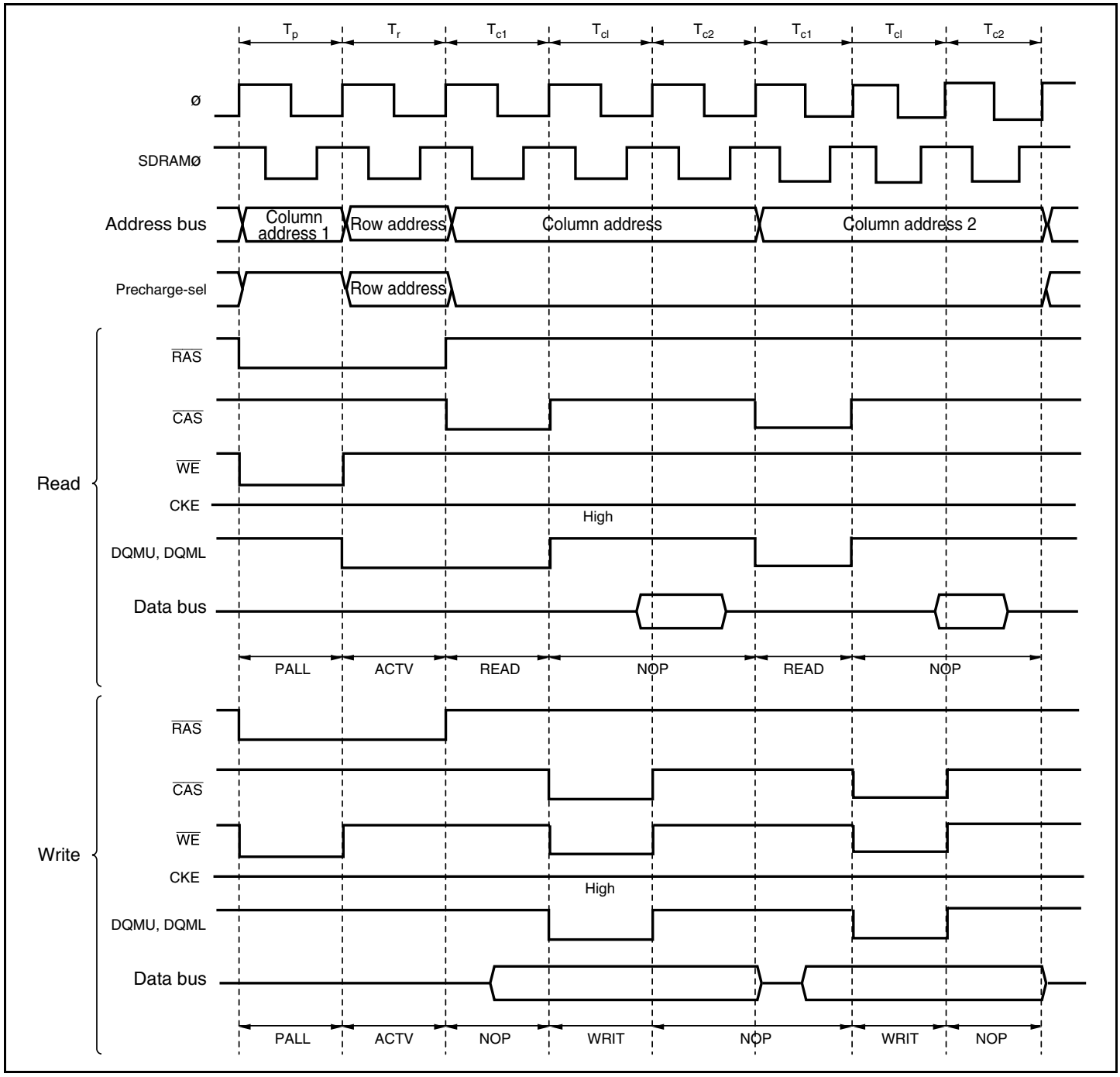


Figure 6.52 Operation Timing of Burst Access (BE = 1, SDWCD = 0, CAS Latency 2)

RAS Down Mode: Even when burst operation is selected, it may happen that access to continuous synchronous DRAM space is not continuous, but is interrupted by access to another space. In this case, if the row address active state is held during the access to the other space, the read or write command can be issued without ACTV command generation similarly to DRAM RAS down mode.

To select RAS down mode, set the BE bit to 1 in DRAMCR regardless of the RCDM bit settings. The operation corresponding to DRAM RAS up mode is not supported by this LSI.

Figure 6.53 shows an example of the timing in RAS down mode.

Note, however, the next continuous synchronous DRAM space access is a full access if:

- a refresh operation is initiated in the RAS down state
- self-refreshing is performed
- the chip enters software standby mode
- the external bus is released
- the BE bit is cleared to 0
- the mode register of the synchronous DRAM is set

There is synchronous DRAM in which time of the active state of each bank is restricted. If it is not guaranteed that other row address are accessed in a period in which program execution ensures the value (software standby, sleep, etc.), auto refresh or self refresh must be set, and the restrictions of the maximum active state time of each bank must be satisfied. When refresh is not used, programs must be developed so that the bank is not in the active state for more than the specified time.

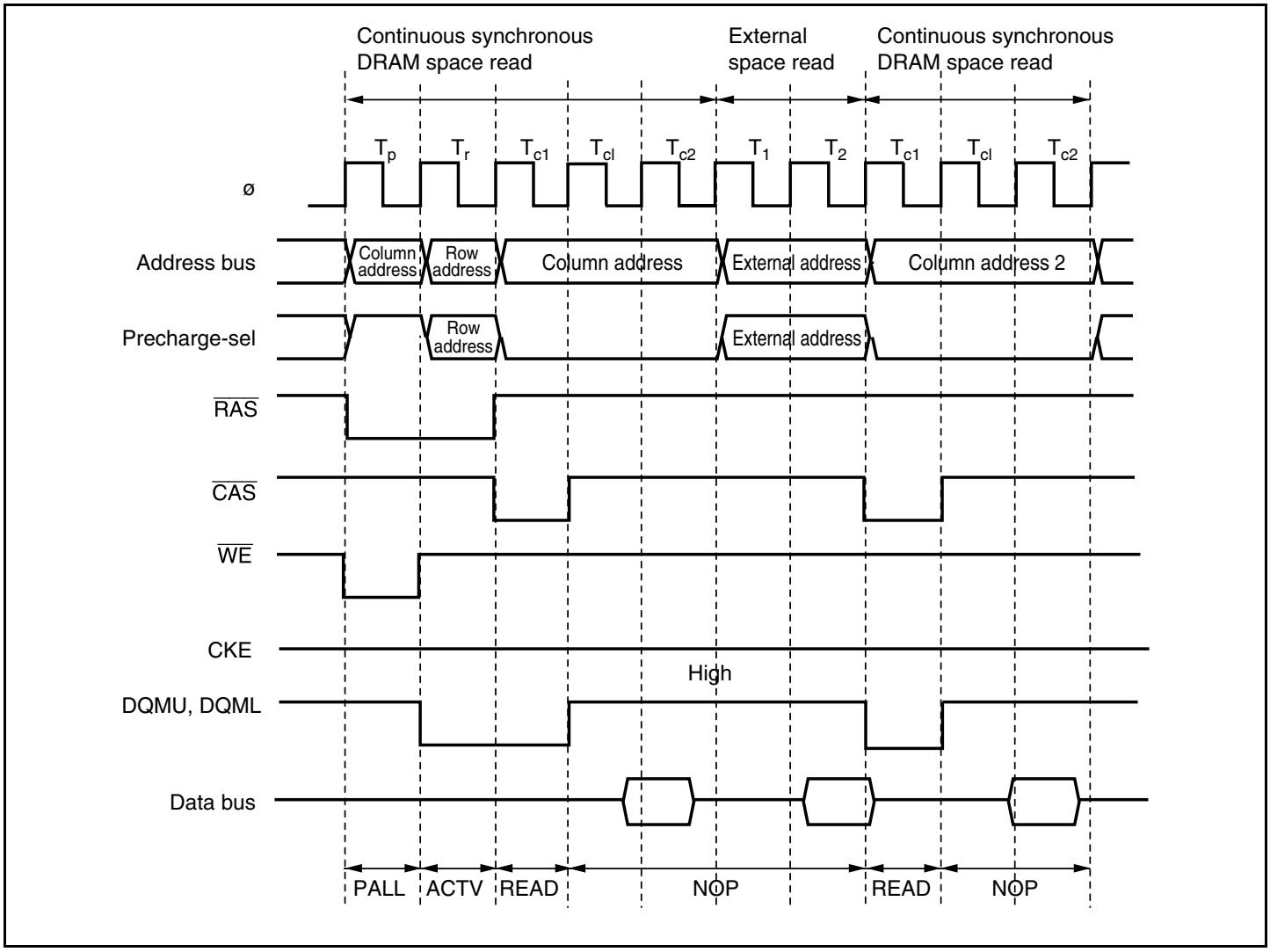


Figure 6.53 Example of Operation Timing in RAS Down Mode (BE = 1, CAS Latency 2)

6.7.13 Refresh Control

This LSI is provided with a synchronous DRAM refresh control function. Auto refreshing is used. In addition, self-refreshing can be executed when the chip enters the software standby state.

Refresh control is enabled when any area is designated as continuous synchronous DRAM space in accordance with the setting of bits RMTS2 to RMTS0 in DRAMCR.

Auto Refreshing: To select auto refreshing, set the RFSHE bit to 1 in REFCR.

With auto refreshing, RTCNT counts up using the input clock selected by bits RTCK2 to RTCK0 in REFCR, and when the count matches the value set in RTCOR (compare match), refresh control is performed. At the same time, RTCNT is reset and starts counting up again from H'00. Refreshing is thus repeated at fixed intervals determined by RTCOR and bits RTCK2 to RTCK0. Set a value in RTCOR and bits RTCK2 to RTCK0 that will meet the refreshing interval specification for the synchronous DRAM used.

When bits RTCK2 to RTCK0 are set, RTCNT starts counting up. RTCNT and RTCOR settings should therefore be completed before setting bits RTCK2 to RTCK0. Auto refresh timing is shown in figure 6.54.

Since the refresh counter operation is the same as the operation in the DRAM interface, see section 6.6.12, Refresh Control.

When the continuous synchronous DRAM space is set, access to external space other than continuous synchronous DRAM space cannot be performed in parallel during the auto refresh period, since the setting of the CBRM bit of REFCR is ignored.

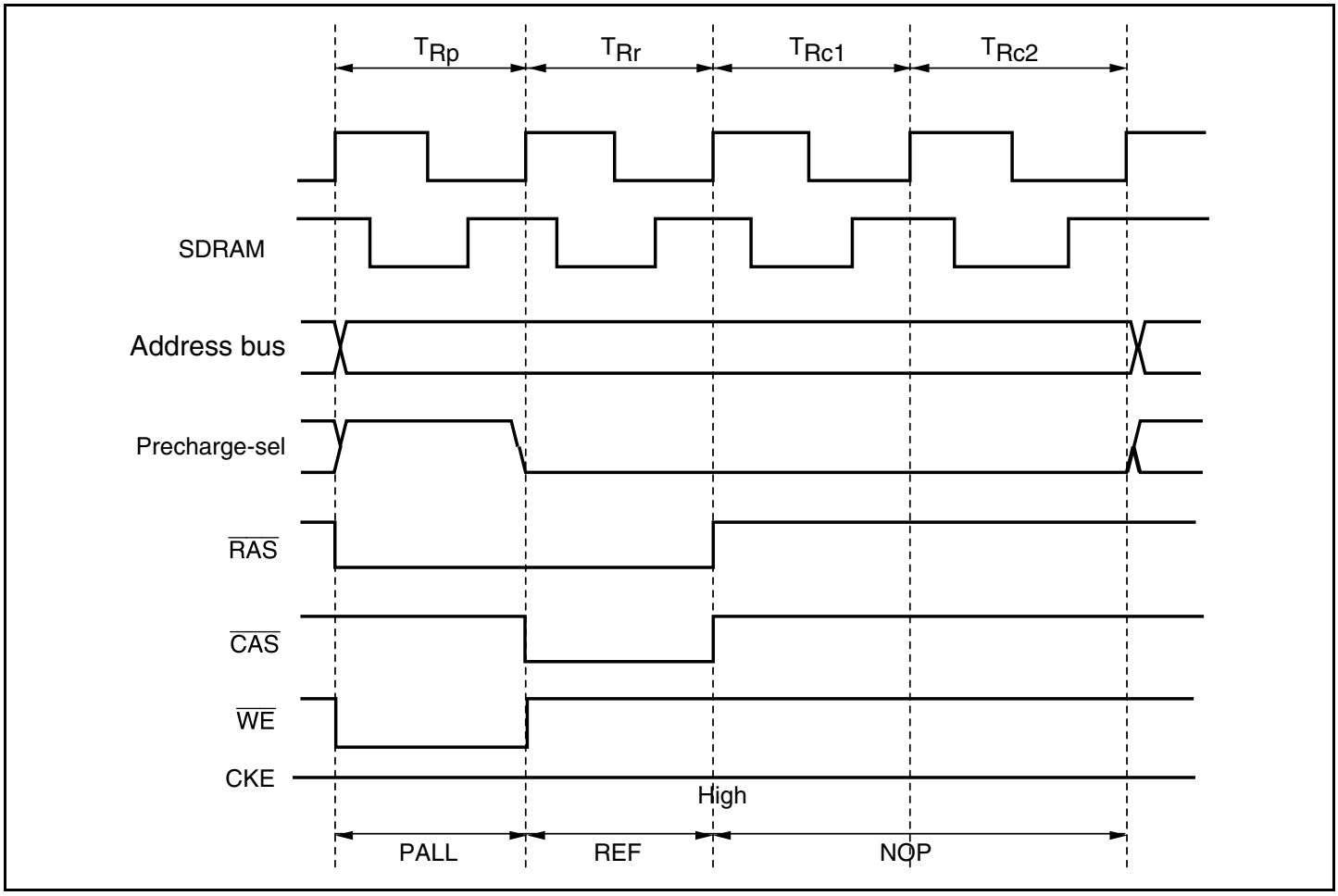


Figure 6.54 Auto Refresh Timing

When the interval specification from the PLL command to the REF command cannot be satisfied, setting the RCW1 and RCW0 bits of REFCR enables one to three wait states to be inserted after the T_{Rp} cycle that is set by the TPC1 and TPC0 bits of DRACCR. Set the optimum number of waits according to the synchronous DRAM connected and the operating frequency of this LSI. Figure 6.55 shows the timing when one wait state is inserted. Since the setting of bits TPC1 and TPC0 of DRACCR is also valid in refresh cycles, the command interval can be extended by the RCW1 and RCW0 bits after the precharge cycles.

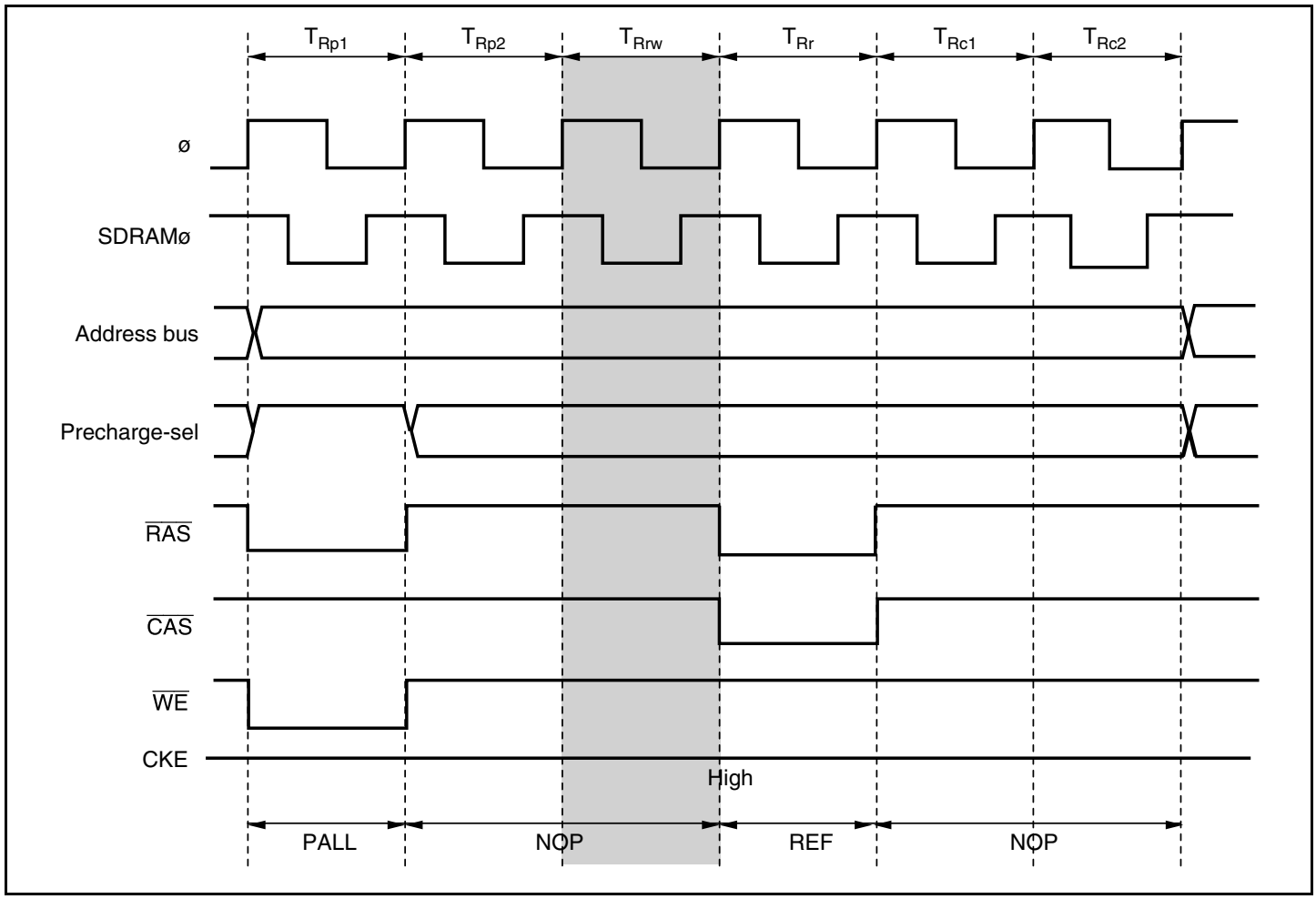


Figure 6.55 Auto Refresh Timing
(TPC = 1, TPC0 = 1, RCW1 = 0, RCW0 = 1)

When the interval specification from the REF command to the ACTV cannot be satisfied, setting the RLW1 and RLW0 bits of REFCR enables one to three wait states to be inserted in the refresh cycle. Set the optimum number of waits according to the synchronous DRAM connected and the operating frequency of this LSI. Figure 6.56 shows the timing when one wait state is inserted.

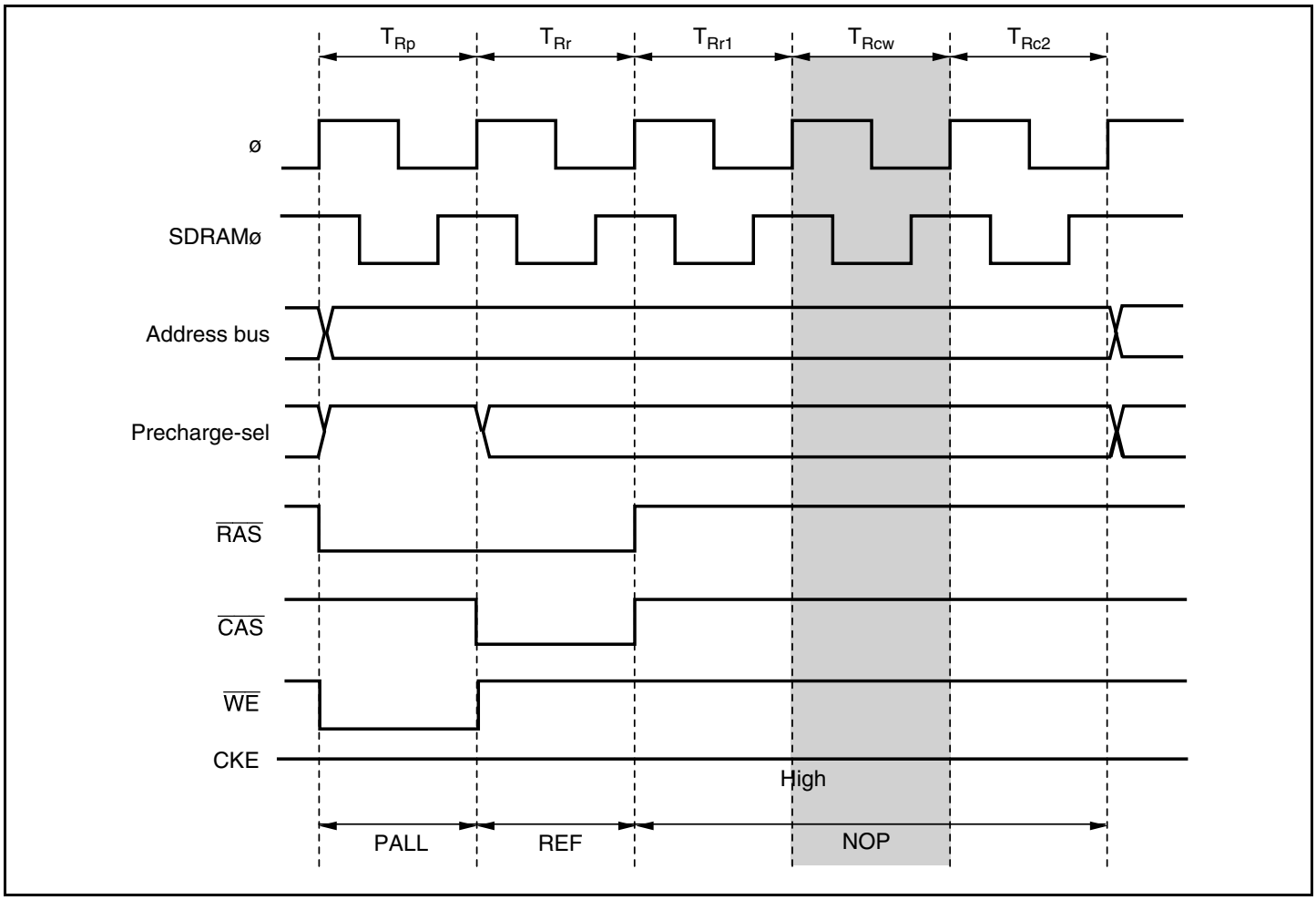


Figure 6.56 Auto Refresh Timing
(TPC = 0, TPC0 = 0, RLW1 = 0, RLW0 = 1)

Self-Refreshing:

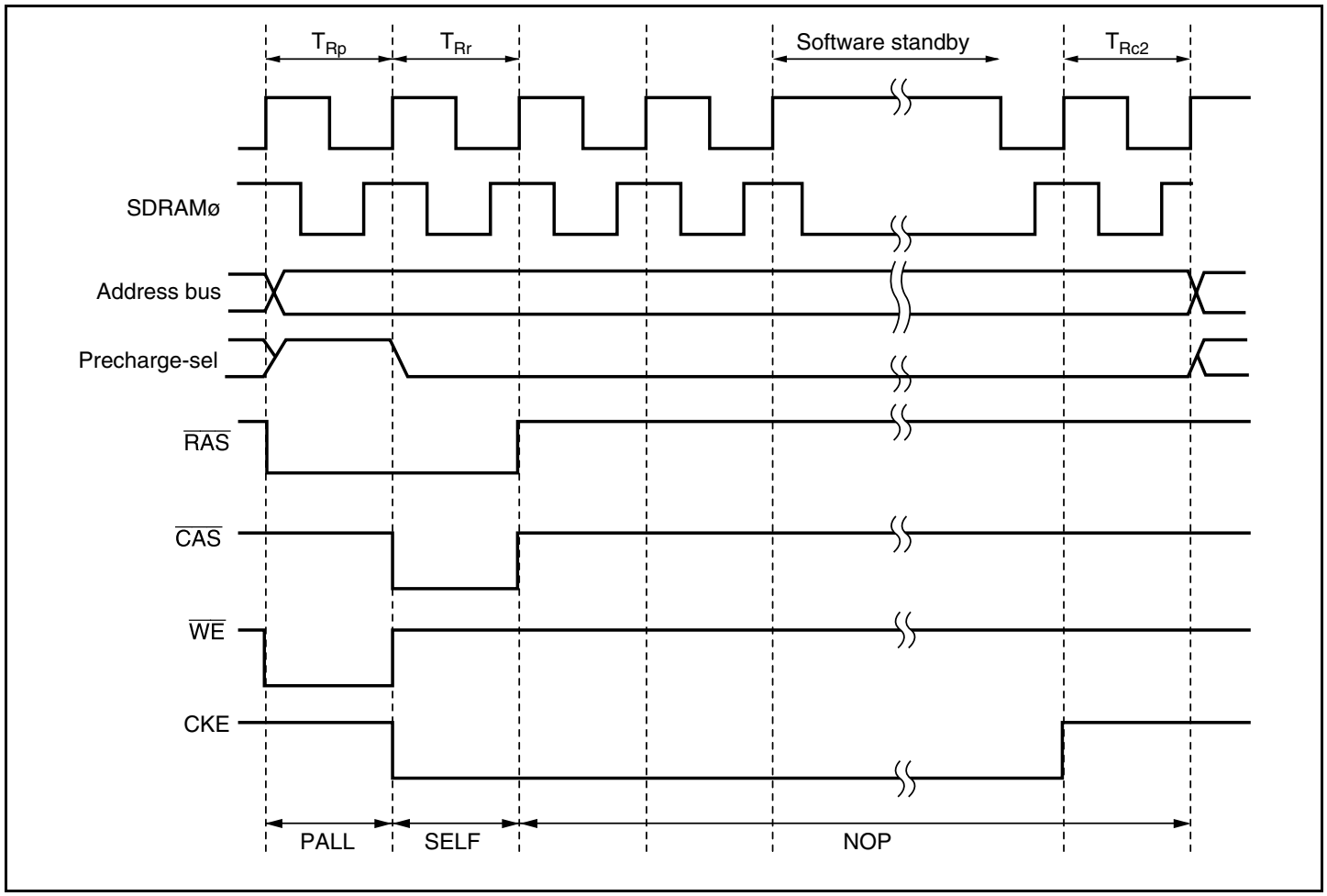


Figure 6.57 Self-Refresh Timing
(TPC1 = 1, TPC0 = 0, RCW1 = 0, RCW0 = 0, RLW1 = 0, RLW0 = 0)

In some synchronous DRAMs provided with a self-refresh mode, the interval between clearing self-refreshing and the next command is specified. A setting can be made in bits TPCS2 to TPCS0 in REFCR to make the precharge time after self-refreshing from 1 to 7 states longer than the normal precharge time. In this case, too, normal precharging is performed according to the setting of bits TPC1 and TPC0 in DRACCR, and therefore a setting should be made to give the optimum post-self-refresh precharge time, including this time. Figure 6.58 shows an example of the timing when the precharge time after self-refreshing is extended by 2 states.

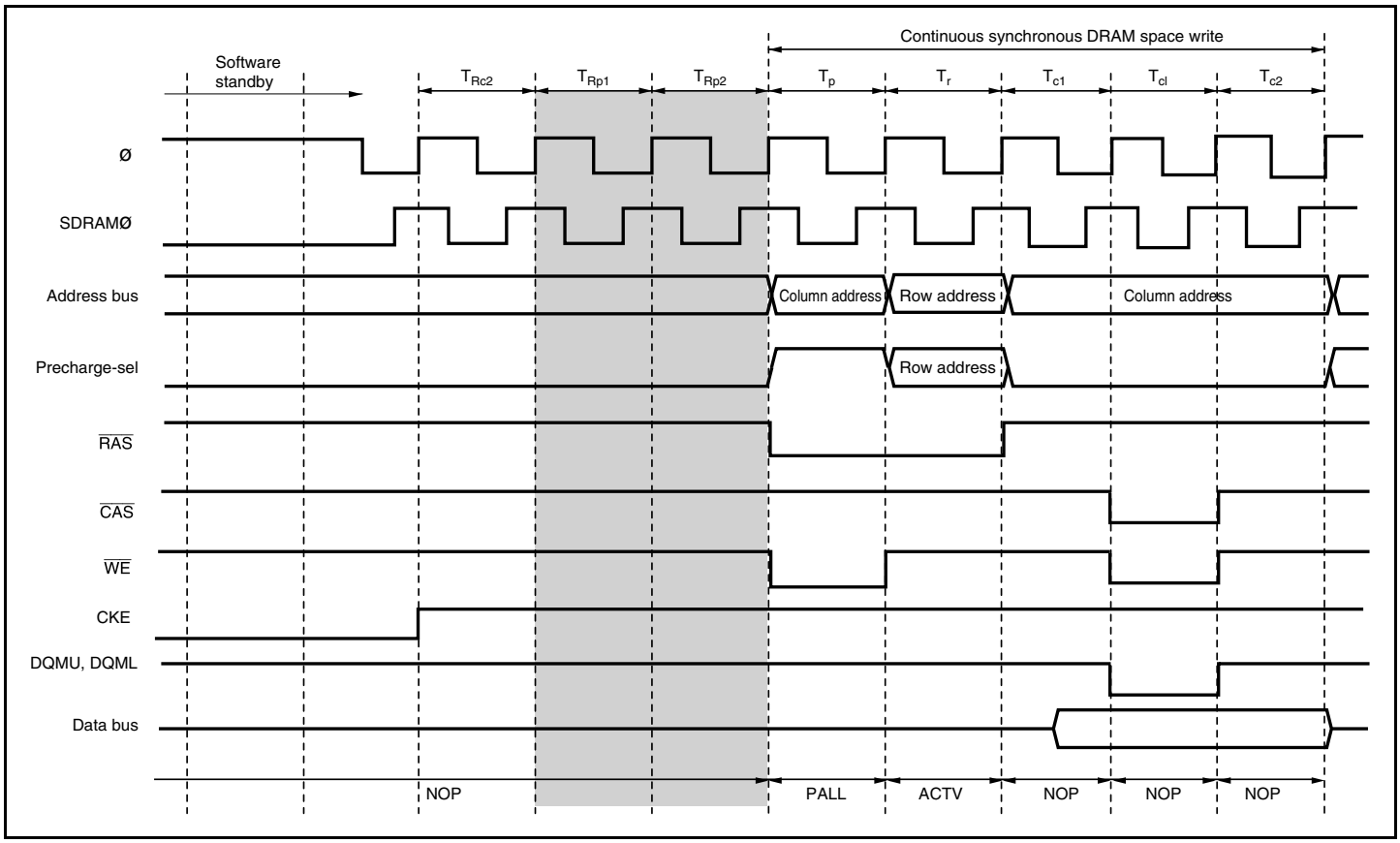


Figure 6.58 Example of Timing when Precharge Time after Self-Refreshing Is Extended by 2 States (TPCS2 to TPCS0 = H'2, TPC1 = 0, TPC0 = 0, CAS Latency 2)

Refreshing and All-Module-Clocks-Stopped Mode: In this LSI, if the ACSE bit is set to 1 in MSTPCRH, and then a SLEEP instruction is executed with the setting for all peripheral module clocks to be stopped (MSTPCR = H'FFFF) or for operation of the 8-bit timer module alone (MSTPCR = H'FFFE), and a transition is made to the sleep state, the all-module-clocks-stopped mode is entered, in which the bus controller and I/O port clocks are also stopped.

As the bus controller clock is also stopped in this mode, auto refreshing is not executed. If synchronous DRAM is connected externally and DRAM data is to be retained in sleep mode, the ACSE bit must be cleared to 0 in MSTPCR.

Software Standby: When a transition is made to normal software standby, the PLL command is not output. If synchronous DRAM is connected and DRAM data is to be retained in software standby, self-refreshing must be set.

6.7.14 Mode Register Setting of Synchronous DRAM

To use synchronous DRAM, mode must be set after power-on. to set mode, set the RMTS2 to RMTS0 bits in DRAMCR to H'5 and enable the synchronous DRAM mode register setting. After that, access the continuous synchronous DRAM space in bytes. When the value to be set in the synchronous DRAM mode register is X, value X is set in the synchronous DRAM mode register by writing to the continuous synchronous DRAM space of address H'400000 + X for 8-bit bus

configuration synchronous DRAM and by writing to the continuous synchronous DRAM space of address H'400000 + 2X for 16-bit bus configuration synchronous DRAM.

The value of the address signal is fetched at the issuance time of the MRS command as the setting value of the mode register in the synchronous DRAM. Mode of burst read/burst write in the synchronous DRAM is not supported by this LSI. For setting the mode register of the synchronous DRAM, set the burst read/single write with the burst length of 1. Figure 6.59 shows the setting timing of the mode in the synchronous DRAM.

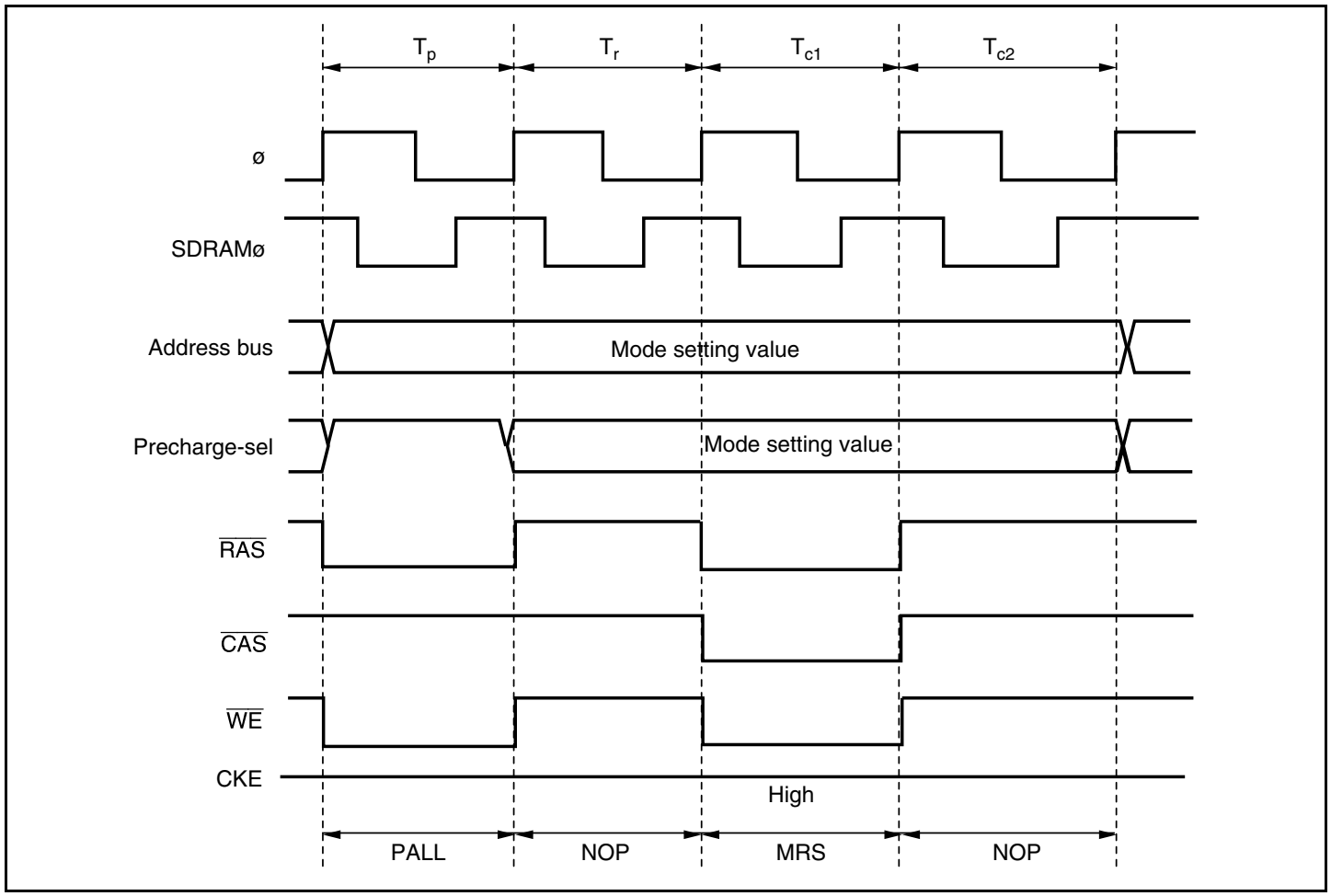


Figure 6.59 Synchronous DRAM Mode Setting Timing

6.7.15 DMAC and EXDMAC Single Address Transfer Mode and Synchronous DRAM Interface

When burst mode is selected on the synchronous DRAM interface, the DACK and EDACK output timing can be selected with the DDS and EDDS bits in DRAMCR. When continuous synchronous DRAM space is accessed in DMAC/EXDMAC single address mode at the same time, these bits select whether or not burst access is to be performed. The establishment time for the read data can be extended in the clock suspend mode irrespective of the settings of the DDS and EDDS bits.

(1) Output Timing of DACK or EDACK

When DDS = 1 or EDDS = 1: Burst access is performed by determining the address only, irrespective of the bus master. With the synchronous DRAM interface, the DACK or EDACK output goes low from the T_{c1} state.

Figure 6.60 shows the DACK or EDACK output timing for the synchronous DRAM interface when DDS = 1 or EDDS = 1.

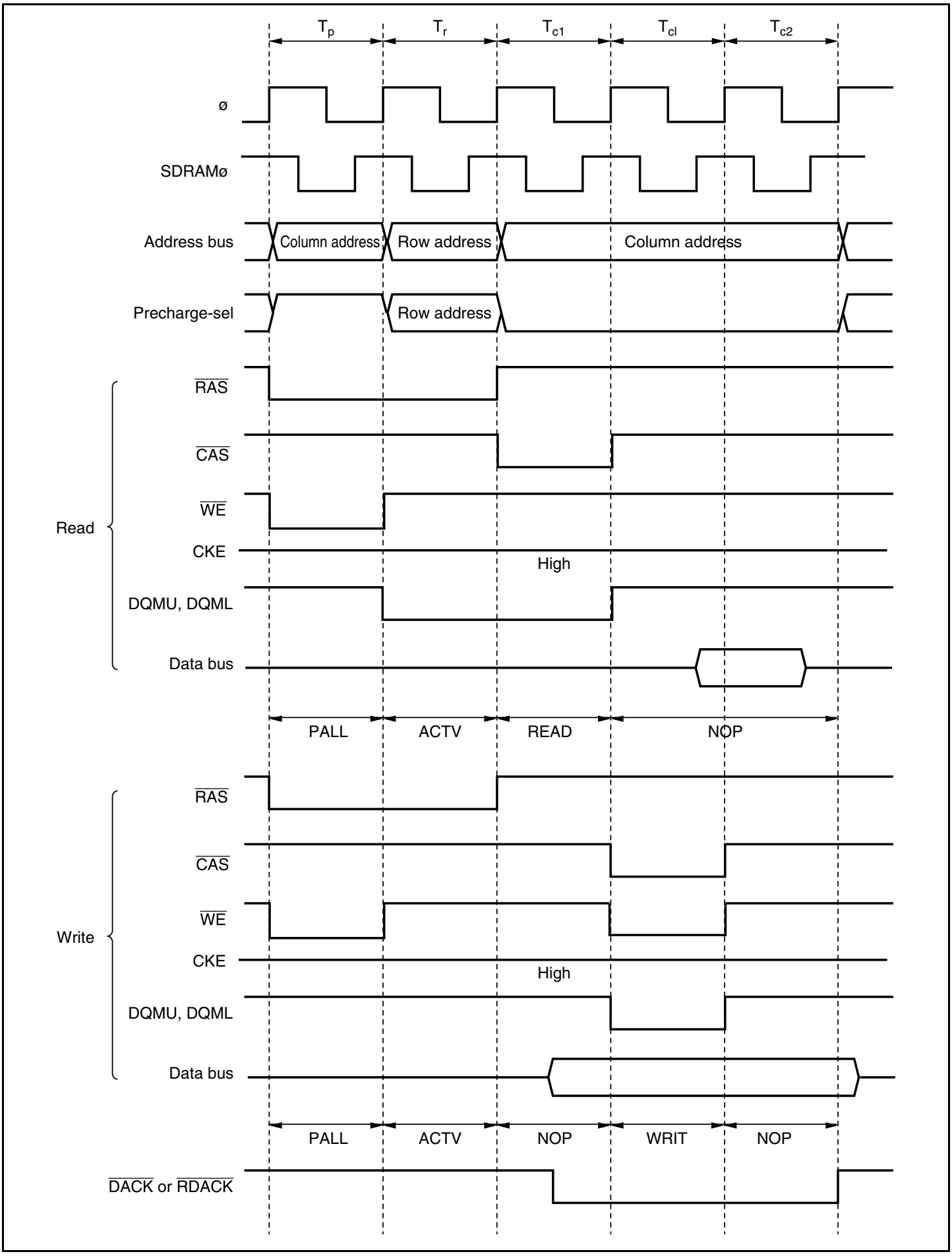


Figure 6.60 Example of DACK/EDACK Output Timing when DDS = 1 or EDDS = 1

When DDS = 0 or EDDS = 0: When continuous synchronous DRAM space is accessed in DMAC or EXDMAC single address transfer mode, full access (normal access) is always performed. With the synchronous DRAM interface, the DACK or EDACK output goes low from the T_r state.

In modes other than DMAC or EXDMAC single address transfer mode, burst access can be used when accessing continuous synchronous DRAM space.

Figure 6.61 shows the DACK or EDACK output timing for the synchronous DRAM interface when DDS = 0 or EDDS = 0.

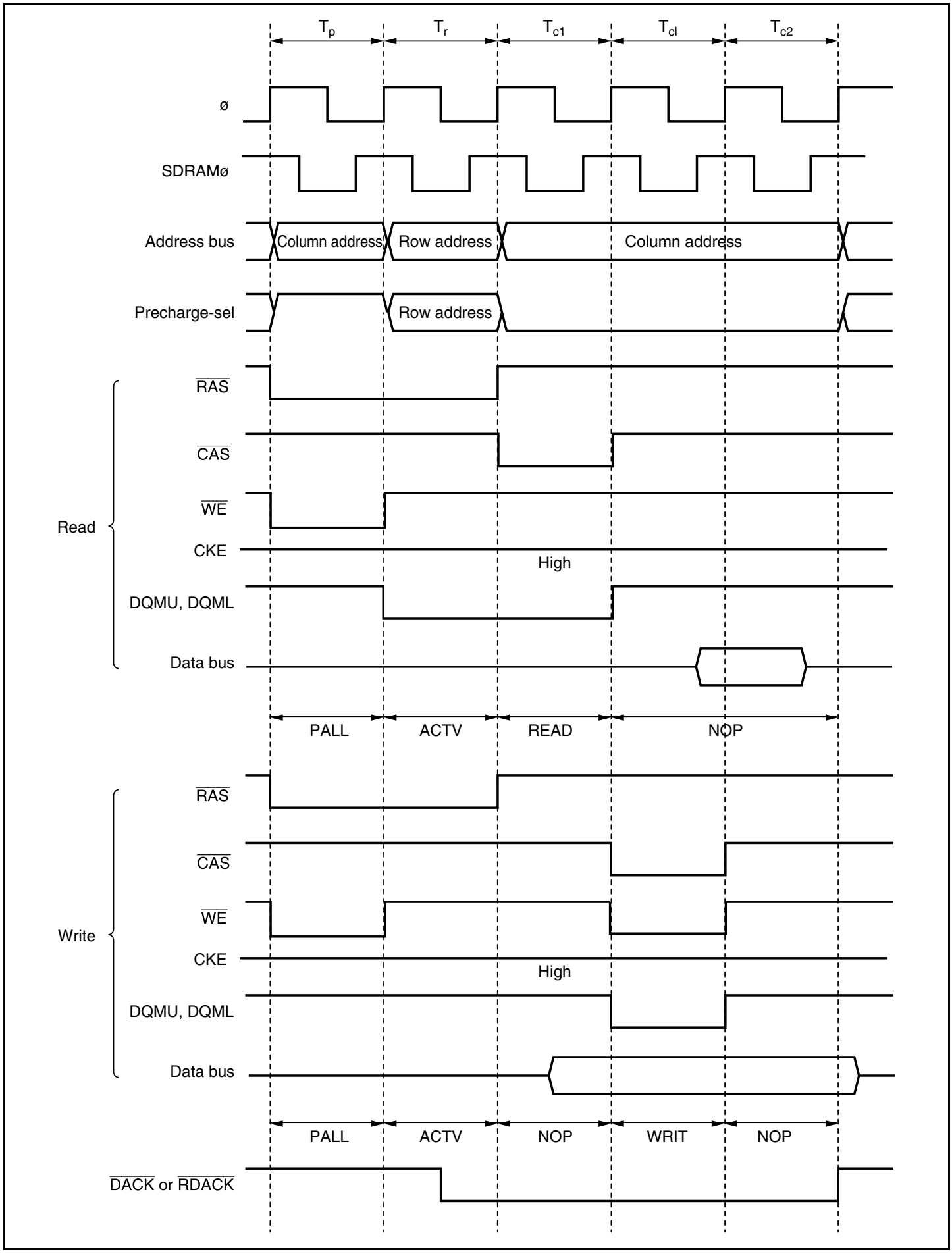


Figure 6.61 Example of DACK/EDACK Output Timing when DDS = 0 or EDDS = 0

(2) Read Data Extension

If the CKSPE bit is set to 1 in DRACCR when the continuous synchronous DRAM space is read-accessed in DMAC/EXDMAC single address mode, the establishment time for the read data can be extended by clock suspend mode. The number of states for insertion of the read data extension cycle (Tsp) is set in bits RDXC1 and RDXC0 in DRACCR. Be sure to set the OEE bit to 1 in DRAMCR when the read data will be extended. The extension of the read data is not in accordance with the bits DDS and EDDS.

Figure 6.62 shows the timing chart when the read data is extended by two cycles.

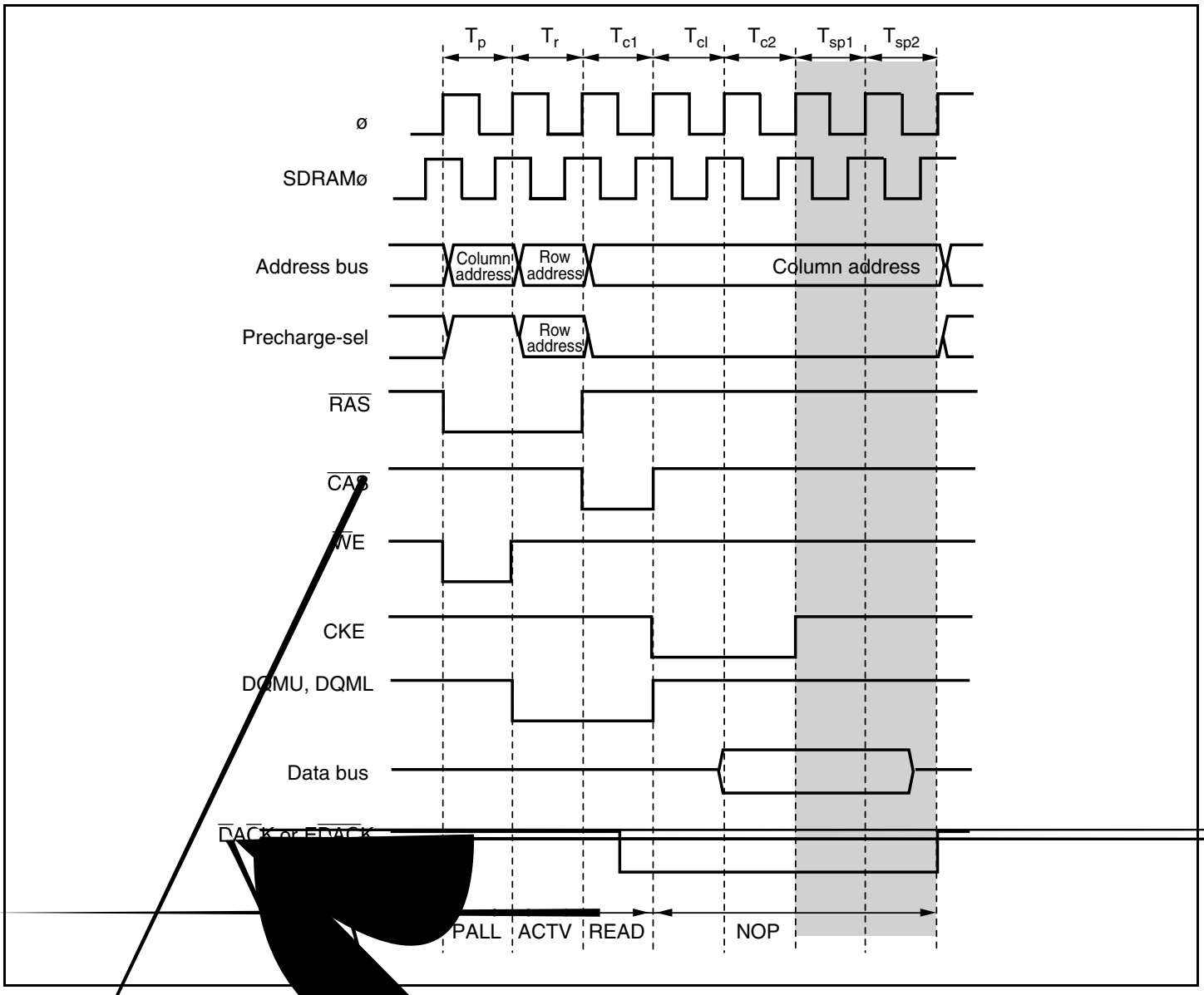


Figure 6.62 Example of Timing when the Read Data is Extended by Two States (DDS = 1, or EDDS = 1, RDXC1 = 0, RDXC0 = 1, CAS Latency 2)

6.8 Burst ROM Interface

In this LSI, external space areas 0 and 1 can be designated as burst ROM space, and burst ROM interfacing performed. The burst ROM space interface enables ROM with burst access capability to be accessed at high speed.

Areas 1 and 0 can be designated as burst ROM space by means of bits BSRM1 and BSRM0 in BROMCR. Continuous burst accesses of 4, 8, 16, or 32 words can be performed, according to the setting of the BSWD11 and BSWD10 bits in BROMCR. From 1 to 8 states can be selected for burst access.

Settings can be made independently for area 0 and area 1.

In burst ROM interface space, burst access covers only CPU read accesses.

6.8.1 Basic Timing

The number of access states in the initial cycle (full access) on the burst ROM interface is determined by the basic bus interface settings in ASTCR, ABWCR, WTCRA, WTCRB, and CSACRH. When area 0 or area 1 is designated as burst ROM interface space, the settings in RDNCR and CSACRL are ignored.

From 1 to 8 states can be selected for the burst cycle, according to the settings of bits BSTS02 to BSTS00 and BSTS12 to BSTS10 in BROMCR. Wait states cannot be inserted. Burst access of up to 32 words is performed, according to the settings of bits BSTS01, BSTS00, BSTS11, and BSTS10 in BROMCR.

The basic access timing for burst ROM space is shown in figures 6.63 and 6.64.

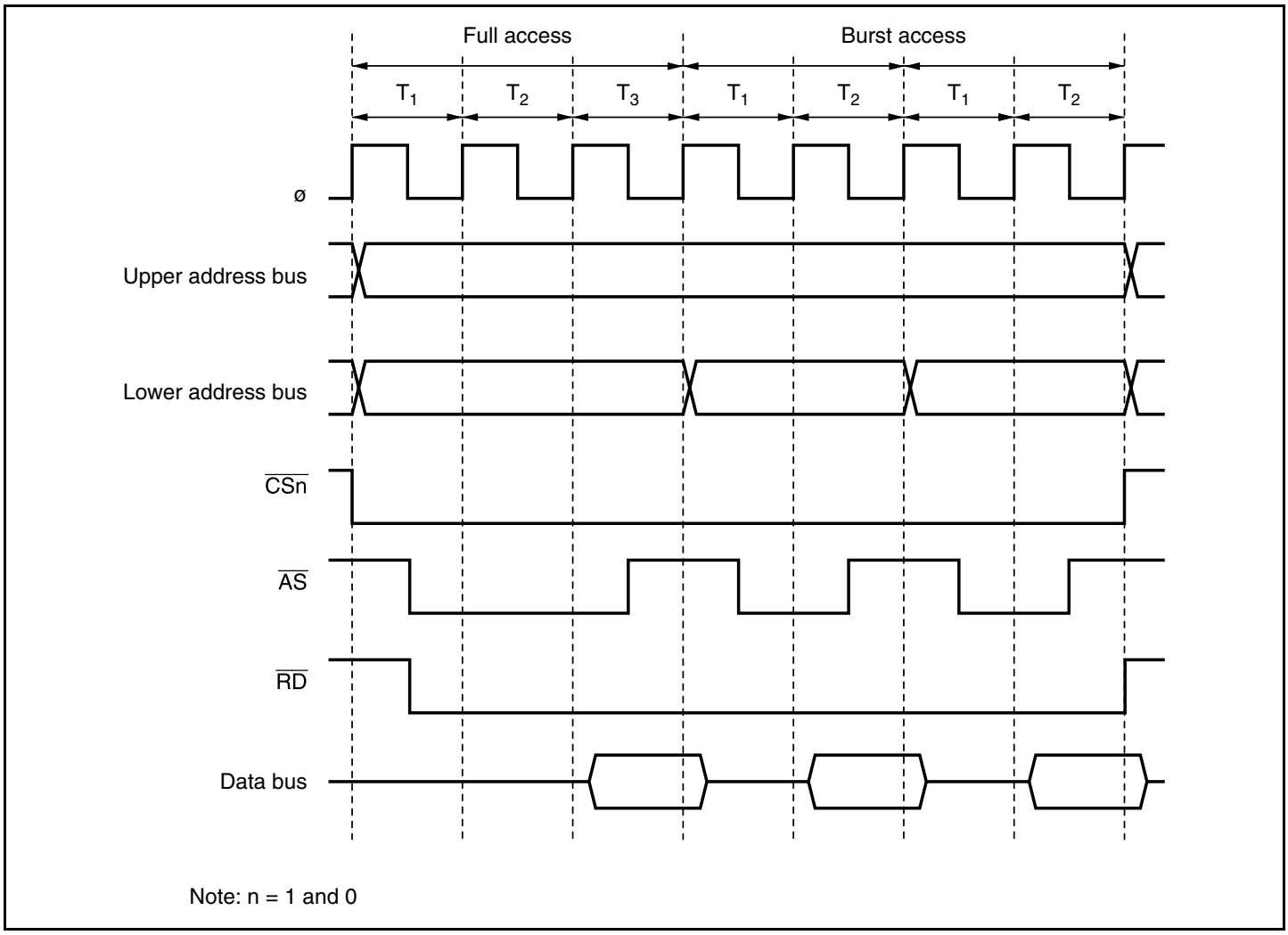


Figure 6.63 Example of Burst ROM Access Timing (ASTn = 1, 2-State Burst Cycle)

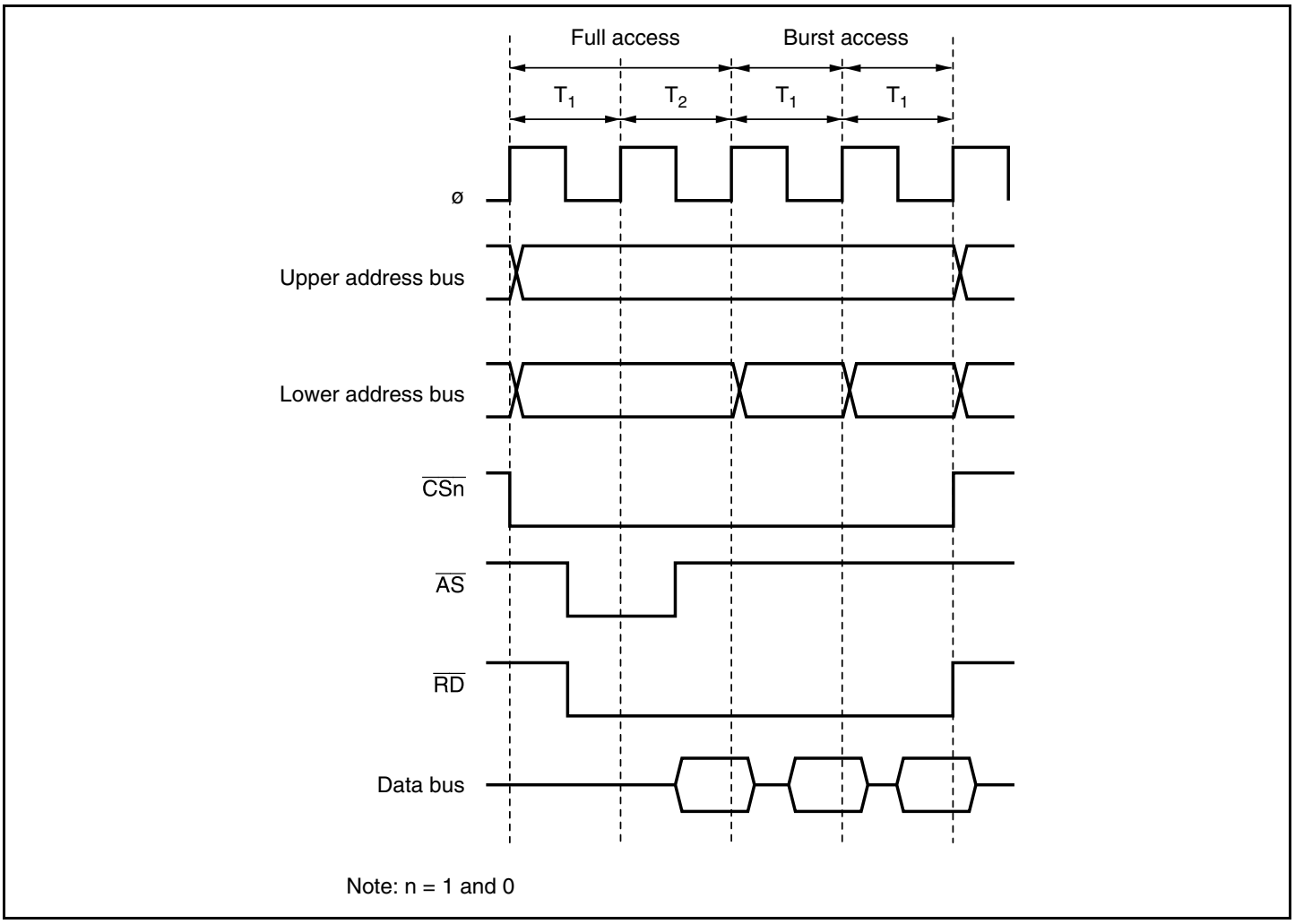


Figure 6.64 Example of Burst ROM Access Timing (ASTn = 0, 1-State Burst Cycle)

6.8.2 Wait Control

As with the basic bus interface, either program wait insertion or pin wait insertion using the WAIT pin can be used in the initial cycle (full access) on the burst ROM interface. See section 6.5.4, Wait Control. Wait states cannot be inserted in a burst cycle.

6.8.3 Write Access

When a write access to burst ROM interface space is executed, burst access is interrupted at that point and the write access is executed in line with the basic bus interface settings. Write accesses are not performed in burst mode even though burst ROM space is designated.

6.9 Idle Cycle

6.9.1 Operation

When this LSI accesses external space, it can insert an idle cycle (T_i) between bus cycles in the following three cases: (1) when read accesses in different areas occur consecutively, (2) when a write cycle occurs immediately after a read cycle, and (3) when a read cycle occurs immediately after a write cycle (in the H8S/2678R Series, it cannot insert an idle cycle in the condition (3)). Insertion of a 1-state or 2-state idle cycle can be selected with the IDLC bit in BCR. By inserting an idle cycle it is possible, for example, to avoid data collisions between ROM, etc., with a long output floating time, and high-speed memory, I/O interfaces, and so on.

Consecutive Reads in Different Areas: If consecutive reads in different areas occur while the ICIS1 bit is set to 1 in BCR, an idle cycle is inserted at the start of the second read cycle.

Figure 6.65 shows an example of the operation in this case. In this example, bus cycle A is a read cycle for ROM with a long output floating time, and bus cycle B is a read cycle for SRAM, each being located in a different area. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is inserted, and a data collision is prevented.

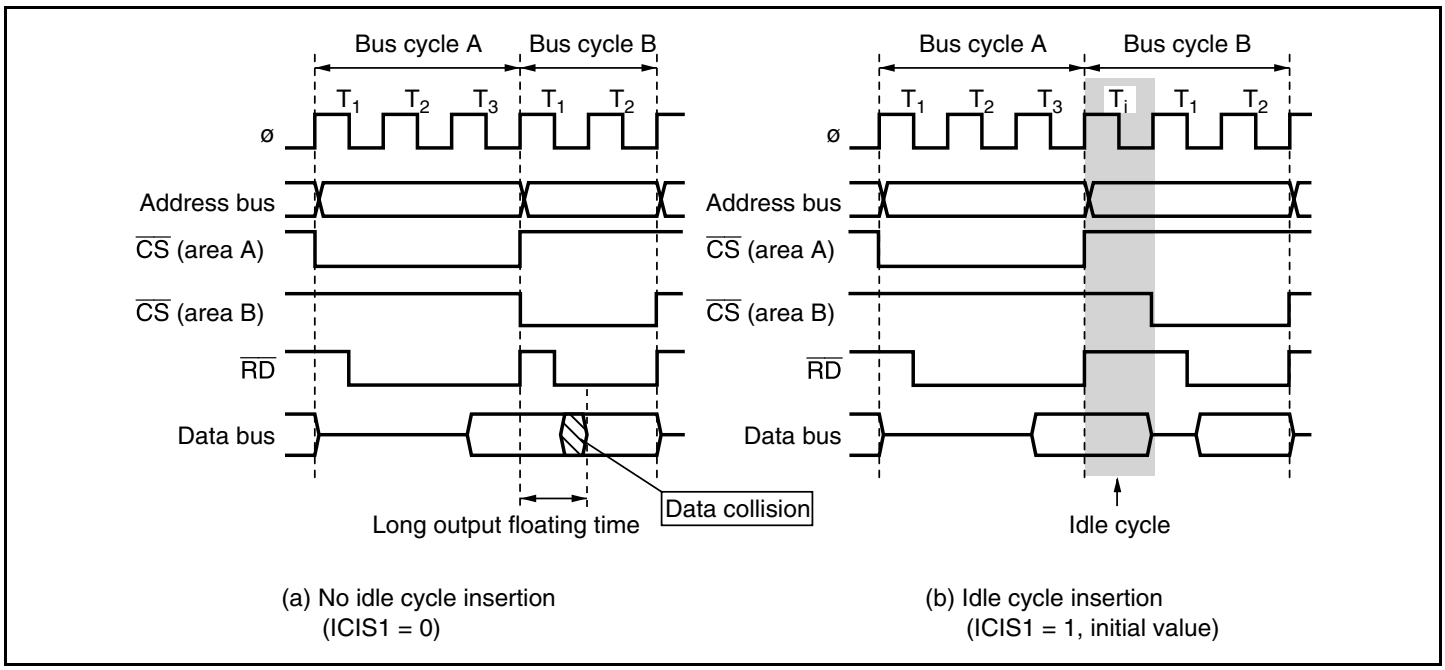


Figure 6.65 Example of Idle Cycle Operation (Consecutive Reads in Different Areas)

Write after Read: If an external write occurs after an external read while the ICIS0 bit is set to 1 in BCR, an idle cycle is inserted at the start of the write cycle.

Figure 6.66 shows an example of the operation in this case. In this example, bus cycle A is a read cycle for ROM with a long output floating time, and bus cycle B is a CPU write cycle. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and the CPU write data. In (b), an idle cycle is inserted, and a data collision is prevented.

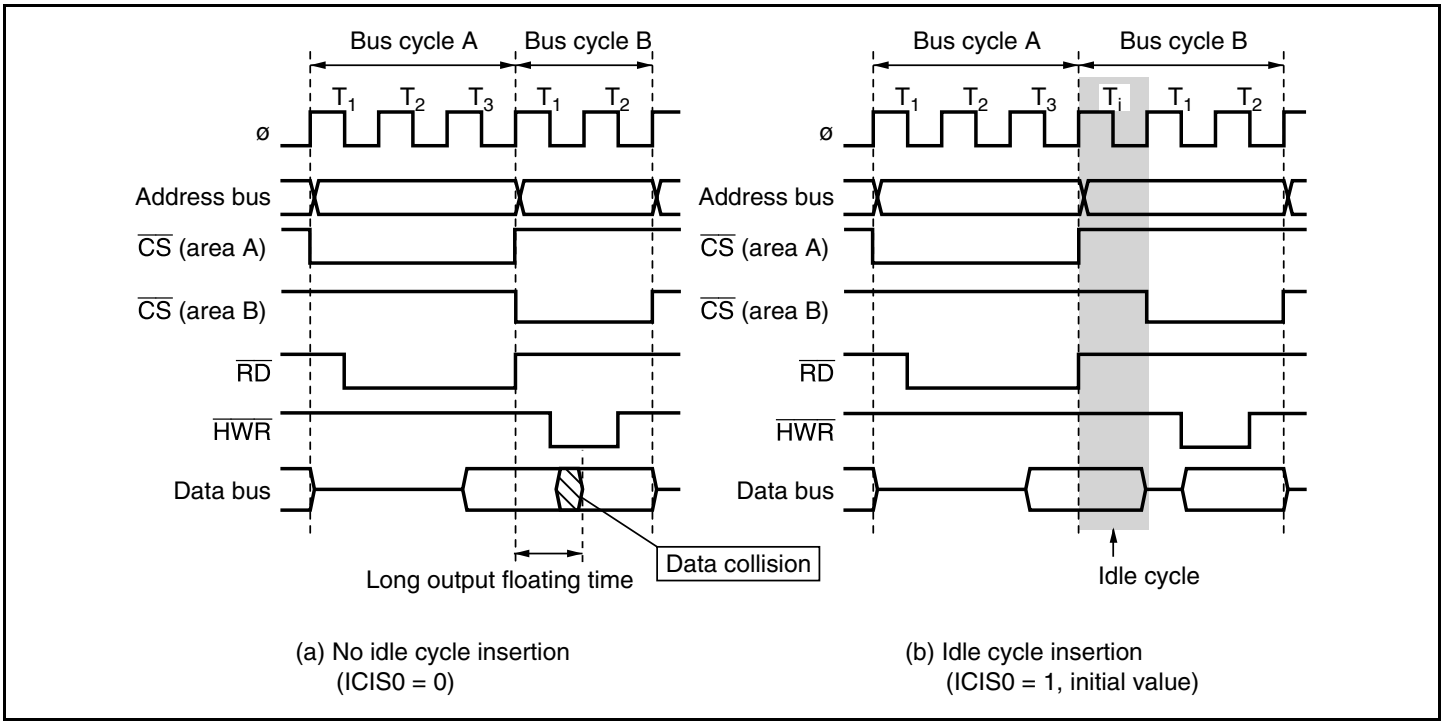


Figure 6.66 Example of Idle Cycle Operation (Write after Read)

Read after Write: If an external read occurs after an external write while the ICIS2 bit is set to 1 in BCR, an idle cycle is inserted at the start of the read cycle.

Figure 6.67 shows an example of the operation in this case. In this example, bus cycle A is a CPU write cycle and bus cycle B is a read cycle from an external device. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the CPU write data and read data from an external device. In (b), an idle cycle is inserted, and a data collision is prevented.

Note: In the H8S/2678 Series, an idle cycle cannot be inserted in the condition (3).

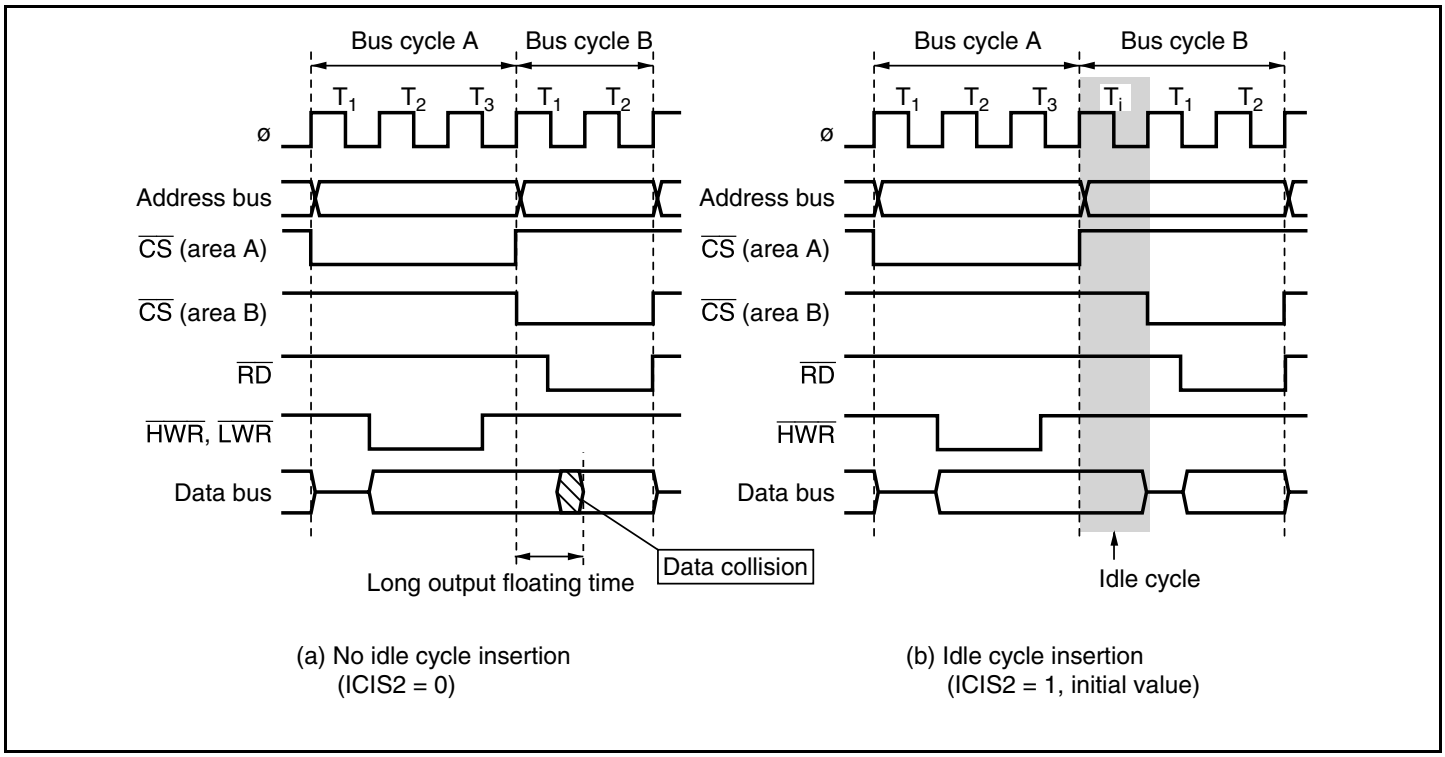


Figure 6.67 Example of Idle Cycle Operation (Read after Write)

Relationship between Chip Select (CS) Signal and Read (RD) Signal: Depending on the system's load conditions, the RD signal may lag behind the CS signal. An example is shown in figure 6.68. In this case, with the setting for no idle cycle insertion (a), there may be a period of overlap between the bus cycle A RD signal and the bus cycle B CS signal. Setting idle cycle insertion, as in (b), however, will prevent any overlap between the RD and CS signals. In the initial state after reset release, idle cycle insertion (b) is set.

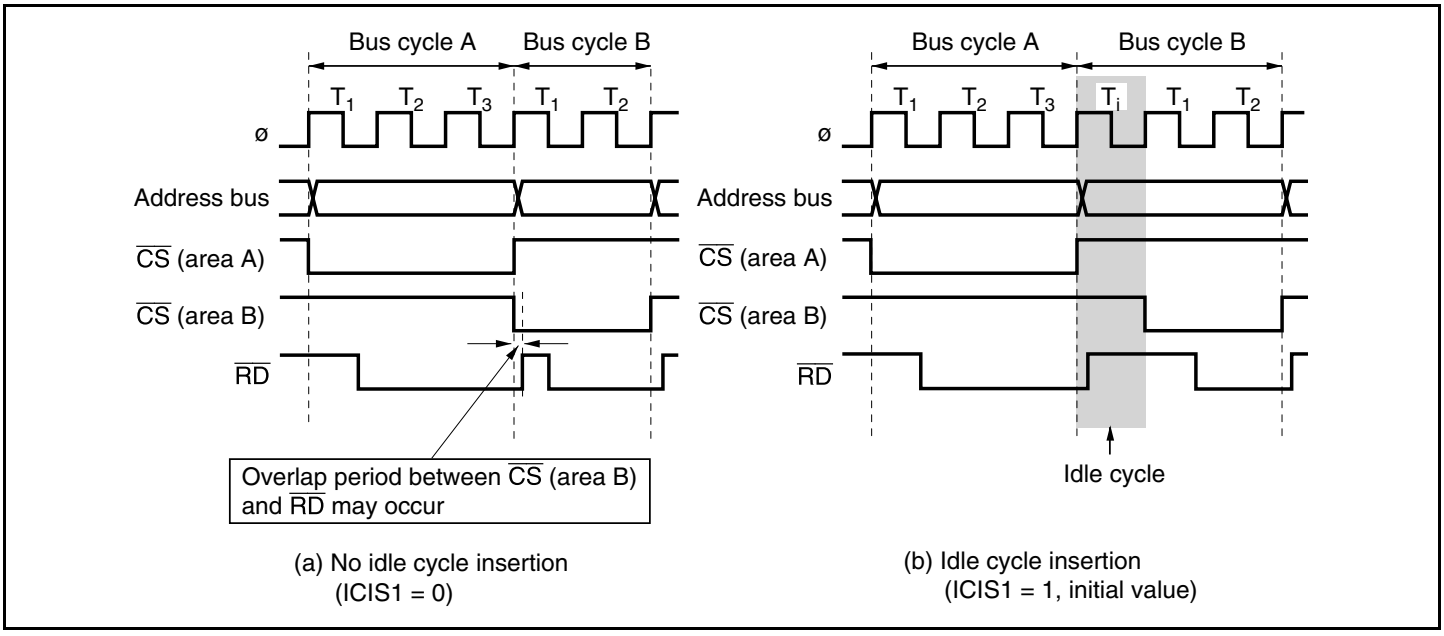


Figure 6.68 Relationship between Chip Select (CS) and Read (RD)

Idle Cycle in Case of DRAM Space Access after Normal Space Access: In a DRAM space access following a normal space access, the settings of bits ICIS2 (not available in the H8S/2678 Series), ICIS1, ICIS0, and IDLC in BCR are valid. However, in the case of consecutive reads in different areas, for example, if the second read is a full access to DRAM space, only a T_p cycle is inserted, and a T_i cycle is not. The timing in this case is shown in figure 6.69.

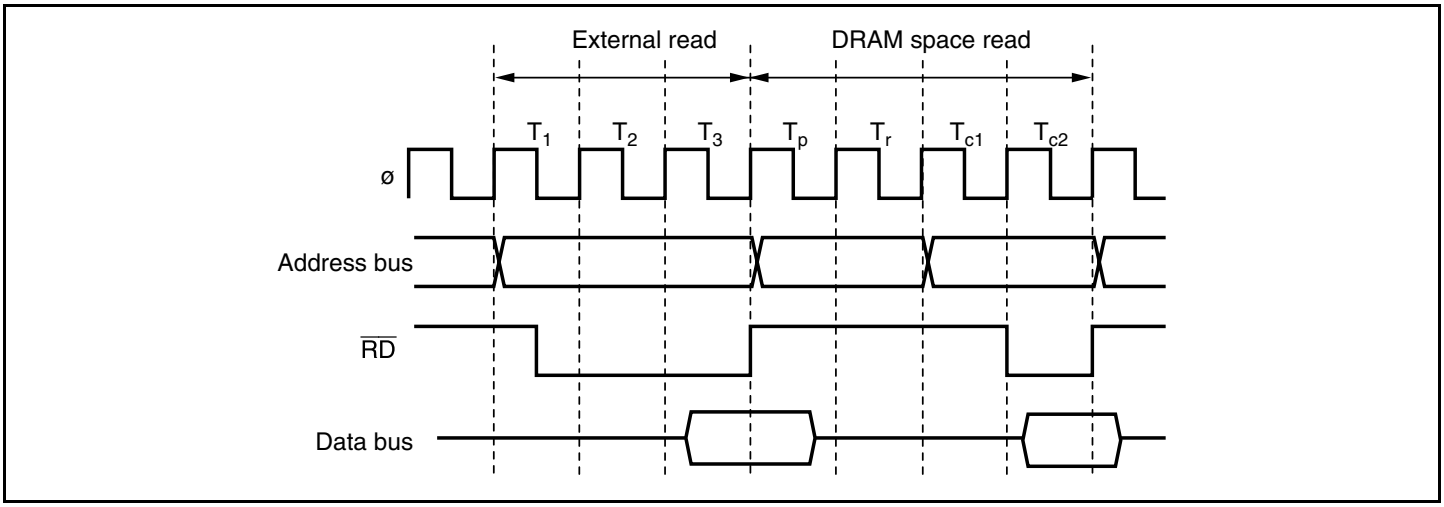


Figure 6.69 Example of DRAM Full Access after External Read (CAST = 0)

In burst access in RAS down mode, the settings of bits ICIS2*, ICIS1, ICIS0, and IDLC are valid and an idle cycle is inserted. The timing in this case is illustrated in figures 6.70 and 6.71.

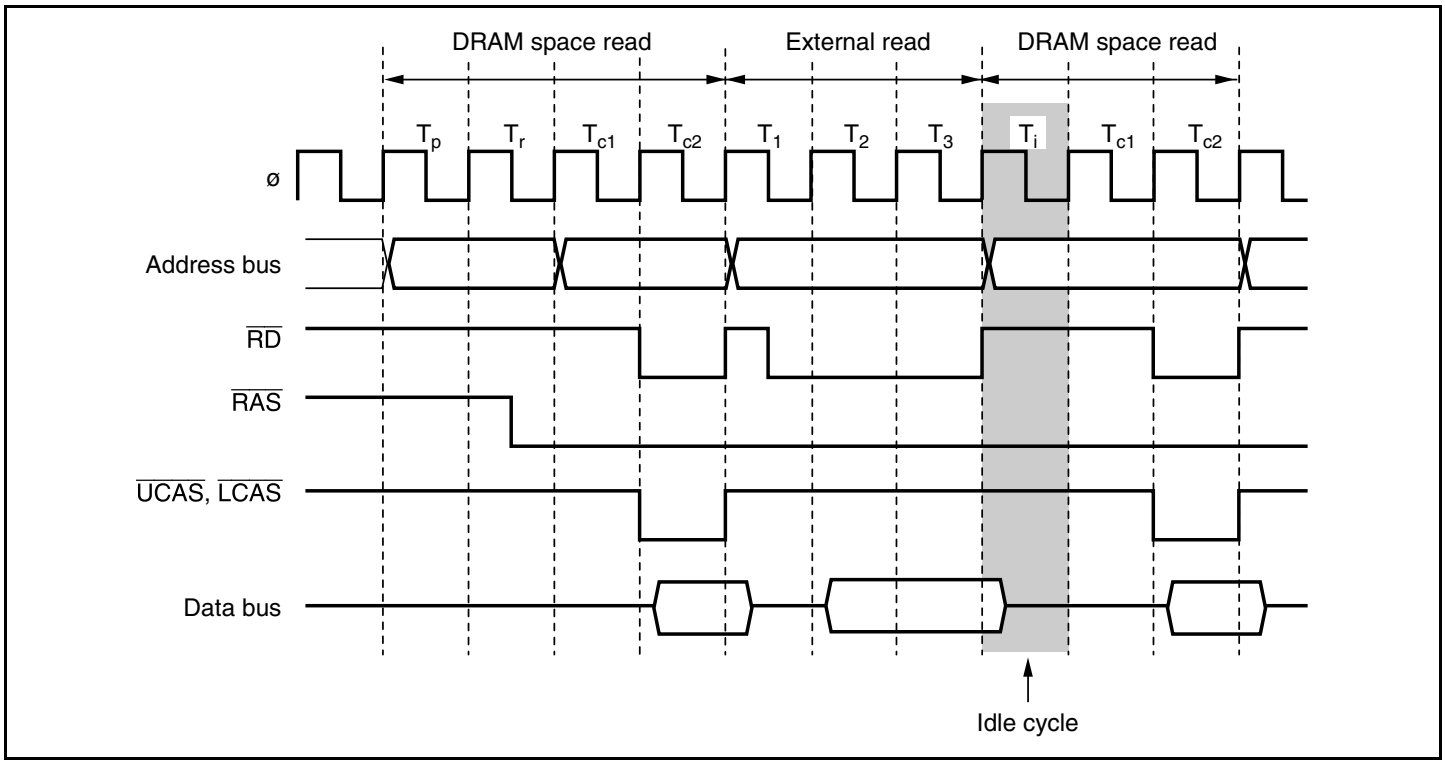


Figure 6.70 Example of Idle Cycle Operation in RAS Down Mode (Consecutive Reads in Different Areas) (IDLC = 0, RAST = 0, CAST = 0)

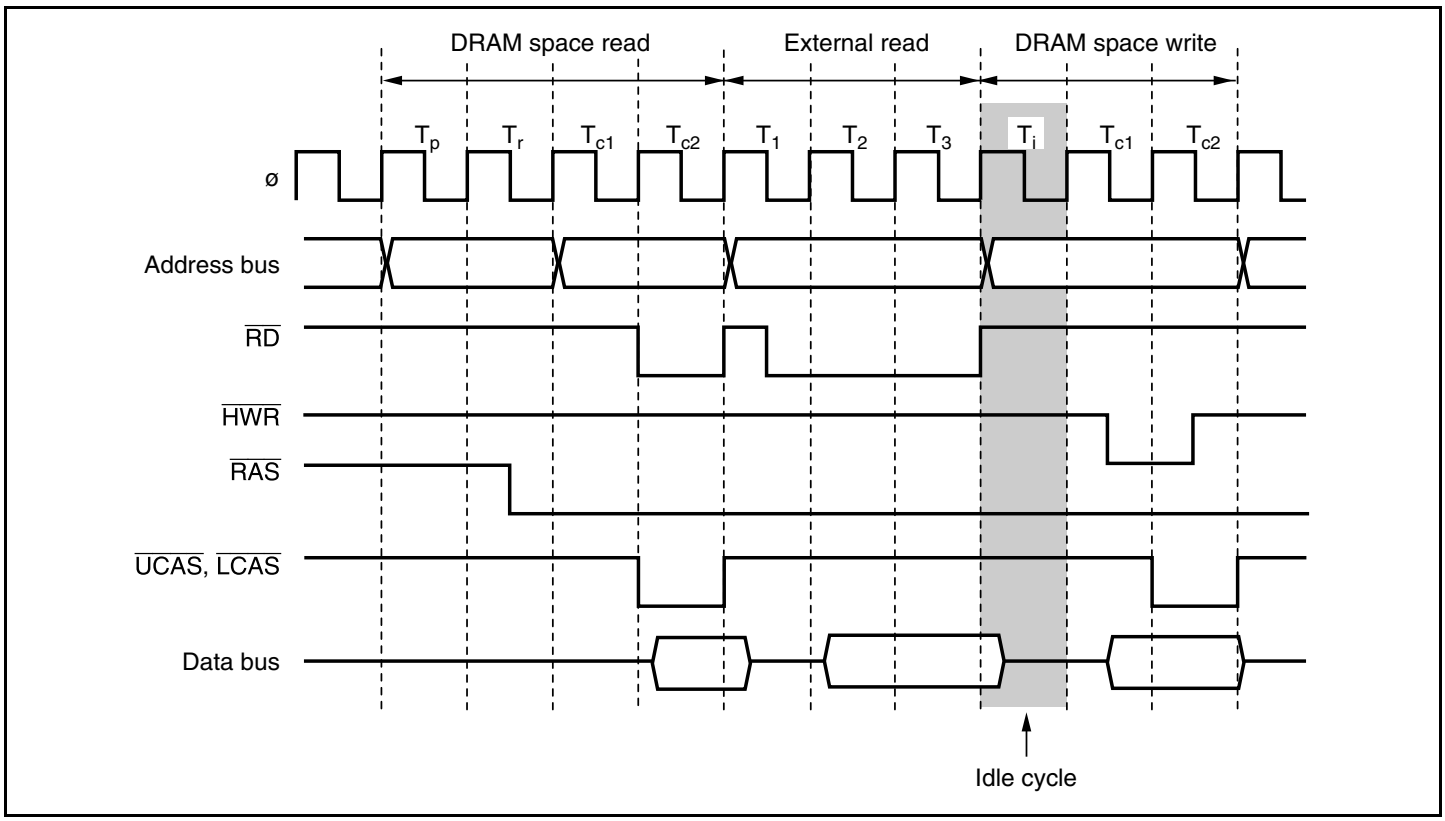


Figure 6.71 Example of Idle Cycle Operation in RAS Down Mode (Write after Read) (IDLC = 0, RAST = 0, CAST = 0)

Idle Cycle in Case of Continuous Synchronous DRAM Space Access after Normal Space Access: In a continuous synchronous DRAM space access following a normal space access, the settings of bits ICIS2, ICIS1, ICIS0, and IDLC in BCR are valid. However, in the case of consecutive reads in different areas, for example, if the second read is a full access to continuous synchronous DRAM space, only Tp cycle is inserted, and Ti cycle is not. The timing in this case is shown in figure 6.72.

Note: In the H8S/2678 Series, the synchronous DRAM interface is not supported.

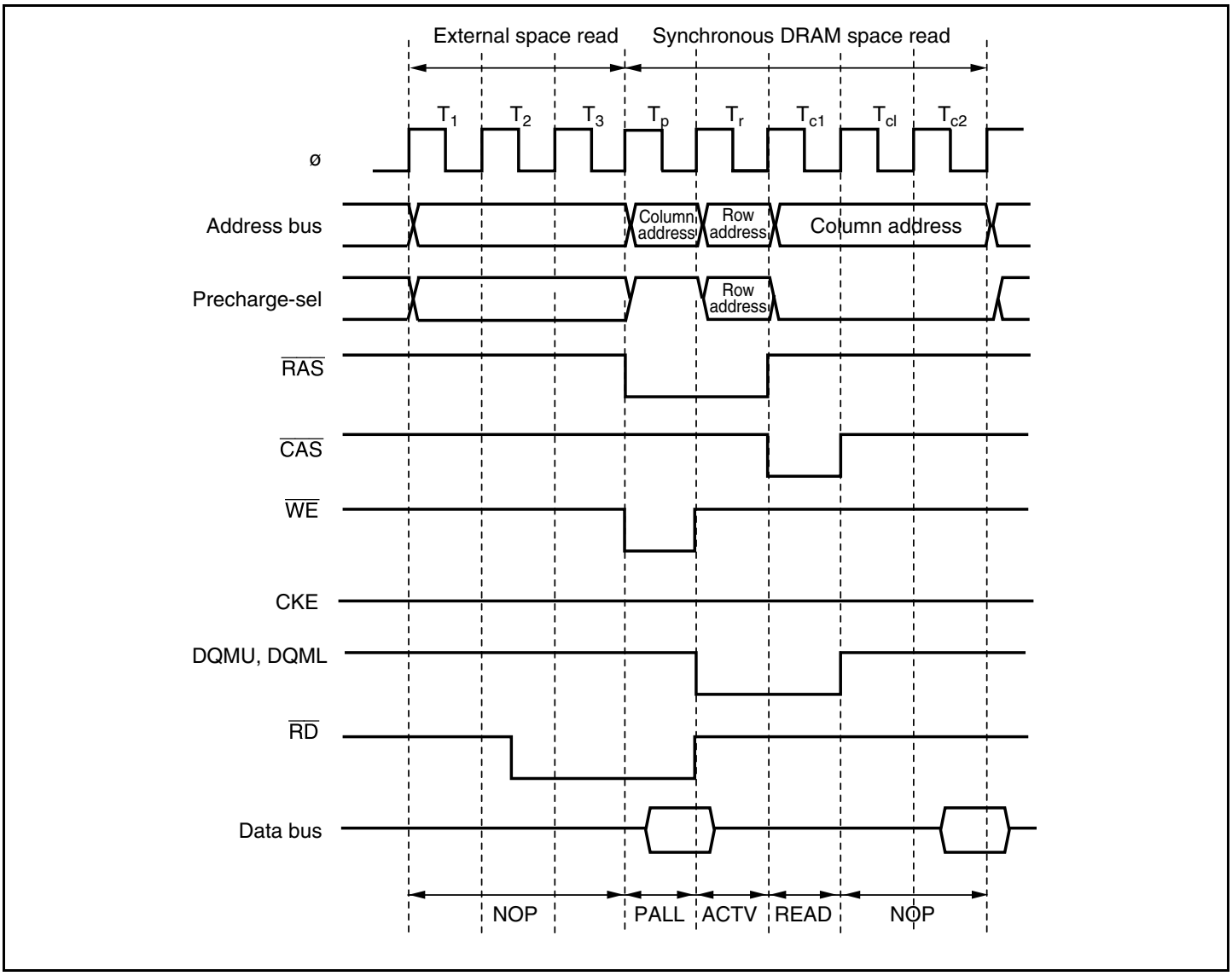


Figure 6.72 Example of Synchronous DRAM Full Access after External Read (CAS Latency 2)

In burst access in RAS down mode, the settings of bits ICIS2, ICIS1, ICIS0, and IDLC are valid and an idle cycle is inserted. However, in read access, note that the timings of DQMU and DQML differ according to the settings of the IDLC bit. The timing in this case is illustrated in figures 6.73 and 6.74. In write access, DQMU and DQML are not in accordance with the settings of the IDLC bit. The timing in this case is illustrated in figure 6.75.

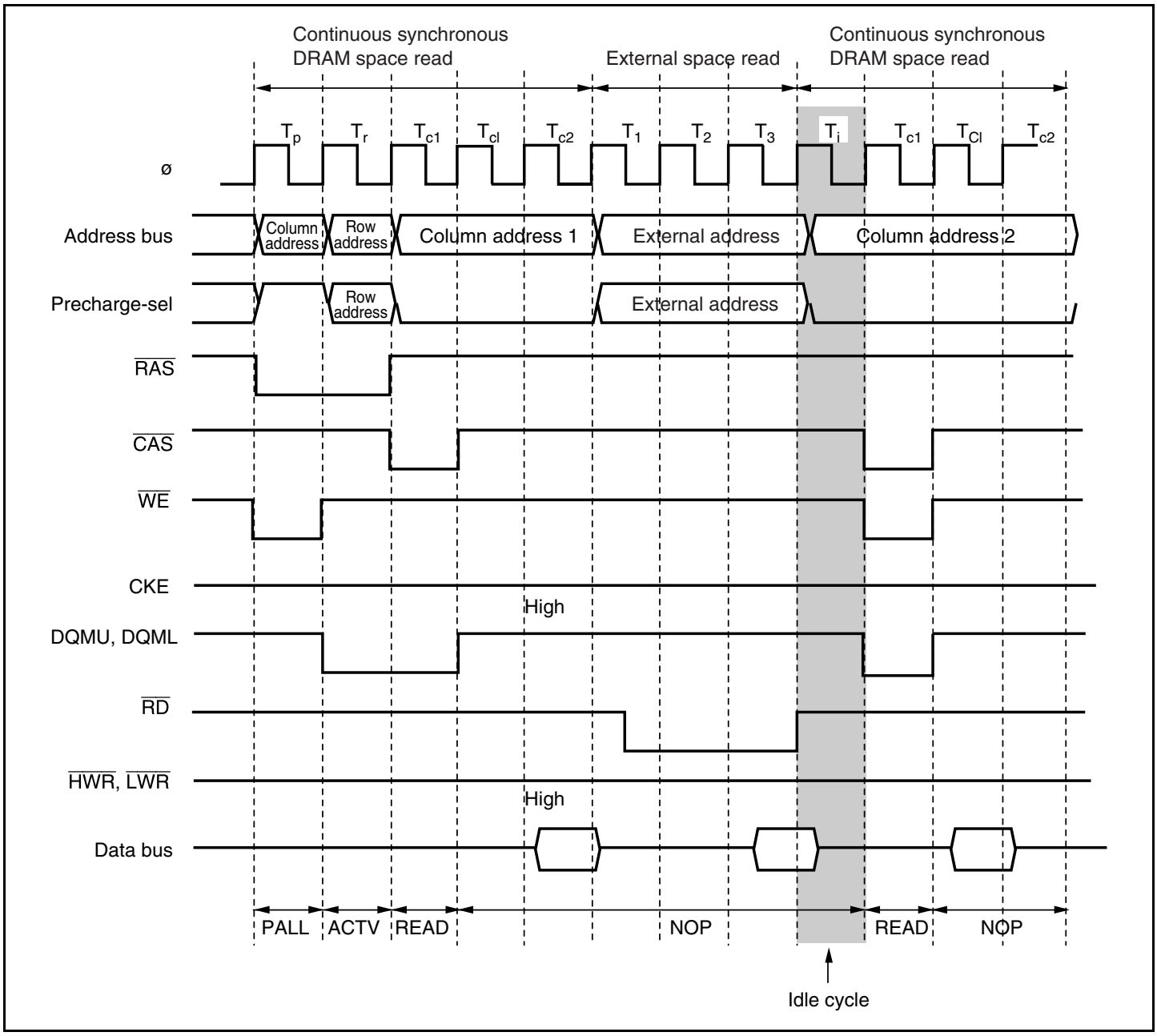
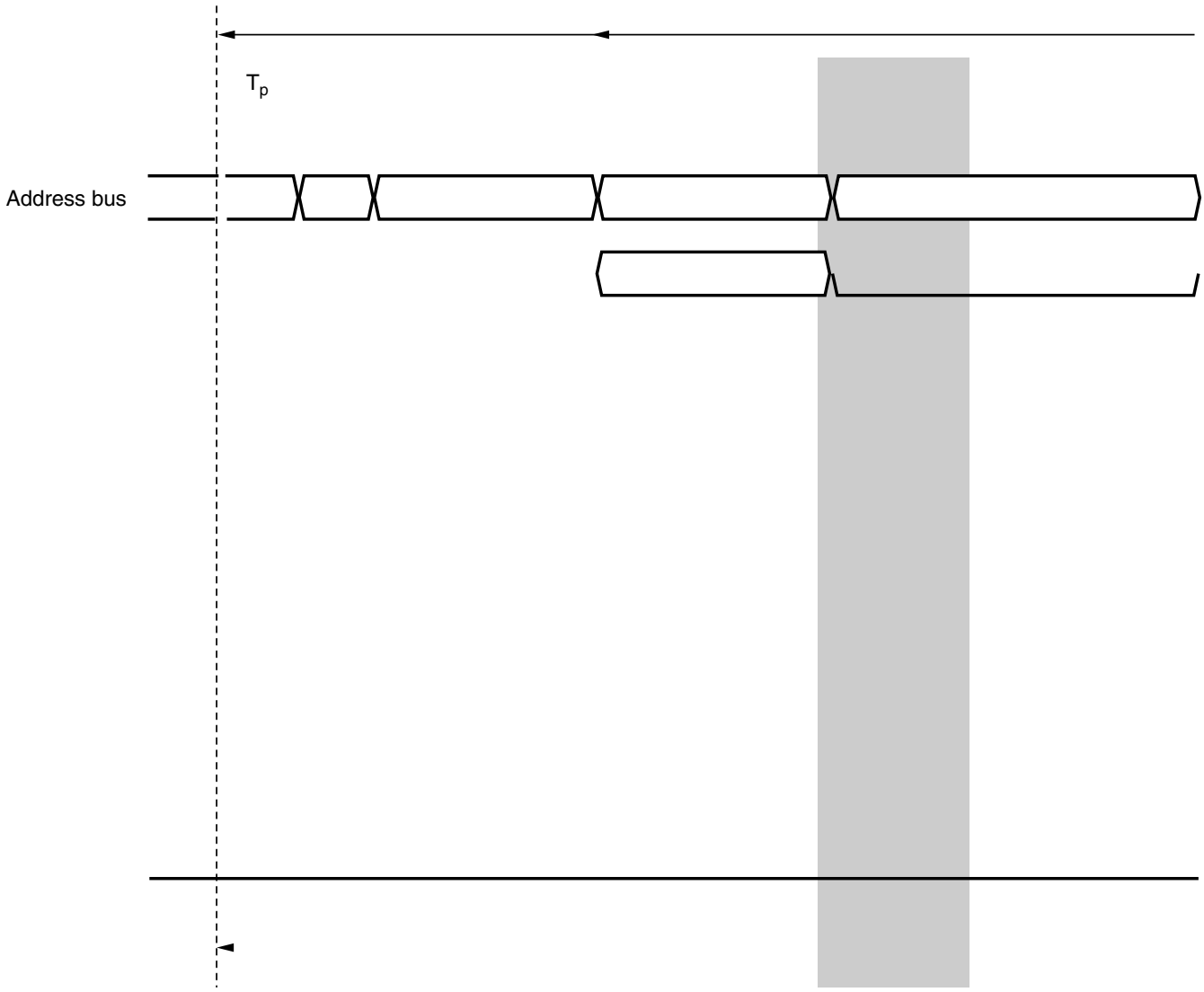


Figure 6.73 Example of Idle Cycle Operation in RAS Down Mode (Read in Different Area) (IDLC = 0, CAS Latency 2)


T_p
Address bus

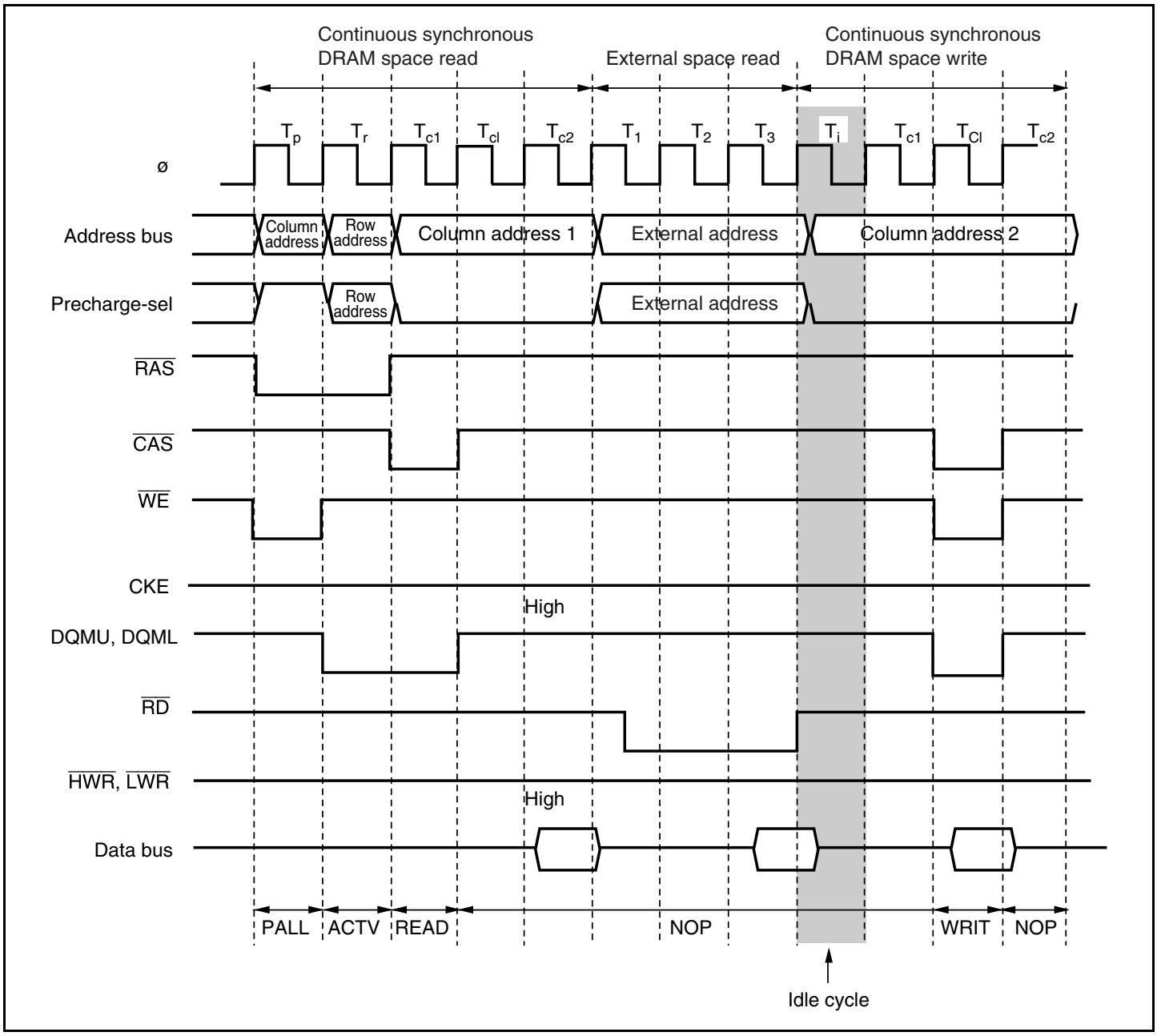


Figure 6.75 Example of Idle Cycle Operation in RAS Down Mode (Write after Read) (IDLC = 0, CAS Latency 2)

Idle Cycle in Case of Normal Space Access after DRAM Space Access:

- Normal space access after DRAM space read access

While the DRMI bit is cleared to 0 in DRACCR, idle cycle insertion after DRAM space access is disabled. Idle cycle insertion after DRAM space access can be enabled by setting the DRMI bit to 1. The conditions and number of states of the idle cycle to be inserted are in accordance with the settings of bits ICIS1, ICIS0, and IDLC in BCR are valid. Figures 6.76 and 6.77 show examples of idle cycle operation when the DRMI bit is set to 1.

When the DRMI bit is cleared to 0, an idle cycle is not inserted after DRAM space access even if bits ICIS1 and ICIS0 are set to 1.

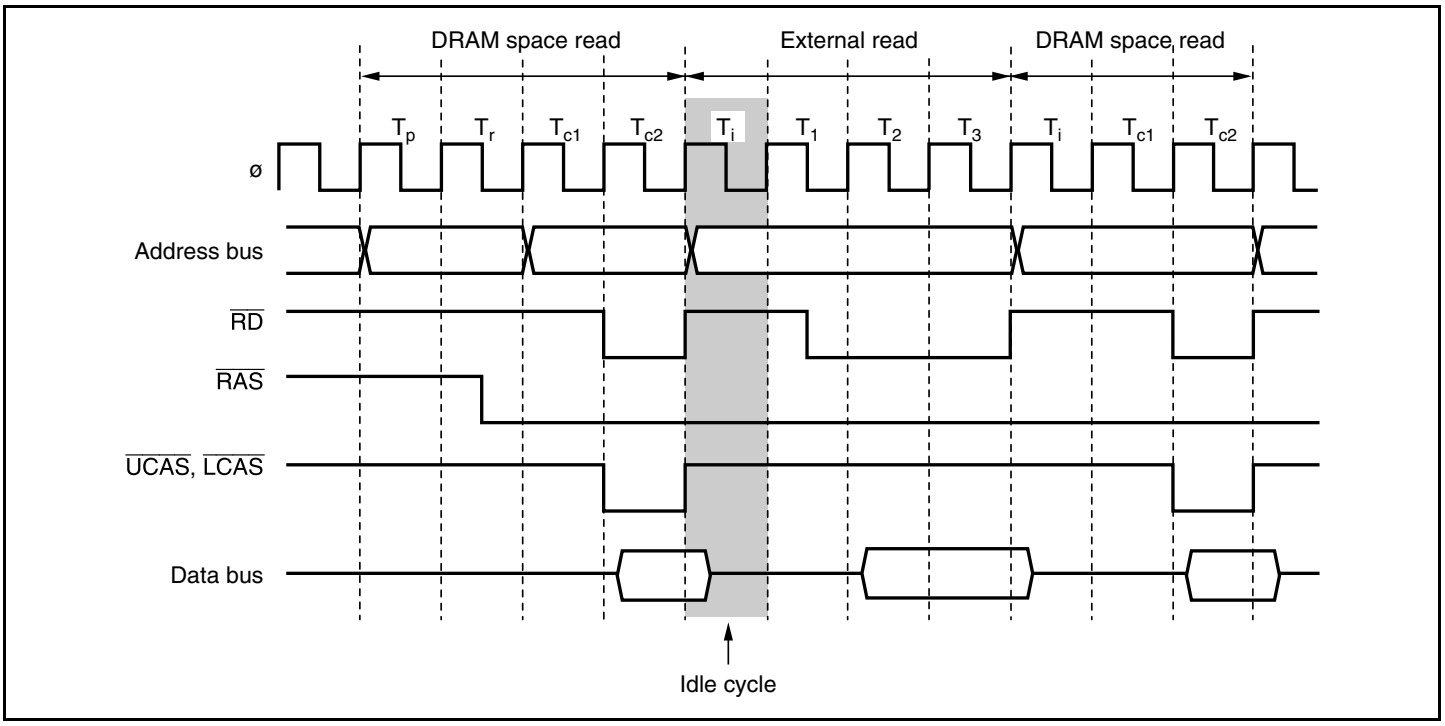


Figure 6.76 Example of Idle Cycle Operation after DRAM Access (Consecutive Reads in Different Areas) (IDLC = 0, RAST = 0, CAST = 0)

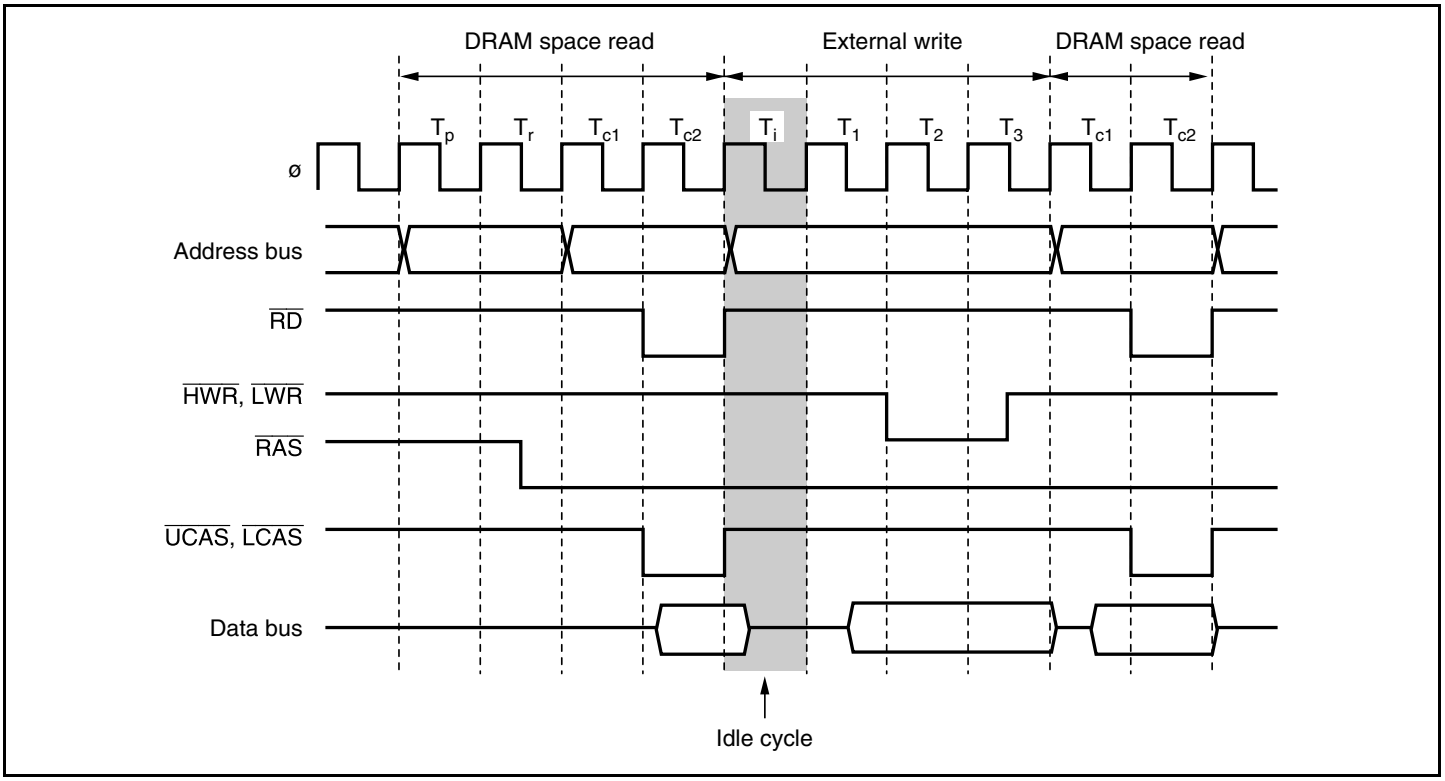


Figure 6.77 Example of Idle Cycle Operation after DRAM Access (Write after Read) (IDLC = 0, RAST = 0, CAST = 0)

- Normal space access after DRAM space write access

While the ICIS2 bit is set to 1 in BCR (there is no ICRS2 bit in the H8S/2678 Series, therefore this setting cannot be made) and a normal space read access occurs after DRAM space write access, idle cycle is inserted in the first read cycle. The number of states of the idle cycle to be inserted is in accordance with the setting of the IDLC bit. It does not depend on the DRMI bit in DRACCR. Figure 6.78 shows an example of idle cycle operation when the ICIS2 bit is set to 1.

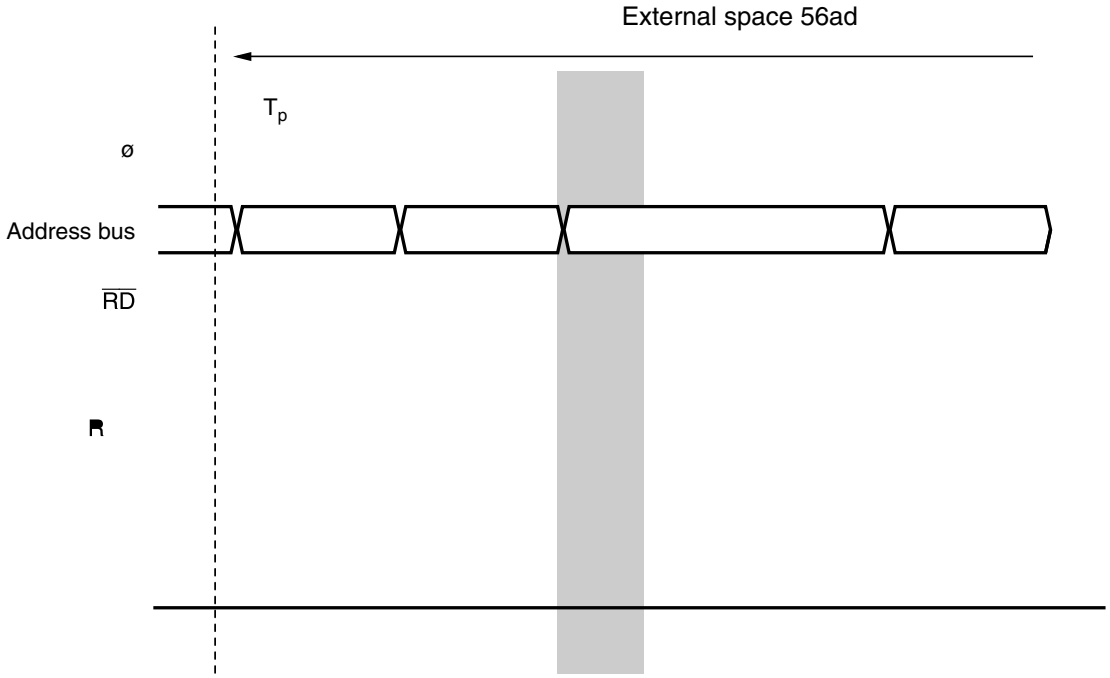


Figure Idle cycle

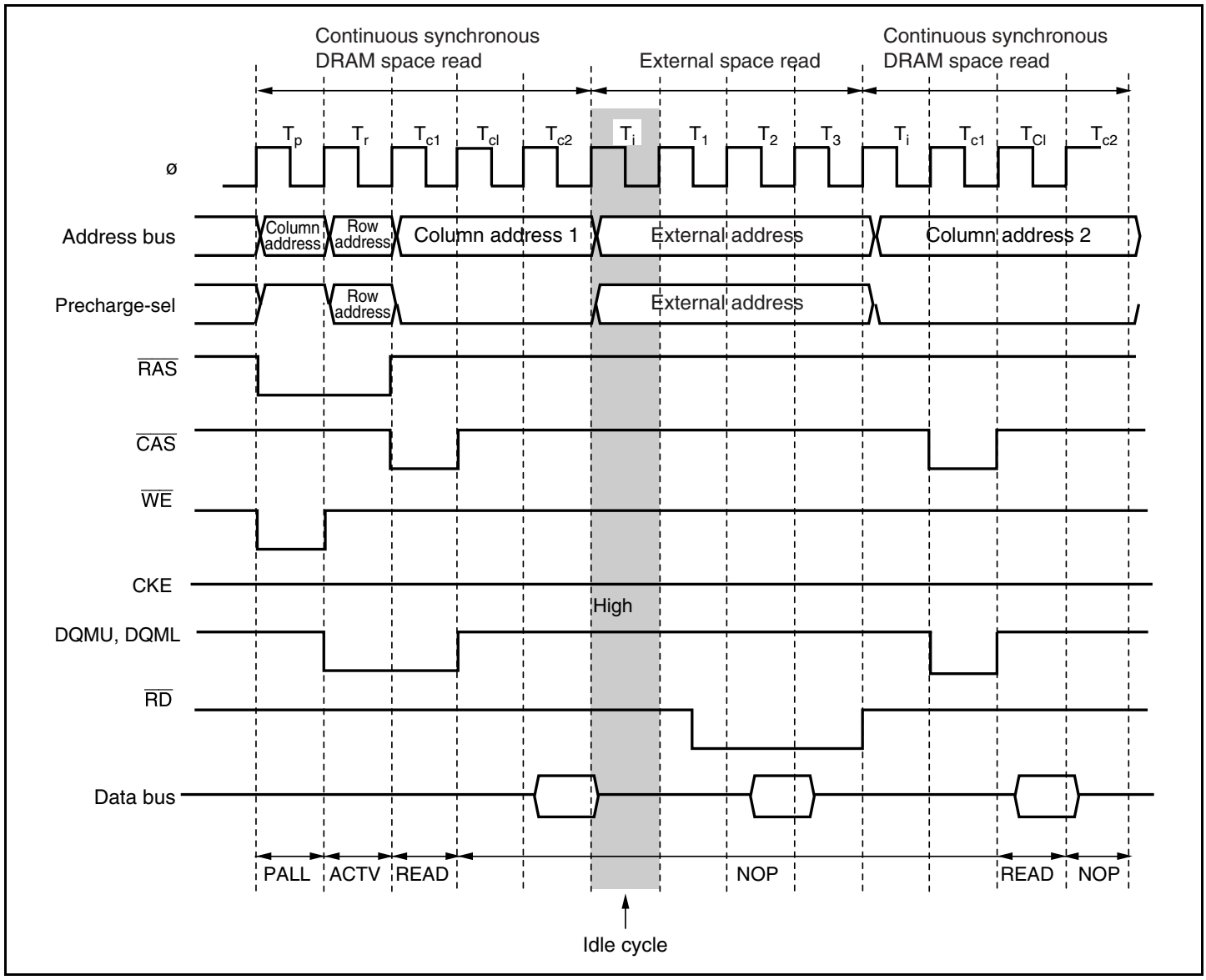


Figure 6.79 Example of Idle Cycle Operation after Continuous Synchronous DRAM Space Read Access (Read between Different Area) (IDLC = 0, CAS Latency 2)

- Normal space access after a continuous synchronous DRAM space write access

If a normal space read cycle occurs after a continuous synchronous DRAM space write access while the ICIS2 bit is set to 1 in BCR, idle cycle is inserted at the start of the read cycle. The number of states of the idle cycle to be inserted is in accordance with the setting of bit IDLC. It is not in accordance with the DRMI bit in DRACCR.

Figure 6.80 shows an example of idle cycle operation when the ICIS2 bit is set to 1.

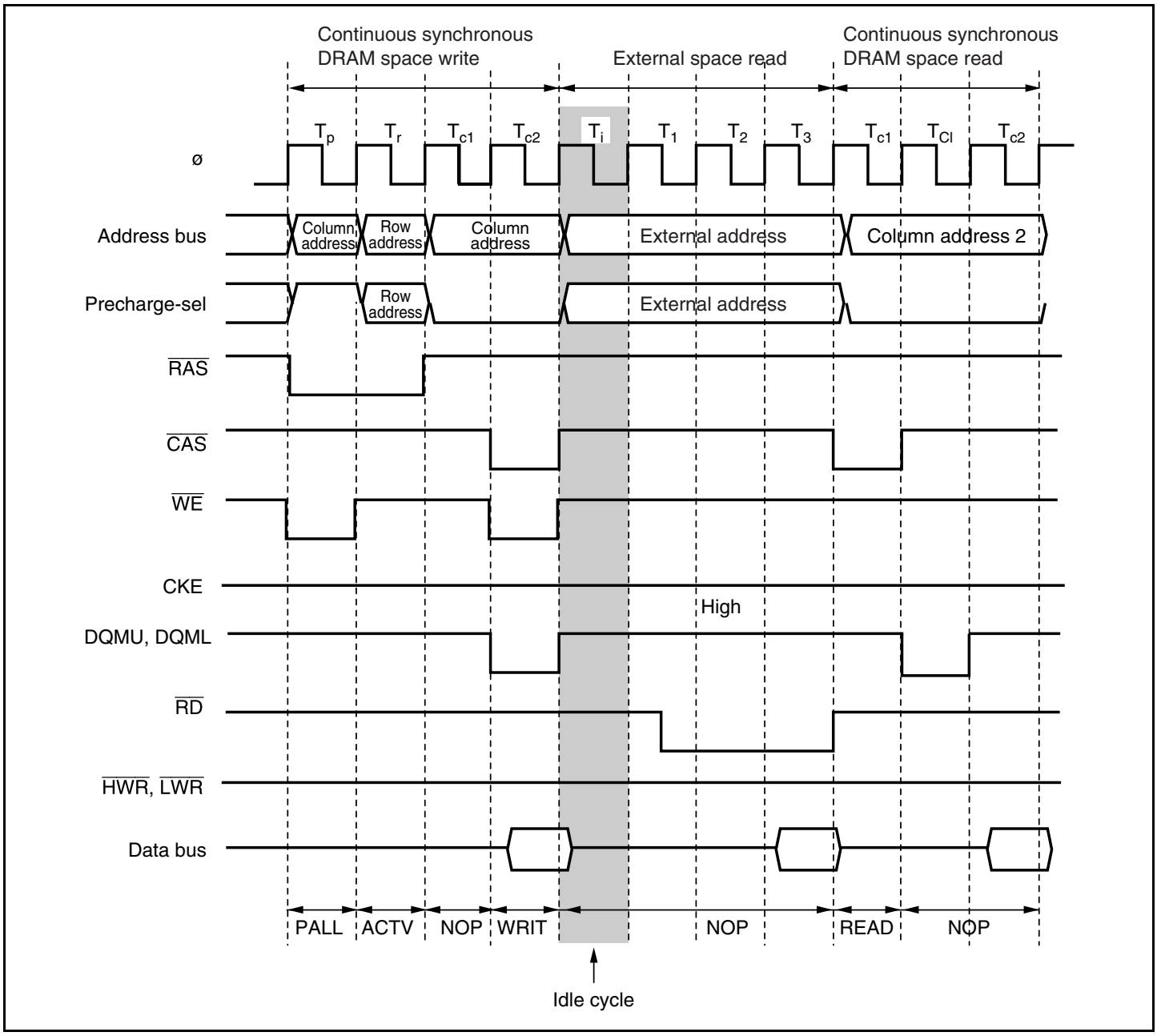


Figure 6.80 Example of Idle Cycle Operation after Continuous Synchronous DRAM Space Write Access (IDLC = 0, ICIS1 = 0, SDWCD = 1, CAS Latency 2)

Table 6.11 shows whether there is an idle cycle insertion or not in the case of mixed accesses to normal space and DRAM space/continuous synchronous DRAM space.

Table 6.11 Idle Cycles in Mixed Accesses to Normal Space and DRAM Continuous Synchronous DRAM Space

Previous Access	Next Access	ICIS2*	ICIS1	ICIS0	DRMI	IDLC	Idle cycle
Normal space read	Normal space read (different area)	—	0	—	—	—	Disabled
		—	1	—	—	0	1 state inserted
						1	2 states inserted
	DRAM/continuous synchronous DRAM* space read	—	0	—	—	—	Disabled
		—	1	—	—	0	1 state inserted
						1	2 states inserted
	Normal space write	—	—	0	—	—	Disabled
		—	—	1	—	0	1 state inserted
						1	2 states inserted
	DRAM/continuous synchronous DRAM* space write	—	—	0	—	—	Disabled
		—	—	1	—	0	1 state inserted
						1	2 states inserted
DRAM/continuous synchronous DRAM space read	Normal space read	—	0	—	—	—	Disabled
		—	1	—	0	—	Disabled
					1	0	1 state inserted
						1	2 states inserted
	DRAM/continuous synchronous DRAM* space read	—	0	—	—	—	Disabled
		—	1	—	0	—	Disabled
					1	0	1 state inserted
						1	2 states inserted
	Normal space write	—	—	0	—	—	Disabled
		—	—	1	0	—	Disabled
					1	0	1 state inserted
						1	2 states inserted
	DRAM/continuous synchronous DRAM* space write	—	—	0	—	—	Disabled
		—	—	1	0	—	Disabled
					1	0	1 state inserted
						1	2 states inserted
Normal space write	Normal space read	0	—	—	—	—	Disabled
		1	—	—	—	0	1 state inserted
						1	2 states inserted
	DRAM/continuous synchronous DRAM* space read	0	—	—	—	—	Disabled
		1	—	—	—	0	1 state inserted
						1	2 states inserted

Previous Access	Next Access	ICIS2*	ICIS1	ICIS0	DRMI	IDLC	Idle cycle
DRAM/continuous synchronous DRAM space write	Normal space read	0	—	—	—	—	Disabled
		1	—	—	—	0	1 state inserted
						1	2 states inserted
	DRAM/continuous synchronous DRAM space read	0	—	—	—	—	Disabled
		1	—	—	—	0	1 state inserted
						1	2 states inserted

Note: In the H8S/2678 Series, the synchronous DRAM interface is not supported.

Setting the DRMI bit in DRACCR to 1 enables an idle cycle to be inserted in the case of consecutive read and write operations in DRAM/continuous synchronous DRAM space burst access. Figures 6.81 and 6.82 show an example of the timing for idle cycle insertion in the case of consecutive read and write accesses to DRAM/continuous synchronous DRAM space.

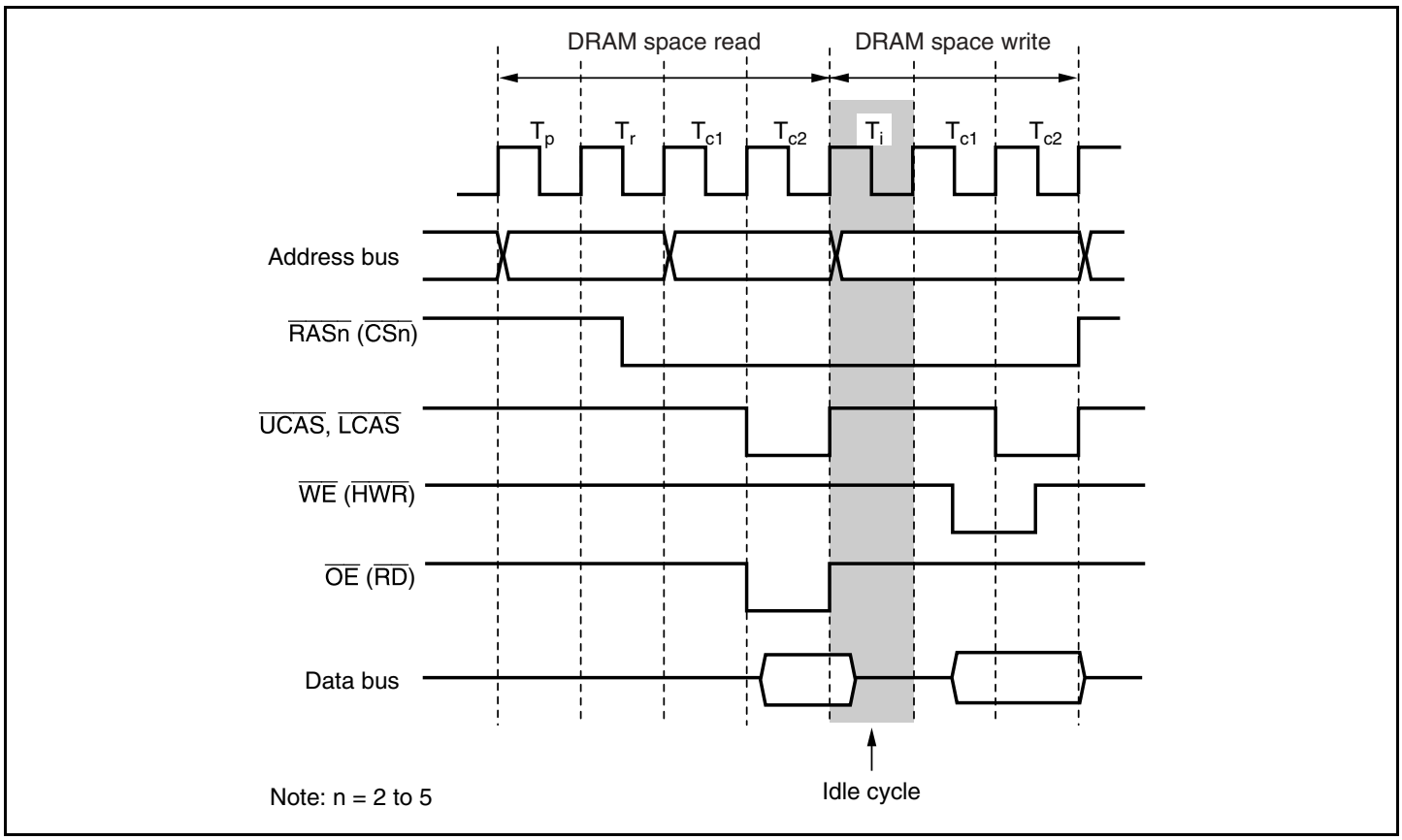


Figure 6.81 Example of Timing for Idle Cycle Insertion in Case of Consecutive Read and Write Accesses to DRAM Space in RAS Down Mode

6.9.2 Pin States in Idle Cycle

Table 6.12 shows the pin states in an idle cycle.

Table 6.12 Pin States in Idle Cycle

Pins	Pin State
A23 to A0	Contents of following bus cycle
D15 to D0	High impedance
CSn (n = 7 to 0)	High*1, *2
UCAS, LCAS	High*2
AS	High
RD	High
(OE)	High
HWR, LWR	High
DACKn (n = 1, 0)	High
EDACKn (n = 3 to 0)	High

Notes: 1. Remains low in DRAM space RAS down mode.
2. Remains low in a DRAM space refresh cycle.

6.10 Write Data Buffer Function

This LSI has a write data buffer function for the external data bus. Using the write data buffer function enables external writes and DMA single address mode transfers to be executed in parallel with internal accesses. The write data buffer function is made available by setting the WDBE bit to 1 in BCR.

Figure 6.83 shows an example of the timing when the write data buffer function is used. When this function is used, if an external write or DMA single address mode transfer continues for two states or longer, and there is an internal access next, an external write only is executed in the first state, but from the next state onward an internal access (on-chip memory or internal I/O register read/write) is executed in parallel with the external write rather than waiting until it ends.

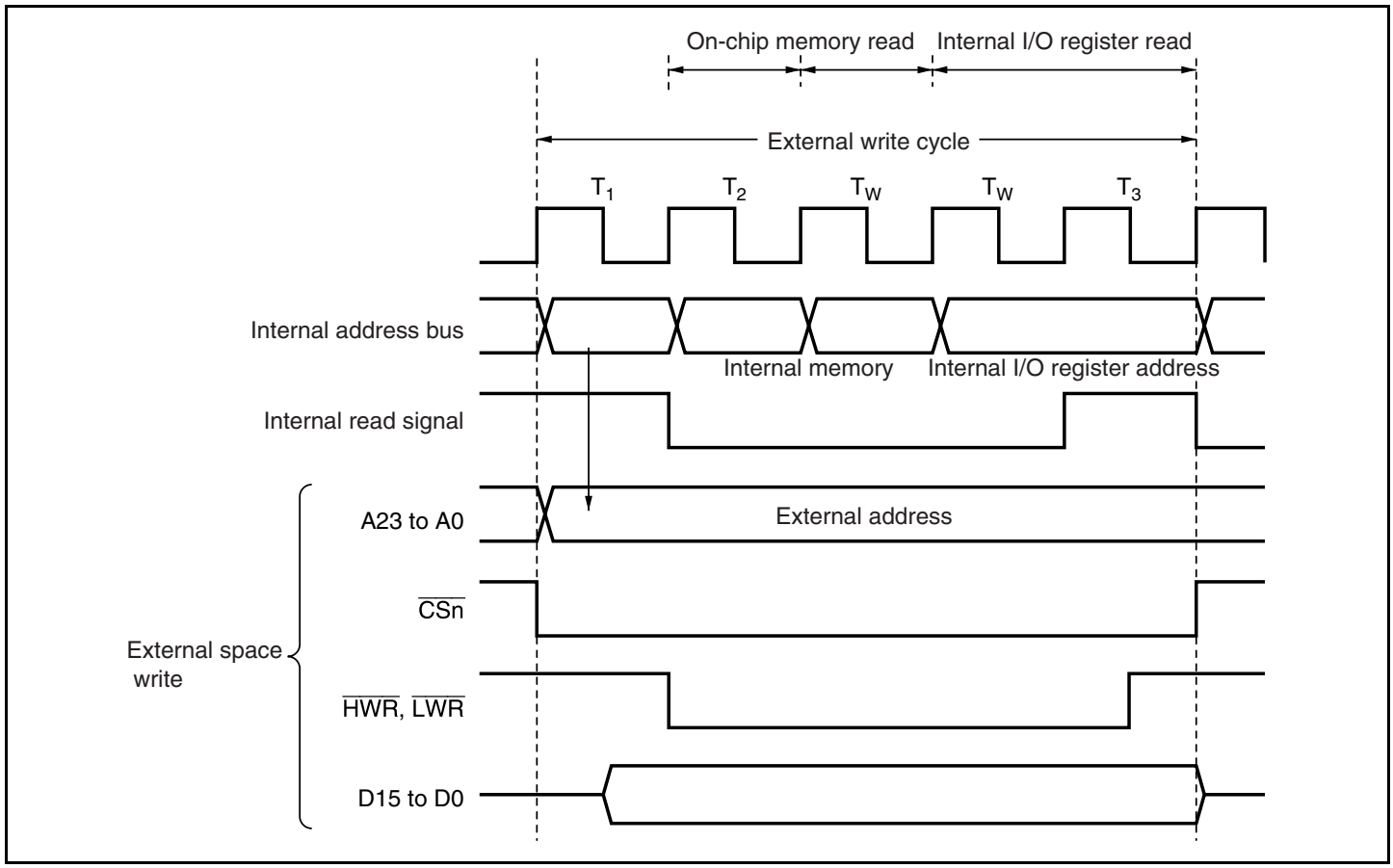


Figure 6.83 Example of Timing when Write Data Buffer Function is Used

6.11 Bus Release

This LSI can release the external bus in response to a bus request from an external device. In the external bus released state, internal bus masters except the EXDMAC continue to operate as long as there is no external access. If any of the following requests are issued in the external bus released state, the BREQO signal can be driven low to output a bus request externally.

- When an internal bus master wants to perform an external access
- When a refresh request is generated
- When a SLEEP instruction is executed to place the chip in software standby mode or all-module-clocks-stopped mode

6.11.1 Operation

In externally expanded mode, the bus can be released to an external device by setting the BRLE bit to 1 in BCR. Driving the BREQ pin low issues an external bus request to this LSI. When the BREQ pin is sampled, at the prescribed timing the BACK pin is driven low, and the address bus, data bus, and bus control signals are placed in the high-impedance state, establishing the external bus released state.

In the external bus released state, internal bus masters except the EXDMAC can perform accesses using the internal bus. When an internal bus master wants to make an external access, it temporarily defers initiation of the bus cycle, and waits for the bus request from the external bus master to be canceled. If a refresh request is generated in the external bus released state, or if a SLEEP instruction is executed to place the chip in software standby mode or all-module-clocks-stopped mode, refresh control and software standby or all-module-clocks-stopped control is deferred until the bus request from the external bus master is canceled.

If the BREQOE bit is set to 1 in BCR, the BREQO pin can be driven low when any of the following requests are issued, to request cancellation of the bus request externally.

- When an internal bus master wants to perform an external access
- When a refresh request is generated
- When a SLEEP instruction is executed to place the chip in software standby mode or all-module-clocks-stopped mode

When the BREQ pin is driven high, the BACK pin is driven high at the prescribed timing and the external bus released state is terminated.

If an external bus release request and external access occur simultaneously, the order of priority is as follows:

(High) External bus release > External access by internal bus master (Low)

If a refresh request and external bus release request occur simultaneously, the order of priority is as follows:

(High) Refresh > External bus release (Low)

6.11.2 Pin States in External Bus Released State

Table 6.13 shows pin states in the external bus released state.

Table 6.13 Pin States in Bus Released State

Pins	Pin State
A23 to A0	High impedance
D15 to D0	High impedance
CSn (n = 7 to 0)	High impedance
UCAS, LCAS	High impedance
AS	High impedance
RD	High impedance
(OE)	High impedance
HWR, LWR	High impedance
DACKn (n = 1, 0)	High
EDACKn (n = 3 to 0)	High

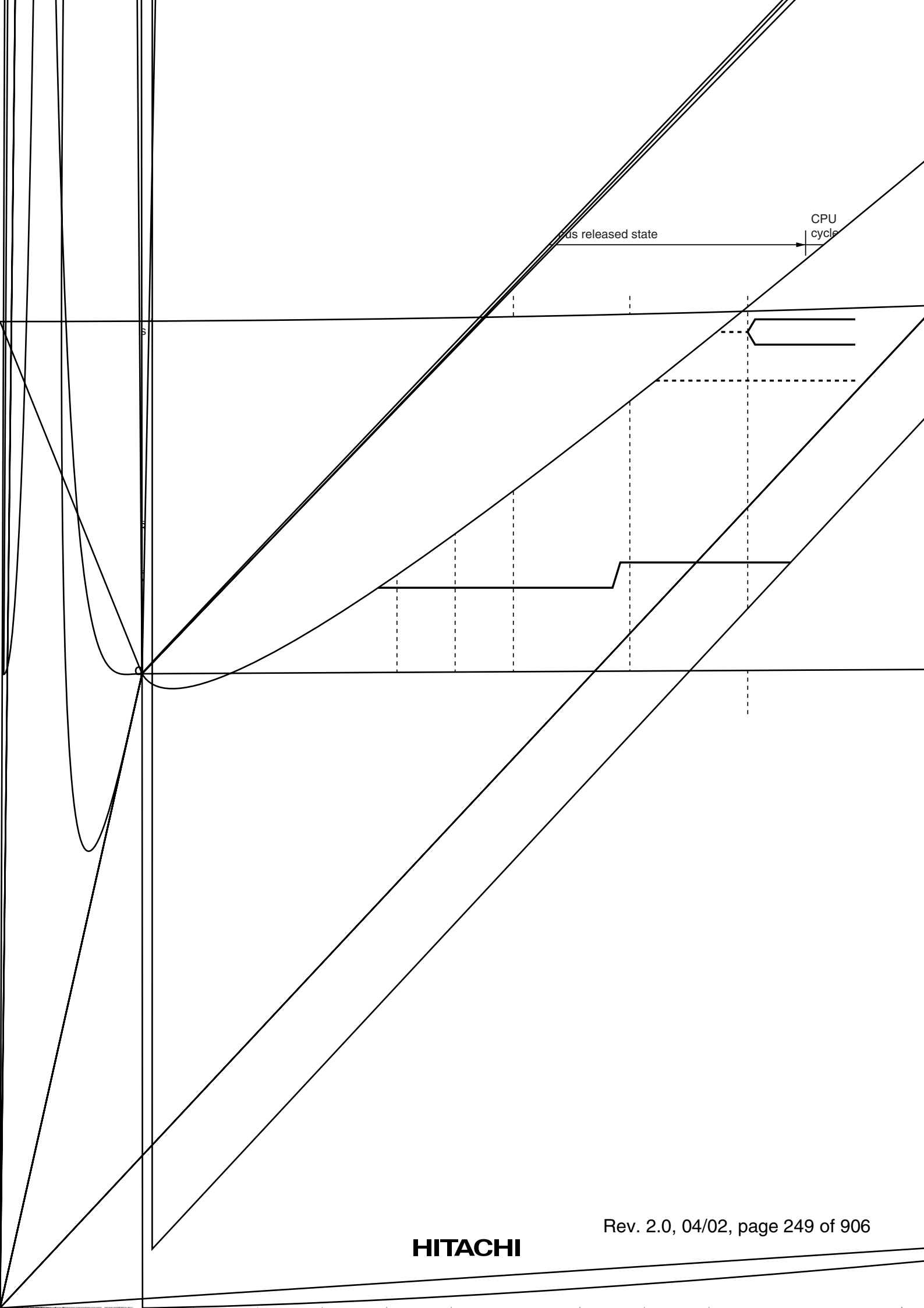
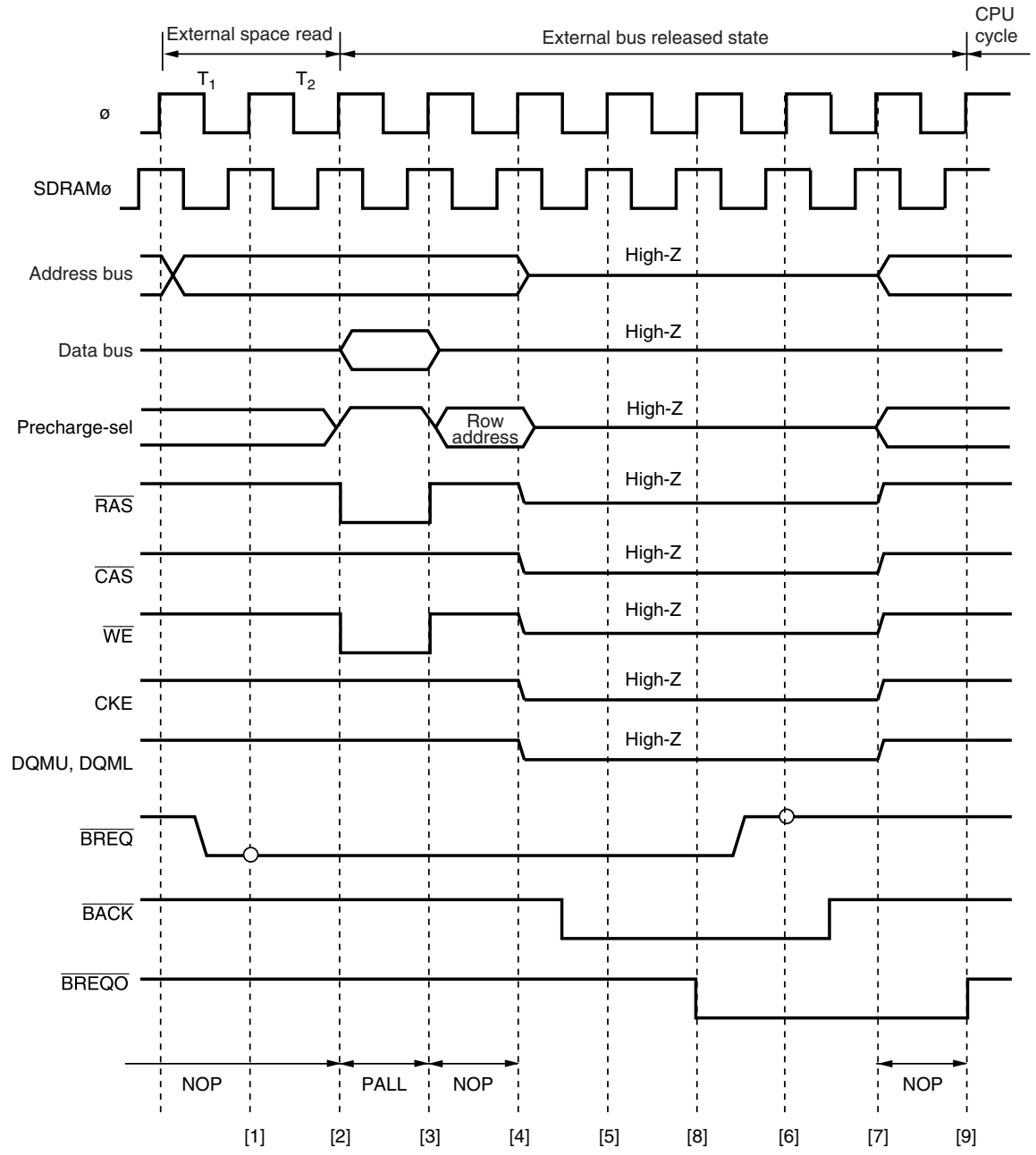

CPU
cycle

Figure 6.85 shows the timing for transition to the bus released state with the synchronous DRAM interface.



[1] Low level of BREQ signal is sampled at rise of f.
[2] PLL command is issued.
[3] Bus control signal returns to be high at end of external space access cycle.
At least one state from sampling of BREQ signal.
[4] BACK signal is driven low, releasing bus to external bus master..
[5] BREQ signal state is also sampled in external bus released state.
[6] High level of BREQ signal is sampled.
[7] BACK signal is driven high, ending external bus release cycle.
[8] When there is external access or refresh request of internal bus master during external bus release while the BREQOE bit is set to 1, BREQO signal goes low.
[9] BREQO signal goes high 1.5 states after rising edge of BACK signal. If BREQO signal is asserted because of auto-refreshing request, it retains low until auto-refresh cycle starts up.

Figure 6.85 Bus Release State Transition Timing when Synchronous DRAM Interface

6.12 Bus Arbitration

This LSI has a bus arbiter that arbitrates bus mastership operations (bus arbitration).

There are four bus masters—the CPU, DTC, DMAC, and EXDMAC—that perform read/write operations when they have possession of the bus. Each bus master requests the bus by means of a bus request signal. The bus arbiter determines priorities at the prescribed timing, and permits use of the bus by means of a bus request acknowledge signal. The selected bus master then takes possession of the bus and begins its operation.

6.12.1 Operation

The bus arbiter detects the bus masters' bus request signals, and if the bus is requested, sends a bus request acknowledge signal to the bus master. If there are bus requests from more than one bus master, the bus request acknowledge signal is sent to the one with the highest priority. When a bus master receives the bus request acknowledge signal, it takes possession of the bus until that signal is canceled.

The order of priority of the bus mastership is as follows:

(High) EXDMAC > DMAC > DTC > CPU (Low)

An internal bus access by internal bus masters except the EXDMAC and external bus release, a refresh when the CBRM bit is 0, and an external bus access by the EXDMAC can be executed in parallel.

If an external bus release request, a refresh request, and an external access by an internal bus master occur simultaneously, the order of priority is as follows:

(High) Refresh > EXDMAC > External bus release (Low)

(High) External bus release > External access by internal bus master except EXDMAC (Low)

As a refresh when the CBRM bit in REFCR is cleared to 0 and an external access other than to DRAM space by an internal bus master can be executed simultaneously, there is no relative order of priority for these two operations.

6.12.2 Bus Transfer Timing

Even if a bus request is received from a bus master with a higher priority than that of the bus master that has acquired the bus and is currently operating, the bus is not necessarily transferred immediately. There are specific timings at which each bus master can relinquish the bus.

CPU: The CPU is the lowest-priority bus master, and if a bus request is received from the DTC, DMAC, or EXDMAC, the bus arbiter transfers the bus to the bus master that issued the request. The timing for transfer of the bus is as follows:

- The bus is transferred at a break between bus cycles. However, if a bus cycle is executed in discrete operations, as in the case of a longword-size access, the bus is not transferred between the component operations.
- With bit manipulation instructions such as BSET and BCLR, the sequence of operations is: data read (read), relevant bit manipulation operation (modify), write-back (write). The bus is not transferred during this read-modify-write cycle, which is executed as a series of bus cycles.
- If the CPU is in sleep mode, the bus is transferred immediately.

DTC: The DTC sends the bus arbiter a request for the bus when an activation request is generated.

The DTC can release the bus after a vector read, a register information read (3 states), a single data transfer, or a register information write (3 states). It does not release the bus during a register information read (3 states), a single data transfer, or a register information write (3 states).

DMAC: The DMAC sends the bus arbiter a request for the bus when an activation request is generated.

In the case of an external request in short address mode or normal mode, and in cycle steal mode, the DMAC releases the bus after a single transfer.

In block transfer mode, it releases the bus after transfer of one block, and in burst mode, after completion of the transfer. However, in the event of an EXDMAC or external bus release request, which have a higher priority than the DMAC, the bus may be transferred to the bus master even if block or burst transfer is in progress.

EXDMAC: The EXDMAC sends the bus arbiter a request for the bus when an activation request is generated.

As the EXDMAC is used exclusively for transfers to and from the external bus, if the bus is transferred to the EXDMAC, internal accesses by other internal bus masters are still executed in parallel.

In normal transfer mode or cycle steal transfer mode, the EXDMAC releases the bus after a single transfer.

In block transfer mode, it releases the bus after transfer of one block, and in burst transfer mode, after completion of the transfer. By setting the BGUP bit to 1 in EDMDR, it is possible to specify temporary release of the bus in the event of an external access request from an internal bus master. For details see section 8, EXDMA Controller.

External Bus Release: When the BREQ pin goes low and an external bus release request is issued while the BRLE bit is set to 1 in BCR, a bus request is sent to the bus arbiter.

External bus release can be performed on completion of an external bus cycle.

6.13 Bus Controller Operation in Reset

In a reset, this LSI, including the bus controller, enters the reset state immediately, and any executing bus cycle is aborted.

6.14 Usage Notes

6.14.1 External Bus Release Function and All-Module-Clocks-Stopped Mode

In this LSI, if the ACSE bit is set to 1 in MSTPCR, and then a SLEEP instruction is executed with the setting for all peripheral module clocks to be stopped (MSTPCR = H'FFFF) or for operation of the 8-bit timer module alone (MSTPCR = H'FFFE), and a transition is made to the sleep state, the all-module-clocks-stopped mode is entered in which the clock is also stopped for the bus controller and I/O ports. In this state, the external bus release function is halted. To use the external bus release function in sleep mode, the ACSE bit in MSTPCR must be cleared to 0. Conversely, if a SLEEP instruction to place the chip in all-module-clocks-stopped mode is executed in the external bus released state, the transition to all-module-clocks-stopped mode is deferred and performed until after the bus is recovered.

6.14.2 External Bus Release Function and Software Standby

In this LSI, internal bus master operation does not stop even while the bus is released, as long as the program is running in on-chip ROM, etc., and no external access occurs. If a SLEEP instruction to place the chip in software standby mode is executed while the external bus is released, the transition to software standby mode is deferred and performed after the bus is recovered.

Also, since clock oscillation halts in software standby mode, if BREQ goes low in this mode, indicating an external bus release request, the request cannot be answered until the chip has recovered from the software standby state.

6.14.3 External Bus Release Function and CBR Refreshing/Auto Refreshing

CBR refreshing/auto refreshing cannot be executed while the external bus is released. Setting the BREQOE bit to 1 in BCR beforehand enables the BREQO signal to be output when a CBR refresh/auto refresh request is issued.

Note: In the H8S/2678 Series, the auto refresh control is not supported.

6.14.4 BREQO Output Timing

When the BREQOE bit is set to 1 and the BREQO signal is output, BREQO may go low before the BACK signal.

This will occur if the next external access request or CBR refresh request occurs while internal bus arbitration is in progress after the chip samples a low level of BREQ.

6.14.5 Notes on Usage of the Synchronous DRAM

Setting of Synchronous DRAM Interface: The DCTL pin must be fixed to 1 to enable the synchronous DRAM interface. Do not change the DCTL pin during operation.

Connection Clock: Be sure to set the clock to be connected to the synchronous DRAM to SDRAMϕ.

WAIT Pin: In the continuous synchronous DRAM space, insertion of the wait state by the WAIT pin is disabled regardless of the setting of the WAITE bit in BCR.

Bank Control: This LSI cannot carry out the bank control of the synchronous DRAM. All banks are selected.

Burst Access: The burst read/burst write mode of the synchronous DRAM is not supported. When setting the mode register of the synchronous DRAM, set to the burst read/single write and set the burst length to 1.

CAS Latency: When connecting a synchronous DRAM having CAS latency of 1, set the BE bit to 0 in the DRAMCR.

Section 7 DMA Controller (DMAC)

This LSI has a built-in DMA controller (DMAC) which can carry out data transfer on up to 4 channels.

7.1 Features

- Selectable as short address mode or full address mode

  Short address mode

  Maximum of 4 channels can be used

  Dual address mode or single address mode can be selected

  In dual address mode, one of the two addresses, transfer source and transfer destination, is specified as 24 bits and the other as 16 bits

  In single address mode, transfer source or transfer destination address only is specified as 24 bits

  In single address mode, transfer can be performed in one bus cycle

  Choice of sequential mode, idle mode, or repeat mode for dual address mode and single address mode

  Full address mode

  Maximum of 2 channels can be used

  Transfer source and transfer destination addresses as specified as 24 bits

  Choice of normal mode or block transfer mode
- 16-Mbyte address space can be specified directly
- Byte or word can be set as the transfer unit
- Activation sources: internal interrupt, external request, auto-request (depending on transfer mode)

  Six 16-bit timer-pulse unit (TPU) compare match/input capture interrupts

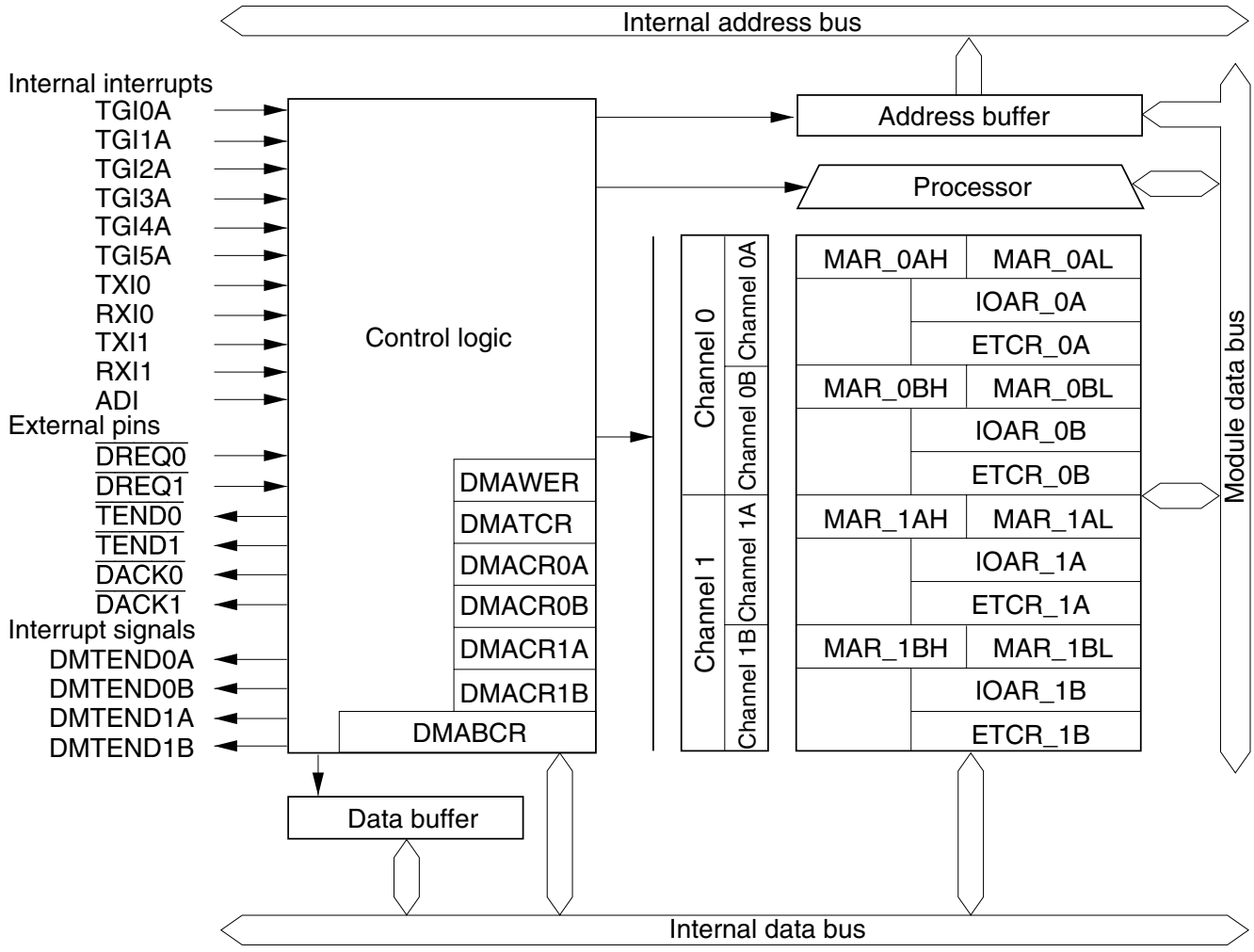
  Serial communication interface (SCI_0, SCI_1) transmission complete interrupt, reception complete interrupt

  A/D converter conversion end interrupt

  External request

  Auto-request
- Module stop mode can be set

A block diagram of the DMAC is shown in figure 7.1.



Legend

DMAWER : DMA write enable register
DMATCR : DMA terminal control register
DMABCR : DMA band control register (for all channels)
DMACR : DMA control register
MAR : Memory address register
IOAR : I/O address register
ETCR : Execute transfer count register

Figure 7.1 Block Diagram of DMAC

7.2 Input/Output Pins

Table 7.1 shows the DMAC pin configuration.

Table 7.1 Pin Configuration

Channel	Pin Name	Symbol	I/O	Function
0	DMA request 0	DREQ0	Input	Channel 0 external request
	DMA transfer acknowledge 0	DACK0	Output	Channel 0 single address transfer acknowledge
	DMA transfer end 0	TEND0	Output	Channel 0 transfer end
1	DMA request 1	DREQ1	Input	Channel 1 external request
	DMA transfer acknowledge 1	DACK1	Output	Channel 1 single address transfer acknowledge
	DMA transfer end 1	TEND1	Output	Channel 1 transfer end

7.3 Register Descriptions

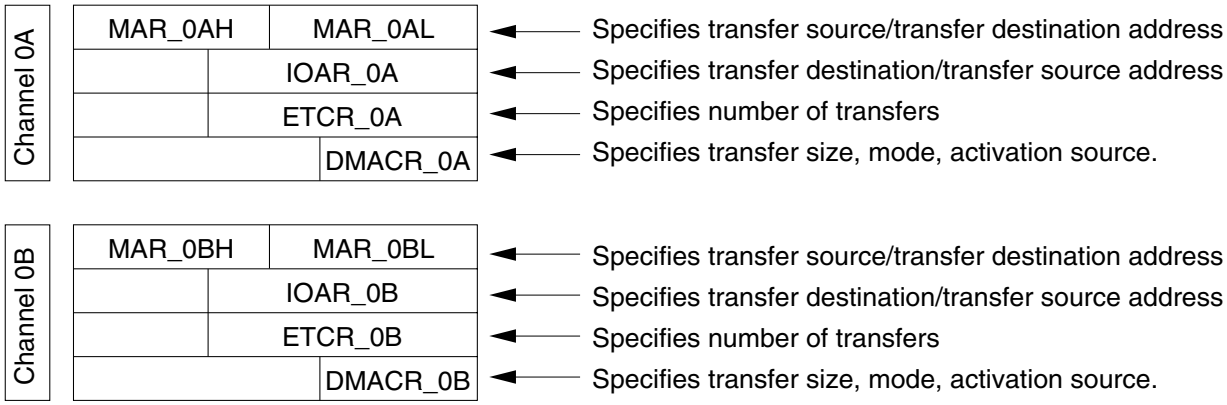
- Memory address register_0AH (MAR_0AH)
- Memory address register_0AL (MAR_0AL)
- I/O address register_0A (IOAR_0A)
- Transfer count register_0A (ECTR_0A)
- Memory address register_0BH (MAR_0BH)
- Memory address register_0BL (MAR_0BL)
- I/O address register_0B (IOAR_0B)
- Transfer count register_0B (ECTR_0B)
- Memory address register_1AH (MAR_1AH)
- Memory address register_1AL (MAR_1AL)
- I/O address register_1A (IOAR_1A)
- Transfer count register_1A (ETCR_1B)
- Memory address register_1BH (MAR_1BH)
- Memory address register_1BL (MAR_1BL)
- I/O address register_1B (IOAR_1B)
- Transfer count register_1B (ETCR_1B)
- DMA control register_0A (DMACR_0A)
- DMA control register_0B (DMACR_0B)
- DMA control register_1A (DMACR_1A)
- DMA control register_1B (DMACR_1B)

- DMA band control register H (DMABCRH)
- DMA band control register L (DMABCRL)
- DMA write enable register (DMAWER)
- DMA terminal control register (DMATCR)

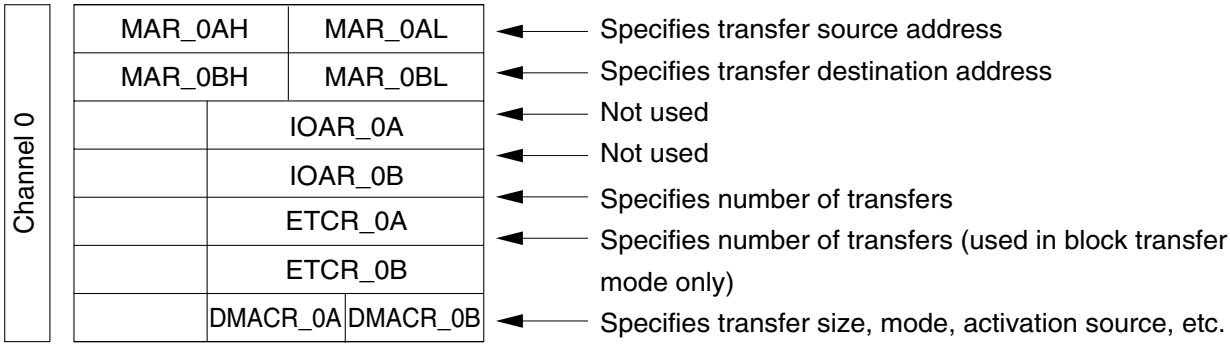
The functions of MAR, IOAR, ETCR, DMACR, and DMABCR differ according to the transfer mode (short address mode or full address mode). The transfer mode can be selected by means of the FAE1 and FAE0 bits in DMABCRH. The register configurations for short address mode and full address mode of channel 0 are shown in table 7.2.

Table 7.2 Short Address Mode and Full Address Mode (Channel 0)

FAE0	Description
0	Short address mode specified (channels 0A and 0B operate independently)



1	Full address mode specified (channels 0A and 0B operate in combination as channel 0)



7.3.1 Memory Address Registers (MARA and MARB)

MAR is a 32-bit readable/writable register that specifies the source address (transfer source address) or destination address (transfer destination address). MAR consists of two 16-bit registers MARH and MARL. The upper 8 bits of MARH are reserved: they are always read as 0, and cannot be modified.

The DMA has four MAR registers: MAR_0A in channel 0 (channel 0A), MAR_0B in channel 0 (channel 0B), MAR_1A in channel 1 (channel 1A), and MAR_1B in channel 1 (channel 1B).

MAR is not initialized by a reset or in standby mode.

Short Address Mode: In short address mode, MARA and MARB operate independently. Whether MAR functions as the source address register or as the destination address register can be selected by means of the DTDIR bit in DMACR.

MAR is incremented or decremented each time a byte or word transfer is executed, so that the address specified by MAR is constantly updated.

Full Address Mode: In full address mode, MARA functions as the source address register, and MARB as the destination address register.

MAR is incremented or decremented each time a byte or word transfer is executed, so that the source or destination address is constantly updated.

7.3.2 I/O Address Registers (IOARA and IOARB)

IOAR is a 16-bit readable/writable register that specifies the lower 16 bits of the source address (transfer source address) or destination address (transfer destination address). The upper 8 bits of the transfer address are automatically set to H'FF.

The DMA has four IOAR registers: IOAR_0A in channel 0 (channel 0A), IOAR_0B in channel 0 (channel 0B), IOAR_1A in channel 1 (channel 1A), and IOAR_1B in channel 1 (channel 1B).

Whether IOAR functions as the source address register or as the destination address register can be selected by means of the DTDIR bit in DMACR.

IOAR is not incremented or decremented each time a data transfer is executed, so the address specified by IOAR is fixed.

IOAR is not initialized by a reset or in standby mode.

IOAR can be used in short address mode but not in full address mode.

7.3.3 Execute Transfer Count Registers (ETCRA and ETCRB)

ETCR is a 16-bit readable/writable register that specifies the number of transfers.

The DMA has four ETCR registers: ETCR_0A in channel 0 (channel 0A), ETCR_0B in channel 0 (channel 0B), ETCR_1A in channel 1 (channel 1A), and ETCR_1B in channel 1 (channel 1B).

ETCR is not initialized by a reset or in standby mode.

Short Address Mode: The function of ETCR in sequential mode and idle mode differs from that in repeat mode.

In sequential mode and idle mode, ETCR functions as a 16-bit transfer counter. ETCR is decremented by 1 each time a transfer is performed, and when the count reaches H'00, the DTE bit in DMABCRL is cleared, and transfer ends.

In repeat mode, ETCRL functions as an 8-bit transfer counter and ETCRH functions as a transfer count holding register. ETCRL is decremented by 1 each time a transfer is performed, and when the count reaches H'00, ETCRL is loaded with the value in ETCRH. At this point, MAR is automatically restored to the value it had when the count was started. The DTE bit in DMABCRL is not cleared, and so transfers can be performed repeatedly until the DTE bit is cleared by the user.

Full Address Mode: The function of ETCR in normal mode differs from that in block transfer mode.

In normal mode, ETCRA functions as a 16-bit transfer counter. ETCRA is decremented by 1 each time a data transfer is performed, and transfer ends when the count reaches H'0000. ETCRB is not used in normal mode.

In block transfer mode, ETCRAL functions as an 8-bit block size counter and ETCRAH functions as a block size holding register. ETCRAL is decremented by 1 each time a 1-byte or 1-word transfer is performed, and when the count reaches H'00, ETCRAL is loaded with the value in ETCRAH. So by setting the block size in ETCRAH and ETCRAL, it is possible to repeatedly transfer blocks consisting of any desired number of bytes or words.

In block transfer mode, ETCRB functions as a 16-bit block transfer counter. ETCRB is decremented by 1 each time a block is transferred, and transfer ends when the count reaches H'0000.

7.3.4 DMA Control Registers (DMACRA and DMACRB)

DMACR controls the operation of each DMAC channel.

The DMA has four DMACR registers: DMACR_0A in channel 0 (channel 0A), DMACR_0B in channel 0 (channel 0B), DMACR_1A in channel 1 (channel 1A), and DMACR_1B in channel 1 (channel 1B).

In short address mode, channels A and B operate independently, and in full address mode, channels A and B operate together. The bit functions in the DMACR registers differ according to the transfer mode.

Short Address Mode:

- DMACR_0A, DMACR_0B, DMACR_1A, and DMARC_1B

Bit	Bit Name	Initial Value	R/W	Description
7	DTSZ	0	R/W	Data Transfer Size Selects the size of data to be transferred at one time. 0: Byte-size transfer 1: Word-size transfer
6	DTID	0	R/W	Data Transfer Increment/Decrement Selects incrementing or decrementing of MAR after every data transfer in sequential mode or repeat mode. In idle mode, MAR is neither incremented nor decremented. 0: MAR is incremented after a data transfer (Initial value) • When DTSZ = 0, MAR is incremented by 1 • When DTSZ = 1, MAR is incremented by 2 1: MAR is decremented after a data transfer • When DTSZ = 0, MAR is decremented by 1 • When DTSZ = 1, MAR is decremented by 2

Bit	Bit Name	Initial Value	R/W	Description
5	RPE	0	R/W	Repeat Enable Used in combination with the DTIE bit in DMABCR to select the mode (sequential, idle, or repeat) in which transfer is to be performed. • When DTIE = 0 (no transfer end interrupt) 0: Transfer in sequential mode 1: Transfer in repeat mode • When DTIE = 1 (with transfer end interrupt) 0: Transfer in sequential mode 1: Transfer in idle mode
4	DTDIR	0	R/W	Data Transfer Direction Used in combination with the SAE bit in DMABCR to specify the data transfer direction (source or destination). The function of this bit is therefore different in dual address mode and single address mode. • When SAE = 0 0: Transfer with MAR as source address and DACK pin as write strobe 1: Transfer with DACK pin as read strobe and MAR as destination address • When SAE = 1 0: Transfer with MAR as source address and IOAR as destination address 1: Transfer with IOAR as source address and MAR as destination address
3	DTF3	0	R/W	Data Transfer Factor 3 to 0 These bits select the data transfer factor (activation source). There are some differences in activation sources for channel A and channel B.
2	DTF2	0	R/W	
1	DTF1	0	R/W	
0	DTF0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
				• Channel A 0000: Setting prohibited 0001: Activated by A/D converter conversion end interrupt 0010: Setting prohibited 0011: Setting prohibited 0100: Activated by SCI channel 0 transmission complete interrupt 0101: Activated by SCI channel 0 reception complete interrupt 0110: Activated by SCI channel 1 transmission complete interrupt 0111: Activated by SCI channel 1 reception complete interrupt 1000: Activated by TPU channel 0 compare match/input capture A interrupt 1001: Activated by TPU channel 1 compare match/input capture A interrupt 1010: Activated by TPU channel 2 compare match/input capture A interrupt 1011: Activated by TPU channel 3 compare match/input capture A interrupt 1100: Activated by TPU channel 4 compare match/input capture A interrupt 1101: Activated by TPU channel 5 compare match/input capture A interrupt 1110: Setting prohibited 1111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
				• Channel B 0000: Setting prohibited 0001: Activated by A/D converter conversion end interrupt 0010: Activated by DREQ pin rising edge input (detected as a low level in the first transfer after transfer is enabled) 0011: Activated by DREQ pin low-level input 0100: Activated by SCI channel 0 transmission complete interrupt 0101: Activated by SCI channel 0 reception complete interrupt 0110: Activated by SCI channel 1 transmission complete interrupt 0111: Activated by SCI channel 1 reception complete interrupt 1000: Activated by TPU channel 0 compare match/input capture A interrupt 1001: Activated by TPU channel 1 compare match/input capture A interrupt 1010: Activated by TPU channel 2 compare match/input capture A interrupt 1011: Activated by TPU channel 3 compare match/input capture A interrupt 1100: Activated by TPU channel 4 compare match/input capture A interrupt 1101: Activated by TPU channel 5 compare match/input capture A interrupt 1110: Setting prohibited 1111: Setting prohibited The same factor can be selected for more than one channel. In this case, activation starts with the highest-priority channel according to the relative channel priorities. For relative channel priorities, see section 7.5.12, Multi-Channel Operation.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	—	All 0	R/W	Reserved These bits can be read from or written to. However, the write value should always be 0.

Legend

x: Don't care

- DMACR_0B and DMACR_1B

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.
6	DAID	0	R/W	Destination Address Increment/Decrement
5	DAIDE	0	R/W	Destination Address Increment/Decrement Enable These bits specify whether destination address register MARB is to be incremented, decremented, or left unchanged, when data transfer is performed. 00: MARB is fixed 01: MARB is incremented after a data transfer • When DTSZ = 0, MARB is incremented by 1 • When DTSZ = 1, MARB is incremented by 2 10: MARB is fixed 11: MARB is decremented after a data transfer • When DTSZ = 0, MARB is decremented by 1 • When DTSZ = 1, MARB is decremented by 2
4	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.
3	DTF3	0	R/W	Data Transfer Factor 3 to 0 These bits select the data transfer factor (activation source). The factors that can be specified differ between normal mode and block transfer mode.
2	DTF2	0	R/W	
1	DTF1	0	R/W	
0	DTF0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
				• Normal Mode 0000: Setting prohibited 0001: Setting prohibited 0010: Activated by DREQ pin falling edge input (detected as a low level in the first transfer after transfer is enabled) 0011: Activated by DREQ pin low-level input 010x: Setting prohibited 0110: Auto-request (cycle steal) 0111: Auto-request (burst) 1xxx: Setting prohibited • Block Transfer Mode 0000: Setting prohibited 0001: Activated by A/D converter conversion end interrupt 0010: Activated by DREQ pin falling edge input (detected as a low level in the first transfer after transfer is enabled) 0011: Activated by DREQ pin low-level input 0100: Activated by SCI channel 0 transmission complete interrupt 0101: Activated by SCI channel 0 reception complete interrupt 0110: Activated by SCI channel 1 transmission complete interrupt 0111: Activated by SCI channel 1 reception complete interrupt 1000: Activated by TPU channel 0 compare match/input capture A interrupt 1001: Activated by TPU channel 1 compare match/input capture A interrupt 1010: Activated by TPU channel 2 compare match/input capture A interrupt 1011: Activated by TPU channel 3 compare match/input capture A interrupt 1100:Activated by TPU channel 4 compare match/input capture A interrupt 1101:Activated by TPU channel 5 compare match/input capture A interrupt 1110: Setting prohibited 1111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
				The same factor can be selected for more than one channel. In this case, activation starts with the highest-priority channel according to the relative channel priorities. For relative channel priorities, see section 7.5.12, Multi-Channel Operation.

Legend
x: Don't care

7.3.5 DMA Band Control Registers H and L (DMABCRH and DMABCRL)

DMABCR controls the operation of each DMAC channel. The bit functions in the DMACR registers differ according to the transfer mode.

Short Address Mode:

- DMABCRH

Bit	Bit Name	Initial Value	R/W	Description
15	FAE1	0	R/W	Full Address Enable 1 Specifies whether channel 1 is to be used in short address mode or full address mode. In short address mode, channels 1A and 1B can be used as independent channels. 0: Short address mode 1: Full address mode
14	FAE0	0	R/W	Full Address Enable 0 Specifies whether channel 0 is to be used in short address mode or full address mode. In short address mode, channels 0A and 0B can be used as independent channels. 0: Short address mode 1: Full address mode
13	SAE1	0	R/W	Single Address Enable 1 Specifies whether channel 1B is to be used for transfer in dual address mode or single address mode. This bit is invalid in full address mode. 0: Dual address mode 1: Single address mode

Bit	Bit Name	Initial Value	R/W	Description
12	SAE0	0	R/W	Single Address Enable 0 Specifies whether channel 0B is to be used for transfer in dual address mode or single address mode. This bit is invalid in full address mode. 0: Dual address mode 1: Single address mode
11	DTA1B	0	R/W	Data Transfer Acknowledge 1B
10	DTA1A	0	R/W	Data Transfer Acknowledge 1A
9	DTA0B	0	R/W	Data Transfer Acknowledge 0B
8	DTA0A	0	R/W	Data Transfer Acknowledge 0A These bits enable or disable clearing when DMA transfer is performed for the internal interrupt source selected by the DTF3 to DTF0 bits in DMACR. It the DTA bit is set to 1 when DTE = 1, the internal interrupt source is cleared automatically by DMA transfer. When DTE = 1 and DTA = 1, the internal interrupt source does not issue an interrupt request to the CPU or DTC. If the DTA bit is cleared to 0 when DTE = 1, the internal interrupt source is not cleared when a transfer is performed, and can issue an interrupt request to the CPU or DTC in parallel. In this case, the interrupt source should be cleared by the CPU or DTC transfer. When DTE = 0, the internal interrupt source issues an interrupt request to the CPU or DTC regardless of the DTA bit setting.

- DMABCRL

Bit	Bit Name	Initial Value	R/W	Description
7	DTE1B	0	R/W	Data Transfer Enable 1B
6	DTE1A	0	R/W	Data Transfer Enable 1A
5	DTE0B	0	R/W	Data Transfer Enable 0B
4	DTE0A	0	R/W	Data Transfer Enable 0A If the DTIE bit is set to 1 when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC. When DTE = 1, data transfer is enabled and the DMAC waits for a request by the activation source selected by the DTF3 to DTF0 bits in DMACR. When a request is issued by the activation source, DMA transfer is executed. [Clearing conditions] • When initialization is performed • When the specified number of transfers have been completed in a transfer mode other than repeat mode • When 0 is written to the DTE bit to forcibly suspend the transfer, or for a similar reason [Setting condition] When 1 is written to the DTE bit after reading DTE = 0
3	DTIE1B	0	R/W	Data Transfer End Interrupt Enable 1B
2	DTIE1A	0	R/W	Data Transfer End Interrupt Enable 1A
1	DTIE0B	0	R/W	Data Transfer End Interrupt Enable 0B
0	DTIE0A	0	R/W	Data Transfer End Interrupt Enable 0A These bits enable or disable an interrupt to the CPU or DTC when transfer ends. If the DTIE bit is set to 1 when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC. A transfer end interrupt can be canceled either by clearing the DTIE bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE bit to 1.

Full Address Mode:

- DMABCRH

Bit	Bit Name	Initial Value	R/W	Description
15	FAE1	0	R/W	Full Address Enable 1 Specifies whether channel 1 is to be used in short address mode or full address mode. In full address mode, channels 1A and 1B are used together as channel 1. 0: Short address mode 1: Full address mode
14	FAE0	0	R/W	Full Address Enable 0 Specifies whether channel 0 is to be used in short address mode or full address mode. In full address mode, channels 0A and 0B are used together as channel 0. 0: Short address mode 1: Full address mode
13	—	0	R/W	Reserved
12	—	0	R/W	These bits can be read from or written to. However, the write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
11	DTA1	0	R/W	Data Transfer Acknowledge 1 These bits enable or disable clearing when DMA transfer is performed for the internal interrupt source selected by the DTF3 to DTF0 bits in DMACR of channel 1. It the DTA1 bit is set to 1 when DTE1 = 1, the internal interrupt source is cleared automatically by DMA transfer. When DTE1 = 1 and DTA1 = 1, the internal interrupt source does not issue an interrupt request to the CPU or DTC. It the DTA1 bit is cleared to 0 when DTE1 = 1, the internal interrupt source is not cleared when a transfer is performed, and can issue an interrupt request to the CPU or DTC in parallel. In this case, the interrupt source should be cleared by the CPU or DTC transfer. When DTE1 = 0, the internal interrupt source issues an interrupt request to the CPU or DTC regardless of the DTA1 bit setting. The state of the DTME1 bit does not affect the above operations.
10	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	DTA0	0	R/W	Data Transfer Acknowledge 0 These bits enable or disable clearing when DMA transfer is performed for the internal interrupt source selected by the DTF3 to DTF0 bits in DMACR of channel 0. It the DTA0 bit is set to 1 when DTE0 = 1, the internal interrupt source is cleared automatically by DMA transfer. When DTE0 = 1 and DTA0 = 1, the internal interrupt source does not issue an interrupt request to the CPU or DTC. It the DTA0 bit is cleared to 0 when DTE0 = 1, the internal interrupt source is not cleared when a transfer is performed, and can issue an interrupt request to the CPU or DTC in parallel. In this case, the interrupt source should be cleared by the CPU or DTC transfer. When DTE0 = 0, the internal interrupt source issues an interrupt request to the CPU or DTC regardless of the DTA0 bit setting. The state of the DTME0 bit does not affect the above operations.
8	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.

- DMABCRL

Bit	Bit Name	Initial Value	R/W	Description
7	DTME1	0	R/W	Data Transfer Master Enable 1 Together with the DTE1 bit, this bit controls enabling or disabling of data transfer on channel 1. When both the DTME1 bit and DTE1 bit are set to 1, transfer is enabled for channel 1. If channel 1 is in the middle of a burst mode transfer when an NMI interrupt is generated, the DTME1 bit is cleared, the transfer is interrupted, and bus mastership passes to the CPU. When the DTME1 bit is subsequently set to 1 again, the interrupted transfer is resumed. In block transfer mode, however, the DTME1 bit is not cleared by an NMI interrupt, and transfer is not interrupted. [Clearing conditions] • When initialization is performed • When NMI is input in burst mode • When 0 is written to the DTME1 bit [Setting condition] When 1 is written to DTME1 after reading DTME1 = 0

Bit	Bit Name	Initial Value	R/W	Description
6	DTE1	0	R/W	Data Transfer Enable 1 Enables or disables DMA transfer for the activation source selected by the DTF3 to DTF0 bits in DMACR of channel 1. When DTE1 = 0, data transfer is disabled and the activation source is ignored. If the activation source is an internal interrupt, an interrupt request is issued to the CPU or DTC. If the DTIE1 bit is set to 1 when DTE1 = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU. When DTE1 = 1 and DTME1 = 1, data transfer is enabled and the DMAC waits for a request by the activation source. When a request is issued by the activation source, DMA transfer is executed. [Clearing conditions] • When initialization is performed • When the specified number of transfers have been completed • When 0 is written to the DTE1 bit to forcibly suspend the transfer, or for a similar reason [Setting condition] When 1 is written to the DTE1 bit after reading DTE1 = 0

Bit	Bit Name	Initial Value	R/W	Description
5	DTME0	0	R/W	Data Transfer Master Enable 0 Together with the DTE0 bit, this bit controls enabling or disabling of data transfer on channel 0. When both the DTME0 bit and DTE0 bit are set to 1, transfer is enabled for channel 0. If channel 0 is in the middle of a burst mode transfer when an NMI interrupt is generated, the DTME0 bit is cleared, the transfer is interrupted, and bus mastership passes to the CPU. When the DTME0 bit is subsequently set to 1 again, the interrupted transfer is resumed. In block transfer mode, however, the DTME0 bit is not cleared by an NMI interrupt, and transfer is not interrupted. [Clearing conditions] • When initialization is performed • When NMI is input in burst mode • When 0 is written to the DTME0 bit [Setting condition] When 1 is written to DTME0 after reading DTME0 = 0

Bit	Bit Name	Initial Value	R/W	Description
4	DTE0	0	R/W	Data Transfer Enable 0 Enables or disables DMA transfer for the activation source selected by the DTF3 to DTF0 bits in DMACR of channel 0. When DTE0 = 0, data transfer is disabled and the activation source is ignored. If the activation source is an internal interrupt, an interrupt request is issued to the CPU or DTC. If the DTE0 bit is cleared to 0 when DTIE0 = 1, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU. When DTE0 = 1 and DTME0 = 1, data transfer is enabled and the DMAC waits for a request by the activation source. When a request is issued by the activation source, DMA transfer is executed. [Clearing conditions] • When initialization is performed • When the specified number of transfers have been completed • When 0 is written to the DTE0 bit to forcibly suspend the transfer, or for a similar reason [Setting condition] When 1 is written to the DTE0 bit after reading DTE0 = 0
3	DTIE1B	0	R/W	Data Transfer Interrupt Enable 1B Enables or disables an interrupt to the CPU or DTC when transfer on channel 1 is interrupted. If the DTME1 bit is cleared to 0 when DTIE1B = 1, the DMAC regards this as indicating a break in the transfer, and issues a transfer break interrupt request to the CPU or DTC. A transfer break interrupt can be canceled either by clearing the DTIE1B bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the DTME1 bit to 1.

Bit	Bit Name	Initial Value	R/W	Description
2	DTIE1A	0	R/W	Data Transfer End Interrupt Enable 1A Enables or disables an interrupt to the CPU or DTC when transfer ends. If the DTE1 bit is cleared to 0 when DTIE1A= 1, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC. A transfer end interrupt can be canceled either by clearing the DTIE1A bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE1 bit to 1.
1	DTIE0B	0	R/W	Data Transfer Interrupt Enable 0B Enables or disables an interrupt to the CPU or DTC when transfer on channel 1 is interrupted. If the DTME0 bit is cleared to 0 when DTIE0B= 1, the DMAC regards this as indicating a break in the transfer, and issues a transfer break interrupt request to the CPU or DTC. A transfer break interrupt can be canceled either by clearing the DTIE0B bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the DTME0 bit to 1.
0	DTIE0A	0	R/W	Data Transfer End Interrupt Enable 0A Enables or disables an interrupt to the CPU or DTC when transfer ends. If the DTE0 bit is cleared to 0 when DTIE0A = 1, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC. A transfer end interrupt can be canceled either by clearing the DTIE0A bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE0 bit to 1.

7.3.6 DMA Write Enable Register (DMAWER)

The DMAC can activate the DTC with a transfer end interrupt, rewrite the channel on which the transfer ended using a DTC chain transfer, and then reactivate the DTC. DMAWER applies restrictions for changing all bits of DMACR, and specific bits for DMATCR and DMABCR for the specific channel, to prevent inadvertent rewriting of registers other than those for the channel concerned. The restrictions applied by DMAWER are valid for the DTC.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	WE1B	0	R/W	Write Enable 1B Enables or disables writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR. 0: Writes are disabled 1: Writes are enabled
2	WE1A	0	R/W	Write Enable 1A Enables or disables writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR. 0: Writes are disabled 1: Writes are enabled
1	WE0B	0	R/W	Write Enable 0B Enables or disables writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, and bit 4 in DMATCR. 0: Writes are disabled 1: Writes are enabled
0	WE0A	0	R/W	Write Enable 0A Enables or disables writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR. 0: Writes are disabled 1: Writes are enabled

Figure 7.2 shows the transfer areas for activating the DTC with a channel 0A transfer end interrupt request, and reactivating channel 0A. The address register and count register areas are set again during the first DTC transfer, then the control register area is set again during the second DTC chain transfer. When re-setting the control register area, perform masking by setting bits in DMAWER to prevent modification of the contents of other channels.

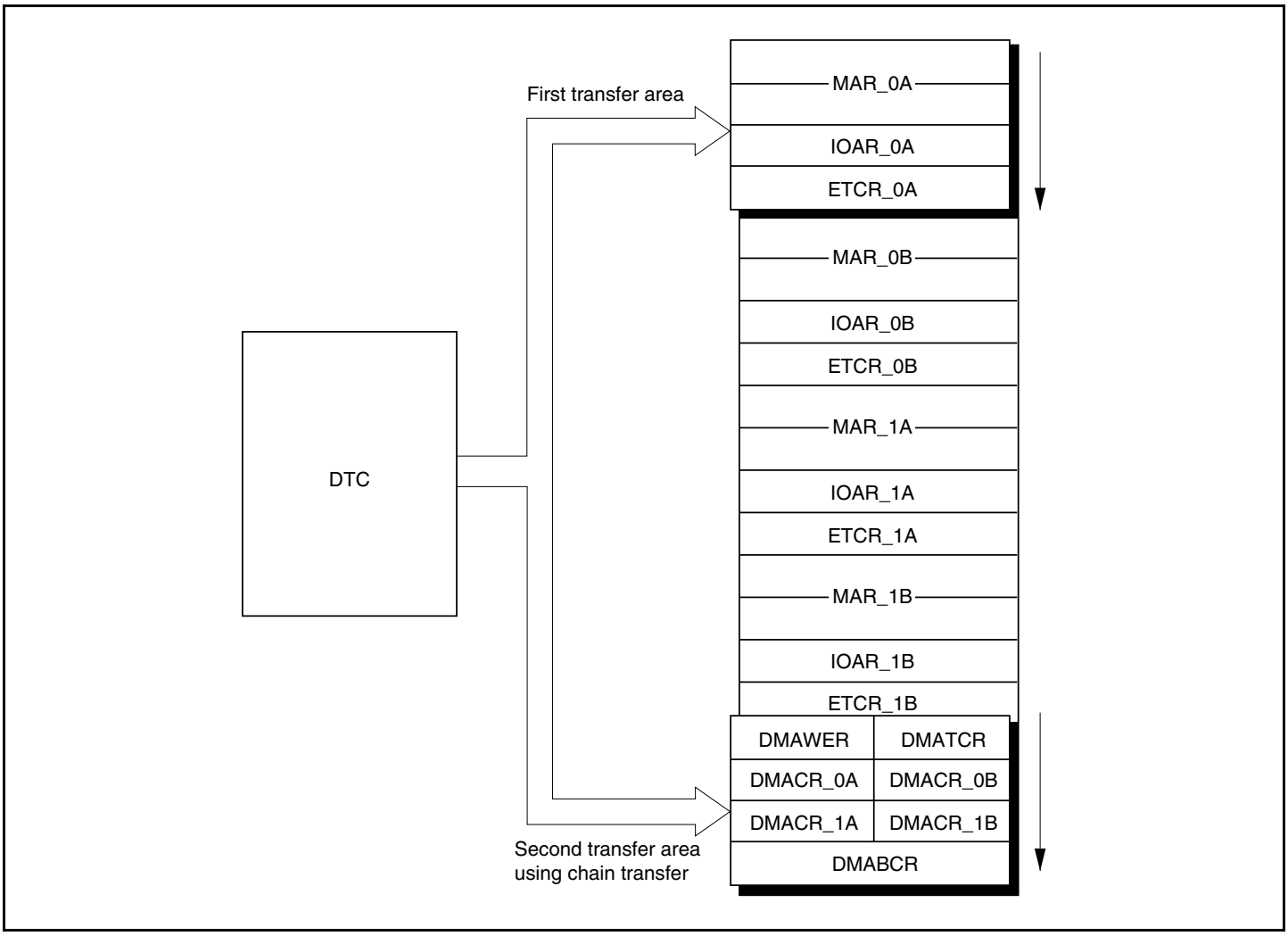


Figure 7.2 Areas for Register Re-Setting by DTC (Channel 0A)

Writes by the DTC to bits 15 to 12 (FAE and SAE) in DMABCR are invalid regardless of the DMAWER settings. These bits should be changed, if necessary, by CPU processing.

In writes by the DTC to bits 7 to 4 (DTE) in DMABCR, 1 can be written without first reading 0. To reactivate a channel set to full address mode, write 1 to both Write Enable A and Write Enable B for the channel to be reactivated.

MAR, IOAR, and ETCR can always be written to regardless of the DMAWER settings. When modifying these registers, the channel to be modified should be halted.

7.3.7 DMA Terminal Control Register (DMATCR)

DMATCR controls enabling or disabling of output from the DMAC transfer end pin. A port can be set for output automatically, and a transfer end signal output, by setting the appropriate bit. The TEND pin is available only for channel B in short address mode. Except for the block transfer mode, a transfer end signal asserts in the transfer cycle in which the transfer counter contents reaches 0 regardless of the activation source. In the block transfer mode, a transfer end signal asserts in the transfer cycle in which the block counter contents reaches 0.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	TEE1	0	R/W	Transfer End Enable 1 Enables or disables transfer end pin 1 (TEND1) output. 0: TEND1 pin output disabled 1: TEND1 pin output enabled
4	TEE0	0	R/W	Transfer End Enable 0 Enables or disables transfer end pin 0 (TEND0) output. 0: TEND0 pin output disabled 1: TEND0 pin output enabled
3 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

7.4 Activation Sources

DMAC activation sources consist of internal interrupt requests, external requests, and auto-requests. The DMAC activation sources that can be specified depend on the transfer mode and channel, as shown in table 7.3.

Table 7.3 DMAC Activation Sources

		Short Address Mode		Full Address Mode	
Activation Source		**Channels 0A and 1A**	**Channels 0B and 1B**	**Normal Mode**	**Block Transfer Mode**
Internal interrupts	ADI	○	○	X	○
	TXI0	○	○	X	○
	RXI0	○	○	X	○
	TXI1	○	○	X	○
	RXI1	○	○	X	○
	TGI0A	○	○	X	○
	TGI1A	○	○	X	○
	TGI2A	○	○	X	○
	TGI3A	○	○	X	○
	TGI4A	○	○	X	○
	TGI5A	○	○	X	○
External requests	DREQ pin falling edge input	X	○	○	○
	DREQ pin low-level input	X	○	○	○
Auto-request		X	X	○	X

Legend

○: Can be specified

X: Cannot be specified

7.4.1 Activation by Internal Interrupt Request

An interrupt request selected as a DMAC activation source can also simultaneously generate an interrupt request for the CPU or DTC. For details, see section 5, Interrupt Controller.

With activation by an internal interrupt request, the DMAC accepts the interrupt request independently of the interrupt controller. Consequently, interrupt controller priority settings are irrelevant.

If the DMAC is activated by a CPU interrupt source or an interrupt request that is not used as a DTC activation source (DTA = 1), the interrupt request flag is cleared automatically by the DMA transfer. With ADI, TXI and RXI interrupts, however, the interrupt source flag is not cleared unless the relevant register is accessed in a DMA transfer. If the same interrupt is used as an activation source for more than one channel, the interrupt request flag is cleared when the highest-priority channel is activated. Transfer requests for other channels are held pending in the DMAC, and activation is carried out in order of priority.

When DTE = 0 after completion of a transfer, an interrupt request from the selected activation source is not sent to the DMAC, regardless of the DTA bit setting. In this case, the relevant interrupt request is sent to the CPU or DTC.

When an interrupt request signal for DMAC activation is also used for an interrupt request to the CPU or DTC activation (DTA = 0), the interrupt request flag is not cleared by the DMAC.

7.4.2 Activation by External Request

If an external request (DREQ pin) is specified as a DMAC activation source, the relevant port should be set to input mode in advance. Level sensing or edge sensing can be used for external requests.

External request operation in normal mode of short address mode or full address mode is described below.

When edge sensing is selected, a byte or word is transferred each time a high-to-low transition is detected on the DREQ pin. The next data transfer may not be performed if the next edge is input before data transfer is completed.

When level sensing is selected, the DMAC stands by for a transfer requ.s07 Tc0.00(n4.081.32 1 Tf6.3374

7.5 Operation

7.5.1 Transfer Modes

Table 7.4 lists the DMAC transfer modes.

Table 7.4 DMAC Transfer Modes

Transfer Mode		Transfer Source	Remarks
Short address mode	Dual address mode • 1-byte or 1-word transfer for a single transfer request (1) Sequential mode • Memory address incremented or decremented by 1 or 2 • Number of transfers: 1 to 65,536 (2) Idle mode • Memory address fixed • Number of transfers: 1 to 65,536 (3) Repeat mode • Memory address incremented or decremented by 1 or 2 • Continues transfer after sending number of transfers (1 to 256) and restoring the initial value	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmission complete interrupt • SCI reception complete interrupt • A/D converter conversion end interrupt • External request	• Up to 4 channels can operate independently • External request applies to channel B only • Single address mode applies to channel B only • Sequential mode, idle mode, and repeat mode can also be specified for single address mode

Transfer Mode		Transfer Source	Remarks
	Single address mode • 1-byte or 1-word transfer for a single transfer request • 1-bus cycle transfer by means of DACK pin instead of using address for specifying I/O • Sequential mode, idle mode, or repeat mode can be specified	• External request	
Full address mode	Normal mode (1) Auto-request • Transfer request is internally held • Number of transfers (1 to 65,536) is continuously sent • Burst/cycle steal transfer can be selected	Auto-request	• Max. 2-channel operation, combining channels A and B • With auto-request, burst mode or cycle steal mode can be selected
	(2) External request • 1-byte or 1-word transfer for a single transfer request • Number of transfers: 1 to 65,536	External request	
	Block transfer mode • Transfer of 1-block, size selected for a single transfer request • Number of transfers: 1 to 65,536 • Source or destination can be selected as block area • Block size: 1 to 256 bytes or word	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmission complete interrupt • SCI reception complete interrupt • A/D converter conversion end interrupt • External request	

7.5.2 Sequential Mode

Sequential mode can be specified by clearing the RPE bit in DMACR to 0. In sequential mode, MAR is updated after each byte or word transfer in response to a single transfer request, and this is executed the number of times specified in ETCR. One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR.

Table 7.5 summarizes register functions in sequential mode.

Table 7.5 Register Functions in Sequential Mode

	Function			
Register	**DTDIR = 0**	**DTDIR = 1**	**Initial Setting**	**Operation**
23 MAR 0	Source address register	Destination address register	Start address of transfer destination or transfer source	Incremented/ decremented every transfer
23 15 H'FF IOAR 0	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
15 ETCR 0	Transfer counter		Number of transfers	Decremented every transfer; transfer ends when count reaches H'0000

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is incremented or decremented by 1 or 2 each time a byte or word is transferred. IOAR specifies the lower 16 bits of the other address. The 8 bits above IOAR have a value of H'FF.

Figure 7.3 illustrates operation in sequential mode.

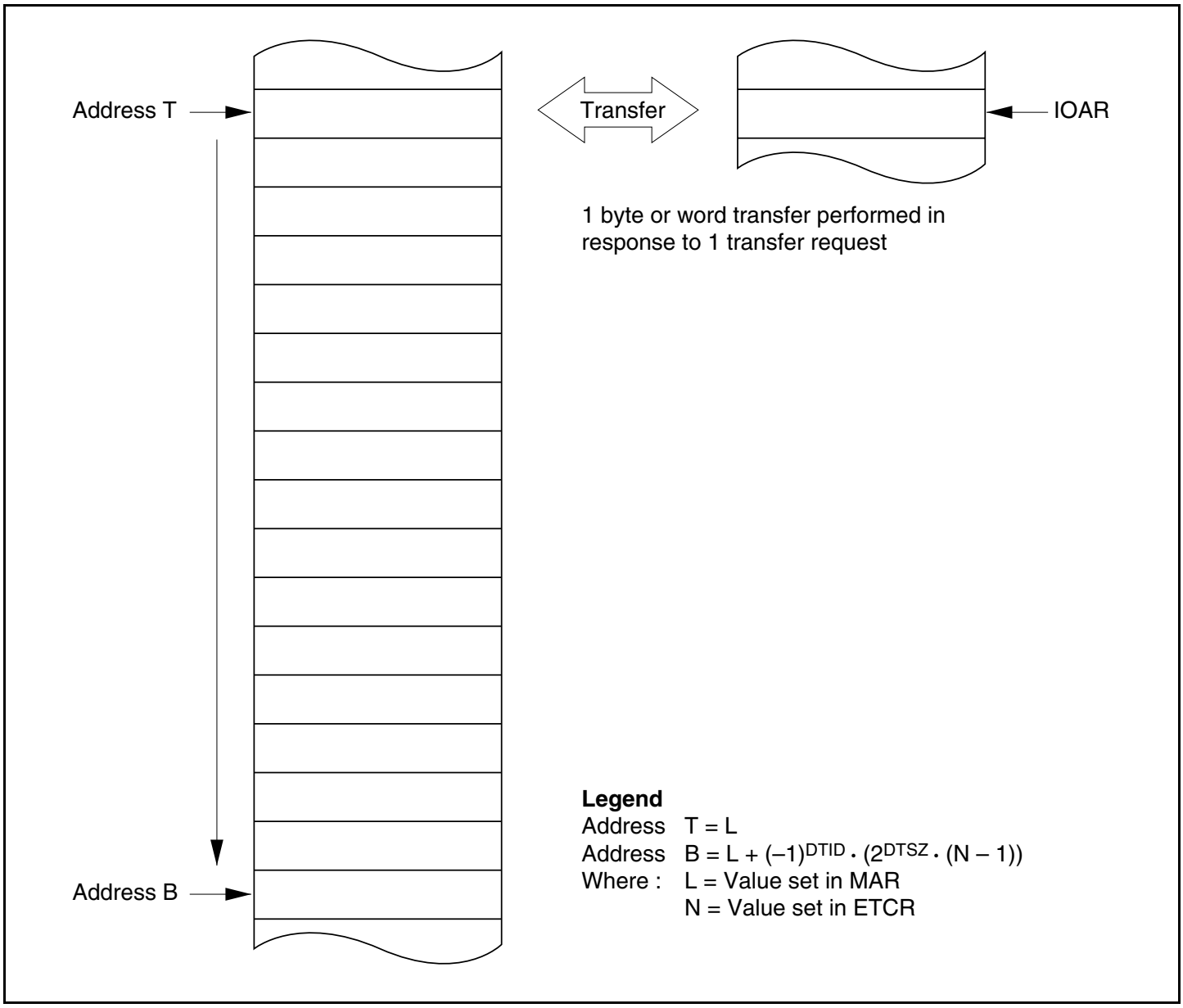
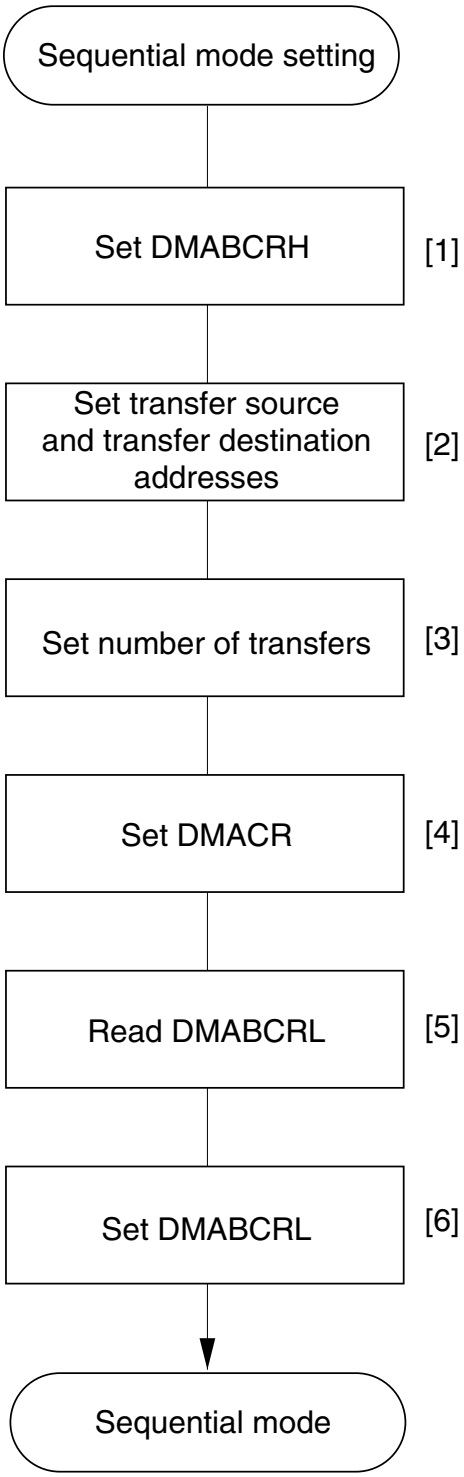


Figure 7.3 Operation in Sequential Mode

The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 each time a data transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and data transfer ends. If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC. The maximum number of transfers, when H'0000 is set in ETCR, is 65,536.

Transfer requests (activation sources) consist of external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts.

Figure 7.4 shows an example of the setting procedure for sequential mode.



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Clear the RPE bit to 0 to select sequential mode.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts with the DTIE bit.
- Set the DTE bit to 1 to enable transfer.

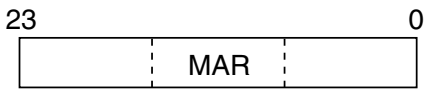
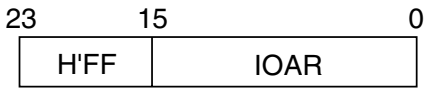
Figure 7.4 Example of Sequential Mode Setting Procedure

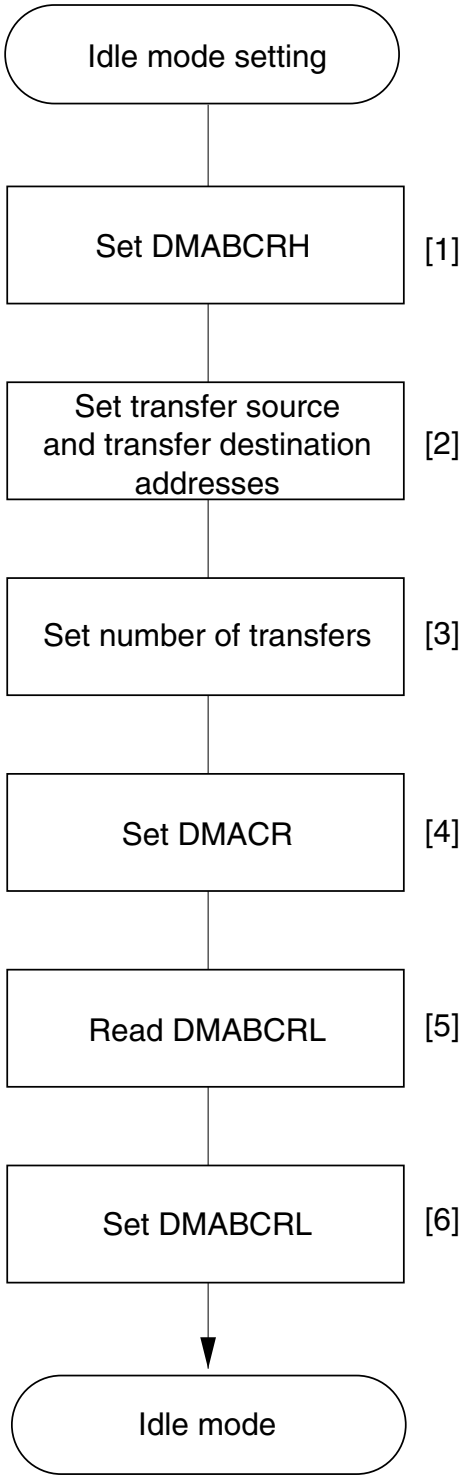
7.5.3 Idle Mode

Idle mode can be specified by setting the RPE bit in DMACR and DTIE bit in DMABCRL to 1. In idle mode, one byte or word is transferred in response to a single transfer request, and this is executed the number of times specified in ETCR. One address is specified by MAR, and the other

by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR. Table 7.6 summarizes register functions in idle mode.

Table 7.6 Register Functions in Idle Mode

	Function			
Register	**DTDIR = 0**	**DTDIR = 1**	**Initial Setting**	**Operation**
	Source address register	Destination address register	Start address of transfer destination or transfer source	Fixed
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
				



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Set the RPE bit to 1.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Set the DTIE bit to 1.
- Set the DTE bit to 1 to enable transfer.

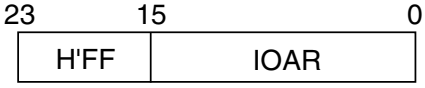
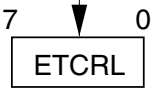
Figure 7.6 Example of Idle Mode Setting Procedure

7.5.4 Repeat Mode

Repeat mode can be specified by setting the RPE bit in DMACR to 1, and clearing the DTIE bit in DMABCRL to 0. In repeat mode, MAR is updated after each byte or word transfer in response to a single transfer request, and this is executed the number of times specified in ETCRL. On completion of the specified number of transfers, MAR and ETCRL are automatically restored to

their original settings and operation continues. One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR. Table 7.7 summarizes register functions in repeat mode.

Table 7.7 Register Functions in Repeat Mode

Register	Function DTDIR = 0	Function DTDIR = 1	Initial Setting	Operation
	Source address register	Destination address register	Start address of transfer destination or transfer source	Incremented/ decremented every transfer. Initial setting is restored when value reaches H'0000
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
	Holds number of transfers		Number of transfers	Fixed
	Transfer counter		Number of transfers	Decremented every transfer. Loaded with ETCRH value when count reaches H'00

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is incremented or decremented by 1 or 2 each time a byte or word is transferred. IOAR specifies the lower 16 bits of the other address. The upper 8 bits of IOAR have a value of H'FF. The number of transfers is specified as 8 bits by ETCRH and ETCRL. The maximum number of transfers, when H'00 is set in both ETCRH and ETCRL, is 256.

In repeat mode, ETCRL functions as the transfer counter, and ETCRH is used to hold the number of transfers. ETCRL is decremented by 1 each time a data transfer is executed, and when its value reaches H'00, it is loaded with the value in ETCRH. At the same time, the value set in MAR is restored in accordance with the values of the DTSZ and DTID bits in DMACR. The MAR restoration operation is as shown below.

$$MAR = MAR - (-1)^{DTID} \cdot 2^{DTSZ} \cdot ETCRH$$

The same value should be set in ETCRH and ETCRL.

In repeat mode, operation continues until the DTE bit in DMABCRL is cleared. To end the transfer operation, therefore, the DTE bit should be cleared to 0. A transfer end interrupt request is

not sent to the CPU or DTC. By setting the DTE bit to 1 again after it has been cleared, the operation can be restarted from the transfer after that terminated when the DTE bit was cleared.

Figure 7.7 illustrates operation in repeat mode.

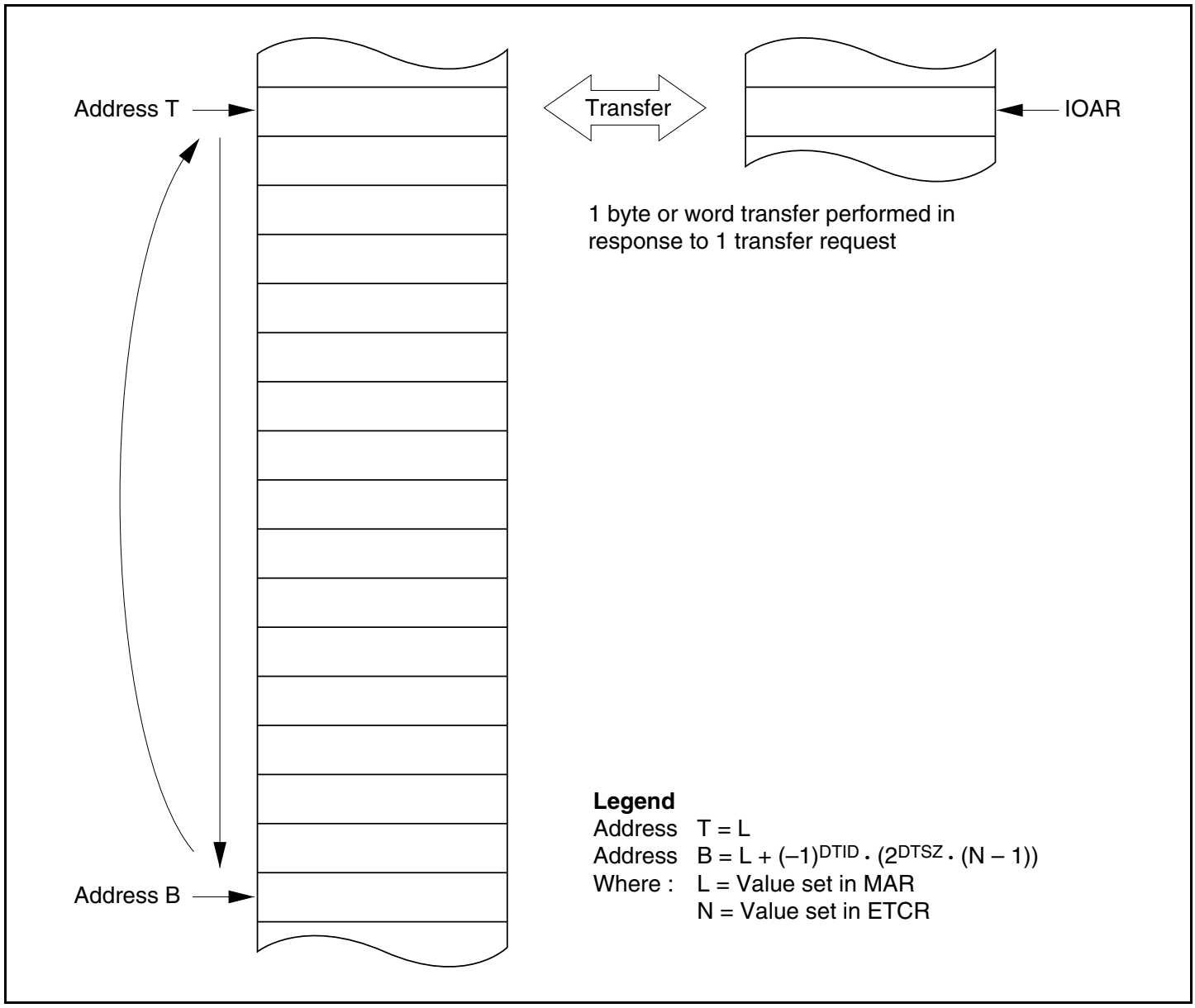
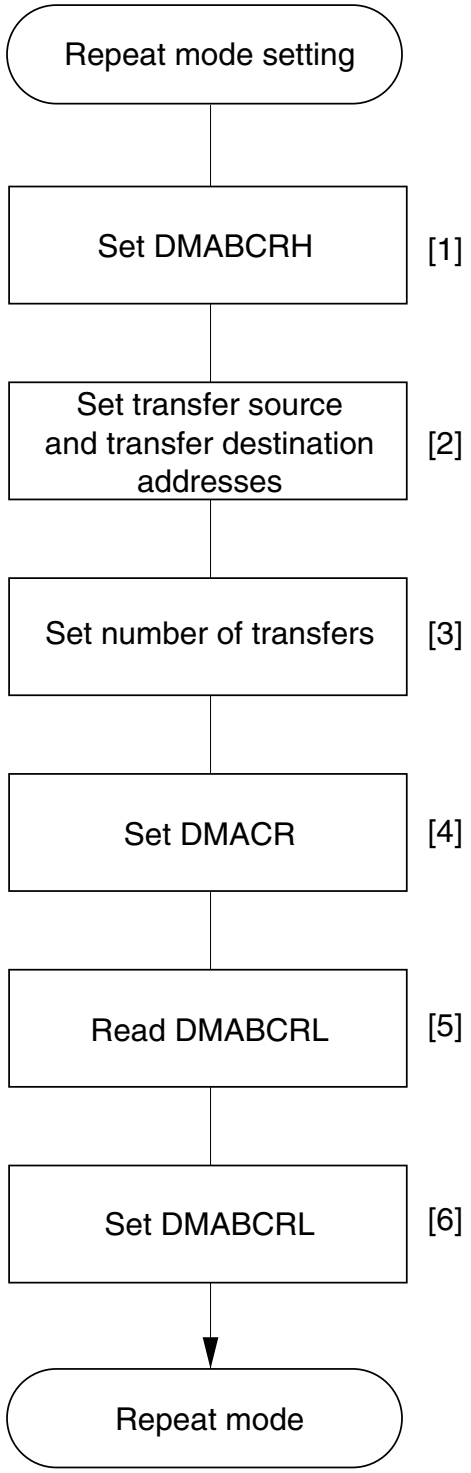


Figure 7.7 Operation in Repeat mode

Transfer requests (activation sources) consist of external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts. External requests can only be specified for channel B.

Figure 7.8 shows an example of the setting procedure for repeat mode.



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in both ETCRH and ETCRL.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Set the RPE bit to 1.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Clear the DTIE bit to 0.
- Set the DTE bit to 1 to enable transfer.

Figure 7.8 Example of Repeat Mode Setting Procedure

7.5.5 Single Address Mode

Single address mode can only be specified for channel B. This mode can be specified by setting the SAE bit in DMABCRH to 1 in short address mode.

One address is specified by MAR, and the other is set automatically to the data transfer acknowledge pin (DACK). The transfer direction can be specified by the DTDIR bit in DMACR. Table 7.8 summarizes register functions in single address mode.

Table 7.8 Register Functions in Single Address Mode

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
23 MAR 0	Source address register	Destination address register	Start address of transfer destination or transfer source	See sections 7.5.2, Sequential Mode, 7.5.3, Idle Mode, and 7.5.4, Repeat Mode.
DACK pin	Write strobe	Read strobe	(Set automatically by SAE bit; IOAR is invalid)	Strobe for external device
15 ETCR 0	Transfer counter		Number of transfers	See sections 7.5.2, Sequential Mode, 7.5.3, Idle Mode, and 7.5.4, Repeat Mode.

MAR specifies the start address of the transfer source or transfer destination as 24 bits. IOAR is invalid; in its place the strobe for external devices (DACK) is output.

Figure 7.9 illustrates operation in single address mode (when sequential mode is specified).

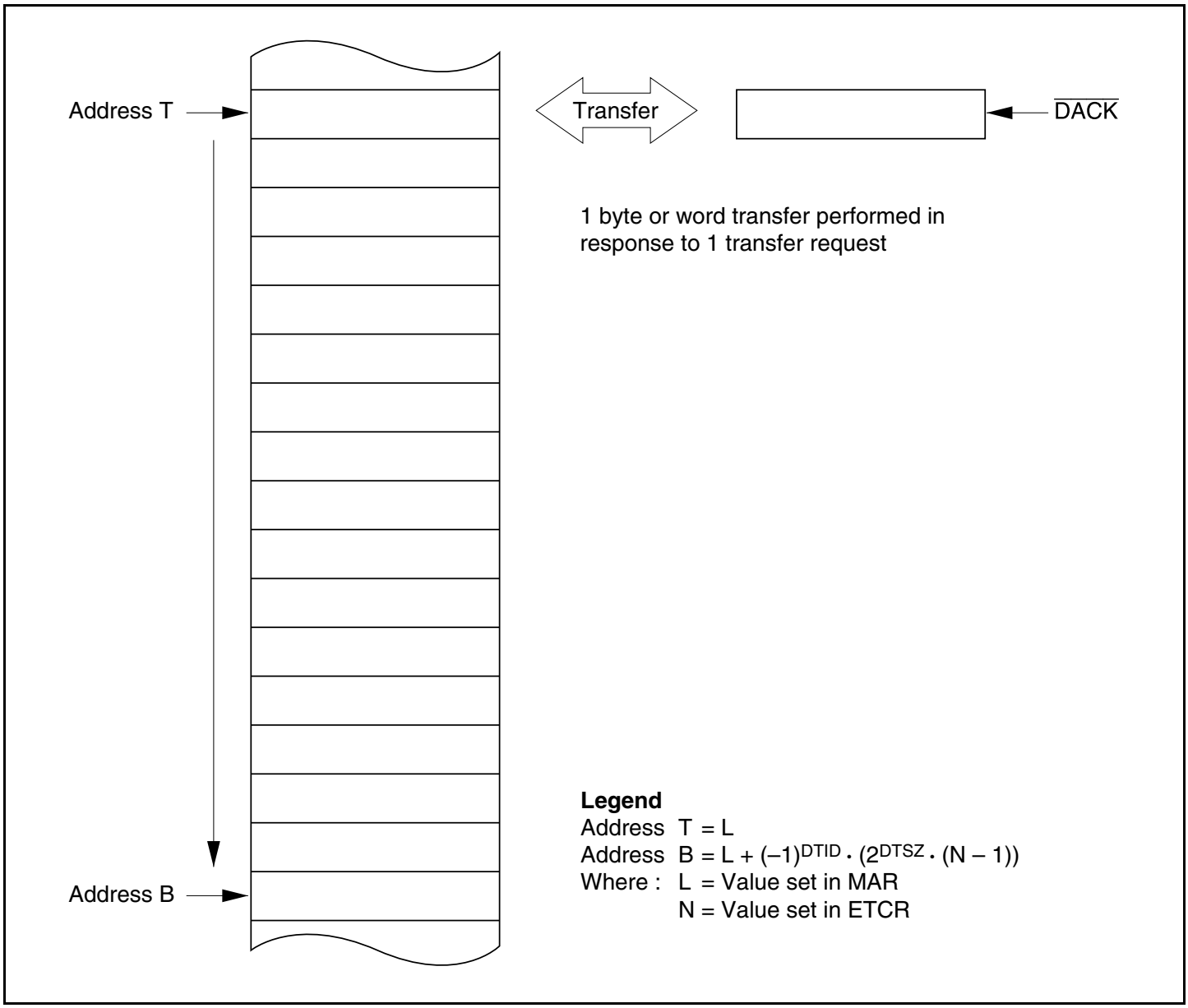
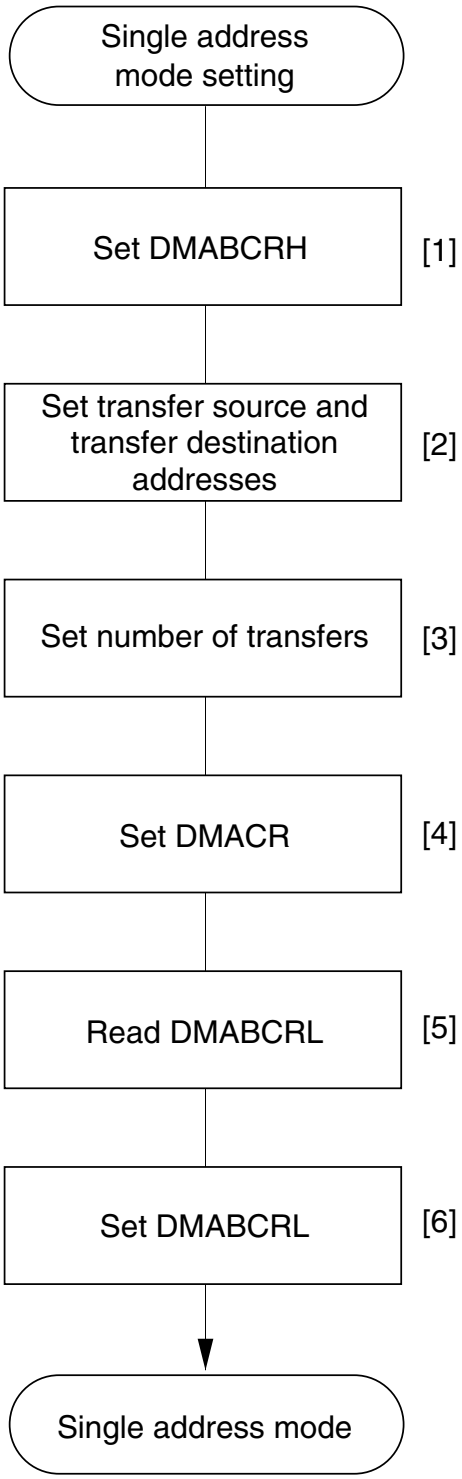


Figure 7.9 Operation in Single Address Mode (When Sequential Mode is Specified)

Figure 7.10 shows an example of the setting procedure for single address mode (when sequential mode is specified).



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Set the SAE bit to 1 to select single address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address/transfer destination address in MAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Clear the RPE bit to 0 to select sequential mode.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts with the DTIE bit.
- Set the DTE bit to 1 to enable transfer.

Figure 7.10 Example of Single Address Mode Setting Procedure (When Sequential Mode is Specified)

7.5.6 Normal Mode

In normal mode, transfer is performed with channels A and B used in combination. Normal mode can be specified by setting the FAE bit in DMABCRH to 1 and clearing the BLKE bit in DMACRA to 0. In normal mode, MAR is updated after data transfer of a byte or word in response

to a single transfer request, and this is executed the number of times specified in ETCRA. The transfer source is specified by MARA, and the transfer destination by MARB. Table 7.9

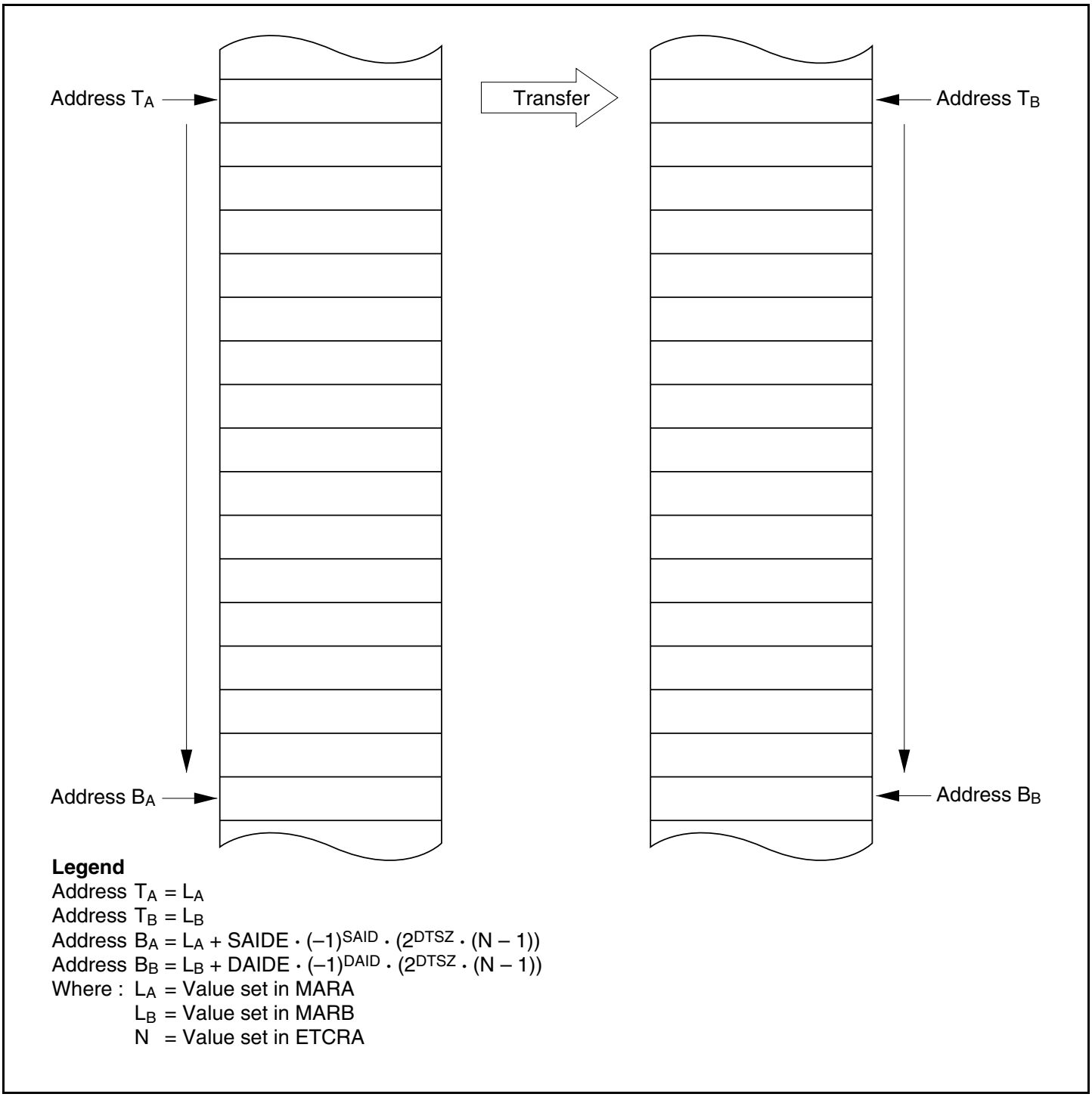
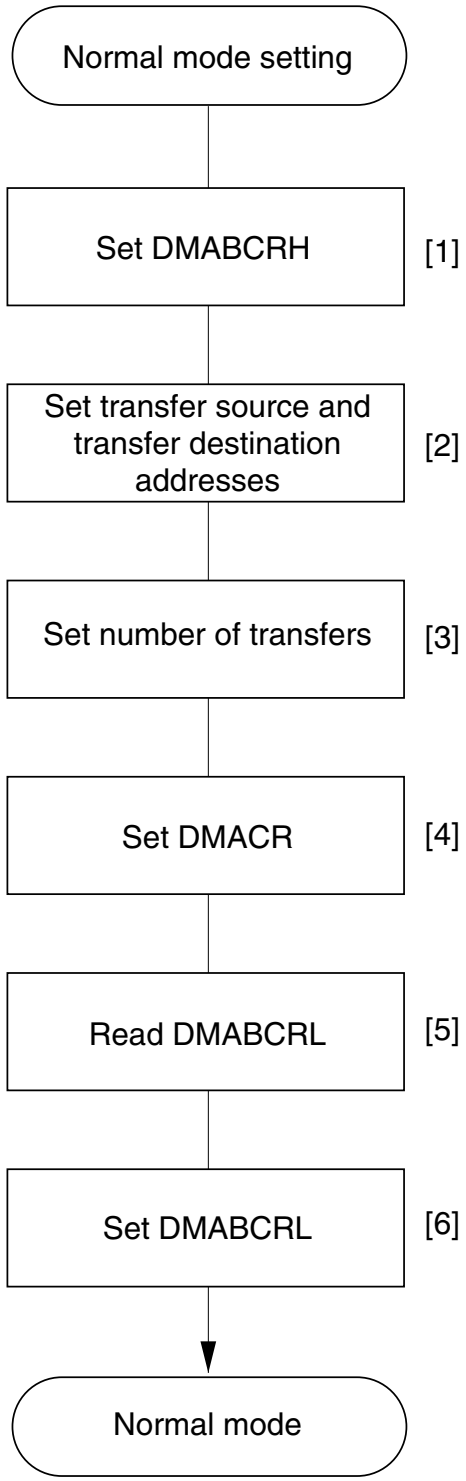


Figure 7.11 Operation in Normal Mode

Transfer requests (activation sources) are external requests and auto-requests. With auto-request, the DMAC is only activated by register setting, and the specified number of transfers are performed automatically. With auto-request, cycle steal mode or burst mode can be selected. In cycle steal mode, the bus is released to another bus master each time a transfer is performed. In burst mode, the bus is held continuously until transfer ends.

Figure 7.12 shows an example of the setting procedure for normal mode.



[1] Set each bit in DMABCRH.
- Set the FAE bit to 1 to select full address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address in MARA, and the transfer destination address in MARB.

[3] Set the number of transfers in ETCRA.

[4] Set each bit in DMACRA and DMACRB.
- Set the transfer data size with the DTSZ bit.
- Specify whether MARA is to be incremented, decremented, or fixed, with the SAID and SAIDE bits.
- Clear the BLKE bit to 0 to select normal mode.
- Specify whether MARB is to be incremented, decremented, or fixed, with the DAID and DAIDE bits.
- Select the activation source with bits DTF3 to DTF0.

[5] Read DTE = 0 and DTME = 0 in DMABCRL.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts with the DTIE bit.
- Set both the DTME bit and the DTE bit to 1 to enable transfer.

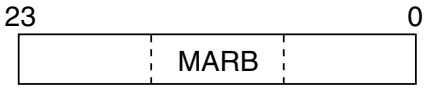
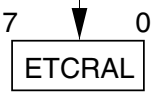
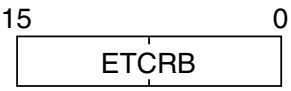
Figure 7.12 Example of Normal Mode Setting Procedure

7.5.7 Block Transfer Mode

In block transfer mode, data transfer is performed with channels A and B used in combination. Block transfer mode can be specified by setting the FAE bit in DMABCRH and the BLKE bit in DMACRA to 1. In block transfer mode, a data transfer of the specified block size is carried out in response to a single transfer request, and this is executed for the number of times specified in

ETCRB. The transfer source is specified by MARA, and the transfer destination by MARB. Either the transfer source or the transfer destination can be selected as a block area (an area composed of a number of bytes or words). Table 7.10 summarizes register functions in block transfer mode.

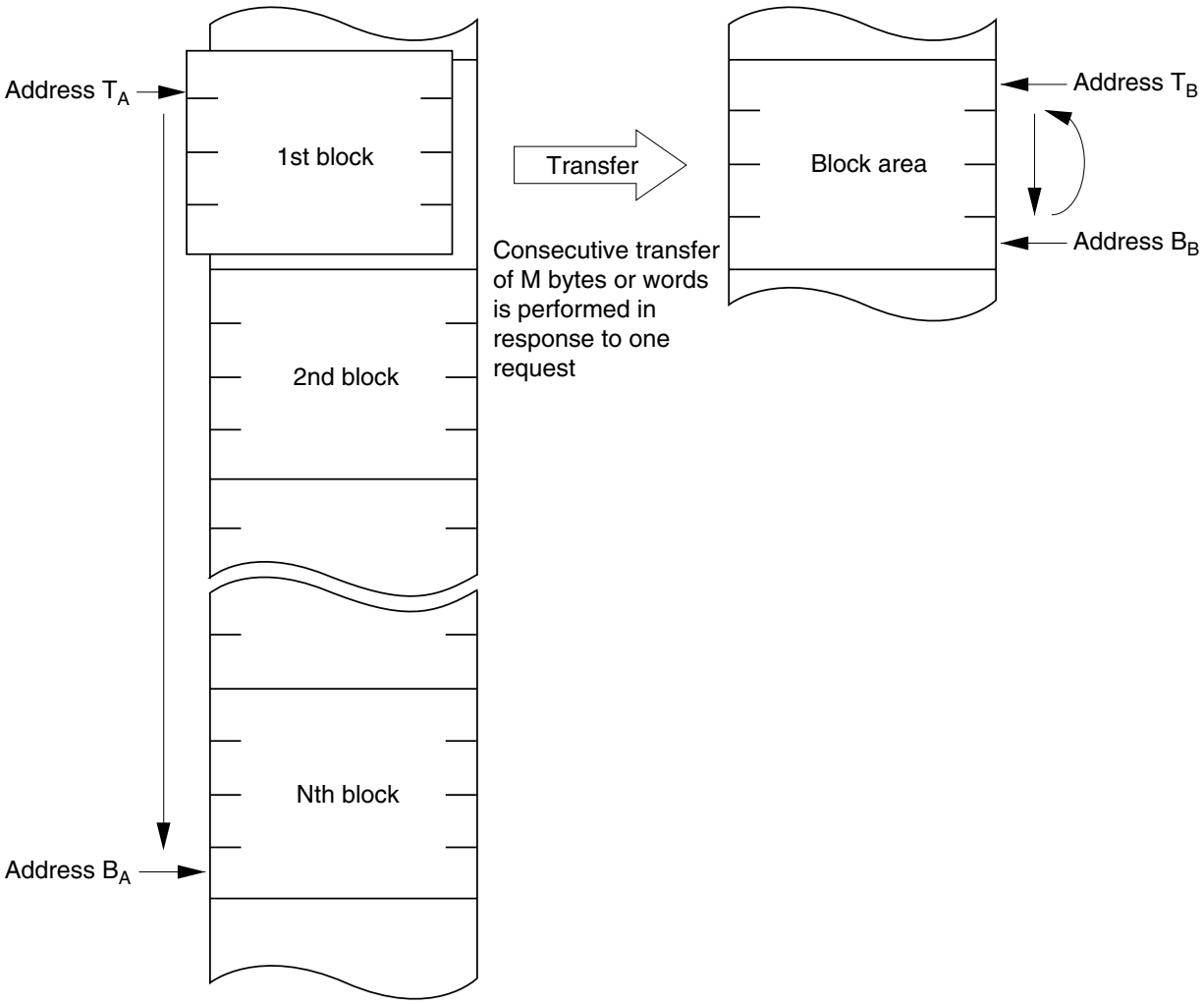
Table 7.10 Register Functions in Block Transfer Mode

Register	Function	Initial Setting	Operation
	Source address register	Start address of transfer source	Incremented/decremented every transfer, or fixed
	Destination address register	Start address of transfer destination	Incremented/decremented every transfer, or fixed
	Holds block size	Block size	Fixed
	Block size counter	Block size	Decremented every transfer; ETCRH value copied when count reaches H'00
	Block transfer counter	Number of block transfers	Decremented every block transfer; transfer ends when count reaches H'0000

MARA and MARB specify the start addresses of the transfer source and transfer destination, respectively, as 24 bits. MAR can be incremented or decremented by 1 or 2 each time a byte or word is transferred, or can be fixed. Incrementing, decrementing, or holding a fixed value can be set separately for MARA and MARB. Whether a block is to be designated for MARA or for MARB is specified by the BLKDIR bit in DMACRA.

To specify the number of transfers, if M is the size of one block (where M = 1 to 256) and N transfers are to be performed (where N = 1 to 65,536), M is set in both ETCRAH and ETCRAL, and N in ETCRB.

Figure 7.13 illustrates operation in block transfer mode when MARB is designated as a block area.



Legend

Address $T_A = L_A$

Address $T_B = L_B$

Address $B_A = L_A + SAIDE \cdot (-1)^{SAID} \cdot (2^{DTSZ} \cdot (M \cdot N - 1))$

Address $B_B = L_B + DAIDE \cdot (-1)^{DAID} \cdot (2^{DTSZ} \cdot (N - 1))$

Where : L_A = Value set in MARA

L_B = Value set in MARB

N = Value set in ETCRB

M = Value set in ETCRAH and ETCRAL

Figure 7.13 Operation in Block Transfer Mode (BLKDIR = 0)

Figure 7.14 illustrates operation in block transfer mode when MARA is designated as a block area.

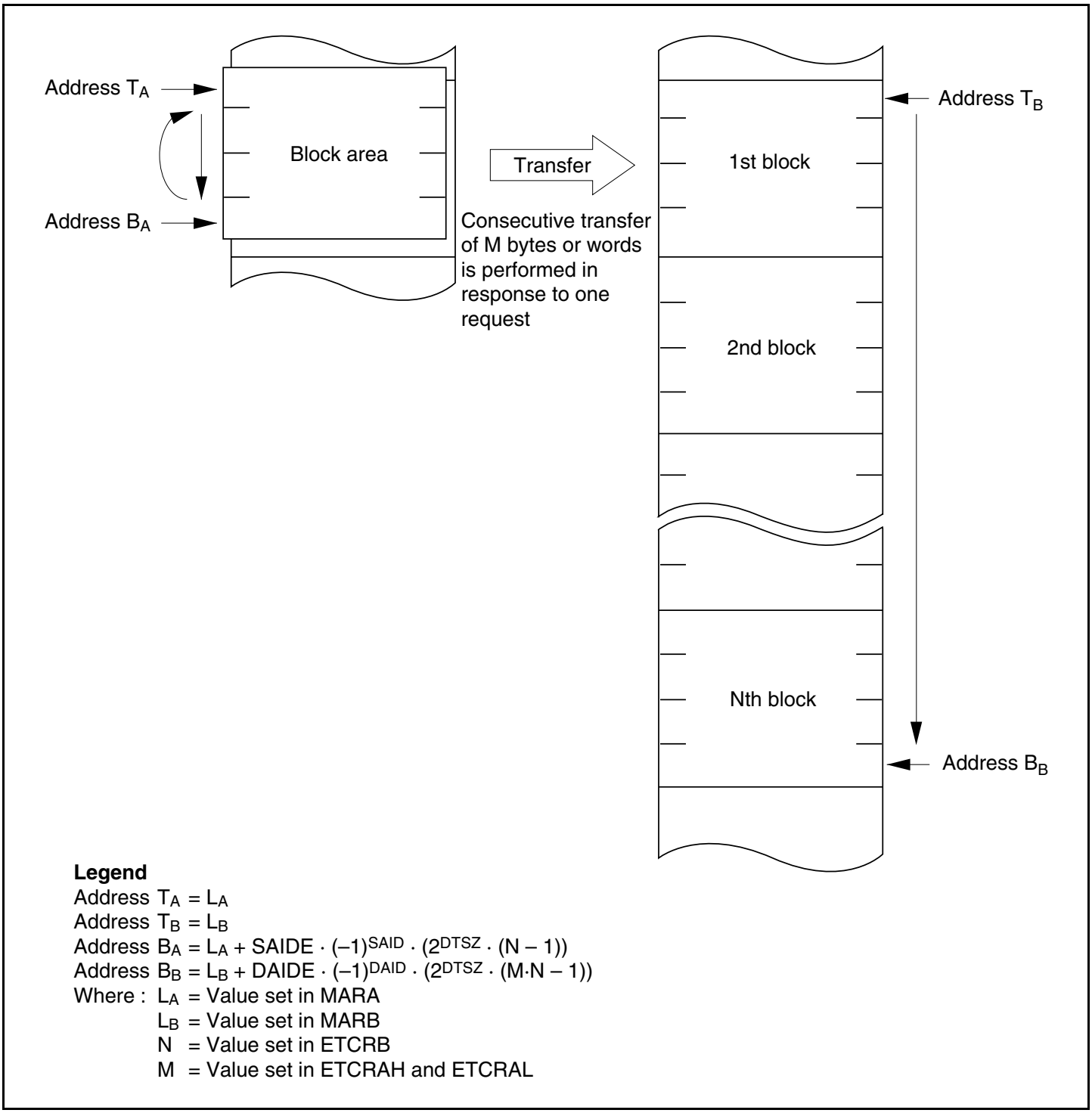


Figure 7.14 Operation in Block Transfer Mode (BLKDIR = 1)

ETCRAL is decremented by 1 each time a byte or word transfer is performed. In response to a single transfer request, burst transfer is performed until the value in ETCRAL reaches H'00. ETCRAL is then loaded with the value in ETCRAH. At this time, the value in the MAR register for which a block designation has been given by the BLKDIR bit in DMACRA is restored in accordance with the DTSZ, SAID/DAID, and SAIDE/DAIDE bits in DMACR.

ETCRB is decremented by 1 after every block transfer, and when the count reaches H'0000 the DTE bit in DMABCRL is cleared and transfer ends. If the DTIE bit in DMABCRL is set to 1 at this point, an interrupt request is sent to the CPU or DTC.

Figure 7.15 shows the operation flow in block transfer mode.

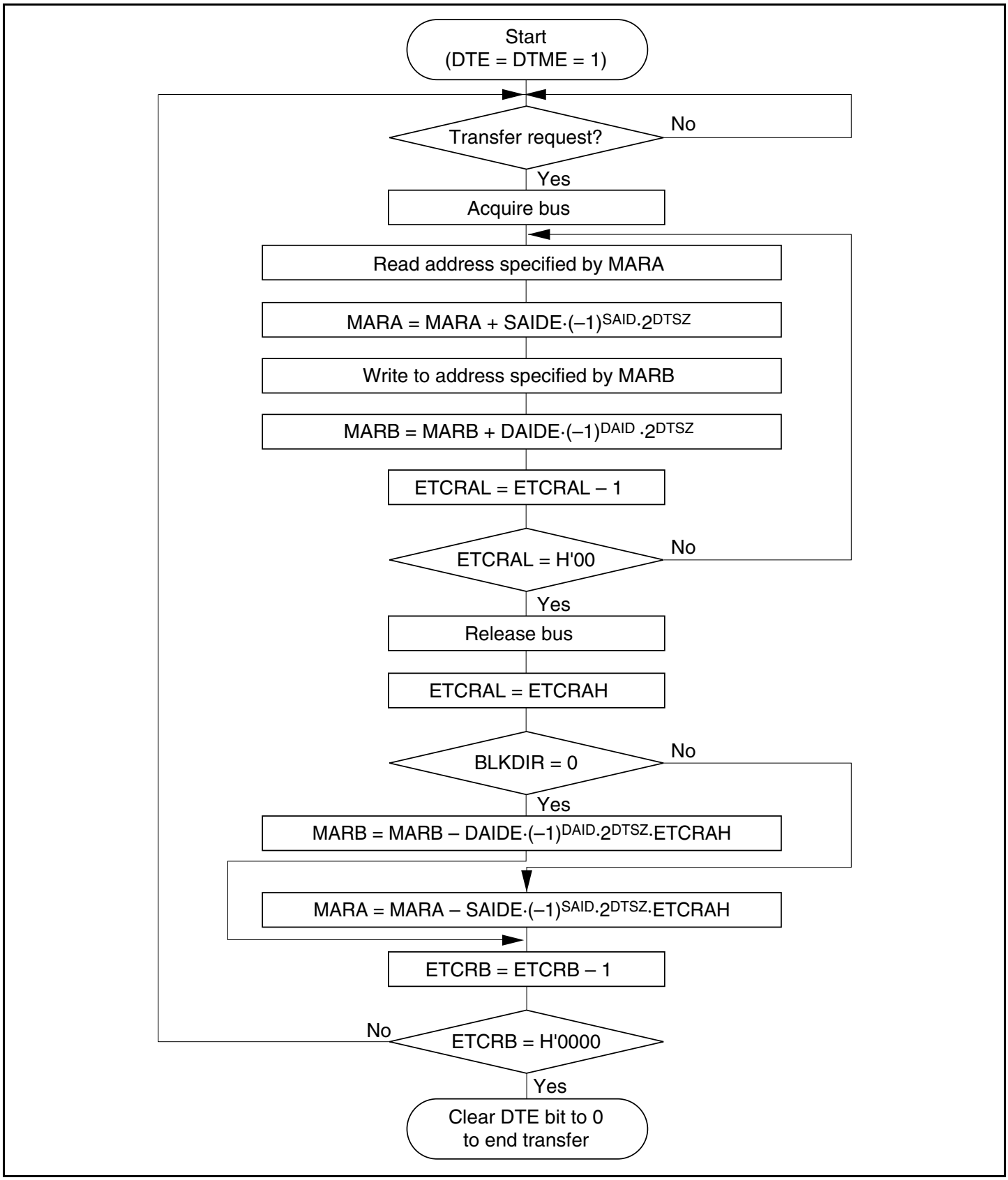
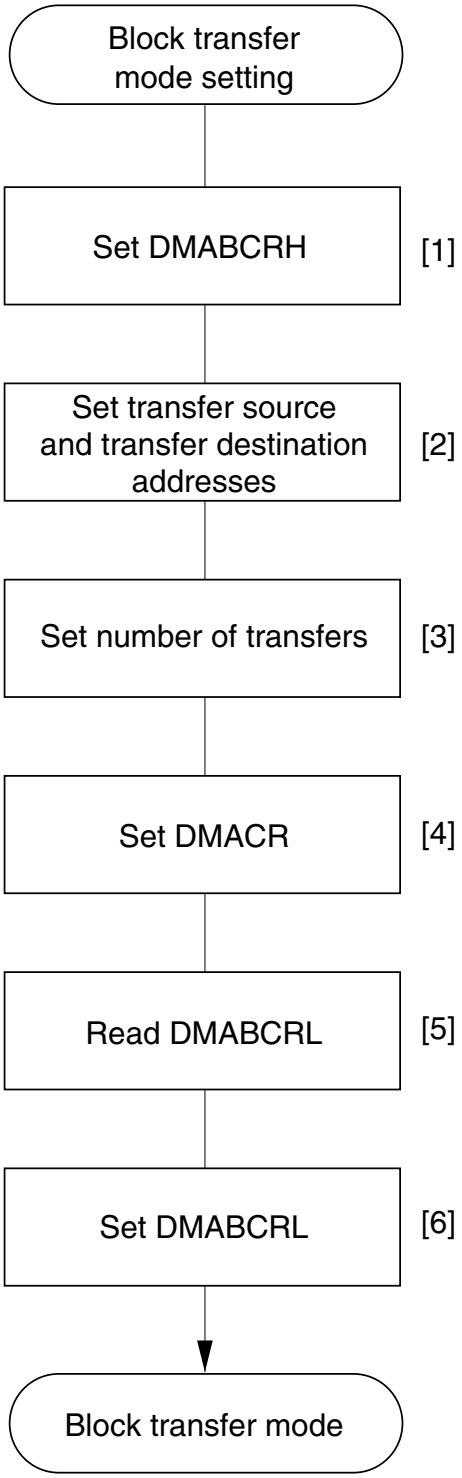


Figure 7.15 Operation Flow in Block Transfer Mode

Transfer requests (activation sources) consist of external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts.

Figure 7.16 shows an example of the setting procedure for block transfer mode.



[1] Set each bit in DMABCRH.
- Set the FAE bit to 1 to select full address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address in MARA, and the transfer destination address in MARB.

[3] Set the block size in both ETCRAH and ETCRAL. Set the number of transfers in ETCRB.

[4] Set each bit in DMACRA and DMACRB.
- Set the transfer data size with the DTSZ bit.
- Specify whether MARA is to be incremented, decremented, or fixed, with the SAID and SAIDE bits.
- Set the BLKE bit to 1 to select block transfer mode.
- Specify whether the transfer source or the transfer destination is a block area with the BLKDIR bit.
- Specify whether MARB is to be incremented, decremented, or fixed, with the DAID and DAIDE bits.
- Select the activation source with bits DTF3 to DTF0.

[5] Read DTE = 0 and DTME = 0 in DMABCRL.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts to the CPU with the DTIE bit.
- Set both the DTME bit and the DTE bit to 1 to enable transfer.

Figure 7.16 Example of Block Transfer Mode Setting Procedure

7.5.8 Basic Bus Cycles

An example of the basic DMAC bus cycle timing is shown in figure 7.17. In this example, word-size transfer is performed from 16-bit, 2-state access space to 8-bit, 3-state access space. When the bus is transferred from the CPU to the DMAC, a source address read and destination address write are performed. The bus is not released in response to another bus request, etc., between these read and write operations. As like CPU cycles, DMA cycles conform to the bus controller settings.

The address is not output to the external address bus in an access to on-chip memory or an internal I/O register.

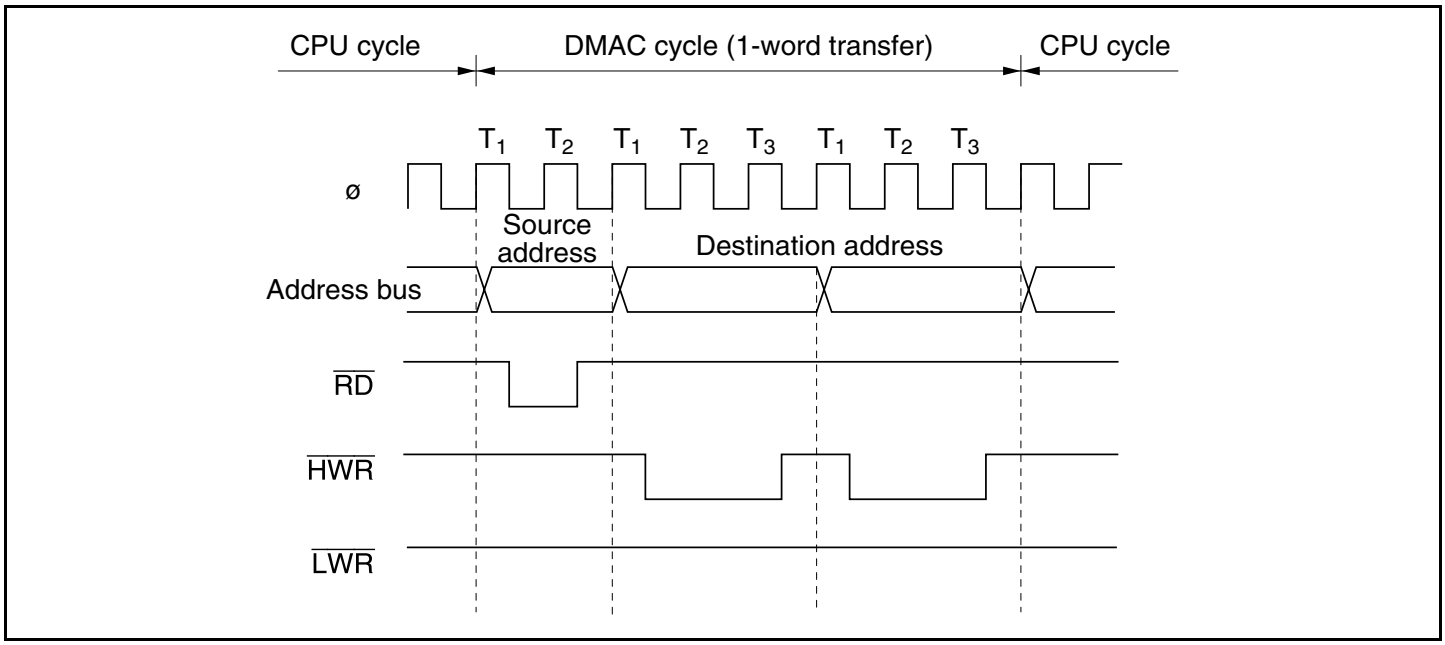
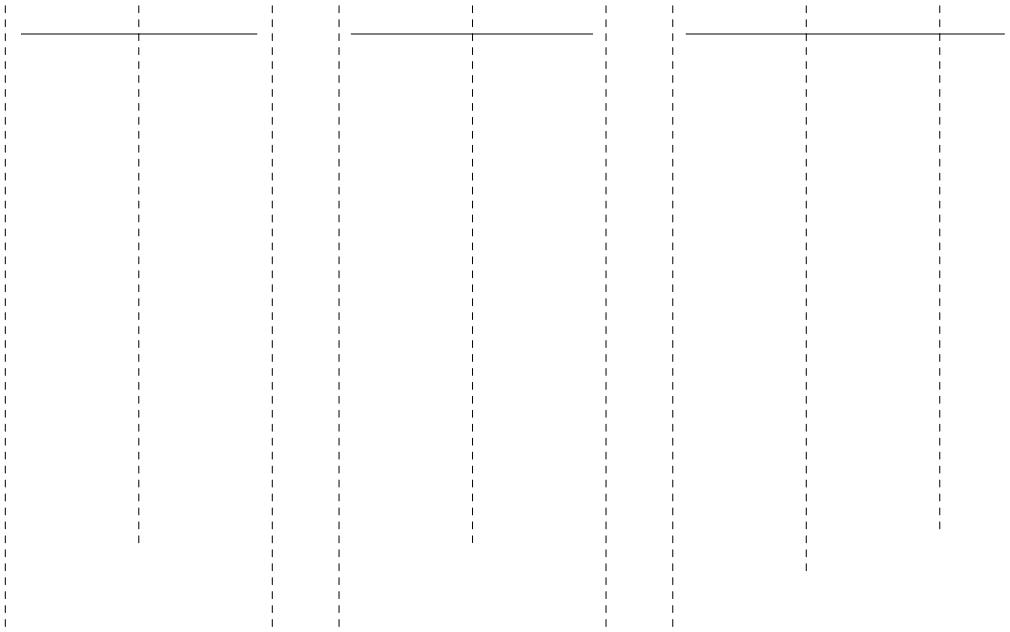


Figure 7.17 Example of DMA Transfer Bus Timing

7.5.9 DMA Bus Cycles (Dual Address Mode)

Short Address Mode: Figure 7.18 shows a transfer example in which TEND output is enabled and byte-size short address mode transfer (sequential/idle/repeat mode) is performed from external 8-bit, 2-state access space to internal I/O space.



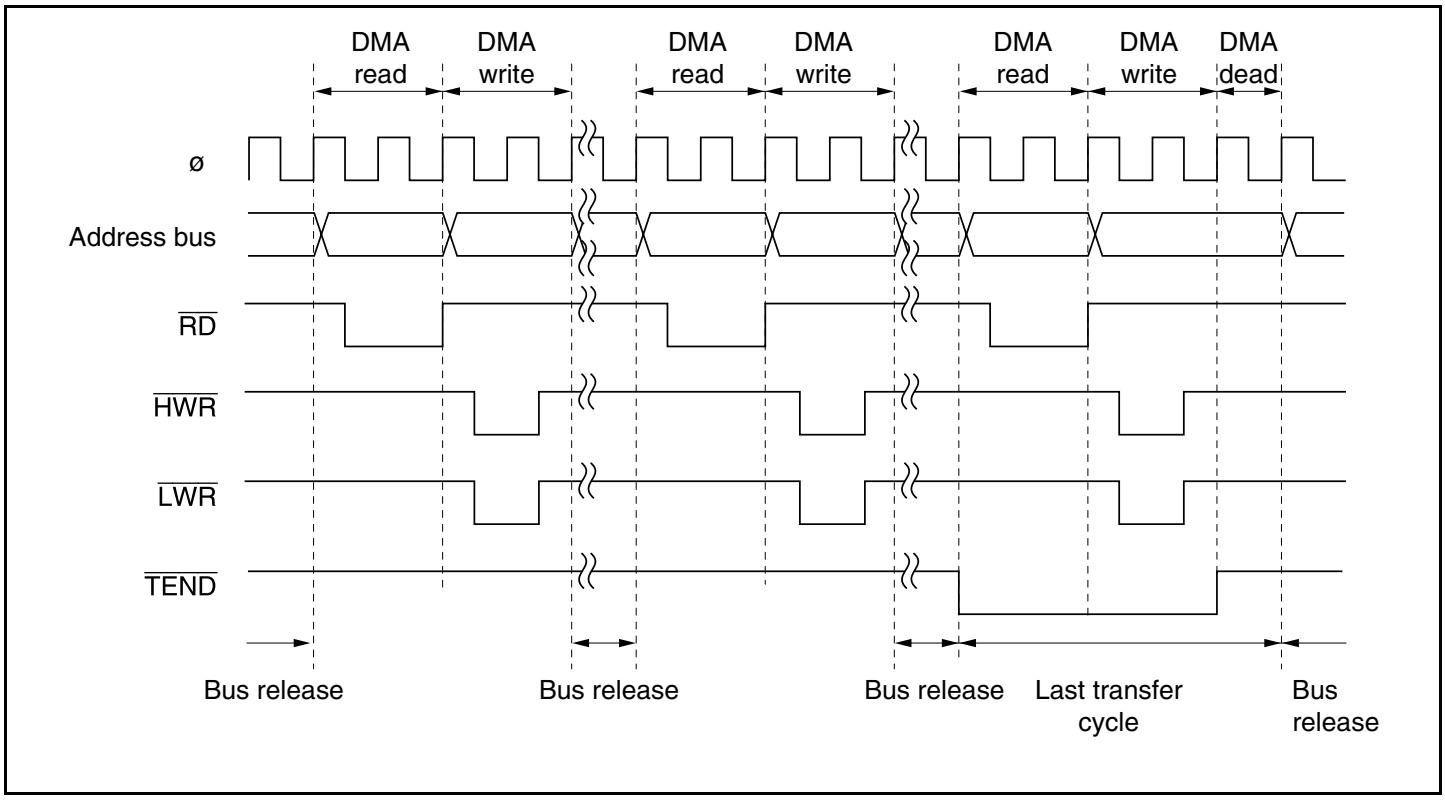


Figure 7.19 Example of Full Address Mode Transfer (Cycle Steal)

A byte or word transfer is performed for a single transfer request, and after the transfer, the bus is released. While the bus is released, one bus cycle is executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

Full Address Mode (Burst Mode): Figure 7.20 shows a transfer example in which TEND output is enabled and word-size full address mode transfer (burst mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

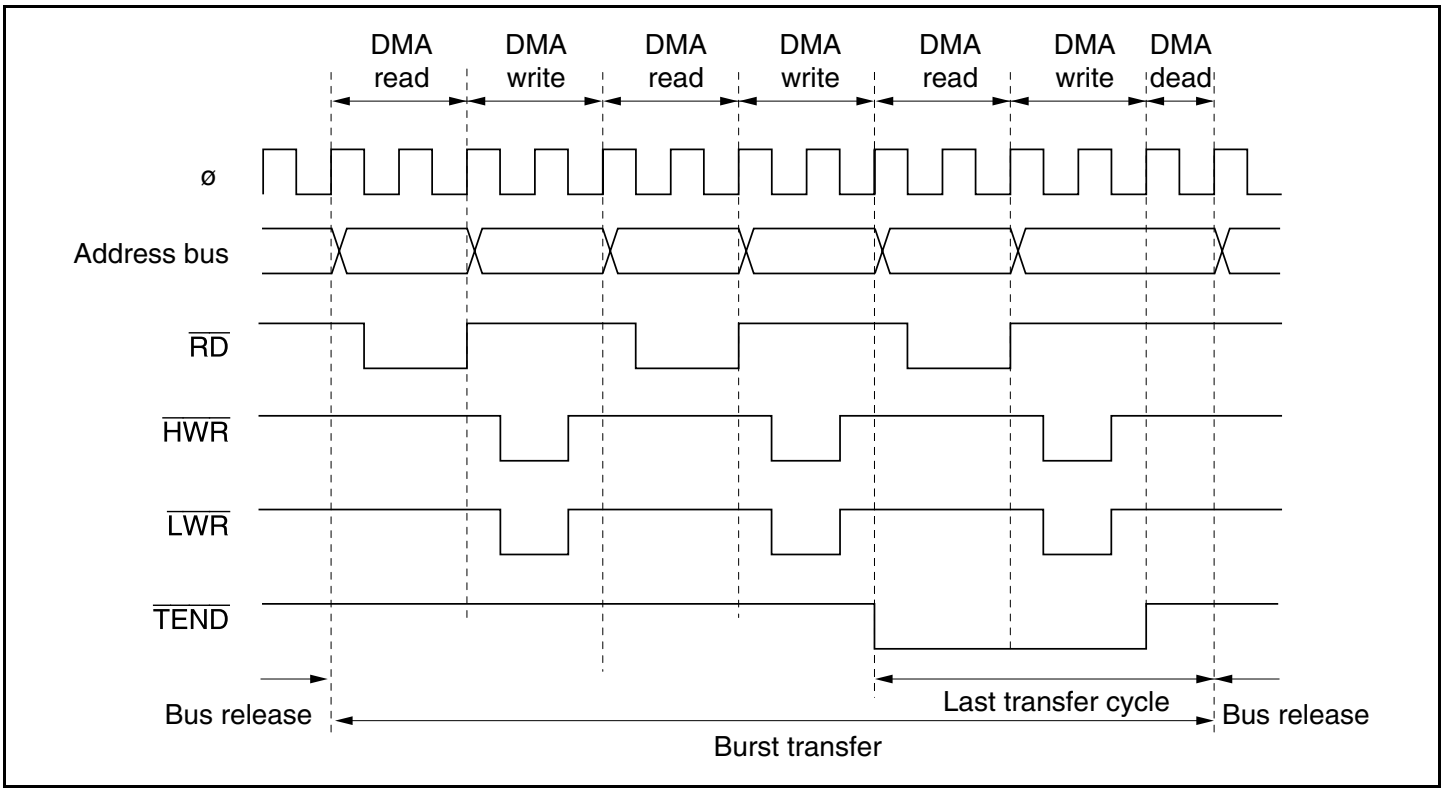


Figure 7.20 Example of Full Address Mode Transfer (Burst Mode)

In burst mode, one-byte or one-word transfers are executed consecutively until transfer ends.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

If a request from another higher-priority channel is generated after burst transfer starts, that channel has to wait until the burst transfer ends.

If an NMI interrupt is generated while a channel designated for burst transfer is in the transfer enabled state, the DTME bit in DMABCRL is cleared and the channel is placed in the transfer disabled state. If burst transfer has already been activated inside the DMAC, the bus is released on completion of a one-byte or one-word transfer within the burst transfer, and burst transfer is suspended. If the last transfer cycle of the burst transfer has already been activated inside the DMAC, execution continues to the end of the transfer even if the DTME bit is cleared.

Full Address Mode (Block Transfer Mode): Figure 7.21 shows a transfer example in which TEND output is enabled and word-size full address mode transfer (block transfer mode) is performed from internal 16-bit, 1-state access space to external 16-bit, 2-state access space.

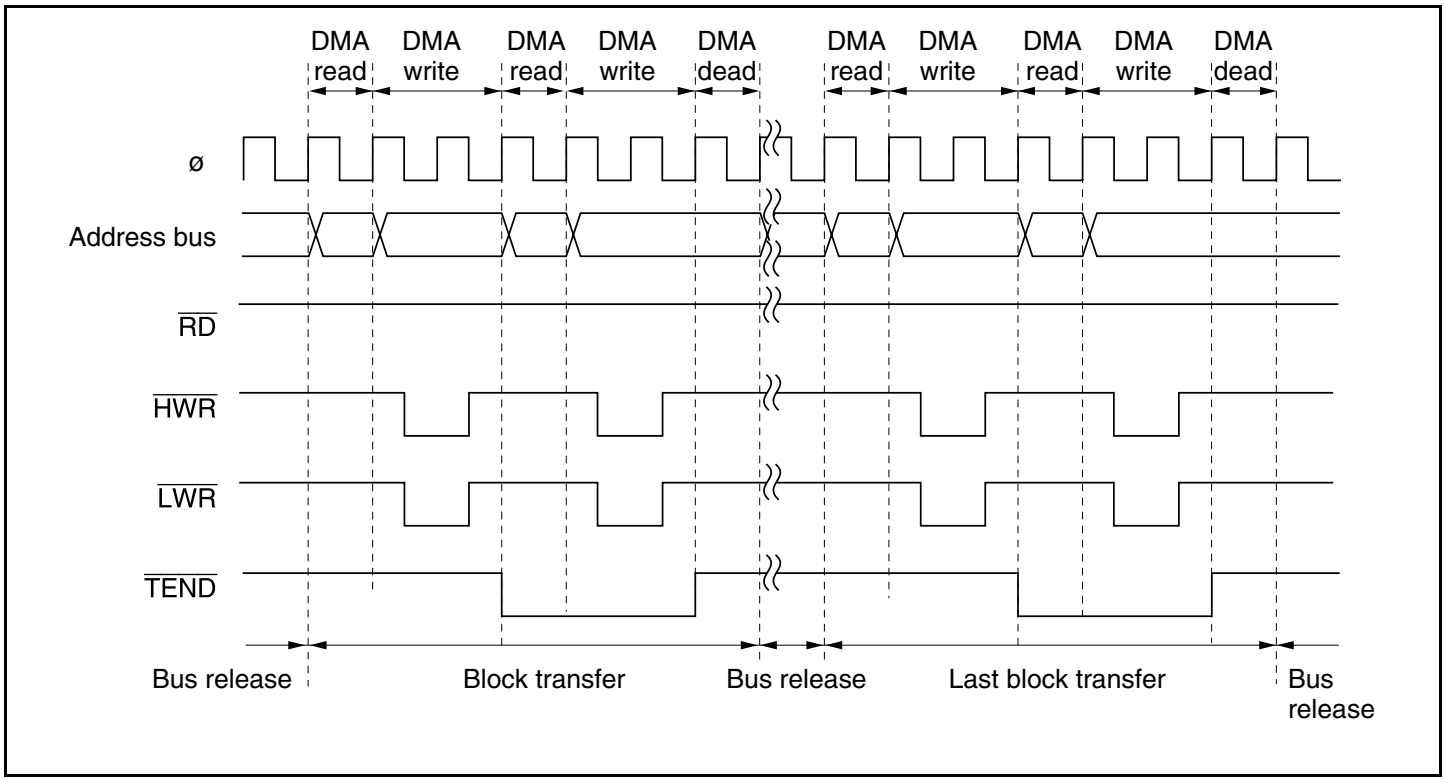


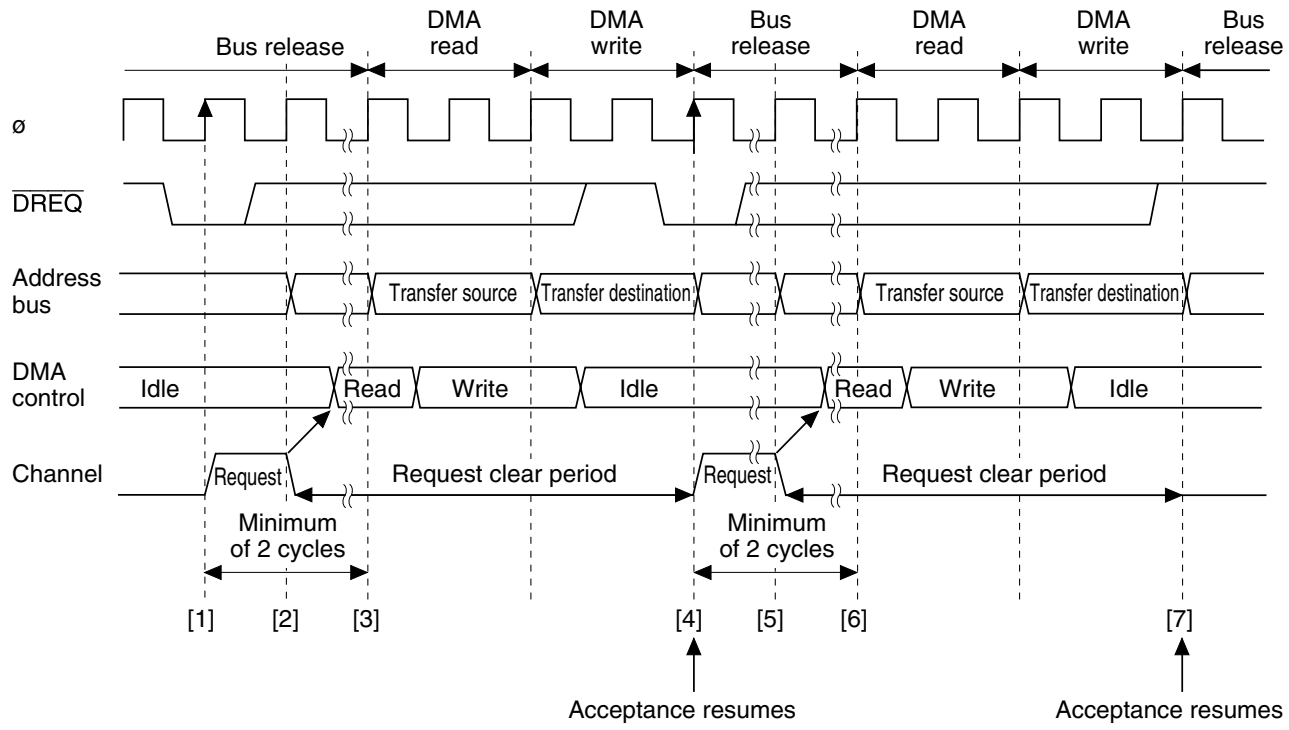
Figure 7.21 Example of Full Address Mode Transfer (Block Transfer Mode)

A one-block transfer is performed for a single transfer request, and after the transfer the bus is released. While the bus is released, one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle of each block (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle. Even if an NMI interrupt is generated during data transfer, block transfer operation is not affected until data transfer for one block has ended.

DREQ Pin Falling Edge Activation Timing: Set the DTA bit in DMABCRH to 1 for the channel for which the DREQ pin is selected.

Figure 7.22 shows an example of normal mode transfer activated by the DREQ pin falling edge.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of ø starts.

[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the write cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.)

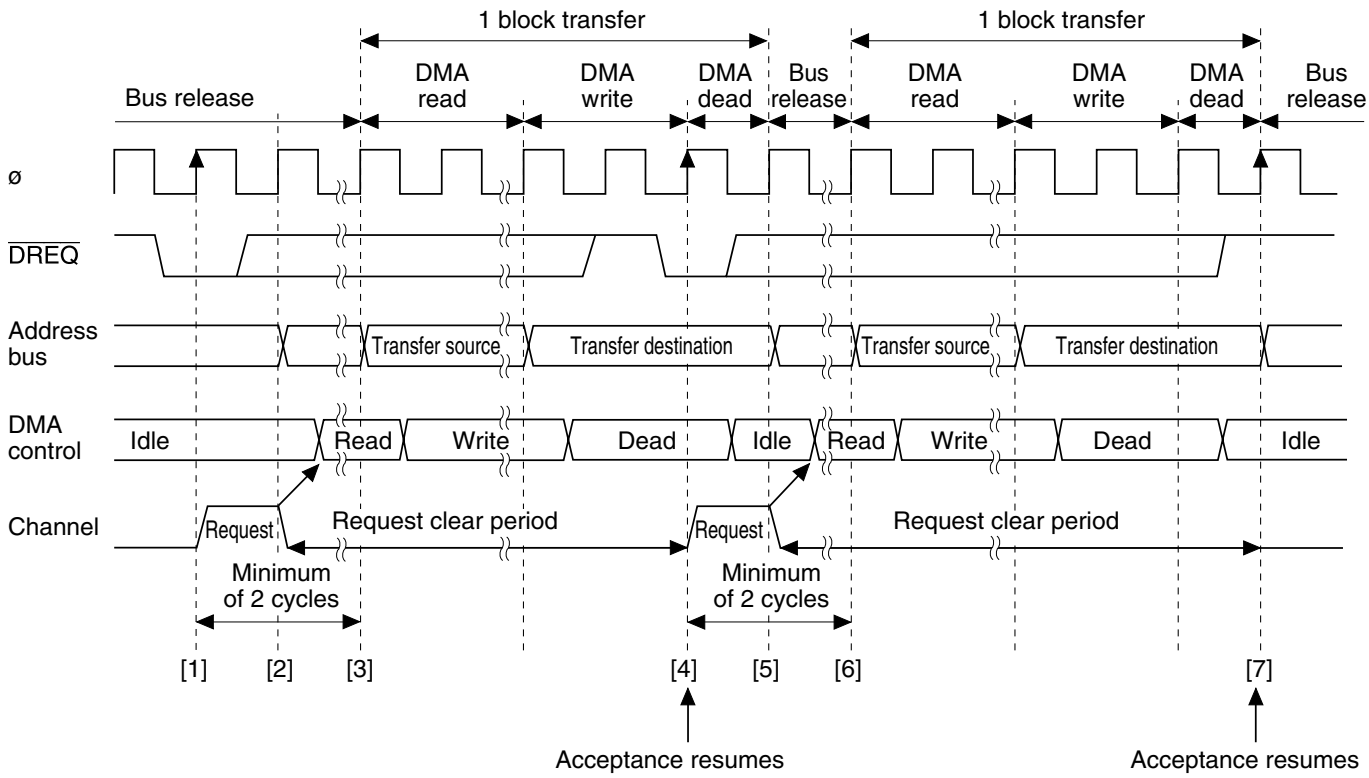
Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.22 Example of DREQ Pin Falling Edge Activated Normal Mode Transfer

DREQ pin sampling is performed every cycle, with the rising edge of the next ø cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the DREQ pin low level is sampled while acceptance by means of the DREQ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and DREQ pin high level sampling for edge detection is started. If DREQ pin high level sampling has been completed by the time the DMA write cycle ends, acceptance resumes after the end of the write cycle, DREQ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

Figure 7.23 shows an example of block transfer mode transfer activated by the DREQ pin falling edge.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of ø starts.

[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the dead cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

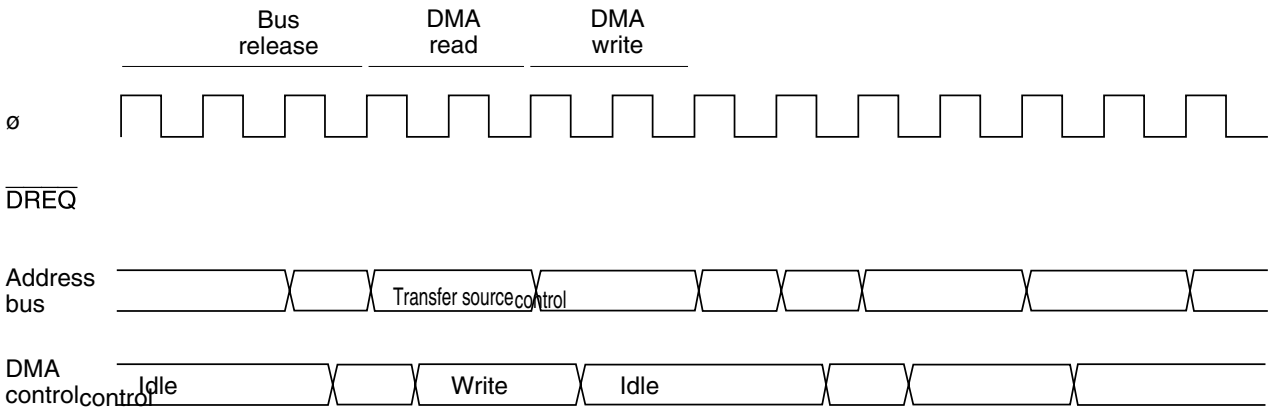
Figure 7.23 Example of DREQ Pin Falling Edge Activated Block Transfer Mode Transfer

DREQ pin sampling is performed every cycle, with the rising edge of the next ø cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

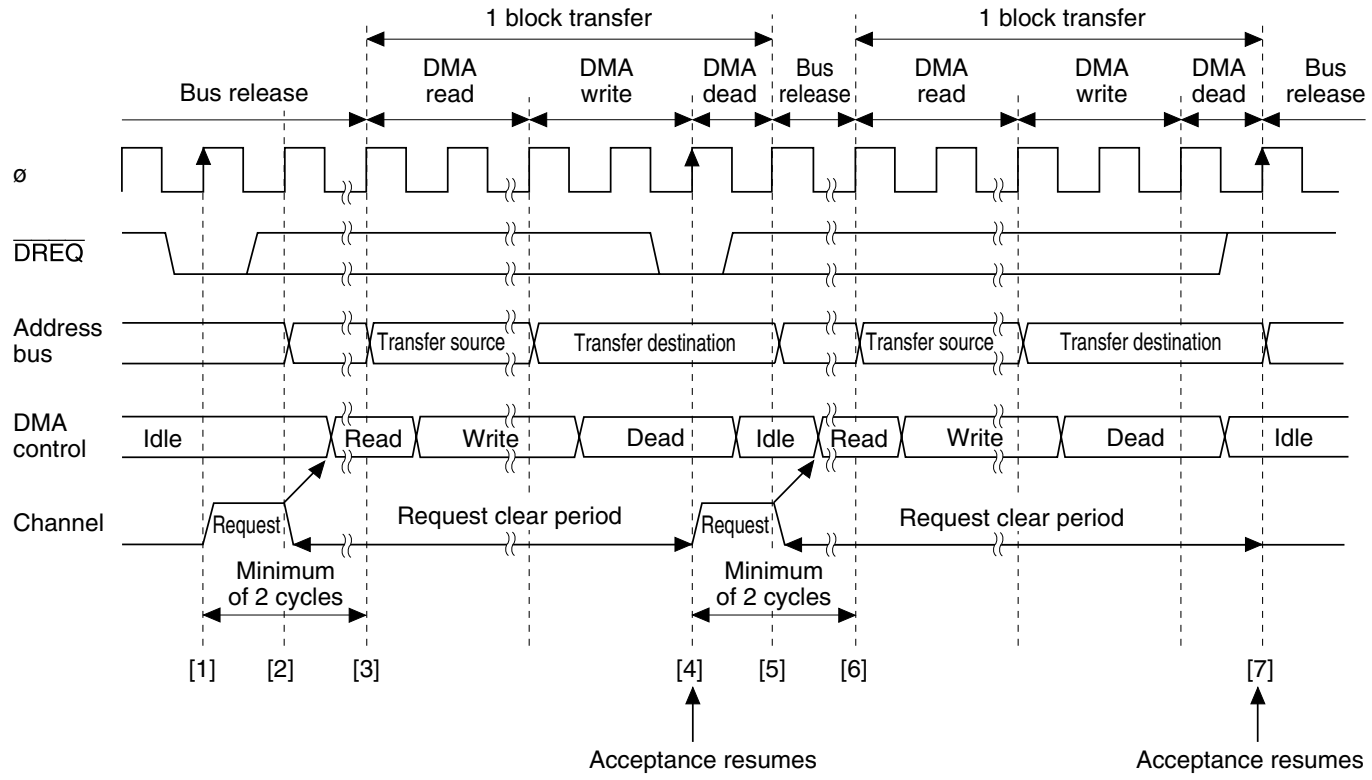
When the DREQ pin low level is sampled while acceptance by means of the DREQ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and DREQ pin high level sampling for edge detection is started. If DREQ pin high level sampling has been completed by the time the DMA dead cycle ends, acceptance resumes after the end of the dead cycle, DREQ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

DREQ Pin Low Level Activation Timing (Normal Mode): Set the DTA bit in DMABCRH to 1 for the channel for which the DREQ pin is selected.

Figure 7.24 shows an example of normal mode transfer activated by the DREQ pin low level.


Bus release
DMA read
DMA write
ø
DREQ
Address bus
Transfer source control
DMA control
Idle
Write
Idle

[1] [2] [3] [4] [5] [6]



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] The DMA cycle is started.

[4] [7] Acceptance is resumed after the dead cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.25 Example of DREQ Pin Low Level Activated Block Transfer Mode Transfer

DREQ pin sampling is performed every cycle, with the rising edge of the next ø cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the DREQ pin low level is sampled while acceptance by means of the DREQ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. After the end of the dead cycle, acceptance resumes, DREQ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

7.5.10 DMA Bus Cycles (Single Address Mode)

Single Address Mode (Read): Figure 7.26 shows a transfer example in which TEND output is enabled and byte-size single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.

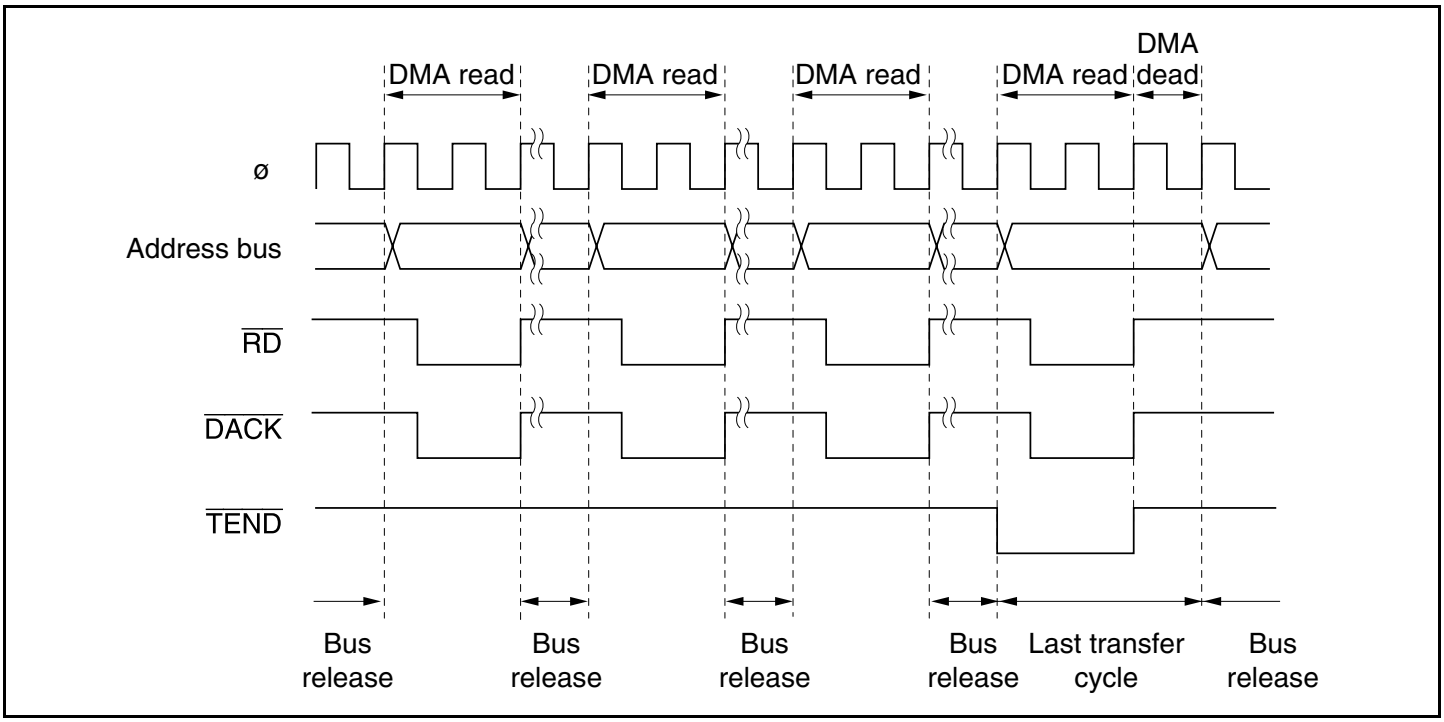


Figure 7.26 Example of Single Address Mode Transfer (Byte Read)

Figure 7.27 shows a transfer example in which TEND output is enabled and word-size single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.

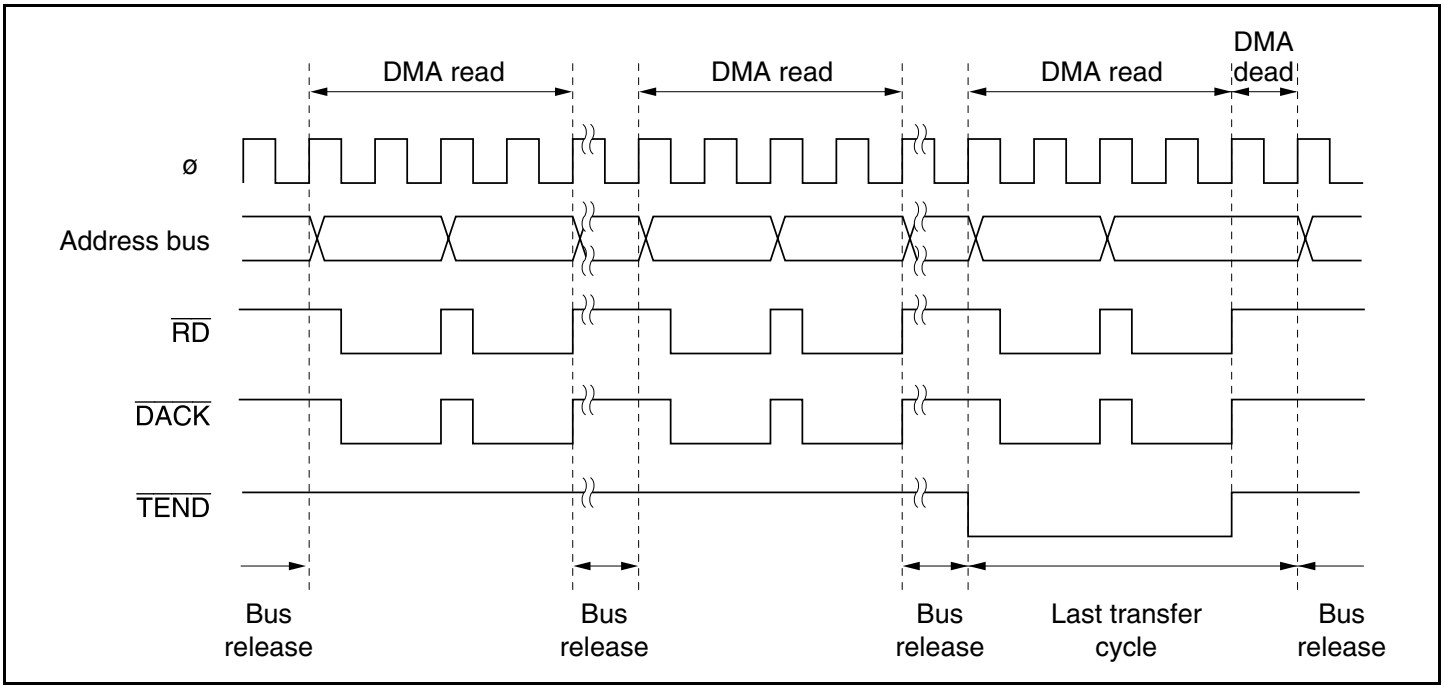


Figure 7.27 Example of Single Address Mode (Word Read) Transfer

A byte or word transfer is performed for a single transfer request, and after the transfer, the bus is released. While the bus is released, one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

Single Address Mode (Write): Figure 7.28 shows a transfer example in which TEND output is enabled and byte-size single address mode transfer (write) is performed from an external device to external 8-bit, 2-state access space.

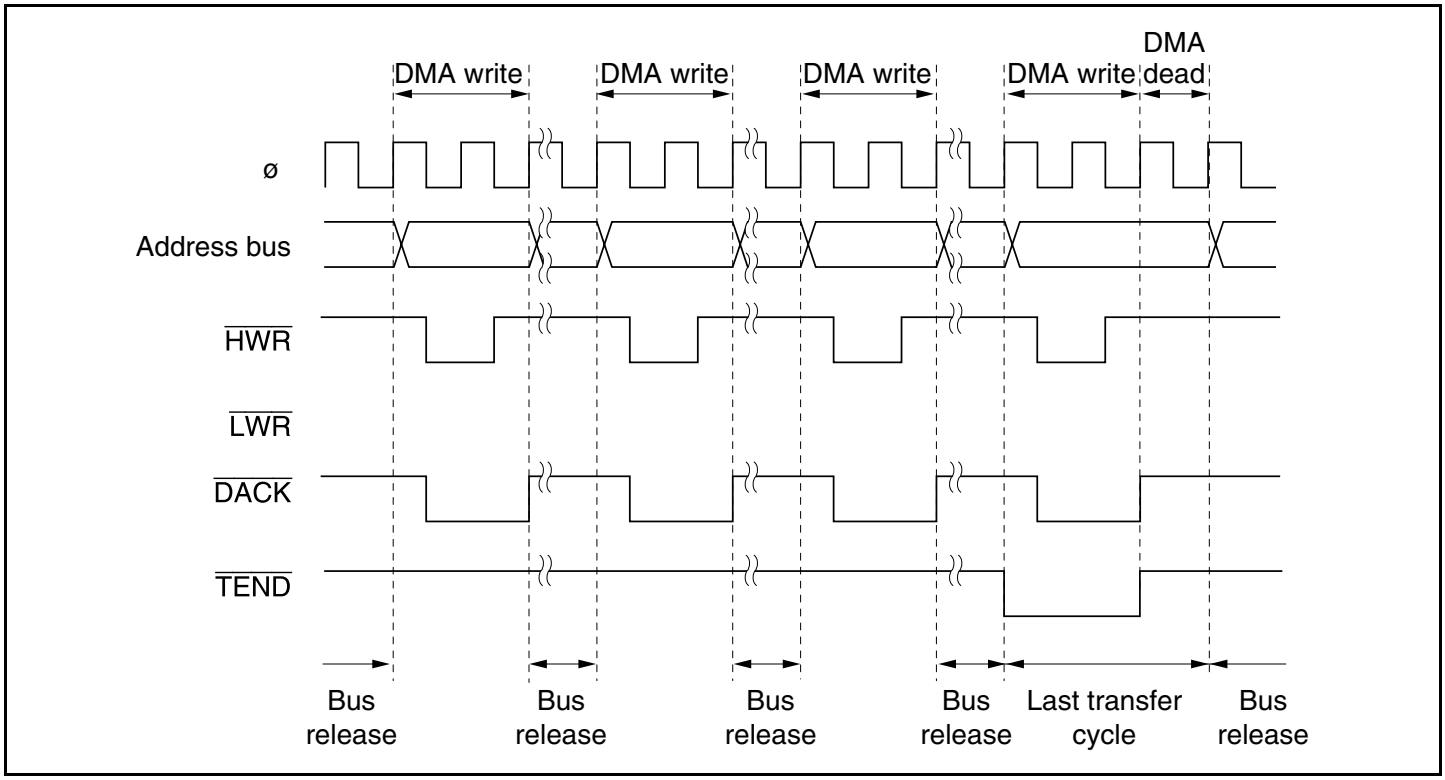


Figure 7.28 Example of Single Address Mode Transfer (Byte Write)

Figure 7.29 shows a transfer example in which TEND output is enabled and word-size single address mode transfer (write) is performed from an external device to external 8-bit, 2-state access space.

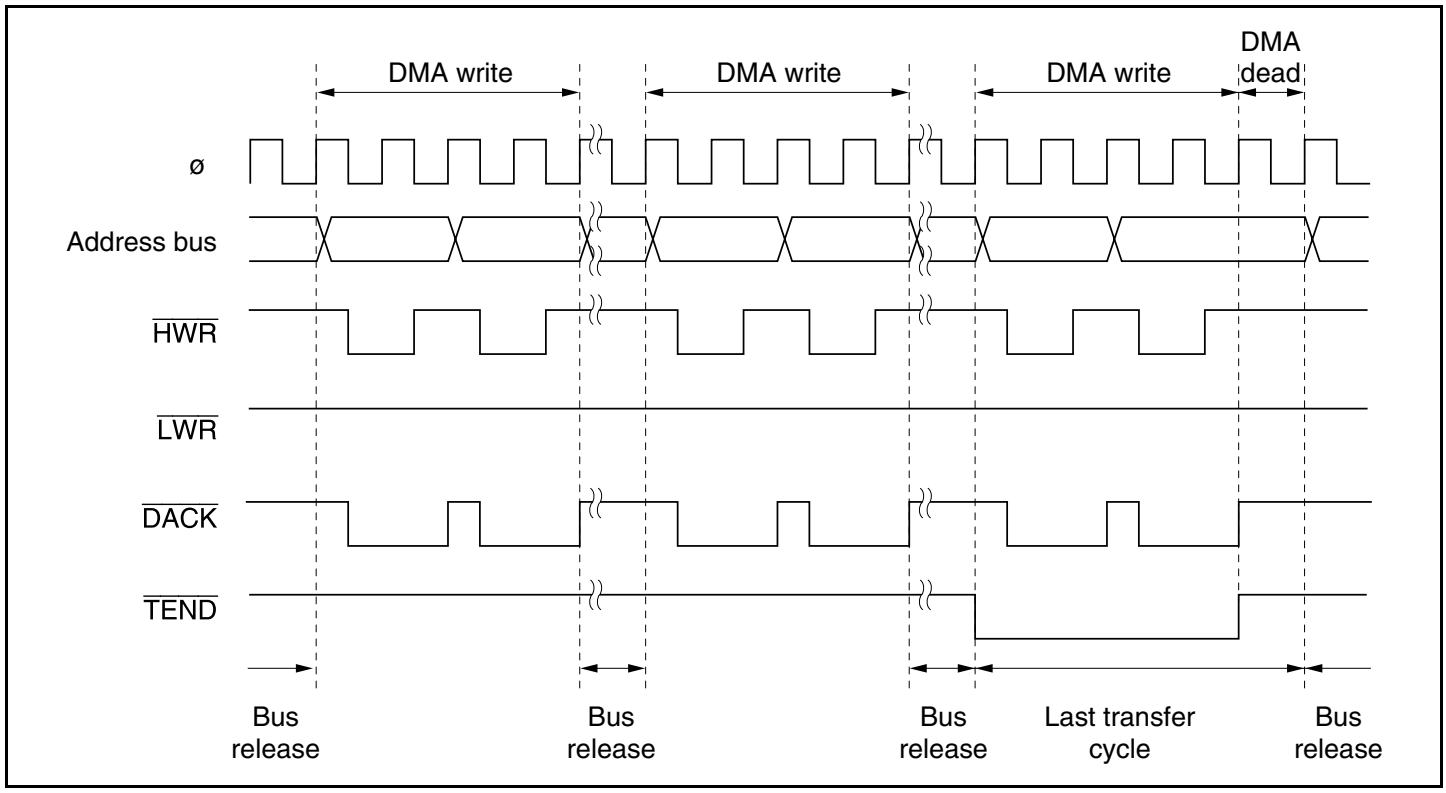


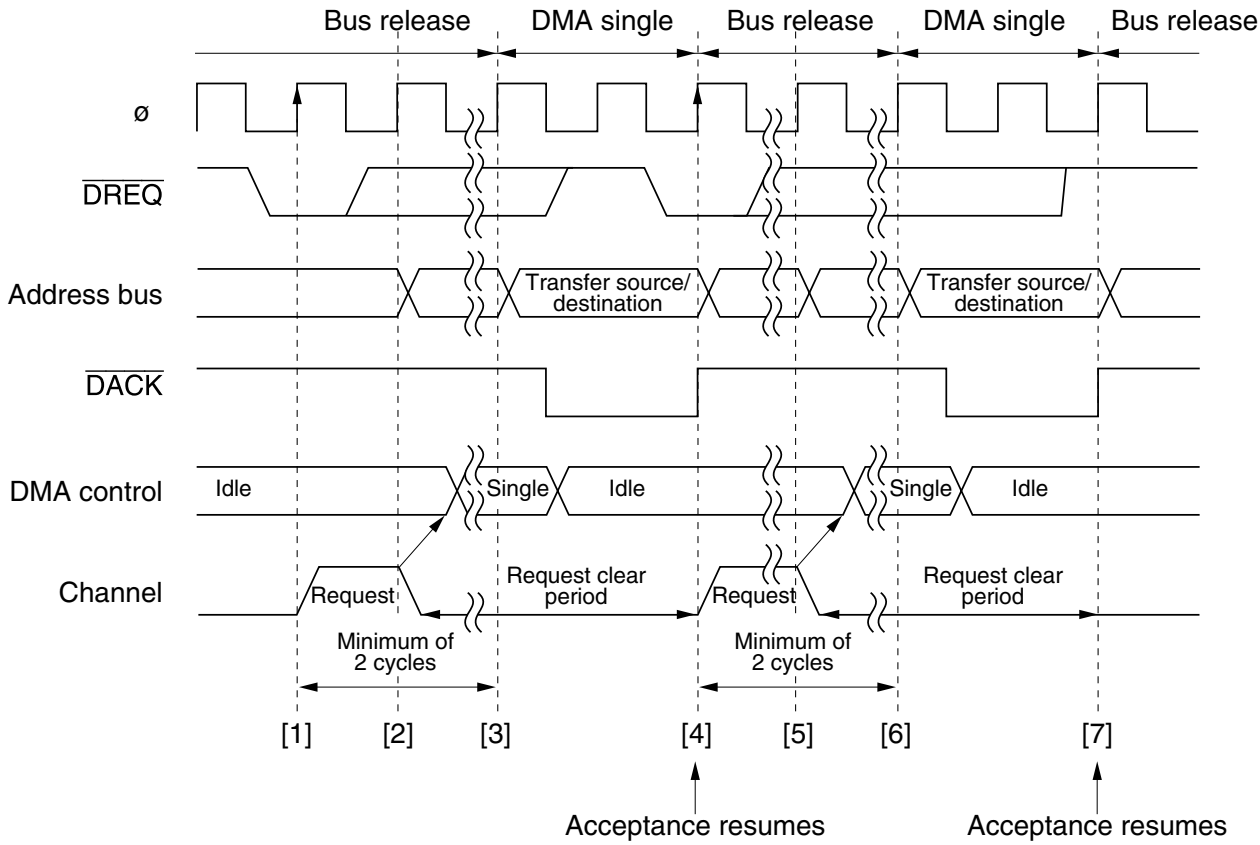
Figure 7.29 Example of Single Address Mode Transfer (Word Write)

A byte or word transfer is performed for a single transfer request, and after the transfer, the bus is released. While the bus is released, one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

DREQ Pin Falling Edge Activation Timing: Set the DTA bit in DMABCRH to 1 for the channel for which the DREQ pin is selected.

Figure 7.30 shows an example of single address mode transfer activated by the DREQ pin falling edge.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of ø starts.

[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the single cycle is completed. (As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.)

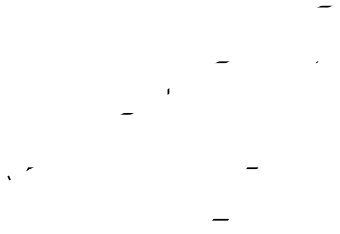
Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.30 Example of DREQ Pin Falling Edge Activated Single Address Mode Transfer

DREQ pin sampling is performed every cycle, with the rising edge of the next ø cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the DREQ pin low level is sampled while acceptance by means of the DREQ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and DREQ pin high level sampling for edge detection is started. If DREQ pin high level sampling has been completed by the time the DMA single cycle ends, acceptance resumes after the end of the single cycle, DREQ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

DREQ Pin Low Level Activation Timing: Set the DTA bit in DMABCRH to 1 for the channel for which the DREQ pin is selected.



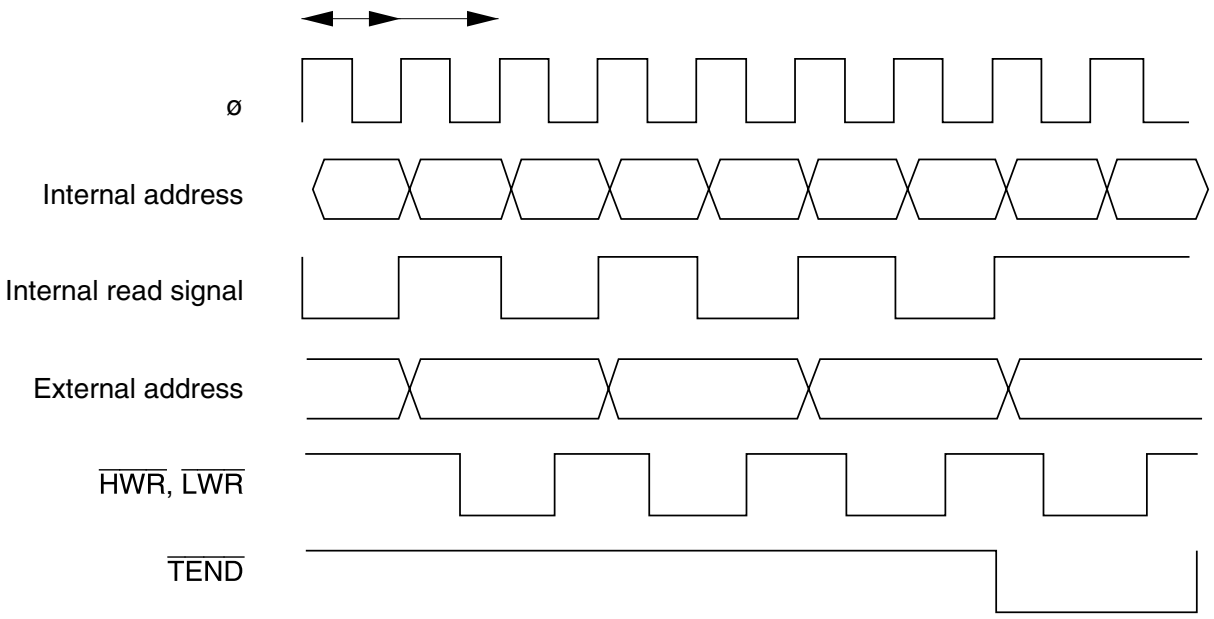
7.5.11 Write Data Buffer Function

DMAC internal-to-external dual address transfers and single address transfers can be executed at high speed using the write data buffer function, enabling system throughput to be improved.

When the WDBE bit of BCR in the bus controller is set to 1, enabling the write data buffer function, dual address transfer external write cycles or single address transfers and internal accesses (on-chip memory or internal I/O registers) are executed in parallel. Internal accesses are independent of the bus master, and DMAC dead cycles are regarded as internal accesses.

A low level can always be output from the TEND pin if the bus cycle in which a low level is to be output from the TEND pin is an external bus cycle. However, a low level is not output from the TEND pin if the bus cycle in which a low level is to be output from the TEND pin is an internal bus cycle, and an external write cycle is executed in parallel with this cycle.

Figure 7.32 shows an example of burst mode transfer from on-chip RAM to external memory using the write data buffer function.



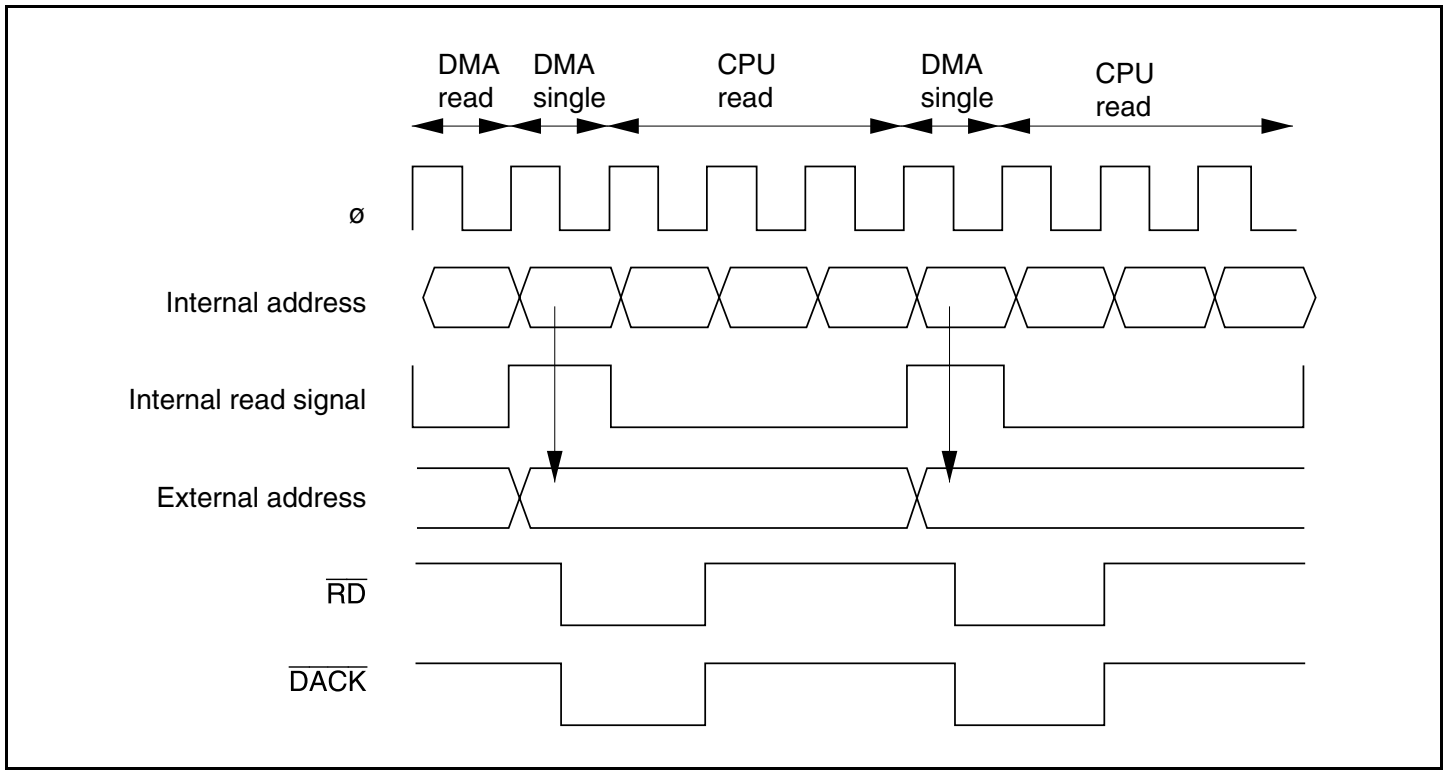


Figure 7.33 Example of Single Address Transfer Using Write Data Buffer Function

When the write data buffer function is activated, the DMAC recognizes that the bus cycle concerned has ended, and starts the next operation. Therefore, DREQ pin sampling is started one state after the start of the DMA write cycle or single address transfer.

7.5.12 Multi-Channel Operation

The DMAC channel priority order is: channel 0 > channel 1, and channel A > channel B. Table 7.11 summarizes the priority order for DMAC channels.

Table 7.11 DMAC Channel Priority Order

Short Address Mode	Full Address Mode	Priority
Channel 0A	Channel 0	High
Channel 0B		
Channel 1A	Channel 1	
Channel 1B		Low

If transfer requests are issued simultaneously for more than one channel, or if a transfer request for another channel is issued during a transfer, when the bus is released, the DMAC selects the highest-priority channel from among those issuing a request according to the priority order shown in table 7.11. During burst transfer, or when one block is being transferred in block transfer, the channel will not be changed until the end of the transfer. Figure 7.34 shows a transfer example in which transfer requests are issued simultaneously for channels 0A, 0B, and 1.

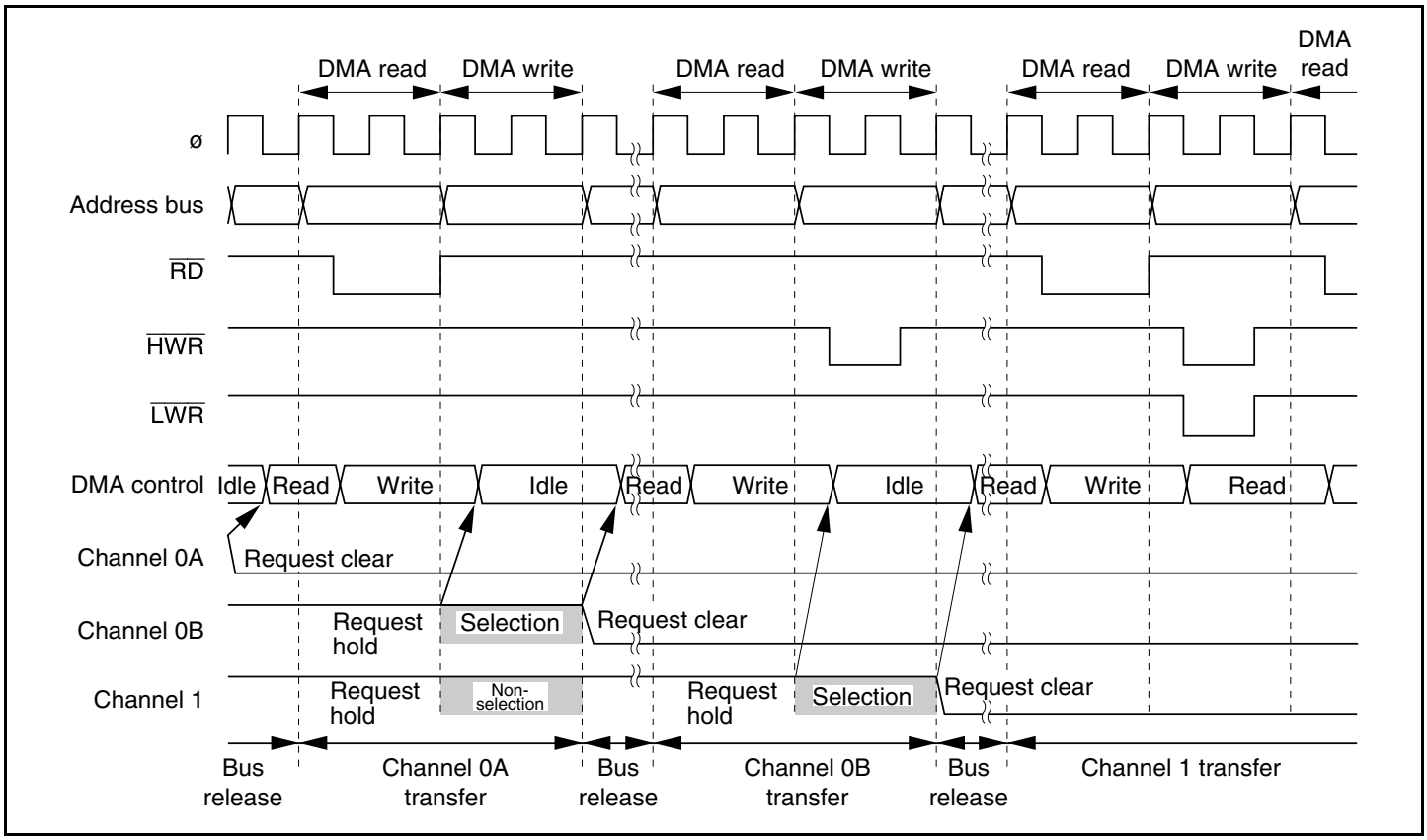


Figure 7.34 Example of Multi-Channel Transfer

7.5.13 Relation between DMAC and External Bus Requests, Refresh Cycles, and EXDMAC

When the DMAC accesses external space, contention with a refresh cycle, EXDMAC cycle, or external bus release cycle may arise. In this case, the bus controller will suspend the transfer and insert a refresh cycle, EXDMAC cycle, or external bus release cycle, in accordance with the external bus priority order, even if the DMAC is executing a burst transfer or block transfer. (An external access by the DTC or CPU, which has a lower priority than the DMAC, is not executed until the DMAC releases the external bus.)

When the DMAC transfer mode is dual address mode, the DMAC releases the external bus after an external write cycle. The external read cycle and external write cycle are inseparable, and so the bus cannot be released between these two cycles.

When the DMAC accesses internal space (on-chip memory or an internal I/O register), the DMAC cycle may be executed at the same time as a refresh cycle, EXDMAC cycle, or external bus release cycle.

7.5.14 DMAC and NMI Interrupts

When an NMI interrupt is requested, burst mode transfer in full address mode is interrupted. An NMI interrupt does not affect the operation of the DMAC in other modes.

In full address mode, transfer is enabled for a channel when both the DTE bit and DTME bit are set to 1. With burst mode setting, the DTME bit is cleared when an NMI interrupt is requested.

If the DTME bit is cleared during burst mode transfer, the DMAC discontinues transfer on completion of the 1-byte or 1-word transfer in progress, then releases the bus, which passes to the CPU.

The channel on which transfer was interrupted can be restarted by setting the DTME bit to 1 again. Figure 7.35 shows the procedure for continuing transfer when it has been interrupted by an NMI interrupt on a channel designated for burst mode transfer.

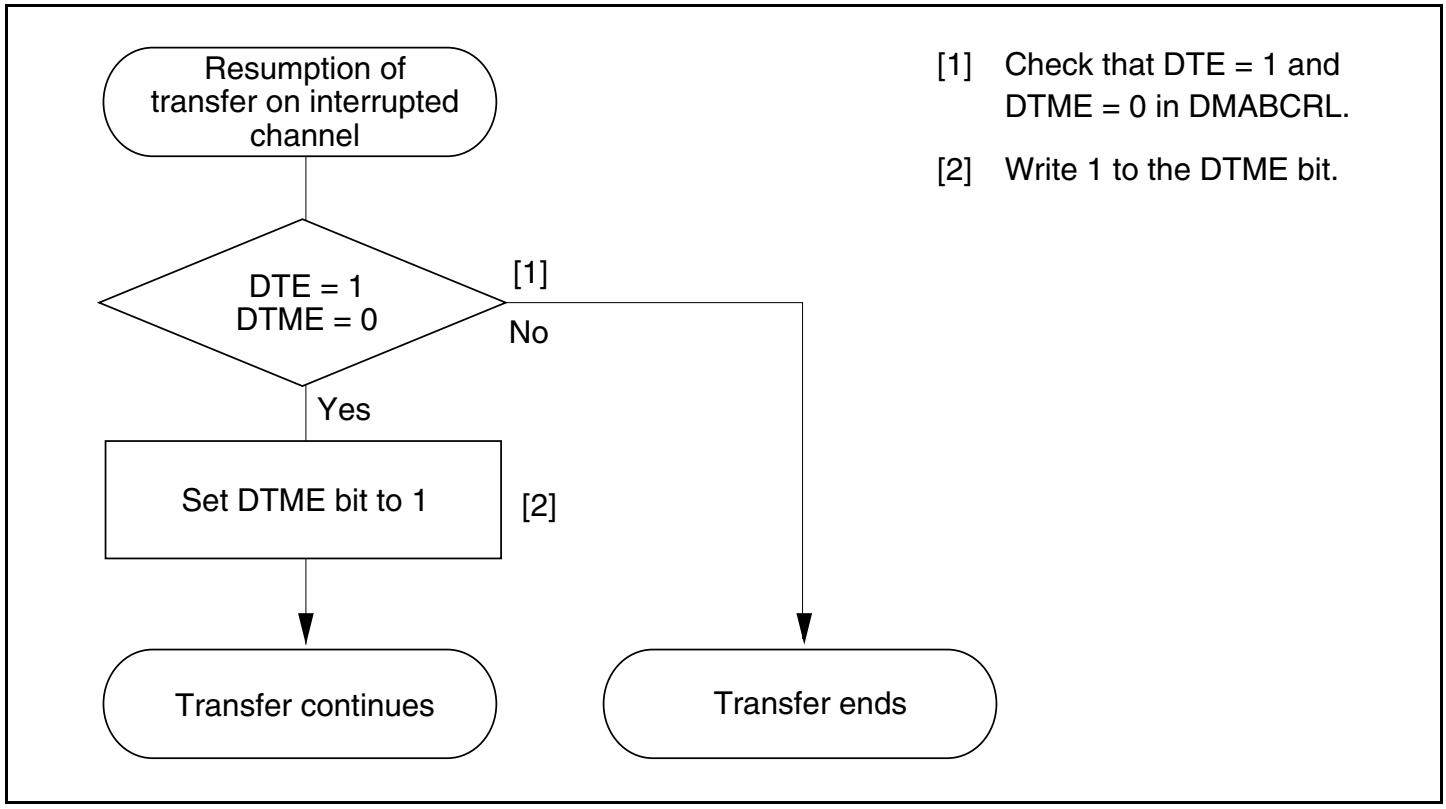
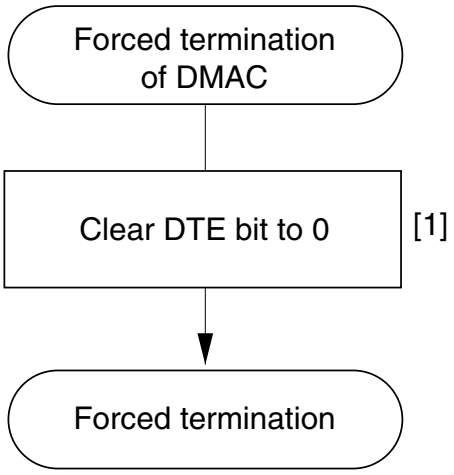


Figure 7.35 Example of Procedure for Continuing Transfer on Channel Interrupted by NMI Interrupt

7.5.15 Forced Termination of DMAC Operation

If the DTE bit in DMABCRL is cleared to 0 for the channel currently operating, the DMAC stops on completion of the 1-byte or 1-word transfer in progress. DMAC operation resumes when the DTE bit is set to 1 again. In full address mode, the same applies to the DTME bit in DMABCRL. Figure 7.36 shows the procedure for forcibly terminating DMAC operation by software.

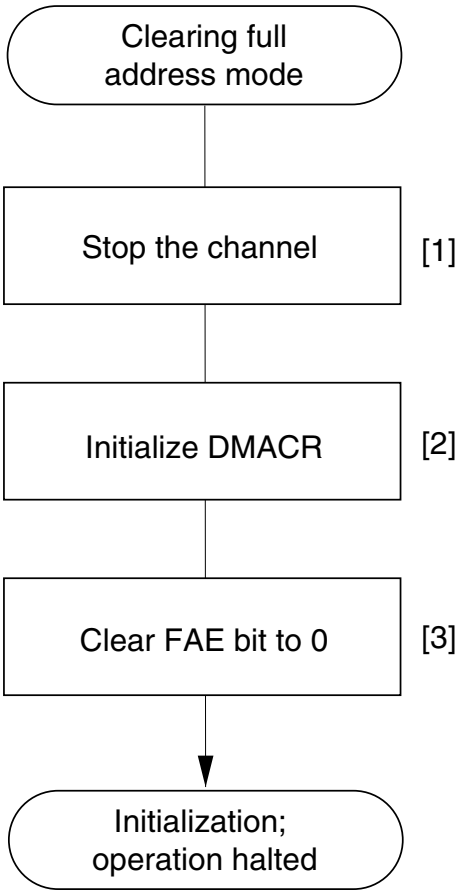


[1] Clear the DTE bit in DMABCRL to 0.
To prevent interrupt generation after forced termination of DMAC operation, clear the DTIE bit to 0 at the same time.

Figure 7.36 Example of Procedure for Forcibly Terminating DMAC Operation

7.5.16 Clearing Full Address Mode

Figure 7.37 shows the procedure for releasing and initializing a channel designated for full address mode. After full address mode has been cleared, the channel can be set to another transfer mode using the appropriate setting procedure.



[1] Clear both the DTE bit and DTME bit in DMABCRL to 0, or wait until the transfer ends and the DTE bit is cleared to 0, then clear the DTME bit to 0. Also clear the corresponding DTIE bit to 0 at the same time.

[2] Clear all bits in DMACRA and DMACRB to 0.

[3] Clear the FAE bit in DMABCRH to 0.

Figure 7.37 Example of Procedure for Clearing Full Address Mode

7.6 Interrupt Sources

The sources of interrupts generated by the DMAC are transfer end and transfer break. Table 7.12 shows the interrupt sources and their priority order.

Table 7.12 Interrupt Sources and Priority Order

Interrupt Name	Interrupt Source		Interrupt Priority Order
	Short Address Mode	Full Address Mode	
DMTEND0A	Interrupt due to end of transfer on channel 0A	Interrupt due to end of transfer on channel 0	High
DMTEND0B	Interrupt due to end of transfer on channel 0B	Interrupt due to break in transfer on channel 0	
DMTEND1A	Interrupt due to end of transfer on channel 1A	Interrupt due to end of transfer on channel 1	
DMTEND1B	Interrupt due to end of transfer on channel 1B	Interrupt due to break in transfer on channel 1	Low

Enabling or disabling of each interrupt source is set by means of the DTIE bit in DMABCRL for the corresponding channel in DMABCRL, and interrupts from each source are sent to the interrupt controller independently. The priority of transfer end interrupts on each channel is decided by the interrupt controller, as shown in table 7.12.

Figure 7.38 shows a block diagram of a transfer end/transfer break interrupt. An interrupt is always generated when the DTIE bit is set to 1 while the DTE bit in DMABCRL is cleared to 0.

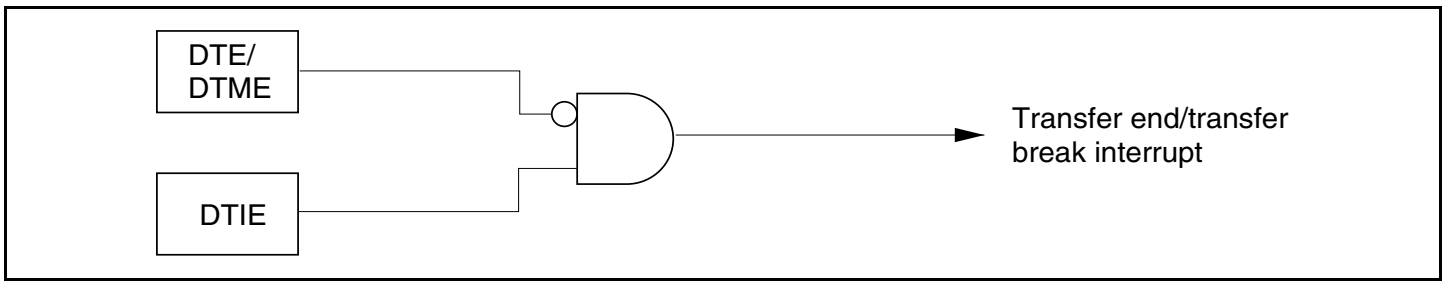


Figure 7.38 Block Diagram of Transfer End/Transfer Break Interrupt

In full address mode, a transfer break interrupt is generated when the DTME bit is cleared to 0 while the DTIEB bit is set to 1. In both short address mode and full address mode, DMABCR should be set so as to prevent the occurrence of a combination that constitutes a condition for interrupt generation during setting.

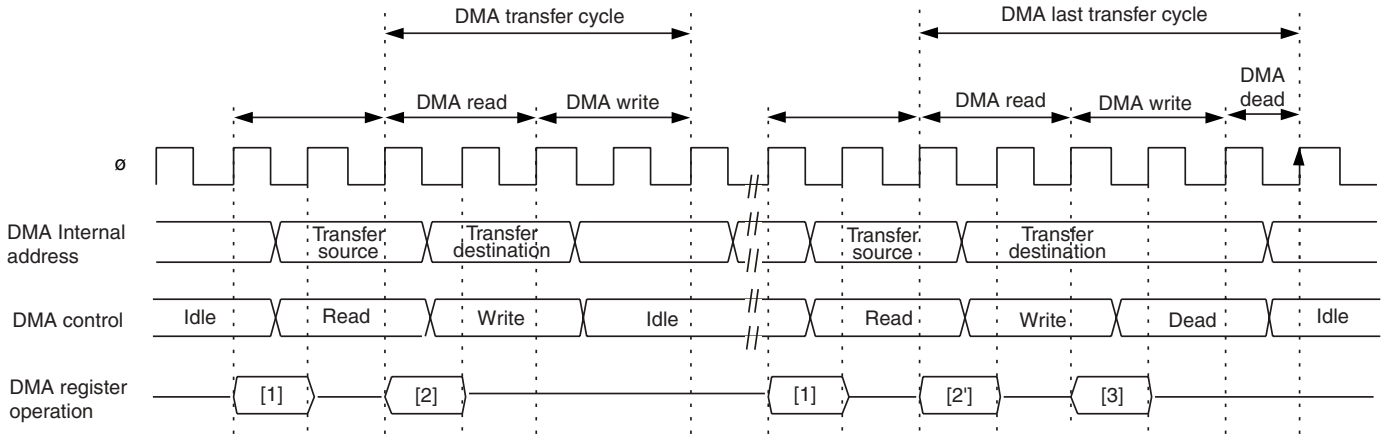
7.7 Usage Notes

7.7.1 DMAC Register Access during Operation

Except for forced termination of the DMAC, the operating (including transfer waiting state) channel setting should not be changed. The operating channel setting should only be changed when transfer is disabled. Also, DMAC registers should not be written to in a DMA transfer.

DMAC register reads during operation (including the transfer waiting state) are described below.

- DMAC control starts one cycle before the bus cycle, with output of the internal address. Consequently, MAR is updated in the bus cycle before DMA transfer. Figure 7.39 shows an example of the update timing for DMAC registers in dual address transfer mode.

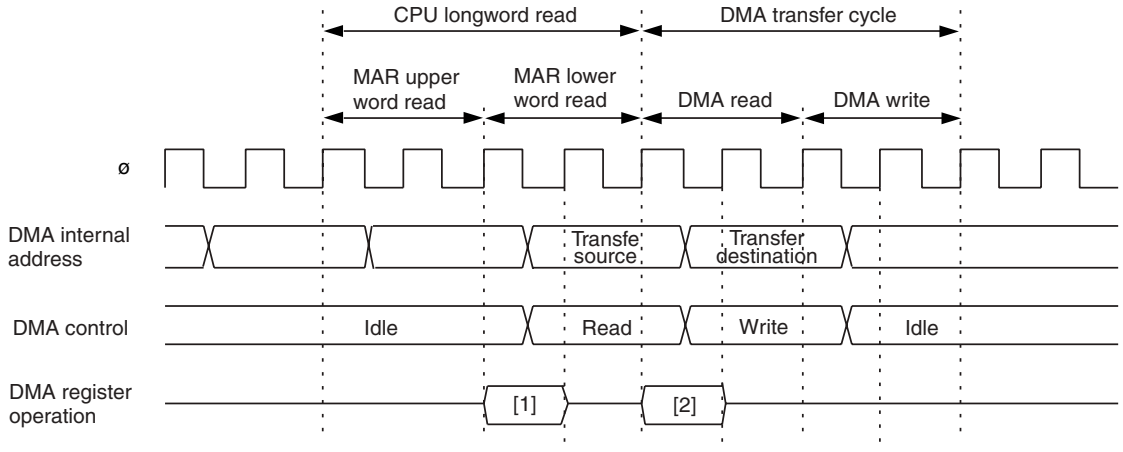


[1] Transfer source address register MAR operation (incremented/decremented/fixed)
Transfer counter ETCR operation (decremented)
Block size counter ETCR operation (decremented in block transfer mode)
[2] Transfer destination address register MAR operation (incremented/decremented/fixed)
[2']Transfer destination address register MAR operation (incremented/decremented/fixed)
Block transfer counter ETCR operation (decremented, in last transfer cycle of a block in block transfer mode)
[3] Transfer address register MAR restore operation (in block or repeat transfer mode)
Transfer counter ETCR restore (in repeat transfer mode)
Block size counter ETCR restore (in block transfer mode)

Note: In single address transfer mode, the update timing is the same as [1].
The MAR operation is post-incrementing/decrementing of the DMA internal address value.

Figure 7.39 DMAC Register Update Timing

- If a DMAC transfer cycle occurs immediately after a DMAC register read cycle, the DMAC register is read as shown in figure 7.40.



Note: The lower word of MAR is the updated value after the operation in [1].

Figure 7.40 Contention between DMAC Register Update and CPU Read

7.7.2 Module Stop

When the MSTP13 bit in MSTPCRH is set to 1, the DMAC clock stops, and the module stop state is entered. However, 1 cannot be written to the MSTP13 bit if any of the DMAC channels is enabled. This setting should therefore be made when DMAC operation is stopped.

When the DMAC clock stops, DMAC register accesses can no longer be made. Since the following DMAC register settings are valid even in the module stop state, they should be invalidated, if necessary, before a module stop.

- Transfer end/break interrupt (DTE = 0 and DTIE = 1)
- TEND pin enable (TEE = 1)
- DACK pin enable (FAE = 0 and SAE = 1)

7.7.3 Write Data Buffer Function

When the WDBE bit of BCR in the bus controller is set to 1, enabling the write data buffer function, dual address transfer external write cycles or single address transfers and internal accesses (on-chip memory or internal I/O registers) are executed in parallel.

- Write data buffer function and DMAC register setting

  If the setting of a register that controls external accesses is changed during execution of an external access by means of the write data buffer function, the external access may not be performed normally. Registers that control external accesses should only be manipulated when external reads, etc., are used with DMAC operation disabled, and the operation is not performed in parallel with external access.
- Write data buffer function and DMAC operation timing

  The DMAC can start its next operation during external access using the write data buffer function. Consequently, the DREQ pin sampling timing, TEND output timing, etc., are different from the case in which the write data buffer function is disabled. Also, internal bus cycles maybe hidden, and not visible.

7.7.4 TEND Output

If the last transfer cycle is for an internal address, note that even if low-level output at the TEND pin has been set, a low level may not be output at the TEND pin under the following external bus conditions since the last transfer cycle (internal bus cycle) and the external bus cycle are executed in parallel.

1. EXDMAC cycle
2. Write cycle with write buffer mode enabled
3. DMAC single address cycle for a different channel with write buffer mode enabled

4. Bus release cycle
5. CBR refresh cycle

Figure 7.41 shows an example in which a low level is not output from the TEND pin in case 2 above.

If the last transfer cycle is an external address cycle, a low level is output at the TEND pin in synchronization with the bus cycle.

However, if the last transfer cycle and a CBR refresh occur simultaneously, note that although the CBR refresh and the last transfer cycle may be executed consecutively, TEND may also go low in this case for the refresh cycle.

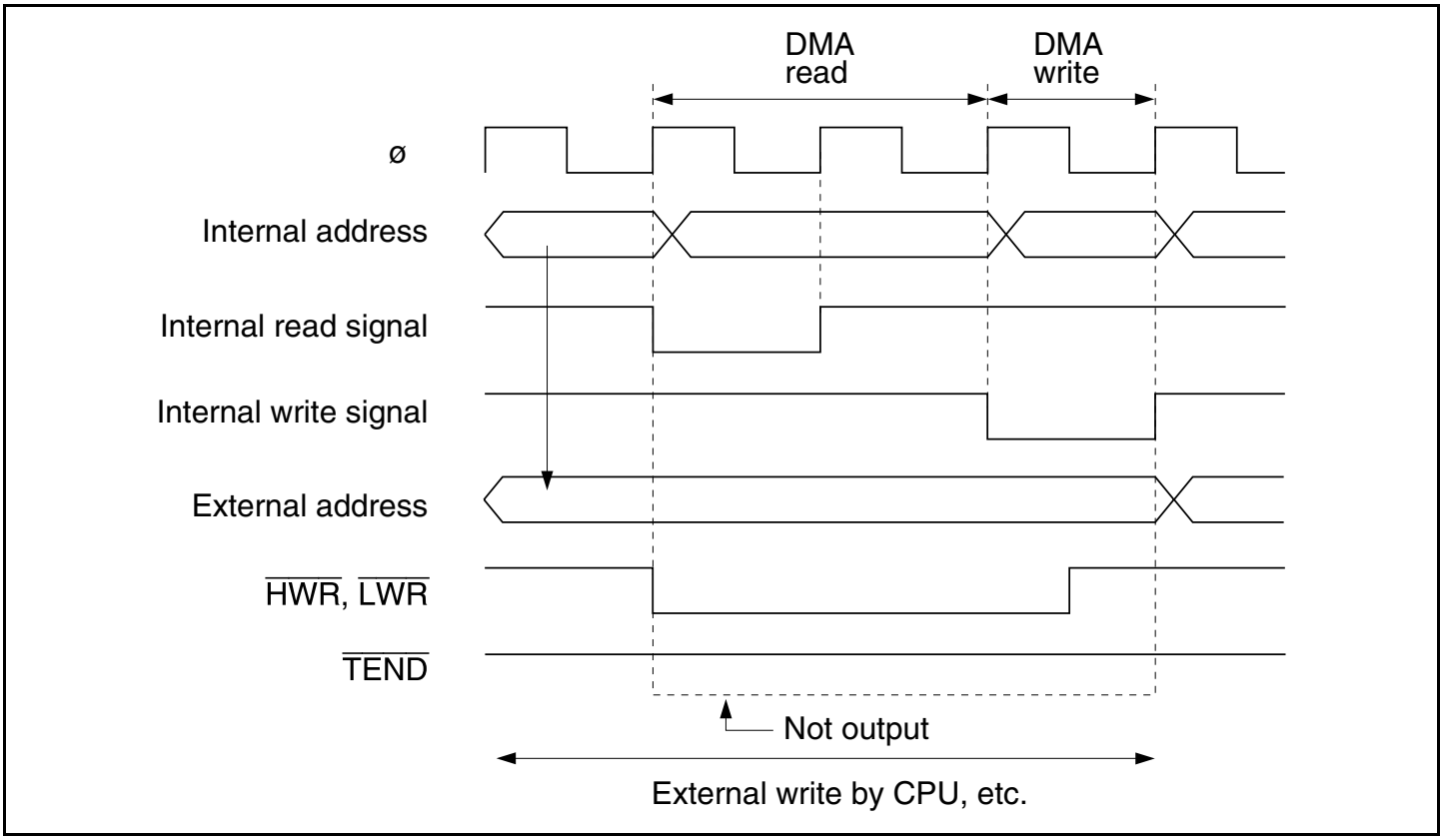


Figure 7.41 Example in Which Low Level is Not Output at TEND Pin

7.7.5 Activation by Falling Edge on DREQ Pin

DREQ pin falling edge detection is performed in synchronization with DMAC internal operations. The operation is as follows:

[1] Activation request wait state: Waits for detection of a low level on the DREQ pin, and switches to [2].

[2] Transfer wait state: Waits for DMAC data transfer to become possible, and switches to [3].

[3] Activation request disabled state: Waits for detection of a high level on the DREQ pin, and switches to [1].

After DMAC transfer is enabled, a transition is made to [1]. Thus, initial activation after transfer is enabled is performed on detection of a low level.

7.7.6 Activation Source Acceptance

At the start of activation source acceptance, a low level is detected in both DREQ pin falling edge sensing and low level sensing. Similarly, in the case of an internal interrupt, the interrupt request is detected. Therefore, a request is accepted from an internal interrupt or DREQ pin low level that occurs before write to DMABCRL to enable transfer.

When the DMAC is activated, take any necessary steps to prevent an internal interrupt or DREQ pin low level remaining from the end of the previous transfer, etc.

7.7.7 Internal Interrupt after End of Transfer

When the DTE bit in DMABCRL is cleared to 0 at the end of a transfer or by a forcible termination, the selected internal interrupt request will be sent to the CPU or DTC even if the DTA bit in DMABCRH is set to 1.

Also, if internal DMAC activation has already been initiated when operation is forcibly terminated, the transfer is executed but flag clearing is not performed for the selected internal interrupt even if the DTA bit is set to 1.

An internal interrupt request following the end of transfer or a forcible termination should be handled by the CPU as necessary.

7.7.8 Channel Re-Setting

To reactivate a number of channels when multiple channels are enabled, use exclusive handling of transfer end interrupts, and perform DMABCR control bit operations exclusively.

Note, in particular, that in cases where multiple interrupts are generated between reading and writing of DMABCR, and a DMABCR operation is performed during new interrupt handling, the DMABCR write data in the original interrupt handling routine will be incorrect, and the write may invalidate the results of the operations by the multiple interrupts. Ensure that overlapping DMABCR operations are not performed by multiple interrupts, and that there is no separation between read and write operations by the use of a bit-manipulation instruction.

Also, when the DTE and DTME bits are cleared by the DMAC or are written with 0, they must first be read while cleared to 0 before the CPU can write 1 to them.

Section 8 EXDMA Controller

This LSI has a built-in four-channel external bus transfer DMA controller (EXDMAC). The EXDMAC can carry out high-speed data transfer, in place of the CPU, to and from external devices and external memory with a DACK (DMA transfer notification) facility.

8.1 Features

- Direct specification of 16-Mbyte address space
- Selection of byte or word transfer data length
- Maximum number of transfers: 16M (16,777,215)/infinite (free-running)
- Selection of dual address mode or single address mode
- Selection of cycle steal mode or burst mode as bus mode
- Selection of normal mode or block transfer mode as transfer mode
- Two kinds of transfer requests: external request and auto-request
- An interrupt request can be sent to the CPU at the end of the specified number of transfers.
- Repeat area designation function:
- Operation in parallel with internal bus master:
- Acceptance of a transfer request and the start of transfer processing can be reported to an external device via the EDRAK pin.
- Module stop mode can be set.

Figure 8.1 shows a block diagram of the EXDMAC.

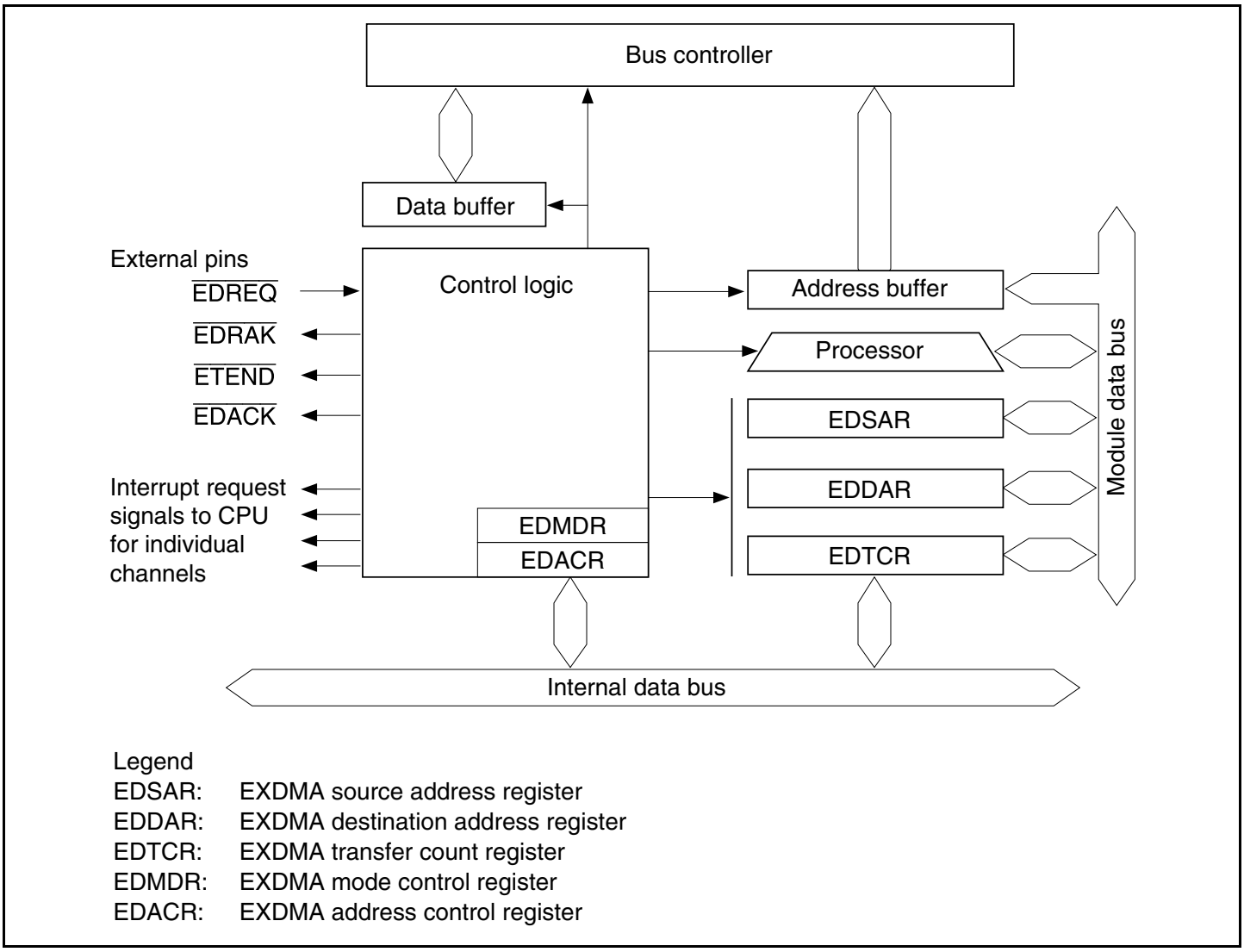


Figure 8.1 Block Diagram of EXDMAC

8.2 Input/Output Pins

Table 8.1 shows the EXDMAC pin configuration.

Table 8.1 Pin Configuration

Channel	Name	Abbre-viation	I/O	Function
0	EXDMA transfer request 0	EDREQ0	Input	Channel 0 external request
	EXDMA transfer acknowledge 0	EDACK0	Output	Channel 0 single address transfer acknowledge
	EXDMA transfer end 0	ETEND0	Output	Channel 0 transfer end
	EDREQ0 acceptance acknowledge	EDRAK0	Output	Notification to external device of channel 0 external request acceptance and start of execution
1	EXDMA transfer request 1	EDREQ1	Input	Channel 1 external request
	EXDMA transfer acknowledge 1	EDACK1	Output	Channel 1 single address transfer acknowledge
	EXDMA transfer end 1	ETEND1	Output	Channel 1 transfer end
	EDREQ1 acceptance acknowledge	EDRAK1	Output	Notification to external device of channel 1 external request acceptance and start of execution
2	EXDMA transfer request 2	EDREQ2	Input	Channel 2 external request
	EXDMA transfer acknowledge 2	EDACK2	Output	Channel 2 single address transfer acknowledge
	EXDMA transfer end 2	ETEND2	Output	Channel 2 transfer end
	EDREQ2 acceptance acknowledge	EDRAK2	Output	Notification to external device of channel 2 external request acceptance and start of execution
3	EXDMA transfer request 3	EDREQ3	Input	Channel 3 external request
	EXDMA transfer acknowledge 3	EDACK3	Output	Channel 3 single address transfer acknowledge
	EXDMA transfer end 3	ETEND3	Output	Channel 3 transfer end
	EDREQ3 acceptance acknowledge	EDRAK3	Output	Notification to external device of channel 3 external request acceptance and start of execution

8.3 Register Descriptions

The EXDMAC has the following registers.

- EXDMA source address register_0 (EDSAR_0)
- EXDMA destination address register_0 (EDDAR_0)
- EXDMA transfer count register_0 (EDTCR_0)
- EXDMA mode control register_0 (EDMDR_0)
- EXDMA address control register_0 (EDACR_0)
- EXDMA source address register_1 (EDSAR_1)
- EXDMA destination address register_1 (EDDAR_1)
- EXDMA transfer count register_1 (EDTCR_1)
- EXDMA mode control register_1 (EDMDR_1)
- EXDMA address control register_1 (EDACR_1)
- EXDMA source address register_2 (EDSAR_2)
- EXDMA destination address register_2 (EDDAR_2)
- EXDMA transfer count register_2 (EDTCR_2)
- EXDMA mode control register_2 (EDMDR_2)
- EXDMA address control register_2 (EDACR_2)
- EXDMA source address register_3 (EDSAR_3)
- EXDMA destination address register_3 (EDDAR_3)
- EXDMA transfer count register_3 (EDTCR_3)
- EXDMA mode control register_3 (EDMDR_3)
- EXDMA address control register_3 (EDACR_3)

8.3.1 EXDMA Source Address Register (EDSAR)

EDSAR is a 32-bit readable/writable register that specifies the transfer source address. An address update function is provided that updates the register contents to the next transfer source address each time transfer processing is performed. In single address mode, the EDSAR value is ignored when a device with DACK is specified as the transfer source. The upper 8 bits of EDSAR are reserved; they are always read as 0 and cannot be modified.

EDSAR can be read at all times by the CPU. When reading EDSAR for a channel on which EXDMA transfer processing is in progress, a longword-size read must be executed. Do not write to EDSAR for a channel on which EXDMA transfer is in progress. The initial values of EDSAR are undefined.

8.3.2 EXDMA Destination Address Register (EDDAR)

EDDAR is a 32-bit readable/writable register that specifies the transfer destination address. An address update function is provided that updates the register contents to the next transfer destination address each time transfer processing is performed. In single address mode, the EDDAR value is ignored when a device with DACK is specified as the transfer destination. The upper 8 bits of EDDAR are reserved; they are always read as 0 and cannot be modified.

EDDAR can be read at all times by the CPU. When reading EDDAR for a channel on which EXDMA transfer processing is in progress, a longword-size read must be executed. Do not write to EDDAR for a channel on which EXDMA transfer is in progress. The initial values of EDDAR are undefined.

8.3.3 EXDMA Transfer Count Register (EDTCR)

EDTCR specifies the number of transfers. The function differs according to the transfer mode. Do not write to EDTCR for a channel on which EXDMA transfer is in progress.

Normal Transfer Mode:

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
23 to 0		All 0	R/W	24-Bit Transfer Counter These bits specify the number of transfers. Setting H'000001 specifies one transfer. Setting H'000000 means no specification for the number of transfers, and the transfer counter function is halted. In this case, there is no transfer end interrupt by the transfer counter. Setting H'FFFFFF specifies the maximum number of transfers, that is 16,777,215. During EXDMA transfer, this counter shows the remaining number of transfers. This counter can be read at all times. When reading EDTCR for a channel on which EXDMA transfer processing is in progress, a longword-size read must be executed.

Block Transfer Mode:

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
23 to 16		Undefined	R/W	Block Size These bits specify the block size (number of bytes or number of words) for block transfer. Setting H'01 specifies one as the block, while setting H'00 specifies the maximum block size, that is 256. The register value always indicates the specified block size.
15 to 0		Undefined	R/W	16-Bit Transfer Counter These bits specify the number of block transfers. Setting H'0001 specifies one block transfer. Setting H'0000 means no specification for the number of transfers, and the transfer counter function is halted. In this case, there is no transfer end interrupt by the transfer counter. Setting H'FFFF specifies the maximum number of block transfers, that is 65,535. During EXDMA transfer, this counter shows the remaining number of block transfers.

8.3.4 EXDMA Mode Control Register (EDMDR)

EDMDR controls EXDMAC operations.

Bit	Bit Name	Initial Value	R/W	Description
15	EDA	0	R/(W)	EXDMA Active Enables or disables data transfer on the corresponding channel. When this bit is set to 1, this indicates that an EXDMA operation is in progress. When auto request mode is specified (by bits MDS1 and MDS0), transfer processing begins when this bit is set to 1. With external requests, transfer processing begins when a transfer request is issued after this bit has been set to 1. When this bit is cleared to 0 during an EXDMA operation, transfer is halted. If this bit is cleared to 0 during an EXDMA operation in block transfer mode, transfer processing is continued for the currently executing one-block transfer, and the bit is cleared on completion of the currently executing one-block transfer. If an external source that ends (aborts) transfer occurs, this bit is automatically cleared to 0 and transfer is terminated. Do not change the operating mode, transfer method, or other parameters while this bit is set to 1. 0: Data transfer disabled on corresponding channel [Clearing conditions] • When the specified number of transfers end • When operation is halted by a repeat area overflow interrupt • When 0 is written to EDA while EDA = 1 (In block transfer mode, write is effective after end of one-block transfer) • Reset, NMI interrupt, hardware standby mode 1: Data transfer enabled on corresponding channel Note: The value written in the EDA bit may not be effective immediately.

Bit	Bit Name	Initial Value	R/W	Description
14	BEF	0	R/(W)*	Block Transfer Error Flag Flag that indicates the occurrence of an error during block transfer. If an NMI interrupt is generated during block transfer, the EXDMAC immediately terminates the EXDMA operation and sets this bit to 1. The address registers indicate the next transfer addresses, but the data for which transfer has been performed within the block size is lost. 0: No block transfer error [Clearing condition] Writing 0 to BEF after reading BEF = 1 1: Block transfer error [Setting condition] NMI interrupt during block transfer
13	EDRAKE	0	R/W	EDRAK Pin Output Enable Enables output from the EDREQ acknowledge/execution start (EDRAK) pin. 0: EDRAK pin output disabled 1: EDRAK pin output enabled
12	ETENDE	0	R/W	ETEND Pin Output Enable Enables output from the EXDMA transfer end (ETEND) pin. 0: ETEND pin output disabled 1: ETEND pin output enabled
11	EDREQS	0	R/W	EDREQ Select Specifies low level sensing or falling edge sensing as the sampling method for the EDREQ pin used in external request mode. 0: Low level sensing (Low level sensing is used for the first transfer after transfer is enabled.) 1: Falling edge sensing
10	AMS	0	R/W	Address Mode Select Selects single address mode or dual address mode. When single address mode is selected, the EDACK pin is valid. 0: Dual address mode 1: Single address mode

Bit	Bit Name	Initial Value	R/W	Description
9	MDS1	0	R/W	Mode Select 1 and 0
8	MDS0	0	R/W	These bits specify the activation source, bus mode, and transfer mode.

Bit	Bit Name	Initial Value	R/W	Description
5	TCEIE	0	R/W	Transfer Counter End Interrupt Enable Enables or disables transfer end interrupt requests by the transfer counter. When transfer ends according to the transfer counter while this bit is set to 1, the IRF bit is set to 1, indicating that an interrupt request has occurred. 0: Transfer end interrupt requests by transfer counter are disabled 1: Transfer end interrupt requests by transfer counter are enabled
4	SDIR	0	R/W	Single Address Direction Specifies the data transfer direction in single address mode. In dual address mode, the specification by this bit is ignored. 0: Transfer direction: EDSAR → external device with DACK 1: Transfer direction: External device with DACK→ EDDAR
3	DTSIZE	0	R/W	Data Transmit Size Specifies the size of data to be transferred. 0: Byte-size 1: Word-size
2	BGUP	0	R/W	Bus Give-Up When this bit is set to 1, the bus can be transferred to an internal bus mastership in burst mode or block transfer mode. This setting is ignored in normal mode and cycle steal mode. 0: Bus is not released 1: Bus is transferred if requested by an internal bus master
1	—	0	R/W	Reserved These bits can be read from or written to. However, the write value should always be 0.
0	—	0	R/W	

Note: Only 0 can be written, to clear the flag.

8.3.5 EXDMA Address Control Register (EDACR)

EDACR specifies address register incrementing/decrementing and use of the repeat area function.

Bit	Bit Name	Initial Value	R/W	Description
15	SAT1	0	R/W	Source Address Update Mode
14	SAT0	0	R/W	These bits specify incrementing/decrementing of the transfer source address (EDSAR). When an external device with DACK is designated as the transfer source in single address mode, the specification by these bits is ignored. 0X: Fixed 10: Incremented (+1 in byte transfer, +2 in word transfer) 11: Decremented (–1 in byte transfer, –2 in word transfer)
13	SARIE	0	R/W	Source Address Repeat Interrupt Enable When this bit is set to 1, in the event of source address repeat area overflow, the IRF bit is set to 1 and the EDA bit cleared to 0 in EDMDR, and transfer is terminated. If the EDIE bit in EDMDR is 1 when the IRF bit in EDMDR is set to 1, an interrupt request is sent to the CPU. When used together with block transfer mode, a source address repeat interrupt is requested at the end of a block-size transfer. If the EDA bit is set to 1 in EDMDR for the channel on which transfer is terminated by a source address repeat interrupt, transfer can be resumed from the state in which it ended. If a source address repeat area has not been designated, this bit is ignored. 0: Source address repeat interrupt is not requested 1: When source address repeat area overflow occurs, the IRF bit in EDMDR is set to 1 and an interrupt is requested

Bit	Bit Name	Initial Value	R/W	Description
12	SARA4	0	R/W	Source Address Repeat Area
11	SARA3	0	R/W	These bits specify the source address (EDSAR) repeat area. The repeat area function updates the specified lower address bits, leaving the remaining upper address bits always the same. A repeat area size of 2 bytes to 8 Mbytes can be specified. The setting interval is a power-of-two number of bytes. When repeat area overflow results from incrementing or decrementing an address, the lower address is the start address of the repeat area in the case of address incrementing, or the last address of the repeat area in the case of address decrementing. If the SARIE bit is set to 1, an interrupt can be requested when repeat area overflow occurs. 00000: Not designated as repeat area 00001: Lower 1 bit (2-byte area) designated as repeat area 00010: Lower 2 bits (4-byte area) designated as repeat area 00011: Lower 3 bits (8-byte area) designated as repeat area 00100: Lower 4 bits (16-byte area) designated as repeat area : : 10011: Lower 19 bits (512-kbyte area) designated as repeat area 10100: Lower 20 bits (1-Mbyte area) designated as repeat area 10101: Lower 21 bits (2-Mbyte area) designated as repeat area 10110: Lower 22 bits (4-Mbyte area) designated as repeat area 10111: Lower 23 bits (8-Mbyte area) designated as repeat area 11XXX: Setting prohibited
10	SARA2	0	R/W	
9	SARA1	0	R/W	
8	SARA0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	DAT1	0	R/W	Destination Address Update Mode
6	DAT0	0	R/W	These bits specify incrementing/decrementing of the transfer destination address (EDDAR). When an external device with DACK is designated as the transfer destination in single address mode, the specification by these bits is ignored. 0X: Fixed 10: Incremented (+1 in byte transfer, +2 in word transfer) 11: Decremented (–1 in byte transfer, –2 in word transfer)
5	DARIE	0	R/W	Destination Address Repeat Interrupt Enable When this bit is set to 1, in the event of destination address repeat area overflow the IRF bit is set to 1 and the EDA bit cleared to 0 in EDMDR, and transfer is terminated. If the EDIE bit in EDMDR is 1 when the IRF bit in EDMDR is set to 1, an interrupt request is sent to the CPU. When used together with block transfer mode, a destination address repeat interrupt is requested at the end of a block-size transfer. If the EDA bit is set to 1 in EDMDR for the channel on which transfer is terminated by a destination address repeat interrupt, transfer can be resumed from the state in which it ended. If a destination address repeat area has not been designated, this bit is ignored. 0: Destination address repeat interrupt is not requested 1: When destination address repeat area overflow occurs, the IRF bit in EDMDR is set to 1 and an interrupt is requested

Bit	Bit Name	Initial Value	R/W	Description
4	DARA4	0	R/W	Destination Address Repeat Area
3	DARA3	0	R/W	These bits specify the destination address (EDDAR) repeat area. The repeat area function updates the specified lower address bits, leaving the remaining upper address bits always the same. A repeat area size of 2 bytes to 8 Mbytes can be specified. The setting interval is a power-of-two number of bytes. When repeat area overflow results from incrementing or decrementing an address, the lower address is the start address of the repeat area in the case of address incrementing, or the last address of the repeat area in the case of address decrementing. If the DARIE bit is set to 1, an interrupt can be requested when repeat area overflow occurs. 00000: Not designated as repeat area 00001: Lower 1 bit (2-byte area) designated as repeat area 00010: Lower 2 bits (4-byte area) designated as repeat area 00011: Lower 3 bits (8-byte area) designated as repeat area 00100: Lower 4 bits (16-byte area) designated as repeat area : : 10011: Lower 19 bits (512-kbyte area) designated as repeat area 10100: Lower 20 bits (1-Mbyte area) designated as repeat area 10101: Lower 21 bits (2-Mbyte area) designated as repeat area 10110: Lower 22 bits (4-Mbyte area) designated as repeat area 10111: Lower 23 bits (8-Mbyte area) designated as repeat area 11XXX: Setting prohibited
2	DARA2	0	R/W	
1	DARA1	0	R/W	
0	DARA0	0	R/W	

Legend

x: Don't care

8.4 Operation

8.4.1 Transfer Modes

The transfer modes of the EXDMAC are summarized in table 8.2.

Table 8.2 EXDMAC Transfer Modes

Transfer Mode			Transfer Origin	Number of Transfers	Address Registers	
					Source	Destination
Dual address mode	Normal transfer mode	Auto request mode • Burst/cycle steal mode	Auto request	1 to 16,777,215 or no specification	EDSAR	EDDAR
		External request mode • Cycle steal mode	External request			
	Block transfer mode	External request mode • Burst transfer of specified block size for a single transfer request • Block size: 1 to 256 bytes or words	External request	1 to 65,535 or no specification		
Single address mode	• Direct data transfer to/from external device using EDACK pin instead of source or destination address register • Above transfer mode can be specified in addition to address register setting • One transfer possible in one bus cycle (Transfer mode variations are the same as in dual address mode.)				EDSAR/ EDACK	EDACK/ EDDAR

The transfer mode can be set independently for each channel.

In normal transfer mode, a one-byte or one-word transfer is executed in response to one transfer request. With auto requests, burst or cycle steal transfer mode can be set. In burst transfer mode, continuous, high-speed transfer can be performed until the specified number of transfers have been executed or the transfer enable bit is cleared to 0.

In block transfer mode, a transfer of the specified block size is executed in response to one transfer request. The block size can be from 1 to 256 bytes or words. Within a block, transfer can be performed at the same high speed as in block transfer mode.

When the “no specification” setting (EDTCR = H'000000) is made for the number of transfers, the transfer counter is halted and there is no limit on the number of transfers, allowing transfer to be performed endlessly.

Incrementing or decrementing the memory address by 1 or 2, or leaving the address unchanged, can be specified independently for each address register.

In all transfer modes, it is possible to set a repeat area comprising a power-of-two number of bytes.

8.4.2 Address Modes

Dual Address Mode: In dual address mode, both the transfer source and transfer destination are specified by registers in the EXDMAC, and one transfer is executed in two bus cycles.

The transfer source address is set in the source address register (EDSAR), and the transfer destination address is set in the transfer destination address register (EDDAR).

In a transfer operation, the value in external memory specified by the transfer source address is read in the first bus cycle, and is written to the external memory specified by the transfer destination address in the next bus cycle.

These consecutive read and write cycles are indivisible: another bus cycle (external access by an internal bus master, refresh cycle, or external bus release cycle) does not occur between these two cycles.

ETEND pin output can be enabled or disabled by means of the ETENDE bit in EDMDR. ETEND is output for two consecutive bus cycles. The EDACK signal is not output.

Figure 8.2 shows an example of the timing in dual address mode.

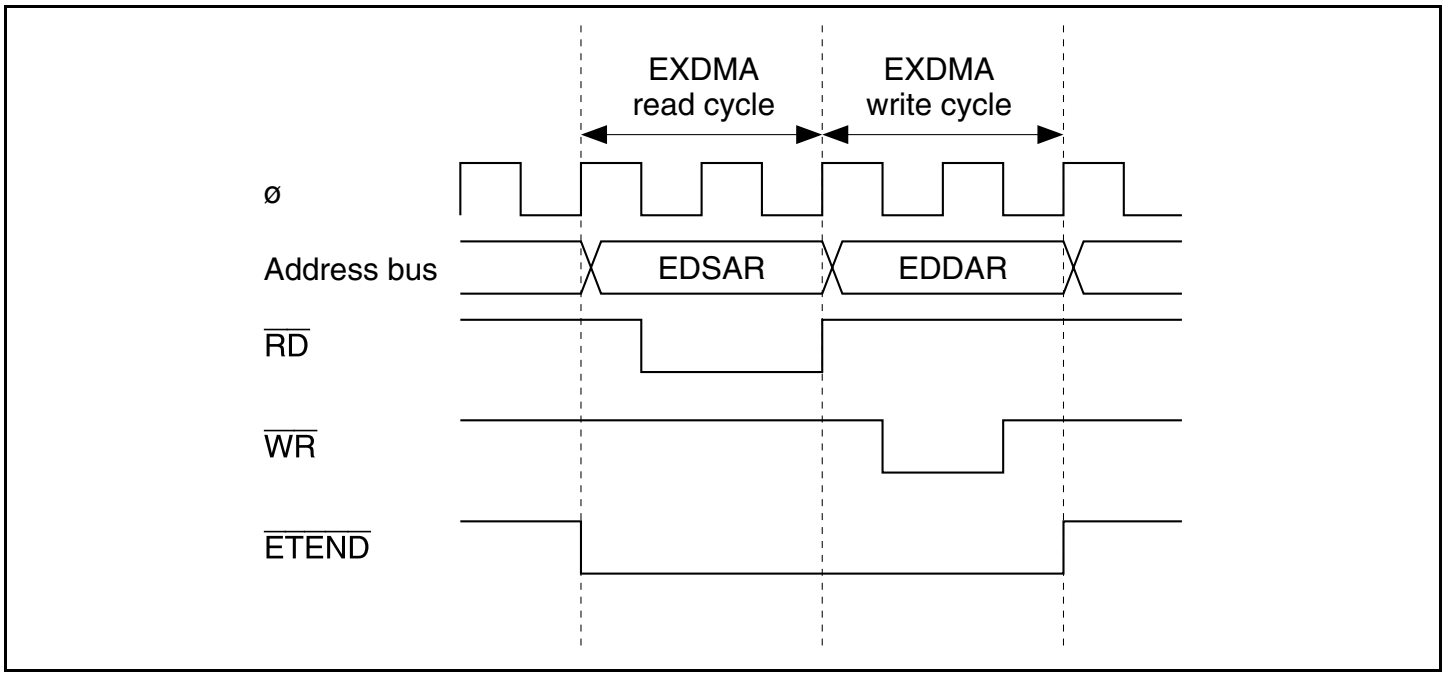


Figure 8.2 Example of Timing in Dual Address Mode

Single Address Mode: In single address mode, the EDACK signal is used instead of the source or destination address register to transfer data directly between an external device and external memory. In this mode, the EXDMAC accesses the transfer source or transfer destination external device by outputting the external I/O strobe signal (EDACK), and at the same time accesses the other external device in the transfer by outputting an address. In this way, DMA transfer can be executed in one bus cycle. In the example of transfer between external memory and an external device with DACK shown in figure 8.3, data is output to the data bus by the external device and written to external memory in the same bus cycle.

The transfer direction, that is whether the external device with DACK is the transfer source or transfer destination, can be specified with the SDIR bit in EDMDR. Transfer is performed from the external memory (EDSAR) to the external device with DACK when SDIR = 0, and from the external device with DACK to the external memory (EDDAR) when SDIR = 1.

The setting in the source or destination address register not used in the transfer is ignored.

The EDACK pin becomes valid automatically when single address mode is selected. The EDACK pin is active-low. ETEND pin output can be enabled or disabled by means of the ETENDE bit in EDMDR. ETEND is output for one bus cycle.

Figure 8.3 shows the data flow in single address mode, and figure 8.4 shows an example of the timing.

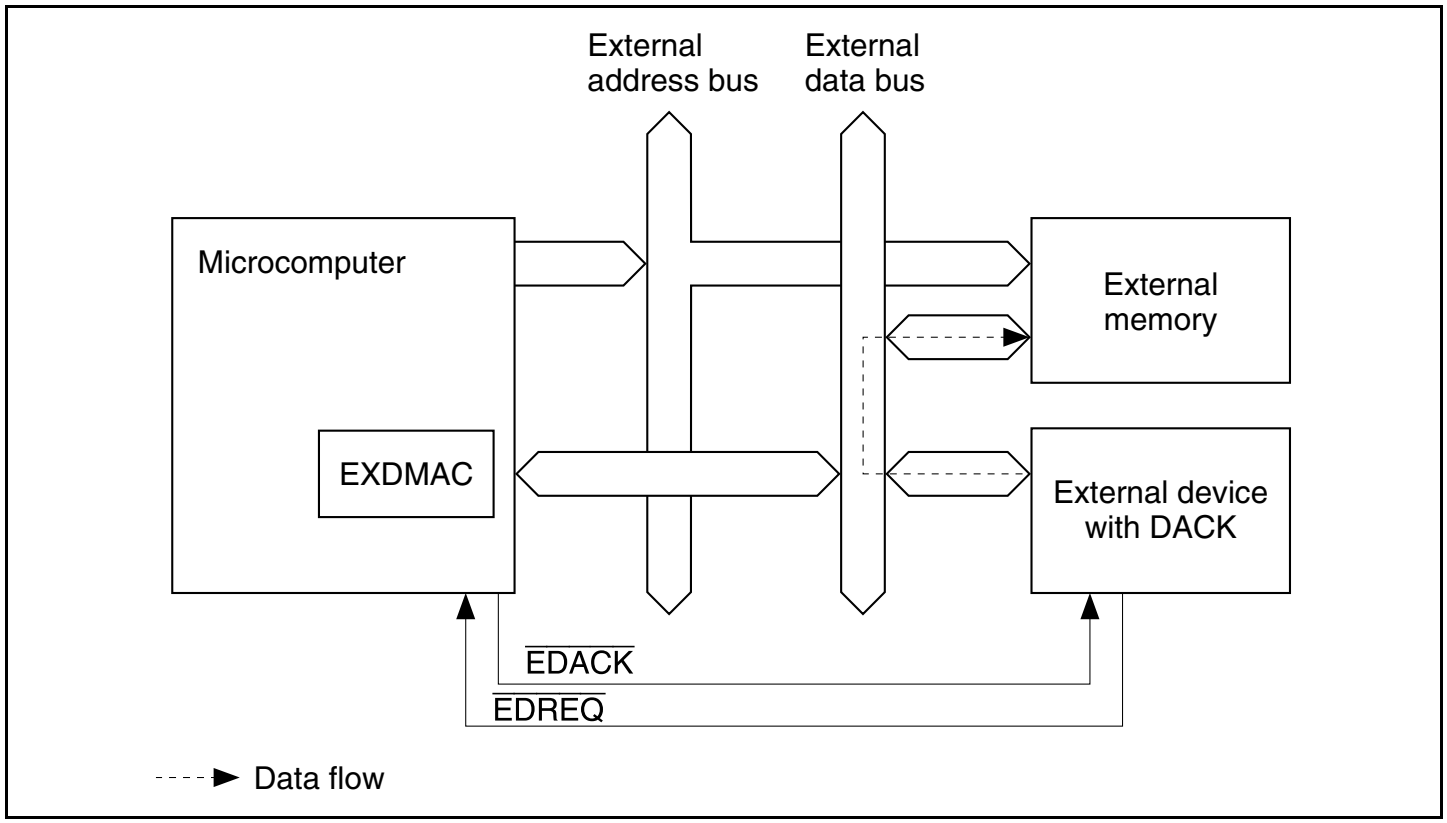


Figure 8.3 Data Flow in Single Address Mode

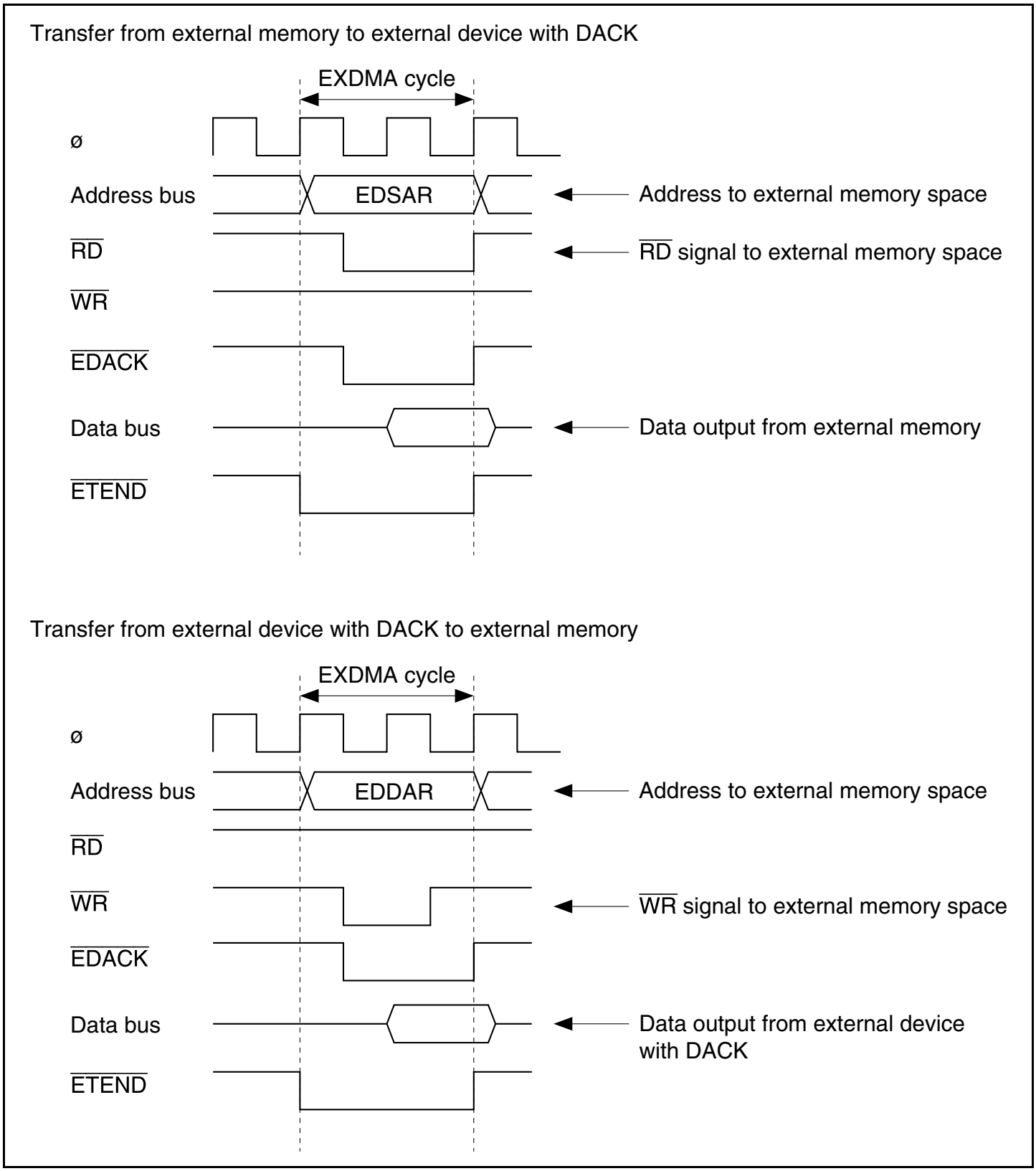


Figure 8.4 Example of Timing in Single Address Mode

8.4.3 DMA Transfer Requests

Auto Request Mode: In auto request mode, transfer request signals are automatically generated within the EXDMAC in cases where a transfer request signal is not issued from outside, such as in transfer between two memories, or between a peripheral module that is not capable of generating transfer requests and memory. In auto request mode, transfer is started when the EDA bit is set to 1 in EDMDR.

In auto request mode, either cycle steal mode or burst mode can be selected as the bus mode. Block transfer mode cannot be used.

External Request Mode: In external request mode, transfer is started by a transfer request signal (EDREQ) from a device external to this LSI. DMA transfer is started when EDREQ is input while DMA transfer is enabled (EDA = 1).

The transfer request source need not be the data transfer source or data transfer destination.

The transfer request signal is accepted via the EDREQ pin. Either falling edge sensing or low level sensing can be selected for the EDREQ pin by means of the EDREQS bit in EDMDR (low level sensing when EDREQS = 0, falling edge sensing when EDREQS = 1).

Setting the EDRAKE bit to 1 in EDMDR enables a signal confirming transfer request acceptance to be output from the EDRAK pin. The EDRAK signal is output when acceptance and transfer processing has been started in response to a single external request. The EDRAK signal enables the external device to determine the timing of EDREQ signal negation, and makes it possible to provide handshaking between the transfer request source and the EXDMAC.

In external request mode, block transfer mode can be used instead of burst mode. Block transfer mode allows continuous execution (burst operation) of the specified number of transfers (the block size) in response to a single transfer request. In block transfer mode, the EDRAK signal is output only once for a one-block transfer, since the transfer request via the EDREQ pin is for a block unit.

8.4.4 Bus Modes

There are two bus modes: cycle steal mode and burst mode. When the activation source is an auto request, either cycle steal mode or burst mode can be selected. When the activation source is an external request, cycle steal mode is used.

Cycle Steal Mode: In cycle steal mode, the EXDMAC releases the bus at the end of each transfer of a transfer unit (byte, word, or block). If there is a subsequent transfer request, the EXDMAC takes back the bus, performs another transfer-unit transfer, and then releases the bus again. This procedure is repeated until the transfer end condition is satisfied.

If a transfer request occurs in another channel during DMA transfer, the bus is temporarily released, then transfer is performed on the channel for which the transfer request was issued. If there is no external space bus request from another bus master, a one-cycle bus release interval is inserted. For details on the operation when there are requests for a number of channels, see section 8.4.8, Channel Priority Order.

Figure 8.5 shows an example of the timing in cycle steal mode.

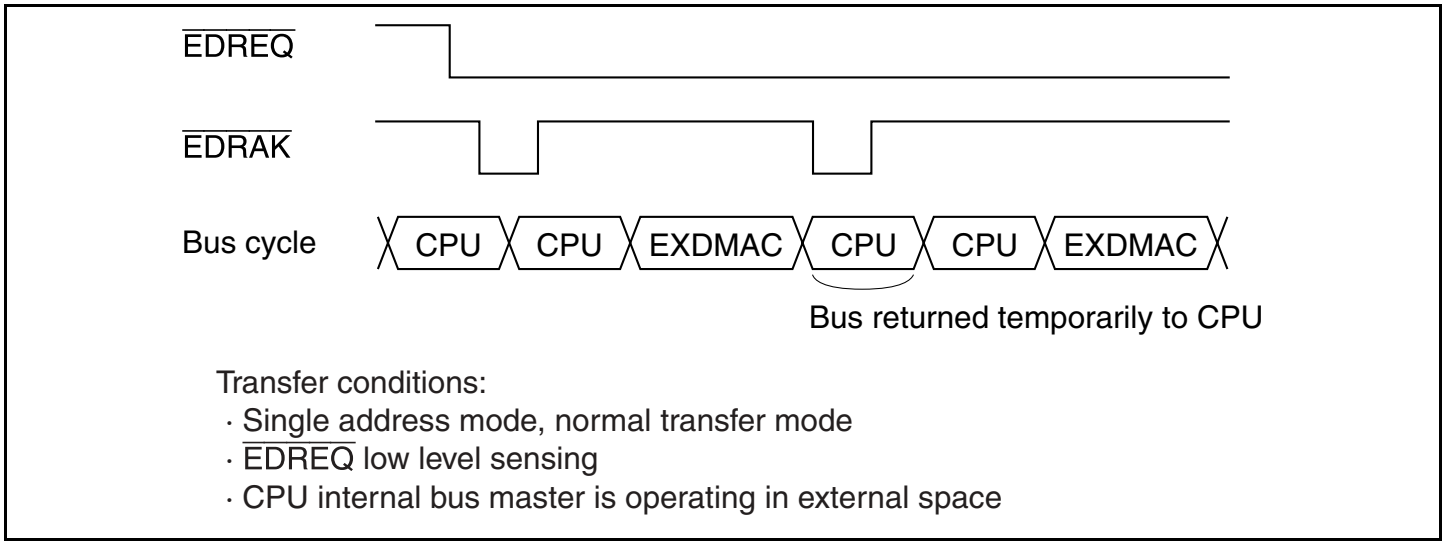


Figure 8.5 Example of Timing in Cycle Steal Mode

Burst Mode: In burst mode, once the EXDMAC acquires the bus it continues transferring data, without releasing the bus, until the transfer end condition is satisfied. There is no burst mode in external request mode.

In burst mode, once transfer is started it is not interrupted even if there is a transfer request from another channel with higher priority. When the burst mode channel finishes its transfer, it releases the bus in the next cycle in the same way as in cycle steal mode.

When the EDA bit is cleared to 0 in EDMDR, DMA transfer is halted. However, DMA transfer is executed for all transfer requests generated within the EXDMAC up until the EDA bit was cleared to 0.

If a repeat area overflow interrupt is generated, the EDA bit is cleared to 0 and transfer is terminated.

When the BGUP bit is set to 1 in EDMDR, the bus is released if a bus request is issued by another bus mastership during burst transfer. If there is no bus request, burst transfer is executed even if the BGUP bit is set to 1.

Figure 8.6 shows examples of the timing in burst mode.

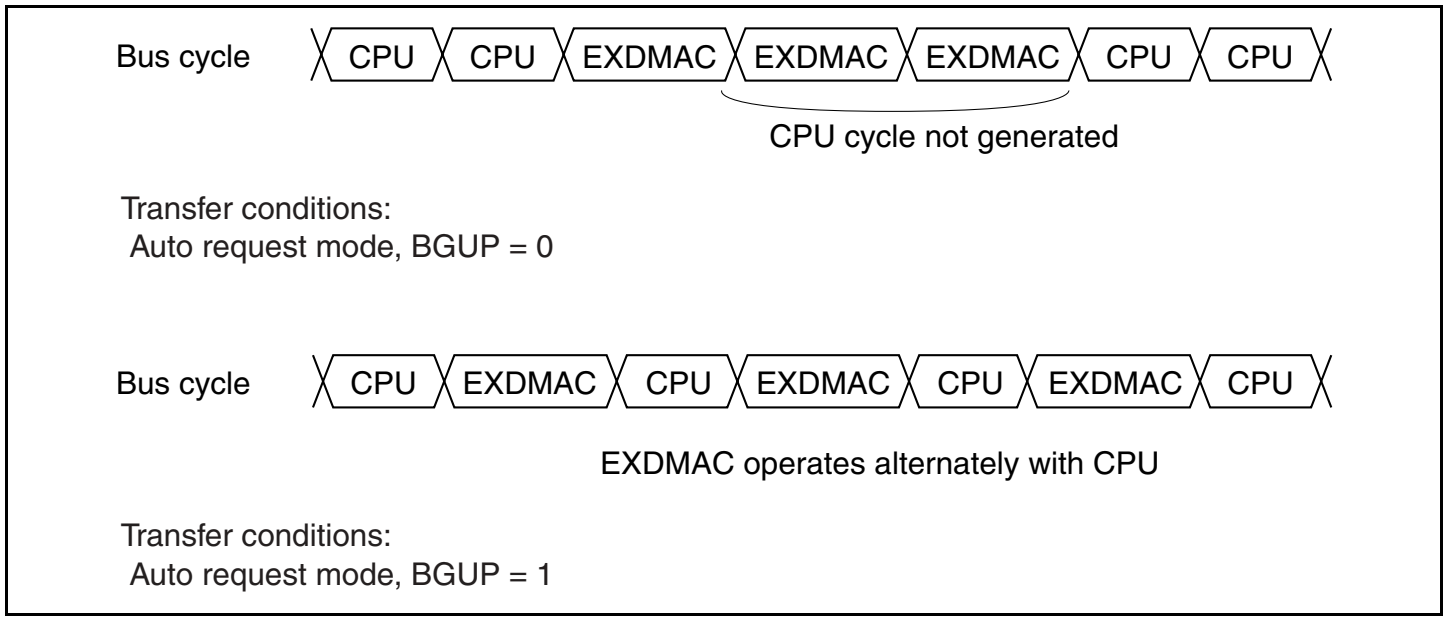


Figure 8.6 Examples of Timing in Burst Mode

8.4.5 Transfer Modes

There are two transfer modes: normal transfer mode and block transfer mode. When the activation source is an external request, either normal transfer mode or block transfer mode can be selected. When the activation source is an auto request, normal transfer mode is used.

Normal Transfer Mode: In normal transfer mode, transfer of one transfer unit is processed in response to one transfer request. EDTCR functions as a 24-bit transfer counter.

The ETEND signal is output only for the last DMA transfer. The EDRAK signal is output each time a transfer request is accepted and transfer processing is started.

Figure 8.7 shows examples of DMA transfer timing in normal transfer mode.

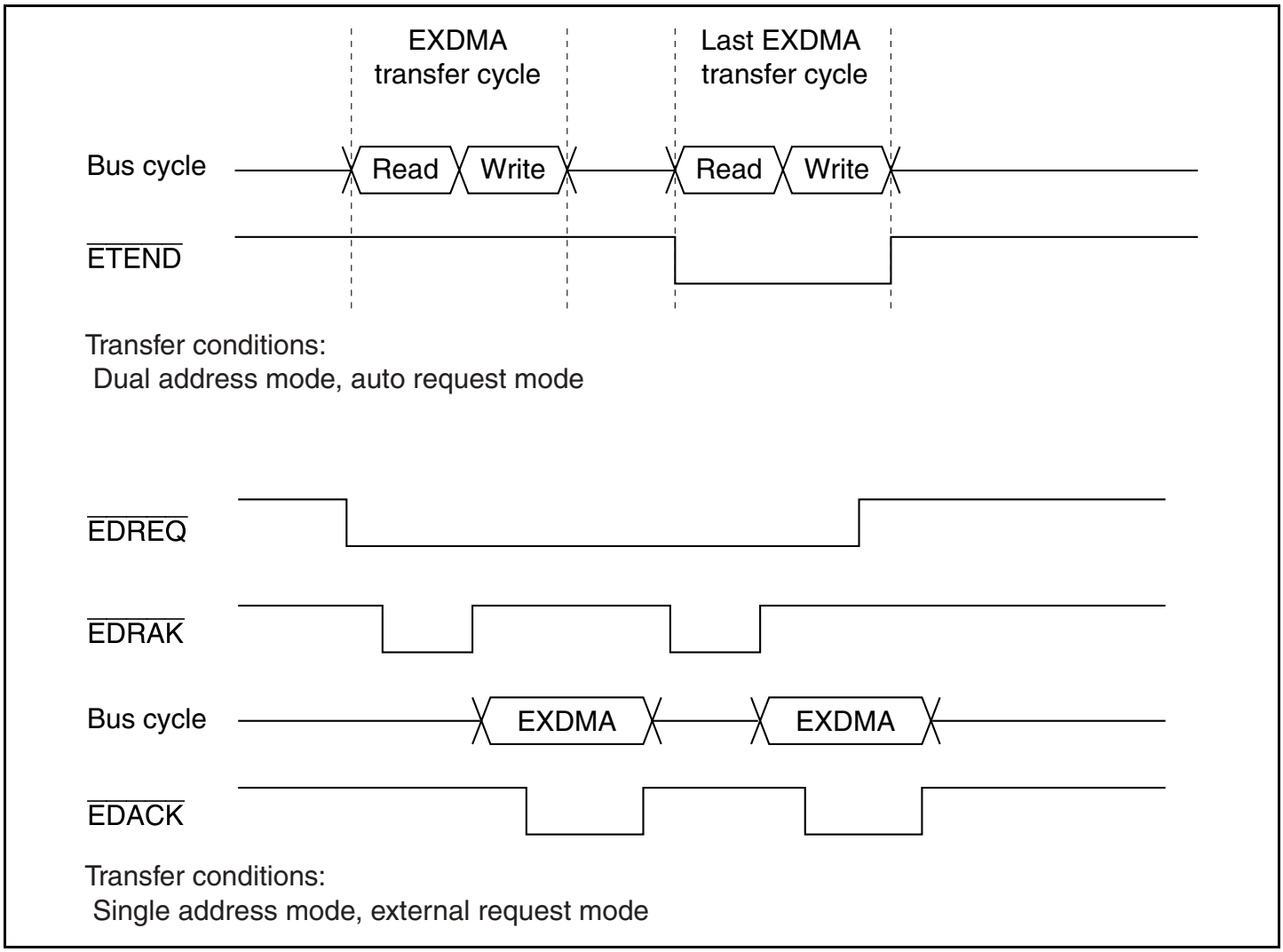


Figure 8.7 Examples of Timing in Normal Transfer Mode

Block Transfer Mode: In block transfer mode, the number of bytes or words specified by the block size is transferred in response to one transfer request. The upper 8 bits of EDTCR specify the block size, and the lower 16 bits function as a 16-bit transfer counter. A block size of 1 to 256 can be specified. During transfer of a block, transfer requests for other higher-priority channels are held pending. When transfer of one block is completed, the bus is released in the next cycle.

When the BGUP bit is set to 1 in EDMDR, the bus is released if a bus request is issued by another bus mastership during block transfer.

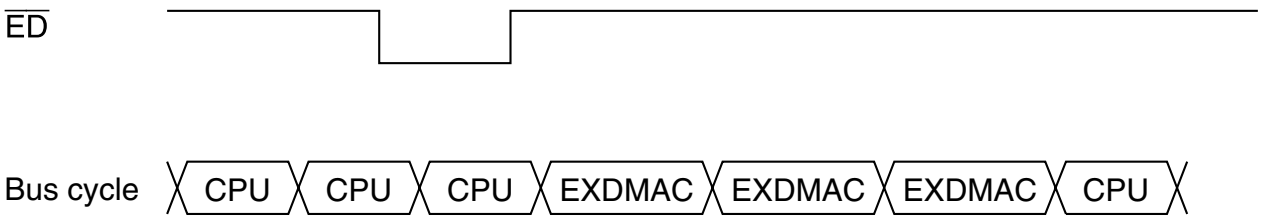
Address register values are updated in the same way as in normal mode. There is no function for restoring the initial address register values after each block transfer.

The ETEND signal is output for each block transfer in the DMA transfer cycle in which the block ends. The EDRAK signal is output once for one transfer request (for transfer of one block).

Caution is required when setting the repeat area overflow interrupt of the repeat area function in block transfer mode. See section 8.4.6, Repeat Area Function, for details.

Block transfer is aborted if an NMI interrupt is generated. See section 8.4.12, Ending DMA Transfer, for details.

Figure 8.8 shows an example of DMA transfer timing in block transfer mode.



IRF bit is set to 1 and the EDA bit cleared to 0 in EDMDR, and transfer is terminated. If EDIE = 1 in EDMDR, an interrupt is requested. If the DARIE bit in EDACR is set to 1, the above applies to the destination address register.

If the EDA bit in EDMDR is set to 1 during interrupt generation, transfer is resumed. Figure 8.9 illustrates the operation of the repeat area function.

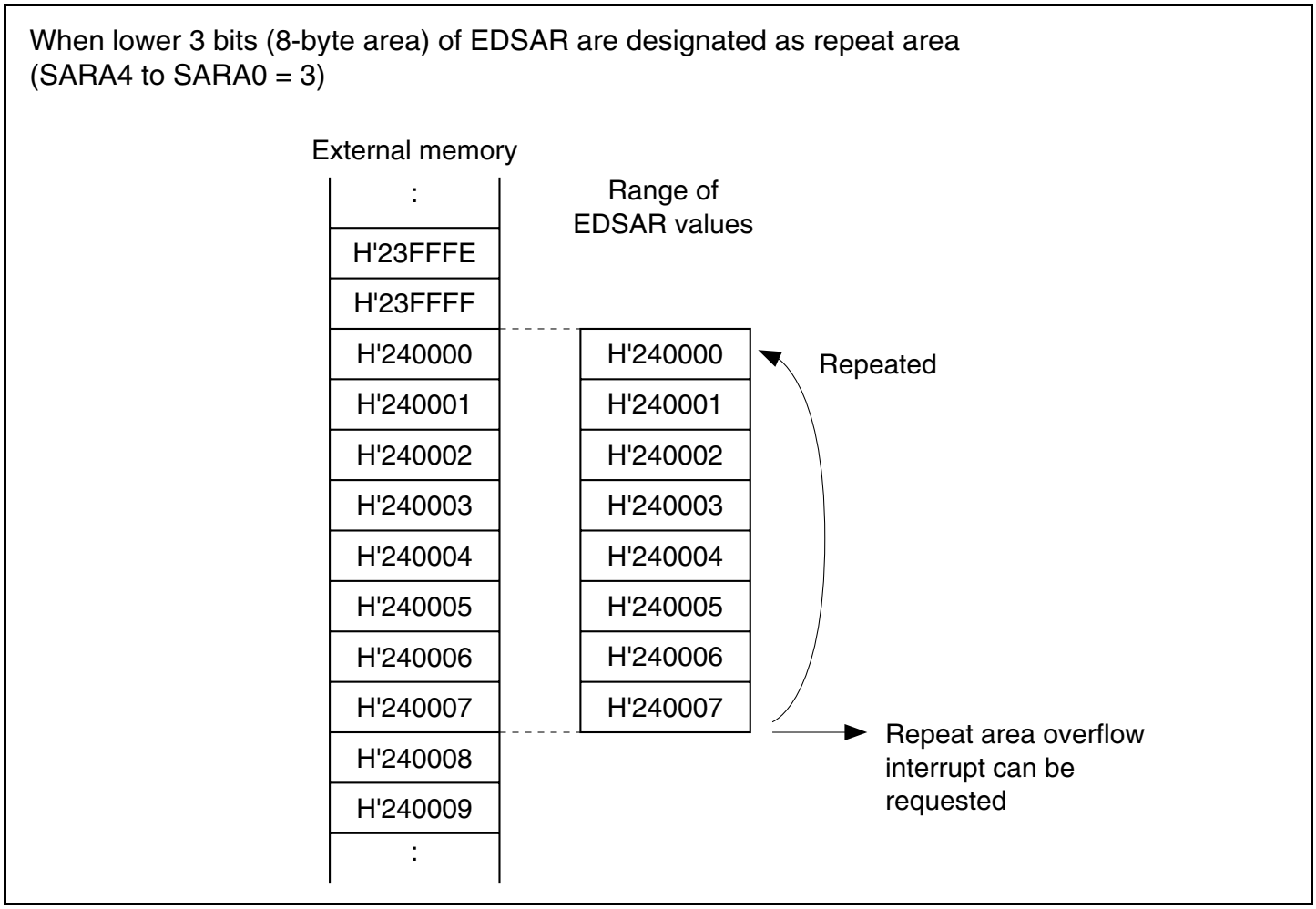


Figure 8.9 Example of Repeat Area Function Operation

Caution is required when the repeat area overflow interrupt function is used together with block transfer mode. If transfer is always terminated when repeat area overflow occurs in block transfer mode, the block size must be a power of two, or alternatively, the address register value must be set so that the end of a block coincides with the end of the repeat area range.

If repeat area overflow occurs while a block is being transferred in block transfer mode, the repeat interrupt request is held pending until the end of the block, and transfer overrun will occur. Figure 8.10 shows an example in which block transfer mode is used together with the repeat area function.

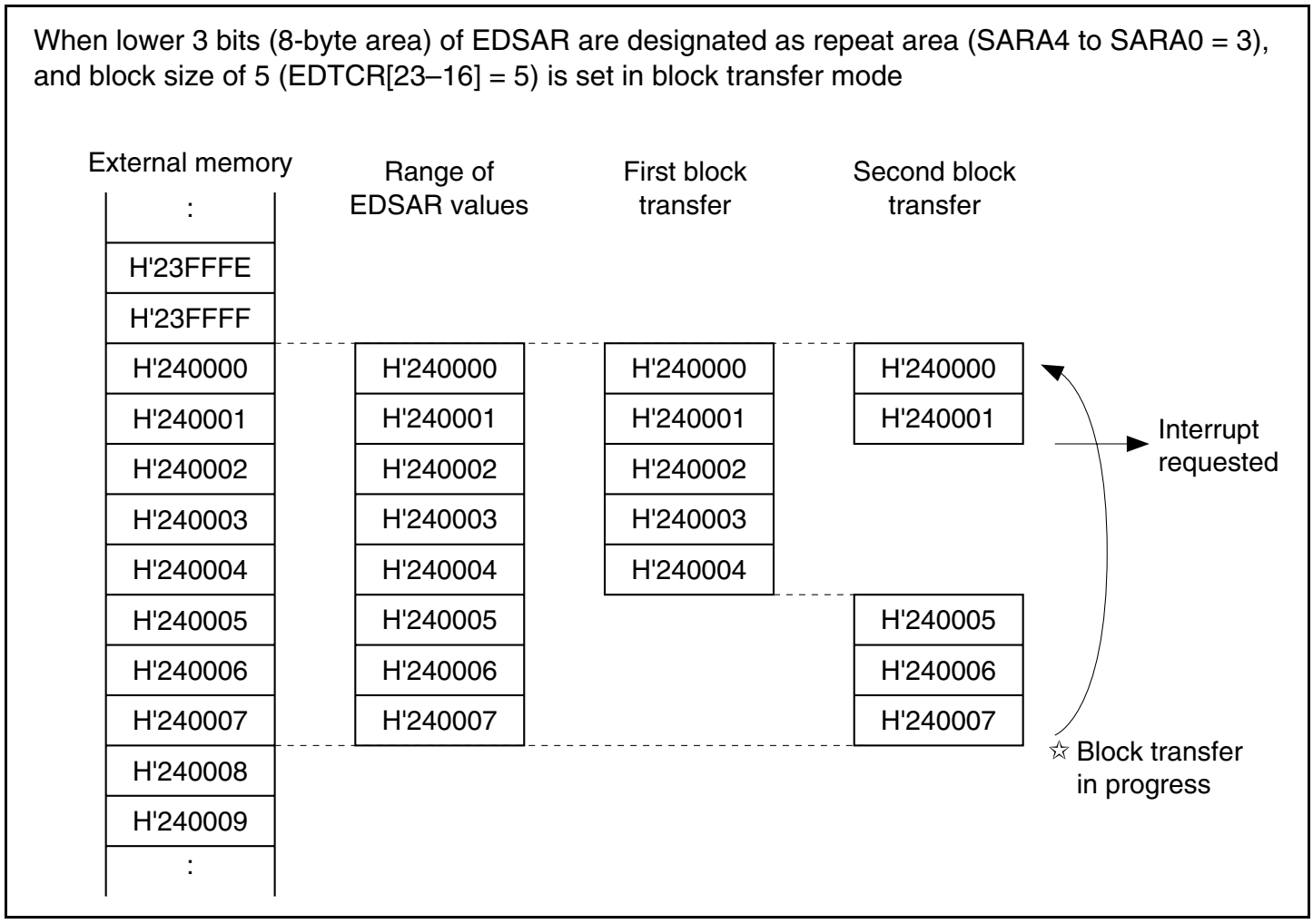


Figure 8.10 Example of Repeat Area Function Operation in Block Transfer Mode

8.4.7 Registers during DMA Transfer Operation

EXDMAC register values are updated as DMA transfer processing is performed. The updated values depend on various settings and the transfer status. The following registers and bits are updated: EDSAR, EDDAR, EDTCR, and bits EDA, BEF, and IRF in EDMDR,

EXDMA Source Address Register (EDSAR): When the EDSAR address is accessed as the transfer source, after the EDSAR value is output, EDSAR is updated with the address to be accessed next. Bits SAT1 and SAT0 in EDACR specify incrementing or decrementing. The address is fixed when SAT1 = 0, incremented when SAT1 = 1 and SAT0 = 0, and decremented when SAT1 = 1 and SAT0 = 1.

The size of the increment or decrement is determined by the size of the data transferred. When the DTSIZE bit in EDMDR = 0, the data is byte-size and the address is incremented or decremented by 1; when DTSIZE = 1, the data is word-size and the address is incremented or decremented by 2.

When a repeat area setting is made, the operation conforms to that setting. The upper part of the address set for the repeat area function is fixed, and is not affected by address updating.

When EDSAR is read during a transfer operation, a longword access must be used. During a transfer operation, EDSAR may be updated without regard to accesses from the CPU, and the correct values may not be read if the upper and lower words are read separately. In a longword access, the EXDMAC buffers the EDSAR value to ensure that the correct value is output.

Do not write to EDSAR for a channel on which a transfer operation is in progress.

EXDMA Destination Address Register (EDDAR): When the EDDAR address is accessed as the transfer destination, after the EDDAR value is output, EDDAR is updated with the address to be accessed next. Bits DAT1 and DAT0 in EDACR specify incrementing or decrementing. The address is fixed when DAT1 = 0, incremented when DAT1 = 1 and DAT0 = 0, and decremented when DAT1 = 1 and DAT0 = 1.

The size of the increment or decrement is determined by the size of the data transferred. When the DTSIZE bit in EDMDR = 0, the data is byte-size and the address is incremented or decremented by 1; when DTSIZE = 1, the data is word-size and the address is incremented or decremented by 2.

When a repeat area setting is made, the operation conforms to that setting. The upper part of the address set for the repeat area function is fixed, and is not affected by address updating.

When EDDAR is read during a transfer operation, a longword access must be used. During a transfer operation, EDDAR may be updated without regard to accesses from the CPU, and the correct values may not be read if the upper and lower words are read separately. In a longword access, the EXDMAC buffers the EDDAR value to ensure that the correct value is output.

Do not write to EDDAR for a channel on which a transfer operation is in progress.

EXDMA Transfer Count Register (EDTCR): When a DMA transfer is performed, the value in EDTCR is decremented by 1. However, when the EDTCR value is 0, transfers are not counted and the EDTCR value does not change.

EDTCR functions differently in block transfer mode. The upper 8 bits, EDTCR[23:16], are used to specify the block size, and their value does not change. The lower 16 bits, EDTCR[15:0], function as a transfer counter, the value of which is decremented by 1 when a DMA transfer is performed. However, when the EDTCR[15:0] value is 0, transfers are not counted and the EDTCR[15:0] value does not change.

In normal transfer mode, all of the lower 24 bits of EDTCR may change, so when EDTCR is read by the CPU during DMA transfer, a longword access must be used. During a transfer operation, EDTCR may be updated without regard to accesses from the CPU, and the correct values may not be read if the upper and lower words are read separately. In a longword access, the EXDMAC buffers the EDTCR value to ensure that the correct value is output.

In block transfer mode, the upper 8 bits are never updated, so there is no problem with using word access.

Do not write to EDTCR for a channel on which a transfer operation is in progress. If there is contention between an address update associated with DMA transfer and a write by the CPU, the CPU write has priority.

In the event of contention between an EDTCR update from 1 to 0 and a write (of a nonzero value) by the CPU, the CPU write value has priority as the EDTCR value, but transfer is terminated. Transfer does not end if the CPU writes 0 to EDTCR.

Figure 8.11 shows EDTCR update operations in normal transfer mode and block transfer mode.

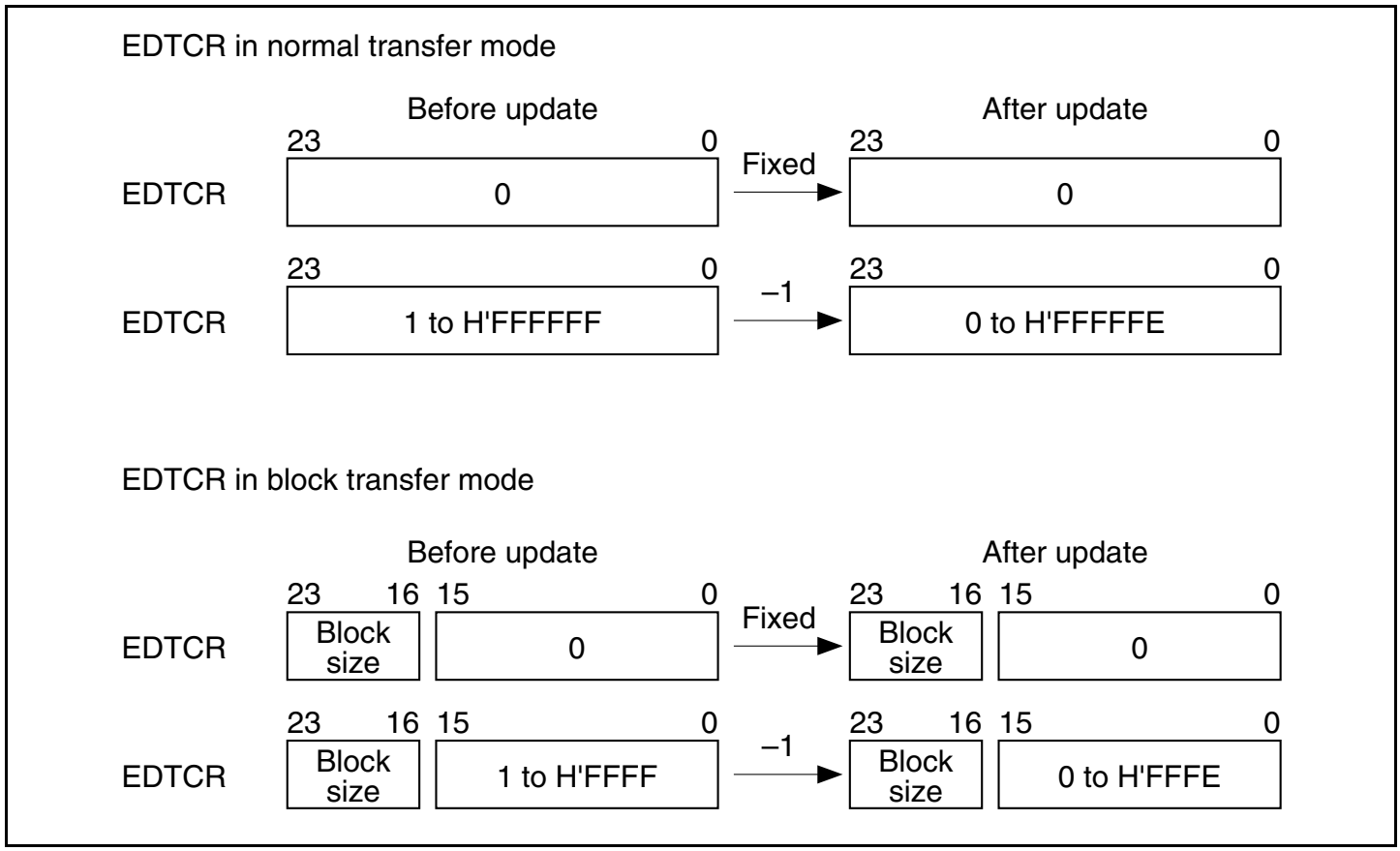


Figure 8.11 EDTCR Update Operations in Normal Transfer Mode and Block Transfer Mode

EDA Bit in EDMDR: The EDA bit in EDMDR is written to by the CPU to control enabling and disabling of data transfer, but may be cleared automatically by the EXDMAC due to the DMA transfer status. There are also periods during transfer when a 0-write to the EDA bit by the CPU is not immediately effective.

Conditions for EDA bit clearing by the EXDMAC include the following:

- When the EDTCR value changes from 1 to 0, and transfer ends
- When a repeat area overflow interrupt is requested, and transfer ends

- When an NMI interrupt is generated, and transfer halts
- A reset
- Hardware standby mode
- When 0 is written to the EDA bit, and transfer halts

When transfer is halted by writing 0 to the EDA bit, the EDA bit remains at 1 during the DMA transfer period. In block transfer mode, since a block-size transfer is carried out without interruption, the EDA bit remains at 1 from the time 0 is written to it until the end of the current block-size transfer.

In burst mode, transfer is halted for up to three DMA transfers following the bus cycle in which 0 is written to the EDA bit. The EDA bit remains set to 1 from the time of the 0-write until the end of the last DMA cycle.

Writes (except to the EDA bit) are prohibited to registers of a channel for which the EDA bit is set to 1. When changing register settings after a 0-write to the EDA bit, it is necessary to confirm that the EDA bit has been cleared to 0.

Figure 8.12 shows the procedure for changing register settings in an operating channel.

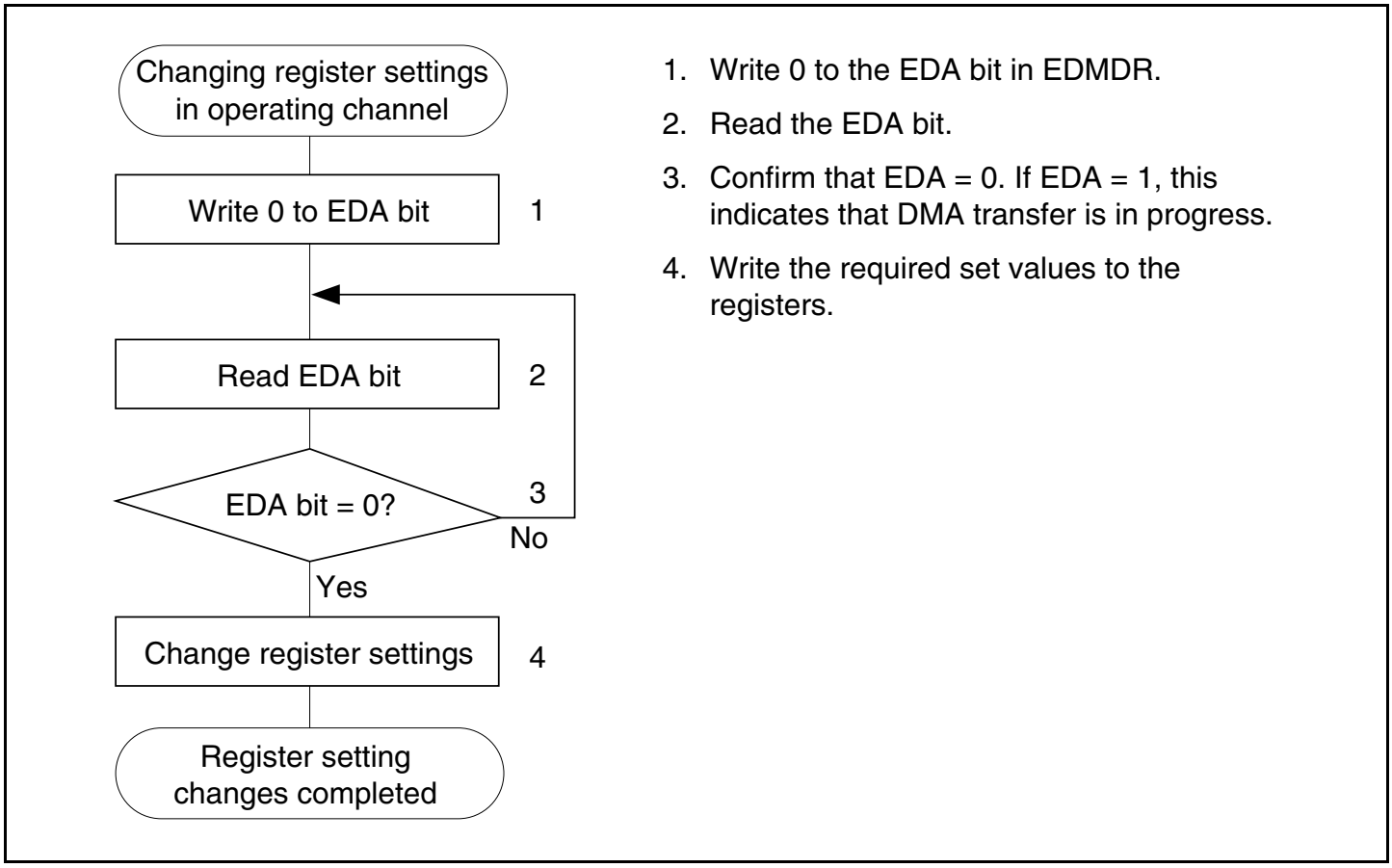


Figure 8.12 Procedure for Changing Register Settings in Operating Channel

BEF Bit in EDMDR: In block transfer mode, the specified number of transfers (equivalent to the block size) is performed in response to a single transfer request. To ensure that the correct number

of transfers is carried out, a block-size transfer is always executed, except in the event of a reset, transition to standby mode, or generation of an NMI interrupt.

If an NMI interrupt is generated during block transfer, operation is halted midway through a block-size transfer and the EDA bit is cleared to 0, terminating the transfer operation. In this case the BEF bit, which indicates the occurrence of an error during block transfer, is set to 1.

IRF Bit in EDMDR: The IRF bit in EDMDR is set to 1 when an interrupt request source occurs. If the EDIE bit in EDMDR is 1 at this time, an interrupt is requested.

The timing for setting the IRF bit to 1 is when the EDA bit in EDMDR is cleared to 0 and transfer ends following the end of the DMA transfer bus cycle in which the source generating the interrupt occurred.

If the EDA bit is set to 1 and transfer is resumed during interrupt handling, the IRF bit is automatically cleared to 0 and the interrupt request is cleared.

For details on interrupts, see section 8.5, Interrupts Sources.

8.4.8 Channel Priority Order

The priority order of the EXDMAC channels is: channel 0 > channel 1 > channel 2 > channel 3. Table 8.3 shows the EXDMAC channel priority order.

Table 8.3 EXDMAC Channel Priority Order

Channel	Priority
Channel 0	High
Channel 1	↑
Channel 2	↑
Channel 3	Low

If transfer requests occur simultaneously for a number of channels, the highest-priority channel according to the priority order in table 8.3 is selected for transfer.

Transfer Requests from Multiple Channels (Except Auto Request Cycle Steal Mode): If transfer requests for different channels are issued during a transfer operation, the highest-priority channel (excluding the currently transferring channel) is selected. The selected channel begins transfer after the currently transferring channel releases the bus. If there is a bus request from a bus mastership other than the EXDMAC at this time, a cycle for the other bus mastership is initiated. If there is no other bus request, the bus is released for one cycle.

Channels are not switched during burst transfer or transfer of a block in block transfer mode.

Figure 8.13 shows an example of the transfer timing when transfer requests occur simultaneously for channels 0, 1, and 2. The example in the figure is for external request cycle steal mode.

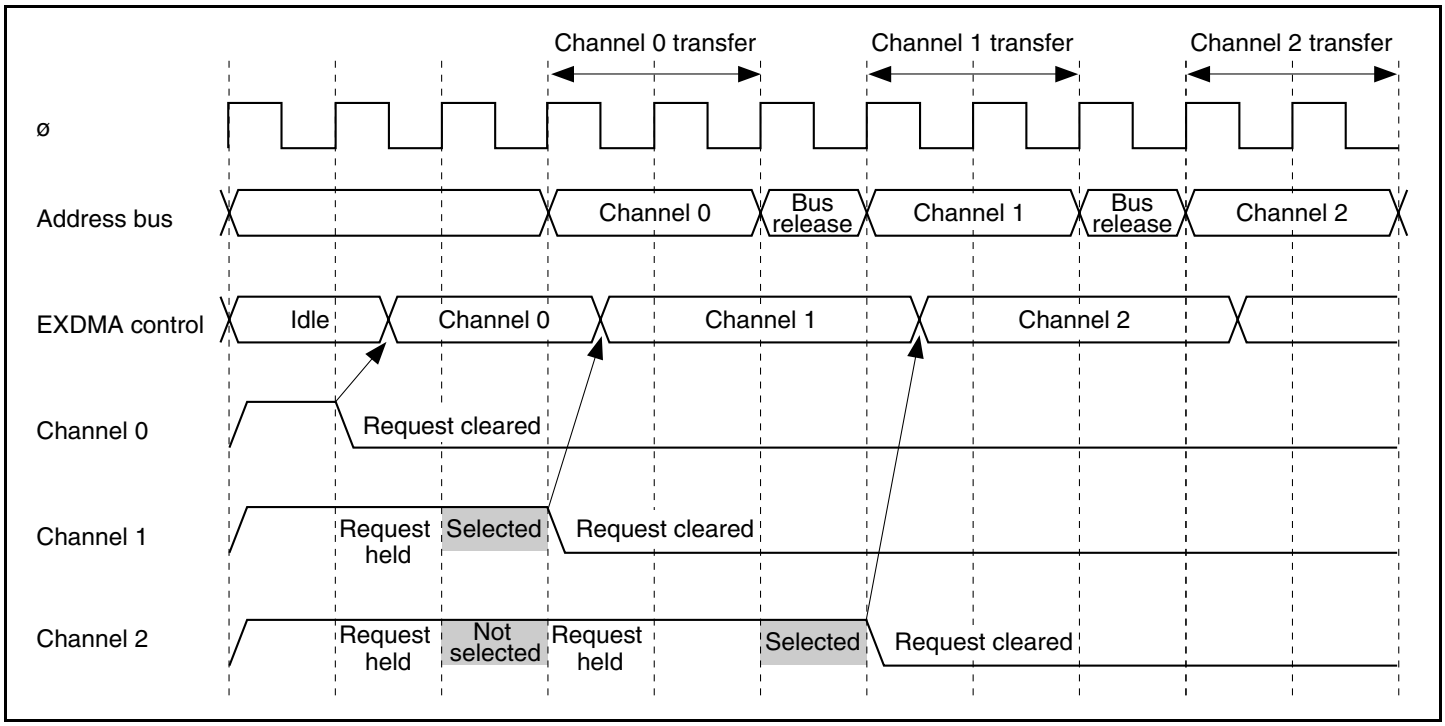


Figure 8.13 Example of Channel Priority Timing

Transfer Requests from Multiple Channels in Auto Request Cycle Steal Mode: If transfer requests for different channels are issued during a transfer in auto request cycle steal mode, the operation depends on the channel priority. If the channel that made the transfer request is of higher priority than the channel currently performing transfer, the channel that made the transfer request is selected.

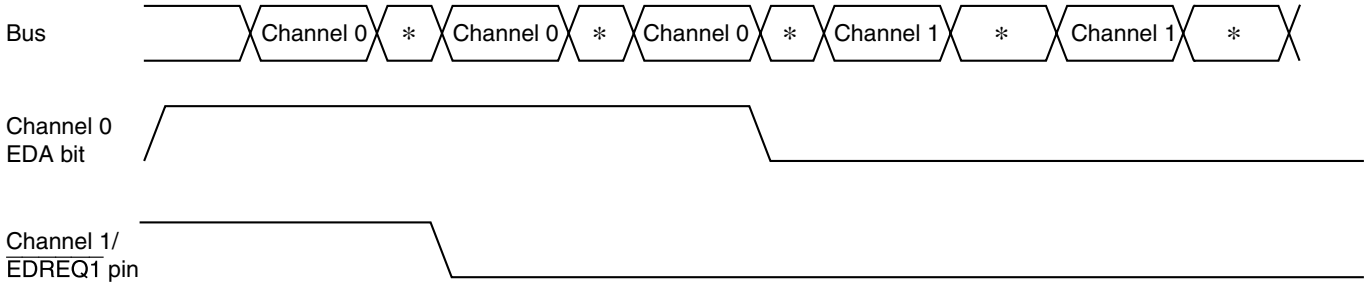
If the channel that made the transfer request is of lower priority than the channel currently performing transfer, that channel's transfer request is held pending, and the currently transferring channel remains selected.

The selected channel begins transfer after the currently transferring channel releases the bus. If there is a bus request from a bus mastership other than the EXDMAC at this time, a cycle for the other bus mastership is initiated. If there is no other bus request, the bus is released for one cycle.

Figure 8.14 shows examples of transfer timing in cases that include auto request cycle steal mode.

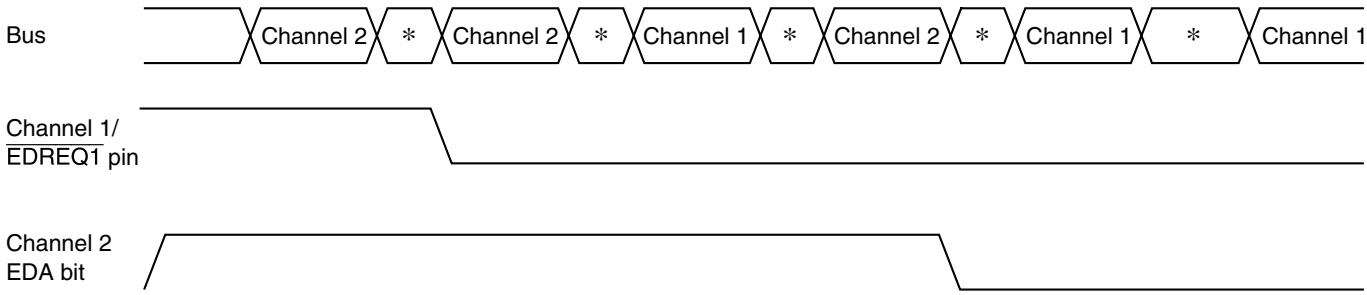
Conditions (1)

Channel 0: Auto request, cycle steal mode
Channel 1: External request, cycle steal mode, low level activation



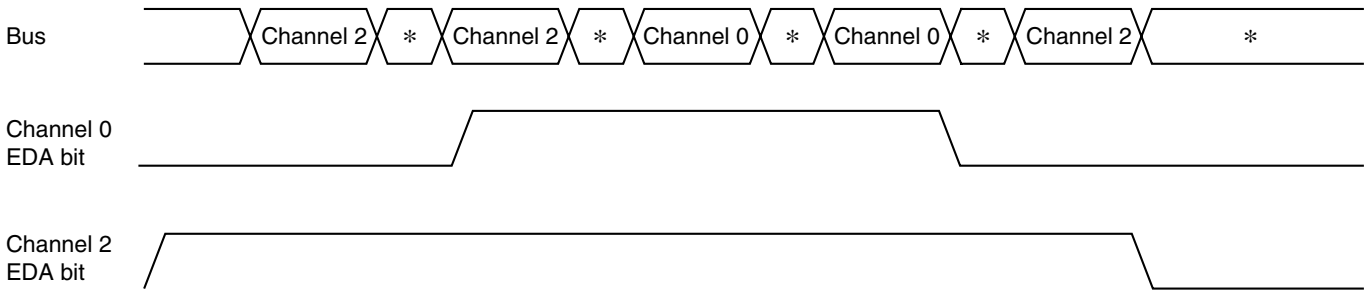
Conditions (2)

Channel 1: External request, cycle steal mode, low level activation
Channel 2: Auto request, cycle steal mode



Conditions (3)

Channel 0: Auto request, cycle steal mode
Channel 2: Auto request, cycle steal mode



*: Bus release

Figure 8.14 Examples of Channel Priority Timing

8.4.9 EXDMAC Bus Cycles (Dual Address Mode)

Normal Transfer Mode (Cycle Steal Mode): Figure 8.15 shows an example of transfer when ETEND output is enabled, and word-size, normal transfer mode (cycle steal mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

After one byte or word has been transferred, the bus is released. While the bus is released, one CPU, DMAC, or DTC bus cycle is initiated.

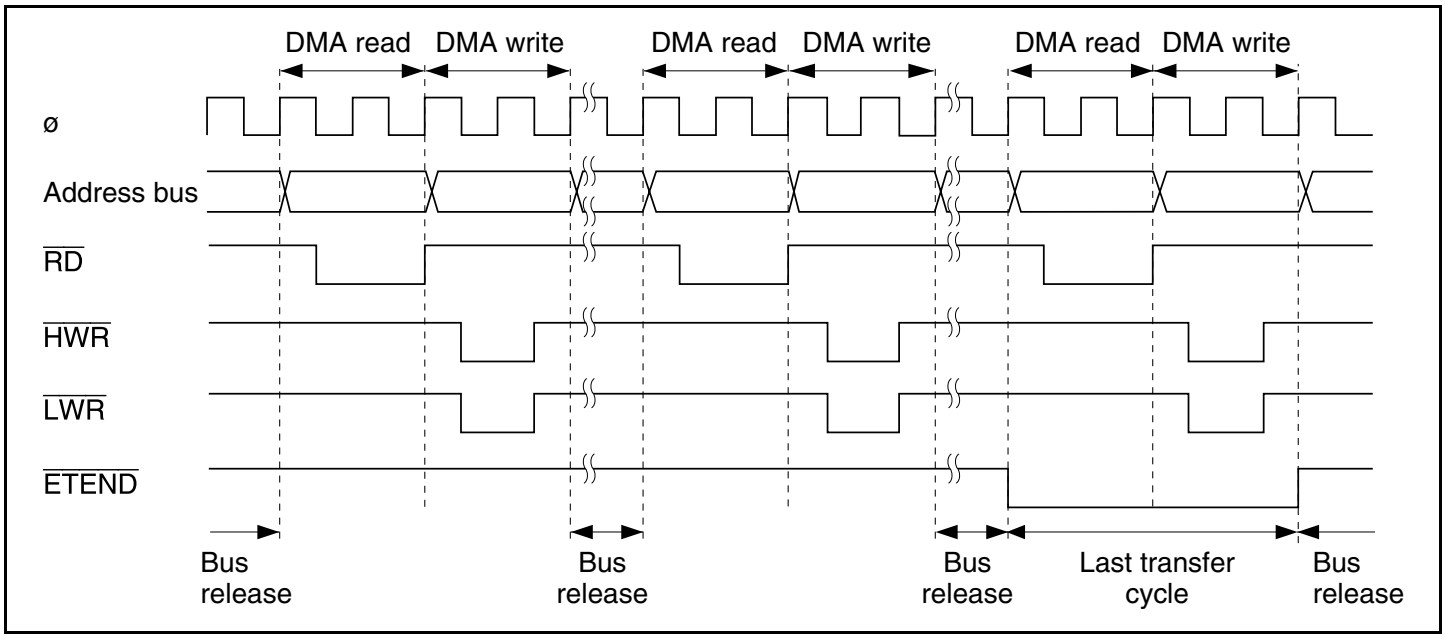


Figure 8.15 Example of Normal Transfer Mode (Cycle Steal Mode) Transfer

Normal Transfer Mode (Burst Mode): Figure 8.16 shows an example of transfer when ETEND output is enabled, and word-size, normal transfer mode (burst mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

In burst mode, one-byte or one-word transfers are executed continuously until transfer ends.

Once burst transfer starts, requests from other channels, even of higher priority, are held pending until transfer ends.

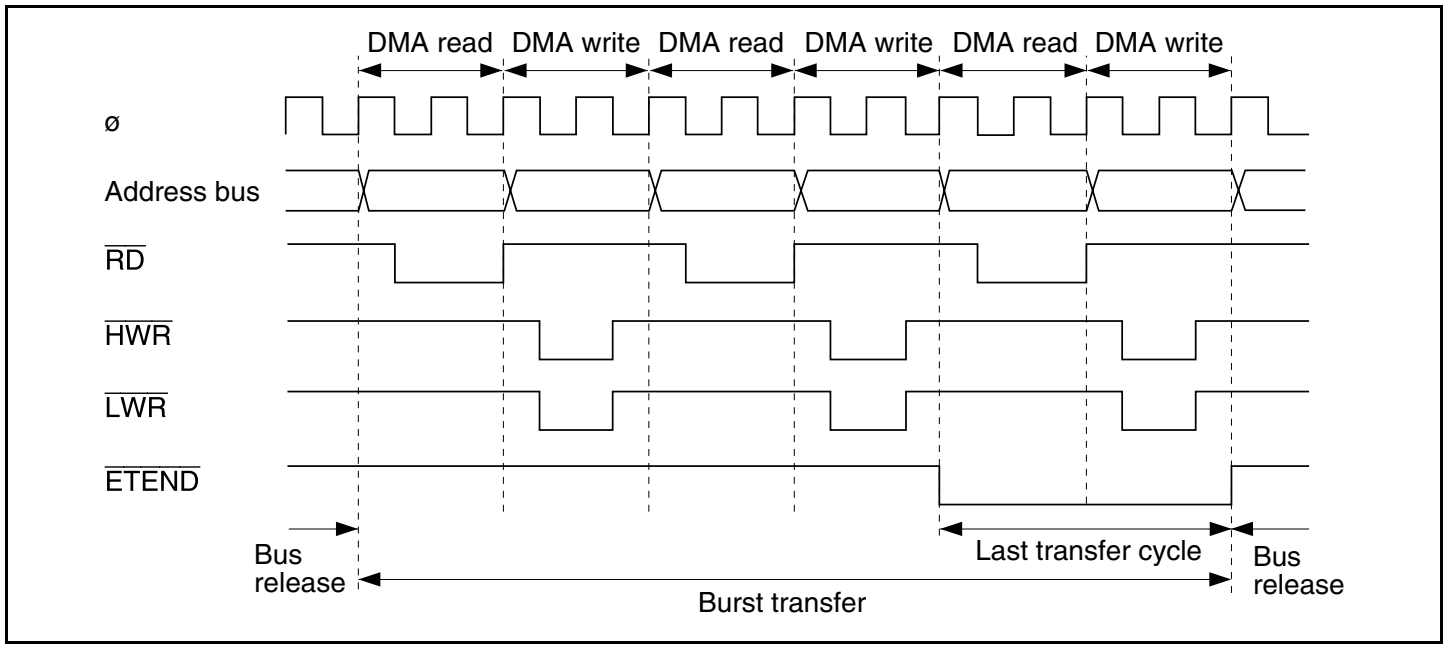


Figure 8.16 Example of Normal Transfer Mode (Burst Mode) Transfer

If an NMI interrupt is generated while a channel designated for burst transfer is enabled for transfer, the EDA bit is cleared and transfer is disabled. If a block transfer has already been initiated within the EXDMAC, the bus is released on completion of the currently executing byte or word transfer, and burst transfer is aborted. If the last transfer cycle in burst transfer has been initiated within the EXDMAC, transfer is executed to the end even if the EDA bit is cleared.

Block Transfer Mode (Cycle Steal Mode): Figure 8.17 shows an example of transfer when ETEND output is enabled, and word-size, block transfer mode (cycle steal mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

One block is transferred in response to one transfer request, and after the transfer, the bus is released. While the bus is released, one or more CPU, DMAC, or DTC bus cycles are initiated.

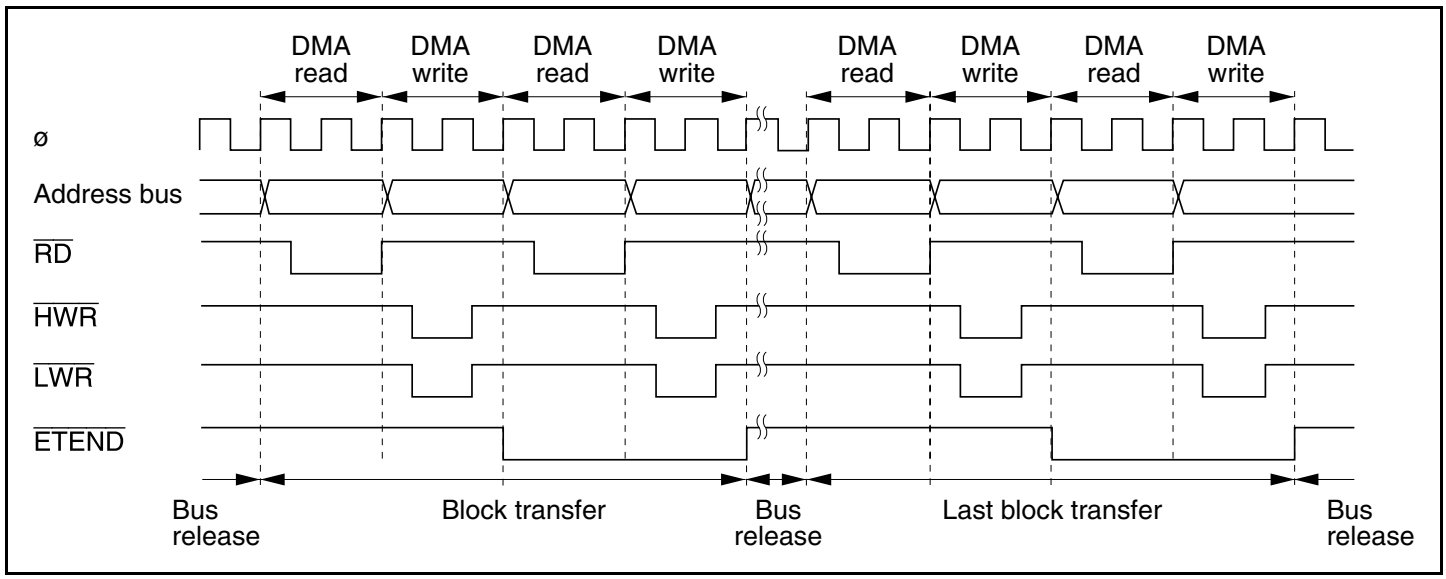
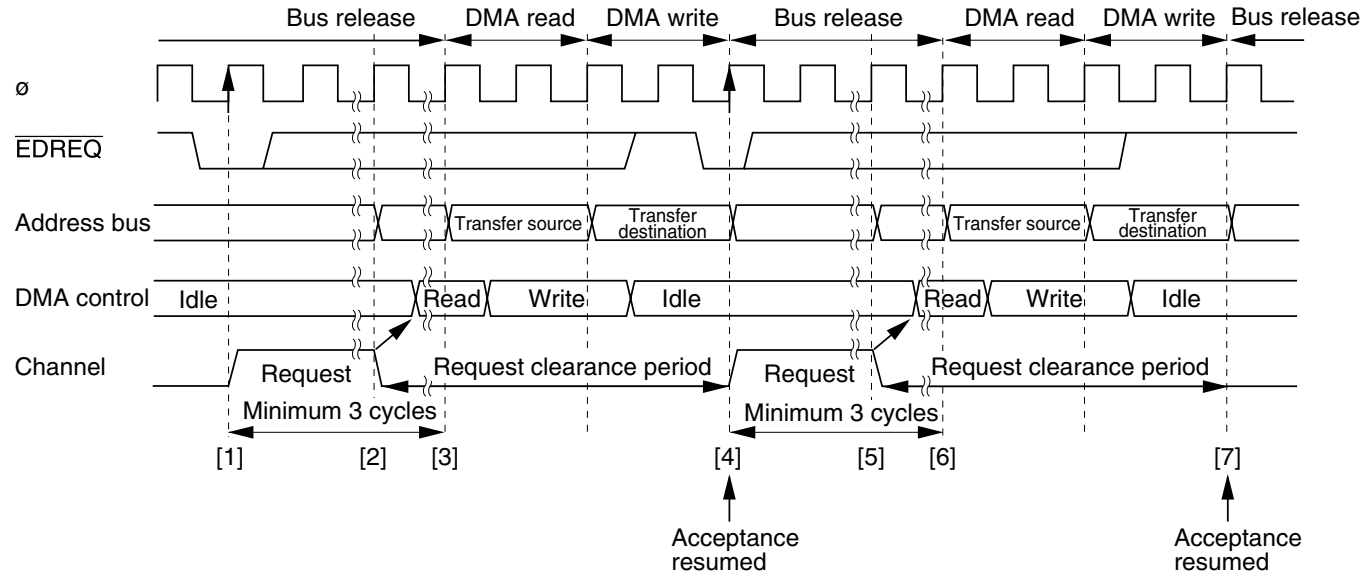


Figure 8.17 Example of Block Transfer Mode (Cycle Steal Mode) Transfer

EDREQ Pin Falling Edge Activation Timing: Figure 8.18 shows an example of normal mode transfer activated by the EDREQ pin falling edge.



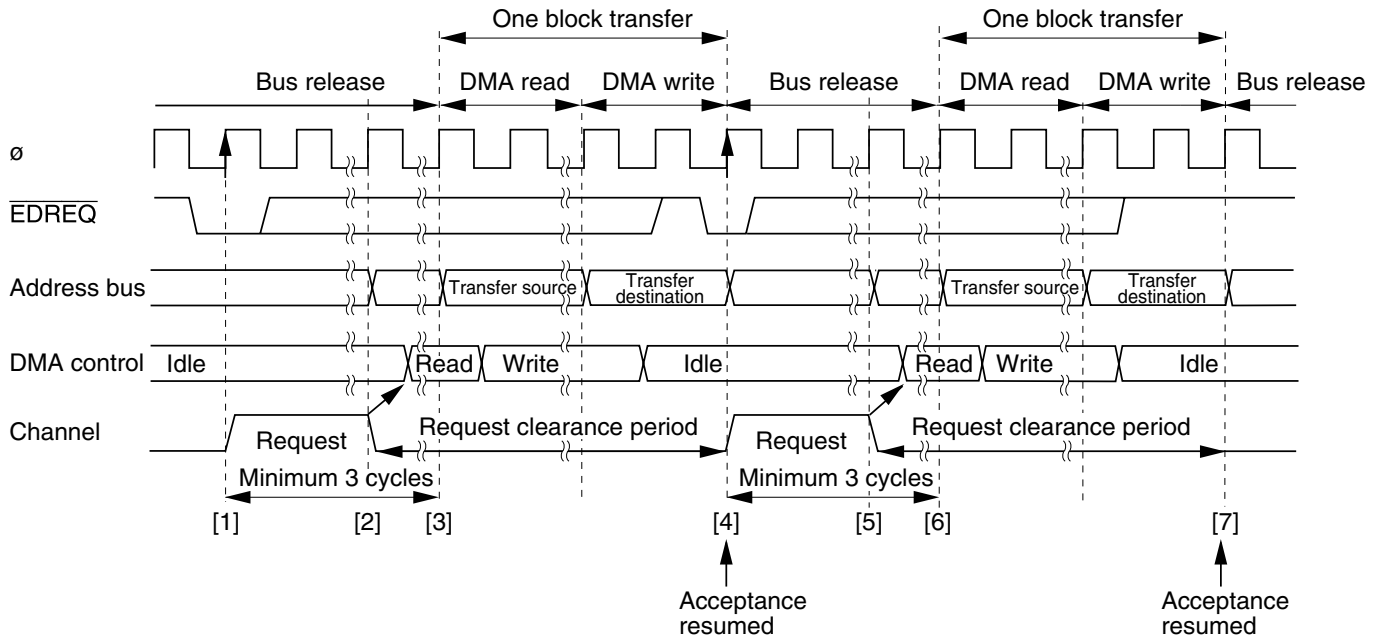
[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ø, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] DMA cycle start; $\overline{\text{EDREQ}}$ pin high level sampling is started at rise of ø.
[4], [7] When $\overline{\text{EDREQ}}$ pin high level has been sampled, acceptance is resumed after completion of write cycle. (As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ø, and request is held.)

Figure 8.18 Example of Normal Mode Transfer Activated by EDREQ Pin Falling Edge

EDREQ pin sampling is performed in each cycle starting at the next rise of ø after the end of the EDMDR write cycle for setting the transfer-enabled state.

When a low level is sampled at the EDREQ pin while acceptance via the EDREQ pin is possible, the request is held within the EXDMAC. Then when activation is initiated within the EXDMAC, the request is cleared, and EDREQ pin high level sampling for edge sensing is started. If EDREQ pin high level sampling is completed by the end of the DMA write cycle, acceptance resumes after the end of the write cycle, and EDREQ pin low level sampling is performed again; this sequence of operations is repeated until the end of the transfer.

Figure 8.19 shows an example of block transfer mode transfer activated by the EDREQ pin falling edge.



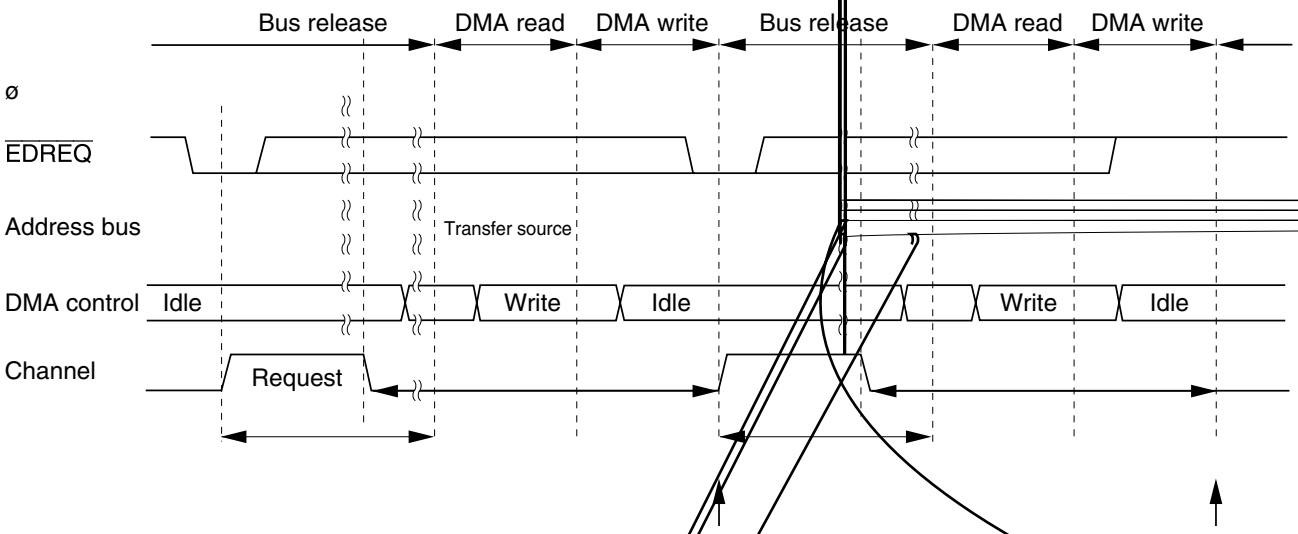
[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ø, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] DMA cycle start; $\overline{\text{EDREQ}}$ pin high level sampling is started at rise of ø.
[4], [7] When $\overline{\text{EDREQ}}$ pin high level has been sampled, acceptance is resumed after completion of dead cycle. (As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ø, and request is held.)

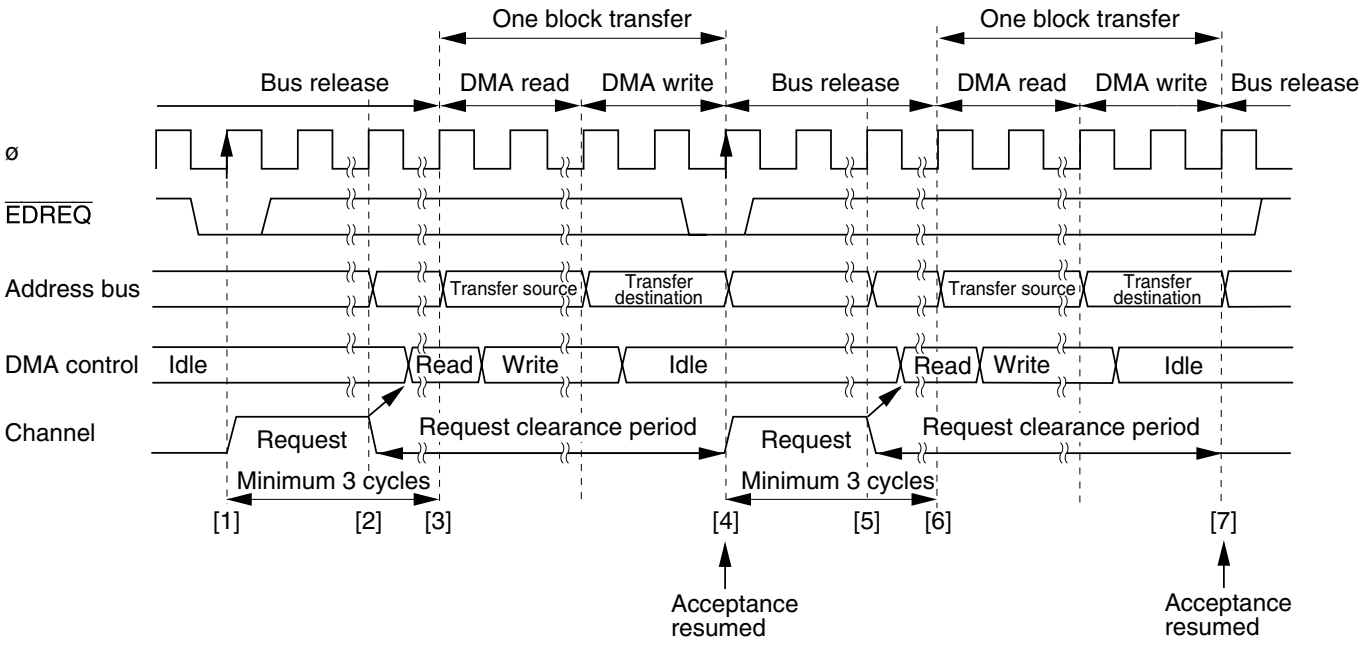
Figure 8.19 Example of Block Transfer Mode Transfer Activated by EDREQ Pin Falling Edge

EDREQ pin sampling is performed in each cycle starting at the next rise of ø after the end of the EDMDR write cycle for setting the transfer-enabled state.

When a low level is sampled at the EDREQ pin while acceptance via the EDREQ pin is possible, the request is held within the EXDMAC. Then when activation is initiated within the EXDMAC, the request is cleared, and EDREQ pin high level sampling for edge sensing is started. If EDREQ pin high level sampling is completed by the end of the DMA write cycle, acceptance resumes after the end of the write cycle, and EDREQ pin low level sampling is performed again; this sequence of operations is repeated until the end of the transfer.

EDREQ Pin Low Level Activation Timing: Figure 8.20 shows an example of normal mode transfer activated by the EDREQ pin low level.


Bus release
DMA read
DMA write
Bus release
DMA read
DMA write
ø
EDREQ
Address bus
Transfer source
DMA control
Idle
Write
Idle
Write
Idle
Channel
Request



[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ø, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] DMA cycle is started.
[4], [7] Acceptance is resumed after completion of dead cycle.
(As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ø, and request is held.)

Figure 8.21 Example of Block Transfer Mode Transfer Activated by EDREQ Pin Low Level

EDREQ pin sampling is performed in each cycle starting at the next rise of ø after the end of the EDMDR write cycle for setting the transfer-enabled state.

When a low level is sampled at the EDREQ pin while acceptance via the EDREQ pin is possible, the request is held within the EXDMAC. Then when activation is initiated within the EXDMAC, the request is cleared. At the end of the write cycle, acceptance resumes and EDREQ pin low level sampling is performed again; this sequence of operations is repeated until the end of the transfer.

8.4.10 EXDMAC Bus Cycles (Single Address Mode)

Single Address Mode (Read): Figure 8.22 shows an example of transfer when ETEND output is enabled, and byte-size, single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.

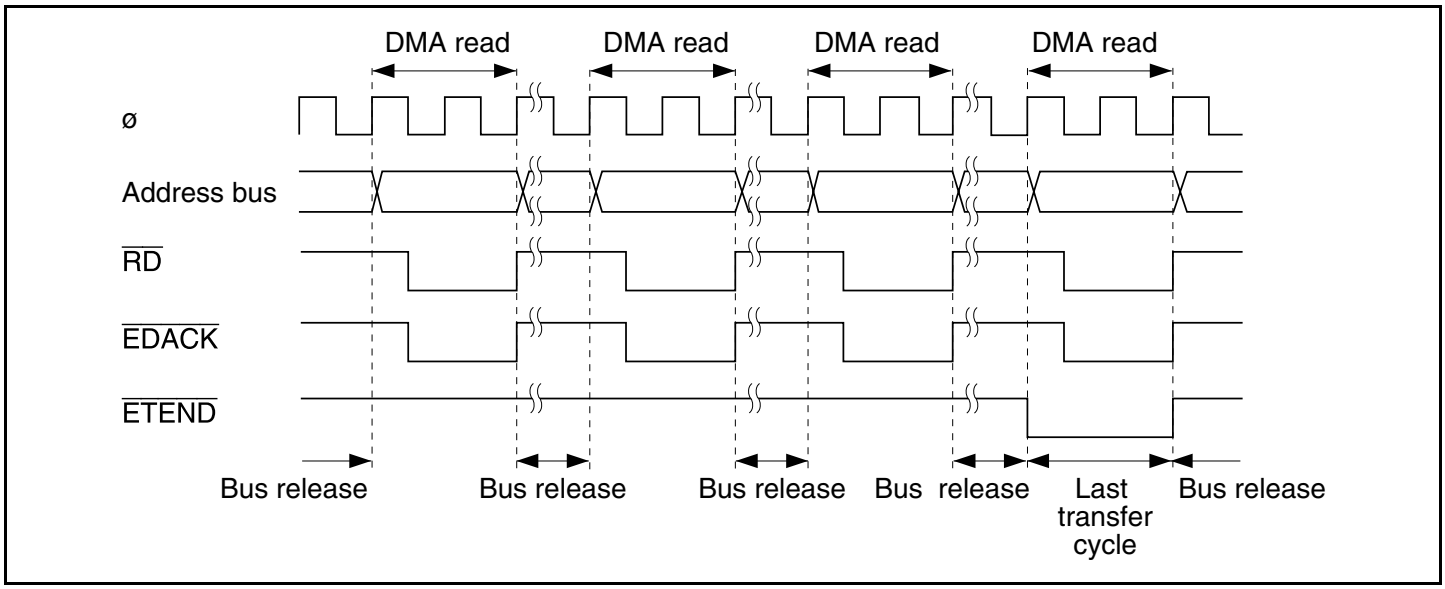


Figure 8.22 Example of Single Address Mode (Byte Read) Transfer

Figure 8.23 shows an example of transfer when ETEND output is enabled, and word-size, single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.

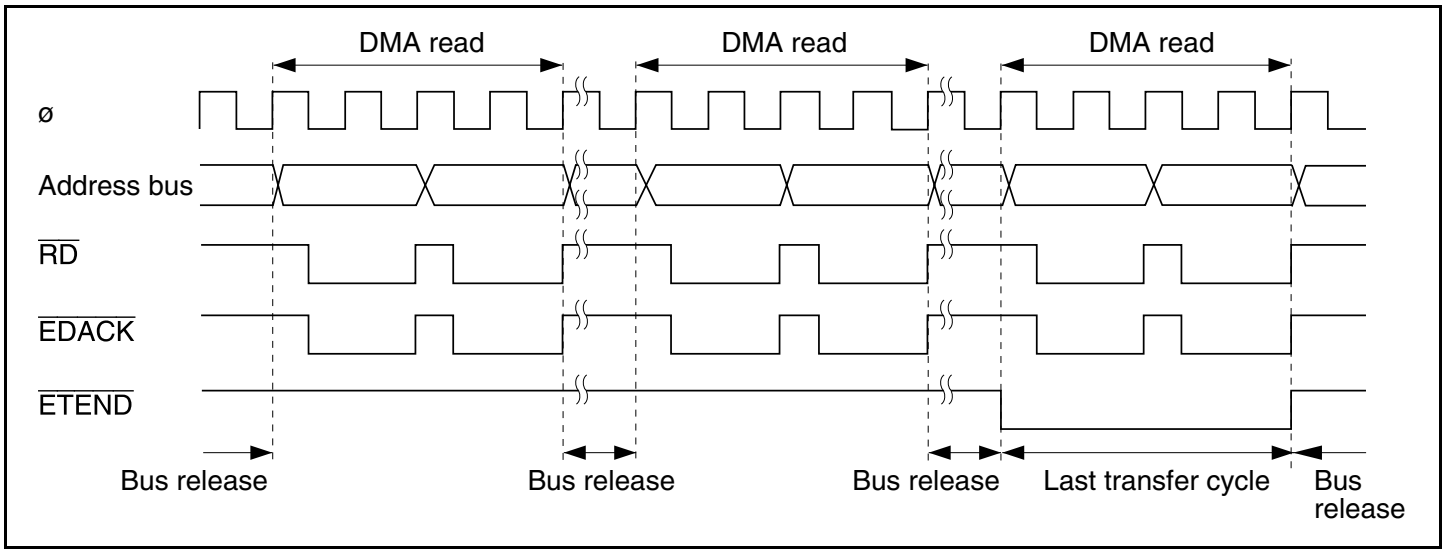


Figure 8.23 Example of Single Address Mode (Word Read) Transfer

After one byte or word has been transferred in response to one transfer request, the bus is released. While the bus is released, one or more CPU, DMAC, or DTC bus cycles are initiated.

Single Address Mode (Write): Figure 8.24 shows an example of transfer when ETEND output is enabled, and byte-size, single address mode transfer (write) is performed from an external device to external 8-bit, 2-state access space.

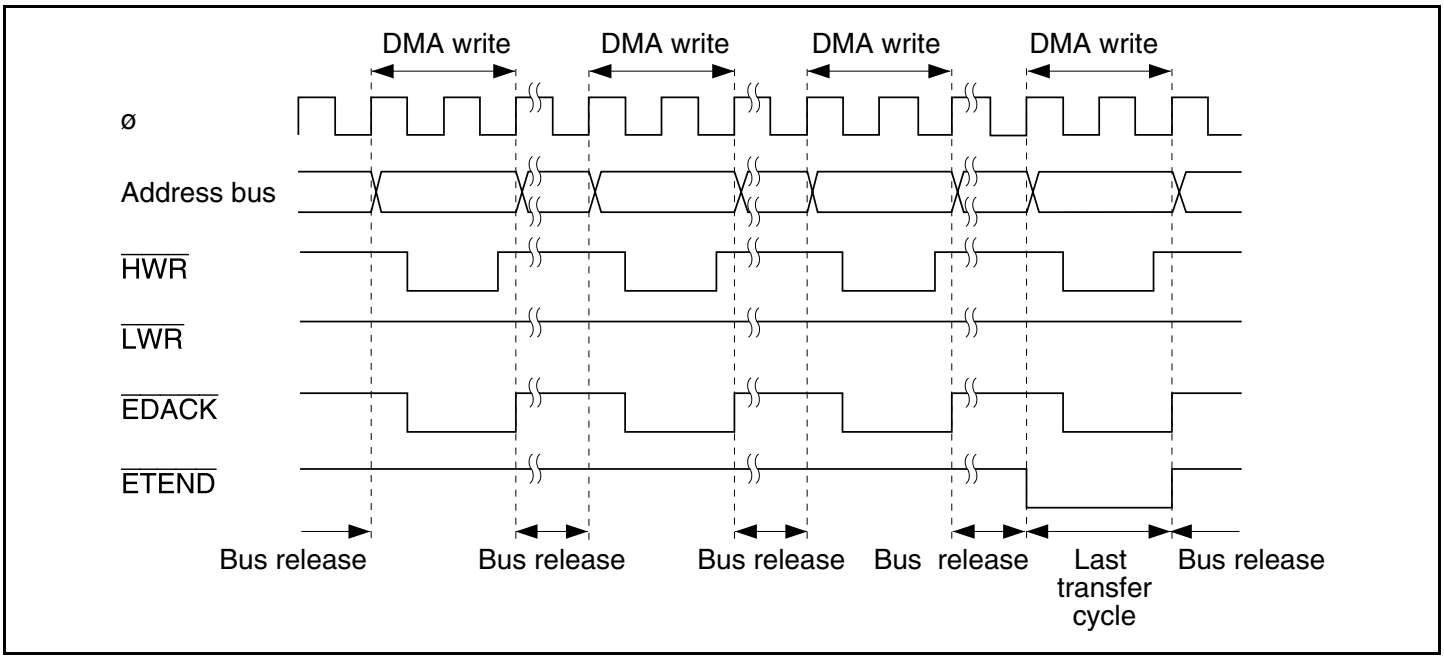


Figure 8.24 Example of Single Address Mode (Byte Write) Transfer

Figure 8.25 shows an example of transfer when ETEND output is enabled, and word-size, single address mode transfer (write) is performed from an external device to external 8-bit, 2-state access space.

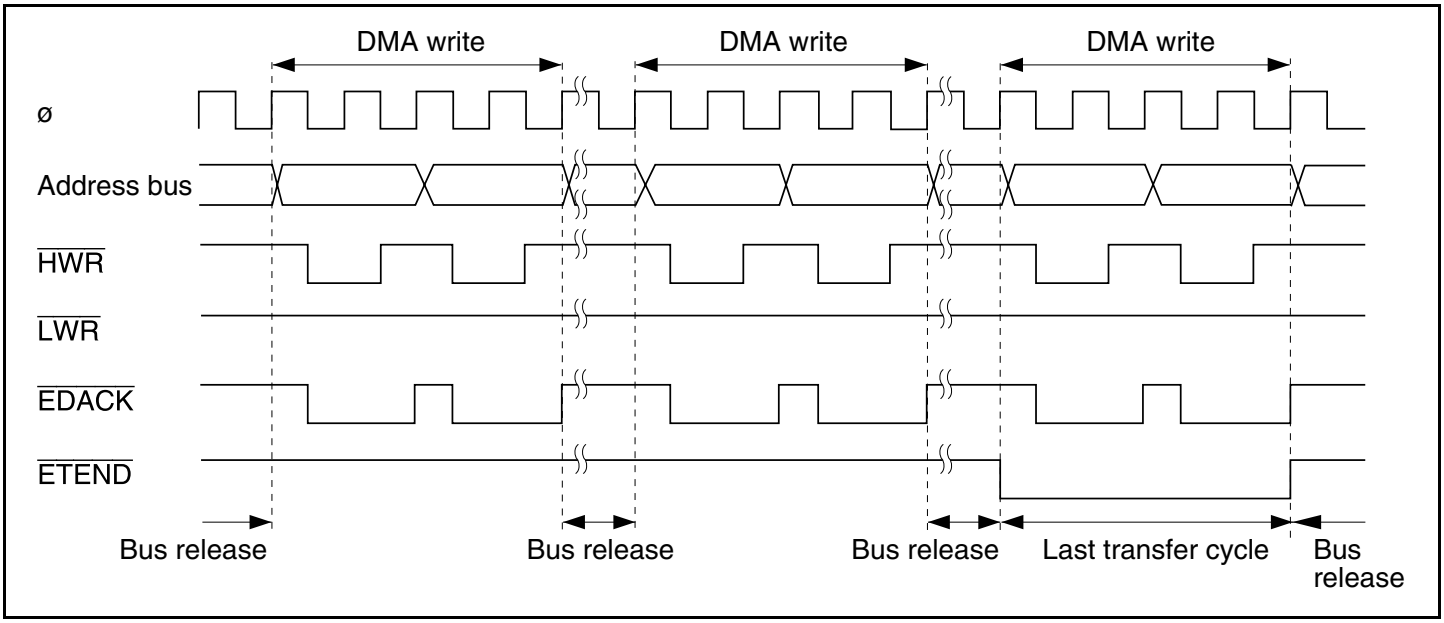
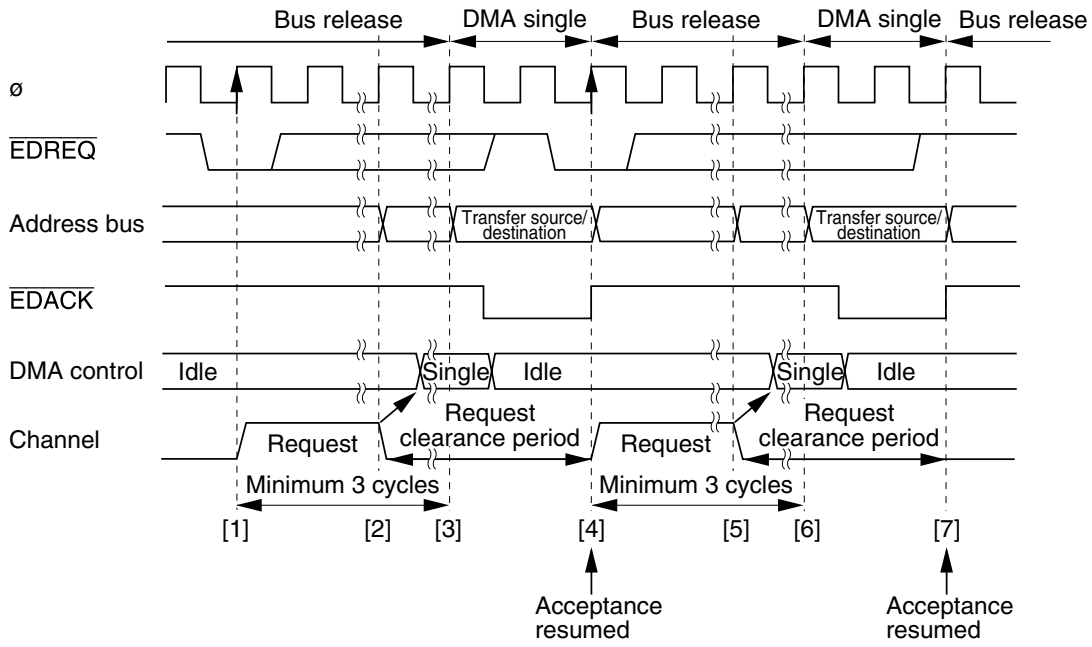


Figure 8.25 Example of Single Address Mode (Word Write) Transfer

After one byte or word has been transferred in response to one transfer request, the bus is released. While the bus is released, one or more CPU, DMAC, or DTC bus cycles are initiated.

EDREQ Pin Falling Edge Activation Timing: Figure 8.26 shows an example of single address mode transfer activated by the EDREQ pin falling edge.



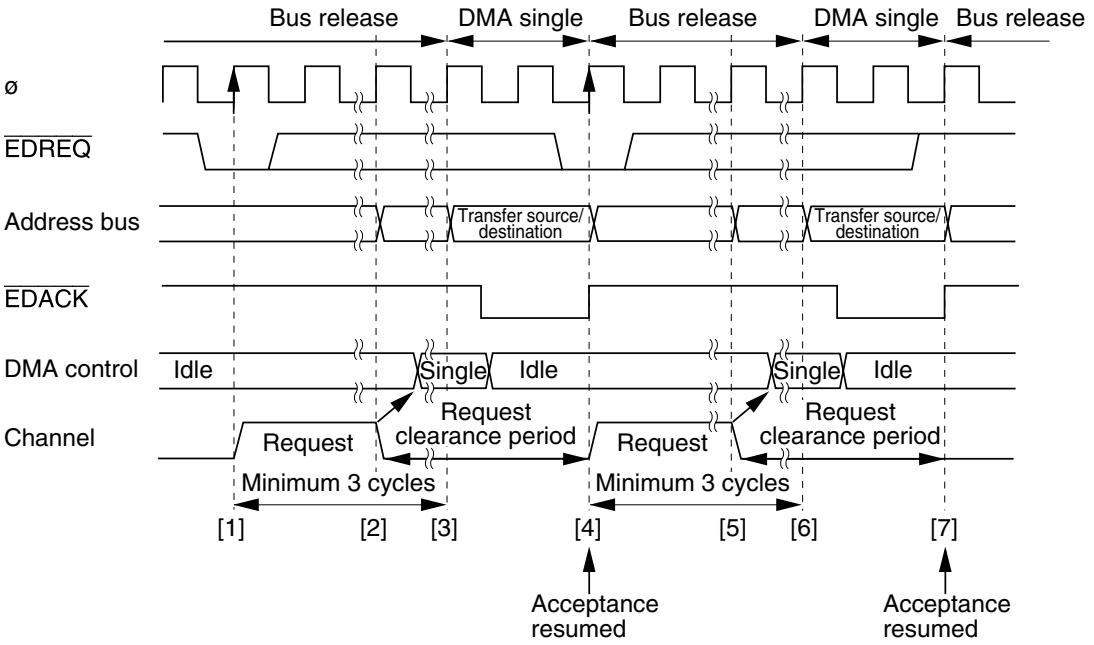
[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ø, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] DMA cycle start; $\overline{\text{EDREQ}}$ pin high level sampling is started at rise of ø.
[4], [7] When $\overline{\text{EDREQ}}$ pin high level has been sampled, acceptance is resumed after completion of single cycle. (As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ø, and request is held.)

Figure 8.26 Example of Single Address Mode Transfer Activated by EDREQ Pin Falling Edge

EDREQ pin sampling is performed in each cycle starting at the next rise of ø after the end of the EDMDR write cycle for setting the transfer-enabled state.

When a low level is sampled at the EDREQ pin while acceptance via the EDREQ pin is possible, the request is held within the EXDMAC. Then when activation is initiated within the EXDMAC, the request is cleared, and EDREQ pin high level sampling for edge sensing is started. If EDREQ pin high level sampling is completed by the end of the DMA single cycle, acceptance resumes after the end of the single cycle, and EDREQ pin low level sampling is performed again; this sequence of operations is repeated until the end of the transfer.

EDREQ Pin Low Level Activation Timing: Figure 8.27 shows an example of single address mode transfer activated by the EDREQ pin low level.



[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ø, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] DMA cycle is started.
[4], [7] Acceptance is resumed after completion of single cycle.
(As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ø, and request is held.)

Figure 8.27 Example of Single Address Mode Transfer Activated by EDREQ Pin Low Level

EDREQ pin sampling is performed in each cycle starting at the next rise of ø after the end of the EDMDR write cycle for setting the transfer-enabled state.

When a low level is sampled at the EDREQ pin while acceptance via the EDREQ pin is possible, the request is held within the EXDMAC. Then when activation is initiated within the EXDMAC, the request is cleared. At the end of the single cycle, acceptance resumes and EDREQ pin low level sampling is performed again; this sequence of operations is repeated until the end of the transfer.

8.4.11 Examples of Operation Timing in Each Mode

Auto Request/Cycle Steal Mode/Normal Transfer Mode: When the EDA bit is set to 1 in EDMDR, an EXDMA transfer cycle is started a minimum of three cycles later. There is a one-cycle bus release interval between the end of a one-transfer-unit EXDMA cycle and the start of the next transfer.

If there is a transfer request for another channel of higher priority, the transfer request by the original channel is held pending, and transfer is performed on the higher-priority channel from the next transfer. Transfer on the original channel is resumed on completion of the higher-priority channel transfer.

Figures 8.28 to 8.30 show operation timing examples for various conditions.

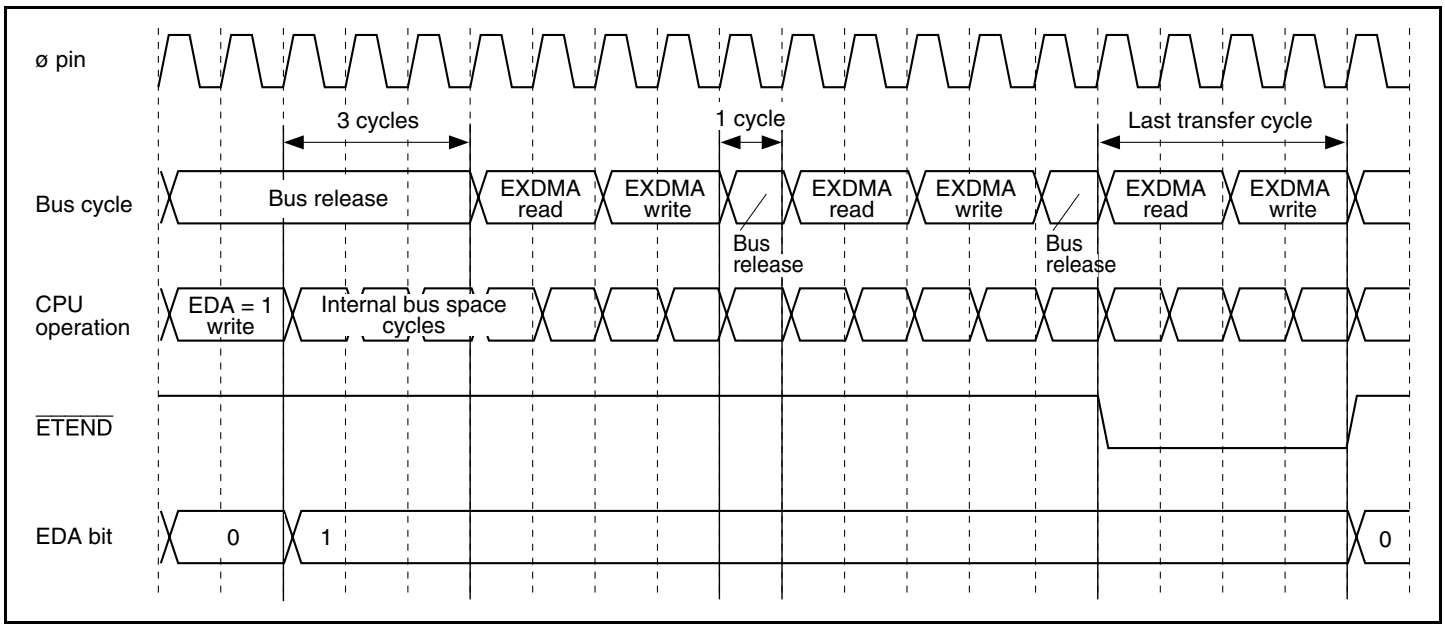


Figure 8.28 Auto Request/Cycle Steal Mode/Normal Transfer Mode (No Contention/Dual Address Mode)

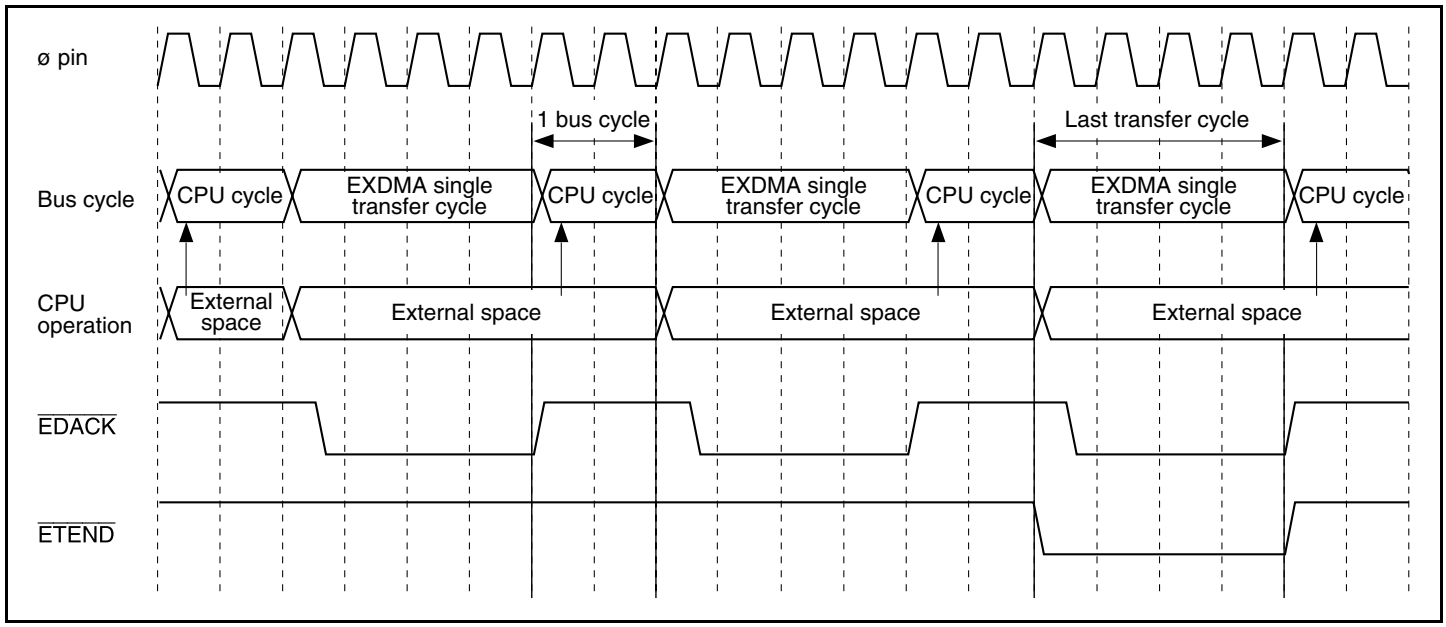


Figure 8.29 Auto Request/Cycle Steal Mode/Normal Transfer Mode (CPU Cycles/Single Address Mode)

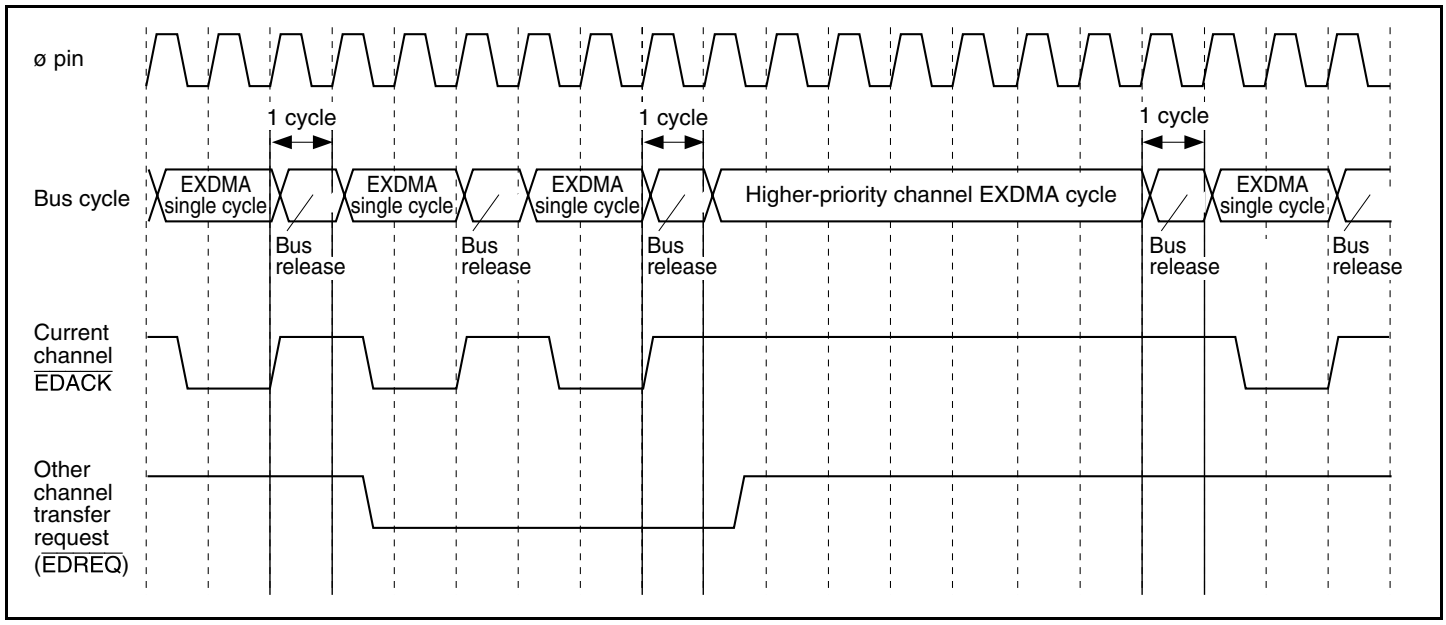


Figure 8.30 Auto Request/Cycle Steal Mode/Normal Transfer Mode (Contention with Another Channel/Single Address Mode)

Auto Request/Burst Mode/Normal Transfer Mode: When the EDA bit is set to 1 in EDMDR, an EXDMA transfer cycle is started a minimum of three cycles later. Once transfer is started, it continues (as a burst) until the transfer end condition is satisfied.

If the BGUP bit is 1 in EDMDR, the bus is transferred in the event of a bus request from another bus master.

Transfer requests for other channels are held pending until the end of transfer on the current channel.

Figures 8.31 to 8.34 show operation timing examples for various conditions.

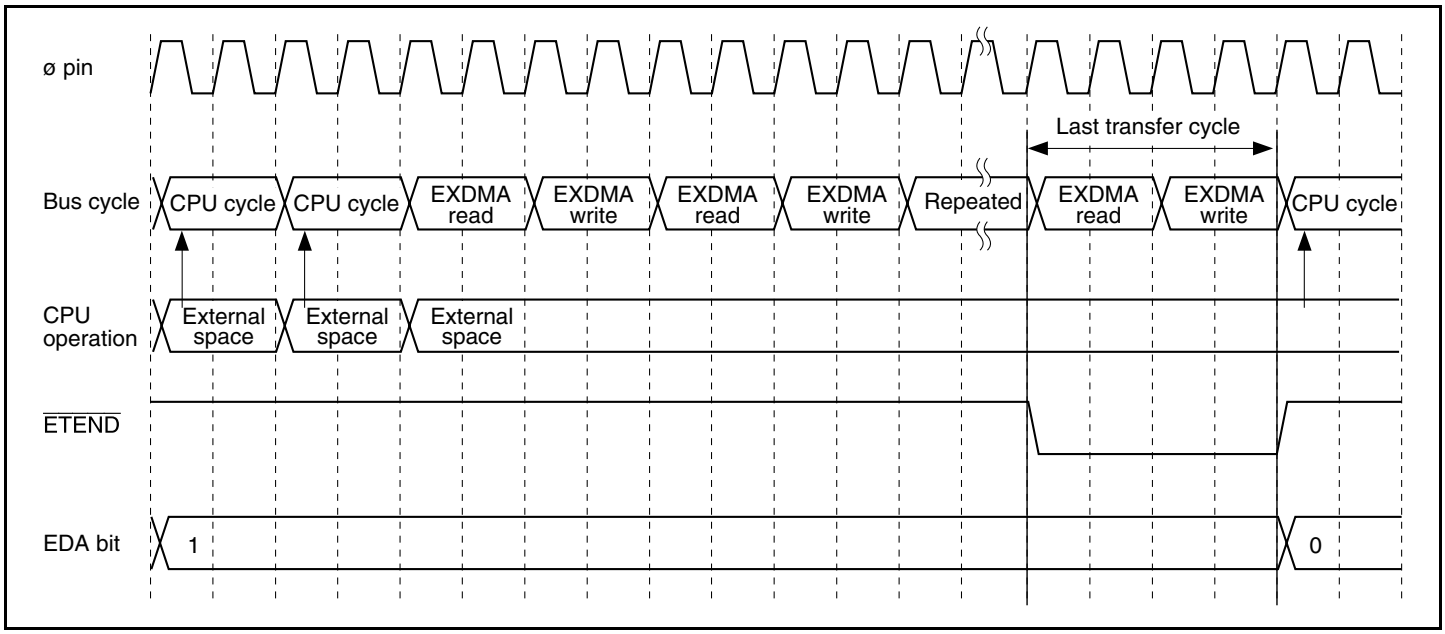


Figure 8.31 Auto Request/Burst Mode/Normal Transfer Mode (CPU Cycles/Dual Address Mode/BGUP = 0)

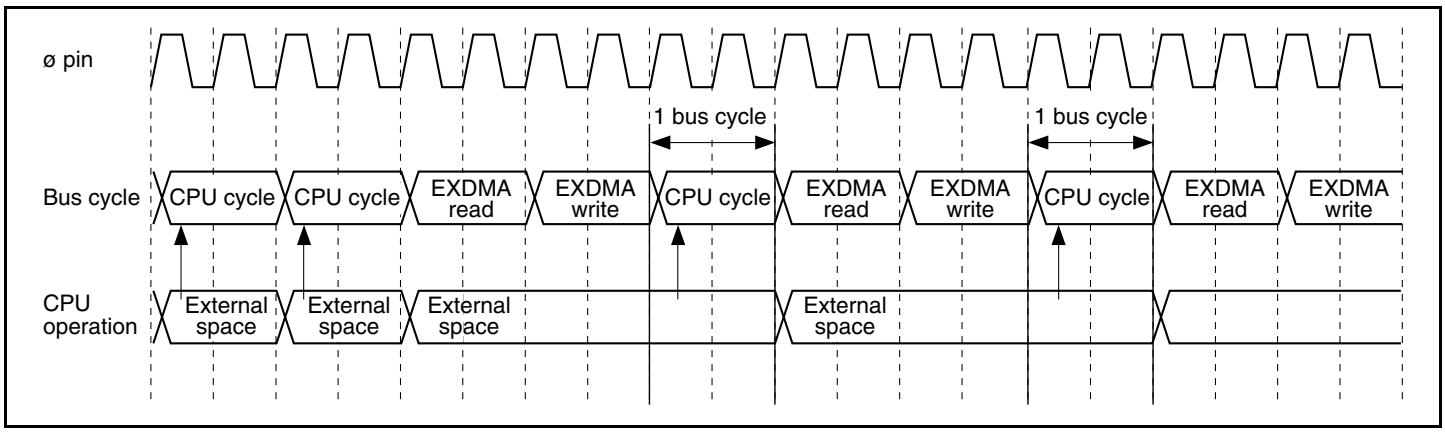


Figure 8.32 Auto Request/Burst Mode/Normal Transfer Mode (CPU Cycles/Dual Address Mode/BGUP = 1)

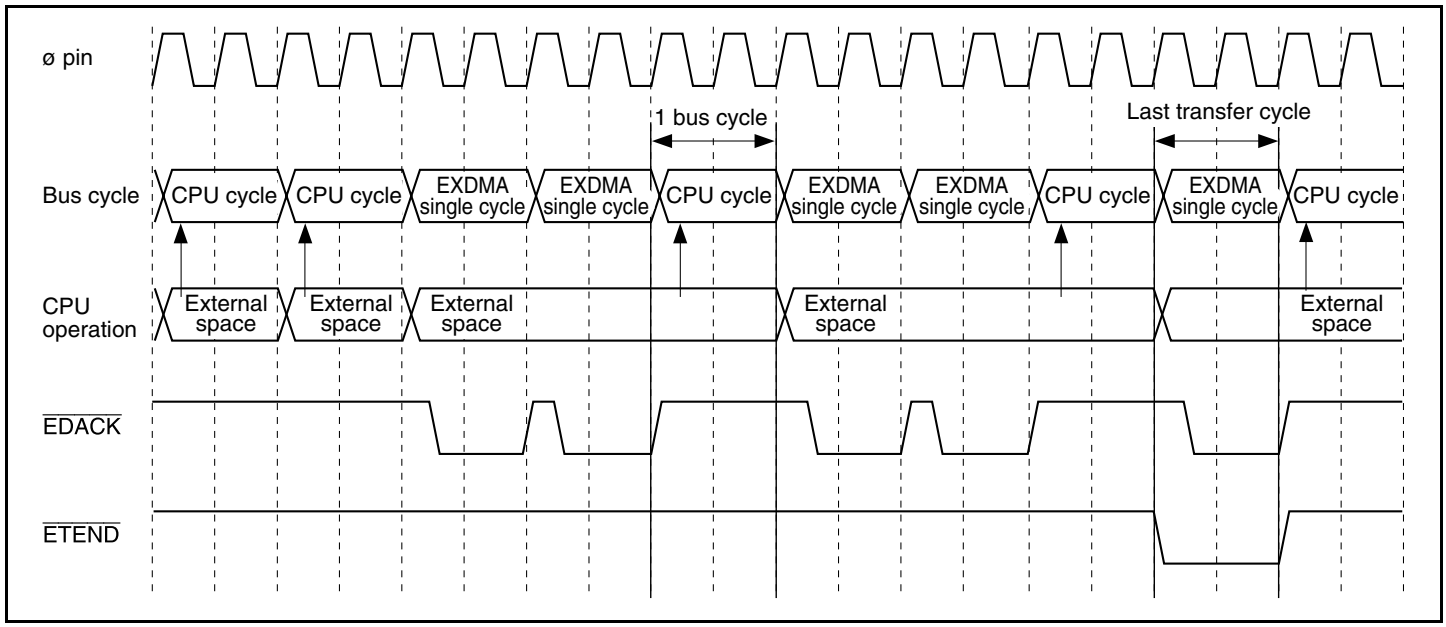


Figure 8.33 Auto Request/Burst Mode/Normal Transfer Mode (CPU Cycles/Single Address Mode/BGUP = 1)

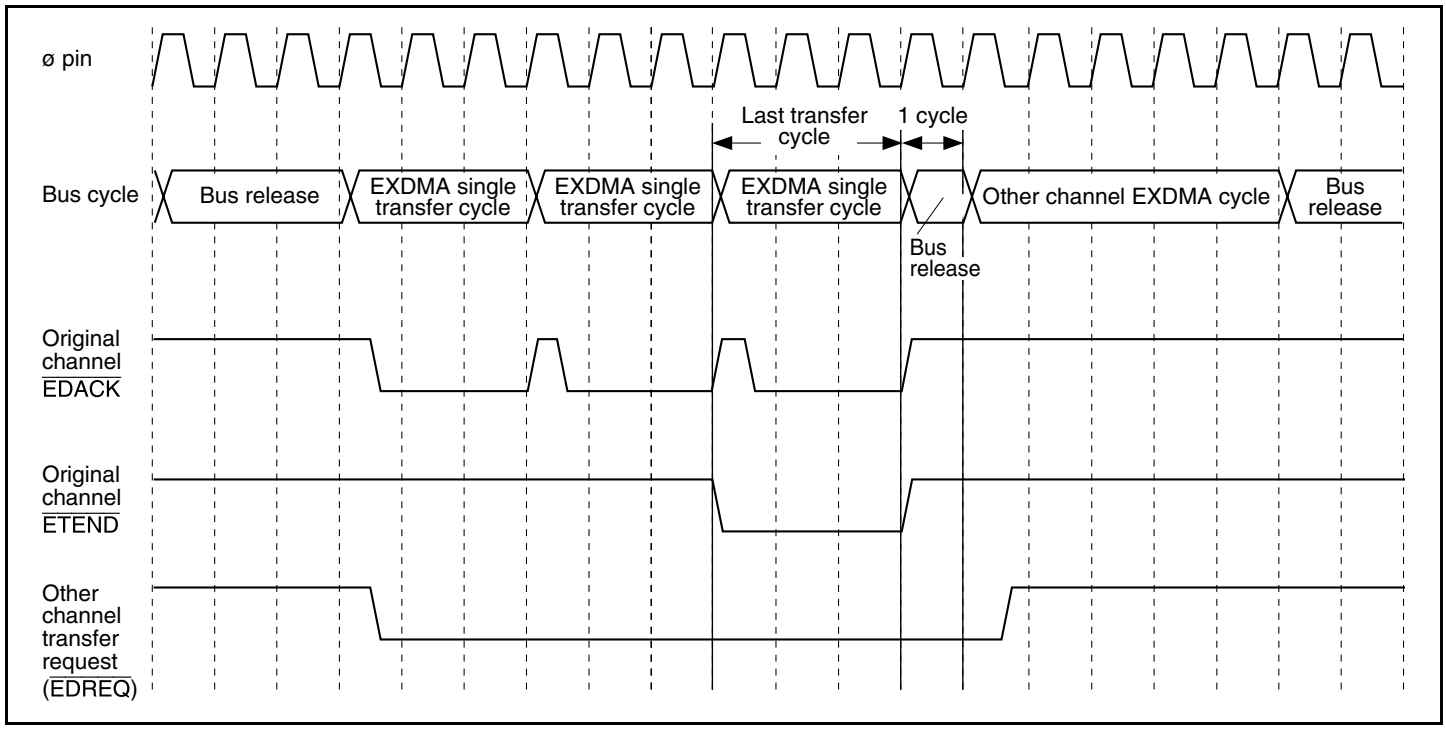


Figure 8.34 Auto Request/Burst Mode/Normal Transfer Mode (Contention with Another Channel/Single Address Mode)

External Request/Cycle Steal Mode/Normal Transfer Mode: In external request mode, an EXDMA transfer cycle is started a minimum of three cycles after a transfer request is accepted. The next transfer request is accepted after the end of a one-transfer-unit EXDMA cycle. For external bus space CPU cycles, at least two bus cycles are generated before the next EXDMA cycle.

If a transfer request is generated for another channel, an EXDMA cycle for the other channel is generated before the next EXDMA cycle.

The EDREQ pin sensing timing is different for low level sensing and falling edge sensing. The same applies to transfer request acceptance and transfer start timing.

Figures 8.35 to 8.38 show operation timing examples for various conditions.

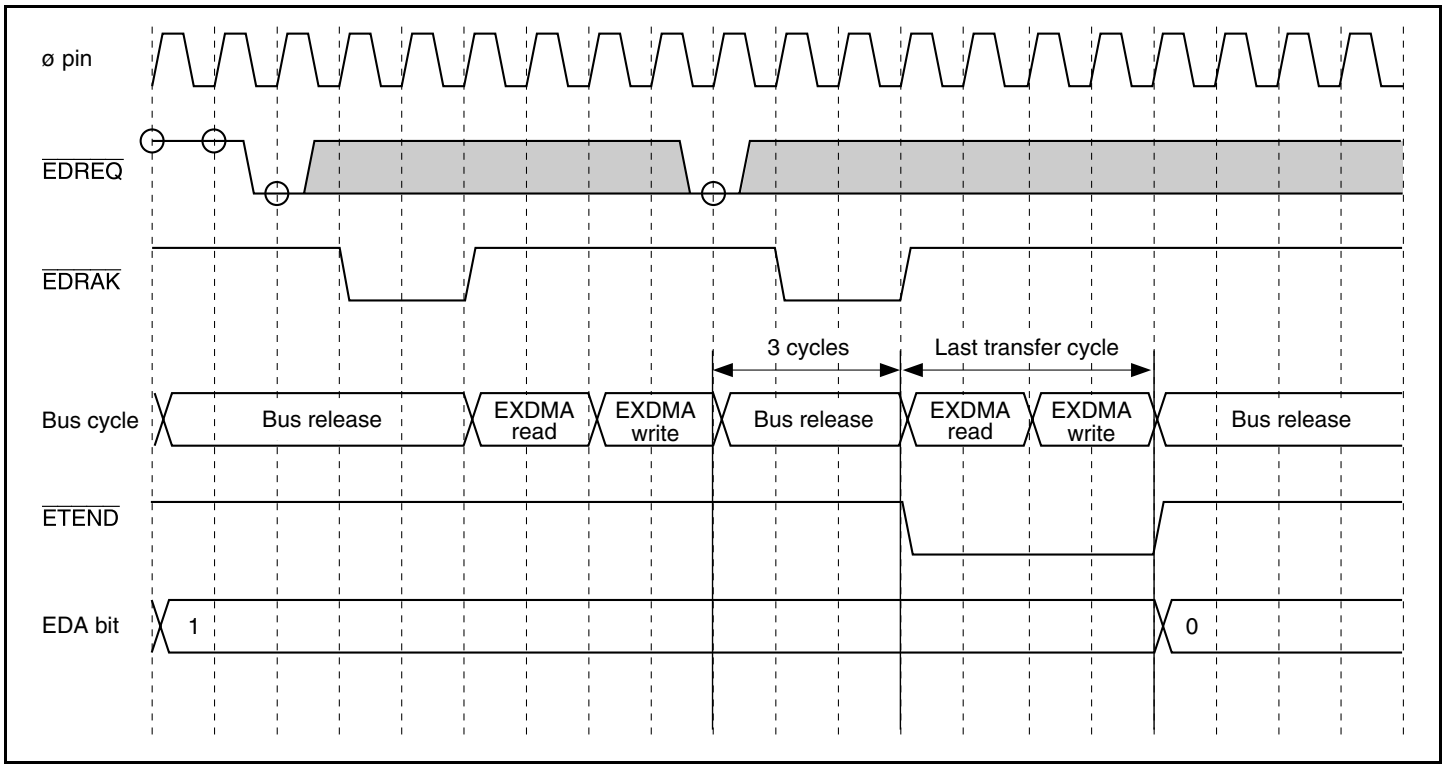


Figure 8.35 External Request/Cycle Steal Mode/Normal Transfer Mode (No Contention/Dual Address Mode/Low Level Sensing)

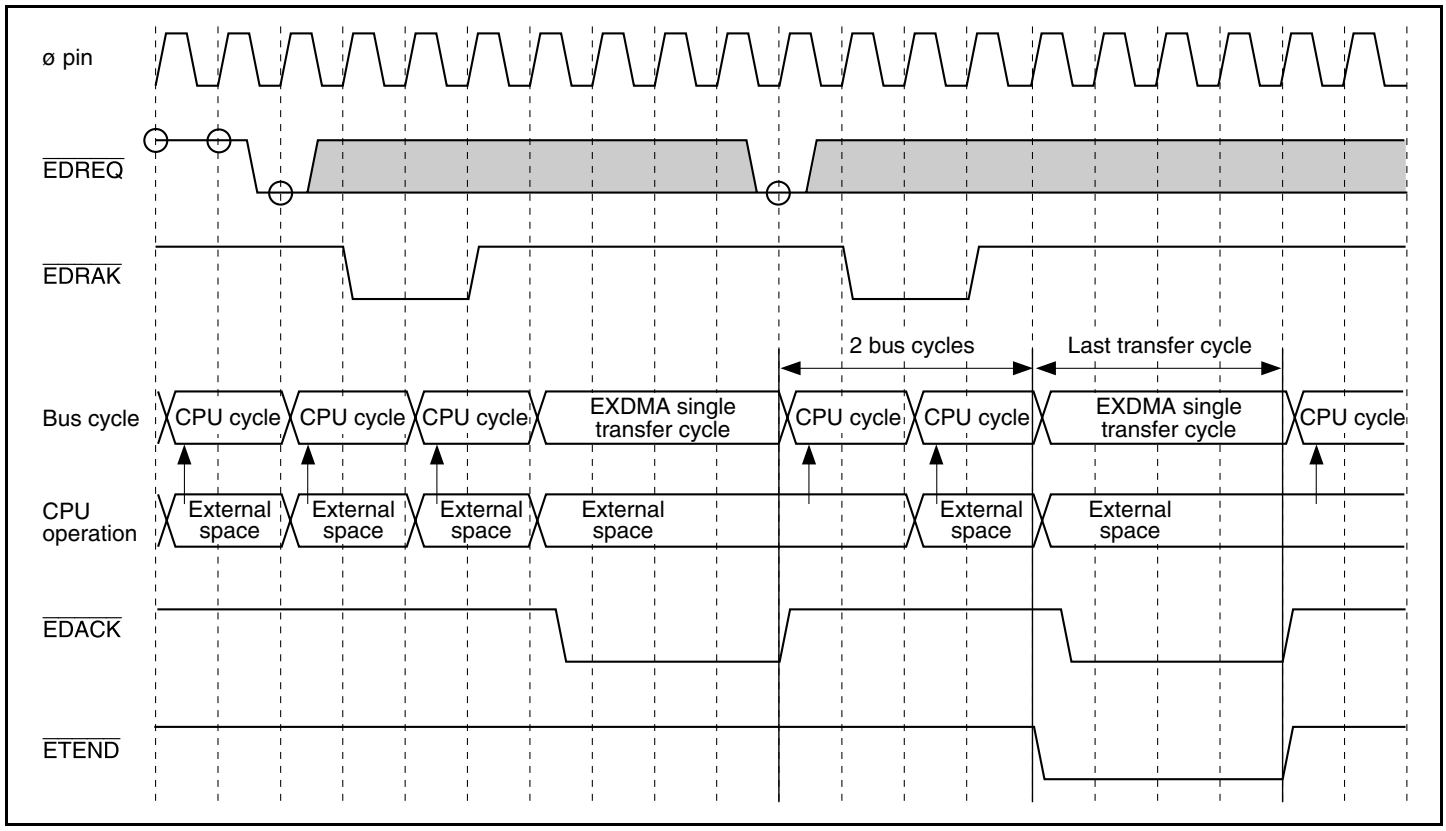


Figure 8.36 External Request/Cycle Steal Mode/Normal Transfer Mode (CPU Cycles/Single Address Mode/Low Level Sensing)

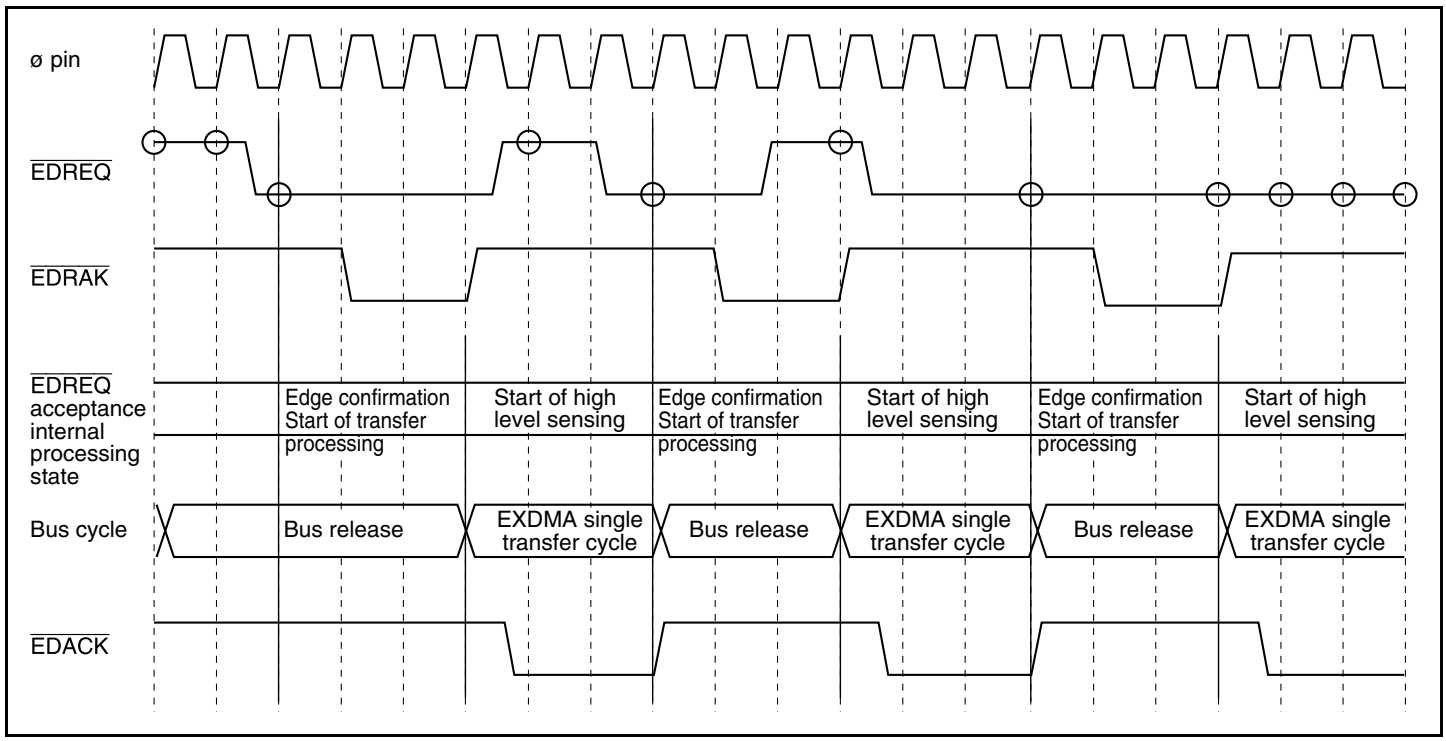
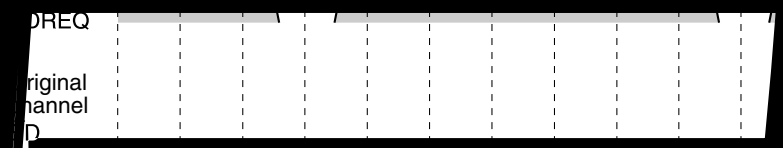


Figure 8.37 External Request/Cycle Steal Mode/Normal Transfer Mode (No Contention/Single Address Mode/Falling Edge Sensing)



HITACHI

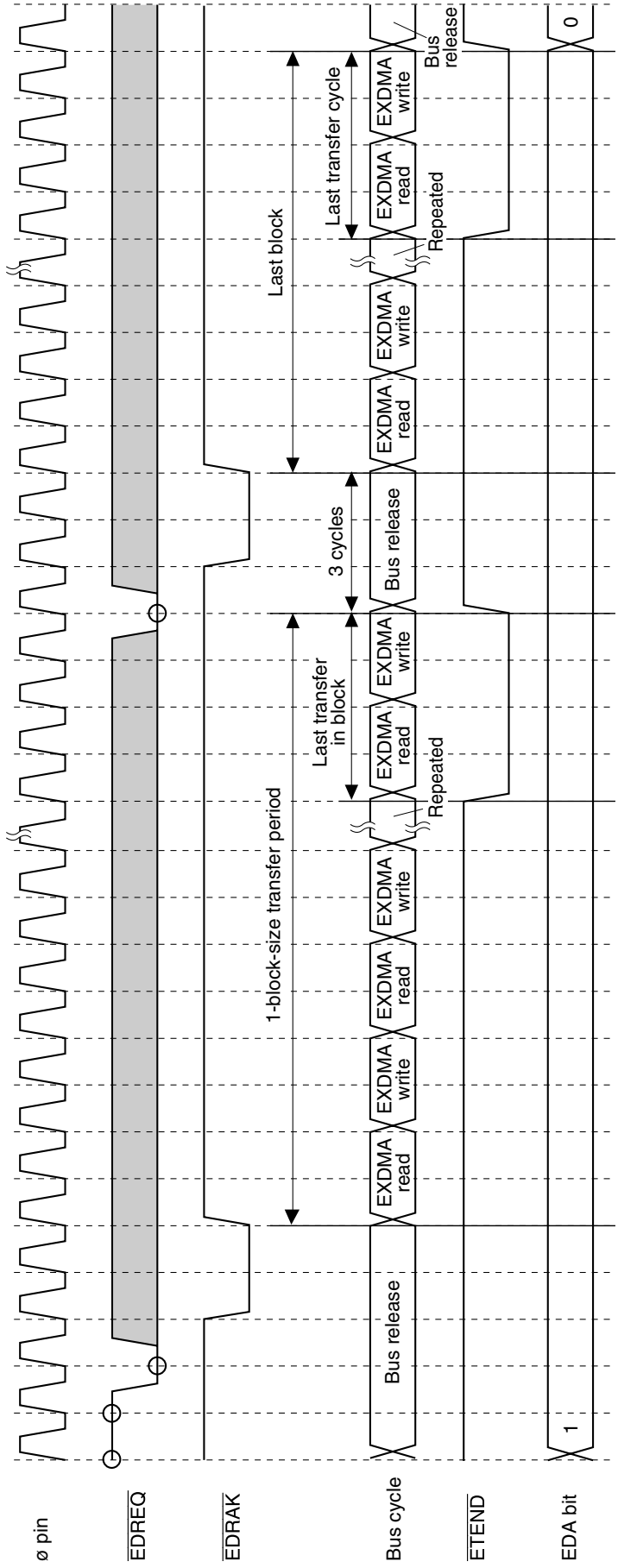


Figure 8.39 External Request/Cycle Steal Mode/Block Transfer Mode (No Contention/Dual Address Mode/Low Level Sensing/BGUP = 0)

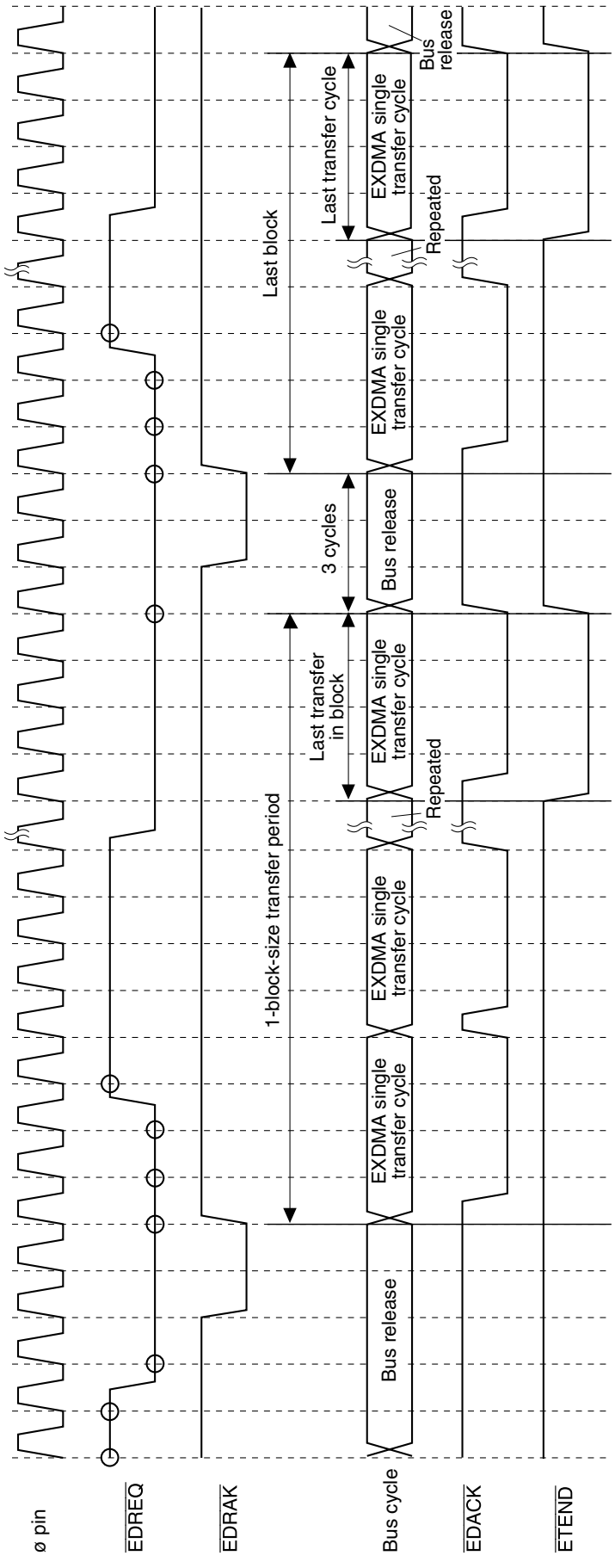


Figure 8.40 External Request/Cycle Steal Mode/Block Transfer Mode (No Contention/Single Address Mode/Falling Edge Sensing/BGUP = 0)

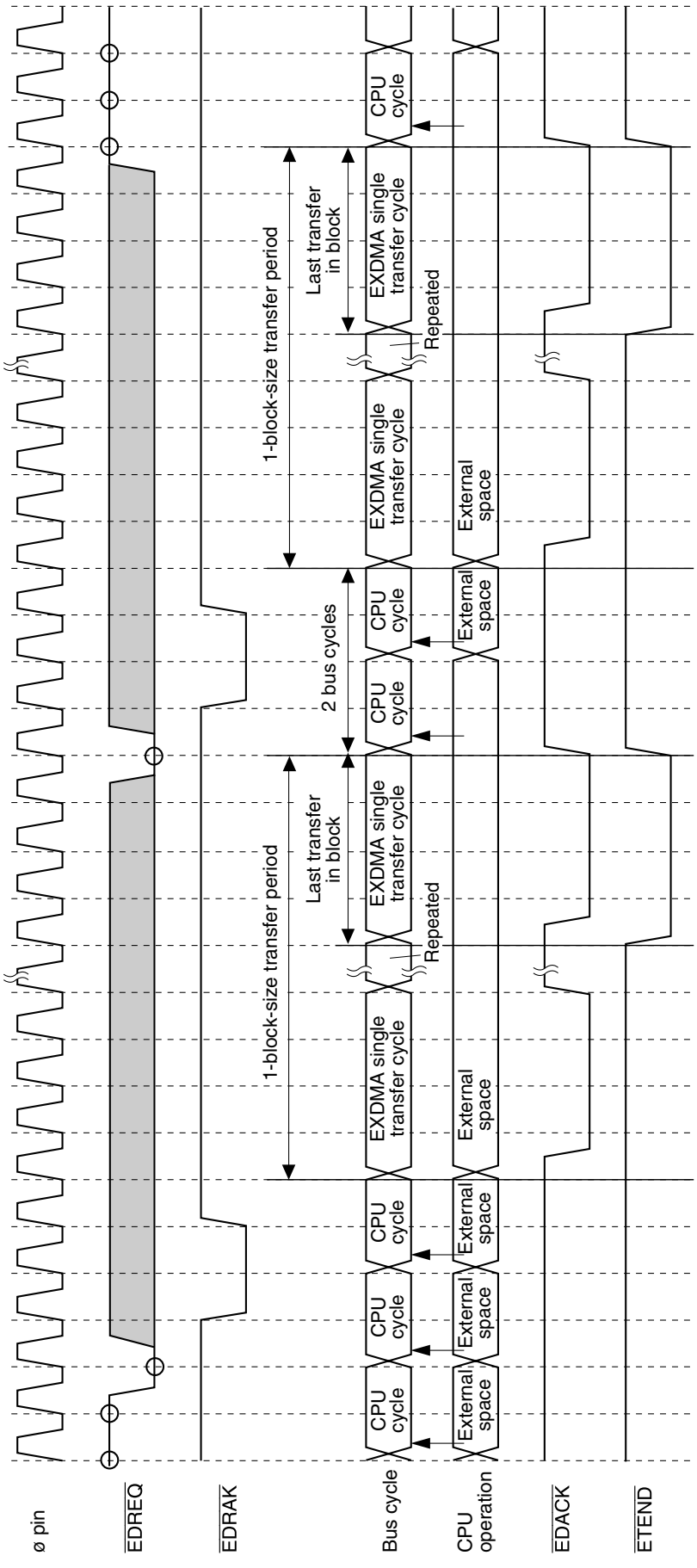


Figure 8.41 External Request/Cycle Steal Mode/Block Transfer Mode (CPU Cycles/Single Address Mode/Low Level Sensing/BGUP = 0)

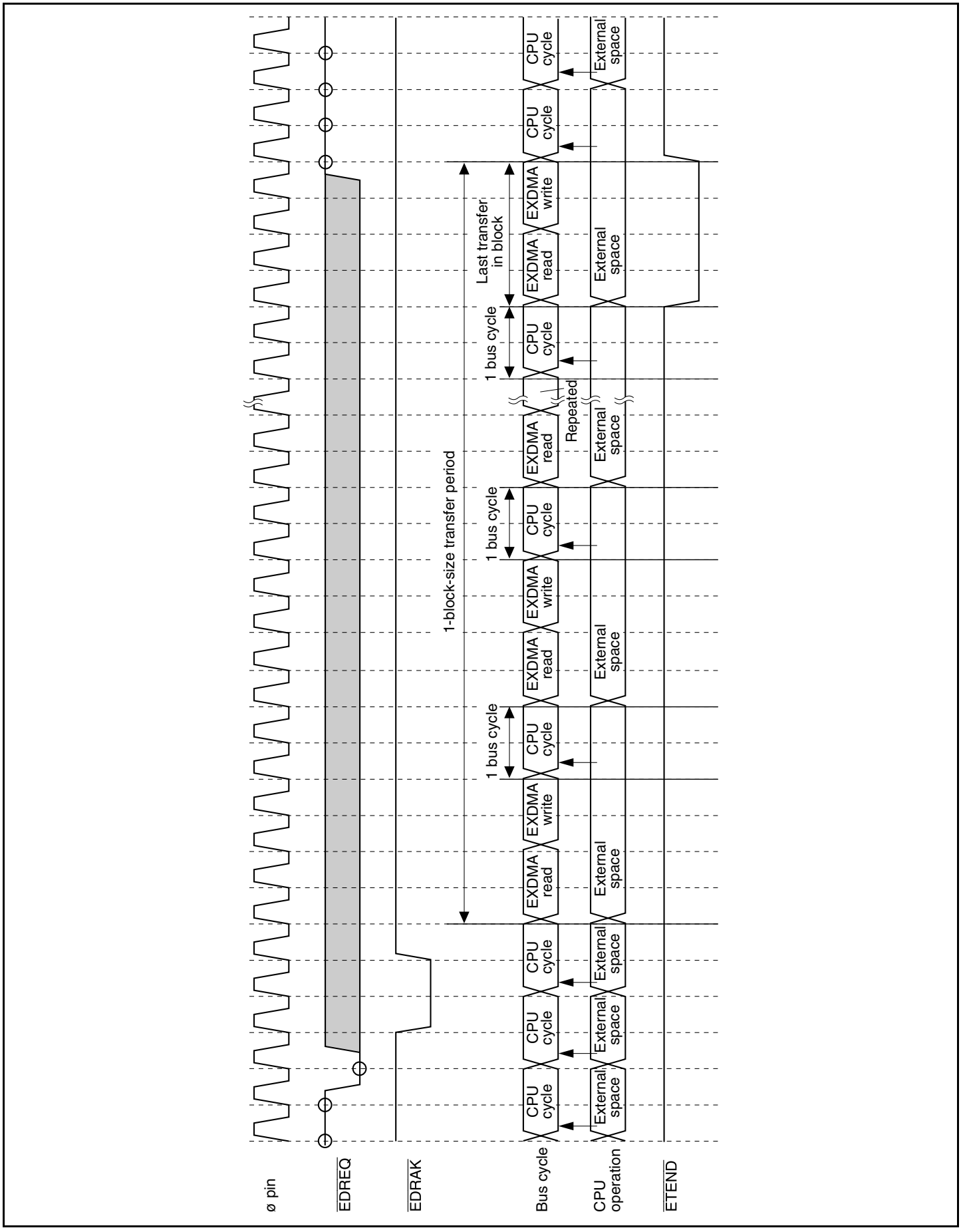


Figure 8.42 External Request/Cycle Steal Mode/Block Transfer Mode (CPU Cycles/Dual Address Mode/Low Level Sensing/BGUP = 1)

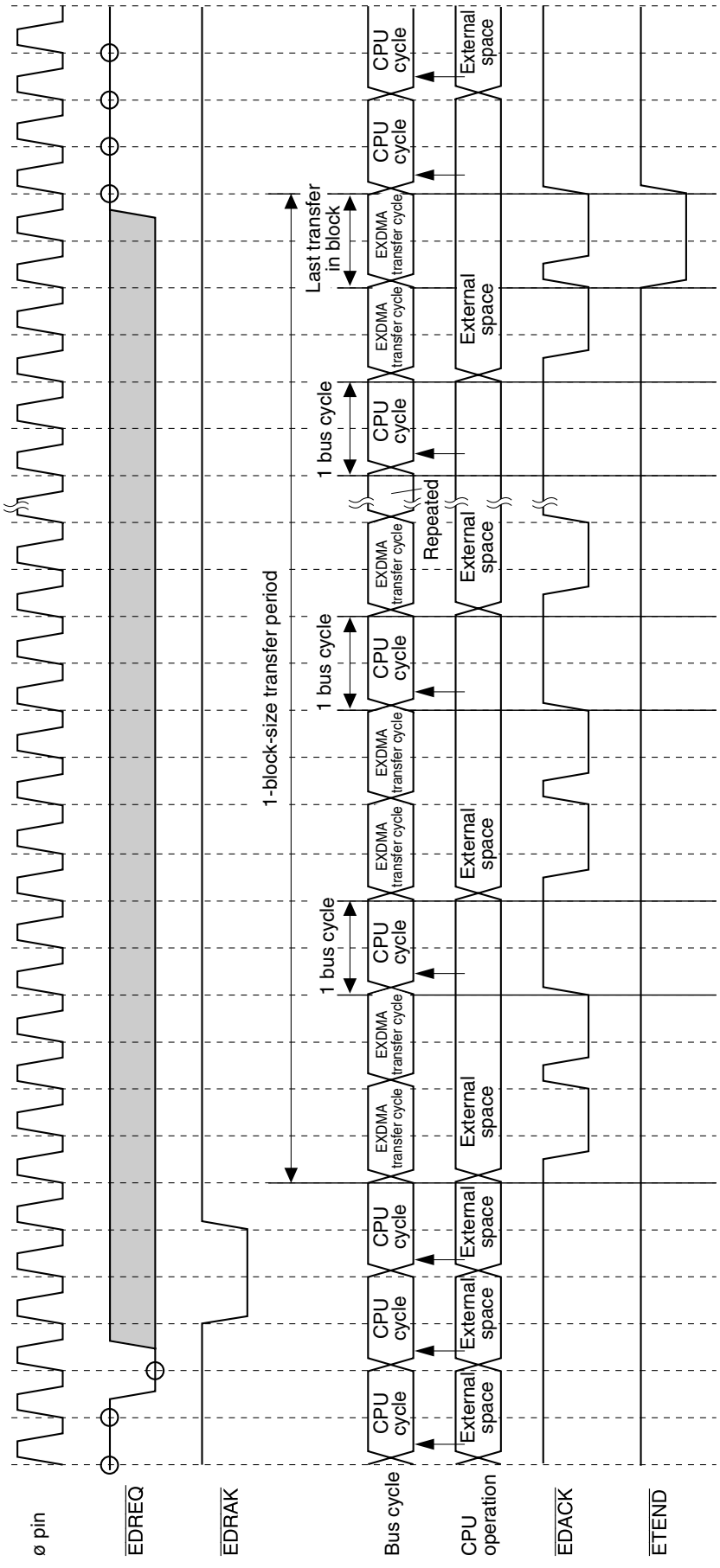


Figure 8.43 External Request/Cycle Steal Mode/Block Transfer Mode (CPU Cycles/Single Address Mode/Low Level Sensing/BGUP = 1)

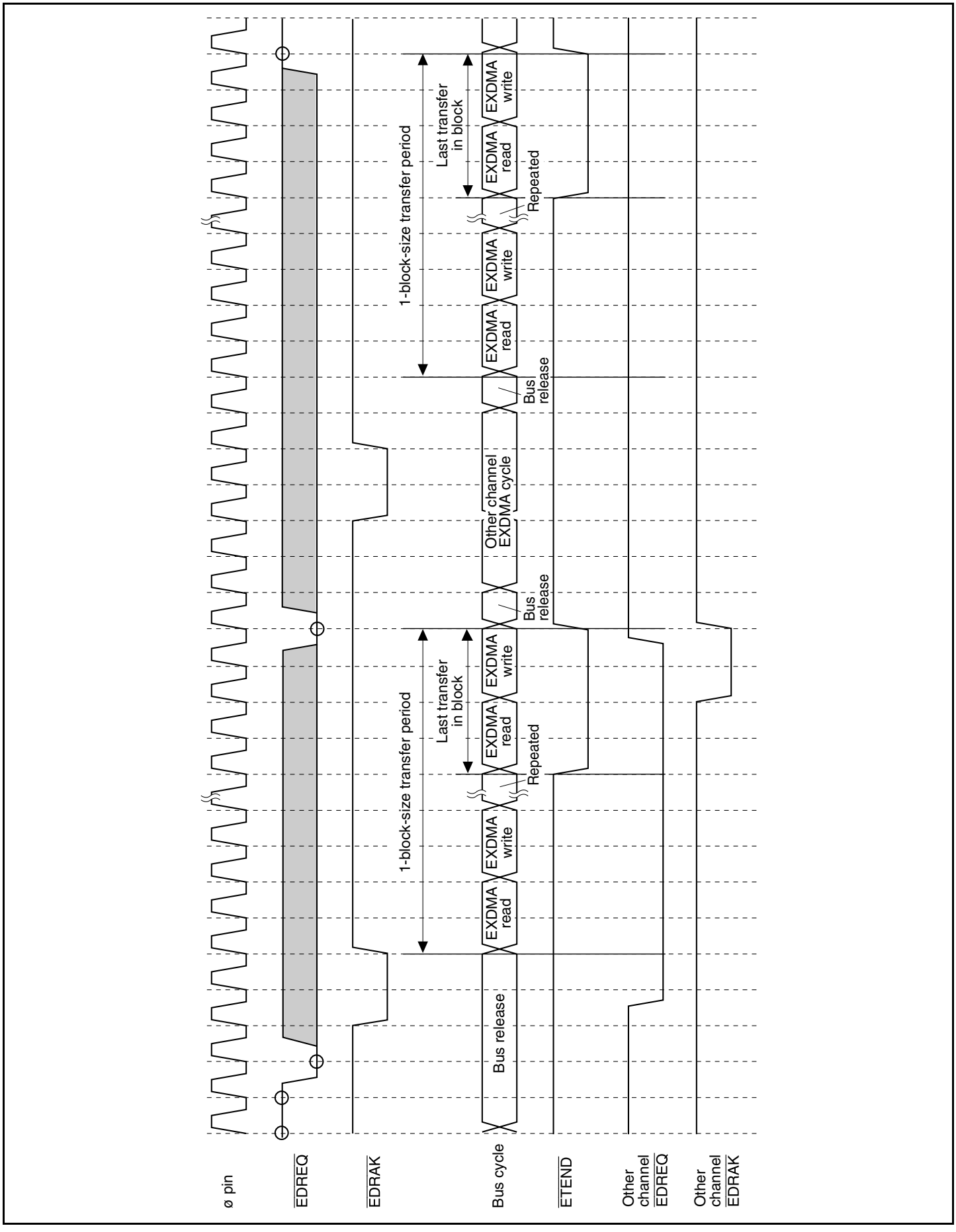


Figure 8.44 External Request/Cycle Steal Mode/Block Transfer Mode (Contention with Another Channel/Dual Address Mode/Low Level Sensing)

8.4.12 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the EDA bit in EDMDR changes from 1 to 0, indicating that DMA transfer has ended.

Transfer End by 1 → 0 Transition of EDTCR: When the value of EDTCR changes from 1 to 0, DMA transfer ends on the corresponding channel and the EDA bit in EDMDR is cleared to 0. If the TCEIE bit in EDMDR is set at this time, a transfer end interrupt request is generated by the transfer counter and the IRF bit in EDMDR is set to 1.

In block transfer mode, DMA transfer ends when the value of bits 15 to 0 in EDTCR changes from 1 to 0.

DMA transfer does not end if the EDTCR value has been 0 since before the start of transfer.

Transfer End by Repeat Area Overflow Interrupt: If an address overflows the repeat area when a repeat area specification has been made and repeat interrupts have been enabled (with the SARIE or DARIE bit in EDACR), a repeat area overflow interrupt is requested. DMA transfer ends, the EDA bit in EDMDR is cleared to 0, and the IRF bit in EDMDR is set to 1.

In dual address mode, if a repeat area overflow interrupt is requested during a read cycle, the following write cycle processing is still executed.

In block transfer mode, if a repeat area overflow interrupt is requested during transfer of a block, transfer continues to the end of the block. Transfer end by means of a repeat area overflow interrupt occurs between block-size transfers.

Transfer End by 0-Write to EDA Bit in EDMDR: When 0 is written to the EDA bit in EDMDR by the CPU, etc., transfer ends after completion of the DMA cycle in which transfer is in progress or a transfer request was accepted.

In block transfer mode, DMA transfer halts after completion of one-block-size transfer.

The EDA bit in EDMDR is not cleared to 0 until all transfer processing has ended. Up to that point, the value of the EDA bit will be read as 1.

Transfer Abort by NMI Interrupt: DMA transfer is aborted when an NMI interrupt is generated. The EDA bit is cleared to 0 in all channels. In external request mode, DMA transfer is performed for all transfer requests for which EDRAK has been output. In dual address mode, processing is executed for the write cycle following the read cycle.

In block transfer mode, operation is aborted even in the middle of a block-size transfer. As the transfer is halted midway through a block, the BEF bit in EDMDR is set to 1 to indicate that the block transfer was not carried out normally.

When transfer is aborted, register values are retained, and as the address registers indicate the next transfer addresses, transfer can be resumed by setting the EDA bit to 1 in EDMDR. If the BEF bit is 1 in EDMDR, transfer can be resumed from midway through a block.

Hardware Standby Mode and Reset Input: The EXDMAC is initialized in hardware standby mode and by a reset. DMA transfer is not guaranteed in these cases.

8.4.13 Relationship between EXDMAC and Other Bus Masters

The read and write operations in a DMA transfer cycle are indivisible, and a refresh cycle, external bus release cycle, or internal bus mastership (CPU, DTC, or DMAC) external space access cycle never occurs between the two.

When read and write cycles occur consecutively, as in burst transfer or block transfer, a refresh or external bus release state may be inserted after the write cycle. As the internal bus masters are of lower priority than the EXDMAC, external space accesses by internal bus masters are not executed until the EXDMAC releases the bus.

The EXDMAC releases the bus in the following cases:

1. When DMA transfer is performed in cycle steal mode
2. When switching to a different channel
3. When transfer ends in burst transfer mode
4. When transfer of one block ends in block transfer mode
5. When burst transfer or block transfer is performed with the BGUP bit in EDMDR set to 1 (however, the bus is not released between read and write cycles)

8.5 Interrupt Sources

EXDMAC interrupt sources are a transfer end indicated by the transfer counter, and repeat area overflow interrupts. Table 8.4 shows the interrupt sources and their priority order.

Table 8.4 Interrupt Sources and Priority Order

Interrupt	Interrupt source	Interrupt Priority
EXDMTEND0	Transfer end indicated by channel 0 transfer counter	High
	Channel 0 source address repeat area overflow	
	Channel 0 destination address repeat area overflow	
EXDMTEND1	Transfer end indicated by channel 1 transfer counter	
	Channel 1 source address repeat area overflow	
	Channel 1 destination address repeat area overflow	
EXDMTEND2	Transfer end indicated by channel 2 transfer counter	
	Channel 2 source address repeat area overflow	
	Channel 2 destination address repeat area overflow	
EXDMTEND3	Transfer end indicated by channel 3 transfer counter	
	Channel 3 source address repeat area overflow	
	Channel 3 destination address repeat area overflow	Low

Interrupt sources can be enabled or disabled by means of the EDIE bit in EDMDR for the relevant channel, and can be sent to the interrupt controller independently. The relative priority order of the channels is determined by the interrupt controller (see table 8.4).

Figure 8.45 shows the transfer end interrupt logic. A transfer end interrupt is generated whenever the EDIE bit is set to 1 while the IRF bit is set to 1 in EDMDR.

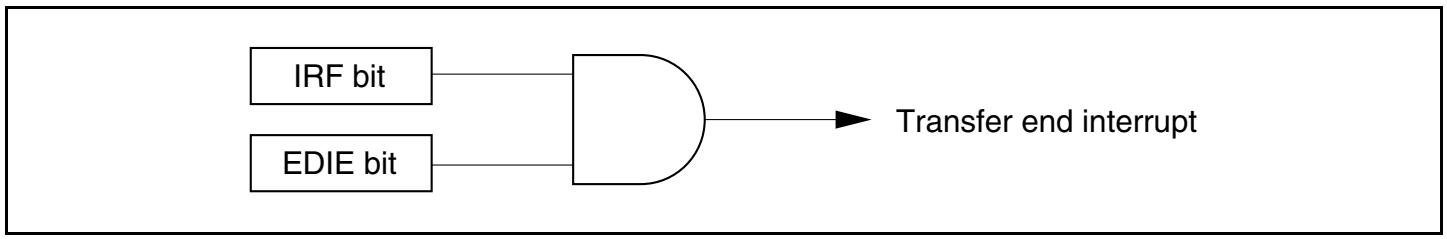
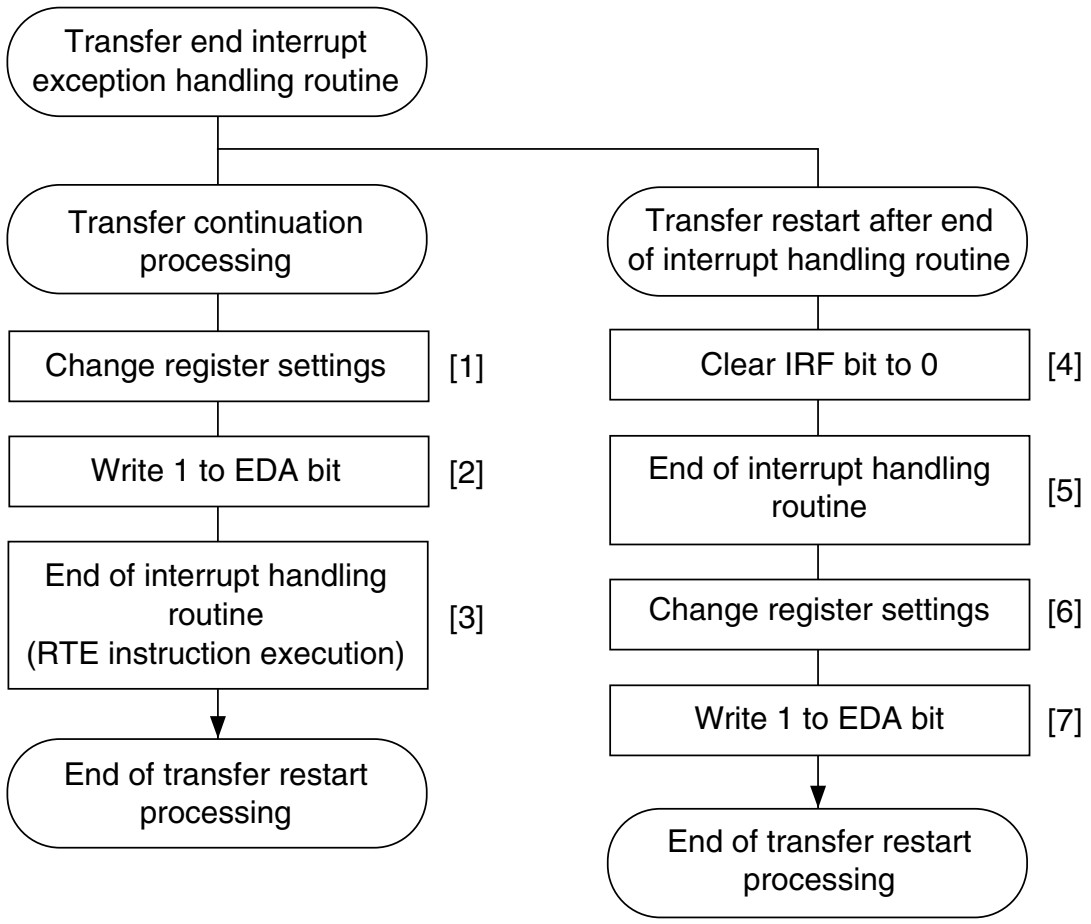


Figure 8.45 Transfer End Interrupt Logic

Interrupt source settings are made individually with the interrupt enable bits in the registers for the relevant channels. The transfer counter's transfer end interrupt is enabled or disabled by means of the TCEIE bit in EDMDR, the source address register repeat area overflow interrupt by means of the SARIE bit in EDACR, and the destination address register repeat area overflow interrupt by means of the DARIE bit in EDACR. When an interrupt source occurs while the corresponding interrupt enable bit is set to 1, the IRF bit in EDMDR is set to 1. The IRF bit is set by all interrupt sources indiscriminately.

The transfer end interrupt can be cleared either by clearing the IRF bit to 0 in EDMDR within the interrupt handling routine, or by re-setting the transfer counter and address registers and then

setting the EDA bit to 1 in EDMDR to perform transfer continuation processing. An example of the procedure for clearing the transfer end interrupt and restarting transfer is shown in figure 8.46.



[1] Write set values to the registers (transfer counter, address registers, etc.).

[2] Write 1 to the EDA bit in EDMDR to restart EXDMA operation. When 1 is written to the EDA bit, the IRF bit in EDMDR is automatically cleared to 0 and the interrupt source is cleared.

[3] The interrupt handling routine is ended with an RTE instruction, etc.

[4] Clear the IRF bit to 0 in EDMDR by first reading 1 from it, then writing 0.

[5] After the interrupt handling routine is ended with an RTE instruction, etc., interrupt masking is cleared.

[6] Write set values to the registers (transfer counter, address registers, etc.).

[7] Write 1 to the EDA bit in EDMDR to restart EXDMA operation.

Figure 8.46 Example of Procedure for Restarting Transfer on Channel in which Transfer End Interrupt Occurred

8.6 Usage Notes

8.6.1 EXDMAC Register Access during Operation

Except for clearing the EDA bit to 0 in EDMDR, settings should not be changed for a channel in operation (including the transfer standby state). Transfer must be disabled before changing a setting for an operational channel.

8.6.2 Module Stop State

When the MSTP14 bit is set to 1 in MSTPCRH, the EXDMAC clock stops and the EXDMAC enters the module stop state. However, 1 cannot be written to the MSTP14 bit when any of the EXDMAC's channels is enabled for transfer, or when an interrupt is being requested. Before setting the MSTP14 bit, first clear the EDA bit in EDMDR to 0, then clear the IRF or EDIE bit in EDMDR to 0.

When the EXDMAC clock stops, EXDMAC registers can no longer be accessed. The following EXDMAC register settings remain valid in the module stop state, and so should be changed, if necessary, before making the module stop transition.

- ETENDE = 1 in EDMDR (ETEND pin enable)
- EDRAKE = 1 in EDMDR (EDRAK pin enable)
- AMS = 1 in EDMDR (EDACK pin enable)

8.6.3 EDREQ Pin Falling Edge Activation

Falling edge sensing on the EDREQ pin is performed in synchronization with EXDMAC internal operations, as indicated below.

[1] Activation request standby state: Waits for low level sensing on EDREQ pin, then goes to [2].

[2] Transfer standby state: Waits for EXDMAC data transfer to become possible, then goes to [3].

[3] Activation request disabled state: Waits for high level sensing on EDREQ pin, then goes to [1].

After EXDMAC transfer is enabled, the EXDMAC goes to state [1], so low level sensing is used for the initial activation after transfer is enabled.

8.6.4 Activation Source Acceptance

At the start of activation source acceptance, low level sensing is used for both falling edge sensing and low level sensing on the EDREQ pin. Therefore, a request is accepted in the case of a low level at the EDREQ pin that occurs before execution of the EDMDR write for setting the transfer-enabled state.

When the EXDMAC is activated, make sure, if necessary, that a low level does not remain at the EDREQ pin from the previous end of transfer, etc.

8.6.5 Enabling Interrupt Requests when IRF = 1 in EDMDR

When transfer is started while the IRF bit is set to 1 in EDMDR, if the EDIE bit is set to 1 in EDMDR together with the EDA bit in EDMDR, enabling interrupt requests, an interrupt will be requested since EDIE = 1 and IRF = 1. To prevent the occurrence of an erroneous interrupt request when transfer starts, ensure that the IRF bit is cleared to 0 before the EDIE bit is set to 1.

8.6.6 ETEND Pin and CBR Refresh Cycle

If the last EXDMAC transfer cycle and a CBR refresh cycle occur simultaneously, note that although the CBR refresh and the last transfer cycle may be executed consecutively, ETEND may also go low in this case for the refresh cycle.

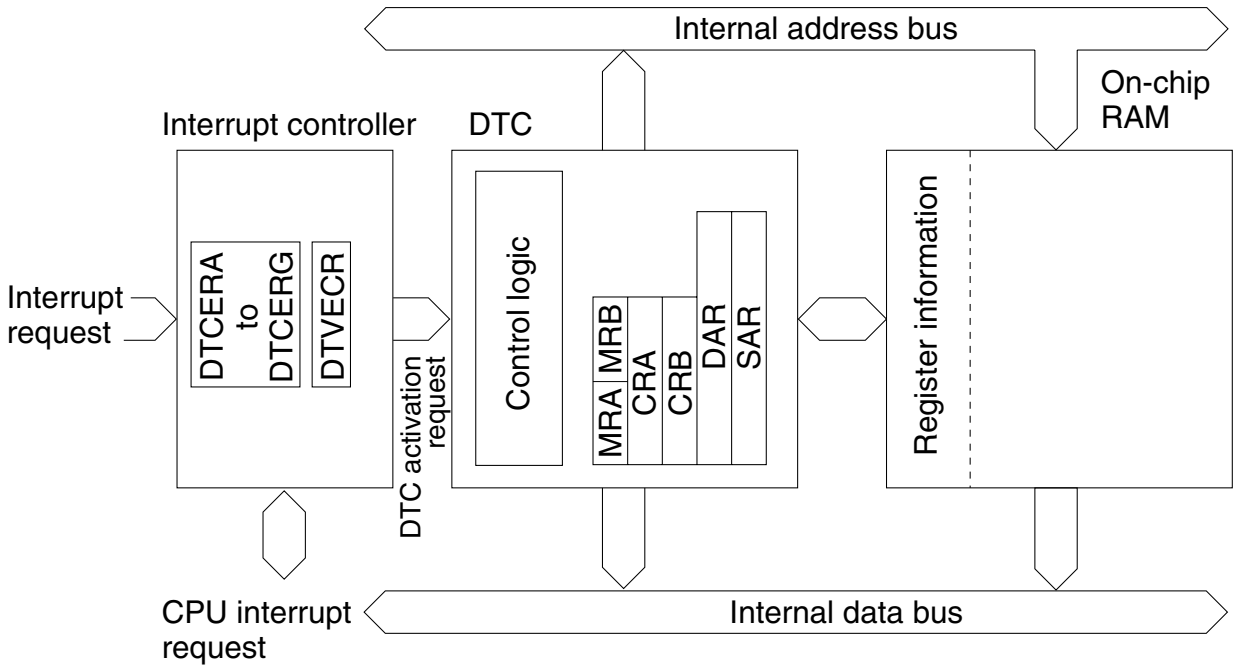
Section 9 Data Transfer Controller (DTC)

This LSI includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

Figure 9.1 shows a block diagram of the DTC. The DTC's register information is stored in the on-chip RAM. When the DTC is used, the RAME bit in SYSCR must be set to 1. A 32-bit bus connects the DTC to the on-chip RAM (1 kbyte), enabling 32-bit/1-state reading and writing of the DTC register information.

9.1 Features

- Transfer possible over any number of channels
- Three transfer modes
  Normal, repeat, and block transfer modes available
- One activation source can trigger a number of data transfers (chain transfer)
- Direct specification of 16-Mbyte address space possible
- Activation by software is possible
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
- Module stop mode can be set



Legend

MRA, MRB	: DTC mode registers A and B
CRA, CRB	: DTC transfer count registers A and B
SAR	: DTC source address register
DAR	: DTC destination address register
DTCERA to DTCERG	: DTC enable registers A to G
DTVECR	: DTC vector register

Figure 9.1 Block Diagram of DTC

9.2 Register Descriptions

DTC has the following registers.

- DTC mode register A (MRA)
- DTC mode register B (MRB)
- DTC source address register (SAR)
- DTC destination address register (DAR)
- DTC transfer count register A (CRA)
- DTC transfer count register B (CRB)

These six registers cannot be directly accessed from the CPU. When activated, the DTC reads a set of register information that is stored in an on-chip RAM to the corresponding DTC registers and transfers data. After the data transfer, it writes a set of updated register information back to the RAM.

- DTC enable registers A to G (DTCERA to DTCERG)

- DTC vector register (DTVECR)

9.2.1 DTC Mode Register A (MRA)

MRA selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	SM1	Undefined	—	Source Address Mode 1 and 0
6	SM0	Undefined	—	These bits specify an SAR operation after a data transfer. 0x: SAR is fixed 10: SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)
5	DM1	Undefined	—	Destination Address Mode 1 and 0
4	DM0	Undefined	—	These bits specify a DAR operation after a data transfer. 0x: DAR is fixed 10: DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: DAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)
3	MD1	Undefined	—	DTC Mode
2	MD0	Undefined	—	These bits specify the DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: Setting prohibited
1	DTS	Undefined	—	DTC Transfer Mode Select Specifies whether the source side or the destination side is set to be a repeat area or block area, in repeat mode or block transfer mode. 0: Destination side is repeat area or block area 1: Source side is repeat area or block area

Bit	Bit Name	Initial Value	R/W	Description
0	Sz	Undefined	—	DTC Data Transfer Size Specifies the size of data to be transferred. 0: Byte-size transfer 1: Word-size transfer

Legend:
X : Don't care

9.2.2 DTC Mode Register B (MRB)

MRB selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable When this bit is set to 1, a chain transfer will be performed. For details, refer to 9.5.4, Chain Transfer. In data transfer with CHNE set to 1, determination of the end of the specified number of transfers, clearing of the activation source flag, and clearing of DTCER is not performed.
6	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, a CPU interrupt request is generated every time after a data transfer ends. When this bit is set to 0, a CPU interrupt request is generated at the time when the specified number of data transfer ends.
5	CHNS	Undefined	—	DTC Chain Transfer Select Specifies the chain transfer condition. 0: Chain transfer every time 1: Chain transfer only when transfer counter = 0
4 to 0	—	Undefined	—	Reserved These bits have no effect on DTC operation, and should always be written with 0.

9.2.3 DTC Source Address Register (SAR)

SAR is a 24-bit register that designates the source address of data to be transferred by the DTC. For word-size transfer, specify an even source address.

9.2.4 DTC Destination Address Register (DAR)

DAR is a 24-bit register that designates the destination address of data to be transferred by the DTC. For word-size transfer, specify an even destination address.

9.2.5 DTC Transfer Count Register A (CRA)

CRA is a 16-bit register that designates the number of times data is to be transferred by the DTC.

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65,536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

In repeat mode or block transfer mode, the CRA is divided into two parts: the upper 8 bits (CRAH) and the lower 8 bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and the contents of CRAH are sent when the count reaches H'00.

9.2.6 DTC Transfer Count Register B (CRB)

CRB is a 16-bit register that designates the number of times data is to be transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65,536) that is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000. This register is not available in normal and repeat modes.

9.2.7 DTC Enable Registers A to G (DTCERA to DTCERG)

DTCER which is comprised of seven registers, DTCERA to DTCERG, is a register that specifies DTC activation interrupt sources. The correspondence between interrupt sources and DTCE bits is shown in table 9.1. For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for reading and writing. If all interrupts are masked, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

Bit	Bit Name	Initial Value	R/W	Description
7	DTCE7	0	R/W	DTC Activation Enable
6	DTCE6	0	R/W	Setting this bit to 1 specifies a relevant interrupt source to a DTC activation source.
5	DTCE5	0	R/W	[Clearing conditions]
4	DTCE4	0	R/W	• When the DISEL bit is 1 and the data transfer has ended
3	DTCE3	0	R/W	• When the specified number of transfers have ended
2	DTCE2	0	R/W	These bits are not cleared when the DISEL bit is 0 and the specified number of transfers have not ended
1	DTCE1	0	R/W	
0	DTCE0	0	R/W	

9.2.8 DTC Vector Register (DTVECR)

DTVECR enables or disables DTC activation by software, and sets a vector number for the software activation interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	SWDTE	0	R/W	DTC Software Activation Enable Setting this bit to 1 activates DTC. Only 1 can be written to this bit. [Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers have not ended • When 0 is written to the DISEL bit after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU. When the DISEL bit is 1 and data transfer has ended or when the specified number of transfers have ended, this bit will not be cleared.
6	DTVEC6	0	R/W	DTC Software Activation Vectors 6 to 0
5	DTVEC5	0	R/W	These bits specify a vector number for DTC software activation.
4	DTVEC4	0	R/W	The vector address is expressed as H'0400 + (vector number × 2). For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420. When the bit SWDTE is 0, these bits can be written.
3	DTVEC3	0	R/W	
2	DTVEC2	0	R/W	
1	DTVEC1	0	R/W	
0	DTVEC0	0	R/W	

9.3 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTVECR by software. An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTCER bit. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source or corresponding DTCER bit is cleared. The activation source flag, in the case of RXI0, for example, is the RDRF flag of SCI_0.

When an interrupt has been designated a DTC activation source, existing CPU mask level and interrupt controller priorities have no effect. If there is more than one activation source at the same time, the DTC operates in accordance with the default priorities.

Figure 9.2 shows a block diagram of activation source control. For details see section 5, Interrupt Controller.

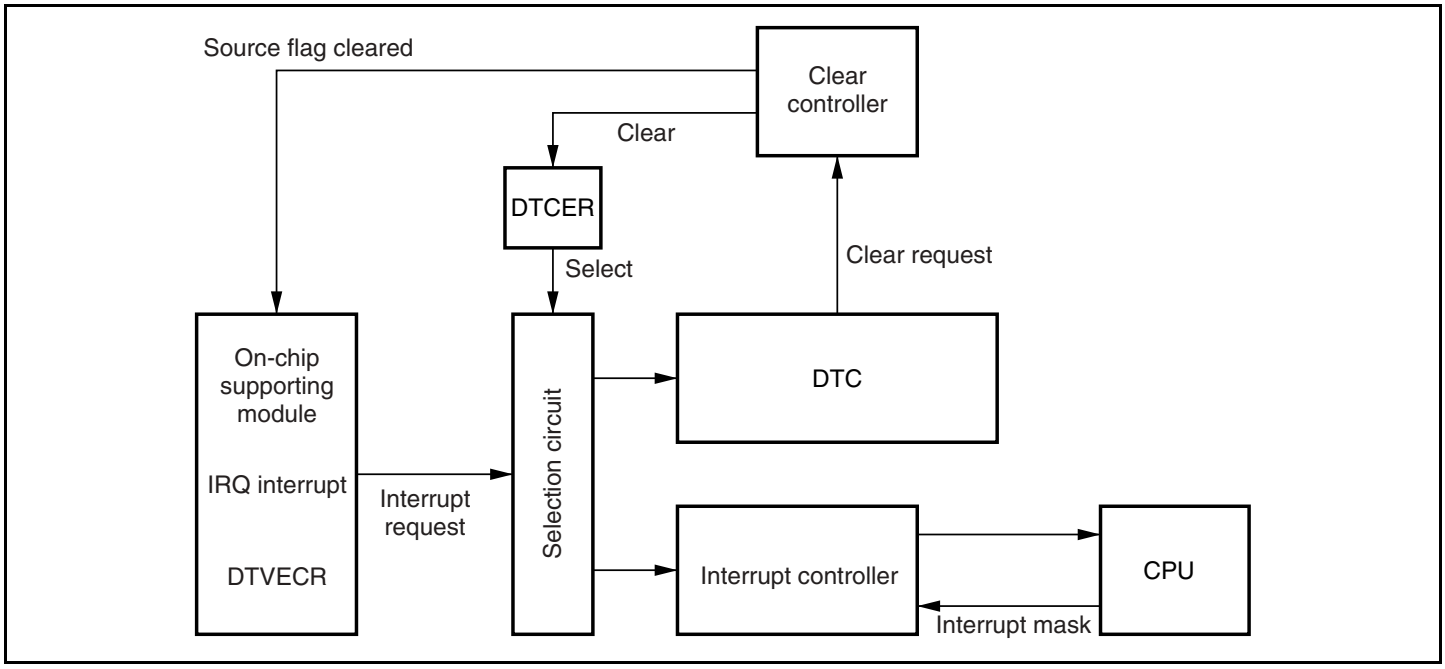


Figure 9.2 Block Diagram of DTC Activation Source Control

9.4 Location of Register Information and DTC Vector Table

Locate the register information in the on-chip RAM (addresses: H'FFBC00 to H'FFBFFF). Register information should be located at the address that is multiple of four within the range. Locating the register information in address space is shown in figure 9.3. Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information. In the case of chain transfer, register information should be located in consecutive areas as shown in figure 9.3 and the register information start address should be located at the corresponding vector address to the activation source. The DTC reads the start address of the register information from the vector address set for each activation source, and then reads the register information from that start address.

When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[6:0] × 2). For example, if DTVECR is H'10, the vector address is H'0420.

The configuration of the vector address is the same in both normal* and advanced modes, a 2-byte unit being used in both cases. These two bytes specify the lower bits of the register information start address.

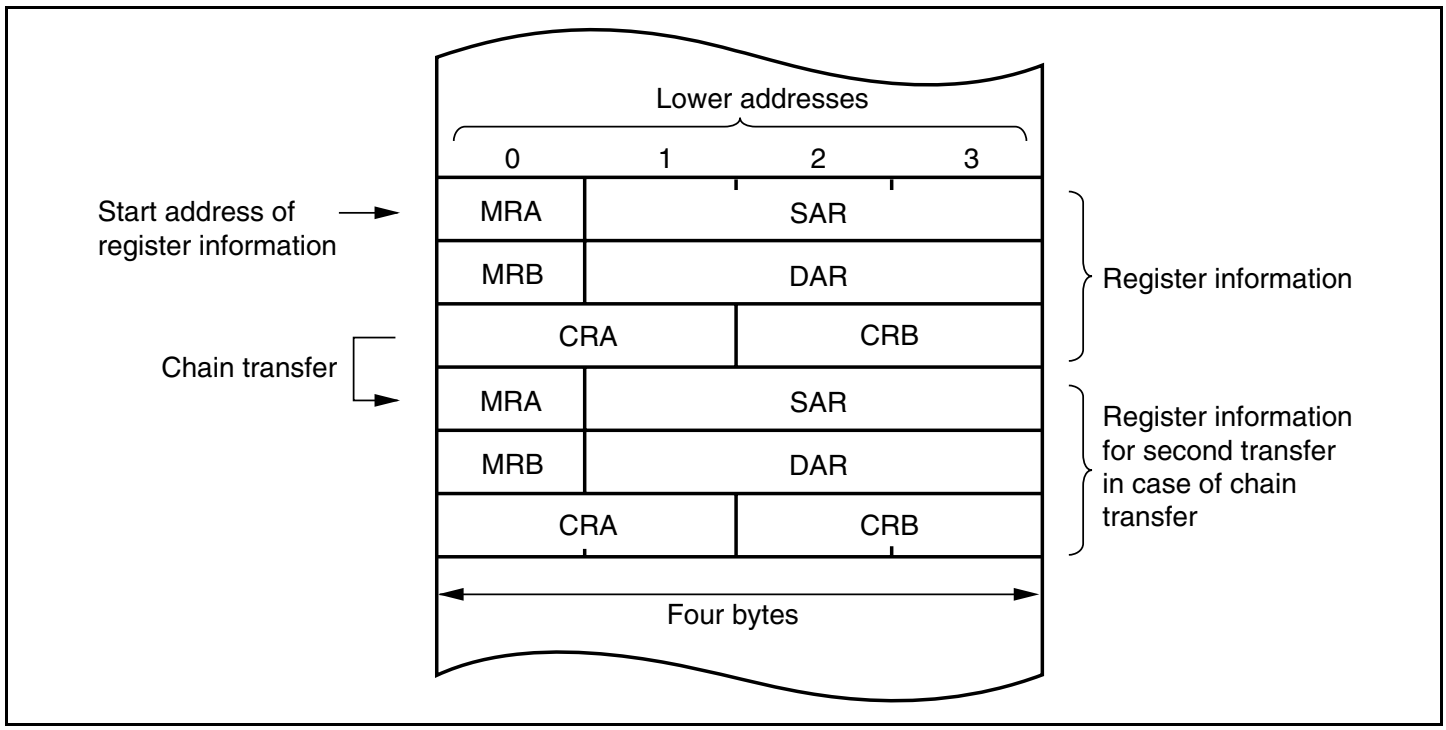


Figure 9.3 Correspondence between DTC Vector Address and Register Information

Note: Not available in this LSI.

Table 9.1 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address	DTCE*	Priority
Software	Write to DTVECR	DTVECR	H'0400 + (DTVECR [6:0] × 2)	—	High
External pin	IRQ0	16	H'0420	DTCEA7	
	IRQ1	17	H'0422	DTCEA6	
	IRQ2	18	H'0424	DTCEA5	
	IRQ3	19	H'0426	DTCEA4	
	IRQ4	20	H'0428	DTCEA3	
	IRQ5	21	H'042A	DTCEA2	
	IRQ6	22	H'042C	DTCEA1	
	IRQ7	23	H'042E	DTCEA0	
	IRQ8	24	H'0430	DTCEB7	
	IRQ9	25	H'0432	DTCEB6	
	IRQ10	26	H'0434	DTCEB5	
	IRQ11	17	H'0436	DTCEB4	
	IRQ12	18	H'0438	DTCEB3	
	IRQ13	19	H'043A	DTCEB2	
	IRQ14	30	H'043C	DTCEB1	
	IRQ15	31	H'043E	DTCEB0	
A/D	ADI	38	H'044C	DTCEC6	
TPU_0	TGI0A	40	H'0450	DTCEC5	
	TGI0B	41	H'0452	DTCEC4	
	TGI0C	42	H'0454	DTCEC3	
	TGI0D	43	H'0456	DTCEC2	
TPU_1	TGI1A	48	H'0460	DTCEC1	
	TGI1B	49	H'0462	DTCEC0	
TPU_2	TGI2A	52	H'0468	DTCED7	
	TGI2B	53	H'046A	DTCED6	
TPU_3	TGI3A	56	H'0470	DTCED5	
	TGI3B	57	H'0472	DTCED4	
	TGI3C	58	H'0474	DTCED3	
	TGI3D	59	H'0476	DTCED2	

TPU_3 11.0187 1451.0187 145. 111.8237 0 0 11.8237 41.018 1tDTCED5

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address	DTCE*	Priority
TMR_0	CMIA0	72	H'0490	DTCEE3	High
	CMIB0	73	H'0492	DTCEE2	
TMR_1	CMIA1	76	H'0498	DTCEE1	
	CMIB1	77	H'049A	DTCEE0	
DMAC	DMTEND0A	80	H'04A0	DTCEF7	
	DMTEND0B	81	H'04A2	DTCEF6	
	DMTEND1A	82	H'04A4	DTCEF5	
	DMTEND1B	83	H'04A6	DTCEF4	
SCI_0	RXI0	89	H'04B2	DTCEF3	
	TXI0	90	H'04B4	DTCEF2	
SCI_1	RXI1	93	H'04BA	DTCEF1	
	TXI1	94	H'04BC	DTCEF0	
SCI_2	RXI2	97	H'04C2	DTCEG7	
	TXI2	98	H'04C4	DTCEG6	Low

Note: DTCE bits with no corresponding interrupt are reserved, and should be written with 0. When clearing the software standby state or all-module-clocks-stop mode with an interrupt, write 0 to the corresponding DTCE bit.

9.5 Operation

The DTC stores register information in the on-chip RAM. When activated, the DTC reads register information that is already stored in the on-chip RAM and transfers data on the basis of that register information. After the data transfer, it writes updated register information back to the on-chip RAM. Pre-storage of register information in the on-chip RAM makes it possible to transfer data over any required number of channels. There are three transfer modes: normal mode, repeat mode, and block transfer mode. Setting the CHNE bit to 1 makes it possible to perform a number of transfers with a single activation (chain transfer). A setting can also be made to have chain transfer performed only when the transfer counter value is 0. This enables DTC re-setting to be performed by the DTC itself.

The 24-bit SAR designates the DTC transfer source address and the 24-bit DAR designates the transfer destination address. After each transfer, SAR and DAR are independently incremented, decremented, or left fixed.

Figure 9.4 shows a flowchart of DTC operation, and table 9.2 summarizes the chain transfer conditions (combinations for performing the second and third transfers are omitted).

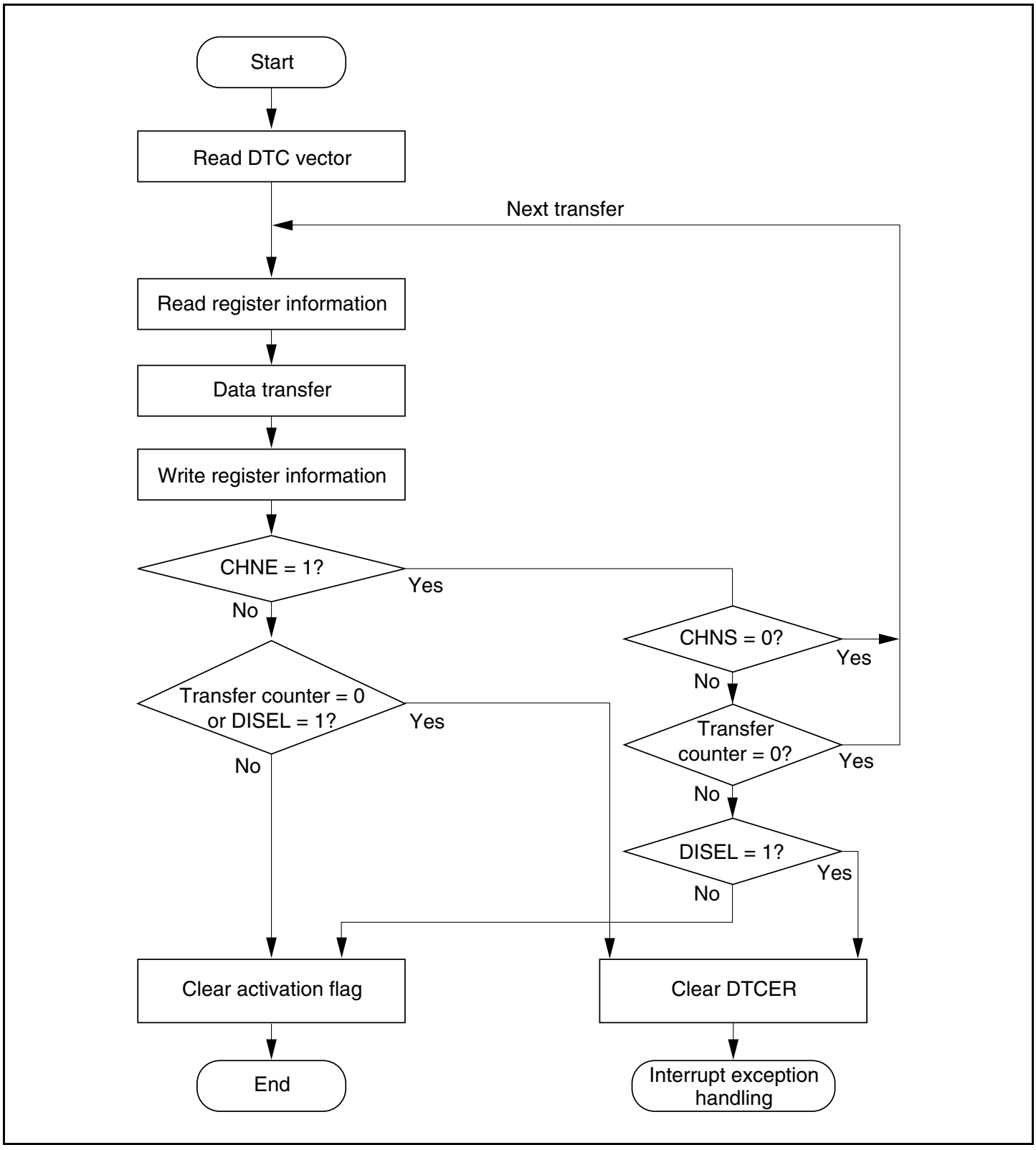


Figure 9.4 Flowchart of DTC Operation

Table 9.2 Chain Transfer Conditions

1st Transfer				2nd Transfer				
CHNE	**CHNS**	**DISEL**	**CR**	**CHNE**	**CHNS**	**DISEL**	**CR**	**DTC Transfer**
0	—	0	Not 0	—	—	—	—	Ends at 1st transfer
0	—	0	0	—	—	—	—	Ends at 1st transfer
0	—	1	—	—	—	—	—	Interrupt request to CPU
1	0	—	—	0	—	0	Not 0	Ends at 2nd transfer
				0	—	0	0	Ends at 2nd transfer
				0	—	1	—	Interrupt request to CPU
1	1	0	Not 0	—	—	—	—	Ends at 1st transfer
1	1	—	0	0	—	0	Not 0	Ends at 2nd transfer
				0	—	0	0	Ends at 2nd transfer
				0	—	1	—	Interrupt request to CPU
1	1	1	Not 0	—	—	—	—	Ends at 1st transfer
								Interrupt request to CPU

9.5.1 Normal Mode

In normal mode, one operation transfers one byte or one word of data. Table 9.3 lists the register function in normal mode. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has ended, a CPU interrupt can be requested.

Table 9.3 Register Function in Normal Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register A	CRA	Designates transfer count
DTC transfer count register B	CRB	Not used

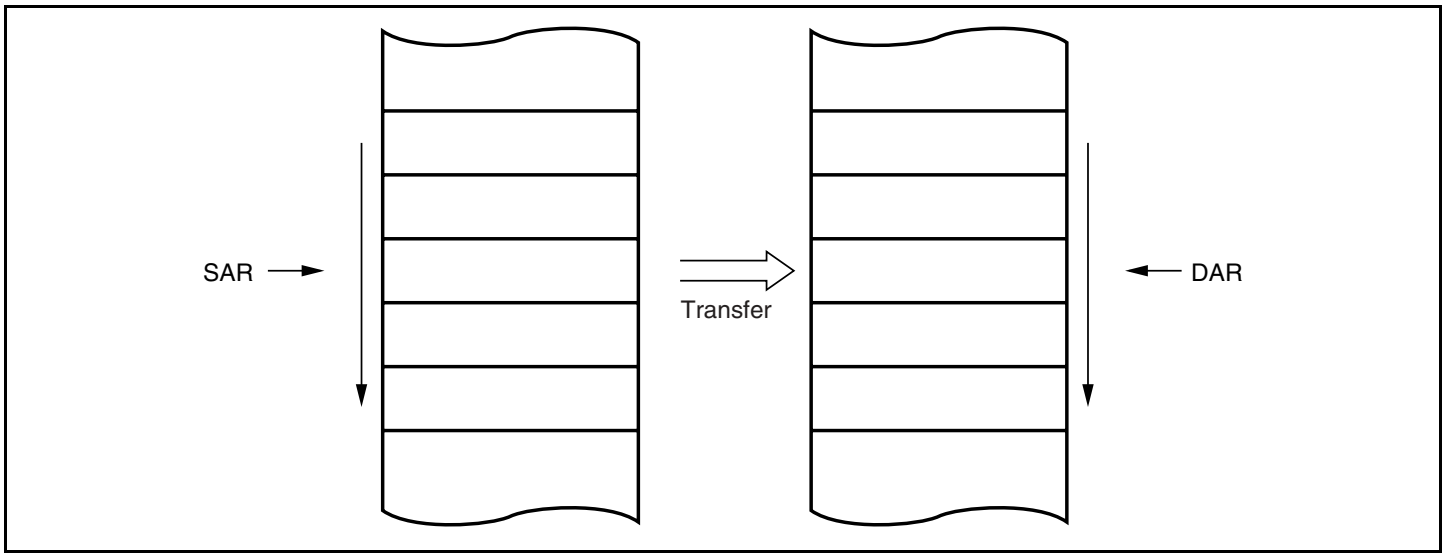


Figure 9.5 Memory Mapping in Normal Mode

9.5.2 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data. Table 9.4 lists the register function in repeat mode. From 1 to 256 transfers can be specified. Once the specified number of transfers has ended, the initial state of the transfer counter and the address register specified as the repeat area is restored, and transfer is repeated. In repeat mode the transfer counter value does not reach H'00, and therefore CPU interrupts cannot be requested when DISEL = 0.

Table 9.4 Register Function in Repeat Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register AH	CRAH	Holds number of transfers
DTC transfer count register AL	CRAL	Designates transfer count
DTC transfer count register B	CRB	Not used

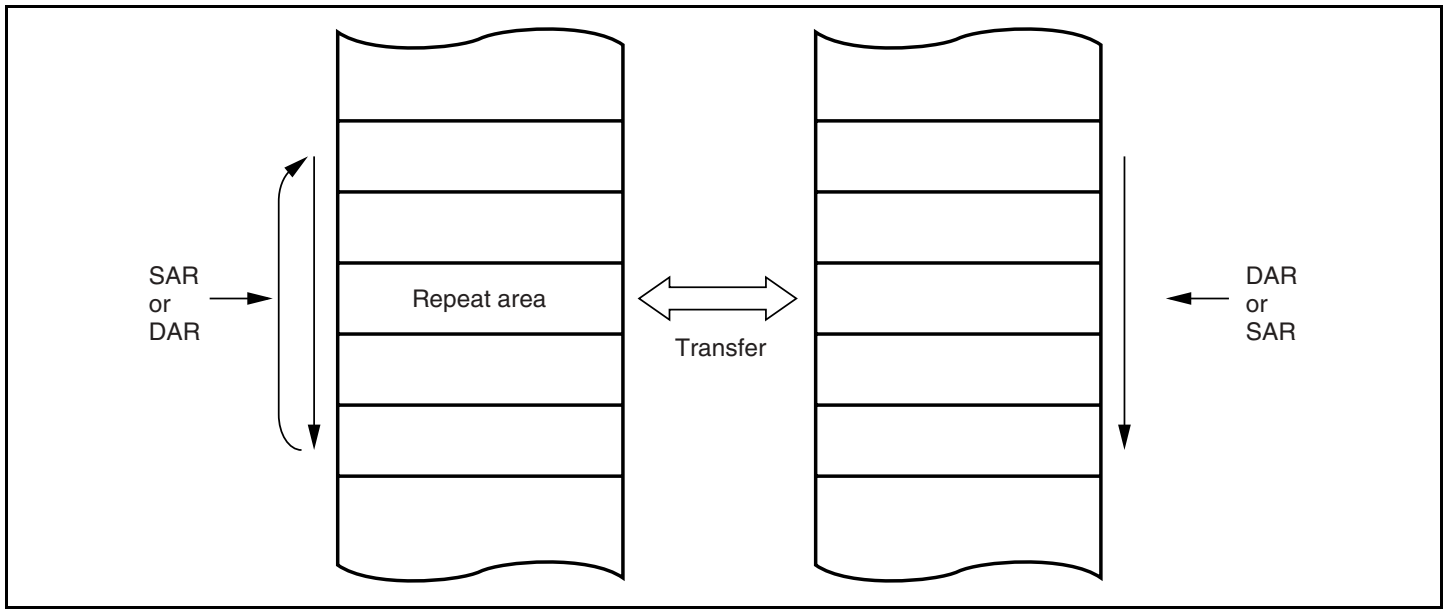


Figure 9.6 Memory Mapping in Repeat Mode

9.5.3 Block Transfer Mode

In block transfer mode, one operation transfers one block of data. Either the transfer source or the transfer destination is designated as a block area. Table 9.5 lists the register function in block transfer mode.

The block size is 1 to 256. When the transfer of one block ends, the initial state of the block size counter and the address register specified as the block area is restored. The other address register is then incremented, decremented, or left fixed. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has ended, a CPU interrupt is requested.

Table 9.5 Register Function in Block Transfer Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register AH	CRAH	Holds block size
DTC transfer count register AL	CRAL	Designates block size count
DTC transfer count register B	CRB	Designates transfer count

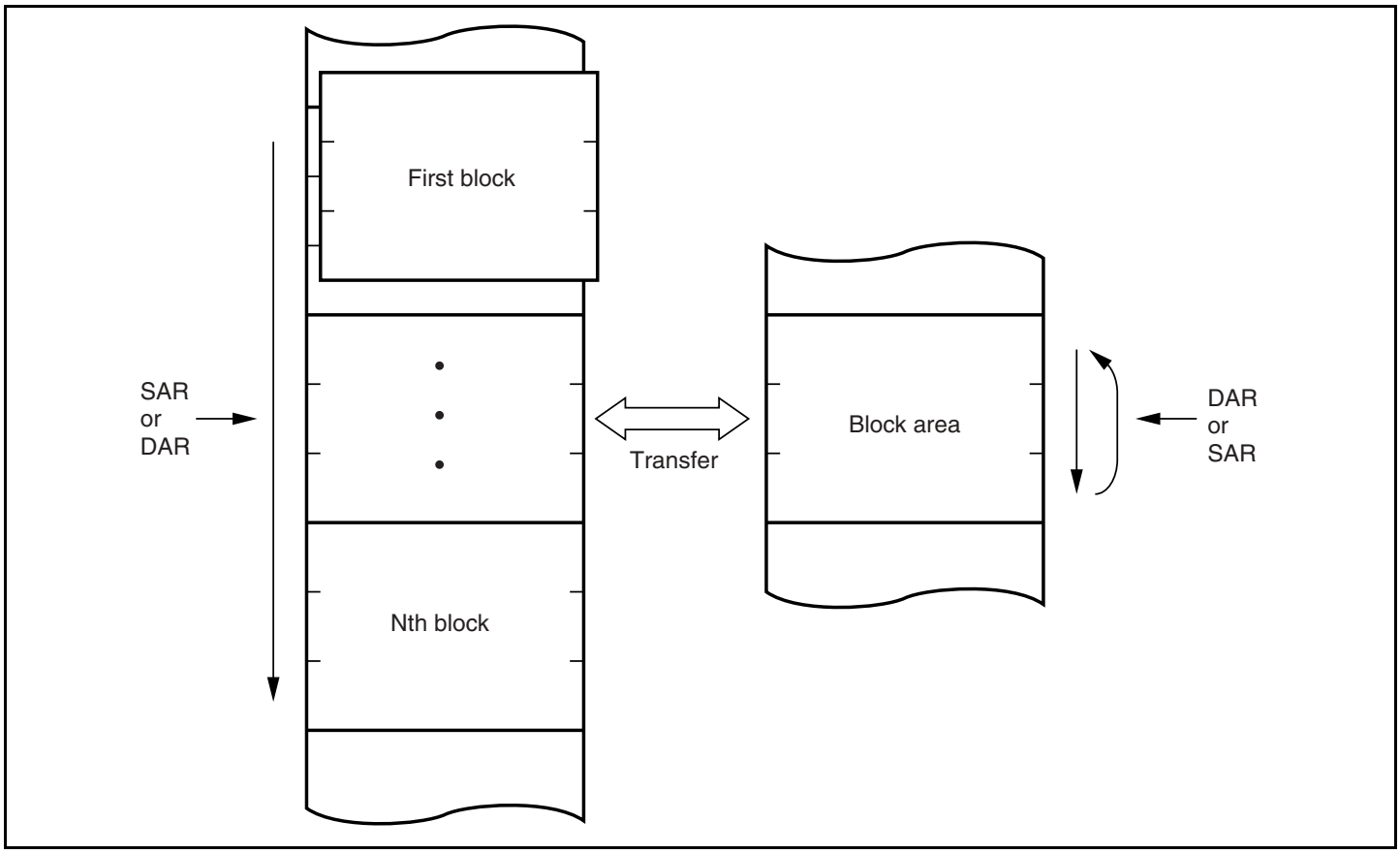


Figure 9.7 Memory Mapping in Block Transfer Mode

9.5.4 Chain Transfer

Setting the CHNE bit to 1 enables a number of data transfers to be performed consecutively in response to a single transfer request. SAR, DAR, CRA, CRB, MRA, and MRB, which define data transfers, can be set independently.

Figure 9.8 shows the operation of chain transfer. When activated, the DTC reads the register information start address stored at the vector address, and then reads the first register information at that start address. The CHNE bit in MRB is checked after the end of data transfer, if the value is 1, the next register information, which is located consecutively, is read and transfer is performed. This operation is repeated until the end of data transfer of register information with CHNE = 0. It is also possible, by setting both the CHNE bit and CHNS bit to 1, to specify execution of chain transfer only when the transfer counter value is 0.

In the case of transfer with CHNE set to 1, an interrupt request to the CPU is not generated at the end of the specified number of transfers or by setting of the DISEL bit to 1, and the interrupt source flag for the activation source is not affected.

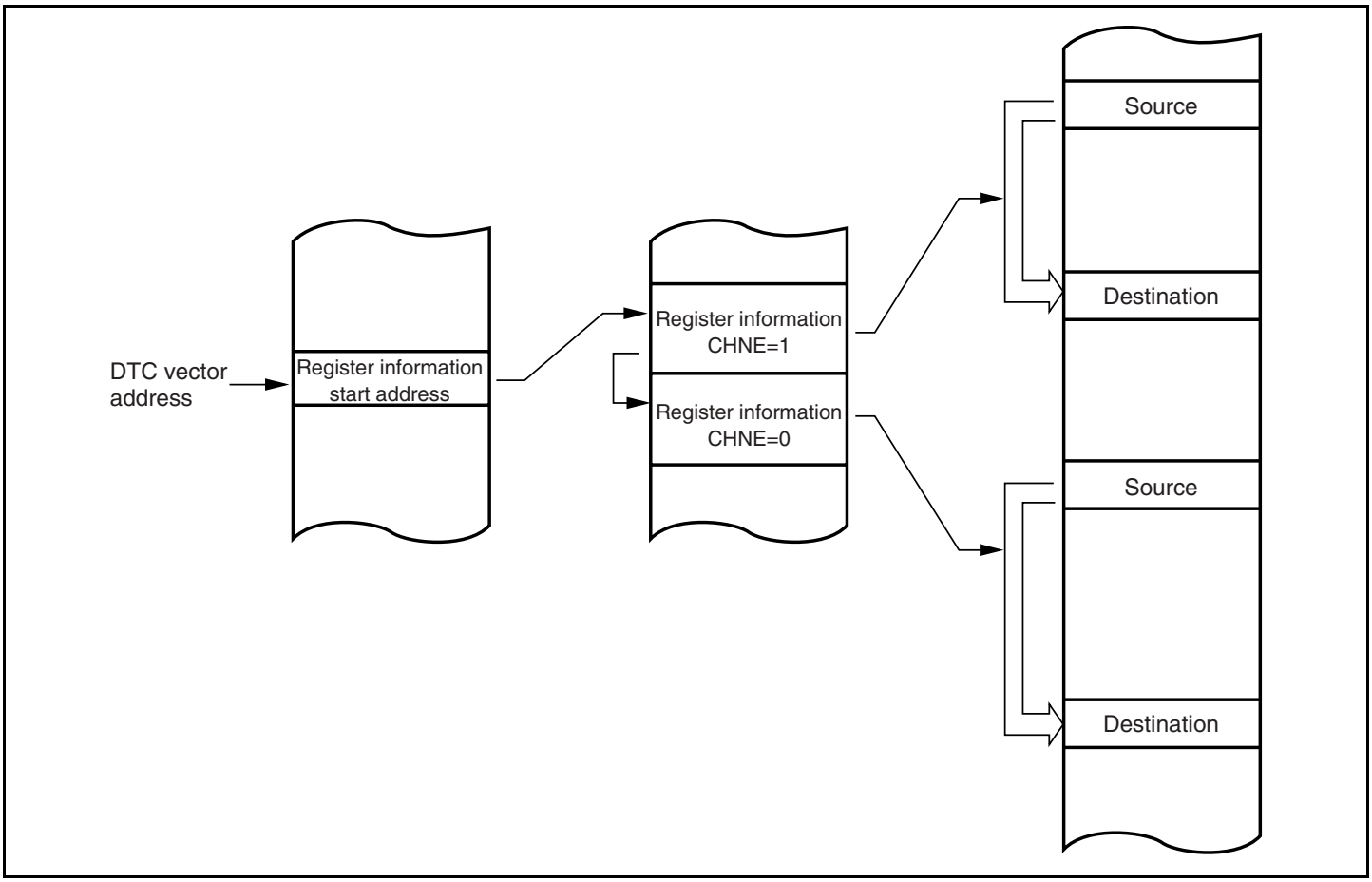


Figure 9.8 Operation of Chain Transfer

9.5.5 Interrupt Sources

An interrupt request is issued to the CPU when the DTC finishes the specified number of data transfers, or a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and interrupt controller priority level control.

In the case of activation by software, a software activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has ended, or the specified number of transfers has ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated. The interrupt handling routine should clear the SWDTE bit to 0.

When the DTC is activated by software, an SWDTEND interrupt is not generated during a data transfer wait or during data transfer even if the SWDTE bit is set to 1.

9.5.6 Operation Timing

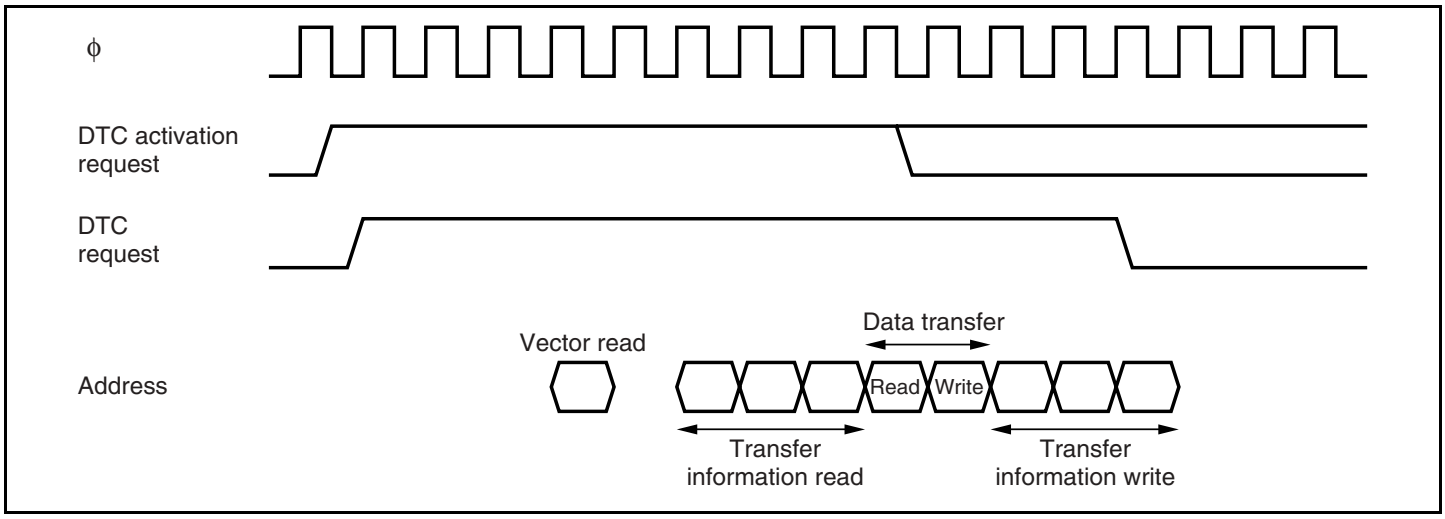


Figure 9.9 DTC Operation Timing (Example in Normal Mode or Repeat Mode)

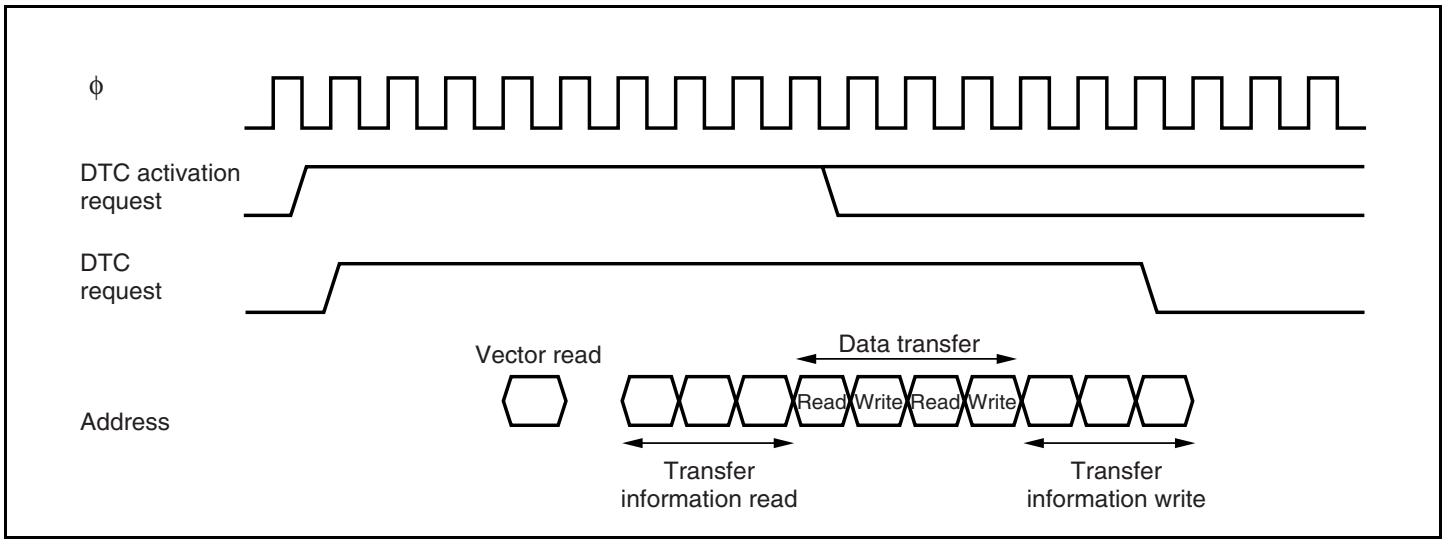


Figure 9.10 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

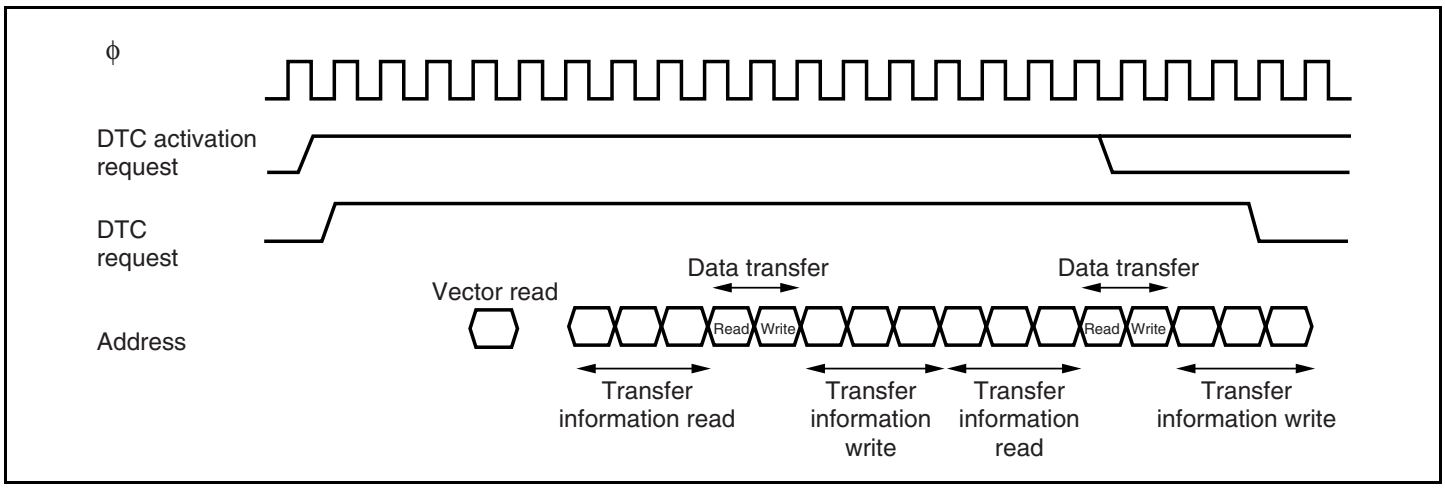


Figure 9.11 DTC Operation Timing (Example of Chain Transfer)

9.5.7 Number of DTC Execution States

Table 9.6 lists execution status for a single DTC data transfer, and table 9.7 shows the number of states required for each execution status.

Table 9.6 DTC Execution Status

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	Internal Operations M
Normal	1	6	1	1	3
Repeat	1	6	1	1	3
Block transfer	1	6	N	N	3

Legend:
N: Block size (initial setting of CRAH and CRAL)

Table 9.7 Number of States Required for Each Execution Status

Object to be Accessed		On-Chip RAM	On-Chip ROM	On-Chip I/O Registers		External Devices			
Bus width		32	16	8	16	8		16	
Access states		1	1	2	2	2	3	2	3
Execution status	Vector read S_I	—	1	—	—	4	6+2m	2	3+m
	Register information read/write S_J	1	—	—	—	—	—	—	—
	Byte data read S_K	1	1	2	2	2	3+m	2	3+m
	Word data read S_K	1	1	4	2	4	6+2m	2	3+m
	Byte data write S_L	1	1	2	2	2	3+m	2	3+m
	Word data write S_L	1	1	4	2	4	6+2m	2	3+m
	Internal operation S_M	1							

The number of execution states is calculated from the formula below. Note that Σ means the sum of all transfers activated by one activation event (the number in which the CHNE bit is set to 1, plus 1).

$$\text{Number of execution states} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$$

For example, when the DTC vector address table is located in on-chip ROM, normal mode is set, and data is transferred from the on-chip ROM to an internal I/O register, the time required for the DTC operation is 13 states. The time from activation to the end of the data write is 10 states.

9.6 Procedures for Using DTC

9.6.1 Activation by Interrupt

The procedure for using the DTC with interrupt activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the enable bits for the interrupt sources to be used as the activation sources to 1. The DTC is activated when an interrupt used as an activation source is generated.
5. After the end of one data transfer, or after the specified number of data transfers have ended, the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the DTC is to continue transferring data, set the DTCE bit to 1.

9.6.2 Activation by Software

The procedure for using the DTC with software activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Check that the SWDTE bit is 0.
4. Write 1 to SWDTE bit and the vector number to DTVECR.
5. Check the vector number written to DTVECR.
6. After the end of one data transfer, if the DISEL bit is 0 and a CPU interrupt is not requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the SWDTE bit to 1. When the DISEL bit is 1, or after the specified number of data transfers have ended, the SWDTE bit is held at 1 and a CPU interrupt is requested.

9.7 Examples of Use of the DTC

9.7.1 Normal Mode

An example is shown in which the DTC is used to receive 128 bytes of data via the SCI.

1. Set MRA to fixed source address (SM1 = SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0). Set the SCI RDR address in SAR, the start address of the RAM area where the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.
2. Set the start address of the register information at the DTC vector address.

3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The interrupt handling routine should perform wrap-up processing.

9.7.2 Chain Transfer

An example of DTC chain transfer is shown in which pulse output is performed using the PPG. Chain transfer can be used to perform pulse output data transfer and PPG output trigger cycle updating. Repeat mode transfer to the PPG's NDR is performed in the first half of the chain transfer, and normal mode transfer to the TPU's TGR in the second half. This is because clearing of the activation source and interrupt generation at the end of the specified number of transfers are restricted to the second half of the chain transfer (transfer when CHNE = 0).

1. Perform settings for transfer to the PPG's NDR. Set MRA to source addresstT5.1(o)-1T0nsf[(z0nss)6.4

9. Each time a TGRA compare match occurs, the next output value is transferred to NDR and the set value of the next output trigger period is transferred to TGRA. The activation source TGFA flag is cleared.
10. When the specified number of transfers are completed (the TPU transfer CRA value is 0), the TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is sent to the CPU. Termination processing should be performed in the interrupt handling routine.

9.7.3 Chain Transfer when Counter = 0

By executing a second data transfer, and performing re-setting of the first data transfer, only when the counter value is 0, it is possible to perform 256 or more repeat transfers.

An example is shown in which a 128-kbyte input buffer is configured. The input buffer is assumed to have been set to start at lower address H'0000. Figure 9.12 shows the chain transfer when the counter value is 0.

1. For the first transfer, set the normal mode for input data. Set fixed transfer source address (G/A, etc.), CRA = H'0000 (65,536 times), and CHNE = 1, CHNS = 1, and DISEL = 0.
2. Prepare the upper 8-bit addresses of the start addresses for each of the 65,536 transfer start addresses for the first data transfer in a separate area (in ROM, etc.). For example, if the input buffer comprises H'200000 to H'21FFFF, prepare H'21 and H'20.
3. For the second transfer, set repeat mode (with the source side as the repeat area) for re-setting the transfer destination address for the first data transfer. Use the upper 8 bits of DAR in the first register information area as the transfer destination. Set CHNE = DISEL = 0. If the above input buffer is specified as H'200000 to H'21FFFF, set the transfer counter to 2.
4. Execute the first data transfer 65,536 times by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper 8 bits of the transfer source address for the first data transfer to H'21. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
5. Next, execute the first data transfer the 65,536 times specified for the first data transfer by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper 8 bits of the transfer source address for the first data transfer to H'20. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
6. Steps 4 and 5 are repeated endlessly. As repeat mode is specified for the second data transfer, an interrupt request is not sent to the CPU.

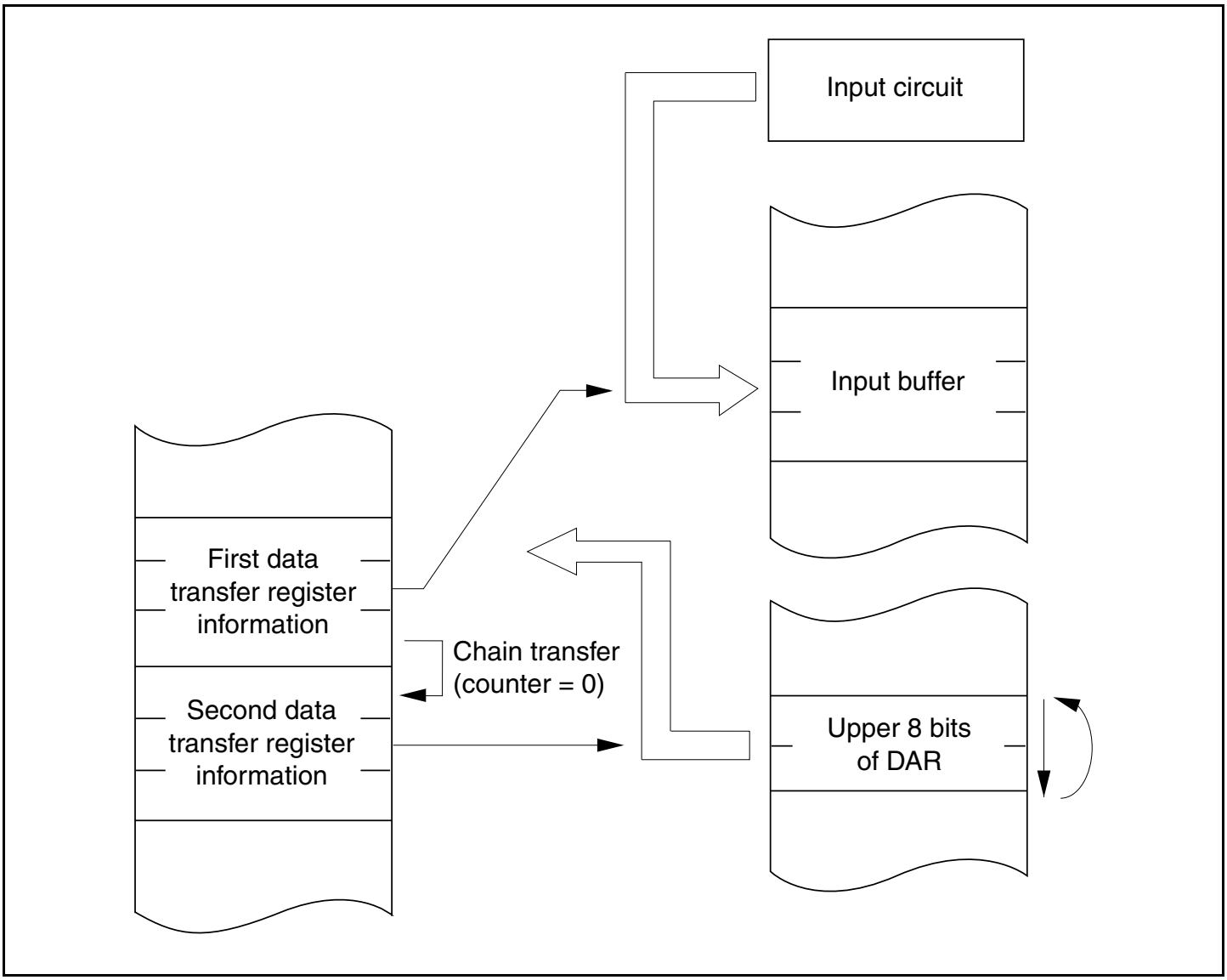


Figure 9.12 Chain Transfer when Counter = 0

9.7.4 Software Activation

An example is shown in which the DTC is used to transfer a block of 128 bytes of data by means of software activation. The transfer source address is H'1000 and the destination address is H'2000. The vector number is H'60, so the vector address is H'04C0.

1. Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt (CHNE = 0). Set the transfer source address (H'1000) in SAR, the destination address (H'2000) in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.
2. Set the start address of the register information at the DTC vector address (H'04C0).
3. Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transfer activated by software.
4. Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data is H'E0.

5. Read DTVECR again and check that it is set to the vector number (H'60). If it is not, this indicates that the write failed. This is presumably because an interrupt occurred between steps 3 and 4 and led to a different software activation. To activate this transfer, go back to step 3.
6. If the write was successful, the DTC is activated and a block of 128 bytes of data is transferred.
7. After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine should clear the SWDTE bit to 0 and perform other wrap-up processing.

9.8 Usage Notes

9.8.1 Module Stop Mode Setting

DTC operation can be disabled or enabled using the module stop control register. The initial setting is for DTC operation to be enabled. Register access is disabled by setting module stop mode. Module stop mode cannot be set while the DTC is activated. For details, refer to section 22, Power-Down Modes.

9.8.2 On-Chip RAM

The MRA, MRB, SAR, DAR, CRA, and CRB registers are all located in on-chip RAM. When the DTC is used, the RAME bit in SYSCR must not be cleared to 0.

9.8.3 DTCE Bit Setting

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all interrupts are disabled, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

- DMAC Transfer End Interrupt

  When DTC transfer is activated by a DMAC transfer end interrupt, regardless of the transfer counter and DISEL bit, the DMAC's DTE bit is not subject to DTC control, and the write data has priority. Consequently, an interrupt request may not be sent to the CPU when the DTC transfer counter reaches 0.
- Chain Transfer

  When chain transfer is used, clearing of the activation source or DTCER is performed when the last of the chain of data transfers is executed. SCI and high-speed A/D converter interrupt/activation sources, on the other hand, are cleared when the DTC reads or writes to the prescribed register.

  Therefore, when the DTC is activated by an interrupt or activation source, if a read/write of the relevant register is not included in the last chained data transfer, the interrupt or activation source will be retained.

Section 10 I/O Ports

Table 10.1 summarizes the port functions. The pins of each port also have other functions such as input/output or external interrupt input pins of on-chip peripheral modules. Each I/O port includes a data direction register (DDR) that controls input/output, a data register (DR) that stores output data, and a port register (PORT) used to read the pin states. The input-only ports do not have a DR or DDR register.

Ports A to E have a built-in pull-up MOS function and a input pull-up MOS control register (PCR) to control the on/off state of input pull-up MOS.

Ports 3 and A include an open-drain control register (ODR) that controls the on/off state of the output buffer PMOS.

Ports 1 to 3, 5 (P50 to P53), and 6 to 8 can drive a single TTL load and 30 pF capacitive load. Ports A to H can drive a single TTL load and 50 pF capacitive load.

All the I/O ports can drive a Darlington transistor when outputting data.

Ports 1 and 2 are Schmitt-triggered inputs. Ports 5,6, F (PF1, PF2), and H (PH2, PH3) are Schmitt-triggered inputs when used as the IRQ input.

Table 10.1 Port Functions

Port	Description	Modes 1 and 5	Modes 2 and 6	Mode 4	Modes 3*, 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port 1	General I/O port also functioning as PPG outputs, TPU I/Os, and EXDMAC outputs	P17/PO15/TIOCB2/TCLKD/ EDRAK3			P17/PO15/TIOCB2/ TCLKD/EDRAK3	P17/PO15/TIOCB2/ TCLKD	Schmitt-triggered input
		P16/PO14/TIOCA2/EDRAK2			P16/PO14/TIOCA2/ EDRAK2	P16/PO14/TIOCA2	
		P15/PO13/TIOCB1/TCLKC					
		P14/PO12/TIOCA1					
		P13/PO11/TIOCD0/TCLKB					
		P12/PO10/TIOCC0/TCLKA					
		P11/PO9/TIOCB0					
		P10/PO8/TIOCA0					
Port 2	General I/O port also functioning as PPG outputs, TPU I/Os, interrupt inputs, and EXDMAC outputs	P27/PO7/TIOCB5/EDRAK1/ (IRQ15)			P27/PO7/TIOCB5/ EDRAK1/(IRQ15)	P27/PO7/TIOCB5/ (IRQ15)	Schmitt-triggered input
		P26/PO6/TIOCA5/EDRAK0/ (IRQ14)			P26/PO6/TIOCA5/ EDRAK0/(IRQ14)	P26/PO6/TIOCA5/ (IRQ14)	
		P25/PO5/TIOCB4/(IRQ13)					
		P24/PO4/TIOCA4/(IRQ12)					
		P23/PO3/TIOCD3/(IRQ11)					
		P22/PO2/TIOCC3/(IRQ10)					
		P21/PO1/TIOCB3/(IRQ9)					
		P20/PO0/TIOCA3/(IRQ8)					
Port 3	General I/O port also functioning as SCI I/Os	P35/SCK1/(OE)/CKE*			P35/SCK1/(OE)/ CKE*	P35/SCK1	Open-drain output capability
		P34/SCK0					
		P33/RxD1					
		P32/RxD0/IrRxD					
		P31/TxD1					
		P30/TxD0/IrTxD					

Port	Description	Modes 1 and 5	Modes 2 and 6	Mode 4	Modes 3*, 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port 4	General I/O port also functioning as A/D converter analog inputs and D/A converter analog outputs	P47/AN7/DA1 P46/AN6/DA0 P45/AN5 P44/AN4 P43/AN3 P42/AN2 P41/AN1 P40/AN0					
Port 5	General I/O port also functioning as interrupt inputs, A/D converter analog inputs, and D/A converter analog outputs	P57/AN15/DA3/IRQ7 P56/AN14/DA2/IRQ6 P55/AN13/IRQ5 P54/AN12/IRQ4					Schmitt-triggered input when used as input
	General I/O port also functioning as interrupt inputs, A/D converter analog inputs, and SCI I/Os	P53/ADTRG/IRQ3 P52/SCK2/IRQ2 P51/RxD2/IRQ1 P50/TxD2/IRQ0					
Port 6	General I/O port also functioning as interrupt inputs, TMR I/Os, and DMAC I/Os	P65/TMO1/DACK1/IRQ13 P64/TMO0/DACK0/IRQ12 P63/TMCI1/TEND1/IRQ11 P62/TMCI0/TEND0/IRQ10 P61/TMRI1/DREQ1/IRQ9 P60/TMRI0/DREQ0/IRQ8					Schmitt-triggered input when used as input
Port 7	General I/O port also functioning as DMAC I/Os and EXDMAC I/Os	P75/EDACK1/(DACK1) P74/EDACK0/(DACK0) P73/ETEND1/(TEND1) P72/ETEND0/(TEND0) P71/EDREQ1/(DREQ1) P70/EDREQ0/(DREQ0)			P75/EDACK1/(DACK1) P74/EDACK0/(DACK0) P73/ETEND1/(TEND1) P72/ETEND0/(TEND0) P71/EDREQ1/(DREQ1) P70/EDREQ0/(DREQ0)	P75/(DACK1) P74/(DACK0) P73/(TEND1) P72/(TEND0) P71/(DREQ1) P70/(DREQ0)	

Port	Description	Modes 1 and 5	Modes 2 and 6	Mode 4	Modes 3*, 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port 8	General I/O port also functioning as EXDMAC I/Os and interrupt inputs	P85/EDACK3/IRQ5			P85/EDACK3/IRQ5	P85/IRQ5	
		P84/EDACK2/IRQ4			P84/EDACK2/IRQ4	P84/IRQ4	
		P83/ETEND3/IRQ3			P83/ETEND3/IRQ3	P83/IRQ3	
		P82/ETEND2/IRQ2			P82/ETEND2/IRQ2	P82/IRQ2	
		P81/EDREQ3/IRQ1			P81/EDREQ3/IRQ1	P81/IRQ1	
		P80/EDREQ2/IRQ0			P80/EDREQ2/IRQ0	P80/IRQ0	
Port A	General I/O port also functioning as address outputs	PA7/A23		PA7/A23	PA7/A23	PA7	Built-in input pull-up MOS Open-drain output capability
		PA6/A22		PA6/A22	PA6/A22	PA6	
		PA5/A21		PA5/A21	PA5/A21	PA5	
		A20		PA4/A20	PA4/A20	PA4	
		A19		PA3/A19	PA3/A19	PA3	
		A18		PA2/A18	PA2/A18	PA2	
		A17		PA1/A17	PA1/A17	PA1	
		A16		PA0/A16	PA0/A16	PA0	
Port B	General I/O port also functioning as address outputs	A15		PB7/A15	PB7/A15	PB7	Built-in input pull-up MOS
		A14		PB6/A14	PB6/A14	PB6	
		A13		PB5/A13	PB5/A13	PB5	
		A12		PB4/A12	PB4/A12	PB4	
		A11		PB3/A11	PB3/A11	PB3	
		A10		PB2/A10	PB2/A10	PB2	
		A9		PB1/A9	PB1/A9	PB1	
		A8		PB0/A8	PB0/A8	PB0	
Port C	General I/O port also functioning as address outputs	A7		PC7/A7	PC7/A7	PC7	Built-in input pull-up MOS
		A6		PC6/A6	PC6/A6	PC6	
		A5		PC5/A5	PC5/A5	PC5	
		A4		PC4/A4	PC4/A4	PC4	
		A3		PC3/A3	PC3/A3	PC3	
		A2		PC2/A2	PC2/A2	PC2	
		A1		PC1/A1	PC1/A1	PC1	
		A0		PC0/A0	PC0/A0	PC0	

Port	Description	Modes 1 and 5	Modes 2 and 6	Mode 4	Modes 3*, 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port D	General I/O port also functioning as data I/Os	D15			D15	PD7	Built-in input pull-up MOS
		D14			D14	PD6	
		D13			D13	PD5	
		D12			D12	PD4	
		D11			D11	PD3	
		D10			D10	PD2	
		D9			D9	PD1	
		D8			D8	PD0	
Port E	General I/O port also functioning as data I/Os	D7	PE7/D7	PE7/D7	PE7/D7	PE7	Built-in input pull-up MOS
		D6	PE6/D6	PE6/D6	PE6/D6	PE6	
		D5	PE5/D5	PE5/D5	PE5/D5	PE5	
		D4	PE4/D4	PE4/D4	PE4/D4	PE4	
		D3	PE3/D3	PE3/D3	PE3/D3	PE3	
		D2	PE2/D2	PE2/D2	PE2/D2	PE2	
		D1	PE1/D1	PE1/D1	PE1/D1	PE1	
		D0	PE0/D0	PE0/D0	PE0/D0	PE0	
Port F	General I/O port also functioning as interrupt inputs and bus control I/Os	PF7/ϕ			PF7/ϕ	PF7/ϕ	Only PF1 and PF2 are Schmitt-triggered inputs when used as the IRQ input
		PF6/AS			PF6/AS	PF6	
		RD			RD	PF5	
		HWR			HWR	PF4	
		PF3/LWR			PF3/LWR	PF3	
		PF2/LCAS/DQML*/IRQ15			PF2/LCAS/DQML*/ IRQ15	PF2/IRQ15	
		PF1/UCAS/DQMU*/IRQ14			PF1/UCAS/DQMU*/ IRQ14	PF1/IRQ14	
		PF0/WAIT			PF0/WAIT	PF0	
Port G	General I/O port also functioning as bus control I/Os	PG6/BREQ			PG6/BREQ	PG6	
		PG5/BACK			PG5/BACK	PG5	
		PG4/BREQO			PG4/BREQO	PG4	
		PG3/CS3/RAS3/CAS*			PG3/CS3/RAS3/ CAS*	PG3	
		PG2/CS2/RAS2/RAS*				PG2	
		PG1/CS1			PG2/CS2/RAS2/ RAS*	PG1	
		PG0/CS0				PG0	
					PG1/CS1		
					PG0/CS0		

Port	Description	Modes 1 and 5	Modes 2 and 6	Mode 4	Modes 3*, 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port H	General I/O port also functioning as interrupt inputs and bus control I/Os	PH3/CS7/OE/CKE*/(IRQ7)			PH3/CS7/OE/CKE*/ (IRQ7)	PH3/(IRQ7)	Only PH2 and PH3 are Schmitt-triggered inputs when used as the IRQ input
		PH2/CS6/(IRQ6)			PH2/CS6/(IRQ6)	PH2/(IRQ6)	
		PH1/CS5/RAS5/SDRAMϕ*			PH1/CS5/RAS5/ SDRAMϕ*	PH1/SDRAMϕ*	
		PH0/CS4/RAS4/WE*			PH0/CS4/RAS4/ WE*	PH0	

Note: Only in H8S/2678R Series.

10.1 Port 1

Port 1 is an 8-bit I/O port that also has other functions. The port 1 has the following registers.

- Port 1 data direction register (P1DDR)
- Port 1 data register (P1DR)
- Port 1 register (PORT1)

10.1.1 Port 1 Data Direction Register (P1DDR)

The individual bits of P1DDR specify input or output for the pins of port 1.

P1DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
6	P16DDR	0	W	
5	P15DDR	0	W	
4	P14DDR	0	W	
3	P13DDR	0	W	
2	P12DDR	0	W	
1	P11DDR	0	W	
0	P10DDR	0	W	

10.1.2 Port 1 Data Register (P1DR)

P1DR stores output data for the port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	P16DR	0	R/W	
5	P15DR	0	R/W	
4	P14DR	0	R/W	
3	P13DR	0	R/W	
2	P12DR	0	R/W	
1	P11DR	0	R/W	
0	P10DR	0	R/W	

10.1.3 Port 1 Register (PORT1)

PORT1 shows the pin states.

PORT1 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	Undefined*	R	If a port 1 read is performed while P1DDR bits are set to 1, the P1DR values are read. If a port 1 read is performed while P1DDR bits are cleared to 0, the pin states are read.
6	P16	Undefined*	R	
5	P15	Undefined*	R	
4	P14	Undefined*	R	
3	P13	Undefined*	R	
2	P12	Undefined*	R	
1	P11	Undefined*	R	
0	P10	Undefined*	R	

Note: Determined by the states of pins P17 to P10.

10.1.4 Pin Functions

Port 1 pins also function as PPG outputs, TPU I/Os, and EXDMAC outputs. The correspondence between the register specification and the pin functions is shown below.

- P17/PO15/TIOCB2/TCLKD/EDRAK3

  The pin function is switched as shown below according to the combination of the TPU channel 2 settings (by bits MD3 to MD0 in TMDR2, bits IOB3 to IOB0 in TIOR2, and bits CCLR1 and CCLR0 in TCR2), bits TPSC2 to TPSC0 in TCR0 and TCR5, bit NDER15 in NDERH, bit EDRAKE in EDMDR3, and bit P17DDR.

Modes 1, 2, 3*[3] (EXPE = 1), 4, 5, 6, 7 (EXPE = 1)

EDRAKE	0				1
TPU channel 2 settings	(1) in table below	(2) in table below			—
P17DDR	—	0	1	1	—
NDER15	—	—	0	1	—
Pin function	TIOCB2 output	P17 input	P17 output	PO15 output	EDRAK3 output
		TIOCB2 input*[1]			
	TCLKD input*[2]				

Modes 3*[3] (EXPE = 0), 7 (EXPE = 0)

EDRAKE	—			
TPU channel 2 settings	(1) in table below	(2) in table below		
P17DDR	—	0	1	1
NDER15	—	—	0	1
Pin function	TIOCB2 output	P17 input	P17 output	PO15 output
		TIOCB2 input*[1]		
	TCLKD input*[2]			

Notes: 1. TIOCB2 input when MD3 to MD0 = B'0000 or B'01xx and IOB3 = 1.
2. TCLKD input when the setting for either TCR0 or TCR5 is TPSC2 to TPSC0 = B'111. TCLKD input when channels 2 and 4 are set to phase counting mode.
3. Only in H8S/2678R Series.

TPU channel 2 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

- P16/PO14/TIOCA2/EDRAK2

  The pin function is switched as shown below according to the combination of the TPU channel 2 settings (by bits MD3 to MD0 in TMDR2, bits IOB3 to IOB0 in TIOR2, and bits CCLR1 and CCLR0 in TCR2), bit NDER14 in NDERH, bit EDRAKE in EDMDR2 and bit P16DDR.

Modes 1, 2, 3*[3] (EXPE = 1), 4, 5, 6, 7 (EXPE = 1)

EDRAKE	0				1
TPU channel 2 settings	(1) in table below	(2) in table below			—
P16DDR	—	0	1	1	—
NDER14	—	—	0	1	—
Pin function	TIOCA2 output	P16 input	P16 output	PO14 output	EDRAK2 output
	TIOCA input*[1]				

Modes 3*[3] (EXPE = 0), 7 (EXPE = 0)

EDRAKE	—			
TPU channel 2 settings	(1) in table below	(2) in table below		
P16DDR	—	0	1	1
NDER14	—	—	0	1
Pin function	TIOCA2 output	P16 input	P16 output	PO14 output
		TIOCA2 input*[1]		

Note: 1. TIOCA2 input when MD3 to MD0 = B'0000 or B'01xx and IOA3 = 1.

TPU channel 2 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

x: Don’t care

Notes: 2. TIOCB2 output disabled.
3. Only in H8S/2678R Series.

- P15/PO13/TIOCB1/TCLKC

  The pin function is switched as shown below according to the combination of the TPU channel 1 settings (by bits MD3 to MD0 in TMDR1, bits IOB3 to IOB0 in TIOR1, and bits CCLR1

TPU channel 1 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don’t care

- P14/PO12/TIOCA1

  The pin function is switched as shown below according to the combination of the TPU channel 1 settings (by bits MD3 to MD0 in TMDR1, bits IOA3 to IOA0 in TIOR1, and bits CCLR1 and CCLR0 in TCR1), bit NDER12 in NDERH, and bit P14DDR.

Th0937 83an276 TD07(n)CCL2 . low
The pin fan23 Tw 697035 PCCL4CL(P)-60.6i2)7.

	TPU channel 1 settings	(1) in table below	(2) in table below	

- P13/PO11/TIOCD0/TCLKB

  The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR0, bits IOD3 to IOD0 in TIOR0L, and bits CCLR2 to CCLR0 in TCR0), bits TPSC2 to TPSC0 in TCR0 to TCR2, bit NDER11 in NDERH, and bit P13DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P13DDR	—	0	1	1
NDER11	—	—	0	1
Pin function	TIOCD0 output	P13 input	P13 output	PO11 output
		TIOCD0 input*[1]		
	TCLKB input*[2]			

Notes: 1. TIOCD0 input when MD3 to MD0 = B'0000 or B’01xx and IOD3 to IOD0 = B'10xx.
2. TCLKB input when the setting for any of TCR0 to TCR2 is TPSC2 to TPSC0 = B'101. TCLKB input when phase counting mode is set for channels 1 and 5.

TPU channel 0 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOD3 to IOD0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2, CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don’t care

- P12/PO10/TIOCC0/TCLKA

  The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR0, bits IOC3 to IOC0 in TIOR0L, and bits CCLR2 to CCLR0 in TCR0), bits TPSC2 to TPSC0 in TCR0 to TCR5, bit NDER10 in NDERH, and bit P12DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P12DDR	—	0	1	1
NDER10	—	—	0	1
Pin function	TIOCC0 output	P12 input	P12 output	PO10 output
		TIOCC0 input*[1]		
	TCLKA input*[2]			

Notes: 1. TIOCC0 input when MD3 to MD0 = B'0000 and IOC3 to IOC0 = B'10xx.
2. TCLKA input when the setting for any of TCR0 to TCR5 is TPSC2 to TPSC0 = B'100.
TCLKA input when phase counting mode is set for channels 1 and 5.

TPU channel 0 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2, CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM*[3] mode 1 output	PWM mode 2 output	—

x: Don't care

Note: 3. TIOCD0 output disabled.
Output disabled and settings (2) effective when BFA = 1 or BFB = 1 in TMDR0.

- P11/PO9/TIOCB0

  The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR0 and bits IOB3 to IOB0 in TIOR0H), bit NDER9 in NDERH, and bit P11DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P11DDR	—	0	1	1
NDER9	—	—	0	1
Pin function	TIOCB0 output	P11 input	P11 output	PO9 output
		TIOCB0 input*[1]		

Note: 1. TIOCB0 input when MD3 to MD0 = B'0000 and IOB3 to IOB0 = B'10xx.

TPU channel 0 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2, CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

- P10/PO8/TIOCA0

  The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR0, bits IOA3 to IOA0 in TIOR0H, and bits CCLR2 to CCLR0 in TCR0), bit NDER8 in NDERH, and bit P10DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P10DDR	—	0	1	1
NDER8	—	—	0	1
Pin function	TIOCA0 output	P10 input	P10 output	PO8 output
		TIOCA0 input*[1]		

Note: 1. TIOCA0 input when MD3 to MD0 = B'0000 and IOA3 to IOA0 = B'10xx.

TPU channel 0 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2, CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

x: Don't care

Note: 2. TIOCB0 output disabled.

10.2 Port 2

Port 2 is an 8-bit I/O port that also has other functions. The port 2 has the following registers.

- Port 2 data direction register (P2DDR)
- Port 2 data register (P2DR)
- Port 2 register (PORT2)

10.2.1 Port 2 Data Direction Register (P2DDR)

The individual bits of P2DDR specify input or output for the pins of port 2.

P2DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
6	P26DDR	0	W	
5	P25DDR	0	W	
4	P24DDR	0	W	
3	P23DDR	0	W	
2	P22DDR	0	W	
1	P21DDR	0	W	
0	P20DDR	0	W	

10.2.2 Port 2 Data Register (P2DR)

P2DR stores output data for the port 2 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	P26DR	0	R/W	
5	P25DR	0	R/W	
4	P24DR	0	R/W	
3	P23DR	0	R/W	
2	P22DR	0	R/W	
1	P21DR	0	R/W	
0	P20DR	0	R/W	

10.2.3 Port 2 Register (PORT2)

PORT2 shows the pin states.

PORT2 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P27	Undefined*	R	If a port 2 read is performed while P2DDR bits are set to 1, the P2DR values are read. If a port 2 read is performed while P2DDR bits are cleared to 0, the pin states are read.
6	P26	Undefined*	R	
5	P25	Undefined*	R	
4	P24	Undefined*	R	
3	P23	Undefined*	R	
2	P22	Undefined*	R	
1	P21	Undefined*	R	
0	P20	Undefined*	R	

Note: Determined by the states of pins P27 to P20.

10.2.4 Pin Functions

Port 2 pins also function as PPG outputs, TPU I/Os, interrupt inputs, and EXDMAC outputs. The correspondence between the register specification and the pin functions is shown below.

- P27/PO7/TIOCB5/IRQ5/EDRAK1

  The pin function is switched as shown below according to the combination of the TPU channel 5 settings (by bits MD3 to MD0 in TMDR5, bits IOB3 to IOB0 in TIOR5, and bits CCLR1 and CCLR0 in TCR5), bit NDER7 in NDERL, bit EDRAKE in EDMDR1, bit P27DDR, and bit ITS15 in ITSR.

Modes 1, 2, 3*[3] (EXPE = 1), 4, 5, 6, 7 (EXPE = 1)

EDRAKE	0				1
TPU channel 5 settings	(1) in table below	(2) in table below			—
P27DDR	—	0	1	1	—
NDER7	—	—	0	1	—
Pin function	TIOCB5 output	P27 input	P27 output	PO7 output	EDRAK1 output
		TIOCB5 input*[1]			
	IRQ5 interrupt input*[2]				

Mode, 3*[3] (EXPE = 0), 7 (EXPE = 0)

EDRAKE	—			
TPU channel 5 settings	(1) in table below	(2) in table below		
P27DDR	—	0	1	1
NDER7	—	—	0	1
Pin function	TIOCB5 output	P27 input	P27 output	PO7 output
		TIOCB5 input*[1]		
	IRQ5 interrupt input*[2]			

Notes: 1. TIOCB5 input when MD3 to MD0 = B'0000 or B'01xx and IOB3 = 1.
2. IRQ5 input when ITS15 = 1.
3. Only in H8S/2678R Series.

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

- P26/PO6/TIOCA5/IRQ14/EDRAK0

  The pin function is switched as shown below according to the combination of the TPU channel 5 settings (by bits MD3 to MD0 in TMDR5, bits IOA3 to IOA0 in TIOR5, and bits CCLR1 and CCLR0 in TCR5), bit NDER6 in NDERL, bit EDRAKE in EDMDR0, bit P26DDR, and bit ITS14 in ITSR.

Modes 1, 2, 3*[4] (EXPE = 1), 4, 5, 6, 7 (EXPE = 1)

EDRAKE	0				1
TPU channel 5 settings	(1) in table below	(2) in table below			—
P26DDR	—	0	1	1	—
NDER6	—	—	0	1	—
Pin function	TIOCA5 output	P26 input	P26 output	PO6 output	EDRAK0 output
		TIOCA input*[1]			
	IRQ14 interrupt input*[2]				

Modes 3*[4] (EXPE = 0), 7 (EXPE = 0)

EDRAKE	—			
TPU channel 5 settings	(1) in table below	(2) in table below		
P26DDR	—	0	1	1
NDER6	—	—	0	1
Pin function	TIOCA5 output	P26 input	P26 output	PO6 output
		TIOCA5 input*[1]		
	IRQ14 interrupt input*[2]			

Notes: 1. TIOCA5 input when MD3 to MD0 = B'0000 or B'01xx and IOA3 = 1.
2. IRQ14 input when ITS14 = 1.

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*[3] mode 1 output	PWM mode 2 output	—

x: Don't care

Note: 3. TIOCB5 output disabled.
4. Only in H8S/2678R Series.

- P25/PO5/TIOCB4/IRQ13

  The pin function is switched as shown below according to the combination of the TPU channel 4 settings (by bits MD3 to MD0 in TMDR4, bits IOB3 to IOB0 in TIOR4, and bits CCLR1 and CCLR0 in TCR4), bit NDER5 in NDERL, bit P25DDR, and bit ITS13 in ITSR.

TPU channel 4 settings	(1) in table below	(2) in table below		
P25DDR	—	0	1	1
NDER5	—	—	0	1
Pin function	TIOCB4 output	P25 input	P25 output	PO5 output
		TIOCB4 input*[1]		
	IRQ13 interrupt input*[2]			

Notes: 1. TIOCB4 input when MD3 to MD0 = B'0000 or B'01xx and IOB3 to IOB0 = B'10xx.
2. IRQ13 input when ITS13 = 1.

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

- P24/PO4/TIOCA4/IRQ12

  The pin function is switched as shown below according to the combination of the TPU channel 4 settings (by bits MD3 to MD0 in TMDR4 and bits IOA3 to IOA0 in TIOR4), bit NDER4 in NDERL, bit P24DDR, and bit ITS12 in ITSR.

TPU channel 4 settings	(1) in table below	(2) in table below		
P24DDR	—	0	1	1
NDER4	—	—	0	1
Pin function	TIOCA4 output	P24 input	P24 output	PO4 output
		TIOCA4 input*[1]		
	IRQ12 interrupt input*[2]			

Notes: 1. TIOCA4 input when MD3 to MD0 = B'0000 or B'01xx and IOA3 to IOA0 = B'10xx.
2. IRQ12 input when ITS12 = 1.

TPU channel 4 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*[3] mode 1 output	PWM mode 2 output	—

x: Don't care

Note: 3. TIOCB4 output disabled.

- P23/PO3/TIOCD3/IRQ11

  The pin function is switched as shown below according to the combination of the TPU channel 3 settings (by bits MD3 to MD0 in TMDR3, bits IOD3 to IOD0 in TIOR3L, and bits CCLR2 to CCLR0 in TCR3), bit NDER3 in NDERL, bit P23DDR, and bit ITS11 in ITSR.

TPU channel 3 settings	(1) in table below	(2) in table below		
P23DDR	—	0	1	1
NDER3	—	—	0	1
Pin function	TIOCD3 output	P23 input	P23 output	PO3 output
		TIOCD3 input*[1]		
	IRQ11 interrupt input*[2]			

Notes: 1. TIOCD3 input when MD3 to MD0 = B'0000 and IOD3 to IOD0 = B'10xx.
2. IRQ11 input when ITS11 = 1.

TPU channel 3 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOD3 to IOD0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

- P22/PO2/TIOCC3/IRQ10

  The pin function is switched as shown below according to the combination of the TPU channel 3 settings (by bits MD3 to MD0 in TMDR3, bits IOC3 to IOC0 in TIOR3L, and bits CCLR2 to CCLR0 in TCR3), bit NDER2 in NDERL, bit P22DDR, and bit ITS10 in ITSR.

TPU channel 3 settings	(1) in table below	(2) in table below		
P22DDR	—	0	1	1
NDER2	—	—	0	1
Pin function	TIOCC3 output	P22 input	P22 output	PO2 output
		TIOCC3 input*[1]		
	IRQ10 interrupt input*[2]			

Notes: 1. TIOCC3 input when MD3 to MD0 = B'0000 and IOC3 to IOC0 = B'10xx.
2. IRQ10 input when ITS10 = 1.

TPU channel 3 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM*[3] mode 1 output	PWM mode 2 output	—

x: Don't care

Note: 3. TIOCD3 output disabled.
Output disabled and settings (2) effective when BFA = 1 or BFB = 1 in TMDR3.

- P21/PO1/TIOCB3/IRQ9

  The pin function is switched as shown below according to the combination of the TPU channel 3 settings (by bits MD3 to MD0 in TMDR3, bits IOB3 to IOB0 in TIOR3H, and bits CCLR2 to CCLR0 in TCR3), bit NDER1 in NDERL, bit P21DDR, and bit ITS9 in ITSR.

TPU channel 3 settings	(1) in table below	(2) in table below		
P21DDR	—	0	1	1
NDER1	—	—	0	1
Pin function	TIOCB3 output	P21 input	P21 output	PO1 output
		TIOCB3 input*[1]		
	IRQ9 interrupt input*[2]			

Notes: 1. TIOCB3 input when MD3 to MD0 = B'0000 and IOB3 to IOB0 = B'10xx.
2. IRQ9 input when ITS9 = 1.

TPU channel 3 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

- P20/PO0/TIOCA3/IRQ8

  The pin function is switched as shown below according to the combination of the TPU channel 3 settings (by bits MD3 to MD0 in TMDR3, bits IOA3 to IOA0 in TIOR3H, and bits CCLR2 to CCLR0 in TCR3), bit NDER0 in NDERL, bit P20DDR, and bit ITS8 in ITSR.

TPU channel 3 settings	(1) in table below	(2) in table below		
P20DDR	—	0	1	1
NDER0	—	—	0	1
Pin function	TIOCA3 output	P20 input	P20 output	PO0 output
		TIOCA3 input*[1]		
	IRQ8 interrupt input*[2]			

Notes: 1. TIOCA3 input when MD3 to MD0 = B'0000 and IOA3 to IOA0 = B'10xx.
2. IRQ8 input when ITS8 = 1.

TPU channel 3 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM*[3] mode 1 output	PWM mode 2 output	—

x: Don't care

Note: 3. TIOCB3 output disabled.

10.3 Port 3

Port 3 is a 6-bit I/O port that also has other functions. The port 3 has the following registers.

- Port 3 data direction register (P3DDR)
- Port 3 data register (P3DR)
- Port 3 register (PORT3)
- Port 3 open drain control register (P3ODR)
- Port function control register 2(PFCR2)

10.3.1 Port 3 Data Direction Register (P3DDR)

The individual bits of P3DDR specify input or output for the pins of port 3.

P3DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P35DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
4	P34DDR	0	W	
3	P33DDR	0	W	
2	P32DDR	0	W	
1	P31DDR	0	W	
0	P30DDR	0	W	

10.3.2 Port 3 Data Register (P3DR)

P3DR stores output data for the port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P35DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
4	P34DR	0	R/W	
3	P33DR	0	R/W	
2	P32DR	0	R/W	
1	P31DR	0	R/W	
0	P30DR	0	R/W	

10.3.3 Port 3 Register (PORT3)

PORT3 shows the pin states.

PORT3 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P35	Undefined*	R	If a port 3 read is performed while P3DDR bits are set to 1, the P3DR values are read. If a port 1 read is performed while P3DDR bits are cleared to 0, the pin states are read.
4	P34	Undefined*	R	
3	P33	Undefined*	R	
2	P32	Undefined*	R	
1	P31	Undefined*	R	
0	P30	Undefined*	R	

Note: Determined by the states of pins P35 to P30.

10.3.4 Port 3 Open Drain Control Register (P3ODR)

P3ODR controls the output status for each port 3 pin.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P35ODR	0	R/W	Setting a P3ODR bit to 1 makes the corresponding port 3 pin an NMOS open-drain output pin, while clearing the bit to 0 makes the pin a CMOS output pin.
4	P34ODR	0	R/W	
3	P33ODR	0	R/W	
2	P32ODR	0	R/W	
1	P31ODR	0	R/W	
0	P30ODR	0	R/W	

10.3.5 Port Function Control Register 2 (PFCR2)

P3ODR controls the I/O port.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	ASOE	1	R/W	AS Output Enable Selects to enable or disable the AS output pin. 0: PF6 is designated as I/O port 1: PF6 is designated as AS output pin
2	LWROE	1	R/W	LWR Output Enable Selects to enable or disable the LWR output pin. 0: PF3 is designated as I/O port 1: PF3 is designated as LWR output pin
1	OES	1	R/W	OE Output Select Selects the OE output pin port when the OEE bit is set to 1 in DRAMCR (enabling OE/CKE* output). 0: P35 is designated as OE output pin 1: PH3 is designated as OE/CKE* output pin
0	DMACS	0	R/W	DMAC Control Pin Select Selects the DMAC control I/O port. 0: PF65 to PF60 are designated as DMAC control pins 1: PF75 to PF70 are designated as DMAC control pins

Note: Only in H8S/2678R Series.

10.3.6 Pin Functions

Port 3 pins also function as SCI I/Os and a bus control signal output. The correspondence between the register specification and the pin functions is shown below.

- P35/SCK1/OE/CKE

The pin function is switched as shown below according to the combination of the C/A bit in SMR of SCI_1, bits CKE0 and CKE1 in SCR, bits RMTS2 to RMTS0 in DRAMCR, bit OES in PFCR2, and bit P35DDR.

Modes 1, 2, 3 (EXPE = 1), 4, 5, 6, 7 (EXPE = 1), H8S/2678R Series

OEE	0					1						
OES	—					1					0	
Area 2 to 5	—					—					Normal space or DRAM space	Continuous synchronous DRAM space
CKE1	0				1	0				1	—	—
C/A	0			1	—	0			1	—	—	—
CKE0	0		1	—	—	0		1	—	—	—	—
P35DDR	0	1	—	—	—	0	1	—	—	—	—	—
Pin function	P35 input	P35 output*[1]	SCK1 output*[1]	SCK1 output*[1]	SCK1 input	P35 input	P35 output*[1]	SCK1 output*[1]	SCK1 output*[1]	SCK1 input	OE output	CKE output

Modes 1, 2, 4, 5, 6, 7 (EXPE = 1), H8S/2678 Series

OEE	0					1					
OES	—					1					0
CKE1	0				1	0				1	—
C/A	0			1	—	0			1	—	—
CKE0	0		1	—	—	0		1	—	—	—
P35DDR	0	1	—	—	—	0	1	—	—	—	—
Pin function	P35 input	P35 output*[1]	SCK1 output*[1]	SCK1 output*[1]	SCK1 input	P35 input	P35 output*[1]	SCK1 output*[1]	SCK1 output*[1]	SCK1 input	OE output

Note: 1. NMOS open-drain output when P35ODR = 1.

Modes 3*[2] (EXPE = 0), 7 (EXPE = 0)

OEE	—				
OES	—				
CKE1	0				—
C/A	0			1	—
CKE0	0		1	—	—
P35DDR	0	1	—	—	—
Pin function	P35 input	P35 output*[1]	SCK1 output*[1]	SCK1 output*[1]	SCK1 input

Notes: 1. NMOS open-drain output when P35ODR = 1.
2. Only in H8S/2678R Series.

- P34/SCK0

The pin function is switched as shown below according to the combination of bit C/A in SMR of SCI_0, bits CKE0 and CKE1 in SCR, and bit P34DDR.

CKE1	0				1
C/A	0			1	—
CKE0	0		1	—	—
P34DDR	0	1	—	—	—
Pin function	P34 input	P34 output*	SCK0 output*	SCK0 output*	SCK0 input

Note: NMOS open-drain output when P34ODR = 1.

- P33/RxD1

The pin function is switched as shown below according to the combination of bit RE in SCR of SCI_1 and bit P33DDR.

RE	0		1
P33DDR	0	1	—
Pin function	P33 input	P33 output*	RxD1 input

Note: NMOS open-drain output when P33ODR = 1.

- P32/RxD0/IrRxD

The pin function is switched as shown below according to the combination of bit RE in SCR of SCI_0 and bit P32DDR.

RE	0		1
P32DDR	0	1	—
Pin function	P32 input	P32 output*	RxD0/IrRxD input

Note: NMOS open-drain output when P32ODR = 1.

- P30/TxD0/IrTxD

The pin function is switched as shown below according to the combination of bit TE in SCR of SCI_0 and bit P30DDR.

TE	0		1
P30DDR	0	1	—
Pin function	P30 input	P30 output*	RxD0/IrRxD output*

Note: NMOS open-drain output when P30ODR = 1.

10.4 Port 4

Port 4 is an 8-bit input-only port. Port 4 has the following register.

- Port 4 register (PORT4)

10.4.1 Port 4 Register (PORT4)

PORT4 is an 8-bit read-only register that shows port 4 pin states.

PORT4 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P47	Undefined*	R	The pin states are always read when a port 4 read is performed.
6	P46	Undefined*	R	
5	P45	Undefined*	R	
4	P44	Undefined*	R	
3	P43	Undefined*	R	
2	P42	Undefined*	R	
1	P41	Undefined*	R	
0	P40	Undefined*	R	

Note: Determined by the states of pins P47 to P40.

10.4.2 Pin Functions

Port 4 also functions as the A/D converter analog input and D/A converter analog output. The correspondence between pins are as follows.

- P47/AN7/DA1

Pin function	AN7 input
	DA1 output

- P46/AN6/DA0

Pin function	AN6 input
	DA0 output

- P45/AN5

Pin function	AN5 input

- P44/AN4

Pin function	AN4 input

- P43/AN3

Pin function	AN3 input

- P42/AN2

Pin function	AN2 input

- P41/AN1

Pin function	AN1 input

- P40/AN0

Pin function	AN0 input

10.5 Port 5

Port 5 comprises a 4-bit I/O port (P53 to P50) and a 4-bit input-only port (P57 to P54). The 4-bit input-only port does not have the data direction register and data register. The port 5 has the following registers.

- Port 5 data direction register (P5DDR)
- Port 5 data register (P5DR)
- Port 5 register (PORT5)

10.5.1 Port 5 Data Direction Register (P5DDR)

The individual bits of P5DDR specify input or output for the pins of port 5.

P5DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	P53DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
2	P52DDR	0	W	
1	P51DDR	0	W	
0	P50DDR	0	W	

10.5.2 Port 5 Data Register (P5DR)

P5DR stores output data for the port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	P53DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
2	P52DR	0	R/W	
1	P51DR	0	R/W	
0	P50DR	0	R/W	

10.5.3 Port 5 Register (PORT5)

PORT5 shows the pin states.

PORT5 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P57	Undefined*	R	When bits P57 to P54 are read, the pin states are always read from bits 7 to 4.
6	P56	Undefined*	R	
5	P55	Undefined*	R	
4	P54	Undefined*	R	
3	P53	Undefined*	R	If bits P53 to P50 are read while P5DDR bits are set to 1, the P5DR values are read. If a port 5 read is performed while P5DDR bits are cleared to 0, the pin states are read.
2	P52	Undefined*	R	
1	P51	Undefined*	R	
0	P50	Undefined*	R	

Note: Determined by the states of pins P57 to P50.

10.5.4 Pin Functions

Port 5 pins also function as SCI I/Os, A/D converter inputs, A/D converter analog inputs, D/A converter analog outputs, and interrupt inputs. The correspondence between the register specification and the pin functions is shown below.

- P57/AN15/DA3/IRQ7

The pin function is switched as shown below according to bit ITS7 in ITSR.

Pin function	IRQ7 interrupt input pin*
	AN15 input
	DA3 output

Note: IRQ7 input when ITS7 = 0.

- P56/AN14/DA2/IRQ6

The pin function is switched as shown below according to bit ITS6 in ITSR.

Pin function	IRQ6 interrupt input pin*
	AN14 input
	DA2 output

Note: IRQ6 input when ITS6 = 0.

- P55/AN13/IRQ5

The pin function is switched as shown below according to bit ITS5 in ITSR.

Pin function	IRQ5 interrupt input*
	AN13 input

Note: IRQ5 input when ITS5 = 0.

- P54/AN12/IRQ4

The pin function is switched as shown below according to bit ITS4 in ITSR.

Pin function	IRQ4 interrupt input*
	AN12 input

Note: IRQ4 input when ITS4 = 0.

- P53/ADTRG/IRQ3

The pin function is switched as shown below according to the combination of bits TRGS1 and TRGS0 in the A/D control register (ADCR), bit ITS3 in ITSR, and bit P53DDR.

P53DDR	0	1
Pin function	P53 input	P53 output
	ADTRG input*[1]	
	IRQ3 interrupt input*[2]	

Notes: 1. ADTRG input when TRGS1 = TRGS0 = 0.
2. IRQ3 input when ITS3 = 0.

- P52/SCK2/IRQ2

The pin function is switched as shown below according to the combination of bit C/A in SMR of SCI_2, bits CKE0 and CKE1 in SCR, bit ITS2 in ITSR, and bit P52DDR.

CKE1	0				1
C/A	0			1	—
CKE0	0		1	—	—
P52DDR	0	1	—	—	—
Pin function	P52 input	P52 output	SCK2 output	SCK2 output	SCK2 input
	IRQ2 interrupt input*				

Note: IRQ2 input when ITS2 = 0.

- P51/RxD2/IRQ1

The pin function is switched as shown below according to the combination of bit RE in SCR of SCI_2, bit ITS1 in ITSR, and bit P51DDR.

RE	0		1
P51DDR	0	1	—
Pin function	P51 input	P51 output	RxD2 input
	IRQ1 interrupt input*		

Note: IRQ1 input when ITS1 = 0.

- P50/TxD2/IRQ0

The pin function is switched as shown below according to the combination of bit TE in SCR of SCI_2, bit ITS0 in ITSR, and bit P50DDR.

TE	0		1
P50DDR	0	1	—
Pin function	P50 input	P50 output	TxD2 input
	IRQ0 interrupt input*		

Note: IRQ0 input when ITS0 = 0.

10.6 Port 6

Port 6 is a 6-bit I/O port that also has other functions. The port 6 has the following registers. For details on the port function control register 2, refer to section 10.3.5, Port Function Control Register 2 (PFCR2).

- Port 6 data direction register (P6DDR)
- Port 6 data register (P6DR)
- Port 6 register (PORT6)
- Port function control register 2 (PFCR2)

10.6.1 Port 6 Data Direction Register (P6DDR)

The individual bits of P6DDR specify input or output for the pins of port 6.

P6DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P65DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
4	P64DDR	0	W	
3	P63DDR	0	W	
2	P62DDR	0	W	
1	P61DDR	0	W	
0	P60DDR	0	W	

10.6.2 Port 6 Data Register (P6DR)

P6DR stores output data for the port 6 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P65DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose I/O.
4	P64DR	0	R/W	
3	P63DR	0	R/W	
2	P62DR	0	R/W	
1	P61DR	0	R/W	
0	P60DR	0	R/W	

10.6.3 Port 6 Register (PORT6)

PORT6 shows the pin states.

PORT6 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved
6	—	Undefined	—	These bits are reserved, if read they will return an undefined value.
5	P65	Undefined*	R	If a port 6 read is performed while P6DDR bits are set to 1, the P6DR values are read. If a port 6 read is performed while P6DDR bits are cleared to 0, the pin states are read.
4	P64	Undefined*	R	
3	P63	Undefined*	R	
2	P62	Undefined*	R	
1	P61	Undefined*	R	
0	P60	Undefined*	R	

Note: Determined by the states of pins P65 to P60.

10.6.4 Pin Functions

Port 6 pins also function as 8-bit timer I/Os, interrupt inputs, and DMAC I/Os. The correspondence between the register specification and the pin functions is shown below.

- P65/TMO1/DACK1/IRQ13

The pin function is switched as shown below according to the combination of bit DMACS in PFCR2, bit SAE1 in DMABCRH, bits OS3 to OS0 in TCSR1 of the 8-bit timer, bit P65DDR, and bit ITS13 in ITSR.

SAE1	0			1			
DMACS	—			1			0
OS3 to OS0	All 0		Not all 0	All 0		Not all 0	—
P65DDR	0	1	—	0	1	—	—
Pin function	P65 input	P65 output	TMO1 output	P65 input	P65 output	TMO1 output	DACK1 output
	IRQ13 interrupt input*						

Note: IRQ13 interrupt input when ITS13 = 0.

- P64/TMO0/DACK0/IRQ12

The pin function is switched as shown below according to the combination of bit DMACS in PFCR2, bit SAE0 in DMABCRH, bits OS3 to OS0 in TCSR_0 of the 8-bit timer, bit P64DDR, and bit ITS12 in ITSR.

SAE0	0			1			
DMACS	—			1			0
OS3 to OS0	All 0		Not all 0	All 0		Not all 0	—
P64DDR	0	1	—	0	1	—	—
Pin function	P64 input	P64 output	TMO0 output	P64 input	P64 output	TMO0 output	DACK0 output
	IRQ12 interrupt input*						

Note: IRQ12 interrupt input when ITS12 = 0.

- P63/TMCI1/TEND1/IRQ11

The pin function is switched as shown below according to the combination of bit DMACS in PFCR2, bit TEE1 in DMATCR of the DMAC, bit P63DDR, and bit ITS11 in ITSR.

TEE1	0		1		
DMACS	—		1		0
P63DDR	0	1	0	1	—
Pin function	P63 input	P63 output	P63 input	P63 output	TEND1 output
	IRQ11 interrupt input*				
	TMCI1 input*[2]				

Notes: 1. IRQ11 interrupt input when ITS11 = 0.
2. When used as the external clock input pin for the TMR, its pin function should be specified to the external clock input by the CKS2 to CKS0 bits in TCR_1.

- P62/TMCI0/TEND0/IRQ10

The pin function is switched as shown below according to the combination of bit DMACS in PFCR2, bit TEE0 in DMATCR of the DMAC, bit P62DDR, and bit ITS10 in ITSR.

TEE0	0		1		
DMACS	—		1		0
P62DDR	0	1	0	1	—
Pin function	P62 input	P62 output	P62 input	P62 output	TEND0 output
	IRQ10 interrupt input*				
	TMCI0 input*[2]				

Note: 1. IRQ10 interrupt input when ITS10 = 0.
2. When used as the external clock input pin for the TMR, its pin function should be specified to the external clock input by the CKS2 to CKS0 bits in TCR_0.

- P61/TMRI1/DREQ1/IRQ9

The pin function is switched as shown below according to the combination of bit P61DDR and bit ITS9 in ITSR.

P61DDR	0	1
Pin function	P61 input	P61 output
	TMRI1 input*[1]	
	DREQ1 input*[2]	
	IRQ9 interrupt input*[2]	

Notes: 1. When used as the counter reset input pin for the TMR, both the CCLR1 and CCLR0 bits in TCR_1 should be set to 1.
2. DREQ1 input when DMAKS = 0.
3. IRQ9 interrupt input when ITS9 = 0.

- P60/TMRI0/DREQ0/IRQ8

The pin function is switched as shown below according to the combination of bit P60DDR and bit ITS8 in ITSR.

P60DDR	0	1
Pin function	P60 input	P60 output
	TMRI0 input*[1]	
	DREQ0 input*[2]	
	IRQ8 interrupt input*[3]	

Notes: 1. When used as the counter reset input pin for the TMR, both the CCLR1 and CCLR0 bits in TCR_0 should be set to 1.
2. DREQ0 input when DMAKS = 0.
3. IRQ8 interrupt input when ITS8 = 0.

10.7 Port 7

Port 7 is a 6-bit I/O port that also has other functions. The port 7 has the following registers. For details on the port function control register 2, refer to section 10.3.5, Port Function Control Register 2 (PFCR2).

- Port 7 data direction register (P7DDR)
- Port 7 data register (P7DR)
- Port 7 register (PORT7)
- Port function control register 2 (PFCR2)

10.7.1 Port 7 Data Direction Register (P7DDR)

The individual bits of P7DDR specify input or output for the pins of port 7.

P7DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P75DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
4	P74DDR	0	W	
3	P73DDR	0	W	
2	P72DDR	0	W	
1	P71DDR	0	W	
0	P70DDR	0	W	

10.7.2 Port 7 Data Register (P7DR)

P7DR stores output data for the port 7 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P75DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
4	P74DR	0	R/W	
3	P73DR	0	R/W	
2	P72DR	0	R/W	
1	P71DR	0	R/W	
0	P70DR	0	R/W	

10.7.3 Port 7 Register (PORT7)

PORT7 shows the pin states.

PORT7 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved
6	—	Undefined	—	These bits are reserved, if read they will return an undefined value.
5	P75	Undefined*	R	If a port 7 read is performed while P7DDR bits are set to 1, the P7DR values are read. If a port 7 read is performed while P7DDR bits are cleared to 0, the pin states are read.
4	P74	Undefined*	R	
3	P73	Undefined*	R	
2	P72	Undefined*	R	
1	P71	Undefined*	R	
0	P70	Undefined*	R	

Note: Determined by the states of pins P75 to P70.

10.7.4 Pin Functions

Port 7 pins also function as DMAC I/Os and EXDMAC I/Os. The correspondence between the register specification and the pin functions is shown below.

- P75/DACK1/EDACK1

The pin function is switched as shown below according to the combination of bit DMACS in PFCR2, bit SAE1 in DMABCRH, bit AMS in EDMDR1, and bit P75DDR.

Modes 1, 2, 3* (EXPE = 1), 4, 5, 6, 7 (EXPE = 1)

AMS	0					1
SAE1	0		1			—
DMACS	—		0		1	—
P75DDR	0	1	0	1	—	—
Pin function	P75 input	P75 output	P75 input	P75 output	DACK1 output	EDACK1 output

Modes 3* (EXPE = 0), 7 (EXPE = 0)

AMS	—				
SAE1	0		1		
DMACS	—		0		1
P75DDR	0	1	0	1	—
Pin function	P75 input	P75 output	P75 input	P75 output	DACK1 output

Note: Only in H8S/2678R Series.

- P74/DACK0/EDACK0

The pin function is switched as shown below according to the combination of bit DMACS in PFCR2, bit SAE0 in DMABCRH, bit AMS in EDMDR0, and bit P74DDR.

Modes 1, 2, 3* (EXPE = 1), 4, 5, 6, 7 (EXPE = 1)

AMS	0					1
SAE0	0		1			—
DMACS	—		0		1	—
P74DDR	0	1	0	1	—	—
Pin function	P74 input	P74 output	P74 input	P74 output	DACK0 output	EDACK0 output

Modes 3* (EXPE = 0), 7 (EXPE = 0)

AMS	—				
SAE0	0		1		
DMACS	—		0		1
P74DDR	0	1	0	1	—
Pin function	P74 input	P74 output	P74 input	P74 output	DACK0 output

Note: Only in H8S/2678R Series.

- P73/TEND1/ETEND1

The pin function is switched as shown below according to the combination of bit DMACS in PFCR2, bit TEE1 in DMATCR of the DMAC, bit ETENDE in EDMDR1 of the EXDMAC, and bit P73DDR.

Modes 1, 2, 3* (EXPE = 1), 4, 5, 6, 7 (EXPE = 1)

ETENDE	0					1
TEE1	0		1			—
DMACS	—		0		1	—
P73DDR	0	1	0	1	—	—
Pin function	P73 input	P73 output	P73 input	P73 output	TEND1 output	ETEND1 output

Modes 3* (EXPE = 0), 7 (EXPE = 0)

ETENDE	—				
TEE1	0		1		
DMACS	—		0		1
P73DDR	0	1	0	1	—
Pin function	P73 input	P73 output	P73 input	P73 output	TEND1 output

Note: Only in H8S/2678R Series.

- P72/TEND0/ETEND0

The pin function is switched as shown below according to the combination of bit DMACS in PFCR2, bit TEE0 in DMATCR of the DMAC, bit ETENDE in EDMDR0 of the EXDMAC, and bit P72DDR.

Modes 1, 2, 3* (EXPE = 1), 4, 5, 6, 7 (EXPE = 1)

ETENDE	0					1
TEE0	0		1			—
DMACS	—		0		1	—
P72DDR	0	1	0	1	—	—
Pin function	P72 input	P72 output	P72 input	P72 output	TEND0 output	ETEND0 output

Modes 3* (EXPE = 0), 7 (EXPE = 0)

ETENDE	—				
TEE0	0		1		
DMACS	—		0		1
P72DDR	0	1	0	1	—
Pin function	P72 input	P72 output	P72 input	P72 output	TEND0 output

Note: Only in H8S/2678R Series.

- P71/DREQ1//

P8DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P85DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
4	P84DDR	0	W	
3	P83DDR	0	W	
2	P82DDR	0	W	
1	P81DDR	0	W	
0	P80DDR	0	W	

10.8.2 Port 8 Data Register (P8DR)

P8DR stores output data for the port 8 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P85DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
4	P84DR	0	R/W	
3	P83DR	0	R/W	
2	P82DR	0	R/W	
1	P81DR	0	R/W	
0	P80DR	0	R/W	

10.8.3 Port 8 Register (PORT8)

PORT8 shows the pin states.

PORT8 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved
6	—	Undefined	—	These bits are reserved, if read they will return an undefined value.
5	P85	Undefined*	R	If a port 8 read is performed while P8DDR bits are set to 1, the P8DR values are read. If a port 8 read is performed while P8DDR bits are cleared to 0, the pin states are read.
4	P84	Undefined*	R	
3	P83	Undefined*	R	
2	P82	Undefined*	R	
1	P81	Undefined*	R	
0	P80	Undefined*	R	

Note: Determined by the states of pins P85 to P80.

10.8.4 Pin Functions

Port 8 pins also function as interrupt inputs and EXDMAC I/Os. The correspondence between the register specification and the pin functions is shown below.

- P85/IRQ5/EDACK3

The pin function is switched as shown below according to the combination of bit AMS in EDMDR3 of the EXDMAC, bit P85DDR, and bit ITS5 in ITSR.

Modes 1, 2, 3 (EXPE = 1), 4, 5, 6, 7 (EXPE = 1)

AMS	0		1
P85DDR	0	1	—
Pin function	P85 input	P85 output	EDACK3 output
	IRQ5 interrupt input*		

Modes 3, 7 (EXPE = 0)

AMS	—	
P85DDR	0	1
Pin function	P85 input	P85 output
	IRQ5 interrupt input*	

Note: IRQ5 input when ITS5 = 1.

- P84/IRQ4/EDACK2

The pin function is switched as shown below according to the combination of bit AMS in EDMDR2 of the EXDMAC, bit P84DDR, and bit ITS4 in ITSR.

Modes 1, 2, 3*[2] (EXPE = 1), 4, 5, 6, 7 (EXPE = 1)

AMS	0		1
P84DDR	0	1	—
Pin function	P84 input	P84 input/output	EDACK2 output
	IRQ4 interrupt input*		

Modes 3*[2] (EXPE = 0), 7 (EXPE = 0)

AMS	—	
P84DDR	0	1
Pin function	P84 input	P84 output
	IRQ4 interrupt input*	

Notes: 1. IRQ4 input when ITS4 = 1.
2. Only in H8S/2678R Series.

- P83/IRQ3/ETEND3

The pin function is switched as shown below according to the combination of bit ETENDE in EDMDR3 of the EXDMAC, bit P83DDR, and bit ITS3 in ITSR.

Modes 1, 2, 3*[2] (EXPE = 1), 4, 5, 6, 7 (EXPE = 1)

ETENDE	0		1
P83DDR	0	1	—
Pin function	P83 input	P83 output	ETEND3 output
	IRQ3 interrupt input*		

Modes 3*[2] (EXPE = 0), 7 (EXPE = 0)

ETENDE	—	
P83DDR	0	1
Pin function	P83 input	P83 output
	IRQ3 interrupt input*	

Notes: 1. IRQ3 input when ITS3 = 1.
2. Only in H8S/2678R Series.

- P82/IRQ2/ETEND2

The pin function is switched as shown below according to the combination of bit ETENDE in EDMDR2 of the EXDMAC, bit P82DDR, and bit ITS2 in ITSR.

Modes 1, 2, 3*[2] (EXPE = 1), 4, 5, 6, 7 (EXPE = 1)

ETENDE	0		1
P82DDR	0	1	—
Pin function	P82 input	P82 output	ETEND2 output
	IRQ2 interrupt input*		

Modes 3*[2] (EXPE = 0), 7 (EXPE = 0)

ETENDE	—	
P82DDR	0	1
Pin function	P82 input	P82 output
	IRQ2 interrupt input*	

Notes: 1. IRQ2 input when ITS2 = 1.
2. Only in H8S/2678R Series.

- P81/IRQ1/EDREQ3

The pin function is switched as shown below according to the combination of bit P81DDR and bit ITS1 in ITSR.

P81DDR	0	1
Pin function	P81 input	P81 output
	EDREQ3 input	
	IRQ1 interrupt input*	

Note: IRQ1 input when ITS1 = 1.

- P80/IRQ0/EDREQ2

The pin function is switched as shown below according to the combination of bit P80DDR and bit ITS0 in ITSR.

P80DDR	0	1
Pin function	P80 input	P80 output
	EDREQ2 input	
	IRQ0 interrupt input*	

Note: IRQ0 input when ITS0 = 1.

10.9 Port A

Port A is an 8-bit I/O port that also has other functions. The port A has the following registers.

- Port A data direction register (PADDR)
- Port A data register (PADR)
- Port A register (PORTA)
- Port A pull-up MOS control register (PAPCR)
- Port A open-drain control register (PAODR)
- Port function control register 1 (PFCR1)

10.9.1 Port A Data Direction Register (PADDR)

The individual bits of PADDR specify input or output for the pins of port A. PADDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DDR	0	W	• Modes 1, 2, 5, and 6 Pins PA4 to PA0 are address outputs regardless of the PADDR settings. For pins PA7 to PA5, when the corresponding bit of A23E to A21E is set to 1, setting a PADDR bit to 1 makes the corresponding port A pin an address output, while clearing the bit to 0 makes the pin an input port. Clearing one of bits A23E to A21E to 0 makes the corresponding port A pin an I/O port, and its function can be switched with PADDR. • Mode 4 When the corresponding bit of A23E to A16E is set to 1, setting a PADDR bit to 1 makes the corresponding port A pin an address output, while clearing the bit to 0 makes the pin an input port. Clearing one of bits A23E to A16E to 0 makes the corresponding port A pin an I/O port, and its function can be switched with PADDR. • Modes 3* and 7 (when EXPE = 1) When the corresponding bit of A23E to A16E is set to 1, setting a PADDR bit to 1 makes the corresponding port A pin an address output, while clearing the bit to 0 makes the pin an input port. Clearing one of bits A23E to A16E to 0 makes the corresponding port A pin an I/O port; setting the corresponding PADDR bit to 1 makes the pin an output port, while clearing the bit to 0 makes the pin an input port. • Modes3* and 7 (when EXPE = 0) Port A is an I/O port, and its pin functions can be switched with PADDR.
6	PA6DDR	0	W	
5	PA5DDR	0	W	
4	PA4DDR	0	W	
3	PA3DDR	0	W	
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR	0	W	

Notes: Only in H8S/2678R Series.

10.9.2 Port A Data Register (PADR)

PADR stores output data for the port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	PA6DR	0	R/W	
5	PA5DR	0	R/W	
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

10.9.3 Port A Register (PORTA)

PORTA shows port A pin states.

PORTA cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7	Undefined*	R	If a port A read is performed while PADDR bits are set to 1, the PADR values are read. If a port A read is performed while PADDR bits are cleared to 0, the pin states are read.
6	PA6	Undefined*	R	
5	PA5	Undefined*	R	
4	PA4	Undefined*	R	
3	PA3	Undefined*	R	
2	PA2	Undefined*	R	
1	PA1	Undefined*	R	
0	PA0	Undefined*	R	

Note: Determined by the states of pins PA7 to PA0.

10.9.4 Port A Pull-Up MOS Control Register (PAPCR)

PAPCR controls the input pull-up MOS function. Bits 7 to 5 are valid in modes 1, 2, 5, and 6, and all the bits are valid in modes 3*, 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7PCR	0	R/W	When a pin function is specified to an input port, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PA6PCR	0	R/W	
5	PA5PCR	0	R/W	
4	PA4PCR	0	R/W	
3	PA3PCR	0	R/W	
2	PA2PCR	0	R/W	
1	PA1PCR	0	R/W	
0	PA0PCR	0	R/W	

Note: Only in H8S/2678R Series.

10.9.5 Port A Open Drain Control Register (PAODR)

PAODR specifies an output type of port A.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7ODR	0	R/W	Setting the corresponding bit to 1 specifies a pin output type to NMOS open-drain output, while clearing this bit to 0 specifies that to CMOS output.
6	PA6ODR	0	R/W	
5	PA5ODR	0	R/W	
4	PA4ODR	0	R/W	
3	PA3ODR	0	R/W	
2	PA2ODR	0	R/W	
1	PA1ODR	0	R/W	
0	PA0ODR	0	R/W	

Note: Only in H8S/2678R Series.

10.9.6 Port Function Control Register 1 (PFCR1)

PFCR1 performs I/O port control. Bits 7 to 5 are valid in modes 1, 2, 5, and 6, and all the bits are valid in modes 3*, 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	A23E	1	R/W	Address 23 Enable Enables or disables output for address output 23 (A23). 0: DR output when PA7DDR = 1 1: A23 output when PA7DDR = 1
6	A22E	1	R/W	Address 22 Enable Enables or disables output for address output 22 (A22). 0: DR output when PA6DDR = 1 1: A22 output when PA6DDR = 1
5	A21E	1	R/W	Address 21 Enable Enables or disables output for address output 21 (A21). 0: DR output when PA5DDR = 1 1: A21 output when PA5DDR = 1
4	A20E	1	R/W	Address 20 Enable Enables or disables output for address output 20 (A20). 0: DR output when PA4DDR = 1 1: A20 output when PA4DDR = 1
3	A19E	1	R/W	Address 19 Enable Enables or disables output for address output 19 (A19). 0: DR output when PA3DDR = 1 1: A19 output when PA3DDR = 1
2	A18E	1	R/W	Address 18 Enable Enables or disables output for address output 18 (A18). 0: DR output when PA2DDR = 1 1: A18 output when PA2DDR = 1
1	A17E	1	R/W	Address 17 Enable Enables or disables output for address output 17 (A17). 0: DR output when PA1DDR = 1 1: A17 output when PA1DDR = 1
0	A16E	1	R/W	Address 16 Enable Enables or disables output for address output 16 (A16). 0: DR output when PA0DDR = 1 1: A16 output when PA0DDR = 1

10.9.7 Pin Functions

Port A pins also function as address outputs. The correspondence between the register specification and the pin functions is shown below.

- PA7/A23, PA6/A22, PA5/A21

The pin function is switched as shown below according to the operating mode, bit EXPE, bits A23E to A21E, and bit PADDR.

Operating mode	1, 2, 4, 5, 6				3*, 7					
EXPE	—				0		1			
AxxE	0		1		—		0		1	
PADDR	0	1	0	1	0	1	0	1	0	1
Pin function	PA input	PA output	PA input	Address output	PA input	PA output	PA input	PA output	PA input	Address output

Note: Only in H8S/2678R Series.

- PA4/A20, PA3/A19, PA2/A18, PA1/A17, PA20/A16

The pin function is switched as shown below according to the operating mode, bit EXPE, bits A23E to A21E, and bit PADDR.

Operating mode	1, 2, 5, 6	4				3*, 7					
EXPE	—	—				0		1			
AxxE	—	0		1		—		0		1	
PADDR	—	0	1	0	1	0	1	0	1	0	1
Pin function	Address output	PA input	PA output	PA input	Address output	PA input	PA output	PA input	PA output	PA input	Address output

Note: Only in H8S/2678R Series.

10.9.8 Port A Input Pull-Up MOS States

Port A has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used by pins PA7 to PA5 in modes 1, 2, 5, and 6, and by all pins in modes 3*, 4, and 7. Input pull-up MOS can be specified as on or off on a bit-by-bit basis.

Table 10.2 summarizes the input pull-up MOS states.

Table 10.2 Input Pull-Up MOS States (Port A)

Mode		Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
3*, 4, 7	PA7 to PA0	Off	Off	On/Off	On/Off
1, 2, 5, 6	PA7 to PA5			On/Off	On/Off
	PA4 to PA0			Off	Off

Legend:
Off: Input pull-up MOS is always off.
On/Off: On when PADDR = 0 and PAPCR = 1; otherwise off.
Note: Only in H8S/2678R Series.

10.10 Port B

Port B is an 8-bit I/O port that also has other functions. The port B has the following registers.

- Port B data direction register (PBDDR)
- Port B data register (PBDR)
- Port B register (PORTB)
- Port B pull-up MOS control register (PBPCR)

10.10.1 Port B Data Direction Register (PBDDR)

The individual bits of PBDDR specify input or output for the pins of port B.

PBDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DDR	0	W	• Modes 1, 2, 5, and 6 Port B pins are address outputs regardless of the PBDDR settings. • Modes 3* (EXPE = 1), 4, and 7 (when EXPE = 1) Setting a PBDDR bit to 1 makes the corresponding port B pin an address output, while clearing the bit to 0 makes the pin an input port. • Modes 3* (EXPE = 1) and 7 (when EXPE = 0) Port B is an I/O port, and its pin functions can be switched with PBDDR.
6	PB6DDR	0	W	
5	PB5DDR	0	W	
4	PB4DDR	0	W	
3	PB3DDR	0	W	
2	PB2DDR	0	W	
1	PB1DDR	0	W	
0	PB0DDR	0	W	

Note: Only in H8S/2678R Series.

10.10.2 Port B Data Register (PBDR)

PBDR is stores output data for the port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

10.10.3 Port B Register (PORTB)

PORTB shows port B pin states.

PORTB cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7	Undefined*	R	If a port B read is performed while PBDDR bits are set to 1, the PBDR values are read. If a port B read is performed while PBDDR bits are cleared to 0, the pin states are read.
6	PB6	Undefined*	R	
5	PB5	Undefined*	R	
4	PB4	Undefined*	R	
3	PB3	Undefined*	R	
2	PB2	Undefined*	R	
1	PB1	Undefined*	R	
0	PB0	Undefined*	R	

Note: Determined by the states of pins PB7 to PB0.

Table 10.3 summarizes the input pull-up MOS states.

Table 10.3 Input Pull-Up MOS States (Port B)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1, 2, 5, 6	Off	Off	Off	Off
3*, 4, 7			On/Off	On/Off

Legend:
Off: Input pull-up MOS is always off.
On/Off: On when PBDDR = 0 and PBPCR = 1; otherwise off.
Note: Only in H8S/2678R Series.

10.11 Port C

Port C is an 8-bit I/O port that also has other functions. The port C has the following registers.

- Port C data direction register (PCDDR)
- Port C data register (PCDR)
- Port C register (PORTC)
- Port C pull-up MOS control register (PCPCR)

10.11.1 Port C Data Direction Register (PCDDR)

The individual bits of PCDDR specify input or output for the pins of port C.

PCDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	• Modes 1, 2, 5, and 6 Port C pins are address outputs regardless of the PCDDR settings. • Modes 3* (EXPE = 1), 4, and 7 (when EXPE = 1) Setting a PCDDR bit to 1 makes the corresponding port C pin an address output, while clearing the bit to 0 makes the pin an input port. • Modes 3* (EXPE = 1) and 7 (when EXPE = 0) Port C is an I/O port, and its pin functions can be switched with PCDDR.
6	PC6DDR	0	W	
5	PC5DDR	0	W	
4	PC4DDR	0	W	
3	PC3DDR	0	W	
2	PC2DDR	0	W	
1	PC1DDR	0	W	
0	PC0DDR	0	W	

Note: Only in H8S/2678R Series.

10.11.2 Port C Data Register (PCDR)

PCDR stores output data for the port C pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

10.11.3 Port C Register (PORTC)

PORTC is shows port C pin states.

PORTC cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7	Undefined*	R	If a port C read is performed while PCDDR bits are set to 1, the PCDR values are read. If a port C read is performed while PCDDR bits are cleared to 0, the pin states are read.
6	PC6	Undefined*	R	
5	PC5	Undefined*	R	
4	PC4	Undefined*	R	
3	PC3	Undefined*	R	
2	PC2	Undefined*	R	
1	PC1	Undefined*	R	
0	PC0	Undefined*	R	

Note: Determined by the states of pins PC7 to PC0.

10.11.4 Port C Pull-Up MOS Control Register (PCPCR)

PCPCR controls the on/off state of input pull-up MOS of port C. PCPCR is valid in modes 3*, 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PCR	0	R/W	When a pin function is specified to an input port, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PC6PCR	0	R/W	
5	PC5PCR	0	R/W	
4	PC4PCR	0	R/W	
3	PC3PCR	0	R/W	
2	PC2PCR	0	R/W	
1	PC1PCR	0	R/W	
0	PC0PCR	0	R/W	

Note: Only in H8S/2678R Series.

10.11.5 Pin Functions

Port C pins also function as address outputs. The correspondence between the register specification and the pin functions is shown below.

- PC7/A7, PC6/A6, PC5/A5, PC4/A4, PC3/A3, PC2/A2, PC1/A1, PC0/A0

The pin function is switched as shown below according to the operating mode, bit EXPE, and bit PCDDR.

Operating mode	1, 2, 5, 6	4		3*, 7			
EXPE	—	—		0		1	
PCDDR	—	0	1	0	1	0	1
Pin function	Address output	PC input	Address output	PC input	PC output	PC input	Address output

Note: Only in H8S/2678R Series.

10.11.6 Port C Input Pull-Up MOS States

Port C has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used in modes 3*, 4, and 7. Input pull-up MOS can be specified as on or off on a bit-by-bit basis.

In modes 3*, 4, and 7, when a PCDDR bit is cleared to 0, setting the corresponding PCPCR bit to 1 turns on the input pull-up MOS for that pin.

Table 10.4 summarizes the input pull-up MOS states.

Table 10.4 Input Pull-Up MOS States (Port C)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1, 2, 5, 6	Off	Off	Off	Off
3*, 4, 7			On/Off	On/Off

Legend:
Off: Input pull-up MOS is always off.
On/Off: On when PCDDR = 0 and PCPCR = 1; otherwise off.
Note: Only in H8S/2678R Series.

10.12 Port D

Port D is an 8-bit I/O port that also has other functions. The port D has the following registers.

- Port D data direction register (PDDDR)
- Port D data register (PDDR)
- Port D register (PORTD)
- Port D pull-up MOS control register (PDPCR)

10.12.1 Port D Data Direction Register (PDDDR)

The individual bits of PDDDR specify input or output for the pins of port D.

PDDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DDR	0	W	• Modes 1, 2, 3* (EXPE = 1), 4, 5, 6, and 7 (when EXPE = 1) Port D is automatically designated for data input/output. • Modes 3* (EXPE = 1) and 7 (when EXPE = 0) Port D is an I/O port, and its pin functions can be switched with PDDDR.
6	PD6DDR	0	W	
5	PD5DDR	0	W	
4	PD4DDR	0	W	
3	PD3DDR	0	W	
2	PD2DDR	0	W	
1	PD1DDR	0	W	
0	PD0DDR	0	W	

Note: Only in H8S/2678R Series.

10.12.2 Port D Data Register (PDDR)

PDDR stores output data for the port D pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

10.12.3 Port D Register (PORTD)

PORTD shows port D pin states.

PORTD cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7	Undefined*	R	If a port D read is performed while PDDDR bits are set to 1, the PDDR values are read. If a port D read is performed while PDDDR bits are cleared to 0, the pin states are read.
6	PD6	Undefined*	R	
5	PD5	Undefined*	R	
4	PD4	Undefined*	R	
3	PD3	Undefined*	R	
2	PD2	Undefined*	R	
1	PD1	Undefined*	R	
0	PD0	Undefined*	R	

Note: Determined by the states of pins PD7 to PD0.

10.12.4 Port D Pull-up Control Register (PDPCR)

PDPCR controls on/off states of the input pull-up MOS of port D. PDPCR is valid in modes3* and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PCR	0	R/W	When the pin is in its input state, the input pull-up MOS of the input pin is on when the corresponding bit is set to 1.
6	PD6PCR	0	R/W	
5	PD5PCR	0	R/W	
4	PD4PCR	0	R/W	
3	PD3PCR	0	R/W	
2	PD2PCR	0	R/W	
1	PD1PCR	0	R/W	
0	PD0PCR	0	R/W	

Note: Only in H8S/2678R Series.

10.12.5 Pin Functions

Port D pins also function as data I/Os. The correspondence between the register specification and the pin functions is shown below.

- PD7/D15, PD6/D14, PD5/D13, PD4/D12, PD3/D11, PD2/D10, PD1/D9, PD0/D8

The pin function is switched as shown below according to the operating mode, bit EXPE, and bit PDDDR.

Operating mode	1, 2, 4, 5, 6	3*, 7		
EXPE	—	0		1
PDDDR	—	0	1	—
Pin function	Data I/O	PD input	PD output	Data I/O

Note: Only in H8S/2678R Series.

10.12.6 Port D Input Pull-Up MOS States

Port D has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used in modes 3*and 7. Input pull-up MOS can be specified as on or off on a bit-by-bit basis.

In modes 3* and 7, when a PDDDR bit is cleared to 0, setting the corresponding PDPCR bit to 1 turns on the input pull-up MOS for that pin.

Table 10.5 summarizes the input pull-up MOS states.

Table 10.5 Input Pull-Up MOS States (Port D)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1, 2, 4, 5, 6	Off	Off	Off	Off
3*, 7			On/Off	On/Off

Legend:
OFF: Input pull-up MOS is always off.
On/Off: On when PDDDR = 0 and PDPCR = 1; otherwise off.
Note: Only in H8S/2678R Series.

10.13 Port E

Port E is an 8-bit I/O port that also has other functions. The port E has the following registers.

- Port E data direction register (PEDDR)
- Port E data register (PEDR)
- Port E register (PORTE)
- Port E pull-up MOS control register (PEPCR)

10.13.1 Port E Data Direction Register (PEDDR)

The individual bits of PEDDR specify input or output for the pins of port E.

PEDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	• Modes 1, 2, 4, 5, and 6 When 8-bit bus mode is selected, port E functions as an I/O port. The pin states can be changed with PEDDR. When 16-bit bus mode is selected, port E is designated for data input/output. For details on 8-bit and 16-bit bus modes, see section 6, Bus Controller. • Modes 3* and 7 (when EXPE = 1) When 8-bit bus mode is selected, port E functions as an I/O port. Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port. When 16-bit bus mode is selected, port E is designated for data input/output. • Modes 3* and 7 (when EXPE = 0) Port E is an I/O port, and its pin functions can be switched with PEDDR.
6	PE6DDR	0	W	
5	PE5DDR	0	W	
4	PE4DDR	0	W	
3	PE3DDR	0	W	
2	PE2DDR	0	W	
1	PE1DDR	0	W	
0	PE0DDR	0	W	

Note: Only in H8S/2678R Series.

10.13.2 Port E Data Register (PEDR)

PEDR stores output data for the port E pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	PE6DR	0	R/W	
5	PE5DR	0	R/W	
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

10.13.3 Port E Register (PORTE)

PORTE shows port E pin states.

PORTE cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7	Undefined*	R	If a port E read is performed while PEDDR bits are set to 1, the PEDR values are read. If a port E read is performed while PEDDR bits are cleared to 0, the pin states are read.
6	PE6	Undefined*	R	
5	PE5	Undefined*	R	
4	PE4	Undefined*	R	
3	PE3	Undefined*	R	
2	PE2	Undefined*	R	
1	PE1	Undefined*	R	
0	PE0	Undefined*	R	

Note: Determined by the states of pins PE7 to PE0.

10.13.4 Port E Pull-up Control Register (PEPCR)

PEPCR controls on/off states of the input pull-up MOS of port E. PEPCR is valid in 8-bit bus mode.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7PCR	0	R/W	When the pin is in its input state, the input pull-up MOS of the input pin is on when the corresponding bit is set to 1.
6	PE6PCR	0	R/W	
5	PE5PCR	0	R/W	
4	PE4PCR	0	R/W	
3	PE3PCR	0	R/W	
2	PE2PCR	0	R/W	
1	PE1PCR	0	R/W	
0	PE0PCR	0	R/W	

10.13.5 Pin Functions

Port E pins also function as data I/Os. The correspondence between the register specification and the pin functions is shown below.

- PE7/D7, PE6/D6, PE5/D5, PE4/D4, PE3/D3, PE2/D2, PE1/D1, PE0/D0

The pin function is switched as shown below according to the operating mode, bus mode, bit EXPE, and bit PEDDR.

Operating mode	1, 2, 4, 5, 6			3*, 7				
Bus mode	All areas 8-bit space		At least one area 16-bit space	—		All areas 8-bit space		At least one area 16-bit space
EXPE	—		—	0		1		1
PEDDR	0	1	—	0	1	0	1	—
Pin function	PE input	PE output	Data I/O	PE input	PE output	PE input	PE output	Data I/O

Note: Only in H8S/2678R Series.

10.13.6 Port E Input Pull-Up MOS States

Port E has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used in 8-bit bus mode. Input pull-up MOS can be specified as on or off on a bit-by-bit basis. In 8-bit bus mode, when a PEDDR bit is cleared to 0, setting the corresponding PEPCR bit to 1 turns on the input pull-up MOS for that pin.

Table 10.6 summarizes the input pull-up MOS states.

Table 10.6 Input Pull-Up MOS States (Port E)

Mode*		**Reset**	**Hardware Standby Mode**	**Software Standby Mode**	**In Other Operations**
1 to 7	8-bit bus	Off	Off	On/Off	On/Off
	16-bit bus			Off	Off

Legend:
Off: Input pull-up MOS is always off.
On/Off: On when PEDDR = 0 and PEPCR = 1; otherwise off.
Note: Mode 3 is available only in H8S/2678R Series.

10.14 Port F

Port F is an 8-bit I/O port that also has other functions. The port F has the following registers. For details on the port function control register 2, refer to section 10.3.5, Port Function Control Register 2 (PFCR2).

- Port F data direction register (PFDDR)

- Port F data register (PFDR)
- Port F register (PORTF)
- Port Function Control Register 2 (PFCR2)

10.14.1 Port F Data Direction Register (PFDDR)

The individual bits of PFDDR specify input or output for the pins of port F.
PFDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DDR	1/0*[1]	W	• Modes 1, 2, 4, 5, and 6 Pin PF7 functions as the ø output pin when the corresponding PFDDR bit is set to 1, and as an input port when the bit is cleared to 0. Pin PF6 functions as the AS output pin when ASOE is set to 1. When ASOE is cleared to 0, pin PF6 is an I/O port and its function can be switched with PF6DDR. Pins PF5 and PF4 are automatically designated as bus control outputs (RD and HWR). Pin PF3 functions as the LWR output pin when LWROE is set to 1. When LWROE is cleared to 0, pin PF3 is an I/O port and its function can be switched with PF3DDR. Pins PF2 and PF1 are designated as I/O ports and their function can be switched with PFDDR. Pins PF0 functions as bus control input/output pin (LCAS, UCAS, and WAIT) when the appropriate bus controller settings are made. Otherwise, these pins are output ports when the corresponding PFDDR bit is set to 1, and input ports when the bit is cleared to 0. • Modes 3* and 7 (when EXPE = 1) Pin PF7 to PF3 function in the same way as in modes 1, 2, 4, 5, and 6.
6	PF6DDR	0	W	
5	PF5DDR	0	W	
4	PF4DDR	0	W	
3	PF3DDR	0	W	
2	PF2DDR	0	W	
1	PF1DDR	0	W	
0	PF0DDR	0	W	

Bit	Bit Name	Initial Value	R/W	Description
				Pins PF2 to PF0 function as bus control input/output pins (LCAS, UCAS, and WAIT) when the appropriate PFCR2 settings are made. Otherwise, these pins are I/O ports, and their functions can be switched with PFDDR. • Modes 3* and 7 (when EXPE = 0) Pin PF7 functions as the ø output pin when the corresponding PFDDR bit is set to 1, and as an input port when the bit is cleared to 0. Pins PF6 to PF0 are I/O ports, and their functions can be switched with PFDDR.

Notes: 1. PF7DDR is initialized to 1 in modes 1, 2, 4, 5, and 6, and to 0 in mode 7.
2. Only in H8S/2678R Series.

10.14.2 Port F Data Register (PFDR)

PFDR stores output data for the port F pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	PF6DR	0	R/W	
5	PF5DR	0	R/W	
4	PF4DR	0	R/W	
3	PF3DR	0	R/W	
2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

10.14.3 Port F Register (PORTF)

PORTF shows port F pin states.

PORTF cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7	Undefined*	R	If a port F read is performed while PFDDR bits are set to 1, the PFDR values are read. If a port F read is performed while PFDDR bits are cleared to 0, the pin states are read.
6	PF6	Undefined*	R	
5	PF5	Undefined*	R	
4	PF4	Undefined*	R	
3	PF3	Undefined*	R	
2	PF2	Undefined*	R	
1	PF1	Undefined*	R	
0	PF0	Undefined*	R	

Note: Determined by the states of pins PF7 to PF0.

10.14.4 Pin Functions

Port F pins also function as external interrupt inputs, bus control signal I/Os, and system clock outputs (ø). The correspondence between the register specification and the pin functions is shown below.

- PF7/ø

The pin function is switched as shown below according to bit PF7DDR.

Operating mode	1 to 7	
PFDDR	0	1
Pin function	PF7 input	ø output

- PF6/AS

The pin function is switched as shown below according to the operating mode, bit EXPE, bit PF6DDR, and bit ASOE.

Operating mode	1, 2, 4, 5, 6			3*, 7				
EXPE	—			0		1		
ASOE	1	0		—		1	0	
PF6DDR	—	0	1	0	1	—	0	1
Pin function	AS output	PF6 input	PF6 output	PF6 input	PF6 output	AS output	PF6 input	PF6 output

Note: Only in H8S/2678R Series.

- PF5/RD

The pin function is switched as shown below according to the operating mode, bit EXPE, and bit PF5DDR.

Operating mode	1, 2, 4, 5, 6	3*, 7		
EXPE	—	0		1
PF5DDR	—	0	1	—
Pin function	RD output	PF5 input	PF5 output	RD output

Note: Only in H8S/2678R Series.

- PF4/HWR

The pin function is switched as shown below according to the operating mode, bit EXPE, and bit PF4DDR.

Operating mode	1, 2, 4, 5, 6	3*, 7		
EXPE	—	0		1
PF4DDR	—	0	1	—
Pin function	HWR output	PF4 input	PF4 output	HWR output

Note: Only in H8S/2678R Series.

- PF3/LWR

The pin function is switched as shown below according to the operating mode, bit EXPE, bit PF3DDR, and bit LWROE.

Operating mode	1, 2, 4, 5, 6			3*, 7				
EXPE	—			0		1		
LWROD	1	0		—		1	0	
PF3DDR	—	0	1	0	1	—	0	1
Pin function	LWR output	PF3 input	PF3 output	PF3 input	PF3 output	LWR output	PF3 input	PF3 output

Note: Only in H8S/2678R Series.

- PF2/LCAS/DQML*[2]/IRQ15

The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits RMTS2 to RMTS0 in DRAMCR, bits ABW5 to ABW2 in ABWCR, and bit PF2DDR.

Operating mode	1, 2, 4, 5, 6			3*[2], 7				
EXPE	—			0		1		
Areas 2 to 5	Any DRAM space area is 16-bit bus space	All DRAM space areas are 8-bit bus space, or areas 2 to 5 are all normal space		—		Any DRAM space area is 16-bit bus space	All DRAM space areas are 8-bit bus space, or areas 2 to 5 are all normal space	
PF2DDR	—	0	1	0	1	—	0	1
Pin function	LCAS output	PF2 input	PF2 output	PF2 input	PF2 output	LCAS output	PF2 input	PF2 output
	IRQ15 interrupt input*							

Notes: 1. IRQ15 interrupt input when bit ITS15 is cleared to 0 in ITSR.
2. Only in H8S/2678R Series.

- PF1/UCAS/DQMU*[2]/IRQ14

The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits RMTS2 to RMTS0 in DRAMCR, and bit PF1DDR.

Operating mode	1, 2, 4, 5,			3*[2], 7				
EXPE	—			0		1		
Areas 2 to 5	Any of areas 2 to 5 is DRAM space	Areas 2 to 5 are all normal space		—		Any of areas 2 to 5 is DRAM space	Areas 2 to 5 are all normal space	
PF1DDR	—	0	1	0	1	—	0	1
Pin function	UCAS output	PF1 input	PF1 output	PF1 input	PF1 output	UCAS output	PF1 input	PF1 output
	IRQ14 interrupt*[1]							

Notes: 1. IRQ14 interrupt input when bit ITS14 is cleared to 0 in ITSR.
2. Only in H8S/2678R Series.

- PF0/WAIT

The pin function is switched as shown below according to the operating mode, bit EXPE, bit WAITE in BCR, and bit PF0DDR.

Operating mode	1, 2, 4, 5, 6			3*, 7				
EXPE	—			0		1		
WAITE	0		1	—		0		1
PF0DDR	0	1	—	0	1	0	1	—
Pin function	PF0 input	PF0 output	WAIT input	PF0 input	PF0 output	PF0 input	PF0 output	WAIT input

Note: Only in H8S/2678R Series.

10.15 Port G

Port G is a 7-bit I/O port that also has other functions. The port G has the following registers.

- Port G data direction register (PGDDR)
- Port G data register (PGDR)
- Port G register (PORTG)
- Port Function Control Register 0 (PFCR0)

10.15.1 Port G Data Direction Register (PGDDR)

The individual bits of PGDDR specify input or output for the pins of port G.

PGDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved If read, it returns an undefined value.
6	PG6DDR	0	W	• Modes 1, 2, 4, 5, and 6 Pins PG6 to PG4 function as bus control input/output pins (BREQO, BACK, and BREQ) when the appropriate bus controller settings are made. Otherwise, these pins are I/O ports, and their functions can be switched with PGDDR. When the CS output enable bits (CS3E to CS0E) are set to 1, pins PG3 to PG0 function as CS output pins when the corresponding PGDDR bit is set to 1, and as input ports when the bit is cleared to 0. When CS3E to CS0E are cleared to 0, pins PG3 to PG0 are I/O ports, and their functions can be switched with PGDDR. • Modes 3*[2], 7 (when EXPE = 1) Pins PG6 to PG4 function as bus control input/output pins (BREQO, BACK, and BREQ) when the appropriate bus controller settings are made. Otherwise, these pins are output ports when the corresponding PGDDR bit is set to 1, and as input ports when the bit is cleared to 0. When the CS output enable bits (CS3E to CS0E) are set to 1, pins PG3 to PG0 function as CS output pins when the corresponding PGDDR bit is set to 1, and as input ports when the bit is cleared to 0. When CS3E to CS0E are cleared to 0, pins PG3 to PG0 are I/O ports, and their functions can be switched with PGDDR. • Modes 3*[2], 7 (when EXPE = 0) Pins PG6 to PG0 are I/O ports, and their functions can be switched with PGDDR.
5	PG5DDR	0	W	
4	PG4DDR	0	W	
3	PG3DDR	0	W	
2	PG2DDR	0	W	
1	PG1DDR	0	W	
0	PG0DDR	1/0*[1]	W	

Notes: 1. PG0DDR is initialized to 1 in modes 1, 2, 5, and 6, and to 0 in modes 3, 4, and 7.
2. Only in H8S/2678R Series.

10.15.2 Port G Data Register (PGDR)

PGDR stores output data for the port G pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0, and cannot be modified.
6	PG6DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose I/O.
5	PG5DR	0	R/W	
4	PG4DR	0	R/W	
3	PG3DR	0	R/W	
2	PG2DR	0	R/W	
1	PG1DR	0	R/W	
0	PG0DR	0	R/W	

10.15.3 Port G Register (PORTG)

PORTG shows port G pin states.

PORTG cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved If this bit is read, it will return an undefined value.
6	PG6	Undefined*	R	If a port G read is performed while PGDDR bits are set to 1, the PGDR values are read. If a port G read is performed while PGDDR bits are cleared to 0, the pin states are read.
5	PG5	Undefined*	R	
4	PG4	Undefined*	R	
3	PG3	Undefined*	R	
2	PG2	Undefined*	R	
1	PG1	Undefined*	R	
0	PG0	Undefined*	R	

Note: Determined by the states of pins PG6 to PG0.

10.15.4 Port Function Control Register 0 (PFCR0)

PFCR0 performs I/O port control.

Bit	Bit Name	Initial Value	R/W	Description
7	CS7E	0	R/W	CS7 to CS0 Enable
6	CS6E	0	R/W	These bits enable or disable the corresponding CSn output.
5	CS5E	0	R/W	0: Pin is designated as I/O port
4	CS4E	0	R/W	1: Pin is designated as CSn output pin
3	CS3E	0	R/W	(n = 7 to 0)
2	CS2E	0	R/W	
1	CS1E	0	R/W	
0	CS0E	0	R/W	

10.15.5 Pin Functions

Port G pins also function as bus control signal I/Os. The correspondence between the register specification and the pin functions is shown below.

- PG6/BREQ

The pin function is switched as shown below according to the operating mode, bit EXPE, bit BRLE, and bit PG6DDR.

Operating mode	1, 2, 4, 5, 6			3*, 7				
EXPE	—			0		1		
BRLE	0		1	—		0		1
PG6DDR	0	1	—	0	1	0	1	—
Pin function	PG6 input	PG6 output	BREQ input	PG6 input	PG6 output	PG6 input	PG6 output	BREQ input

Note: Only in H8S/2678R Series.

- PG5/BACK

The pin function is switched as shown below according to the operating mode, bit EXPE, bit BRLE, and bit PG5DDR.

Operating mode	1, 2, 4, 5, 6			3*, 7				
EXPE	—			0		1		
BRLE	0		1	—		0		1
PG5DDR	0	1	—	0	1	0	1	—
Pin function	PG5 input	PG5 output	BACK output	PG5 input	PG5 output	PG5 input	PG5 output	BACK output

Note: Only in H8S/2678R Series.

- PG4/BREQO

The pin function is switched as shown below according to the operating mode, bit EXPE, bit BRLE, bit BREQO, and bit PG4DDR.

Operating mode	1, 2, 4, 5, 6					3*, 7						
EXPE	—					0		1				
BRLE	0		1			—		0		1		
BREQO	—		0		1	—		—		0		1
PG4DDR	0	1	0	1	—	0	1	0	1	0	1	—
Pin function	PG4 input	PG4 output	PG4 input	PG4 output	BREQO output	PG4 input	PG4 output	PG4 input	PG4 output	PG4 input	PG4 output	BREQO output

Note: Only in H8S/2678R Series.

- PG3/CS3/RAS3*/CAS*, PG2/CS2/RAS2*/RAS*

The pin function is switched as shown below according to the operating mode, bit PGnDDR, bit CSnE, and bits RMTS2 to RMTS0.

Operating mode	1, 2, 4, 5, 6							3*, 7								
EXPE	—							0		1						
CSnE	0		1					—		0		1				
RMTS2 to RMTS0	—		Area n normal space		Area n DRAM space	Area 3 synchro-nous DRAM* space	Area 2 synchro-nous DRAM* space	—		—		Area n normal space		Area n DRAM space	Area 3 synchro-nous DRAM* space	Area 2 synchro-nous DRAM* space
PGnDDR	0	1	0	1	—	—	—	0	1	0	1	0	1	—	—	—
Pin function	PGn input	PGn output	PGn input	CSn output	RASn output	CAS* output	RAS* output	PGn input	PGn output	PGn input	PGn output	PGn input	CSn output	RASn output	CAS* output	RAS* output

(n = 3 or 2)

Note: Only in H8S/2678R Series.

- PG1/CS1, PG0/CS0

The pin function is switched as shown below according to the operating mode, bit PGnDDR, and bit CSnE.

Operating mode	1, 2, 4, 5, 6				3*, 7					
EXPE	—				0		1			
CSnE	0		1		—		0		1	
PGnDDR	0	1	0	1	0	1	0	1	0	1
Pin function	PGn input	PGn output	PGn input	CSn output	PGn input	PGn output	PGn input	PGn output	PGn input	CSn output

(n =1 or 0)

Note: Only in H8S/2678R Series.

10.16 Port H

Port H is a 4-bit I/O port that also has other functions. The port H has the following registers. For details on the port function control register 0, refer to section 10.15.4, Port Function Control Register 0 (PFCR0), and for details on the port function control register 2, refer to section 10.3.5, Port Function Control Register 2 (PFCR2).

- Port H data direction register (PHDDR)
- Port H data register (PHDR)
- Port H register (PORTH)

- Port Function Control Register 0 (PFCR0)
- Port Function Control Register 2 (PFCR2)

10.16.1 Port H Data Direction Register (PHDDR)

The individual bits of PHDDR specify input or output for the pins of port H.

PHDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved If these bits are read, they will return an undefined value.
3	PH3DDR	0	W	• Modes 1, 2, 3* (when EXPE = 1), 4, 5, 6, and 7 (when EXPE = 1) When the OE output enable bit (OEE) and OE output select bit (OES) are set to 1, pin PH3 functions as the OE output pin. Otherwise, when bit CS7E is set to 1, pin PH3 functions as a CS output pin when the corresponding PHDDR bit is set to 1, and as an input port when the bit is cleared to 0. When bit CS7E is cleared to 0, pin PH3 is an I/O port, and its function can be switched with PHDDR. When the CS output enable bits (CS6E to CS4E) are set to 1, pins PH2 to PH0 function as CS output pins when the corresponding PHDDR bit is set to 1, and as I/O ports when the bit is cleared to 0. When CS6E to CS4E are cleared to 0, pins PH2 to PH0 are I/O ports, and their functions can be switched with PHDDR. • Mode3* (EXPE = 0) and Mode 7 (when EXPE = 0) Pins PH3 to PH0 are I/O ports, and their functions can be switched with PHDDR.
2	PH2DDR	0	W	
1	PH1DDR	0	W	
0	PH0DDR	0	W	

Note: Only in H8S/2678R Series.

10.16.2 Port H Data Register (PHDR)

PHDR stores output data for the port H pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are reserved; they are always read as 0 and cannot be modified.
3	PH3DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
2	PH2DR	0	R/W	
1	PH1DR	0	R/W	
0	PH0DR	0	R/W	

10.16.3 Port H Register (PORTH)

PORTH shows port H pin states.

PORTH cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved If these bits are read, they will return an undefined value.
3	PH3	Undefined*	R	If a port H read is performed while PHDDR bits are set to 1, the PHDR values are read. If a port H read is performed while PHDDR bits are cleared to 0, the pin states are read.
2	PH2	Undefined*	R	
1	PH1	Undefined*	R	
0	PH0	Undefined*	R	

Note: Determined by the states of pins PH3 to PH0.

10.16.4 Pin Functions

Port H pins also function as bus control signal I/Os and external interrupt inputs. The correspondence between the register specification and the pin functions is shown below.

- PH3/CS7/OE/CKE*[2]/(IRQ7)

The pin function is switched as shown below according to the operating mode, bit EXPE, bit OEE, bit OES, bit CS7E, and bit PH3DDR.

Operating mode	1, 2, 4, 5, 6										3*2, 7											
EXPE	—										0		1									
CEE	0				1						—		0				1					
CES	—				0				1		—		0				1					
Area 2 to 5	—				—				Normal space or DRAM space	syn-chronous DRAM space*2	—		—				—				Normal space or DRAM space	syn-chronous DRAM space*2
CS7E	0		1		0		1		—	—	—		0		1		0		1		—	—
PH3DDR	0	1	0	1	0	1	0	1	—	—	0	1	0	1	0	1	0	1	0	1	—	—
Pin function	PH3 input	PH3 output	PH3 input	CS7 output	PH3 input	PH3 output	PH3 input	CS7 output	OE output	CKE*2 output	PH3 input	PH3 output	PH3 input	PH3 output	PH3 input	CS7 output	PH3 input	PH3 output	PH3 input	CS7 output	OE output	CKE*2 output
	IRQ7 input*1																					

Notes: 1. IRQ7 interrupt input pin when bit ITS7 is set to 1 in ITSR
2. Only in H8S/2678R Series.

- PH2/CS6/(IRQ6)

The pin function is switched as shown below according to the operating mode, bit PH2DDR, and bit CS6E.

Operating mode	1, 2, 4, 5, 6				3*2, 7					
EXPE	—				0		1			
CS6E	0		1		—		0		1	
PH2DDR	0	1	0	1	0	1	0	1	0	1
Pin function	PH2 input	PH2 output	PH2 input	CS6 output	PH2 input	PH2 output	PH2 input	PH2 output	PH2 input	CS6 output
	IRQ6 interrupt input*									

Notes: 1. IRQ6 interrupt input pin when bit ITS6 is set to 1 in ITSR.
2. Only in H8S/2678R Series.

- PH1/CS5/RAS5*/SDRAMϕ*

The pin function is switched as shown below according to the operating mode, bit EXPE, bit CS5E, bits RMTS2 to RMTS0, and bit PH1DDR.

Operating mode	1, 2, 4, 5, 6							3*, 7									—
EXPE	—							0		1							—
Area 5	Normal space				DRAM space			—		Normal space				DRAM space			—
DCTL	0																1
CS5E	0		1		0		1	—		0		1		0		1	—
PH1DDR	0	1	0	1	0	1	—	0	1	0	1	0	1	0	1	—	—
Pin function	PH1 input	PH1 output	PH1 input	CS5 output	PH1 input	PH1 output	RAS5* output	PH1 input	PH1 output	PH1 input	PH1 output	PH1 input	CS5 output	PH1 input	PH1 output	RAS5* output	SDRAM* ϕ output

Note: Only in H8S/2678R Series.

- PH0/CS4/RAS4*/WE*

The pin function is switched as shown below according to the operating mode, bit EXPE, bit CS4E, bits RMTS2 to RMTS0, and bit PH0DDR.

Operating mode	1, 2, 4, 5, 6						3*, 7							
EXPE	—						0		1					
Area 4	—		Normal space		DRAM space	Synchronous DRAM* space	—		—		Normal space		DRAM space	Synchronous DRAM* space
SC4E	0		1				—		0		1			
PH1DDR	0	1	0	1	—	—	0	1	0	1	0	1	—	—
Pin function	PH0 input	PH0 output	PH0 input	CS4 output	RAS4* output	WE* output	PH0 input	PH0 output	PH0 input	PH0 output	PH0 input	CS4 output	RAS4* output	WE* output

Note: Only in H8S/2678R Series.

Section 11 16-Bit Timer Pulse Unit (TPU)

This LSI has an on-chip 16-bit timer pulse unit (TPU) that comprises six 16-bit timer channels. The function list of the 16-bit timer unit and its block diagram are shown in table 11.1 and figure 11.1, respectively.

11.1 Features

- Maximum 16-pulse input/output
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  Waveform output at compare match
  Input capture function
  Counter clear operation
  Synchronous operations:
    Multiple timer counters (TCNT) can be written to simultaneously
    Simultaneous clearing by compare match and input capture possible
    Register simultaneous input/output possible by counter synchronous operation
  Maximum of 15-phase PWM output possible by combination with synchronous operation
- Buffer operation settable for channels 0 and 3
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
- Cascaded operation
- Fast access via internal 16-bit bus
- 26 interrupt sources
- Automatic transfer of register data
- Programmable pulse generator (PPG) output trigger can be generated
- A/D converter conversion start trigger can be generated
- Module stop mode can be set

Table 11.1 TPU Functions

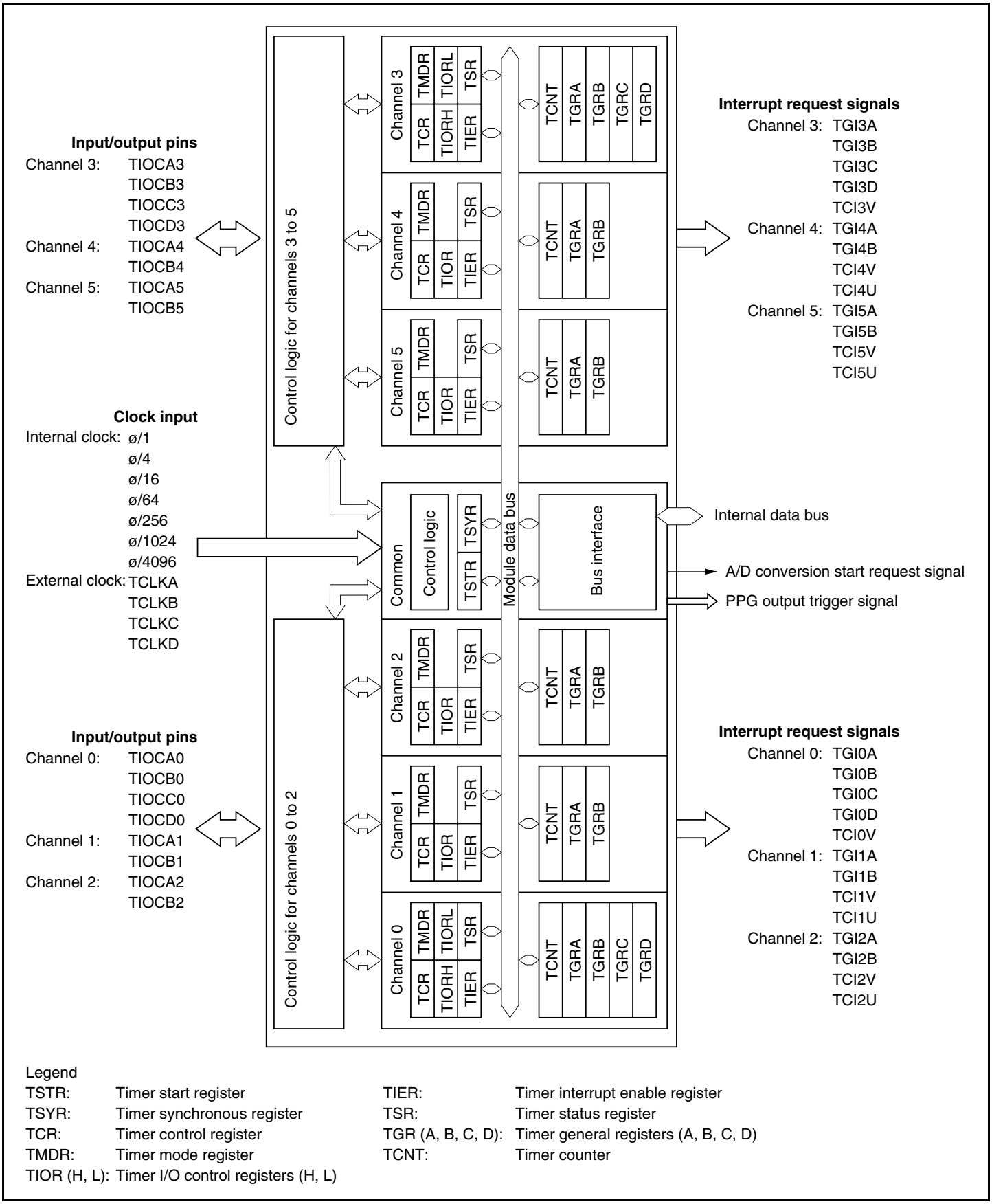
Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock		ø/1 ø/4 ø/16 ø/64 TCLKA TCLKB TCLKC TCLKD	ø/1 ø/4 ø/16 ø/64 ø/256 TCLKA TCLKB	ø/1 ø/4 ø/16 ø/64 ø/1024 TCLKA TCLKB TCLKC	ø/1 ø/4 ø/16 ø/64 ø/256 ø/1024 ø/4096 TCLKA	ø/1 ø/4 ø/16 ø/64 ø/1024 TCLKA TCLKC	ø/1 ø/4 ø/16 ø/64 ø/256 TCLKA TCLKC TCLKD
General registers (TGR)		TGRA_0 TGRB_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4	TGRA_5 TGRB_5
General registers/ buffer registers		TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TIOCA5 TIOCB5
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	○	○	○	○	○	○
	1 output	○	○	○	○	○	○
	Toggle output	○	○	○	○	○	○
Input capture function		○	○	○	○	○	○
Synchronous operation		○	○	○	○	○	○
PWM mode		○	○	○	○	○	○
Phase counting mode		—	○	○	—	○	○
Buffer operation		○	—	—	○	—	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
DMAC activation	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture
A/D converter trigger	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture
PPG trigger	TGRA/ TGRB compare match or input capture	TGRA/ TGRB compare match or input capture	TGRA/ TGRB compare match or input capture	TGRA/ TGRB compare match or input capture	—	—
Interrupt sources	5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	4 sources • Compare match or input capture 4A • Compare match or input capture 4B • Overflow • Underflow	4 sources • Compare match or input capture 5A • Compare match or input capture 5B • Overflow • Underflow

Legend

○ : Possible

— : Not possible



Legend

TSTR:	Timer start register	TIER:	Timer interrupt enable register
TSYR:	Timer synchronous register	TSR:	Timer status register
TCR:	Timer control register	TGR (A, B, C, D):	Timer general registers (A, B, C, D)
TMDR:	Timer mode register	TCNT:	Timer counter
TIOR (H, L):	Timer I/O control registers (H, L)		

Figure 11.1 Block Diagram of TPU

11.2 Input/Output Pins

Table 11.2 Pin Configuration

Channel	Symbol	I/O	Function
All	TCLKA	Input	External clock A input pin (Channel 1 and 5 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channel 1 and 5 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 and 4 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 and 4 phase counting mode B phase input)
0	TIOCA0	I/O	TGRA_0 input capture input/output compare output/PWM output pin
	TIOCB0	I/O	TGRB_0 input capture input/output compare output/PWM output pin
	TIOCC0	I/O	TGRC_0 input capture input/output compare output/PWM output pin
	TIOCD0	I/O	TGRD_0 input capture input/output compare output/PWM output pin
1	TIOCA1	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOCB1	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOCA2	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOCB2	I/O	TGRB_2 input capture input/output compare output/PWM output pin
3	TIOCA3	I/O	TGRA_3 input capture input/output compare output/PWM output pin
	TIOCB3	I/O	TGRB_3 input capture input/output compare output/PWM output pin
	TIOCC3	I/O	TGRC_3 input capture input/output compare output/PWM output pin
	TIOCD3	I/O	TGRD_3 input capture input/output compare output/PWM output pin
4	TIOCA4	I/O	TGRA_4 input capture input/output compare output/PWM output pin
	TIOCB4	I/O	TGRB_4 input capture input/output compare output/PWM output pin
5	TIOCA5	I/O	TGRA_5 input capture input/output compare output/PWM output pin
	TIOCB5	I/O	TGRB_5 input capture input/output compare output/PWM output pin

11.3 Register Descriptions

The TPU has the following registers in each channel.

- Timer control register_0 (TCR_0)
- Timer mode register_0 (TMDR_0)
- Timer I/O control register H_0 (TIORH_0)
- Timer I/O control register L_0 (TIORL_0)
- Timer interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)
- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register _1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)
- Timer control register_2 (TCR_2)
- Timer mode register_2 (TMDR_2)
- Timer I/O control register_2 (TIOR_2)
- Timer interrupt enable register_2 (TIER_2)
- Timer status register_2 (TSR_2)
- Timer counter_2 (TCNT_2)
- Timer general register A_2 (TGRA_2)
- Timer general register B_2 (TGRB_2)
- Timer control register_3 (TCR_3)
- Timer mode register_3 (TMDR_3)
- Timer I/O control register H_3 (TIORH_3)
- Timer I/O control register L_3 (TIORL_3)
- Timer interrupt enable register_3 (TIER_3)
- Timer status register_3 (TSR_3)
- Timer counter_3 (TCNT_3)

- Timer general register A_3 (TGRA_3)
- Timer general register B_3 (TGRB_3)
- Timer general register C_3 (TGRC_3)
- Timer general register D_3 (TGRD_3)
- Timer control register_4 (TCR_4)
- Timer mode register_4 (TMDR_4)
- Timer I/O control register _4 (TIOR_4)
- Timer interrupt enable register_4 (TIER_4)
- Timer status register_4 (TSR_4)
- Timer counter_4 (TCNT_4)
- Timer general register A_4 (TGRA_4)
- Timer general register B_4 (TGRB_4)
- Timer control register_5 (TCR_5)
- Timer mode register_5 (TMDR_5)
- Timer I/O control register_5 (TIOR_5)
- Timer interrupt enable register_5 (TIER_5)
- Timer status register_5 (TSR_5)
- Timer counter_5 (TCNT_5)
- Timer general register A_5 (TGRA_5)
- Timer general register B_5 (TGRB_5)

Common Registers

- Timer start register (TSTR)
- Timer synchronous register (TSYR)

11.3.1 Timer Control Register (TCR)

The TCR registers control the TCNT operation for each channel. The TPU has a total of six TCR registers, one for each channel. TCR register settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial Value	R/W	Description
7	CCLR2	0	R/W	Counter Clear 2 to 0
6	CCLR1	0	R/W	These bits select the TCNT counter clearing source. See tables 11.3 and 11.4 for details.
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. ø/4 both edges = ø/2 rising edge). If phase counting mode is used on channels 1, 2, 4, and 5, this setting is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is ø/4 or slower. This setting is ignored if the input clock is ø/1, or when overflow/underflow of another channel is selected. 00: Count at rising edge 01: Count at falling edge 1x: Count at both edges Legend: x: Don't care
2	TPSC2	0	R/W	Time Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 11.5 to 11.10 for details.
0	TPSC0	0	R/W	

Table 11.3 CCLR2 to CCLR0 (Channels 0 and 3)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*[1]
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*[2]
		1	0	TCNT cleared by TGRD compare match/input capture*[2]
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*[1]

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

Table 11.4 CCLR2 to CCLR0 (Channels 1, 2, 4, and 5)

Channel	Bit 7 Reserved*[2]	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*[1]

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot be modified.

Table 11.5 TPSC2 to TPSC0 (Channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on ø/1
			1	Internal clock: counts on ø/4
		1	0	Internal clock: counts on ø/16
			1	Internal clock: counts on ø/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	External clock: counts on TCLKD pin input

Table 11.6 TPSC2 to TPSC0 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on ø/1
			1	Internal clock: counts on ø/4
		1	0	Internal clock: counts on ø/16
			1	Internal clock: counts on ø/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on ø/256
			1	Counts on TCNT2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

Table 11.7 TPSC2 to TPSC0 (Channel 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on ø/1
			1	Internal clock: counts on ø/4
		1	0	Internal clock: counts on ø/16
			1	Internal clock: counts on ø/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	Internal clock: counts on ø/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 11.8 TPSC2 to TPSC0 (Channel 3)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3	0	0	0	Internal clock: counts on ø/1
			1	Internal clock: counts on ø/4
		1	0	Internal clock: counts on ø/16
			1	Internal clock: counts on ø/64
	1	0	0	External clock: counts on TCLKA pin input
			1	Internal clock: counts on ø/1024
		1	0	Internal clock: counts on ø/256
			1	Internal clock: counts on ø/4096

Table 11.9 TPSC2 to TPSC0 (Channel 4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
4	0	0	0	Internal clock: counts on ø/1
			1	Internal clock: counts on ø/4
		1	0	Internal clock: counts on ø/16
			1	Internal clock: counts on ø/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKC pin input
		1	0	Internal clock: counts on ø/1024
			1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

Table 11.10 TPSC2 to TPSC0 (Channel 5)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	0	Internal clock: counts on ø/1
			1	Internal clock: counts on ø/4
		1	0	Internal clock: counts on ø/16
			1	Internal clock: counts on ø/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKC pin input
		1	0	Internal clock: counts on ø/256
			1	External clock: counts on TCLKD pin input

Note: This setting is ignored when channel 5 is in phase counting mode.

11.3.2 Timer Mode Register (TMDR)

TMDR registers are used to set the operating mode for each channel. The TPU has six TMDR registers, one for each channel. TMDR register settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved
6	—	1	—	These bits are always read as 1 and cannot be modified.
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: TGRB operates normally 1: TGRB and TGRD used together for buffer operation
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified. 0: TGRA operates normally 1: TGRA and TGRC used together for buffer operation
3	MD3	0	R/W	Modes 3 to 0 These bits are used to set the timer operating mode. MD3 is a reserved bit. In a write, it should always be written with 0. See table 11.11 for details.
2	MD2	0	R/W	
1	MD1	0	R/W	
0	MD0	0	R/W	

Table 11.11 MD3 to MD0

Bit 3 MD3*[1]	Bit 2 MD2*[2]	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	x	x	x	—

Legend: x: Don't care

Notes: 1. MD3 is a reserved bit. In a write, it should always be written with 0.
2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 should always be written to MD2.

11.3.3 Timer I/O Control Register (TIOR)

TIOR registers control the TGR registers. The TPU has eight TIOR registers, two each for channels 0 and 3, and one each for channels 1, 2, 4, and 5. Care is required since TIOR is affected by the TMDR setting.

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIOR_4, TIOR_5

Bit	Bit Name	Initial Value	R/W	Description
7	IOB3	0	R/W	I/O Control B3 to B0
6	IOB2	0	R/W	Specify the function of TGRB. For details, see tables 11.12, 11.14, 11.15, 11.16, 11.18, and 11.19.
5	IOB1	0	R/W	
4	IOB0	0	R/W	
3	IOA3	0	R/W	I/O Control A3 to A0
2	IOA2	0	R/W	Specify the function of TGRA. For details, see tables 11.20, 11.22, 11.23, 11.24, 11.26, and 11.27.
1	IOA1	0	R/W	
0	IOA0	0	R/W	

TIORL_0, TIORL_3

Bit	Bit Name	Initial Value	R/W	Description
7	IOD3	0	R/W	I/O Control D3 to D0
6	IOD2	0	R/W	Specify the function of TGRD. For details, see tables 11.13, and 11.17.
5	IOD1	0	R/W	
4	IOD0	0	R/W	
3	IOC3	0	R/W	I/O Control C3 to C0
2	IOC2	0	R/W	Specify the function of TGRC. For details, see tables 11.21, and 11.25
1	IOC1	0	R/W	
0	IOC0	0	R/W	

Table 11.12 TIORH_0

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_0 Function**	**TIOCB0 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB0 pin Input capture at rising edge
			1		Capture input source is TIOCB0 pin Input capture at falling edge
		1	x		Capture input source is TIOCB0 pin Input capture at both edges
	1	x	x		Capture input source is channel 1/count clock Input capture at TCNT_1 count- up/count-down*

Legend: x: Don’t care

Note: When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and ø/1 is used as the TCNT_1 count clock, this setting is invalid and input capture is not generated.

Table 11.13 TIORL_0

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_0 Function**	**TIOCD0 Pin Function**
0	0	0	0	Output compare register*2	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD0 pin Input capture at rising edge
			1		Capture input source is TIOCD0 pin Input capture at falling edge
		1	x		Capture input source is TIOCD0 pin Input capture at both edges
	1	x	x		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down*1

Legend: x: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and ø/1 is used as the TCNT_1 count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 11.14 TIOR_1

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_1 Function**	**TIOCB1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 pin Input capture at rising edge
			1		Capture input source is TIOCB1 pin Input capture at falling edge
		1	x		Capture input source is TIOCB1 pin Input capture at both edges
	1	x	x		TGRC_0 compare match/input capture Input capture at generation of TGRC_0 compare match/input capture

Legend: x: Don’t care

Table 11.15 TIOR_2

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_2 Function**	**TIOCB2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCB2 pin Input capture at rising edge
			1		Capture input source is TIOCB2 pin Input capture at falling edge
		1	x		Capture input source is TIOCB2 pin Input capture at both edges

Legend: x: Don’t care

Table 11.16 TIORH_3

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_3 Function**	**TIOCB3 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB3 pin Input capture at rising edge
			1		Capture input source is TIOCB3 pin Input capture at falling edge
		1	x		Capture input source is TIOCB3 pin Input capture at both edges
	1	x	x		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down*

Legend: x: Don’t care

Note: When bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and ø/1 is used as the TCNT_4 count clock, this setting is invalid and input capture is not generated.

Table 11.17 TIORL_3

Table 11.18 TIOR_4

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_4 Function**	**TIOCB4 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB4 pin Input capture at rising edge
			1		Capture input source is TIOCB4 pin Input capture at falling edge
		1	x		Capture input source is TIOCB4 pin Input capture at both edges
	1	x	x		Capture input source is TGRC_3 compare match/input capture Input capture at generation of TGRC_3 compare match/input capture

Legend: x: Don’t care

Table 11.19 TIOR_5

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_5 Function**	**TIOCB5 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCB5 pin Input capture at rising edge
			1		Capture input source is TIOCB5 pin Input capture at falling edge
		1	x		Capture input source is TIOCB5 pin Input capture at both edges

Legend: x: Don’t care

Table 11.20 TIORH_0

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_0 Function**	**TIOCA0 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin Input capture at rising edge
			1		Capture input source is TIOCA0 pin Input capture at falling edge
		1	x		Capture input source is TIOCA0 pin Input capture at both edges
	1	x	x		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

Legend: x: Don’t care

Table 11.21 TIORL_0

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_0 Function**	**TIOCC0 Pin Function**
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCC0 pin Input capture at rising edge
			1		Capture input source is TIOCC0 pin Input capture at falling edge
		1	x		Capture input source is TIOCC0 pin Input capture at both edges
	1	x	x		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

Legend: x: Don't care

Note: When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 11.22 TIOR_1

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_1 Function**	**TIOCA1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA1 pin Input capture at rising edge
			1		Capture input source is TIOCA1 pin Input capture at falling edge
		1	x		Capture input source is TIOCA1 pin Input capture at both edges
	1	x	x		Capture input source is TGRA_0 compare match/input capture Input capture at generation of channel 0/TGRA_0 compare match/input capture

Legend: x: Don’t care

Table 11.23 TIOR_2

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_2 Function**	**TIOCA2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCA2 pin Input capture at rising edge
			1		Capture input source is TIOCA2 pin Input capture at falling edge
		1	x		Capture input source is TIOCA2 pin Input capture at both edges

Legend: x: Don’t care

Table 11.24 TIORH_3

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_3 Function**	**TIOCA3 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA3 pin Input capture at rising edge
			1		Capture input source is TIOCA3 pin Input capture at falling edge
		1	x		Capture input source is TIOCA3 pin Input capture at both edges
	1	x	x		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down

Legend: x: Don’t care

Table 11.25 TIORL_3

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_3 Function**	**TIOCC3 Pin Function**
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCC3 pin Input capture at rising edge
			1		Capture input source is TIOCC3 pin Input capture at falling edge
		1	x		Capture input source is TIOCC3 pin Input capture at both edges
	1	x	x		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down

Legend: x: Don't care

Note: When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 11.26 TIOR_4

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_4 Function**	**TIOCA4 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA4 pin Input capture at rising edge
			1		Capture input source is TIOCA4 pin Input capture at falling edge
		1	x		Capture input source is TIOCA4 pin Input capture at both edges
	1	x	x		Capture input source is TGRA_3 compare match/input capture Input capture at generation of TGRA_3 compare match/input capture

Legend: x: Don’t care

Table 11.27 TIOR_5

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_5 Function**	**TIOCA5 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	x	0	0	Input capture register	Input capture source is TIOCA5 pin Input capture at rising edge
			1		Input capture source is TIOCA5 pin Input capture at falling edge
		1	x		Input capture source is TIOCA5 pin Input capture at both edges

Legend: x: Don’t care

11.3.4 Timer Interrupt Enable Register (TIER)

TIER registers control enabling or disabling of interrupt requests for each channel. The TPU has six TIER registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TTGE	0	R/W	A/D Conversion Start Request Enable Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match. 0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled
6	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1, 2, 4, and 5. In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0 and 3. In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled

Bit	Bit Name	Initial value	R/W	Description
2	TGIEC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channels 0 and 3. In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGIC) by TGFC bit disabled 1: Interrupt requests (TGIC) by TGFC bit enabled
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

11.3.5 Timer Status Register (TSR)

TSR registers indicate the status of each channel. The TPU has six TSR registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in channels 1, 2, 4, and 5. In channels 0 and 3, bit 7 is reserved. It is always read as 1 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1, 2, 4, and 5 are set to phase counting mode. In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified. [Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF) [Clearing condition] When 0 is written to TCFU after reading TCFU = 1
4	TCFV	0	R/(W)*	Overflow Flag Status flag that indicates that TCNT overflow has occurred. [Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000) [Clearing condition] When 0 is written to TCFV after reading TCFV = 1

Bit	Bit Name	Initial value	R/W	Description
3	TGFD	0	R/(W)*	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of TGRD input capture or compare match in channels 0 and 3. In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register [Clearing conditions] • When DTC is activated by TGID interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFD after reading TGFD = 1
2	TGFC	0	R/(W)*	Input Capture/Output Compare Flag C Status flag that indicates the occurrence of TGRC input capture or compare match in channels 0 and 3. In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIC interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFC after reading TGFC = 1

Bit	Bit Name	Initial value	R/W	Description
1	TGFB	0	R/(W)*	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGRB input capture or compare match. [Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
0	TGFA	0	R/(W)*	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA input capture or compare match. [Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1

Note: Only 0 can be written, for flag clearing.

11.3.6 Timer Counter (TCNT)

The TCNT registers are 16-bit readable/writable counters. The TPU has six TCNT counters, one for each channel.

The TCNT counters are initialized to H'0000 by a reset, or in hardware standby mode.

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

11.3.7 Timer General Register (TGR)

The TGR registers are 16-bit readable/writable registers with a dual function as output compare and input capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two each for channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for operation as buffer registers. The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit. TGR buffer register combinations are TGRA–TGRC and TGRB–TGRD.

11.3.8 Timer Start Register (TSTR)

TSTR selects operation/stoppage for channels 0 to 5. When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bit	Bit Name	Initial value	R/W	Description
7 6	—	0	—	Reserved These bits should always be written with 0.
5	CST5	0	R/W	Counter Start 5 to 0 These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_5 to TCNT_0 count operation is stopped 1: TCNT_5 to TCNT_0 performs count operation
4	CST4	0	R/W	
3	CST3	0	R/W	
2	CST2	0	R/W	
1	CST1	0	R/W	
0	CST0	0	R/W	

11.3.9 Timer Synchronous Register (TSYR)

TSYR selects independent operation or synchronous operation for the TCNT counters of channels 0 to 5. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

Bit	Bit Name	Initial value	R/W	Description
7	—	—	R/W	Reserved These bits should always be written with 0.
6	—	—	R/W	
5	SYNC5	0	R/W	Timer Synchronization 5 to 0 These bits select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, synchronous presetting of multiple channels, and synchronous clearing through counter clearing on another channel are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR2 to CCLR0 in TCR. 0: TCNT_5 to TCNT_0 operates independently (TCNT presetting /clearing is unrelated to other channels) 1: TCNT_5 to TCNT_0 performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is possible)
4	SYNC4	0	R/W	
3	SYNC3	0	R/W	
2	SYNC2	0	R/W	
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

11.4 Operation

11.4.1 Basic Functions

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Counter Operation: When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

1. Example of count operation setting procedure
   Figure 11.2 shows an example of the count operation setting procedure.

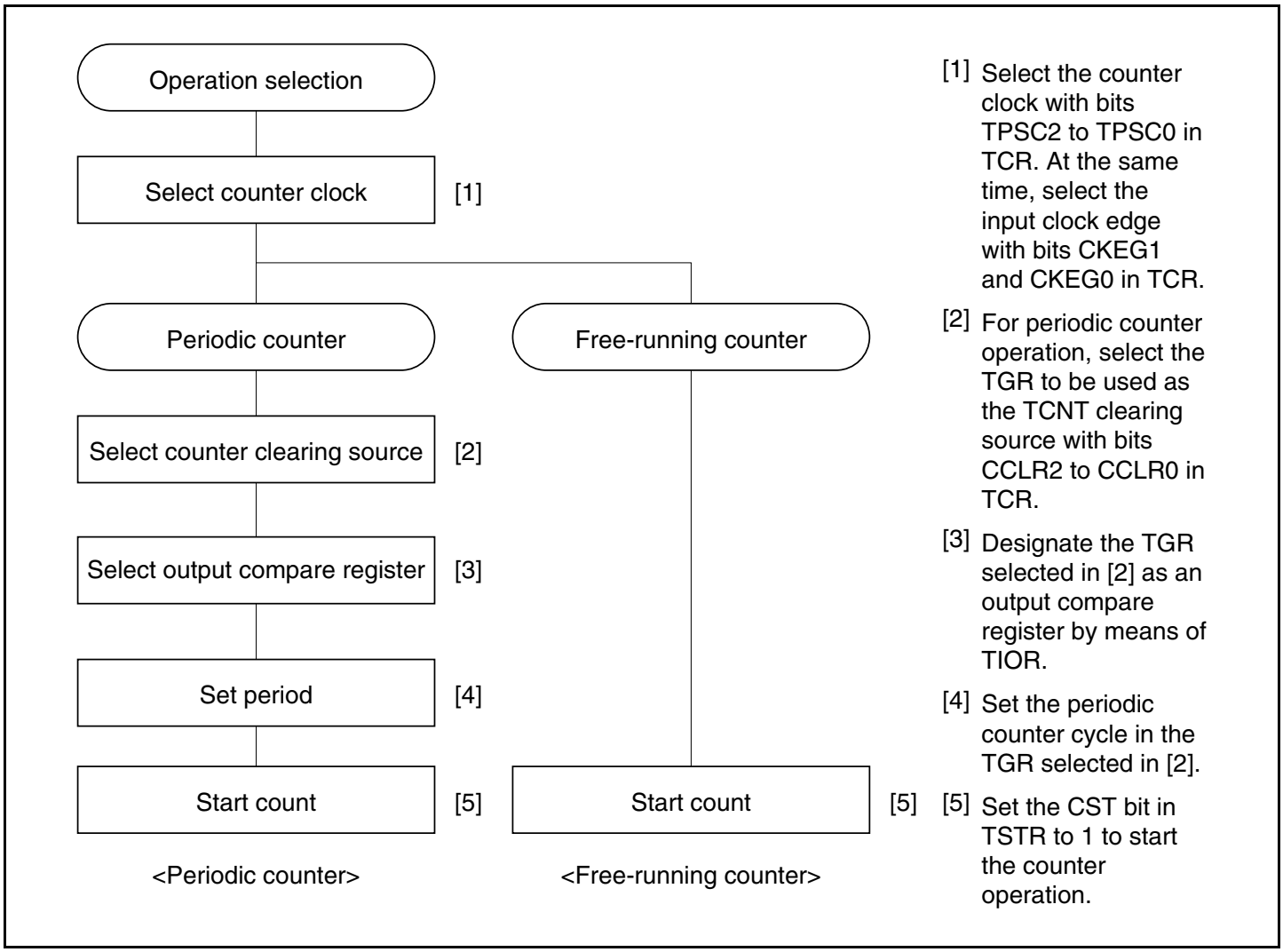


Figure 11.2 Example of Counter Operation Setting Procedure

2. Free-running count operation and periodic count operation

   Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (changes from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

   Figure 11.3 illustrates free-running counter operation.

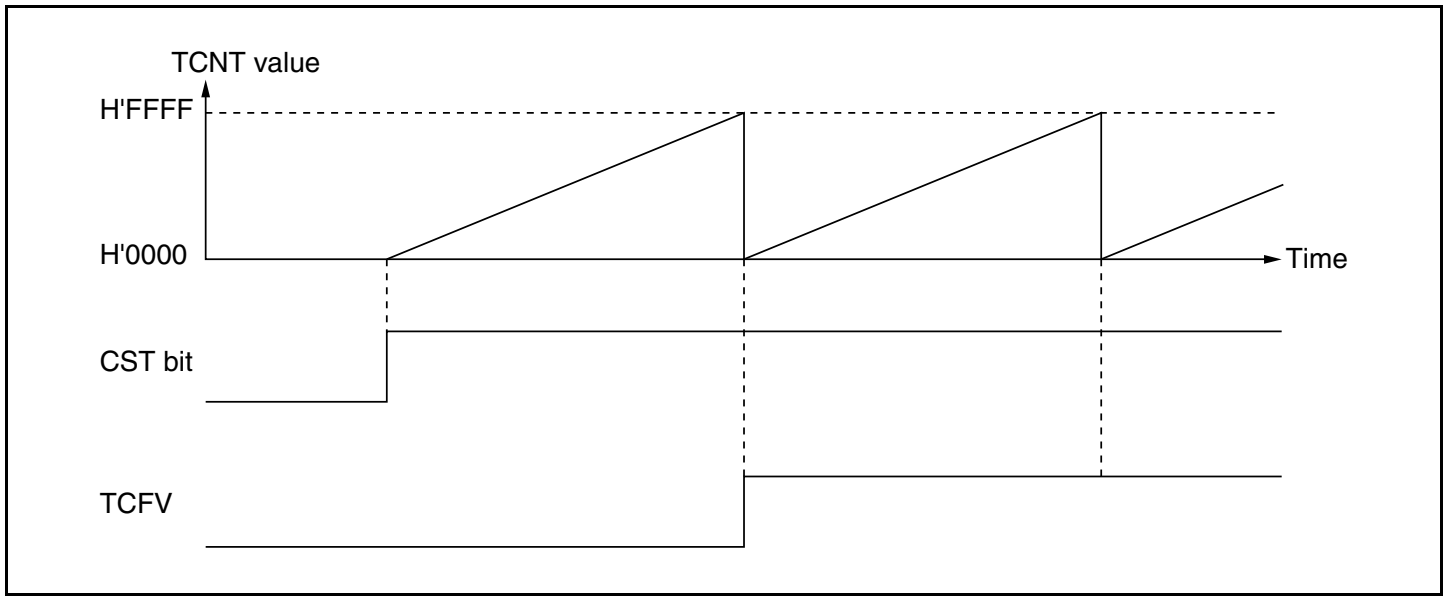


Figure 11.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts count-up operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 11.4 illustrates periodic counter operation.

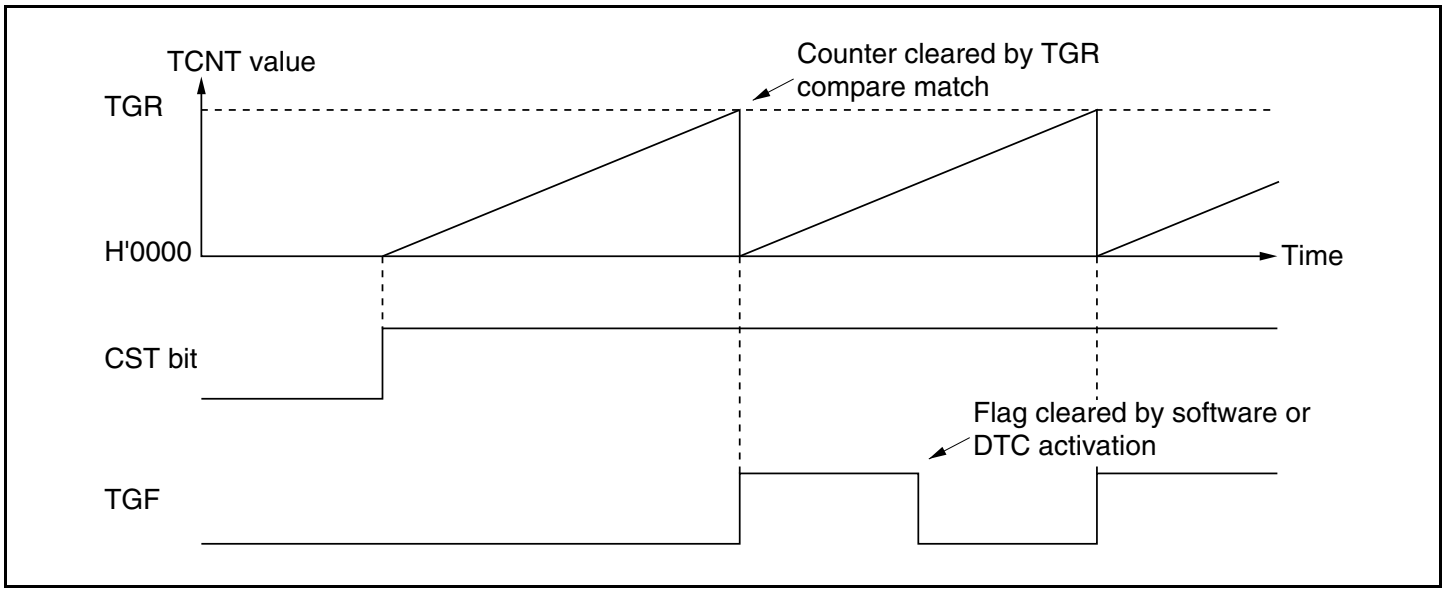
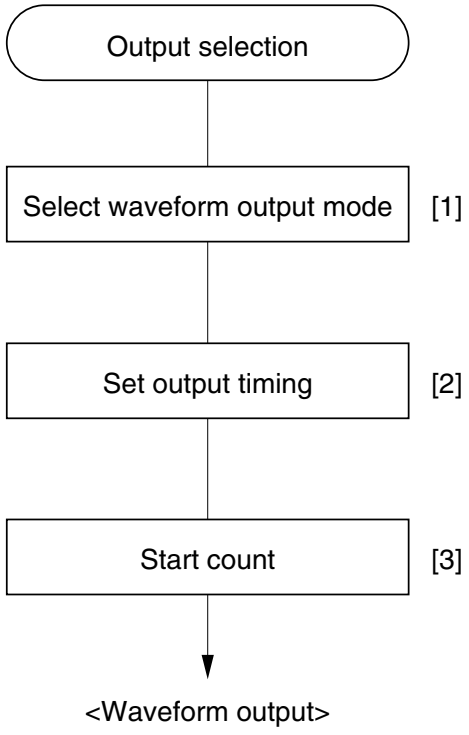


Figure 11.4 Periodic Counter Operation

Waveform Output by Compare Match: The TPU can perform 0, 1, or toggle output from the corresponding output pin using a compare match.

1. Example of setting procedure for waveform output by compare match

   Figure 11.5 shows an example of the setting procedure for waveform output by a compare match.



[1] Select initial value 0 output or 1 output, and compare match output value 0 output, 1 output, or toggle output, by means of TIOR. The set initial value is output at the TIOC pin until the first compare match occurs.

[2] Set the timing for compare match generation in TGR.

[3] Set the CST bit in TSTR to 1 to start the count operation.

Figure 11.5 Example of Setting Procedure for Waveform Output by Compare Match

2. Examples of waveform output operation

Figure 11.6 shows an example of 0 output/1 output.

In this example, TCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level match, the pin level does not change.

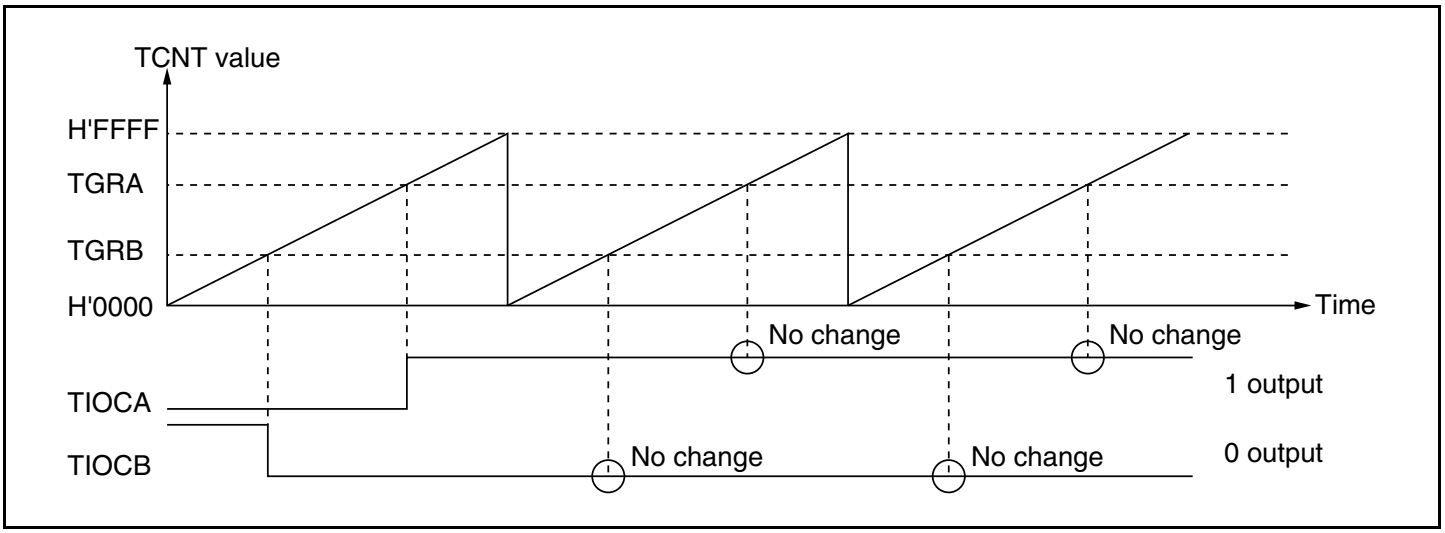


Figure 11.6 Example of 0 Output/1 Output Operation

Figure 11.7 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

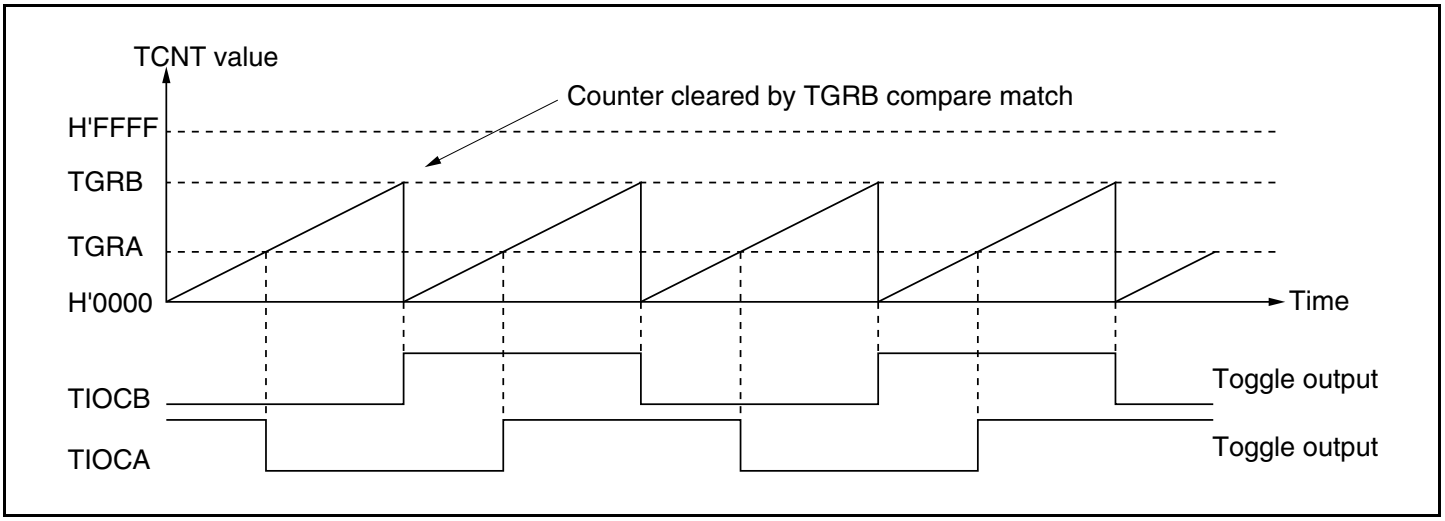


Figure 11.7 Example of Toggle Output Operation

Input Capture Function: The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detection edge. For channels 0, 1, 3, and 4, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 3, ø/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if ø/1 is selected.

1. Example of setting procedure for input capture operation

   Figure 11.8 shows an example of the setting procedure for input capture operation.

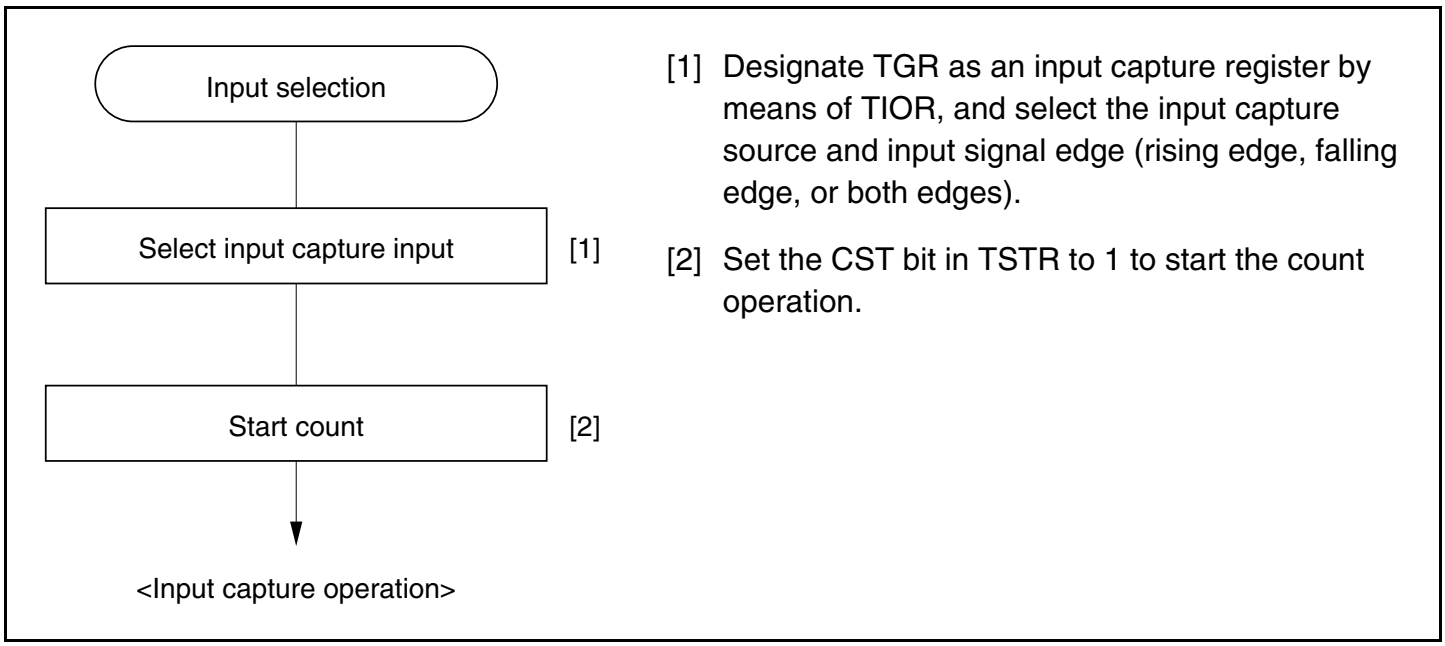


Figure 11.8 Example of Setting Procedure for Input Capture Operation

2. Example of input capture operation

   Figure 11.9 shows an example of input capture operation.

   In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

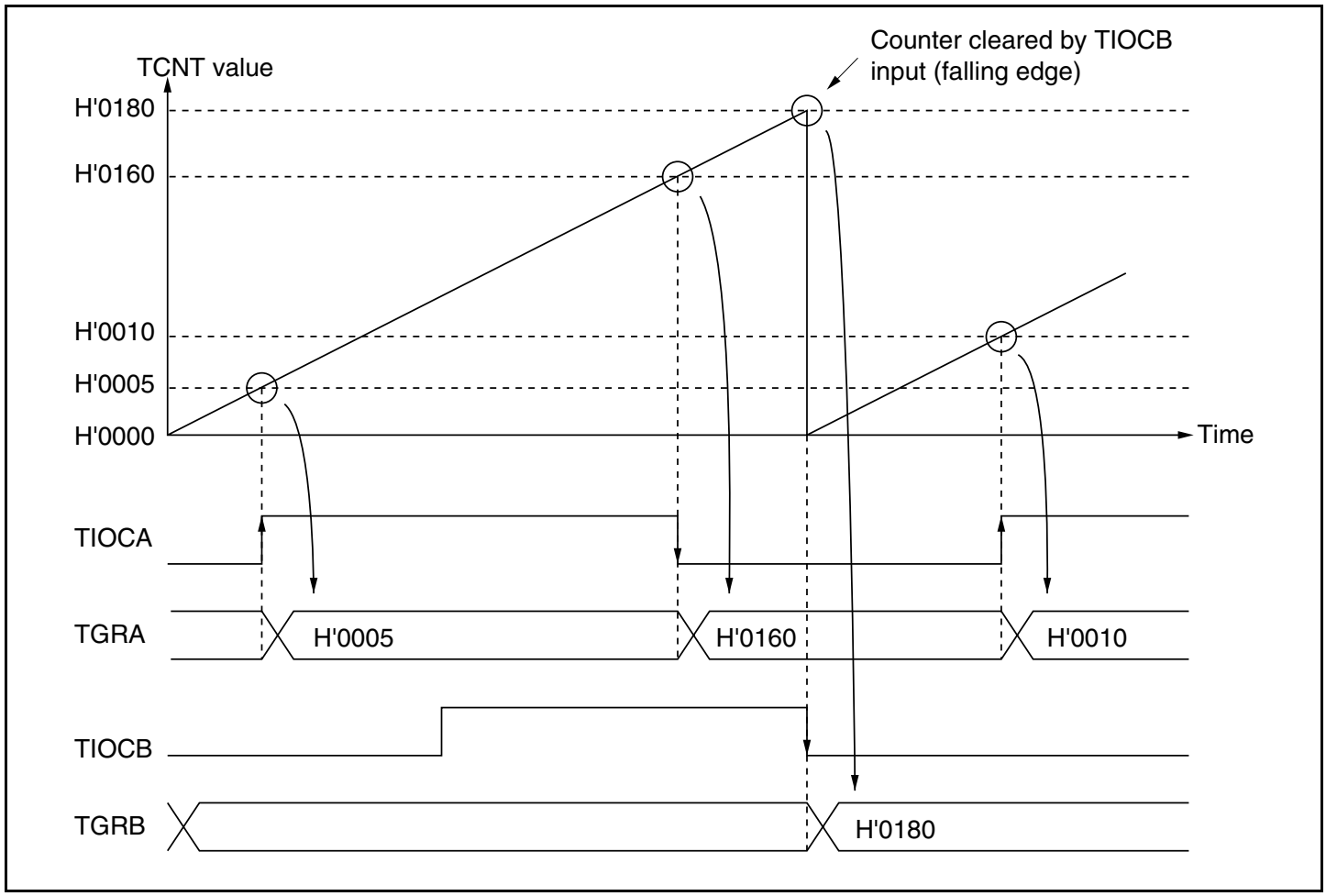


Figure 11.9 Example of Input Capture Operation

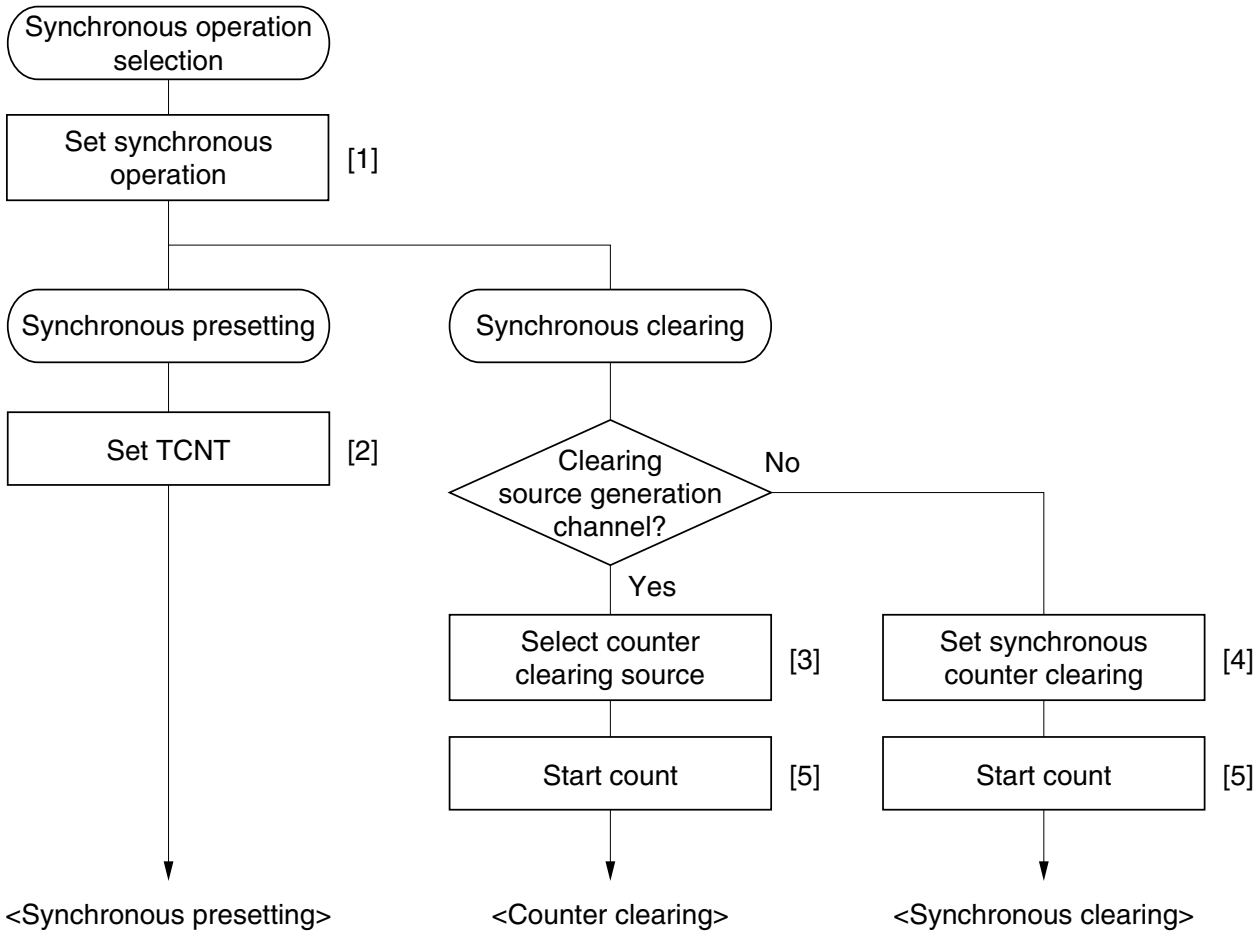
11.4.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be rewritten simultaneously (synchronous presetting). Also, multiple of TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 5 can all be designated for synchronous operation.

Example of Synchronous Operation Setting Procedure: Figure 11.10 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.

[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.

[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.

[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.

[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

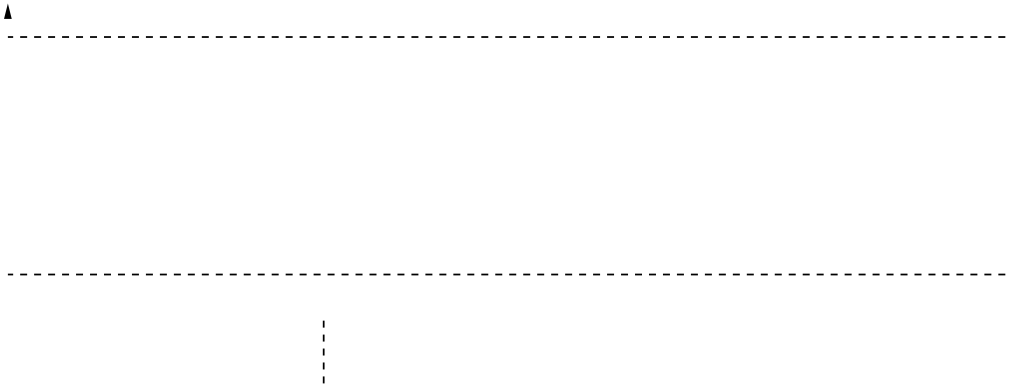
Figure 11.10 Example of Synchronous Operation Setting Procedure

Example of Synchronous Operation: Figure 11.11 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOCA0, TIOCA1, and TIOCA2. At this time, synchronous presetting, and synchronous clearing by TGRB_0 compare match, is performed for channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the PWM cycle.

For details on PWM modes, see section 11.4.5, PWM Modes.



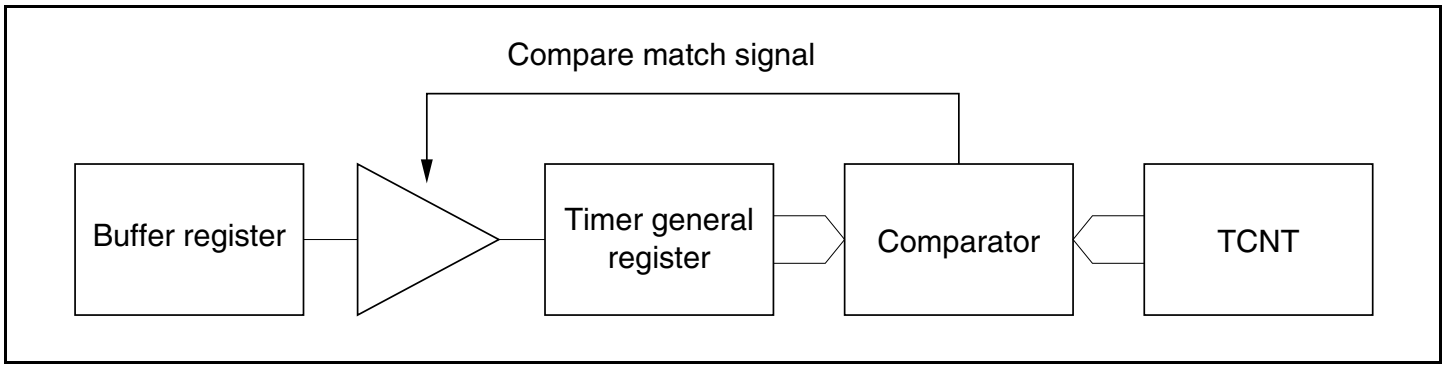


Figure 11.12 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

  This operation is illustrated in figure 11.13.

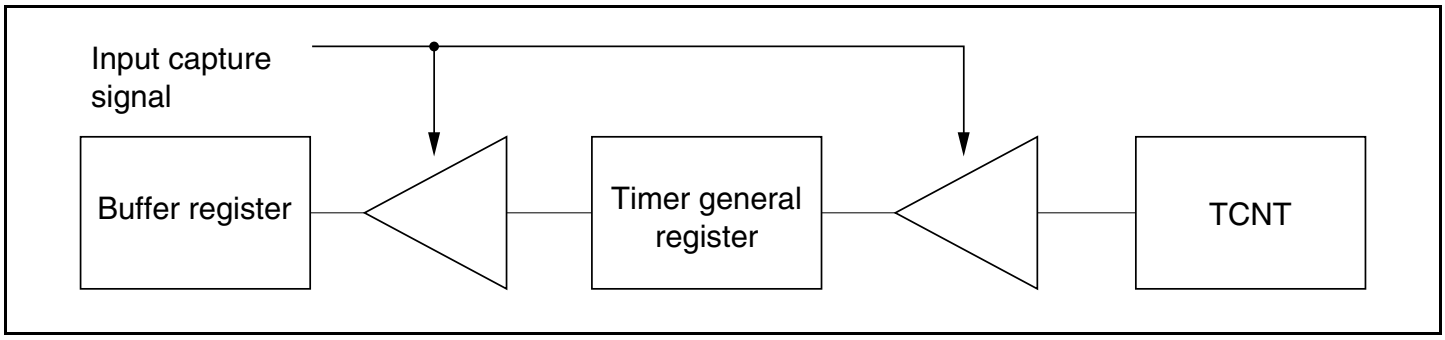


Figure 11.13 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 11.14 shows an example of the buffer operation setting procedure.

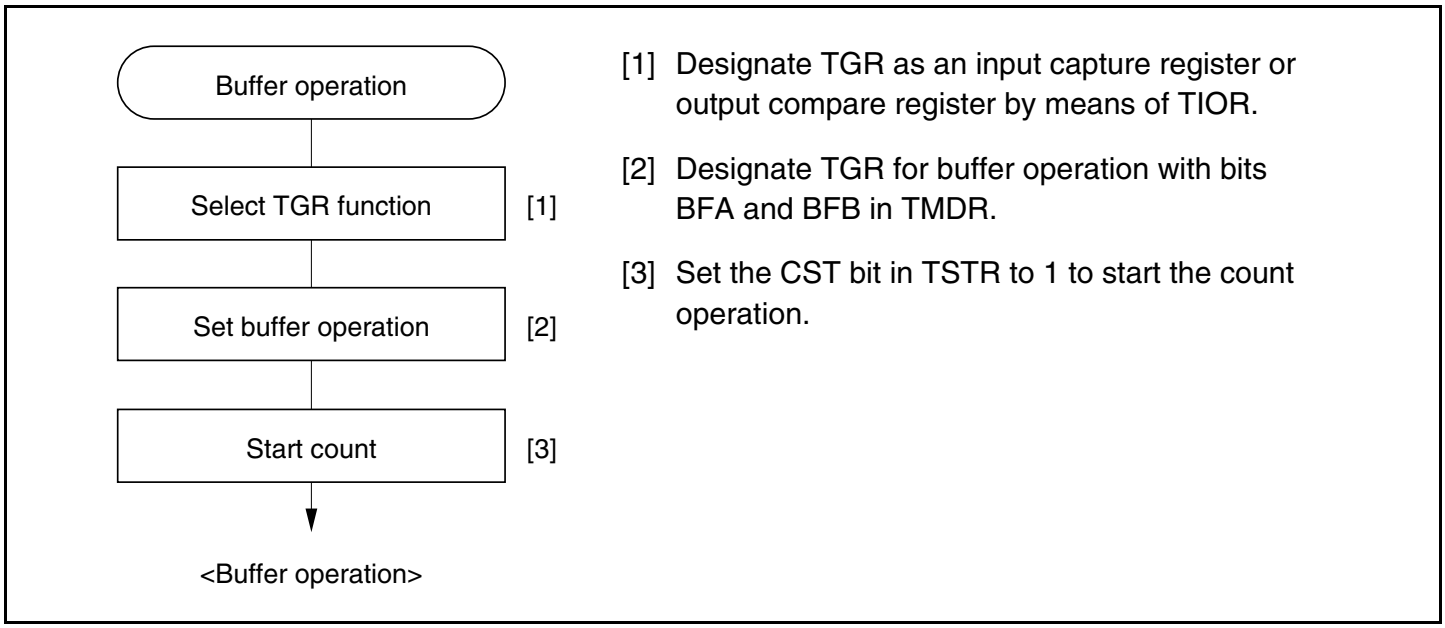


Figure 11.14 Example of Buffer Operation Setting Procedure

Examples of Buffer Operation:

1. When TGR is an output compare register

   Figure 11.15 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B.

   As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

   For details on PWM modes, see section 11.4.5, PWM Modes.

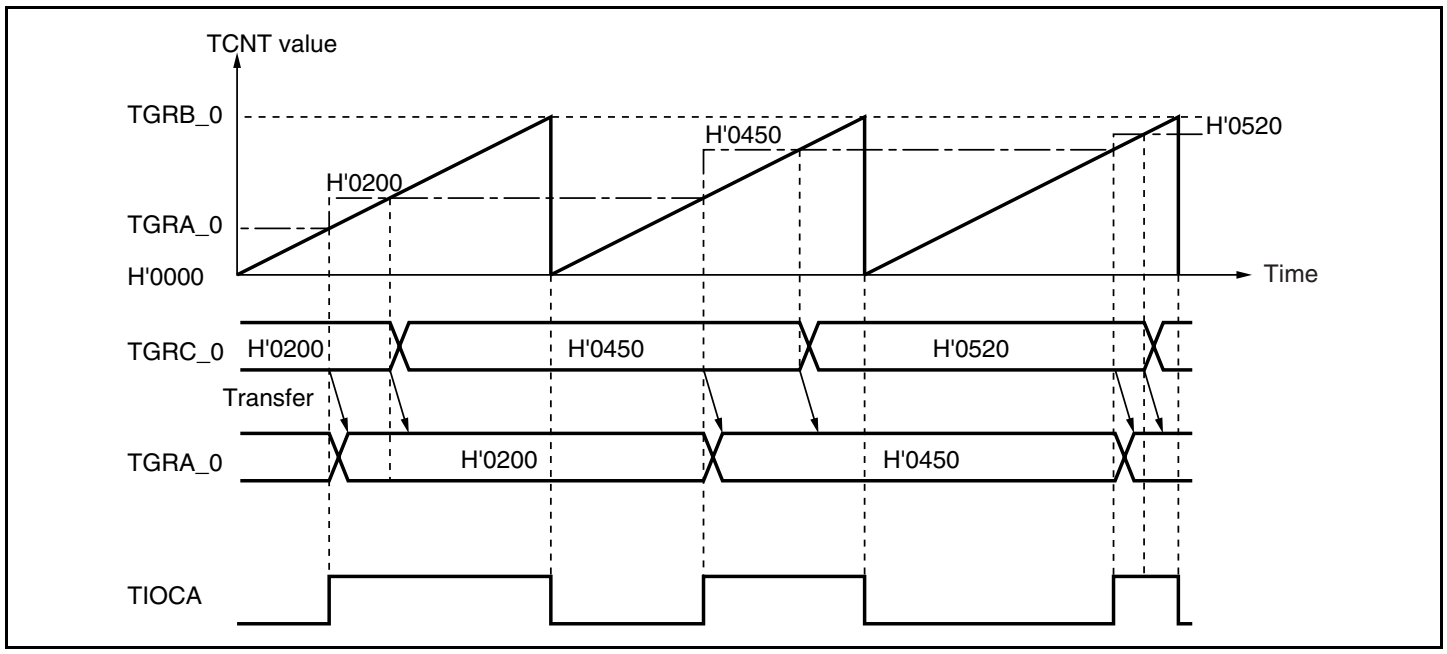


Figure 11.15 Example of Buffer Operation (1)

2. When TGR is an input capture register

   Figure 11.16 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

   Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

   As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

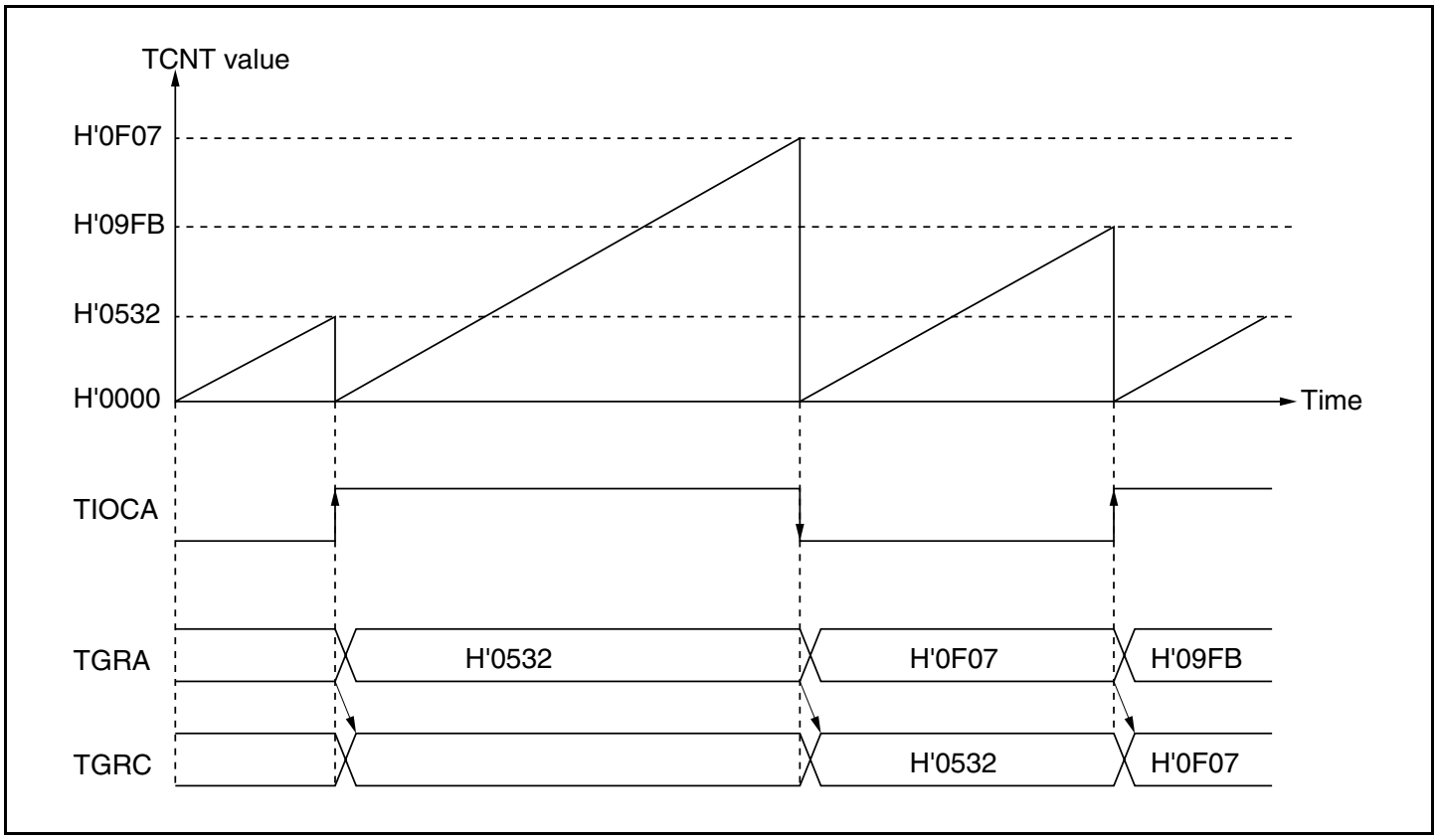


Figure 11.16 Example of Buffer Operation (2)

11.4.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 (channel 4) counter clock at overflow/underflow of TCNT_2 (TCNT_5) as set in bits TPSC2 to TPSC0 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase-counting mode.

Table 11.29 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is invalid and the counter operates independently in phase counting mode.

Table 11.29 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2
Channels 4 and 5	TCNT_4	TCNT_5

Example of Cascaded Operation Setting Procedure: Figure 11.17 shows an example of the setting procedure for cascaded operation.

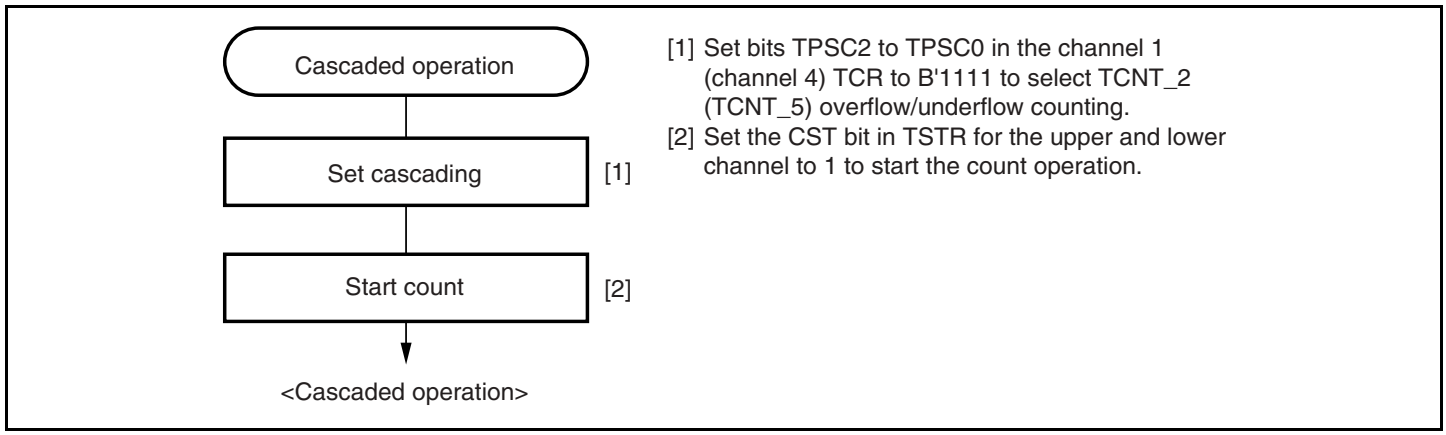


Figure 11.17 Cascaded Operation Setting Procedure

Examples of Cascaded Operation: Figure 11.18 illustrates the operation when counting upon TCNT_2 overflow/underflow has been set for TCNT_1, TGRA_1 and TGRA_2 have been designated as input capture registers, and the TIOC pin rising edge has been selected.

When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TGRA_1, and the lower 16 bits to TGRA_2.

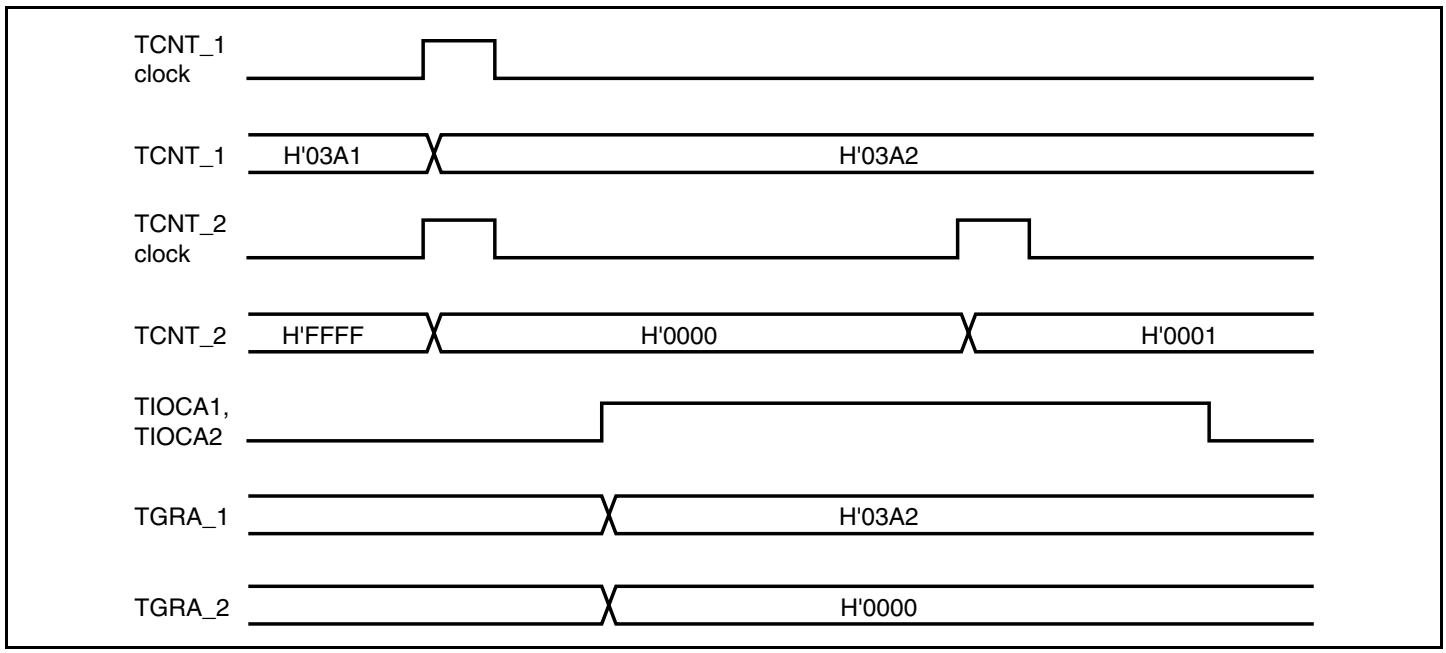


Figure 11.18 Example of Cascaded Operation (1)

Figure 11.19 illustrates the operation when counting upon TCNT_2 overflow/underflow has been set for TCNT_1, and phase counting mode has been designated for channel 2.

TCNT_1 is incremented by TCNT_2 overflow and decremented by TCNT_2 underflow.

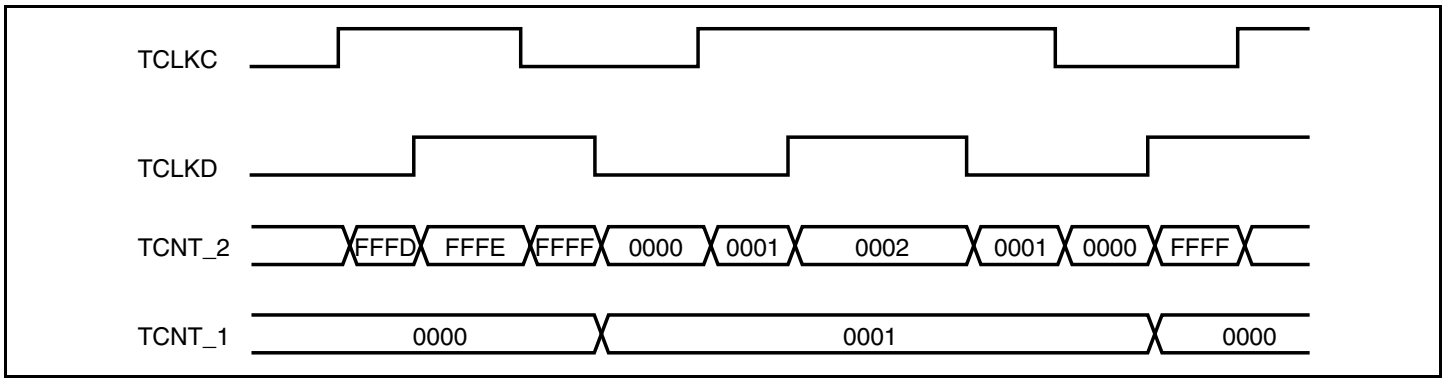


Figure 11.19 Example of Cascaded Operation (2)

11.4.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0, 1, or toggle output can be selected as the output level in response to compare match of each TGR.

Settings of TGR registers can output a PWM waveform in the range of 0–% to 100–% duty cycle.

Designating TGR compare match as the counter clearing source enables the cycle to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The outputs specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR are output from the TIOCA and TIOCC pins at compare matches A and C, respectively. The outputs specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR are output at compare matches B and D, respectively. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.

  In PWM mode 1, a maximum 8-phase PWM output is possible.

- PWM mode 2

  PWM output is generated using one TGR as the cycle register and the others as duty cycle registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a synchronization register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty cycle registers are identical, the output value does not change when a compare match occurs.

  In PWM mode 2, a maximum 15-phase PWM output is possible by combined use with synchronous operation.

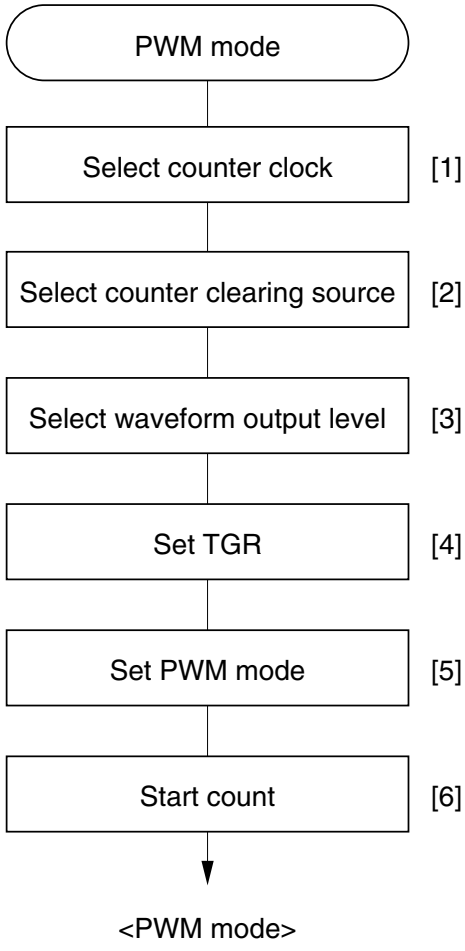
The correspondence between PWM output pins and registers is shown in table 11.30.

Table 11.30 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGRA_0	TIOCA0	TIOCA0
	TGRB_0		TIOCB0
	TGRC_0	TIOCC0	TIOCC0
	TGRD_0		TIOCD0
1	TGRA_1	TIOCA1	TIOCA1
	TGRB_1		TIOCB1
2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2
3	TGRA_3	TIOCA3	TIOCA3
	TGRB_3		TIOCB3
	TGRC_3	TIOCC3	TIOCC3
	TGRD_3		TIOCD3
4	TGRA_4	TIOCA4	TIOCA4
	TGRB_4		TIOCB4
5	TGRA_5	TIOCA5	TIOCA5
	TGRB_5		TIOCB5

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the cycle is set.

Example of PWM Mode Setting Procedure: Figure 11.20 shows an example of the PWM mode setting procedure.



[1] Select the counter clock with bits TPSC2 to TPSC0 in TCR. At the same time, select the input clock edge with bits CKEG1 and CKEG0 in TCR.

[2] Use bits CCLR2 to CCLR0 in TCR to select the TGR to be used as the TCNT clearing source.

[3] Use TIOR to designate the TGR as an output compare register, and select the initial value and output value.

[4] Set the cycle in the TGR selected in [2], and set the duty in the other TGRs.

[5] Select the PWM mode with bits MD3 to MD0 in TMDR.

[6] Set the CST bit in TSTR to 1 to start the count operation.

Figure 11.20 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation: Figure 11.21 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the cycle, and the values set in TGRB registers as the duty cycle.

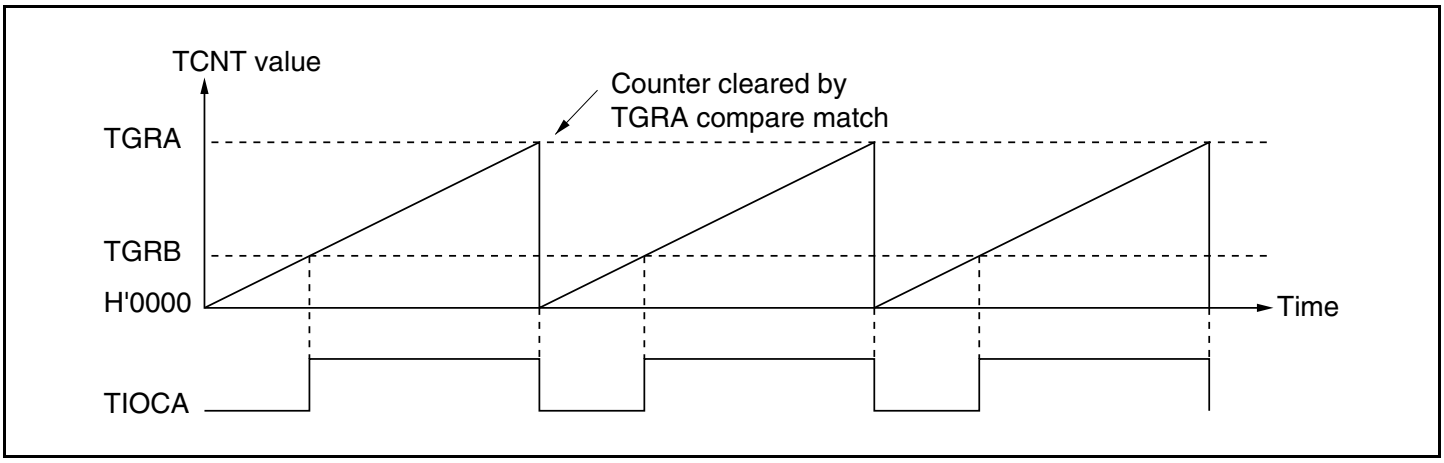


Figure 11.21 Example of PWM Mode Operation (1)

Figure 11.22 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), to output a 5-phase PWM waveform.

In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other TGRs as the duty cycle.

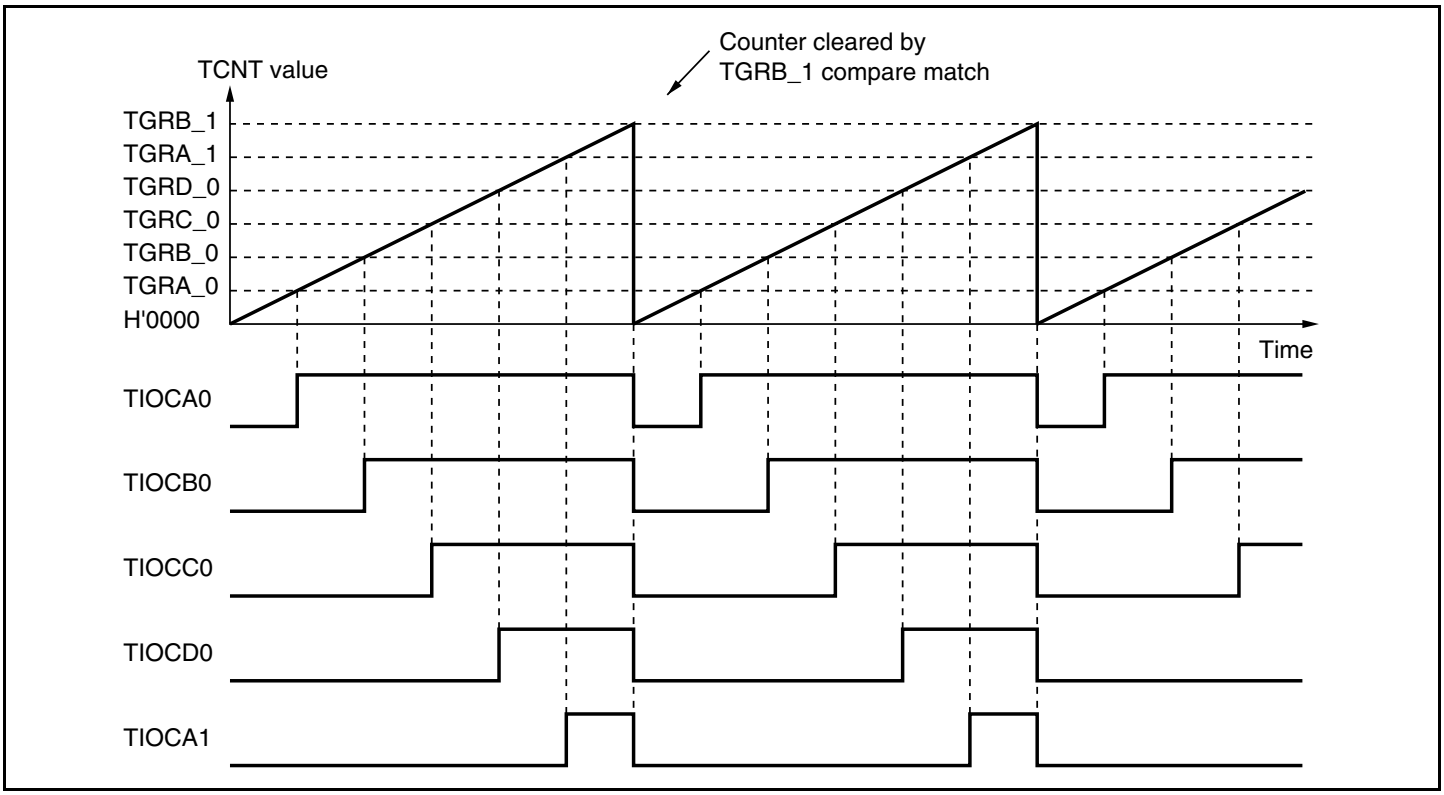


Figure 11.22 Example of PWM Mode Operation (2)

Figure 11.23 shows examples of PWM waveform output with 0% duty cycle and 100% duty cycle in PWM mode.

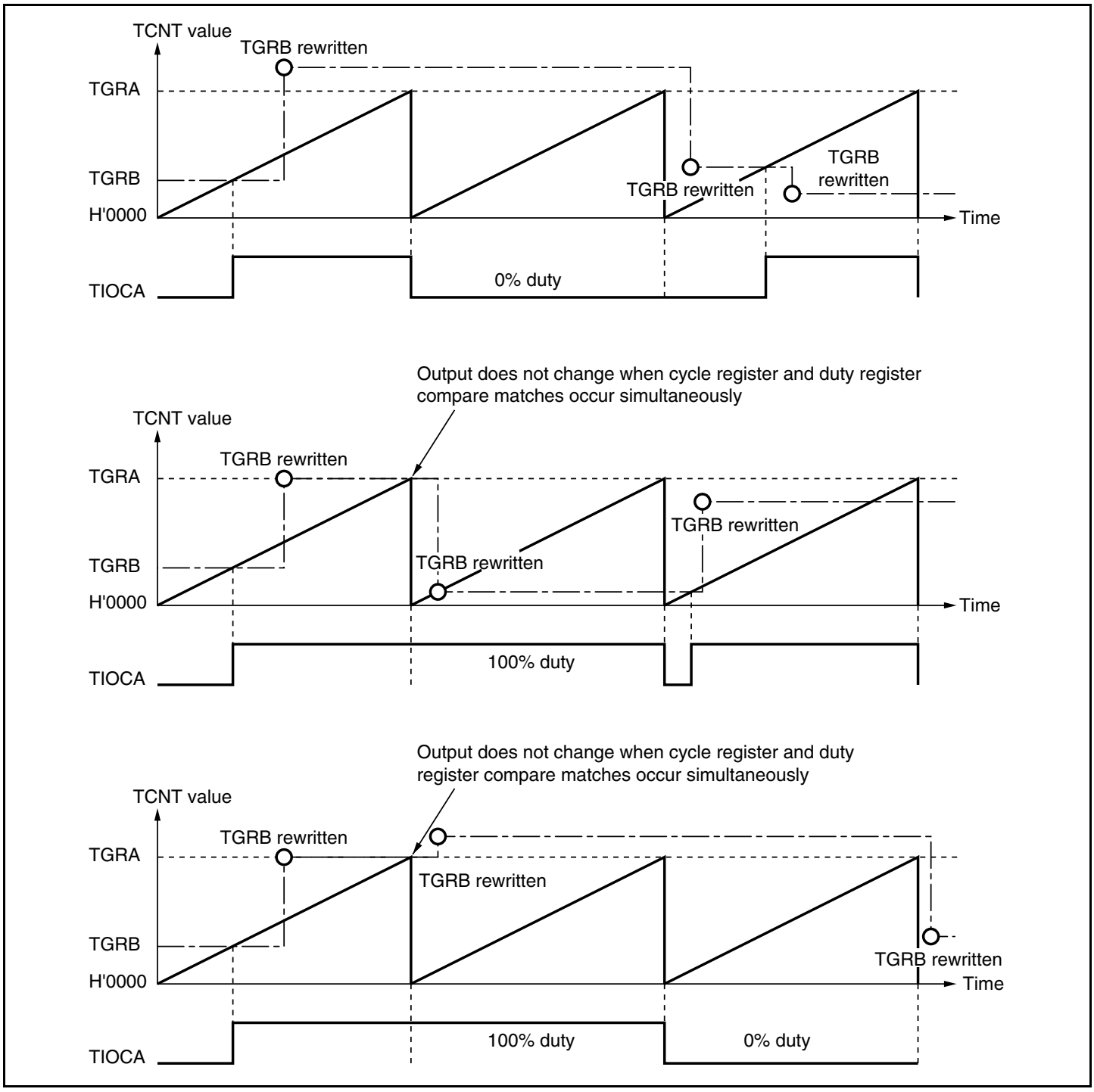


Figure 11.23 Example of PWM Mode Operation (3)

11.4.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1, 2, 4, and 5.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, the functions of bits CCLR1 and CCLR0 in TCR, and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.

This can be used for two-phase encoder pulse input.

When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when underflow occurs while TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.

Table 11.31 shows the correspondence between external clock pins and channels.

Table 11.31 Clock Input Pins in Phase Counting Mode

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure: Figure 11.24 shows an example of the phase counting mode setting procedure.

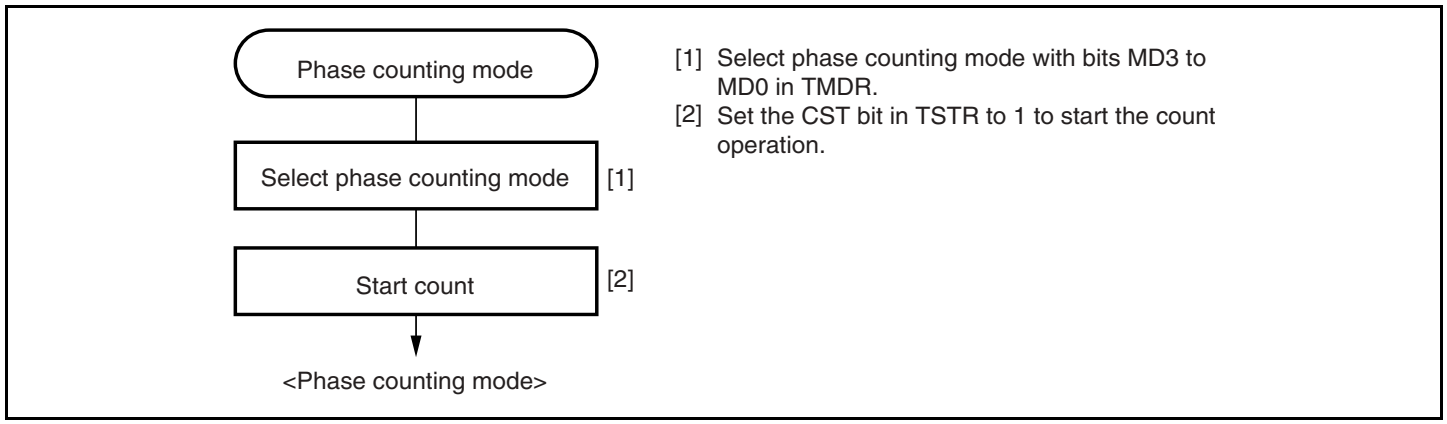


Figure 11.24 Example of Phase Counting Mode Setting Procedure

Examples of Phase Counting Mode Operation: In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

1. Phase counting mode 1

   Figure 11.25 shows an example of phase counting mode 1 operation, and table 11.32 summarizes the TCNT up/down-count conditions.

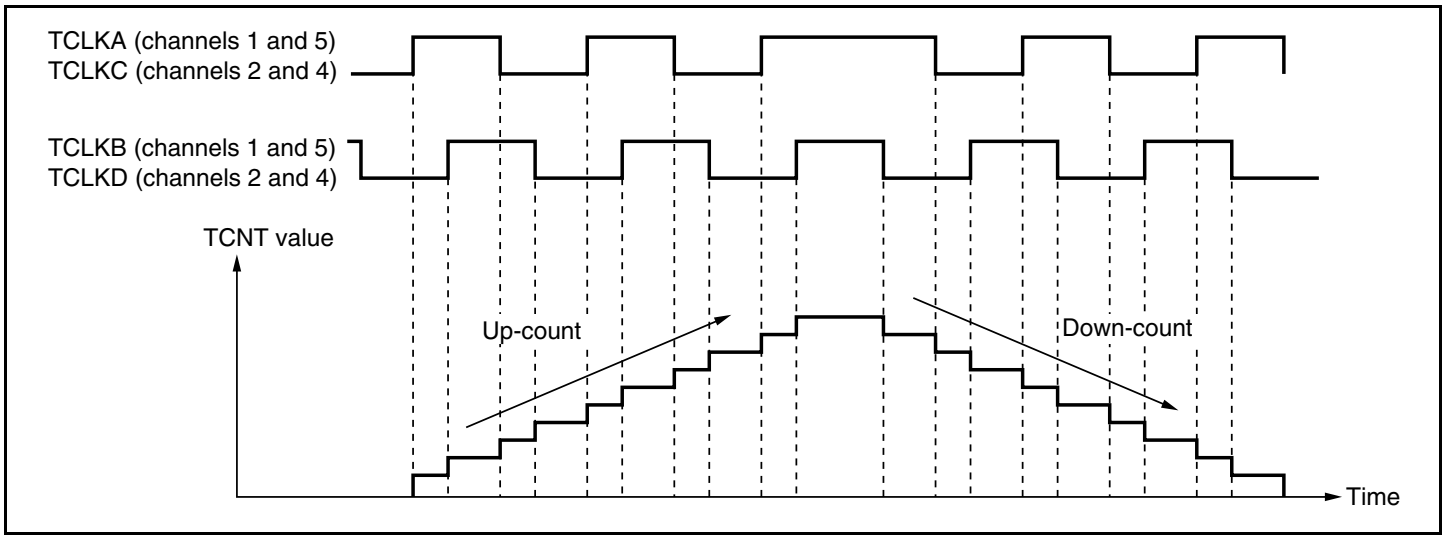


Figure 11.25 Example of Phase Counting Mode 1 Operation

Table 11.32 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

Legend

↑ : Rising edge

↓ : Falling edge

2. Phase counting mode 2

Figure 11.26 shows an example of phase counting mode 2 operation, and table 11.33 summarizes the TCNT up/down-count conditions.

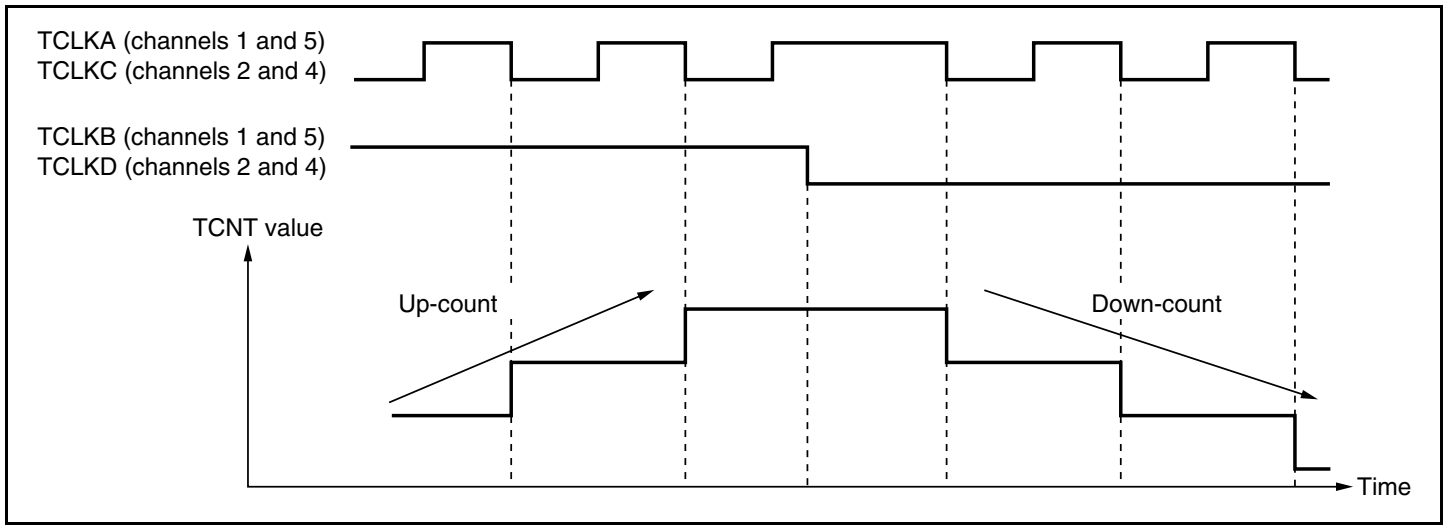


Figure 11.26 Example of Phase Counting Mode 2 Operation

Table 11.33 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	⌐↑	Don't care
Low level	⌐↓	Don't care
⌐↑	Low level	Don't care
⌐↓	High level	Up-count
High level	⌐↓	Don't care
Low level	⌐↑	Don't care
⌐↑	High level	Don't care
⌐↓	Low level	Down-count

Legend

⌐↑ : Rising edge

⌐↓ : Falling edge

3. Phase counting mode 3

Figure 11.27 shows an example of phase counting mode 3 operation, and table 11.34 summarizes the TCNT up/down-count conditions.

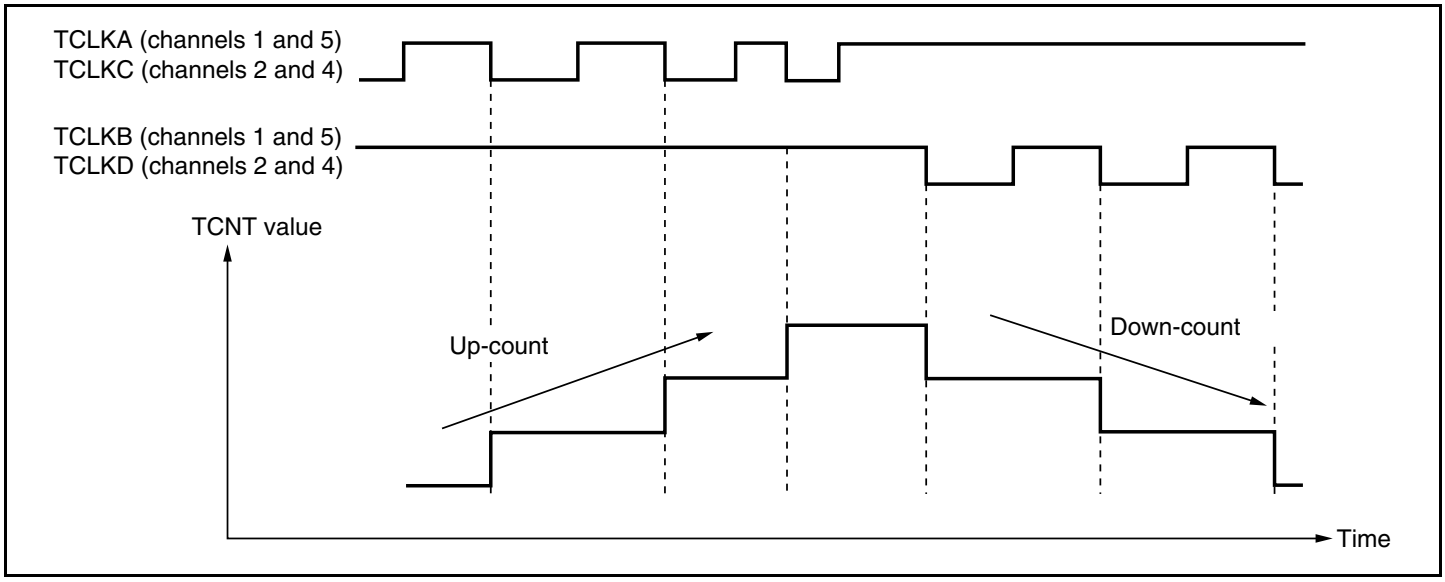


Figure 11.27 Example of Phase Counting Mode 3 Operation

Table 11.34 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	⌐↑	Don't care
Low level	⌐↓	Don't care
⌐↑	Low level	Don't care
⌐↓	High level	Up-count
High level	⌐↓	Down-count
Low level	⌐↑	Don't care
⌐↑	High level	Don't care
⌐↓	Low level	Don't care

Legend

⌐↑ : Rising edge

⌐↓ : Falling edge

4. Phase counting mode 4

   Figure 11.28 shows an example of phase counting mode 4 operation, and table 11.35 summarizes the TCNT up/down-count conditions.

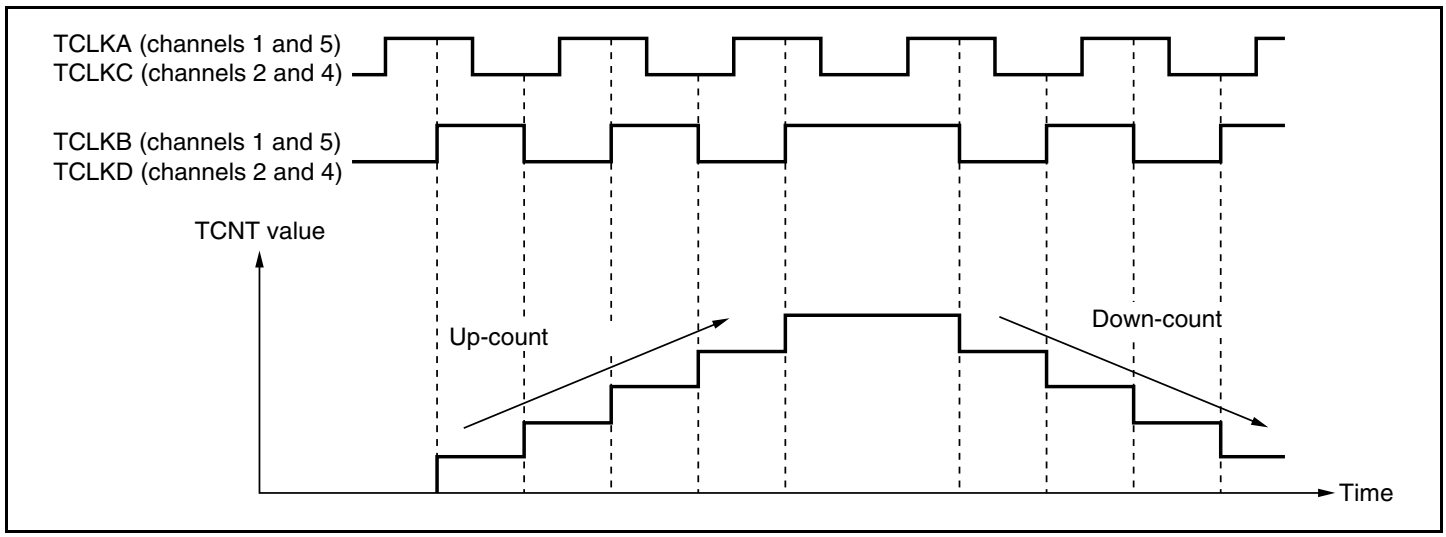


Figure 11.28 Example of Phase Counting Mode 4 Operation

Table 11.35 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count

Phase Counting Mode Application Example: Figure 11.29 shows an example in which phase counting mode is designated for channel 1, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect the position or speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to TCLKA and TCLKB.

Channel 0 operates with TCNT counter clearing by TGRC_0 compare match; TGRA_0 and TGRC_0 are used for the compare match function, and are set with the speed control cycle and position control cycle. TGRB_0 is used for input capture, with TGRB_0 and TGRD_0 operating in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input capture source, and detection of the pulse width of 2-phase encoder 4-multiplication pulses is performed.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, channel 0 TGRA_0 and TGRC_0 compare matches are selected as the input capture source, and the up/down-counter values for the control cycles are stored.

This procedure enables accurate position/speed detection to be achieved.

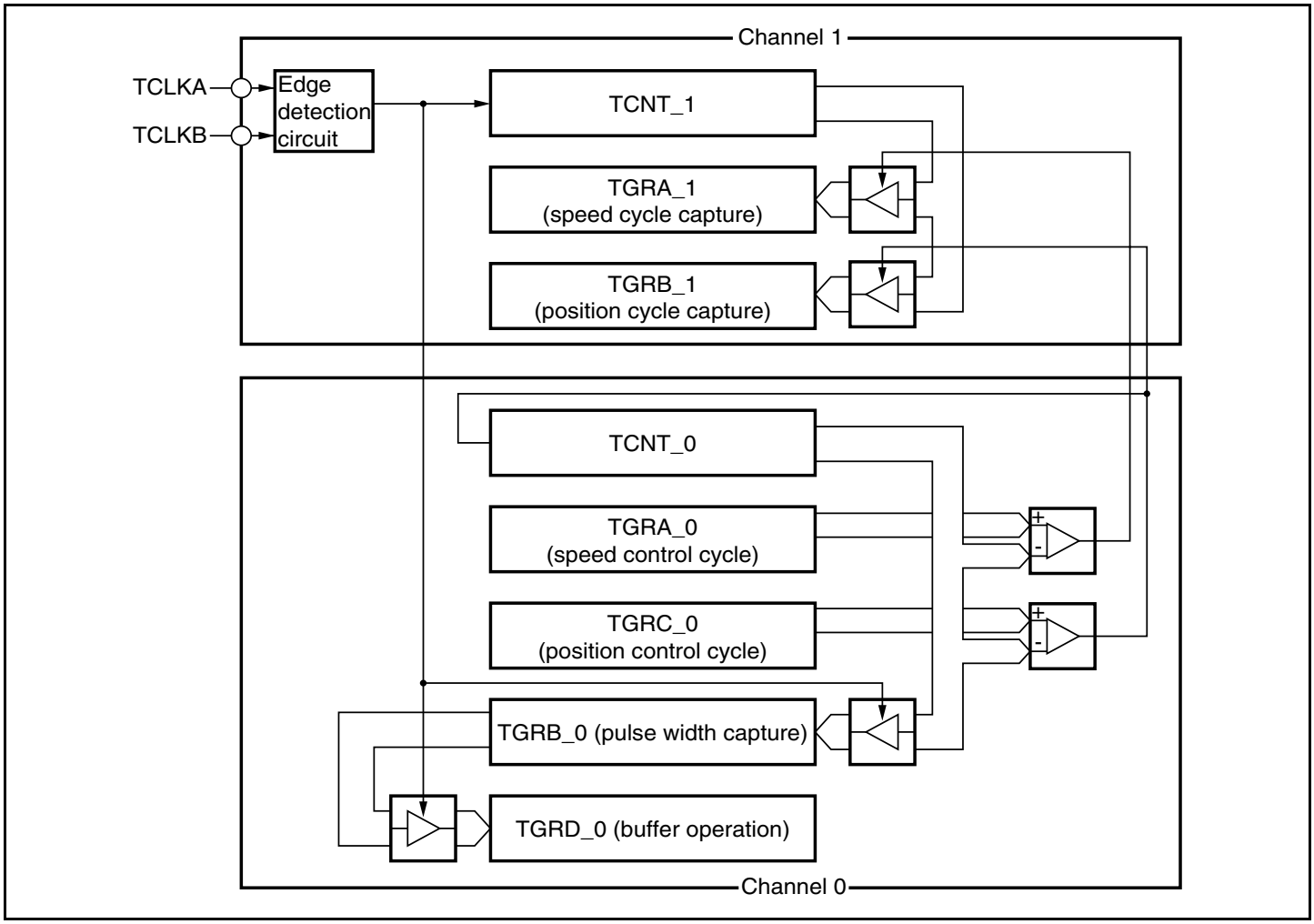


Figure 11.29 Phase Counting Mode Application Example

11.5 Interrupt Sources

There are three kinds of TPU interrupt source: TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disable bit, allowing generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, but the priority order within a channel is fixed. For details, see section 5, Interrupt Controller.

Table 11.36 lists the TPU interrupt sources.

Table 11.36 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
0	TGI0A	TGRA_0 input capture/compare match	TGFA_0	Possible	Possible
	TGI0B	TGRB_0 input capture/compare match	TGFB_0	Possible	Not possible
	TGI0C	TGRC_0 input capture/compare match	TGFC_0	Possible	Not possible
	TGI0D	TGRD_0 input capture/compare match	TGFD_0	Possible	Not possible
	TGI0V	TCNT_0 overflow	TCFV_0	Not possible	Not possible
1	TGI1A	TGRA_1 input capture/compare match	TGFA_1	Possible	Possible
	TGI1B	TGRB_1 input capture/compare match	TGFB_1	Possible	Not possible
	TCI1V	TCNT_1 overflow	TCFV_1	Not possible	Not possible
	TCI1U	TCNT_1 underflow	TCFU_1	Not possible	Not possible
2	TGI2A	TGRA_2 input capture/compare match	TGFA_2	Possible	Possible
	TGI2B	TGRB_2 input capture/compare match	TGFB_2	Possible	Not possible
	TCI2V	TCNT_2 overflow	TCFV_2	Not possible	Not possible
	TCI2U	TCNT_2 underflow	TCFU_2	Not possible	Not possible
3	TGI3A	TGRA_3 input capture/compare match	TGFA_3	Possible	Possible
	TGI3B	TGRB_3 input capture/compare match	TGFB_3	Possible	Not possible
	TGI3C	TGRC_3 input capture/compare match	TGFC_3	Possible	Not possible
	TGI3D	TGRD_3 input capture/compare match	TGFD_3	Possible	Not possible
	TCI3V	TCNT_3 overflow	TCFV_3	Not possible	Not possible
4	TGI4A	TGRA_4 input capture/compare match	TGFA_4	Possible	Possible
	TGI4B	TGRB_4 input capture/compare match	TGFB_4	Possible	Not possible
	TCI4V	TCNT_4 overflow	TCFV_4	Not possible	Not possible
	TCI4U	TCNT_4 underflow	TCFU_4	Not possible	Not possible
5	TGI5A	TGRA_5 input capture/compare match	TGFA_5	Possible	Possible
	TGI5B	TGRB_5 input capture/compare match	TGFB_5	Possible	Not possible
	TCI5V	TCNT_5 overflow	TCFV_5	Not possible	Not possible
	TCI5U	TCNT_5 underflow	TCFU_5	Not possible	Not possible

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Input Capture/Compare Match Interrupt: An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has 16 input capture/compare match interrupts, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

Overflow Interrupt: An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The TPU has six overflow interrupts, one for each channel.

Underflow Interrupt: An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The TPU has four underflow interrupts, one each for channels 1, 2, 4, and 5.

11.6 DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt for a channel. For details, see section 9, Data Transfer Controller.

A total of 16 TPU input capture/compare match interrupts can be used as DTC activation sources, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

11.7 DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt for a channel. For details, see section 7, DMA Controller.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as DMAC activation sources, one for each channel.

11.8 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a channel.

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

11.9 Operation Timing

11.9.1 Input/Output Timing

TCNT Count Timing: Figure 11.30 shows TCNT count timing in internal clock operation, and figure 11.31 shows TCNT count timing in external clock operation.

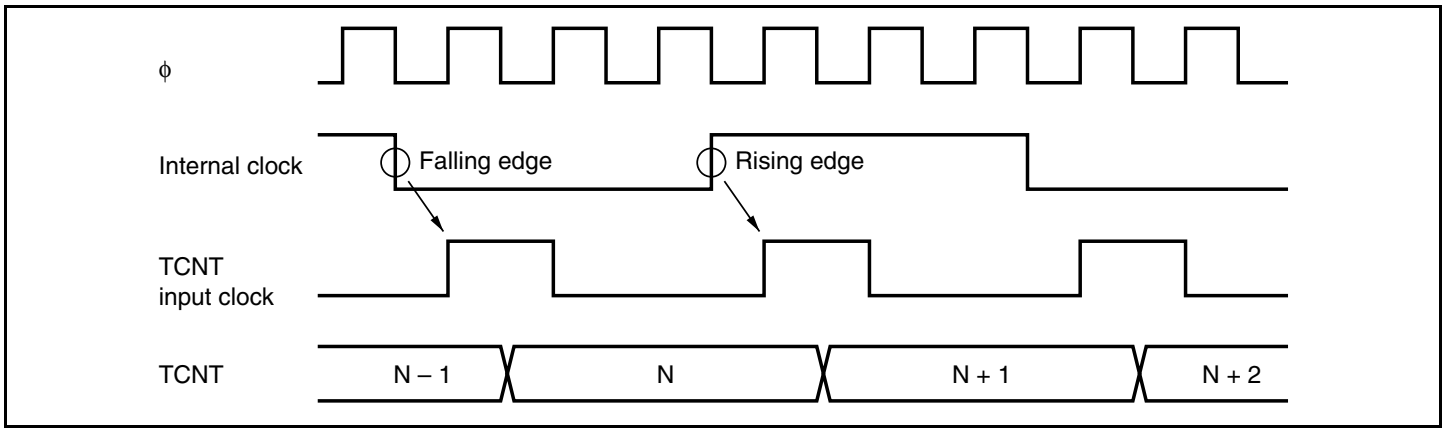


Figure 11.30 Count Timing in Internal Clock Operation

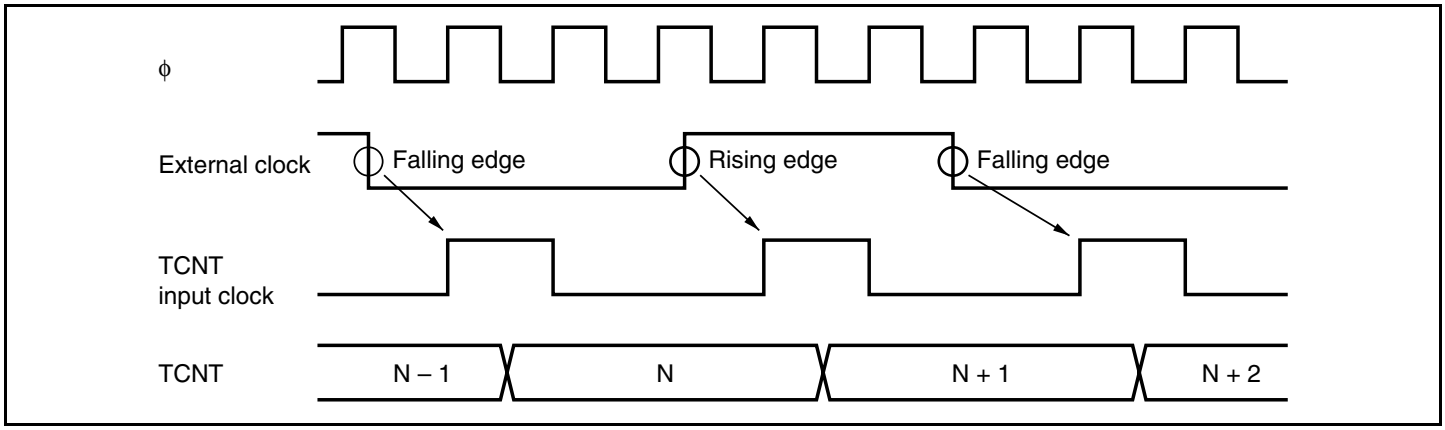


Figure 11.31 Count Timing in External Clock Operation

Output Compare Output Timing: A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin. After a match between TCNT and TGR, the compare match signal is not generated until the (TIOC pin) TCNT input clock is generated.

Figure 11.32 shows output compare output timing.

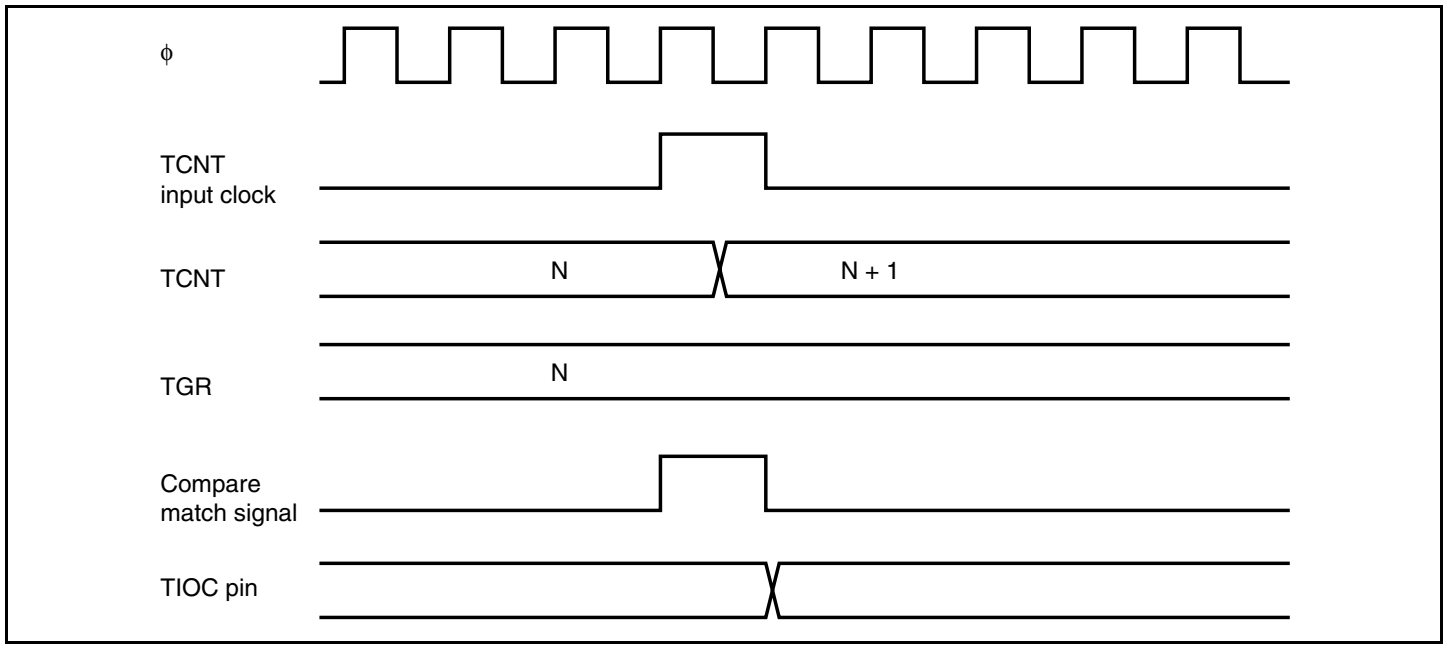


Figure 11.32 Output Compare Output Timing

Input Capture Signal Timing: Figure 11.33 shows input capture signal timing.

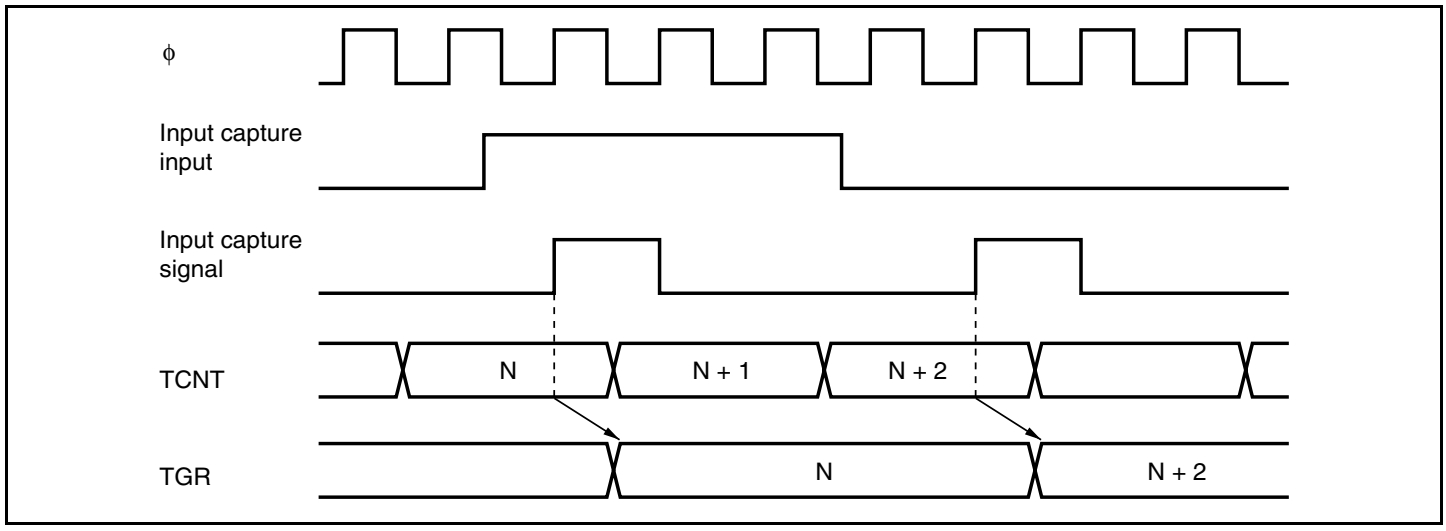


Figure 11.33 Input Capture Input Signal Timing

Timing for Counter Clearing by Compare Match/Input Capture: Figure 11.34 shows the timing when counter clearing by compare match occurrence is specified, and figure 11.35 shows the timing when counter clearing by input capture occurrence is specified.

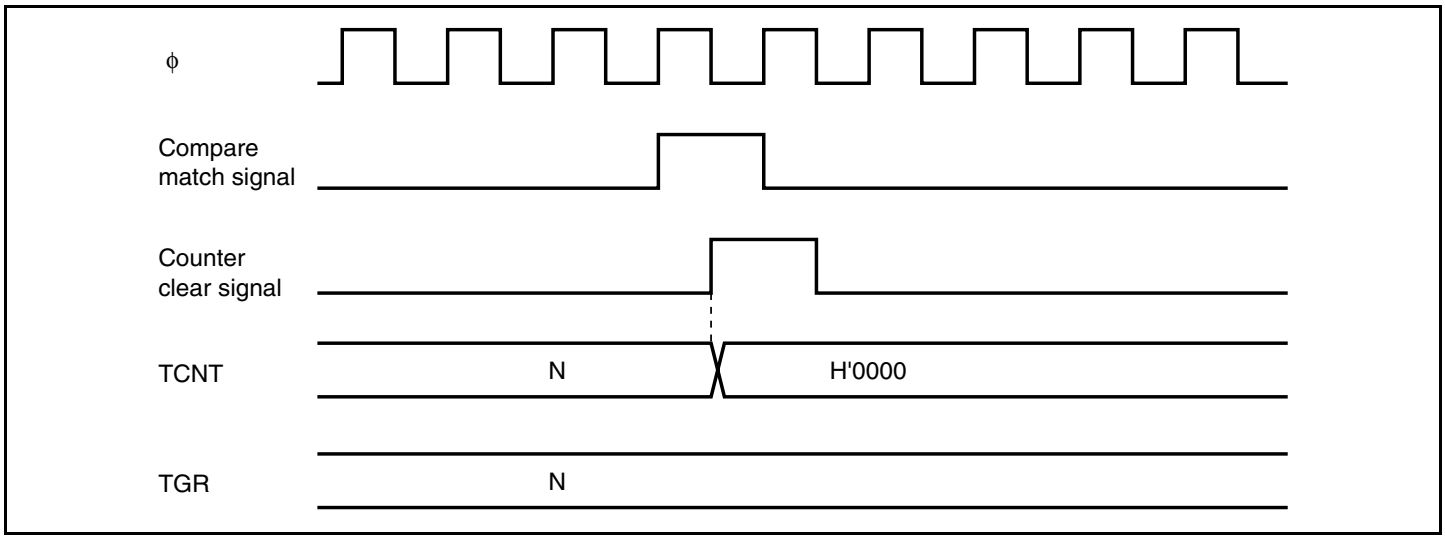


Figure 11.34 Counter Clear Timing (Compare Match)

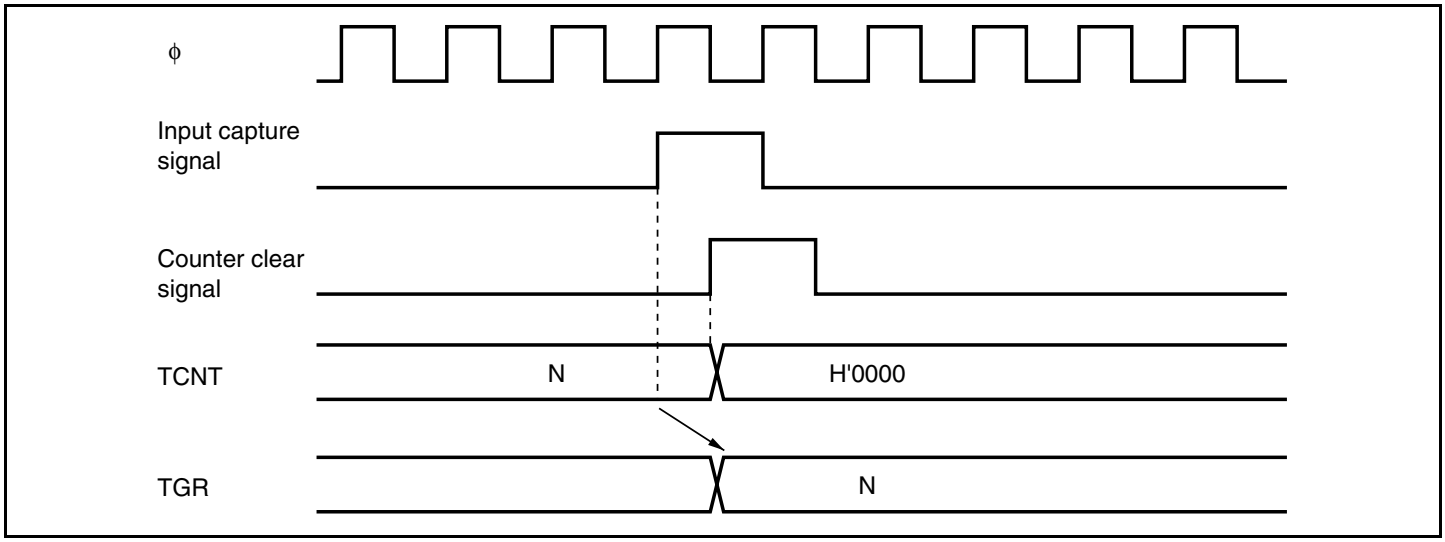


Figure 11.35 Counter Clear Timing (Input Capture)

Buffer Operation Timing: Figures 11.36 and 11.37 show the timings in buffer operation.

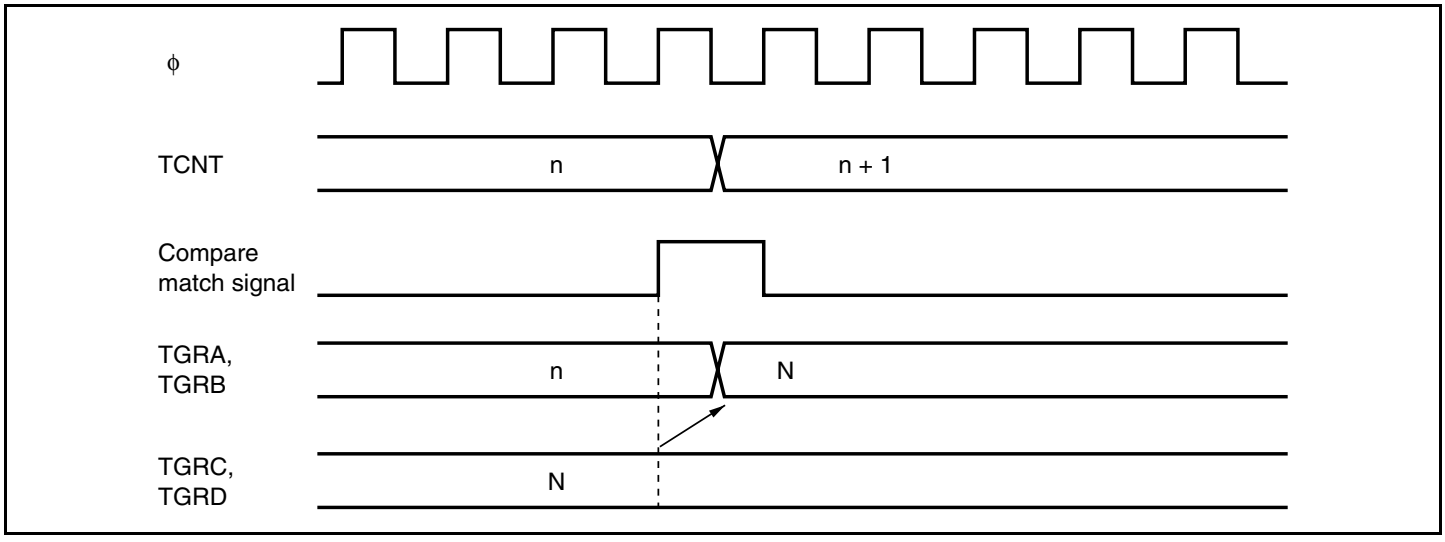


Figure 11.36 Buffer Operation Timing (Compare Match)

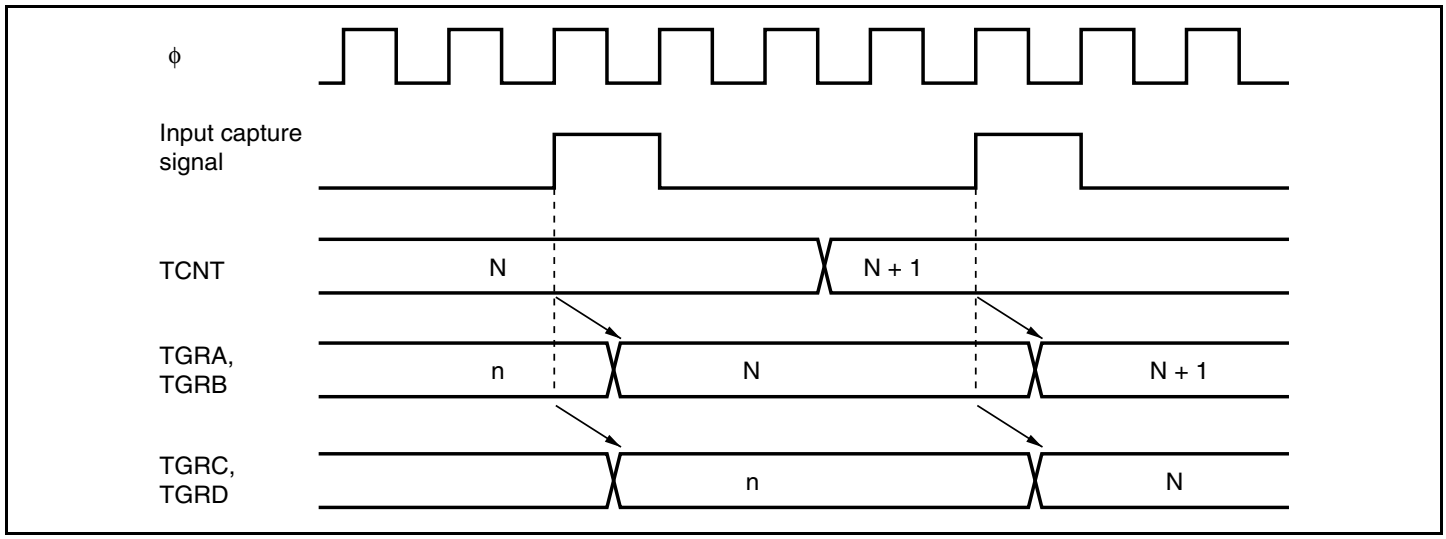


Figure 11.37 Buffer Operation Timing (Input Capture)

11.9.2 Interrupt Signal Timing

TGF Flag Setting Timing in Case of Compare Match: Figure 11.38 shows the timing for setting of the TGF flag in TSR by compare match occurrence, and the TGI interrupt request signal timing.

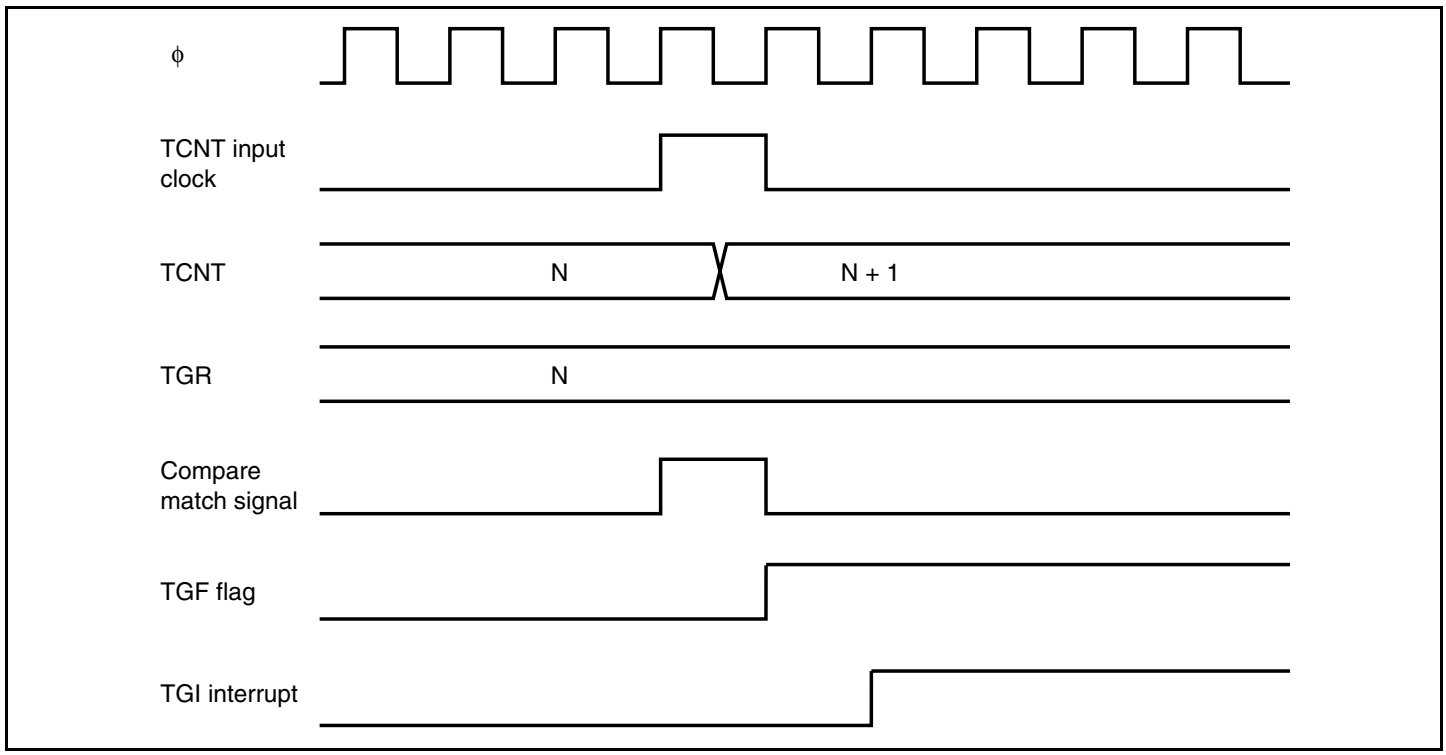


Figure 11.38 TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture: Figure 11.39 shows the timing for setting of the TGF flag in TSR by input capture occurrence, and the TGI interrupt request signal timing.

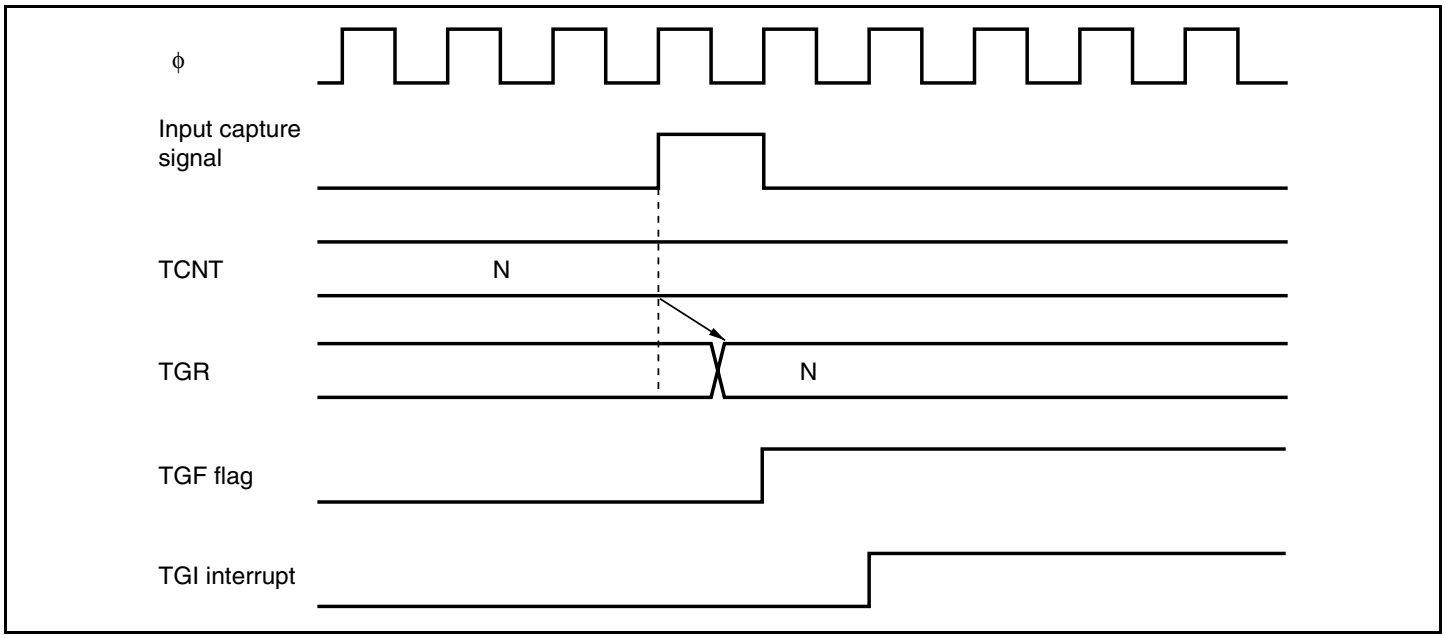


Figure 11.39 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing: Figure 11.40 shows the timing for setting of the TCFV flag in TSR by overflow occurrence, and the TCIV interrupt request signal timing.

Figure 11.41 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and the TCIU interrupt request signal timing.

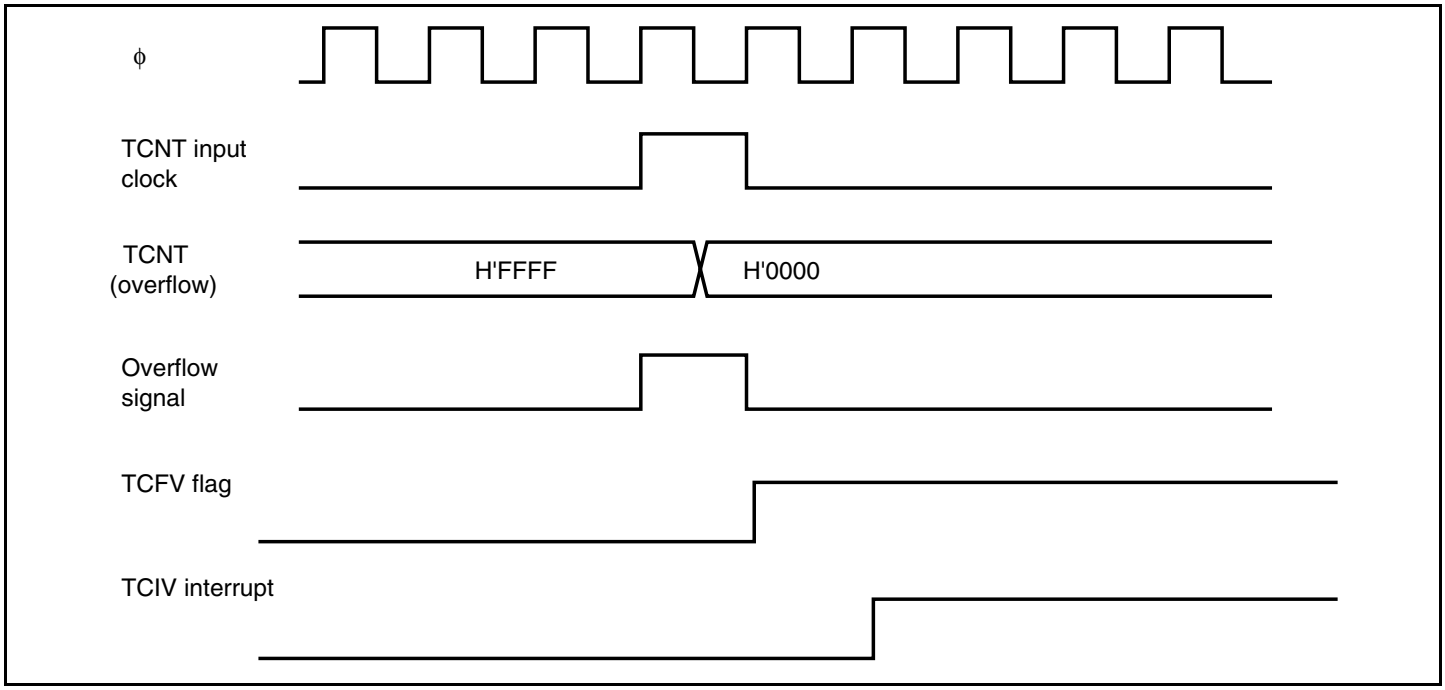


Figure 11.40 TCIV Interrupt Setting Timing

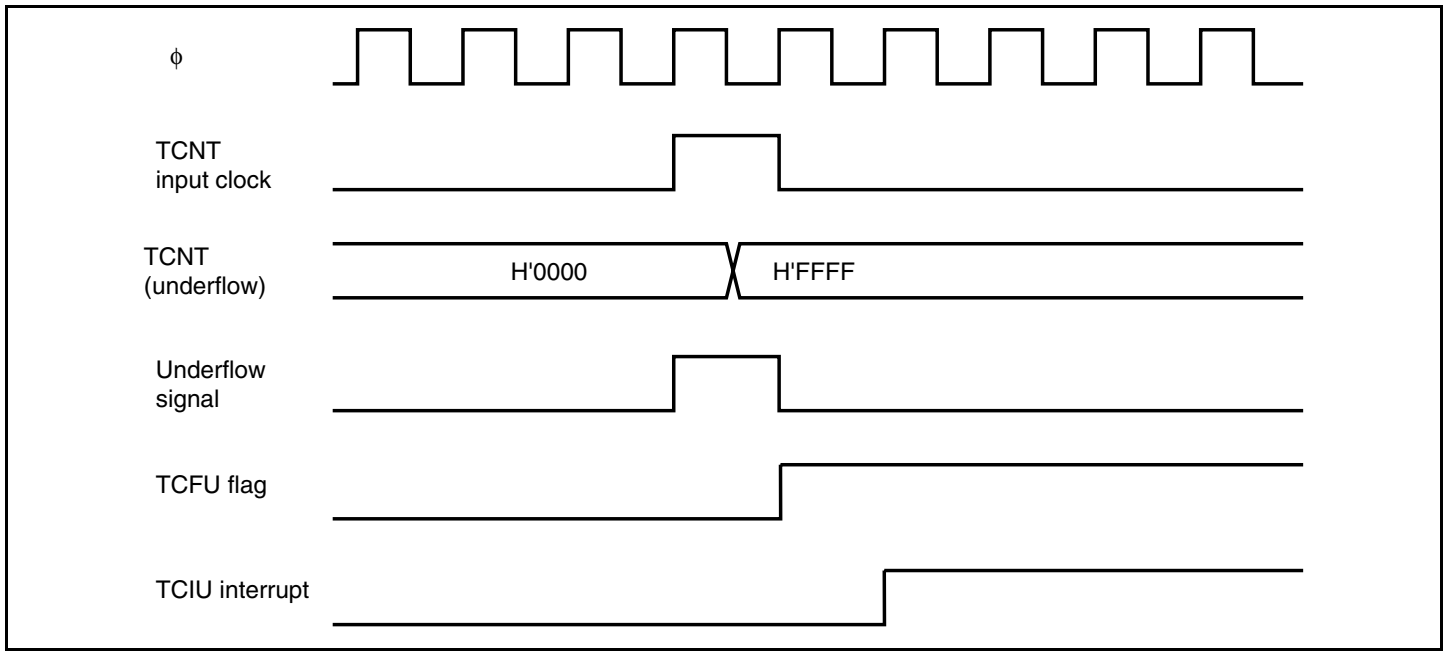


Figure 11.41 TCIU Interrupt Setting Timing

Status Flag Clearing Timing: After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DTC or DMAC is activated, the flag is cleared automatically. Figure 11.42 shows the timing for status flag clearing by the CPU, and figure 11.43 shows the timing for status flag clearing by the DTC or DMAC.

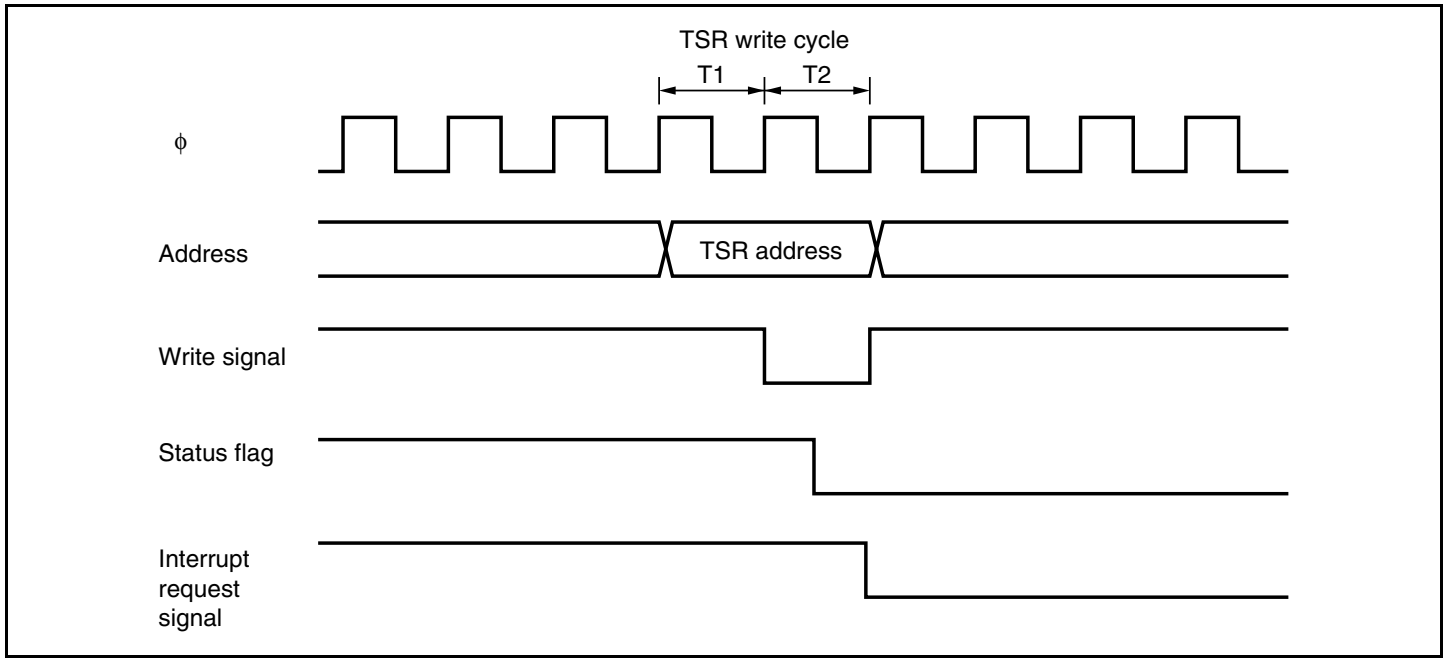


Figure 11.42 Timing for Status Flag Clearing by CPU

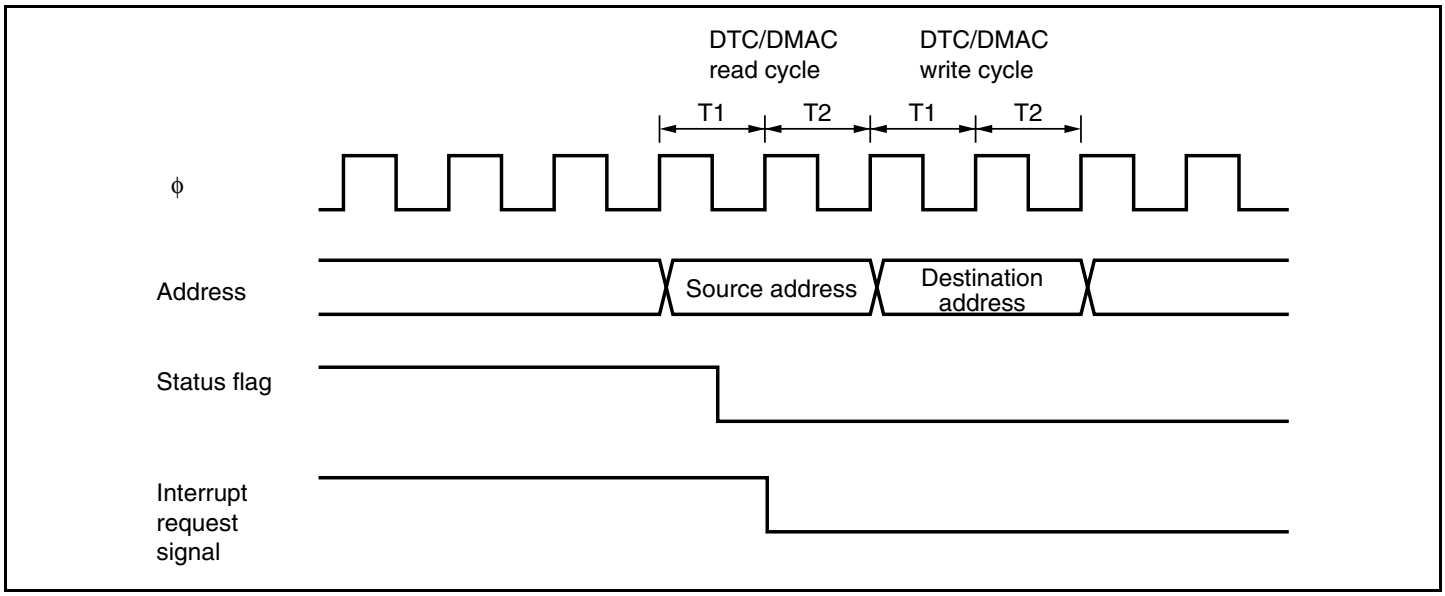


Figure 11.43 Timing for Status Flag Clearing by DTC/DMAC Activation

11.10 Usage Notes

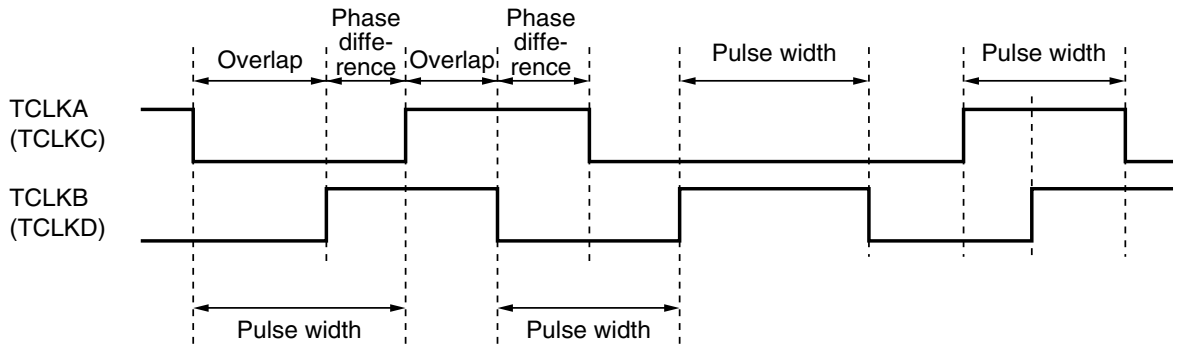
11.10.1 Module Stop Mode Setting

TPU operation can be disabled or enabled using the module stop control register. The initial setting is for TPU operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

11.10.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 11.44 shows the input clock conditions in phase counting mode.



Notes: Phase difference and overlap : 1.5 states or more
Pulse width : 2.5 states or more

Figure 11.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

11.10.3 Caution on Cycle Setting

When counter clearing by compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{\phi}{(N + 1)}$$

Where f: Counter frequency
ø: Operating frequency
N: TGR set value

11.10.4 Contention between TCNT Write and Clear Operations

If the counter clearing signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed. Figure 11.45 shows the timing in this case.

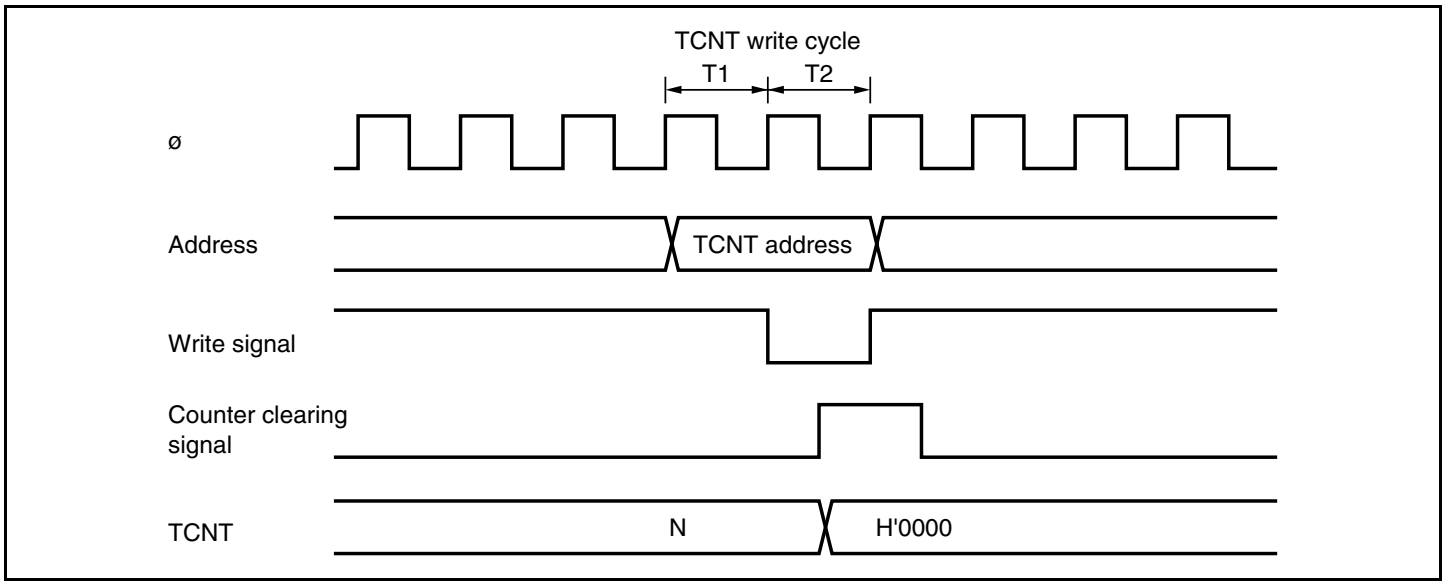


Figure 11.45 Contention between TCNT Write and Clear Operations

11.10.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented. Figure 11.46 shows the timing in this case.

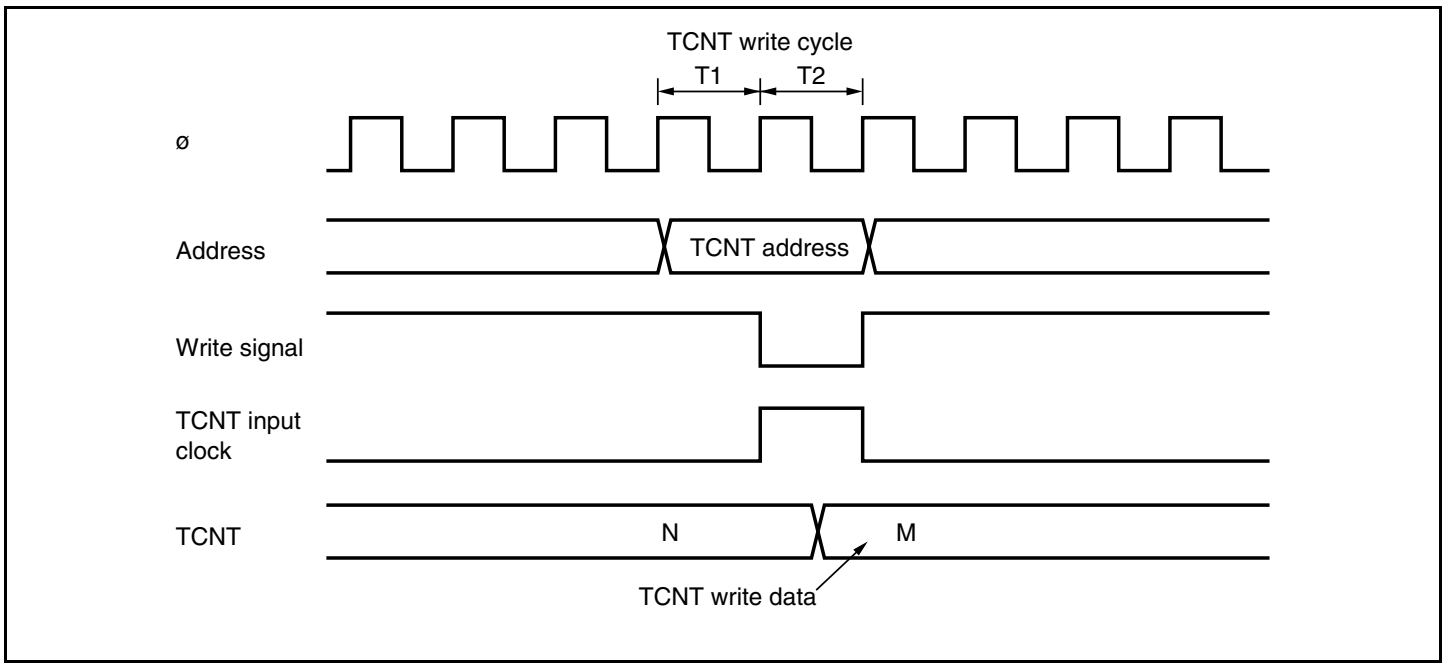


Figure 11.46 Contention between TCNT Write and Increment Operations

11.10.6 Contention between TGR Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write takes precedence and the compare match signal is disabled. A compare match also does not occur when the same value as before is written.

Figure 11.47 shows the timing in this case.

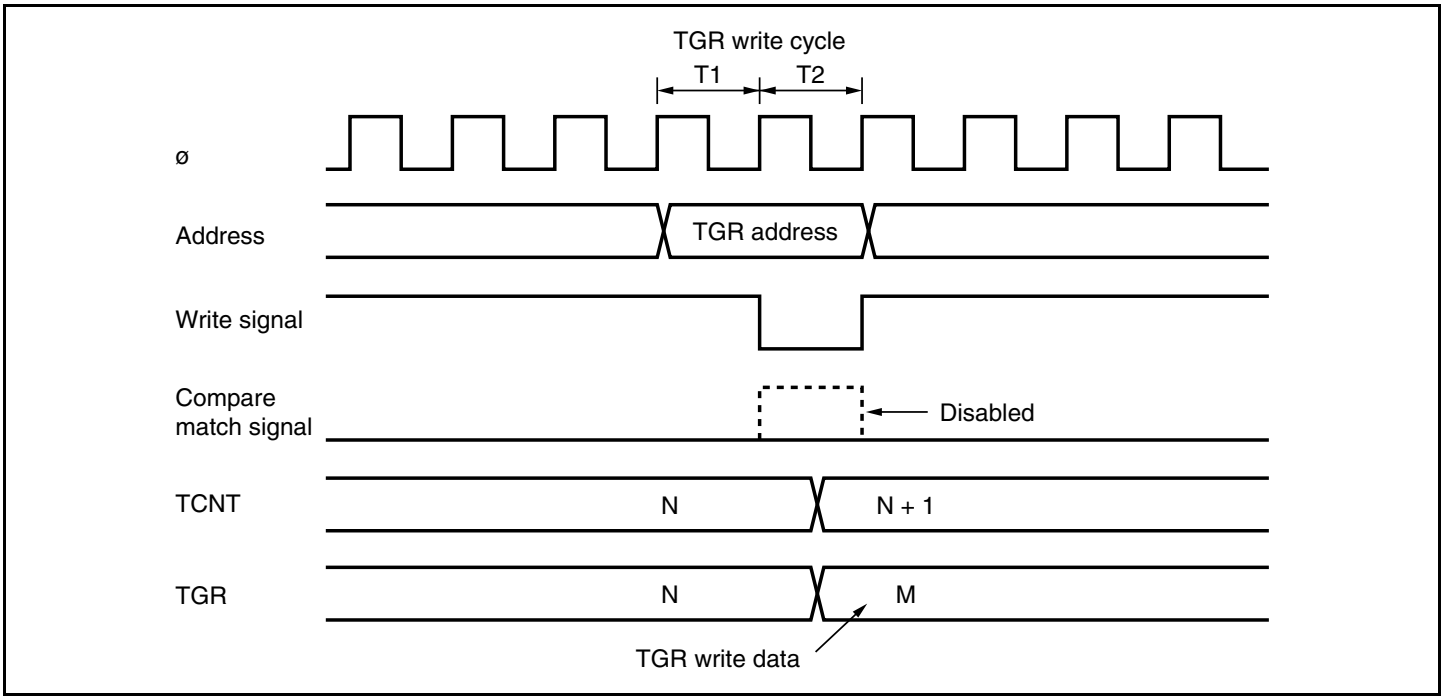


Figure 11.47 Contention between TGR Write and Compare Match

11.10.7 Contention between Buffer Register Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the data transferred to TGR by the buffer operation will be the data prior to the write.

Figure 11.48 shows the timing in this case.

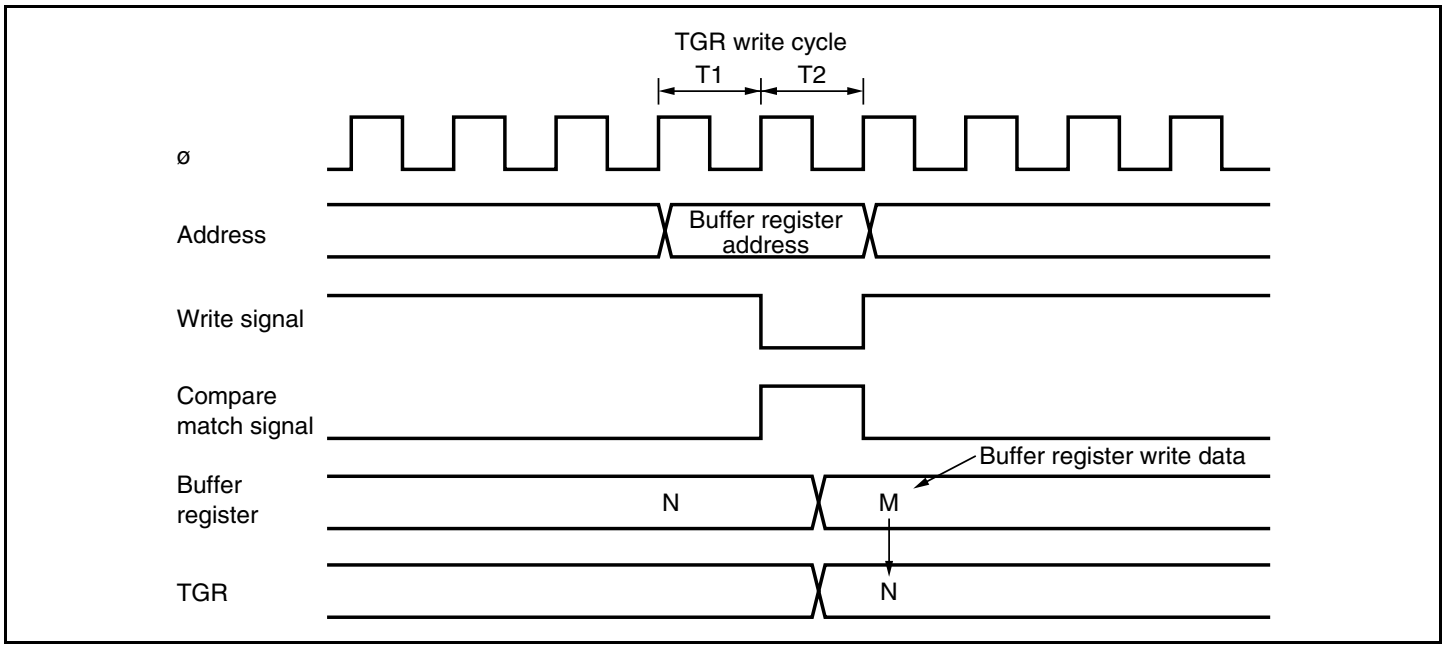


Figure 11.48 Contention between Buffer Register Write and Compare Match

11.10.8 Contention between TGR Read and Input Capture

If the input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data after input capture transfer.

Figure 11.49 shows the timing in this case.

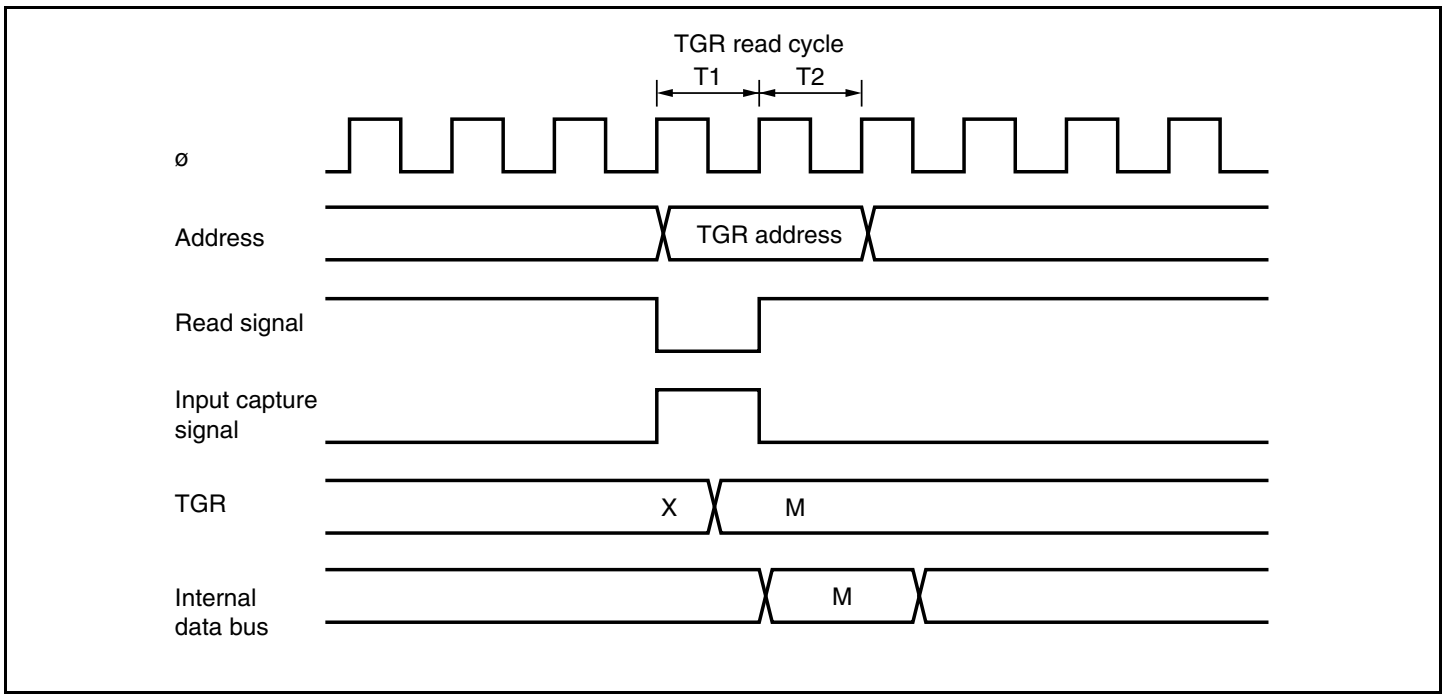


Figure 11.49 Contention between TGR Read and Input Capture

11.10.9 Contention between TGR Write and Input Capture

If the input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed.

Figure 11.50 shows the timing in this case.

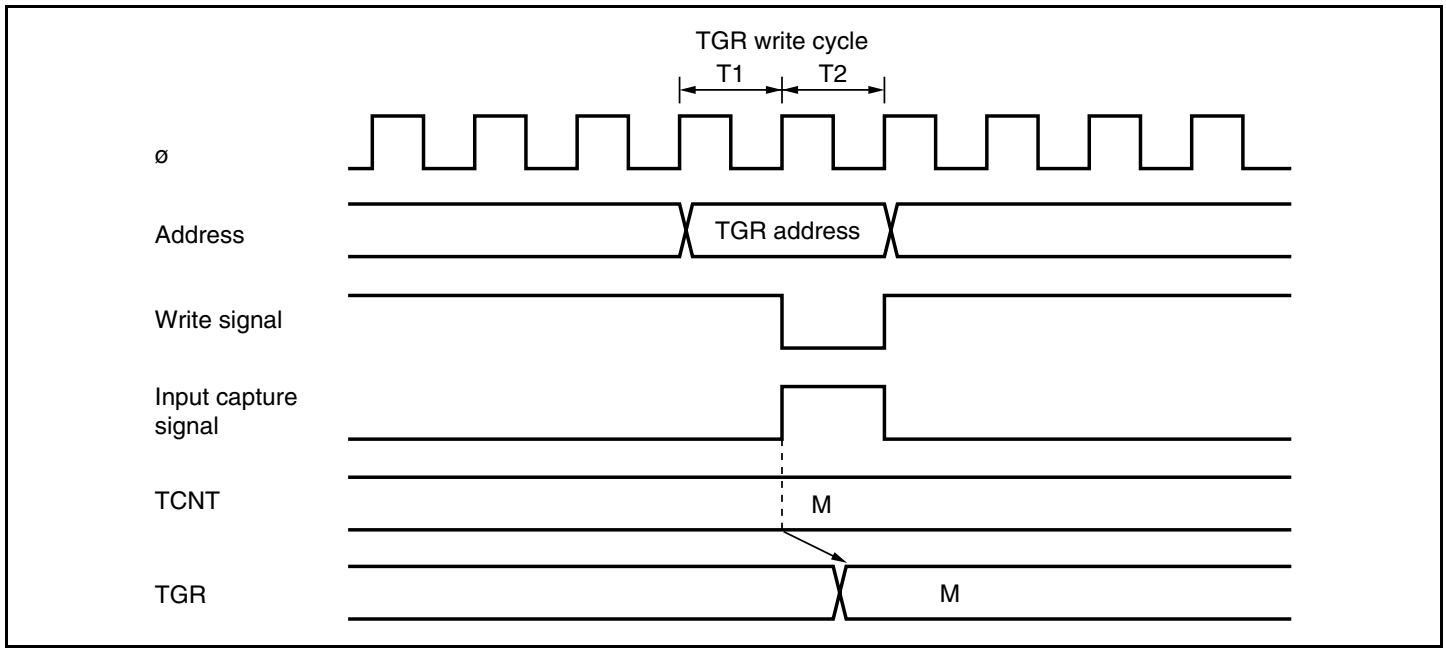


Figure 11.50 Contention between TGR Write and Input Capture

11.10.10 Contention between Buffer Register Write and Input Capture

If the input capture signal is generated in the T2 state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 11.51 shows the timing in this case.

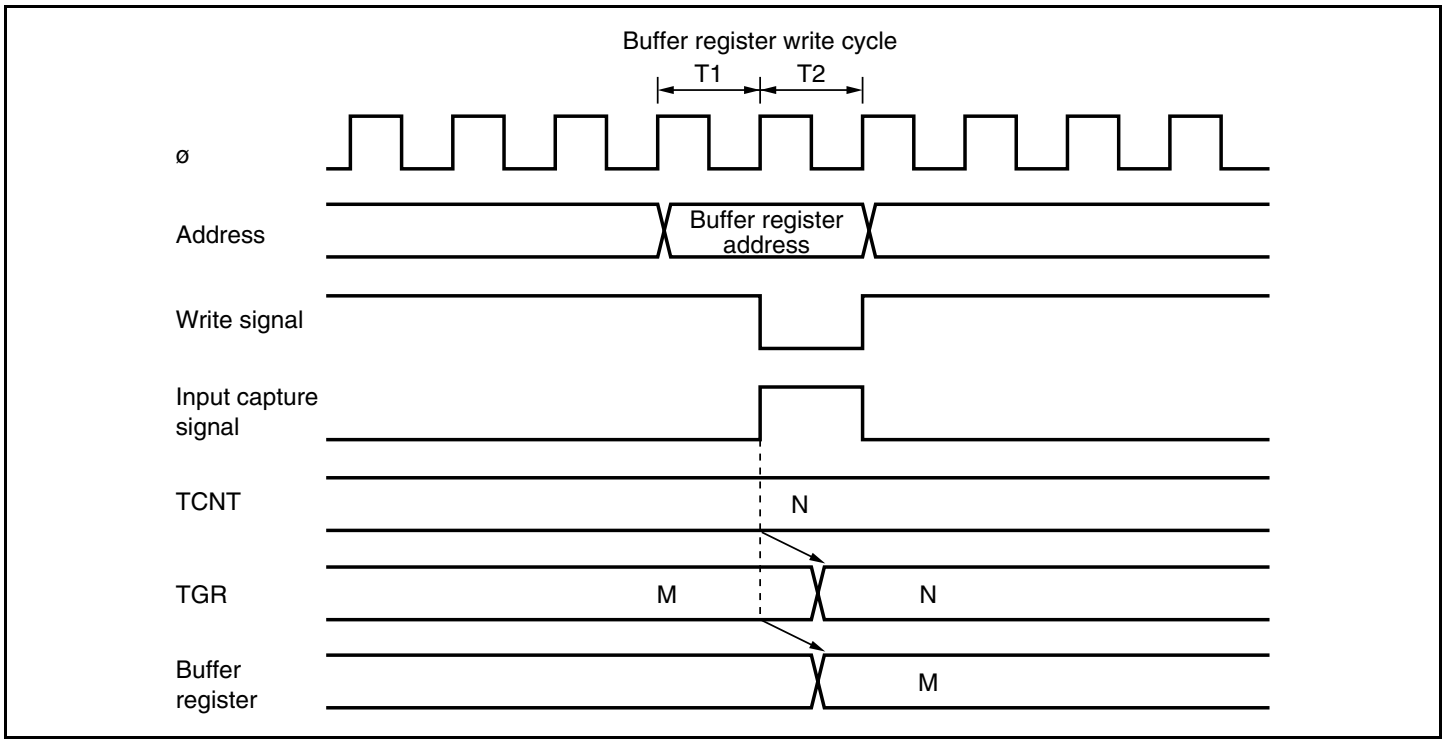


Figure 11.51 Contention between Buffer Register Write and Input Capture

11.10.11 Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 11.52 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.

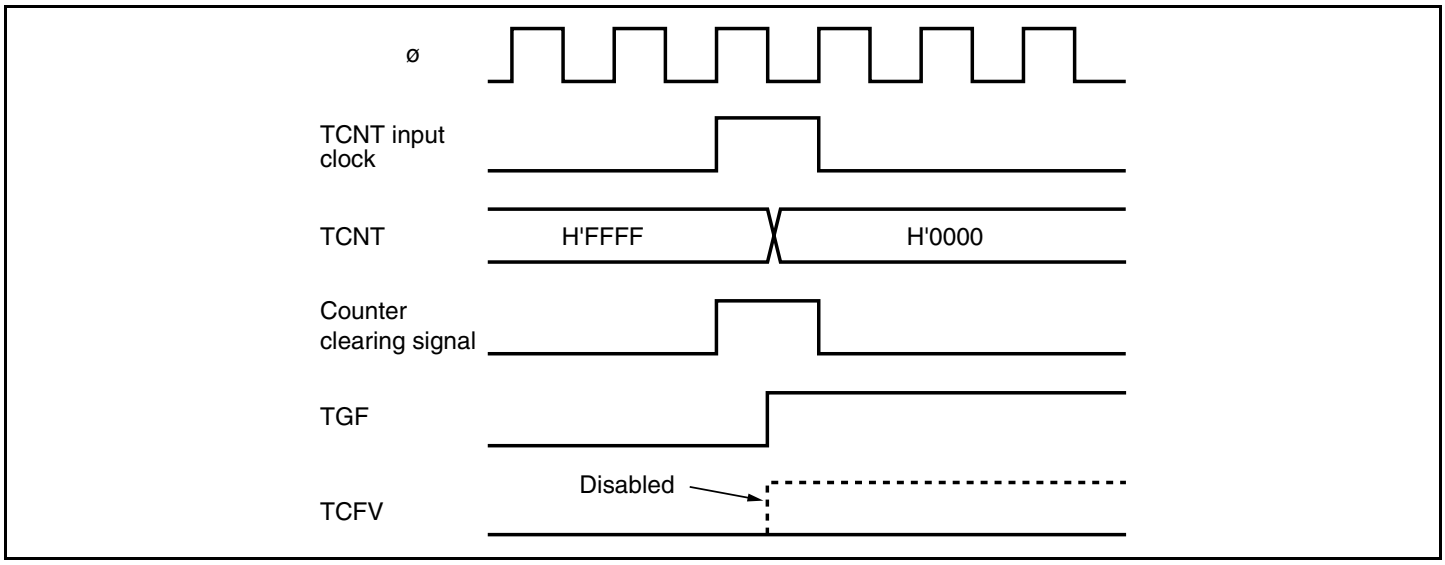


Figure 11.52 Contention between Overflow and Counter Clearing

11.10.12 Contention between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T2 state of a TCNT write cycle, when overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 11.53 shows the operation timing when there is contention between TCNT write and overflow.

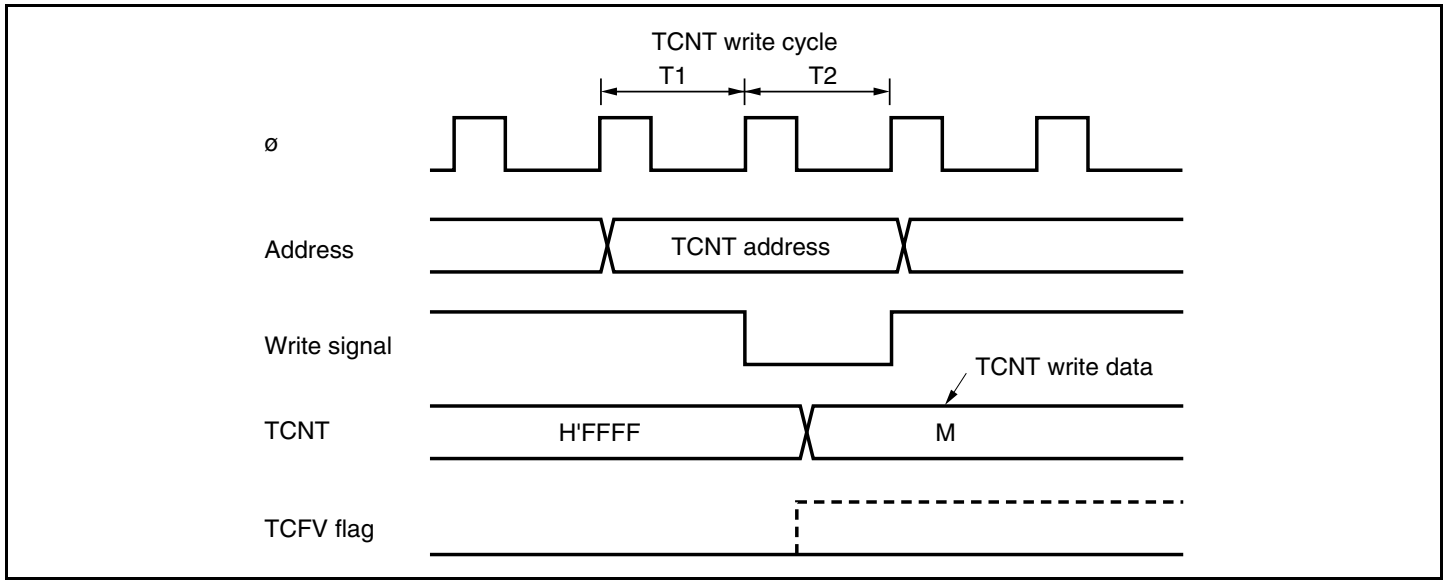


Figure 11.53 Contention between TCNT Write and Overflow

11.10.13 Multiplexing of I/O Pins

In this LSI, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLKD input pin with the TIOCB2 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

11.10.14 Interrupts and Module Stop Mode

If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DMAC or DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

Section 12 Programmable Pulse Generator (PPG)

The programmable pulse generator (PPG) provides pulse outputs by using the 16-bit timer pulse unit (TPU) as a time base. The PPG pulse outputs are divided into 4-bit groups (groups 3 to 0) that can operate both simultaneously and independently. The block diagram of PPG is shown in figure 12.1

12.1 Features

- 16-bit output data
- Four output groups
- Selectable output trigger signals
- Non-overlap mode
- Can operate together with the data transfer controller (DTC) and the DMA controller (DMAC)
- Settable inverted output
- Module stop mode can be set

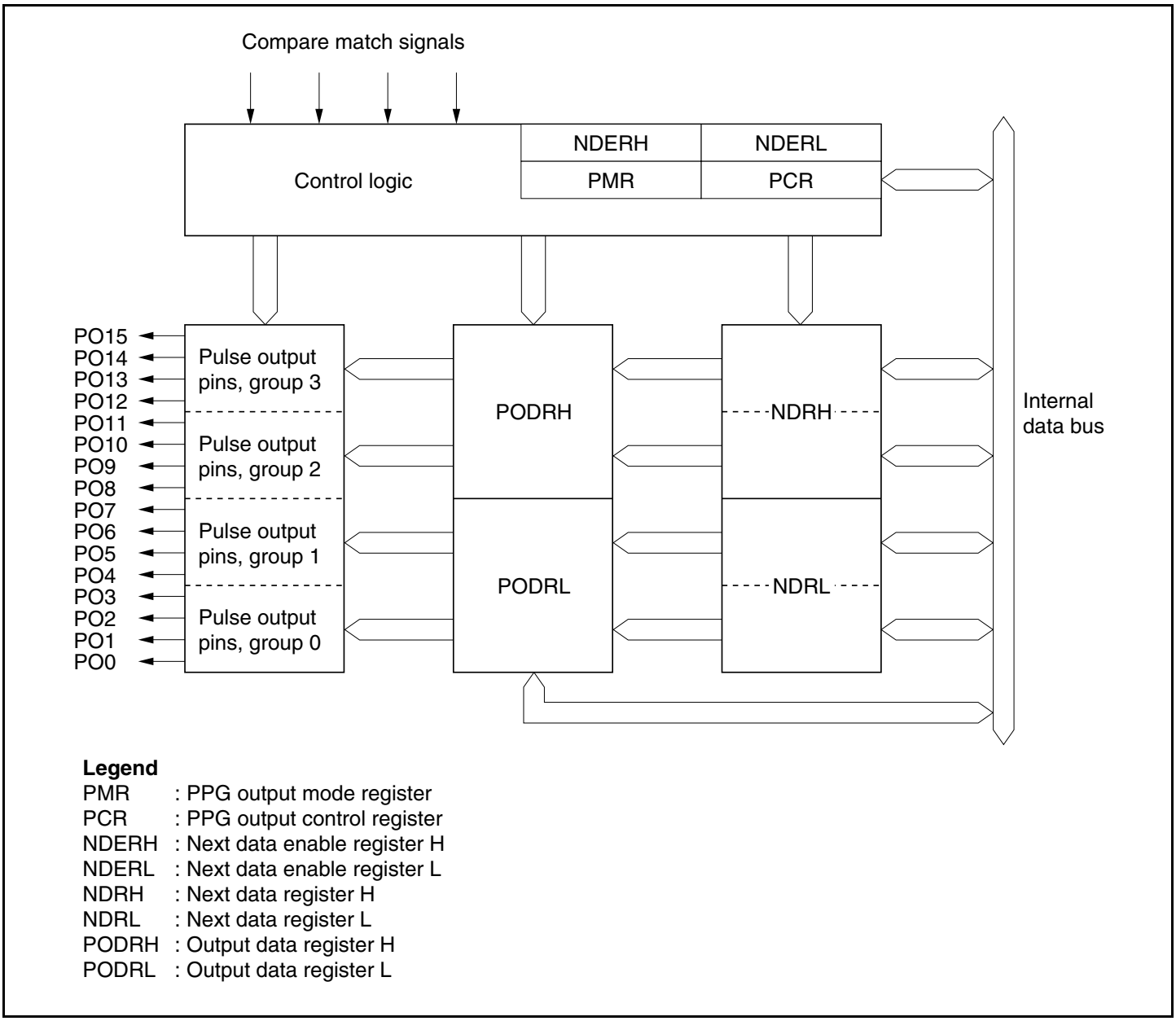


Figure 12.1 Block Diagram of PPG

12.2 Input/Output Pins

Table 12.1 shows the PPG pin configuration.

Table 12.1 Pin Configuration

Pin Name	I/O	Function
PO15	Output	Group 3 pulse output
PO14	Output	
PO13	Output	
PO12	Output	
PO11	Output	Group 2 pulse output
PO10	Output	
PO9	Output	
PO8	Output	
PO7	Output	Group 1 pulse output
PO6	Output	
PO5	Output	
PO4	Output	
PO3	Output	Group 0 pulse output
PO2	Output	
PO1	Output	
PO0	Output	

12.3 Register Descriptions

The PPG has the following registers.

- Next data enable register H (NDERH)
- Next data enable register L (NDERL)
- Output data register H (PODRH)
- Output data register L (PODRL)
- Next data register H (NDRH)
- Next data register L (NDRL)
- PPG output control register (PCR)
- PPG output mode register (PMR)

12.3.1 Next Data Enable Registers H, L (NDERH, NDERL)

NDERH, NDERL enable or disable pulse output on a bit-by-bit basis. For outputting pulse by the PPG, set the corresponding DDR to 1.

NDERH

Bit	Bit Name	Initial Value	R/W	Description
7	NDER15	0	R/W	Next Data Enable 15 to 8
6	NDER14	0	R/W	When a bit is set to 1, the value in the corresponding NDRH bit is transferred to the PODRH bit by the selected output trigger. Values are not transferred from NDRH to PODRH for cleared bits.
5	NDER13	0	R/W	
4	NDER12	0	R/W	
3	NDER11	0	R/W	
2	NDER10	0	R/W	
1	NDER9	0	R/W	
0	NDER8	0	R/W	

NDERL

Bit	Bit Name	Initial Value	R/W	Description
7	NDER7	0	R/W	Next Data Enable 7 to 0
6	NDER6	0	R/W	When a bit is set to 1, the value in the corresponding NDRL bit is transferred to the PODRL bit by the selected output trigger. Values are not transferred from NDRL to PODRL for cleared bits.
5	NDER5	0	R/W	
4	NDER4	0	R/W	
3	NDER3	0	R/W	
2	NDER2	0	R/W	
1	NDER1	0	R/W	
0	NDER0	0	R/W	

12.3.2 Output Data Registers H, L (PODRH, PODRL)

PODRH and PODRL store output data for use in pulse output. A bit that has been set for pulse output by NDER is read-only and cannot be modified.

PODRH

Bit	Bit Name	Initial Value	R/W	Description
7	POD15	0	R/W	Output Data Register 15 to 8
6	POD14	0	R/W	For bits which have been set to pulse output by NDERH, the output trigger transfers NDRH values to this register during PPG operation. While NDERH is set to 1, the CPU cannot write to this register. While NDERH is cleared, the initial output value of the pulse can be set.
5	POD13	0	R/W	
4	POD12	0	R/W	
3	POD11	0	R/W	
2	POD10	0	R/W	
1	POD9	0	R/W	
0	POD8	0	R/W	

PODRL

Bit	Bit Name	Initial Value	R/W	Description
7	POD7	0	R/W	Output Data Register 7 to 0
6	POD6	0	R/W	For bits which have been set to pulse output by NDERL, the output trigger transfers NDRL values to this register during PPG operation. While NDERL is set to 1, the CPU cannot write to this register. While NDERL is cleared, the initial output value of the pulse can be set.
5	POD5	0	R/W	
4	POD4	0	R/W	
3	POD3	0	R/W	
2	POD2	0	R/W	
1	POD1	0	R/W	
0	POD0	0	R/W	

12.3.3 Next Data Registers H, L (NDRH, NDRL)

NDRH, NDRL store the next data for pulse output. The NDR addresses differ depending on whether pulse output groups have the same output trigger or different output triggers.

NDRH

If pulse output groups 2 and 3 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 15 to 8
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3	NDR11	0	R/W	
2	NDR10	0	R/W	
1	NDR9	0	R/W	
0	NDR8	0	R/W	

If pulse output groups 2 and 3 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 15 to 12
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
3	NDR11	0	R/W	Next Data Register 11 to 8 The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
2	NDR10	0	R/W	
1	NDR9	0	R/W	
0	NDR8	0	R/W	

NDRL

If pulse output groups 0 and 1 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 7 to 0 The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
6	NDR6	0	R/W	
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3	NDR3	0	R/W	
2	NDR2	0	R/W	
1	NDR1	0	R/W	
0	NDR0	0	R/W	

If pulse output groups 0 and 1 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 7 to 4 The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
6	NDR6	0	R/W	
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
3	NDR3	0	R/W	Next Data Register 3 to 0 The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
2	NDR2	0	R/W	
1	NDR1	0	R/W	
0	NDR0	0	R/W	

12.3.4 PPG Output Control Register (PCR)

PCR selects output trigger signals on a group-by-group basis. For details on output trigger selection, refer to section 12.3.5, PPG Output Mode Register (PMR).

Bit	Bit Name	Initial Value	R/W	Description
7	G3CMS1	1	R/W	Group 3 Compare Match Select 1 and 0
6	G3CMS0	1	R/W	Select output trigger of pulse output group 3. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
5	G2CMS1	1	R/W	Group 2 Compare Match Select 1 and 0
4	G2CMS0	1	R/W	Select output trigger of pulse output group 2. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
3	G1CMS1	1	R/W	Group 1 Compare Match Select 1 and 0
2	G1CMS0	1	R/W	Select output trigger of pulse output group 1. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3

Bit	Bit Name	Initial Value	R/W	Description
1	G0CMS1	1	R/W	Group 0 Compare Match Select 1 and 0
0	G0CMS0	1	R/W	Select output trigger of pulse output group 0. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3

12.3.5 PPG Output Mode Register (PMR)

PMR selects the pulse output mode of the PPG for each group. If inverted output is selected, a low-level pulse is output when PODRH is 1 and a high-level pulse is output when PODRH is 0. If non-overlapping operation is selected, PPG updates its output values at compare match A or B of the TPU that becomes the output trigger. For details, refer to section 12.4.4, Non-Overlapping Pulse Output.

Bit	Bit Name	Initial Value	R/W	Description
7	G3INV	1	R/W	Group 3 Inversion Selects direct output or inverted output for pulse output group 3. 0: Inverted output 1: Direct output
6	G2INV	1	R/W	Group 2 Inversion Selects direct output or inverted output for pulse output group 2. 0: Inverted output 1: Direct output
5	G1INV	1	R/W	Group 1 Inversion Selects direct output or inverted output for pulse output group 1. 0: Inverted output 1: Direct output
4	G0INV	1	R/W	Group 0 Inversion Selects direct output or inverted output for pulse output group 0. 0: Inverted output 1: Direct output

Bit	Bit Name	Initial Value	R/W	Description
3	G3NOV	0	R/W	Group 3 Non-Overlap Selects normal or non-overlapping operation for pulse output group 3. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)
2	G2NOV	0	R/W	Group 2 Non-Overlap Selects normal or non-overlapping operation for pulse output group 2. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)
1	G1NOV	0	R/W	Group 1 Non-Overlap Selects normal or non-overlapping operation for pulse output group 1. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)
0	G0NOV	0	R/W	Group 0 Non-Overlap Selects normal or non-overlapping operation for pulse output group 0. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)

12.4 Operation

Figure 12.2 shows an overview diagram of the PPG. PPG pulse output is enabled when the corresponding bits in P1DDR, P2DDR, and NDER are set to 1. An initial output value is determined by its corresponding PODR initial setting. When the compare match event specified by PCR occurs, the corresponding NDR bit contents are transferred to PODR to update the output values. Sequential output of data of up to 16 bits is possible by writing new output data to NDR before the next compare match.

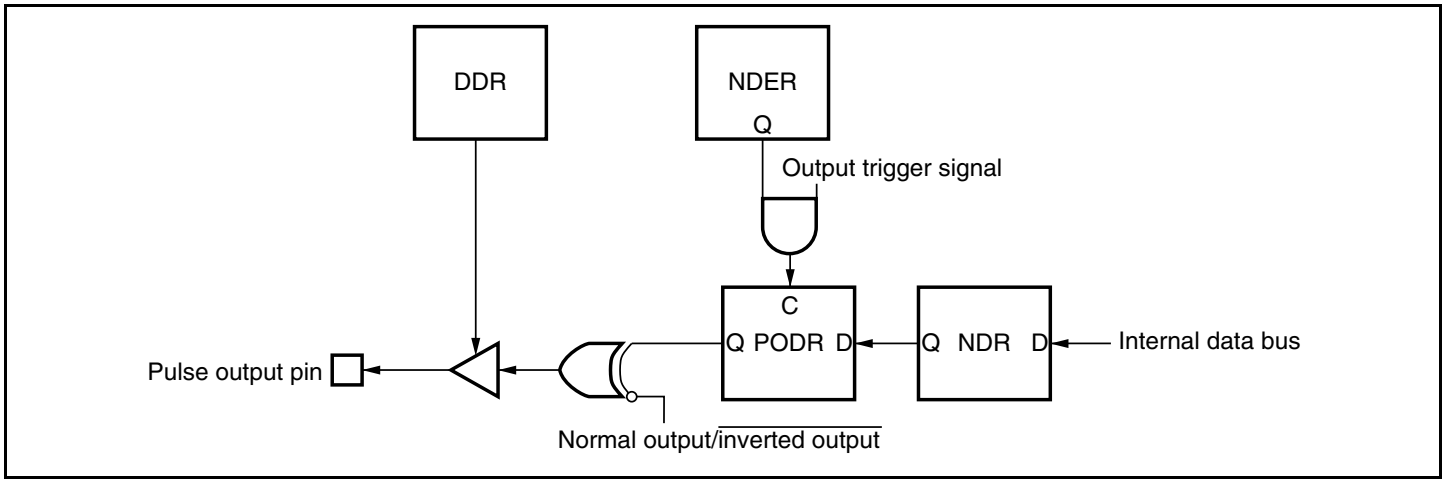


Figure 12.2 Overview Diagram of PPG

12.4.1 Output Timing

If pulse output is enabled, NDR contents are transferred to PODR and output when the specified compare match event occurs. Figure 12.3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare match A.

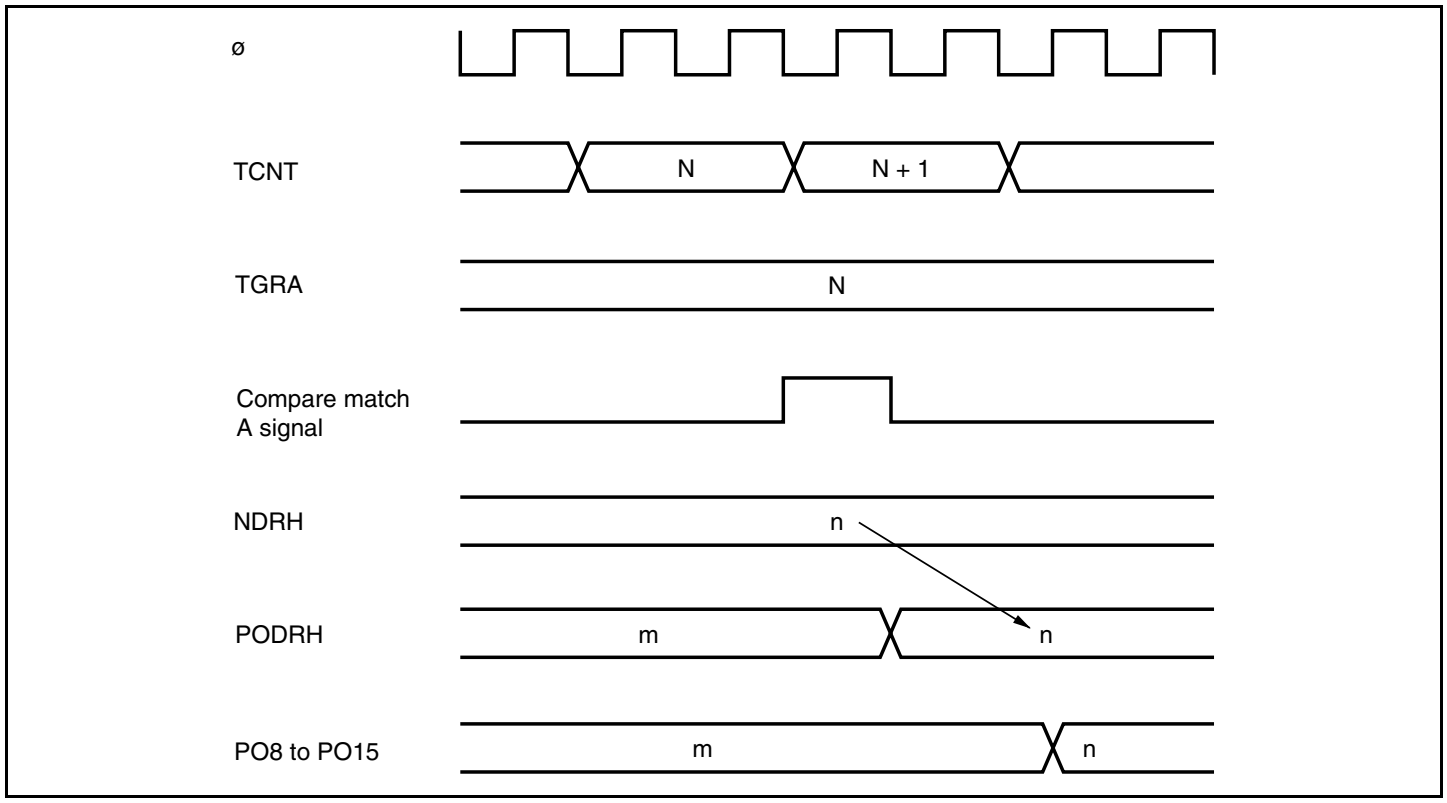
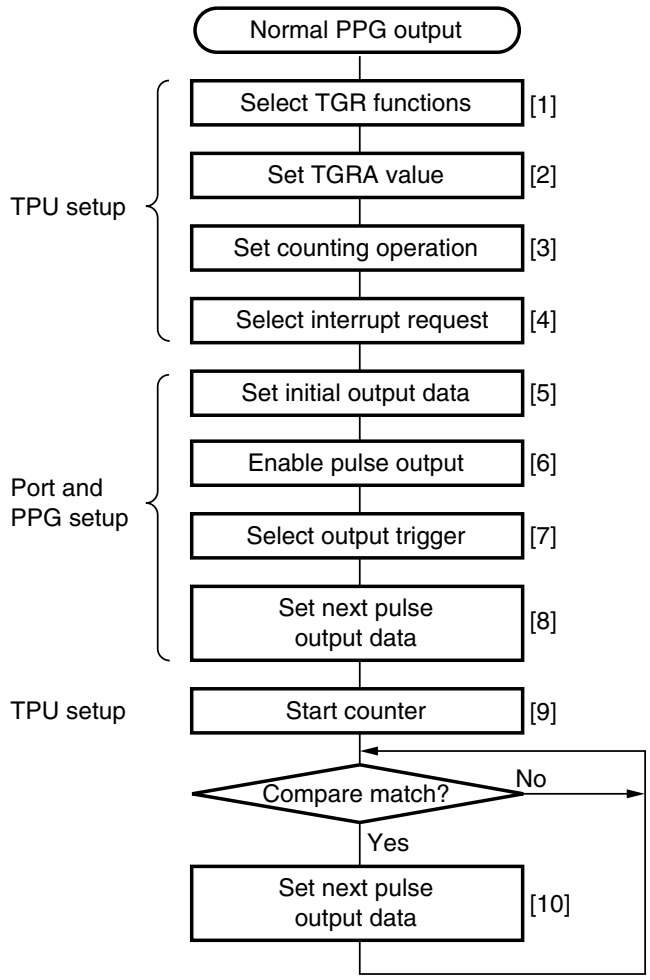


Figure 12.3 Timing of Transfer and Output of NDR Contents (Example)

12.4.2 Sample Setup Procedure for Normal Pulse Output

Figure 12.4 shows a sample procedure for setting up normal pulse output.



[1] Set TIOR to make TGRA an output compare register (with output disabled)

[2] Set the PPG output trigger period

[3] Select the counter clock source with bits TPSC2 to TPSC0 in TCR. Select the counter clear source with bits CCLR2 to CCLR0.

[4] Enable the TGIA interrupt in TIER. The DTC or DMAC can also be set up to transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the DDR and NDER bits for the pins to be used for pulse output to 1.

[7] Select the TPU compare match event to be used as the output trigger in PCR.

[8] Set the next pulse output values in NDR.

[9] Set the CST bit in TSTR to 1 to start the TCNT counter.

[10] At each TGIA interrupt, set the next output values in NDR.

Figure 12.4 Setup Procedure for Normal Pulse Output (Example)

12.4.3 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

Figure 12.5 shows an example in which pulse output is used for cyclic five-phase pulse output.

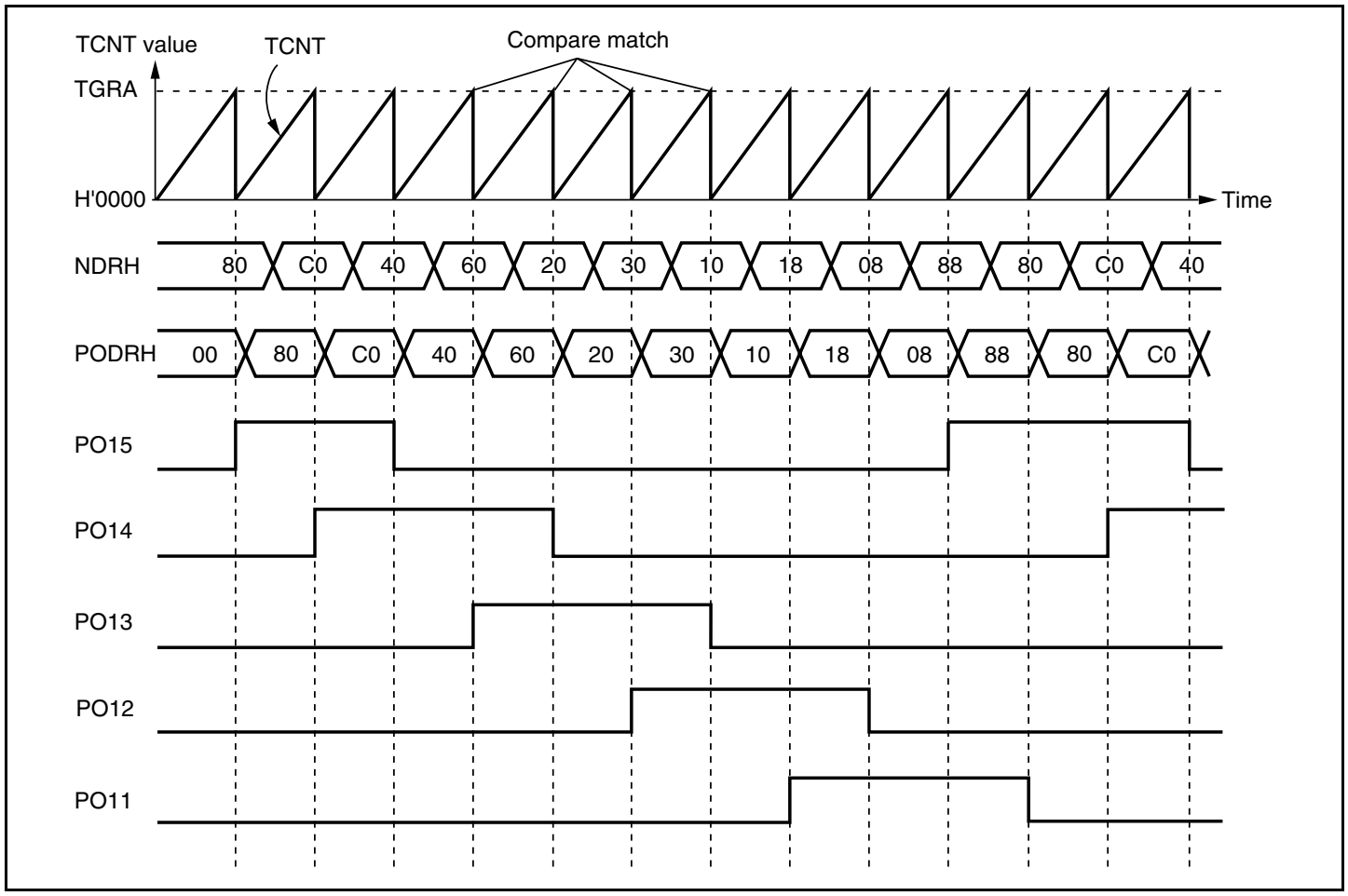


Figure 12.5 Normal Pulse Output Example (Five-Phase Pulse Output)

1. Set up TGRA in TPU which is used as the output trigger to be an output compare register. Set a cycle in TGRA so the counter will be cleared by compare match A. Set the TGIEA bit in TIER to 1 to enable the compare match/input capture A (TGIA) interrupt.
2. Write H'F8 in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Write output data H'80 in NDRH.
3. The timer counter in the TPU channel starts. When compare match A occurs, the NDRH contents are transferred to PODRH and output. The TGIA interrupt handling routine writes the next output data (H'C0) in NDRH.
4. Five-phase pulse output (one or two phases active at a time) can be obtained subsequently by writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88... at successive TGIA interrupts.

   If the DTC or DMAC is set for activation by the TGIA interrupt, pulse output can be obtained without imposing a load on the CPU.

12.4.4 Non-Overlapping Pulse Output

During non-overlapping operation, transfer from NDR to PODR is performed as follows:

- NDR bits are always transferred to PODR bits at compare match A.
- At compare match B, NDR bits are transferred only if their value is 0. Bits are not transferred if their value is 1.

Figure 12.6 illustrates the non-overlapping pulse output operation.

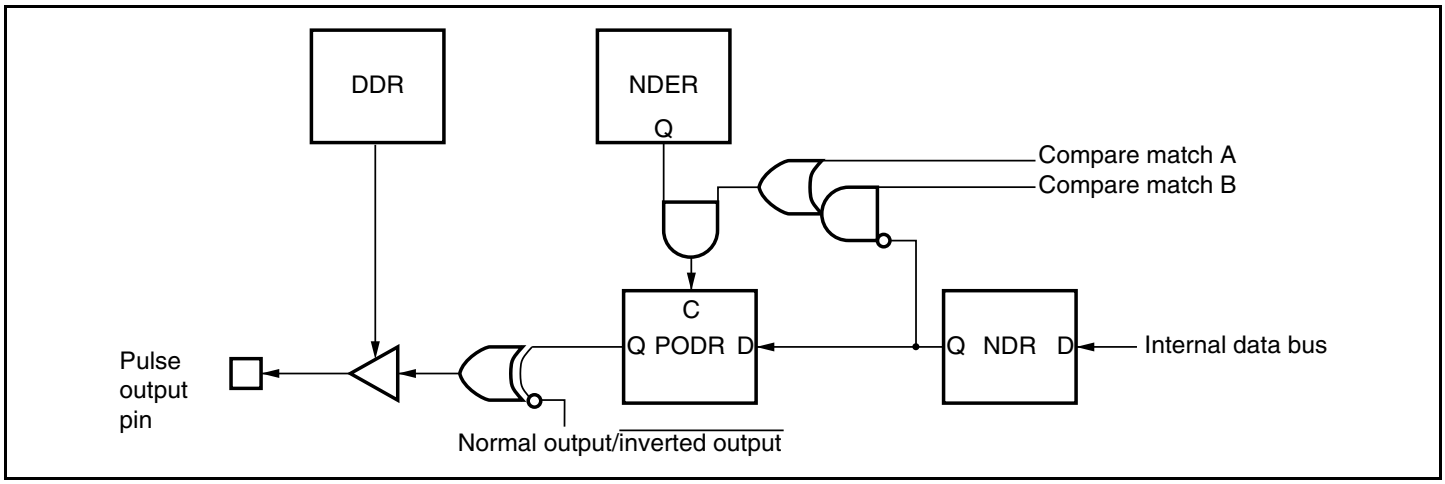


Figure 12.6 Non-Overlapping Pulse Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur before compare match A.

The NDR contents should not be altered during the interval from compare match B to compare match A (the non-overlap margin).

This can be accomplished by having the TGIA interrupt handling routine write the next data in NDR, or by having the TGIA interrupt activate the DTC or DMAC. Note, however, that the next data must be written before the next compare match B occurs.

Figure 12.7 shows the timing of this operation.

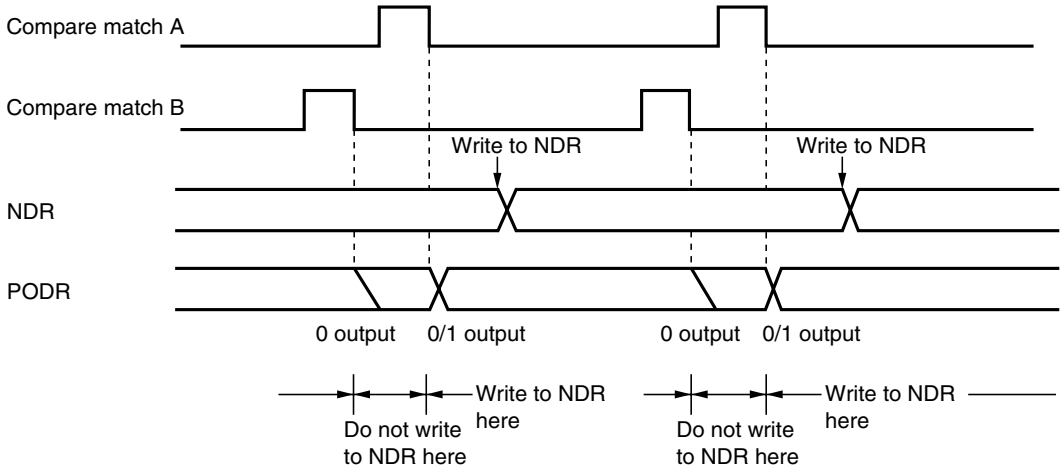
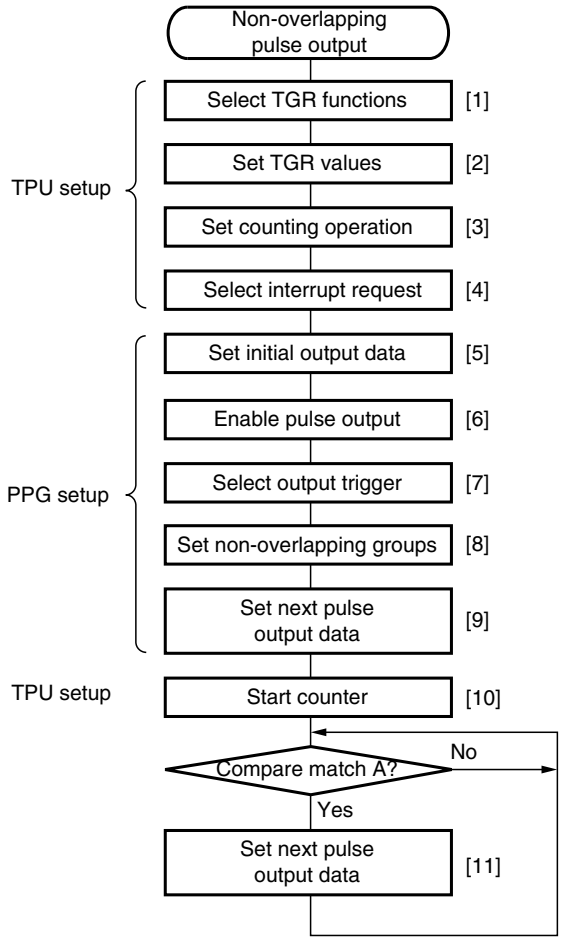


Figure 12.7 Non-Overlapping Operation and NDR Write Timing

12.4.5 Sample Setup Procedure for Non-Overlapping Pulse Output

Figure 12.8 shows a sample procedure for setting up non-overlapping pulse output.



[1] Set TIOR to make TGRA and TGRB an output compare registers (with output disabled)

[2] Set the pulse output trigger period in TGRB and the non-overlap period in TGRA.

[3] Select the counter clock source with bits TPSC2 to TPSC0 in TCR. Select the counter clear source with bits CCLR2 to CCLR0.

[4] Enable the TGIA interrupt in TIER. The DTC or DMAC can also be set up to transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the DDR and NDER bits for the pins to be used for pulse output to 1.

[7] Select the TPU compare match event to be used as the pulse output trigger in PCR.

[8] In PMR, select the groups that will operate in non-overlap mode.

[9] Set the next pulse output values in NDR.

[10] Set the CST bit in TSTR to 1 to start the TCNT counter.

[11] At each TGIA interrupt, set the next output values in NDR.

Figure 12.8 Setup Procedure for Non-Overlapping Pulse Output (Example)

12.4.6 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)

Figure 12.9 shows an example in which pulse output is used for four-phase complementary non-overlapping pulse output.

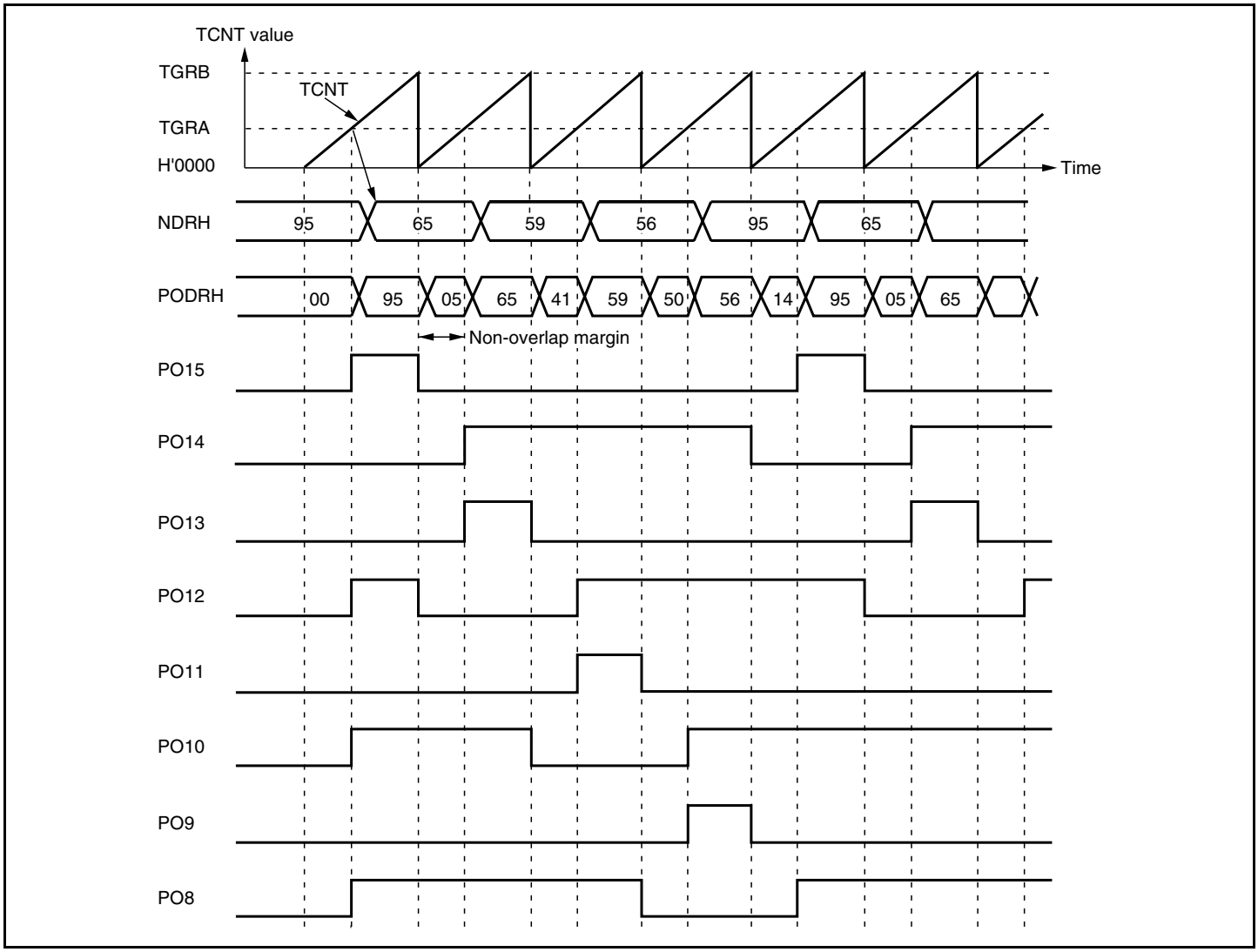


Figure 12.9 Non-Overlapping Pulse Output Example (Four-Phase Complementary)

1. Set up the TPU channel to be used as the output trigger channel so that TGRA and TGRB are output compare registers. Set the trigger period in TGRB and the non-overlap margin in TGRA, and set the counter to be cleared by compare match B. Set the TGIEA bit in TIER to 1 to enable the TGIA interrupt.
2. Write H'FF in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Set the G3NOV and G2NOV bits in PMR to 1 to select non-overlapping output. Write output data H'95 in NDRH.
3. The timer counter in the TPU channel starts. When a compare match with TGRB occurs, outputs change from 1 to 0. When a compare match with TGRA occurs, outputs change from 0

to 1 (the change from 0 to 1 is delayed by the value set in TGRA). The TGIA interrupt handling routine writes the next output data (H'65) in NDRH.

4. Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing H'59, H'56, H'95... at successive TGIA interrupts.

   If the DTC or DMAC is set for activation by the TGIA interrupt, pulse output can be obtained without imposing a load on the CPU.

12.4.7 Inverted Pulse Output

If the G3INV, G2INV, G1INV, and G0INV bits in PMR are cleared to 0, values that are the inverse of the PODR contents can be output.

Figure 12.10 shows the outputs when G3INV and G2INV are cleared to 0, in addition to the settings of figure 12.9.

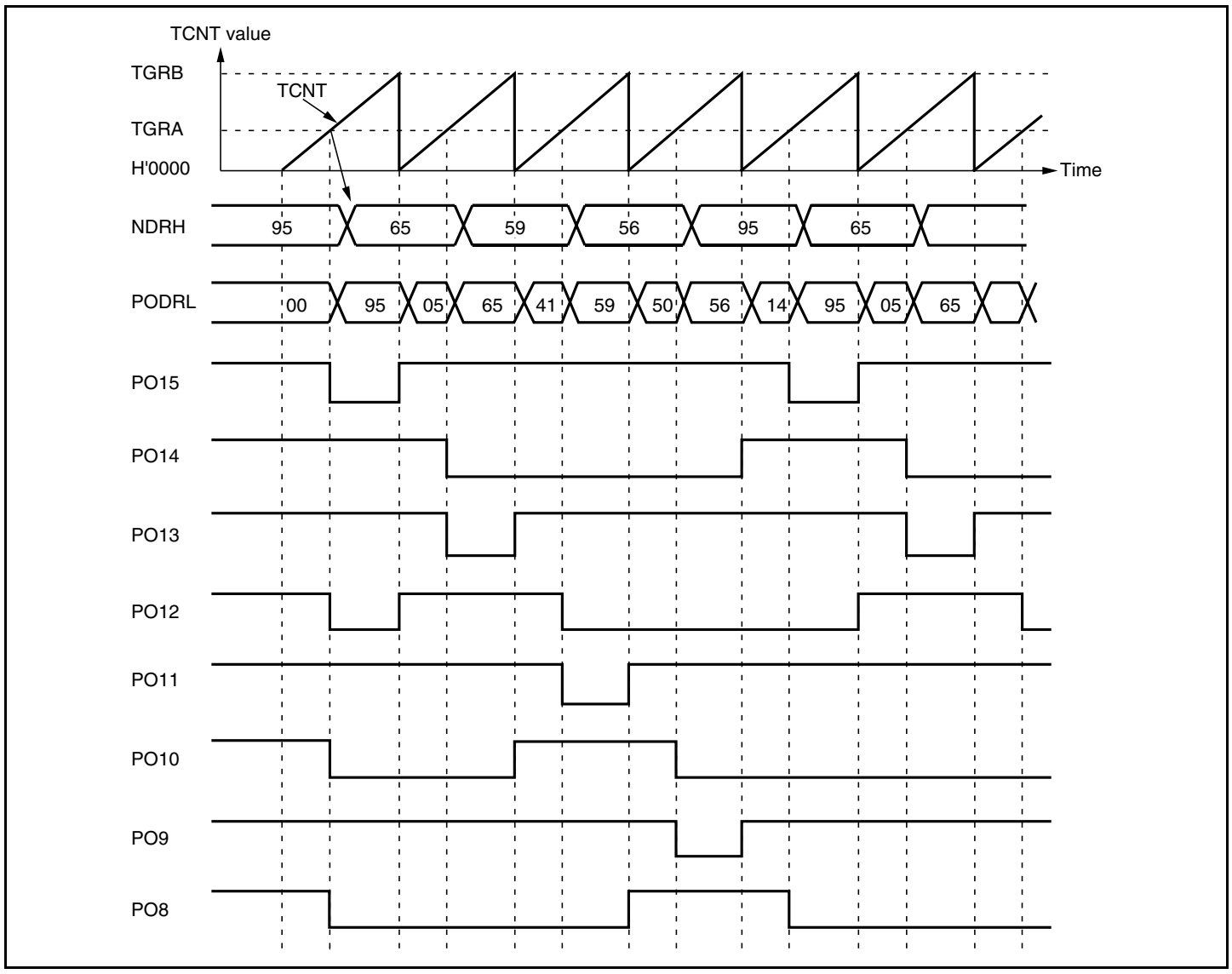


Figure 12.10 Inverted Pulse Output (Example)

12.4.8 Pulse Output Triggered by Input Capture

Pulse output can be triggered by TPU input capture as well as by compare match. If TGRA functions as an input capture register in the TPU channel selected by PCR, pulse output will be triggered by the input capture signal.

Figure 12.11 shows the timing of this output.

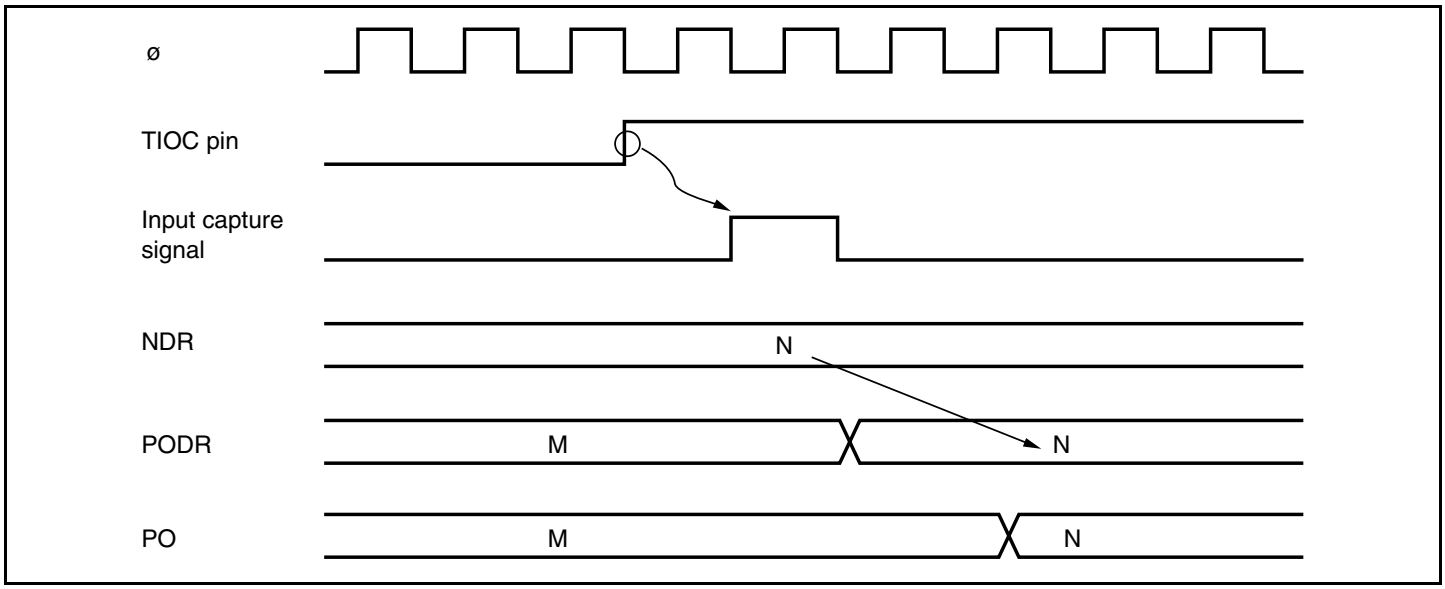


Figure 12.11 Pulse Output Triggered by Input Capture (Example)

12.5 Usage Notes

12.5.1 Module Stop Mode Setting

PPG operation can be disabled or enabled using the module stop control register. The initial value is for PPG operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

12.5.2 Operation of Pulse Output Pins

Pins PO0 to PO15 are also used for other peripheral functions such as the TPU. When output by another peripheral function is enabled, the corresponding pins cannot be used for pulse output. Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of the usage of the pins.

Pin functions should be changed only under conditions in which the output trigger event will not occur.

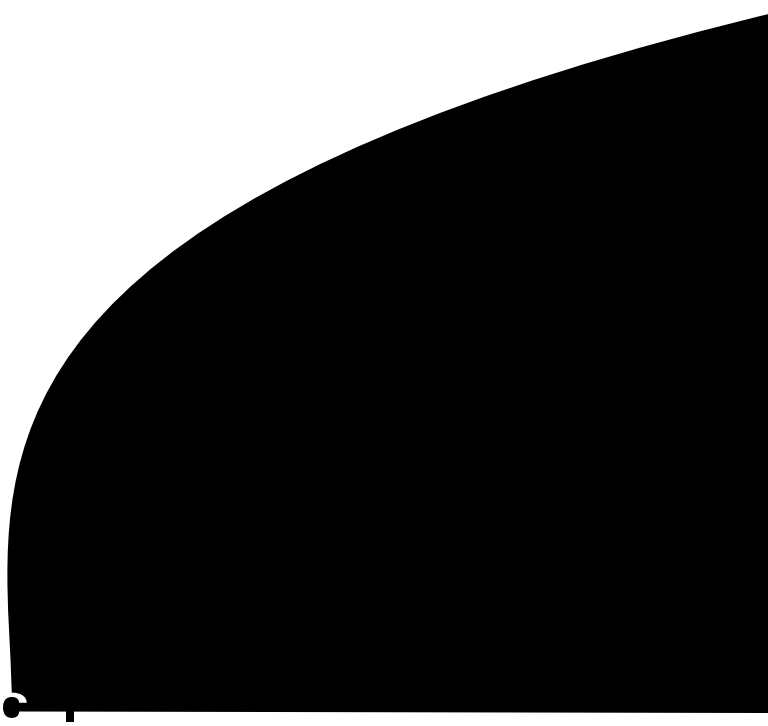
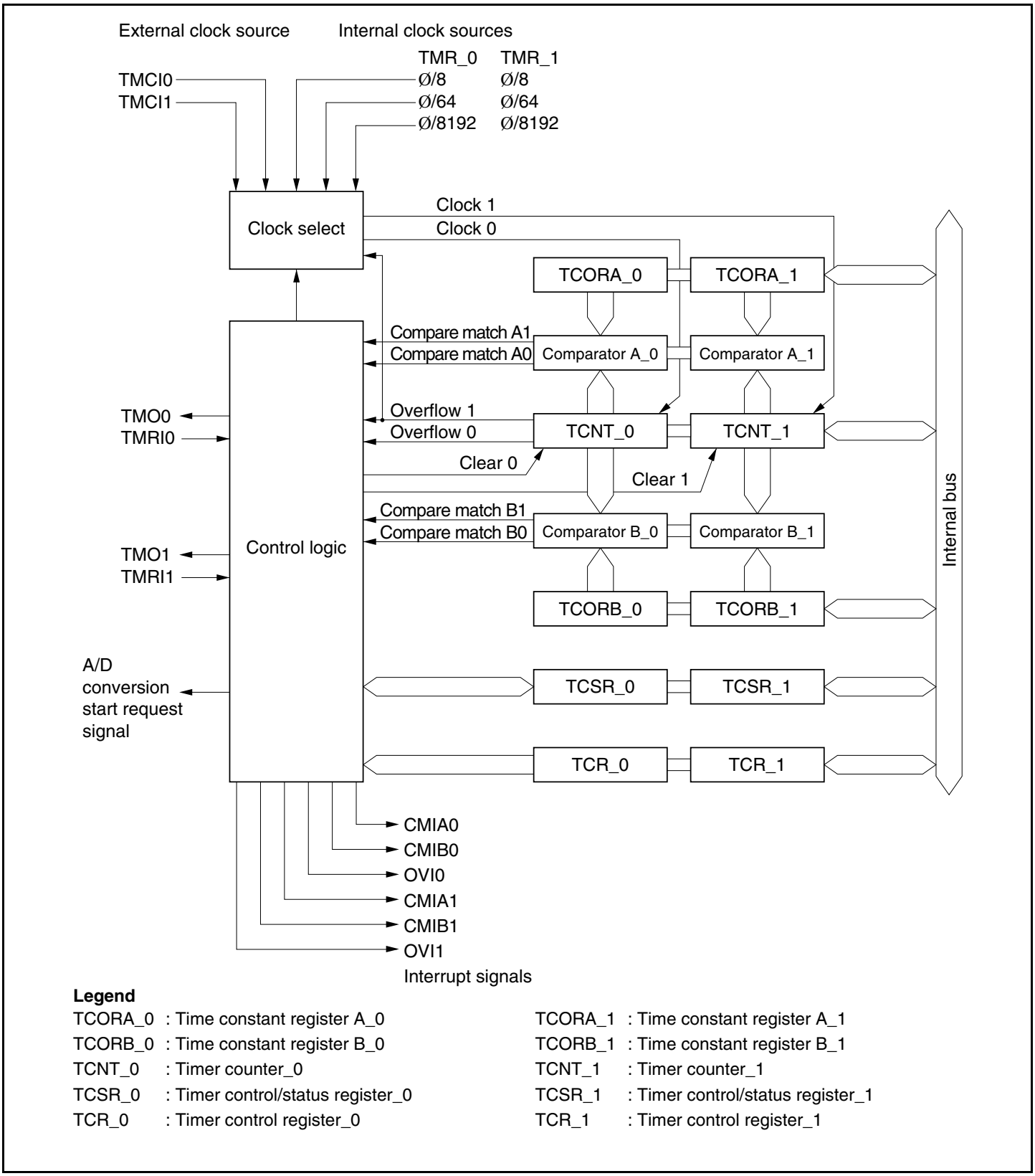


Figure 13.1 shows a block diagram of the 8-bit timer module (TMR_0 and TMR_1).



Legend

TCORA_0	: Time constant register A_0	TCORA_1	: Time constant register A_1
TCORB_0	: Time constant register B_0	TCORB_1	: Time constant register B_1
TCNT_0	: Timer counter_0	TCNT_1	: Timer counter_1
TCSR_0	: Timer control/status register_0	TCSR_1	: Timer control/status register_1
TCR_0	: Timer control register_0	TCR_1	: Timer control register_1

Figure 13.1 Block Diagram of 8-Bit Timer Module

13.2 Input/Output Pins

Table 13.1 shows the pin configuration of the 8-bit timer.

Table 13.1 Pin Configuration

Channel	Name	Symbol	I/O	Function
0	Timer output pin	TMO0	Output	Outputs at compare match
	Timer clock input pin	TMCI0	Input	Inputs external clock for counter
	Timer reset input pin	TMRI0	Input	Inputs external reset to counter
1	Timer output pin	TMO1	Output	Outputs at compare match
	Timer clock input pin	TMCI1	Input	Inputs external clock for counter
	Timer reset input pin	TMRI1	Input	Inputs external reset to counter

13.3 Register Descriptions

The 8-bit timer module has the following registers. For details on the module stop control register, refer to section 22.1.2 Module Stop Control Registers H, L (MSTPCRH, MSTPCRL).

- Timer counter_0 (TCNT_0)
- Time constant register A_0 (TCORA_0)
- Time constant register B_0 (TCORB_0)
- Timer control register_0 (TCR_0)
- Timer control/status register_0 (TCSR_0)
- Timer counter_1 (TCNT_1)
- Time constant register A_1 (TCORA_1)
- Time constant register B_1 (TCORB_1)
- Timer control register_1 (TCR_1)
- Timer control/status register_1 (TCSR_1)

13.3.1 Timer Counter (TCNT)

TCNT is 8-bit up-counter. TCNT_0 and TCNT_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction. Bits CKS2 to CKS0 in TCR are used to select a clock. TCNT can be cleared by an external reset input or by a compare match signal A or B. Which signal is to be used for clearing is selected by bits CCLR1 and CCLR0 in TCR. When TCNT overflows from H'FF to H'00, OVF in TCSR is set to 1. TCNT is initialized to H'00.

13.3.2 Time Constant Register A (TCORA)

TCORA is 8-bit readable/writable register. TCORA_0 and TCORA_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction.

The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding CMFA flag in TCSR is set to 1. Note, however, that comparison is disabled during the T2 state of a TCORA write cycle.

The timer output from the TMO pin can be freely controlled by this compare match signal (compare match A) and the settings of bits OS1 and OS0 in TCSR.

TCORA is initialized to H'FF.

13.3.3 Time Constant Register B (TCORB)

TCORB is 8-bit readable/writable register. TCORB_0 and TCORB_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction.

TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding CMFB flag in TCSR is set to 1. Note, however, that comparison is disabled during the T2 state of a TCOBR write cycle.

The timer output from the TMO pin can be freely controlled by this compare match signal (compare match B) and the settings of bits OS3 and OS2 in TCSR.

TCORB is initialized to H'FF.

13.3.4 Timer Control Register (TCR)

TCR selects the clock source and the time at which TCNT is cleared, and controls interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare Match Interrupt Enable B Selects whether CMFB interrupt requests (CMIB) are enabled or disabled when the CMFB flag in TCSR is set to 1. 0: CMFB interrupt requests (CMIB) are disabled 1: CMFB interrupt requests (CMIB) are enabled
6	CMIEA	0	R/W	Compare Match Interrupt Enable A Selects whether CMFA interrupt requests (CMIA) are enabled or disabled when the CMFA flag in TCSR is set to 1. 0: CMFA interrupt requests (CMIA) are disabled 1: CMFA interrupt requests (CMIA) are enabled
5	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether OVF interrupt requests (OVI) are enabled or disabled when the OVF flag in TCSR is set to 1. 0: OVF interrupt requests (OVI) are disabled 1: OVF interrupt requests (OVI) are enabled
4 3	CCLR1 CCLR0	0 0	R/W R/W	Counter Clear 1 and 0 These bits select the method by which TCNT is cleared 00: Clearing is disabled 01: Clear by compare match A 10: Clear by compare match B 11: Clear by rising edge of external reset input
2 1 0	CKS2 CKS1 CKS0	0 0 0	R/W R/W R/W	Clock Select 2 to 0 These bits select the clock input to TCNT and count condition. See table 13.2.

Table 13.2 Clock Input to TCNT and Count Condition

Channel	TCR Bit 2 CKS2	TCR Bit 1 CKS1	TCR Bit 0 CKS0	Description
TMR_0	0	0	0	Clock input disabled
			1	Internal clock, counted at falling edge of ø/8
		1	0	Internal clock, counted at falling edge of ø/64
			1	Internal clock, counted at falling edge of ø/8192
	1	0	0	Count at TCNT_1 overflow signal*
TMR_1	0	0	0	Clock input disabled
			1	Internal clock, counted at falling edge of ø/8
		1	0	Internal clock, counted at falling edge of ø/64
			1	Internal clock, counted at falling edge of ø/8192
	1	0	0	Count at TCNT_0 compare match A*
All	1	0	1	External clock, counted at rising edge
		1	0	External clock, counted at falling edge
		1	1	External clock, counted at both rising and falling edges

Note: If the count input of TMR_0 is the TCNT_1 overflow signal and that of TMR_1 is the TCNT_0 compare match signal, no incrementing clock is generated. Do not use this setting.

13.3.5 Timer Control/Status Register (TCSR)

TCSR displays status flags, and controls compare match output.

TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare Match Flag B [Setting condition] • Set when TCNT matches TCORB [Clearing conditions] • Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB • When DTC is activated by CMIB interrupt while DISEL bit of MRB in DTC is 0

Bit	Bit Name	Initial Value	R/W	Description
6	CMFA	0	R/(W)*	Compare Match Flag A [Setting condition] • Set when TCNT matches TCORA [Clearing conditions] • Cleared by reading CMFA when CMFA = 1, then writing 0 to CMFA • When DTC is activated by CMIA interrupt while DISEL bit of MRB in DTC is 0
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] Set when TCNT overflows from H'FF to H'00 [Clearing condition] Cleared by reading OVF when OVF = 1, then writing 0 to OVF
4	ADTE	0	R/W	A/D Trigger Enable Selects enabling or disabling of A/D converter start requests by compare match A. 0: A/D converter start requests by compare match A are disabled 1: A/D converter start requests by compare match A are enabled
3 2	OS3 OS2	0 0	R/W R/W	Output Select 3 and 2 These bits select a method of TMO pin output when compare match B of TCORB and TCNT occurs. 00: No change when compare match B occurs 01: 0 is output when compare match B occurs 10: 1 is output when compare match B occurs 11: Output is inverted when compare match B occurs (toggle output)

Bit	Bit Name	Initial Value	R/W	Description
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits select a method of TMO pin output when compare match A of TCORA and TCNT occurs. 00: No change when compare match A occurs 01: 0 is output when compare match A occurs 10: 1 is output when compare match A occurs 11: Output is inverted when compare match A occurs (toggle output)

Note: Only 0 can be written to bits 7 to 5, to clear these flags.

TCSR_1

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare Match Flag B [Setting condition] • Set when TCNT matches TCORB [Clearing conditions] • Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB • When DTC is activated by CMIB interrupt while DISEL bit of MRB in DTC is 0
6	CMFA	0	R/(W)*	Compare Match Flag A [Setting condition] • Set when TCNT matches TCORA [Clearing conditions] • Cleared by reading CMFA when CMFA = 1, then writing 0 to CMFA • When DTC is activated by CMIA interrupt while DISEL bit of MRB in DTC is 0
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] Set when TCNT overflows from H'FF to H'00 [Clearing condition] Cleared by reading OVF when OVF = 1, then writing 0 to OVF

Bit	Bit Name	Initial Value	R/W	Description
4	—	1	R	Reserved This bit is always read as 1 and cannot be modified.
3 2	OS3 OS2	0 0	R/W R/W	Output Select 3 and 2 These bits select a method of TMO pin output when compare match B of TCORB and TCNT occurs. 00: No change when compare match B occurs 01: 0 is output when compare match B occurs 10: 1 is output when compare match B occurs 11: Output is inverted when compare match B occurs (toggle output)
1 0	OS1 OS0	0 0	R/W R/W	Output Select 1 and 0 These bits select a method of TMO pin output when compare match A of TCORA and TCNT occurs. 00: No change when compare match A occurs 01: 0 is output when compare match A occurs 10: 1 is output when compare match A occurs 11: Output is inverted when compare match A occurs (toggle output)

Note: Only 0 can be written to bits 7 to 5, to clear these flags.

13.4 Operation

13.4.1 Pulse Output

Figure 13.2 shows an example that the 8-bit timer is used to generate a pulse output with a selected duty cycle. The control bits are set as follows:

[1] In TCR, bit CCLR1 is cleared to 0 and bit CCLR0 is set to 1 so that the timer counter is

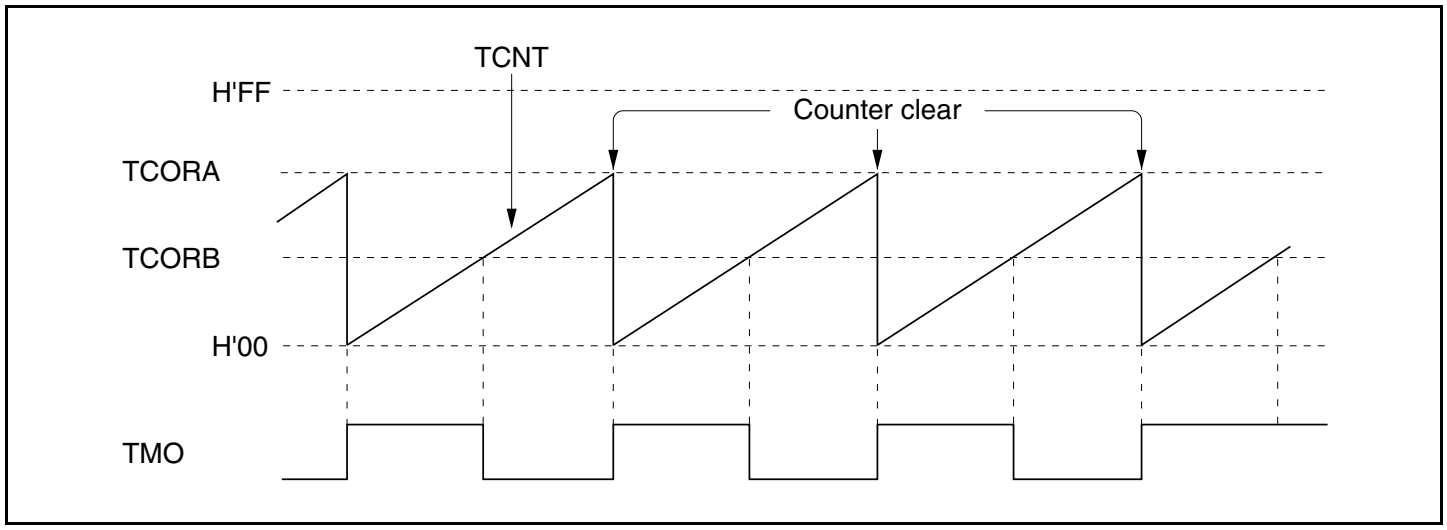


Figure 13.2 Example of Pulse Output

13.5 Operation Timing

13.5.1 TCNT Incrementation Timing

Figure 13.3 shows the count timing for internal clock input. Figure 13.4 shows the count timing for external clock signal. Note that the external clock pulse width must be at least 1.5 states for incrementation at a single edge, and at least 2.5 states for incrementation at both edges. The counter will not increment correctly if the pulse width is less than these values.

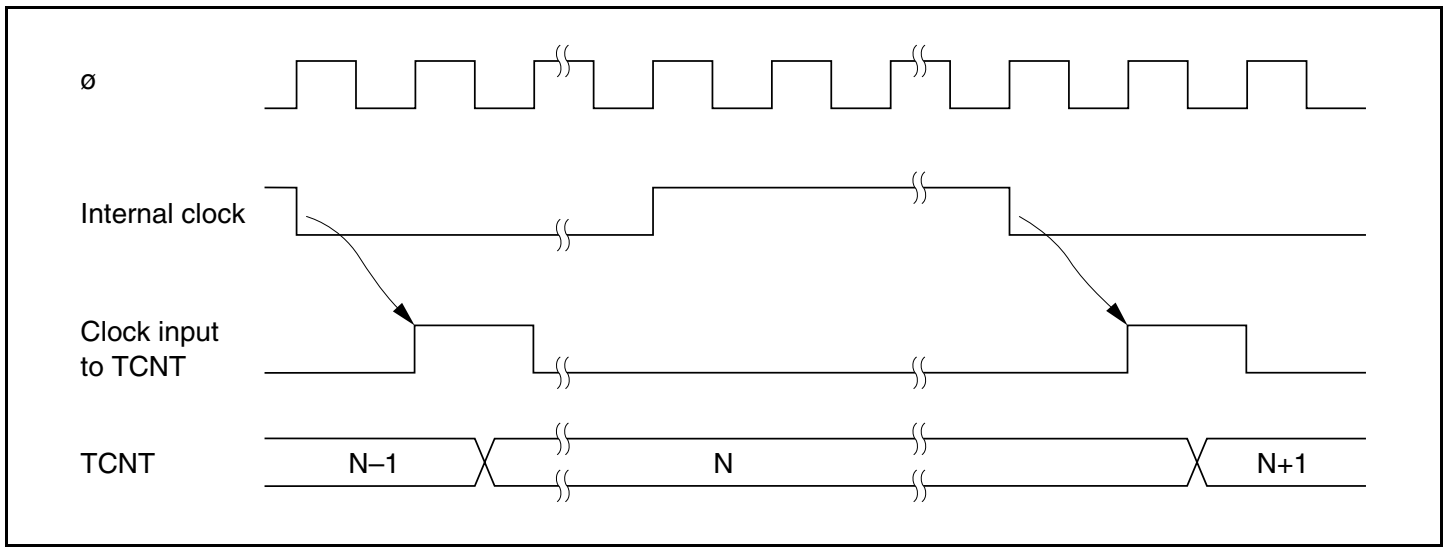


Figure 13.3 Count Timing for Internal Clock Input

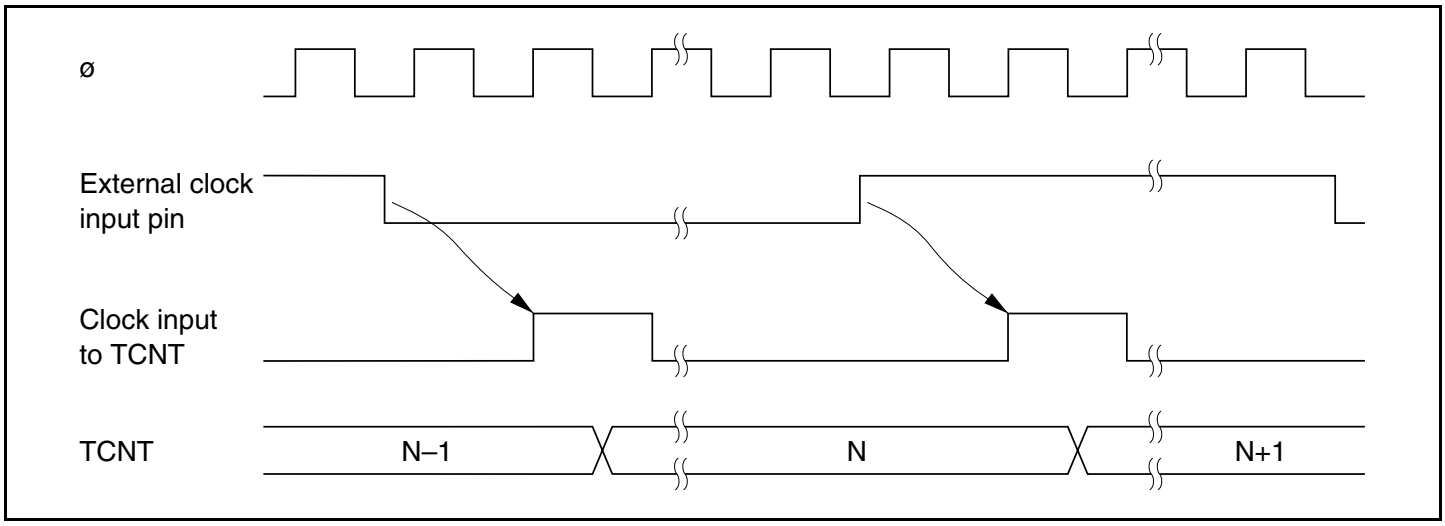


Figure 13.4 Count Timing for External Clock Input

13.5.2 Timing of CMFA and CMFB Setting when Compare-Match Occurs

The CMFA and CMFB flags in TCSR are set to 1 by a compare match signal generated when the TCOR and TCNT values match. The compare match signal is generated at the last state in which the match is true, just before the timer counter is updated. Therefore, when TCOR and TCNT match, the compare match signal is not generated until the next incrementation clock input. Figure 13.5 shows this timing.

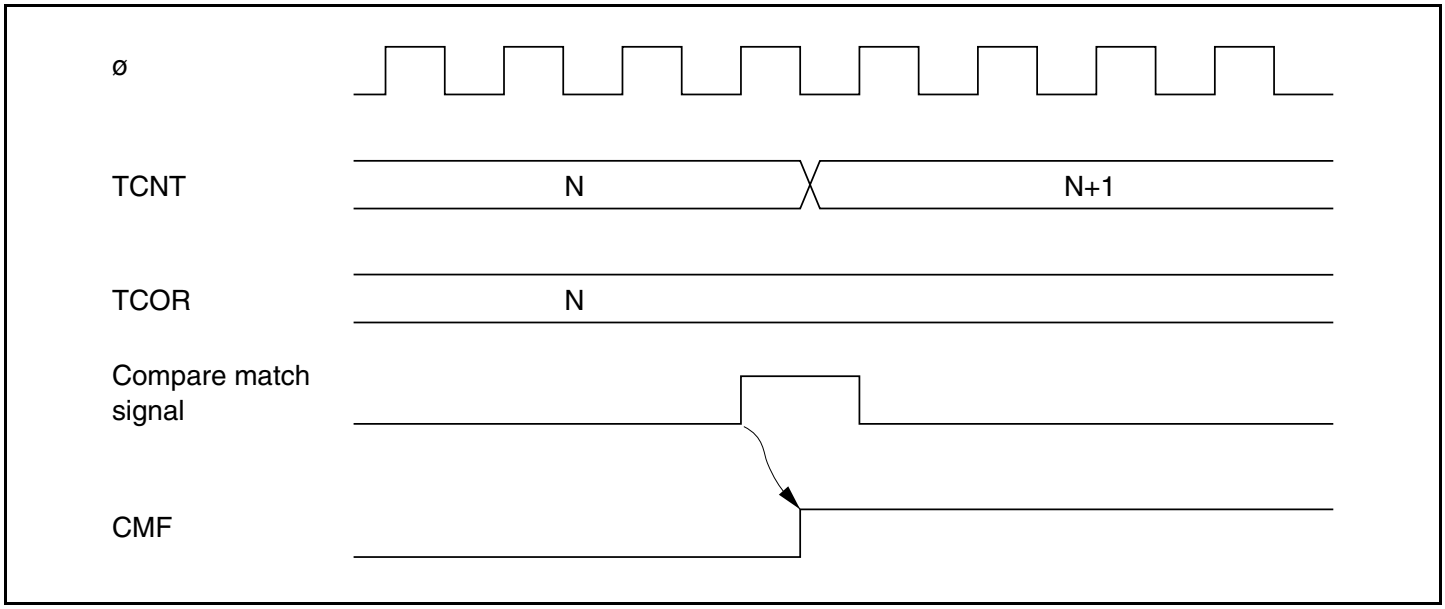


Figure 13.5 Timing of CMF Setting

13.5.3 Timing of Timer Output when Compare-Match Occurs

When compare match A or B occurs, the timer output changes as specified by bits OS3 to OS0 in TCSR.

Figure 13.6 shows the timing when the output is set to toggle at compare match A.

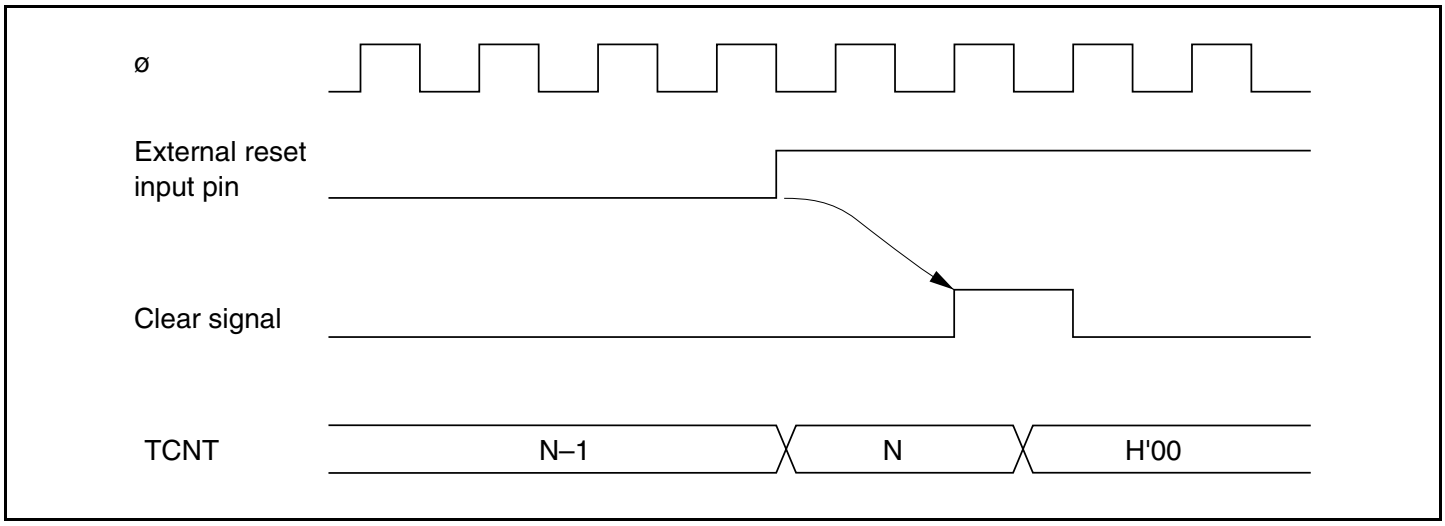


Figure 13.8 Timing of Clearance by External Reset

13.5.6 Timing of Overflow Flag (OVF) Setting

The OVF in TCSR is set to 1 when TCNT overflows (changes from H'FF to H'00). Figure 13.9 shows the timing of this operation.

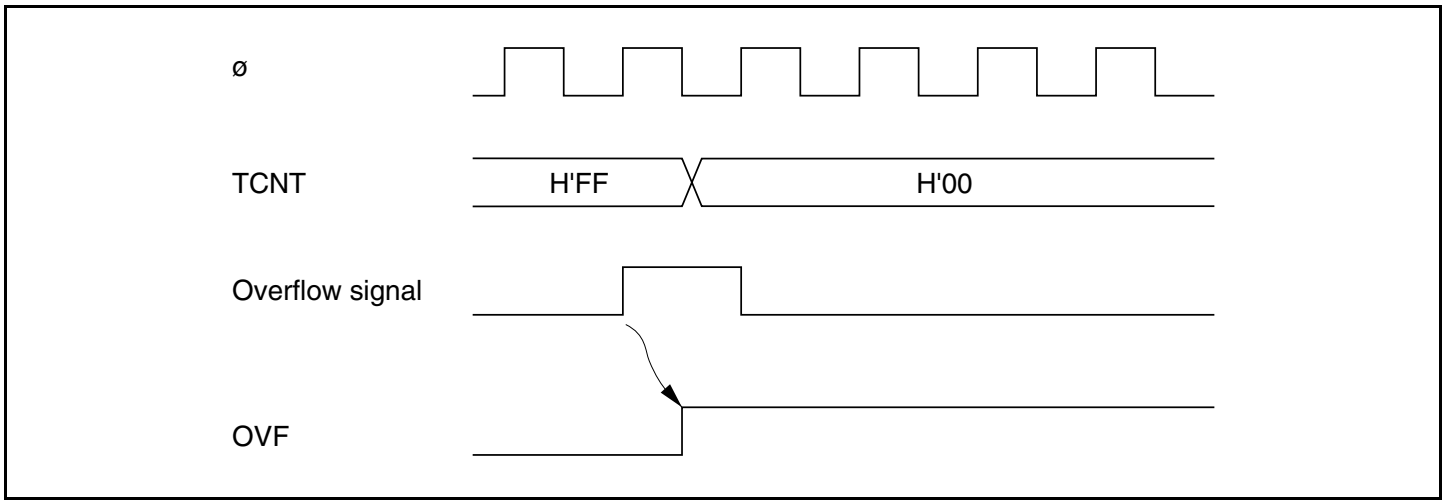


Figure 13.9 Timing of OVF Setting

13.6 Operation with Cascaded Connection

If bits CKS2 to CKS0 in either TCR_0 or TCR_1 are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, a single 16-bit timer could be used (16-bit counter mode) or compare matches of the 8-bit channel 0 could be counted by the timer of channel 1 (compare match count mode). In this case, the timer operates as below.

13.6.1 16-Bit Counter Mode

When bits CKS2 to CKS0 in TCR_0 are set to B'100, the timer functions as a single 16-bit timer with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bits.

[1] Setting of compare match flags

- The CMF flag in TCSR_0 is set to 1 when a 16-bit compare match event occurs.
- The CMF flag in TCSR_1 is set to 1 when a lower 8-bit compare match event occurs.

[2] Counter clear specification

- If the CCLR1 and CCLR0 bits in TCR_0 have been set for counter clear at compare match, the 16-bit counters (TCNT_0 and TCNT_1 together) are cleared when a 16-bit compare match event occurs. The 16-bit counters (TCNT0 and TCNT1 together) are cleared even if counter clear by the TMRI0 pin has also been set.
- The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bits cannot be cleared independently.

[3] Pin output

- Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR_0 is in accordance with the 16-bit compare match conditions.
- Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR_1 is in accordance with the lower 8-bit compare match conditions.

13.6.2 Compare Match Count Mode

When bits CKS2 to CKS0 in TCR_1 are B'100, TCNT_1 counts compare match A's for channel 0.

Channels 0 and 1 are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clear are in accordance with the settings for each channel.

13.7 Interrupts

13.7.1 Interrupt Sources and DTC Activation

There are three 8-bit timer interrupt sources: CMIA, CMIB, and OVI. Their relative priorities are shown in table 13.3. Each interrupt source is set as enabled or disabled by the corresponding interrupt enable bit in TCR or TCSR, and independent interrupt requests are sent for each to the interrupt controller. It is also possible to activate the DTC by means of CMIA and CMIB interrupts.

Table 13.3 8-Bit Timer Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority
CMIA0	TCORA_0 compare match	CMFA	Possible	High
CMIB0	TCORB_0 compare match	CMFB	Possible	↑
OVI0	TCNT_0 overflow	OVF	Not possible	Low
CMIA1	TCORA_1 compare match	CMFA	Possible	High
CMIB1	TCORB_1 compare match	CMFB	Possible	↑
OVI1	TCNT_1 overflow	OVF	Not possible	Low

13.7.2 A/D Converter Activation

The A/D converter can be activated only by TMR_0 compare match A.

If the ADTE bit in TCSR0 is set to 1 when the CMFA flag is set to 1 by the occurrence of TMR_0 compare match A, a request to start A/D conversion is sent to the A/D converter. If the 8-bit timer conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

13.8 Usage Notes

13.8.1 Contention between TCNT Write and Clear

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the clear takes priority, so that the counter is cleared and the write is not performed.

Figure 13.10 shows this operation.

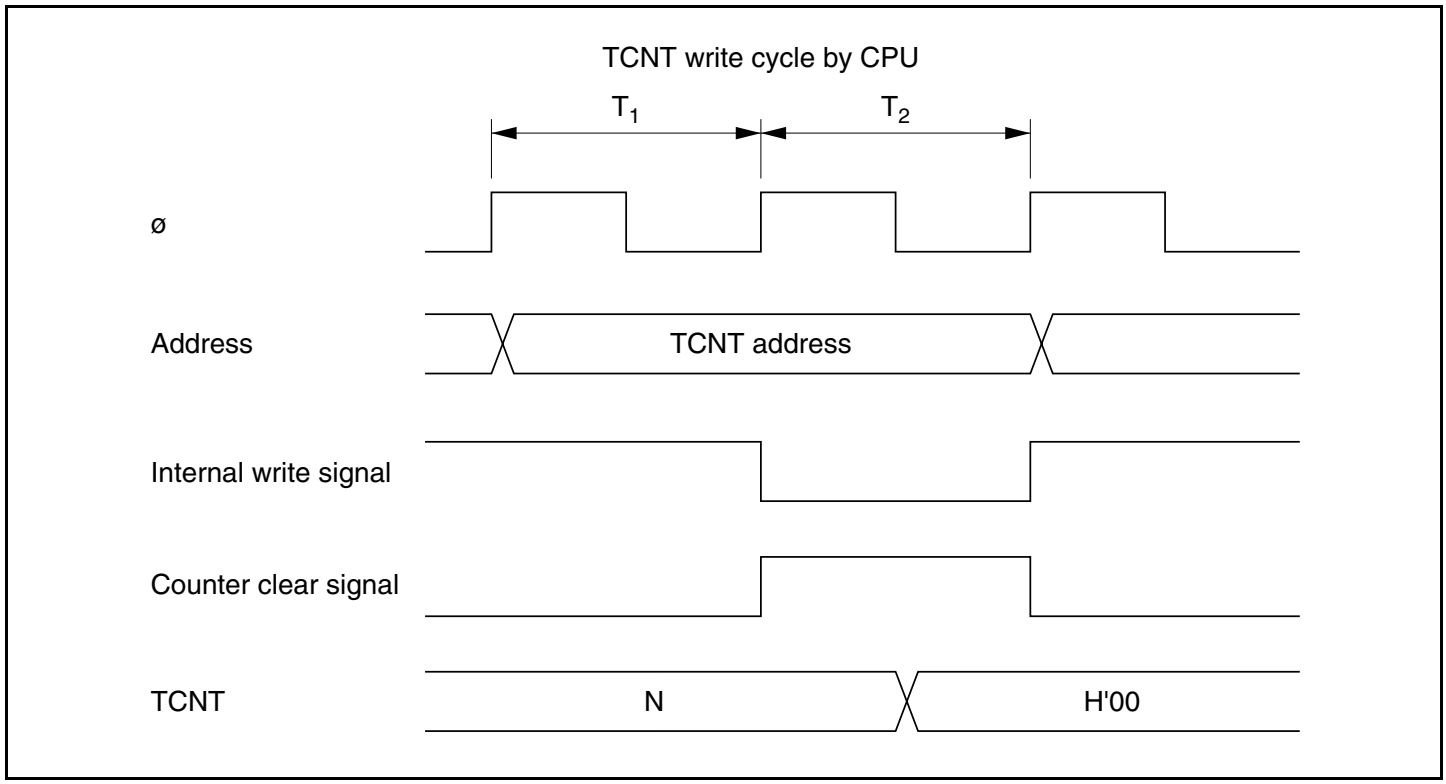


Figure 13.10 Contention between TCNT Write and Clear

13.8.2 Contention between TCNT Write and Increment

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the write takes priority and the counter is not incremented.

Figure 13.11 shows this operation.

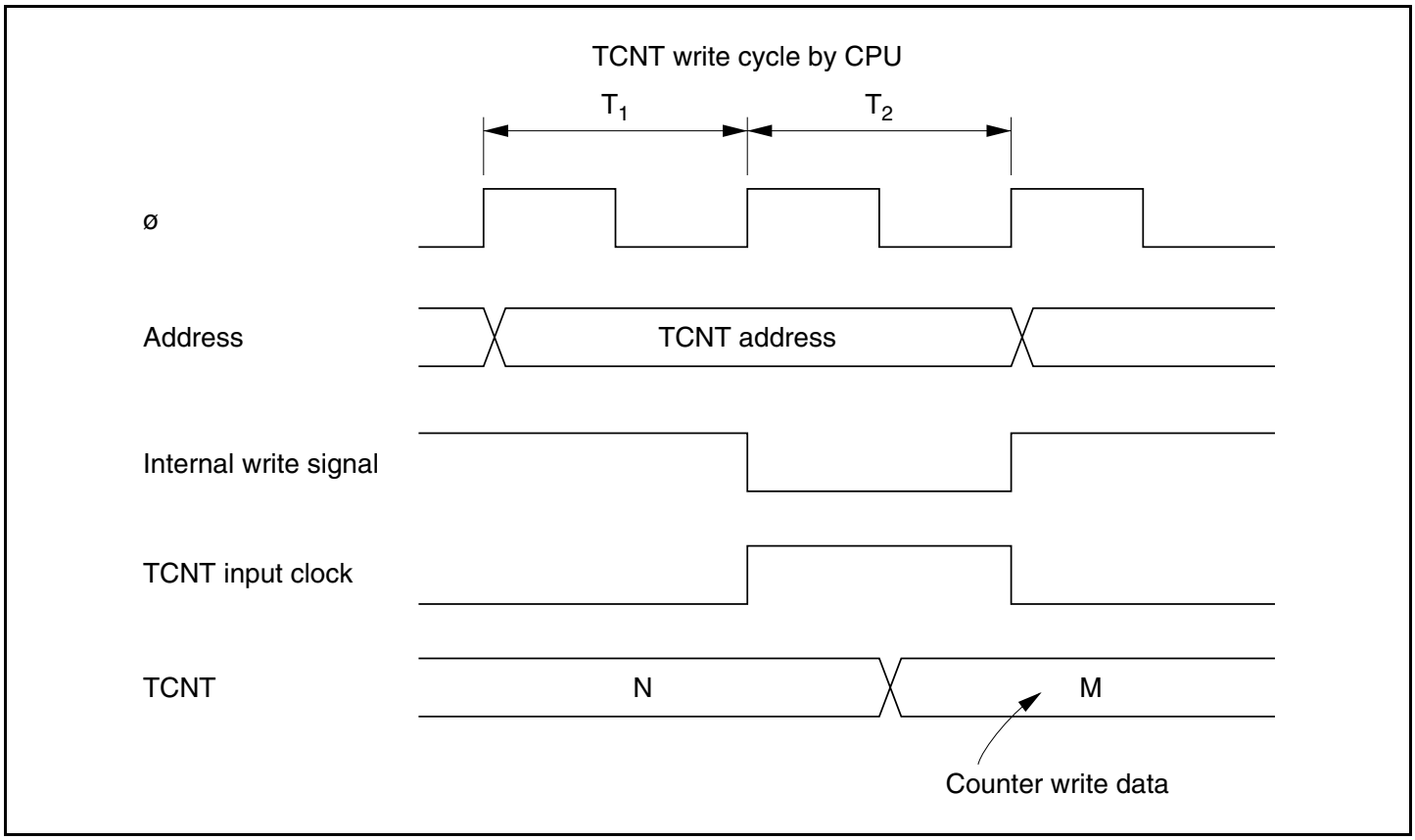


Figure 13.11 Contention between TCNT Write and Increment

13.8.3 Contention between TCOR Write and Compare Match

During the T_2 state of a TCOR write cycle, the TCOR write has priority and the compare match signal is inhibited even if a compare match event occurs as shown in figure 13.12.

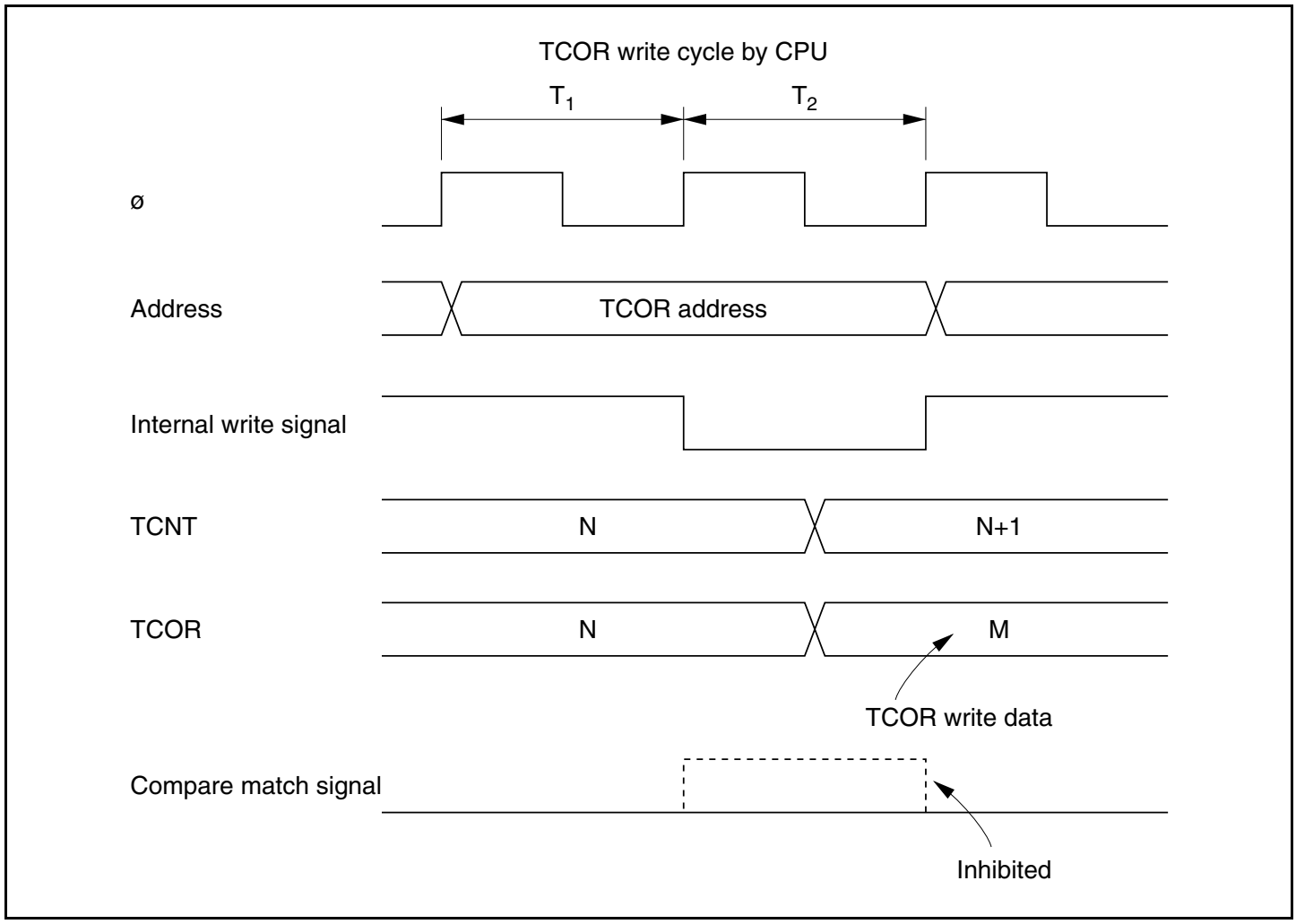


Figure 13.12 Contention between TCOR Write and Compare Match

13.8.4 Contention between Compare Matches A and B

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output statuses set for compare match A and compare match B, as shown in table 13.4.

Table 13.4 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1 output	
0 output	
No change	Low

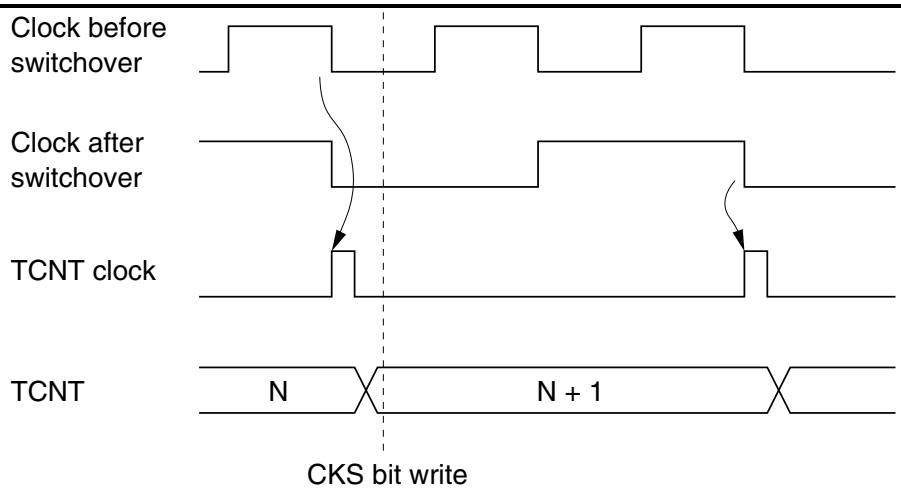
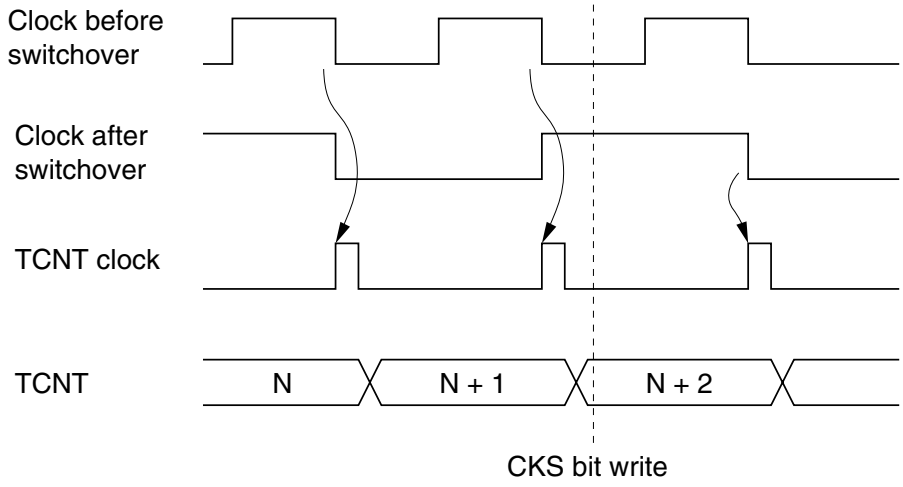
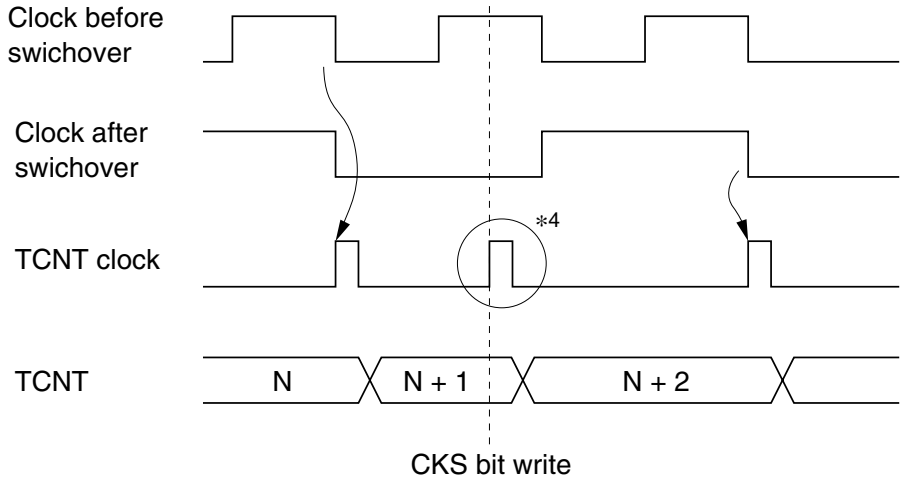
13.8.5 Switching of Internal Clocks and TCNT Operation

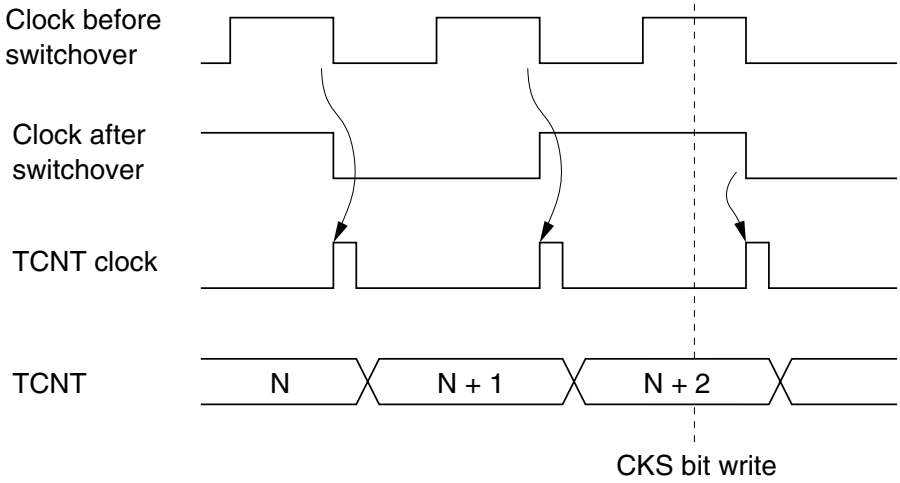
TCNT may increment erroneously when the internal clock is switched over. Table 13.5 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1 and CKS0 bits) and the TCNT operation.

When the TCNT clock is generated from an internal clock, the falling edge of the internal clock pulse is detected. If clock switching causes a change from high to low level, as shown in case 3 in table 13.5, a TCNT clock pulse is generated on the assumption that the switchover is a falling edge. This increments TCNT.

The erroneous incrementation can also happen when switching between internal and external clocks.

Table 13.5 Switching of Internal Clock and TCNT Operation

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Switching from low to low*1	
2	Switching from low to high*2	
3	Switching from high to low*3	

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
4	Switching from high to high	

Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated on the assumption that the switchover is a falling edge; TCNT is incremented.

13.8.6 Mode Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, input clocks for TCNT_0 and TCNT_1 are not generated, and the counter stops. Do not specify 16-bit counter and compare match count modes simultaneously.

Section 14 Watchdog Timer

The watchdog timer (WDT) is an 8-bit timer that outputs an overflow signal (WDTOVF) if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow.

At the same time, the WDT can also generate an internal reset signal.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

The block diagram of the WDT is shown in figure 14.1.

14.1 Features

- Selectable from eight counter input clocks
- Switchable between watchdog timer mode and interval timer mode

In watchdog timer mode

- If the counter overflows, the WDT outputs WDTOVF. It is possible to select whether or not the entire chip is reset at the same time.

In interval timer mode

- If the counter overflows, the WDT generates an interval timer interrupt (WOVI).

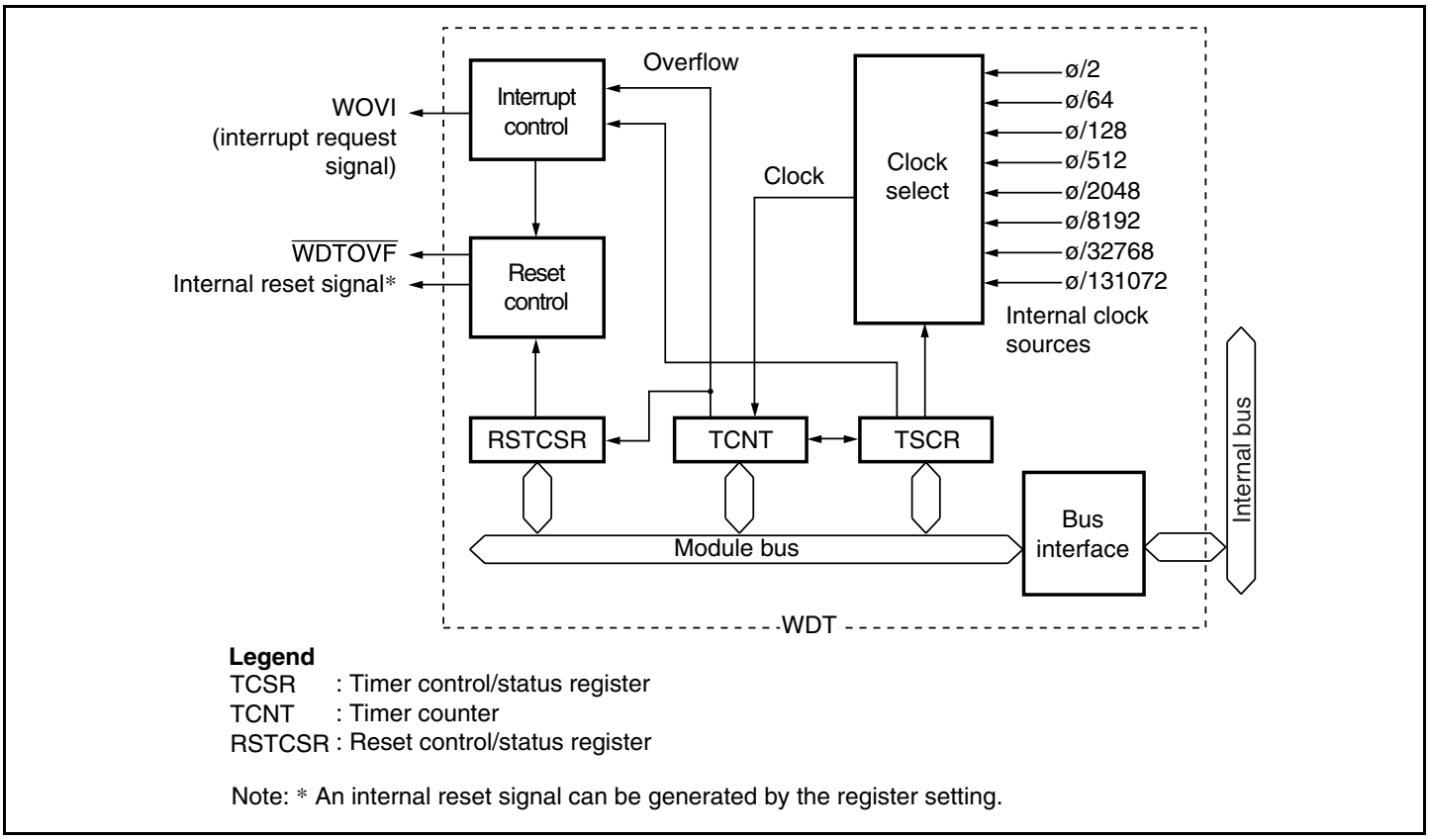


Figure 14.1 Block Diagram of WDT

14.2 Input/Output Pin

Table 14.1 shows the WDT pin configuration.

Table 14.1 Pin configuration

Name	Symbol	I/O	Function
Watchdog timer overflow	WDTOVF	Output	Outputs counter overflow signal in watchdog timer mode

14.3 Register Descriptions

The WDT has the following three registers. To prevent accidental overwriting, TCSR, TCNT, and RSTCSR have to be written to in a method different from normal registers. For details, refer to section 14.6.1, Notes on Register Access.

- Timer counter (TCNT)
- Timer control/status register (TCSR)
- Reset control/status register (RSTCSR)

14.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT is initialized to H'00 when the TME bit in TCSR is cleared to 0.

14.3.2 Timer Control/Status Register (TCSR)

TCSR selects the clock source to be input to TCNT, and the timer mode.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Overflow Flag Indicates that TCNT has overflowed in interval timer mode. Only a write of 0 is permitted, to clear the flag. [Setting condition] When TCNT overflows in interval timer mode (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing conditions] Cleared by reading TCSR when OVF = 1, then writing 0 to OVF
6	WT/IT	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. 0: Interval timer mode When TCNT overflows, an interval timer interrupt (WOVI) is requested. 1: Watchdog timer mode When TCNT overflows, the WDTOVF signal is output.
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.

Bit	Bit Name	Initial Value	R/W	Description
4	—	1	—	Reserved
3	—	1	—	These bits are always read as 1 and cannot be modified.
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	Selects the clock source to be input to TCNT. The overflow frequency for ø = 20 MHz is enclosed in parentheses.
0	CKS0	0	R/W	000: Clock ø/2 (frequency: 25.6 µs)
				001: Clock ø/64 (frequency: 819.2 µs)
				010: Clock ø/128 (frequency: 1.6 ms)
				011: Clock ø/512 (frequency: 6.6 ms)
				100: Clock ø/2048 (frequency: 26.2 ms)
				101: Clock ø/8192 (frequency: 104.9 ms)
				110: Clock ø/32768 (frequency: 419.4 ms)
				111: Clock ø/131072 (frequency: 1.68 s)

Note: Only a write of 0 is permitted, to clear the flag.

14.3.3 Reset Control/Status Register (RSTCSR)

RSTCSR controls the generation of the internal reset signal when TCNT overflows, and selects the type of internal reset signal. RSTCSR is initialized to H'1F by a reset signal from the RES pin, but not by the WDT internal reset signal caused by overflows.

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)*	Watchdog Timer Overflow Flag This bit is set when TCNT overflows in watchdog timer mode. This bit cannot be set in interval timer mode, and only 0 can be written. [Setting condition] Set when TCNT overflows (changed from H'FF to H'00) in watchdog timer mode [Clearing condition] Cleared by reading RSTCSR when WOVF = 1, and then writing 0 to WOVF
6	RSTE	0	R/W	Reset Enable Specifies whether or not a reset signal is generated in the chip if TCNT overflows during watchdog timer operation. 0: Reset signal is not generated even if TCNT overflows (Though this LSI is not reset, TCNT and TCSR in WDT are reset) 1: Reset signal is generated if TCNT overflows
5	—	0	R/W	Reserved Can be read and written, but does not affect operation.
4 to 0	—	1	—	Reserved These bits are always read as 1 and cannot be modified.

Note: Only a write of 0 is permitted, to clear the flag.

14.4 Operation

14.4.1 Watchdog Timer Mode

To use the WDT as a watchdog timer mode, set the WT/IT and TME bits in TCSR to 1.

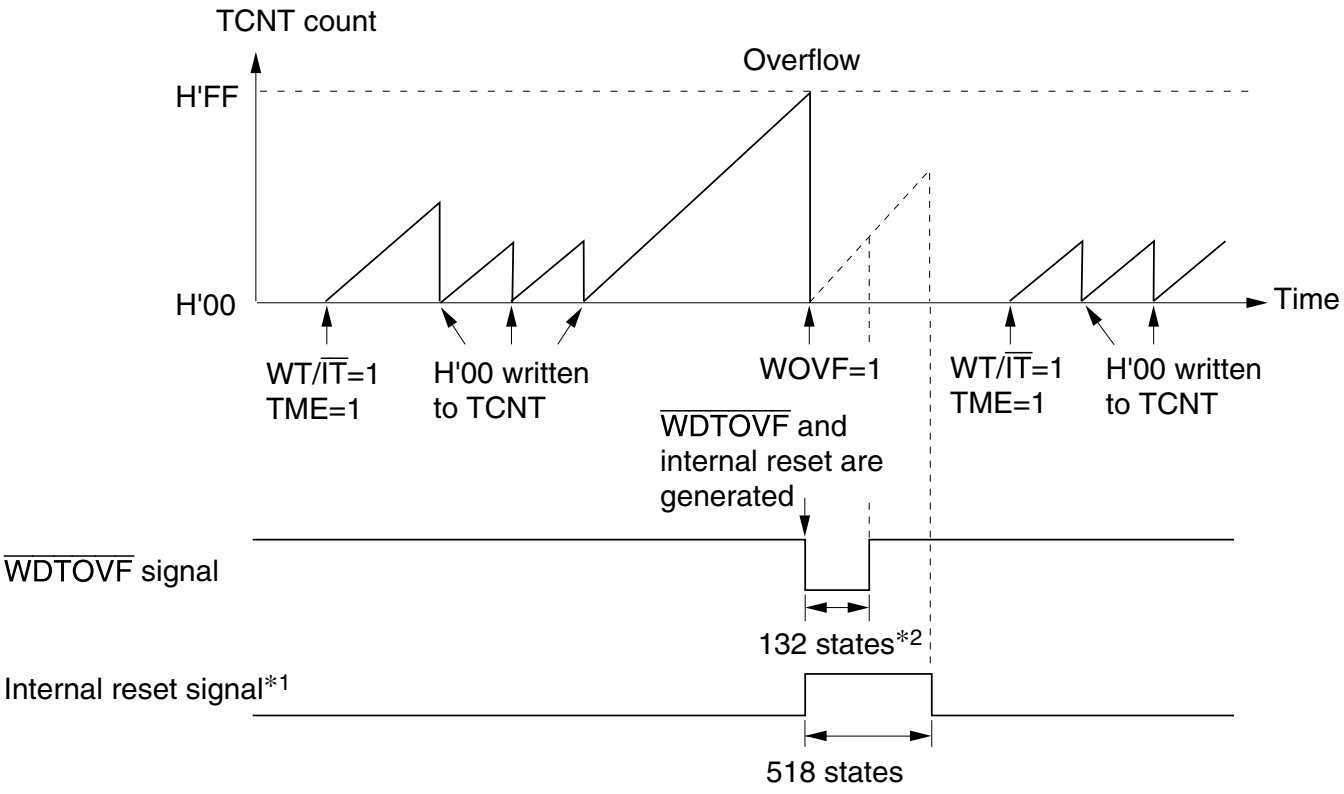
If TCNT overflows without being rewritten because of a system crash or other error, the WDTOVF signal is output.

This ensures that TCNT does not overflow while the system is operating normally. Software must prevent TCNT overflows by rewriting the TCNT value (normally be writing H'00) before overflow occurs. This WDTOVF signal can be used to reset the chip internally in watchdog timer mode.

If TCNT overflows when 1 is set in the RSTE bit in RSTCSR, a signal that resets this LSI internally is generated at the same time as the WDTOVF signal. If a reset caused by a signal input to the RES pin occurs at the same time as a reset caused by a WDT overflow, the RES pin reset has priority and the WOVF bit in RSTCSR is cleared to 0.

The WDTOVF signal is output for 132 states when RSTE = 1, and for 130 states when RSTE = 0. The internal reset signal is output for 518 states.

When TCNT overflows in watchdog timer mode, the WOVF bit in RSTCSR is set to 1. If TCNT overflows when 1 is set in the RSTE bit in RSTCSR, an internal reset signal is generated to the entire chip.



Legend

WT/$\overline{IT}$: Timer mode select bit
TME : Timer enable bit

Notes: 1. If TCNT overflows when the RSTE bit is set to 1, an internal reset signal is generated.
2. 130 states when the RSTE bit is cleared to 0.

Figure 14.2 Operation in Watchdog Timer Mode

14.4.2 Interval Timer Mode

To use the WDT as an interval timer, set the WT/IT bit to 0 and TME bit in TCSR to 1.

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is generated each time the TCNT overflows. Therefore, an interrupt can be generated at intervals.

When the TCNT overflows in interval timer mode, an interval timer interrupt (WOVI) is requested at the same time the OVF bit in the TCSR is set to 1.

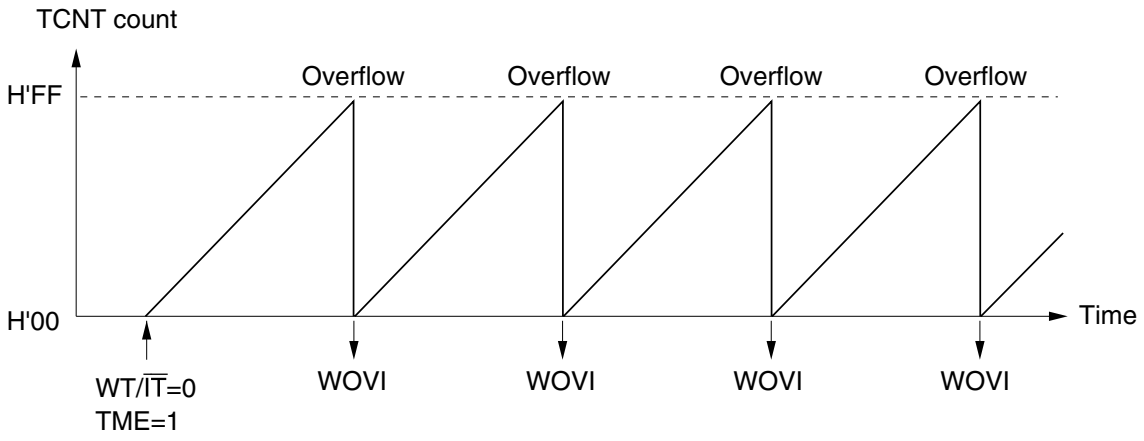


Figure 14.3 Operation in Interval Timer Mode

14.5 Interrupt Source

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. OVF must be cleared to 0 in the interrupt handling routine.

Table 14.2 WDT Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation
WOVI	TCNT overflow	OVF	Impossible

14.6 Usage Notes

14.6.1 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in being more difficult to write to. The procedures for writing to and reading these registers are given below.

Writing to TCNT, TCSR, and RSTCSR

TCNT and TCSR must be written to by a word transfer instruction. They cannot be written to by a byte transfer instruction.

TCNT and TCSR both have the same write address. Therefore, satisfy the relative condition shown in figure 14.4 to write to TCNT or TCSR. The transfer instruction writes the lower byte data to TCNT or TCSR according to the satisfied condition.

To write to RSTCSR, execute a word transfer instruction for address H'FFBE. A byte transfer instruction cannot perform writing to RSTCSR.

The method of writing 0 to the WOVF bit differs from that of writing to the RSTE bit. To write 0 to the WOVF bit, satisfy the lower condition shown in figure 14.4.

If satisfied, the transfer instruction clears the WOVF bit to 0, but has no effect on the RSTE bit. To write to the RSTE bit, satisfy the above condition shown in figure 14.4. If satisfied, the transfer instruction writes the value in bit 6 of the lower byte into the RSTE bit, but has no effect on the WOVF bit.

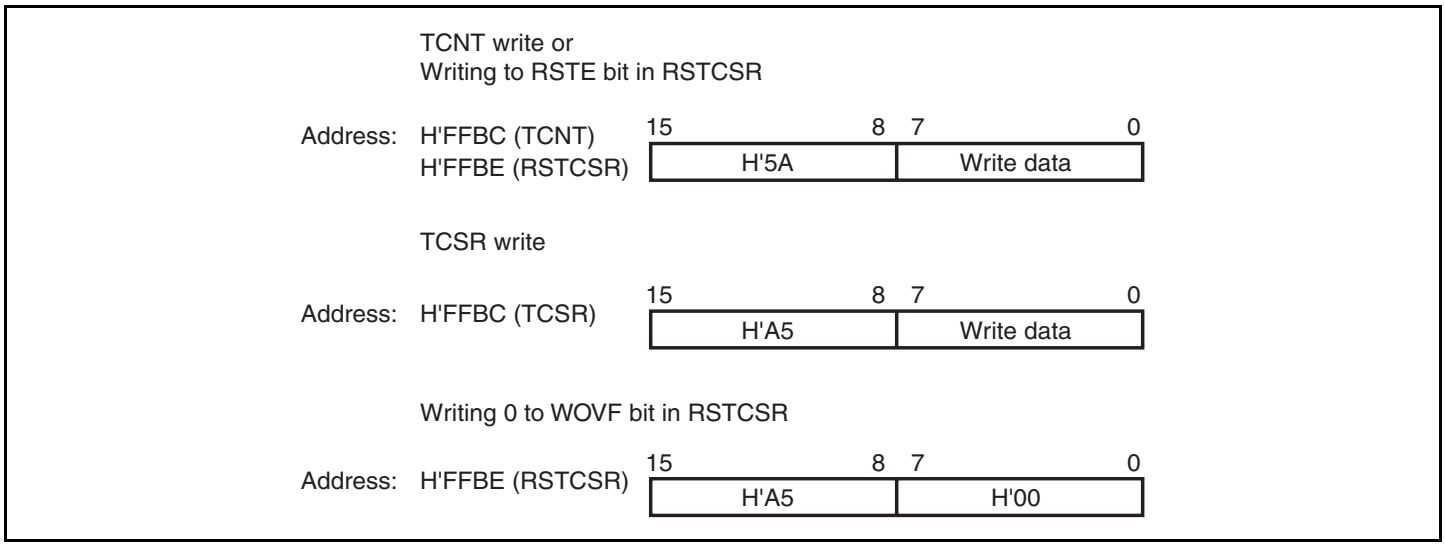


Figure 14.4 Writing to TCNT, TCSR, and RSTCSR

Reading TCNT, TCSR, and RSTCSR

These registers are read in the same way as other registers. The read addresses are H'FFBC for TCSR, H'FFBD for TCNT, and H'FFBF for RSTCSR.

14.6.2 Contention between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the next cycle after the T2 state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 14.5 shows this operation.

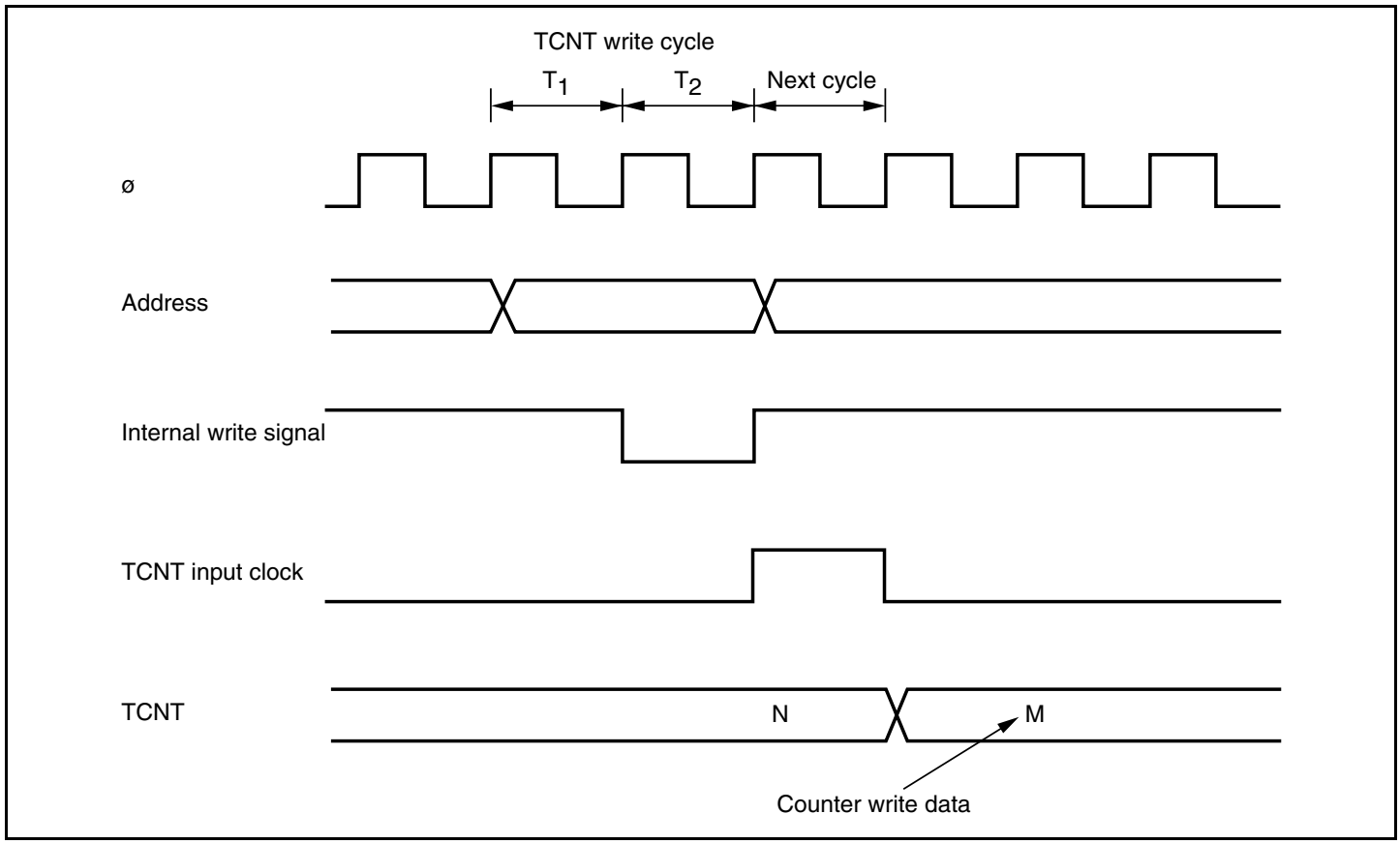


Figure 14.5 Contention between TCNT Write and Increment

14.6.3 Changing Value of CKS2 to CKS0

If bits CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before changing the value of bits CKS2 to CKS0.

14.6.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer, while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

14.6.5 Internal Reset in Watchdog Timer Mode

This LSI is not reset internally if TCNT overflows while the RSTE bit is cleared to 0 during watchdog timer mode operation, but TCNT and TCSR of the WDT are reset.

TCNT, TCSR, and RSTCR cannot be written to while the WDTOVF signal is low. Also note that a read of the WOVF flag is not recognized during this period. To clear the WOVF flag, therefore, read TCSR after the WDTOVF signal goes high, then write 0 to the WOVF flag.

14.6.6 System Reset by WDTOVF Signal

If the WDTOVF output signal is input to the RES pin, the chip will not be initialized correctly. Make sure that the WDTOVF signal is not input logically to the RES pin.

To reset the entire system by means of the WDTOVF signal, use the circuit shown in figure 14.6.

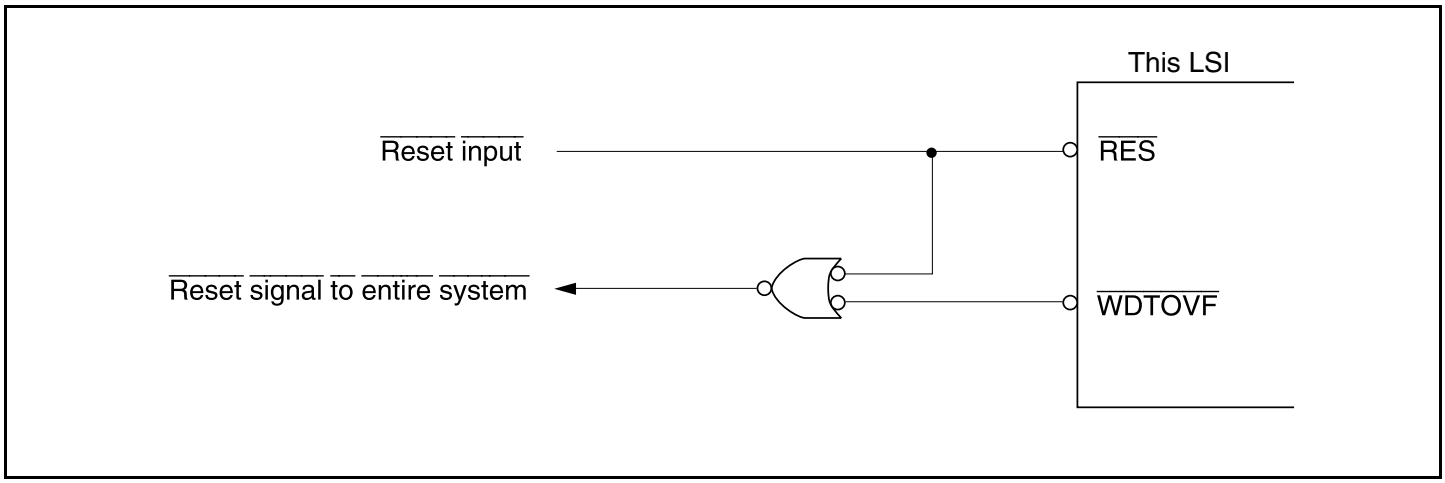


Figure 14.6 Circuit for System Reset by WDTOVF Signal (Example)

Section 15 Serial Communication Interface (SCI, IrDA)

This LSI has three independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function) in asynchronous mode. The SCI also supports an IC card (Smart Card) interface conforming to ISO/IEC 7816-3 (Identification Card) as an asynchronous serial communication interface extension function. One of the three SCI channels (SCI_0) can generate an IrDA communication waveform conforming to IrDA specification version 1.0.

Figure 15.1 shows a block diagram of the SCI.

15.1 Features

- Choice of asynchronous or clocked synchronous serial communication mode
- Full-duplex communication capability

  The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously. Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected

  External clock can be selected as a transfer clock source (except for in Smart Card interface mode).
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode 7-bit data)
- Four interrupt sources

  Four interrupt sources — transmit-end, transmit-data-empty, receive-data-full, and receive error — that can issue requests. The transmit-data-empty interrupt and receive data full interrupts can activate the data transfer controller (DTC) or DMA controller (DMAC).
- Module stop mode can be set

Asynchronous mode

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors
- Break detection: Break can be detected by reading the RxD pin level directly in case of a framing error

- Average transfer rate generator (only for H8S/2678R Series): The following transfer rate can be selected (SCI_2 only)
  115.152 or 460.606 kbps at 10.667 MHz operation
  115.196, 460.784 or 720 kbps at 16 MHz operation
  720 kbps at 32 MHz operation

Clocked Synchronous mode

- Data length: 8 bits
- Receive error detection: Overrun errors detected

Smart Card Interface

- Automatic transmission of error signal (parity error) in receive mode
- Error signal detection and automatic data retransmission in transmit mode
- Direct convention and inverse convention both supported

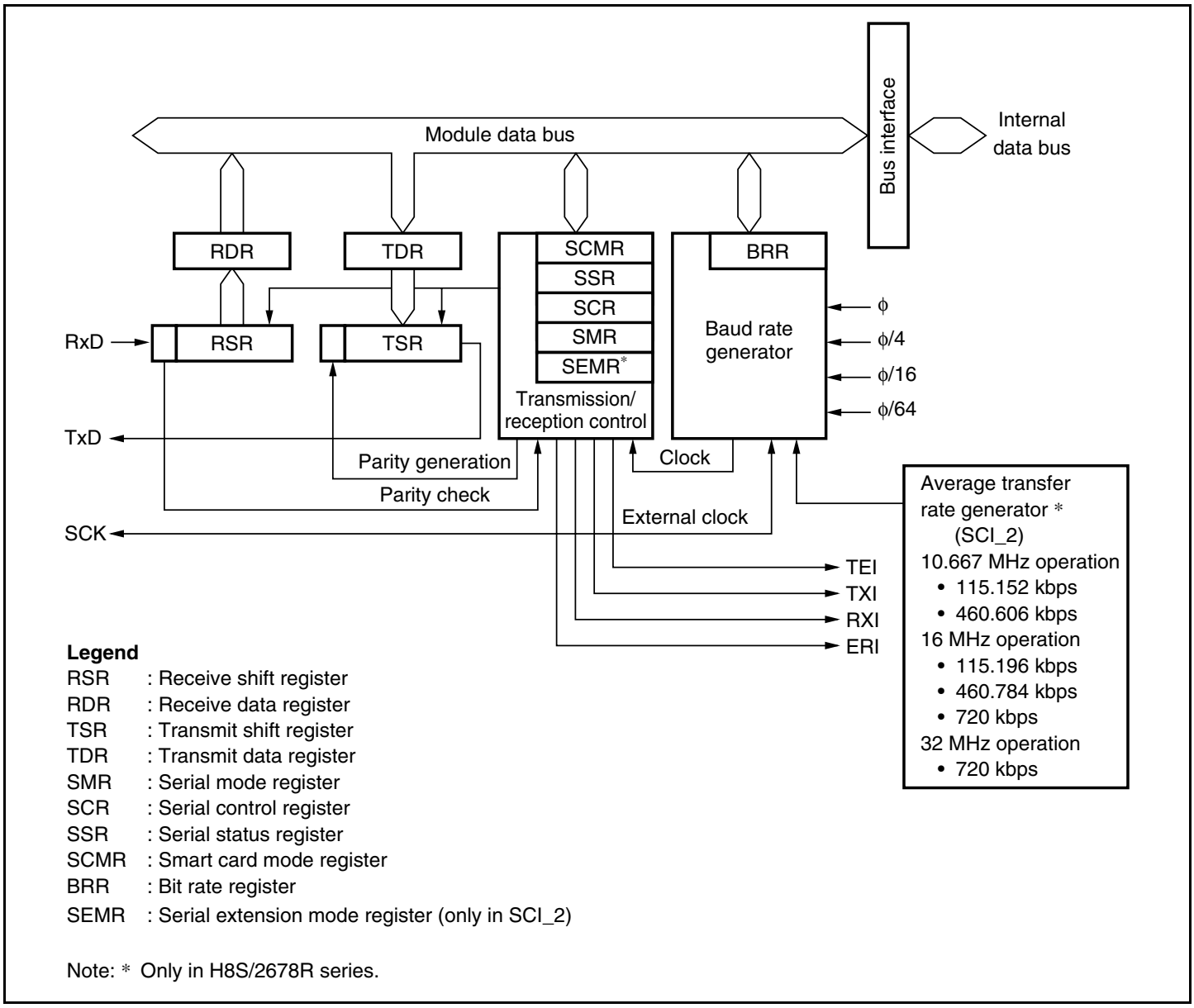


Figure 15.1 Block Diagram of SCI

15.2 Input/Output Pins

Table 15.1 shows the pin configuration of the serial communication interface.

Table 15.1 Pin Configuration

Channel	Pin Name*	I/O	Function
0	SCK0	I/O	Channel 0 clock input/output
	RxD0/IrRxD	Input	Channel 0 receive data input (normal/IrDA)
	TxD0/IrTxD	Output	Channel 0 transmit data output (normal/IrDA)
1	SCK1	I/O	Channel 1 clock input/output
	RxD1	Input	Channel 1 receive data input
	TxD1	Output	Channel 1 transmit data output
2	SCK2	I/O	Channel 2 clock input/output
	RxD2	Input	Channel 2 receive data input
	TxD2	Output	Channel 2 transmit data output

Note: Pin names SCK, RxD, and TxD are used in the text for all channels, omitting the channel designation.

15.3 Register Descriptions

The SCI has the following registers. The serial mode register (SMR), serial status register (SSR), and serial control register (SCR) are described separately for normal serial communication interface mode and Smart Card interface mode because their bit functions partially differ.

- Receive shift register_0 (RSR_0)
- Transmit shift register_0 (TSR_0)
- Receive data register_0 (RDR_0)
- Transmit data register_0 (TDR_0)
- Serial mode register_0 (SMR_0)
- Serial control register_0 (SCR_0)
- Serial status register_0 (SSR_0)
- Smart card mode register_0 (SCMR_0)
- Bit rate register_0 (BRR_0)
- IrDA control register_0 (IrCR_0)
- Receive shift register_1 (RSR_1)
- Transmit shift register_1 (TSR_1)
- Receive data register_1 (RDR_1)
- Transmit data register_1 (TDR_1)
- Serial mode register_1 (SMR_1)
- Serial control register_1 (SCR_1)
- Serial status register_1 (SSR_1)

- Smart card mode register_1 (SCMR_1)
- Bit rate register_1 (BRR_1)
- Receive shift register_2 (RSR_2)
- Transmit shift register_2 (TSR_2)
- Receive data register_2 (RDR_2)
- Transmit data register_2 (TDR_2)
- Serial mode register_2 (SMR_2)
- Serial control register_2 (SCR_2)
- Serial status register_2 (SSR_2)
- Smart card mode register_2 (SCMR_2)
- Bit rate register_2 (BRR_2)
- Serial extension mode register (SEMR)*

Note: Only in H8S/2678R Series.

15.3.1 Receive Shift Register (RSR)

RSR is a shift register used to receive serial data that is input to the RxD pin and convert it into parallel data. When one byte of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

15.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores receive data. When the SCI has received one byte of serial data, it transfers the received serial data from RSR to RDR where it is stored. After this, RSR is receive-enabled. Since RSR and RDR function as a double buffer in this way, enables continuous receive operations to be performed. After confirming that the RDRF bit in SSR is set to 1, read RDR for only once. RDR cannot be written to by the CPU.

15.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structures of TDR and TSR enable continuous serial transmission. If the next transmit data has already been written to TDR during serial transmission, the SCI transfers the written data to TSR to continue transmission. Although TDR can be read or written to by the CPU at all times, to achieve reliable serial transmission, write transmit data to TDR for only once after confirming that the TDRE bit in SSR is set to 1.

15.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, then sends the data to the TxD pin starting. TSR cannot be directly accessed by the CPU.

15.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the on-chip baud rate generator clock source.

Some bit functions of SMR differ in normal serial communication interface mode and Smart Card interface mode.

Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	C/A	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is fixed and the MSB (bit 7) of TDR is not transmitted in transmission. In clocked synchronous mode, a fixed data length of 8 bits is used.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. For a multiprocessor format, parity bit addition and checking are not performed regardless of the PE bit setting.
4	O/E	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous mode) Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked regardless of the STOP bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit character.
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynchronous mode) When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and O/E bit settings are invalid in multiprocessor mode.

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 clock source for the on-chip baud rate generator. 00: ø clock (n = 0) 01: ø/4 clock (n = 1) 10: ø/16 clock (n = 2) 11: ø/64 clock (n = 3) For the relation between the bit rate register setting and the baud rate, see section 15.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 15.3.9, Bit Rate Register (BRR)).

Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	GSM Mode When this bit is set to 1, the SCI operates in GSM mode. In GSM mode, the timing of the TEND setting is advanced by 11.0 etu (Elementary Time Unit: the time for transfer of one bit), and clock output control mode addition is performed. For details, refer to section 15.7.8, Clock Output Control.
6	BLK	0	R/W	When this bit is set to 1, the SCI operates in block transfer mode. For details on block transfer mode, refer to section 15.7.3, Block Transfer Mode.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. In Smart Card interface mode, this bit must be set to 1.
4	O/E	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity. For details on setting this bit in Smart Card interface mode, refer to section 15.7.2, Data Format (Except for Block Transfer Mode).

Bit	Bit Name	Initial Value	R/W	Description
3	BCP1	0	R/W	Basic Clock Pulse 1 and 0
2	BCP0	0	R/W	These bits select the number of basic clock periods in a 1-bit transfer interval on the Smart Card interface. 00: 32 clock (S = 32) 01: 64 clock (S = 64) 10: 372 clock (S = 372) 11: 256 clock (S = 256) For details, refer to section 15.7.4, Receive Data Sampling Timing and Reception Margin. S stands for the value of S in BRR (see section 15.3.9, Bit Rate Register (BRR)).
1	CKS1	0	R/W	Clock Select 1 and 0:
0	CKS0	0	R/W	These bits select the clock source for the on-chip baud rate generator. 00: ø clock (n = 0) 01: ø/4 clock (n = 1) 10: ø/16 clock (n = 2) 11: ø/64 clock (n = 3) For the relation between the bit rate register setting and the baud rate, see section 15.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 15.3.9, Bit Rate Register (BRR)).

15.3.6 Serial Control Register (SCR)

SCR performs enabling or disabling of SCI transfer operations and interrupt requests, and selection of the transfer/receive clock source. For details on interrupt requests, refer to section 15.9, Interrupts Sources. Some bit functions of SCR differ in normal serial communication interface mode and Smart Card interface mode.

Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, TXI interrupt request is enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled.
5	TE	0	R/W	Transmit Enable When this bit s set to 1, transmission is enabled.
4	RE	0	R/W	Receive Enable: When this bit is set to 1, reception is enabled.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and ORER status flags in SSR is prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, refer to section 15.5, Multiprocessor Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, TEI interrupt request is enabled.
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	Selects the clock source and SCK pin function. Asynchronous mode 00: On-chip baud rate generator SCK pin functions as I/O port 01: On-chip baud rate generator (Outputs a clock of the same frequency as the bit rate from the SCK pin.) 1X: External clock (Inputs a clock with a frequency 16 times the bit rate from the SCK pin.) Clocked synchronous mode 0X: Internal clock (SCK pin functions as clock output) 1X: External clock (SCK pin functions as clock input)

Note: X: Don’t care

Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, TXI interrupt request is enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode) Write 0 to this bit in Smart Card interface mode.
2	TEIE	0	R/W	Transmit End Interrupt Enable Write 0 to this bit in Smart Card interface mode.
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0		Enables or disables clock output from the SCK pin. The clock output can be dynamically switched in GSM mode. For details, refer to section 15.7.8, Clock Output Control. When the GM bit in SMR is 0: 00: Output disabled (SCK pin can be used as an I/O port pin) 01: Clock output 1X: Reserved When the GM bit in SMR is 1: 00: Output fixed low 01: Clock output 10: Output fixed high 11: Clock output

Note: X: Don’t care

15.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER; they can only be cleared. Some bit functions of SSR differ in normal serial communication interface mode and Smart Card interface mode.

Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt request and transfers data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or DTC is activated by an RXI interrupt and transferred data from RDR The RDRF flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W)*	Overrun Error [Setting condition] • When the next serial reception is completed while RDRF = 1 [Clearing condition] • When 0 is written to ORER after reading ORER = 1
4	FER	0	R/(W)*	Framing Error [Setting condition] • When the stop bit is 0 [Clearing condition] • When 0 is written to FER after reading FER = 1 In 2-stop-bit mode, only the first stop bit is checked.
3	PER	0	R/(W)*	Parity Error [Setting condition] • When a parity error is detected during reception [Clearing condition] • When 0 is written to PER after reading PER = 1
2	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR
1	MPB	0	R	Multiprocessor Bit MPB stores the multiprocessor bit in the receive data. When the RE bit in SCR is cleared to 0 its previous state is retained.

Bit	Bit Name	Initial Value	R/W	Description
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT sets the multiprocessor bit to be added to the transmit data.

Note: Only 0 can be written, to clear the flag.

Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt request and transfers data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or DTC is activated by an RXI interrupt and transferred data from RDR The RDRF flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W)*	Overrun Error [Setting condition] • When the next serial reception is completed while RDRF = 1 [Clearing condition] • When 0 is written to ORER after reading ORER = 1
4	ERS	0	R/(W)*	Error Signal Status [Setting condition] • When the low level of the error signal is sampled [Clearing conditions] • When 0 is written to ERS after reading ERS = 1
3	PER	0	R/(W)*	Parity Error [Setting condition] • When a parity error is detected during reception [Clearing condition] • When 0 is written to PER after reading PER = 1
2	TEND	1	R	Transmit End This bit is set to 1 when no error signal has been sent back from the receiving end and the next transmit data is ready to be transferred to TDR. [Setting conditions] • When the TE bit in SCR is 0 and the ERS bit is also 0 • If the ERS bit is 0 and the TDRE bit is 1 after the specified interval after transmission of 1-byte data Timing to set this bit differs according to the register settings. GM = 0, BLK = 0: 2.5 etu after transmission GM = 0, BLK = 1: 1.5 etu after transmission GM = 1, BLK = 0: 1.0 etu after transmission GM = 1, BLK = 1: 1.0 etu after transmission

Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	[Clearing conditions] • When 0 is written to TEND after reading TEND = 1 • When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR
1	MPB	0	R	Multiprocessor Bit This bit is not used in Smart Card interface mode.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in Smart Card interface mode.

Note: Only 0 can be written, to clear the flag.

15.3.8 Smart Card Mode Register (SCMR)

SCMR selects Smart Card interface mode and its format.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	1	—	Reserved These bits are always read as 1.
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the serial/parallel conversion format. 0: LSB-first in transfer 1: MSB-first in transfer The bit setting is valid only when the transfer data format is 8 bits. For 7-bit data, LSB-first is fixed.
2	SINV	0	R/W	Smart Card Data Invert Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit. To invert the parity bit, invert the O/E bit in SMR. 0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR. 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR.
1	—	1	—	Reserved This bit is always read as 1.

Bit	Bit Name	Initial Value	R/W	Description
0	SMIF	0	R/W	Smart Card Interface Mode Select This bit is set to 1 to make the SCI operate in Smart Card interface mode. 0: Normal asynchronous mode or clocked synchronous mode 1: Smart card interface mode

15.3.9 Bit Rate Register (BRR)

BRR is an 8-bit register that adjusts the bit rate. As the SCI performs baud rate generator control independently for each channel, different bit rates can be set for each channel. Table 15.2 shows the relationships between the N setting in BRR and bit rate B for normal asynchronous mode, clocked synchronous mode, and Smart Card interface mode. The initial value of BRR is H'FF, and it can be read or written to by the CPU at all times.

Table 15.2 Relationships between N Setting in BRR and Bit Rate B

Mode	Bit Rate	Error
Asynchronous Mode	$B = \frac{ø \times 10^6}{64 \times 2^{2n-1} \times (N + 1)}$	$\text{Error (\%)} = \{ \frac{ø \times 10^6}{B \times 64 \times 2^{2n-1} \times (N + 1)} - 1 \} \times 100$
Clocked Synchronous Mode	$B = \frac{ø \times 10^6}{8 \times 2^{2n-1} \times (N + 1)}$	
Smart Card Interface Mode	$B = \frac{ø \times 10^6}{S \times 2^{2n-1} \times (N + 1)}$	$\text{Error (\%)} = \{ \frac{ø \times 10^6}{B \times S \times 2^{2n-1} \times (N + 1)} - 1 \} \times 100$

Note: B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \le N \le 255$)
ø: Operating frequency (MHz)
n and S: Determined by the SMR settings shown in the following tables.

SMR Setting		
CKS1	CKS0	n
0	0	0
0	1	1
1	0	2
1	1	3

SMR Setting		
BCP1	BCP0	S
0	0	32
0	1	64
1	0	372
1	1	256

Table 15.3 shows sample N settings in BRR in normal asynchronous mode. Table 15.4 shows the maximum bit rate for each frequency in normal asynchronous mode. Table 15.6 shows sample N settings in BRR in clocked synchronous mode. Table 15.8 shows sample N settings in BRR in Smart Card interface mode. In Smart Card interface mode, S (the number of basic clock periods in a 1-bit transfer interval) can be selected. For details, refer to section 15.7.4, Receive Data Sampling Timing and Reception Margin. Tables 15.5 and 15.7 show the maximum bit rates with external clock input.

Table 15.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (1)

	Operating Frequency ø (MHz)											
	2			2.097152			2.4576			3		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	141	0.03	1	148	–0.04	1	174	–0.26	1	212	0.03
150	1	103	0.16	1	108	0.21	1	127	0.00	1	155	0.16
300	0	207	0.16	0	217	0.21	0	255	0.00	1	77	0.16
600	0	103	0.16	0	108	0.21	0	127	0.00	0	155	0.16
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77	0.16
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38	0.16
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19	–2.34
9600	—	—	—	0	6	–2.48	0	7	0.00	0	9	–2.34
19200	—	—	—	—	—	—	0	3	0.00	0	4	–2.34
31250	0	1	0.00	—	—	—	—	—	—	0	2	0.00
38400	—	—	—	—	—	—	0	1	0.00	—	—	—

Table 15.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (2)

	Operating Frequency ø (MHz)											
	3.6864			4			4.9152			5		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
300	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
19200	0	5	0.00	—	—	—	0	7	0.00	0	7	1.73
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4	0.00
38400	0	2	0.00	—	—	—	0	3	0.00	0	3	1.73

	Operating Frequency ø (MHz)											
	6			6.144			7.3728			8		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	106	–0.44	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	1	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	0.00	0	5	2.40	—	—	—	0	7	0.00
38400	0	4	–2.34	0	4	0.00	0	5	0.00	—	—	—

Table 15.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (3)

	Operating Frequency ø (MHz)											
	9.8304			10			12			12.288		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

	Operating Frequency ø (MHz)											
	14			14.7456			16			17.2032		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	64	0.70	3	70	0.03	3	75	0.48
150	2	181	0.16	2	191	0.00	2	207	0.16	2	223	0.00
300	2	90	0.16	2	95	0.00	2	103	0.16	2	111	0.00
600	1	181	0.16	1	191	0.00	1	207	0.16	1	223	0.00
1200	1	90	0.16	1	95	0.00	1	103	0.16	1	111	0.00
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	223	0.00
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	111	0.00
9600	0	45	–0.93	0	47	0.00	0	51	0.16	0	55	0.00
19200	0	22	–0.93	0	23	0.00	0	25	0.16	0	27	0.00
31250	0	13	0.00	0	14	–1.70	0	15	0.00	0	16	1.20
38400	—	—	—	0	11	0.00	0	12	0.16	0	13	0.00

Table 15.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (4)

	Operating Frequency ø (MHz)											
	18			19.6608			20			25		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	79	–0.12	3	86	0.31	3	88	–0.25	3	110	–0.02
150	2	233	0.16	2	255	0.00	3	64	0.16	3	80	–0.47
300	2	116	0.16	2	127	0.00	2	129	0.16	2	162	0.15
600	1	233	0.16	1	255	0.00	2	64	0.16	2	80	–0.47
1200	1	116	0.16	1	127	0.00	1	129	0.16	1	162	0.15
2400	0	233	0.16	0	255	0.00	1	64	0.16	1	80	–0.47
4800	0	116	0.16	0	127	0.00	0	129	0.16	0	162	0.15
9600	0	58	–0.69	0	63	0.00	0	64	0.16	0	80	–0.47
19200	0	28	1.02	0	31	0.00	0	32	–1.36	0	40	–0.76
31250	0	17	0.00	0	19	–1.70	0	19	0.00	0	24	0.00
38400	0	14	–2.34	0	15	0.00	0	15	1.73	0	19	1.73

	Operating Frequency ø (MHz)					
	30			33		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)
110	3	132	0.13	3	145	0.33
150	3	97	–0.35	3	106	0.39
300	2	194	0.16	2	214	–0.07
600	2	97	–0.35	2	106	0.39
1200	1	194	0.16	1	214	–0.07
2400	1	97	–0.35	1	106	0.39
4800	0	194	0.16	0	214	–0.07
9600	0	97	–0.35	0	106	0.39
19200	0	48	–0.35	0	53	–0.54
31250	0	29	0	0	32	0
38400	0	23	1.73	0	26	–0.54

Table 15.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ø (MHz)	Maximum Bit Rate (bit/s)	n	N	ø (MHz)	Maximum Bit Rate (bit/s)	n	N
2	62500	0	0	10	312500	0	0
2.097152	65536	0	0	12	375000	0	0
2.4576	76800	0	0	12.288	384000	0	0
3	93750	0	0	14	437500	0	0
3.6864	115200	0	0	14.7456	460800	0	0
4	125000	0	0	16	500000	0	0
4.9152	153600	0	0	17.2032	537600	0	0
5	156250	0	0	18	562500	0	0
6	187500	0	0	19.6608	614400	0	0
6.144	192000	0	0	20	625000	0	0
7.3728	230400	0	0	25	781250	0	0
8	250000	0	0	30	937500	0	0
9.8304	307200	0	0	33	1031250	0	0

Table 15.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ø (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	ø (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
2	0.5000	31250	10	2.5000	156250
2.097152	0.5243	32768	12	3.0000	187500
2.4576	0.6144	38400	12.288	3.0720	192000
3	0.7500	46875	14	3.5000	218750
3.6864	0.9216	57600	14.7456	3.6864	230400
4	1.0000	62500	16	4.0000	250000
4.9152	1.2288	76800	17.2032	4.3008	268800
5	1.2500	78125	18	4.5000	281250
6	1.5000	93750	19.6608	4.9152	307200
6.144	1.5360	96000	20	5.0000	312500
7.3728	1.8432	115200	25	6.2500	390625
8	2.0000	125000	30	7.5000	468750
9.8304	2.4576	153600	33	8.2500	515625

Table 15.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate	Operating Frequency ø (MHz)													
	2		4		8		10		16		20		25	
(bit/s)	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—										
250	2	124	2	249	3	124	—	—	3	249				
500	1	249	2	124	2	249	—	—	3	124	—	—		
1 k	1	124	1	249	2	124	—	—	2	249	—	—	3	97
2.5 k	0	199	1	99	1	199	1	249	2	99	2	124	2	155
5 k	0	99	0	199	1	99	1	124	1	199	1	249	2	77
10 k	0	49	0	99	0	199	0	249	1	99	1	124	1	155
25 k	0	19	0	39	0	79	0	99	0	159	0	199	0	249
50 k	0	9	0	19	0	39	0	49	0	79	0	99	0	124
100 k	0	4	0	9	0	19	0	24	0	39	0	49	0	62
250 k	0	1	0	3	0	7	0	9	0	15	0	19	0	24
500 k	0	0*	0	1	0	3	0	4	0	7	0	9	—	—
1 M			0	0*	0	1			0	3	0	4	—	—
2.5 M							0	0*			0	1	—	—
5 M											0	0*	—	—

Bit Rate (bit/s)	Operating Frequency ø (MHz) 30 n	30 N	33 n	33 N
110				
250				
500	3	233		
1 k	3	116	3	128
2.5 k	2	187	2	205
5 k	2	93	2	102
10 k	1	187	1	205
25 k	1	74	1	82
50 k	0	149	0	164
100 k	0	74	0	82
250 k	0	29	0	32
500 k	0	14	—	—
1 M	—	—	—	—
2.5 M	0	2	—	—
5 M	—	—	—	—

Legend

Blank: Cannot be set.

—: Can be set, but there will be a degree of error.

*: Continuous transfer is not possible.

Table 15.7 Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode)

ø (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	ø (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
2	0.3333	333333.3	16	2.6667	2666666.7
4	0.6667	666666.7	18	3.0000	3000000.0
6	1.0000	1000000.0	20	3.3333	3333333.3
8	1.3333	1333333.3	25	4.1667	4166666.7
10	1.6667	1666666.7	30	5.0000	5000000.0
12	2.0000	2000000.0	33	5.5000	5500000.0
14	2.3333	2333333.3			

Table 15.8 Examples of Bit Rate for Various BRR Settings (Smart Card Interface Mode) (when n = 0 and S = 372)

	Operating Frequency ø (MHz)											
	7.1424			10.00			10.7136			13.00		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	1	1	0.00	0	1	30	0	1	25	0	1	8.99

	Operating Frequency ø (MHz)											
	14.2848			16.00			18.00			20.00		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	15.99	0	2	6.60

	Operating Frequency ø (MHz)								
	25.00			30.00			33.00		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	3	12.49	0	3	5.01	0	4	7.59

Table 15.9 Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode) (when S = 372)

ø (MHz)	Maximum Bit Rate (bit/s)	n	N	ø (MHz)	Maximum Bit Rate (bit/s)	n	N
7.1424	9600	0	0	18.00	24194	0	0
10.00	13441	0	0	20.00	26882	0	0
10.7136	14400	0	0	25.00	33602	0	0
13.00	17473	0	0	30.00	40323	0	0
14.2848	19200	0	0	33.00	44355	0	0
16.00	21505	0	0				

15.3.10 IrDA Control Register (IrCR)

IrCR selects the function of SCI_0.

Bit	Bit Name	Initial Value	R/W	Description

15.3.11 Serial Extension Mode Register (SEMR)

SEMR selects the clock source in asynchronous mode. The basic clock can be automatically set by selecting the average transfer rate. SEMR is supported only in SCI_2 of the H8S/2678R Series.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved If these bits are read, an undefined value will be returned and cannot be modified.
3	ABCS	0	R/W	Asynchronous basic clock selection (valid only in asynchronous mode) Selects the basic clock for 1-bit period in asynchronous mode. 0: Operates on a basic clock with a frequency of 16 times the transfer rate. 1: Operates on a basic clock with a frequency of 8 times the transfer rate.

Bit	Bit Name	Initial Value	R/W	Description
2	ACS2	0	R/W	Asynchronous clock source selection (valid when CKS1 = 1 in asynchronous mode)
1	ACS1	0	R/W	
0	ACS0	0	R/W	Selects the clock source for the average transfer rate. The basic clock can be automatically set by selecting the average transfer rate in spite of the value of ABCS. 000: External clock input 001: Selects 115.152 kbps which is the average transfer rate dedicated for ϕ= 10.667 MHz. (Operates on a basic clock with a frequency of 16 times the transfer rate.) 010: Selects 460.606 kbps which is the average transfer rate dedicated for ϕ= 10.667 MHz. (Operates on a basic clock with a frequency of 8 times the transfer rate.) 011: Selects 720 kbps which is the average transfer rate dedicated for ϕ = 32 MHz. (Operates on a basic clock with a frequency of 16 times the transfer rate.) 100: Reserved 101: Selects 115.196 kbps which is the average transfer rate dedicated for ϕ = 16 MHz (Operates on a basic clock with a frequency of 16 times the transfer rate.) 110: Selects 460.784 kbps which is the average transfer rate dedicated for ϕ = 16 MHz (Operates on a basic clock with a frequency of 16 times the transfer rate.) 111: Selects 720 kbps which is the average transfer rate dedicated for ϕ = 16 MHz (Operates on a basic clock with a frequency of 8 times the transfer rate.) Note that the average transfer rate does not correspond to the frequency other than 10.667, 16, or 32 MHz.

15.4 Operation in Asynchronous Mode

Figure 15.2 shows the general format for asynchronous serial communication. One frame consists of a start bit (low level), followed by transfer data, a parity bit, and finally stop bits (high level). In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. In asynchronous serial communication, the communication line is usually held in the mark state (high level). The SCI monitors the communication line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

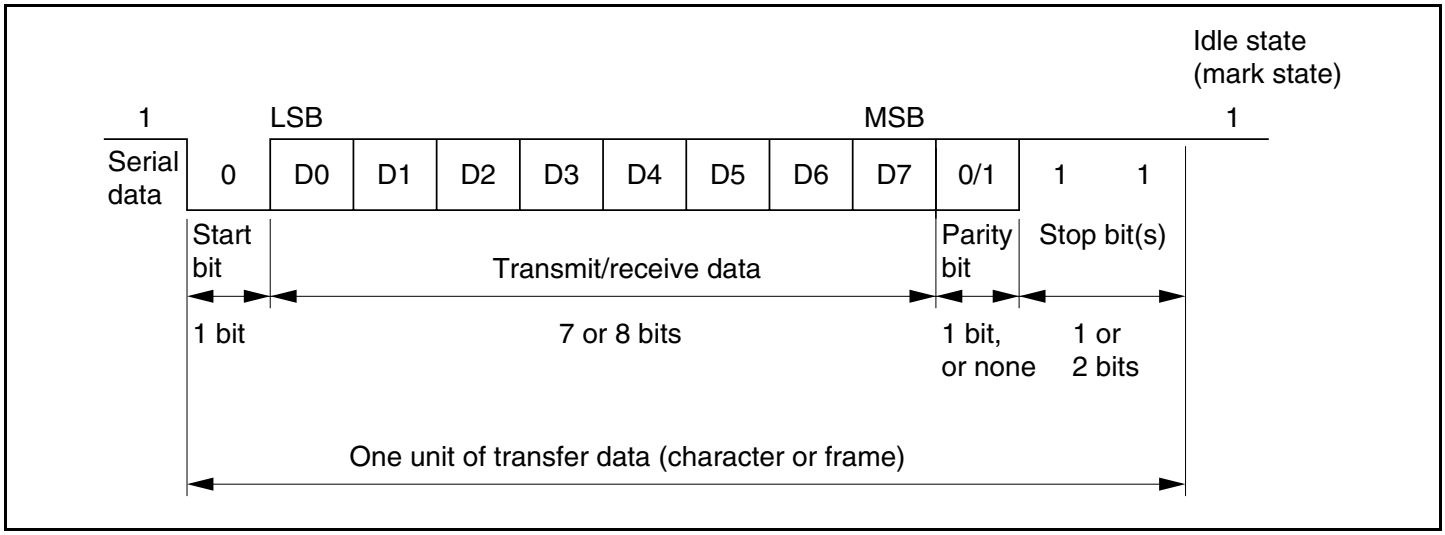


Figure 15.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

15.4.1 Data Transfer Format

Table 15.10 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected according to the SMR setting. For details on the multiprocessor bit, refer to section 15.5, Multiprocessor Communication Function.

Table 15.10 Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STOP
0	1	0	0	S 8-bit data P STOP
0	1	0	1	S 8-bit data P STOP STOP
1	0	0	0	S 7-bit data STOP
1	0	0	1	S 7-bit data STOP STOP
1	1	0	0	S 7-bit data P STOP
1	1	0	1	S 7-bit data P STOP STOP
0	—	1	0	S 8-bit data MPB STOP
0	—	1	1	S 8-bit data MPB STOP STOP
1	—	1	0	S 7-bit data MPB STOP
1	—	1	1	S 7-bit data MPB STOP STOP

Legend

S : Start bit
STOP : Stop bit
P : Parity bit
MPB : Multiprocessor bit

15.4.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a basic clock with a frequency of 16 times the bit rate. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched at the middle of each bit by sampling the data at the rising edge of the 8th pulse of the basic clock as shown in figure 15.3. Thus the reception margin in asynchronous mode is given by formula (1) below.

$$M = \{ (0.5 - \frac{1}{2N}) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \} \times 100 [\%]$$

... Formula (1)

Where M: Reception Margin

N: Ratio of bit rate to clock (N = 16)

D: Clock duty cycle (D = 0.5 to 1.0)

L: Frame length (L = 9 to 12)

F: Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in formula (1), a reception margin is given by formula below.

$M = \{0.5 - 1/(2 \times 16)\} \times 100 [\%] = 46.875\%$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

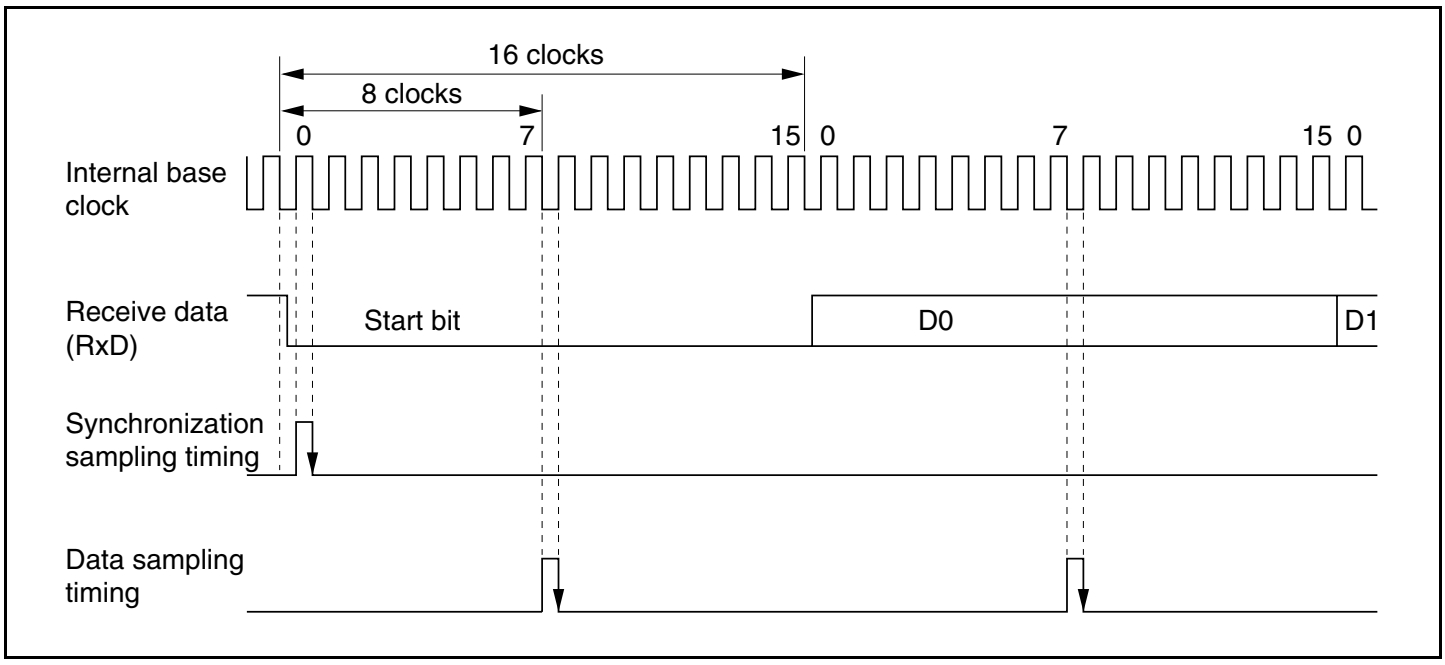


Figure 15.3 Receive Data Sampling Timing in Asynchronous Mode

15.4.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the C/A bit in SMR and the CKE1 and CKE0 bits in SCR. When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 15.4.

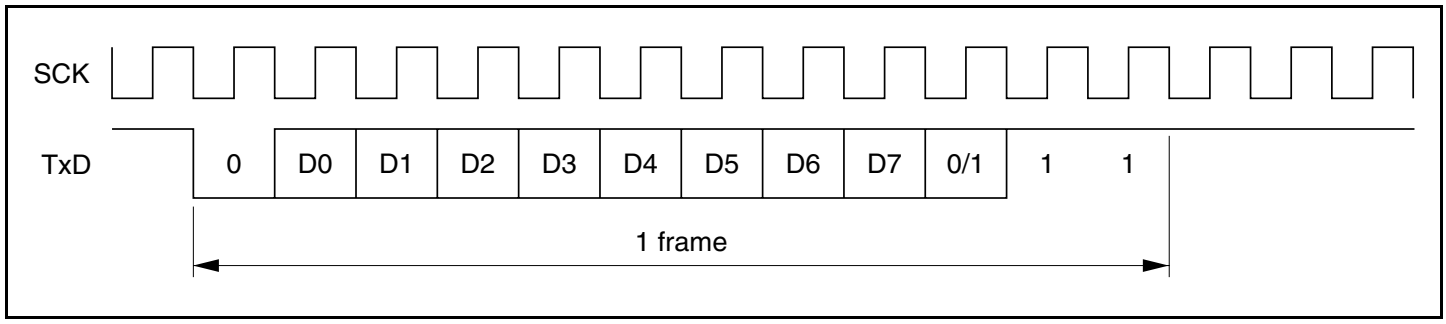
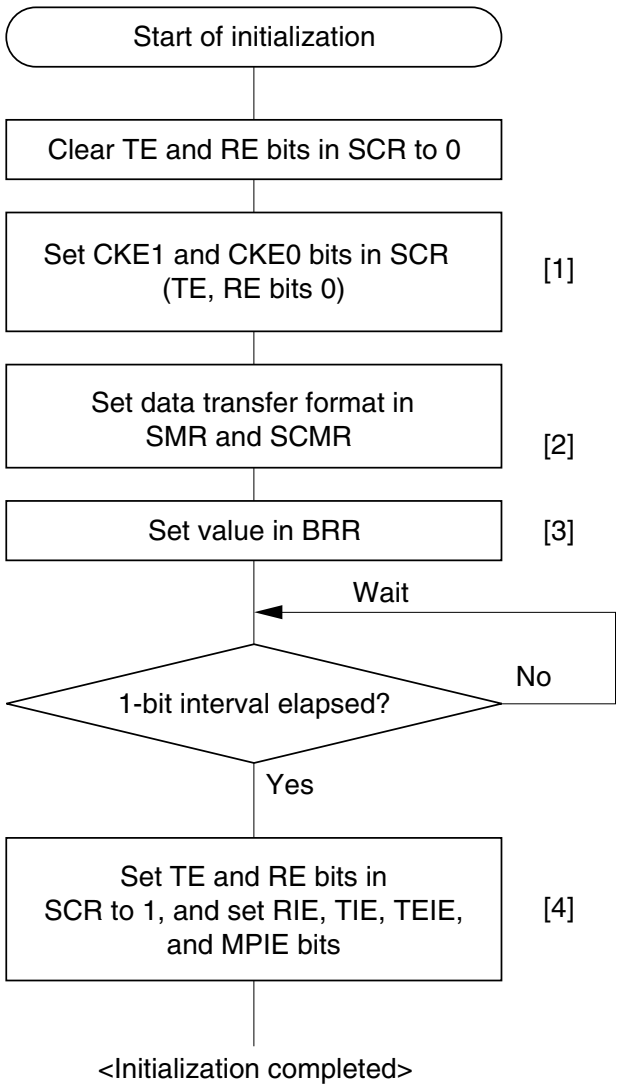


Figure 15.4 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)

15.4.4 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as shown in figure 15.5. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock is selected in asynchronous mode, it is output immediately after SCR settings are made.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. (Not necessary if an external clock is used.)

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.

Setting the TE and RE bits enables the TxD and RxD pins to be used.

Figure 15.5 Sample SCI Initialization Flowchart

15.4.5 Data Transmission (Asynchronous Mode)

Figure 15.6 shows an example of the operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if is cleared to 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit is set to 1 at this time, a transmit data empty interrupt request (TXI) is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit or multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.
6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the "mark state" is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 15.7 shows a sample flowchart for transmission in asynchronous mode.

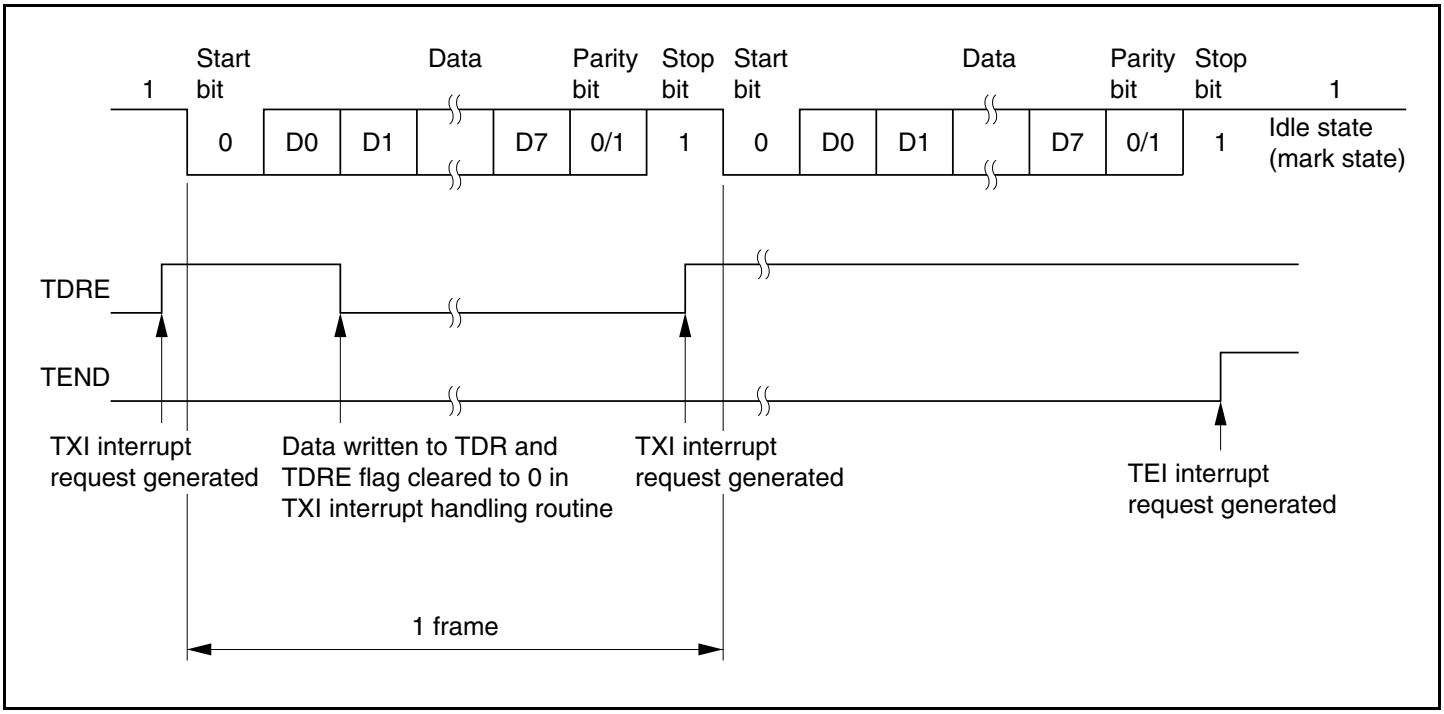


Figure 15.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

15.4.6 Serial Data Reception (Asynchronous Mode)

Figure 15.8 shows an example of the operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI monitors the communication line, and if a start bit is detected, performs internal synchronization, receives receive data in RSR, and checks the parity bit and stop bit.
2. If an overrun error (when reception of the next data is completed while the RDRF flag is still set to 1) occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
5. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

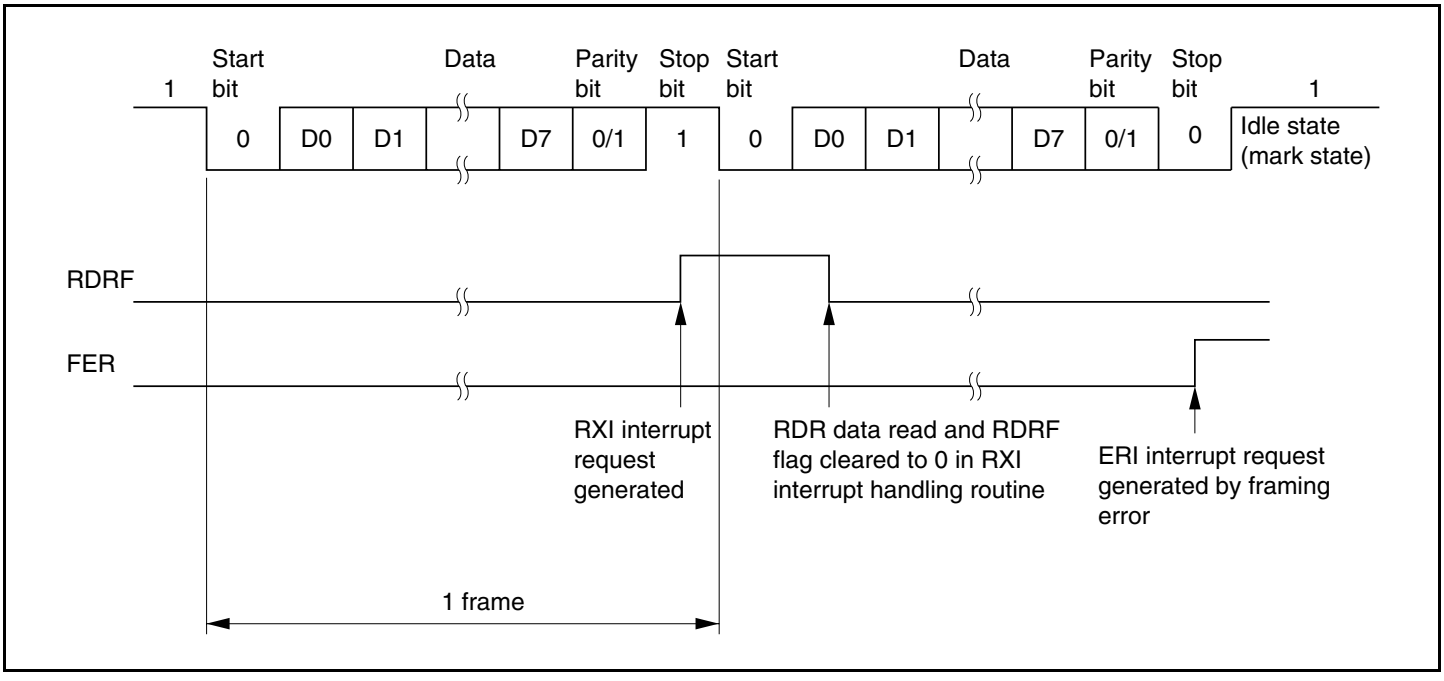


Figure 15.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

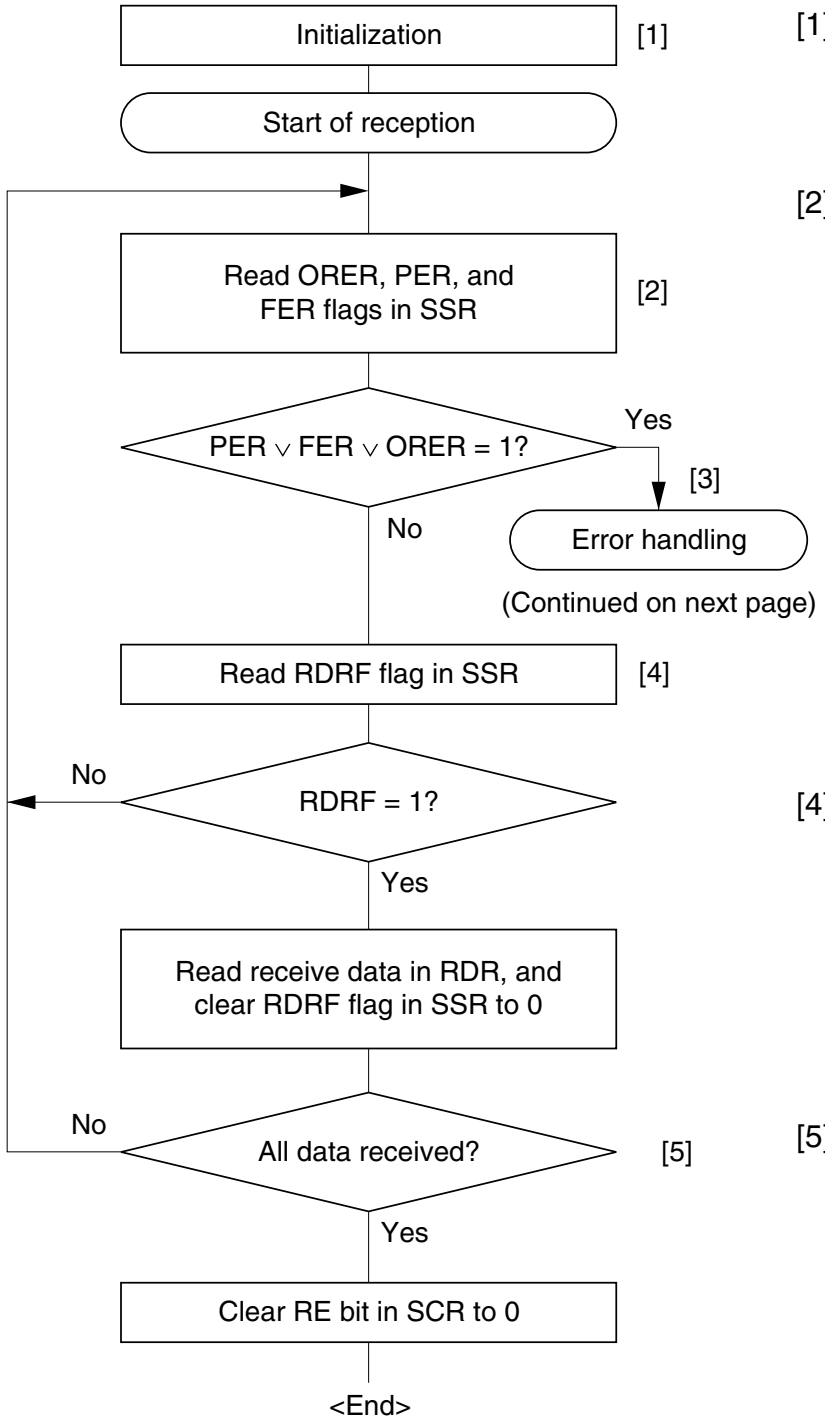
Table 15.11 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the

ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 15.9 shows a sample flowchart for serial data reception.

Table 15.11 SSR Status Flags and Receive Data Handling

SSR Status Flag					
RDRF*	**ORER**	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	1	0	0	Lost	Overrun error
0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + framing error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

Note: The RDRF flag retains its state before data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error handling and break detection:
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

[4] SCI status check and receive data read :
Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0. The RDRF flag is cleared automatically when the DMAC or DTC is activated by an RXI interrupt and the RDR value is read.

Figure 15.9 Sample Serial Reception Data Flowchart (1)

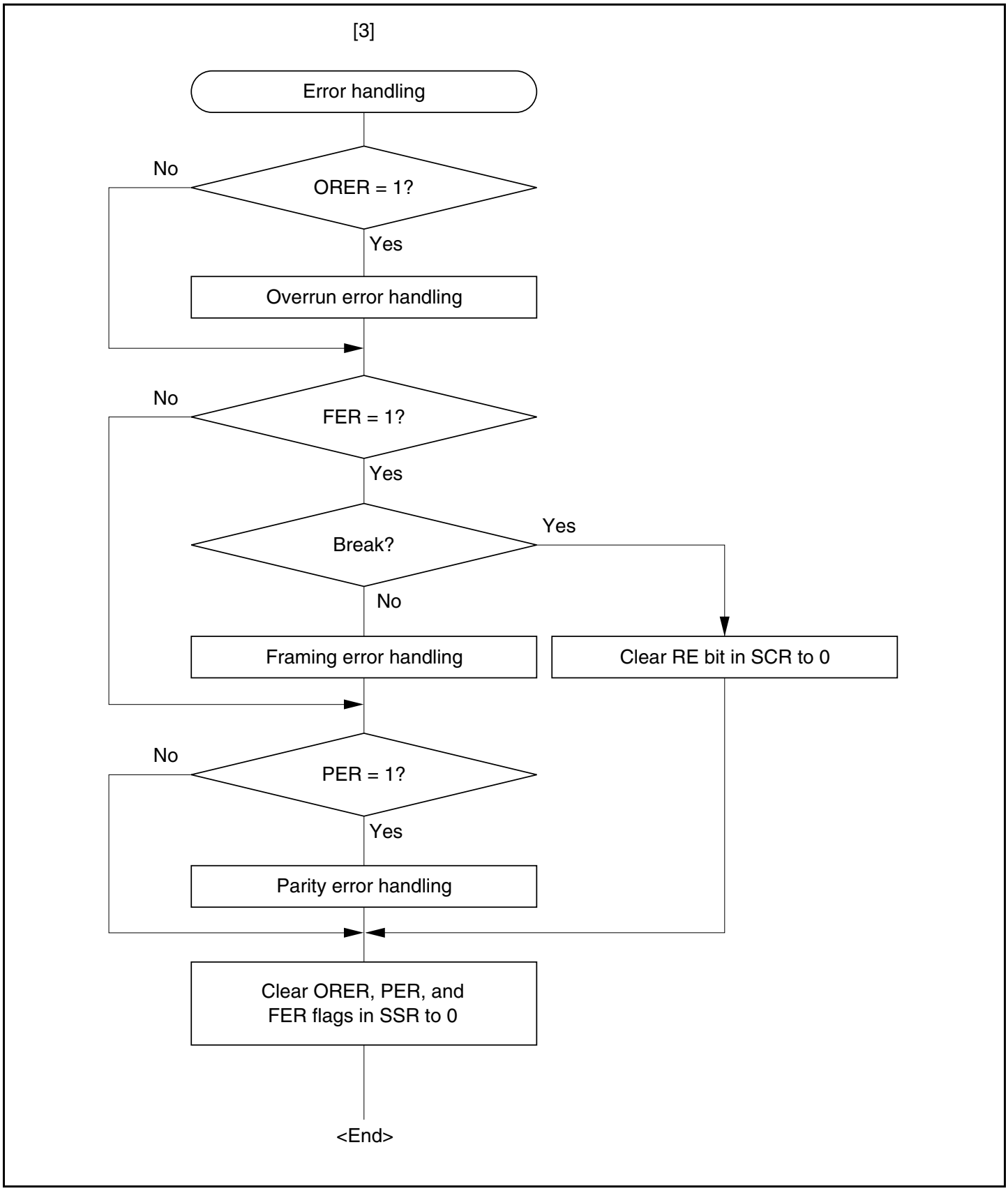


Figure 15.9 Sample Serial Reception Data Flowchart (2)

15.5 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer to be performed among a number of processors sharing communication lines by means of asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle to the specified receiving station. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle, and if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 15.10 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends communication data with a 1 multiprocessor bit added to the ID code of the receiving station. It then sends transmit data as data with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set to 1, transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status flags, RDRF, FER, and ORER to 1 are inhibited until data with a 1 multiprocessor bit is received. On reception of receive character with a 1 multiprocessor bit, the MPBR bit in SSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is invalid. All other bit settings are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

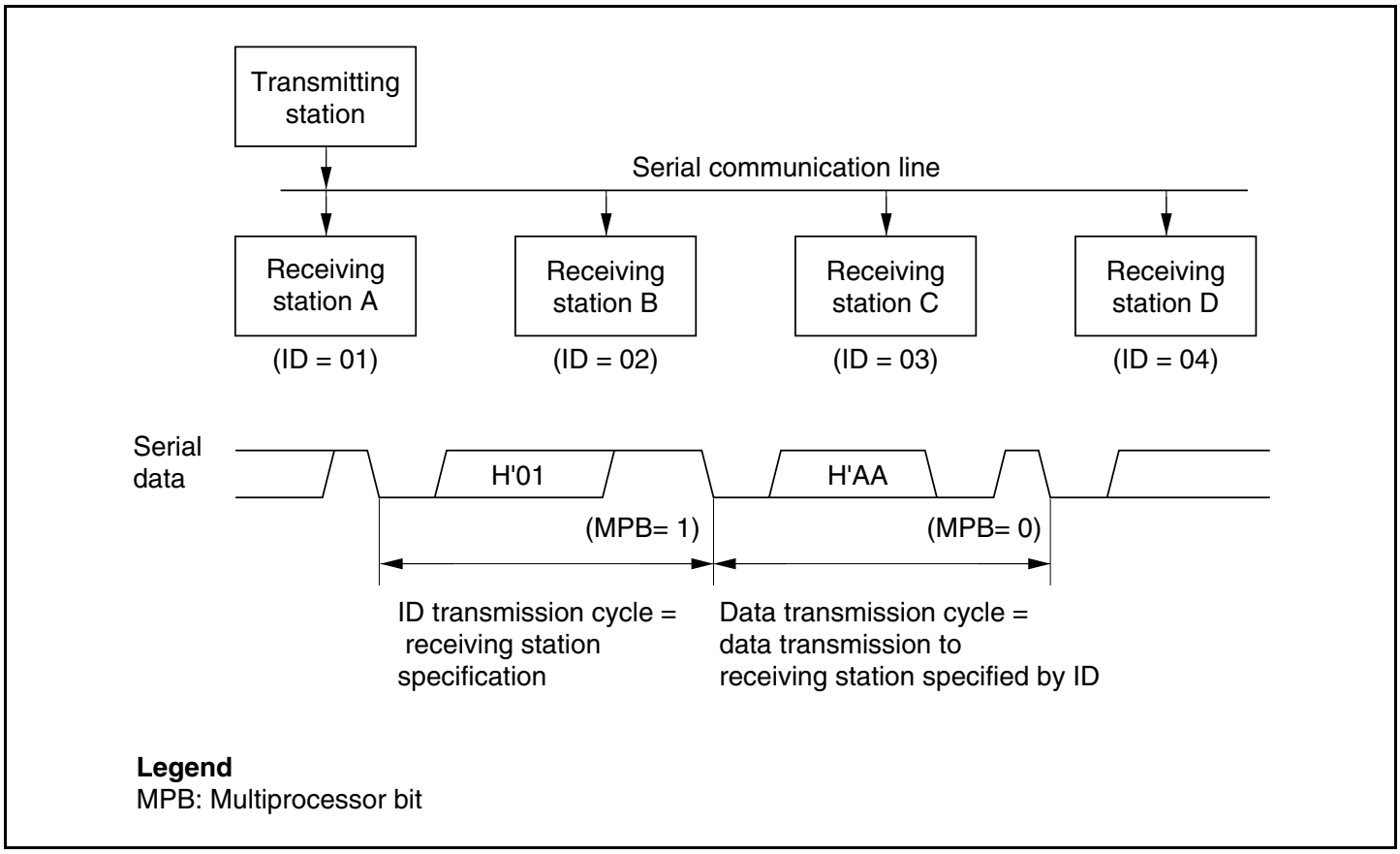
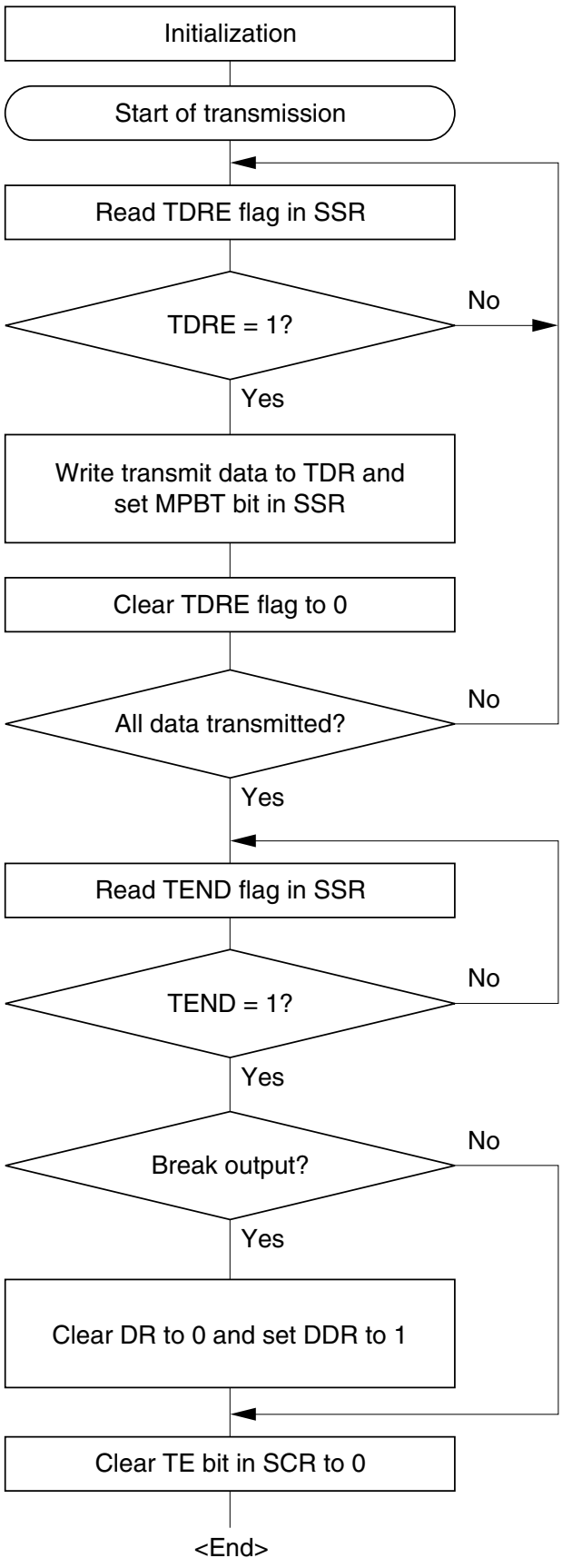


Figure 15.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

15.5.1 Multiprocessor Serial Data Transmission

Figure 15.11 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI operations are the same as those in asynchronous mode.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit-data-empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the port DDR to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 15.11 Sample Multiprocessor Serial Transmission Flowchart

15.5.2 Multiprocessor Serial Data Reception

Figure 15.13 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR is set to 1, data is skipped until data with a 1 multiprocessor bit is received. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 15.12 shows an example of SCI operation for multiprocessor format reception.

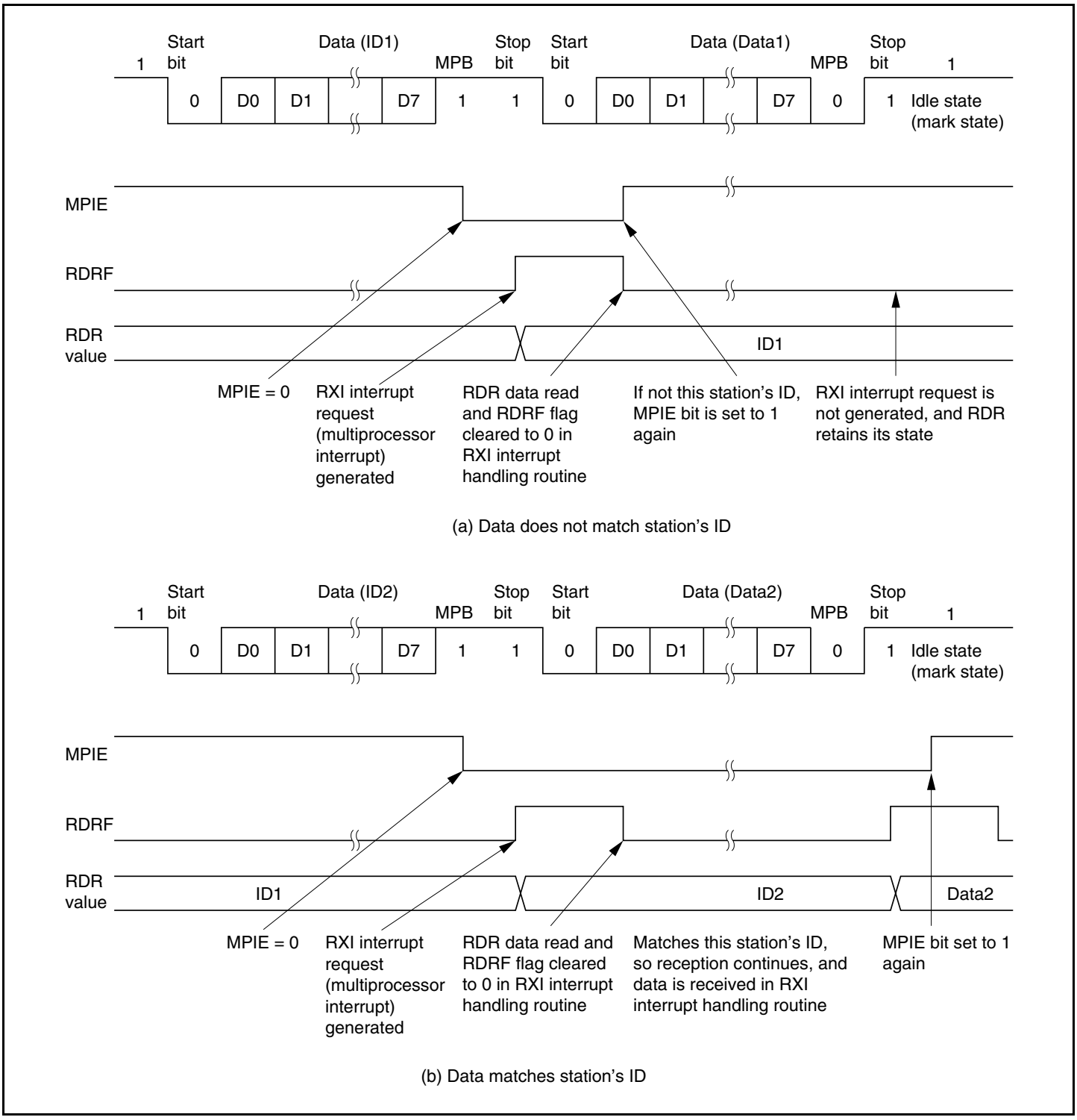
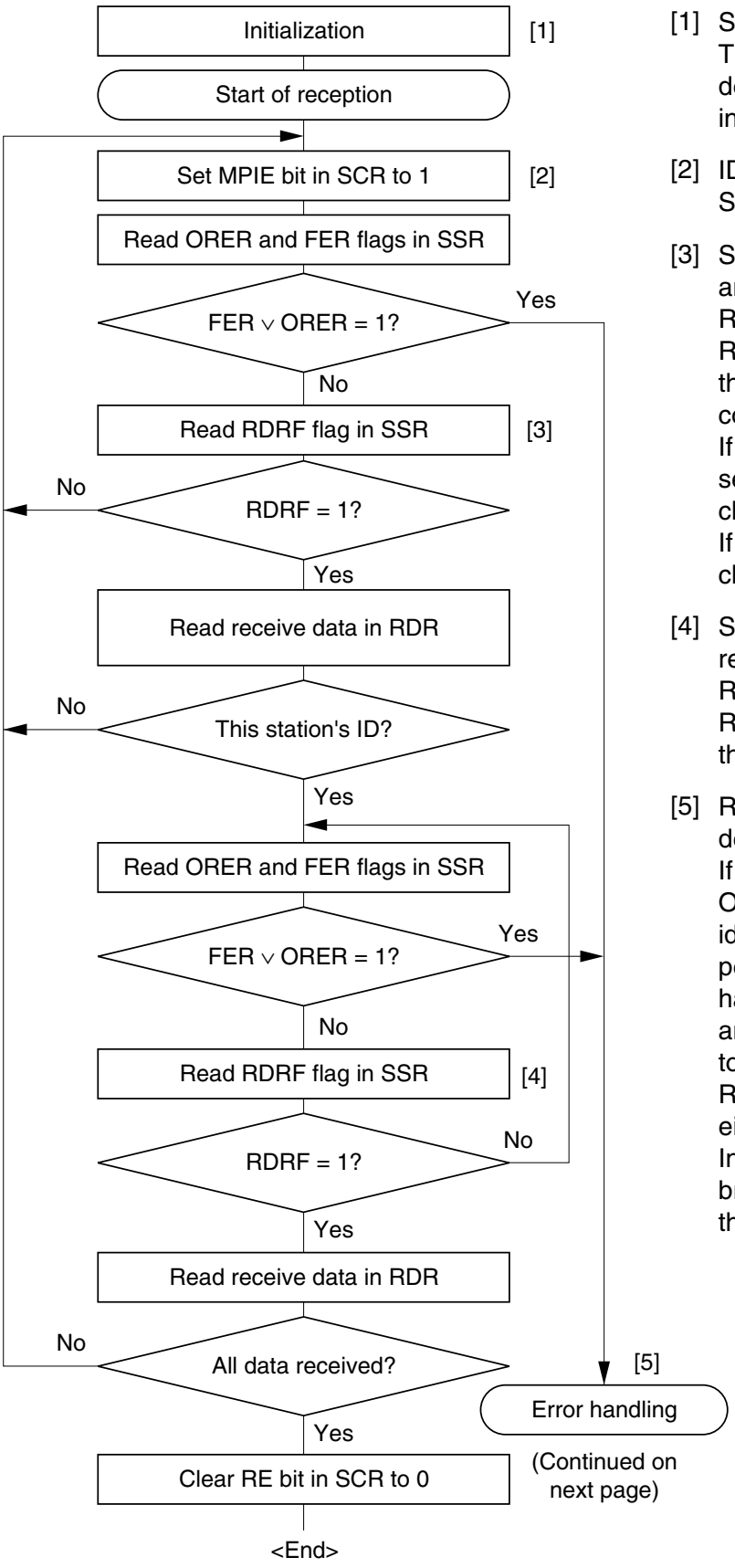


Figure 15.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] ID reception cycle:
Set the MPIE bit in SCR to 1.

[3] SCI status check, ID reception and comparison:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and compare it with this station's ID.
If the data is not this station's ID, set the MPIE bit to 1 again, and clear the RDRF flag to 0.
If the data is this station's ID, clear the RDRF flag to 0.

[4] SCI status check and data reception:
Read SSR and check that the RDRF flag is set to 1, then read the data in RDR.

[5] Receive error handling and break detection:
If a receive error occurs, read the ORER and FER flags in SSR to identify the error. After performing the appropriate error handling, ensure that the ORER and FER flags are both cleared to 0.
Reception cannot be resumed if either of these flags is set to 1.
In the case of a framing error, a break can be detected by reading the RxD pin value.

Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (1)

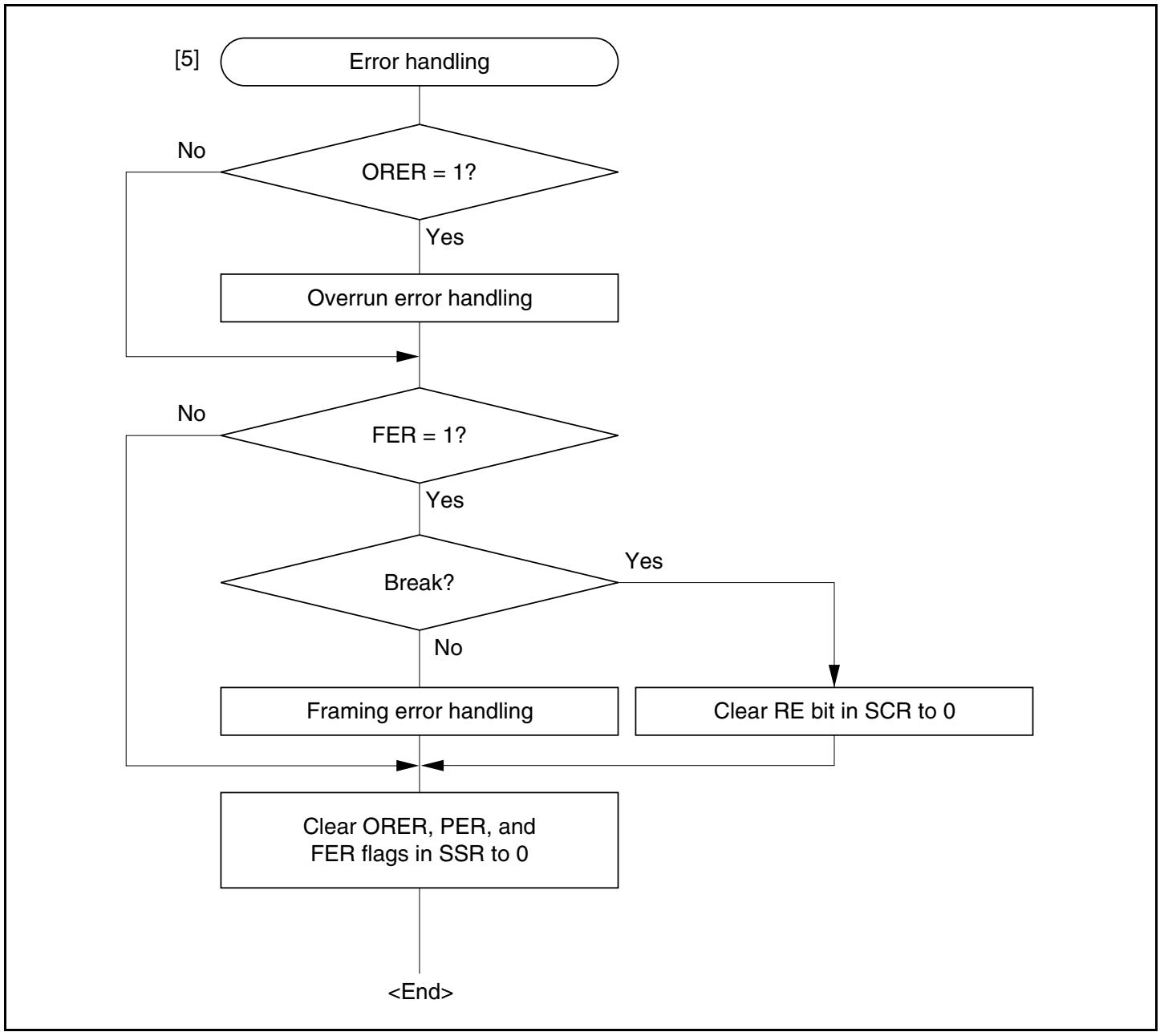


Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (2)

15.6 Operation in Clocked Synchronous Mode

Figure 15.14 shows the general format for clocked synchronous communication. In clocked synchronous mode, data is transmitted or received in synchronization with clock pulses. One character of communication data consists of 8-bit data. In clocked synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. In clocked synchronous mode, the SCI receives data in synchronization with the rising edge of the serial clock. After 8-bit data is output, the transmission line holds the MSB state. In clocked synchronous mode, no parity or multiprocessor bit is added. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

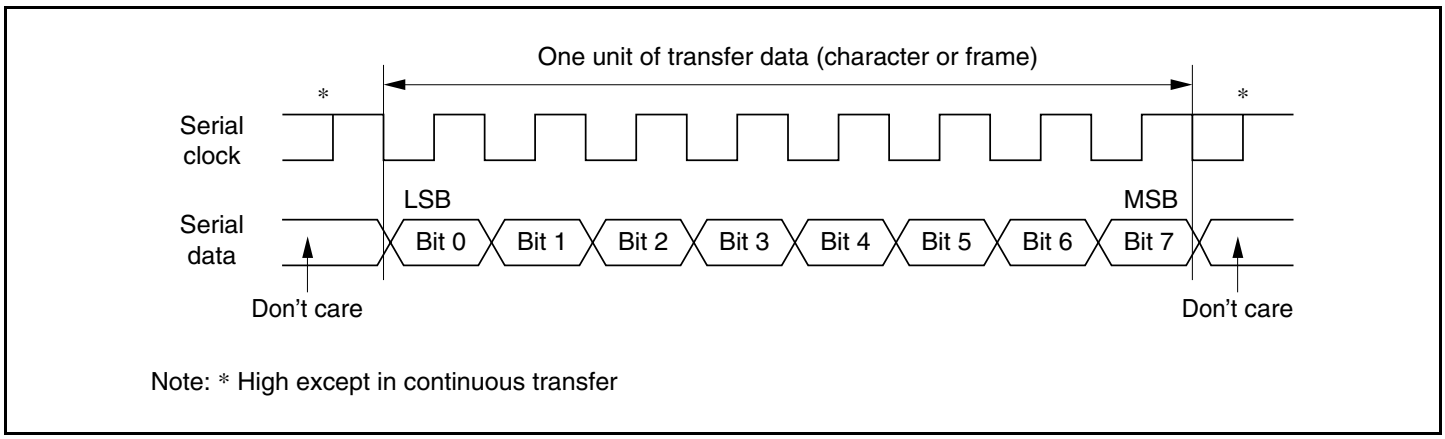


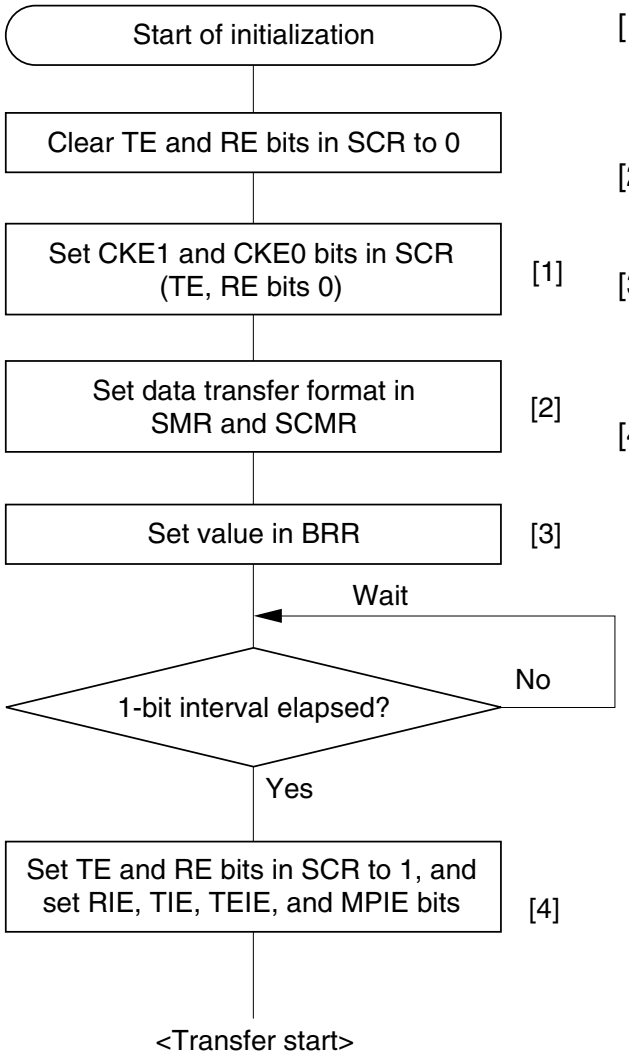
Figure 15.14 Data Format in Clocked Synchronous Communication (For LSB-First)

15.6.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK pin can be selected, according to the setting of CKE1 and CKE0 bits in SCR. When the SCI is operated on an internal clock, the serial clock is output from the SCK pin. Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high.

15.6.2 SCI Initialization (Clocked Synchronous Mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described in a sample flowchart in figure 15.15. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, TE and RE, to 0.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. (Not necessary if an external clock is used.)

[4] Wait at least one bit interval, then set the TE and RE bits in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.
Setting the TE and RE bits enable the TxD and RxD pins to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 15.15 Sample SCI Initialization Flowchart

15.6.3 Serial Data Transmission (Clocked Synchronous Mode)

Figure 15.16 shows an example of SCI operation for transmission in clocked synchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a TXI interrupt request is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. 8-bit data is sent from the TxD pin synchronized with the output clock when output clock mode has been specified and synchronized with the input clock when use of an external clock has been specified.
4. The SCI checks the TDRE flag at the timing for sending the MSB.
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TxD pin maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated. The SCK pin is fixed high.

Figure 15.17 shows a sample flowchart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) is set to 1. Make sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

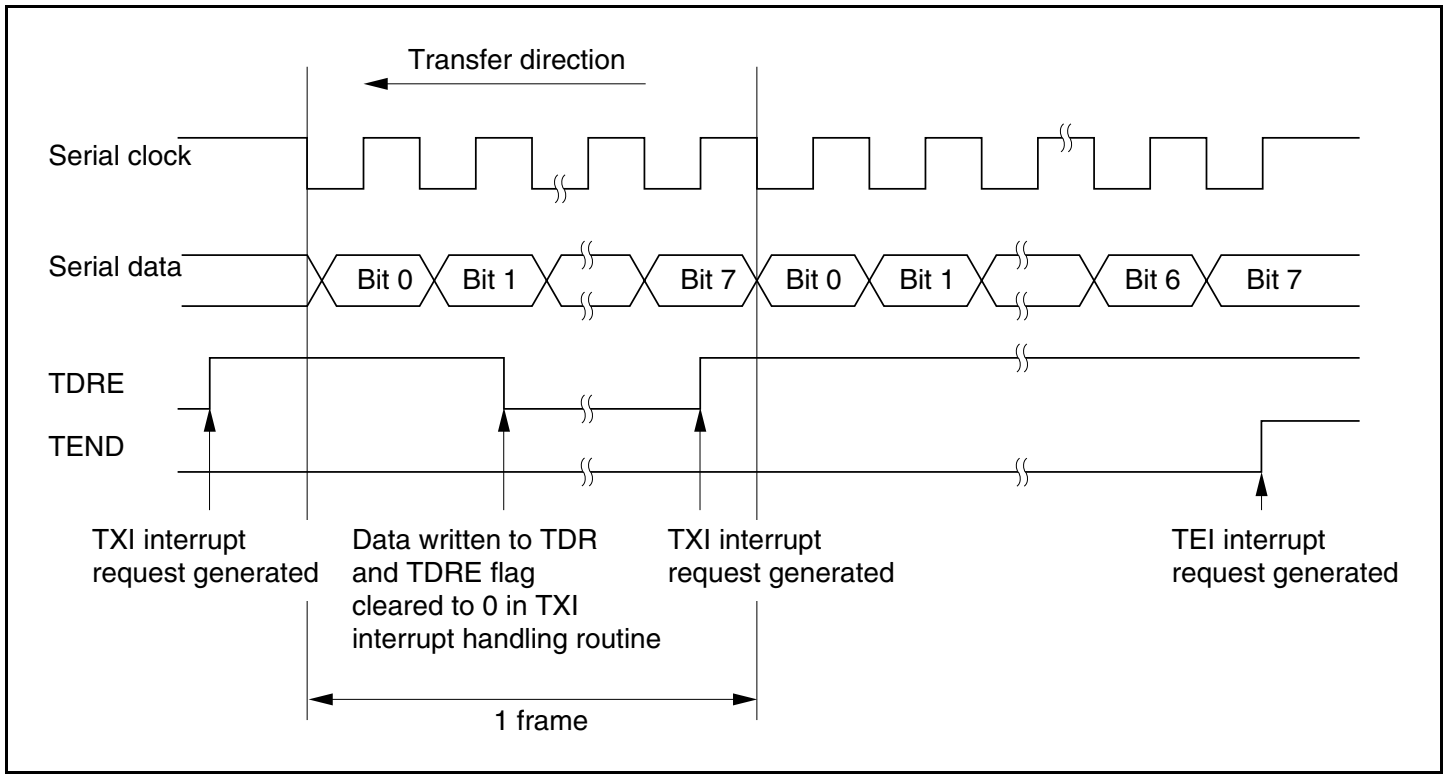
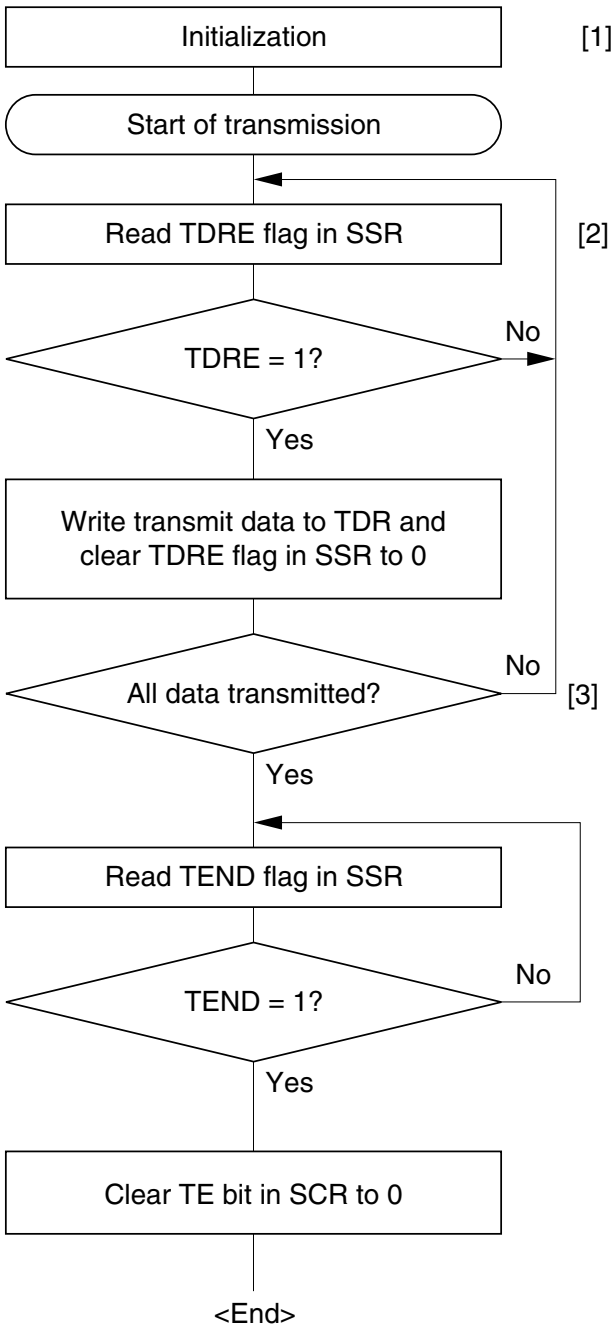


Figure 15.16 Sample SCI Transmission Operation in Clocked Synchronous Mode



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit-data-empty interrupt (TXI) request and data is written to TDR.

Figure 15.17 Sample Serial Transmission Flowchart

15.6.4 Serial Data Reception (Clocked Synchronous Mode)

Figure 15.18 shows an example of SCI operation for reception in clocked synchronous mode. In serial reception, the SCI operates as described below.

1. The SCI performs internal initialization in synchronization with a synchronization clock input or output, starts receiving data, and stores the received data in RSR.
2. If an overrun error (when reception of the next data is completed while the RDRF flag is still set to 1) occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
3. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

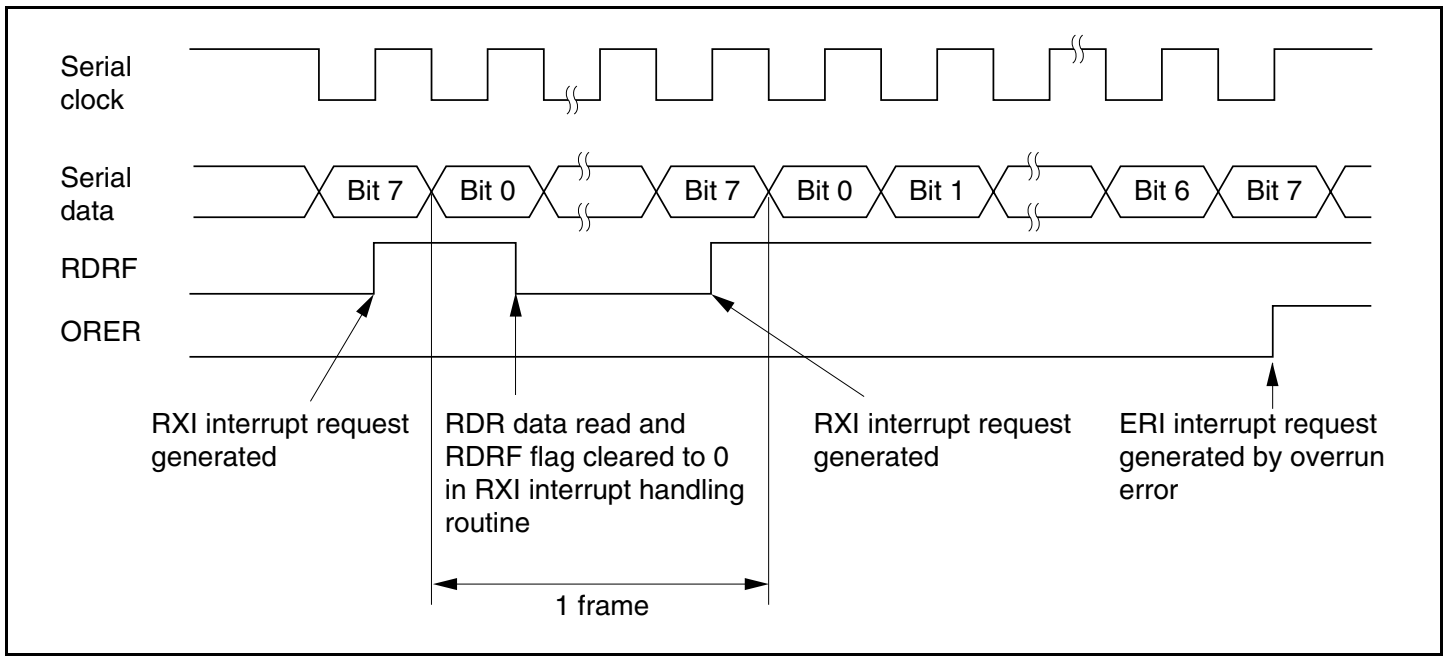
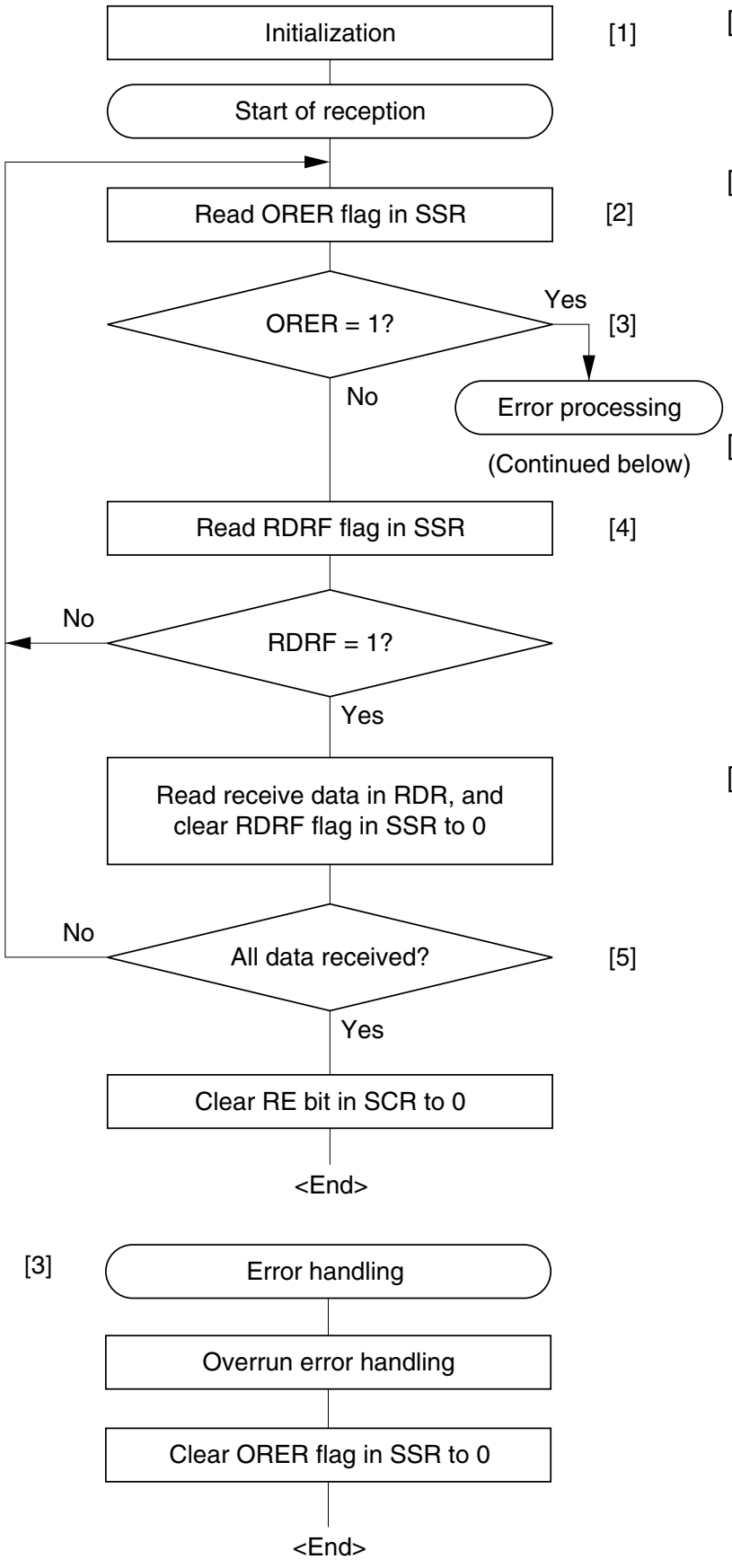


Figure 15.18 Example of SCI Operation in Reception

Transfer cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 15.19 shows a sample flowchart for serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error handling:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error handling, clear the ORER flag to 0. Transfer cannot be resumed if the ORER flag is set to 1.

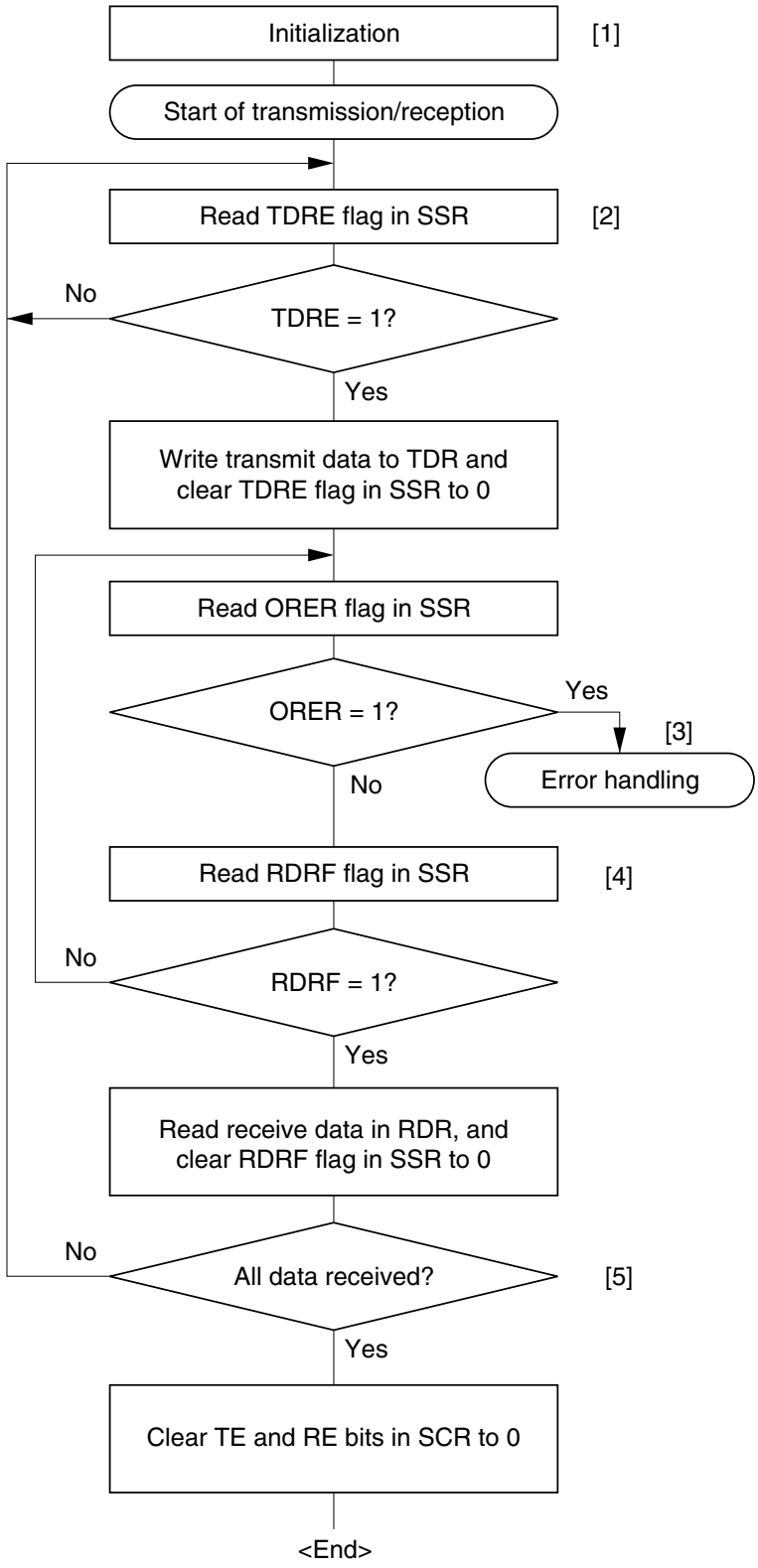
[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. The RDRF flag is cleared automatically when the DMAC or DTC is activated by a receive-data-full interrupt (RXI) request and the RDR value is read.

Figure 15.19 Sample Serial Reception Flowchart

15.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 15.20 shows a sample flowchart for simultaneous serial transmit and receive operations. The following procedure should be used for simultaneous serial data transmit and receive operations after the SCI is initialized. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.



[1] SCI initialization:
The TxD pin is designated as the transmit data output pin, and the RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.
Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error handling:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error handling, clear the ORER flag to 0.
Transmission/reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit-data-empty interrupt (TXI) request and data is written to TDR. Also, the RDRF flag is cleared automatically when the DMAC or DTC is activated by a receive-data-full interrupt (RXI) request and the RDR value is read.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE and RE bits to 0, then set both these bits to 1 simultaneously.

Figure 15.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

15.7 Operation in Smart Card Interface Mode

The SCI supports an IC card (Smart Card) interface conforming to ISO/IEC 7816-3 (Identification Card) as a serial communication interface extension function. Switching between the normal serial communication interface and the Smart Card interface is carried out by means of a register setting.

15.7.1 Pin Connection Example

Figure 15.21 shows an example of connection with the Smart Card. In communication with an IC card, since both transmission and reception are carried out on a single data transmission line, the TxD pin and RxD pin should be connected with the LSI pin. The data transmission line should be pulled up to the V_{CC} power supply with a resistor. If an IC card is not connected, and the TE and RE bits are both set to 1, closed transmission/reception is possible, enabling self-diagnosis to be carried out. When the clock generated on the SCI is used by an IC card, the SCK pin output is input to the CLK pin of the IC card. This LSI port output is used as the reset signal.

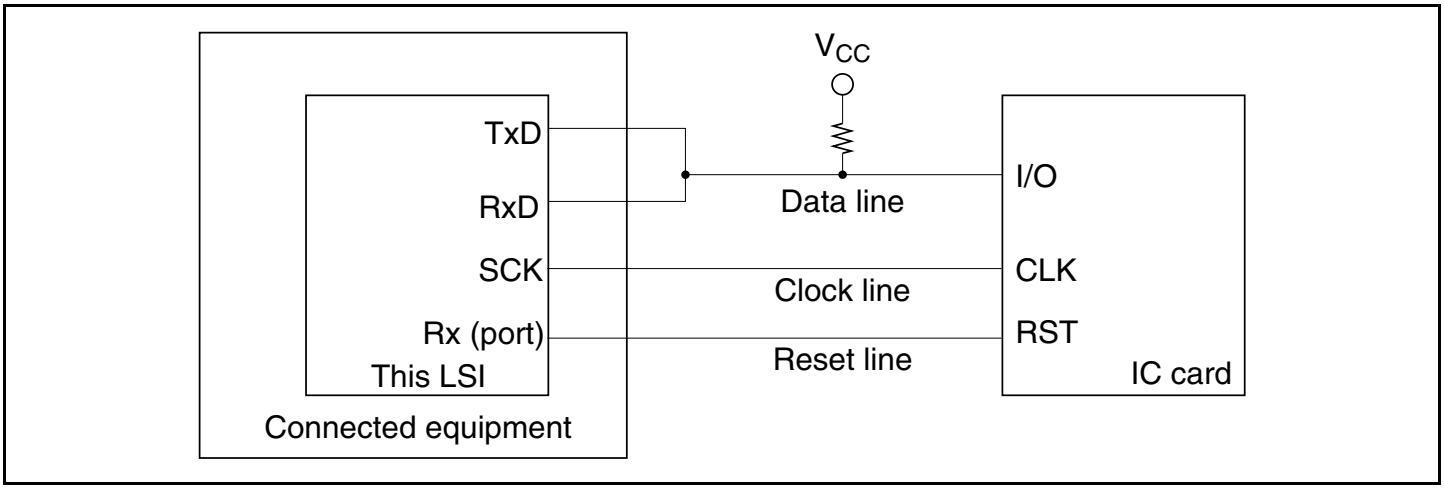


Figure 15.21 Schematic Diagram of Smart Card Interface Pin Connections

15.7.2 Data Format (Except for Block Transfer Mode)

Figure 15.22 shows the transfer data format in Smart Card interface mode.

- One frame consists of 8-bit data plus a parity bit in asynchronous mode.
- In transmission, a guard time of at least 2 etu (Elementary Time Unit: the time for transfer of one bit) is left between the end of the parity bit and the start of the next frame.
- If a parity error is detected during reception, a low error signal level is output for one etu period, 10.5 etu after the start bit.
- If an error signal is sampled during transmission, the same data is retransmitted automatically after the elapse of 2 etu or longer.

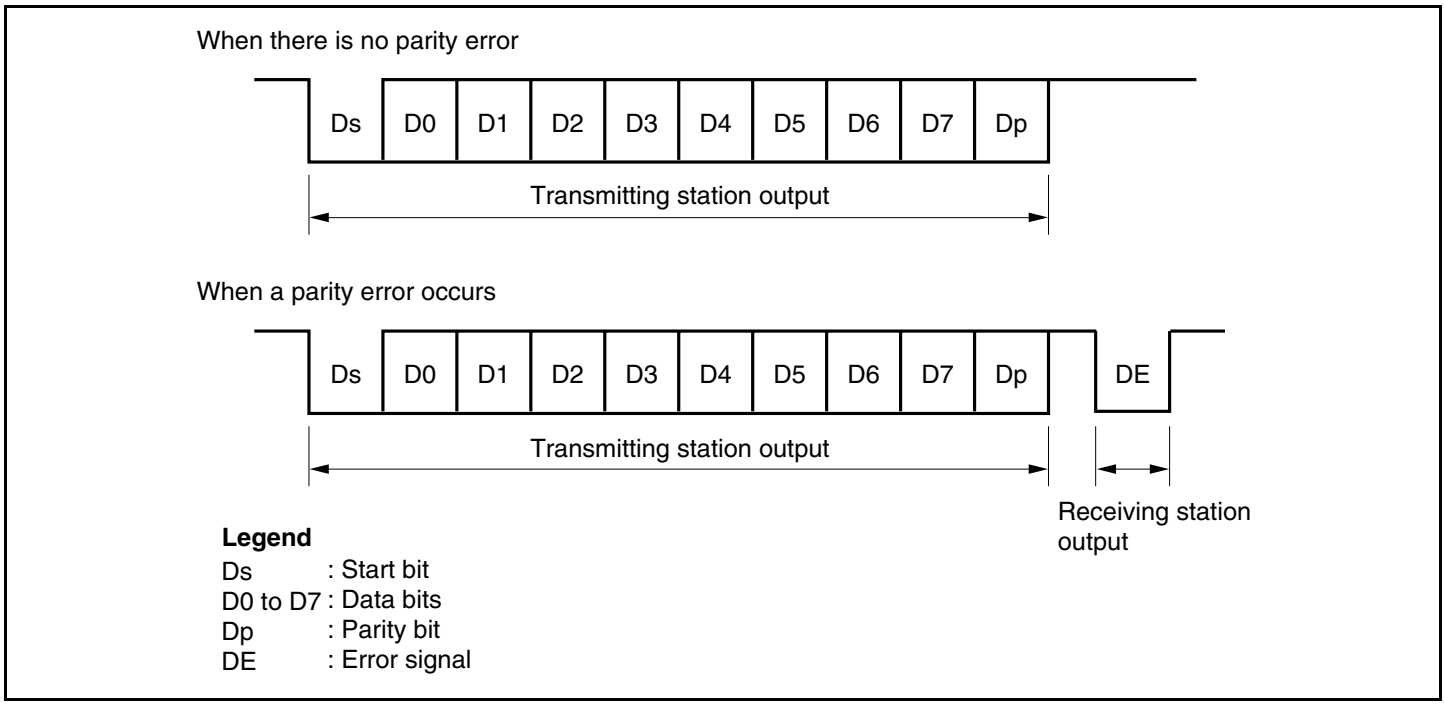


Figure 15.22 Normal Smart Card Interface Data Format

Data transfer with the types of IC cards (direct convention and inverse convention) are performed as described in the following.

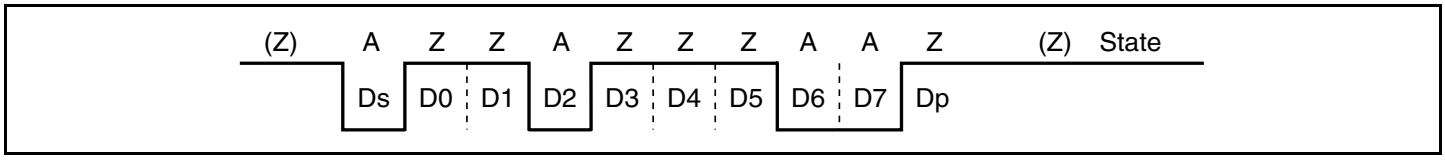


Figure 15.23 Direct Convention (SDIR = SINV = O/E = 0)

As in the above sample start character, with the direct convention type, the logic 1 level corresponds to state Z and the logic 0 level to state A, and transfer is performed in LSB-first order. The start character data above is H'3B. For the direct convention type, clear the SDIR and SINV bits in SCMR to 0. According to the Smart Card regulations, clear the O/E bit in SMR to 0 to select even parity mode.

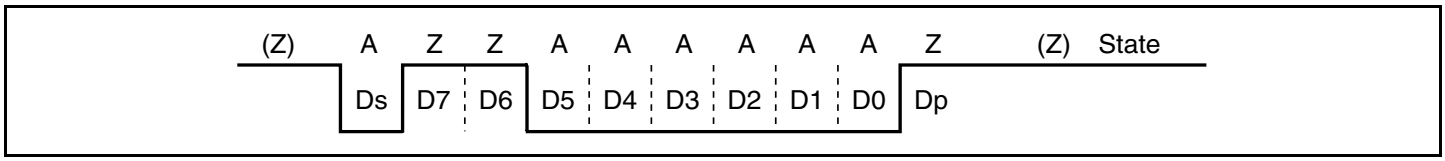


Figure 15.24 Inverse Convention (SDIR = SINV = O/E = 1)

With the inverse convention type, the logic 1 level corresponds to state A and the logic 0 level to state Z, and transfer is performed in MSB-first order. The start character data above is H'3F. For

15.7.3 Block Transfer Mode

Operation in block transfer mode is the same as that in normal Smart Card interface, except for the following points.

- In reception, though the parity check is performed, no error signal is output even if an error is detected. However, the PER bit in SSR is set to 1 and must be cleared before receiving the parity bit of the next frame.
- In transmission, a guard time of at least 1 etu is left between the end of the parity bit and the start of the next frame.
- In transmission, because retransmission is not performed, the TEND flag is set to 1, 11.5 etu after transmission start.
- As with the normal Smart Card interface, the ERS flag indicates the error signal status, but since error signal transfer is not performed, this flag is always cleared to 0.

15.7.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator is used as transmit/receive clock in Smart Card interface. In Smart Card interface mode, the SCI operates on a basic clock with a frequency of 32, 64, 372, or 256 times the bit rate (fixed at 16 times in normal asynchronous mode) as determined by bits BCP1 and BCP0. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. As shown in figure 15.25, by sampling receive data at the rising-edge of the 16th, 32nd, 186th, or 128th pulse of the basic clock, data can be latched at the middle of the bit. The reception margin is given by the following formula.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

Where M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, and 256)

D: Clock duty cycle (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5 and N = 372 in the above formula, the reception margin formula is as follows.

$$M = (0.5 - 1/2 \times 372) \times 100\%$$
$$= 49.866\%$$

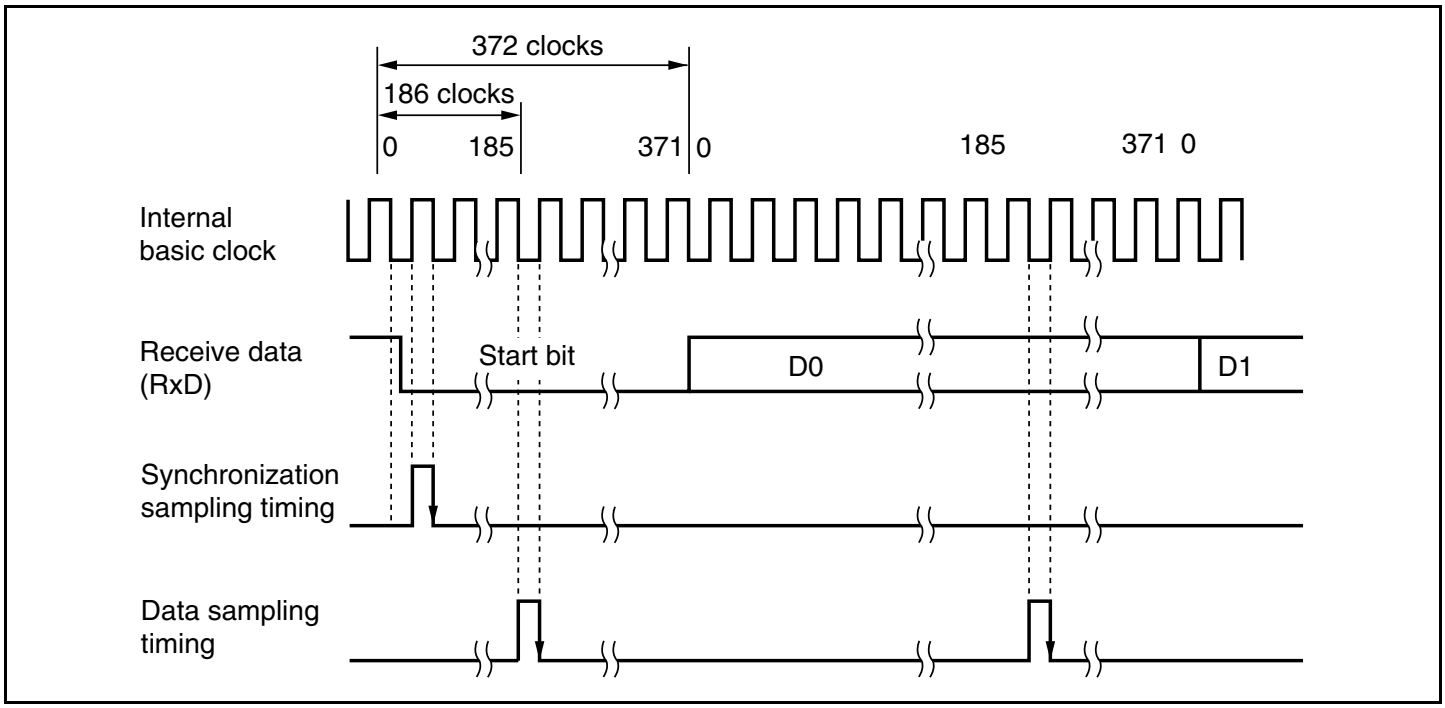


Figure 15.25 Receive Data Sampling Timing in Smart Card Mode (Using Clock of 372 Times the Bit Rate)

15.7.5 Initialization

Before transmitting and receiving data, initialize the SCI as described below. Initialization is also necessary when switching from transmit mode to receive mode, or vice versa.

1. Clear the TE and RE bits in SCR to 0.
2. Clear the error flags ERS, PER, and ORER in SSR to 0.
3. Set the GM, BLK, O/E, BCP1, BCP0, CKS1, and CKS0 bits in SMR. Set the PE bit to 1.
4. Set the SMIF, SDIR, and SINV bits in SCMR.
   When the SMIF bit is set to 1, the TxD and RxD pins are both switched from ports to SCI pins, and are placed in the high-impedance state.
5. Set the value corresponding to the bit rate in BRR.
6. Set the CKE0 and CKE1 bits in SCR. Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0.
   If the CKE0 bit is set to 1, the clock is output from the SCK pin.
7. Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do not set the TE bit and RE bit at the same time, except for self-diagnosis.

To switch from receive mode to transmit mode, after checking that the SCI has finished reception, initialize the SCI, and clear RE to 0 and set TE to 1. Whether SCI has finished reception can be checked with the RDRF, PER, or ORER flag. To switch from transmit mode to receive mode, after checking that the SCI has finished transmission, initialize the SCI, and clear TE to 0 and set RE to 1. Whether SCI has finished transmission can be checked with the TEND flag.

15.7.6 Data Transmission (Except for Block Transfer Mode)

As data transmission in Smart Card interface mode involves error signal sampling and retransmission processing, the operations are different from those in normal serial communication interface mode (except for block transfer mode). Figure 15.26 illustrates the retransfer operation when the SCI is in transmit mode.

1. If an error signal is sampled from the receiving end after transmission of one frame is completed, the ERS bit in SSR is set to 1. If the RIE bit in SCR is set at this time, an ERI interrupt request is generated. The ERS bit in SSR should be cleared to 0 before the next parity bit is sampled.
2. The TEND bit in SSR is not set for a frame for which an error signal is received. Data is retransferred from TDR to TSR, and retransmitted automatically.
3. If an error signal is not sent back from the receiving end, the ERS bit in SSR is not set. Transmission of one frame, including a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1. If the TIE bit in SCR is set at this time, a TXI interrupt request is generated. Writing transmit data to TDR transfers the next transmit data.

Figure 15.28 shows a flowchart for transmission. The sequence of transmit operations can be performed automatically by specifying the DTC or DMAC to be activated with a TXI interrupt source. In a transmit operation, the TDRE flag is also set to 1 at the same time as the TEND flag in SSR, and a TXI interrupt will be generated if the TIE bit in SCR has been set to 1. If the TXI request is designated beforehand as a DTC or DMAC activation source, the DTC or DMAC will be activated by the TXI request, and transfer of the transmit data will be carried out. The TDRE and TEND flags are automatically cleared to 0 when data transfer is performed by the DTC or DMAC. In the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC will automatically transmit the specified number of bytes in the event of an error, including retransmission. However, the ERS flag is not cleared automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When performing transfer using the DTC or DMAC, it is essential to set and enable the DTC or DMAC before carrying out SCI setting. For details on the DTC or DMAC setting procedures, refer to section 9, Data Transfer Controller (DTC) or section 7, DMA Controller (DMAC).

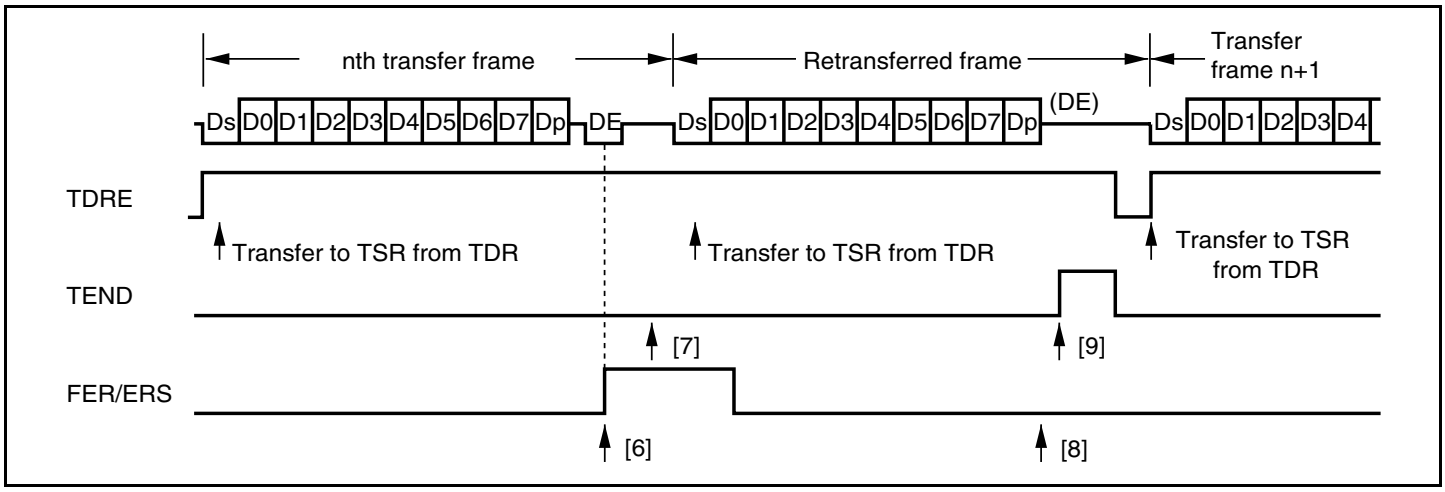


Figure 15.26 Retransfer Operation in SCI Transmit Mode

The timing for setting the TEND flag depends on the value of the GM bit in SMR. The TEND flag generation timing is shown in figure 15.27.

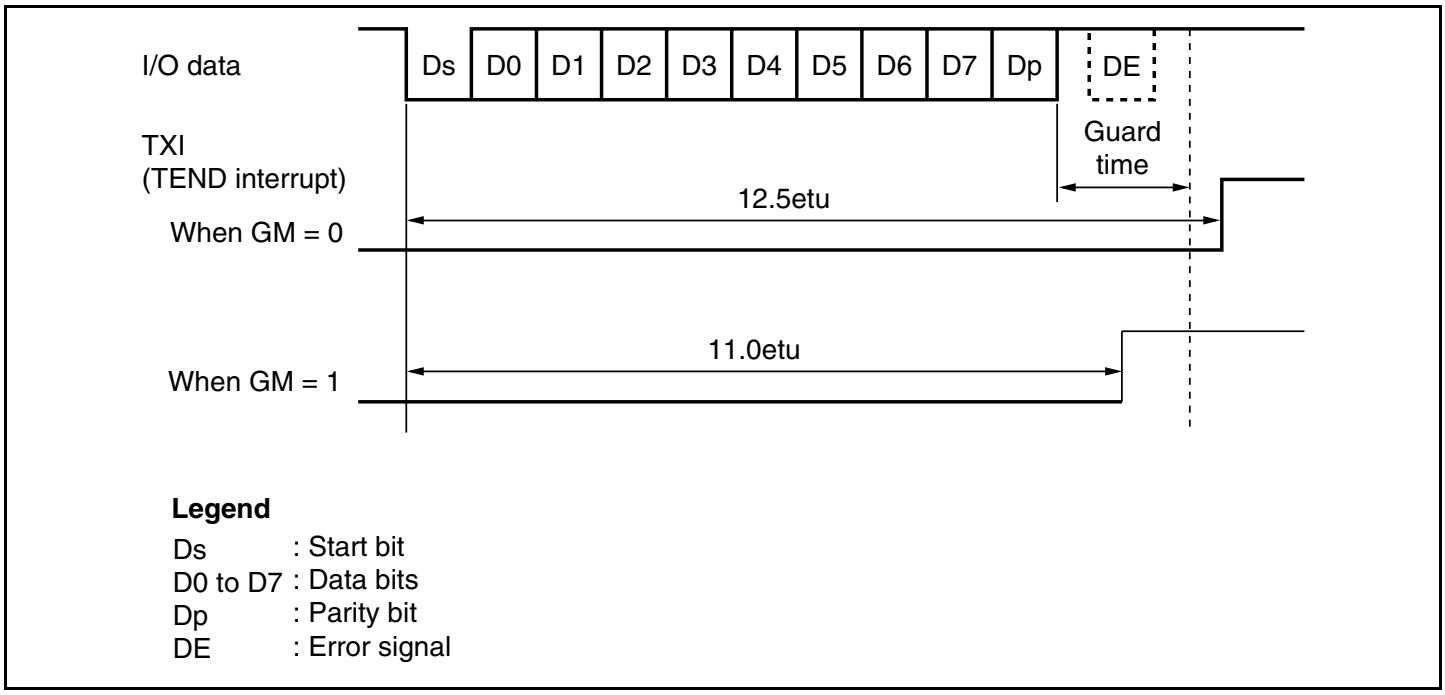


Figure 15.27 TEND Flag Generation Timing in Transmission Operation

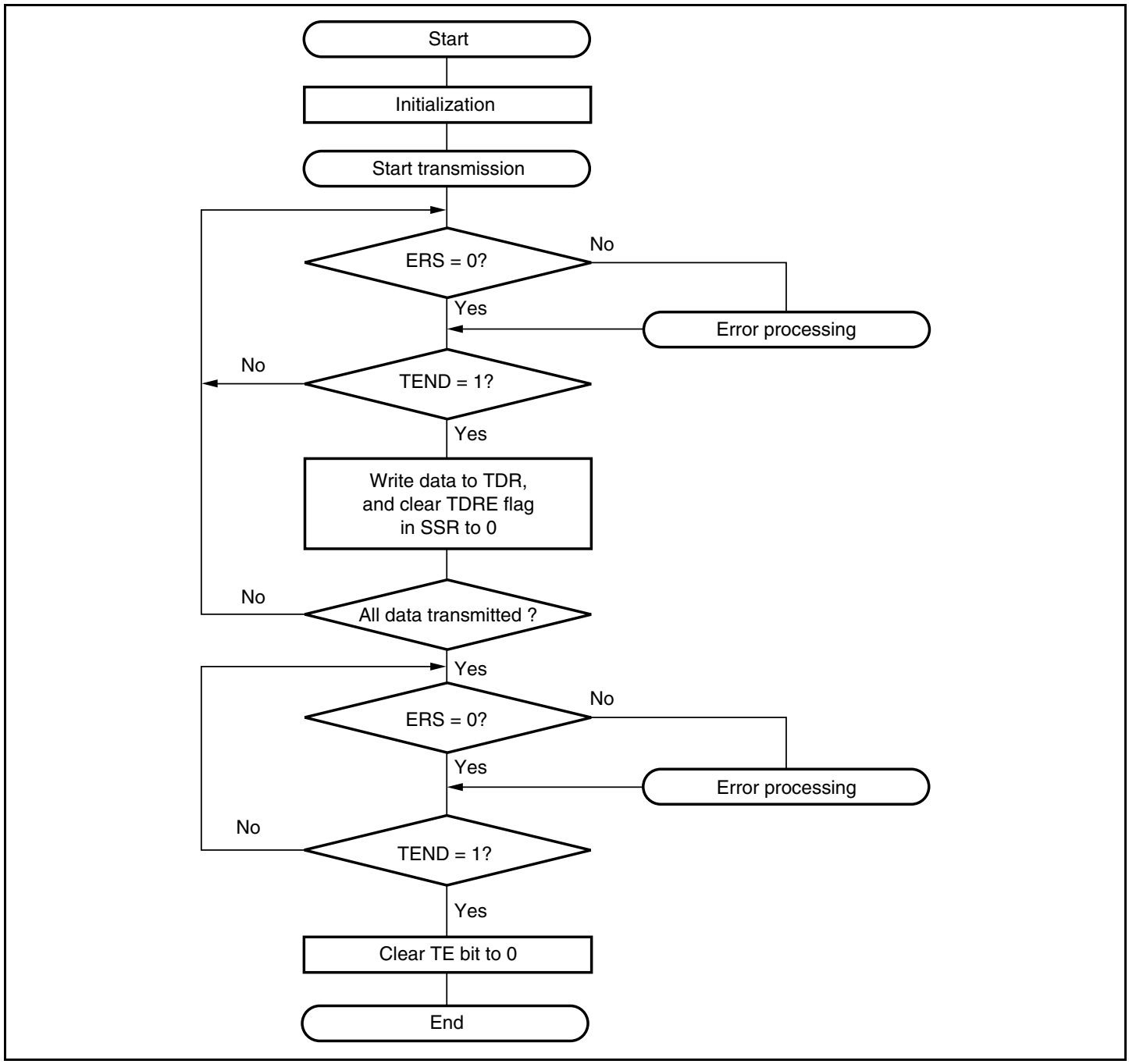


Figure 15.28 Example of Transmission Processing Flow

15.7.7 Serial Data Reception (Except for Block Transfer Mode)

Data reception in Smart Card interface mode uses the same operation procedure as for normal serial communication interface mode. Figure 15.29 illustrates the retransfer operation when the SCI is in receive mode.

1. If an error is found when the received parity bit is checked, the PER bit in SSR is automatically set to 1. If the RIE bit in SCR is set at this time, an ERI interrupt request is generated. The PER bit in SSR should be cleared to 0 before the next parity bit is sampled.
2. The RDRF bit in SSR is not set for a frame in which an error has occurred.
3. If no error is found when the received parity bit is checked, the PER bit in SSR is not set to 1. The receive operation is judged to have been completed normally, and the RDRF flag in SSR is automatically set to 1. If the RIE bit in SCR is set at this time, an RXI interrupt request is generated.

Figure 15.30 shows a flowchart for reception. The sequence of receive operations can be performed automatically by specifying the DTC or DMAC to be activated with an RXI interrupt source. In a receive operation, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DTC or DMAC activation source, the DTC or DMAC will be activated by the RXI request, and transfer of the receive data will be carried out. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC or DMAC. If an error occurs in receive mode and the ORER or PER flag is set to 1, a transfer error interrupt (ERI) request will be generated, and so the error flag must be cleared to 0. In the event of an error, the DTC or DMAC is not activated and receive data is skipped. Therefore, receive data is transferred for only the specified number of bytes in the event of an error. Even when a parity error occurs in receive mode and the PER flag is set to 1, the data that has been received is transferred to RDR and can be read from there.

Note: For details on receive operations in block transfer mode, refer to section 15.4, Operation in Asynchronous Mode.

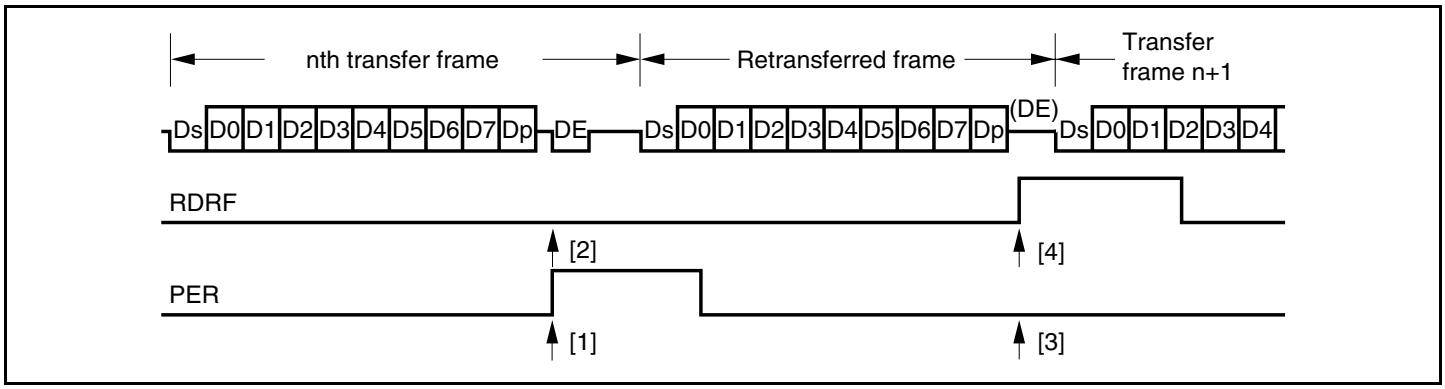


Figure 15.29 Retransfer Operation in SCI Receive Mode

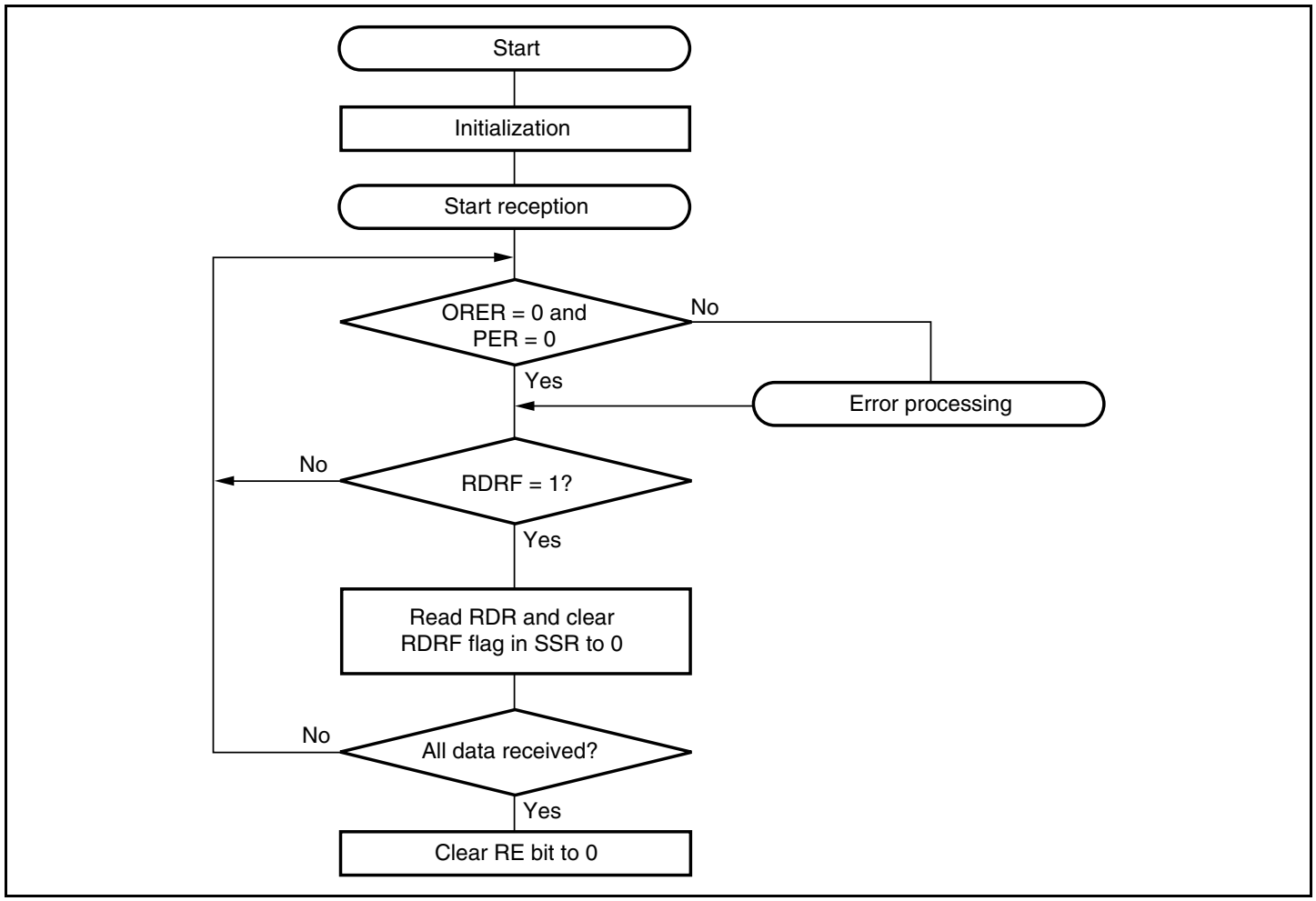


Figure 15.30 Example of Reception Processing Flow

15.7.8 Clock Output Control

When the GM bit in SMR is set to 1, the clock output level can be fixed with bits CKE1 and CKE0 in SCR. At this time, the minimum clock pulse width can be made the specified width.

Figure 15.31 shows the timing for fixing the clock output level. In this example, GM is set to 1, CKE1 is cleared to 0, and the CKE0 bit is controlled.

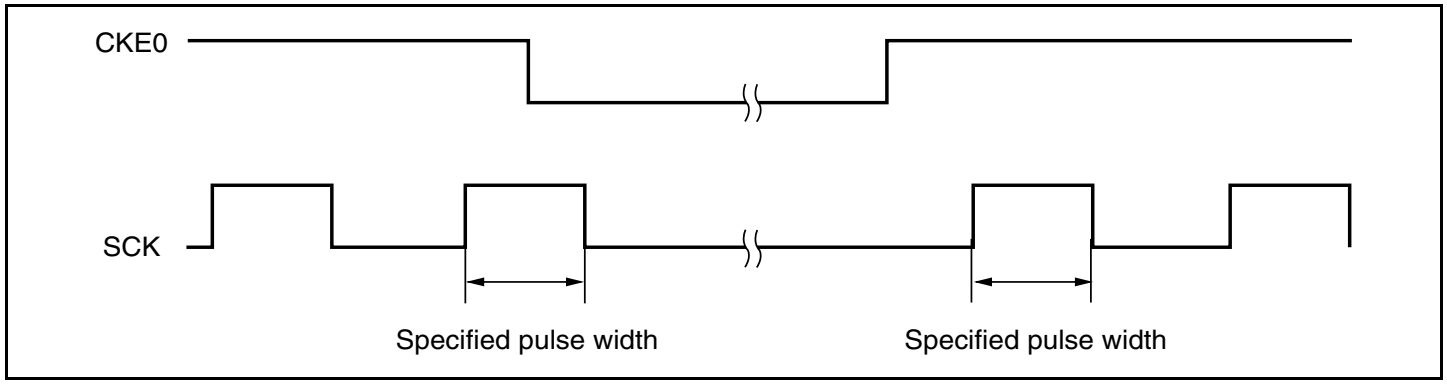


Figure 15.31 Timing for Fixing Clock Output Level

15.8 IrDA Operation

When the IrDA function is enabled with bit IrE in IrCR, the SCI_0 TxD0 and RxD0 signals are subjected to waveform encoding/decoding conforming to IrDA specification version 1.0 (IrTxD and IrRxD pins). By connecting these pins to an infrared transceiver/receiver, it is possible to implement infrared transmission/reception conforming to the IrDA specification version 1.0 system.

In the IrDA specification version 1.0 system, communication is started at a transfer rate of 9600 bps, and subsequently the transfer rate can be varied as necessary. As the IrDA interface in this LSI does not include a function for varying the transfer rate automatically, the transfer rate setting must be changed by software.

Figure 15.33 shows a block diagram of the IrDA function.

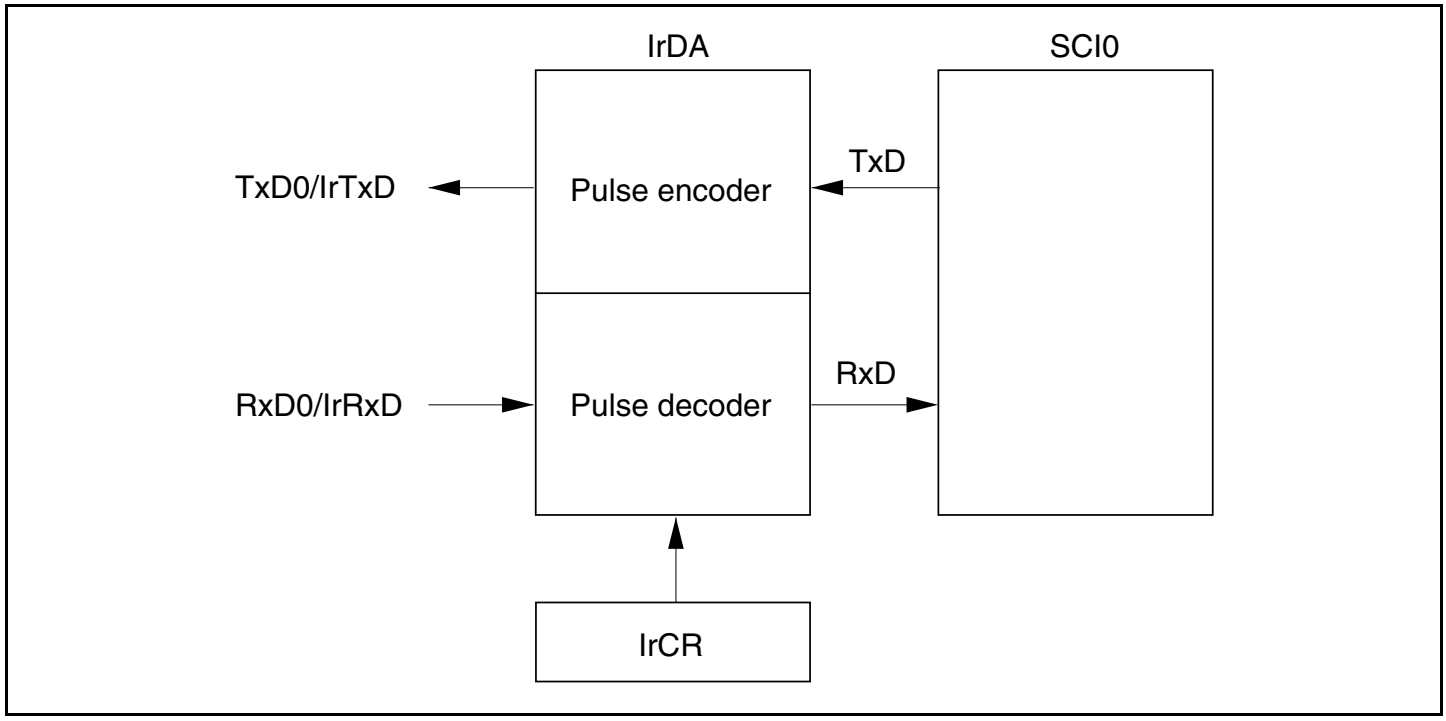


Figure 15.33 Block Diagram of IrDA

Transmission: In transmission, the output signal (UART frame) from the SCI is converted to an IR frame by the IrDA interface (see figure 15.34).

When the serial data is 0, a high pulse of 3/16 the bit rate (interval equivalent to the width of one bit) is output (initial value). The high-level pulse can be varied according to the setting of bits IrCKS2 to IrCKS0 in IrCR.

In the specification, the high pulse width is fixed at a minimum of 1.41 µs, and a maximum of (3/16 + 2.5%) × bit rate or (3/16 × bit rate) + 1.08 µs. When system clock ø is 20 MHz, 1.6 µs can be set for a high pulse width with a minimum value of 1.41 µs.

When the serial data is 1, no pulse is output.

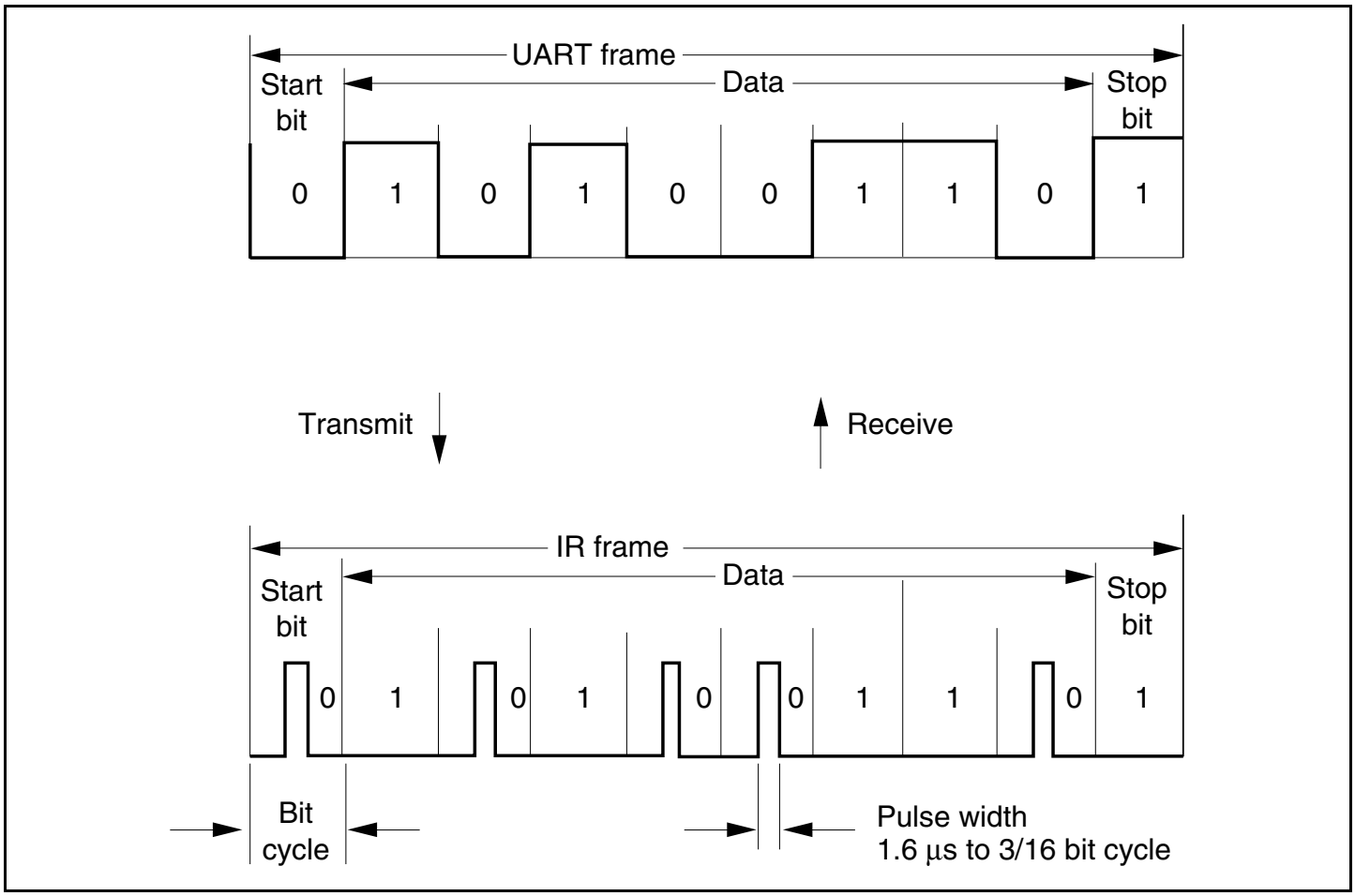


Figure 15.34 IrDA Transmit/Receive Operations

Reception: In reception, IR frame data is converted to a UART frame by the IrDA interface, and input to the SCI.

When a high pulse is detected, 0 data is output, and if there is no pulse during a one-bit interval, 1 data is output. Note that a pulse shorter than the minimum pulse width of 1.41 µs will be identified as a 0 signal.

High Pulse Width Selection: Table 15.12 shows possible settings for bits IrCKS2 to IrCKS0 (minimum pulse width), and operating frequencies of this LSI and bit rates, for making the pulse width shorter than 3/16 times the bit rate in transmission.

Table 15.12 Settings of Bits IrCKS2 to IrCKS0

Operating Frequency ø (MHz)	Bit Rate (bps) (Above) /Bit Period × 3/16 (µs) (Below)					
	2400	9600	19200	38400	57600	115200
	78.13	19.53	9.77	4.88	3.26	1.63
2	010	010	010	010	010	—
2.097152	010	010	010	010	010	—
2.4576	010	010	010	010	010	—
3	011	011	011	011	011	—
3.6864	011	011	011	011	011	011
4.9152	011	011	011	011	011	011
5	011	011	011	011	011	011
6	100	100	100	100	100	100
6.144	100	100	100	100	100	100
7.3728	100	100	100	100	100	100
8	100	100	100	100	100	100
9.8304	100	100	100	100	100	100
10	100	100	100	100	100	100
12	101	101	101	101	101	101
12.288	101	101	101	101	101	101
14	101	101	101	101	101	101
14.7456	101	101	101	101	101	101
16	101	101	101	101	101	101
16.9344	101	101	101	101	101	101
17.2032	101	101	101	101	101	101
18	101	101	101	101	101	101
19.6608	101	101	101	101	101	101
20	101	101	101	101	101	101
25	110	110	110	110	110	110

Legend

—: A bit rate setting cannot be made on the SCI side.

15.9 Interrupt Sources

15.9.1 Interrupts in Normal Serial Communication Interface Mode

Table 15.13 shows the interrupt sources in normal serial communication interface mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled using the enable bits in SCR.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the TEND flag in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the DTC or DMAC to perform data transfer. The TDRE flag is cleared to 0 automatically when data transfer is performed by the DTC or DMAC.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interrupt request can activate the DTC or DMAC to perform data transfer. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC or DMAC.

A TEI interrupt is generated when the TEND flag is set to 1 while the TEIE bit is set to 1. If a TEI interrupt and a TXI interrupt are generated simultaneously, the TXI interrupt has priority for acceptance. However, note that if the TDRE and TEND flags are cleared simultaneously by the TXI interrupt routine, the SCI cannot branch to the TEI interrupt routine later.

Table 15.13 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority
0	ERI0	Receive Error	ORER, FER, PER	Not possible	Not possible	High
	RXI0	Receive Data Full	RDRF	Possible	Possible	
	TXI0	Transmit Data Empty	TDRE	Possible	Possible	
	TEI0	Transmission End	TEND	Not possible	Not possible	
1	ERI1	Receive Error	ORER, FER, PER	Not possible	Not possible	
	RXI1	Receive Data Full	RDRF	Possible	Possible	
	TXI1	Transmit Data Empty	TDRE	Possible	Possible	
	TEI1	Transmission End	TEND	Not possible	Not possible	
2	ERI2	Receive Error	ORER, FER, PER	Not possible	Not possible	
	RXI2	Receive Data Full	RDRF	Possible	Not possible	
	TXI2	Transmit Data Empty	TDRE	Possible	Not possible	
	TEI2	Transmission End	TEND	Not possible	Not possible	Low

15.9.2 Interrupts in Smart Card Interface Mode

Table 15.14 shows the interrupt sources in Smart Card interface mode. The transmit end interrupt (TEI) request cannot be used in this mode.

Table 15.14 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority
0	ERI0	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	High
	RXI0	Receive Data Full	RDRF	Possible	Possible	
	TXI0	Transmit Data Empty	TEND	Possible	Possible	
1	ERI1	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	
	RXI1	Receive Data Full	RDRF	Possible	Possible	
	TXI1	Transmit Data Empty	TEND	Possible	Possible	
2	ERI2	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	
	RXI2	Receive Data Full	RDRF	Possible	Not possible	
	TXI2	Transmit Data Empty	TEND	Possible	Not possible	Low

In Smart Card interface mode, as in normal serial communication interface mode, transfer can be carried out using the DTC or DMAC. In transmit operations, the TDRE flag is also set to 1 at the same time as the TEND flag in SSR, and a TXI interrupt is generated. If the TXI request is designated beforehand as a DTC or DMAC activation source, the DTC or DMAC will be activated by the TXI request, and transfer of the transmit data will be carried out. The TDRE and TEND flags are automatically cleared to 0 when data transfer is performed by the DTC or DMAC. In the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC will automatically transmit the specified number of bytes in the event of an error, including retransmission. However, the ERS flag is not cleared automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When performing transfer using the DTC or DMAC, it is essential to set and enable the DTC or DMAC before carrying out SCI setting. For details on the DTC or DMAC setting procedures, refer to section 9, Data Transfer Controller (DTC) or section 7, DMA Controller (DMAC).

In receive operations, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DTC or DMAC activation source, the DTC or DMAC will be activated by the RXI request, and transfer of the receive data will be carried out. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC or DMAC. If an error occurs, an error flag is set but the RDRF flag is not. Consequently, the DTC or

DMAC is not activated, but instead, an ERI interrupt request is sent to the CPU. Therefore, the error flag should be cleared.

15.10 Usage Notes

15.10.1 Module Stop Mode Setting

SCI operation can be disabled or enabled using the module stop control register. The initial setting is for SCI operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

15.10.2 Break Detection and Processing

When framing error detection is performed, a break can be detected by reading the RxD pin value directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag is set, and the PER flag may also be set. Note that, since the SCI continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

15.10.3 Mark State and Break Sending

When TE is 0, the TxD pin is used as an I/O port whose direction (input or output) and level are determined by DR and DDR. This can be used to set the TxD pin to mark state or send a break during serial data transmission. To maintain the communication line at mark state until TE is set to 1, set both PCR and PDR to 1. Since TE is cleared to 0 at this point, the TxD pin becomes an I/O port, and 1 is output from the TxD pin. To send a break during serial transmission, first set PCR to 1 and clear PDR to 0, and then clear TE to 0. When TE is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TxD pin becomes an I/O port, and 0 is output from the TxD pin.

15.10.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only)

Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

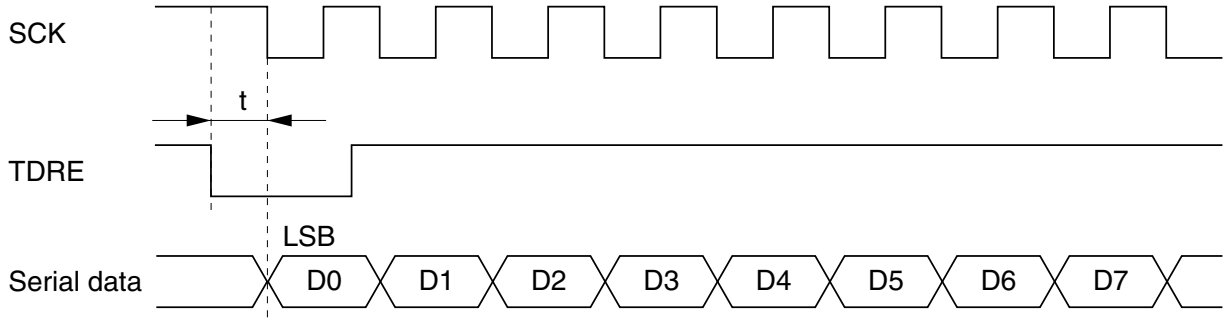
15.10.5 Relation between Writes to TDR and the TDRE Flag

The TDRE flag in SSR is a status flag that indicates that transmit data has been transferred from TDR to TSR. When the SCI transfers data from TDR to TSR, the TDRE flag is set to 1.

Data can be written to TDR regardless of the state of the TDRE flag. However, if new data is written to TDR when the TDRE flag is cleared to 0, the data stored in TDR will be lost since it has not yet been transferred to TSR. It is therefore essential to check that the TDRE flag is set to 1 before writing transmit data to TDR.

15.10.6 Restrictions on Use of DMAC or DTC

- When an external clock source is used as the serial clock, the transmit clock should not be input until at least 5 ø clock cycles after TDR is updated by the DMAC or DTC. Misoperation may occur if the transmit clock is input within 4 ø clocks after TDR is updated. (Figure 15.35)
- When RDR is read by the DMAC or DTC, be sure to set the activation source to the relevant SCI receive-data-full interrupt (RXI).



Note: When operating on an external clock, set t > 4 clocks.

Figure 15.35 Example of Synchronous Transmission Using DTC

15.10.7 Operation in Case of Mode Transition

- Transmission

  Operation should be stopped (by clearing TE, TIE, and TEIE to 0) before making a module stop mode or software standby mode transition. TSR, TDR, and SSR are reset. The output pin states in module stop mode or software standby mode depend on the port settings, and become high-level output after the relevant mode is cleared. If a transition is made during transmission, the data being transmitted will be undefined.

  When transmitting without changing the transmit mode after the relevant mode is cleared, transmission can be started by setting TE to 1 again, and performing the following sequence: SSR read → TDR write → TDRE clearance. To transmit with a different transmit mode after clearing the relevant mode, the procedure must be started again from initialization.

  Figure 15.36 shows a sample flowchart for mode transition during transmission. Port pin states during mode transition are shown in figures 15.37 and 15.38.

Operation should also be stopped (by clearing TE, TIE, and TEIE to 0) before making a transition from transmission by DTC transfer to module stop mode or software standby mode transition. To perform transmission with the DTC after the relevant mode is cleared, setting TE and TIE to 1 will set the TXI flag and start DTC transmission.

- Reception

  Receive operation should be stopped (by clearing RE to 0) before making a module stop mode or software standby mode transition. RSR, RDR, and SSR are reset. If a transition is made during reception, the data being received will be invalid.

  To continue receiving without changing the reception mode after the relevant mode is cleared, set RE to 1 before starting reception. To receive with a different receive mode, the procedure must be started again from initialization.

Figure 15.39 shows a sample flowchart for mode transition during reception.

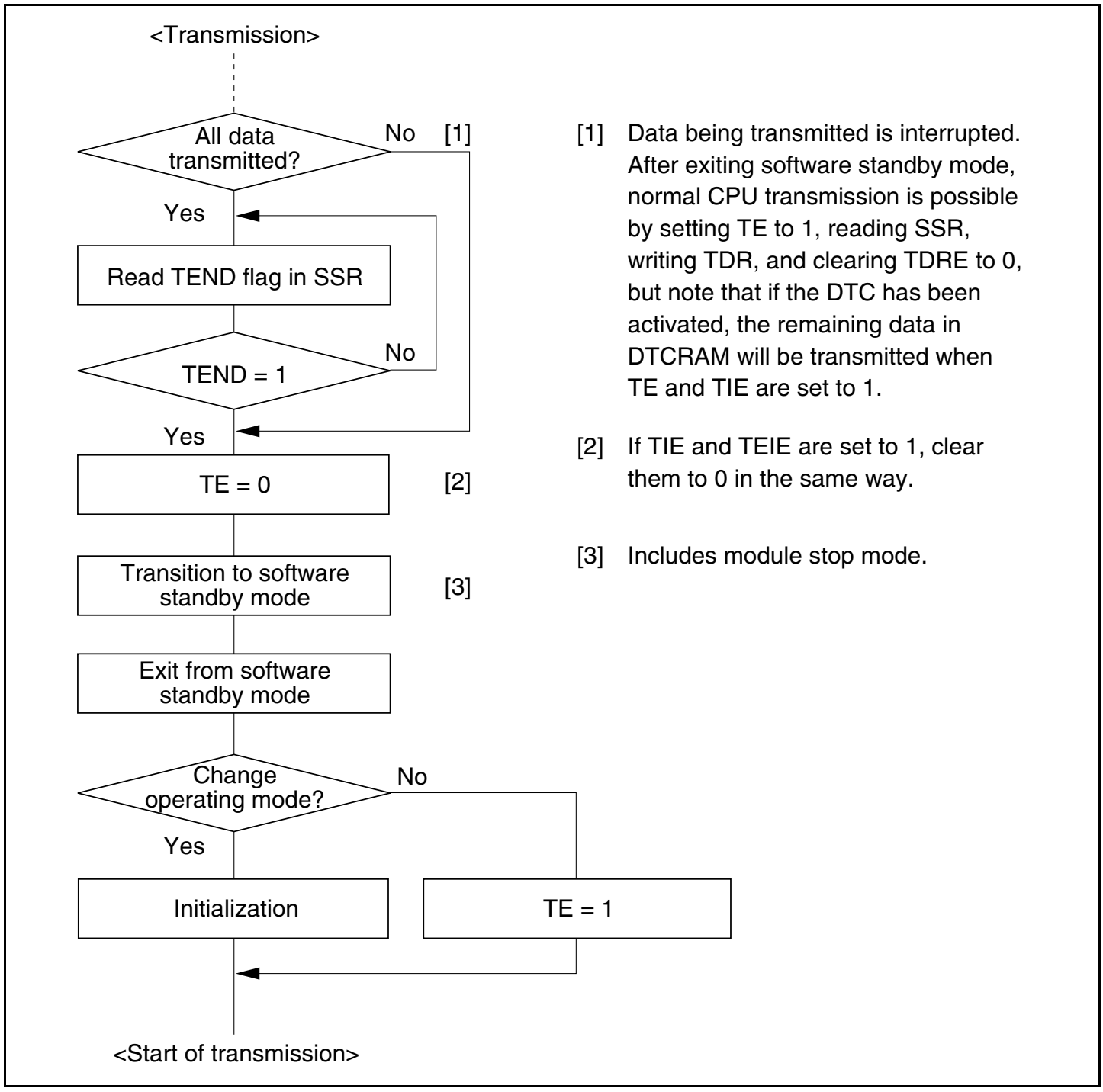


Figure 15.36 Sample Flowchart for Mode Transition during Transmission

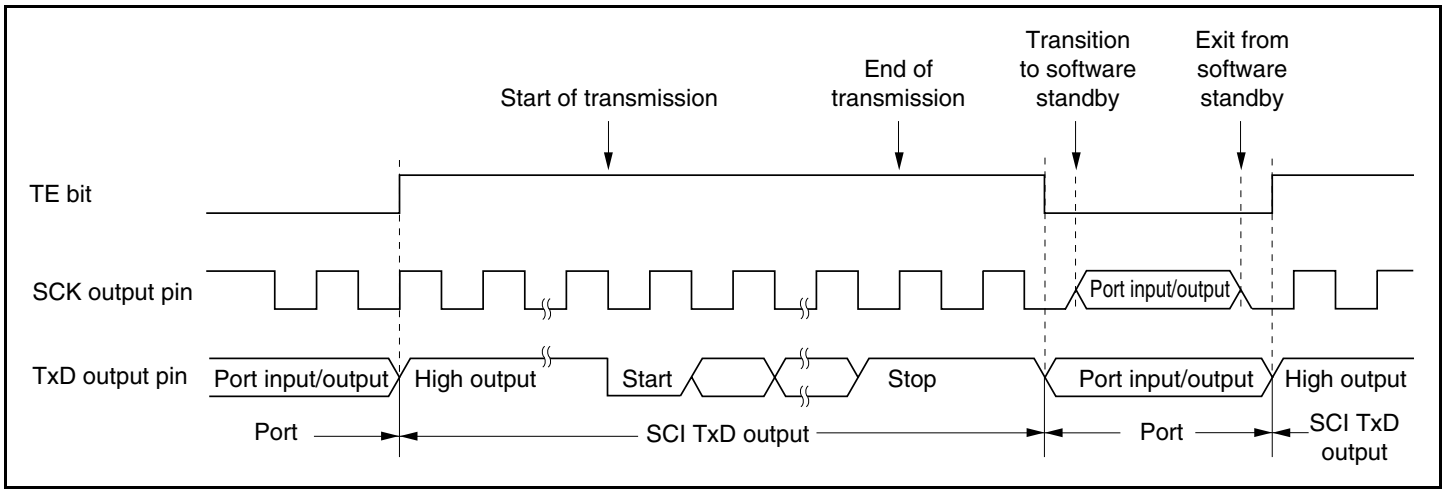


Figure 15.37 Port Pin States during Mode Transition (Internal Clock, Asynchronous Transmission)

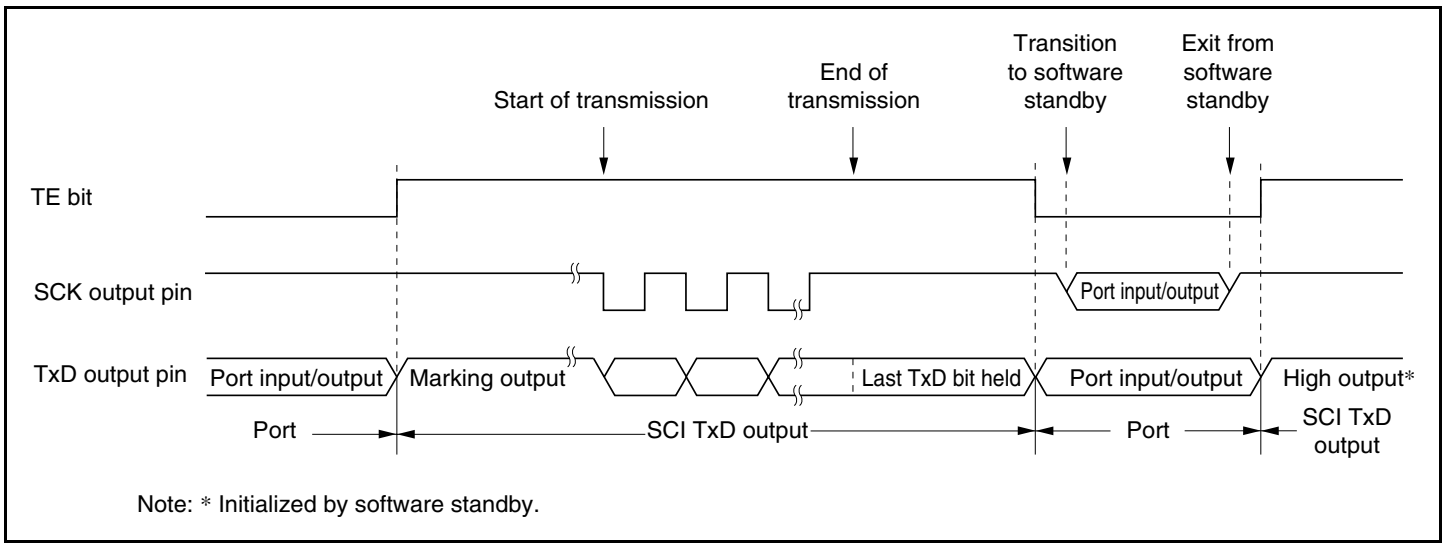


Figure 15.38 Port Pin States during Mode Transition (Internal Clock, Synchronous Transmission)

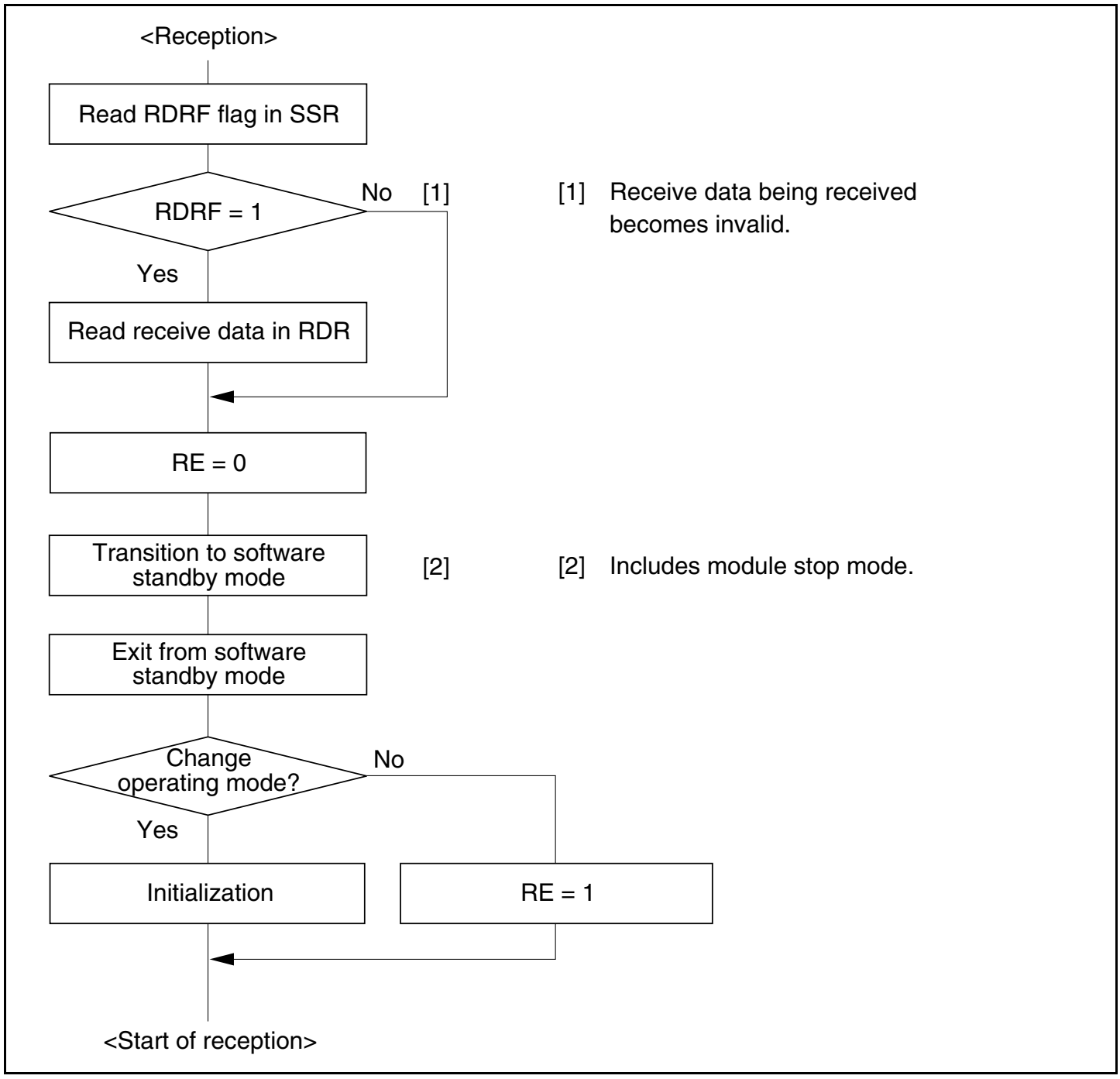


Figure 15.39 Sample Flowchart for Mode Transition during Reception

Section 16 A/D Converter

This LSI includes a successive approximation type 10-bit A/D converter that allows up to twelve analog input channels to be selected. The block diagram of A/D converter is shown in figure 16.1.

16.1 Features

- 10-bit resolution
- Twelve input channels
- Conversion time: 6.7 µs per channel (at 20 MHz operation)
- Two kinds of operating modes
  - Single mode: Single-channel A/D conversion
  - Scan mode: Continuous A/D conversion on 1 to 4 channels, or 1 to 8 channels (H8S/2678R Series)
- Four data registers (H8S/2678 Series) or eight data registers (H8S/2678R Series)
  - Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of conversion start
  - Conversion can be started by software, 16-bit timer pulse unit (TPU), conversion start trigger by 8-bit timer (TMR), or external trigger signal.
- Interrupt request
  - A/D conversion end interrupt (ADI) request can be generated
- Module stop mode can be set

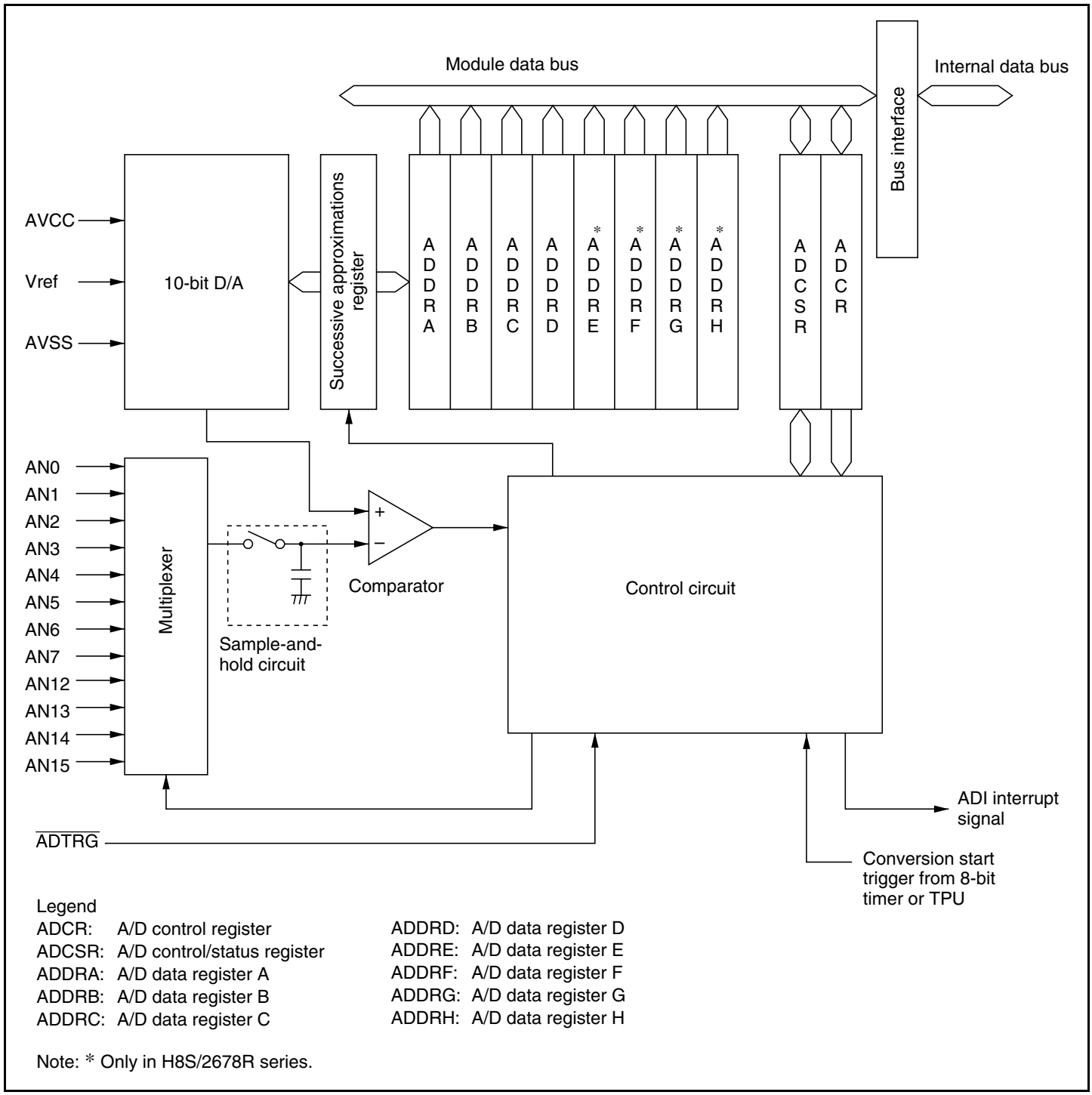


Figure 16.1 Block Diagram of A/D Converter

16.2 Input/Output Pins

Table 16.1 shows the pin configuration of the A/D converter.

The twelve analog input pins are divided into two channel sets: channel set 0 (AN0 to AN7) and channel set 1 (AN12 to AN15).

In the H8S/2678 Series, each channel set is divided into four channels × two groups: group 0 in channel set 0 (AN0 to AN3), group 1 in channel set 0 (AN4 to AN7), and group1 in channel set 1 (AN12 to AN15).

The AV_{CC} and AV_{SS} pins are the power supply pins for the analog block in the A/D converter. The Vref pin is the A/D conversion reference voltage pin.

Table 16.1 A/D Converter Pins

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog block power supply
Analog ground pin	AV_{SS}	Input	Analog block ground
Reference voltage pin	Vref	Input	A/D conversion reference voltage
Analog input pin 0	AN0	Input	Channel set 0 analog inputs
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 12	AN12	Input	Channel set 1 analog inputs
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	
A/D external trigger input pin	ADTRG	Input	External trigger input for starting A/D conversion

16.3 Register Descriptions

The A/D converter has the following registers.

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRD)
- A/D data register E (ADDRE)
- A/D data register F (ADDRF)
- A/D data register G (ADDRG)

- A/D data register H (ADDRH)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

16.3.1 A/D Data Registers A to H (ADDRA to ADDRH)

There are four 16-bit read-only ADDR registers, ADDRA to ADDRD (H8S/2678 Series) and eight 16-bit read-only ADDR registers, ADDRA to ADDRH (H8S/2678R Series), used to store the results of A/D conversion. The ADDR registers, which store a conversion result for each channel, are shown in table 16.2.

The converted 10-bit data is stored to bits 15 to 6. The lower 6-bit data is always read as 0. ADDR must not be accessed in 8-bit units and must be accessed in 16-bit units.

In the H8S/2678 Series, the data bus between the CPU and the A/D converter is 8-bit width. The upper byte can be read directly from the CPU, but the lower byte should be read via a temporary register. The temporary register contents are transferred from the ADDR when the upper byte data is read. When reading the ADDR, read the only upper byte, or read in word unit.

In the H8S/2678R Series, the data bus between the CPU and the A/D converter is 16-bit width. The data can be read directly from the CPU.

Table 16.2 Analog Input Channels and Corresponding ADDR Registers

- H8S/2678 Series

Analog Input Channel				A/D Data Register which stores conversion result
Channel Set 0 (CH3 = 1)		Channel Set 1 (CH3 = 0)		
Group 0 (CH2 = 0)	Group 1 (CH2 = 1)	Group 0 (CH2 = 0)	Group 1 (CH2 = 1)	
AN0	AN4	Setting prohibited	AN12	ADDRA
AN1	AN5	Setting prohibited	AN13	ADDRB
AN2	AN6	Setting prohibited	AN14	ADDRC
AN3	AN7	Setting prohibited	AN15	ADDRD

Analog Input Channel		A/D Data Register which stores conversion result
Channel Set 0 (CH3 = 0)	**Channel Set 1 (CH3 = 1)**	
AN0	Nothing	ADDRA
AN1	Nothing	ADDRB

A71 AN1 ADDHB

A61 AN1 ADDGB

Bit	Bit Name	Initial Value	R/W	Description
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion, and the A/D converter enters wait state. When this bit is set to 1 by software, TPU (trigger), TMR (trigger), or the ADTRG pin, A/D conversion starts. This bit remains set to 1 during A/D conversion. In single mode, cleared to 0 automatically when conversion on the specified channel ends. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by a reset, or a transition to hardware standby mode or software.
4	SCAN	0	R/W	Scan Mode Selects single mode or scan mode as the A/D conversion operating mode. 0: Single mode 1: Scan mode
3	CKS	0	R/W	Clock Select Used together with the CKS1 bit in ADCR to set the A/D conversion time. When CKS1 = 0 0: 530 states (max) 1: 68 states (max) When CKS = 1 0: 266 states (max) 1: 134 states (max)

Bit	Bit Name	Initial Value	R/W	Description
2	CH2	0	R/W	Channel Select 2 to 0
1	CH1	0	R/W	These bits are used together with the SCAN bit in ADCSR and the CH3 bit in ADCR to select the analog input channels.
0	CH0	0	R/W	When SCAN = 0, CH3 = 0 / When SCAN = 1, CH3 = 0 0XXX: Setting prohibited / 0XXX: Setting prohibited 100: AN12 / 100: AN12 101: AN13 / 101: AN12 and AN13 110: AN14 / 110: AN12 to AN14 111: AN15 / 111: AN12 to AN15 When SCAN = 0, CH3 = 1 / When SCAN = 1, CH3 = 1 000: AN0 / 000: AN0 001: AN1 / 001: AN0 and AN1 010: AN2 / 010: AN0 to AN2 011: AN3 / 011: AN0 to AN3 100: AN4 / 100: AN4 101: AN5 / 101: AN4 and AN5 110: AN6 / 110: AN4 to AN6 111: AN7 / 111: AN4 to AN7

Note: Only 0 can be written in bit 7, to clear the flag.

Legend: X: Don't care.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conversion. [Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all specified channels in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When the DTC or DMAC is activated by an ADI interrupt and ADDR is read
6	ADIE	0	R/W	A/D Interrupt Enable A/D conversion end interrupt (ADI) request enabled when 1 is set
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion, and the A/D converter enters wait state. Setting this bit to 1 starts an A/D conversion. In single mode, cleared to 0 automatically when conversion on the specified channel ends. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by software, a reset, or a transition to software standby mode, hardware standby mode or module stop mode.
4	—	0	—	Reserved This bit is always read as 0 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
3	CH3	0	R/W	Channel select 3 to 0
2	CH2	0	R/W	Selects analog input together with bits SCANE and SCANS in ADCR.
1	CH1	0	R/W	
0	CH0	0	R/W	When SCANE = 0 and SCANS = X
				0000: AN0 1000: Setting prohibited
				0001: AN1 1001: Setting prohibited
				0010: AN2 1010: Setting prohibited
				0011: AN3 1011: Setting prohibited
				0100: AN4 1100: AN12
				0101: AN5 1101: AN13
				0110: AN6 1110: AN14
				0111: AN7 1111: AN15
				When SCANE = 1 and SCANS = 0
				0000: AN0 1000: Setting prohibited
				0001: AN0 and AN1 1001: Setting prohibited
				0010: AN0 to AN2 1010: Setting prohibited
				0011: AN0 to AN3 1011: Setting prohibited
				0100: AN4 1100: AN12
				0101: AN4 and AN5 1101: AN12 and AN13
				0110: AN4 to AN6 1110: AN12 to AN14
				0111: AN4 to AN7 1111: AN12 to AN15
				When SCANE = 1 and SCANS = 1
				0000: AN0 1000: Setting prohibited
				0001: AN0 and AN1 1001: Setting prohibited
				0010: AN0 to AN2 1010: Setting prohibited
				0011: AN0 to AN3 1011: Setting prohibited
				0100: AN0 to AN4 1100: Setting prohibited
				0101: AN0 to AN5 1101: Setting prohibited
				0110: AN0 to AN6 1110: Setting prohibited
				0111: AN0 to AN7 1111: Setting prohibited

Note: Only 0 can be written in bit 7, to clear the flag.

Legend: X: Don’t care.

16.3.3 A/D Control Register (ADCR)

ADCR enables A/D conversion start by an external trigger input.

- H8S/2678 Series

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0:
6	TRGS0	0	R/W	These bits select enabling or disabling of the start of A/D conversion by a trigger signal. 00: A/D conversion start by external trigger is disabled 01: A/D conversion start by external trigger (TPU) is enabled 10: A/D conversion start by external trigger (TMR) is enabled 11: A/D conversion start by external trigger pin (ADTRG) is enabled
5	—	1	—	Reserved
4	—	1	—	These bits are always read as 1 and cannot be modified.
3	CKS1	1	R/W	Clock Select 1 Used together with the CKS bit in ADCSR to set the A/D conversion time. See the description of the CKS bit for details.
2	CH3	1	R/W	Channel Select 3 Used together with bits CH2, CH1, and CH0 in ADCSR to select the analog input channel(s). See the description of bits CH2, CH1, and CH0 for details.
1	—	1	—	Reserved
0	—	1	—	These bits are always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0
6	TRGS0	0	R/W	These bits select enabling or disabling of the start of A/D conversion by a trigger signal. 00: A/D conversion start by external trigger is disabled 01: A/D conversion start by external trigger (TPU) is enabled 10: A/D conversion start by external trigger (TMR) is enabled 11: A/D conversion start by external trigger pin (ADTRG) is enabled
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	Selects single mode or scan mode as the A/D conversion operating mode. 0x: Single mode 10: Scan mode. A/D conversion is performed continuously for channels 1 to 4 11: Scan mode. A/D conversion is performed continuously for channels 1 to 8.
3	CKS1	0	R/W	Clock Select 1 to 0
2	CKS0	0	R/W	Sets the A/D conversion time. Only set bits CKS1 and CKS0 while conversion is stopped (ADST = 0). 00: A/D conversion time = 530 states (max) 01: A/D conversion time = 266 states (max) 10: A/D conversion time = 134 states (max) 11: A/D conversion time = 68 states (max)
1	—	0	—	Reserved
0	—	0	—	These bits are always read as 0 and cannot be modified.

Legend: X: Don’t care.

16.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes: single mode and scan mode. When changing the operating mode or analog input channel, to prevent incorrect operation, first clear the bit ADST to 0 in ADCSR to halt A/D conversion. The ADST bit can be set at the same time as the operating mode or analog input channel is changed.

16.4.1 Single Mode

In single mode, A/D conversion is to be performed only once on the specified single channel. Operations are as follows.

1. A/D conversion is started when the ADST bit in ADCSR is set to 1, according to the software or external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A/D data register to the channel.
3. On completion of conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion, and is automatically cleared to 0 when conversion ends. When the ADST bit is cleared to 0 during A/D conversion, A/D conversion stops and the A/D converter enters wait state.

16.4.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the specified channels: maximum four channels or maximum eight channels (H8S/2678R Series). Operations are as follows.

1. When the ADST bit in ADCSR is set to 1 by a software, TPU or external trigger input, A/D conversion starts on the first channel in the group.

   In the H8S/2678 Series, the A/D conversion starts on AN0 when CH3 and CH2 =10, AN4 when CH3 and CH2 = 11, or AN12 when CH3 and CH2 = 01.

   In the H8S/2678R Series, the consecutive A/D conversion on maximum four channels (SCANE and SCANS = 10) or on maximum eight channels (SCANE and SCANS = 11) can be selected. When the consecutive A/D conversion is performed on the four channels, the A/D conversion starts on AN0 when CH3 and CH2 =00, AN4 when CH3 and CH2 = 01, or AN12 when CH3 and CH2 = 11. When the consecutive A/D conversion is performed on the eight channels, the A/D conversion starts on AN0 when SH3 and SH2 =00.
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the corresponding A/D data register to each channel.

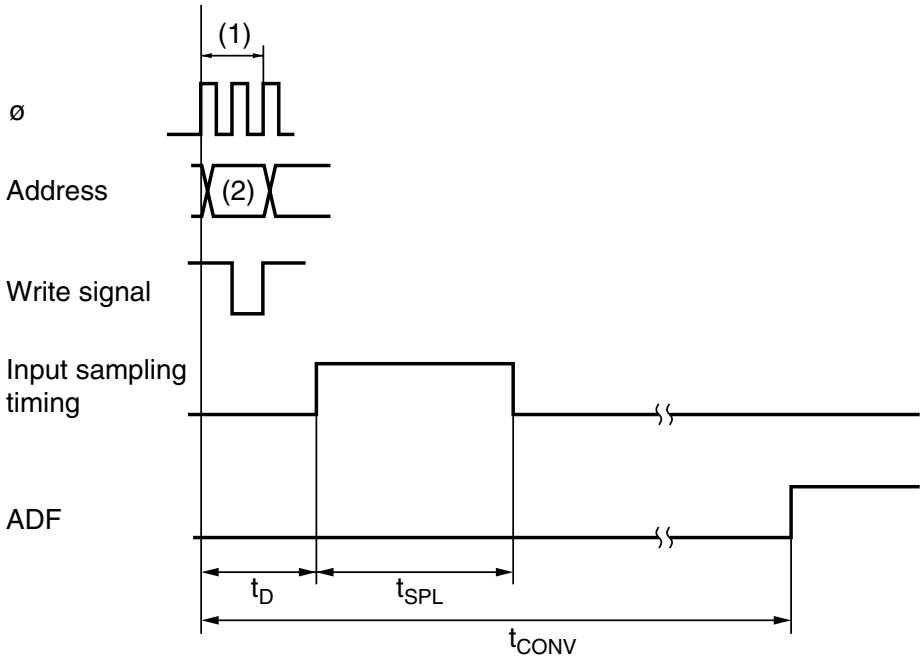
3. When conversion of all the selected channels is completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested. Conversion of the first channel in the group starts again.
4. The ADST bit is not cleared automatically, and steps [2] to [3] are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter enters wait state. If the ADST bit is later set to 1, A/D conversion starts again from the first channel in the group.

16.4.3 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 when A/D conversion start delay time (t_D) passes after the ADST bit is set to 1, then starts conversion. Figure 16.2 shows the A/D conversion timing. Table 16.3 indicates the A/D conversion time.

As indicated in figure 16.2, the A/D conversion time (t_{CONV}) includes t_D and the input sampling time (t_{SPL}). 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 tables 16.3.

In scan mode, the values given in tables 16.3 apply to the first conversion time. The values given in tables 16.4 apply to the second and subsequent conversions. The conversion time must be within the ranges indicated in the descriptions, A/D Conversion Characteristics in section 24, Electrical Characteristics. Therefore the CKS and CKS1 bits (H8S/2678 Series) or CKS1 and CKS0 bits (H8S/2678R Series) must be set to satisfy this condition.



Legend
(1) : ADCSR write cycle
(2) : ADCSR address
t_D : A/D conversion start delay time
t_{SPL} : Input sampling time
t_{CONV} : A/D conversion time

Figure 16.2 A/D Conversion Timing

Table 16.3 A/D Conversion Time (Single Mode)

- H8S/2678 Series

		CKS1 = 0						CKS1 = 1					
		CKS = 0			CKS = 1			CKS = 0			CKS = 1		
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
A/D conversion start delay time	t_D	18	—	33	4	—	5	10	—	17	6	—	9
Input sampling time	t_{SPL}	—	127	—	—	15	—	—	63	—	—	31	—
A/D conversion time	t_{CONV}	515	—	530	67	—	68	259	—	266	131	—	134

Note: Values in the table are the number of states.

- H8S/2678R Series

		CKS1 = 0						CKS1 = 1					
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
A/D conversion start delay time	t_D	18	—	33	10	—	17	6	—	9	4	—	5
Input sampling time	t_{SPL}	—	127	—	—	63	—	—	31	—	—	15	—
A/D conversion time	t_{CONV}	515	—	530	259	—	266	131	—	134	67	—	68

Note: Values in the table are the number of states.

Table 16.4 A/D Conversion Time (Scan Mode)

- H8S/2678 Series

CKS1	CKS	Conversion Time (State)
0	0	512 (Fixed)
	1	64 (Fixed)
1	0	256 (Fixed)
	1	128 (Fixed)

- H8S/2678R Series

CKS1	CKS0	Conversion Time (State)
0	0	512 (Fixed)
	1	256 (Fixed)
1	0	128 (Fixed)
	1	64 (Fixed)

16.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set to 11 in ADCR, external trigger input is enabled at the ADTRG pin. A falling edge at the ADTRG pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as when the bit ADST has been set to 1 by software. Figure 16.3 shows the timing.

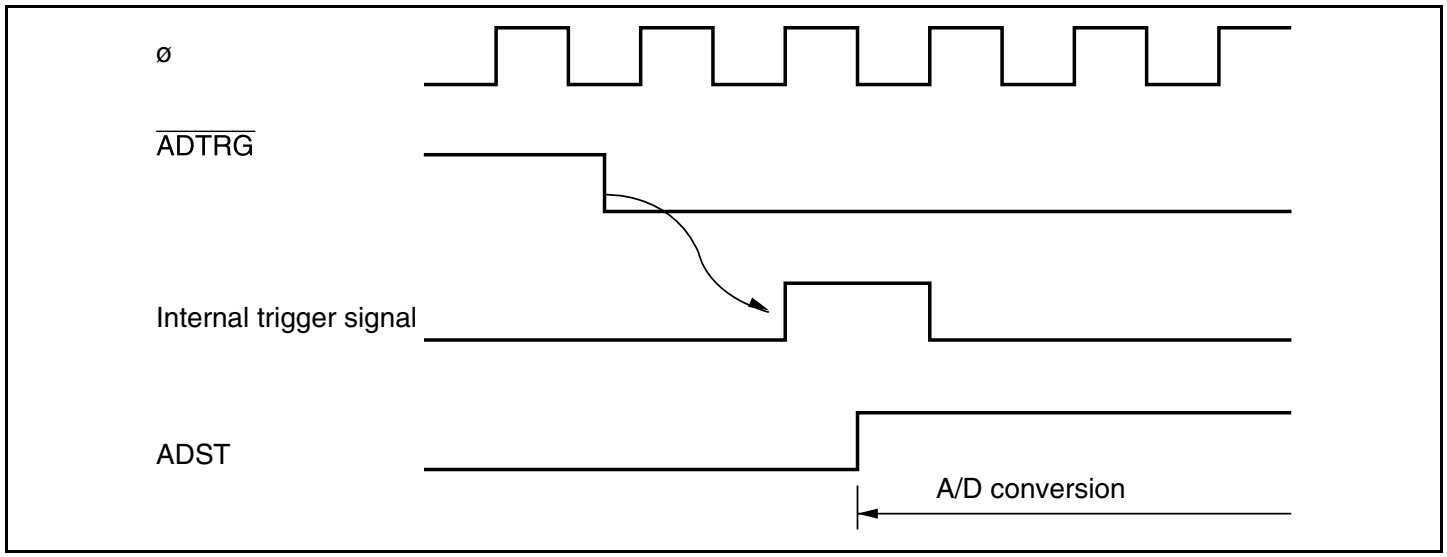


Figure 16.3 External Trigger Input Timing

16.5 Interrupt Source

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. Setting the ADIE bit to 1 enables an ADI interrupt requests while the bit ADF in ADCSR is set to 1 after A/D conversion is completed. The DTC or DMAC can be activated by an ADI interrupt. Having the converted data read by the DTC or DMAC in response to an ADI interrupt enables continuous conversion to be achieved without imposing a load on software.

Table 16.5 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ADI	End of conversion	ADF	Possible	Possible

16.6 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution
  The number of A/D converter digital output codes
- Quantization error
  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 16.4).
- Offset error
  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value B'0000000000 (H'000) to B'0000000001 (H'001) (see figure 16.5).
- Full-scale error

The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H'3FF) (see figure 16.5).

- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error (see figure 16.5).

- Absolute precision

  The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.

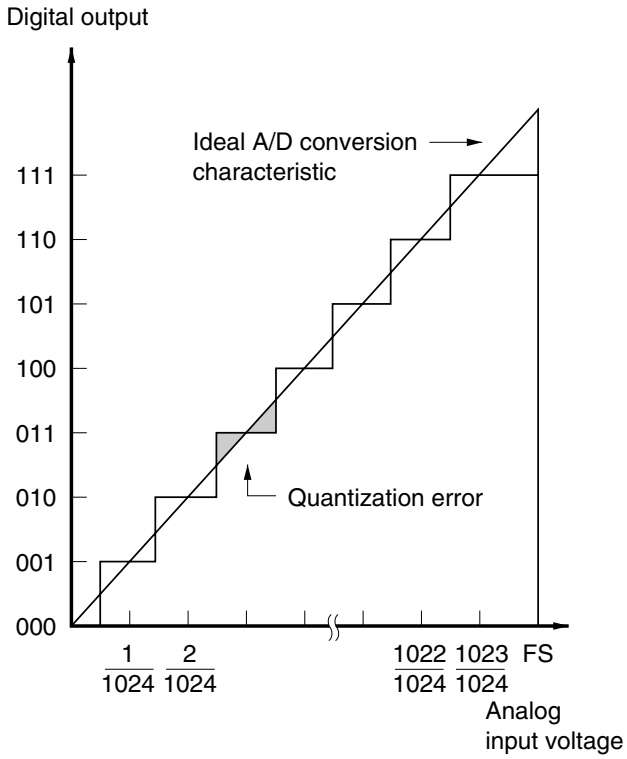


Figure 16.4 A/D Conversion Accuracy Definitions

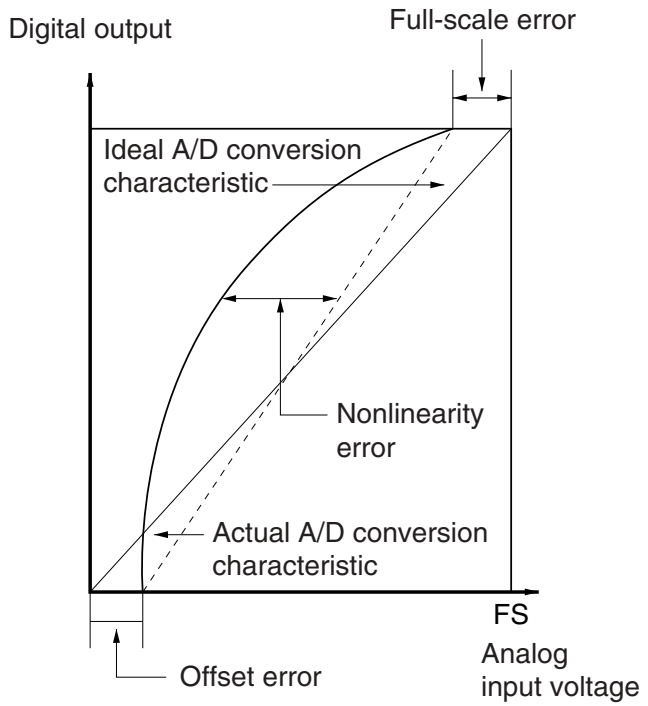


Figure 16.5 A/D Conversion Accuracy Definitions

16.7 Usage Notes

16.7.1 Module Stop Mode Setting

Operation of the A/D converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the A/D converter to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

16.7.2 Permissible Signal Source Impedance

This LSI's analog input is designed so that conversion precision is guaranteed for an input signal for which the signal source impedance is 10 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 10 kΩ, charging may be insufficient and it may not be possible to guarantee the A/D conversion accuracy. However, if a large capacitance is provided externally for conversion in single mode, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, since a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/μs or greater) (see figure 16.6). When converting a high-speed analog signal or conversion in scan mode, a low-impedance buffer should be inserted.

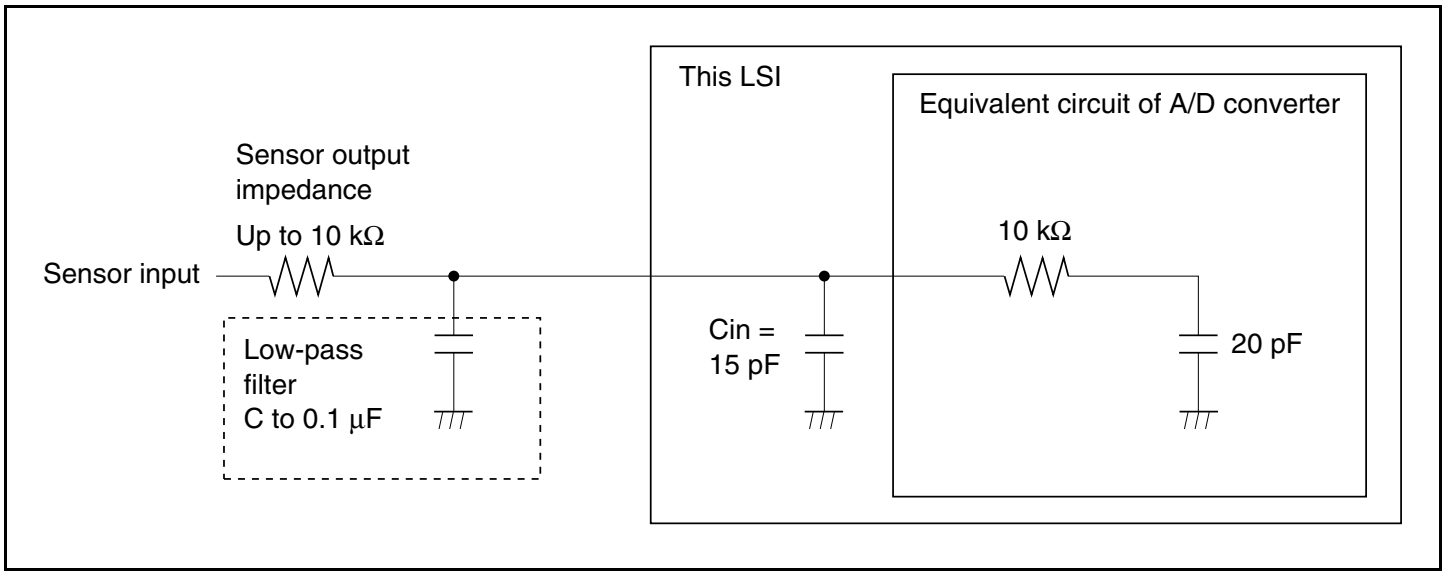


Figure 16.6 Example of Analog Input Circuit

16.7.3 Influences on Absolute Precision

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AVss.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board, so acting as antennas.

16.7.4 Setting Range of Analog Power Supply and Other Pins

If conditions shown below are not met, the reliability of the device may be adversely affected.

- Analog input voltage range
  The voltage applied to analog input pin ANn during A/D conversion should be in the range AVss ≤ V_{AN} ≤ Vref.
- Relation between AVcc, AVss and Vcc, Vss
  As the relationship between AVcc, AVss and Vcc, Vss, set AVcc ≥ Vcc and AVss = Vss. If the A/D converter is not used, the AVcc and AVss pins must not be left open.
- Vref setting range
  The reference voltage at the Vref pin should be set in the range Vref ≤ AVcc.

16.7.5 Notes on Board Design

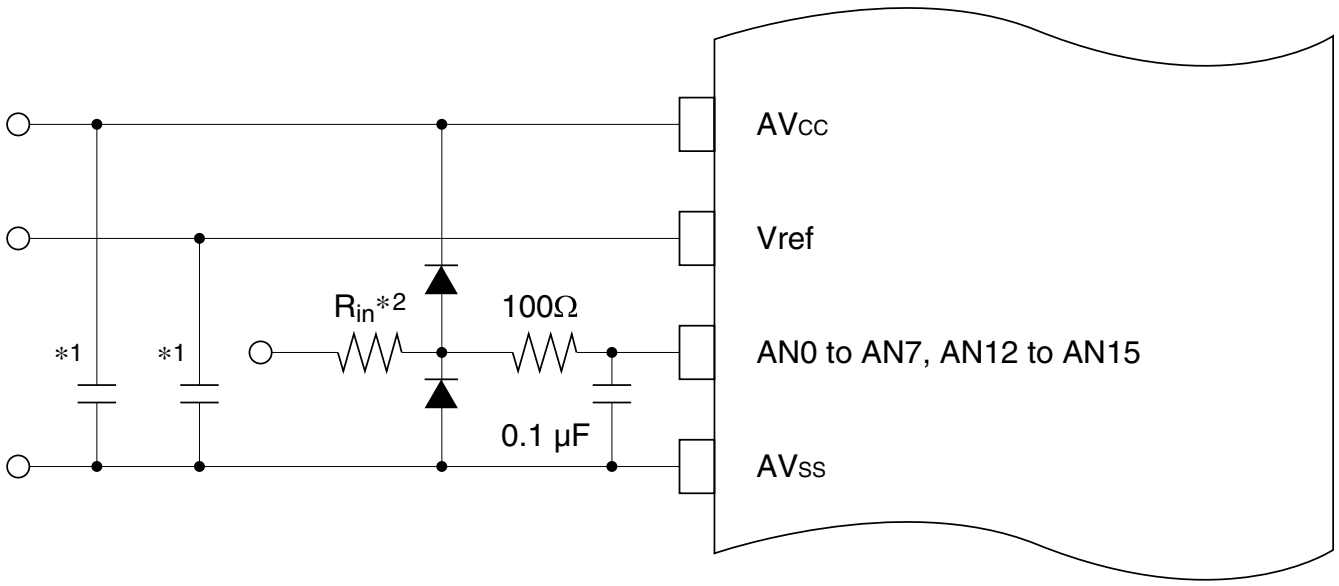
In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Also, digital circuitry must be isolated from the analog input signals (AN0 to AN7 and AN12 to AN15), analog reference power supply (Vref), and analog power supply (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable digital ground (Vss) on the board.

16.7.6 Notes on Noise Countermeasures

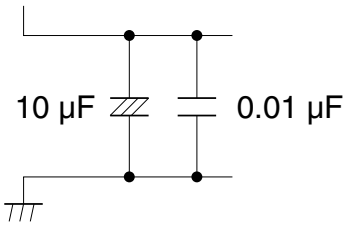
A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (AN0 to AN7 and AN12 to AN15) should be connected between AVcc and AVss as shown in figure 16.7. Also, the bypass capacitors connected to AVcc and the filter capacitor connected to AN0 to AN7 and AN12 to AN15 must be connected to AVss.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN7 and AN12 to AN15) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (R_{in}), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding the circuit constants.



Notes: Values are reference values.

1.

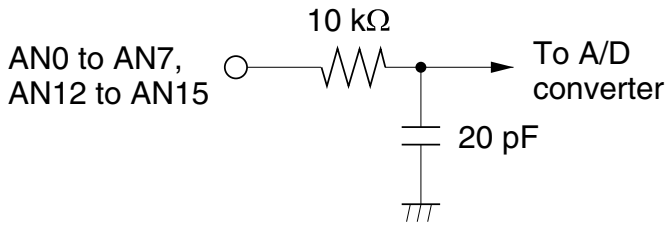


2. R_{in}: Input impedance

Figure 16.7 Example of Analog Input Protection Circuit

Table 16.6 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	10	kΩ



Note: Values are reference values.

Figure 16.8 Analog Input Pin Equivalent Circuit

Section 17 D/A Converter

17.1 Features

D/A converter features are listed below.

- 8-bit resolution
- Four output channels
- Maximum conversion time of 10 μs (with 20 pF load)
- Output voltage of 0 V to Vref
- D/A output hold function in software standby mode
- Setting the module stop mode

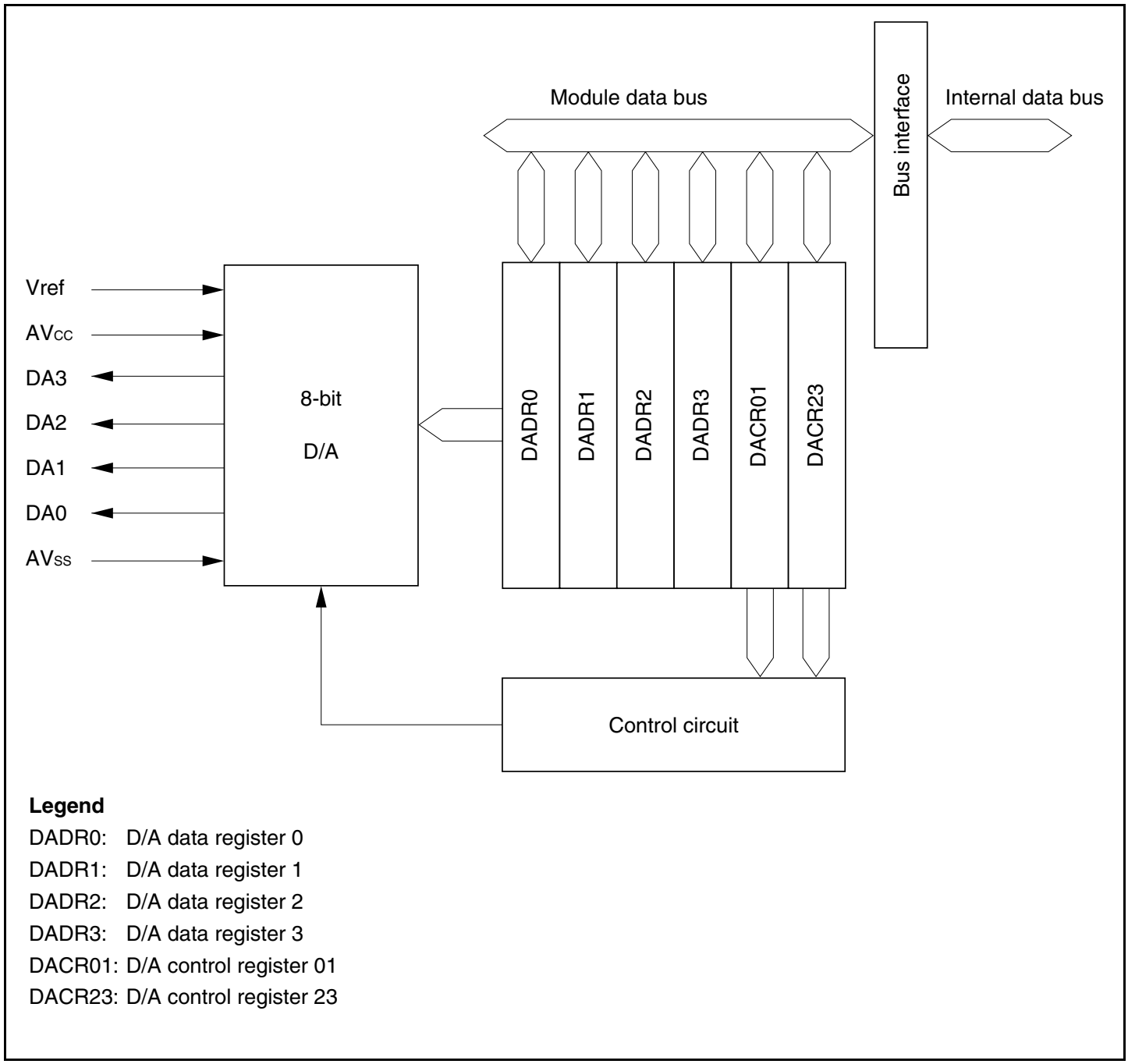


Figure 17.1 Block Diagram of D/A Converter

17.2 Input/Output Pins

Table 17.1 shows the pin configuration of the D/A converter.

DACR01

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: Analog output (DA1) is disabled 1: Channel 1 D/A conversion is enabled; analog output (DA1) is enabled
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output. 0: Analog output (DA0) is disabled 1: Channel 0 D/A conversion is enabled; analog output (DA0) is enabled
5	DAE	0	R/W	D/A Enable Used together with the DAOE0 and DAOE1 bits to control D/A conversion. When the DAE bit is cleared to 0, channel 0 and 1 D/A conversions are controlled independently. When the DAE bit is set to 1, channel 0 and 1 D/A conversions are controlled together. Output of conversion results is always controlled independently by the DAOE0 and DAOE1 bits. For details, see table 17.2 Control of D/A Conversion.
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Table 17.2 Control of D/A Conversion

Bit 5 DAE	Bit 7 DAOE1	Bit 6 DAOE0	Description
0	0	0	D/A conversion disabled
		1	Channel 0 D/A conversion enabled, channel1 D/A conversion disabled
	1	0	Channel 1 D/A conversion enabled, channel0 D/A conversion disabled
		1	Channel 0 and 1 D/A conversions enabled
1	0	0	D/A conversion disabled
		1	Channel 0 and 1 D/A conversions enabled
	1	0	
		1	

DACR23

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE3	0	R/W	D/A Output Enable 3 Controls D/A conversion and analog output. 0: Analog output (DA3) is disabled 1: Channel 3 D/A conversion is enabled; analog output (DA3) is enabled
6	DAOE2	0	R/W	D/A Output Enable 2 Controls D/A conversion and analog output. 0: Analog output (DA2) is disabled 1: Channel 2 D/A conversion is enabled; analog output (DA2) is enabled
5	DAE	0	R/W	D/A Enable Used together with the DAOE2 and DAOE3 bits to control D/A conversion. When the DAE bit is cleared to 0, channel 2 and 3 D/A conversions are controlled independently. When the DAE bit is set to 1, channel 2 and 3 D/A conversions are controlled together. Output of conversion results is always controlled independently by the DAOE2 and DAOE3 bits. For details, see table 17.3 Control of D/A Conversion.
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Table 17.3 Control of D/A Conversion

Bit 5 DAE	Bit 7 DAOE3	Bit 6 DAOE2	Description
0	0	0	D/A conversion disabled
		1	Channel 2 D/A conversion enabled, channel3 D/A conversion disabled
	1	0	Channel 3 D/A conversion enabled, channel2 D/A conversion disabled
		1	Channel 2 and 3 D/A conversions enabled
1	0	0	D/A conversion disabled
		1	Channel 2 and 3 D/A conversions enabled
	1	0	
		1	

17.4 Operation

The D/A converter includes D/A conversion circuits for four channels, each of which can operate independently.

When DAOE bit in DACR01 or DACR23 is set to 1, D/A conversion is enabled and the conversion result is output.

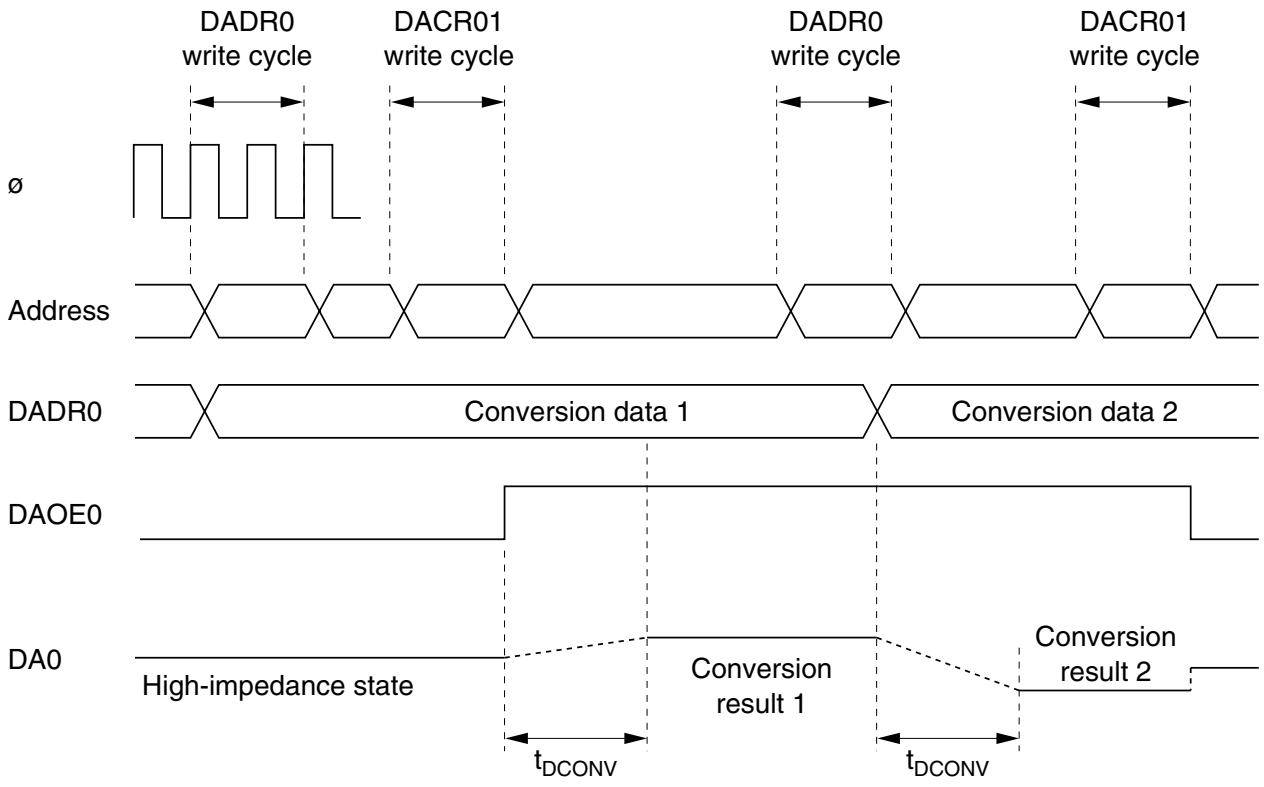
The operation example concerns D/A conversion on channel 0. Figure 17.2 shows the timing of this operation.

[1] Write the conversion data to DADR0.

[2] Set the DAOE0 bit in DACR01 to 1. D/A conversion is started. The conversion result is output from the analog output pin DA0 after the conversion time t_{DCONV} has elapsed. The conversion result is continued to output until DADR0 is written to again or the DAOE0 bit is cleared to 0. The output value is expressed by the following formula:

[3] If DADR0 is written to again, the conversion is immediately started. The conversion result is output after the conversion time t_{DCONV} has elapsed.

[4] If the DAOE0 bit is cleared to 0, analog output is disabled.



Legend
t_{DCONV}: D/A conversion time

Figure 17.2 Example of D/A Converter Operation

17.5 Usage Notes

17.5.1 Setting for Module Stop Mode

It is possible to enable/disable the D/A converter operation using the module stop control register, the D/A converter does not operate by the initial value of the register. The register can be accessed by releasing the module stop mode. For details, see section 22, Power-Down Modes.

17.5.2 D/A Output Hold Function in Software Standby Mode

If D/A conversion is enabled and this LSI enters software standby mode, D/A output is held and analog power supply current remains at the same level during D/A conversion. When the analog power supply current is required to go low in software standby mode, bits DAOE0 to DAOE3 and DAE should be cleared to 0, and D/A output should be disabled.

Section 18 RAM

This LSI has an on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling one-state access by the CPU to both byte data and word data.

The on-chip RAM can be enabled or disabled by means of the RAME bit in the system control register (SYSCR). For details on the system control register (SYSCR), refer to section 3.2.2, System Control Register (SYSCR).

Product Type	Name	ROM Type	RAM Capacitance	RAM Address
H8S/2678 Series	HD64F2676	Flash memory version	8 kbytes	H′FFA000 to H′FFBFFF
	HD6432676	Masked ROM version	8 kbytes	H′FFA000 to H′FFBFFF
	HD6432675		8 kbytes	H′FFA000 to H′FFBFFF
	HD6432673		8 kbytes	H′FFA000 to H′FFBFFF
	HD6412670	ROMless version	8 kbytes	H′FFA000 to H′FFBFFF
H8S/2678R Series	HD6412674R	ROMless version	32 kbytes	H′FF4000 to H′FFBFFF

Section 19 Flash Memory (F-ZTAT Version)

The features of the flash memory included in the flash memory version are summarized below. The block diagram of the flash memory is shown in figure 19.1.

19.1 Features

- Size

Product Classification		ROM Size	ROM Address
H8S/2678 Series	HD64F2676	256 kbytes	H'000000 to H'03FFFF (Modes 3, 4, 7, 10, and 11) H'100000 to H'13FFFF (Modes 5, 6, 13, and 14)

- Programming/erase methods

  The flash memory is programmed 128 bytes at a time. Erase is performed in single-block units. The flash memory of 384 kbytes is configured as follows: 64 kbytes × 5 blocks, 32 kbytes × 1 block, and 4 kbytes × 8 block. The 256-kbyte flash memory is configured as follows: 64 kbytes × 3 blocks, 32 kbytes × 1 block, and 4 kbytes × 8 blocks. To erase the entire flash memory, each block must be erased in turn.
- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.
- Two on-board programming modes

  Boot mode

  User program mode

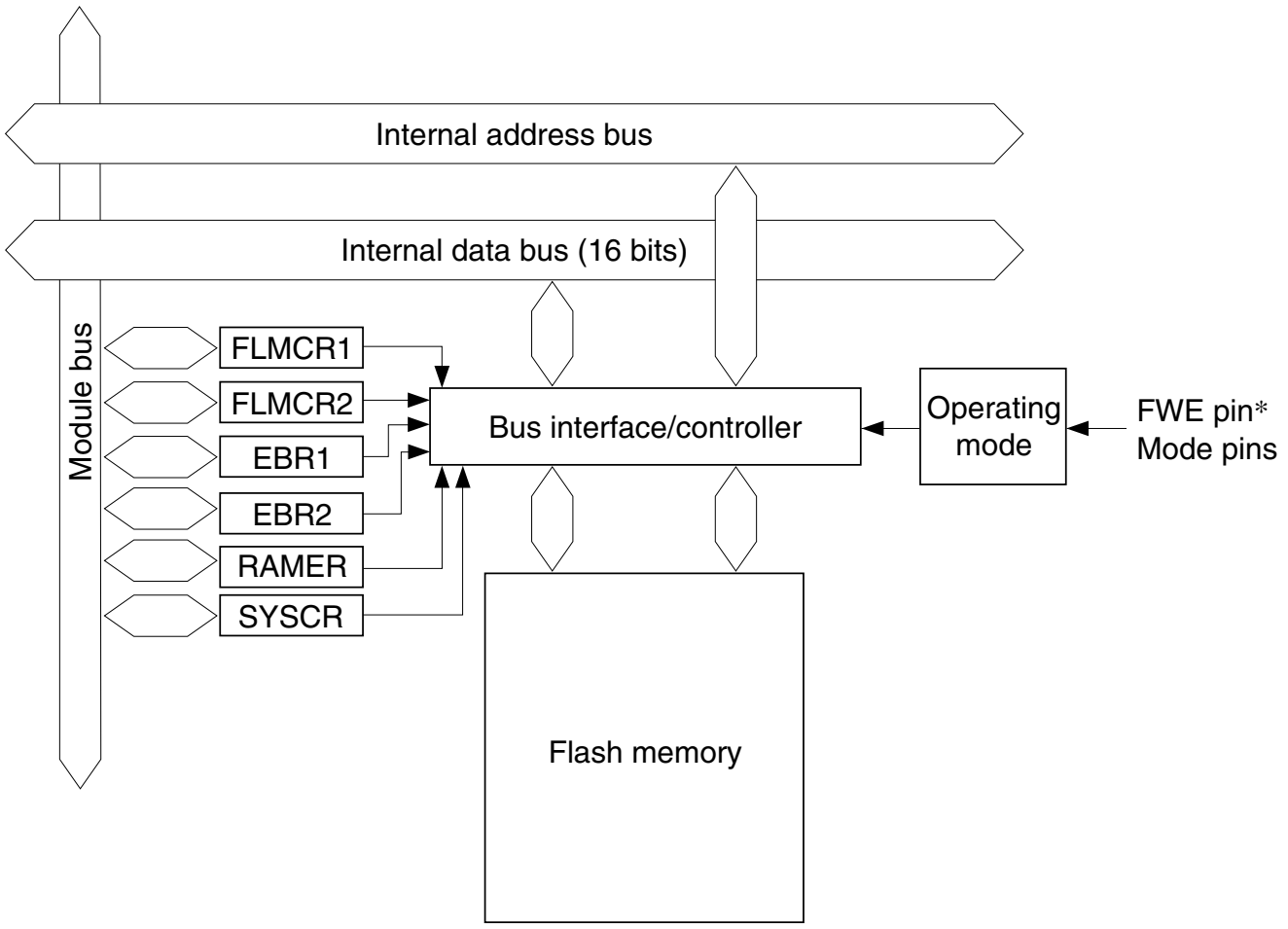
  On-board programming/erasing can be done in boot mode in which the on-chip boot program is started for erase or programming of the entire flash memory. In normal user program mode, individual blocks can be erased or programmed.
- Programmer mode

  Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.
- Automatic bit rate adjustment

  With data transfer in boot mode, the bit rate of this LSI can be automatically adjusted to match the transfer bit rate of the host.
- Flash memory emulation by RAM

  Part of the RAM area can be overlapped onto flash memory, to emulate flash memory updates in real time.
- Programming/erasing protection

  There are three protect modes, hardware, software, and error protect, which allow protected status to be designated for flash memory program/erase operations.



Legend
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1: Erase block register 1
EBR2: Erase block register 2
RAMER: RAM emulation register
SYSCR: System control register

Note: * Only in H8S/2678 series.

Figure 19.1 Block Diagram of Flash Memory

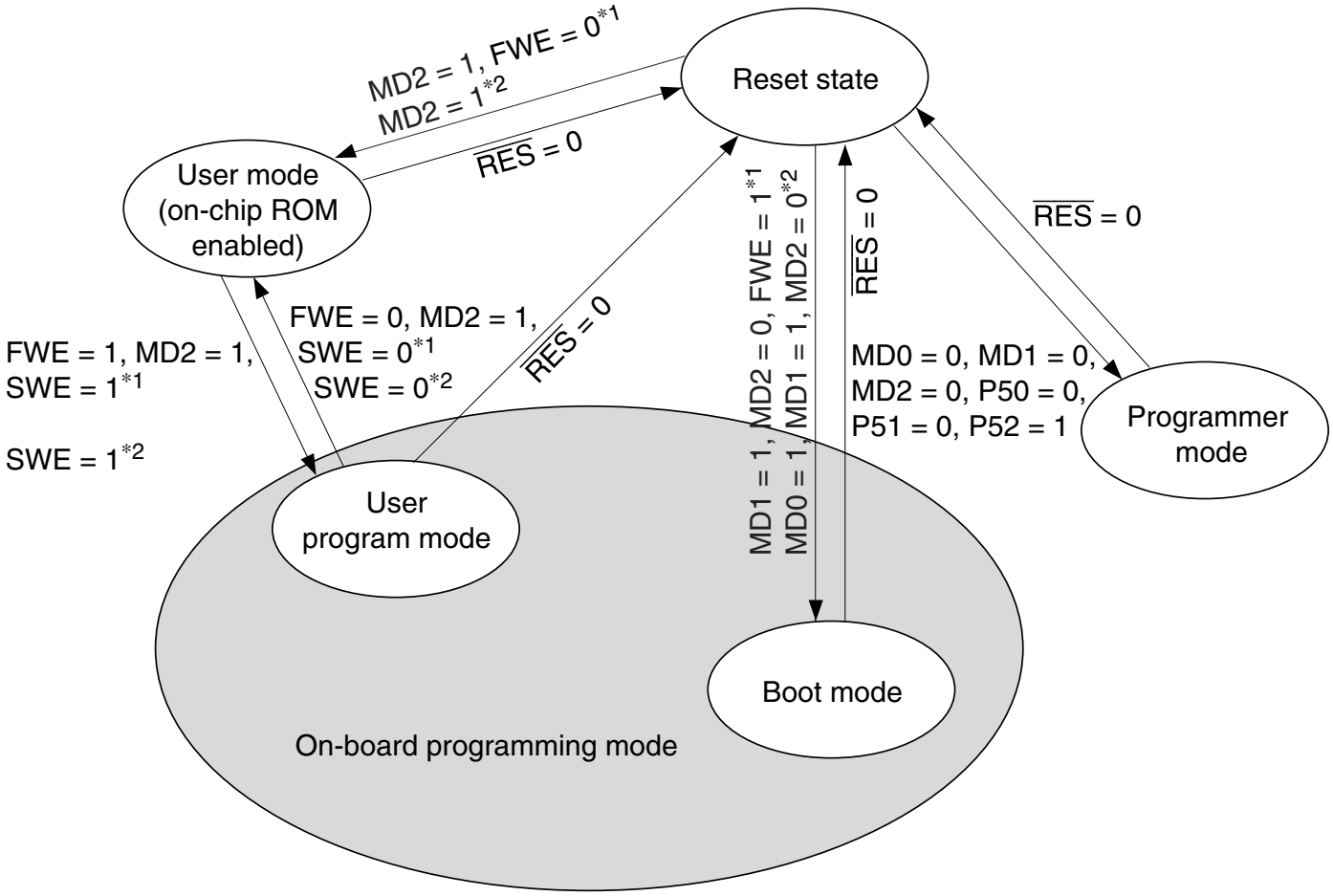
19.2 Mode Transitions

When the mode pins and the FWE pin* are set in the reset state and a reset-start is executed, this LSI enters an operating mode as shown in figure 19.2. In user mode, flash memory can be read but not programmed or erased.

The boot, user program and programmer modes are provided as modes to write and erase the flash memory.

The differences between boot mode and user program mode are shown in table 19.1. Figure 19.3 shows boot mode. Figure 19.4 shows user program mode.

Note: Only in the H8S/2678 Series.



Notes: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

*1 Only in H8S/2678 series.
*2 Only in H8S/2678R series.

Figure 19.2 Flash Memory State Transitions

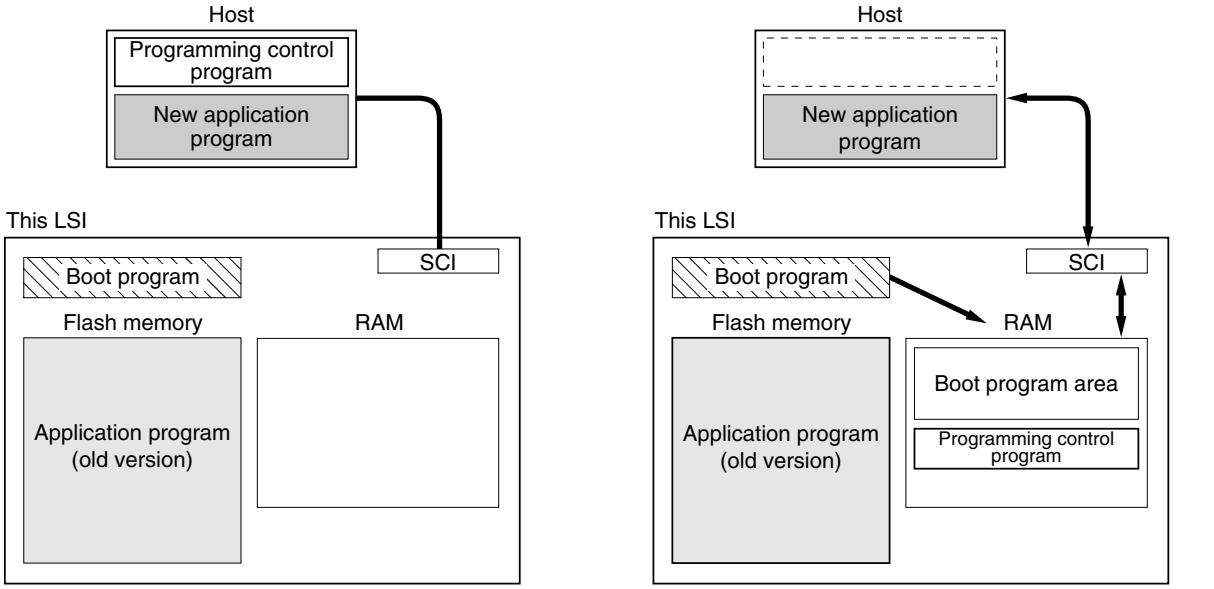
Table 19.1 Differences between Boot Mode and User Program Mode

	Boot Mode	User Program Mode
Total erase	Yes	Yes
Block erase	No	Yes
Programming control program*	Program/program-verify	Erase/erase-verify/program/ program-verify emulation

Note: To be provided by the user, in accordance with the recommended algorithm.

1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.

2. Programming control program transfer
   When boot mode is entered, the boot program in the chip (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.



3. Flash memory initialization
   The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, entire flash memory erasure is performed, without regard to blocks.

4. Writing new application program
   The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.

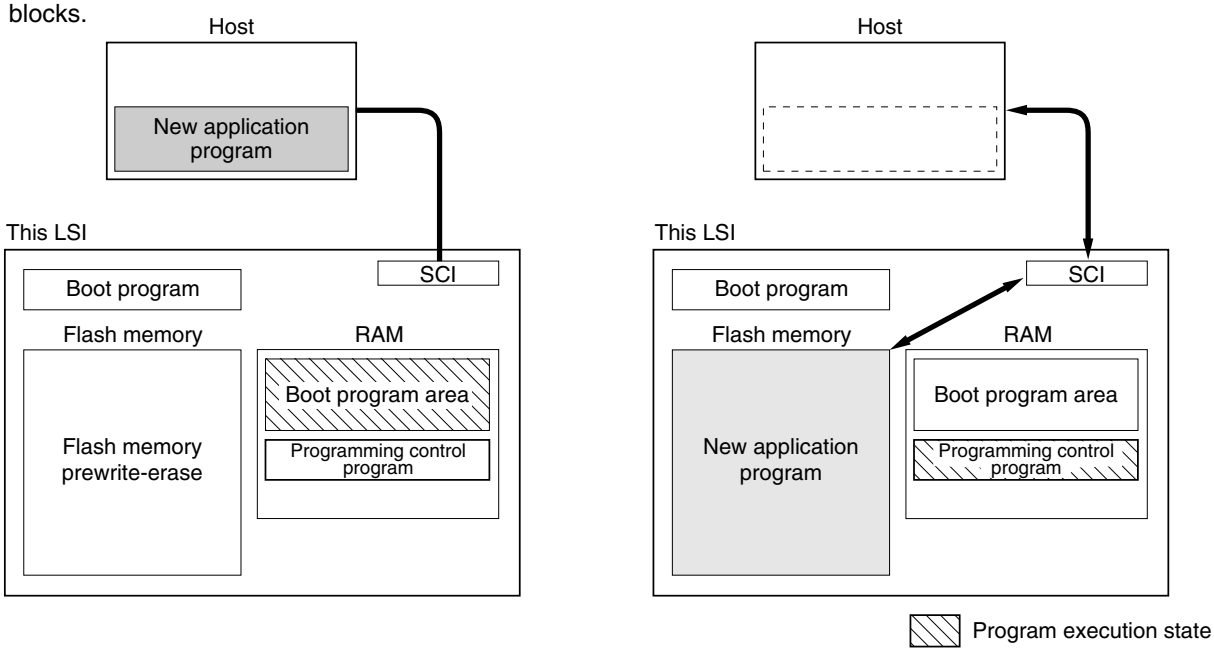
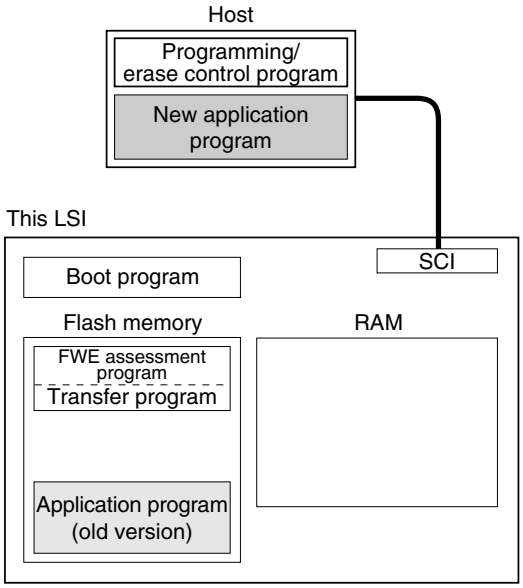
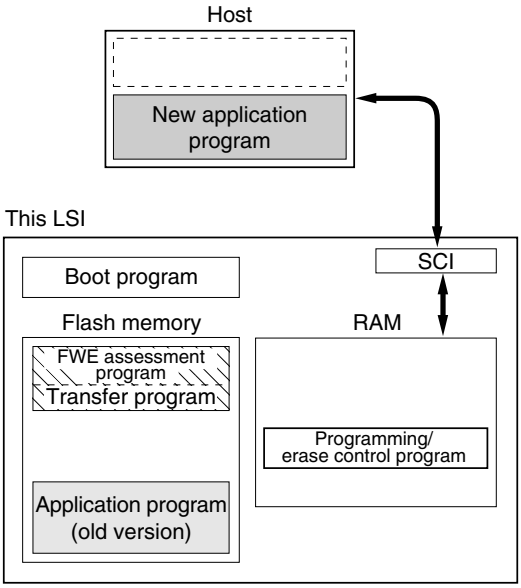


Figure 19.3 Boot Mode

1. Initial state
   (1) The FWE assessment program that confirms that user program mode is entered, and (2) the program that will transfer the programming/ erase control program to on-chip RAM should be written into the flash memory by the user beforehand. (3) The programming/erase control program should be prepared in the host or in the flash memory.

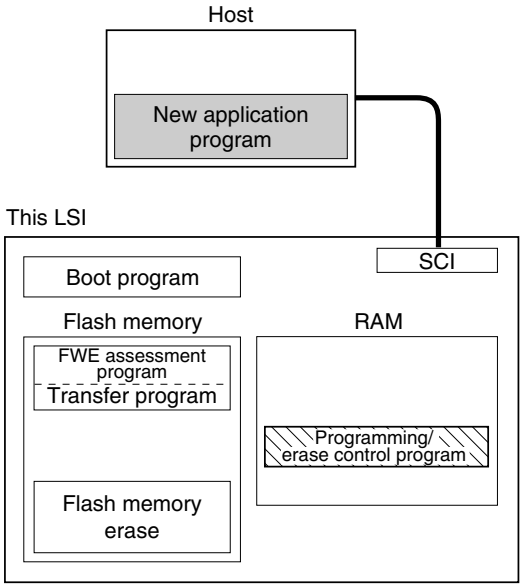
2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes the transfer program in the flash memory, and transfers the programming/erase control program to RAM.

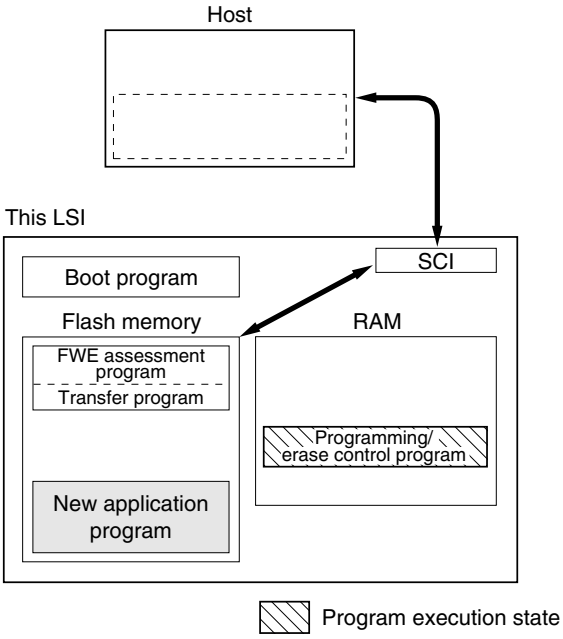




3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.

4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.



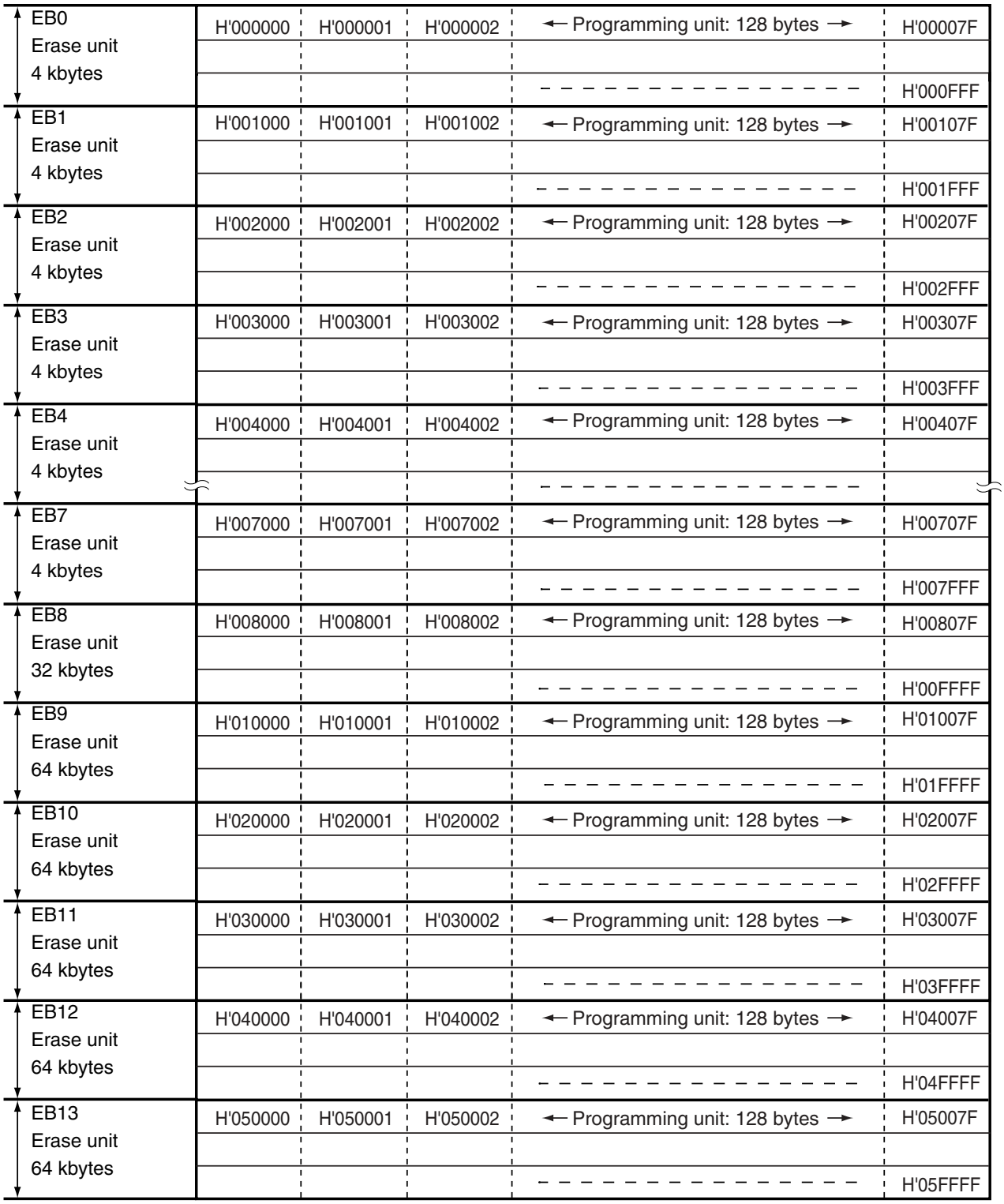


Note: The FWE assessment program is not available in the H8S/2678R series.

Figure 19.4 User Program Mode

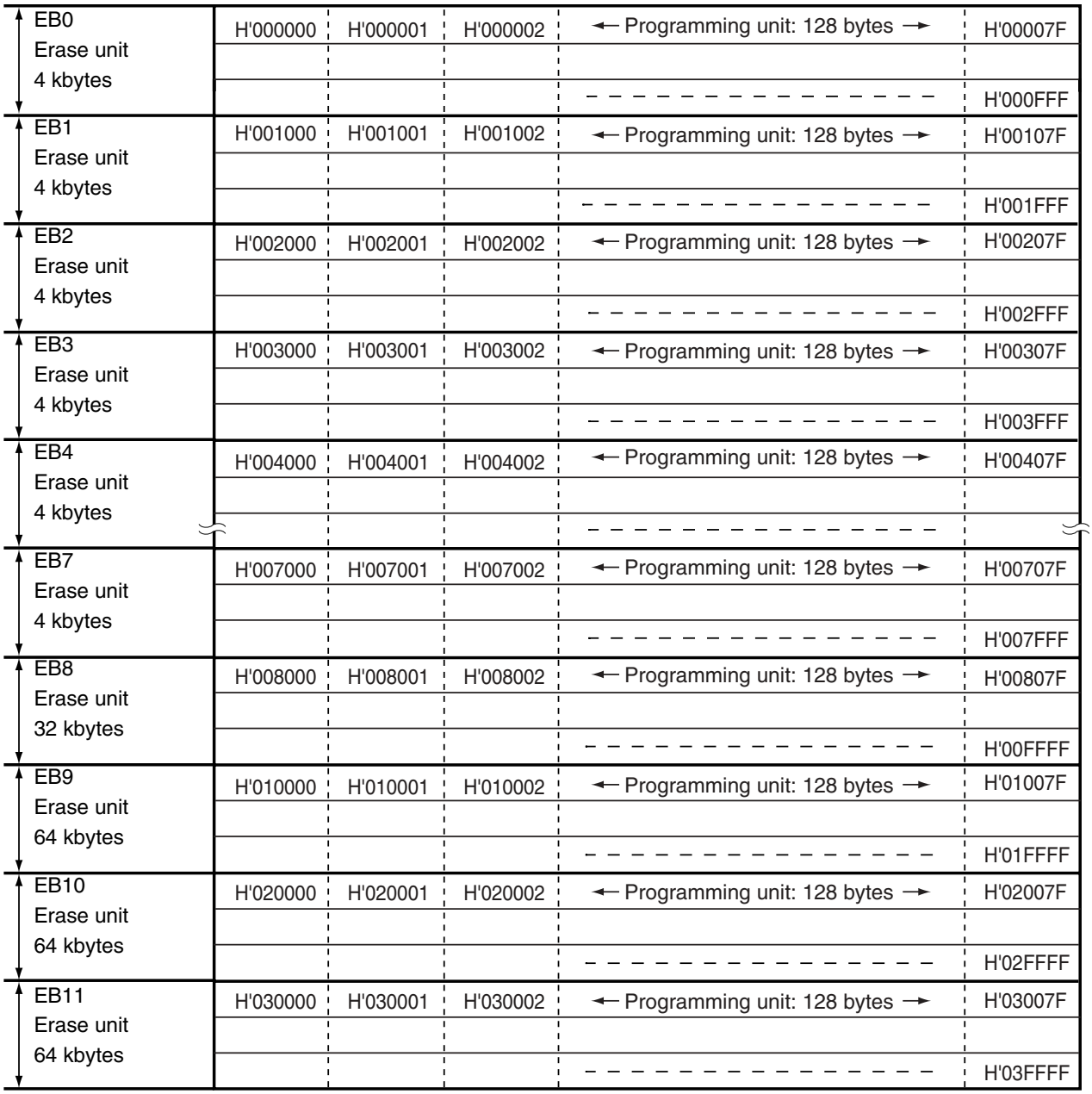
19.3 Block Configuration

Figure 19.5 shows the block configuration of 384-kbyte flash memory and figure 19.6 shows that of 256-kbyte flash memory. The thick lines indicate erasing units, the narrow lines indicate programming units, and the values are addresses. The 384-kbyte flash memory is divided into 64 kbytes (5 blocks), 32 kbytes (1 block), and 4 kbytes (8 blocks). The 256-kbyte flash memory is divided into 64 kbytes (3 blocks), 32 kbytes (1 block), and 4 kbytes (8 blocks). Erasing is performed in these divided units. Programming is performed in 128-byte units starting from an address whose lower eight bits are H'00 or H'80.



Note: Addresses H'100000 to H'15FFFF are allocated in modes 5 and 6.

Figure 19.5 384-Kbyte Flash Memory Block Configuration (Modes 3, 4, and 7)



Note: Addresses H'100000 to H'13FFFF are allocated in modes 5, 6, 13, and 14.

Figure 19.6 256-Kbyte Flash Memory Block Configuration (Modes 4, 7, 10, and 11)

19.4 Input/Output Pins

Table 19.2 shows the pin configuration of the flash memory.

Table 19.2 Pin Configuration

Pin Name	I/O	Function
RES	Input	Reset
FWE*	Input	Flash program/erase protection by hardware
MD2	Input	Sets this LSI's operating mode
MD1	Input	Sets this LSI's operating mode
MD0	Input	Sets this LSI's operating mode
P52	Input	Sets operating mode in programmer mode
P51	Input	Sets operating mode in programmer mode
P50	Input	Sets operating mode in programmer mode
TxD1	Output	Serial transmit data output
RxD1	Input	Serial receive data input

Note: Only in H8S/2678 Series.

19.5 Register Descriptions

The flash memory has the following registers. For details on the system control register, refer to section 3.2.2, System Control Register (SYSCR).

- Flash memory control register 1 (FLMCR1)
- Flash memory control register 2 (FLMCR2)
- Erase block register 1 (EBR1)
- Erase block register 2 (EBR2)
- RAM emulation register (RAMER)

19.5.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory transit to program mode, program-verify mode, erase mode, or erase-verify mode. For details on register setting, refer to section 19.8, Flash Memory Programming/Erasing.

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	0/1	R	Flash Write Enable Reflects the input level at the FWE pin. It is set to 1 when a high level is input to the FWE pin, and cleared to 0 when a low level is input. When this bit is cleared to 0, the flash memory transits to the hardware protection state. Note: In the H8S/2678R Series, this bit is reserved. This bit is always read as 0 in modes 1 and 2. This bit is always read as 1 in modes 3 to 7. The initial value should not be changed.
6	SWE	0	R/W	Software Write Enable When this bit is set to 1, flash memory programming/erasing is enabled. When this bit is cleared to 0, other FLMCR1 register bits and all EBR1 and EBR2 bits cannot be set.
5	ESU	0	R/W	Erase Setup When this bit is set to 1 while FWE = 1* and SWE = 1, the flash memory transits to the erase setup state. When it is cleared to 0, the erase setup state is cancelled.
4	PSU	0	R/W	Program Setup When this bit is set to 1 while FWE = 1* and SWE = 1, the flash memory transits to the program setup state. When it is cleared to 0, the program setup state is cancelled.
3	EV	0	R/W	Erase-Verify When this bit is set to 1 while FWE = 1* and SWE = 1, the flash memory transits to erase-verify mode. When it is cleared to 0, erase-verify mode is cancelled.
2	PV	0	R/W	Program-Verify When this bit is set to 1 while FWE = 1* and SWE = 1, the flash memory transits to program-verify mode. When it is cleared to 0, program-verify mode is cancelled.
1	E	0	R/W	Erase When this bit is set to 1 while FWE = 1*, SWE = 1, and ESU = 1, the flash memory transits to erase mode. When it is cleared to 0, erase mode is cancelled.

Bit	Bit Name	Initial Value	R/W	Description
0	P	0	R/W	Program When this bit is set to 1 while FWE = 1*, SWE = 1, and PSU = 1, the flash memory transits to program mode. When it is cleared to 0, program mode is cancelled.

Note: Only in H8S/2678 Series.

19.5.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is a register that displays the state of flash memory programming/erasing. FLMCR2 is a read-only register, and should not be written to. When the on-chip flash memory is disabled, the contents of FLMCR2 are always read as H'00.

Bit	Bit Name	Initial Value	R/W	Description
7	FLER	0	R	Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state. See 19.9.3 Error Protection, for details.
6 to 0	—	All 0	R	Reserved These bits always read as 0.

19.5.3 Erase Block Register 1 (EBR1)

EBR1 specifies the flash memory erase area block. EBR1 is initialized to H'00 when the SWE bit in FLMCR1 is 0. Set only one bit in EBR1 and EBR2 together (do not set more than one bit at the same time). Setting more than one bit will automatically clear all EBR1 and EBR2 bits to 0. For details, see table 19.3, Erase Blocks.

Bit	Bit Name	Initial Value	R/W	Description
7	EB7	0	R/W	When this bit is set to 1, 4 kbytes of EB7 are to be erased.
6	EB6	0	R/W	When this bit is set to 1, 4 kbytes of EB6 are to be erased.
5	EB5	0	R/W	When this bit is set to 1, 4 kbytes of EB5 are to be erased.
4	EB4	0	R/W	When this bit is set to 1, 4 kbytes of EB4 are to be erased.
3	EB3	0	R/W	When this bit is set to 1, 4 kbytes of EB3 is to be erased.
2	EB2	0	R/W	When this bit is set to 1, 4 kbytes of EB2 is to be erased.
1	EB1	0	R/W	When this bit is set to 1, 4 kbytes of EB1 is to be erased.
0	EB0	0	R/W	When this bit is set to 1, 4 kbytes of EB0 is to be erased.

19.5.4 Erase Block Register 2 (EBR2)

EBR2 specifies the flash memory erase area block. EBR2 is initialized to H'00 when the SWE bit in FLMCR1 is 0. Set only one bit in EBR2 and EBR1 together (do not set more than one bit at the same time). Setting more than one bit will automatically clear all EBR1 and EBR2 bits to 0. For details, see table 19.3, Erase Blocks.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	0	R/W	Reserved The initial value should not be changed.
5	EB13	0	R/W	When this bit is set to 1, 64 kbytes of EB13 are to be erased. Note: In the H8S/2678 Series, this bit is reserved. The initial value should not be changed.
4	EB12	0	R/W	When this bit is set to 1, 64 kbytes of EB12 are to be erased. Note: In the H8S/2678 Series, this bit is reserved. The initial value should not be changed.
3	EB11	0	R/W	When this bit is set to 1, 64 kbytes of EB11 are to be erased.
2	EB10	0	R/W	When this bit is set to 1, 64 kbytes of EB10 are to be erased.
1	EB9	0	R/W	When this bit is set to 1, 64 kbytes of EB9 are to be erased.
0	EB8	0	R/W	When this bit is set to 1, 32 kbytes of EB8 are to be erased.

Table 19.3 Erase Blocks

Block (Size)	Address	
	H8S/2678R Series: Modes 3, 4, and 7 H8S/2678 Series: Modes 4, 7, 10, and 11	H8S/2678R Series: Modes 5 and 6 H8S/2678 Series: Modes 5, 6, 13, and 14
EB0 (4 kbytes)	H'000000 to H'000FFF	H'100000 to H'100FFF
EB1 (4 kbytes)	H'001000 to H'001FFF	H'101000 to H'101FFF
EB2 (4 kbytes)	H'002000 to H'002FFF	H'102000 to H'102FFF
EB3 (4 kbytes)	H'003000 to H'003FFF	H'103000 to H'103FFF
EB4 (4 kbytes)	H'004000 to H'004FFF	H'104000 to H'104FFF
EB5 (4 kbytes)	H'005000 to H'005FFF	H'105000 to H'105FFF
EB6 (4 kbytes)	H'006000 to H'006FFF	H'106000 to H'106FFF
EB7 (4 kbytes)	H'007000 to H'007FFF	H'107000 to H'107FFF
EB8 (32 kbytes)	H'008000 to H'00FFFF	H'108000 to H'10FFFF
EB9 (64 kbytes)	H'010000 to H'01FFFF	H'110000 to H'11FFFF
EB10 (64 kbytes)	H'020000 to H'02FFFF	H'120000 to H'12FFFF
EB11 (64 kbytes)	H'030000 to H'03FFFF	H'130000 to H'13FFFF
EB12 (64 kbytes)	H'040000 to H'04FFFF	H'140000 to H'14FFFF
EB13 (64 kbytes)	H'050000 to H'05FFFF	H'150000 to H'15FFFF

Note: The erase blocks of the 384-kbyte flash memory are EB0 to EB13.
The erase blocks of the 256-kbyte flash memory are EB0 to EB11.

19.5.5 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER settings should be made in user mode or user program mode. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
7 and 5	—	All 0	R	Reserved These bits always read as 0.
4	—	0	R/W	Reserved The initial value should not be changed.
3	RAMS	0	R/W	RAM Select Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, the flash memory is overlapped with part of RAM, and all flash memory block are in the program/erase-protect state. When this bit is cleared to 0, the RAM emulation function is invalid.
2 1 0	RAM2 RAM1 RAM0	0 0 0	R/W R/W R/W	Flash Memory Area Selection When the RAMS bit is set to 1, selects one of the following flash memory areas to overlap the RAM area. The areas correspond with 4-kbyte erase blocks. H8S/2678R Series: Modes 3, 4, and 7 H8S/2678 Series: Modes 4, 7, 10, and 11 000: H'000000 to H'000FFF (EB0) 001: H'001000 to H'001FFF (EB1) 010: H'002000 to H'002FFF (EB2) 011: H'003000 to H'003FFF (EB3) 100: H'004000 to H'004FFF (EB4) 101: H'005000 to H'005FFF (EB5) 110: H'006000 to H'006FFF (EB6) 111: H'007000 to H'007FFF (EB7) H8S/2678R Series: Modes 5 and 6 H8S/2678 Series: Modes 5, 6, 13, and 14 000: H'100000 to H'100FFF (EB0) 001: H'101000 to H'101FFF (EB1) 010: H'102000 to H'102FFF (EB2) 011: H'103000 to H'103FFF (EB3) 100: H'104000 to H'104FFF (EB4) 101: H'105000 to H'105FFF (EB5) 110: H'106000 to H'106FFF (EB6) 111: H'107000 to H'107FFF (EB7)

19.6 On-Board Programming Modes

In an on-board programming mode, programming, erasing, and verification for the on-chip flash memory can be performed. There are two on-board programming modes: boot mode and user program mode. Table 19.4 shows how to select boot mode. User program mode can be selected by setting the control bits by software. For a diagram that shows mode transitions of flash memory, see figure 19.2.

Table 19.4 Setting On-Board Programming Modes

- H8S/2678 Series

Mode Setting		FWE	MD2	MD1	MD0
Boot mode	Expanded mode with on-chip ROM enabled	1	0	1	0
	Single-chip activation expanded mode with on-chip ROM enabled	1	0	1	1
User program mode	Expanded mode with on-chip ROM enabled	1	1	0	0
	External ROM activation expanded mode with on-chip ROM enabled*[1]	1	1	0	1
	External ROM activation expanded mode with on-chip ROM enabled*[2]	1	1	1	0
	Single-chip activation expanded mode with on-chip ROM enabled	1	1	1	1

Notes: 1. The initial setting for the external bus width is 16 bits.
2. The initial setting for the external bus width is 8 bits

- H8S/2678R Series

Mode Setting		MD2	MD1	MD0
Boot mode	Single-chip activation expanded mode with on-chip ROM enabled	0	1	1

19.6.1 Boot Mode

When this LSI enters boot mode, the embedded boot program is started. The boot program transfers the programming control program from the externally connected host to the on-chip RAM via the SCI_1. When the flash memory is all erased, the programming control program is executed.

Table 19.5 shows the boot mode operations between reset end and branching to the programming control program.

1. When the boot program is initiated, the SCI_1 should be set to asynchronous mode, the chip measures the low-level period of asynchronous SCI communication data (H'00) transmitted continuously from the host. The chip then calculates the bit rate of transmission from the host, and adjusts the SCI_1 bit rate to match that of the host. The transfer format is 8-bit data, 1 stop bit, and no parity. The reset should end with the RxD pin high. The RxD and TxD pins should be pulled up on the board if necessary. After the reset ends, it takes approximately 100 states before the chip is ready to measure the low-level period.
2. After matching the bit rates, the chip transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the chip. If reception could not be performed normally, initiate boot mode again by a reset. Depending on the host's transfer bit rate and system clock frequency of this LSI, there will be a discrepancy between the bit rates of the host and the chip. To operate the SCI properly, set the host's transfer bit rate and system clock frequency of this LSI within the ranges listed in table 19.6.
3. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 19.8, Flash Memory Programming/Erasing.
4. Before branching to the programming control program, the chip terminates transfer operations by the SCI_1 (by clearing the RE and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. Therefore, the programming control program can still use it for transfer of program data or verify data with the host. The TxD pin is high. The contents of the CPU general registers are undefined immediately after branching to the programming control program. These registers must be initialized at the beginning of the programming control program, since the stack pointer (SP), in particular, is used implicitly in subroutine calls, etc.
5. In boot mode, if flash memory contains data (all data is not 1), all blocks of flash memory are erased. Boot mode is used for the initial programming in the on-board state or for a forcible return when a program that is to be initiated in user program mode was accidentally erased and could not be executed in user program mode.

Notes: 1. In boot mode, a part of the on-chip RAM area (H'FF8000 to H'FF87FF) is used by the boot program. Addresses H'FF8800 to H'FFBFFF is the area to which the programming control program is transferred from the host. The boot program area cannot be used until the execution state in boot mode switches to the programming control program.

2. Boot mode can be cleared by a reset. Release the reset by setting the MD pins, after waiting at least 20 states since driving the reset pin low. Boot mode is also cleared when the WDT overflow reset occurs.
3. Do not change the MD pin input levels in boot mode.
4. All interrupts are disabled during programming or erasing of the flash memory.

Table 19.5 Boot Mode Operation

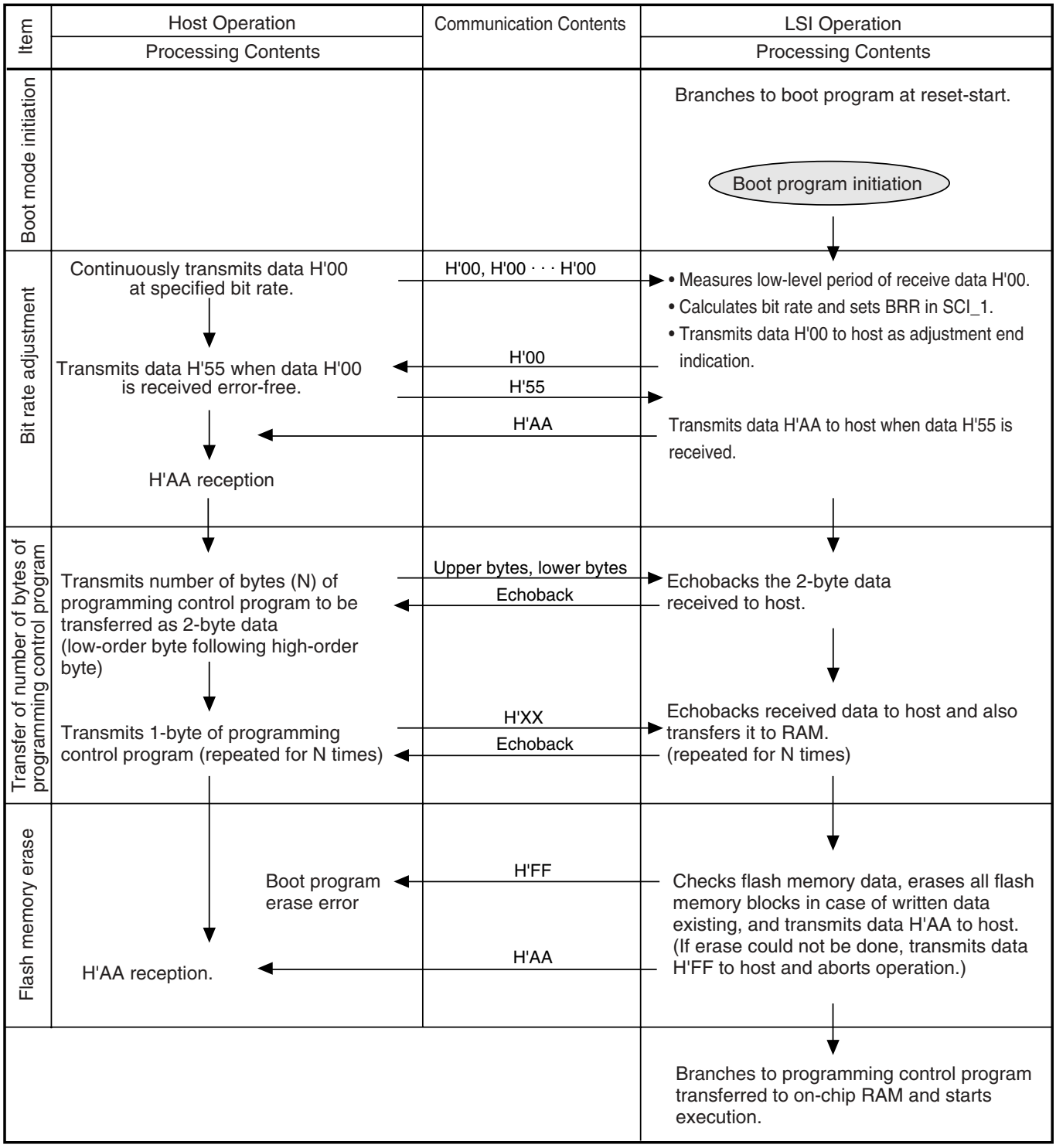


Table 19.6 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible

Host Bit Rate	System Clock Frequency Range of LSI
19,200 bps	8 to 25 MHz
9,600 bps	8 to 25 MHz

19.6.2 User Program Mode

On-board programming/erasing of an individual flash memory block can also be performed in user program mode by branching to a user program/erase program. The user must set branching conditions and provide on-board means of supplying programming data. The flash memory must contain the program/erase program or a program which provides the program/erase program from external memory. Because the flash memory itself cannot be read during programming/erasing, transfer the program/erase program to on-chip RAM, as like in boot mode. Figure 19.7 shows a sample procedure for programming/erasing in user program mode. Prepare a program/erase program in accordance with the description in section 19.8, Flash Memory Programming/Erasing.

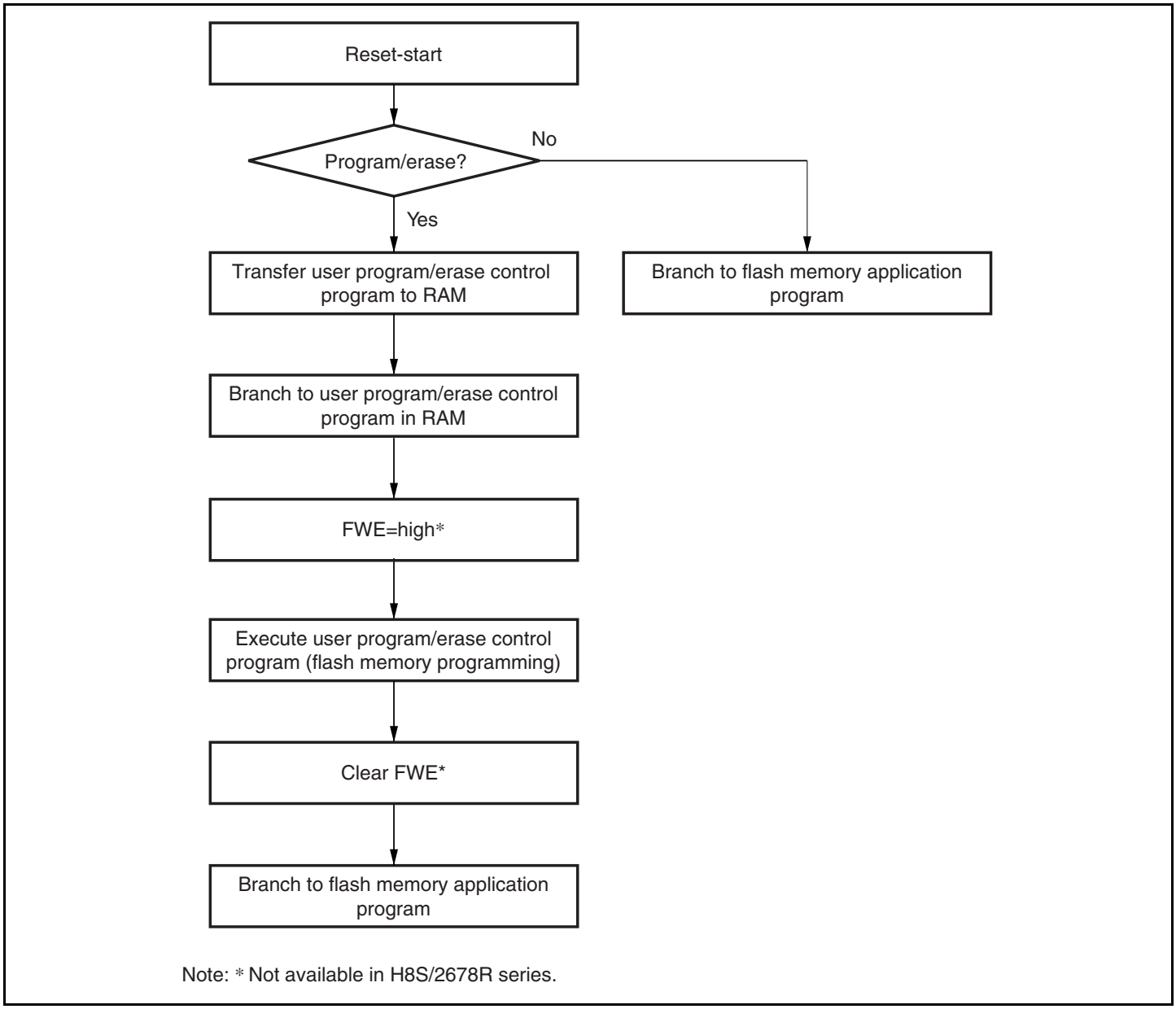


Figure 19.7 Programming/Erasing Flowchart Example in User Program Mode

19.7 Flash Memory Emulation in RAM

Making a setting in the RAM emulation register (RAMER) enables RAM to be overlapped onto the part of flash memory area so that data to be programmed to flash memory can be emulated in the on-chip RAM in real time. Emulation can be performed in user mode or user program mode. Figure 19.8 shows an example of emulation of real-time flash memory programming.

1. Set RAMER to overlap RAM onto the area for which real-time programming is required.
2. Emulation is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, thus releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB0).

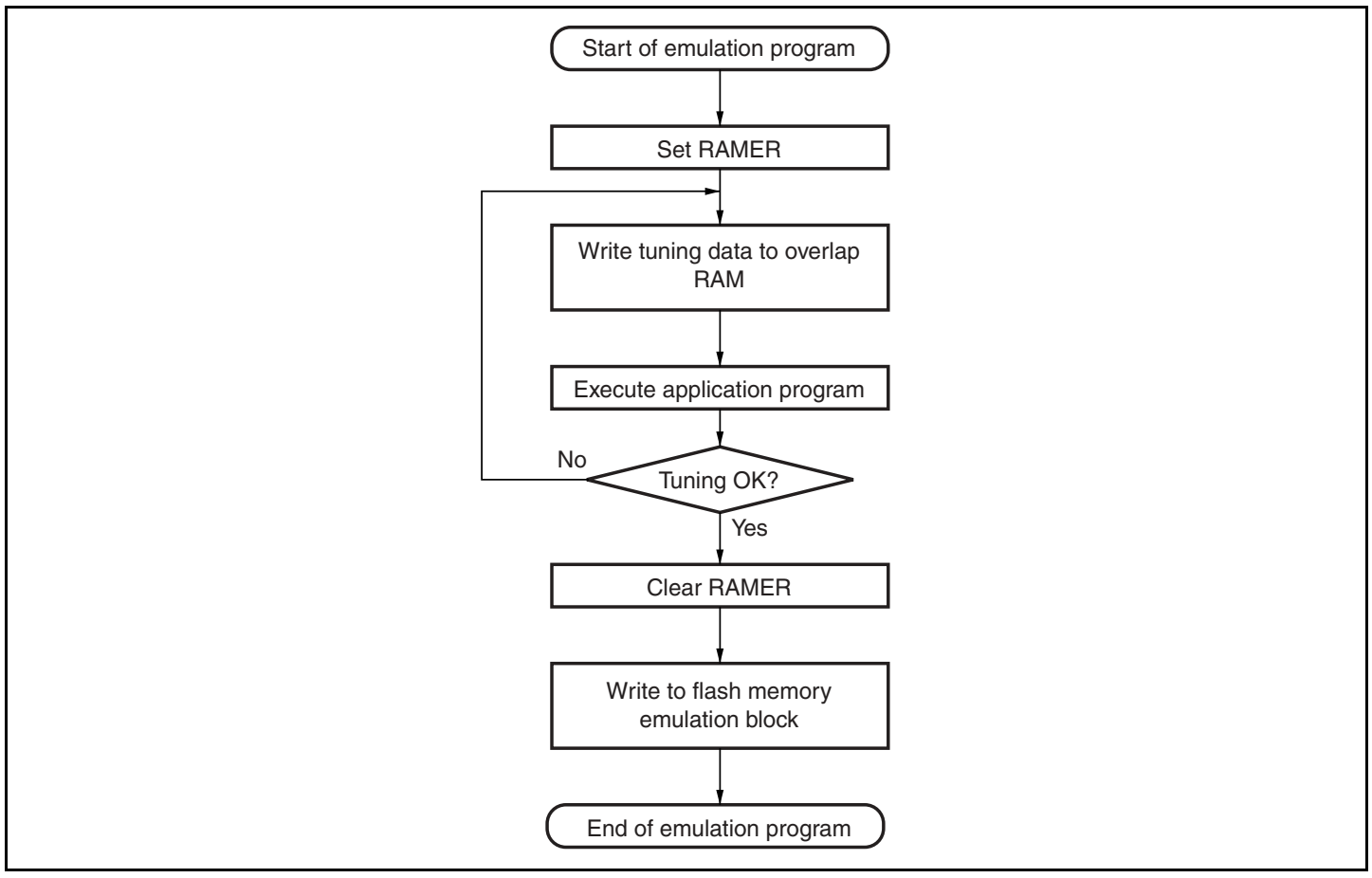


Figure 19.8 Flowchart for Flash Memory Emulation in RAM

Example in which flash memory block is overlapped is shown in figure 19.9.

1. The RAM area to be overlapped is fixed at a 4-kbyte area in the range of H'FFA000 to H'FFAFFF.
2. The flash memory area to overlap is selected by RAMER from a 4-kbyte area among one of the EB0 to EB7 blocks.

3. The overlapped RAM area can be accessed from both the flash memory addresses and RAM addresses.

Notes: 1. When the RAMS bit in RAMER is set to 1, program/erase protection is enabled for all flash memory blocks (emulation protection). In this state, setting the P or E bit in FLMCR1 to 1 does not cause a transition to program mode or erase mode.

2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm.

3. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlap RAM.

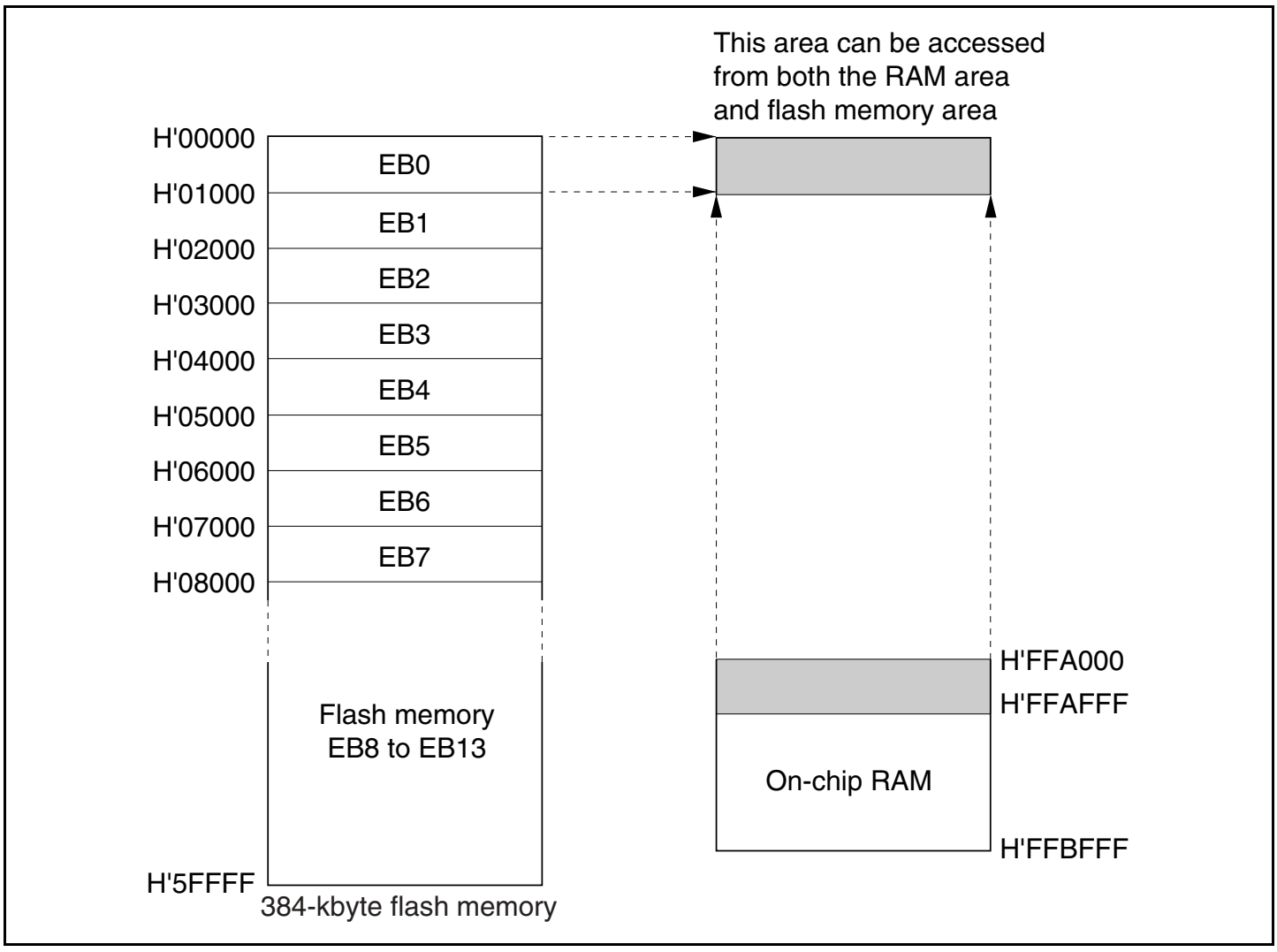


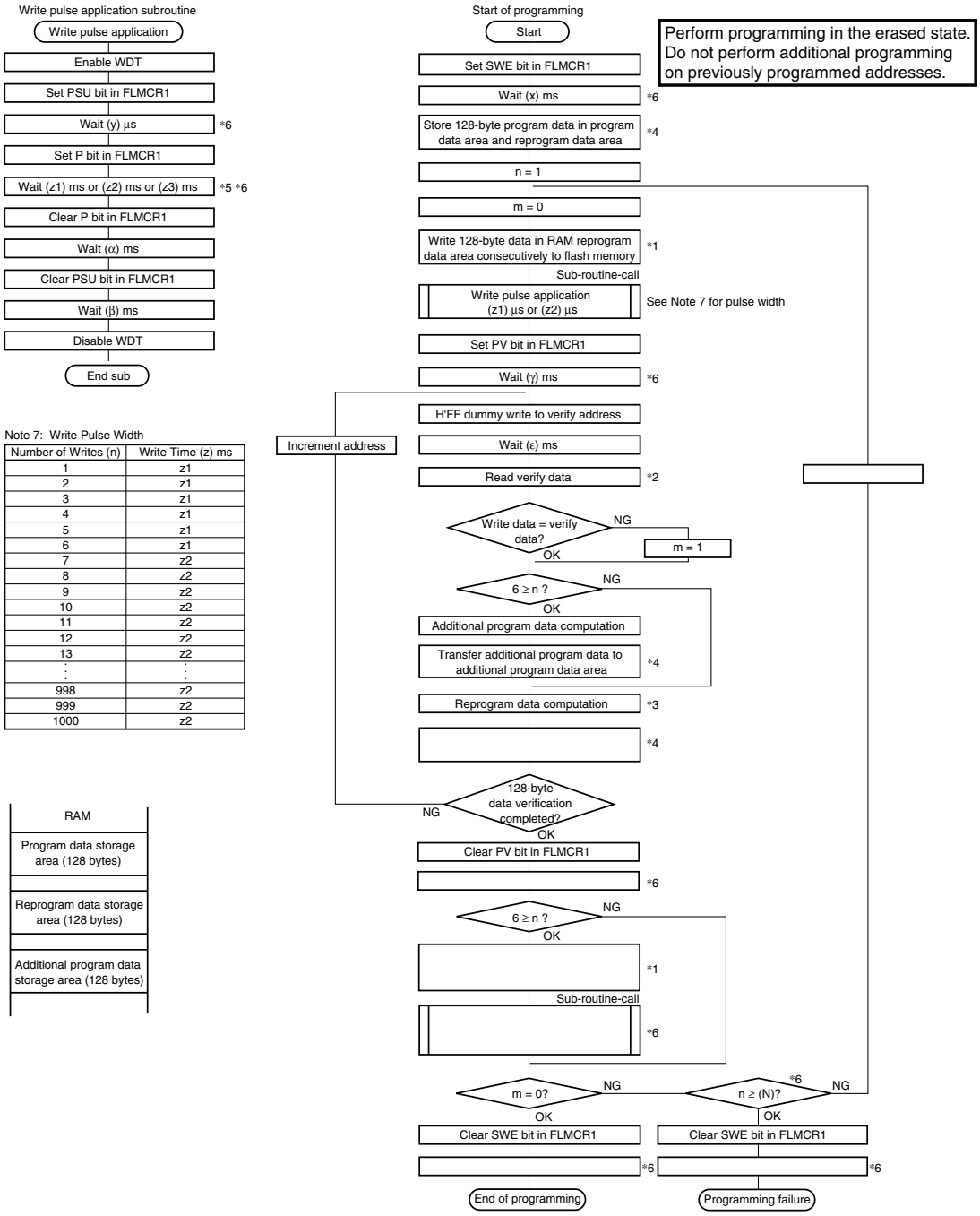
Figure 19.9 Example of RAM Overlap Operation

19.8 Flash Memory Programming/Erasing

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. Depending on the FLMCR1 and FLMCR2 setting, the flash memory operates in one of the following four modes: program mode, erase mode, program-verify mode, and erase-verify mode. The programming control program in boot mode and the user program/erase program in user mode use these operating modes in combination to perform programming/erasing. Flash memory programming and erasing should be performed in accordance with the descriptions in section 19.8.1, Program/Program-Verify and section 19.8.2, Erase/Erase-Verify, respectively.

19.8.1 Program/Program-Verify

When programming data or programs to the flash memory, the program/program-verify flowchart shown in figure 19.10 should be followed. Performing programming operations according to this flowchart will enable data or programs to be programmed to the flash memory without subjecting the chip to voltage stress or sacrificing program data reliability.



Note 7: Write Pulse Width

Number of Writes (n)	Write Time (z) ms
1	z1
2	z1
3	z1
4	z1
5	z1
6	z1
7	z2
8	z2
9	z2
10	z2
11	z2
12	z2
13	z2
:	:
998	z2
999	z2
1000	z2

Notes:
1. Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00 or H'80. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.
2. Verify data is read in 16-bit (W) units.
3. The reprogram data is given by the operation of the following tables (comparison between stored data in the program data area and verify data). Programming is executed for the bits of reprogram data 0 in the next reprogram loop. Even bits for which programming has been completed will be subjected to additional programming if they fail the subsequent verify operation.
4. A 128-byte areas for storing program data, reprogram data, and additional program data must be provided in the RAM. The contents of the reprogram and additional program data are modified as programming proceeds.
5. A write pulse of (z1) or (z2) µs should be applied according to the progress of the programming operation. See Note 7 for the pulse widths. When writing of additional-programming data is executed, a (z3) µs write pulse should be applied. Reprogram data X' means reprogram data when the write pulse is applied.
6. For the values of x, y, z1, z2, z3, a, b, g, e, h, q, and N, see section 24.6, Flash Memory Characteristics.

z807 ramming data is executed, a (6.p3ted, a (6rnmq0.9uhe reprol m197.492 588.56 l96.836 582.933525,

Additional Program Data Operation Chart

Reprogram Data (X')	Verify Data (V)	Additional Program Data (Y)	Comments
0	0	0	Additional programming executed
	1	1	Additional programming not executed
1	0		Additional programming not executed
	1		Additional programming not executed

19.8.2 Erase/Erase-Verify

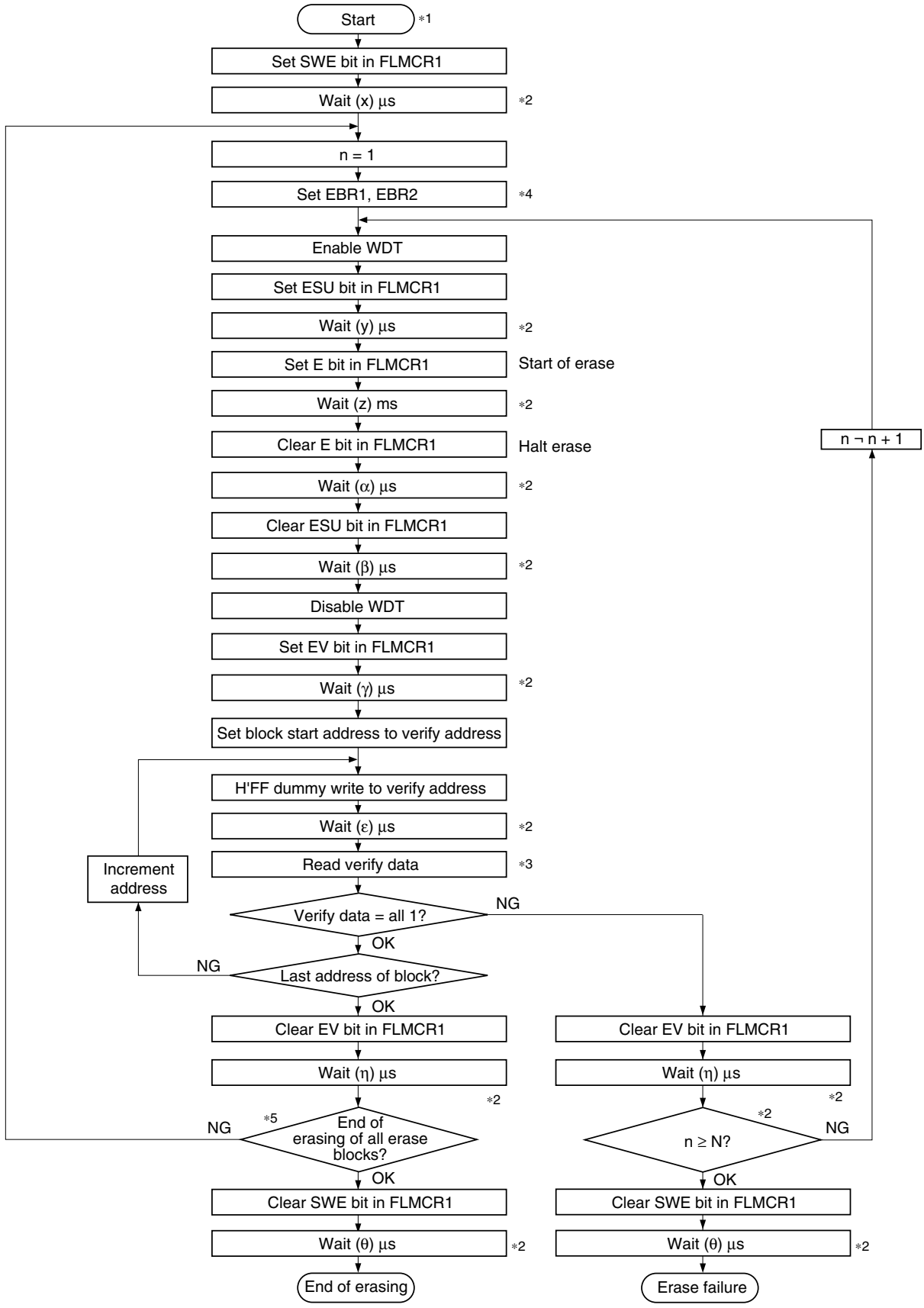
When erasing flash memory, the erase/erase-verify flowchart shown in figure 19.11 should be followed.

1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Erasing is performed in block units. Make only a single-bit specification in the erase block registers (EBR1 and EBR2). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E bit is set to 1 is the flash memory erase time.
4. The watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. Set a value greater than (y + z + α + β) ms as the WDT overflow period.
5. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower two bits are B'00. Verify data can be read in longwords from the address to which a dummy write was performed.
6. If the read data is unerased, set erase mode again, and repeat the erase/erase-verify sequence as before. The maximum number of repetitions of the erase/erase-verify sequence (N) must not be exceeded.

19.8.3 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI input, are disabled when flash memory is being programmed or erased, and while the boot program is executing in boot mode. There are three reasons for this:

1. Interrupt during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.
2. If the interrupt exception handling is started when the vector address has not been programmed yet or the flash memory is being programmed or erased, the vector would not be read correctly, possibly resulting in CPU runaway.
3. If an interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.



Notes: 1. Prewriting (setting erase block data to all 0) is not necessary.
2. The values of x, y, z, α, β, γ, ε, η, θ, and N are shown in section 24.6, Flash Memory Characteristics.
3. Verify data is read in 16-bit (W) units.
4. Set only one bit in EBR1or EBR2. More than one bit cannot be set.
5. Erasing is performed in block units. To erase a number of blocks, the individual blocks must be erased sequentially.

Figure 19.11 Erase/Erase-Verify Flowchart

19.9 Program/Erase Protection

There are three kinds of flash memory program/erase protection: hardware protection, software protection, and error protection.

19.9.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted because of a transition to reset (including an overflow reset by the WDT) or standby mode. Flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), erase block register 1 (EBR1), and erase block register 2 (EBR2) are initialized. In a reset via the RES pin, the reset state is not entered unless the RES pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the RES pin low for the RES pulse width specified in the AC Characteristics section.

19.9.2 Software Protection

Software protection can be implemented against programming/erasing of all flash memory blocks by clearing the SWE bit in FLMCR1 (this operation must be executed in the on-chip RAM or external memory). When software protection is in effect, setting the P or E bit in FLMCR1 does not cause a transition to program mode or erase mode. By setting the erase block register 1 (EBR1) and erase block register 2 (EBR2), erase protection can be set for individual blocks. When EBR1 and EBR2 are set to H'00, erase protection is set for all blocks.

19.9.3 Error Protection

In error protection, an error is detected when the CPU's runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is forcibly aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

When the following errors are detected during programming/erasing of flash memory, the FLER bit in FLMCR2 is set to 1, and the error protection state is entered.

- When flash memory is read during programming/erasing (including a vector read or instruction fetch)
- When an exception handling (excluding a reset) is started during programming/erasing
- When a SLEEP instruction is executed during programming/erasing
- When the CPU releases the bus mastership during programming/erasing

The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, but program mode or erase mode is forcibly aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit setting is enabled, and a

transition can be made to verify mode. The error protection state can be canceled by a power-on reset or in hardware standby mode.

19.10 Programmer Mode

In programmer mode, a PROM programmer can perform programming/erasing via a socket adapter, just like for a discrete flash memory. Use a PROM programmer which supports the Hitachi 512-kbyte flash memory on-chip MCU device type (FZTAT512V3A). A 12-MHz input clock is needed.

19.11 Power-Down States for Flash Memory

In user mode, the flash memory will operate in either of the following states:

- Normal operating mode
  The flash memory can be read.
- Standby mode
  All flash memory circuits are halted.

Table 19.7 shows the correspondence between the operating modes of this LSI and the flash memory. When the flash memory returns to normal operation from a standby state, a power supply circuit stabilization period is needed. When the flash memory returns to its normal operating state, bits STS3 to STS0 in SBYCR must be set to provide a wait time of at least 100 µs, even when the external clock is being used.

Table 19.7 Flash Memory Operating States

Operating Mode	Flash Memory Operating State
Active mode	Normal operating state
Sleep mode	Normal operating state
Standby mode	Standby state

19.12 Usage Notes

Precautions concerning the use of on-board programming mode, the RAM emulation function, and programmer mode are summarized below.

1. Use the specified voltages and timing for programming and erasing.
   Applied voltages in excess of the rating can permanently damage the device. Use a PROM programmer that supports the Hitachi microcomputer device type with 512-kbyte on-chip flash memory (FZTAT512V3A).

Do not select the HN27C4096 setting for the PROM programmer, and only use the specified socket adapter.

2. Reset the flash memory before turning on/off the power.

   When applying or disconnecting Vcc power, fix the RES pin low and place the flash memory in the hardware protection state. The power-on and power-off timing requirements should also be satisfied in the event of a power failure and subsequent recovery.

3. Powering on and off.

   Do not apply a high level to the FWE pin until V_{CC} has stabilized. Also, drive the FWE pin low before turning off V_{CC}.

   When applying or disconnecting V_{CC} power, fix the FWE pin low and place the flash memory in the hardware protection state.

   The power-on and power-off timing requirements should also be satisfied in the event of a power failure and subsequent recovery. The power-on and power-off timing in the H8S/2678 Series is shown in figure 19.12.

4. FWE application/disconnection.

   FWE application should be carried out when this LSI operation is in a stable condition. If this LSI operation is not stable, fix the FWE pin low and set the protection state.

The following points must be observed concerning FWE application and disconnection to prevent unintentional programming or erasing of flash memory:

- Apply FWE when the V_{CC} voltage has stabilized within its rated voltage range.
- In boot mode, apply and disconnect FWE during a reset.
- In user program mode, FWE can be switched between high and low level regardless of the reset state. FWE input can also be switched during execution of a program in flash memory.
- Do not apply FWE if program runaway has occurred.
- Disconnect FWE only when the SWE, ESU, PSU, EV, PV, and E bits in FLMCR1 are cleared.

5. Do not apply a constant high level to the FWE pin.

   Apply a high level to the FWE pin only when programming or erasing flash memory. Also,

When the SWE bit is set, data in flash memory can be rewritten. When the SWE bit is set to 1, data in flash memory can be read only in program-verify/erase-verify mode. Access flash memory only for verify operations (verification during programming/erasing). Also, do not clear the SWE bit during programming, erasing, or verifying. Similarly, when using the RAM emulation function, the SWE bit must be cleared before executing a program or reading data in flash memory.

However, the RAM area overlapping flash memory space can be read and written to regardless of whether the SWE bit is set or cleared.

8. Do not use interrupts while flash memory is being programmed or erased.

   All interrupt requests, including NMI, should be disabled during programming/erasing the flash memory to give priority to program/erase operations.

9. Do not perform additional programming. Erase the memory before reprogramming.

   In on-board programming, perform only one programming operation on a 128-byte programming unit block. In programmer mode, too, perform only one programming operation on a 128-byte programming unit block. Programming should be carried out with the entire programming unit block erased.

10. Before programming, check that the chip is correctly mounted in the PROM programmer.

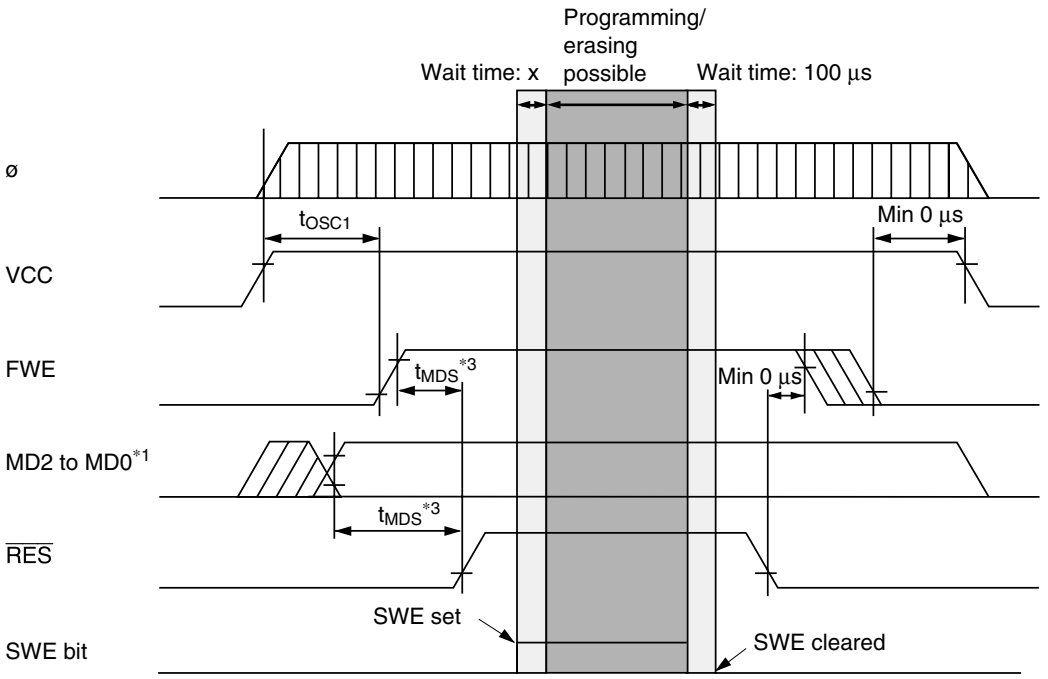
    Overcurrent damage to the device can result if the index marks on the PROM programmer socket, socket adapter, and chip are not correctly aligned.

11. Do not touch the socket adapter or chip during programming.

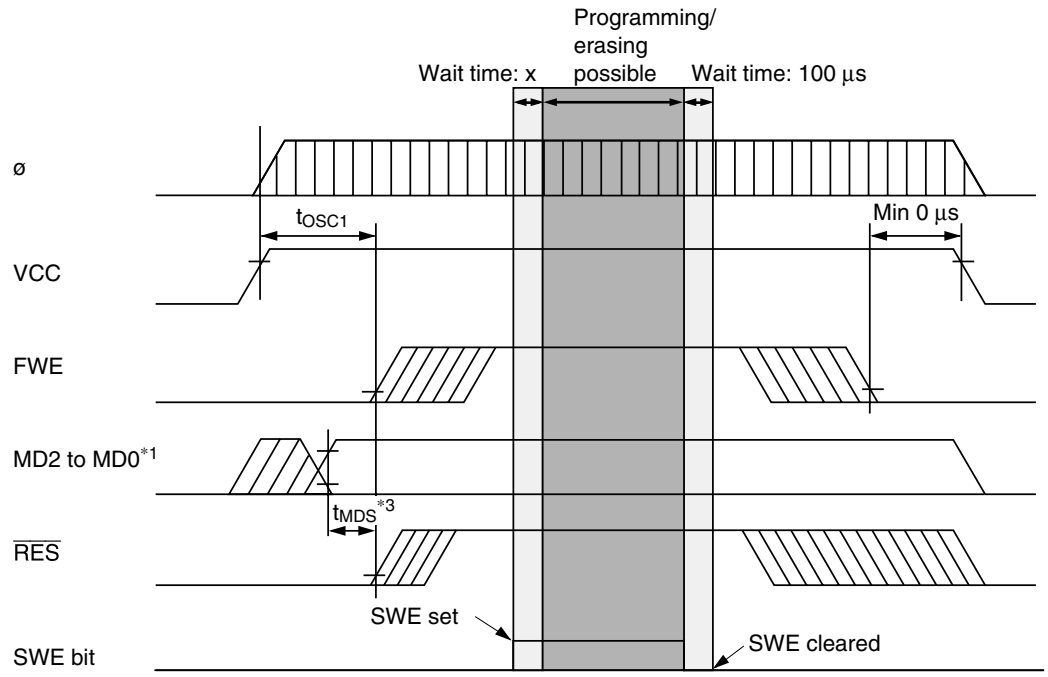
    Touching either of these can cause contact faults and write errors.

12. Apply the reset signal after the SWE, bit is cleared during its operation.

    The reset signal is applied at least 100 µs after the SWE bit has been cleared.



(1) Boot Mode



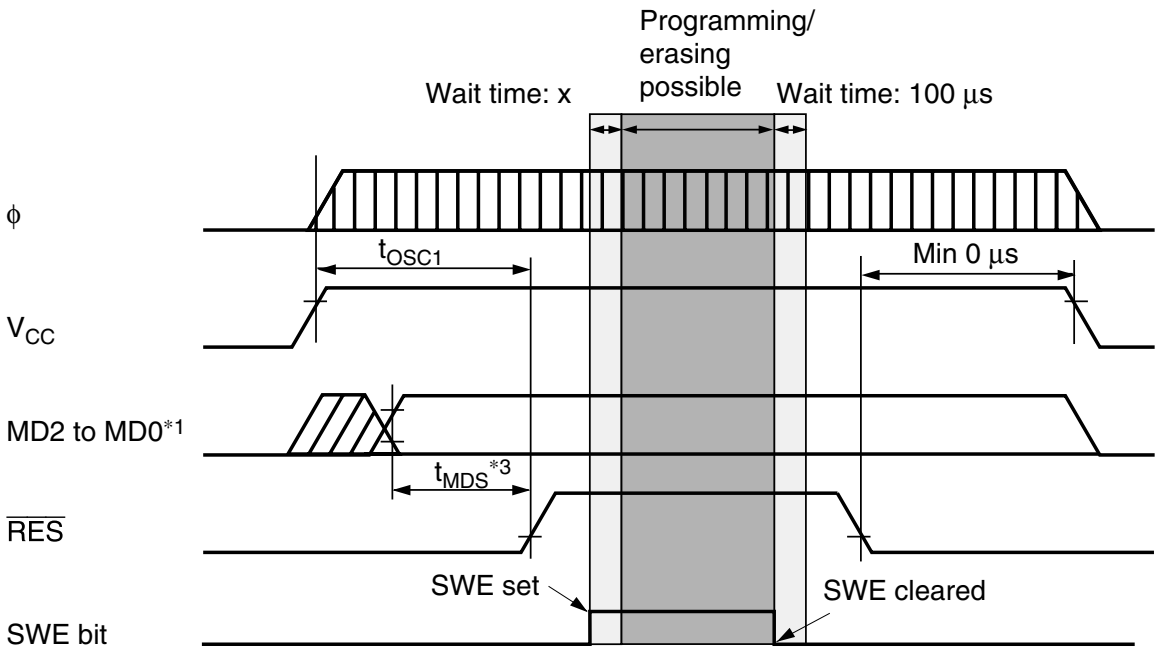
(2) User Program Mode

Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*2

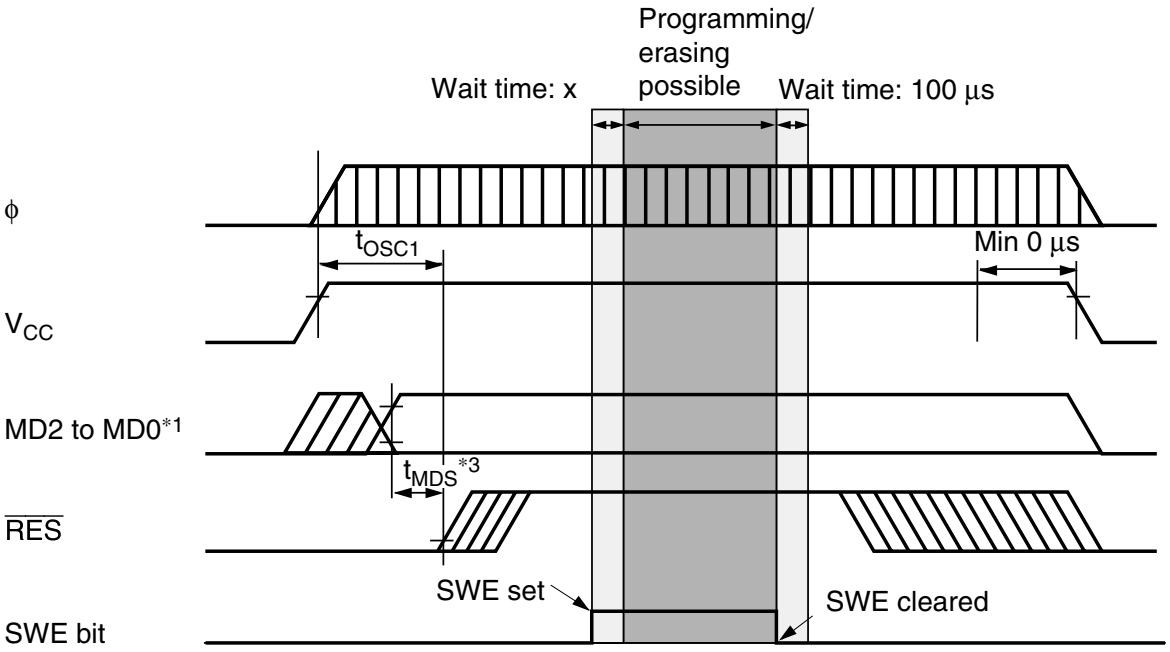
Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: 1. Except when switching modes, the level of the mode pins (MD2 to MD0) must be fixed until power-off by pulling the pins up or down.
2. See section 24.6, Flash Memory Characteristics.
3. Mode programming setup time t_{MDS} (min) = 200 ns

Figure 19.12 Power-On/Off Timing (H8S/2678 Series)



(1) Boot Mode



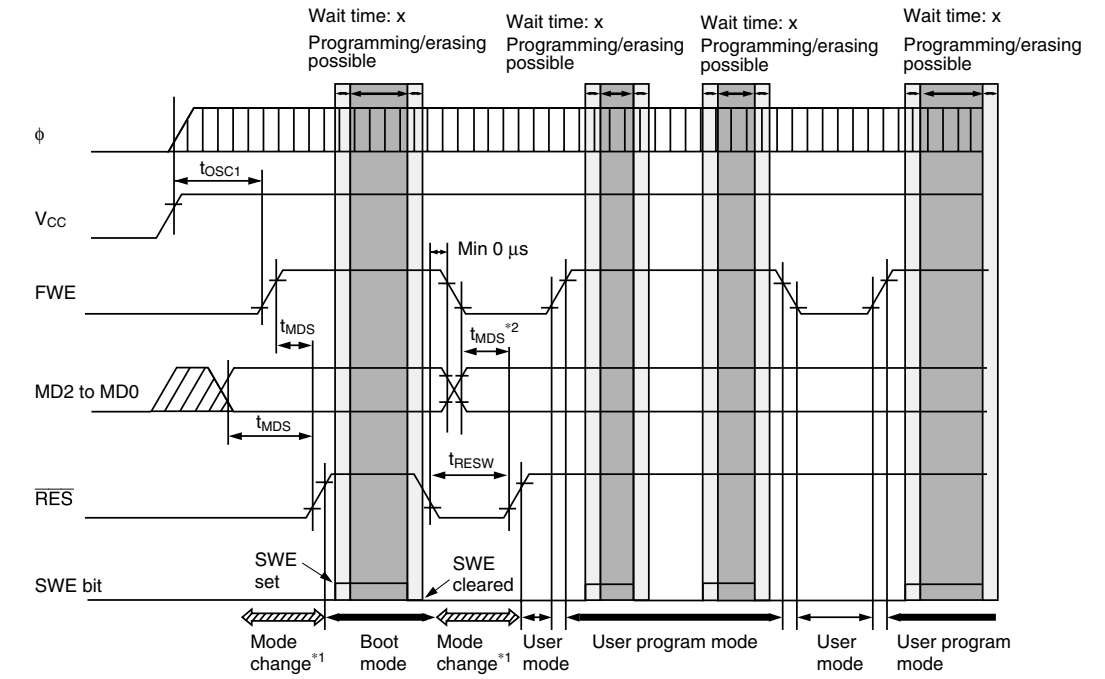
(2) User Program Mode

Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*2

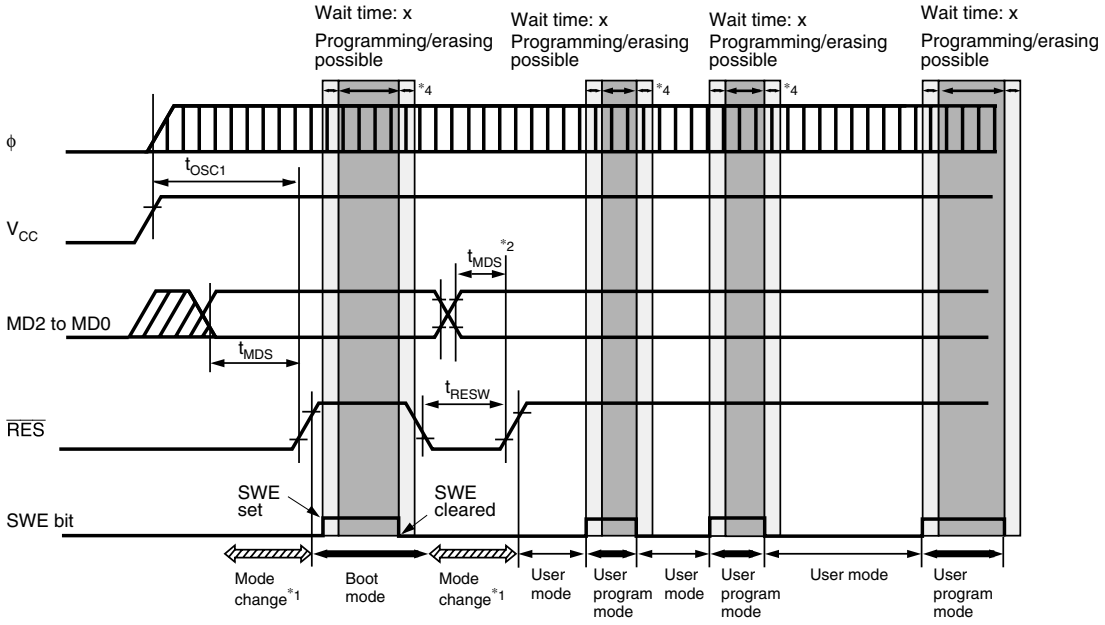
Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: 1. Except when switching modes, the level of the mode pins (MD2 to MD0) must be fixed until power-off by pulling the pins up or down.
2. See section 24.6, Flash Memory Characteristics.
3. Mode programming setup time t_{MDS} (min) = 200 ns

Figure 19.13 Power-On/Off Timing (H8S/2678R Series)



(1) H8S/2678 Series



(2) H8S/2678R Series

Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*3

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: 1. When entering boot mode or making a transition from boot mode to another mode, mode switching must be carried out by means of RES input. The state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$) will change during this switchover interval (the interval during which the $\overline{RES}$ pin input is low), and therefore these pins should not be used as output signals during this time.
2. When making a transition from boot mode to another mode, a mode programming setup time t_{MDS} (min) of 200 ns is necessary with respect to $\overline{RES}$ clearance timing.
3. See section 24.6, Flash Memory Characteristics.
4. Wait time: 100 µs

Figure 19.14 Mode Transition Timing
(Example: Boot Mode → User Mode ↔ User Program Mode)

19.13 Note on Switching from F-ZTAT Version to Masked ROM Version

Care is required if application software developed on the F-ZTAT version is used when the F-ZTAT version is switched to the masked ROM version product.

If an address in which a register for the F-ZTAT version is present is read (see section 23.1, Register Addresses) in the masked ROM version, an undefined value will be returned.

If application software developed on the F-ZTAT version is used in the masked ROM version product, the state of the FWE pin cannot be judged. The program must be modified so that the part of reprogramming (erasing/programming) the flash memory and the part of the RAM emulation are not started.

Also, the mode pin of boot mode must not be set in the masked ROM version.

Note: This note is applied to all products in the F-ZTAT version and in the masked ROM version of same series with the different ROM size.

Section 20 Masked ROM

This series microcomputer has 64, 128, or 256 kbytes of on-chip masked ROM. The on-chip ROM is connected to the CPU, data transfer controller (DTC), and DMA controller (DMAC) with a 16-bit data bus. The on-chip ROM can be accessed by the CPU, DTC, and DMAC in 8 or 16-bit units. The data in the on-chip ROM can always be accessed in one state.

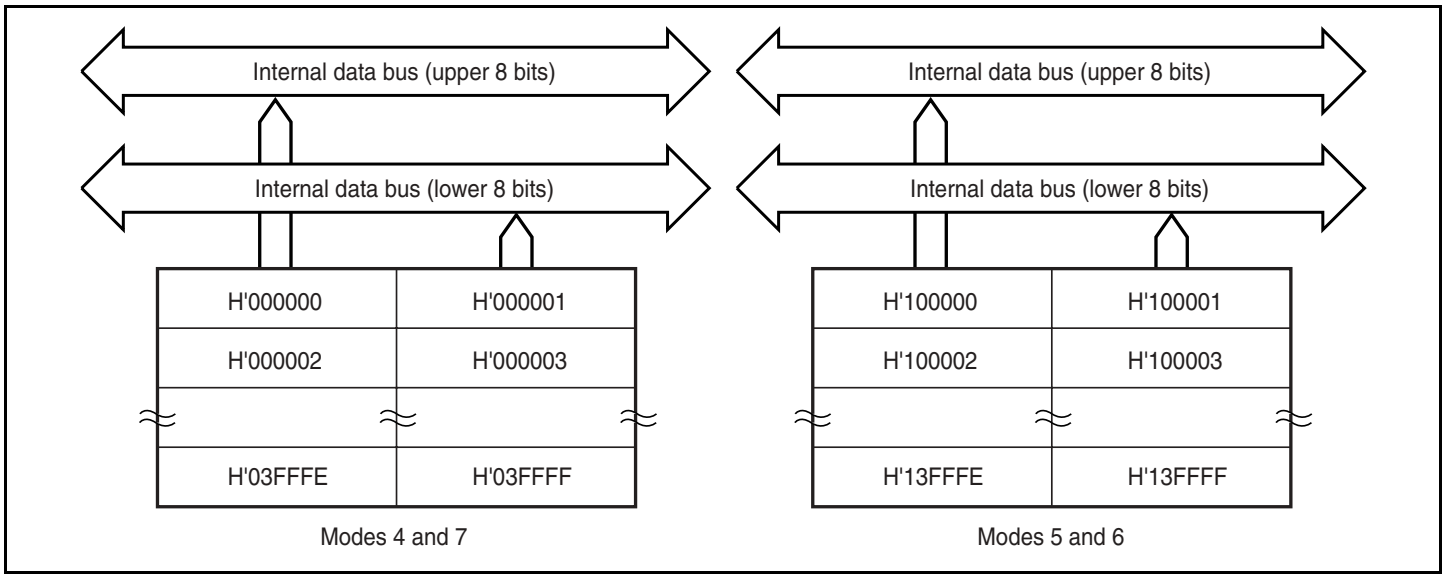


Figure 20.1 Block Diagram of 256-Kbyte Masked ROM (HD6432676)

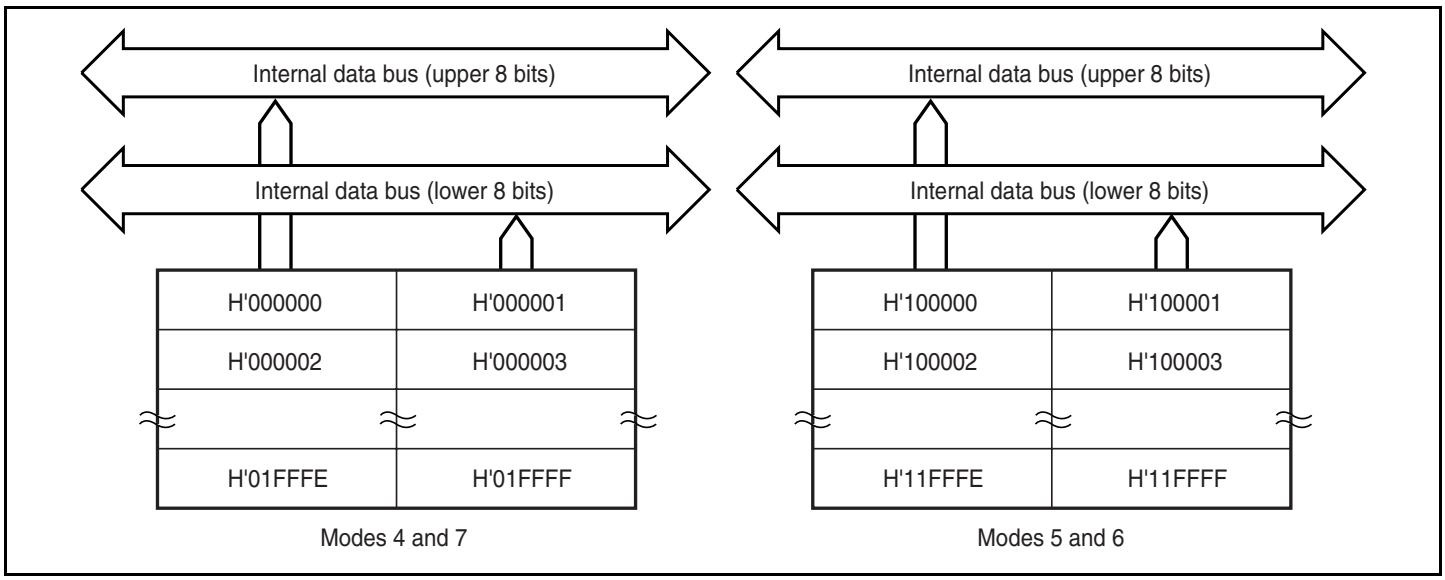
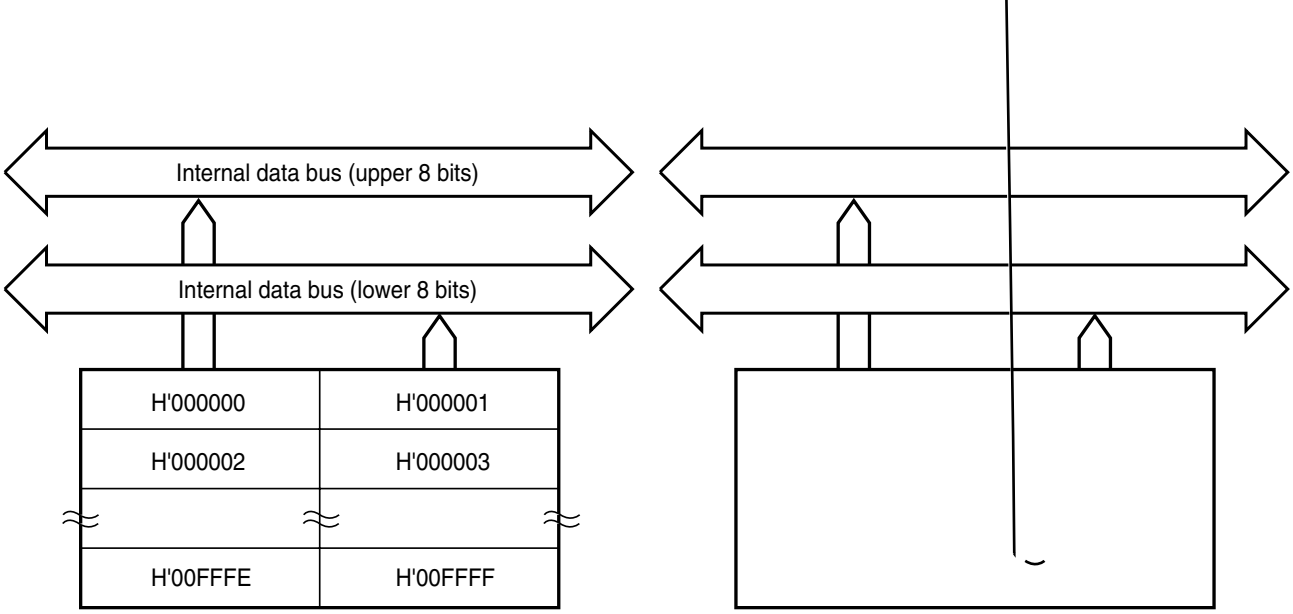


Figure 20.2 Block Diagram of 128-Kbyte Masked ROM (HD6432675)



Modes 4 and 7

Modes 5 and 6

Section 21 Clock Pulse Generator

This LSI has an on-chip clock pulse generator (CPG) that generates the system clock (ø) and internal clocks.

The clock pulse generator consists of an oscillator circuit, PLL circuit, and divider.

Figure 21.1 shows a block diagram of the clock pulse generator.

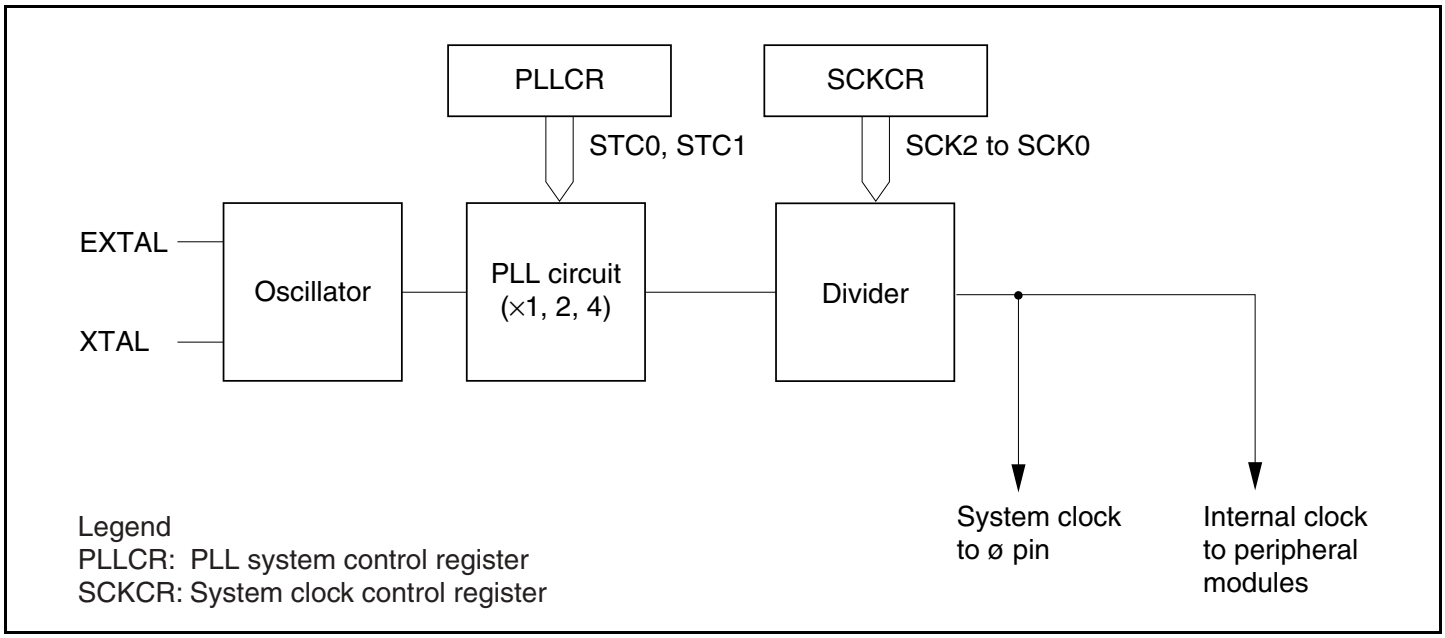


Figure 21.1 Block Diagram of Clock Pulse Generator

The frequency can be changed by means of the PLL circuit. Frequency changes are made by software by means of settings in the PLL control register (PLLCR) and the system clock control register (SCKCR).

21.1 Register Descriptions

The clock pulse generator has the following registers.

- System clock control register (SCKCR)
- PLL control register (PLLCR)

21.1.1 System Clock Control Register (SCKCR)

SCKCR controls ø clock output and selects operation when the frequency multiplication factor used by the PLL circuit is changed, and the division ratio used by the divider.

Bit	Bit Name	Initial Value	R/W	Description
7	PSTOP	0	R/W	ø Clock Output Disable Controls ø output. Normal Operation 0: ø output 1: Fixed high Sleep Mode 0: ø output 1: Fixed high Software Standby Mode 0: Fixed high 1: Fixed high Hardware Standby Mode 0: High impedance 1: High impedance All module clock stop mode 0: ø output 1: Fixed high
6	—	0	R/W	Reserved This bit can be read from or written to. However, The write value should always be 0.
5	—	0	R/W	Reserved
4	—	0	R/W	These bits are always read as 0. However, the write value should always be 0.
3	STCS	0	R/W	Frequency Multiplication Factor Switching Mode Select Selects the operation when the PLL circuit frequency multiplication factor is changed. 0: Specified multiplication factor is valid after transition to software standby mode 1: Specified multiplication factor is valid immediately after STC1 and STC0 bits are rewritten

Bit	Bit Name	Initial Value	R/W	Description
2	SCK2	0	R/W	System Clock Select 2 to 0
1	SCK1	0	R/W	Select the division ratio.
0	SCK0	0	R/W	000: 1/1
				001: 1/2
				010: 1/4
				011: 1/8
				100: 1/16
				101: 1/32
				11X: Setting prohibited

X: Don't care

21.1.2 PLL Control Register (PLLCR)

PLLCR sets the frequency multiplication factor used by the PLL circuit.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	0	—	Reserved These bits are always read as 0 and cannot be modified.
3	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.
2	—	0	R/W	Reserved This bit is always read as 0 and cannot be modified.
1	STC1	0	R/W	Frequency Multiplication Factor
0	STC0	0	R/W	The STC bits specify the frequency multiplication factor used by the PLL circuit. 00: × 1 01: × 2 10: × 4 11: Setting prohibited

21.2 Oscillator

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock.

21.2.1 Connecting a Crystal Resonator

A crystal resonator can be connected as shown in the example in figure 21.2. Select the damping resistance R_d according to table 20.1. An AT-cut parallel-resonance type should be used.

Figure 21.3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 21.2.

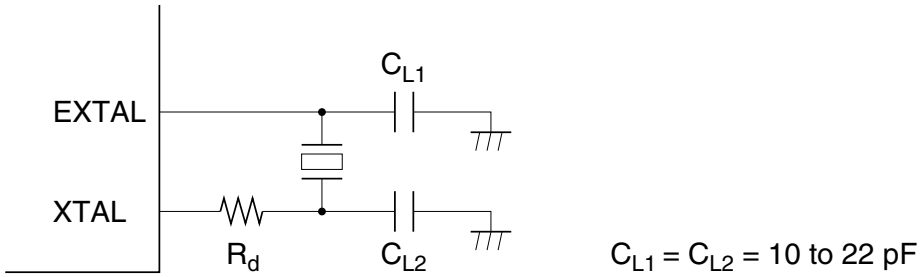


Figure 21.2 Connection of Crystal Resonator (Example)

Table 21.1 Damping Resistance Value

Frequency (MHz)	8	12	16	20	25
R_d (Ω)	200	0	0	0	0

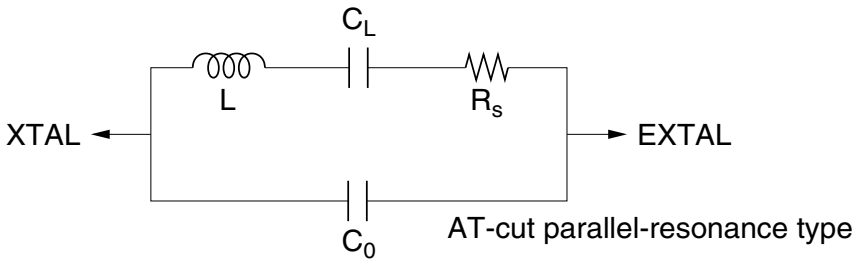


Figure 21.3 Crystal Resonator Equivalent Circuit

Table 21.2 Crystal Resonator Characteristics

Frequency (MHz)	8	12	16	20	25
R_S max (Ω)	80	60	50	40	40
C_0 max (pF)	7	7	7	7	7

21.2.2 External Clock Input

An external clock signal can be input as shown in the examples in figure 21.4. If the XTAL pin is left open, make sure that parasitic capacitance is no more than 10 pF. When the counter clock is input to the XTAL pin, make sure that the external clock is held high in standby mode.

Table 21.3 shows the input conditions for the external clock

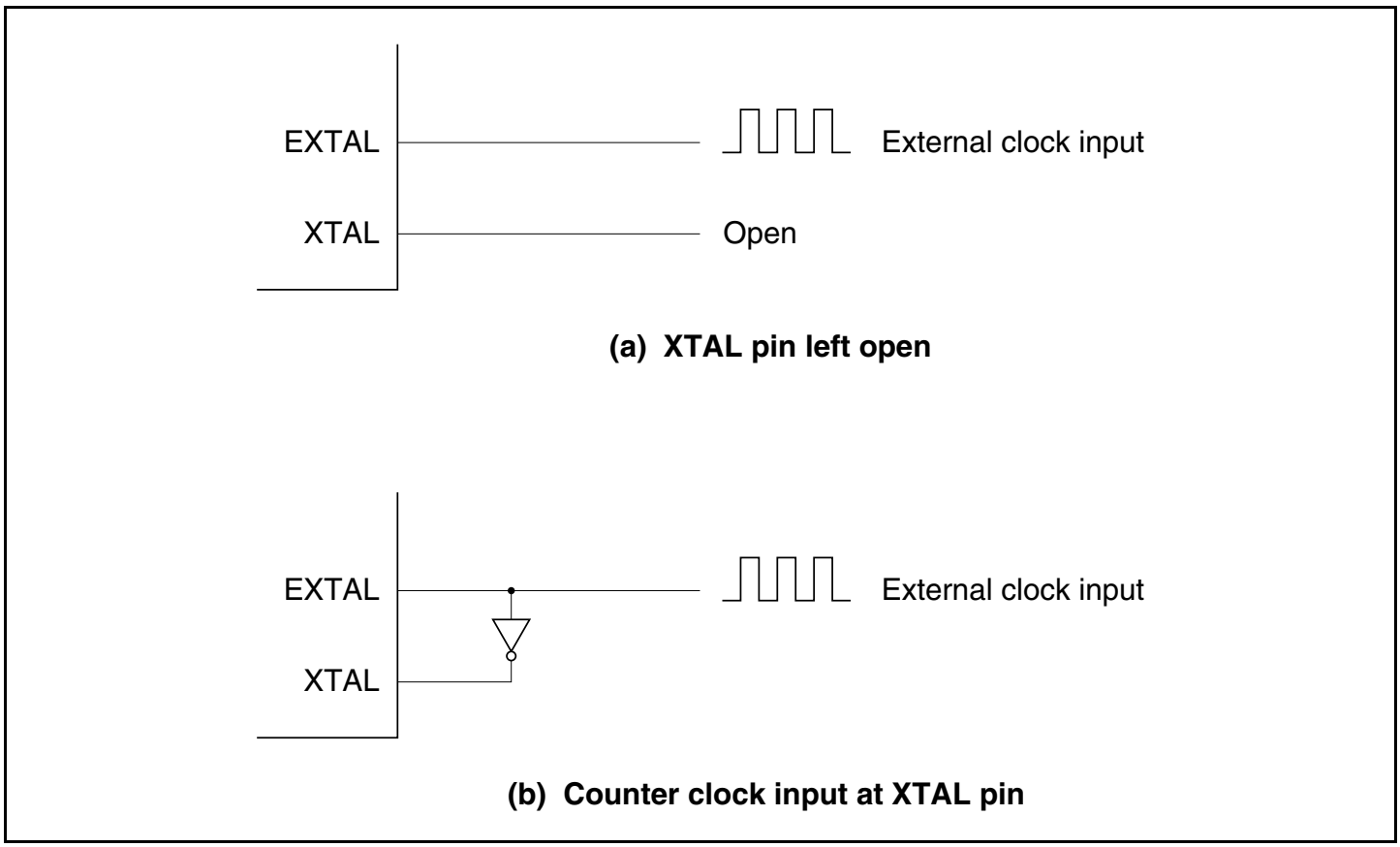


Figure 21.4 External Clock Input (Examples)

Table 21.3 External Clock Input Conditions

Item	Symbol	V_{cc} = 3.0 V to 3.6 V Min	Max	Unit	Test Conditions
External clock input low pulse width	t_{EXL}	15	—	ns	Figure 21.5
External clock input high pulse width	t_{EXH}	15	—	ns	
External clock rise time	t_{EXr}	—	5	ns	
External clock fall time	t_{EXf}	—	5	ns	
Clock low pulse width	t_{CL}	0.4	0.6	t_{cyc}	
Clock high pulse width	t_{CH}	0.4	0.6	t_{cyc}	

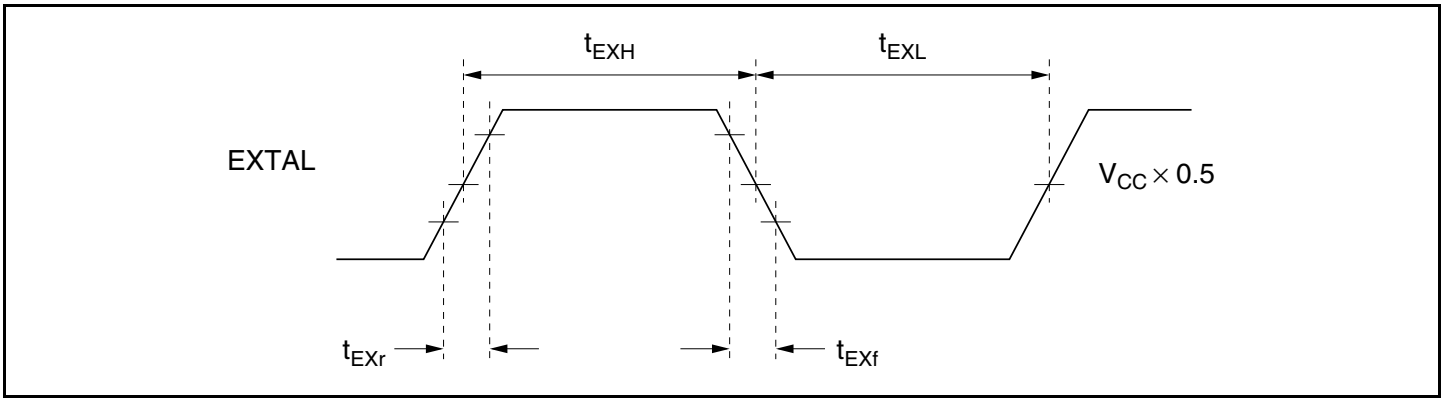


Figure 21.5 External Clock Input Timing

21.3 PLL Circuit

The PLL circuit has the function of multiplying the frequency of the clock from the oscillator by a factor of 1, 2, or 4. The multiplication factor is set with the STC1 and the STC0 bits in PLLCR. The phase of the rising edge of the internal clock is controlled so as to match that of the rising edge of the EXTAL pin.

When the multiplication factor of the PLL circuit is changed, the operation varies according to the setting of the STCS bit in SCKCR.

When STCS = 0, the setting becomes valid after a transition to software standby mode. The transition time count is performed in accordance with the setting of bits STS3 to STS0 in SBYCR. For details on SBYCR, refer to section 22.1.1, Standby Control Register (SBYCR).

1. The initial PLL circuit multiplication factor is 1.
2. A value is set in bits STS3 to STS0 to give the specified transition time.

3. The target value is set in bits STC1 and STC0, and a transition is made to software standby mode.
4. The clock pulse generator stops and the value set in STC1 and STC0 becomes valid.
5. Software standby mode is cleared, and a transition time is secured in accordance with the setting in STS3 to STS0.
6. After the set transition time has elapsed, this LSI resumes operation using the target multiplication factor.

When STCS = 1, this LSI operates using the new multiplication factor immediately after bits STC1 and STC0 are rewritten.

21.4 Frequency Divider

The frequency divider divides the PLL output clock to generate a 1/2, 1/4, 1/8, 1/16, or 1/32 clock.

21.5 Usage Notes

21.5.1 Notes on Clock Pulse Generator

1. The following points should be noted since the frequency of ∅ changes according to the setting of SCKCR and PLLCR.

   Select the clock division ratio that is within the operation guaranteed range of clock cycle time tcyc shown in the AC timing of Electrical Characteristics. In other words, the range of ∅ must be specified from 8 MHz (min) to 33 MHz (max); outside of this range must be prevented.
2. All the on-chip peripheral modules operate on the ∅. Therefore, note that the time processing of modules such as a timer and SCI differ before and after changing the clock division ratio. In addition, wait time for clearing software standby mode differs by changing the clock division ratio. See the description, Setting Oscillation Stabilization Time after Clearing Software Standby Mode in section 22.2.3, Software Standby Mode, for details.
3. Note that the frequency of ∅ will be changed when setting SCKCR or PLLCR while executing the external bus cycle with the write-data-buffer function or the EXDMAC.

21.5.2 Notes on Resonator

Since various characteristics related to the resonator are closely linked to the user's board design, thorough evaluation is necessary on the user's part, using the oscillator connection examples shown in this section as a guide. As the parameters for the oscillation circuit will depend on the floating capacitance of the resonator and the user board, the parameters should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is not applied to the oscillator pin.

21.5.3 Notes on Board Design

When using the crystal resonator, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins. Other signal lines should be routed away from the oscillation circuit to prevent induction from interfering with correct oscillation. See figure 21.6.

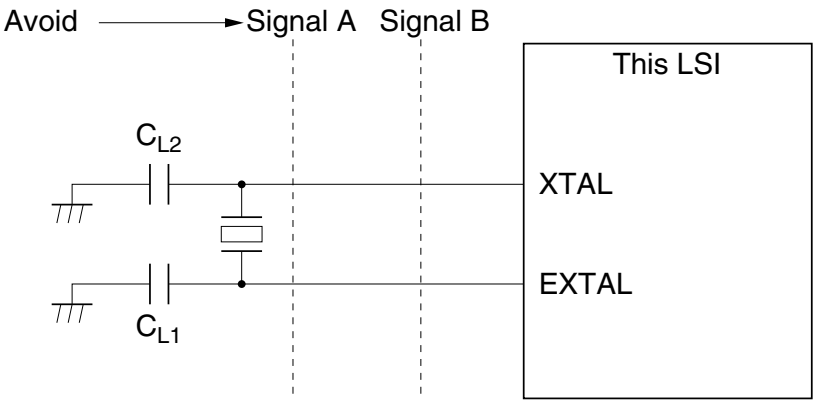
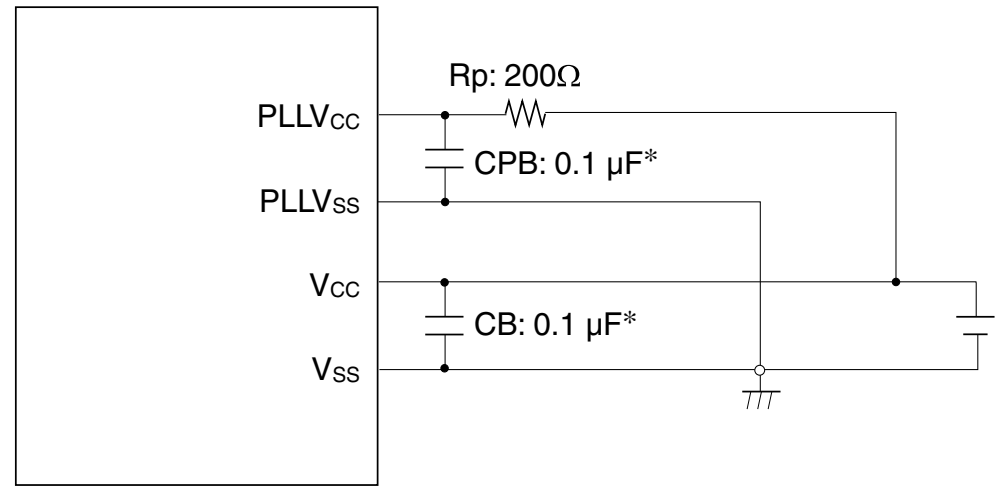


Figure 21.6 Note on Board Design for Oscillation Circuit

Figure 21.7 shows the external circuitry recommended for the PLL circuit. Separate PLLVcc and PLLVss from the other Vcc and Vss lines at the board power supply source, and be sure to insert bypass capacitors CPB and CB close to the pins.



Note: * CB and CPB are laminated ceramic capacitors.

Figure 21.7 Recommended External Circuitry for PLL Circuit

Section 22 Power-Down Modes

In addition to the normal program execution state, this LSI has power-down modes in which operation of the CPU and oscillator is halted and power dissipation is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip peripheral modules, and so on.

This LSI's operating modes are high-speed mode and six power down modes:

- Clock division mode
- Sleep mode
- Module stop mode
- All module clock stop mode
- Software standby mode
- Hardware standby mode

Sleep mode is a CPU state, clock division mode is a CPU and bus master state, and module stop mode is an on-chip peripheral function (including bus masters other than the CPU) state. A combination of these modes can be set.

After a reset, this LSI is in high-speed mode.

Table 22.1 shows the internal states of this LSI in each mode. Figure 21.1 shows the mode transition diagram.

Table 22.1 Operating Modes

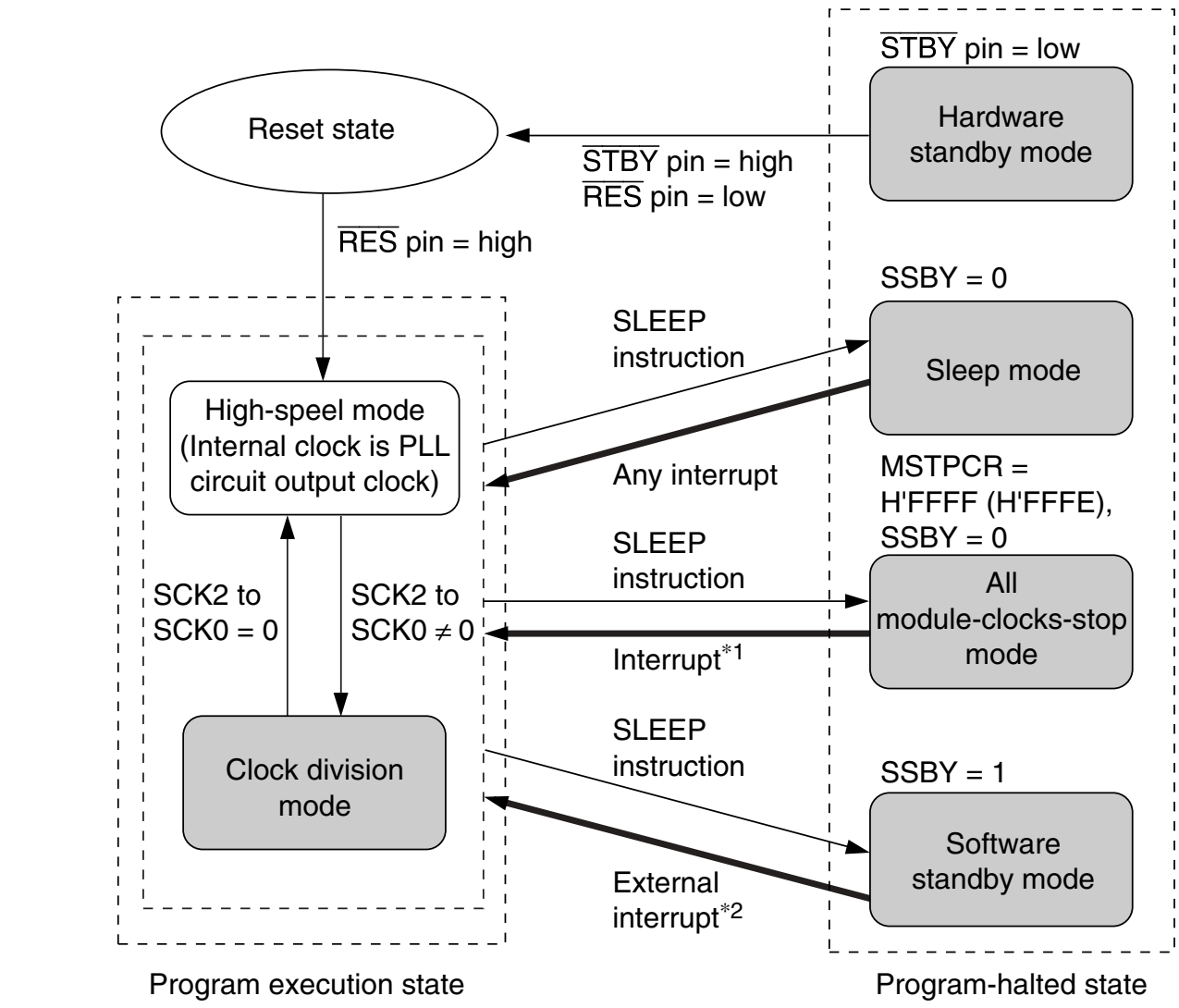
Operating State		High Speed Mode	Clock Division Mode	Sleep Mode	Module Stop Mode	All Module Clock Stop Mode	Software Standby Mode	Hardware Standby Mode
Clock pulse generator		Functions	Functions	Functions	Functions	Functions	Halted	Halted
CPU	Instruction execution	Functions	Functions	Halted	Functions	Halted	Halted	Halted
	Register			Retained			Retained	Undefined
External interrupts	NMI	Functions	Functions	Functions	Functions	Functions	Functions	Halted
	IRQ0 to 15							
Peripheral functions	WDT	Functions	Functions	Functions	Functions	Functions	Halted (Retained)	Halted (Reset)
	TMR	Functions	Functions	Functions	Halted (Retained)	Functions/ Halted (Retained)*	Halted (Retained)	Halted (Reset)
	EXDMAC	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	DMAC	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	DTC	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	TPU	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	PPG	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	D/A	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	A/D	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Reset)	Halted (Reset)
	SCI	Functions	Functions	Functions	Halted (Reset)	Halted (Reset)	Halted (Reset)	Halted (Reset)
	RAM	Functions	Functions	Functions	Functions	Functions	Retained	Retained
	I/O	Functions	Functions	Functions	Functions	Retained	Retained	High impedance

Notes: "Halted (Retained)" in the table means that internal register values are retained and internal operations are suspended.

"Halted (Reset)" in the table means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).

* The active or halted state can be selected by means of the MSTP0 bit in MSTPCR.



: Transition after exception handling : Power- down mode

Notes: 1. NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$, 8-bit timer interrupts, watchdog timer interrupts.
(8-bit timer interrupts are valid when MSTP0 = 0.)
2. NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$
(IRQ0 to IRQ15 are valid when the corresponding bit in SSIER is 1.)

- When a transition is made between modes by means of an interrupt, the transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.
- From any state, a transition to hardware standby mode occurs when $\overline{\text{STBY}}$ is driven low.
- From any state except hardware standby mode, a transition to the reset state occurs when $\overline{\text{RES}}$ is driven low.

Figure 22.1 Mode Transitions

22.1 Register Descriptions

The registers relating to the power-down mode are shown below. For details on the system clock control register (SCKCR), refer to section 21.1.1, System Clock Control Register (SCKCR).

- System clock control register (SCKCR)
- Standby control register (SBYCR)
- Module stop control register H (MSTPCRH)
- Module stop control register L (MSTPCRL)

22.1.1 Standby Control Register (SBYCR)

SBYCR performs software standby mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby This bit specifies the transition mode after executing the SLEEP instruction 0: Shifts to sleep mode after the SLEEP instruction is executed 1: Shifts to software standby mode after the SLEEP instruction is executed This bit does not change when clearing the software standby mode by using external interrupts and shifting to normal operation. This bit should be written 0 when clearing.
6	OPE	1	R/W	Output Port Enable Specifies whether the output of the address bus and bus control signals (CS0 to CS7, AS, RD, HWR, LWR, UCAS, LCAS) is retained or set to the high-impedance state in software standby mode. 0: In software standby mode, address bus and bus control signals are high-impedance 1: In software standby mode, address bus and bus control signals retain output state

Bit	Bit Name	Initial Value	R/W	Description
5	—	0	—	Reserved
4	—	0	—	These bits are always read as 0. The initial value should not be changed.
3	STS3	1	R/W	Standby Timer Select 3 to 0
2	STS2	1	R/W	These bits select the time the MCU waits for the clock to stabilize when software standby mode is cleared by an external interrupt. With crystal oscillation, refer to table 22.2 and make a selection according to the operating frequency so that the standby time is at least the oscillation stabilization time. With an external clock, a PLL circuit stabilization time is necessary. Refer to table 22.2 to set the wait time. When DRAM is used and self-refreshing in the software standby state is selected, note that the DRAM's tRAS (self-refresh RAS pulse width) specification must be satisfied. With the F-ZTAT version, a flash memory stabilization time must be provided. 0000: Setting prohibited 0001: Setting prohibited 0010: Setting prohibited 0011: Setting prohibited 0100: Setting prohibited 0101: Standby time = 64 states 0110: Standby time = 512 states 0111: Standby time = 1024 states 1000: Standby time = 2048 states 1001: Standby time = 4096 states 1010: Standby time = 16384 states 1011: Standby time = 32768 states 1100: Standby time = 65536 states 1101: Standby time = 131072 states 1110: Standby time = 262144 states 1111: Standby time = 524288 states
1	STS1	1	R/W	
0	STS0	1	R/W	

22.1.2 Module Stop Control Registers H and L (MSTPCRH, MSTPCRL)

MSTPCR performs module stop mode control.

Setting a bit to 1, the corresponding module enters module stop mode, while clearing the bit to 0 clears the module stop mode.

MSTPCRH

Bit	Bit Name	Initial Value	R/W	Module
15	ACSE	0	R/W	All-Module-Clocks-Stop Mode Enable Enables or disables all-module-clocks-stop mode, in which, when the CPU executes a SLEEP instruction after module stop mode has been set for all the on-chip peripheral functions controlled by MSTPCR or the on-chip peripheral functions except the TMR. 0: All-module-clocks-stop mode disabled 1: All-module-clocks-stop mode enabled
14	MSTP14	0	R/W	EXDMA controller (EXDMAC)
13	MSTP13	0	R/W	DMA controller (DMAC)
12	MSTP12	0	R/W	Data transfer controller (DTC)
11	MSTP11	1	R/W	16-bit timer-pulse unit (TPU)
10	MSTP10	1	R/W	Programmable pulse generator (PPG)
9	MSTP9	1	R/W	D/A converter (channels 0 and 1)
8	MSTP8	1	R/W	D/A converter (channels 2 and 3)

MSTPCRL

Bit	Bit Name	Initial Value	R/W	Module
7	MSTP7	1	R/W	—
6	MSTP6	1	R/W	A/D converter
5	MSTP5	1	R/W	—
4	MSTP4	1	R/W	—
3	MSTP3	1	R/W	Serial communication interface 2 (SCI_2)
2	MSTP2	1	R/W	Serial communication interface 1 (SCI_1)
1	MSTP1	1	R/W	Serial communication interface 0 (SCI_0)
0	MSTP0	1	R/W	8-bit timer (TMR)

22.2 Operation

22.2.1 Clock Division Mode

When bits SCK2 to SCK0 in SCKCR are set to a value from 001 to 101, a transition is made to clock division mode at the end of the bus cycle. In clock division mode, the CPU, bus masters, and on-chip peripheral functions all operate on the operating clock (1/2, 1/4, 1/8, 1/16, or 1/32) specified by bits SCK2 to SCK0.

Clock division mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition is made to high-speed mode at the end of the bus cycle, and clock division mode is cleared.

If a SLEEP instruction is executed while the SSBY bit in SBYCR is cleared to 0, the chip enters sleep mode. When sleep mode is cleared by an interrupt, clock division mode is restored.

If a SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1, the chip enters software standby mode. When software standby mode is cleared by an external interrupt, clock division mode is restored.

When the RES pin is driven low, the reset state is entered and clock division mode is cleared. The same applies to a reset caused by watchdog timer overflow.

When the STBY pin is driven low, a transition is made to hardware standby mode.

22.2.2 Sleep Mode

Transition to Sleep Mode: When the SLEEP instruction is executed when the SSBY bit is 0 in SBYCR, the CPU enters the sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registers are retained. Other peripheral functions do not stop.

Exiting Sleep Mode: Sleep mode is exited by any interrupt, or signals at the RES, or STBY pins.

- Exiting Sleep Mode by Interrupts:
  When an interrupt occurs, sleep mode is exited and interrupt exception processing starts. Sleep mode is not exited if the interrupt is disabled, or interrupts other than NMI are masked by the CPU.
- Exiting Sleep Mode by RES pin:
  Setting the RES pin level low selects the reset state. After the stipulated reset input duration, driving the RES pin high starts the CPU performing reset exception processing.
- Exiting Sleep Mode by STBY Pin:
  When the STBY pin level is driven low, a transition is made to hardware standby mode.

22.2.3 Software Standby Mode

Transition to Software Standby Mode: If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, software standby mode is entered. In this mode, the CPU, on-chip peripheral functions, and oscillator all stop. However, the contents of the CPU's internal registers, RAM data, and the states of on-chip peripheral functions other than the SCI and A/D converter, and I/O ports, are retained. Whether the address bus and bus control signals are placed in the high-impedance state or retain the output state can be specified by the OPE bit in SBYCR.

In this mode the oscillator stops, and therefore power dissipation is significantly reduced.

Clearing Software Standby Mode: Software standby mode is cleared by an external interrupt (NMI pin, or pins IRQ0 to IRQ15), or by means of the RES pin or STBY pin. Setting the SSI bit in SSIER to 1 enables IRQ0 to IRQ15 to be used as software standby mode clearing sources.

Clearing with an Interrupt:

When an NMI or IRQ0 to IRQ15 interrupt request signal is input, clock oscillation starts, and after the elapse of the time set in bits STS2 to STS0 in SBYCR, stable clocks are supplied to the entire LSI, software standby mode is cleared, and interrupt exception handling is started.

When clearing software standby mode with an IRQ0 to IRQ15 interrupt, set the corresponding enable bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ0 to IRQ15 is generated. Software standby mode cannot be cleared if the interrupt has been masked on the CPU side or has been designated as a DTC activation source.

Clearing with the RES Pin:

When the RES pin is driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire LSI. Note that the RES pin must be held low until clock oscillation stabilizes. When the RES pin goes high, the CPU begins reset exception handling.

Clearing with the STBY Pin:

When the STBY pin is driven low, a transition is made to hardware standby mode.

Setting Oscillation Stabilization Time after Clearing Software Standby Mode: Bits STS3 to STS0 in SBYCR should be set as described below.

Using a Crystal Oscillator:

Set bits STS3 to STS0 so that the standby time is more than the oscillation stabilization time.

Table 22.2 shows the standby times for operating frequencies and settings of bits STS3 to STS0.

Using an External Clock:

A PLL circuit stabilization time is necessary. Refer to table 22.2 to set the wait time.

Table 22.2 Oscillation Stabilization Time Settings

					ø* [MHz]						
STS3	STS2	STS1	STS0	Standby Time	33	25	20	13	10	8	Unit
0	0	0	0	Reserved	—	—	—	—	—	—	µs
			1	Reserved	—	—	—	—	—	—	
		1	0	Reserved	—	—	—	—	—	—	
			1	Reserved	—	—	—	—	—	—	
	1	0	0	Reserved	—	—	—	—	—	—	
			1	64	1.9	2.6	3.2	4.9	6.4	8.0	
		1	0	512	15.5	20.5	25.6	39.4	51.2	64.0	
			1	1024	31.0	41.0	51.2	78.8	102.4	128.0	
1	0	0	0	2048	62.1	81.9	102.4	157.5	204.8	256.0	
			1	4096	0.12	0.16	0.20	0.32	0.41	0.51	ms
		1	0	16384	0.50	0.66	0.82	1.26	1.64	2.05	
			1	32765	0.99	1.31	1.64	2.52	3.28	4.10	
	1	0	0	65536	1.99	2.62	3.28	5.04	6.55	8.19	
			1	131072	3.97	5.24	6.55	10.08	13.11	16.38	
		1	0	262144	7.94	10.49	13.11	20.16	26.21	32.77	
			1	524288	15.89	20.97	26.21	40.33	52.43	65.54	

☐ : Recommended time setting

Note: ø is the frequency divider output.

Software Standby Mode Application Example: Figure 22.2 shows an example in which a transition is made to software standby mode at the falling edge on the NMI pin, and software standby mode is cleared at the rising edge on the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in INTCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then cleared at the rising edge on the NMI pin.

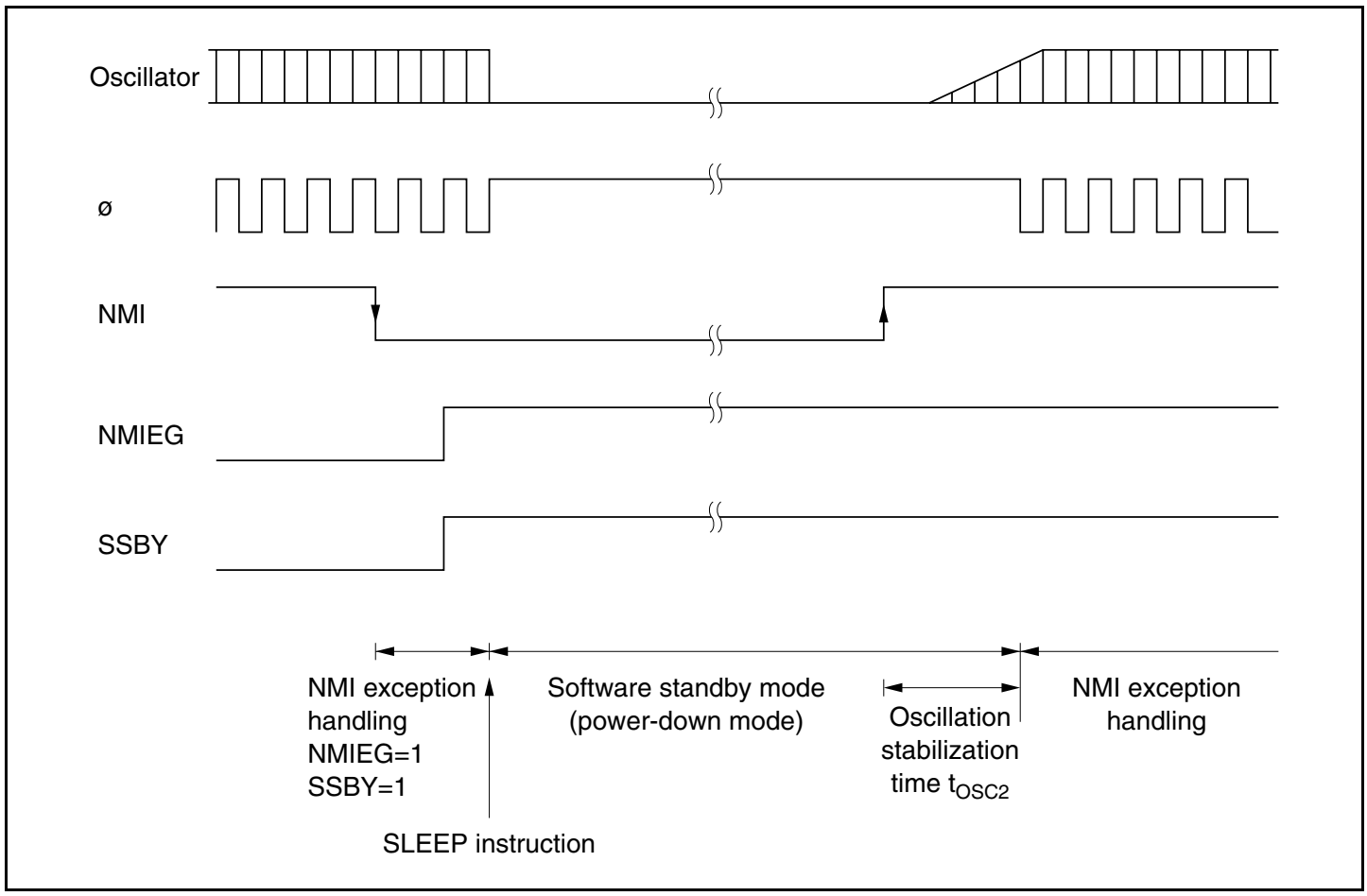


Figure 22.2 Software Standby Mode Application Example

22.2.4 Hardware Standby Mode

Transition to Hardware Standby Mode: When the STBY pin is driven low, a transition is made to hardware standby mode from any mode.

In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power dissipation. As long as the prescribed voltage is supplied, on-chip RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the STBY pin low. Do not change the state of the mode pins (MD2 to MD0) while this LSI is in hardware standby mode.

Clearing Hardware Standby Mode: Hardware standby mode is cleared by means of the STBY pin and the RES pin. When the STBY pin is driven high while the RES pin is low, the reset state is set and clock oscillation is started. Ensure that the RES pin is held low until the clock oscillator stabilizes (for details on the oscillation stabilization time, refer to table 22.2). When the RES pin is subsequently driven high, a transition is made to the program execution state via the reset exception handling state.

Hardware Standby Mode Timing: Figure 22.3 shows an example of hardware standby mode timing.

When the STBY pin is driven low after the RES pin has been driven low, a transition is made to hardware standby mode. Hardware standby mode is cleared by driving the STBY pin high, waiting for the oscillation stabilization time, then changing the RES pin from low to high.

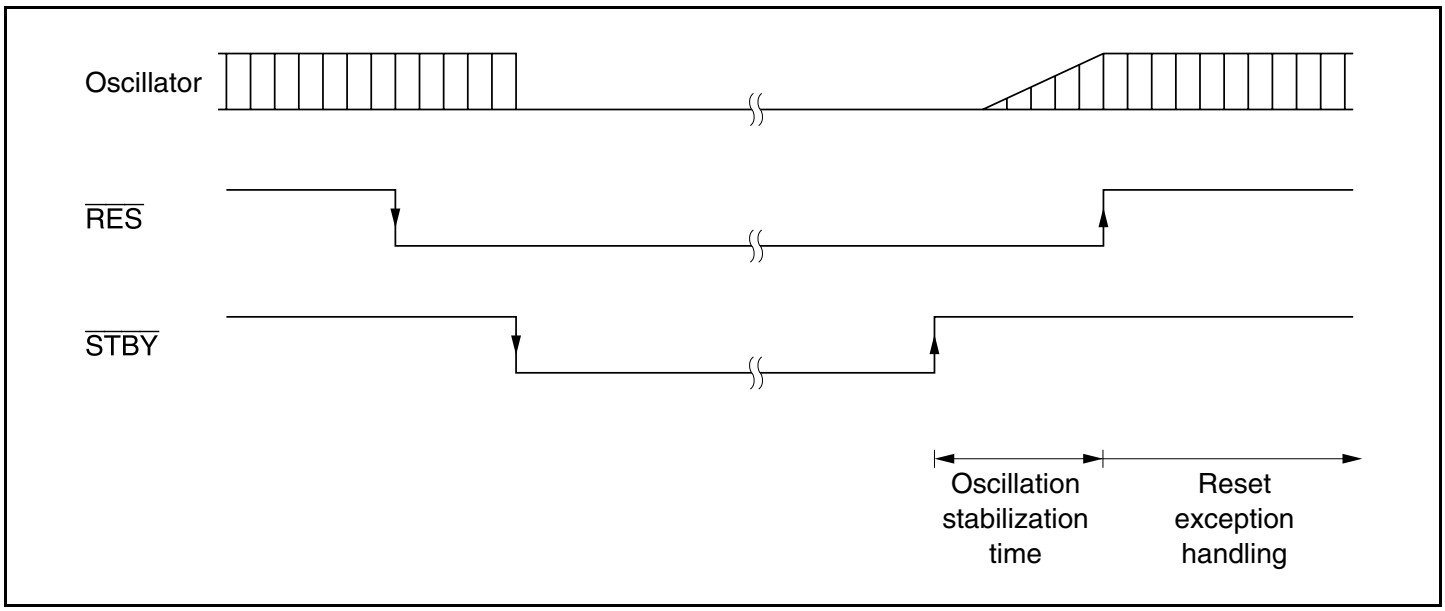


Figure 22.3 Hardware Standby Mode Timing

22.2.5 Module Stop Mode

Module stop mode can be set for individual on-chip peripheral modules.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating at the end of the bus cycle. In module stop mode, the internal states of modules other than the SCI are retained.

After reset clearance, all modules other than the EXDMAC, DMAC, and DTC are in module stop mode.

The module registers which are set in module stop mode cannot be read or written to.

22.2.6 All-Module-Clocks-Stop Mode

When the ACSE bit in MSTPCRH is set to 1 and module stop mode is set for all the on-chip peripheral functions controlled by MSTPCR (MSTPCR = H'FFFF), or for all the on-chip peripheral functions except the 8-bit timer (MSTPCR = H'FFFE), executing a SLEEP instruction while the SSBY bit in SBYCR is cleared to 0 will cause all the on-chip peripheral functions (except the 8-bit timer and watchdog timer), the bus controller, and the I/O ports to stop operating, and a transition to be made to all-module-clocks-stop mode, at the end of the bus cycle.

Operation or halting of the 8-bit timer can be selected by means of the MSTP0 bit.

All-module-clocks-stop mode is cleared by an external interrupt (NMI, IRQ0 to IRQ15 pins), RES pin input, or an internal interrupt (8-bit timer, watchdog timer), and the CPU returns to the normal program execution state via the exception handling state. All-module-clocks-stop mode is not cleared if interrupts are disabled, if interrupts other than NMI are masked by the CPU, or if the relevant interrupt is designated as a DTC activation source.

When the STBY pin is driven low, a transition is made to hardware standby mode.

22.3 ø Clock Output Control

Output of the ø clock can be controlled by means of the PSTOP bit in SCKCR, and DDR for the corresponding port. When the PSTOP bit is set to 1, the ø clock stops at the end of the bus cycle, and ø output goes high. ø clock output is enabled when the PSTOP bit is cleared to 0. When DDR for the corresponding port is cleared to 0, ø clock output is disabled and input port mode is set. Table 22.3 shows the state of the ø pin in each processing state.

Table 22.3 ø Pin State in Each Processing State

Register Setting						
DDR	**PSTOP**	**Normal operating state**	**Sleep mode**	**Software standby mode**	**Hardware standby mode**	**All-module-clocks-stop mode**
0	X	High impedance	High impedance	High impedance	High impedance	High impedance
1	0	ø output	ø output	Fixed high	High impedance	ø output
1	1	Fixed high	Fixed high	Fixed high	High impedance	Fixed high

22.4 Usage Notes

22.4.1 I/O Port Status

In software standby mode, I/O port states are retained. Therefore, there is no reduction in current dissipation for the output current when a high-level signal is output.

22.4.2 Current Dissipation during Oscillation Stabilization Standby Period

Current dissipation increases during the oscillation stabilization standby period.

22.4.3 EXDMAC/DMAC/DTC Module Stop

Depending on the operating status of the EXDMAC, DMAC, or DTC, the MSTP14 to MSTP12 bits may not be set to 1. Setting of the EXDMAC, DMAC, or DTC module stop mode should be carried out only when the respective module is not activated.

For details, refer to section 8, EXDMA Controller (EXDMAC), section 7, DMA Controller (DMAC), and section 9, Data Transfer Controller (DTC).

22.4.4 On-Chip Peripheral Module Interrupts

Relevant interrupt operations cannot be performed in module stop mode. Consequently, if module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the EXDMAC, DMAC, or DTC activation source.

Interrupts should therefore be disabled before entering module stop mode.

22.4.5 Writing to MSTPCR

MSTPCR should only be written to by the CPU.

Section 23 List of Registers

This section gives information on the on-chip I/O registers and is configured as described below.

1. Register Addresses (by functional module, in order of the corresponding section numbers)

- Descriptions by functional module, in order of the corresponding section numbers
  Entries that consist of — lines are for separation of the functional modules.
- Access to reserved addresses which are not described in this list is prohibited.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given, on the presumption of a big-endian system.

2. Register Bits

- Bit configurations of the registers are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- Reserved bits are indicated by — in the bit name.
- No entry in the bit-name column indicates that the whole register is allocated as a counter or for holding data.
- When registers consist of 16 or 32 bits, bits are described from the MSB side.
  The order in which bytes are described is on the presumption of a big-endian system.

3. Register States in Each Operating Mode

- Register states are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- For the initial state of each bit, refer to the description of the register in the corresponding section.
- The register states described are for the basic operating modes. If there is a specific reset for an on-chip module, refer to the section on that on-chip module.

23.1 Register Addresses (by functional module, in order of the corresponding section numbers)

Entries under Access size indicates numbers of bits.

Note: Access to undefined or reserved addresses is prohibited. Since operation or continued operation is not guaranteed when these registers are accessed, do not attempt such access.

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
DTC mode register A	MRA	8	H'BC00 to	DTC	16/32	2
DTC source address register	SAR	24	H'BFFF	DTC	16/32	2
DTC mode register B	MRB	8		DTC	16/32	2
DTC destination address register	DAR	24		DTC	16/32	2
DTC transfer count register A	CRA	16		DTC	16/32	2
DTC transfer count register B	CRB	18		DTC	16/32	2
Serial expansion mode register*[1]	SEMR	8	H'FDA8	SCI_2	8	2
EXDMA source address register_0	EDSAR_0	32	H'FDC0	EXDMAC_0	16	2
EXDMA destination address register_0	EDDAR_0	32	H'FDC4	EXDMAC_0	16	2
EXDMA transfer count register_0	EDTCR_0	32	H'FDC8	EXDMAC_0	16	2
EXDMA mode control register_0	EDMDR_0	16	H'FDCC	EXDMAC_0	16	2
EXDMA address control register_0	EDACR_0	16	H'FDCE	EXDMAC_0	16	2
EXDMA source address register_1	EDSAR_1	32	H'FDD0	EXDMAC_1	16	2
EXDMA destination address register_1	EDDAR_1	32	H'FDD4	EXDMAC_1	16	2
EXDMA transfer count register_1	EDTCR_1	32	H'FDD8	EXDMAC_1	16	2
EXDMA mode control register_1	EDMDR_1	16	H'FDDC	EXDMAC_1	16	2
EXDMA address control register_1	EDACR_1	16	H'FDDE	EXDMAC_1	16	2
EXDMA source address register_2	EDSAR_2	32	H'FDE0	EXDMAC_2	16	2
EXDMA destination address register_2	EDDAR_2	32	H'FDE4	EXDMAC_2	16	2
EXDMA transfer count register_2	EDTCR_2	32	H'FDE8	EXDMAC_2	16	2
EXDMA mode control register_2	EDMDR_2	16	H'FDEC	EXDMAC_2	16	2
EXDMA address control register_2	EDACR_2	16	H'FDEE	EXDMAC_2	16	2
EXDMA source address register_3	EDSAR_3	32	H'FDF0	EXDMAC_3	16	2
EXDMA destination address register_3	EDDAR_3	32	H'FDF4	EXDMAC_3	16	2
EXDMA transfer count register 3	EDTCR_3	32	H'FDF8	EXDMAC_3	16	2
EXDMA mode control register 3	EDMDR_3	16	H'FDFC	EXDMAC_3	16	2
EXDMA address control register 3	EDACR_3	16	H'FDFE	EXDMAC_3	16	2
Interrupt priority register A	IPRA	16	H'FE00	INT	16	2
Interrupt priority register B	IPRB	16	H'FE02	INT	16	2
Interrupt priority register C	IPRC	16	H'FE04	INT	16	2
Interrupt priority register D	IPRD	16	H'FE06	INT	16	2
Interrupt priority register E	IPRE	16	H'FE08	INT	16	2
Interrupt priority register F	IPRF	16	H'FE0A	INT	16	2

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
Interrupt priority register G	IPRG	16	H'FE0C	INT	16	2
Interrupt priority register H	IPRH	16	H'FE0E	INT	16	2
Interrupt priority register I	IPRI	16	H'FE10	INT	16	2
Interrupt priority register J	IPRJ	16	H'FE12	INT	16	2
Interrupt priority register K	IPRK	16	H'FE14	INT	16	2
IRQ pin select register	ITSR	16	H'FE16	INT	16	2
Software standby release IRQ enable register	SSIER	16	H'FE18	INT	16	2
IRQ sense control register H	ISCRH	16	H'FE1A	INT	16	2
IRQ sense control register L	ISCRL	16	H'FE1C	INT	16	2
IrDA control register_0	IrCR_0	8	H'FE1E	IrDA_0	8	2
Port 1 data direction register	P1DDR	8	H'FE20	PORT	8	2
Port 2 data direction register	P2DDR	8	H'FE21	PORT	8	2
Port 3 data direction register	P3DDR	8	H'FE22	PORT	8	2
Port 5 data direction register	P5DDR	8	H'FE24	PORT	8	2
Port 6 data direction register	P6DDR	8	H'FE25	PORT	8	2
Port 7 data direction register	P7DDR	8	H'FE26	PORT	8	2
Port 8 data direction register	P8DDR	8	H'FE27	PORT	8	2
Port A data direction register	PADDR	8	H'FE29	PORT	8	2
Port B data direction register	PBDDR	8	H'FE2A	PORT	8	2
Port C data direction register	PCDDR	8	H'FE2B	PORT	8	2
Port D data direction register	PDDDR	8	H'FE2C	PORT	8	2
Port E data direction register	PEDDR	8	H'FE2D	PORT	8	2
Port F data direction register	PFDDR	8	H'FE2E	PORT	8	2
Port G data direction register	PGDDR	8	H'FE2F	PORT	8	2
Port function control register 0	PFCR0	8	H'FE32	PORT	8	2
Port function control register 1	PFCR1	8	H'FE33	PORT	8	2
Port function control register 2	PFCR2	8	H'FE34	PORT	8	2
Port A pull-up MOS control register	PAPCR	8	H'FE36	PORT	8	2
Port B pull-up MOS control register	PBPCR	8	H'FE37	PORT	8	2
Port C pull-up MOS control register	PCPCR	8	H'FE38	PORT	8	2
Port D pull-up MOS control register	PDPCR	8	H'FE39	PORT	8	2

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
Port E pull-up MOS control register	PEPCR	8	H'FE3A	PORT	8	2
Port 3 open drain control register	P3ODR	8	H'FE3C	PORT	8	2
Port A open drain control register	PAODR	8	H'FE3D	PORT	8	2
Timer control register_3	TCR_3	8	H'FE80	TPU_3	16	2
Timer mode register_3	TMDR_3	8	H'FE81	TPU_3	16	2
Timer I/O control register H_3	TIORH_3	8	H'FE82	TPU_3	16	2
Timer I/O control register L_3	TIORL_3	8	H'FE83	TPU_3	16	2
Timer interrupt enable register_3	TIER_3	8	H'FE84	TPU_3	16	2
Timer status register_3	TSR_3	8	H'FE85	TPU_3	16	2
Timer counter_3	TCNT_3	16	H'FE86	TPU_3	16	2
Timer general register A_3	TGRA_3	16	H'FE88	TPU_3	16	2
Timer general register B_3	TGRB_3	16	H'FE8A	TPU_3	16	2
Timer general register C_3	TGRC_3	16	H'FE8C	TPU_3	16	2
Timer general register D_3	TGRD_3	16	H'FE8E	TPU_3	16	2
Timer control register_4	TCR_4	8	H'FE90	TPU_4	16	2
Timer mode register_4	TMDR_4	8	H'FE91	TPU_4	16	2
Timer I/O control register_4	TIOR_4	8	H'FE92	TPU_4	16	2
Timer interrupt enable register_4	TIER_4	8	H'FE94	TPU_4	16	2
Timer status register_4	TSR_4	8	H'FE95	TPU_4	16	2
Timer counter_4	TCNT_4	16	H'FE96	TPU_4	16	2
Timer general register A_4	TGRA_4	16	H'FE98	TPU_4	16	2
Timer general register B_4	TGRB_4	16	H'FE9A	TPU_4	16	2
Timer control register_5	TCR_5	8	H'FEA0	TPU_5	16	2
Timer mode register_5	TMDR_5	8	H'FEA1	TPU_5	16	2
Timer I/O control register_5	TIOR_5	8	H'FEA2	TPU_5	16	2
Timer interrupt enable register_5	TIER_5	8	H'FEA4	TPU_5	16	2
Timer status register_5	TSR_5	8	H'FEA5	TPU_5	16	2
Timer counter_5	TCNT_5	16	H'FEA6	TPU_5	16	2
Timer general register A_5	TGRA_5	16	H'FEA8	TPU_5	16	2
Timer general register B_5	TGRB_5	16	H'FEAA	TPU_5	16	2
Bus width control register	ABWCR	8	H'FEC0	BSC	16	2
Access state control register	ASTCR	8	H'FEC1	BSC	16	2
Wait control register AH	WTCRAH	8	H'FEC2	BSC	16	2

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
Wait control register AL	WTCRAL	8	H'FEC3	BSC	16	2
Wait control register BH	WTCRBH	8	H'FEC4	BSC	16	2
Wait control register BL	WTCRBL	8	H'FEC5	BSC	16	2
Read strobe timing control register	RDNCR	8	H'FEC6	BSC	16	2
Chip select assertion period control registers H	CSACRH	8	H'FEC8	BSC	16	2
Chip select assertion period control register L	CSACRL	8	H'FEC9	BSC	16	2
Burst ROM interface control register H	BROMCRH	8	H'FECA	BSC	16	2
Burst ROM interface control register L	BROMCRL	8	H'FECB	BSC	16	2
Bus control register	BCR	16	H'FECC	BSC	16	2
RAM emulation register*[3]	RAMER	8	H'FECE	FLASH	16	2
DRAM control register L	DRAMCR	16	H'FED0	BSC	16	2
DRAM access control register	DRACCR	8/16*[2]	H'FED2	BSC	16	2
Refresh control register	REFCR	16	H'FED4	BSC	16	2
Refresh timer counter	RTCNT	8	H'FED6	BSC	16	2
Refresh time constant register	RTCOR	8	H'FED7	BSC	16	2
Memory address register 0AH	MAR0AH	16	H'FEE0	DMAC	16	2
Memory address register 0AL	MAR_0AL	16	H'FEE2	DMAC	16	2
I/O address register 0A	IOAR_0A	16	H'FEE4	DMAC	16	2
Transfer count register 0A	ETCR_0A	16	H'FEE6	DMAC	16	2
Memory address register 0BH	MAR_0BH	16	H'FEE8	DMAC	16	2
Memory address register 0BL	MAR_0BL	16	H'FEEA	DMAC	16	2
I/O address register 0B	IOAR_0B	16	H'FEEC	DMAC	16	2
Transfer count register 0B	ETCR_0B	16	H'FEEE	DMAC	16	2
Memory address register 1AH	MAR_1AH	16	H'FEF0	DMAC	16	2
Memory address register 1AL	MAR_1AL	16	H'FEF2	DMAC	16	2
I/O address register 1A	IOAR_1A	16	H'FEF4	DMAC	16	2
Transfer count register 1A	ETCR_1A	16	H'FEF6	DMAC	16	2
Memory address register 1BH	MAR_1BH	16	H'FEF8	DMAC	16	2
Memory address register 1BL	MAR_1BL	16	H'FEFA	DMAC	16	2
I/O address register 1B	IOAR_1B	16	H'FEFC	DMAC	16	2
Transfer count register 1B	ETCR_1B	16	H'FEFE	DMAC	16	2

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
DMA write enable register	DMAWER	8	H'FF20	DMAC	8	2
DMA terminal control register	DMATCR	8	H'FF21	DMAC	8	2
DMA control register 0A	DMACR_0A	8	H'FF22	DMAC	16	2
DMA control register 0B	DMACR_0B	8	H'FF23	DMAC	16	2
DMA control register 1A	DMACR_1A	8	H'FF24	DMAC	16	2
DMA control register 1B	DMACR_1B	8	H'FF25	DMAC	16	2
DMA band control register H	DMABCRH	8	H'FF26	DMAC	16	2
DMA band control register L	DMABCRL	8	H'FF27	DMAC	16	2
DTC enable register A	DTCERA	8	H'FF28	DTC	16	2
DTC enable register B	DTCERB	8	H'FF29	DTC	16	2
DTC enable register C	DTCERC	8	H'FF2A	DTC	16	2
DTC enable register D	DTCERD	8	H'FF2B	DTC	16	2
DTC enable register E	DTCERE	8	H'FF2C	DTC	16	2
DTC enable register F	DTCERF	8	H'FF2D	DTC	16	2
DTC enable register G	DTCERG	8	H'FF2E	DTC	16	2
DTC vector register	DTVECR	8	H'FF30	DTC	16	2
Interrupt control register	INTCR	8	H'FF31	INT	16	2
IRQ enable register	IER	16	H'FF32	INT	16	2
IRQ status register	ISR	16	H'FF34	INT	16	2
Standby control register	SBYCR	8	H'FF3A	SYSTEM	8	2
System clock control register	SCKCR	8	H'FF3B	SYSTEM	8	2
System control register	SYSCR	8	H'FF3D	SYSTEM	8	2
Mode control register	MDCR	8	H'FF3E	SYSTEM	8	2
Module stop control register H	MSTPCRH	8	H'FF40	SYSTEM	8	2
Module stop control register L	MSTPCRL	8	H'FF41	SYSTEM	8	2
PLL control register	PLLCR	8	H'FF45	SYSTEM	8	2
PPG output control register	PCR	8	H'FF46	PPG	8	2
PPG output mode register	PMR	8	H'FF47	PPG	8	2
Next data enable register H	NDERH	8	H'FF48	PPG	8	2
Next data enable register L	NDERL	8	H'FF49	PPG	8	2
Output data register H	PODRH	8	H'FF4A	PPG	8	2
Output data register L	PODRL	8	H'FF4B	PPG	8	2
Next data register H*[4]	NDRH	8	H'FF4C	PPG	8	2

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
Next data register L*[4]	NDRL	8	H'FF4D	PPG	8	2
Next data register H*[4]	NDRH	8	H'FF4E	PPG	8	2
Next data register L*[4]	NDRL	8	H'FF4F	PPG	8	2
Port 1 register	PORT1	8	H'FF50	PORT	8	2
Port 2 register	PORT2	8	H'FF51	PORT	8	2
Port 3 register	PORT3	8	H'FF52	PORT	8	2
Port 4 register	PORT4	8	H'FF53	PORT	8	2
Port 5 register	PORT5	8	H'FF54	PORT	8	2
Port 6 register	PORT6	8	H'FF55	PORT	8	2
Port 7 register	PORT7	8	H'FF56	PORT	8	2
Port 8 register	PORT8	8	H'FF57	PORT	8	2
Port A register	PORTA	8	H'FF59	PORT	8	2
Port B register	PORTB	8	H'FF5A	PORT	8	2
Port C register	PORTC	8	H'FF5B	PORT	8	2
Port D register	PORTD	8	H'FF5C	PORT	8	2
Port E register	PORTE	8	H'FF5D	PORT	8	2
Port F register	PORTF	8	H'FF5E	PORT	8	2
Port G register	PORTG	8	H'FF5F	PORT	8	2
Port 1 data register	P1DR	8	H'FF60	PORT	8	2
Port 2 data register	P2DR	8	H'FF61	PORT	8	2
Port 3 data register	P3DR	8	H'FF62	PORT	8	2
Port 5 data register	P5DR	8	H'FF64	PORT	8	2
Port 6 data register	P6DR	8	H'FF65	PORT	8	2
Port 7 data register	P7DR	8	H'FF66	PORT	8	2
Port 8 data register	P8DR	8	H'FF67	PORT	8	2
Port A data register	PADR	8	H'FF69	PORT	8	2
Port B data register	PBDR	8	H'FF6A	PORT	8	2
Port C data register	PCDR	8	H'FF6B	PORT	8	2
Port D data register	PDDR	8	H'FF6C	PORT	8	2
Port E data register	PEDR	8	H'FF6D	PORT	8	2
Port F data register	PFDR	8	H'FF6E	PORT	8	2
Port G data register	PGDR	8	H'FF6F	PORT	8	2

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
Port H register	PORTH	8	H'FF70	PORT	8	2
Port H data register	PHDR	8	H'FF72	PORT	8	2
Port H data direction register	PHDDR	8	H'FF74	PORT	8	2
Serial mode register_0	SMR_0	8	H'FF78	SCI_0	8	2
Bit rate register_0	BRR_0	8	H'FF79	SCI_0	8	2
Serial control register_0	SCR_0	8	H'FF7A	SCI_0	8	2
Transmit data register_0	TDR_0	8	H'FF7B	SCI_0	8	2
Serial status register_0	SSR_0	8	H'FF7C	SCI_0	8	2
Receive data register_0	RDR_0	8	H'FF7D	SCI_0	8	2
Smart card mode register_0	SCMR_0	8	H'FF7E	SCI_0	8	2
Serial mode register_1	SMR_1	8	H'FF80	SCI_1	8	2
Bit rate register_1	BRR_1	8	H'FF81	SCI_1	8	2
Serial control register_1	SCR_1	8	H'FF82	SCI_1	8	2
Transmit data register_1	TDR_1	8	H'FF83	SCI_1	8	2
Serial status register_1	SSR_1	8	H'FF84	SCI_1	8	2
Receive data register_1	RDR_1	8	H'FF85	SCI_1	8	2
Smart card mode register_1	SCMR_1	8	H'FF86	SCI_1	8	2
Serial mode register_2	SMR_2	8	H'FF88	SCI_2	8	2
Bit rate register_2	BRR_2	8	H'FF89	SCI_2	8	2
Serial control register_2	SCR_2	8	H'FF8A	SCI_2	8	2
Transmit data register_2	TDR_2	8	H'FF8B	SCI_2	8	2
Serial status register_2	SSR_2	8	H'FF8C	SCI_2	8	2
Receive data register_2	RDR_2	8	H'FF8D	SCI_2	8	2
Smart card mode register_2	SCMR_2	8	H'FF8E	SCI_2	8	2
A/D data register A (H8S/2678R Series)	ADDRA	16	H'FF90	A/D	16	2
A/D data register AH (H8S/2678 Series)	ADDRAH	8	H'FF90	A/D	8	2
A/D data register AL (H8S/2678 Series)	ADDRAL	8	H'FF91	A/D	8	2
A/D data register B (H8S/2678R Series)	ADDRB	16	H'FF92	A/D	16	2
A/D data register BH (H8S/2678 Series)	ADDRBH	8	H'FF92	A/D	8	2

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
A/D data register BL (H8S/2678 Series)	ADDRBL	8	H'FF93	A/D	8	2
A/D data register C (H8S/2678R Series)	ADDRC	16	H'FF94	A/D	16	2
A/D data register CH (H8S/2678 Series)	ADDRCH	8	H'FF94	A/D	8	2
A/D data register CL (H8S/2678 Series)	ADDRCL	8	H'FF95	A/D	8	2
A/D data register D (H8S/2678R Series)	ADDRD	16	H'FF96	A/D	16	2
A/D data register DH (H8S/2678 Series)	ADDRDH	8	H'FF96	A/D	8	2
A/D data register DL (H8S/2678 Series)	ADDRDL	8	H'FF97	A/D	8	2
A/D control/status register (H8S/2678 Series)	ADCSR	8	H'FF98	A/D	8	2
A/D data register E (H8S/2678R Series)	ADDRE	16	H'FF98	A/D	16	2
A/D control register (H8S/2678 Series)	ADCR	8	H'FF99	A/D	8	2
A/D data register F (H8S/2678R Series)	ADDRF	16	H'FF9A	A/D	16	2
A/D data register G (H8S/2678R Series)	ADDRG	16	H'FF9C	A/D	16	2
A/D data register H (H8S/2678R Series)	ADDRH	16	H'FF9E	A/D	16	2
A/D control/status register (H8S/2678R Series)	ADCSR	8	H'FFA0	A/D	16	2
A/D control register (H8S/2678R Series)	ADCR	8	H'FFA1	A/D	16	2
D/A data register 0	DADR0	8	H'FFA4	D/A	8	2
D/A data register 1	DADR1	8	H'FFA5	D/A	8	2
D/A control register 01	DACR01	8	H'FFA6	D/A	8	2
D/A data register 2	DADR2	8	H'FFA8	D/A	8	2
D/A data register 3	DADR3	8	H'FFA9	D/A	8	2
D/A control register 23	DACR23	8	H'FFAA	D/A	8	2
Timer control register 0	TCR_0	8	H'FFB0	TMR_0	16	2
Timer control register 1	TCR_1	8	H'FFB1	TMR_1	16	2

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
Timer control/status register 0	TCSR_0	8	H'FFB2	TMR_0	16	2
Timer control/status register 1	TCSR_1	8	H'FFB3	TMR_1	16	2
Time constant register A0	TCORA_0	8	H'FFB4	TMR_0	16	2
Time constant register A1	TCORA_1	8	H'FFB5	TMR_1	16	2
Time constant register B0	TCORB_0	8	H'FFB6	TMR_0	16	2
Time constant register B1	TCORB_1	8	H'FFB7	TMR_1	16	2
Timer counter 0	TCNT_0	8	H'FFB8	TMR_0	16	2
Timer counter 1	TCNT_1	8	H'FFB9	TMR_1	16	2
Timer control/status register	TCSR	8	H'FFBC*[4] (Write) H'FFBC (Read)	WDT	16	2
Timer counter	TCNT	8	H'FFBC*[4] (Write) H'FFBD (Read)	WDT	16	2
Reset control/status register	RSTCSR	8	H'FFBE*[4] (Write) H'FFBF (Read)	WDT	16	2
Timer start register	TSTR	8	H'FFC0	TPU	16	2
Timer synchronous register	TSYR	8	H'FFC1	TPU	16	2
Flash memory control register 1*[3]	FLMCR1	8	H'FFC8	FLASH	8	2
Flash memory control register 2*[3]	FLMCR2	8	H'FFC9	FLASH	8	2
Erase block register 1*[3]	EBR1	8	H'FFCA	FLASH	8	2
Erase block register 2*[3]	EBR2	8	H'FFCB	FLASH	8	2
Timer control register_0	TCR_0	8	H'FFD0	TPU_0	16	2
Timer mode register_0	TMDR_0	8	H'FFD1	TPU_0	16	2
Timer I/O control register H_0	TIORH_0	8	H'FFD2	TPU_0	16	2
Timer I/O control register L_0	TIORL_0	8	H'FFD3	TPU_0	16	2
Timer interrupt enable register_0	TIER_0	8	H'FFD4	TPU_0	16	2
Timer status register_0	TSR_0	8	H'FFD5	TPU_0	16	2
Timer counter_0	TCNT_0	16	H'FFD6	TPU_0	16	2
Timer general register A_0	TGRA_0	16	H'FFD8	TPU_0	16	2
Timer general register B_0	TGRB_0	16	H'FFDA	TPU_0	16	2

Register Name	Abbrevia-tion	Bit No.	Address	Module	Data Width	Access States
Timer general register C_0	TGRC_0	16	H'FFDC	TPU_0	16	2
Timer general register D_0	TGRD_0	16	H'FFDE	TPU_0	16	2
Timer control register_1	TCR_1	8	H'FFE0	TPU_1	16	2
Timer mode register_1	TMDR_1	8	H'FFE1	TPU_1	16	2
Timer I/O control register_1	TIOR_1	8	H'FFE2	TPU_1	16	2
Timer interrupt enable register_1	TIER_1	8	H'FFE4	TPU_1	16	2
Timer status register_1	TSR_1	8	H'FFE5	TPU_1	16	2
Timer counter_1	TCNT_1	16	H'FFE6	TPU_1	16	2
Timer general register A_1	TGRA_1	16	H'FFE8	TPU_1	16	2
Timer general register B_1	TGRB_1	16	H'FFEA	TPU_1	16	2
Timer control register_2	TCR_2	8	H'FFF0	TPU_2	16	2
Timer mode register_2	TMDR_2	8	H'FFF1	TPU_2	16	2
Timer I/O control register_2	TIOR_2	8	H'FFF2	TPU_2	16	2
Timer interrupt enable register_2	TIER_2	8	H'FFF4	TPU_2	16	2
Timer status rgister_2	TSR_2	8	H'FFF5	TPU_2	16	2
Timer counter_2	TCNT_2	16	H'FFF6	TPU_2	16	2
Timer general register A_2	TGRA_2	16	H'FFF8	TPU_2	16	2
Timer general register B_2	TGRB_2	16	H'FFFA	TPU_2	16	2

Notes: 1. Not available in the H8S/2678 Series.

2. In the H8S/2678 Series: 8 bits, in the H8S/2678R Series: 16 bits.
3. Register of the flash memory version. Not available in the masked ROM version and ROM-less version.
4. If the pulse output group 2 and pulse output group 3 output triggers are the same according to the PCR setting, the NDRH address will be H'FF4C, and if different, the address of NDRH for group 2 will be H'FF4E, and that for group 3 will be H'FF4C. Similarly, if the pulse output group 0 and pulse output group 1 output triggers are the same according to the PCR setting, the NDRL address will be H'FF4D, and if different, the address of NDRL for group 0 will be H'FF4F, and that for group 1 will be H'FF4D.
5. For writing, refer to section 14.6.1, Notes on register access.

23.2 Register Bits

Register addresses and bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, and 16-bit and 32-bit registers are shown as 2 or 4 lines, respectively.

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MRA	AM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz	DTC*[9]
SAR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
MRB	CHNE	DISEL	CHNS	—	—	—	—	—	
DAR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CRA	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CRB	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
SEMR*[8]	—	—	—	—	ABCS	ACS2	ACS1	ACS0	SCI_2 Smart card interface 2
EDSAR_0	—	—	—	—	—	—	—	—	EXDMAC_0
EDDAR_0	—	—	—	—	—	—	—	—	
EDTCR_0	—	—	—	—	—	—	—	—	
EDMDR_0	EDA	BEF	EDRAKE	ETENDE	EDREQS	AMS	MDS1	MDS0	
	EDIE	IRF	TCEIE	SDIR	DTSIZE	BGUP	—	—	
EDACR_0	SAT1	SAT0	SARIE	SARA4	SARA3	SARA2	SARA1	SARA0	
	DAT1	DAT0	DARIE	DARA4	DARA3	DARA2	DARA1	DARA0	
EDSAR_1	—	—	—	—	—	—	—	—	EXDMAC_1
EDDAR_1	—	—	—	—	—	—	—	—	
EDTCR_1	—	—	—	—	—	—	—	—	
EDMDR_1	EDA	BEF	EDRAKE	ETENDE	EDREQS	AMS	MDS1	MDS0	
	EDIE	IRF	TCEIE	SDIR	DTSIZE	BGUP	—	—	
EDACR_1	SAT1	SAT0	SARIE	SARA4	SARA3	SARA2	SARA1	SARA0	
	DAT1	DAT0	DARIE	DARA4	DARA3	DARA2	DARA1	DARA0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
EDSAR_2	—	—	—	—	—	—	—	—	EXDMAC_2
EDDAR_2	—	—	—	—	—	—	—	—	
EDTCR_2	—	—	—	—	—	—	—	—	
EDMDR_2	EDA	BEF	EDRAKE	ETENDE	EDREQS	AMS	MDS1	MDS0	
	EDIE	IRF	TCEIE	SDIR	DTSIZE	BGUP	—	—	
EDACR_2	SAT1	SAT0	SARIE	SARA4	SARA3	SARA2	SARA1	SARA0	
	DAT1	DAT0	DARIE	DARA4	DARA3	DARA2	DARA1	DARA0	
EDSAR_3	—	—	—	—	—	—	—	—	EXDMAC_3
EDDAR_3	—	—	—	—	—	—	—	—	
EDTCR_3	—	—	—	—	—	—	—	—	
EDMDR_3	EDA	BEF	EDRAKE	ETENDE	EDREQS	AMS	MDS1	MDS0	
	EDIE	IRF	TCEIE	SDIR	DTSIZE	BGUP	—	—	
EDACR_3	SAT1	SAT0	SARIE	SARA4	SARA3	SARA2	SARA1	SARA0	
	DAT1	DAT0	DARIE	DARA4	DARA3	DARA2	DARA1	DARA0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
IPRA	—	IPRA14	IPRA13	IPRA12	—	IPRA10	IPRA9	IPRA8	INT
	—	IPRA6	IPRA5	IPRA4	—	IPRA2	IPRA1	IPRA0	
IPRB	—	IPRB14	IPRB13	IPRB12	—	IPRB10	IPRB9	IPRB8	
	—	IPRB6	IPRB5	IPRB4	—	IPRB2	IPRB1	IPRB0	
IPRC	—	IPRC14	IPRC13	IPRC12	—	IPRC10	IPRC9	IPRC8	
	—	IPRC6	IPRC5	IPRC4	—	IPRC2	IPRC1	IPRC0	
IPRD	—	IPRD14	IPRD13	IPRD12	—	IPRD10	IPRD9	IPRD8	
	—	IPRD6	IPRD5	IPRD4	—	IPRD2	IPRD1	IPRD0	
IPRE	—	IPRE14	IPRE13	IPRE12	—	IPRE10	IPRE9	IPRE8	
	—	IPRE6	IPRE5	IPRE4	—	IPRE2	IPRE1	IPRE0	
IPRF	—	IPRF14	IPRF13	IPRF12	—	IPRF10	IPRF9	IPRF8	
	—	IPRF6	IPRF5	IPRF4	—	IPRF2	IPRF1	IPRF0	
IPRG	—	IPRG14	IPRG13	IPRG12	—	IPRG10	IPRG9	IPRG8	
	—	IPRG6	IPRG5	IPRG4	—	IPRG2	IPRG1	IPRG0	
IPRH	—	IPRH14	IPRH13	IPRH12	—	IPRH10	IPRH9	IPRH8	
	—	IPRH6	IPRH5	IPRH4	—	IPRH2	IPRH1	IPRH0	
IPRI	—	IPRI14	IPRI13	IPRI12	—	IPRI10	IPRI9	IPRI8	
	—	IPRI6	IPRI5	IPRI4	—	IPRI2	IPRI1	IPRI0	
IPRJ	—	IPRJ14	IPRJ13	IPRJ12	—	IPRJ10	IPRJ9	IPRJ8	
	—	IPRJ6	IPRJ5	IPRJ4	—	IPRJ2	IPRJ1	IPRJ0	
IPRK	—	IPRK14	IPRK13	IPRK12	—	IPRK10	IPRK9	IPRK8	
	—	IPRK6	IPRK5	IPRK4	—	IPRK2	IPRK1	IPRK0	
ITSR	ITS15	ITS14	ITS13	ITS12	ITS11	ITS10	ITS9	ITS8	
	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	ITS0	
SSIER	SSI15	SSI14	SSI13	SSI12	SSI11	SSI10	SSI9	SSI8	
	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	SSI0	
ISCRH	IRQ15SCB	IRQ15SCA	IRQ14SCB	IRQ14SCA	IRQ13SCB	IRQ13SCA	IRQ12SCB	IRQ12SCA	
	IRQ11SCB	IRQ11SCA	IRQ10SCB	IRQ10SCA	IRQ9SCB	IRQ9SCA	IRQ8SCB	IRQ8SCA	
ISCRL	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	
	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA	
IrCR_0	IrE	IrCKS2	IrCKS1	IrCKS0	—	—	—	—	IrDA_0

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	PORT
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR	
P3DDR	—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR	
P5DDR	—	—	—	—	P53DDR	P52DDR	P51DDR	P50DDR	
P6DDR	—	—	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR	
P7DDR	—	—	P75DDR	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR	
P8DDR	—	—	P85DDR	P84DDR	P83DDR	P82DDR	P81DDR	P80DDR	
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR	
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR	
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR	
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR	
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR	
PGDDR	—	PG6DDR	PG5DDR	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR	
PFCR0	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E	
PFCR1	A23E	A22E	A21E	A20E	A19E	A18E	A17E	A16E	
PFCR2	—	—	—	—	ASOE	LWROE	OES	DMACS	
PAPCR	PA7PCR	PA6PCR	PA5PCR	PA4PCR	PA3PCR	PA2PCR	PA1PCR	PA0PCR	
PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR	
PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR	
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR	
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR	
P3ODR	—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR	
PAODR	PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	PA0ODR	
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_3
TMDR_3	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_3	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_3	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TGRA_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	TPU_3
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRC_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRD_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_4
TMDR_4	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_5
TMDR_5	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	BSC
ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	
WTCRAH	—	W72	W71	W70	—	W62	W61	W60	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
WTCRAL	—	W52	W51	W50	—	W42	W41	W40	BSC
WTCRBH	—	W32	W31	W30	—	W22	W21	W20	
WTCRBL	—	W12	W11	W10	—	W02	W01	W00	
RDNCR	RDN7	RDN6	RDN5	RDN4	RDN3	RDN2	RDN1	RDN0	
CSACRH	CSXH7	CSXH6	CSXH5	CSXH4	CSXH3	CSXH2	CSXH1	CSXH0	
CSACRL	CSXT7	CSXT6	CSXT5	CSXT4	CSXT3	CSXT2	CSXT1	CSXT0	
BROMCRH	BSRM0	BSTS02	BSTS01	BSTS00	—	—	BSWD01	BSWD00	
BROMCRL	BSRM1	BSTS12	BSTS11	BSTS10	—	—	BSWD11	BSWD10	
BCR	BRLE	BREQ0E	—	IDLC	ICIS1	ICIS0	WDBE	WAITE	
	—	—	—	—	—	ICIS2*[8]	—	—	
RAMER*[7]	—	—	—	—	RAMS	RAM2	RAM1	RAM0	FLASH
DRAMCR	0EE	RAST	—	CAST	—	RMTS2	RMTS1	RMTS0	BSC
	BE	RCDM	DDS	EDDS	—	MXC2	MXC1	MXC0	
DRACCR*[1]	DRMI	—	TPC1	TPC0	SDWCD*[8]	—	RCD1	RCD0	
	—	—	—	—	CKSPE*[8]	—	RDXC1*[8]	RDXC0*[8]	
REFCR	CMF	CMIE	RCW1	RCW0	—	RTCK2	RTCK1	RTCK0	
	RFSHE	CBRM	RLW1	RLW0	SLFRF	TPCS2	TPCS1	TPCS0	
RTCNT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
RTCOR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_0AH	—	—	—	—	—	—	—	—	DMAC
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_0AL	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
IOAR_0A	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
ETCR_0A	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_0BH	—	—	—	—	—	—	—	—	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_0BL	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
IOAR_0B	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ETCR_0B	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	DMAC
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_1AH	—	—	—	—	—	—	—	—	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_1AL	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
IOAR_1A	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
ETCR_1A	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_1BH	—	—	—	—	—	—	—	—	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_1BL	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
IOAR_1B	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
ETCR_1B	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DMAWER	—	—	—	—	WE1B	WE1A	WE0B	WE0A	
DMATCR	—	—	TEE1	TEE0	—	—	—	—	
DMACR_0A*[11]	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR_0A*[12]	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—	
DMACR_0B*[11]	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR_0B*[12]	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	
DMACR_1A*[11]	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR_1A*[12]	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—	
DMACR_1B*[11]	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR_1B*[12]	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	
DMABCRH*[11]	FAE1	FAE0	SAE1	SAE0	DTA1B	DTA1A	DTA0B	DTA0A	
DMABCRH*[12]	FAE1	FAE0	—	—	DTA1	—	DTA0	—	
DMABCRL*[11]	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A	
DMABCRL*[12]	DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B	DTIE0A	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC*[10]
DTCERB	DTCEB7	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0	
DTCERC	—	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0	
DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0	
DTCERE	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0	
DTCERF	DTCEF7	DTCEF6	DTCEF5	DTCEF4	DTCEF3	DTCEF2	DTCEF1	DTCEF0	
DTCERG	DTCEG7	DTCEG6	—	—	—	—	—	—	
DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
INTCR	—	—	INTM1	INTM0	NMIEG	—	—	—	INT
IER	IRQ15E	IRQ14E	IRQ13E	IRQ12E	IRQ11E	IRQ10E	IRQ9E	IRQ8E	
	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
ISR	IRQ15F	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	IRQ8F	
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
SBYCR	SSBY	OPE	—	—	STS3	STS2	STS1	STS0	SYSTEM
SCKCR	PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0	
SYSCR	—	—	MACS	—	FLSHE	—	EXPE	RAME	
MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	
MSTPCRH	ACSE	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	
MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0	
PLLCR	—	—	—	—	—	—	STC1	STC0	
PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	PPG
PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV	
NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8	
NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0	
PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8	
PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0	
NDRH	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8	
NDRL	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0	
NDRH	—	—	—	—	NDR11	NDR10	NDR9	NDR8	
NDRL	—	—	—	—	NDR3	NDR2	NDR1	NDR0	
PORT1	P17	P16	P15	P14	P13	P12	P11	P10	PORT
PORT2	P27	P26	P25	P24	P23	P22	P21	P20	
PORT3	—	—	P35	P34	P33	P32	P31	P30	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PORT4	P47	P46	P45	P44	P43	P42	P41	P40	PORT
PORT5	P57	P56	P55	P54	P53	P52	P51	P50	
PORT6	—	—	P65	P64	P63	P62	P61	P60	
PORT7	—	—	P75	P74	P73	P72	P71	P70	
PORT8	—	—	P85	P84	P83	P82	P81	P80	
PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	
PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	
PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	
PORTG	—	PG6	PG5	PG4	PG3	PG2	PG1	PG0	
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	
P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR	
P3DR	—	—	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR	
P5DR	—	—	—	—	P53DR	P52DR	P51DR	P50DR	
P6DR	—	—	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR	
P7DR	—	—	P75DR	P74DR	P73DR	P72DR	P71DR	P70DR	
P8DR	—	—	P85DR	P84DR	P83DR	P82DR	P81RD	P80DR	
PADR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
PGDR	—	PG6DR	PG5DR	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR	
PORTH	—	—	—	—	PH3	PH2	PH1	PH0	
PHDR	—	—	—	—	PH3DR	PH2DR	PH1DR	PH0DR	
PHDDR	—	—	—	—	PH3DDR	PH2DDR	PH1DDR	PH0DDR	
SMR_0	C/A/ GM*[2]	CHR/ BLK*[3]	PE	O/E	STOP/ BCP1*[4]	MP/ BCP0*[5]	CKS1	CKS0	SCI_0, Smart card interface 0
BRR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SCI_0, Smart card interface 0
SSR_0	TDRE	RDRF	ORER	FER/ ERS*[6]	PER	TEND	MPB	MPBT	
RDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_1	C/A/ GM*[2]	CHR/ BLK*[3]	PE	O/E	STOP/ BCP1*[4]	MP/ BCP0*[5]	CKS1	CKS0	SCI_1, Smart card interface 1
BRR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_1	TDRE	RDRF	ORER	FER/ ERS*[6]	PER	TEND	MPB	MPBT	
RDR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_2	C/A/ GM*[2]	CHR/ BLK*[3]	PE	O/E	STOP/ BCP1*[4]	MP/ BCP0*[5]	CKS1	CKS0	SCI_2, Smart card interface 2
BRR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_2									
SSR_2	TDRE	RDRF	ORER	FER/ ERS*[6]	PER	TEND	MPB	MPBT	
RDR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF	
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
	AD1	AD0	—	—	—	—	—	—	
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRE*[8]	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRF*[8]	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ADDRG*[8]	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
	AD1	AD0	—	—	—	—	—	—	
ADDRH*[8]	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN*[9]/ —*[8]	CKS*[9]/ CH3*[8]	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	—*[9]/ SCANE*[8]	—*[9]/ SCANS*[8]	CKS1	CH3*[9]/ CKS0*[8]	—	—	
DADR0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	D/A
DADR1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DACR01	DAOE1	DAOE0	DAE	—	—	—	—	—	
DADR2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DADR3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DACR23	DAOE3	DAOE2	DAE	—	—	—	—	—	
TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_0
TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_1
TCSR_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	
TCSR_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
TCORA_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCORA_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCORB_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCORB_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCNT_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCNT_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCSR	OVF	WT/IT	TME	—	—	CKS2	CKS1	CKS0	WDT
TCNT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
RSTCSR	WOVF	RSTE	—	—	—	—	—	—	
TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU
TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	
FLMCR1*[7]	FWE	SWE	ESU	PSU	EV	PV	E	P	FLASH
FLMCR2*[7]	FLER	—	—	—	—	—	—	—	
EBR1*[7]	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
EBR2*[7]	—	—	EB13	EB12	EB11	EB10	EB9	EB8	
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_0

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	TPU_0
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRC_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRD_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_1
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_2
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TGRA_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	TPU_2
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Notes: 1. In the H8S/2678 Series: 8 bits, in the H8S/2678R Series: 16 bits.
2. Functions as C/A for SCI use, and as GM for smart card interface use.
3. Functions as CHR for SCI use, and as BLK for smart card interface use.
4. Functions as STOP for SCI use, and as BCP1 for smart card interface use.
5. Functions as MP for SCI use, and as BCP0 for smart card interface use.
6. Functions as FER for SCI use, and as ERS for smart card interface use.
7. Register of the flash memory version. Not available in the masked ROM version and ROM-less version.
8. Not available in the H8S/2678 Series.
9. Not available in the H8S/2678R Series.
10. Loaded in on-chip RAM. The bus width is 32 bits when the DTC accesses this area as register information, and 16 bits otherwise.
11. For short address mode
12. For full address mode

23.3 Register States in Each Operating Mode

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
MRA	Initialized	—	—	—	—	—	—	Initialized	DTC
SAR	Initialized	—	—	—	—	—	—	Initialized	
MRB	Initialized	—	—	—	—	—	—	Initialized	
DAR	Initialized	—	—	—	—	—	—	Initialized	
CRA	Initialized	—	—	—	—	—	—	Initialized	
CRB	Initialized	—	—	—	—	—	—	Initialized	
SEMR*[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	SCI2
EDSAR_0	Initialized	—	—	—	—	—	—	Initialized	EXDMA_C
EDDAR_0	Initialized	—	—	—	—	—	—	Initialized	
EDTCR_0	Initialized	—	—	—	—	—	—	Initialized	
EDMDR_0	Initialized	—	—	—	—	—	—	Initialized	
EDACR_0	Initialized	—	—	—	—	—	—	Initialized	
EDSAR_1	Initialized	—	—	—	—	—	—	Initialized	EXDMA_1
EDDAR_1	Initialized	—	—	—	—	—	—	Initialized	
EDTCR_1	Initialized	—	—	—	—	—	—	Initialized	
EDMDR_1	Initialized	—	—	—	—	—	—	Initialized	
EDACR_1	Initialized	—	—	—	—	—	—	Initialized	
EDSAR_2	Initialized	—	—	—	—	—	—	Initialized	EXDMA_2
EDDAR_2	Initialized	—	—	—	—	—	—	Initialized	
EDTCR_2	Initialized	—	—	—	—	—	—	Initialized	
EDMDR_2	Initialized	—	—	—	—	—	—	Initialized	
EDACR_2	Initialized	—	—	—	—	—	—	Initialized	
EDSAR_3	Initialized	—	—	—	—	—	—	Initialized	EXDMA_3
EDDAR_3	Initialized	—	—	—	—	—	—	Initialized	
EDTCR_3	Initialized	—	—	—	—	—	—	Initialized	
EDMDR_3	Initialized	—	—	—	—	—	—	Initialized	
EDACR_3	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
IPRA	Initialized	—	—	—	—	—	—	Initialized	INT
IPRB	Initialized	—	—	—	—	—	—	Initialized	
IPRC	Initialized	—	—	—	—	—	—	Initialized	
IPRD	Initialized	—	—	—	—	—	—	Initialized	
IPRE	Initialized	—	—	—	—	—	—	Initialized	
IPRF	Initialized	—	—	—	—	—	—	Initialized	
IPRG	Initialized	—	—	—	—	—	—	Initialized	
IPRH	Initialized	—	—	—	—	—	—	Initialized	
IPRI	Initialized	—	—	—	—	—	—	Initialized	
IPRJ	Initialized	—	—	—	—	—	—	Initialized	
IPRK	Initialized	—	—	—	—	—	—	Initialized	
ITSR	Initialized	—	—	—	—	—	—	Initialized	
SSIER	Initialized	—	—	—	—	—	—	Initialized	
ISCRH	Initialized	—	—	—	—	—	—	Initialized	
ISCRL	Initialized	—	—	—	—	—	—	Initialized	
IrCR_0	Initialized	—	—	—	—	—	—	Initialized	IrDA_0

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
P1DDR	—	—	—	—	—	—	—	Initialized	PORT
P2DDR	—	—	—	—	—	—	—	Initialized	
P3DDR	—	—	—	—	—	—	—	Initialized	
P5DDR	—	—	—	—	—	—	—	Initialized	
P6DDR	—	—	—	—	—	—	—	Initialized	
P7DDR	—	—	—	—	—	—	—	Initialized	
P8DDR	—	—	—	—	—	—	—	Initialized	
PADDR	—	—	—	—	—	—	—	Initialized	
PBDDR	—	—	—	—	—	—	—	Initialized	
PCDDR	—	—	—	—	—	—	—	Initialized	
PDDDR	—	—	—	—	—	—	—	Initialized	
PEDDR	—	—	—	—	—	—	—	Initialized	
PFDDR	—	—	—	—	—	—	—	Initialized	
PGDDR	—	—	—	—	—	—	—	Initialized	
PFCR0	—	—	—	—	—	—	—	Initialized	
PFCR1	—	—	—	—	—	—	—	Initialized	
PFCR2	—	—	—	—	—	—	—	Initialized	
PAPCR	—	—	—	—	—	—	—	Initialized	
PBPCR	—	—	—	—	—	—	—	Initialized	
PCPCR	—	—	—	—	—	—	—	Initialized	
PDPCR	—	—	—	—	—	—	—	Initialized	
PEPCR	—	—	—	—	—	—	—	Initialized	
P3ODR	—	—	—	—	—	—	—	Initialized	
PAODR	—	—	—	—	—	—	—	Initialized	
TCR_3	Initialized	—	—	—	—	—	—	Initialized	TPU_3
TMDR_3	Initialized	—	—	—	—	—	—	Initialized	
TIORH_3	Initialized	—	—	—	—	—	—	Initialized	
TIORL_3	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
TIER_3	Initialized	—	—	—	—	—	—	Initialized	TPU_3
TSR_3	Initialized	—	—	—	—	—	—	Initialized	
TCNT_3	Initialized	—	—	—	—	—	—	Initialized	
TGRA_3	Initialized	—	—	—	—	—	—	Initialized	
TGRB_3	Initialized	—	—	—	—	—	—	Initialized	
TGRC_3	Initialized	—	—	—	—	—	—	Initialized	
TGRD_3	Initialized	—	—	—	—	—	—	Initialized	
TCR_4	Initialized	—	—	—	—	—	—	Initialized	TPU_4
TMDR_4	Initialized	—	—	—	—	—	—	Initialized	
TIOR_4	Initialized	—	—	—	—	—	—	Initialized	
TIER_4	Initialized	—	—	—	—	—	—	Initialized	
TSR_4	Initialized	—	—	—	—	—	—	Initialized	
TCNT_4	Initialized	—	—	—	—	—	—	Initialized	
TGRA_4	Initialized	—	—	—	—	—	—	Initialized	
TGRB_4	Initialized	—	—	—	—	—	—	Initialized	
TCR_5	Initialized	—	—	—	—	—	—	Initialized	TPU_5
TMDR_5	Initialized	—	—	—	—	—	—	Initialized	
TIOR_5	Initialized	—	—	—	—	—	—	Initialized	
TIER_5	Initialized	—	—	—	—	—	—	Initialized	
TSR_5	Initialized	—	—	—	—	—	—	Initialized	
TCNT_5	Initialized	—	—	—	—	—	—	Initialized	
TGRA_5	Initialized	—	—	—	—	—	—	Initialized	
TGRB_5	Initialized	—	—	—	—	—	—	Initialized	
ABWCR	Initialized	—	—	—	—	—	—	Initialized	BSC
ASTCR	Initialized	—	—	—	—	—	—	Initialized	
WTCRAH	Initialized	—	—	—	—	—	—	Initialized	
WTCRAL	Initialized	—	—	—	—	—	—	Initialized	
WTCRBH	Initialized	—	—	—	—	—	—	Initialized	
WTCRBL	Initialized	—	—	—	—	—	—	Initialized	
RDNCR	Initialized	—	—	—	—	—	—	Initialized	
CSACRH	Initialized	—	—	—	—	—	—	Initialized	
CSACRL	Initialized	—	—	—	—	—	—	Initialized	
BROMCRH	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
BROMCRL	Initialized	—	—	—	—	—	—	Initialized	BSC
BCR	Initialized	—	—	—	—	—	—	Initialized	
RAMER*[2]	Initialized	—	—	—	—	—	—	Initialized	FLASH
DRAMCR	Initialized	—	—	—	—	—	—	Initialized	BSC
DRACCR	Initialized	—	—	—	—	—	—	Initialized	
REFCR	Initialized	—	—	—	—	—	—	Initialized	
RTCNT	Initialized	—	—	—	—	—	—	Initialized	
RTCOR	Initialized	—	—	—	—	—	—	Initialized	
MAR_0AH	Initialized	—	—	—	—	—	—	Initialized	
MAR_0AL	Initialized	—	—	—	—	—	—	Initialized	
IOAR_0A	Initialized	—	—	—	—	—	—	Initialized	
ETCR_0A	Initialized	—	—	—	—	—	—	Initialized	
MAR_0BH	Initialized	—	—	—	—	—	—	Initialized	
MAR_0BL	Initialized	—	—	—	—	—	—	Initialized	
IOAR_0B	Initialized	—	—	—	—	—	—	Initialized	
ETCR_0B	Initialized	—	—	—	—	—	—	Initialized	
MAR_1AH	Initialized	—	—	—	—	—	—	Initialized	
MAR_1AL	Initialized	—	—	—	—	—	—	Initialized	
IOAR_1A	Initialized	—	—	—	—	—	—	Initialized	
ETCR_1A	Initialized	—	—	—	—	—	—	Initialized	DMAC
MAR_1BH	Initialized	—	—	—	—	—	—	Initialized	
MAR_1BL	Initialized	—	—	—	—	—	—	Initialized	
IOAR_1B	Initialized	—	—	—	—	—	—	Initialized	
ETCR_1B	Initialized	—	—	—	—	—	—	Initialized	
DMAWER	Initialized	—	—	—	—	—	—	Initialized	
DMATCR	Initialized	—	—	—	—	—	—	Initialized	
DMACR_0A	Initialized	—	—	—	—	—	—	Initialized	
DMACR_0B	Initialized	—	—	—	—	—	—	Initialized	
DMACR_1A	Initialized	—	—	—	—	—	—	Initialized	
DMACR_1B	Initialized	—	—	—	—	—	—	Initialized	
DMABCRH	Initialized	—	—	—	—	—	—	Initialized	
DMABCRL	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
DTCERA	Initialized	—	—	—	—	—	—	Initialized	DTC
DTCERB	Initialized	—	—	—	—	—	—	Initialized	
DTCERC	Initialized	—	—	—	—	—	—	Initialized	
DTCERD	Initialized	—	—	—	—	—	—	Initialized	
DTCERE	Initialized	—	—	—	—	—	—	Initialized	
DTCERF	Initialized	—	—	—	—	—	—	Initialized	
DTCERG	Initialized	—	—	—	—	—	—	Initialized	
DTVECR	Initialized	—	—	—	—	—	—	Initialized	
INTCR	Initialized	—	—	—	—	—	—	Initialized	INT
IER	Initialized	—	—	—	—	—	—	Initialized	
ISR	Initialized	—	—	—	—	—	—	Initialized	
SBYCR	Initialized	—	—	—	—	—	—	Initialized	SYSTEM
SCKCR	Initialized	—	—	—	—	—	—	Initialized	
SYSCR	Initialized	—	—	—	—	—	—	Initialized	
MDCR	Initialized	—	—	—	—	—	—	Initialized	
MSTPCRH	Initialized	—	—	—	—	—	—	Initialized	
MSTPCRL	Initialized	—	—	—	—	—	—	Initialized	
PLLCR	Initialized	—	—	—	—	—	—	Initialized	
PCR	Initialized	—	—	—	—	—	—	Initialized	PPG
PMR	Initialized	—	—	—	—	—	—	Initialized	
NDERH	Initialized	—	—	—	—	—	—	Initialized	
NDERL	Initialized	—	—	—	—	—	—	Initialized	
PODRH	Initialized	—	—	—	—	—	—	Initialized	
PODRL	Initialized	—	—	—	—	—	—	Initialized	
NDRH	Initialized	—	—	—	—	—	—	Initialized	
NDRL	Initialized	—	—	—	—	—	—	Initialized	
NDRH	Initialized	—	—	—	—	—	—	Initialized	
NDRL	Initialized	—	—	—	—	—	—	Initialized	
PORT1	—	—	—	—	—	—	—	—	PORT
PORT2	—	—	—	—	—	—	—	—	
PORT3	—	—	—	—	—	—	—	—	
PORT4	—	—	—	—	—	—	—	—	
PORT5	—	—	—	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
PORT6	—	—	—	—	—	—	—	—	PORT
PORT7	—	—	—	—	—	—	—	—	
PORT8	—	—	—	—	—	—	—	—	
PORTA	—	—	—	—	—	—	—	—	
PORTB	—	—	—	—	—	—	—	—	
PORTC	—	—	—	—	—	—	—	—	
PORTD	—	—	—	—	—	—	—	—	
PORTE	—	—	—	—	—	—	—	—	
PORTF	—	—	—	—	—	—	—	—	
PORTG	—	—	—	—	—	—	—	—	
P1DR	Initialized	—	—	—	—	—	Initialized	—	
P2DR	Initialized	—	—	—	—	—	Initialized	—	
P3DR	Initialized	—	—	—	—	—	Initialized	—	
P5DR	Initialized	—	—	—	—	—	Initialized	—	
P6DR	Initialized	—	—	—	—	—	Initialized	—	
P7DR	Initialized	—	—	—	—	—	Initialized	—	
P8DR	Initialized	—	—	—	—	—	Initialized	—	
PADR	Initialized	—	—	—	—	—	Initialized	—	
PBDR	Initialized	—	—	—	—	—	Initialized	—	
PCDR	Initialized	—	—	—	—	—	Initialized	—	
PDDR	Initialized	—	—	—	—	—	Initialized	—	
PEDR	Initialized	—	—	—	—	—	Initialized	—	
PFDR	Initialized	—	—	—	—	—	Initialized	—	
PGDR	Initialized	—	—	—	—	—	Initialized	—	
PORTH	Initialized	—	—	—	—	—	—	—	
PHDR	Initialized	—	—	—	—	—	Initialized	—	
PHDDR	Initialized	—	—	—	—	—	Initialized	—	
SMR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	SCI_0
BRR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
TDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
SCMR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	SCI_0
SMR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	SCI_1
BRR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
TDR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RDR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCMR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SMR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	SCI_2
BRR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
TDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCMR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRA	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	A/D
ADDRB	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRC	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRD	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRE*[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRF*[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRG*[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRH*[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADCSR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
DADR0	Initialized	—	—	—	—	—	—	Initialized	D/A
DADR1	Initialized	—	—	—	—	—	—	Initialized	
DACR01	Initialized	—	—	—	—	—	—	Initialized	
DADR2	Initialized	—	—	—	—	—	—	Initialized	
DADR3	Initialized	—	—	—	—	—	—	Initialized	
DACR23	Initialized	—	—	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	—	—	Initialized	TMR_0
TCR_1	Initialized	—	—	—	—	—	—	Initialized	TMR_1

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
TCSR_0	Initialized	—	—	—	—	—	—	Initialized	TMR_0
TCSR_1	Initialized	—	—	—	—	—	—	Initialized	TMR_1
TCORA_0	Initialized	—	—	—	—	—	—	Initialized	
TCORA_1	Initialized	—	—	—	—	—	—	Initialized	
TCORB_0	Initialized	—	—	—	—	—	—	Initialized	
TCORB_1	Initialized	—	—	—	—	—	—	Initialized	
TCNT_0	Initialized	—	—	—	—	—	—	Initialized	
TCNT_1	Initialized	—	—	—	—	—	—	Initialized	
TCSR	Initialized	—	—	—	—	—	—	Initialized	WDT
TCNT	Initialized	—	—	—	—	—	—	Initialized	
RSTCSR	Initialized	—	—	—	—	—	—	—	
TSTR	Initialized	—	—	—	—	—	—	Initialized	TPU
TSYR	Initialized	—	—	—	—	—	—	Initialized	
FLMCR1*[2]	Initialized	—	—	—	—	—	—	Initialized	FLASH
FLMCR2*[2]	Initialized	—	—	—	—	—	—	Initialized	
EBR1*[2]	Initialized	—	—	—	—	—	—	Initialized	
EBR2*[2]	Initialized	—	—	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	—	—	Initialized	TPU_0
TMDR_0	Initialized	—	—	—	—	—	—	Initialized	
TIORH_0	Initialized	—	—	—	—	—	—	Initialized	
TIORL_0	Initialized	—	—	—	—	—	—	Initialized	
TIER_0	Initialized	—	—	—	—	—	—	Initialized	
TSR_0	Initialized	—	—	—	—	—	—	Initialized	
TCNT_0	Initialized	—	—	—	—	—	—	Initialized	
TGRA_0	Initialized	—	—	—	—	—	—	Initialized	
TGRB_0	Initialized	—	—	—	—	—	—	Initialized	
TGRC_0	Initialized	—	—	—	—	—	—	Initialized	
TGRD_0	Initialized	—	—	—	—	—	—	Initialized	
TCR_1	Initialized	—	—	—	—	—	—	Initialized	TPU_1
TMDR_1	Initialized	—	—	—	—	—	—	Initialized	
TIOR_1	Initialized	—	—	—	—	—	—	Initialized	
TIER_1	Initialized	—	—	—	—	—	—	Initialized	
TSR_1	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
TCNT_1	Initialized	—	—	—	—	—	—	Initialized	TPU_1
TGRA_1	Initialized	—	—	—	—	—	—	Initialized	
TGRB_1	Initialized	—	—	—	—	—	—	Initialized	
TCR_2	Initialized	—	—	—	—	—	—	Initialized	TPU_2
TMDR_2	Initialized	—	—	—	—	—	—	Initialized	
TIOR_2	Initialized	—	—	—	—	—	—	Initialized	
TIER_2	Initialized	—	—	—	—	—	—	Initialized	
TSR_2	Initialized	—	—	—	—	—	—	Initialized	
TCNT_2	Initialized	—	—	—	—	—	—	Initialized	
TGRA_2	Initialized	—	—	—	—	—	—	Initialized	
TGRB_2	Initialized	—	—	—	—	—	—	Initialized	

Notes: 1. Not available in the H8S/2678 Series.

2. Register of the flash memory version. Not available in the masked ROM version and ROM-less version.

Section 24 Electrical Characteristics

24.1 Absolute Maximum Ratings

Table 24.1 lists the absolute maximum ratings.

Table 24.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC} $PLLV_{CC}$	–0.3 to + 4.6*	V
Input voltage (except port 4, P54 to P57)	V_{in}	–0.3 to V_{CC} + 0.3	V
Input voltage (port 4, P54 to P57)	V_{in}	–0.3 to AV_{CC} + 0.3	V
Reference power supply voltage	V_{ref}	–0.3 to AV_{CC} + 0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to + 4.6*	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} + 0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to + 75*	°C
		Wide-range specifications: –40 to + 85	°C
Storage temperature	T_{stg}	–55 to + 125*	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

Note: F-ZTAT version:

Ranges of power supply voltage and analog power supply voltage:

– 0.3 to 4.0 V

Ranges of operating temperature when flash memory is programmed/erased:

Regular specifications: 0 to +75°C

Wide-range specifications: 0 to +85°C

24.2 DC Characteristics

Table 24.2 DC Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V*[1], T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	Port 1, port 2, P50 to P53*[2], port 6*[2], port 8*[2], PF1*[2], PF2*[2], PH2*[2], PH3*[2]	VT^-	$V_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.07$	—	—	V	
	P54 to P57*[2]	VT^-	$AV_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$AV_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$AV_{CC} \times 0.07$	—	—	V	
Input high voltage	STBY, MD2 to MD0, DCTL*[4]	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	RES, NMI		$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Port 3, P50 to P53*[3], ports 6 to 8*[3], ports A to H*[3]		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Port 4, P54 to P57*[3]		$AV_{CC} \times 0.7$	—	AV_{CC} + 0.3	V	
Input low voltage	RES, STBY, MD2 to MD0, DCTL*[4]	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL		–0.3	—	$V_{CC} \times 0.2$	V	
	Ports 3 to 8, ports A to H*[3]		–0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 μA
			V_{CC} – 1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input leakage current	RES	$\|I_{in}\|$	—	—	10.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
	STBY, NMI, MD2 to MD0, DCTL*[4]		—	—	1.0	µA	
	Port 4, P54 to P57		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} – 0.5 V

Notes: 1. If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

2. When used as IRQ0 to IRQ15.
3. When used as other than IRQ0 to IRQ15.
4. Not supported in the H8S/2678 Series.

Table 24.3 DC Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V*[1], T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Three-state leakage current (off state)	Ports 1 to 3, P50 to P53, ports 6 to 8, ports A to H	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
Input pull-up MOS current	Ports A to E	$-I_p$	10	—	300	µA	V_{CC} = 2.7 to 3.6 V V_{in} = 0 V
Input capacitance	RES	C_{in}	—	—	30	pF	V_{in} = 0 V
	NMI		—	—	30	pF	f = 1 MHz
	All input pins except RES and NMI		—	—	15	pF	T_a = 25°C
Current consamption *[2]	Normal operation	I_{CC}*[4]	—	80 (3.3 V)	150	mA	f = 33 MHz
	Sleep mode		—	70 (3.3 V)	125	mA	f = 33 MHz
	Standby mode*[3]		—	0.01	10	µA	$T_a \leq$ 50°C
			—	—	80	µA	50°C < T_a
	All module clocks stopped*[5]		—	50 (3.3 V)	125	µA	
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	0.2 (3.0 V)	2.0	mA	
	Idle		—	0.01	5.0	µA	
Reference power supply current	During A/D and D/A conversion	AI_{CC}	—	1.4 (3.0 V)	4.0	mA	
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

2. Current dissipation values are for V_{IH}min = V_{CC} – 0.5 V and V_{IL}max = 0.5 V with all output pins unloaded and all input pull-up MOSs in the off state.

3. The values are for $V_{RAM} \leq V_{CC}$ < 3.0 V, V_{IH}min = $V_{CC} \times 0.9$, and V_{IL}max = 0.3 V.

4. I_{CC} depends on V_{CC} and f as follows:

$I_{CC}max = 1.0\ (mA) + 1.2\ (mA/(MHz \times V)) \times V_{CC} \times f$ (normal operation)

$I_{CC}max = 1.0\ (mA) + 1.0\ (mA/(MHz \times V)) \times V_{CC} \times f$ (sleep mode)

5. The values are for reference.

Table 24.4 Permissible Output Currents

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
$V_{SS} = AV_{SS}$ = 0 V*, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	All output pins	I_{OL}	—	—	2.0	mA
Permissible output low current (total)	Total of all output pins	ΣI_{OL}	—	—	80	mA
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	40	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in table 24.3.

Note: If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

24.3 AC Characteristics

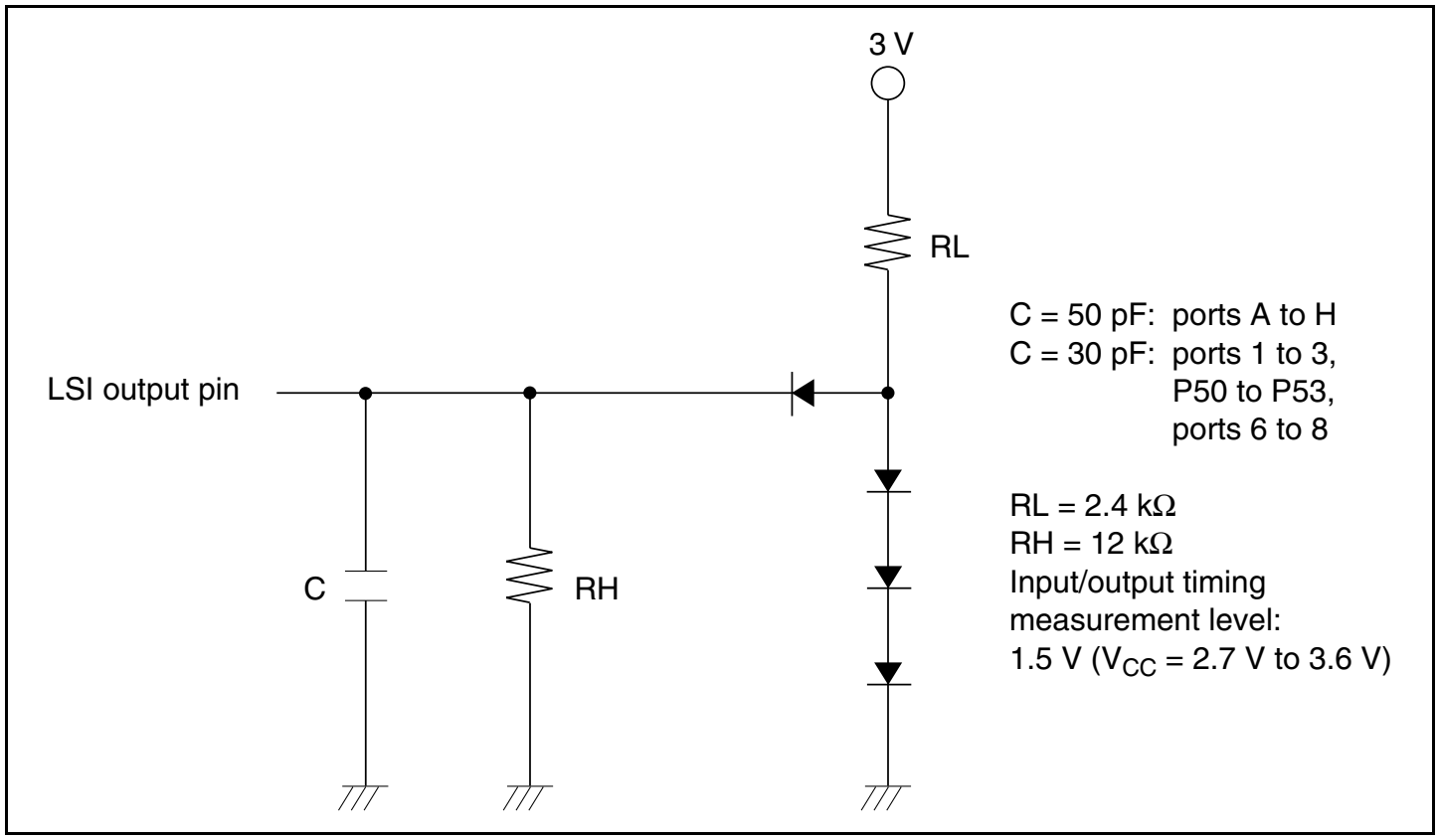


Figure 24.1 Output Load Circuit

Clock Timing

Table 24.5 Clock Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ø = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
Clock cycle time	t_{cyc}	30.3	500	ns	Figure 24.2
Clock pulse high width	t_{CH}	10	—	ns	Figure 24.2
Clock pulse low width	t_{CL}	10	—	ns	
Clock rise time	t_{Cr}	—	5	ns	
Clock fall time	t_{Cf}	—	5	ns	
Reset oscillation stabilization time (crystal)	t_{OSC1}	10	—	ms	Figure 24.4 (1)
Software standby oscillation stabilization time (crystal)	t_{OSC2}	10	—	ms	Figure 24.4 (2)
External clock output delay stabilization time	t_{DEXT}	500	—t		

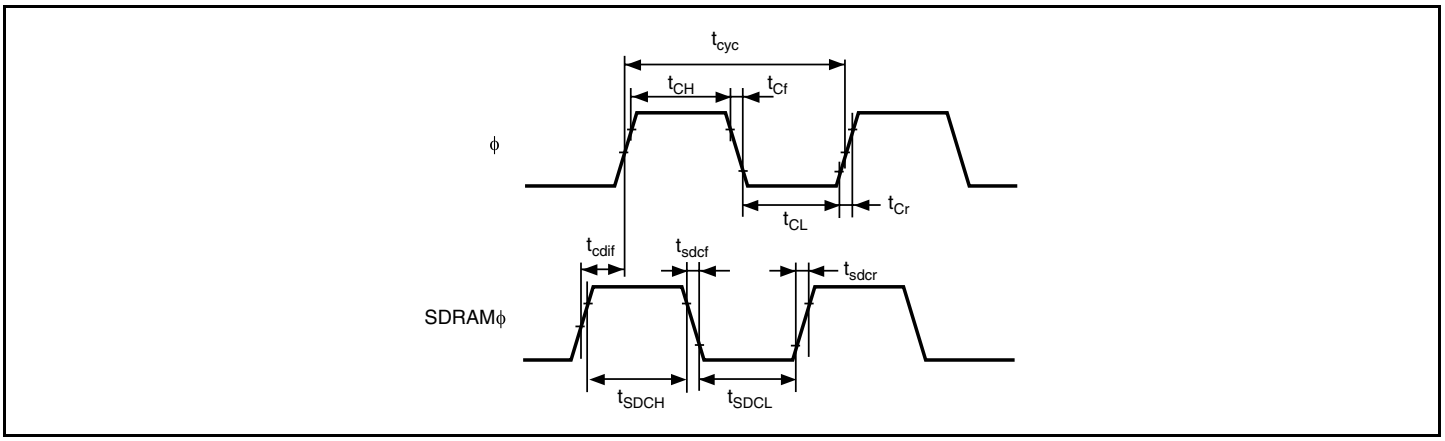


Figure 24.3 SDRAMϕ Timing*

Note: Not supported in the H8S/2678 Series.

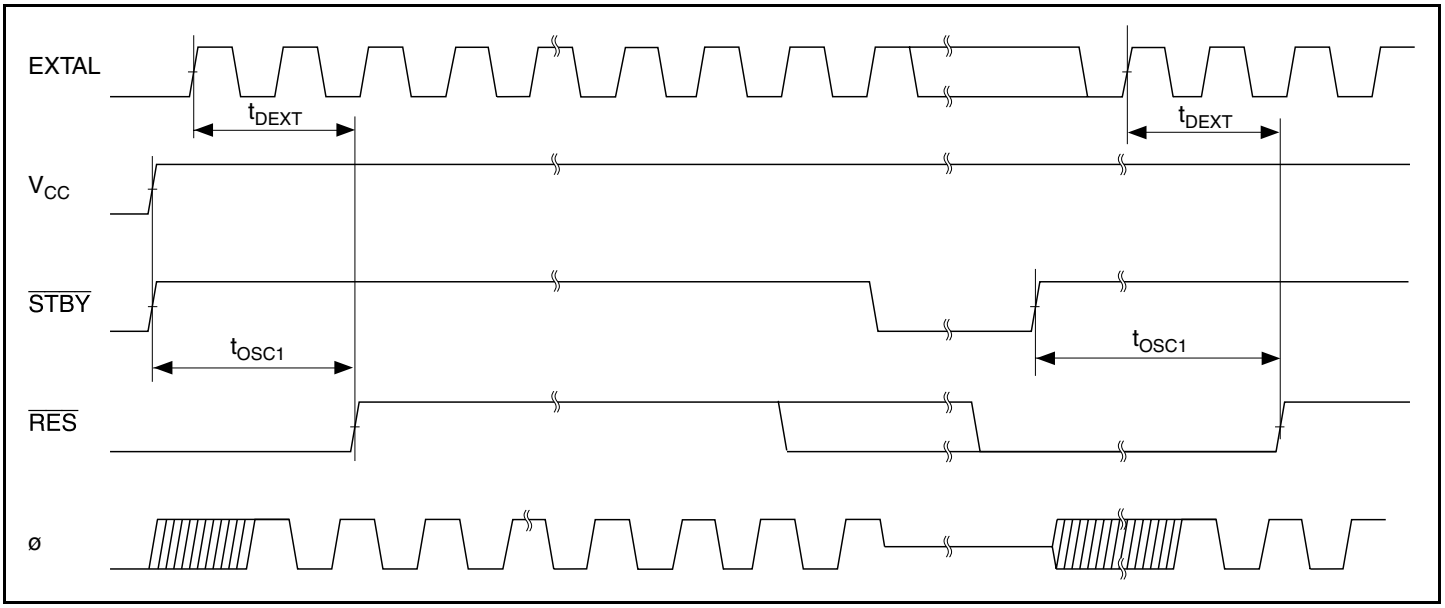


Figure 24.4 (1) Oscillation Stabilization Timing

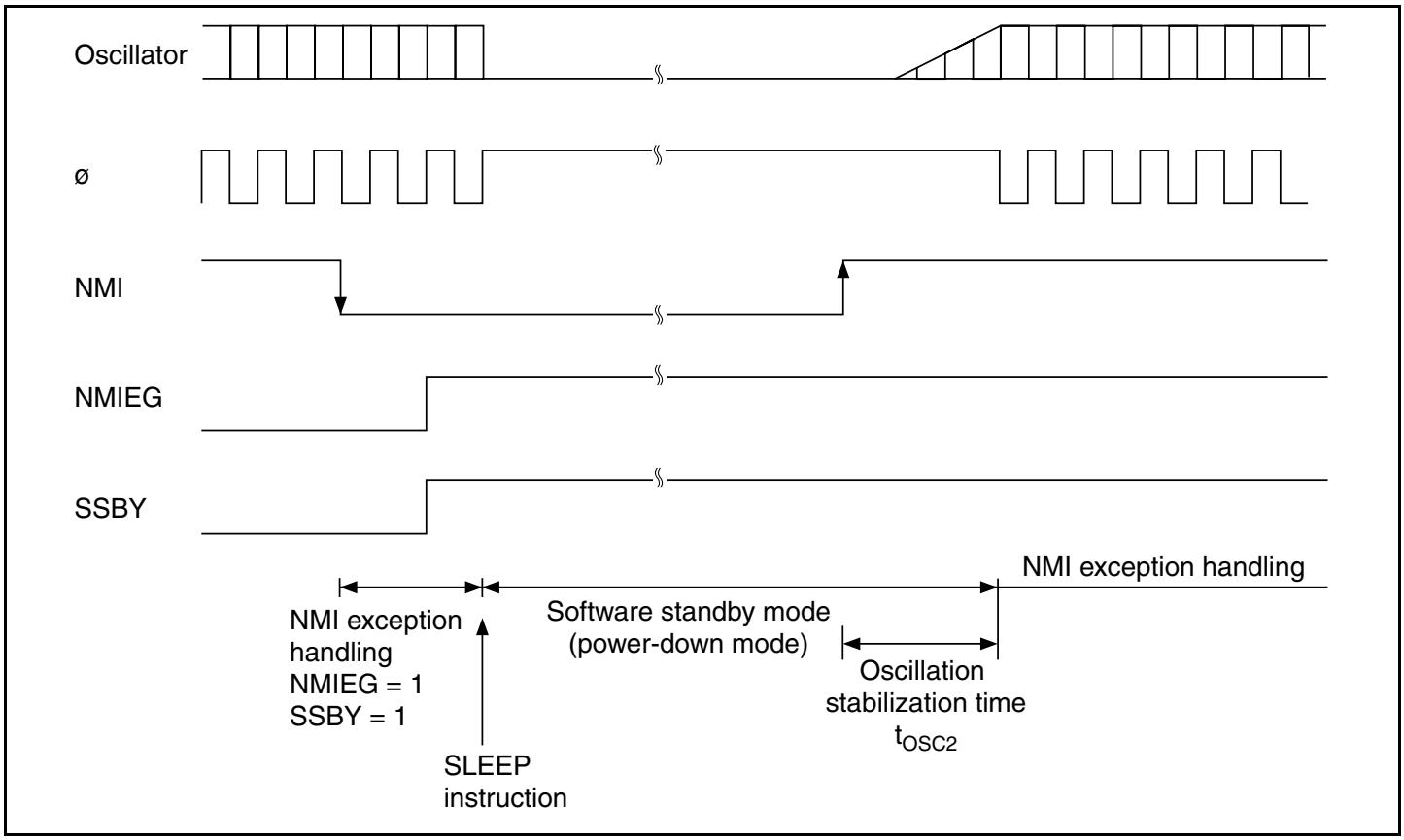


Figure 24.4 (2) Oscillation Stabilization Timing

Control Signal Timing

Table 24.6 Control Signal Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ø = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
RES setup time	t_{RESS}	200	—	ns	Figure 24.5
RES pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 24.6
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (in recovery from software standby mode)	t_{NMIW}	200	—		
IRQ setup time	t_{IRQS}	150	—	ns	
IRQ hold time	t_{IRQH}	10	—		
IRQ pulse width (in recovery from software standby mode)	t_{IRQW}	200	—		

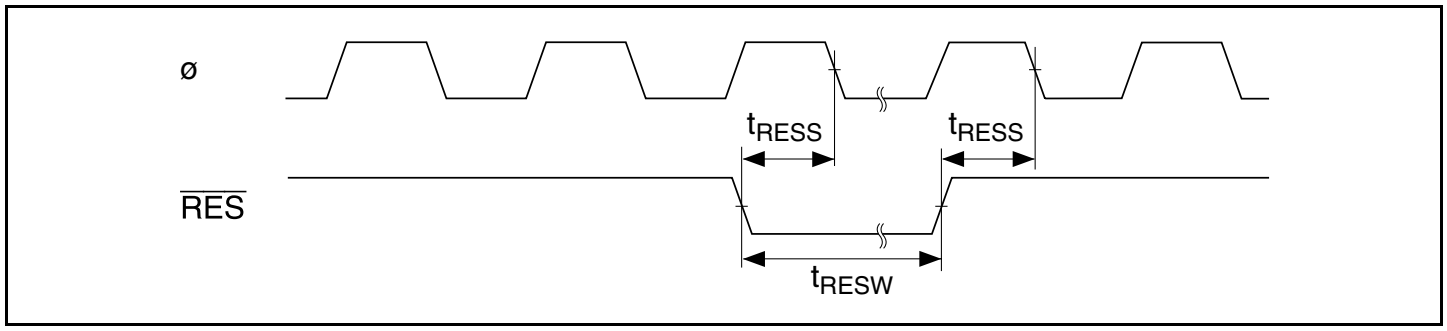
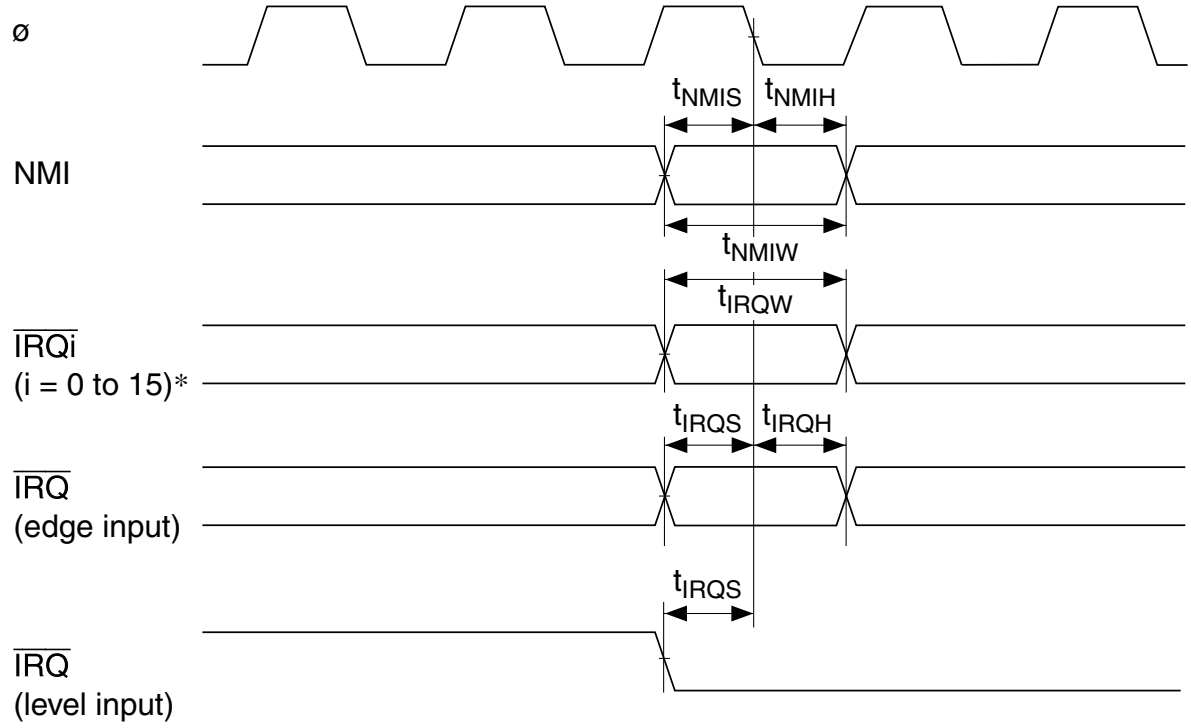


Figure 24.5 Reset Input Timing



Note: * Necessary for SSIER setting to clear software standby mode.

Figure 24.6 Interrupt Input Timing

Bus Timing

Table 24.7 Bus Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
Address delay time	t_{AD}	—	20	ns	Figures 24.7 to 24.21
Address setup time 1	t_{AS1}	$0.5 \times t_{cyc} - 13$	—	ns	
Address setup time 2	t_{AS2}	$1.0 \times t_{cyc} - 13$	—	ns	
Address setup time 3	t_{AS3}	$1.5 \times t_{cyc} - 13$	—	ns	
Address setup time 4	t_{AS4}	$2.0 \times t_{cyc} - 13$	—	ns	
Address hold time 1	t_{AH1}	$0.5 \times t_{cyc} - 8$	—	ns	
Address hold time 2	t_{AH2}	$1.0 \times t_{cyc} - 8$	—	ns	
Address hold time 3	t_{AH3}	$1.5 \times t_{cyc} - 8$	—	ns	
CS delay time 1	t_{CSD1}	—	15	ns	
CS delay time 2	t_{CSD2}	—	15	ns	
CS delay time 3	t_{CSD3}	—	20	ns	
AS delay time	t_{ASD}	—	15	ns	
RD delay time 1	t_{RSD1}	—	15	ns	
RD delay time 2	t_{RSD2}	—	15	ns	
Read data setup time 1	t_{RDS1}	15	—	ns	
Read data setup time 2	t_{RDS2}	15	—	ns	
Read data hold time 1	t_{RDH1}	0	—	ns	
Read data hold time 2	t_{RDH2}	0	—	ns	
Read data access time 1	t_{AC1}	—	$1.0 \times t_{cyc} - 20$	ns	
Read data access time 2	t_{AC2}	—	$1.5 \times t_{cyc} - 20$	ns	
Read data access time 3	t_{AC3}	—	$2.0 \times t_{cyc} - 20$	ns	
Read data access time 4	t_{AC4}	—	$2.5 \times t_{cyc} - 20$	ns	
Read data access time 5	t_{AC5}	—	$1.0 \times t_{cyc} - 20$	ns	
Read data access time 6	t_{AC6}	—	$2.0 \times t_{cyc} - 20$	ns	
Read data access time 7	t_{AC7}	—	$4.0 \times t_{cyc} - 20$	ns	
Read data access time 8	t_{AC8}	—	$3.0 \times t_{cyc} - 20$	ns	
Address read data access time 1	t_{AA1}	—	$1.0 \times t_{cyc} - 20$	ns	
Address read data access time 2	t_{AA2}	—	$1.5 \times t_{cyc} - 20$	ns	
Address read data access time 3	t_{AA3}	—	$2.0 \times t_{cyc} - 20$	ns	
Address read data access time 4	t_{AA4}	—	$2.5 \times t_{cyc} - 20$	ns	
Address read data access time 5	t_{AA5}	—	$3.0 \times t_{cyc} - 20$	ns	

Table 24.8 Bus Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
WR delay time 1	t_{WRD1}	—	15	ns	Figures 24.7 to 24.21
WR delay time 2	t_{WRD2}	—	15	ns	
WR pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 13$	—	ns	
WR pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 13$	—	ns	
Write data delay time	t_{WDD}	—	20	ns	
Write data setup time 1	t_{WDS1}	$0.5 \times t_{cyc} - 13$	—	ns	
Write data setup time 2	t_{WDS2}	$1.0 \times t_{cyc} - 13$	—	ns	
Write data setup time 3	t_{WDS3}	$1.5 \times t_{cyc} - 13$	—	ns	
Write data hold time 1	t_{WDH1}	$0.5 \times t_{cyc} - 8$	—	ns	
Write data hold time 2	t_{WDH2}	$1.0 \times t_{cyc} - 8$	—	ns	
Write data hold time 3	t_{WDH3}	$1.5 \times t_{cyc} - 8$	—	ns	
Write command setup time 1	t_{WCS1}	$0.5 \times t_{cyc} - 10$	—	ns	
Write command setup time 2	t_{WCS2}	$1.0 \times t_{cyc} - 10$	—	ns	
Write command hold time 1	t_{WCH1}	$0.5 \times t_{cyc} - 10$	—	ns	
Write command hold time 2	t_{WCH2}	$1.0 \times t_{cyc} - 10$	—	ns	
Read command setup time 1	t_{RCS1}	$1.5 \times t_{cyc} - 10$	—	ns	
Read command setup time 2	t_{RCS2}	$2.0 \times t_{cyc} - 10$	—	ns	
Read command hold time	t_{RCH}	$0.5 \times t_{cyc} - 10$	—	ns	
CAS delay time 1	t_{CASD1}	—	15	ns	
CAS delay time 2	t_{CASD2}	—	15	ns	
CAS setup time 1	t_{CSR1}	$0.5 \times t_{cyc} - 10$	—	ns	
CAS setup time 2	t_{CSR2}	$1.5 \times t_{cyc} - 10$	—	ns	
CAS pulse width 1	t_{CASW1}	$1.0 \times t_{cyc} - 20$	—	ns	
CAS pulse width 2	t_{CASW2}	$1.5 \times t_{cyc} - 20$	—	ns	
CAS precharge time 1	t_{CPW1}	$1.0 \times t_{cyc} - 20$	—	ns	
CAS precharge time 2	t_{CPW2}	$1.5 \times t_{cyc} - 20$	—	ns	
OE delay time 1	t_{OED1}	—	15	ns	
OE delay time 2	t_{OED2}	—	15	ns	
Precharge time 1	t_{PCH1}	$1.0 \times t_{cyc} - 20$	—	ns	
Precharge time 2	t_{PCH2}	$1.5 \times t_{cyc} - 20$	—	ns	

Item	Symbol	Min	Max	Unit	Test Conditions
Self-refresh precharge time 1	t_{RPS1}	$2.5 \times t_{cyc} - 20$	—	ns	Figure 24.22 Figure 24.23
Self-refresh precharge time 2	t_{RPS2}	$3.0 \times t_{cyc} - 20$	—	ns	
WAIT setup time	t_{WTS}	25	—	ns	Figure 24.15
WAIT hold time	t_{WTH}	5	—	ns	
BREQ setup time	t_{BREQS}	30	—	ns	Figure 24.24
BACK delay time	t_{BACD}	—	15	ns	
Bus floating time	t_{BZD}	—	40	ns	
BREQO delay time	t_{BRQOD}	—	25	ns	Figure 24.25
Address delay time 2*	t_{AD2}	—	16.5	ns	Figure 24.26
CS delay time 4*	t_{CSD4}	—	16.5	ns	Figure 24.26
DQM delay time*	t_{DQMD}	—	16.5	ns	Figure 24.26
CKE delay time*	t_{CKED}	—	16.5	ns	Figure 24.27
Read data setup time 3*	t_{RDS3}	15	—	ns	Figure 24.26
Read data hold time 3*	t_{RDH3}	0	—	ns	Figure 24.26
Write data delay time 2*	t_{WDD2}	—	31.5	ns	Figure 24.26
Write data hold time 4*	t_{WDH3}	2	—	ns	Figure 24.26

Note: Not supported in the H8S/2678 Series.

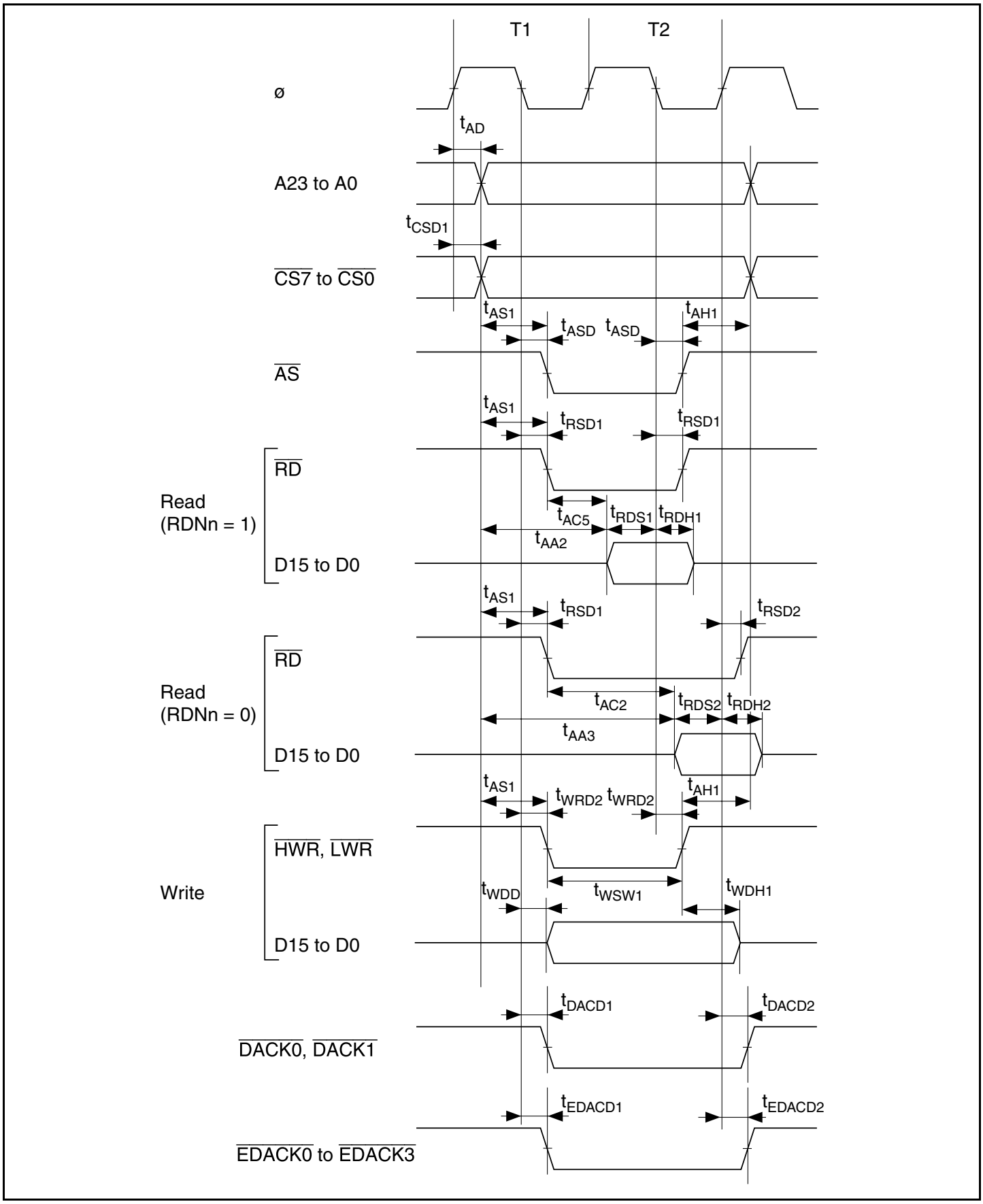


Figure 24.7 Basic Bus Timing: Two-State Access

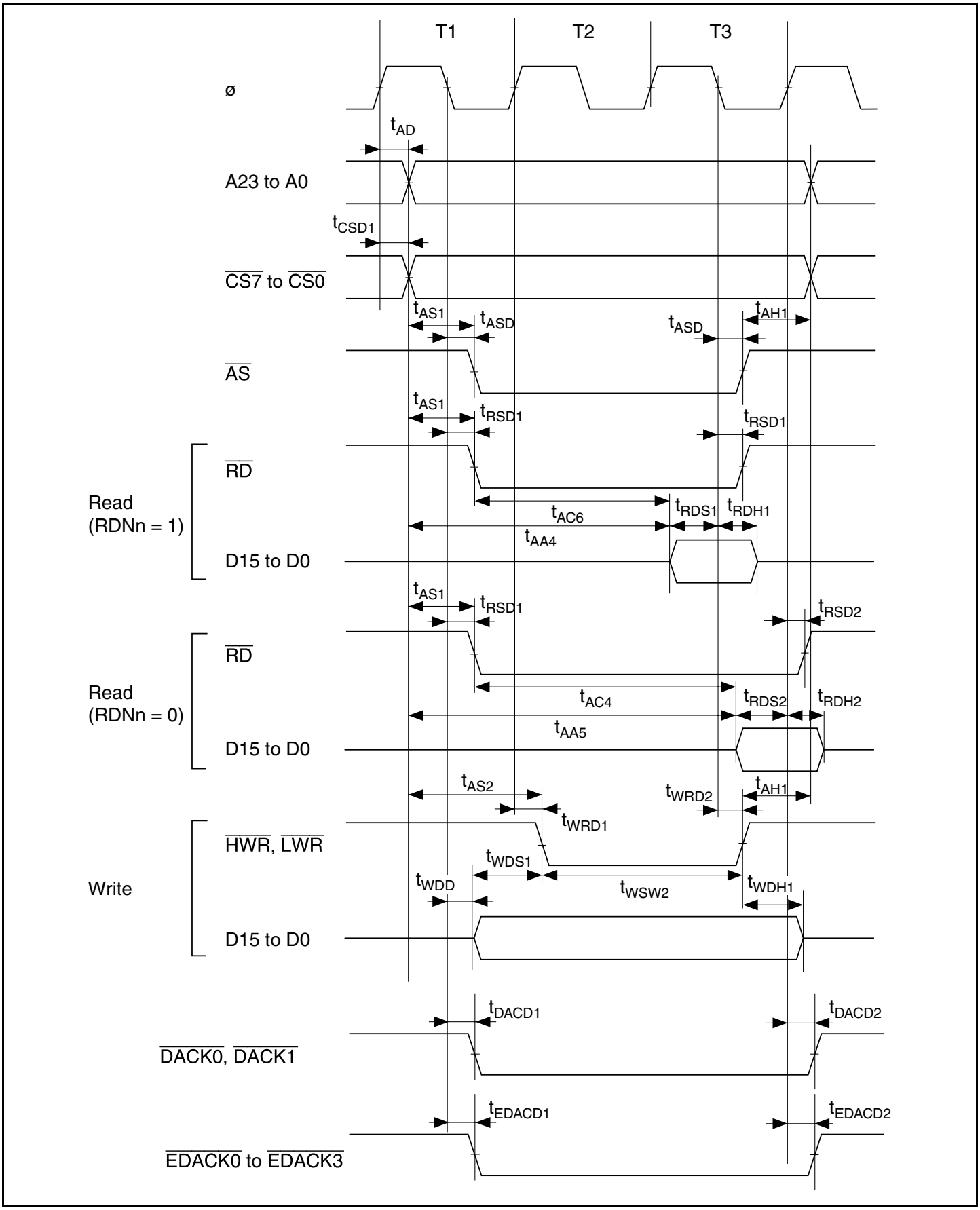


Figure 24.8 Basic Bus Timing: Three-State Access

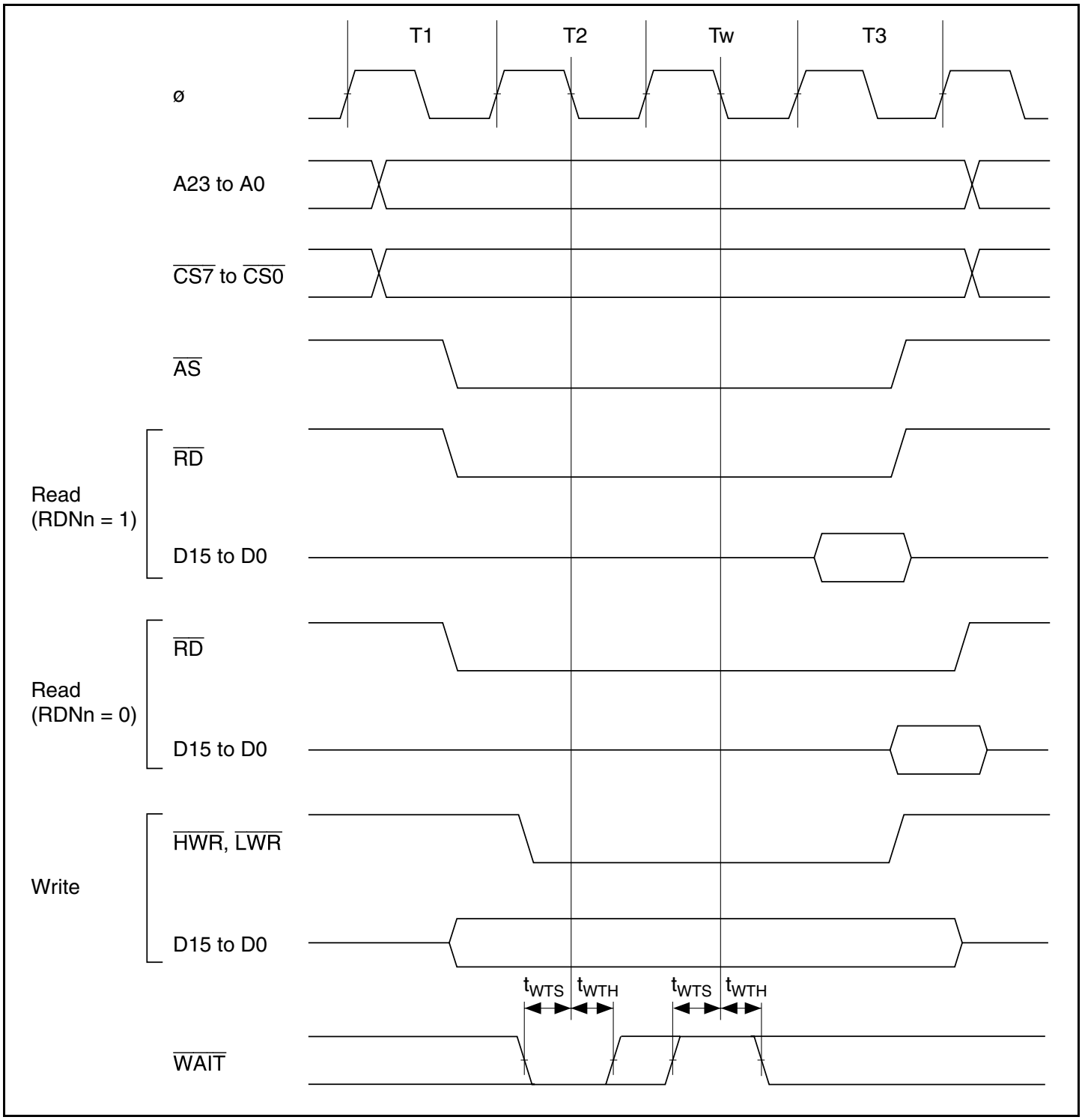


Figure 24.9 Basic Bus Timing: Three-State Access, One Wait

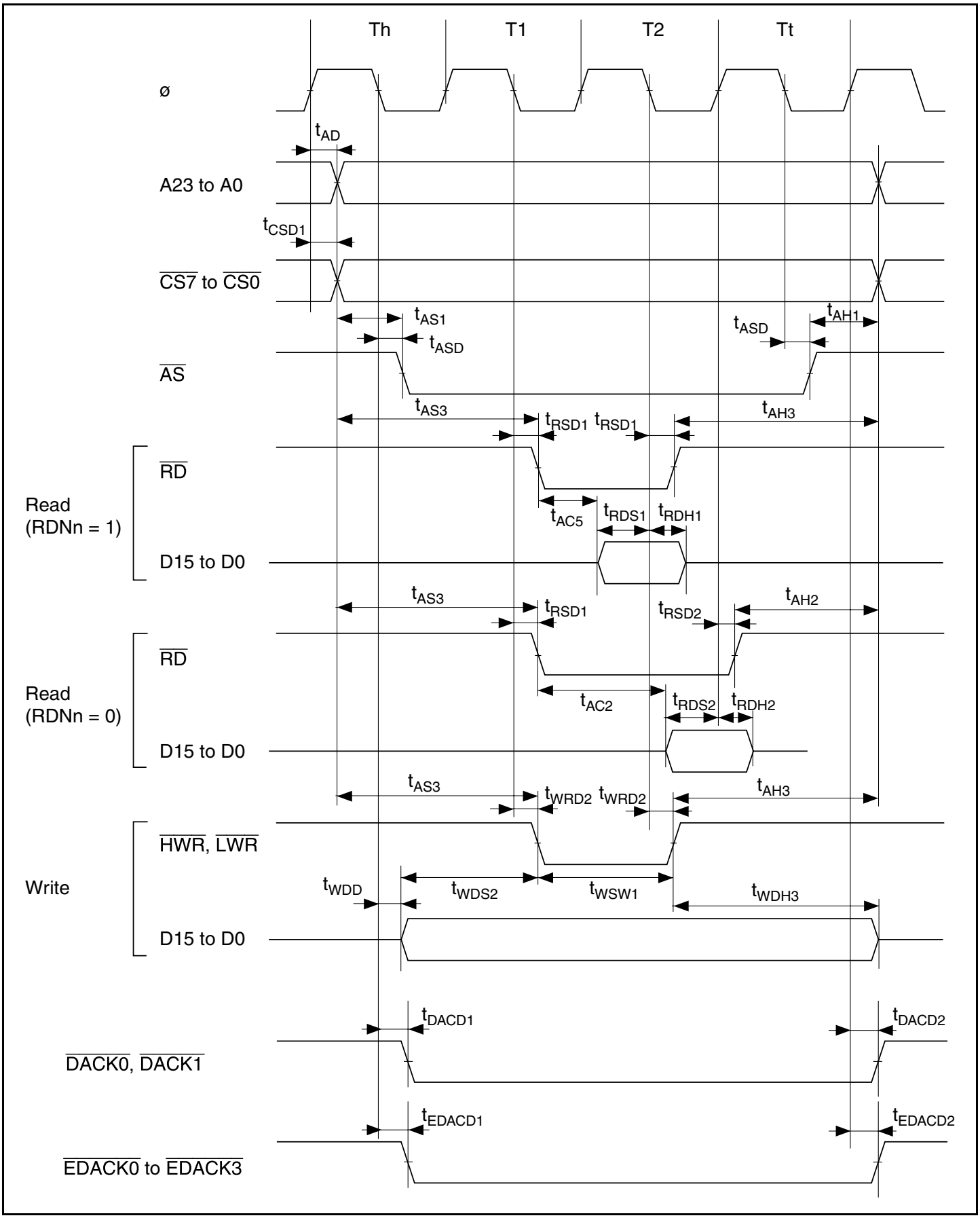
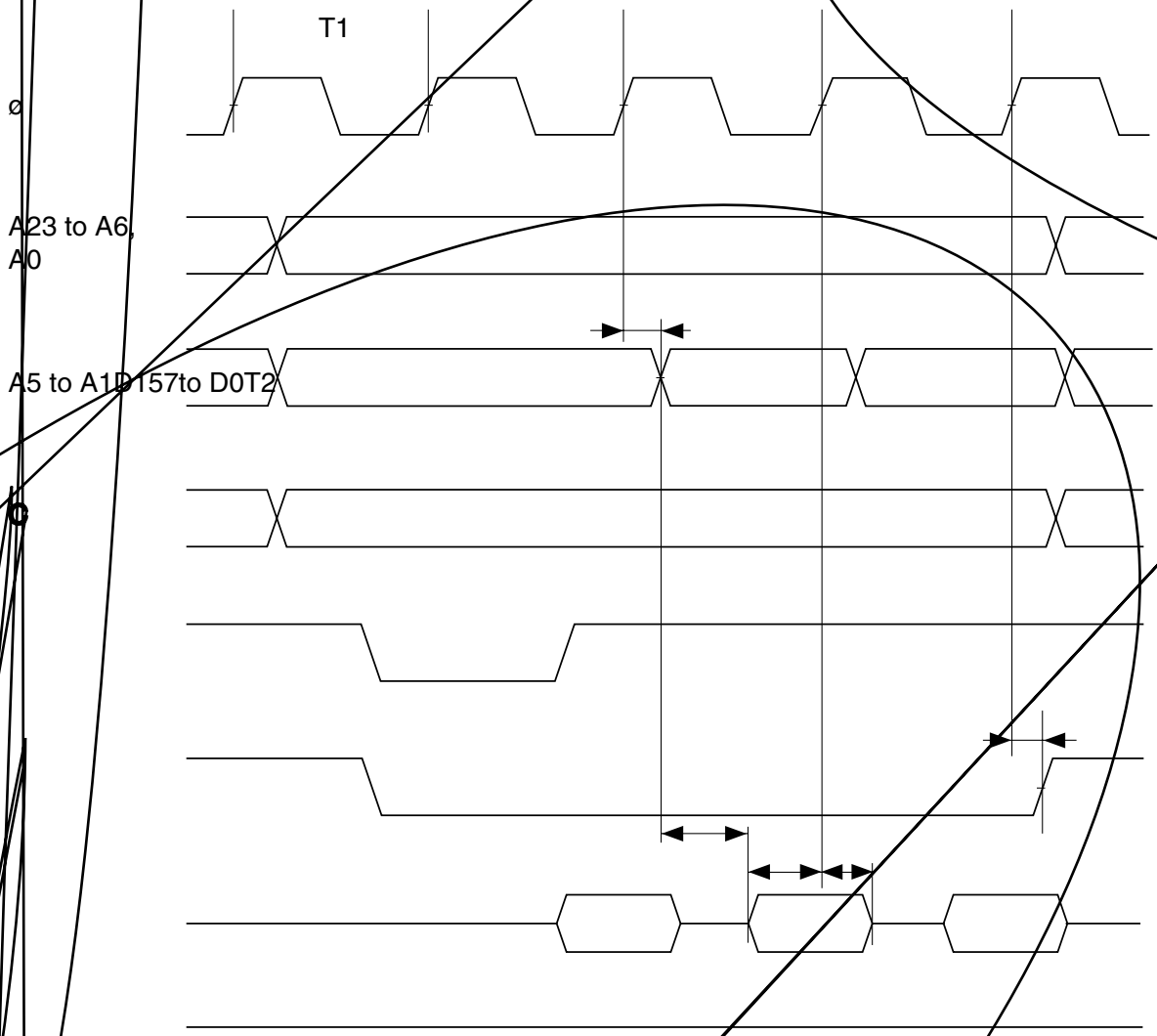


Figure 24.10 Basic Bus Timing: Two-State Access (CS Assertion Period Extended)


T1
ø
A23 to A6, A0
A5 to A1
D15
to D0
T2

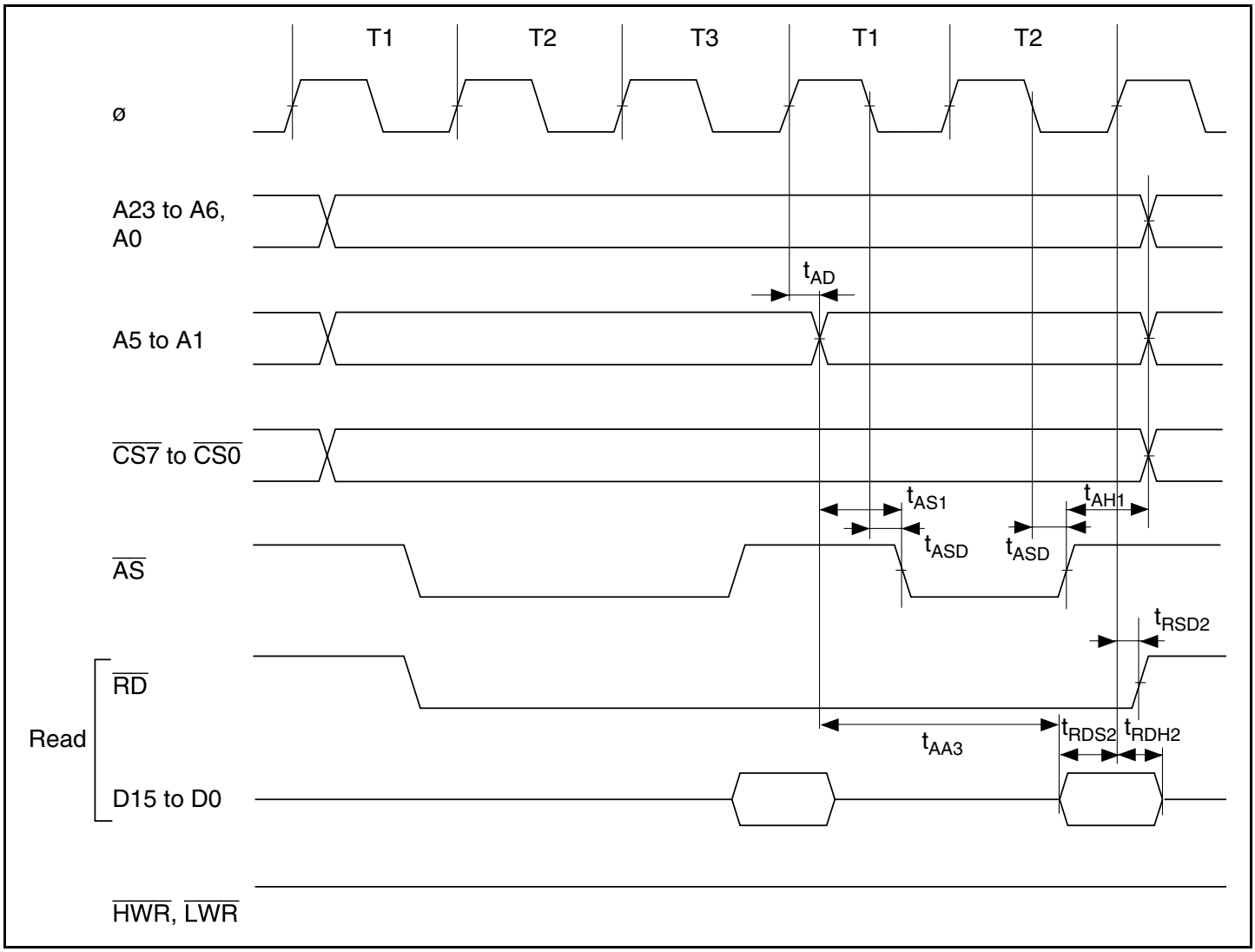
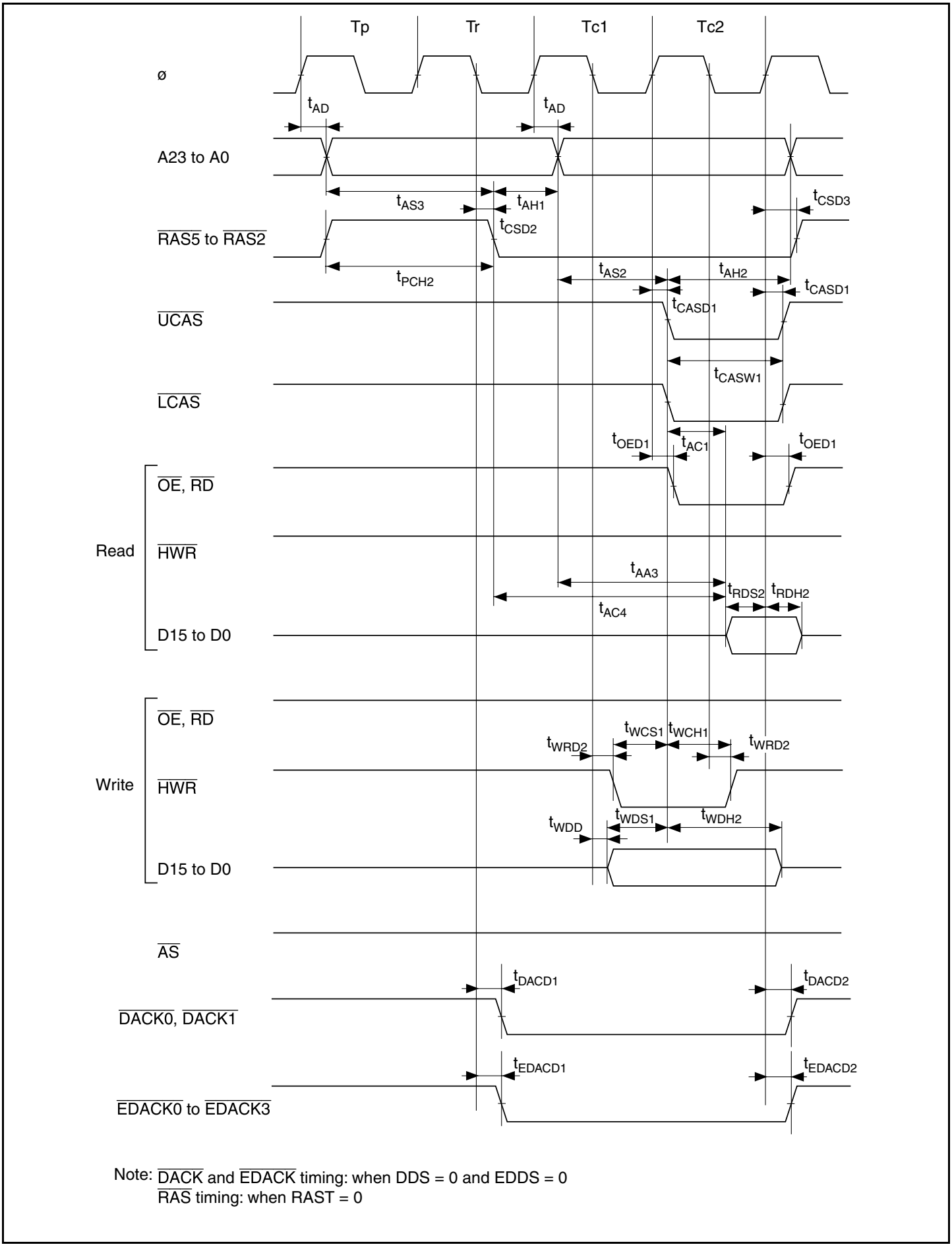
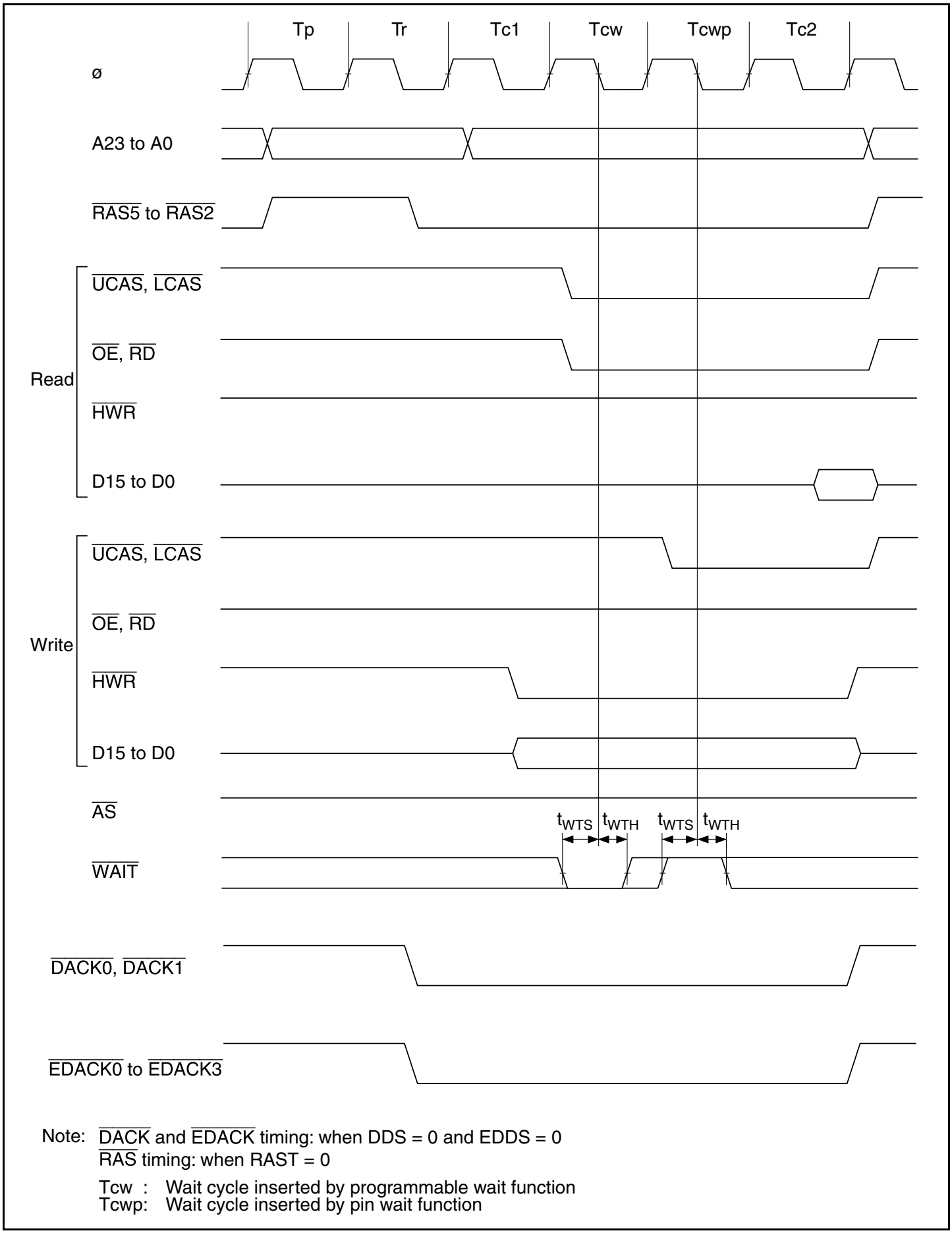


Figure 24.13 Burst ROM Access Timing: Two-State Burst Access



Note: DACK and EDACK timing: when DDS = 0 and EDDS = 0
RAS timing: when RAST = 0

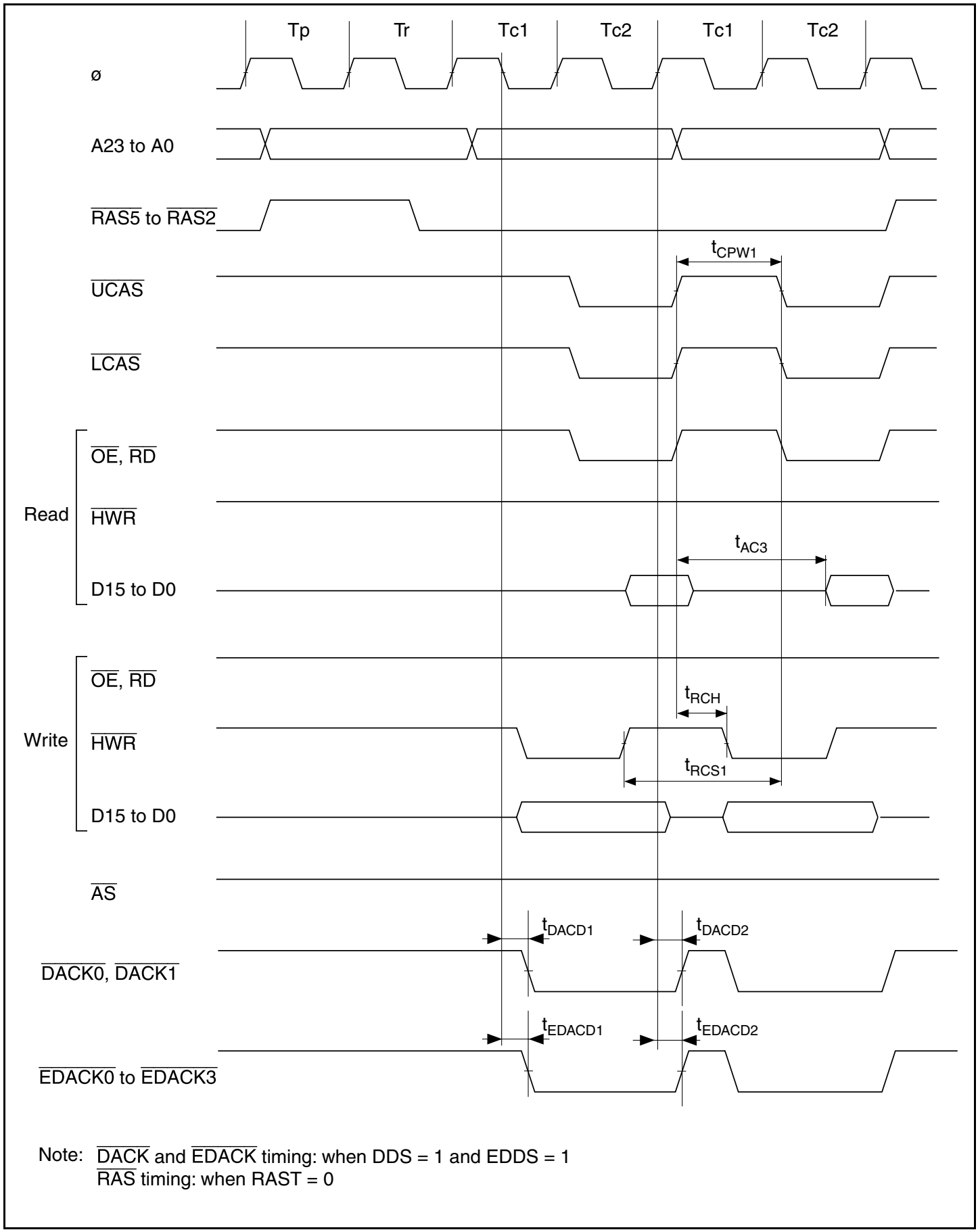
Figure 24.14 DRAM Access Timing: Two-State Access



Note: DACK and EDACK timing: when DDS = 0 and EDDS = 0
RAS timing: when RAST = 0

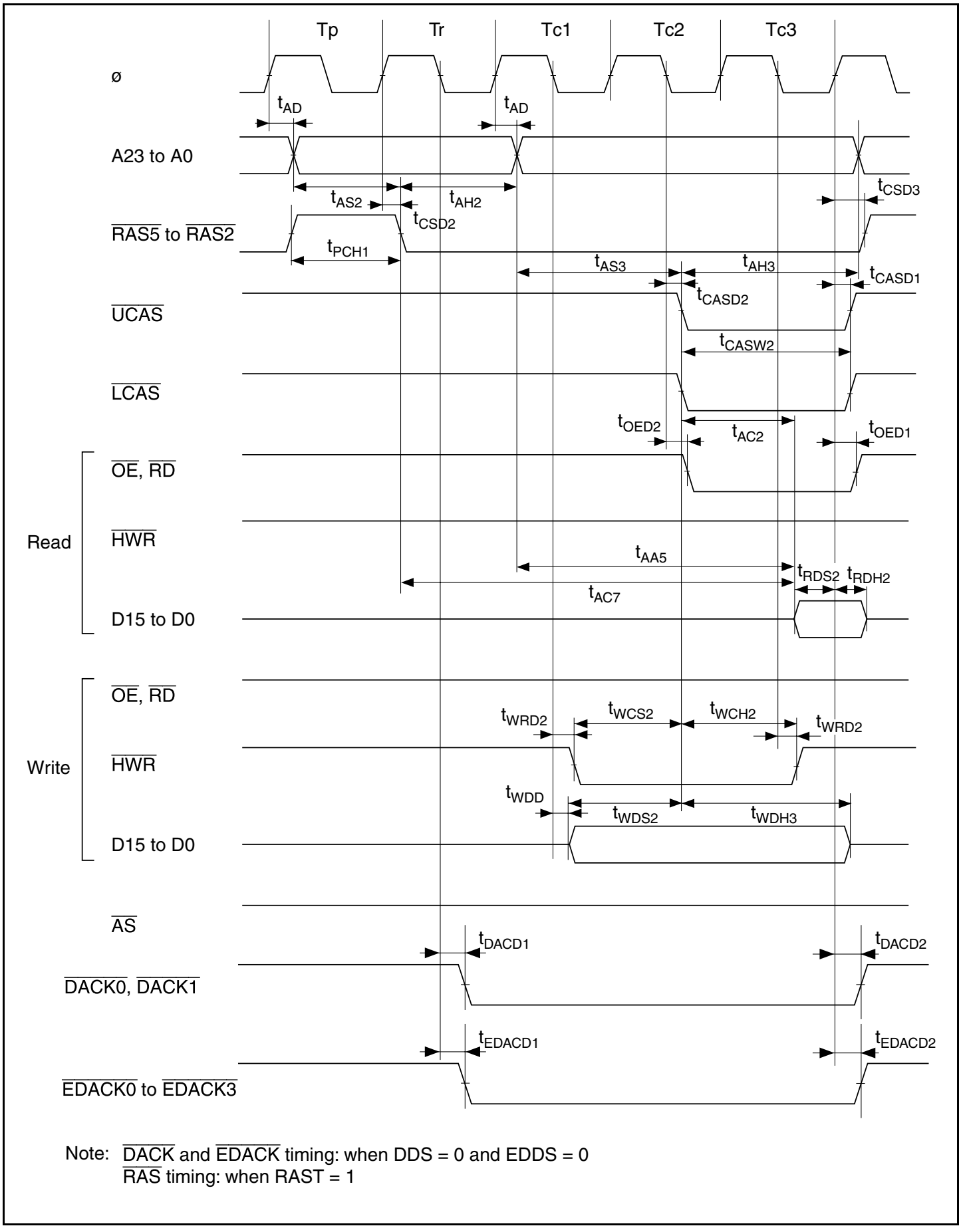
Tcw : Wait cycle inserted by programmable wait function
Tcwp: Wait cycle inserted by pin wait function

Figure 24.15 DRAM Access Timing: Two-State Access, One Wait



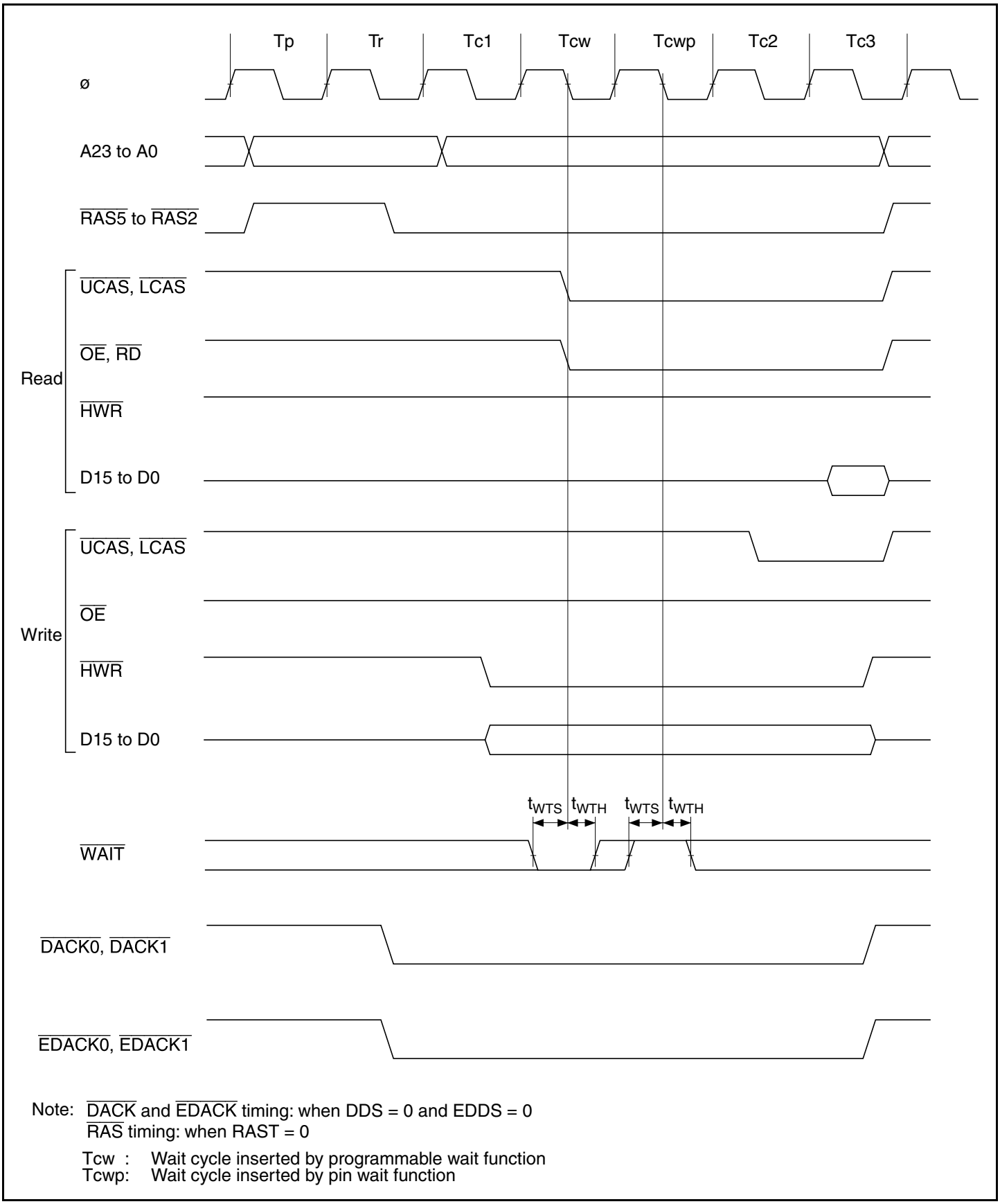
Note: DACK and EDACK timing: when DDS = 1 and EDDS = 1
RAS timing: when RAST = 0

Figure 24.16 DRAM Access Timing: Two-State Burst Access



Note: DACK and EDACK timing: when DDS = 0 and EDDS = 0
RAS timing: when RAST = 1

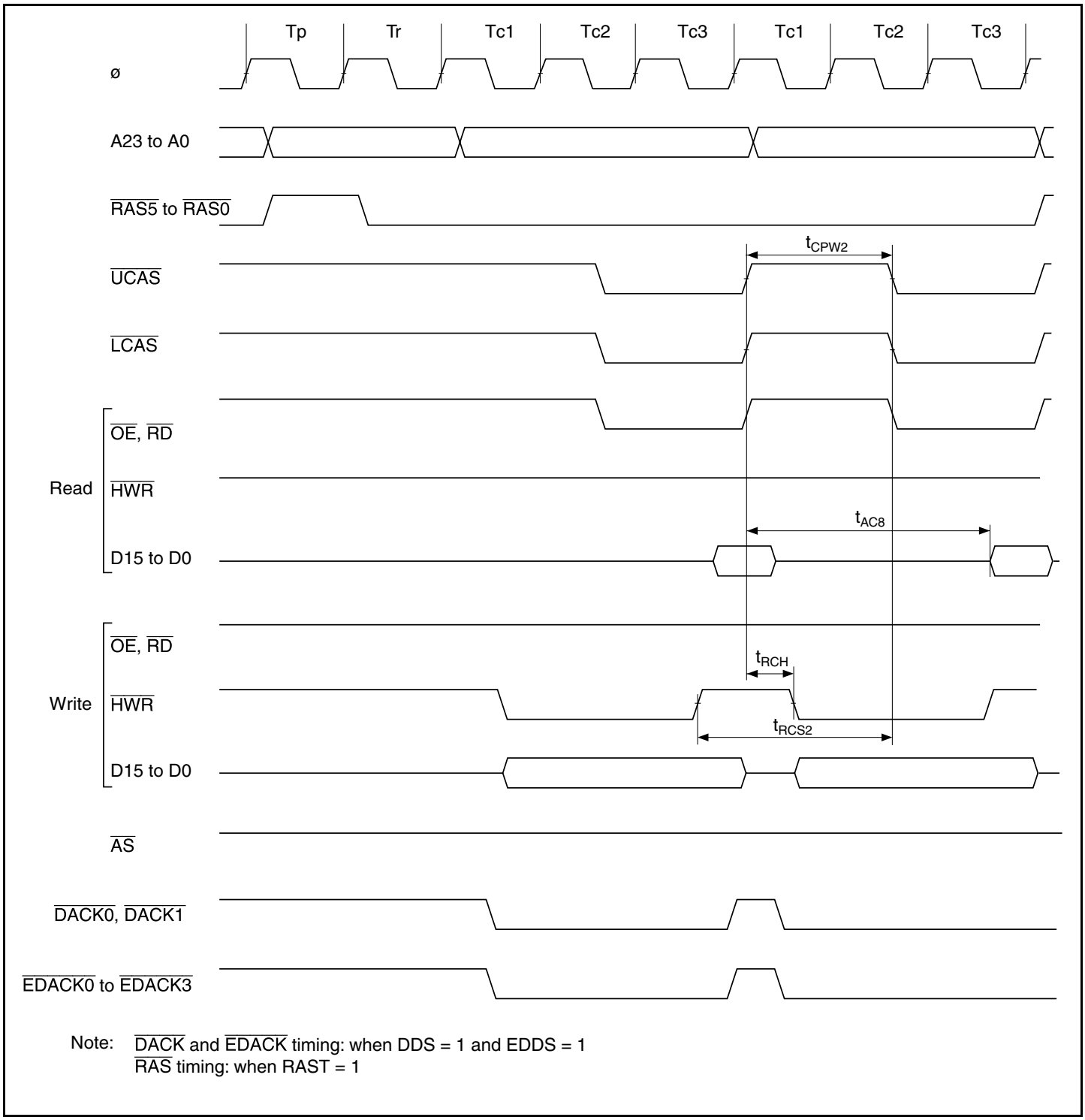
Figure 24.17 DRAM Access Timing: Three-State Access (RAST = 1)



Note: DACK and EDACK timing: when DDS = 0 and EDDS = 0
RAS timing: when RAST = 0

Tcw : Wait cycle inserted by programmable wait function
Tcwp: Wait cycle inserted by pin wait function

Figure 24.18 DRAM Access Timing: Three-State Access, One Wait



Note: DACK and EDACK timing: when DDS = 1 and EDDS = 1
RAS timing: when RAST = 1

Figure 24.19 DRAM Access Timing: Three-State Burst Access

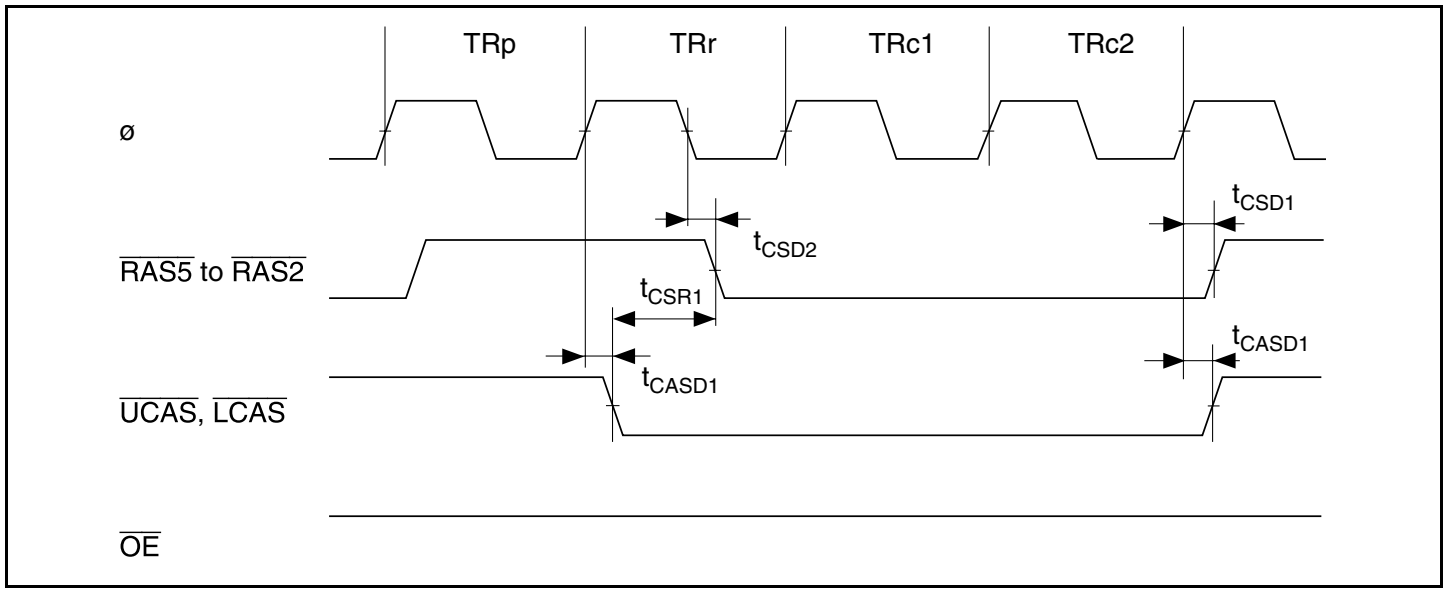


Figure 24.20 CAS-Before-RAS Refresh Timing

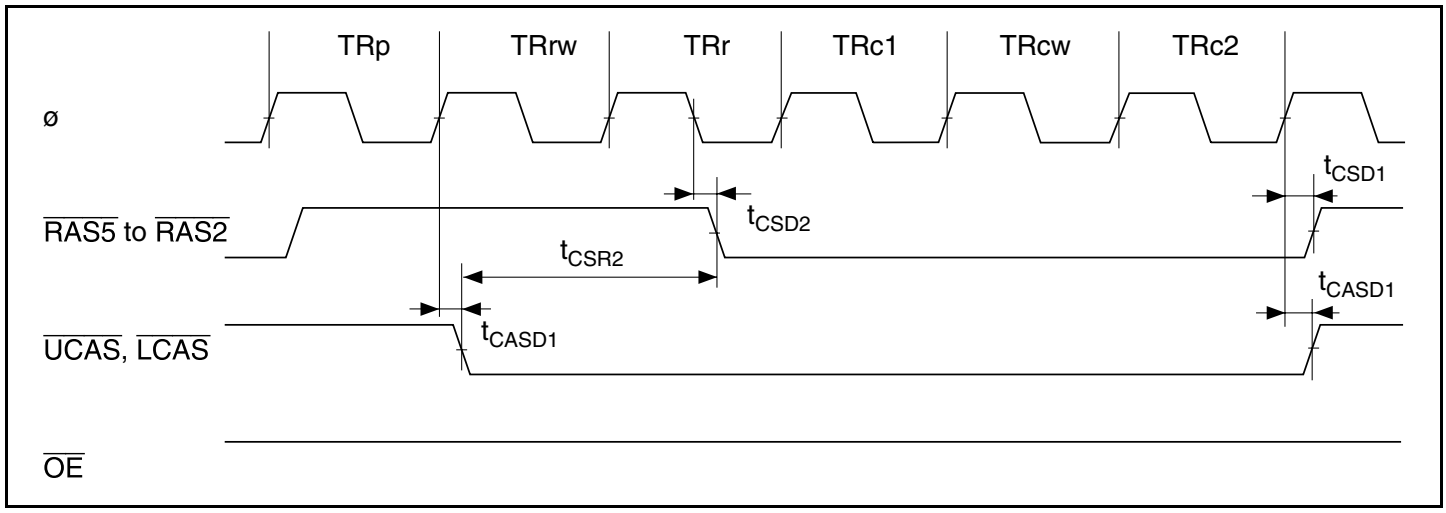


Figure 24.21 CAS-Before-RAS Refresh Timing (with Wait Cycle Insertion)

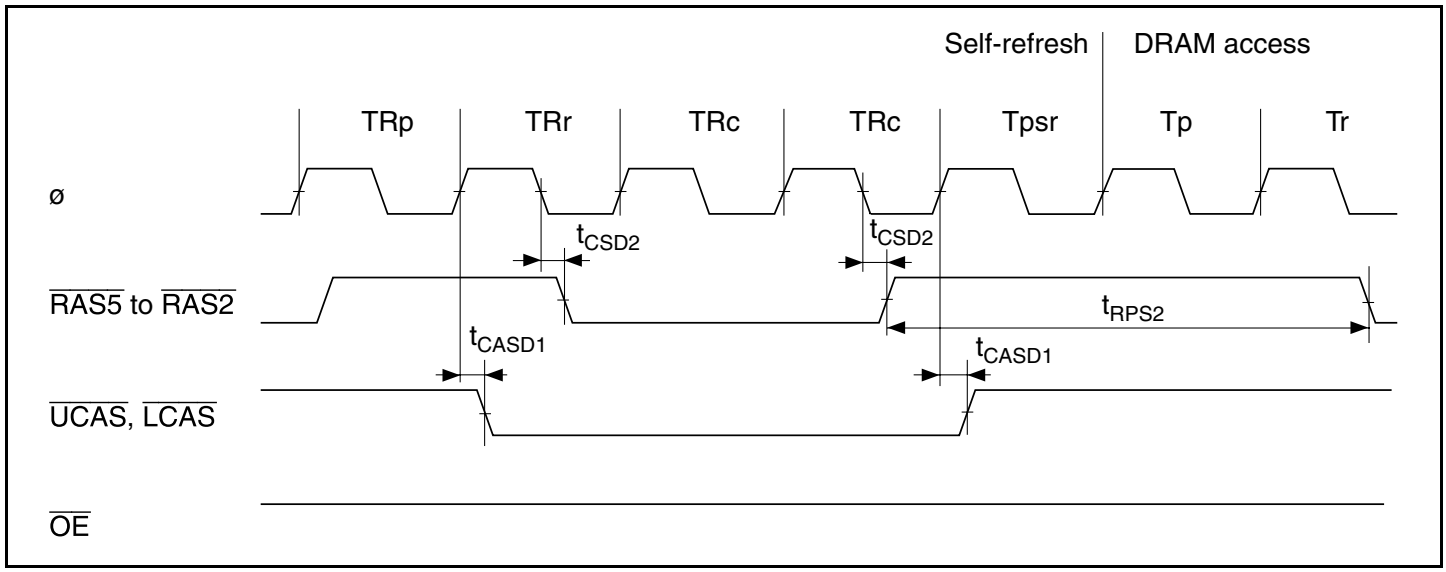


Figure 24.22 Self-Refresh Timing
(Return from Software Standby Mode: RAST = 0)

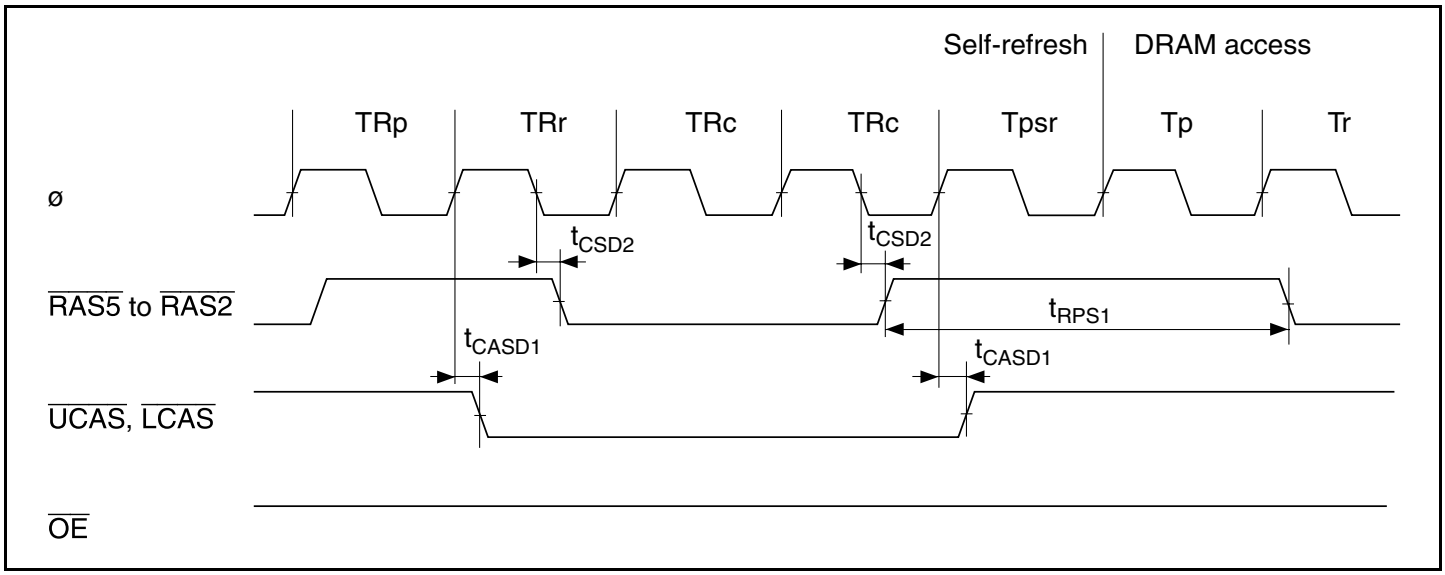


Figure 24.23 Self-Refresh Timing
(Return from Software Standby Mode: RAST = 1)

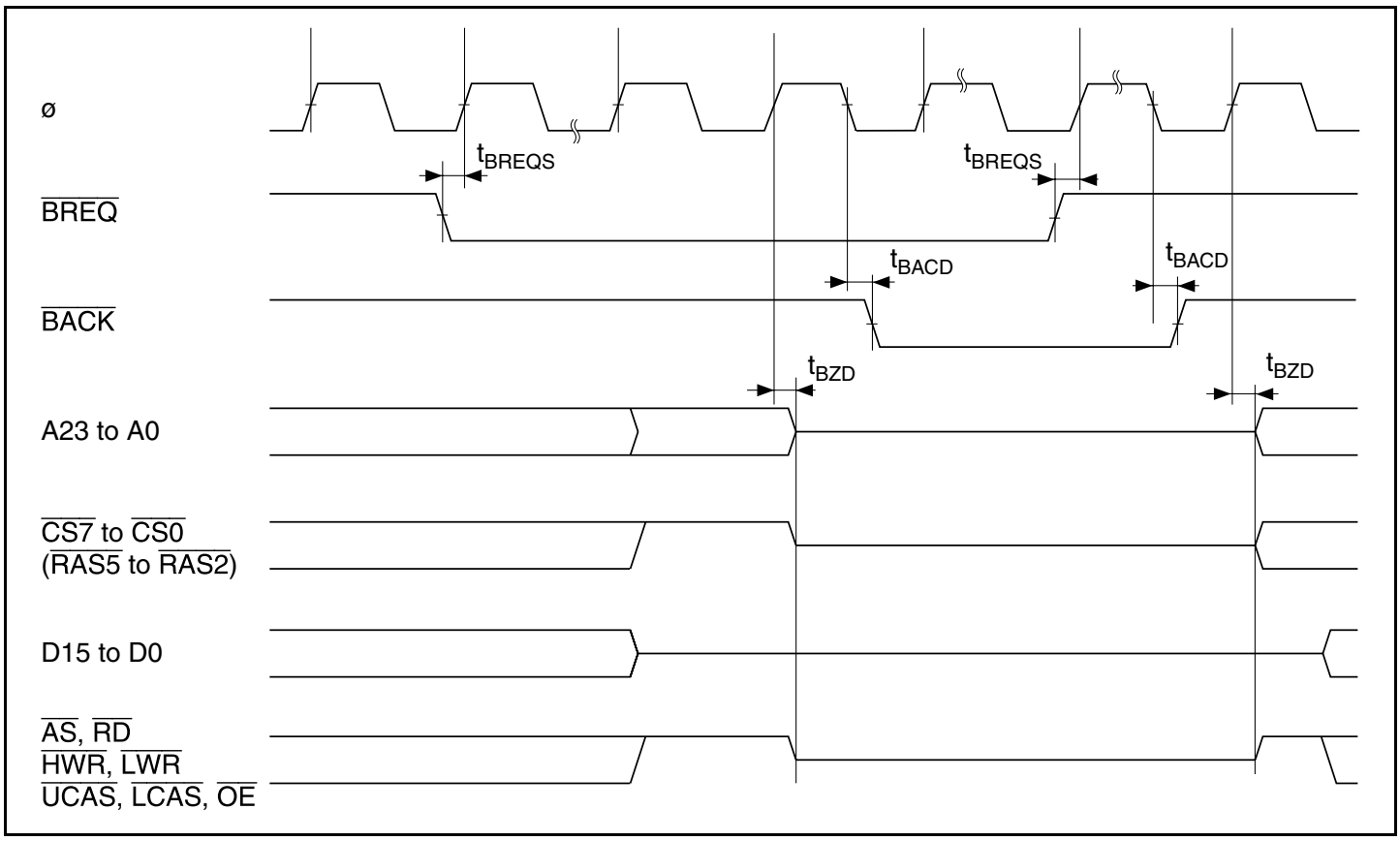


Figure 24.24 External Bus Release Timing

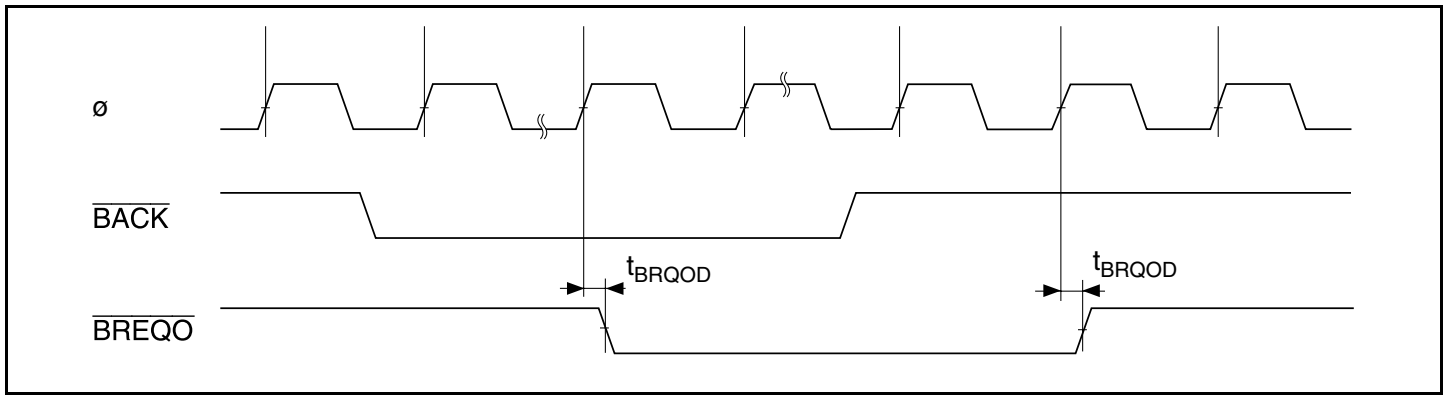


Figure 24.25 External Bus Request Output Timing

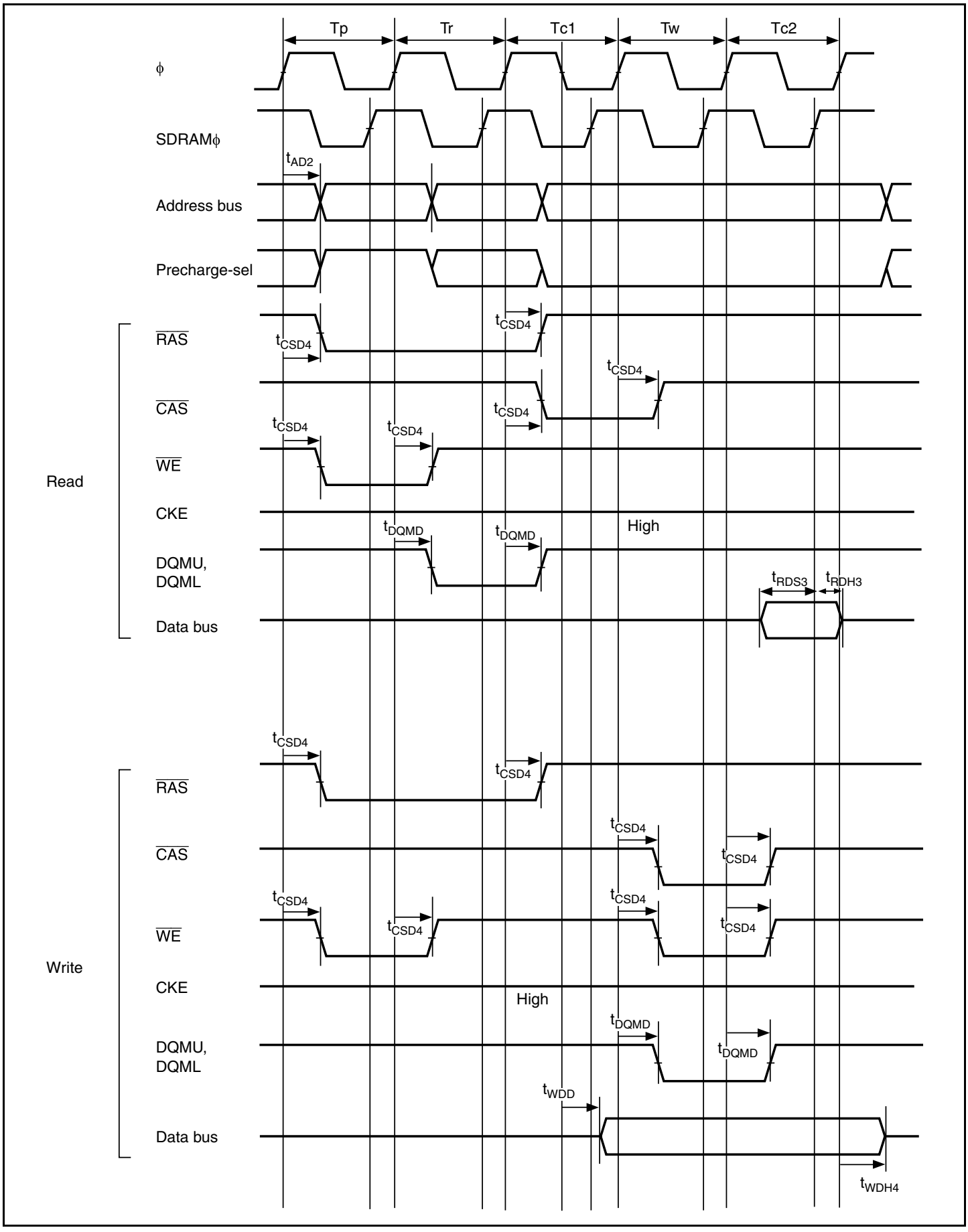


Figure 24.26 Synchronous DRAM Basic Access Timing (CAS Latency 2)

Note: Not supported in the H8S/2678 Series.

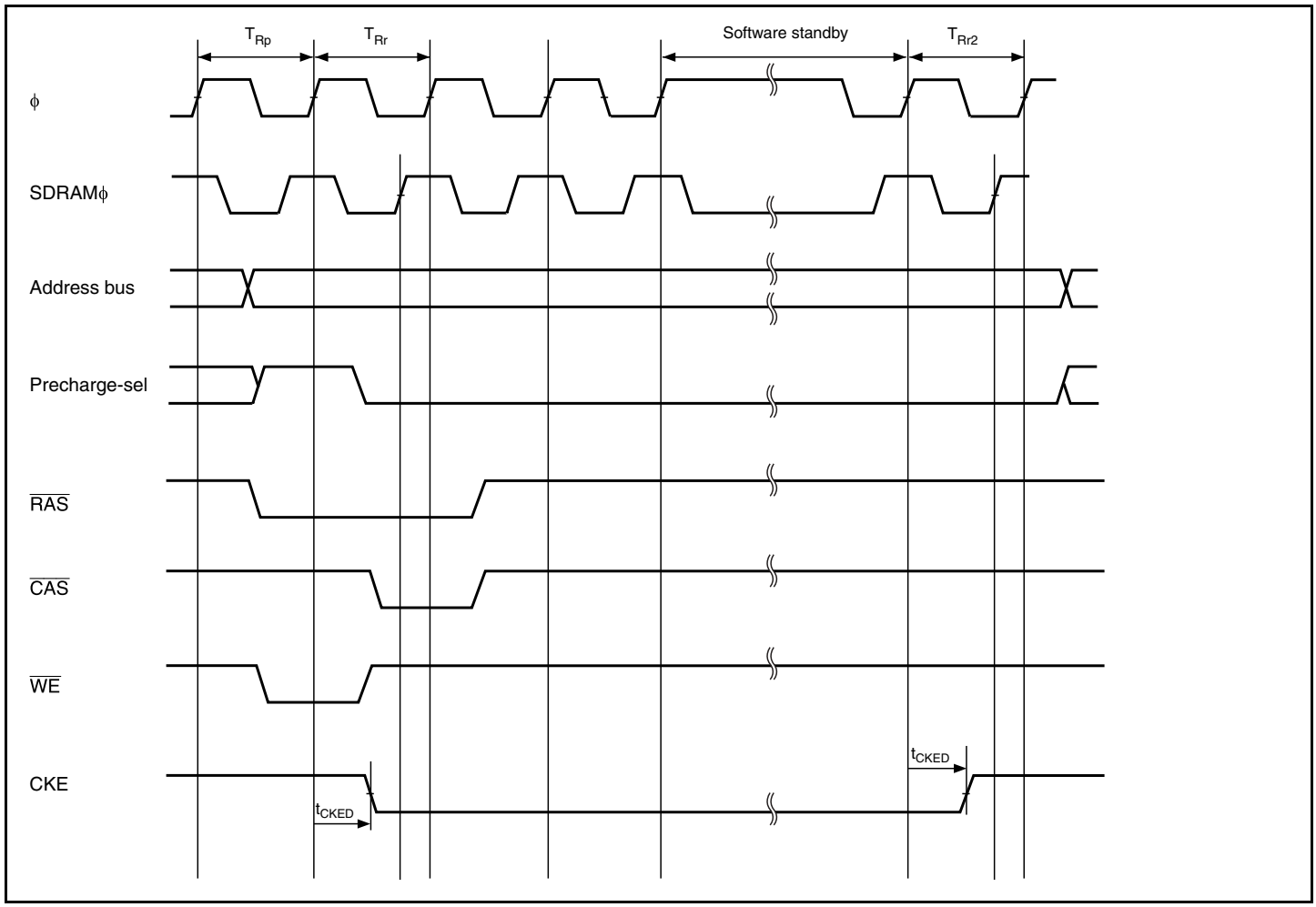


Figure 24.27 Synchronous DRAM Self-Refresh Timing

Note: Not supported in the H8S/2678 Series.

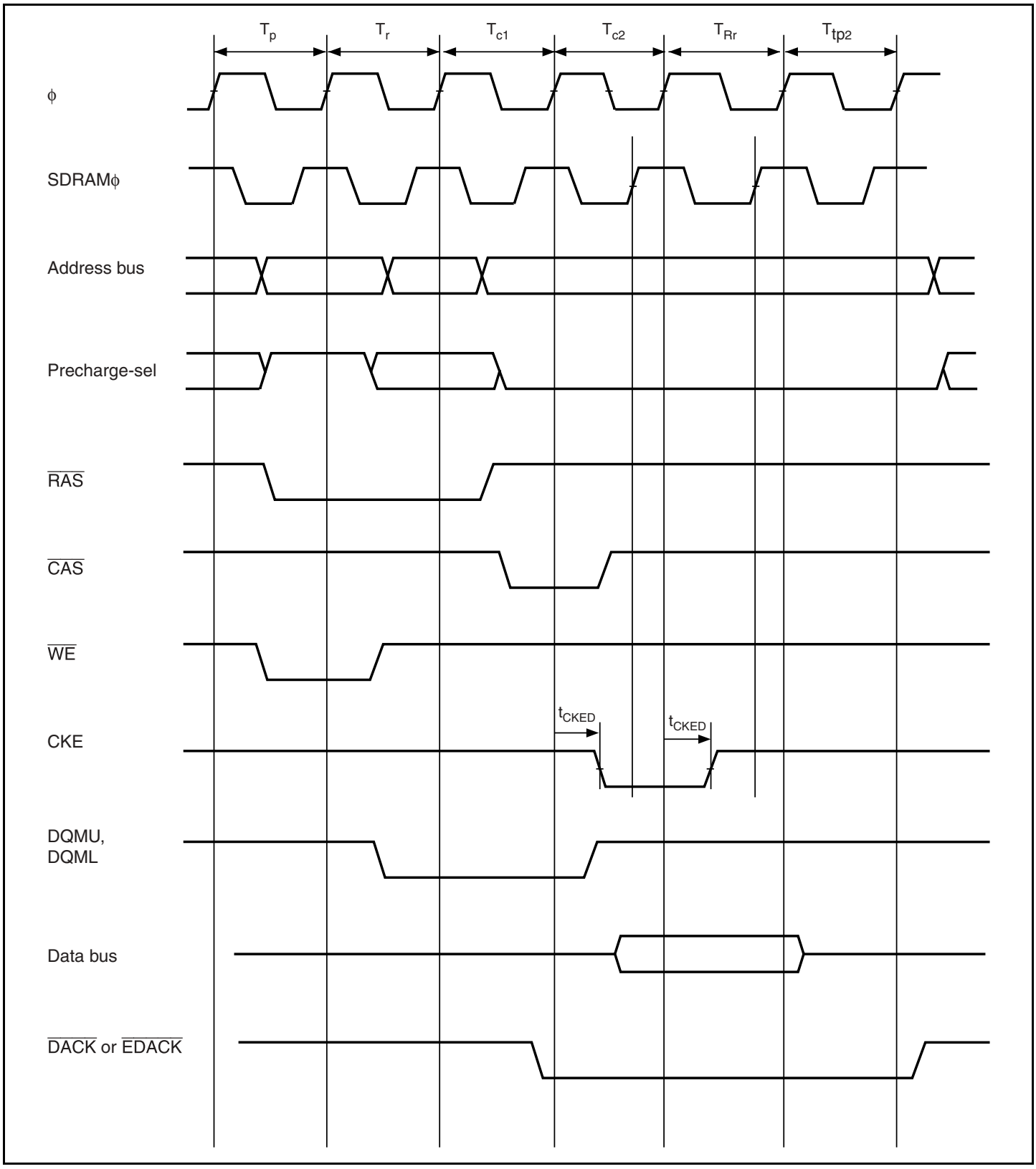


Figure 24.28 Read Data: Two-State Expansion (CAS Latency 2)

Note: Not supported in the H8S/2678 Series.

DMAC and EXDMAC Timing

Table 24.9 DMAC and EXDMAC Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ø = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
DREQ setup time	t_{DRQS}	25	—	ns	Figure 24.32
DREQ hold time	t_{DRQH}	10	—		
TEND delay time	t_{TED}	—	18	ns	Figure 24.31
DACK delay time 1	t_{DACD1}	—	18		Figure 24.29
DACK delay time 2	t_{DACD2}	—	18		Figure 24.30
EDREQ setup time	t_{EDRQS}	25	—	ns	Figure 24.32
EDREQ hold time	t_{EDRQH}	10	—		
ETEND delay time	t_{ETED}	—	18	ns	Figure 24.31
EDACK delay time 1	t_{EDACD1}	—	18		Figure 24.29
EDACK delay time 2	t_{EDACD2}	—	18		Figure 24.30
EDRAK delay time	t_{EDRKD}	—	18	ns	Figure 24.33

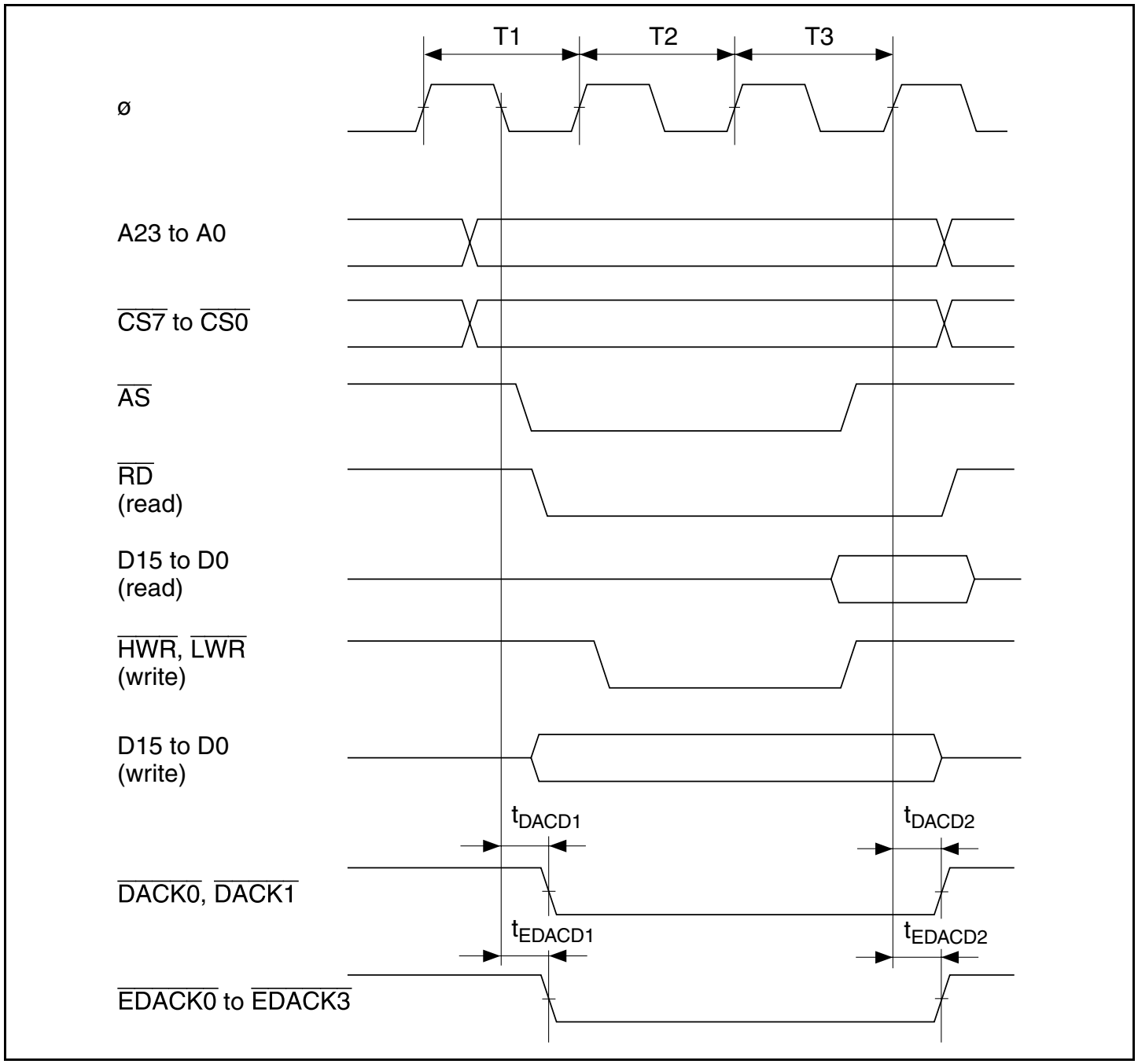
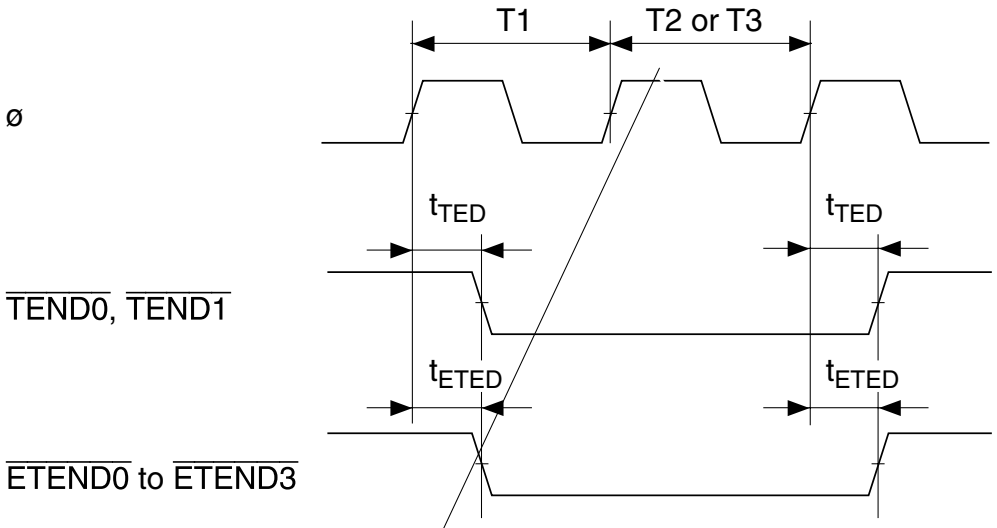


Figure 24.30 DMAC and EXDMAC Single Address Transfer Timing: Three-State Access


T1
T2 or T3
ø
tTED
tTED
TEND0, TEND1
tETED
tETED
ETEND0 to ETEND3

Timing of On-Chip Peripheral Modules

Table 24.10 Timing of On-Chip Peripheral Modules

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item			Symbol	Min	Max	Unit	Test Conditions
I/O ports	Output data delay time		t_{PWD}	—	40	ns	Figure 24.34
	Input data setup time		t_{PRS}	25	—	ns	
	Input data hold time		t_{PRH}	25	—	ns	
PPG	Pulse output delay time		t_{POD}	—	40	ns	Figure 24.35
TPU	Timer output delay time		t_{TOCD}	—	40	ns	Figure 24.36
	Timer input setup time		t_{TICS}	25	—	ns	
	Timer clock input setup time		t_{TCKS}	25	—	ns	Figure 24.37
	Timer clock pulse width	Single-edge specification	t_{TCKWH}	1.5	—	t_{cyc}	
		Both-edge specification	t_{TCKWL}	2.5	—	t_{cyc}	
8-bit timer	Timer output delay time		t_{TMOD}	—	40	ns	Figure 24.38
	Timer reset input setup time		t_{TMRS}	25	—	ns	Figure 24.40
	Timer clock input setup time		t_{TMCS}	25	—	ns	Figure 24.39
	Timer clock pulse width	Single-edge specification	t_{TMCWH}	1.5	—	t_{cyc}	
		Both-edge specification	t_{TMCWL}	2.5	—	t_{cyc}	
WDT	Overflow output delay time		t_{WOVD}	—	40	ns	Figure 24.41
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 24.42
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5		
	Transmit data delay time		t_{TXD}	—	40	ns	Figure 24.43
	Receive data setup time (synchronous)		t_{RXS}	40	—	ns	
	Receive data hold time (synchronous)		t_{RXH}	40	—	ns	
A/D converter	Trigger input setup time		t_{TRGS}	30	—	ns	Figure 24.44

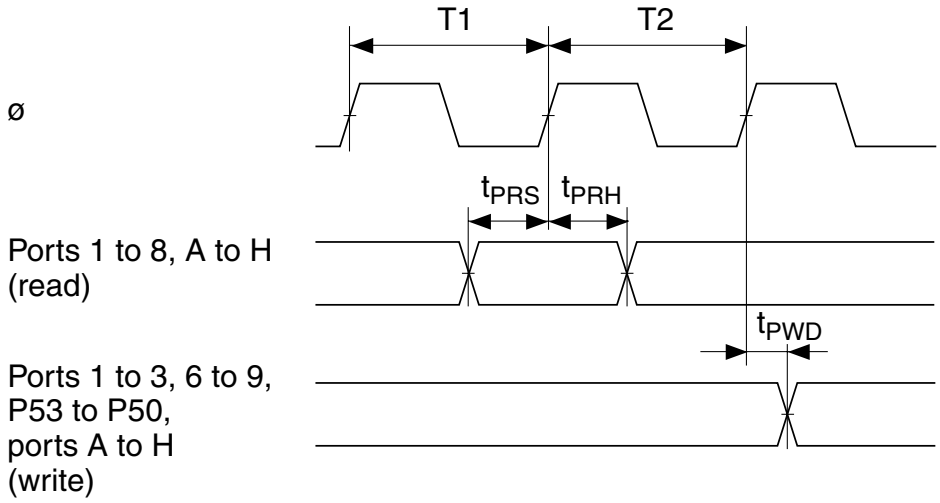


Figure 24.34 I/O Port Input/Output Timing

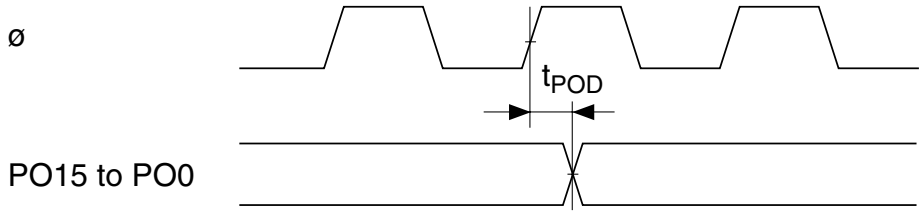
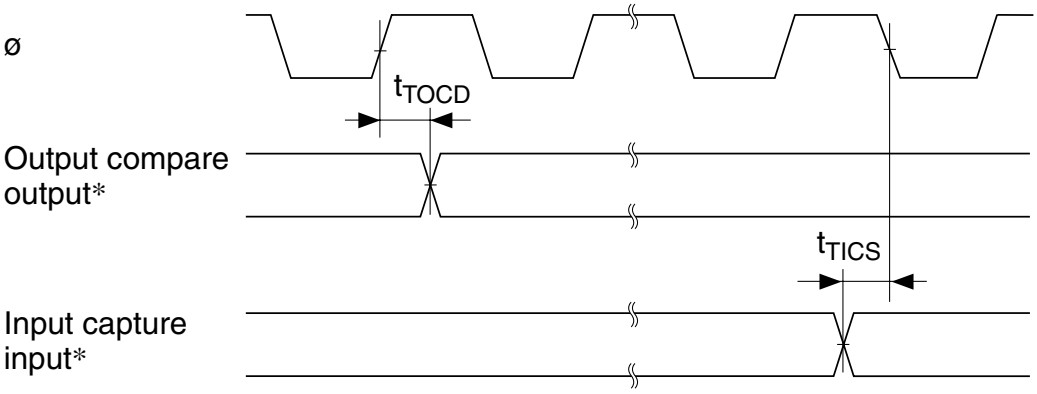


Figure 24.35 PPG Output Timing



Note: * TIOCA0 to TIOCA5, TIOCB0 to TIOCB5, TIOCC0, TIOCC3, TIOCD0, TIOCD3

Figure 24.36 TPU Input/Output Timing

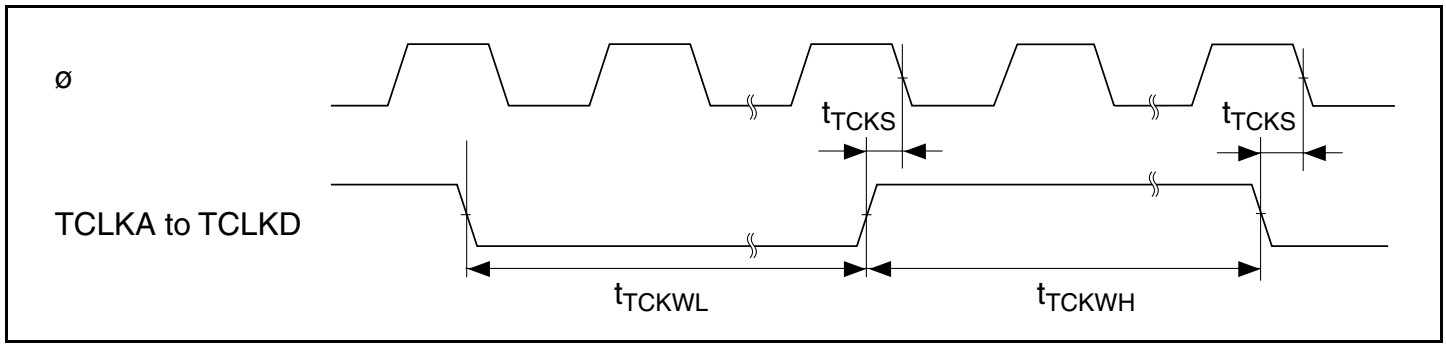


Figure 24.37 TPU Clock Input Timing

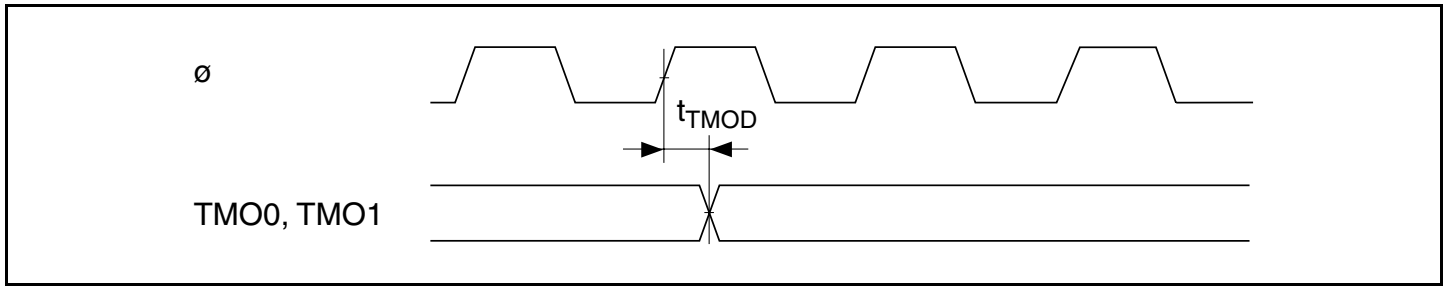


Figure 24.38 8-Bit Timer Output Timing

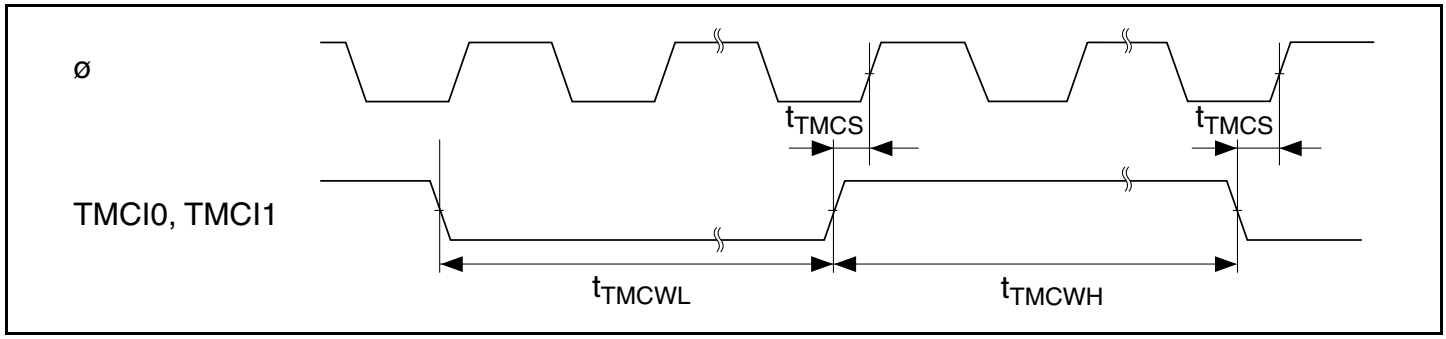


Figure 24.39 8-Bit Timer Clock Input Timing

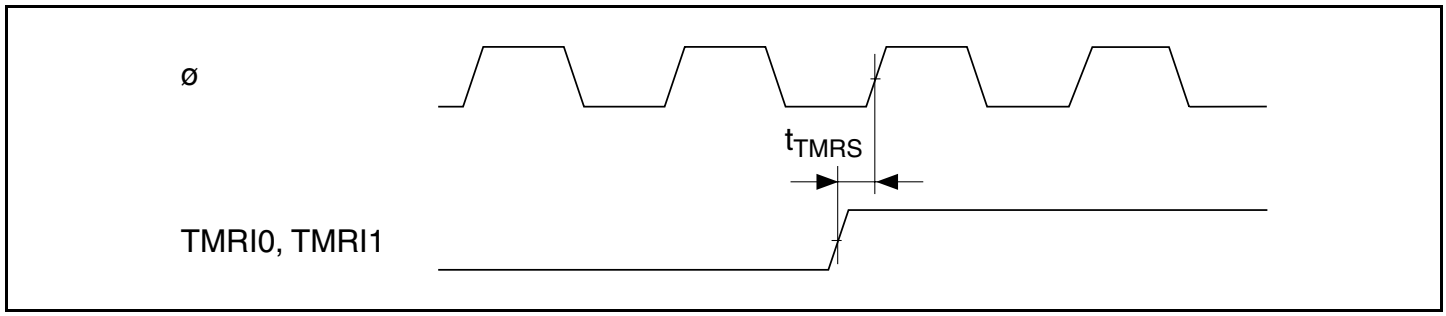


Figure 24.40 8-Bit Timer Reset Input Timing

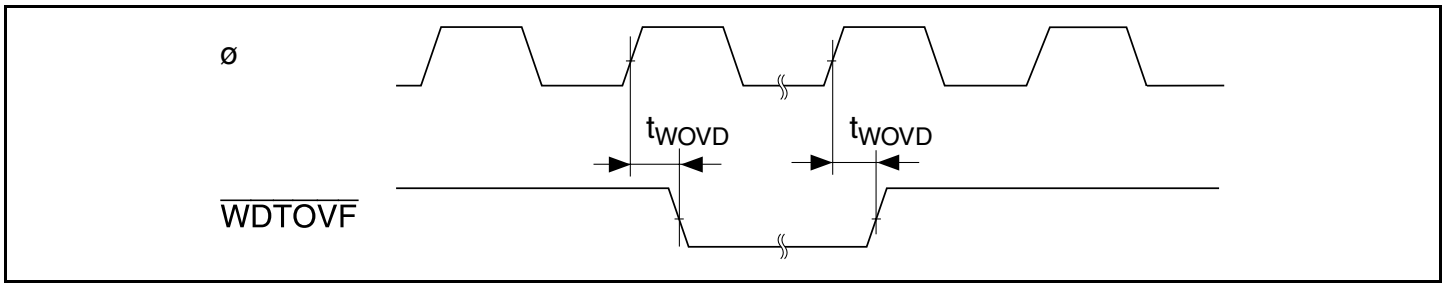


Figure 24.41 WDT Output Timing

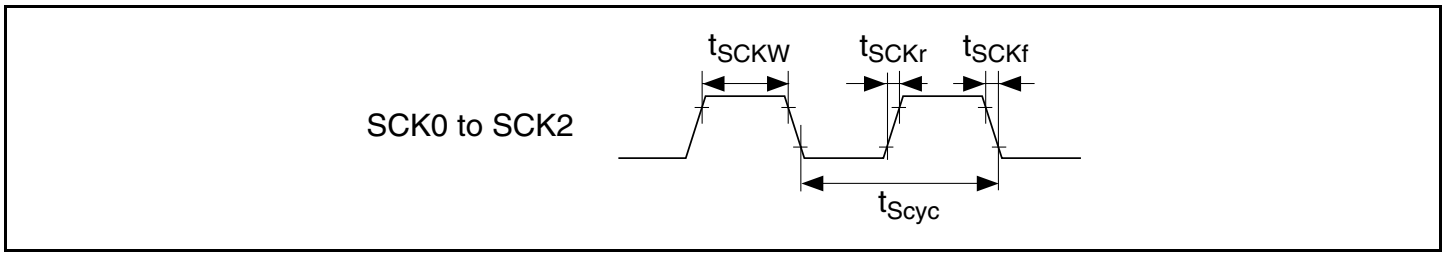


Figure 24.42 SCK Clock Input Timing

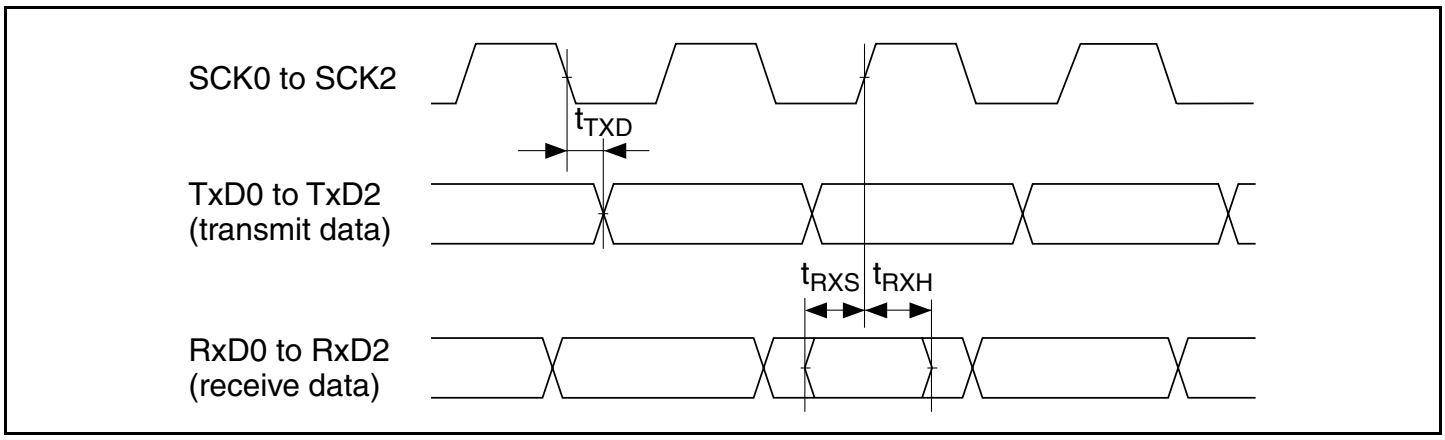


Figure 24.43 SCI Input/Output Timing: Synchronous Mode

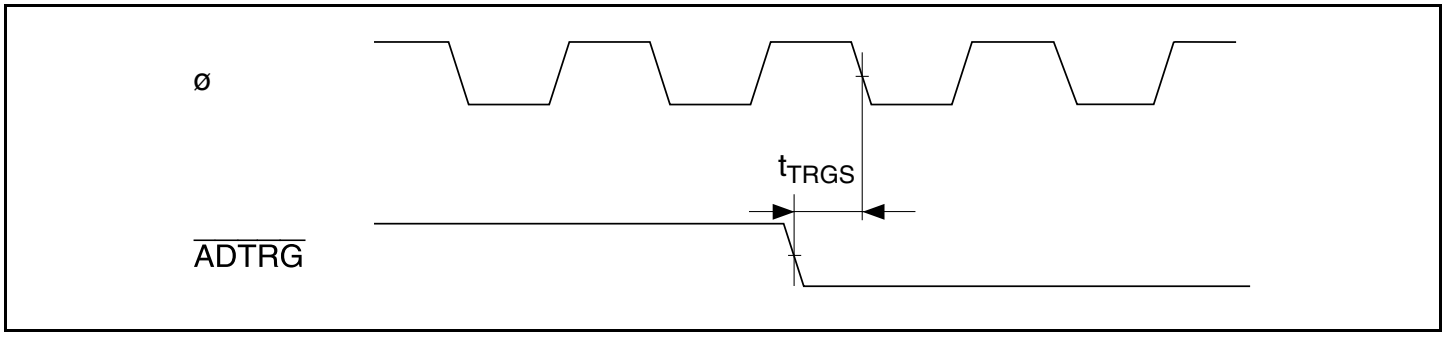


Figure 24.44 A/D Converter External Trigger Input Timing

24.4 A/D Conversion Characteristics

Table 24.11 A/D Conversion Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ø = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Min	Typ	Max	Unit
Resolution	10	10	10	Bit
Conversion time	—	—	8.1	µs
Analog input capacitance	—	—	20	pF
Permissible signal source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±7.5	LSB
Offset error	—	—	±7.5	LSB
Full-scale error	—	—	±7.5	LSB
Quantization error	—	±0.5	—	LSB
Absolute accuracy	—	—	±8.0	LSB

24.5 D/A Conversion Characteristics

Table 24.12 D/A Conversion Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ø = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Min	Typ	Max	Unit	Test Conditions
Resolution	8	8	8	Bit	
Conversion time	—	—	10	µs	20 pF capacitive load
Absolute accuracy	—	±2.0	±3.0	LSB	2 MΩ resistive load
	—	—	±2.0	LSB	4 MΩ resistive load

24.6 Flash Memory Characteristics

Table 24.13 Flash Memory Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = 0°C to 75°C (program/erase operating temperature range: regular specifications), T_a = 0°C to 85°C (program/erase operating temperature range: wide-range specifications)

Item		Symbol		Min	Typ	Max	Unit	Test Conditions
Programming time*[1, 2, 4]		t_P		—	10	200	ms/ 128 bytes	
Erase time*[1, 3, 6]		t_E		—	50	1000	ms/ 128 bytes	
Rewrite times		N_{WEC}		—	—	100	Times	
Programming	Wait time after SWE bit setting*[1]	x		1	—	—	µs	
	Wait time after PSU bit setting*[1]	y		50	—	—	µs	
	Wait time after P bit setting*[1, 4]	z	z1	—	—	30	µs	1 ≤ n ≤ 6
			z2	—	—	200	µs	7 ≤ n ≤ 1000
			z3	—	—	10	µs	Additional programming wait
	Wait time after P bit clearing*[1]	α		5	—	—	µs	
	Wait time after PSU bit clearing*[1]	β		5	—	—	µs	
	Wait time after PV bit setting*[1]	γ		4	—	—	µs	
	Wait time after H'FF dummy write*[1]	ε		2	—	—	µs	
	Wait time after PV bit clearing*[1]	η		2	—	—	µs	
	Wait time after SWE bit clearing*[1]	θ		100	—	—	µs	
	Maximum number of writes*[1, 4]	N		—	—	1000*[5]	Times	

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Erasing	Wait time after SWE bit setting*[1]	x	1	—	—	µs	
	Wait time after ESU bit setting*[1]	y	100	—	—	µs	
	Wait time after E bit setting*[1, *6]	z	—	—	10	µs	Erase time wait
	Wait time after E bit clearing*[1]	α	10	—	—	µs	
	Wait time after ESU bit clearing*[1]	β	10	—	—	µs	
	Wait time after EV bit setting*[1]	γ	20	—	—	µs	
	Wait time after H'FF dummy write*[1]	ε	2	—	—	µs	
	Wait time after EV bit clearing*[1]	η	4	—	—	µs	
	Wait time after SWE bit clearing*[1]	θ	100	—	—	µs	
	Maximum number of erases*[1, *6]	N	—	—	100	Times	

Notes: 1. Follow the program/erase algorithms when making the time settings.

2. Programming time per 128 bytes. (Indicates the total time during which the P bit is set in flash memory control register 1 (FLMCR1). Does not include the program-verify time.)

3. Time to erase one block. (Indicates the time during which the E bit is set in FLMCR1. Does not include the erase-verify time.)

4. Maximum programming time

$$t_P\ (\max) = \sum_{i=1}^{N} \text{wait time after P bit setting (z)}$$

5. The maximum number of writes (N) should be set as shown below according to the actual set value of (z) so as not to exceed the maximum programming time (t_P(max)).

   The wait time after P bit setting (z) should be changed as follows according to the number of writes (n).

   Number of writes (n)

   $1 \leq n \leq 6$ z = 30 µs

   $7 \leq n \leq 1000$ z = 200 µs

   (Additional programming)

   Number of writes (n)

   $1 \leq n \leq 6$ z = 10 µs

6. For the maximum erase time (t_E(max)), the following relationship applies between the wait time after E bit setting (z) and the maximum number of erases (N):

   t_E(max) = Wait time after E bit setting (z) × maximum number of erases (N)

24.7 Usage Note

The F-ZTAT and masked ROM versions both satisfy the electrical characteristics shown in this manual, but actual electrical characteristic values, operating margins, noise margins, and other properties may vary due to differences in manufacturing process, on-chip ROM, layout patterns, and so on.

When system evaluation testing is carried out using the F-ZTAT version, the same evaluation testing should also be conducted for the masked ROM version when changing over to that version.

Appendix

A. I/O Port States in Each Pin State

Port Name	MCU Operating Mode*[1]	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port 1	1 to 7	T	T	Keep	Keep	I/O port
Port 2	1 to 7	T	T	Keep	Keep	I/O port
P34 to P30	1 to 7	T	T	Keep	Keep	I/O port
P35/OE/CKE	1 to 7	T	T	[OPE = 0, OE (CKE)*[2] output]	[OE (CKE)*[2] output]	[OE (CKE)*[2] output]
				T	T	OE (CKE)*[2]
				[OPE = 1, OE (CKE)*[2] output]	[Other than the above]	[Other than the above]
				H	Keep	I/O port
				[Other than the above]		
				Keep		
P47/DA1	1 to 7	T	T	[DAOE1 = 1]	Keep	Input port
				Keep		
				[DAOE1 = 0]		
				T		
P46/DA0	1 to 7	T	T	[DAOE0 = 1]	Keep	Input port
				Keep		
				[DAOE0 = 0]		
				T		
P45 to P40	1 to 7	T	T	T	T	Input port
P57/DA3	1 to 7	T	T	[DAOE3 = 1]	Keep	Input port
				Keep		
				[DAOE3 = 0]		
				T		
P56/DA2	1 to 7	T	T	[DAOE2 = 1]	Keep	Input port
				Keep		
				[DAOE2 = 0]		
				T		

Port Name	MCU Operating Mode*[1]	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
P55, P54	1 to 7	T	T	T	T	Input port
P53 to P50	1 to 7	T	T	Keep	Keep	I/O port
Port 6	1 to 7	T	T	Keep	Keep	I/O port
Port 7	1 to 7	T	T	Keep	Keep	I/O port
Port 8	1 to 7	T	T	Keep	Keep	I/O port
PA7/A23 PA6/A22 PA5/A21	1 to 7	T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A23 to A21 [Other than the above] I/O port
PA4/A20 PA3/A19 PA2/A18 PA1/A17 PA0/A16	1, 2, 5, 6	L	T	[OPE = 0] T [OPE = 1] Keep	T	Address output A20 to A16
	3, 4, 7	T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A20 to A16 [Other than the above] I/O port
Port B	1, 2, 5, 6	L	T	[OPE = 0] T [OPE = 1] Keep	T	Address output A15 to A8

Port Name	MCU Operating Mode*[1]	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port B	3, 4	T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A15 to A8 [Other than the above] I/O port
	3, 7	T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A15 to A8 [Other than the above] I/O port
Port C	1, 2, 5, 6	L	T	[OPE = 0] T [OPE = 1] Keep	T	Address output A7 to A0
	4	T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A7 to A0 [Other than the above] I/O port

Port Name	MCU Operating Mode*[1]		Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port C	3, 7		T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A7 to A0 [Other than the above] I/O port
Port D	1, 2, 4 to 6		T	T	T	T	D15 toD8
	3, 7		T	T	[Data bus] T [Other than the above] Keep	[Data bus] T [Other than the above] Keep	[Data bus] D15 to D8 [Other than the above] I/O port
Port E	1, 2, 4 to 6	8-bit bus	T	T	Keep	Keep	I/O port
		16-bit bus	T	T	T	T	D7 to D0
	3, 7	8-bit bus	T	T	Keep	Keep	I/O port
		16-bit bus	T	T	[Data bus] T [Other than the above] Keep	[Data bus] T [Other than the above] Keep	[Data bus] D7 to D0 [Other than the above] I/O port
PF7/ø	1, 2, 4 to 6		Clock output	T	[Clock output] H [Other than the above] Keep	[Clock output] Clock output [Other than the above] Keep	[Clock output] Clock output [Other than the above] Input port
	3, 7		T				

Port Name	MCU Operating Mode*[1]	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PF6/AS	1, 2, 4 to 6	H	T	[OPE = 0, AS output] T [OPE = 1, AS output] H [Other than the above] Keep	[AS output] T [Other than the above] Keep	[AS output] AS [Other than the above] I/O port
	3, 7	T				
PF5/RD PF4/HWR	1, 2, 4 to 6	H	T	[OPE = 0] T [OPE = 1] H	T	RD, HWR
	3, 7	T		[OPE = 0, RD, HWR output] T [OPE = 1, RD, HWR output] H [Other than the above] Keep	[RD, HWR output] T [Other than the above] Keep	[RD, HWR output] RD, HWR [Other than the above] I/O port
PF3/LWR	1, 2, 4 to 6	H	T	[OPE = 0, LWR output] T [OPE = 1, LWR output] H [Other than the above] Keep	[LWR output] T [Other than the above] Keep	[LWR output] LWR [Other than the above] I/O port
	3, 7	T				

Port Name	MCU Operating Mode*[1]	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PF2/LCAS/ DQML*[2]	1 to 7	T	T	[OPE = 0, LCAS (DQML) output] T [OPE = 1, LCAS (DQML) output] H [Other than the above] Keep	[LCAS (DQML) output] T [Other than the above] Keep	[LCAS (DQML) output] LCAS (DQML) [Other than the above] I/O port
PF1/UCAS/ (DQMU)*[2]	1 to 7	T	T	[OPE = 0, UCAS (DQMU) output] T [OPE = 1, UCAS (DQMU) output] H [Other than the above] Keep	[UCAS (DQMU) output] T [Other than the above] Keep	[UCAS (DQMU) output] UCAS [Other than the above] I/O port
PF0/WAIT	1 to 7	T	T	[WAIT input] T [Other than the above] Keep	[WAIT input] T [Other than the above] Keep	[WAIT input] WAIT [Other than the above] I/O port
PG6/BREQ	1 to 7	T	T	[BREQ input] T [Other than the above] Keep	[BREQ input] BREQ	[BREQ input] BREQ [Other than the above] I/O port

Port Name	MCU Operating Mode*[1]	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PG5/BACK	1 to 7	T	T	[BACK output] T [Other than the above] Keep	BACK	[BACK output] BACK [Other than the above] I/O port
PG4/ BREQO	1 to 7	T	T	[BREQO output] T [Other than the above] Keep	[BREQO output] BREQO [Other than the above] Keep	[BREQO output] BREQO [Other than the above] I/O port
PG3/CS3 PG2/CS2 PG1/CS1	1 to 7	T	T	[OPE = 0, CS output] T [OPE = 1, CS output] H [Other than the above] Keep	[CS output] T [Other than the above] Keep	[CS output] CS [Other than the above] I/O port
PG0/CS0	1, 2, 5, 6	H	T	[OPE = 0, CS output] T [OPE = 1, CS output] H [Other than the above] Keep	[CS output] T [Other than the above] Keep	[CS output] CS [Other than the above] I/O port
	3, 4, 7	T				

Port Name	MCU Operating Mode*[1]	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PH3/OE/ CKE/CS7	1 to 7	T	T	[OPE = 0, OE output] T [OPE = 1, OE output] H [OPE = 0, CS output] T [OPE = 1, CS output] H [Other than the above] Keep	[OE output] T [CS output] T [Other than the above] Keep	[OE output] OE [CS output] CS [Other than the above] I/O port
PH2/CS6	1 to 7	T	T	[OPE = 0, CS output] T [OPE = 1, CS output] H [Other than the above] Keep	[CS output] T [Other than the above] Keep	[CS output] CS [Other than the above] I/O port
PH1/CS5/ SDRAMϕ*[2]	1 to 7	[DCTL = 1] Clock output [DCTL = 0] T	[DCTL = 1] L [DCTL = 0] T	[DCTL = 1] L [DCTL = 0, OPE = 0, CS output] T [DCTL = 0, OPE = 1, CS output] H [Other than the above] Keep	[DCTL = 1] Clock output [DCTL = 0, CS output] T [Other than the above] Keep	[DCTL = 1] Clock output [DCTL = 0, CS output] CS [Other than the above] I/O port

Port Name	MCU Operating Mode*[1]	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PH0/CS4	1 to 7	T	T	[OPE = 0, CS output] T [OPE = 1, CS output] H [Other than the above] Keep	[CS output] T [Other than the above] Keep	[CS output] CS [Other than the above] I/O port

Legend:

L: Low level

H: High level

Keep: Input port becomes high-impedance, output port retains state

T: High impedance

DDR: Data direction register

OPE: Output port enable

Notes: 1. Mode 3 is not supported in the H8S/2678 Series.

2. Not available in the H8S/2678 Series.

B. Product Lineup

Product		Type Name	Model Marking	Package (Code)
H8S/2676	Flash memory version	HD64F2676	HD64F2676	144-pin QFP (FP-144G)
	Masked ROM version	HD6432676	HD64F2676(***)	144-pin QFP (FP-144G)
H8S/2675	Masked ROM version	HD6432675	HD6432675(***)	144-pin QFP (FP-144G)
H8S/2674R	ROM-less version	HD6412674R	HD6412674	144-pin LQFP (FP-144H)
H8S/2673	Masked ROM version	HD6432673	HD6432673(***)	144-pin QFP (FP-144G)
H8S/2670	ROM-less version	HD6412670	HD6412670	144-pin QFP (FP-144G)

[Symbols]

(***): ROM code

C. Package Dimensions

For package dimensions, dimensions described in Hitachi Semiconductor Packages have priority.

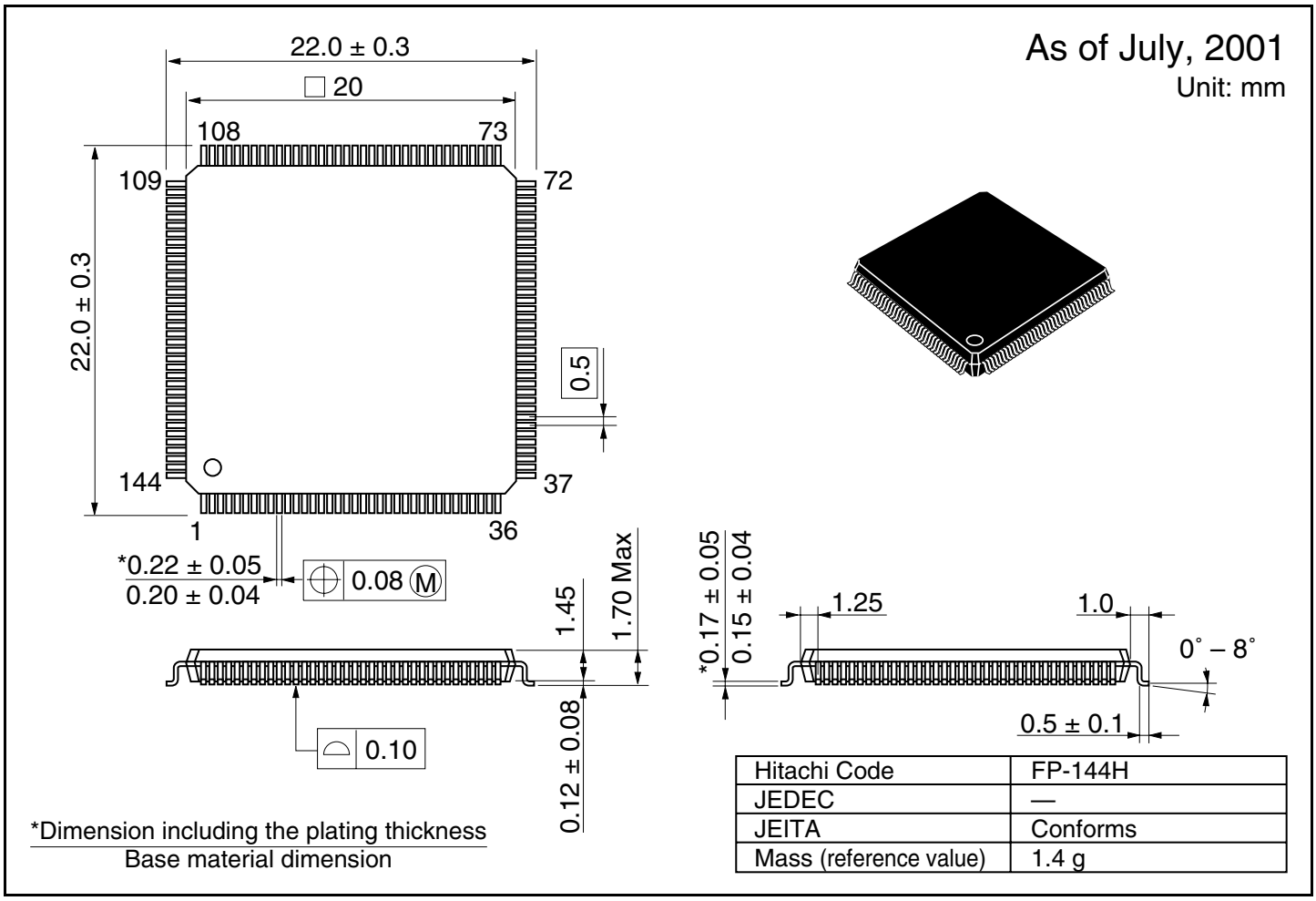


Figure C.1 Package Dimensions (FP-144H)

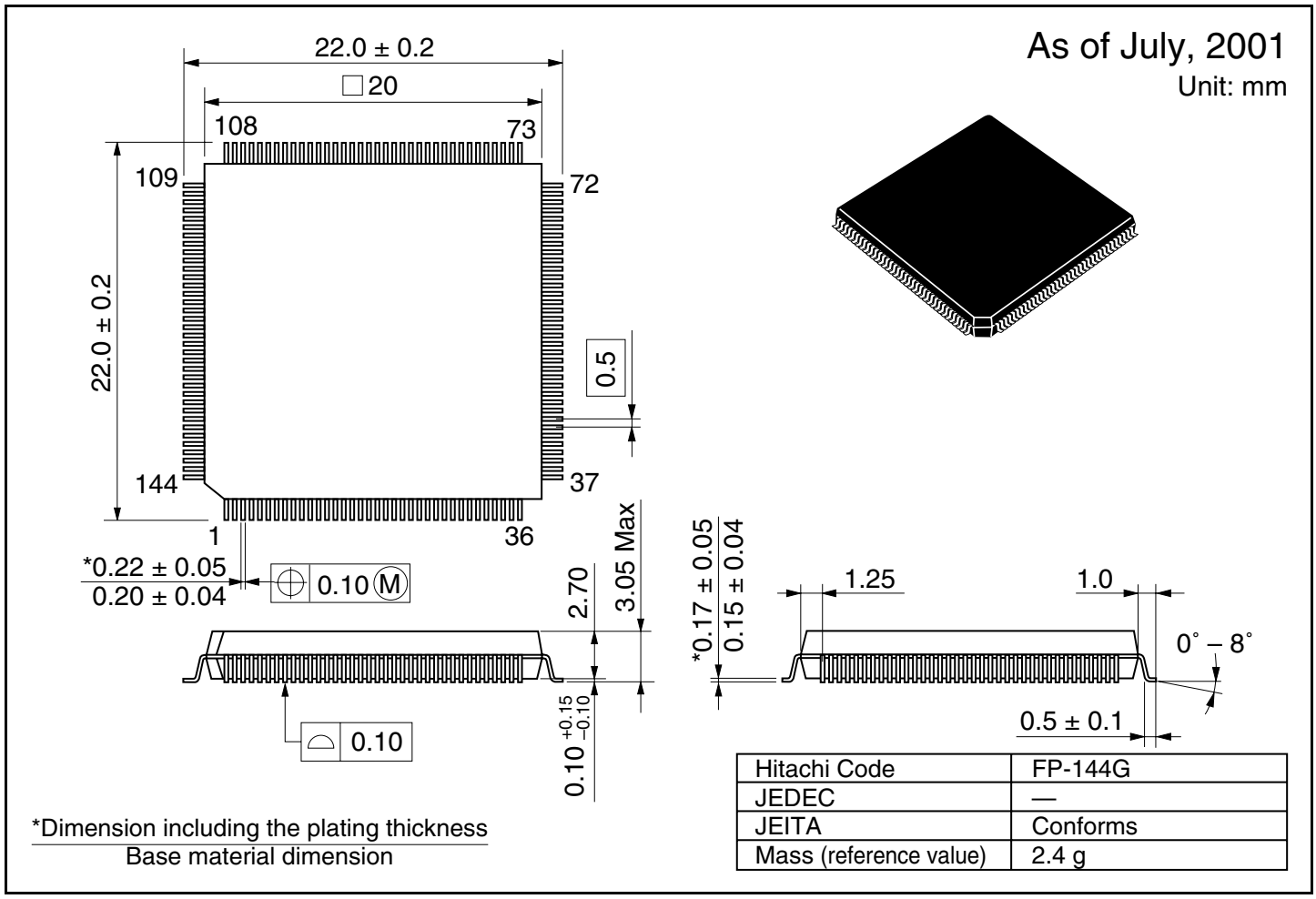


Figure C.2 Package Dimensions (FP-144G)

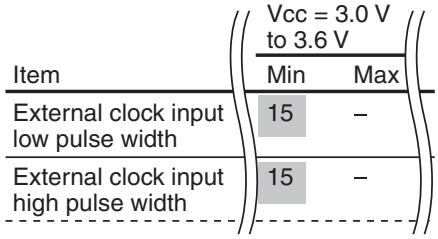
Main Revisions and Additions in this Edition

Item	Page	Revisions (See Manual for Details)
1.1 Features	1	The following product deleted. Model: HD64F2677R
3.4 Memory Map in Each Operating Mode	65	Address map for H8S/2677R deleted.
5.3.1 Interrupt Control Register (INTCR) Bits 7 to 6	88	Bit 7: R/W, Reserved Bit 6: R/W, These bits can be read from or written to. However, the write value should always be 0.
5.3.1 Interrupt Control Register (INTCR) Bits 2 to 0	88	Bits 2 to 0: R/W, Reserved These bits can be read from or written to. However, the write value should always be 0.
5.7.6 Note on IRQ Status Register (ISR)	118	Section 5.7.6 added.
6.3.7 Bus Control Register (BCR)	134	Description changed. Bit 15: External Bus Release Enable
6.3.7 Bus Control Register (BCR)	134	Description changed. Bit 13: This bit can be read from or written to. However, the write value should always be 0.
6.3.8 DRAM Control Register (DRAMCR)	136	Descriptions changed. Bits 13, 11, 3: This bit can be read from or written to. However, the write value should always be 0.
6.3.9 DRAM Access Control Register (DRACCR)	143 to 145	Descriptions changed. H8S/2678 Series Bit 6: This bit can be read from or written to. However, the write value should always be 0. H8S/2678R Series Bits 14, 10, 7 to 4, 2: This bit (These bits) can be read from or written to. However, the write value should always be 0.
Figure 6.5 CAS Latency Control Cycle Disable Timing during Continuous Synchronous DRAM Space Write Access (for CAS Latency 2)	146	Error in figure 6.5 corrected. (Error) DQMU, DQML → (Correction) DQMU, DQML
6.3.10 Refresh Control Register (REFCR)	148	Description changed. Bit 11: This bit can be read from or written to. However, the write value should always be 0.

Item	Page	Revisions (See Manual for Details)
7.3.4 DMA Control Registers (DMACRA and DMACRB) Full Address Mode	266	Description of DMACR changed. Bits 10 to 8, 7, 4: This bit (These bits) can be read from or written to. However, the write value should always be 0.
7.3.4 DMA Control Registers (DMACRA and DMACRB) Full Address Mode DMACR_0B and DMACR_1B	267	Desctiption of bits DTF3 to DTF0 added. 0010: Activated by DREQ pin falling edge input (detected as a low level in the first transfer after transfer is enabled)
7.3.5 DMA Band Control Registers H and L (DMABCRH and DMABCRL) Full Address Mode	271 to 273	Descriptions of DMABCRH changed. Bits 13, 12, 10, 8: This bit (These bits) can be read from or written to. However, the write value should always be 0.
7.3.5 DMA Band Control Registers H and L (DMABCRH and DMABCRL) Full Address Mode	277, 278	Descriptions of bits 3 to 0 in DMABCRL changed. Bit 3: (Error) If the DTIE1B bit is set to 1 when DTME1 = 0, → (Correction) If the DTME1 bit is cleared to 0 when DTIE1B = 1, Bit 2: (Error) If the DTIE1A bit is set to 1 when DTE1 = 0, → (Correction) If the DTE1 bit is cleared to 0 when DTIE1A= 1, Bit 1: (Error) If the DTIE0B bit is set to 1 when DTME0 = 0, → (Correction) If the DTME0 bit is cleared to 0 when DTIE0B= 1, Bit 0: (Error) If the DTIE0A bit is set to 1 when DTE0 = 0, → (Correction) If the DTE0 bit is cleared to 0 when DTIE0A = 1,
7.3.7 DMA Terminal Control Register (DMATCR)	281	Description on DMATCR added.
7.4.1 Activation by Internal Interrupt Request	283	With ADI, TXI, and RXI interrupts,… When an interrupt request signal for DMAC activation is also used for an interrupt request to the CPU or DTC activation (DTA = 0),
7.5.11 Write Data Buffer Function	319	DMAC internal-to-external dual address transfers and single address transfers can be executed… …, dual address transfer external write cycles or single transfer and internal accesses…
8.3.4 EXDMA Mode Control Register (EDMDR)	340	Desription changed. Bits 1, 0: These bits are always read as 0. The initial values should not be modified.
9.2.6 DTC Transfer Count Register B (CRB)	397	Description added. This register is not available in normal and repeat modes.

Item	Page	Revisions (See Manual for Details)
10.1.4 Pin Functions P17/PO15/TIOCB2/TCLKD/EDRAK3	425	<table><tr><td>TPU channel 2 settings</td><td>(2)</td><td>(1)</td><td>(2)</td></tr><tr><td>MD3 to MD0</td><td colspan="3">B'0011</td></tr><tr><td>IOB3 to IOB0</td><td>B'xx00</td><td colspan="2">Other than B'xx00</td></tr></table>
10.1.4 Pin Functions P15/PO13/TIOCB1/TCLKC	426	Notes amended. (Error) TIOCB1 input when MD3 to MD0 = B'0000 or B'01XX and IOB3 = B'10xx. (Correction) TIOCB1 input when MD3 to MD0 = B'0000 or B'01xx and IOB3 to IOB0 = B'10xx.
10.1.4 Pin Functions P14/PO12/TIOCA1	427	Notes amended. (Error) TIOCA1 input when MD3 to MD0 = B'0000, B'000, and B'01xx and IOA3 = B'10xx. (Correction) TIOCA1 input when MD3 to MD0 = B'0000 or B'01xx and IOA3 to IOA0 = B'10xx.
10.2.4 Pin Functions P27/PO7/TIOCB5/(IRQ5)/EDRAK1	434	The values of MD3 to MD0 in the subordinated table amended. (Error) B'0000 to B'0011 → (Correction) B'0000, B'01xx
10.2.4 Pin Functions P26/PO6/TIOCA5/(IRQ14)/EDRAK0	435	The following values in the subordinated table amended. MD3 to MD0: (Error) B'0000 to B'0011 → (Correction) B'0000, B'01xx MD3 to MD0: (Error) B'0010 → (Correction) B'001x CCLR1, CCLR0: (Error) Other than B'10 → (Correction) Other than B'01 CCLR1, CCLR0: (Error) B'10 → (Correction) B'01
10.2.4 Pin Functions P25/PO5/TIOCB4/(IRQ13)	436	The values of MD3 to MD0 in the subordinated table amended. (Error) B'0000 to B'0011 → (Correction) B'0000, B'01xx
10.2.4 Pin Functions P24/PO4/TIOCA4/RxD4/(IRQ12)	437	The following values in the subordinated table amended. MD3 to MD0: (Error) B'0000 to B'0011 → (Correction) B'0000, B'01xx CCLR1, CCLR0: (Error) Other than B'10 → (Correction) Other than B'01 CCLR1, CCLR0: (Error) B'10 → (Correction) B'01
10.2.4 Pin Functions P23/PO3/TIOCD3/TxD4/(IRQ11)	438	The values of MD3 to MD0 in the subordinated table amended. (Error) B'0001 to B'0011 → (Correction) B'0000
10.2.4 Pin Functions P22/PO2/TIOCC3/(IRQ10)	439	The values of MD3 to MD0 in the subordinated table amended. (Error) B'0001 to B'01xx → (Correction) B'0000

Item	Page	Revisions (See Manual for Details)
10.2.4 Pin Functions P21/PO1/TIOCB3/(IRQ9)	440	The values of MD3 to MD0 in the subordinated table amended. (Error) B'0001 to B'0011 → (Correction) B'0000
10.2.4 Pin Functions P20/PO0/TIOCA3/(IRQ8)	441	The values of MD3 to MD0 in the subordinated table amended. (Error) B'0001 to B'01xx → (Correction) B'0000
10.3.6 Pin Functions P35/SCK1/SCL0/(OE)/(CKE*3)	444	(Correction) …bits CKE0 and CKE1 in SCR, bits OEE and RMTS2 to RMTS0 in DRAMCR, bit OES in PFCR2, and bit P35DDR.
10.6.4 Pin Functions P63/TMCI1/TEND1/IRQ11	456	Note added. When used as the external clock input pin for the TMR, its pin function should be specified to the external clock input by the CKS2 to CKS0 bits in TCR_1.
10.6.4 Pin Functions P62/TMCI0/TEND0/IRQ10	456	Note added. When used as the external clock input pin for the TMR, its pin function should be specified to the external clock input by the CKS2 to CKS0 bits in TCR_1.
10.6.4 Pin Functions P61/TMRI1/DREQ1/IRQ9	457	Note added. When used as the counter reset input pin for the TMR, both the CCLR1 and CCLR0 bits in TCR_1 should be set to 1.
10.6.4 Pin Functions P60/TMRI0/DREQ0/IRQ8	457	Note added. When used as the counter reset input pin for the TMR, both the CCLR1 and CCLR0 bits in TCR_1 should be set to 1.
10.14.4 Pin Functions PF0/WAIT	491	(Correction) …bit EXPE, bit WAITE in BCR, and bit PF0DDR.
10.15.5 Pin Functions PG3/CS3/RAS3*/CAS*, PG2/CS2/RAS2*/RAS*	496	<table><tr><td>Operating mode</td><td colspan="9">3*, 7</td></tr><tr><td>PGnDDR</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>–</td><td>–</td><td>–</td></tr><tr><td>Pin function</td><td>PGn input</td><td>PGn output</td><td>PGn input</td><td>PGn output</td><td>PGn input</td><td>$\overline{CSn}$ output</td><td>$\overline{RASn}$ output</td><td>$\overline{CAS}$* output</td><td>$\overline{RAS}$* output</td></tr></table>
14.6.2 Contention between Timer Counter (TCNT) Write and Increment	629	If a timer counter clock pulse is generated during the next cycle after the T2 state of a TCNT write cycle,
16.3.2 I2C Bus Control Register B (ICCRB)	714	<table><tr><td>Bit</td><td>Description</td></tr><tr><td>5</td><td>A/D Start Clearing this bit to 0 stops A/D conversion, and the A/D converter enters wait state. When this bit is set to 1 by software, TPU (trigger), TMR (trigger), or the ADTRG pin, A/D conversion starts. This bit remains set to 1 during A/D conversion. In single mode, cleared to 0 automatically when conversion on the specified channel ends. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by a reset, or a transition to hardware standby mode or software.</td></tr></table>

Item	Page	Revisions (See Manual for Details)
16.4.2 Scan Mode	721	Description added to item 4. If the ADST bit is later set to 1, A/D conversion starts again from the first channel in the group.
Section 18 RAM	739	The following product of the H8S/2678R Series deleted. Product type name: H8S/2677R
19.1 Features	741	The following product of the H8S/2678R Series deleted. Product type name: H8S/2677R
19.5.2 Flash Memory Control Register 2 (FLMCR2)	751	Description added. When the on-chip flash memory is disabled, the contents of FLMCR2 are always read as H'00.
19.6 On-Board Programming Modes	756	Description amended.
19.6.1 Boot Mode	756	Description amended.
19.6.1 Boot Mode Table 19.6 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible	758	Table 19.6 amended. 19,200 bps (Correction) 8 to 25 MHz 9,600 bps (Correction) 8 to 25 MHz 4,800 bps (Correction) deleted
19.8.1 Program/Program-Verify	762	4. Consecutively transfer 128 bytes of data in byte units from the programming data area,
21.1.1 System Clock Control Register (SCKCR)	778	Description of bit 6 amended. This bit can be read from or written to. However, The write value should always be 0.
21.1.2 PLL Control Register (PLLCR)	779	Description of bit 3 amended. This bit can be read from or written to. However, The write value should always be 0.
21.2.2 External Clock Input Table 21.3 External Clock Input Conditions	782	
21.5.2 Notes on Resonator	783	As the parameters for the oscillation circuit will depend on the floating capacitance of the resonator and the user board, the parameters should be determined in consultation with the resonator manufacturer.

Item	Page	Revisions (See Manual for Details)
23.3 Register States in Each Operating Mode	826	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Hardware Standby
P1DDR	–	–	–	–	Initialized
P2DDR	–	–	–	–	Initialized
PAODR	–	–	–	–	Initialized

23.3 Register States in Each Operating Mode	829, 830	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Hardware Standby
PORT1	–	–	–	–	–
PORT2	–	–	–	–	–
PORTG	–	–	–	–	–

23.3 Register States in Each Operating Mode	831, 832	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Hardware Standby
TCR_0	Initialized	–	–	–	Initialized
TCR_1	Initialized	–	–	–	Initialized
TCNT	Initialized	–	–	–	Initialized
RSTCSR	Initialized	–	–	–	–

Section 24 Electrical Characteristics	836	TBD deleted.
24.2 DC Characteristics Table 24.2 DC Characteristics	836, 837	Pin added. Input high voltage: STBY, MD2 to MD0, DCTL*[4] Input low voltage: RES, STBY, MD2 to MD0, DCTL*[4] Input leakage current: STBY, NMI, MD2 to MD0, DCTL*[4]
24.3 AC Characteristics Figure 24.4 (1) Oscillation Stabilization Timing	842	All timing stipulated at 1/2 Vcc.
24.3 AC Characteristics Table 24.8 Bus Timing	848	

Item	Symbol	Min	Max	Unit	Test Conditions
Address delay time 2*	tAD2	–	16.5	ns	Figure 24.26
CS delay time 4*	tCSD4	–	16.5	ns	Figure 24.26
Read data setup time 3*	tRDS3	15	–	ns	Figure 24.26
Read data hold time 3*	tRDH3	0	–	ns	Figure 24.26
Write data delay time 2*	tWDD2	–	31.5	ns	Figure 24.26
Write data hold time 4*	tWDH3	2	–	ns	Figure 24.26

24.3 AC Characteristics Figure 24.26 Synchronous DRAM Basic Access Timing (CAS Latency 2) Figure 24.27 Synchronous DRAM Self-Refresh Timing Figure 24.28 Read Data: Two-State Expansion (CAS Latency 2)	865 to 867	All timing stipulated at 1/2 Vcc.

HITACHI

Index

H8S/2678 Series, H8S/2678R Series Hardware Manual

Publication Date: 1st Edition, September 2001
2nd Edition, April 2002
Published by: Business Operation Division
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
Hitachi Kodaira Semiconductor Co., Ltd.