To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

RENESAS

Notice

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

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

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

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

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

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

User's Manual

32

H8SX/1663Group

Hardware Manual

Renesas 32-Bit CISC
Microcomputer
H8SX Family / H8SX/1600 Series

H8SX/1663	R5F61663
H8SX/1664	R5F61664

Renesas Electronics
www.renesas.com

Rev.1.00 2006.06

RENESAS

are in their open states, intermediate levels are induced by noise in the vicinity, a
through current flows internally, and a malfunction may occur.

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

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

The configuration of the functional description of each module differs according to t 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. Ea includes notes in relation to the descriptions given, and usage notes are given, as require 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 ve This does not include all of the revised contents. For details, see the actual locations in t manual.

11. Index

characteristics of the H8SX/1663 Group to the target users.
Refer to the H8SX Family Software Manual for a detailed description of
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 i on the CPU, system control functions, peripheral functions and electrical characteristi
- In order to understand the details of the CPU's functions

  Read the H8SX Family Software 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 register. The addresses, bits, and initial values of the registers are summarized in secti List of Registers.

Examples:	Register name:	The following notation is used for cases when the similar function, e.g. 16-bit timer pulse unit or seri communication interface, is implemented on more channel: XXX_N (XXX is the register name and N is the c 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
	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 Please ensure you have the latest versions of all documents you requ http://www.renesas.com/

Section 7 DMA Controller (DMAC)

Section 8 Data Transfer Controller (DTC)

Section 11 Programmable Pulse Generator (PPG)

Section 12 8-Bit Timers (TMR)

Section 16 USB Function Module (USB)

Section 17 I²C Bus Interface2 (IIC2)

RENESAS

Appendix

Section 8 Data Transfer Controller (DTC)

Section 9 I/O Ports

Section 10 16-Bit Timer Pulse Unit (TPU)

Section 16 USB Function Module (USB)

Section 17 I²C Bus Interface2 (IIC2)

RENESAS

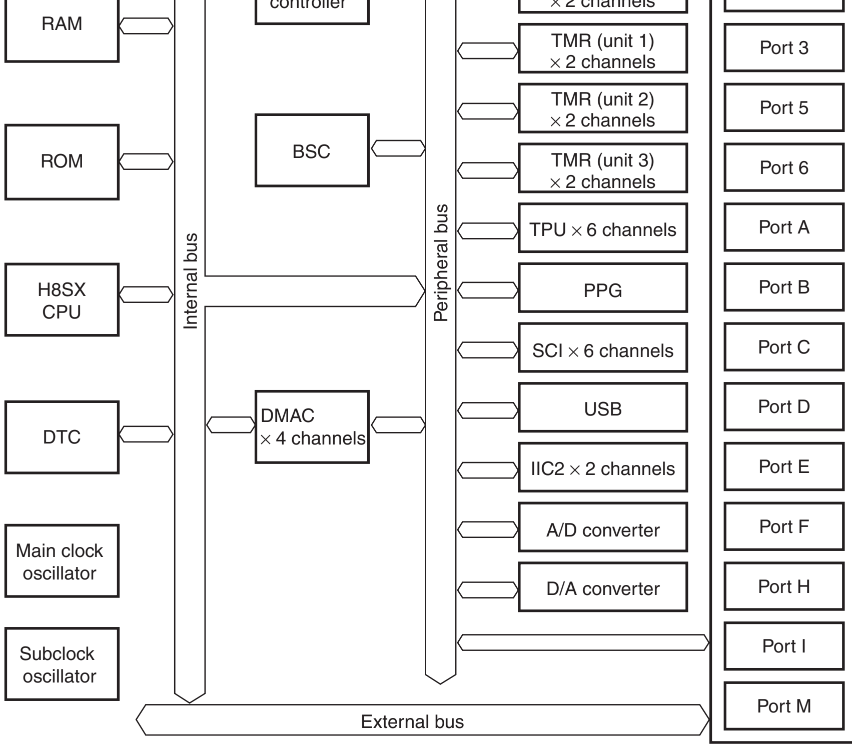
Appendix

- Extensive peripheral functions
  DMA controller (DMAC)
  Data transfer controller (DTC)
  16-bit timer pulse unit (TPU)
  Programmable pulse generator (PPG)
  8-bit timer (TMR)
  Watchdog timer (WDT)
  Serial communication interface (SCI) can be used in asynchronous or clocked synch
  mode
  Universal Serial Bus Interface (USB)
  I^2C bus interface 2 (IIC2)
  10-bit A/D converter
  8-bit D/A converter
  Clock pulse generator

- On-chip memory

Product Classification		Product Model	ROM	RA
Flash memory version	H8SX/1663	R5F61663	384 kbytes	40
	H8SX/1664	R5F61664	512 kbytes	40

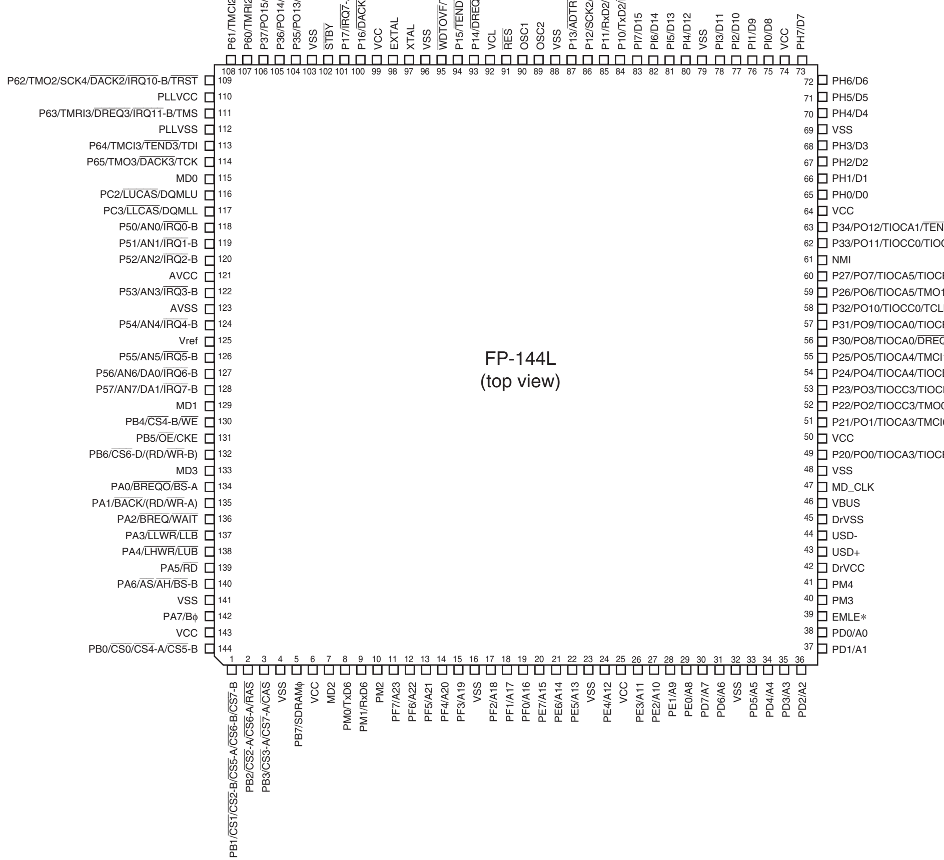
- General I/O port
  92 input/output ports
  Nine input ports
- Supports power-down modes
- Small package



[Legend]

CPU:	Central processing unit	TPU:	16-bit timer pulse unit
DTC:	Data transfer controller	PPG:	Programmable pulse generator
BSC:	Bus controller	SCI:	Serial communication interface
DMAC:	DMA controller	USB:	Universal Serial Bus Interface
WDT:	Watchdog timer	IIC2:	I^{2}C Bus Interface 2
TMR:	8-bit timer	TM32K:	32K timer

Figure 1.1 Block Diagram



Note: * This is an on-chip emulator enable pin. Drive this pin low for the connection in normal operating mode. When this pin is driven high, the on-chip emulation function is enabled and pins P62, P63, P64, P65, and WDTOVF are dedicated to the on-chip emulator. For details on a connection example with the E10A, refer to the E10A Emulator User's Manual.

Figure 1.2 Pin Assignments

5	PB7/SDRAMϕ	PB7/SDRAMϕ
6	VCC	VCC
7	MD2	MD2
8	PM0/TxD6	PM0/TxD6
9	PM1/RxD6	PM1/RxD6
10	PM2	PM2
11	PF7/A23	PF7/A23
12	PF6/A22	PF6/A22
13	PF5/A21	PF5/A21
14	PF4/A20	PF4/A20
15	PF3/A19	PF3/A19
16	VSS	VSS
17	PF2/A18	PF2/A18
18	PF1/A17	PF1/A17
19	PF0/A16	PF0/A16
20	PE7/A15	PE7/A15
21	PE6/A14	PE6/A14
22	PE5/A13	PE5/A13
23	VSS	VSS
24	PE4/A12	PE4/A12
25	VCC	VCC
26	PE3/A11	PE3/A11
27	PE2/A10	PE2/A10
28	PE1/A9	PE1/A9
29	PE0/A8	PE0/A8

37	PD1/A1	PD1/A1
38	PD0/A0	PD0/A0
39	EMLE	EMLE
40	PM3	PM3
41	PM4	PM4
42	DrVCC	DrVCC
43	USD+	USD+
44	USD-	USD-
45	DrVSS	DrVSS
46	VBUS	VBUS
47	MD_CLK	MD_CLK
48	VSS	VSS
49	P20/PO0/TIOCA3/TIOCB3/TMRI0/SCK0/$\overline{\text{IRQ8}}$-A	P20/PO0/TIOCA3/TIOCB3/TMRI0/SC
50	VCC	VCC
51	P21/PO1/TIOCA3/TMCI0/RxD0/$\overline{\text{IRQ9}}$-A	P21/PO1/TIOCA3/TMCI0/RxD0/$\overline{\text{IRQ9}}$-
52	P22/PO2/TIOCC3/TMO0/TxD0/$\overline{\text{IRQ10}}$-A	P22/PO2/TIOCC3/TMO0/TxD0/$\overline{\text{IRQ10}}$
53	P23/PO3/TIOCC3/TIOCD3/$\overline{\text{IRQ11}}$-A	P23/PO3/TIOCC3/TIOCD3/$\overline{\text{IRQ11}}$-A
54	P24/PO4/TIOCA4/TIOCB4/TMRI1/SCK1	P24/PO4/TIOCA4/TIOCB4/TMRI1/SC
55	P25/PO5/TIOCA4/TMCI1/RxD1	P25/PO5/TIOCA4/TMCI1/RxD1
56	P30/PO8/TIOCA0/$\overline{\text{DREQ0}}$-B	P30/PO8/TIOCA0/$\overline{\text{DREQ0}}$-B
57	P31/PO9/TIOCA0/TIOCB0/$\overline{\text{TEND0}}$-B	P31/PO9/TIOCA0/TIOCB0/$\overline{\text{TEND0}}$-B
58	P32/PO10/TIOCC0/TCLKA-A/$\overline{\text{DACK0}}$-B	P32/PO10/TIOCC0/TCLKA-A/$\overline{\text{DACK0}}$-
59	P26/PO6/TIOCA5/TMO1/TxD1	P26/PO6/TIOCA5/TMO1/TxD1
60	P27/PO7/TIOCA5/TIOCB5	P27/PO7/TIOCA5/TIOCB5

68	PH3/D3	PH3/D3
69	VSS	VSS
70	PH4/D4	PH4/D4
71	PH5/D5	PH5/D5
72	PH6/D6	PH6/D6
73	PH7/D7	PH7/D7
74	VCC	VCC
75	PI0/D8	PI0/D8
76	PI1/D9	PI1/D9
77	PI2/D10	PI2/D10
78	PI3/D11	PI3/D11
79	VSS	VSS
80	PI4/D12	PI4/D12
81	PI5/D13	PI5/D13
82	PI6/D14	PI6/D14
83	PI7/D15	PI7/D15
84	P10/TxD2/$\overline{\text{DREQ0}}$-A/$\overline{\text{IRQ0}}$-A	P10/TxD2/$\overline{\text{DREQ0}}$-A/$\overline{\text{IRQ0}}$-A
85	P11/RxD2/$\overline{\text{TEND0}}$-A/$\overline{\text{IRQ1}}$-A	P11/RxD2/$\overline{\text{TEND0}}$-A/$\overline{\text{IRQ1}}$-A
86	P12/SCK2/$\overline{\text{DACK0}}$-A/$\overline{\text{IRQ2}}$-A	P12/SCK2/$\overline{\text{DACK0}}$-A/$\overline{\text{IRQ2}}$-A
87	P13/$\overline{\text{ADTRG0}}$/$\overline{\text{IRQ3}}$-A	P13/$\overline{\text{ADTRG0}}$/$\overline{\text{IRQ3}}$-A
88	VSS	VSS
89	OSC2	OSC2
90	OSC1	OSC1
91	$\overline{\text{RES}}$	$\overline{\text{RES}}$

98	EXTAL	EXTAL
99	VCC	VCC
100	P16/DACK1-A/IRQ6-A/TCLKC-B/SDA0	P16/DACK1-A/IRQ6-A/TCLKC-B/SDA
101	P17/IRQ7-A/TCLKD-B/SCL0	P17/IRQ7-A/TCLKD-B/SCL0
102	STBY	STBY
103	VSS	VSS
104	P35/PO13/TIOCA1/TIOCB1/TCLKC-A/DACK1-B	P35/PO13/TIOCA1/TIOCB1/TCLKC-A
105	P36/PO14/TIOCA2	P36/PO14/TIOCA2
106	P37/PO15/TIOCA2/TIOCB2/TCLKD-A	P37/PO15/TIOCA2/TIOCB2/TCLKD-A
107	P60/TMRI2/TxD4/DREQ2/IRQ8-B	P60/TMRI2/TxD4/DREQ2/IRQ8-B
108	P61/TMCI2/RxD4/TEND2/IRQ9-B	P61/TMCI2/RxD4/TEND2/IRQ9-B
109	P62/TMO2/SCK4/DACK2/IRQ10-B/TRST	P62/TMO2/SCK4/DACK2/IRQ10-B/TR
110	PLLVCC	PLLVCC
111	P63/TMRI3/DREQ3/IRQ11-B/TMS	P63/TMRI3/DREQ3/IRQ11-B/TMS
112	PLLVSS	PLLVSS
113	P64/TMCI3/TEND3/TDI	P64/TMCI3/TEND3/TDI
114	P65/TMO3/DACK3/TCK	P65/TMO3/DACK3/TCK
115	MD0	MD0
116	PC2/LUCAS/DQMLU	PC2/LUCAS/DQMLU
117	PC3/LLCAS/DQMLL	PC3/LLCAS/DQMLL
118	P50/AN0/IRQ0-B	P50/AN0/IRQ0-B
119	P51/AN1/IRQ1-B	P51/AN1/IRQ1-B
120	P52/AN2/IRQ2-B	P52/AN2/IRQ2-B

128	P57/AN7/DA1/IRQ7-B	P57/AN7/DA1/IRQ7-B
129	MD1	MD1
130	PB4/$\overline{CS4}$-B/$\overline{WE}$	PB4/$\overline{CS4}$-B/$\overline{WE}$
131	PB5/$\overline{OE}$/CKE	PB5/$\overline{OE}$/CKE
132	PB6/$\overline{CS6}$-D/(RD/$\overline{WR}$-B)	PB6/$\overline{CS6}$-D/(RD/$\overline{WR}$-B)
133	MD3	MD3
134	PA0/$\overline{BREQO}$/$\overline{BS}$-A	PA0/$\overline{BREQO}$/$\overline{BS}$-A
135	PA1/$\overline{BACK}$/(RD/$\overline{WR}$-A)	PA1/$\overline{BACK}$/(RD/$\overline{WR}$-A)
136	PA2/$\overline{BREQ}$/$\overline{WAIT}$	PA2/$\overline{BREQ}$/$\overline{WAIT}$
137	PA3/$\overline{LLWR}$/$\overline{LLB}$	PA3/$\overline{LLWR}$/$\overline{LLB}$
138	PA4/$\overline{LHWR}$/$\overline{LUB}$	PA4/$\overline{LHWR}$/$\overline{LUB}$
139	PA5/$\overline{RD}$	PA5/$\overline{RD}$
140	PA6/$\overline{AS}$/$\overline{AH}$/$\overline{BS}$-B	PA6/$\overline{AS}$/$\overline{AH}$/$\overline{BS}$-B
141	VSS	VSS
142	PA7/Bϕ	PA7/Bϕ
143	VCC	VCC
144	PB0/$\overline{CS0}$/$\overline{CS4}$-A/$\overline{CS5}$-B	PB0/$\overline{CS0}$/$\overline{CS4}$-A/$\overline{CS5}$-B

				pin.)
	V_{ss}	4, 16, 23, 32, 48, 69, 79, 88, 96, 103, 141	Input	Ground pins. Connect to the system supply (0 V).
	$PLLV_{cc}$	110	Input	Power supply pin for the PLL circuit to the system power supply.
	$PLLV_{ss}$	112	Input	Ground pin for the PLL circuits.
	DrVCC	42	Input	Power supply pin for USB on-chip tr Connect to the system power suppl
	DrVSS	45	Input	Ground pin for USB on-chip transce
Clock	XTAL	97	Input	Pins for a crystal resonator. Externa be input to the EXTAL pin. For a co example, see section 22, Clock Puls Generator.
	EXTAL	98	Input	
	OSC1	90	Input	Connects the 32.768-kHz crystal re
	OSC2	89	Input	Connect the 32.768-kHz crystal res
	Bϕ	142	Output	Outputs the system clock for extern
	SDRAMϕ	5	Output	Connects to the CLK pin of synchro DRAM when synchronous DRAM is connected. For details, see section Controller (BSC).
Operating mode control	MD3 MD2 MD1 MD0	133 7 129 115	Input	Pins for setting the operating mode. levels of these pins must not be cha during operation.
	MD_CLK	47	Input	Pin for changing the multiplication ra clock pulse generator. The signal le pin must not be changed during ope

emulator	TMS	111	Input	driven high, these pins are dedicated on-chip emulator.
	TDI	113	Input	
	TCK	114	Input	
	TDO	95	Output	
Address bus	A23 A22 A21 A20 A19 A18 A17 A16 A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0	11 12 13 14 15 17 18 19 20 21 22 24 26 27 28 29 30 31 33 34 35 36 37 38	Output	Output pins for the addresses.

	D4	70		
	D3	68		
	D2	67		
	D1	66		
	D0	65		
Bus control	$\overline{\text{BREQ}}$	136	Input	External bus masters request the bu signal.
	$\overline{\text{BREQO}}$	134	Output	The internal bus masters request th access the external space in the ex released state.
	$\overline{\text{BACK}}$	135	Output	Bus acknowledge signal which indic the bus has been released.
	$\overline{\text{BS}}$-A/$\overline{\text{BS}}$-B	134/140	Output	Indicates the start of a bus cycle.
	$\overline{\text{AS}}$	140	Output	Strobe signal which indicates that th address on the address bus is valid accessing the basic bus interface o control SRAM interface space.
	$\overline{\text{AH}}$	140	Output	This signal is used to hold the addre accessing the address/data multiple interface space.
	$\overline{\text{RD}}$	139	Output	Strobe signal to indicates that the b interface space is being read from.
	RD/$\overline{\text{WR}}$-A/RD/$\overline{\text{WR}}$-B	135/132	Output	Indicates the direction (input/output) data bus.
	$\overline{\text{LHWR}}$	138	Output	Strobe signal which indicates that th byte (D15 to D8) is valid when acce basic bus interface space.
	$\overline{\text{LLWR}}$	137	Output	Strobe signal which indicates that th byte (D7 to D0) is valid when acces basic bus interface space.

$\overline{CS3}$-A $\overline{CS4}$-A/$\overline{CS4}$-B $\overline{CS5}$-A/$\overline{CS5}$-B $\overline{CS6}$-A/$\overline{CS6}$-B/$\overline{CS6}$-D $\overline{CS7}$-A/$\overline{CS7}$-B	3 144/130 1/144 2/1/132 3/1		
$\overline{WAIT}$	136	Input	Requests wait cycles when accessin external space.
$\overline{RAS}$	2	Output	• Row address strobe signal for D when area 2 is specified as DRA interface space. • Row address strobe signal when specified as synchronous DRAM space.
$\overline{CAS}$	3	Output	Column address strobe signal when specified as synchronous DRAM inte space.
$\overline{WE}$	130	Output	• Write enable signal for DRAM sp • Write enable signal when area 2 specified as synchronous DRAM space.
$\overline{OE}$/CKE	131	Output	• Output enable signal for DRAM i space. • Clock enable signal for synchron DRAM interface space.

				• Lower-data mask enable signal synchronous DRAM interface s • Data mask enable signal for 8-b synchronous DRAM interface s
Interrupt	NMI	61	Input	Non-maskable interrupt request sig this pin is not in use, this signal mus high.
	$\overline{IRQ11}$-A/$\overline{IRQ11}$-B $\overline{IRQ10}$-A/$\overline{IRQ10}$-B $\overline{IRQ9}$-A/$\overline{IRQ9}$-B $\overline{IRQ8}$-A/$\overline{IRQ8}$-B $\overline{IRQ7}$-A/$\overline{IRQ7}$-B $\overline{IRQ6}$-A/$\overline{IRQ6}$-B $\overline{IRQ5}$-A/$\overline{IRQ5}$-B $\overline{IRQ4}$-A/$\overline{IRQ4}$-B $\overline{IRQ3}$-A/$\overline{IRQ3}$-B $\overline{IRQ2}$-A/$\overline{IRQ2}$-B $\overline{IRQ1}$-A/$\overline{IRQ1}$-B $\overline{IRQ0}$-A/$\overline{IRQ0}$-B	53/111 52/109 51/108 49/107 101/128 100/127 94/126 93/124 87/122 86/120 85/119 84/118	Input	Maskable interrupt request signal.
DMA controller (DMAC)	$\overline{DREQ0}$-A/$\overline{DREQ0}$-B $\overline{DREQ1}$-A/$\overline{DREQ1}$-B $\overline{DREQ2}$ $\overline{DREQ3}$	84/56 93/62 107 111	Input	Requests DMAC activation.
	$\overline{DACK0}$-A/$\overline{DACK0}$-B $\overline{DACK1}$-A/$\overline{DACK1}$-B $\overline{DACK2}$ $\overline{DACK3}$	86/58 100/104 109 114	Output	DMAC single address transfer ackn signal.
	$\overline{TEND0}$-A/$\overline{TEND0}$-B $\overline{TEND1}$-A/$\overline{TEND1}$-B $\overline{TEND2}$ $\overline{TEND3}$	85/57 94/63 108 113	Output	Indicates DMAC data transfer end.

				compare outputs/PWM outputs.
	TIOCA2 TIOCB2	105, 106 106	Input/ output	Signals for TGRA_2 and TGRB_2. Th used for the input capture inputs/outp compare outputs/PWM outputs.
	TIOCA3 TIOCB3 TIOCC3 TIOCD3	49, 51 49 52, 53 53	Input/ output	Signals for TGRA_3 and TGRB_3. Th used for the input capture inputs/outp compare outputs/PWM outputs.
	TIOCA4 TIOCB4	54, 55 54	Input/ output	Signals for TGRA_4 and TGRB_4. Th used for the input capture inputs/outp compare outputs/PWM outputs.
	TIOCA5 TIOCB5	59, 60 60	Input/ output	Signals for TGRA_5 and TGRB_5. Th used for the input capture inputs/outp compare outputs/PWM outputs.
Programmable pulse generator (PPG)	PO15 PO14 PO13 PO12 PO11 PO10 PO9 PO8 PO7 PO6 PO5 PO4 PO3 PO2 PO1 PO0	106 105 104 63 62 58 57 56 60 59 55 54 53 52 51 49	Output	Output pins for the pulse signals.

	TMRI1 TMRI2 TMRI3	[illegible] 107 111		
Watchdog timer (WDT)	$\overline{\text{WDTOVF}}$	95	Output	Output pin for the counter overflow watchdog timer mode.
Serial communication interface (SCI)	TxD0 TxD1 TxD2 TxD4 TxD5 TxD6	52 59 84 107 93 8	Output	Output pins for transmit data.
	RxD0 RxD1 RxD2 RxD4 RxD5 RxD6	51 55 85 108 94 9	Input	Input pins for receive data.
	SCK0 SCK1 SCK2 SCK4	49 54 86 109	Input/ output	Input/output pins for clock signals.
SCI with IrDA (SCI)	IrTxD	93	Output	Output pin that outputs decoded da
	IrRxD	94	Input	Input pin that inputs decoded data f

	AN6	127		converter.
	AN5	126		
	AN4	124		
	AN3	122		
	AN2	120		
	AN1	119		
	AN0	118		
	$\overline{\text{ADTRG0}}$	87	Input	Input pin for the external trigger signa A/D conversion.
D/A converter	DA1 DA0	128 127	Output	Output pins for the analog signals for converter.
A/D converter, D/A converter	AV_{cc}	121	Input	Analog power supply pin for the A/D converters. When the A/D and D/A co are not in use, connect to the system supply.
	AV_{ss}	123	Input	Ground pin for the A/D and D/A conv Connect to the system power supply
	Vref	125	Input	Reference power supply pin for the A D/A converters. When the A/D and D converters are not in use, connect to system power supply.

P25	55		
P24	54		
P23	53		
P22	52		
P21	51		
P20	49		
P37	106	Input/ output	8-bit input/output pins.
P36	105		
P35	104		
P34	63		
P33	62		
P32	58		
P31	57		
P30	56		
P57	128	Input	8-bit input pins.
P56	127		
P55	126		
P54	124		
P53	122		
P52	120		
P51	119		
P50	118		
P65	114	Input/ output	6-bit input/output pins.
P64	113		
P63	111		
P62	109		
P61	108		
P60	107		

	PB6	132	output	
	PB5	131		
	PB4	130		
	PB3	3		
	PB2	2		
	PB1	1		
	PB0	144		
	PC3	117	Input/	2-bit input/output pins.
	PC2	116	output	
	PD7	30	Input/	8-bit input/output pins.
	PD6	31	output	
	PD5	33		
	PD4	34		
	PD3	35		
	PD2	36		
	PD1	37		
	PD0	38		
	PE7	20	Input/	8-bit input/output pins.
	PE6	21	output	
	PE5	22		
	PE4	24		
	PE3	26		
	PE2	27		
	PE1	28		
	PE0	29		

PH5	71		
PH4	70		
PH3	68		
PH2	67		
PH1	66		
PH0	65		
PI7	83	Input/ output	8-bit input/output pins.
PI6	82		
PI5	81		
PI4	80		
PI3	78		
PI2	77		
PI1	76		
PI0	75		
PM4	41	Input/ output	5-bit input/output pins.
PM3	40		
PM2	10		
PM1	9		
PM0	8		

- Upward-compatible with H8/300, H8/300H, and H8S CPUs
  — Can execute H8/300, H8/300H, and H8S/2000 object programs
- Sixteen 16-bit general registers
  — Also usable as sixteen 8-bit registers or eight 32-bit registers
- 87 basic instructions
  — 8/16/32-bit arithmetic and logic instructions
  — Multiply and divide instructions
  — Bit field transfer instructions
  — Powerful bit-manipulation instructions
  — Bit condition branch instructions
  — Multiply-and-accumulate instruction
- Eleven addressing modes
  — Register direct [Rn]
  — Register indirect [@ERn]
  — Register indirect with displacement [@(d:2,ERn), @(d:16,ERn), or @(d:32,ERn)
  — Index register indirect with displacement [@(d:16,RnL.B), @(d:32,RnL.B), @(d:16,Rn.W), @(d:32,Rn.W), @(d:16,ERn.L), or @(d:32,ERn.L)]
  — Register indirect with pre-/post-increment or pre-/post-decrement [@+ERn, @–E @ERn+, or @ERn–]
  — Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  — Immediate [#xx:3, #xx:4, #xx:8, #xx:16, or #xx:32]
  — Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  — Program-counter relative with index register [@(RnL.B,PC), @(Rn.W,PC), or @(ERn.L,PC)]
  — Memory indirect [@@aa:8]
  — Extended memory indirect [@@vec:7]

— 8 × 8-bit register-register multiply: 1 state

— 16 ÷ 8-bit register-register divide: 10 states

— 16 × 16-bit register-register multiply: 1 state

— 32 ÷ 16-bit register-register divide: 18 states

— 32 × 32-bit register-register multiply: 5 states

— 32 ÷ 32-bit register-register divide: 18 states

- Four CPU operating modes
  — Normal mode
  — Middle mode
  — Advanced mode
  — Maximum mode
- Power-down modes
  — Transition is made by execution of SLEEP instruction
  — Choice of CPU operating clocks

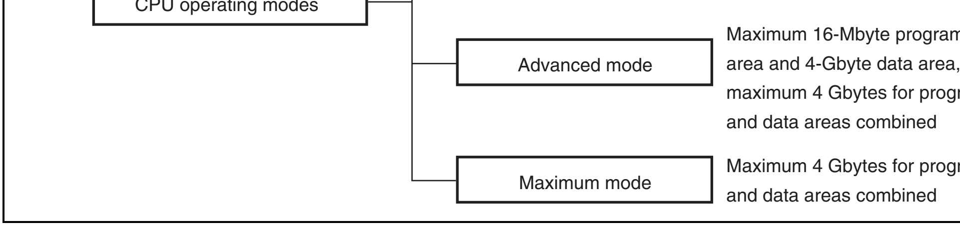


Figure 2.1 CPU Operating Modes

2.2.1 Normal Mode

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

Note: Normal mode is not supported in this LSI.

- Address Space

  The maximum address space of 64 kbytes can be accessed.
- Extended Registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16- segments of 32-bit registers. When the extended register En is used as a 16-bit regist contain any value, even when the corresponding general register Rn is used as an ad register. (If the general register Rn is referenced in the register indirect addressing m pre-/post-increment or pre-/post-decrement and a carry or borrow occurs, however, t the corresponding extended register En will be affected.)
- Instruction Set

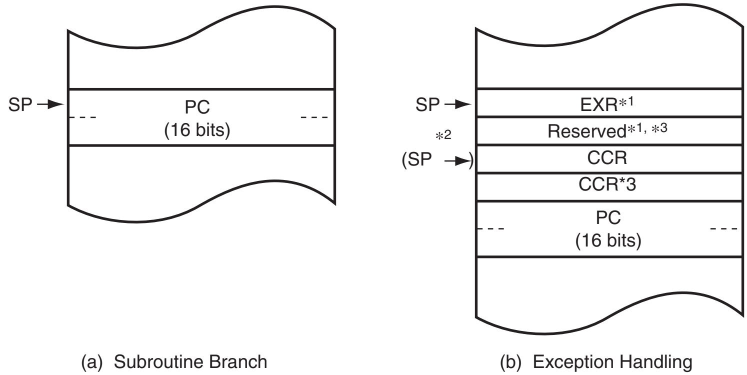
  All instructions and addressing modes can be used. Only the lower 16 bits of effectiv addresses (EA) are valid.

Figure 2.2   Exception Vector Table (Normal Mode)

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressin
are used in the JMP and JSR instructions. An 8-bit absolute address included in the in
code specifies a memory location. Execution branches to the contents of the memory

- Stack Structure

  The stack structure of PC at a subroutine branch and that of PC and CCR at an except
  handling are shown in figure 2.3. The PC contents are saved or restored in 16-bit units



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

Figure 2.3   Stack Structure (Normal Mode)

The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-
segments of 32-bit registers. When the extended register En is used as a 16-bit regist
other than the JMP and JSR instructions), it can contain any value even when the
corresponding general register Rn is used as an address register. (If the general regis
referenced in the register indirect addressing mode with pre-/post-increment or pre-/
decrement and a carry or borrow occurs, however, the value in the corresponding ex
register En will be affected.)

- Instruction Set

  All instructions and addressing modes can be used. Only the lower 16 bits of effecti
  addresses (EA) are valid and the upper eight bits are sign-extended.

- Exception Vector Table and Memory Indirect Branch Addresses

  In middle mode, the top area starting at H'000000 is allocated to the exception vecto
  One branch address is stored per 32 bits. The upper eight bits are ignored and the lo
  are stored. The structure of the exception vector table is shown in figure 2.4.

  The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressi
  are used in the JMP and JSR instructions. An 8-bit absolute address included in the i
  code specifies a memory location. Execution branches to the contents of the memory

  In middle mode, an operand is a 32-bit (longword) operand, providing a 32-bit branc
  The upper eight bits are reserved and assumed to be H'00.

- Stack Structure

  The stack structure of PC at a subroutine branch and that of PC and CCR at an excep
  handling are shown in figure 2.5. The PC contents are saved or restored in 24-bit uni

- 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 ve table. One branch address is stored per 32 bits. The upper eight bits are ignored and th 24 bits are stored. The structure of the exception vector table is shown in figure 2.4.

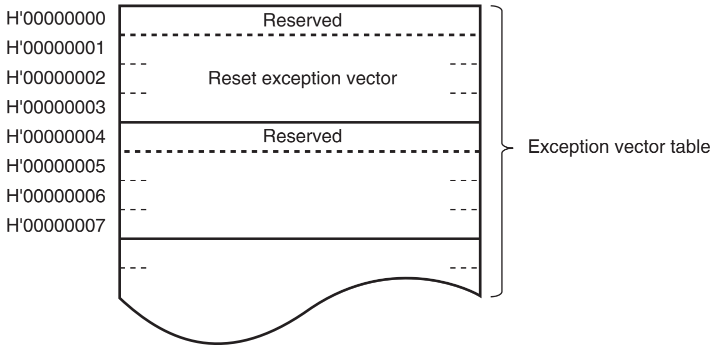


Figure 2.4 Exception Vector Table (Middle and Advanced Modes)

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressin are used in the JMP and JSR instructions. An 8-bit absolute address included in the in code specifies a memory location. Execution branches to the contents of the memory

In advanced mode, an operand is a 32-bit (longword) operand, providing a 32-bit bran address. The upper eight bits are reserved and assumed to be H'00.

(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 on return.

Figure 2.5 Stack Structure (Middle and Advanced Modes)

2.2.4 Maximum Mode

The program area is extended to 4 Gbytes as compared with that in advanced mode.

- Address Space

  The maximum address space of 4 Gbytes can be linearly accessed.
- Extended Registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers or as the upper 16-b 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 maximum mode, the top area starting at H'00000000 is allocated to the exception table. One branch address is stored per 32 bits. The structure of the exception vector shown in figure 2.6.

Figure 2.6 Exception Vector Table (Maximum Modes)

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressin are used in the JMP and JSR instructions. An 8-bit absolute address included in the in code specifies a memory location. Execution branches to the contents of the memory

In maximum mode, an operand is a 32-bit (longword) operand, providing a 32-bit bra address.

- Stack Structure

The stack structure of PC at a subroutine branch and that of PC and CCR at an except handling are shown in figure 2.7. The PC contents are saved or restored in 32-bit units EXR contents are saved or restored regardless of whether or not EXR is in use.

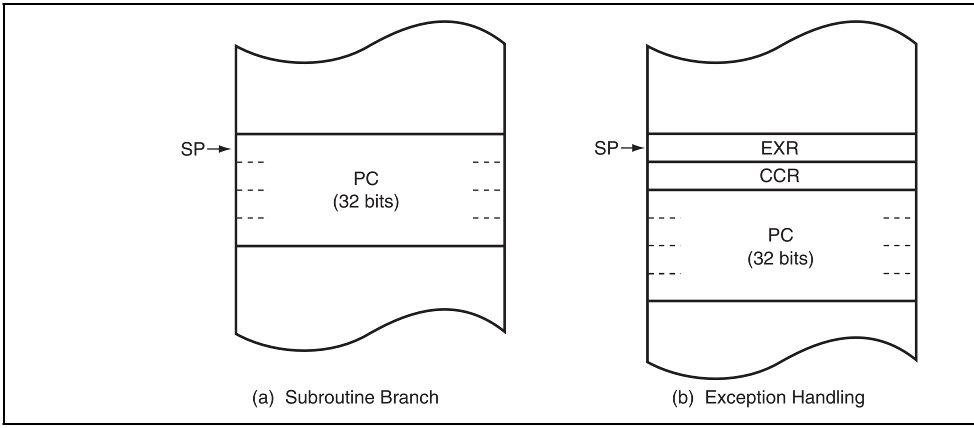


Figure 2.7 Stack Structure (Maximum Mode)

Figure 2.8 shows a memory map of the H8SX CPU. The address space differs dependin
CPU operating mode.

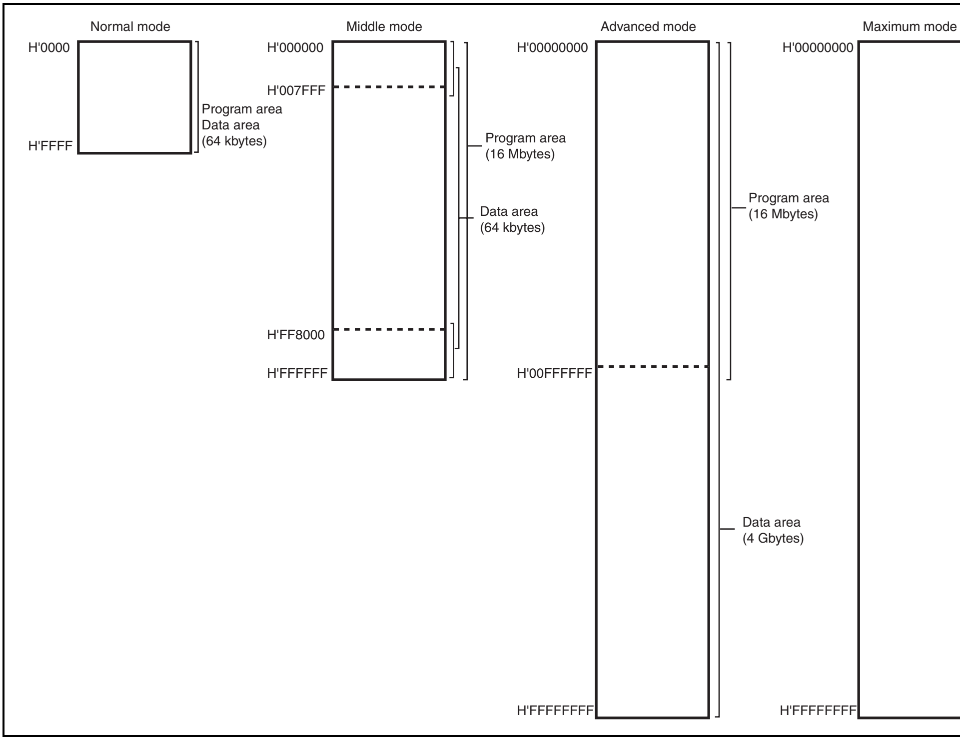


Figure 2.8 Memory Map

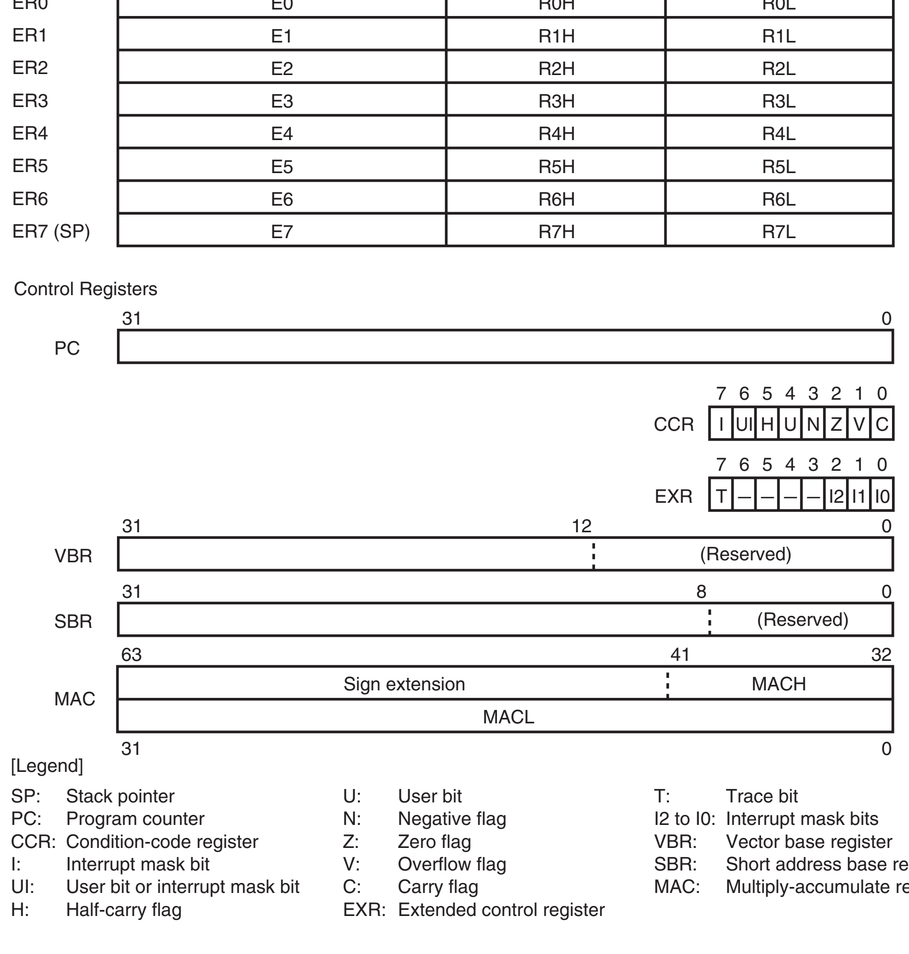


Figure 2.9 CPU Registers

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

When the general registers are used as 8-bit registers, the R registers are divided into 8-b
registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These regist
functionally equivalent, providing a maximum sixteen 8-bit registers.

The general registers ER (ER0 to ER7), R (R0 to R7), and RL (R0L to R7L) are also use
registers. The size in the operand field determines which register is selected.

The usage of each register can be selected independently.

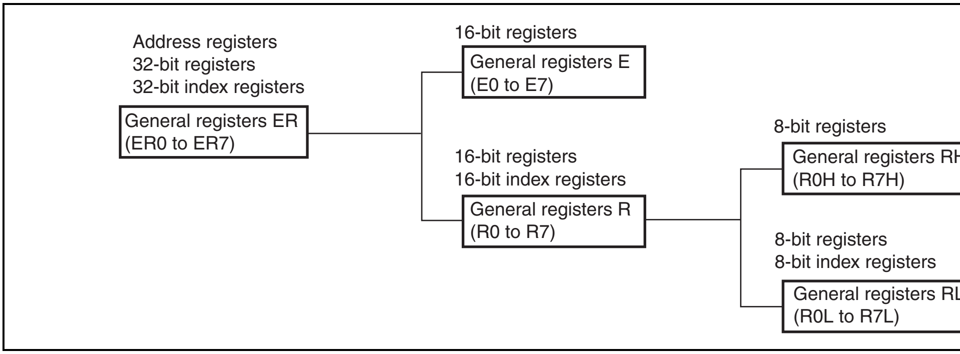


Figure 2.10 Usage of General Registers

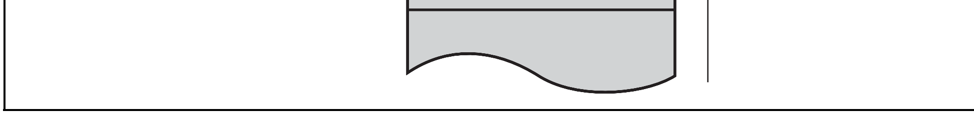


Figure 2.11 Stack

2.5.2 Program Counter (PC)

PC is a 32-bit counter that indicates the address of the next instruction the CPU will exec length of all CPU instructions is 16 bits (one word) or a multiple of 16 bits, so the least si bit is ignored. (When the instruction code is fetched, the least significant bit is regarded a

Bit	Bit Name	Value	R/W	Description
7	I	1	R/W	Interrupt Mask Bit Masks interrupts when set to 1. This bit is set start of an exception handling.
6	UI	Undefined	R/W	User Bit Can be written to and read from by software u LDC, STC, ANDC, ORC, and XORC instructi
5	H	Undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUBX.B, or NEG.B instruction is executed, this flag is s there is a carry or borrow at bit 3, and cleared otherwise. When the ADD.W, SUB.W, CMP.W NEG.W instruction is executed, this flag is se there is a carry or borrow at bit 11, and cleare otherwise. When the ADD.L, SUB.L, CMP.L, instruction is executed, this flag is set to 1 if th carry or borrow at bit 27, and cleared to 0 oth
4	U	Undefined	R/W	User Bit Can be written to and read from by software u LDC, STC, ANDC, ORC, and XORC instructi
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit (re sign bit) of data.

otherwise. A carry has the following types:

- Carry from the result of addition
- Borrow from the result of subtraction
- Carry from the result of shift or rotation

The carry flag is also used as a bit accumulato
manipulation instructions.

2.5.4 Extended Control Register (EXR)

EXR is an 8-bit register that contains the trace bit (T) and three interrupt mask bits (I2 to

Operations can be performed on the EXR bits by the LDC, STC, ANDC, ORC, and XOR
instructions.

For details, see section 4, Exception Handling.

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 g each time an instruction is executed. When thi cleared to 0, instructions are executed in sequ
6 to 3	—	All 1	R/W	Reserved These bits are always read as 1.
2	I2	1	R/W	Interrupt Mask Bits
1	I1	1	R/W	These bits designate the interrupt mask level (
0	I0	1	R/W	

8-bit absolute address addressing mode (@aa:8), this register is used as the upper addres
initial value is H'FFFFFF00. The SBR contents are changed with the LDC and STC inst

2.5.7 Multiply-Accumulate Register (MAC)

MAC is a 64-bit register that stores the results of multiply-and-accumulate operations. It of two 32-bit registers denoted MACH and MACL. The lower 10 bits of MACH are val upper bits are sign extended. The MAC contents are changed with the MAC, CLRMAC and STMAC instructions.

2.5.8 Initial Values of CPU Registers

Reset exception handling loads the start address from the vector table into the PC, clears in EXR to 0, and sets the I bits in CCR and EXR to 1. The general registers, MAC, and bits in CCR are not initialized. In particular, the initial value of the stack pointer (ER7) i undefined. The SP should therefore be initialized using an MOV.L instruction executed immediately after a reset.

Figure 2.12 shows the data formats in general registers.

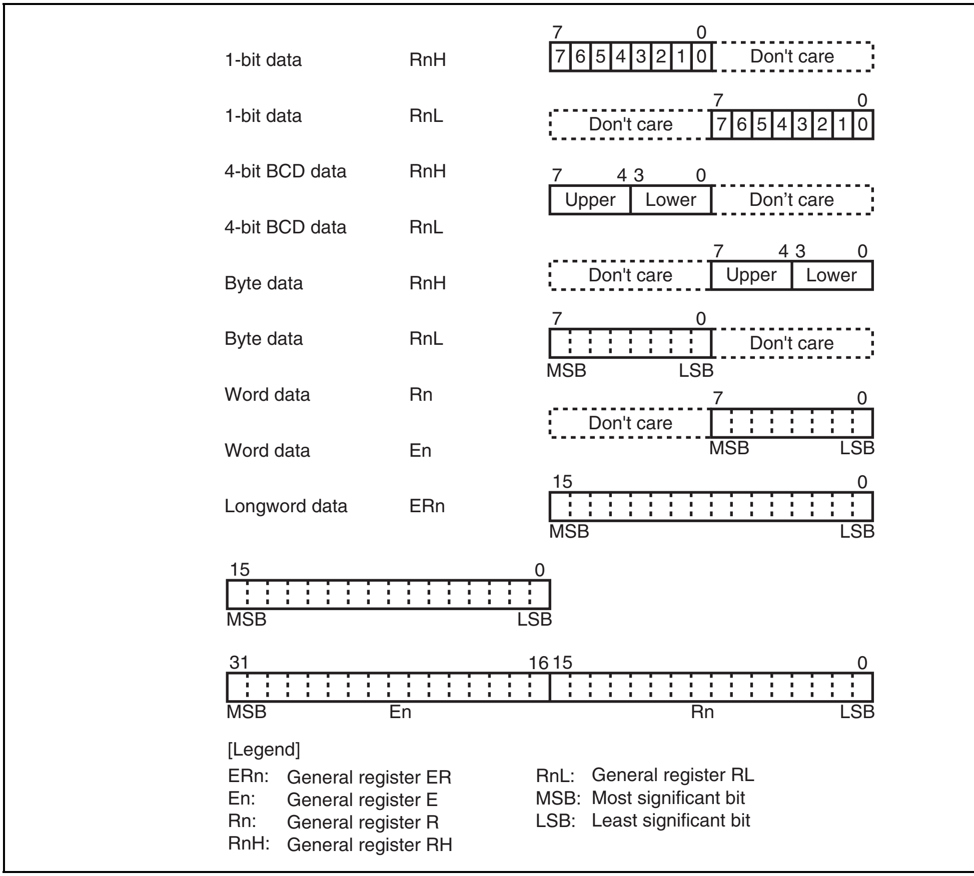


Figure 2.12 General Register Data Formats

the stack manipulation, branch table manipulation, block transfer instructions, and MAC instruction should be located to even addresses.

When SP (ER7) is used as an address register to access the stack, the operand size shoul size or longword size.

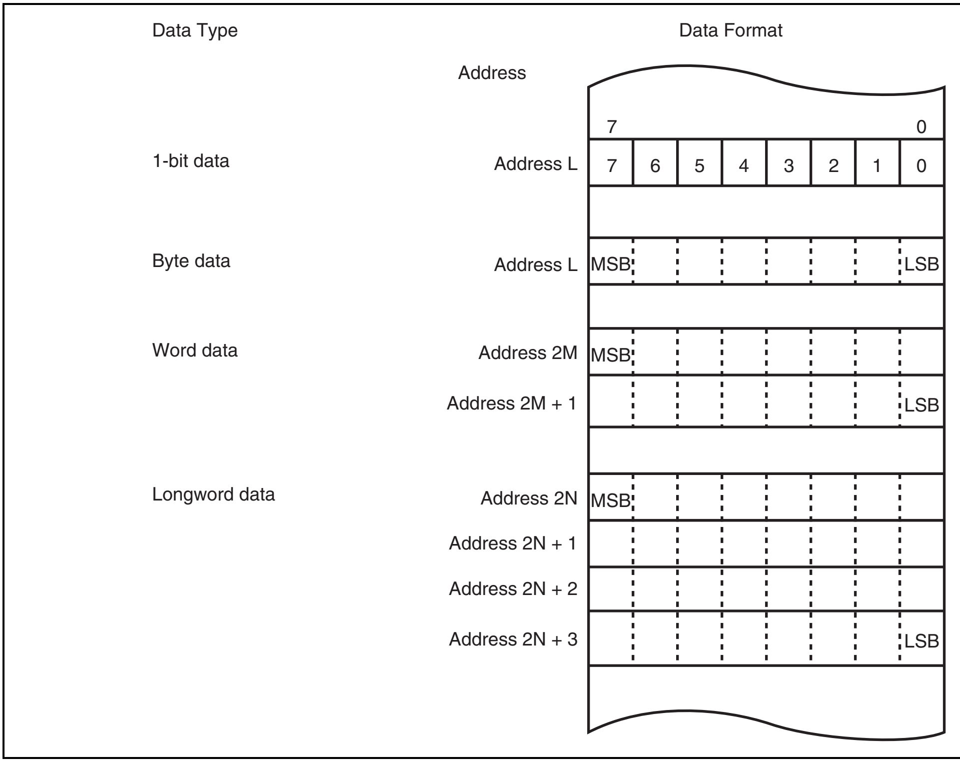


Figure 2.13 Memory Data Formats

	POP, PUSH*[1]	W/L
	LDM, STM	L
	MOVA	B/W*[2]
Block transfer	EEPMOV	B
	MOVMD	B/W/L
	MOVSD	B
Arithmetic operations	ADD, ADDX, SUB, SUBX, CMP, NEG, INC, DEC	B/W/L
	DAA, DAS	B
	ADDS, SUBS	L
	MULXU, DIVXU, MULXS, DIVXS	B/W
	MULU, DIVU, MULS, DIVS	W/L
	MULU/U, MULS/U	L
	EXTU, EXTS	W/L
	TAS	B
	MAC	—
	LDMAC, STMAC	—
	CLRMAC	—
Logic operations	AND, OR, XOR, NOT	B/W/L
Shift	SHLL, SHLR, SHAL, SHAR, ROTL, ROTR, ROTXL, ROTXR	B/W/L
Bit manipulation	BSET, BCLR, BNOT, BTST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST	B
	BSET/EQ, BSET/NE, BCLR/EQ, BCLR/NE, BSTZ, BISTZ	B
	BFLD, BFST	B

[Legend]

B: Byte size

W: Word size

L: Longword size

Notes: 1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W
@−SP.
POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.I
@−SP.
2. Size of data to be added with a displacement
3. Size of data to specify a branch condition
4. Bcc is the generic designation of a conditional branch instruction.
5. Size of general register to be restored
6. Not available in this LSI.

Data transfer	MOV	B/W/L	S	SD	SD	SD	SD	SD		SD
		B		S/D					S/D	
	MOVFPE, MOVTPE*[12]	B		S/D						S/D*
	POP, PUSH	W/L		S/D				S/D*[2]		
	LDM, STM	L		S/D				S/D*[2]		
	MOVA*[4]	B/W		S	S	S	S	S		S
Block transfer	EEPMOV	B								
	MOVMD	B/W/L								
	MOVSD	B								
Arithmetic operations	ADD, CMP	B	S	D	D	D	D	D	D	D
		B		S	D	D	D	D	D	D
		B		D	S	S	S	S	S	S
		B			SD	SD	SD	SD		SD
		W/L	S	SD	SD	SD	SD	SD		SD
	SUB	B	S		D	D	D	D	D	D
		B		S	D	D	D	D	D	D
		B		D	S	S	S	S	S	S
		B			SD	SD	SD	SD		SD
		W/L	S	SD	SD	SD	SD	SD		SD
	ADDX, SUBX	B/W/L	S	SD						
		B/W/L	S		SD					
		B/W/L	S					SD*[5]		
	INC, DEC	B/W/L		D						
	ADDS, SUBS	L		D						
	DAA, DAS	B		D						
	MULXU, DIVXU	B/W	S:4	SD						
	MULU, DIVU	W/L	S:4	SD						

	MAC	—								
	CLRMAC	—								
	LDMAC	—		S						
	STMAC	—		D						
Logic operations	AND, OR, XOR	B		S	D	D	D	D	D	D
		B		D	S	S	S	S	S	S
		B			SD	SD	SD	SD		SD
		W/L	S	SD	SD	SD	SD	SD		SD
	NOT	B		D	D	D	D	D	D	D
		W/L		D	D	D	D	D		D
Shift	SHLL, SHLR	B		D	D	D	D	D	D	D
		B/W/L*6		D	D	D	D	D		D
		B/W/L*7		D						
	SHAL, SHAR ROTL, ROTR ROTXL, ROTXR	B		D	D	D	D	D	D	D
		W/L		D	D	D	D	D		D
Bit manipu-lation	BSET, BCLR, BNOT, BTST, BSET/cc, BCLR/cc	B		D	D				D	D
	BAND, BIAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST, BSTZ, BISTZ	B		D	D				D	D

LDC (VBR, SBR)	L		S				
STC (CCR, EXR)	B/W*[9]		D	D	D	D*[11]	D
STC (VBR, SBR)	L		D				
ANDC, ORC, XORC	B	S					
SLEEP	—						
NOP	—						

[Legend]

d: d:16 or d:32

S: Can be specified as a source operand.

D: Can be specified as a destination operand.

SD: Can be specified as either a source or destination operand or both.

S/D: Can be specified as either a source or destination operand.

S:4: 4-bit immediate data can be specified as a source operand.

Notes: 1. Only @aa:16 is available.

2. @ERn+ as a source operand and @−ERn as a destination operand
3. Specified by ER5 as a source address and ER6 as a destination address for d transfer.
4. Size of data to be added with a displacement
5. Only @ERn− is available
6. When the number of bits to be shifted is 1, 2, 4, 8, or 16
7. When the number of bits to be shifted is specified by 5-bit immediate data or a register
8. Size of data to specify a branch condition
9. Byte when immediate or register direct, otherwise, word
10. Only @ERn+ is available
11. Only @−ERn is available
12. Not available in this LSI.

	Bcc	—		O					
	BRA	—		O	O				
	BRA/S	—		O*					
	JMP	—	O			O	O	O	O
	BSR	—		O					
	JSR	—	O			O	O	O	O
	RTS, RTS/L	—							
System control	TRAPA	—							
	RTE, RTE/L	—							

[Legend]

d: d:8 or d:16

Note: * Only @(d:8, PC) is available.

ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
VBR	Vector base register
SBR	Short address base 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
~	Logical 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 regis
to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

		Saves general register contents on the stack.
LDM	L	@SP+ → Rn (register list) Restores the data from the stack to multiple general registers. T or four general registers which have serial register numbers car specified.
STM	L	Rn (register list) → @−SP Saves the contents of multiple general registers on the stack. Tv or four general registers which have serial register numbers car specified.
MOVA	B/W	EA → Rd Zero-extends and shifts the contents of a specified general regis memory data and adds them with a displacement. The result is general register.

Note: Not available in this LSI.

MOVMD.W	W	Transfers a data block. Transfers word data which begins at a memory location specified to a memory location specified by ER6. The number of word data transferred is specified by R4.
MOVMD.L	L	Transfers a data block. Transfers longword data which begins at a memory location spec ER5 to a memory location specified by ER6. The number of long data to be transferred is specified by R4.
MOVSD.B	B	Transfers a data block with zero data detection. Transfers byte data which begins at a memory location specified to a memory location specified by ER6. The number of byte data transferred is specified by R4. When zero data is detected during the transfer stops and execution branches to a specified address

INC DEC	B/W/L	$Rd \pm 1 \rightarrow Rd$, $Rd \pm 2 \rightarrow Rd$ Increments or decrements a general register by 1 or 2. (Byte op can be incremented or decremented by 1 only.)
ADDS SUBS	L	$Rd \pm 1 \rightarrow Rd$, $Rd \pm 2 \rightarrow Rd$, $Rd \pm 4 \rightarrow Rd$ Adds or subtracts the value 1, 2, or 4 to or from data in a genera
DAA DAS	B	Rd (decimal adjust) $\rightarrow$ Rd Decimal-adjusts an addition or subtraction result in a general re referring to the CCR to produce 2-digit 4-bit BCD data.
MULXU	B/W	$Rd \times Rs \rightarrow Rd$ Performs unsigned multiplication on data in two general register bits $\times$ 8 bits $\rightarrow$ 16 bits, or 16 bits $\times$ 16 bits $\rightarrow$ 32 bits.
MULU	W/L	$Rd \times Rs \rightarrow Rd$ Performs unsigned multiplication on data in two general register bits $\times$ 8 bits $\rightarrow$ 16 bits, or 16 bits $\times$ 16 bits $\rightarrow$ 32 bits.
MULU/U	L	$Rd \times Rs \rightarrow Rd$ Performs unsigned multiplication on data in two general register $\times$ 32 bits $\rightarrow$ upper 32 bits).
MULXS	B/W	$Rd \times Rs \rightarrow Rd$ Performs signed multiplication on data in two general registers: bits $\times$ 8 bits $\rightarrow$ 16 bits, or 16 bits $\times$ 16 bits $\rightarrow$ 32 bits.
MULS	W/L	$Rd \times Rs \rightarrow Rd$ Performs signed multiplication on data in two general registers: bits $\times$ 16 bits $\rightarrow$ 16 bits, or 32 bits $\times$ 32 bits $\rightarrow$ 32 bits.
MULS/U	L	$Rd \times Rs \rightarrow Rd$ Performs signed multiplication on data in two general registers (32 bits $\rightarrow$ upper 32 bits).
DIVXU	B/W	$Rd \div Rs \rightarrow Rd$ Performs unsigned division on data in two general registers: eith $\div$ 8 bits $\rightarrow$ 8-bit quotient and 8-bit remainder, or 32 bits $\div$ 16 bits quotient and 16-bit remainder.

			Compares data between immediate data, general registers, and and stores the result in CCR.
NEG	B/W/L	0 – (EAd) → (EAd)	Takes the two's complement (arithmetic complement) of data in a register or the contents of a memory location.
EXTU	W/L	(EAd) (zero extension) → (EAd)	Performs zero-extension on the lower 8 or 16 bits of data in a ge register or memory to word or longword size. The lower 8 bits to word or longword, or the lower 16 bits to long be zero-extended.
EXTS	W/L	(EAd) (sign extension) → (EAd)	Performs sign-extension on the lower 8 or 16 bits of data in a ge register or memory to word or longword size. The lower 8 bits to word or longword, or the lower 16 bits to long be sign-extended.
TAS	B	@ERd – 0, 1 → (<bit 7> of @EAd)	Tests memory contents, and sets the most significant bit (bit 7) to
MAC	—	(EAs) × (EAd) + MAC → MAC	Performs signed multiplication on memory contents and adds the MAC.
CLRMAC	—	0 → MAC	Clears MAC to zero.
LDMAC	—	Rs → MAC	Loads data from a general register to MAC.
STMAC	—	MAC → Rd	Stores data from MAC to a general register.

		Performs a logical exclusive OR operation on data between imm data, general registers, and memory.
NOT	B/W/L	~ (EAd) → (EAd) Takes the one's complement of the contents of a general registe memory location.

Table 2.8 Shift Operation Instructions

Instruction	Size	Function
SHLL SHLR	B/W/L	(EAd) (shift) → (EAd) Performs a logical shift on the contents of a general register or a location. The contents of a general register or a memory location can be 1, 2, 4, 8, or 16 bits. The contents of a general register can be s any bits. In this case, the number of bits is specified by 5-bit imm data or the lower 5 bits of the contents of a general register.
SHAL SHAR	B/W/L	(EAd) (shift) → (EAd) Performs an arithmetic shift on the contents of a general registe memory location. 1-bit or 2-bit shift is possible.
ROTL ROTR	B/W/L	(EAd) (rotate) → (EAd) Rotates the contents of a general register or a memory location 1-bit or 2-bit rotation is possible.
ROTXL ROTXR	B/W/L	(EAd) (rotate) → (EAd) Rotates the contents of a general register or a memory location carry bit. 1-bit or 2-bit rotation is possible.

BCLR	B	0 → (<bit-No.> of <EAd>) Clears a specified bit in the contents of a general register or a me location to 0. The bit number is specified by 3-bit immediate data lower three bits of a general register.
BCLR/cc	B	if cc, 0 → (<bit-No.> of <EAd>) If the specified condition is satisfied, this instruction clears a spec in a memory location to 0. The bit number can be specified by 3- immediate data, or by the lower three bits of a general register. T status can be specified as a condition.
BNOT	B	~ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in the contents of a general register or a m location. The bit number is specified by 3-bit immediate data or th three bits of a general register.
BTST	B	~ (<bit-No.> of <EAd>) → Z Tests a specified bit in the contents of a general register or a me location and sets or clears the Z flag accordingly. The bit number specified by 3-bit immediate data or the lower three bits of a gene register.
BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the carry flag with a specified bit in the contents of a gener register or a memory location and stores the result in the carry fla bit number is specified by 3-bit immediate data.
BIAND	B	C ∧ [~ (<bit-No.> of <EAd>)] → C ANDs the carry flag with the inverse of a specified bit in the conte general register or a memory location and stores the result in the 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 the contents of a genera or a memory location and stores the result in the carry flag. The b number is specified by 3-bit immediate data.

		Exclusive-ORs the carry flag with the inverse of a specified bit i contents of a general register or a memory location and stores t in the carry flag. The bit number is specified by 3-bit immediate
BLD	B	(<bit-No.> of <EAd>) → C Transfers a specified bit in the contents of a general register or location to the carry flag. The bit number is specified by 3-bit im data.
BILD	B	~ (<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in the contents of a gene register or a memory location to the carry flag. The bit number i by 3-bit immediate data.
BST	B	C → (<bit-No.> of <EAd>) Transfers the carry flag value to a specified bit in the contents o general register or a memory location. The bit number is specifi immediate data.
BSTZ	B	Z → (<bit-No.> of <EAd>) Transfers the zero flag value to a specified bit in the contents of memory location. The bit number is specified by 3-bit immediate
BIST	B	~ C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in contents of a general register or a memory location. The bit nun specified by 3-bit immediate data.

field in memory location contents.

Table 2.10 Branch Instructions

Instruction	Size	Function
BRA/BS BRA/BC	B	Tests a specified bit in memory location contents. If the specified condition is satisfied, execution branches to a specified address.
BSR/BS BSR/BC	B	Tests a specified bit in memory location contents. If the specified condition is satisfied, execution branches to a subroutine at a spe address.
Bcc	—	Branches to a specified address if the specified condition is satis
BRA/S	—	Branches unconditionally to a specified address after executing t instruction. The next instruction should be a 1-word instruction ex the block transfer and branch instructions.
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.
RTS/L	—	Returns from a subroutine, restoring data from the stack to multip general registers.

		location to CCR or EXR.
		Although CCR and EXR are 8-bit registers, word-size transfers performed between them and memory. The upper 8 bits are val
	L	Rs → VBR, Rs → SBR Transfers the general register contents to VBR or SBR.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers the contents of CCR or EXR to a general register or m Although CCR and EXR are 8-bit registers, word-size transfers performed between them and memory. The upper 8 bits are val
	L	VBR → Rd, SBR → Rd Transfers the contents of VBR or SBR to a general register.
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 immedi
NOP	—	PC + 2 → PC Only increments the program counter.

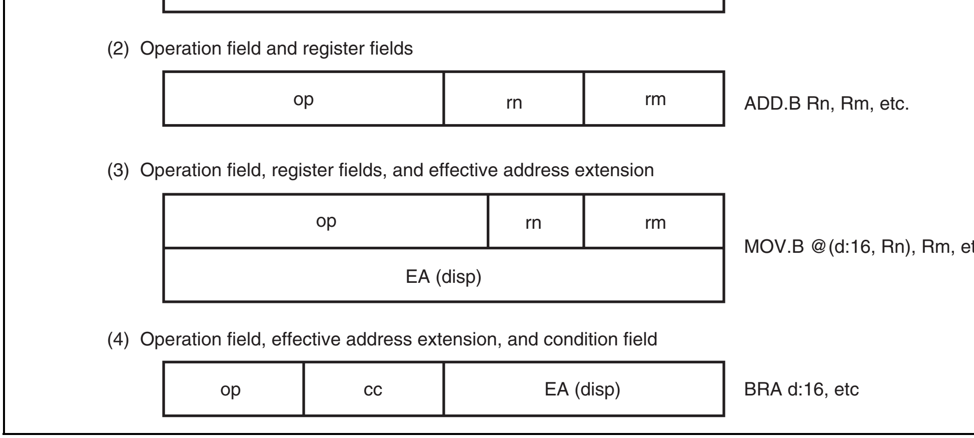


Figure 2.14 Instruction Formats

- Operation Field

  Indicates the function of the instruction, and specifies the addressing mode and operat carried out on the operand. The operation field always includes the first four bits of th instruction. Some instructions have two operation fields.

- Register Field

  Specifies a general register. Address registers are specified by 3 bits, data registers by 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 branch condition of Bcc instructions.

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:2,ERn)/@(d:16,ERn)/@(d:32,E
4	Index register indirect with displacement	@(d:16, RnL.B)/@(d:16,Rn.W)/@(d: @(d:32, RnL.B)/@(d:32,Rn.W)/@(d:
5	Register indirect with post-increment	@ERn+
	Register indirect with pre-decrement	@−ERn
	Register indirect with pre-increment	@+ERn
	Register indirect with post-decrement	@ERn−
6	Absolute address	@aa:8/@aa:16/@aa:24/@aa:32
7	Immediate	#xx:3/#xx:4/#xx:8/#xx:16/#xx:32
8	Program-counter relative	@(d:8,PC)/@(d:16,PC)
9	Program-counter relative with index register	@(RnL.B,PC)/@(Rn.W,PC)/@(ERn.
10	Memory indirect	@@aa:8
11	Extended memory indirect	@@vec:7

2.8.1 Register Direct—Rn

The operand value is the contents of an 8-, 16-, or 32-bit general register which is specif register field in the instruction code.

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.

The operand value is the contents of a memory location which is pointed to by the sum of contents of an address register (ERn) and a 16- or 32-bit displacement. ERn is specified b register field of the instruction code. The displacement is included in the instruction code 16-bit displacement is sign-extended when added to ERn.

This addressing mode has a short format (@(d:2, ERn)). The short format can be used wh displacement is 1, 2, or 3 and the operand is byte data, when the displacement is 2, 4, or 6 operand is word data, or when the displacement is 4, 8, or 12 and the operand is longword

2.8.4 Index Register Indirect with Displacement—@(d:16,RnL.B), @(d:32,RnL. @(d:16,Rn.W), @(d:32,Rn.W), @(d:16,ERn.L), or @(d:32,ERn.L)

The operand value is the contents of a memory location which is pointed to by the sum of following operation result and a 16- or 32-bit displacement: a specified bits of the conten address register (RnL, Rn, ERn) specified by the register field in the instruction code are extended to 32-bit data and multiplied by 1, 2, or 4. The displacement is included in the i code and the 16-bit displacement is sign-extended when added to ERn. If the operand is b ERn is multiplied by 1. If the operand is word or longword data, ERn is multiplied by 2 o respectively.

The operand value is the contents of a memory location which is pointed to by the fo
operation result: the value 1, 2, or 4 is subtracted from the contents of an address reg
(ERn). ERn is specified by the register field of the instruction code. After that, the op
value is stored in the address register. The value subtracted is 1 for byte access, 2 for
access, or 4 for longword access.

- Register indirect with pre-increment—@+ERn

  The operand value is the contents of a memory location which is pointed to by the fo
  operation result: the value 1, 2, or 4 is added to the contents of an address register (E
  is specified by the register field of the instruction code. After that, the operand value
  in the address register. The value added is 1 for byte access, 2 for word access, or 4 f
  longword access.

- Register indirect with post-decrement—@ERn–

  The operand value is the contents of a memory location which is pointed to by the co
  an address register (ERn). ERn is specified by the register field of the instruction coc
  the memory location is accessed, 1, 2, or 4 is subtracted from the address register cor
  the remainder is stored in the address register. The value subtracted is 1 for byte acce
  word access, or 4 for longword access.

using this addressing mode, data to be written is the contents of the general register after
calculating an effective address. If the same general register is specified in an instruction
effective addresses are calculated, the contents of the general register after the first calcu
an effective address is used in the second calculation of an effective address.

Example 1:

MOV.W R0, @ER0+
When ER0 before execution is H'12345678, H'567A is written at H'12345678.

There are 8-bit (@aa:8), 16-bit (@aa:16), 24-bit (@aa:24), and 32-bit (@aa:32) absolute addresses.

To access the data area, the absolute address of 8 bits (@aa:8), 16 bits (@aa:16), or 32 bi (@aa:32) is used. For an 8-bit absolute address, the upper 24 bits are specified by SBR. F bit absolute address, the upper 16 bits are sign-extended. A 32-bit absolute address can ac entire address space.

To access the program area, the absolute address of 24 bits (@aa:24) or 32 bits (@aa:32) For a 24-bit absolute address, the upper 8 bits are all assumed to be 0 (H'00).

Table 2.13 shows the accessible absolute address ranges.

Table 2.13 Absolute Address Access Ranges

Absolute Address		Normal Mode	Middle Mode	Advanced Mode	Maximu Mode
Data area	8 bits (@aa:8)	A consecutive 256-byte area (the upper address is set in SBI			
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF	H'00000000 to H'0000 H'FFFF8000 to H'FFFF	
	32 bits (@aa:32)			H'00000000 to H'FFFF	
Program area	24 bits (@aa:24)		H'000000 to H'FFFFFF	H'00000000 to H'00FF	
	32 bits (@aa:32)			H'00000000 to H'00FFFFFF	H'0000 H'FFFF

manipulation instructions contain 3-bit immediate data in the instruction code, for specif number. The BFLD and BFST instructions contain 8-bit immediate data in the instructio for specifying a bit field. The TRAPA instruction contains 2-bit immediate data in the in code, for specifying a vector address.

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

This mode is used in the Bcc and BSR instructions. The operand value is a 32-bit branch which is the sum of an 8- or 16-bit displacement in the instruction code and the 32-bit ad the PC contents. The 8-bit or 16-bit displacement is sign-extended to 32 bits when added contents. The PC contents to which the displacement is added is the address of the first b next instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 word –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The res value should be an even number. In advanced mode, only the lower 24 bits of this branc are valid; the upper 8 bits are all assumed to be 0 (H'00).

2.8.9 Program-Counter Relative with Index Register—@(RnL.B, PC), @(Rn.W or @(ERn.L, PC)

This mode is used in the Bcc and BSR instructions. The operand value is a 32-bit branch which is the sum of the following operation result and the 32-bit address of the PC conte contents of an address register specified by the register field in the instruction code (RnI ERn) is zero-extended and multiplied by 2. The PC contents to which the displacement i the address of the first byte of the next instruction. In advanced mode, only the lower 24 this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00).

advanced mode, the first byte of the longword-size data is assumed to be all 0 (H'00).

Note that the top part of the address range is also used as the exception handling vector ar
vector address of an exception handling other than a reset or a CPU address error can be c
by VBR.

Figure 2.15 shows an example of specification of a branch address using this addressing r

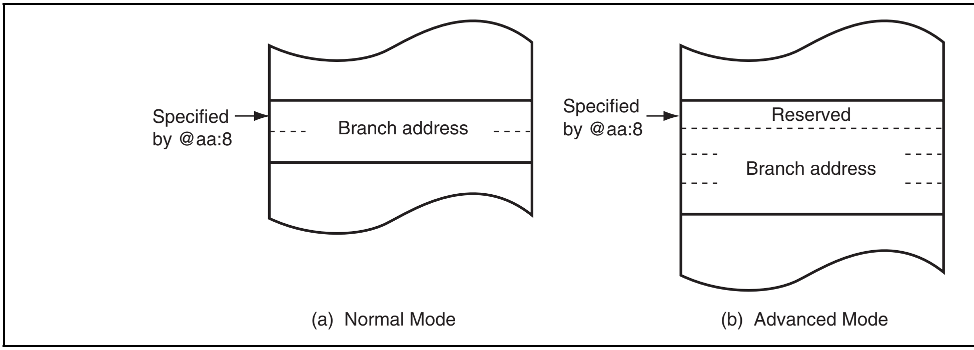


Figure 2.15 Branch Address Specification in Memory Indirect Mode

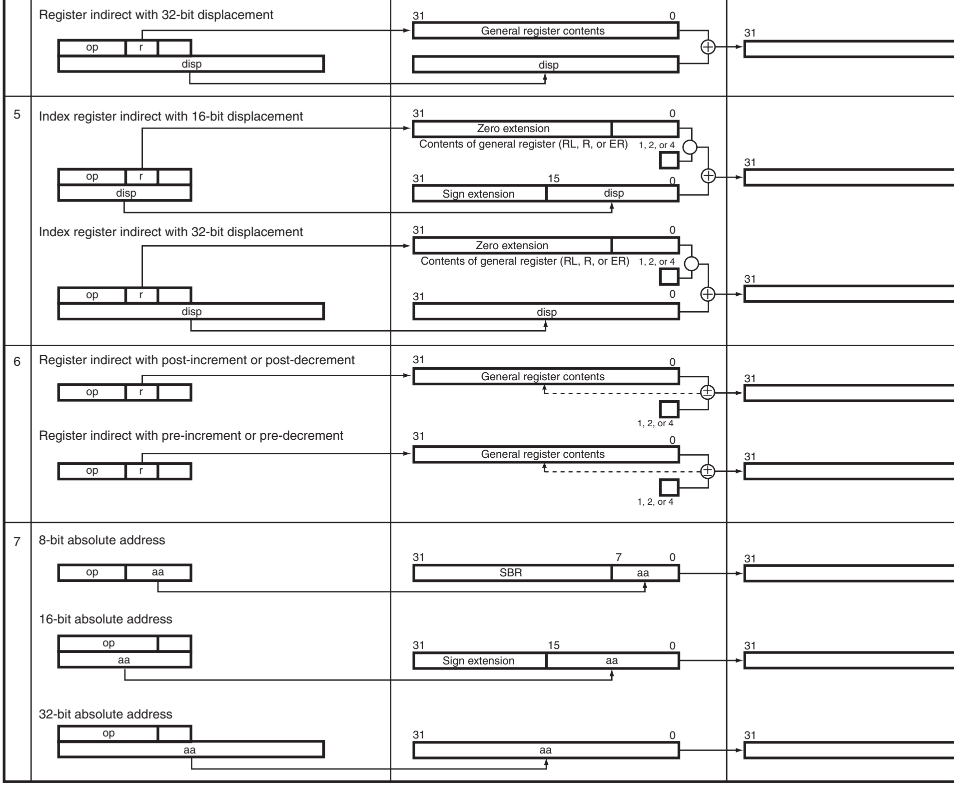
advanced mode, the first byte of the longword-size data is assumed to be all 0 (H'00).

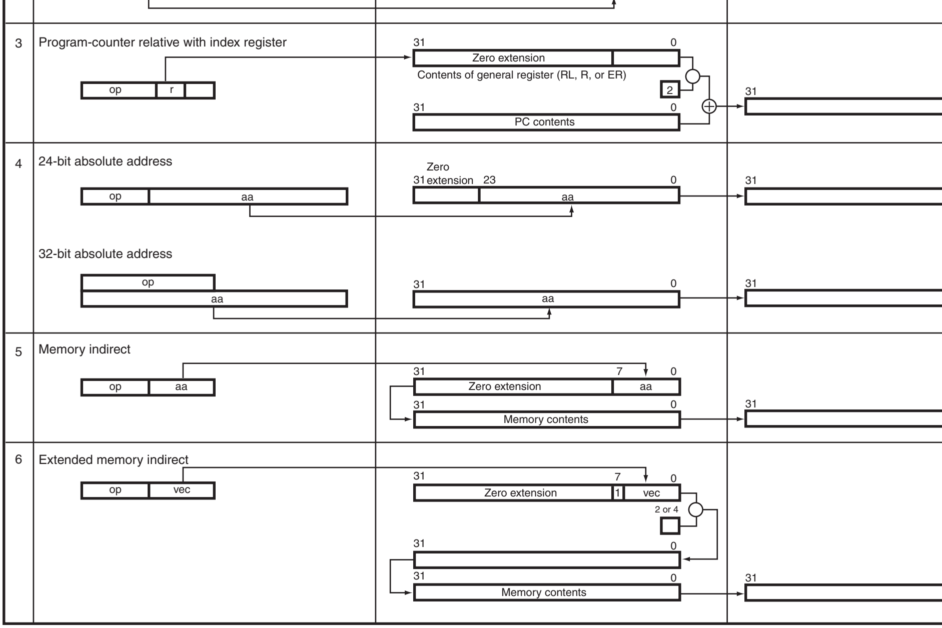
2.8.12 Effective Address Calculation

Tables 2.14 and 2.15 show how effective addresses are calculated in each addressing mo
lower bits of the effective address are valid and the upper bits are ignored (zero extende
extended) according to the CPU operating mode.

The valid bits in middle mode are as follows:

- The lower 16 bits of the effective address are valid and the upper 16 bits are sign-ext the transfer and operation instructions.
- The lower 24 bits of the effective address are valid and the upper eight bits are zero- for the branch instructions.


Register indirect with 32-bit displacement
op
r
disp
31
0
General register contents
disp
31
5
Index register indirect with 16-bit displacement
op
r
disp
Zero extension
Contents of general register (RL, R, or ER)
1, 2, or 4
15
Sign extension
disp
Index register indirect with 32-bit displacement
op
r
disp
Zero extension
Contents of general register (RL, R, or ER)
1, 2, or 4
disp
6
Register indirect with post-increment or post-decrement
op
r
General register contents
1, 2, or 4
Register indirect with pre-increment or pre-decrement
op
r
General register contents
1, 2, or 4
7
8-bit absolute address
op
aa
SBR
7
aa
16-bit absolute address
op
aa
Sign extension
aa
32-bit absolute address
op
aa
aa



2.8.13 MOVA Instruction

The MOVA instruction stores the effective address in a general register.

1. Firstly, data is obtained by the addressing mode shown in item 2 of table 2.14.
2. Next, the effective address is calculated using the obtained data as the index by the a mode shown in item 5 of table 2.14. The obtained data is used instead of the general The result is stored in a general register. For details, see H8SX Family Software Ma

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

- Exception-handling state

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

- Program execution state

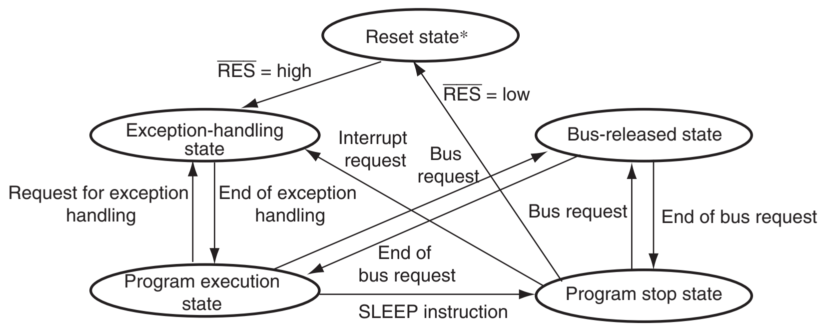
  In this state the CPU executes program instructions in sequence.

- Bus-released state

  The bus-released state occurs when the bus has been released in response to a bus req 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 when a SLEEP instruction is executed or the CPU enters hardware standby mode. For see section 23, Power-Down Modes.



Note: * A transition to the reset state occurs whenever the $\overline{RES}$ signal goes low. A transition can also be made to the reset state when the watchdog timer overflows.

Figure 2.16 State Transitions

MCU Operating Mode	MD2	MD1	MD0	CPU Operating Mode	Address Space	LSI Initiation Mode	On-Chip ROM	Exte Bu Defa
2	0	1	0	Advanced	16 Mbytes	Boot mode	Enabled	8 bits
4	1	0	0			On-chip ROM disabled extended mode	Disabled	16 bit
5	1	0	1				Disabled	8 bits
6	1	1	0			On-chip ROM enabled extended mode	Enabled	8 bits
7	1	1	1			Single-chip mode	Enabled	8 bits

Table 3.2 SDRAM Interface Selection for MCU Operating Mode

MD3	SDRAM Interface
0	Disabled
1	Enabled

In this LSI, an advanced mode as the CPU operating mode and a 16-Mbyte address spac available. The initial external bus widths are eight or 16 bits. As the LSI initiation mode external extended mode, on-chip ROM initiation mode, or single-chip initiation mode ca selected.

Mode 2 is the boot mode in which the flash memory can be programmed and erased. Fo on the boot mode, see section 21, Flash Memory (0.18-µm F-ZTAT Version).

If 16-bit address space is designated for any one area, it is called the 16-bit bus widths mo
bit address space is designated for all areas, it is called the 8-bit bus width mode.

3.2 Register Descriptions

The following registers are related to the operating mode setting.

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

3.2.1 Mode Control Register (MDCR)

MDCR indicates the current operating mode. When MDCR is read from, the states of sig
MD3 to MD0 are latched. Latching is released by a reset.

Bit	15	14	13	12	11	10	9	
Bit Name	MDS7	—	—	—	MDS3	MDS2	MDS1	
Initial Value	Undefined*	1	0	1	Undefined*	Undefined*	Undefined*	Un
R/W	R	R	R	R	R	R	R	

Bit	7	6	5	4	3	2	1	
Bit Name	—	—	—	—	—	—	—	
Initial Value	Undefined*	1	0	1	Undefined*	Undefined*	Undefined*	Un
R/W	R	R	R	R	R	R	R	

Note: * Determined by pins MD3 to MD0.

8	MDS0	Undefined*	R	When MDCR is read, the signal levels input MD2 to MD0 are latched into these bits. The latches are released by a reset.
7	—	Undefined*	R	Reserved
6	—	1	R	These are read-only bits and cannot be mod
5	—	0	R	
4	—	1	R	
3	—	Undefined*	R	
2	—	Undefined*	R	
1	—	Undefined*	R	
0	—	Undefined*	R	

Note: * Determined by pins MD3 to MD0.

Table 3.3 Settings of Bits MDS3 to MDS0

MCU Operating Mode	Mode Pins			MDCR		
	MD2	MD1	MD0	MDS3	MDS2	MDS1
2	0	1	0	1	1	0
4	1	0	0	0	0	1
5	1	0	1	0	0	0
6	1	1	0	0	1	0
7	1	1	1	0	1	0

Bit Name	—	—	—	—	—	—	DTCMD
Initial Value	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Descriptions
15	—	1	R/W	Reserved
14	—	1	R/W	These bits are always read as 1. The write valu always be 1.
13	MACS	0	R/W	MAC Saturation Operation Control Selects either saturation operation or non-satu operation for the MAC instruction. 0: MAC instruction is non-saturation operation 1: MAC instruction is saturation operation
12	—	1	R/W	Reserved This bit is always read as 1. The write value sh always be 1.
11	FETCHMD	0	R/W	Instruction Fetch Mode Select This LSI can prefetch an instruction in units of 32 bits. Select the bus width for instruction fetc depending on the used memory for the storage programs*[1]. 0: 32-bit mode 1: 16-bit mode

				When writing 0 to this bit after reading EXPE external bus cycle should not be executed. The external bus cycle may be carried out in p with the internal bus cycle depending on the s the write data buffer function. 0: External bus disabled 1: External bus enabled
8	RAME	1	R/W	RAM Enable Enables or disables the on-chip RAM. This bi initialized when the reset state is released. Do 0 during access to the on-chip RAM. 0: On-chip RAM disabled 1: On-chip RAM enabled
7 to 2	—	All 0	R/W	Reserved These bits are always read as 0. The write va always be 0.
1	DTCMD	1	R/W	DTC Mode Select Selects DTC operating mode. 0: DTC is in full-address mode 1: DTC is in short address mode
0	—	1	R/W	Reserved This bit is always read as 1. The write value s always be 1.

Notes: 1. For details on instruction fetch mode, see section 2.3, Instruction Fetch.

2. The initial value depends on the LSI initiation mode.

The CPU operating mode is advanced mode in which the address space is 16 Mbytes, and
chip ROM is disabled.

The initial bus width mode immediately after a reset is 16 bits, with 16-bit access to all ar
Ports D, E, and F function as an address bus, ports H and I function as a data bus, and par
ports A and B function as bus control signals. However, if all areas are designated as an 8
access space by the bus controller, the bus mode switches to eight bits, and only port H fu
as a data bus.

3.3.3 Mode 5

The CPU operating mode is advanced mode in which the address space is 16 Mbytes, and
chip ROM is disabled.

The initial bus width mode immediately after a reset is eight bits, with 8-bit access to all a
Ports D, E, and F function as an address bus, port H functions as a data bus, and parts of p
and B function as bus control signals. However, if any area is designated as a 16-bit acces
by the bus controller, the bus width mode switches to 16 bits, and ports H and I function a
bus.

3.3.5 Mode 7

The CPU operating mode is advanced mode in which the address space is 16 Mbytes, ar
chip ROM is enabled.

In the initial state, all areas are designated to 8-bit access space and all I/O ports can be u
general input/output ports. The external address space cannot be accessed in the initial s
setting the EXPE bit in the system control register (SYSCR) to 1 enables the external ad
space. After the external address space is enabled, ports D, E, and F can be used as an ad
output bus and ports H and I as a data bus by specifying the data direction register (DDR
port. For details, see section 9, I/O Ports.

Port B	PB7 to PB1	P*/C	P*/C	P*/C	P*/C	P*/
	PB0	P*/C	P/C*	P/C*	P*/C	P*/
Port C	PC3, PC2	P*/C	P*/C	P*/C	P*/C	P*/
Port D		P*/A	A	A	P*/A	P*/
Port E		P*/A	A	A	P*/A	P*/
Port F	PF4 to PF0	P*/A	A	A	P*/A	P*/
	PF7 to PF5	P*/A	P*/A	P*/A	P*/A	P*/
Port H		P*/D	D	D	D	P*/
Port I		P*/D	P/D*	P*/D	P*/D	P*/

[Legend]

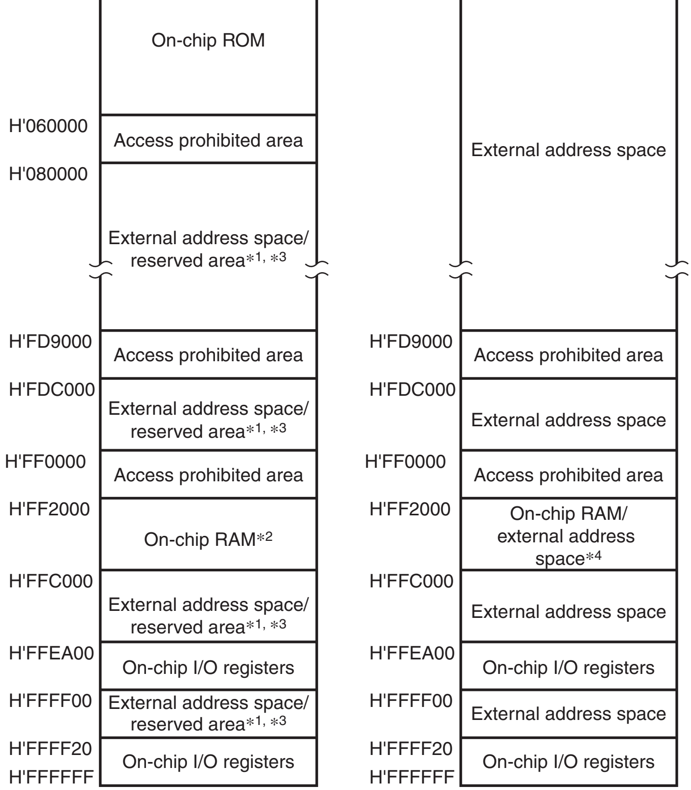
P: I/O port

A: Address bus output

D: Data bus input/output

C: Control signals, clock input/output

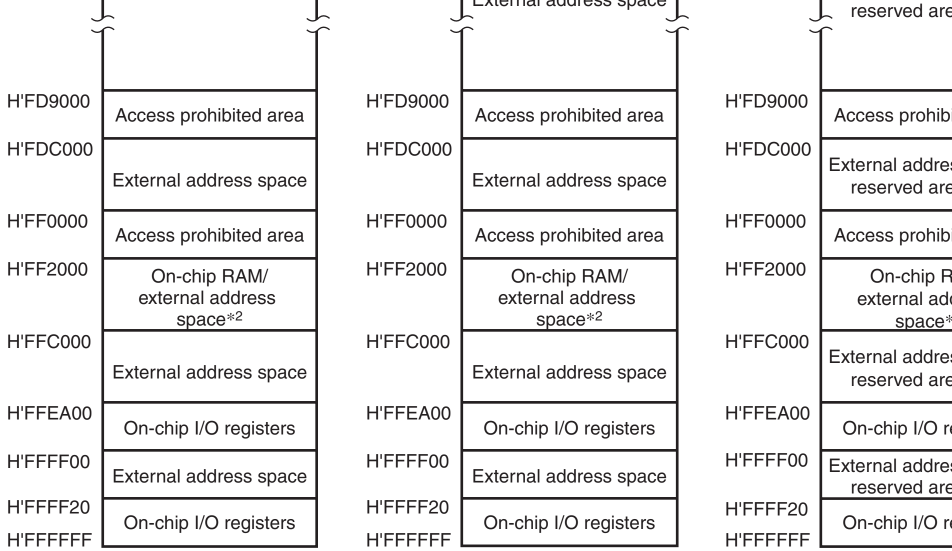
*: Immediately after a reset



Notes: 1. This area is specified as the external address space when EXPE = 1 and the reserved area when E

2. The on-chip RAM is used for flash memory programming. Do not clear the RAME bit to 0.
3. Do not access the reserved areas.
4. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.

Figure 3.1 Address Map in Each Operating Mode of H8SX/1663 (1)

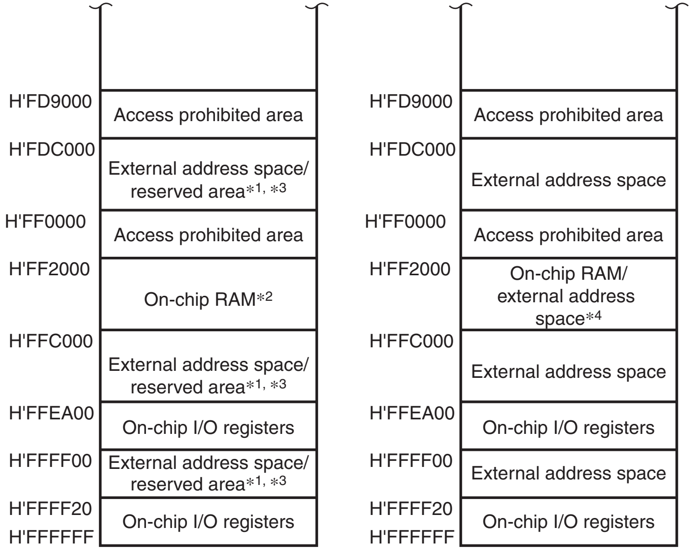


Notes: 1. This area is specified as the external address space when EXPE = 1 and the reserved area when EXPE

2. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.

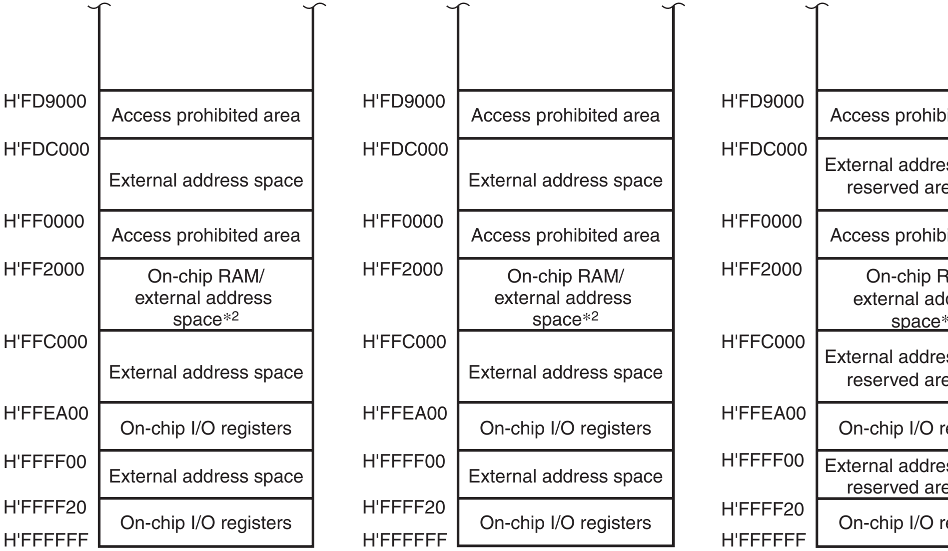
3. Do not access the reserved areas.

Figure 3.1 Address Map in Each Operating Mode of H8SX/1663 (2)



Notes: 1. This area is specified as the external address space when EXPE = 1 and the reserved area when E
2. The on-chip RAM is used for flash memory programming. Do not clear the RAME bit to 0.
3. Do not access the reserved areas.
4. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.

Figure 3.2 Address Map in Each Operating Mode of H8SX/1664 (1)



Notes: 1. This area is specified as the external address space when EXPE = 1 and the reserved area when EXPE

2. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.

3. Do not access the reserved areas.

Figure 3.2 Address Map in Each Operating Mode of H8SX/1664 (2)

Table 4.1 Exception Types and Priority

Priority	Exception Type	Exception Handling Start Timing
High	Reset	Exception handling starts at the timing of level cha low to high on the $\overline{\text{RES}}$ pin, or when the watchdog overflows. The CPU enters the reset state when t pin is low.
	Illegal instruction	Exception handling starts when an undefined cod executed.
	Trace*1	Exception handling starts after execution of the cu instruction or exception handling, if the trace (T) b is set to 1.
	Address error	After an address error has occurred, exception ha starts on completion of instruction execution.
	Interrupt	Exception handling starts after execution of the cu instruction or exception handling, if an interrupt re occurred.*2
	Sleep instruction	Exception handling starts by execution of a sleep (SLEEP), if the SSBY bit in SBYCR is set to 0 and SLPIE bit in SBYCR is set to 1.
Low	Trap instruction*3	Exception handling starts by execution of a trap in (TRAPA).

Notes: 1. Traces are enabled only in interrupt control mode 2. Trace exception handlin executed after execution of an RTE instruction.

2. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or instruction execution, or on completion of reset exception handling.

3. Trap instruction exception handling requests and sleep instruction exception requests are accepted at all times in program execution state.

Exception Source	Vector Number	Vector Table Address Offse Normal Mode*2	 Advanced, M Maximum*2
Reset	0	H'0000 to H'0001	H'0000 to H'0
Reserved for system use	1	H'0002 to H'0003	H'0004 to H'0
	2	H'0004 to H'0005	H'0008 to H'0
	3	H'0006 to H'0007	H'000C to H'
Illegal instruction	4	H'0008 to H'0009	H'0010 to H'0
Trace	5	H'000A to H'000B	H'0014 to H'0
Reserved for system use	6	H'000C to H'000D	H'0018 to H'0
Interrupt (NMI)	7	H'000E to H'000F	H'001C to H'
Trap instruction (#0)	8	H'0010 to H'0011	H'0020 to H'0
(#1)	9	H'0012 to H'0013	H'0024 to H'0
(#2)	10	H'0014 to H'0015	H'0028 to H'0
(#3)	11	H'0016 to H'0017	H'002C to H'
CPU address error	12	H'0018 to H'0019	H'0030 to H'0
DMA address error*3	13	H'001A to H'001B	H'0034 to H'0
Reserved for system use	14	H'001C to H'001D	H'0038 to H'0
	17	H'0022 to H'0023	H'0044 to H'0
Sleep interrupt	18	H'0024 to H'0025	H'0048 to H'0

	IRQ2	66	H'0084 to H'0085	H'0108 to H
	IRQ3	67	H'0086 to H'0087	H'010C to H
	IRQ4	68	H'0088 to H'0089	H'0110 to H
	IRQ5	69	H'008A to H'008B	H'0114 to H
	IRQ6	70	H'008C to H'008D	H'0118 to H
	IRQ7	71	H'008E to H'008F	H'011C to H
	IRQ8	72	H'0090 to H'0091	H'0120 to H
	IRQ9	73	H'0092 to H'0093	H'0124 to H
	IRQ10	74	H'0094 to H'0095	H'0128 to H
	IRQ11	75	H'0096 to H'0097	H'012C to H
Reserved for system use		76	H'0098 to H'0099	H'0130 to H
		79	H'009E to H'009F	H'013C to H
Internal interrupt*4		80	H'00A0 to H'00A1	H'0140 to H
		255	H'01FE to H'01FF	H'03FC to H

Notes: 1. Lower 16 bits of the address.
2. Not available in this LSI.
3. A DMA address error is generated by the DTC and DMAC.
4. For details of internal interrupt vectors, see section 5.5, Interrupt Exception H Vector Table.

A reset has priority over any other exception. When the $\overline{\text{RES}}$ pin goes low, all processing this LSI enters the reset state. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for at ms with the $\overline{\text{STBY}}$ pin driven high when the power is turned on. When operation is in pro hold the $\overline{\text{RES}}$ pin low for at least 20 cycles.

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

A reset initializes the internal state of the CPU and the registers of the on-chip peripheral The interrupt control mode is 0 immediately after a reset.

4.3.1 Reset Exception Handling

When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary time, this LSI starts re exception handling as follows:

1. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized, VBR is cleared to H'00000000, the T bit is cleared to 0 in EXR, and the I b set to 1 in EXR and CCR.
2. The reset exception handling vector address is read and transferred to the PC, and pro execution starts from the address indicated by the PC.

Figures 4.1 and 4.2 show examples of the reset sequence.

After the reset state is released, MSTPCRA and MSTPCRB are initialized to H'0FFF an
respectively, and all modules except the DTC and DMAC enter the module stop state.

Consequently, on-chip peripheral module registers cannot be read or written to. Register
and writing is enabled when the module stop state is canceled.

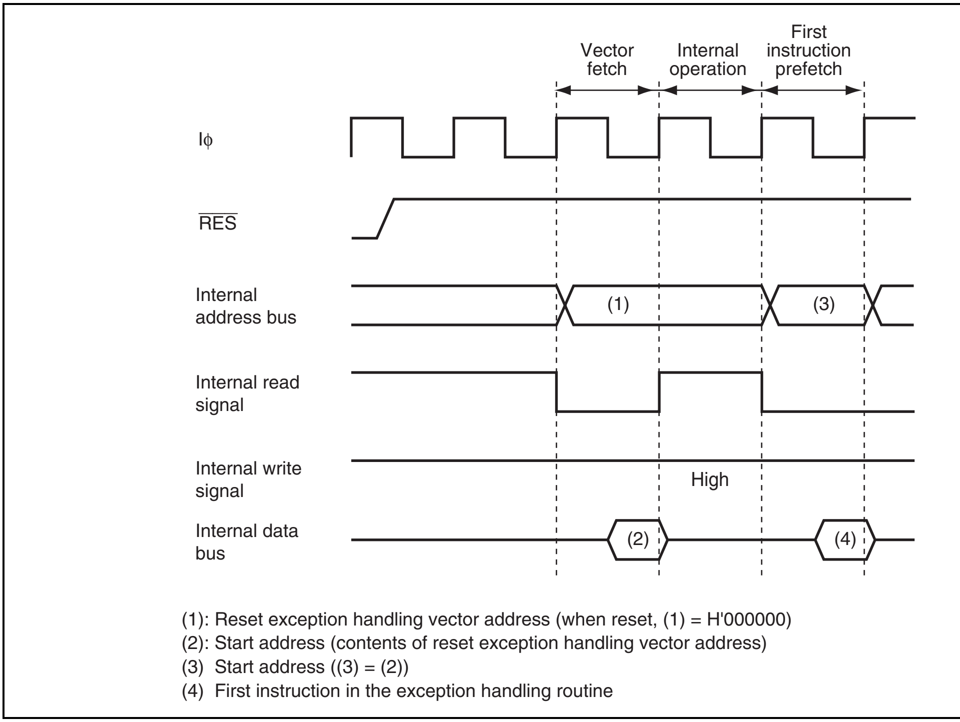
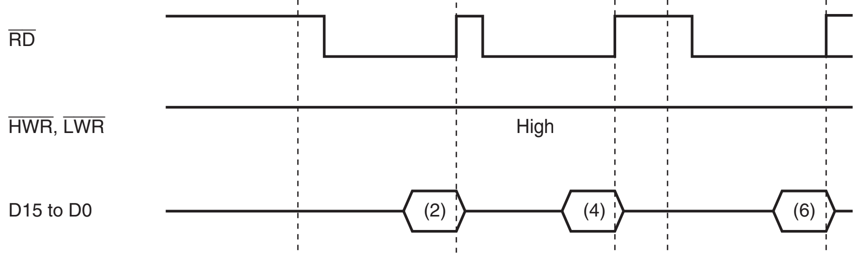


Figure 4.1 Reset Sequence (On-chip ROM Enabled Advanced Mode)



(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 instruction in the exception handling routine

Note: * Seven program wait cycles are inserted.

Figure 4.2 Reset Sequence (16-Bit External Access in On-chip ROM Disabled Advanced Mode)

handling routine by the RTE instruction, trace mode resumes. Trace exception handling
carried out after execution of the RTE instruction.

Interrupts are accepted even within the trace exception handling routine.

Table 4.4 Status of CCR and EXR after Trace Exception Handling

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

[Legend]

1: Set to 1

0: Cleared to 0

—: Retains the previous value.

Instruction fetch	CPU	Fetches instructions from even addresses	No (nor
		Fetches instructions from odd addresses	Occurs
		Fetches instructions from areas other than on-chip peripheral module space*[1]	No (nor
		Fetches instructions from on-chip peripheral module space*[1]	Occurs
		Fetches instructions from external memory space in single-chip mode	Occurs
		Fetches instructions from access prohibited area.*[2]	Occurs
Stack operation	CPU	Accesses stack when the stack pointer value is even address	No (nor
		Accesses stack when the stack pointer value is odd	Occurs
Data read/write	CPU	Accesses word data from even addresses	No (nor
		Accesses word data from odd addresses	No (nor
		Accesses external memory space in single-chip mode	Occurs
		Accesses to access prohibited area*[2]	Occurs
Data read/write	DTC or DMAC	Accesses word data from even addresses	No (nor
		Accesses word data from odd addresses	No (nor
		Accesses external memory space in single-chip mode	Occurs
		Accesses to access prohibited area*[2]	Occurs
Single address transfer	DMAC	Address access space is the external memory space for single address transfer	No (nor
		Address access space is not the external memory space for single address transfer	Occurs

Notes: 1. For on-chip peripheral module space, see section 6, Bus Controller (BSC).
2. For the access prohibited area, refer to figure 3.1 in section 3.4, Address Map.

program execution starts from that address.

Even though an address error occurs during a transition to an address error exception ha
address error is not accepted. This prevents an address error from occurring due to stack
exception handling, thereby preventing infinitive stacking.

If the SP contents are not a multiple of 2 when an address error exception handling occu
stacked values (PC, CCR, and EXR) are undefined.

When an address error occurs, the following is performed to halt the DTC and DMAC.

- The ERR bit of DTCCR in the DTC is set to 1.
- The ERRF bit of DMDR_0 in the DMAC is set to 1.
- The DTE bits of DMDRs for all channels in the DMAC are cleared to 0 to forcibly t
  transfer.

Table 4.6 shows the state of CCR and EXR after execution of the address error exceptio
handling.

Table 4.6 Status of CCR and EXR after Address Error Exception Handling

	CCR		EXR	
Interrupt Control Mode	I	UI	T	I2 to
0	1	—	—	—
2	1	—	0	7

[Legend]
1: Set to 1
0: Cleared to 0
—: Retains the previous value.

IRQ0 to IRQ11	Pins IRQ0 to IRQ11 (external input)	12
On-chip peripheral module	DMA controller (DMAC)	8
	Watchdog timer (WDT)	1
	A/D converter	1
	16-bit timer pulse unit (TPU)	26
	8-bit timer (TMR)	16
	Serial communications interface (SCI)	24
	I^2C bus interface 2 (IIC2)	2
	USB function module (USB)	5

Different vector numbers and vector table offsets are assigned to different interrupt sourc vector number and vector table offset, refer to table 5.2, Interrupt Sources, Vector Addres Offsets, and Interrupt Priority in section 5, Interrupt Controller.

3. An exception handling vector table address corresponding to the interrupt source is g
   the start address of the exception service routine is loaded from the vector table to P
   program execution starts from that address.

4.7 Instruction Exception Handling

There are three instructions that cause exception handling: trap instruction, sleep instruc illegal instruction.

4.7.1 Trap Instruction

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

1. The contents of PC, CCR, and EXR are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. An exception handling vector table address corresponding to the vector number spec the TRAPA instruction is generated, the start address of the exception service routin from the vector table to PC, and program execution starts from that address.

A start address is read from the vector table corresponding to a vector number from 0 to specified in the instruction code.

Table 4.8 shows the state of CCR and EXR after execution of trap instruction exception

4.7.2 Sleep Instruction Exception Handling

The sleep instruction exception handling starts when a sleep instruction is executed with t
bit in SBYCR set to 0 and the SLPIE bit in SBYCR set to 1. The sleep instruction except
handling can always be executed in the program execution state. In the exception handlin
CPU operates as follows.

1. The contents of PC, CCR, and EXR are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. An exception handling vector table address corresponding to the vector number speci the SLEEP instruction is generated, the start address of the exception service routine i from the vector table to PC, and program execution starts from that address.

Bus masters other than the CPU may gain the bus mastership after a sleep instruction has executed. In such cases the sleep instruction will be started when the transactions of a bus other than the CPU has been completed and the CPU has gained the bus mastership.

Table 4.9 shows the state of CCR and EXR after execution of sleep instruction exception handling.

4.7.3 Exception Handling by Illegal Instruction

The illegal instructions are general illegal instructions and slot illegal instructions. The e
handling by the general illegal instruction starts when an undefined code is executed. Th
exception handling by the slot illegal instruction starts when a particular instruction (e.g
length is two words or more, or it changes the PC contents) at a delay slot (immediately
delayed branch instruction) is executed. The exception handling by the general illegal in
and slot illegal instruction is always executable in the program execution state.

The exception handling is as follows:

1. The contents of PC, CCR, and EXR are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. An exception handling vector table address corresponding to the occurred exception generated, the start address of the exception service routine is loaded from the vector PC, and program execution starts from that address.

Table 4.10 shows the state of CCR and EXR after execution of illegal instruction except handling.

4.8 Stack Status after Exception Handling

Figure 4.3 shows the stack after completion of exception handling.

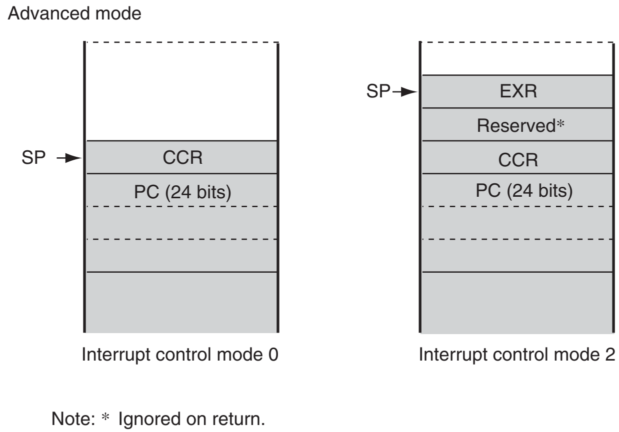


Figure 4.3 Stack Status after Exception Handling

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

Performing stack manipulation while SP is set to an odd value leads to an address error. shows an example of operation when the SP value is odd.

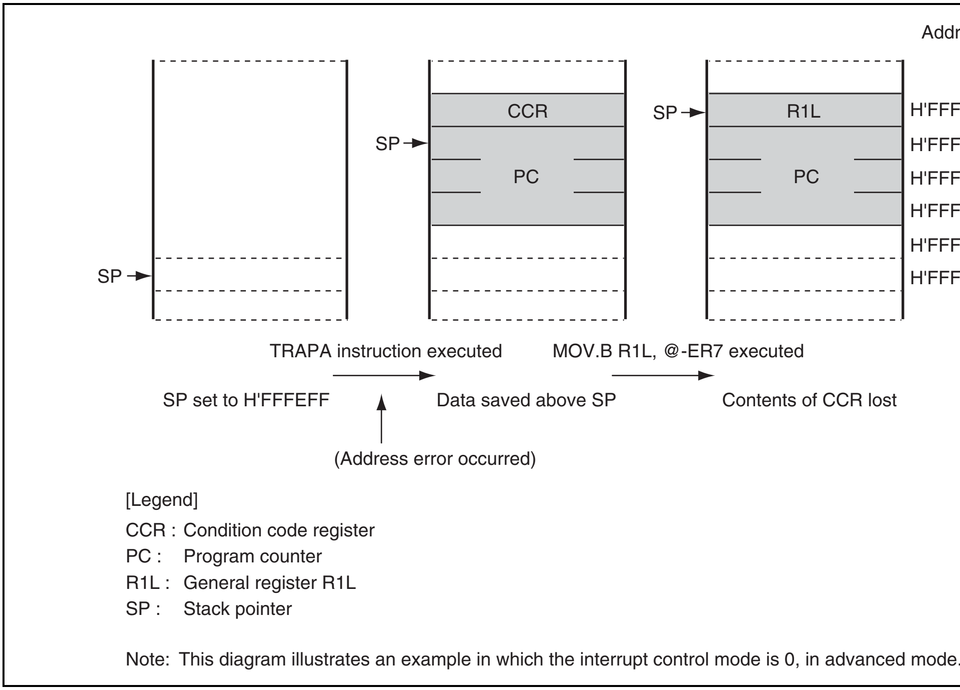


Figure 4.4 Operation when SP Value is Odd

interrupts except for the interrupt requests listed below. The following seven interrup
are given priority of 8, therefore they are accepted at all times.

— NMI

— Illegal instructions

— Trace

— Trap instructions

— CPU address error

— DMA address error (occurred in the DTC and DMAC)

— Sleep instruction

- Independent vector addresses

  All interrupt sources are assigned independent vector addresses, making it unnecessa
  source to be identified in the interrupt handling routine.

- Thirteen external interrupts

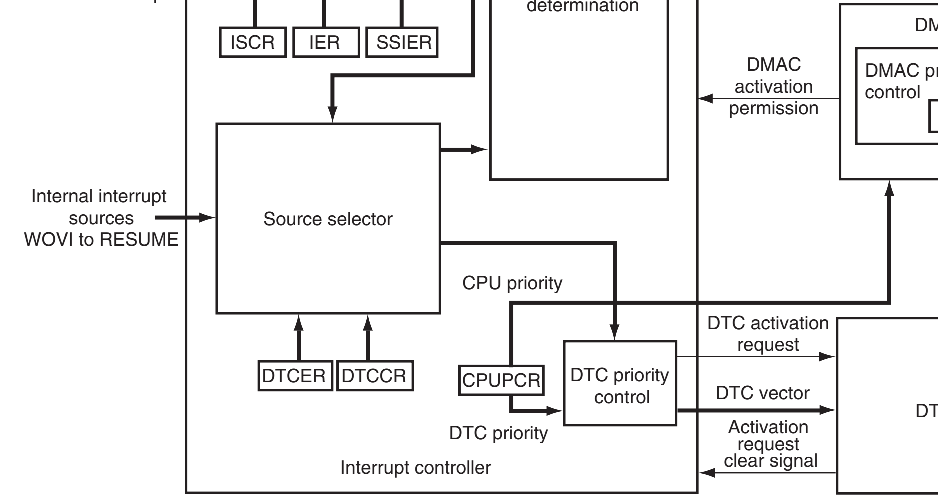
  NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or fall
  detection can be selected for NMI. Falling edge, rising edge, or both edge detection,
  sensing, can be selected for $\overline{\text{IRQ11}}$ to $\overline{\text{IRQ0}}$.

- DTC and DMAC control

  DTC and DMAC can be activated by means of interrupts.

- CPU priority control function

  The priority levels can be assigned to the CPU, DTC, and DMAC. The priority level
  CPU can be automatically assigned on an exception generation. Priority can be given
  CPU interrupt exception handling over that of the DTC and DMAC transfer.



[Legend]

INTCR:	Interrupt control register	SSIER:	Software standby release IRQ enable register
CPUPCR:	CPU priority control register	IPR:	Interrupt priority register
ISCR:	IRQ sense control register	DTCER:	DTC enable register
IER:	IRQ enable register	DTCCR:	DTC control register
ISR:	IRQ status register		

Figure 5.1 Block Diagram of Interrupt Controller

Rising, falling, or both edges, or level sensing, can be independently selected.

5.3 Register Descriptions

The interrupt controller has the following registers.

- Interrupt control register (INTCR)
- CPU priority control register (CPUPCR)
- Interrupt priority registers A to C, E to I, K, L, Q, and R (IPRA to IPRC, IPRE to IPRI, IPRK, IPRL, IPRQ, and IPRR)
- IRQ enable register (IER)
- IRQ sense control registers H and L (ISCRH, ISCRL)
- IRQ status register (ISR)
- Software standby release IRQ enable register (SSIER)

Bit	Bit Name	Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These are read-only bits and cannot be modified
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 m the interrupt controller. 00: Interrupt control mode 0 Interrupts are controlled by I bit in CCR. 01: Setting prohibited. 10: Interrupt control mode 2 Interrupts are controlled by bits I2 to I0 in EX 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 N 1: Interrupt request generated at rising edge of N
2 to 0	—	All 0	R	Reserved These are read-only bits and cannot be modified

R/W R/W R/W R/W R/W R/W R/(W) R/(W)

Note: * When the IPSETE bit is set to 1, the CPU priority is automatically updated, so these bits cannot be mo

Bit	Bit Name	Initial Value	R/W	Description
7	CPUPCE	0	R/W	CPU Priority Control Enable Controls the CPU priority control function. Setti to 1 enables the CPU priority control over the D DMAC. 0: CPU always has the lowest priority 1: CPU priority control enabled
6	DTCP2	0	R/W	DTC Priority Level 2 to 0
5	DTCP1	0	R/W	These bits set the DTC priority level.
4	DTCP0	0	R/W	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)
3	IPSETE	0	R/W	Interrupt Priority Set Enable Controls the function which automatically assig interrupt priority level of the CPU. Setting this b automatically sets bits CPUP2 to CPUP0 by the interrupt mask bit (I bit in CCR or bits I2 to I0 in 0: Bits CPUP2 to CPUP0 are not updated autor 1: The interrupt mask bit value is reflected in bi to CPUP0

011: Priority level 3

100: Priority level 4

101: Priority level 5

110: Priority level 6

111: Priority level 7 (highest)

Note: * When the IPSETE bit is set to 1, the CPU priority is automatically updated, so t cannot be modified.

5.3.3 Interrupt Priority Registers A to I, K, L, Q, and R (IPRA to IPRI, IPRK, IPRL, IPRQ, and IPRR)

IPR sets priory (levels 7 to 0) for interrupts other than NMI.

Setting a value in the range from B'000 to B'111 in the 3-bit groups of bits 14 to 12, 10 to and 2 to 0 assigns a priority level to the corresponding interrupt. For the correspondence b the interrupt sources and the IPR settings, see table 5.2.

Bit	15	14	13	12	11	10	9
Bit Name	—	IPR14	IPR13	IPR12	—	IPR10	IPR9
Initial Value	0	1	1	1	0	1	1
R/W	R	R/W	R/W	R/W	R	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	—	IPR6	IPR5	IPR4	—	IPR2	IPR1
Initial Value	0	1	1	1	0	1	1
R/W	R	R/W	R/W	R/W	R	R/W	R/W

				101: Priority level 5 110: Priority level 6 111: Priority level 7 (highest)
11	—	0	R	Reserved This is a read-only bit and cannot be modified.
10 9 8	IPR10 IPR9 IPR8	1 1 1	R/W R/W R/W	Sets the priority level of the corresponding inter 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	R	Reserved This is a read-only bit and cannot be modified.
6 5 4	IPR6 IPR5 IPR4	1 1 1	R/W R/W R/W	Sets the priority level of the corresponding inter 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)

100: Priority level 4

101: Priority level 5

110: Priority level 6

111: Priority level 7 (highest)

5.3.4 IRQ Enable Register (IER)

IER enables interrupt requests IRQ15, and IRQ11 to IRQ0.

Bit	15	14	13	12	11	10	9
Bit Name	IRQ15E	—	—	—	IRQ11E	IRQ10E	IRQ9E
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ15E	0	R/W	IRQ15 Enable The IRQ15 interrupt request is enabled when t 1. IRQ15 is internally connected to the 32KOV in the TM32K.
14 to 12	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.

8	IRQ8E	0	R/W	IRQ8 Enable The IRQ8 interrupt request is enabled when t
7	IRQ7E	0	R/W	IRQ7 Enable The IRQ7 interrupt request is enabled when t
6	IRQ6E	0	R/W	IRQ6 Enable The IRQ6 interrupt request is enabled when t
5	IRQ5E	0	R/W	IRQ5 Enable The IRQ5 interrupt request is enabled when t
4	IRQ4E	0	R/W	IRQ4 Enable The IRQ4 interrupt request is enabled when t
3	IRQ3E	0	R/W	IRQ3 Enable The IRQ3 interrupt request is enabled when t
2	IRQ2E	0	R/W	IRQ2 Enable The IRQ2 interrupt request is enabled when t
1	IRQ1E	0	R/W	IRQ1 Enable The IRQ1 interrupt request is enabled when t
0	IRQ0E	0	R/W	IRQ0 Enable The IRQ0 interrupt request is enabled when t

- ISCRH

Bit	15	14	13	12	11	10	9
Bit Name	IRQ15SR	IRQ15SF	—	—	—	—	—
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	IRQ11SR	IRQ11SF	IRQ10SR	IRQ10SF	IRQ9SR	IRQ9SF	IRQ8SR
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- ISCRL

Bit	15	14	13	12	11	10	9
Bit Name	IRQ7SR	IRQ7SF	IRQ6SR	IRQ6SF	IRQ5SR	IRQ5SF	IRQ4SR
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	IRQ3SR	IRQ3SF	IRQ2SR	IRQ2SF	IRQ1SR	IRQ1SF	IRQ0SR
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

				11: Setting prohibited
13 to 8	—	All 0	R/W	Reserved These bits are always read as 0. The write va always be 0.
7 6	IRQ11SR IRQ11SF	0 0	R/W R/W	IRQ11 Sense Control Rise IRQ11 Sense Control Fall 00: Interrupt request generated by low level o 01: Interrupt request generated at falling edge 10: Interrupt request generated at rising edge 11: Interrupt request generated at both falling edges of $\overline{\text{IRQ11}}$
5 4	IRQ10SR IRQ10SF	0 0	R/W R/W	IRQ10 Sense Control Rise IRQ10 Sense Control Fall 00: Interrupt request generated by low level o 01: Interrupt request generated at falling edge 10: Interrupt request generated at rising edge 11: Interrupt request generated at both falling edges of $\overline{\text{IRQ10}}$
3 2	IRQ9SR IRQ9SF	0 0	R/W R/W	IRQ9 Sense Control Rise IRQ9 Sense Control Fall 00: Interrupt request generated by low level o 01: Interrupt request generated at falling edge 10: Interrupt request generated at rising edge 11: Interrupt request generated at both falling edges of $\overline{\text{IRQ9}}$

ISCRL

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ7SR	0	R/W	IRQ7 Sense Control Rise
14	IRQ7SF	0	R/W	IRQ7 Sense Control Fall 00: Interrupt request generated by low level of $\overline{\text{IR}}$ 01: Interrupt request generated at falling edge of 10: Interrupt request generated at rising edge of 11: Interrupt request generated at both falling an edges of $\overline{\text{IRQ7}}$
13	IRQ6SR	0	R/W	IRQ6 Sense Control Rise
12	IRQ6SF	0	R/W	IRQ6 Sense Control Fall 00: Interrupt request generated by low level of $\overline{\text{IR}}$ 01: Interrupt request generated at falling edge of 10: Interrupt request generated at rising edge of 11: Interrupt request generated at both falling an edges of $\overline{\text{IRQ6}}$
11	IRQ5SR	0	R/W	IRQ5 Sense Control Rise
10	IRQ5SF	0	R/W	IRQ5 Sense Control Fall 00: Interrupt request generated by low level of $\overline{\text{IR}}$ 01: Interrupt request generated at falling edge of 10: Interrupt request generated at rising edge of 11: Interrupt request generated at both falling an edges of $\overline{\text{IRQ5}}$

				00: Interrupt request generated by low level of I 01: Interrupt request generated at falling edge o 10: Interrupt request generated at rising edge o 11: Interrupt request generated at both falling a edges of IRQ3
5 4	IRQ2SR IRQ2SF	0 0	R/W R/W	IRQ2 Sense Control Rise IRQ2 Sense Control Fall 00: Interrupt request generated by low level of I 01: Interrupt request generated at falling edge o 10: Interrupt request generated at rising edge o 11: Interrupt request generated at both falling a edges of IRQ2
3 2	IRQ1SR IRQ1SF	0 0	R/W R/W	IRQ1 Sense Control Rise IRQ1 Sense Control Fall 00: Interrupt request generated by low level of I 01: Interrupt request generated at falling edge o 10: Interrupt request generated at rising edge o 11: Interrupt request generated at both falling a edges of IRQ1
1 0	IRQ0SR IRQ0SF	0 0	R/W R/W	IRQ0 Sense Control Rise IRQ0 Sense Control Fall 00: Interrupt request generated by low level of I 01: Interrupt request generated at falling edge o 10: Interrupt request generated at rising edge o 11: Interrupt request generated at both falling a edges of IRQ0

Bit Name	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	I
Initial Value	0	0	0	0	0	0	0	
R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R

Note: * Only 0 can be written, to clear the flag. The bit manipulation instructions or memory operation instructi
be used to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ15F	0	R/(W)*	[Setting condition] • When the interrupt selected by ISCR occur [Clearing conditions] • Writing 0 after reading IRQnF = 1 (n = 15) • When IRQn interrupt exception handling is while falling-edge sensing is selected
14 to 12	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.

3	IRQ3F	0	R/(W)*	selected • When the DTC is activated by an IRQn int and the DISEL bit in MRB of the DTC is cl
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.

5.3.7 Software Standby Release IRQ Enable Register (SSIER)

SSIER selects the IRQ interrupt used to leave software standby mode.

The IRQ interrupt used to leave software standby mode should not be set as the DTC ac source.

Bit	15	14	13	12	11	10	9
Bit Name	SSI15	—	—	—	SSI11	SSI10	SSI9
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

				These bits are always read as 0. The write valu always be 0.
11	SSI11	0	R/W	Software Standby Release IRQ Setting
10	SSI10	0	R/W	These bits select the IRQn interrupt used to lea software standby mode (n = 11 to 0).
9	SSI9	0	R/W	0: An IRQn request is not sampled in software mode
8	SSI8	0	R/W	1: When an IRQn request occurs in software st mode, this LSI leaves software standby moc the oscillation settling time has elapsed
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	

the CPU regardless of the interrupt control mode or the settings of the CPU interrupt ma
The NMIEG bit in INTCR selects whether an interrupt is requested at the rising or fallin
the NMI pin.

When an NMI interrupt is generated, the interrupt controller determines that an error has
and performs the following procedure.

- Sets the ERR bit of DTCCR in the DTC to 1.
- Sets the ERRF bit of DMDR_0 in the DMAC to 1
- Clears the DTE bits of DMDRs for all channels in the DMAC to 0 to forcibly termin
  transfer

(2) IRQn Interrupts

An IRQn interrupt is requested by a signal input on pins $\overline{\text{IRQ11}}$ to $\overline{\text{IRQ0}}$. $\overline{\text{IRQn}}$ (n = 11 t
the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level,
  edge, rising edge, or both edges, on pins $\overline{\text{IRQn}}$.
- Enabling or disabling of interrupt requests IRQn can be selected by IER.
- The interrupt priority can be set by IPR.
- The status of interrupt requests IRQn is indicated in ISR. ISR flags can be cleared to
  software. The bit manipulation instructions and memory operation instructions shoul
  to clear the flag.

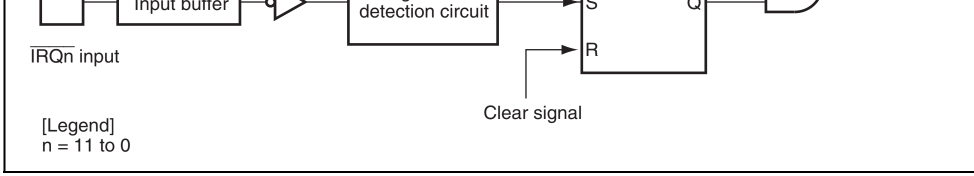


Figure 5.2 Block Diagram of Interrupts IRQn

When the IRQ sensing control in ISCR is set to a low level of signal $\overline{\text{IRQn}}$, the level of $\overline{\text{I}}$
should be held low until an interrupt handling starts. Then set the corresponding input sig
to high in the interrupt handling routine and clear the IRQnF to 0. Interrupts may not be e
when the corresponding input signal $\overline{\text{IRQn}}$ is set to high before the interrupt handling beg

5.4.2 Internal Interrupts

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

- For each on-chip peripheral module there are flags that indicate the interrupt request s and enable bits that enable or disable these interrupts. They can be controlled indepen When the enable bit is set to 1, an interrupt request is issued to the interrupt controller
- The interrupt priority can be set by means of IPR.
- The DTC and DMAC can be activated by a TPU, SCI, or other interrupt request.
- The priority levels of DTC and DMAC activation can be controlled by the DTC and D priority control functions.

Classification	Interrupt Source	Vector Number	Vector Address Offset*	IPR	Priority	DTC Activation
External pin	NMI	7	H'001C	—	High	—
External pin	IRQ0	64	H'0100	IPRA14 to IPRA12	↑	O
	IRQ1	65	H'0104	IPRA10 to IPRA8		O
	IRQ2	66	H'0108	IPRA6 to IPRA4		O
	IRQ3	67	H'010C	IPRA2 to IPRA0		O
	IRQ4	68	H'0110	IPRB14 to IPRB12		O
	IRQ5	69	H'0114	IPRB10 to IPRB8		O
	IRQ6	70	H'0118	IPRB6 to IPRB4		O
	IRQ7	71	H'011C	IPRB2 to IPRB0		O
	IRQ8	72	H'0120	IPRC14 to IPRC12		O
	IRQ9	73	H'0124	IPRC10 to IPRC8		O
	IRQ10	74	H'0128	IPRC6 to IPRC4		O
	IRQ11	75	H'012C	IPRC2 to IPRC0		O
—	Reserved for system use	76	H'0130	—		—
		77	H'0134			—
		78	H'0138			—
TM32K	32KOVI (IRQ15)	79	H'013C	IPRD2 to IPRD0		—
—	Reserved for system use	80	H'0140	—		—
WDT	WOVI	81	H'0144	IPRE10 to IPRE8	Low	—

	system use						
TPU_0	TGI0A	88	H'0160	IPRF6 to IPRF4		O	O
	TGI0B	89	H'0164			O	—
	TGI0C	90	H'0168			O	—
	TGI0D	91	H'016C			O	—
	TCI0V	92	H'0170			—	—
TPU_1	TGI1A	93	H'0174	IPRF2 to IPRF0		O	O
	TGI1B	94	H'0178			O	—
	TCI1V	95	H'017C			—	—
	TCI1U	96	H'0180			—	—
TPU_2	TGI2A	97	H'0184	IPRG14 to IPRG12		O	O
	TGI2B	98	H'0188			O	—
	TCI2V	99	H'018C			—	—
	TCI2U	100	H'0190			—	—
TPU_3	TGI3A	101	H'0194	IPRG10 to IPRG8		O	O
	TGI3B	102	H'0198			O	—
	TGI3C	103	H'019C			O	—
	TGI3D	104	H'01A0			O	—
	TCI3V	105	H'01A4			—	—
TPU_4	TGI4A	106	H'01A8	IPRG6 to IPRG4		O	O
	TGI4B	107	H'01AC			O	—
	TCI4V	108	H'01B0			—	—
	TCI4U	109	H'01B4		Low	—	—

	CMI0B	117	H'01D4			O
	OV0I	118	H'01D8			—
TMR_1	CMI1A	119	H'01DC	IPRH10 to IPRH8		O
	CMI1B	120	H'01E0			O
	OV1I	121	H'01E4			—
TMR_2	CMI2A	122	H'01E8	IPRH6 to IPRH4		O
	CMI2B	123	H'01EC			O
	OV2I	124	H'01F0			—
TMR_3	CMI3A	125	H'01F4	IPRH2 to IPRH0		O
	CMI3B	126	H'01F8			O
	OV3I	127	H'01FC			—
DMAC	DMTEND0	128	H'0200	IPRI14 to IPRI12		O
	DMTEND1	129	H'0204	IPRI10 to IPRI8		O
	DMTEND2	130	H'0208	IPRI6 to IPRI4		O
	DMTEND3	131	H'020C	IPRI2 to IPRI0		O
—	Reserved for system use	132	H'0210	—		—
		133	H'0214			—
		134	H'0218			—
		135	H'021C			—
DMAC	DMEEND0	136	H'0220	IPRK14 to IPRK12		O
	DMEEND1	137	H'0224			O
	DMEEND2	138	H'0228			O
	DMEEND3	139	H'022C		Low	O

	TXI0	146	H'0248			O	O
	TEI0	147	H'024C			—	—
SCI_1	ERI1	148	H'0250	IPRK2 to IPRK0		—	—
	RXI1	149	H'0254			O	O
	TXI1	150	H'0258			O	O
	TEI1	151	H'025C			—	—
SCI_2	ERI2	152	H'0260	IPRL14 to IPRL12		—	—
	RXI2	153	H'0264			O	O
	TXI2	154	H'0268			O	O
	TEI2	155	H'026C			—	—
—	Reserved for system use	156	H'0270	—		—	—
		157	H'0274			—	—
		158	H'0278			—	—
		159	H'027C			—	—
SCI_4	ERI4	160	H'0280	IPRL6 to IPRL4		—	—
	RXI4	161	H'0284			O	O
	TXI4	162	H'0288			O	O
	TEI4	163	H'028C			—	—
—	Reserved for system use	164 | 215	H'0290 | H'035C	—	Low	— | —	— —

	TXI5	221	H'0374			
	ERI5	222	H'0378			—
	TEI5	223	H'037C			—
SCI_6	RXI6	224	H'0380	IPRR14 to IPRR12		—
	TXI6	225	H'0384			—
	ERI6	226	H'0388			—
	TEI6	227	H'038C			—
TMR_4	CMIA4 or CMIB4	228	H'0390	IPRR10 to IPRR8		—
TMR_5	CMIA5 or CMIB5	229	H'0394			—
TMR_6	CMIA6 or CMIB6	230	H'0398			—
TMR_7	CMIA7 or CMIB7	231	H'039C			—
USB	USBINTN0	232	H'03A0	IPRR6 to IPRR4		—
	USBINTN1	233	H'03A4			—
	USBINTN2	234	H'03A8			—
	USBINTN3	235	H'03AC			—
—	Reserved for system use	236	H'03B0	IPRR2 to IPRR0		—
		237	H'03B4			—
USB	resume	238	H'03B8			—
—	Reserved for system use	239 | 255	H'03BC | H'03FC	—	Low	— | —

Note: * Lower 16 bits of the start address in advanced, middle, and maximum modes

Mode	Register	Mask Bit	Description
0	Default	I	The priority levels of the interrupt sources are f default settings. The interrupts except for NMI is masked by the
2	IPR	I2 to I0	Eight priority levels can be set for interrupt sou except for NMI with IPR. 8-level interrupt mask control is performed by b I0.

5.6.1 Interrupt Control Mode 0

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

1. If an interrupt request occurs when the corresponding interrupt enable bit is set to 1, th interrupt request is sent to the interrupt controller.
2. If the I bit in CCR is set to 1, NMI is accepted, and other interrupt requests are held pe the I bit is cleared to 0, an interrupt request is accepted.
3. For multiple interrupt requests, the interrupt controller selects the interrupt request wi highest priority, sends the request to the CPU, and holds other interrupt requests pend
4. When the CPU accepts the interrupt request, it starts interrupt exception handling afte execution of the current instruction has been completed.
5. The PC and CCR contents are saved to the stack area during the interrupt exception h The PC contents saved on the stack is the address of the first instruction to be execute returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.

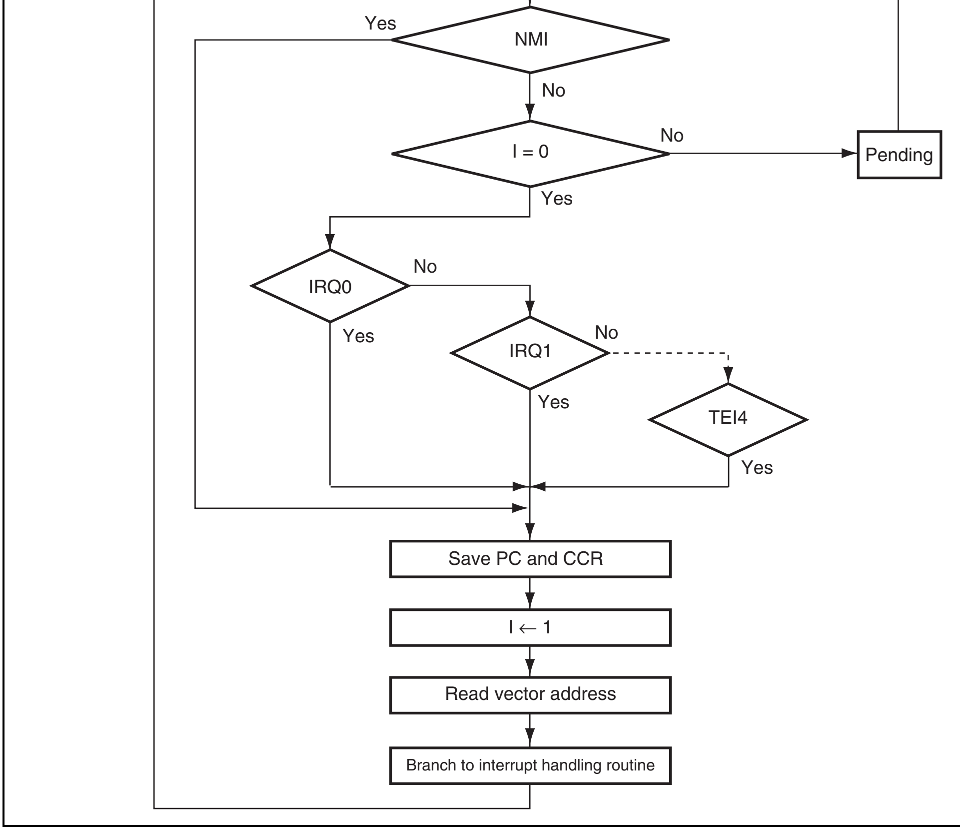


Figure 5.3 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

multiple interrupt requests have the same priority, an interrupt request is selected acc
the default setting shown in table 5.2.

3. Next, the priority of the selected interrupt request is compared with the interrupt mask in EXR. When the interrupt request does not have priority over the mask level set, it i pending, and only an interrupt request with a priority over the interrupt mask level is a
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 contents are saved to the stack area during interrupt exceptio handling. The PC saved on the stack is the address of the first instruction to be execut returning from the interrupt handling routine.
6. The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priorit accepted interrupt. If the accepted interrupt is NMI, the interrupt mask level is set to H
7. The CPU generates a vector address for the accepted interrupt and starts execution of interrupt handling routine at the address indicated by the contents of the vector addres vector table.

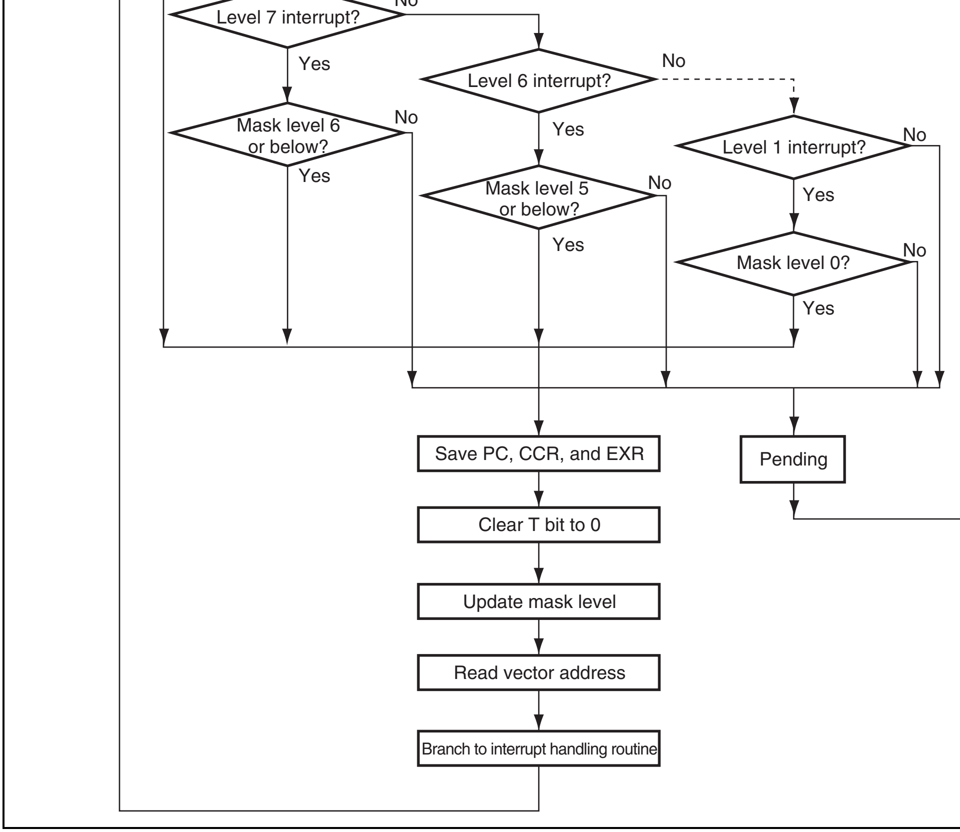


Figure 5.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

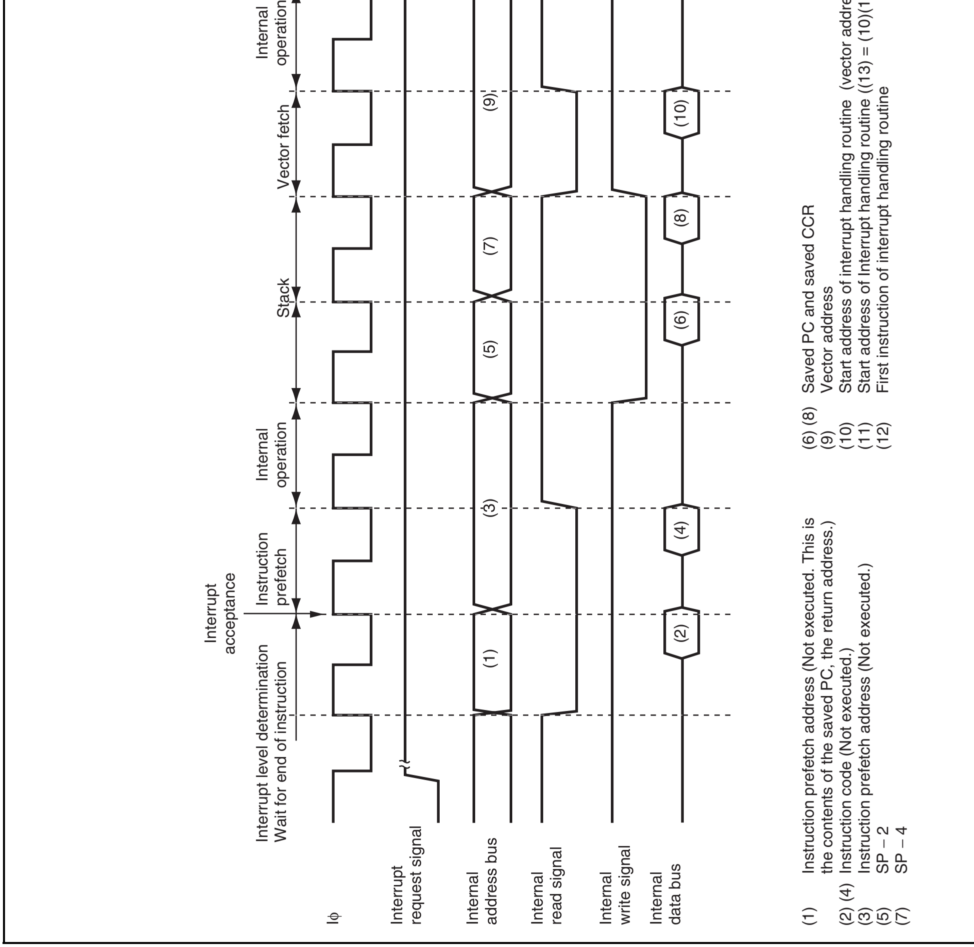


Figure 5.5 Interrupt Exception Handling

Execution State	Normal Mode* Interrupt Control Mode 0	 Interrupt Control Mode 2	Advanced Mode Interrupt Control Mode 0	 Interrupt Control Mode 2	Maximu Interrupt Control Mode 0
Interrupt priority determination*1	3				
Number of states until executing instruction ends*2	1 to 19 + $2 \cdot S_I$				
PC, CCR, EXR stacking	S_K to $2 \cdot S_K$*6	$2 \cdot S_K$	S_K to $2 \cdot S_K$*6	$2 \cdot S_K$	$2 \cdot S_K$
Vector fetch	S_h				
Instruction fetch*3	$2 \cdot S_I$				
Internal processing*4	2				
Total (using on-chip memory)	10 to 31	11 to 31	10 to 31	11 to 31	11 to 31

Notes: 1. Two states for an internal interrupt.
2. In the case of the MULXS or DIVXS instruction
3. Prefetch after interrupt acceptance or for an instruction in the interrupt handlin
4. Internal operation after interrupt acceptance or after vector fetch
5. Not available in this LSI.
6. When setting the SP value to 4n, the interrupt response time is S_K; when setti
2, the interrupt response time is $2 \cdot S_K$.

[Legend]
m: Number of wait cycles 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
available:

- Interrupt request to the CPU
- Activation request to the DTC
- Activation request to the DMAC
- Combination of the above

For details on interrupt requests that can be used to activate the DTC and DMAC, see tab
section 7, DMA Controller (DMAC), and section 8, Data Transfer Controller (DTC).

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

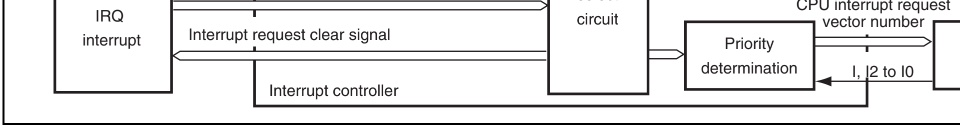


Figure 5.6 Block Diagram of DTC, DMAC, and Interrupt Controller

(1) Selection of Interrupt Sources

The activation source for each DMAC channel is selected by DMRSR. The selected acti
source is input to the DMAC through the select circuit. When transfer by an on-chip mo
interrupt is enabled (DTF1 = 1, DTF0 = 0, and DTE = 1 in DMDR) and the DTA bit in
set to 1, the interrupt source selected for the DMAC activation source is controlled by th
and cannot be used as a DTC activation source or CPU interrupt source.

Interrupt sources that are not controlled by the DMAC are set for DTC activation source
interrupt sources by the DTCE bit in DTCERA to DTCERH of the DTC.

Specifying the DISEL bit in MRB of the DTC generates an interrupt request to the CPU
clearing the DTCE bit to 0 after the individual DTC data transfer.

Note that when the DTC performs a predetermined number of data transfers and the tran
counter indicates 0, an interrupt request is made to the CPU by clearing the DTCE bit to
DTC data transfer.

When the same interrupt source is set as both the DTC and DMAC activation source and
interrupt source, the DTC and DMAC must be given priority over the CPU. If the IPSET
CPUPCR is set to 1, the priority is determined according to the IPR setting. Therefore, t
setting or the IPR setting corresponding to the interrupt source must be set to lower than
to the DTCP and DMAP setting. If the CPU is given priority over the DTC or DMAC, t
DMAC may not be activated, and the data transfer may not be performed.

are performed independently.

Table 5.6 lists the selection of interrupt sources and interrupt source clear control by setti
DTA bit in DMDR of the DMAC, the DTCE bit in DTCERA to DTCERH of the DTC, a
DISEL bit in MRB of the DTC.

Table 5.6 Interrupt Source Selection and Clear Control

DMAC Setting	DTC Setting		Interrupt Source Selection/Clear C		
DTA	DTCE	DISEL	DMAC	DTC	CPU
0	0	*	O	X	√
	1	0	O	√	X
		1	O	O	√
1	*	*	√	X	X

[Legend]

√: The corresponding interrupt is used. The interrupt source is cleared.
(The interrupt source flag must be cleared in the CPU interrupt handling routine.)

O: The corresponding interrupt is used. The interrupt source is not cleared.

X: The corresponding interrupt is not available.

*: Don't care.

(4) Usage Note

The interrupt sources of the SCI, and A/D converter are cleared according to the setting s
table 5.6, when the DTC or DMAC reads/writes the prescribed register.

To initiate multiple channels for the DTC with the same interrupt, the same priority (DTC
DMAP) should be assigned.

The priority control function over the DTC and DMAC is enabled by setting the CPUPC
CPUPCR to 1. When the CPUPCE bit is 1, the DTC and DMAC activation sources are
according to the respective priority levels.

The DTC activation source is controlled according to the priority level of the CPU indic
bits CPUP2 to CPUP0 and the priority level of the DTC indicated by bits DTCP2 to DT
CPU has priority, the DTC activation source is held. The DTC is activated when the con
which the activation source is held is cancelled (CPUPCE = 1 and value of bits CPUP2
is greater than that of bits DTCP2 to DTCP0). The priority level of the DTC is assigned
DTCP2 to DTCP0 bits regardless of the activation source.

For the DMAC, the priority level can be specified for each channel. The DMAC activati
is controlled according to the priority level of each DMAC channel indicated by bits DM
DMAP0 and the priority level of the CPU. If the CPU has priority, the DMAC activatio
held. The DMAC is activated when the condition by which the activation source is held
cancelled (CPUPCE = 1 and value of bits CPUP2 to CPUP0 is greater than that of bits D
DMAP0). If different priority levels are specified for channels, the channels of the high
levels continue transfer and the activation sources for the channels of lower priority leve
that of the CPU are held.

There are two methods for assigning the priority level to the CPU by the IPSETE bit in
Setting the IPSETE bit to 1 enables a function to automatically assign the value of the in
mask bit of the CPU to the CPU priority level. Clearing the IPSETE bit to 0 disables the
to automatically assign the priority level. Therefore, the priority level is assigned directl
software rewriting bits CPUP2 to CPUP0. Even if the IPSETE bit is 1, the priority level
CPU is software assignable by rewriting the interrupt mask bit of the CPU (I bit in CCR
bits in EXR).

Interrupt Control Mode	Interrupt Priority	Interrupt Mask Bit	IPSETE in CPUPCR	CPUP2 to CPUP0	Updating of C to CPUP0
0	Default	I = any	0	B'111 to B'000	Enabled
		I = 0	1	B'000	Disabled
		I = 1		B'100	
2	IPR setting	I2 to I0	0	B'111 to B'000	Enabled
			1	I2 to I0	Disabled

		B'100	B'000	B'000	Masked	Mas
		B'100	B'000	B'011	Masked	Mas
		B'100	B'111	B'101	Enabled	Enal
		B'000	B'111	B'101	Enabled	Enal
2	0	Any	Any	Any	Enabled	Enal
	1	B'000	B'000	B'000	Enabled	Enal
		B'000	B'011	B'101	Enabled	Enal
		B'011	B'011	B'101	Enabled	Enal
		B'100	B'011	B'101	Masked	Enal
		B'101	B'011	B'101	Masked	Enal
		B'110	B'011	B'101	Masked	Mas
		B'111	B'011	B'101	Masked	Mas
		B'101	B'011	B'101	Masked	Enal
		B'101	B'110	B'101	Enabled	Enal

be executed on completion of the instruction. However, if there is an interrupt request wi over that interrupt, interrupt exception handling will be executed for the interrupt with pri and another interrupt will be ignored. The same also applies when an interrupt source flag cleared to 0. Figure 5.7 shows an example in which the TCIEV bit in TIER of the TPU is to 0. The above conflict will not occur if an enable bit or interrupt source flag is cleared t the interrupt is masked.

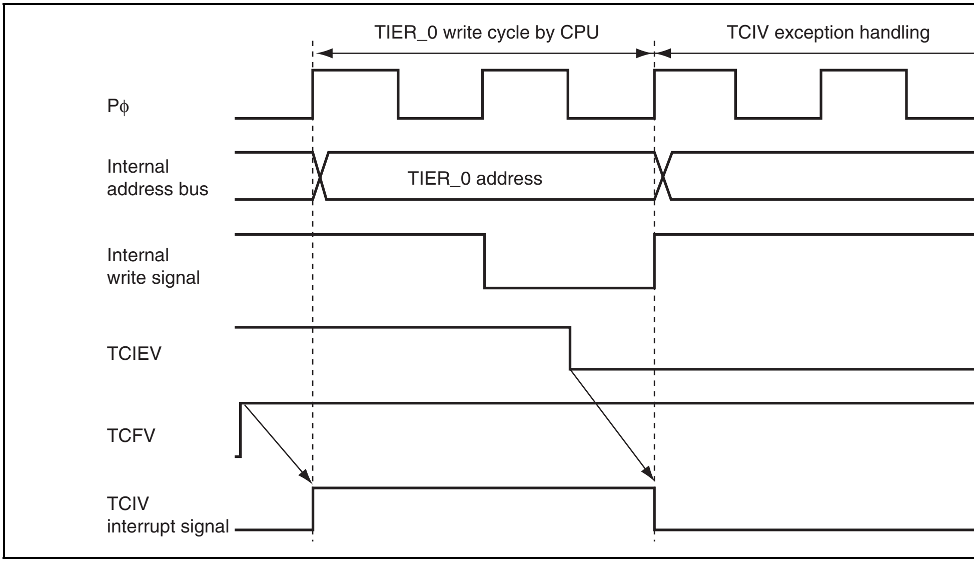


Figure 5.7 Conflict between Interrupt Generation and Disabling

Similarly, when an interrupt is requested immediately before the DTC enable bit is chang activate the DTC, DTC activation and the interrupt exception handling by the CPU are bo executed. When changing the DTC enable bit, make sure that an interrupt is not requested

The interrupt controller disables interrupt acceptance for a 3-state period after the CPU h
updated the mask level with an LDC, ANDC, ORC, or XORC instruction, and for a peri
writing to the registers of the interrupt controller.

5.8.4 Interrupts during Execution of EEPMOV Instruction

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

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

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

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

5.8.5 Interrupts during Execution of MOVMD and MOVSD Instructions

With the MOVMD or MOVSD instruction, if an interrupt request is issued during the tr
interrupt exception handling starts at the end of the individual transfer cycle. The PC val
on the stack in this case is the address of the MOVMD or MOVSD instruction. The tran
remaining data is resumed after returning from the interrupt handling routine.

- Manages external address space in area units
  Manages the external address space divided into eight areas
  Chip select signals ($\overline{\text{CS0}}$ to $\overline{\text{CS7}}$) can be output for each area
  Bus specifications can be set independently for each area
  8-bit access or 16-bit access can be selected for each area
  Burst ROM, byte control SRAM, or address/data multiplexed I/O interface can be se
  An endian conversion function is provided to connect a device of little endian
- Basic bus interface
  This interface can be connected to the SRAM and ROM
  2-state access or 3-state access can be selected for each area
  Program wait cycles can be inserted for each area
  Wait cycles can be inserted by the $\overline{\text{WAIT}}$ pin.
  Extension cycles can be inserted while $\overline{\text{CSn}}$ is asserted for each area (n = 0 to 7)
  The negation timing of the read strobe signal ($\overline{\text{RD}}$) can be modified
- Byte control SRAM interface
  Byte control SRAM interface can be set for areas 0 to 7
  The SRAM that has a byte control pin can be directly connected
- Burst ROM interface
  Burst ROM interface can be set for areas 0 and 1
  Burst ROM interface parameters can be set independently for areas 0 and 1
- Address/data multiplexed I/O interface
  Address/data multiplexed I/O interface can be set for areas 3 to 7

Synchronous DRAM interface is available as area 2

Row/column address-multiplexed output (8, 9, 10, or 11 bits)

DQM signals control byte access for 16-bit data bus device

Auto refresh and self refresh are selectable

$\overline{\text{CAS}}$ latency can be selected from 2 to 4

- Idle cycle insertion

  Idle cycles can be inserted between external read accesses to different areas

  Idle cycles can be inserted before the external write access after an external read acce

  Idle cycles can be inserted before the external read access after an external write acce

  Idle cycles can be inserted before the external access after a DMAC single address tra (write access)

- Write buffer function

  External write cycles and internal accesses can be executed in parallel

  Write accesses to the on-chip peripheral module and on-chip memory accesses can be in parallel

  DMAC single address transfers and internal accesses can be executed in parallel

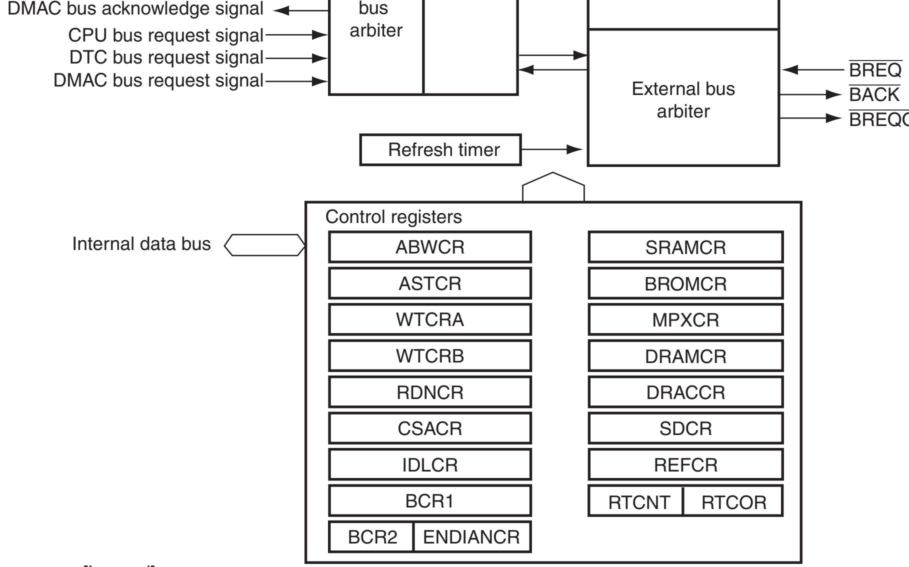
- External bus release function
- Bus arbitration function

  Includes a bus arbiter that arbitrates bus mastership among the CPU, DMAC, DTC, ar external bus master

- Multi-clock function

  The internal peripheral functions can be operated in synchronization with the peripher module clock (Pϕ). Accesses to the external address space can be operated in synchro with the external bus clock (Bϕ).

- The bus start ($\overline{\text{BS}}$) and read/write (RD/$\overline{\text{WR}}$) signals can be output.



[Legend]

ABWCR: Bus width control register
ASTCR: Access state control register
WTCRA: Wait control register A
WTCRB: Wait control register B
RDNCR: Read strobe timing control register
CSACR: $\overline{CS}$ assertion period control register
IDLCR: Idle control register
BCR1: Bus control register 1
BCR2: Bus control register 2
ENDIANCR: Endian control register

SRAMCR: SRAM mode control register
BROMCR: Burst ROM interface control registe
MPXCR: Address/data multiplexed I/O contro
DRAMCR: DRAM control register
DRACCR: DRAM access control register
SDCR: Synchronous DRAM control registe
REFCR: Refresh control register
RTCNT: Refresh timer counter
RTCOR: Refresh time constant register

Figure 6.1 Block Diagram of Bus Controller

- Idle control register (IDLCR)
- Bus control register 1 (BCR1)
- Bus control register 2 (BCR2)
- Endian control register (ENDIANCR)
- SRAM mode control register (SRAMCR)
- Burst ROM interface control register (BROMCR)
- Address/data multiplexed I/O control register (MPXCR)
- DRAM control register (DRAMCR)
- DRAM access control register (DRACCR)
- Synchronous DRAM control register (SDCR)
- Refresh control register (REFCR)
- Refresh timer counter (RTCNT)
- Refresh time constant register (RTCOR)

Bit Name	ABWL7	ABWL6	ABWL5	ABWL4	ABWL3	ABWL2	ABWL1
Initial Value	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Initial value at 16-bit bus initiation is H'FEFF, and that at 8-bit bus initiation is H'FFFF.

Bit	Bit Name	Initial Value*1	R/W	Description
15	ABWH7	1	R/W	Area 7 to 0 Bus Width Control
14	ABWH6	1	R/W	These bits select whether the corresponding ar designated as 8-bit access space or 16-bit acce
13	ABWH5	1	R/W	
12	ABWH4	1	R/W	ABWHn ABWLn (n = 7 to 0)
11	ABWH3	1	R/W	× 0: Setting prohibited
10	ABWH2	1	R/W	0 1: Area n is designated as 16- access space
9	ABWH1	1	R/W	
8	ABWL0	1/0	R/W	1 1: Area n is designated as 8-b space*2
7	ABWL7	1	R/W	
6	ABWL6	1	R/W	
5	ABWL5	1	R/W	
4	ABWL4	1	R/W	
3	ABWL3	1	R/W	
2	ABWL2	1	R/W	
1	ABWL1	1	R/W	
0	ABWL0	1	R/W	

[Legend]

×: Don't care

Notes: 1. Initial value at 16-bit bus initiation is H'FEFF, and that at 8-bit bus initiation is

2. An address space specified as byte control SRAM interface must not be spec bit access space.

Bit	7	6	5	4	3	2	1
Bit Name	—	—	—	—	—	—	—
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	AST7	1	R/W	Area 7 to 0 Access State Control
14	AST6	1	R/W	These bits select whether the corresponding are designated as 2-state access space or 3-state ac space. Wait cycle insertion is enabled or disable same time.
13	AST5	1	R/W	
12	AST4	1	R/W	
11	AST3	1	R/W	0: Area n is designated as 2-state access space
10	AST2	1	R/W	Wait cycle insertion in area n access is disabl
9	AST1	1	R/W	1: Area n is designated as 3-state access space
8	AST0	1	R/W	Wait cycle insertion in area n access is enable
				(n = 7 to 0)
7 to 0	—	All 0	R	Reserved
				These are read-only bits and cannot be modified

Bit	7	6	5	4	3	2	1
Bit Name	—	W52	W51	W50	—	W42	W41
Initial Value	0	1	1	1	0	1	1
R/W	R	R/W	R/W	R/W	R	R/W	R/W

• WTCRB

Bit	15	14	13	12	11	10	9
Bit Name	—	W32	W31	W30	—	W22	W21
Initial Value	0	1	1	1	0	1	1
R/W	R	R/W	R/W	R/W	R	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	—	W12	W11	W10	—	W02	W01
Initial Value	0	1	1	1	0	1	1
R/W	R	R/W	R/W	R/W	R	R/W	R/W

				001: 1 program wait cycle inserted 010: 2 program wait cycles inserted 011: 3 program wait cycles inserted 100: 4 program wait cycles inserted 101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted
11	—	0	R	Reserved This is a read-only bit and cannot be modified.
10 9 8	W62 W61 W60	1 1 1	R/W R/W R/W	Area 6 Wait Control 2 to 0 These bits select the number of program wait cy when accessing area 6 while bit AST6 in ASTCF 000: Program wait cycle not inserted 001: 1 program wait cycle inserted 010: 2 program wait cycles inserted 011: 3 program wait cycles inserted 100: 4 program wait cycles inserted 101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted
7	—	0	R	Reserved This is a read-only bit and cannot be modified.

				101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted
3	—	0	R	Reserved This is a read-only bit 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 c when accessing area 4 while bit AST4 in ASTC 000: Program wait cycle not inserted 001: 1 program wait cycle inserted 010: 2 program wait cycles inserted 011: 3 program wait cycles inserted 100: 4 program wait cycles inserted 101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted

- WTCRB

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This is a read-only bit and cannot be modified

				101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted
11	—	0	R	Reserved This is a read-only bit and cannot be modified.
10 9 8	W22 W21 W20	1 1 1	R/W R/W R/W	Area 2 Wait Control 2 to 0 These bits select the number of program wait c when accessing area 2 while bit AST2 in ASTC When SDRAM is connected, the CAS latency is specified. At this time, W22 is ignored. The CAS can be specified even if the wait cycle insertion disabled by ASTCR. Selection of number of program wait cycles: 000: Program wait cycle not inserted 001: 1 program wait cycle inserted 010: 2 program wait cycles inserted 011: 3 program wait cycles inserted 100: 4 program wait cycles inserted 101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted Setting of CAS latency (W22 is ignored.): 00: Setting prohibited 01: SDRAM with a CAS latency of 2 is connecte 10: SDRAM with a CAS latency of 3 is connecte 11: SDRAM with a CAS latency of 4 is connecte

				011: 3 program wait cycles inserted 100: 4 program wait cycles inserted 101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted
3	—	0	R	Reserved This is a read-only bit 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 when accessing area 0 while bit AST0 in AST 000: Program wait cycle not inserted 001: 1 program wait cycle inserted 010: 2 program wait cycles inserted 011: 3 program wait cycles inserted 100: 4 program wait cycles inserted 101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted

Bit	7	6	5	4	3	2	1
Bit Name	—	—	—	—	—	—	—
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	RDN7	0	R/W	Read Strobe Timing Control
14	RDN6	0	R/W	RDN7 to RDN0 set the negation timing of the strobe in a corresponding area read access.
13	RDN5	0	R/W	
12	RDN4	0	R/W	As shown in figure 6.2, the read strobe for an which the RDNn bit is set to 1 is negated one cycle earlier than that for an area for which the bit is cleared to 0. The read data setup and ho are also given one half-cycle earlier.
11	RDN3	0	R/W	
10	RDN2	0	R/W	
9	RDN1	0	R/W	
8	RDN0	0	R/W	0: In an area n read access, the $\overline{RD}$ signal is n at the end of the read cycle 1: In an area n read access, the $\overline{RD}$ signal is n one half-cycle before the end of the read cy (n = 7 to 0)
7 to 0	—	All 0	R	Reserved These are read-only bits and cannot be modifi

Notes: 1. In an external address space which is specified as byte control SRAM interfac RDNCR setting is ignored and the same operation when RDNn = 1 is performe

2. In an external address space which is specified as burst ROM interface, the R setting is ignored during CPU read accesses and the same operation when RD performed.

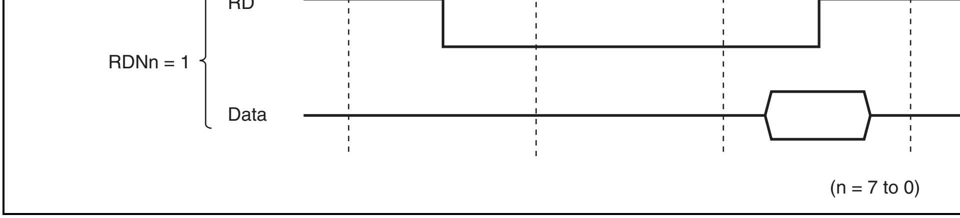


Figure 6.2 Read Strobe Negation Timing (Example of 3-State Access Spac

6.2.5 $\overline{\text{CS}}$ Assertion Period Control Registers (CSACR)

CSACR selects whether or not the assertion periods of the chip select signals ($\overline{\text{CSn}}$) and signals for the basic bus, byte-control SRAM, burst ROM, and address/data multiplexed interface are to be extended. Extending the assertion period of the $\overline{\text{CSn}}$ and address sign the setup time and hold time of read strobe ($\overline{\text{RD}}$) and write strobe ($\overline{\text{LHWR}}$/$\overline{\text{LLWR}}$) to be and to make the write data setup time and hold time for the write strobe become flexible

Bit	15	14	13	12	11	10	9
Bit Name	CSXH7	CSXH6	CSXH5	CSXH4	CSXH3	CSXH2	CSXH1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	CSXT7	CSXT6	CSXT5	CSXT4	CSXT3	CSXT2	CSXT1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

				period (Th) is extended (n = 7 to 0)
7	CSXT7	0	R/W	$\overline{CS}$ and Address Signal Assertion Period Contr
6	CSXT6	0	R/W	These bits specify whether or not the Tt cycle is
5	CSXT5	0	R/W	inserted (see figure 6.3). When an area for whic
4	CSXT4	0	R/W	CSXTn is set to 1 is accessed, one Tt cycle, in
3	CSXT3	0	R/W	the $\overline{CSn}$ and address signals are retained, is in after the normal access cycle.
2	CSXT2	0	R/W	0: In access to area n, the $\overline{CSn}$ and address as
1	CSXT1	0	R/W	period (Tt) is not extended
0	CSXT0	0	R/W	1: In access to area n, the $\overline{CSn}$ and address as period (Tt) is extended (n = 7 to 0)

Note: * In burst ROM interface, the CSXTn settings are ignored during CPU read acce

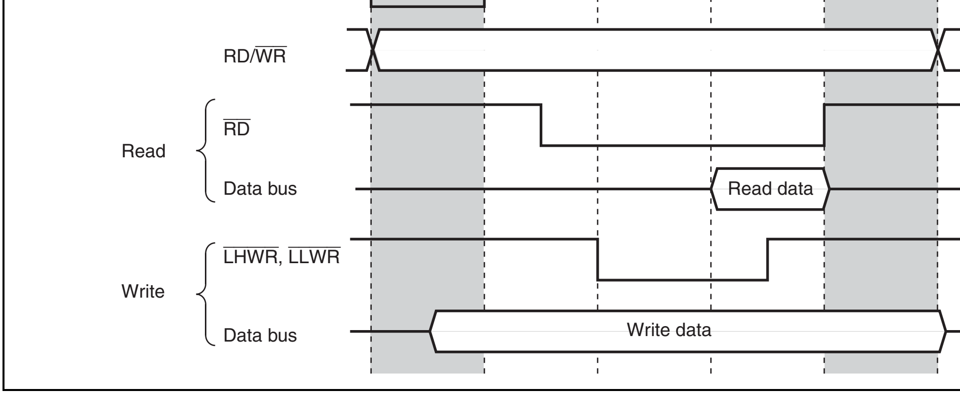


Figure 6.3 $\overline{CS}$ and Address Assertion Period Extension (Example of Basic Bus Interface, 3-State Access Space, and RDNn = 0)

Bit Name	IDLSEL7	IDLSEL6	IDLSEL5	IDLSEL4	IDLSEL3	IDLSEL2	IDLSEL1	IL
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Bit Name	Initial Value	R/W	Description
15	IDLS3	1	R/W	Idle Cycle Insertion 3 Inserts an idle cycle between the bus cycles wh DMAC single address transfer (write cycle) is fo by external access. 0: No idle cycle is inserted 1: An idle cycle is inserted
14	IDLS2	1	R/W	Idle Cycle Insertion 2 Inserts an idle cycle between the bus cycles wh external write cycle is followed by external read 0: No idle cycle is inserted 1: An idle cycle is inserted
13	IDLS1	1	R/W	Idle Cycle Insertion 1 Inserts an idle cycle between the bus cycles wh external read cycles of different areas continue 0: No idle cycle is inserted 1: An idle cycle is inserted

				00: No idle cycle is inserted 01: 2 idle cycles are inserted 00: 3 idle cycles are inserted 01: 4 idle cycles are inserted
9	IDLCA1	1	R/W	Idle Cycle State Number Select A
8	IDLCA0	1	R/W	Specifies the number of idle cycles to be inser the idle condition specified by IDLS3 to IDLS0 00: 1 idle cycle is inserted 01: 2 idle cycles are inserted 10: 3 idle cycles are inserted 11: 4 idle cycles are inserted
7	IDLSEL7	0	R/W	Idle Cycle Number Select
6	IDLSEL6	0	R/W	Specifies the number of idle cycles to be inser each area for the idle insertion condition speci IDLS1 and IDLS0.
5	IDLSEL5	0	R/W	
4	IDLSEL4	0	R/W	0: Number of idle cycles to be inserted for area specified by IDLCA1 and IDLCA0.
3	IDLSEL3	0	R/W	
2	IDLSEL2	0	R/W	1: Number of idle cycles to be inserted for area specified by IDLCB1 and IDLCB0.
1	IDLSEL1	0	R/W	
0	IDLSEL0	0	R/W	(n = 7 to 0)

Bit Name	DKC	—	—	—	—	—	—
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	BRLE	0	R/W	External Bus Release Enable Enables/disables external bus release. 0: External bus release disabled $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$, and $\overline{\text{BREQO}}$ pins can be us ports 1: External bus release enabled* For details, see section 9, I/O Ports.
14	BREQOE	0	R/W	$\overline{\text{BREQO}}$ Pin Enable Controls outputting the bus request signal (BF the external bus master in the external bus rel state when an internal bus master performs a external address space access. 0: $\overline{\text{BREQO}}$ output disabled $\overline{\text{BREQO}}$ pin can be used as I/O port 1: $\overline{\text{BREQO}}$ output enabled

				The changed setting may not affect an externa immediately after the change. 0: Write data buffer function not used 1: Write data buffer function used
8	WAITE	0	R/W	$\overline{\text{WAIT}}$ Pin Enable Selects enabling/disabling of wait input by the pin. 0: Wait input by $\overline{\text{WAIT}}$ pin disabled $\overline{\text{WAIT}}$ pin can be used as I/O port 1: Wait input by $\overline{\text{WAIT}}$ pin enabled For details, see section 9, I/O Ports.
7	DKC	0	R/W	$\overline{\text{DACK}}$ Control Selects the timing of DMAC transfer acknowle signal assertion. 0: $\overline{\text{DACK}}$ signal is asserted at the Bϕ falling ed 1: $\overline{\text{DACK}}$ signal is asserted at the Bϕ rising ed
6	—	0	R/W	Reserved This bit is always read as 0. The write value s always be 0.
5 to 0	—	All 0	R	Reserved These are read-only bits and cannot be modifi

Note: When external bus release is enabled or input by the $\overline{\text{WAIT}}$ pin is enabled, make the ICR bit to 1. For details, see section 9, I/O Ports.

Bit	Bit Name	Value	R/W	Description
7, 6	—	All 0	R	Reserved These are read-only bits and cannot be modifie
5	—	0	R/W	Reserved This bit is always read as 0. The write value sh always be 0.
4	IBCCS	0	R/W	Internal Bus Cycle Control Select Selects the internal bus arbiter function. 0: Releases the bus mastership according to th 1: Executes the bus cycles alternatively when a bus mastership request conflicts with a DMA DTC bus mastership request
3, 2	—	All 0	R	Reserved These are read-only bits and cannot be modifie
1	—	1	R/W	Reserved This bit is always read as 1. The write value sh always be 1.
0	PWDBE	0	R/W	Peripheral Module Write Data Buffer Enable Specifies whether or not to use the write data b function for the peripheral module write cycles. 0: Write data buffer function not used 1: Write data buffer function used

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	LE7	0	R/W	Little Endian Select
6	LE6	0	R/W	Selects the endian for the corresponding area
5	LE5	0	R/W	0: Data format of area n is specified as big enc
4	LE4	0	R/W	1: Data format of area n is specified as little er
3	LE3	0	R/W	(n = 7 to 2)
2	LE2	0	R/W	
1, 0	—	All 0	R	Reserved These are read-only bits and cannot be modifi

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1
Bit Name	—	—	—	—	—	—	—
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	BCSEL7	0	R/W	Byte Control SRAM Interface Select
14	BCSEL6	0	R/W	Selects the bus interface for the corresponding
13	BCSEL5	0	R/W	When setting a bit to 1, the bus interface select
12	BCSEL4	0	R/W	BROMCR and MPXCR must be cleared to 0.
11	BCSEL3	0	R/W	0: Area n is basic bus interface
10	BCSEL2	0	R/W	1: Area n is byte control SRAM interface
9	BCSEL1	0	R/W	(n = 7 to 0)
8	BCSEL0	0	R/W	
7 to 0	—	All 0	R	Reserved These are read-only bits and cannot be modifie

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSRM0	0	R/W	Area 0 Burst ROM Interface Select Specifies the area 0 bus interface. To set this clear bit BCSEL0 in SRAMCR to 0. 0: Basic bus interface or byte-control SRAM i 1: Burst ROM interface
14	BSTS02	0	R/W	Area 0 Burst Cycle Select
13	BSTS01	0	R/W	Specifies the number of burst cycles of area 0
12	BSTS00	0	R/W	000: 1 cycle 001: 2 cycles 010: 3 cycles 011: 4 cycles 100: 5 cycles 101: 6 cycles 110: 7 cycles 111: 8 cycles
11, 10	—	All 0	R	Reserved These are read-only bits and cannot be modifi

				Specifies the area 1 bus interface as a basic int or a burst ROM interface. To set this bit to 1, cl BCSEL1 in SRAMCR to 0. 0: Basic bus interface or byte-control SRAM int 1: Burst ROM interface
6	BSTS12	0	R/W	Area 1 Burst Cycle Select
5	BSTS11	0	R/W	Specifies the number of cycles of area 1 burst c
4	BSTS10	0	R/W	000: 1 cycle 001: 2 cycles 010: 3 cycles 011: 4 cycles 100: 5 cycles 101: 6 cycles 110: 7 cycles 111: 8 cycles
3, 2	—	All 0	R	Reserved These are read-only bits and cannot be modifie
1	BSWD11	0	R/W	Area 1 Burst Word Number Select
0	BSWD10	0	R/W	Selects the number of words in burst access to 1 burst ROM interface 00: Up to 4 words (8 bytes) 01: Up to 8 words (16 bytes) 10: Up to 16 words (32 bytes) 11: Up to 32 words (64 bytes)

Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	MPXE7	0	R/W	Address/Data Multiplexed I/O Interface Select
14	MPXE6	0	R/W	Specifies the bus interface for the correspondi
13	MPXE5	0	R/W	To set this bit to 1, clear the BCSELn bit in SF 0.
12	MPXE4	0	R/W	0: Area n is specified as a basic interface or a control SRAM interface.
11	MPXE3	0	R/W	1: Area n is specified as an address/data mult I/O interface (n = 7 to 3)
10 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modifi
0	ADDEX	0	R/W	Address Output Cycle Extension Specifies whether a wait cycle is inserted for t address output cycle of address/data multiplex interface. 0: No wait cycle is inserted for the address ou 1: One wait cycle is inserted for the address o cycle

Bit Name	BE	RCDM	DDS	—	—	—	MXC1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	DRAME	0	R/W	Area 2 DRAM Interface Select Selects whether or not area 2 is specified as th DRAM/SDRAM interface. When this bit is set t select the type of DRAM to be used in area 2 w DTYPE bit. When this bit is set to 1, the BCSE SRAMCR should be set to 0. 0: Basic bus interface or byte-control SRAM in 1: DRAM/SDRAM interface
14	DTYPE	0	R/W	DRAM Select Selects the type of DRAM to be used in area 2 0: DRAM is used in area 2 1: SDRAM is used in area 2
13, 12	—	All 0	R	Reserved The initial value should not be changed.
11	OEE	0	R/W	$\overline{OE}$ Output Enable The $\overline{OE}$ signal is output when DRAM with the E page mode is connected, whereas the CKE sig output when SDRAM is connected. 0: $\overline{OE}$/CKE signal output disabled (the $\overline{OE}$/CKE be used as an I/O port) 1: $\overline{OE}$/CKE signal enabled

				1: RAS signal is asserted at the rising edge o signal in the Tr cycle
9	—	0	R	Reserved The initial value should not be changed.
8	CAST	0	R/W	Column Address Output Cycle Count Select Selects whether the number of column addres cycles is two or three during a DRAM access. When SDRAM is used, the setting of this bit d affect operation. 0: Column address is output for two cycles 1: Column address is output for three cycles
7	BE	0	R/W	Burst Access Enable Enables or disables a burst access to the DRAM/SDRAM. The DRAM/SDRAM is acces high-speed page mode. When DRAM with the page mode is used, connect the $\overline{OE}$ signal of the $\overline{OE}$ signal of DRAM. 0: DRAM/SDRAM is accessed with full acces 1: DRAM/SDRAM is accessed in high-speed mode

				interface, the READ/WRIT command is issued issuance of the ACTV command when the sam address is accessed consecutively. 0: RAS up mode when the DRAM/SDRAM is a 1: RAS down mode when the DRAM/SDRAM i accessed
5	DDS	0	R/W	DMAC Single Address Transfer Option Selects whether a DMAC single address transf through the DRAM/SDRAM interface is enable full access mode or is also enabled in fast-pag mode. When clearing the BE bit to 0 to disable a burs to the DRAM/SDRAM interface, a DMAC singl transfer is performed in full access mode regar this bit. This bit does not affect an external access by o masters or a DMAC dual address transfer. Set bit to 1 changes the $\overline{\text{DACK}}$ output timing. 0: DMAC single address transfer through the DRAM/SDRAM is enabled only in full acces 1: DMAC single address transfer through the DRAM/SDRAM is also enabled in fast-page mode

00: Shifted by 8 bits

A23 to A8 are compared for 8-bit access spac

A23 to A9 are compared for 16-bit access spa

01: Shifted by 9 bits

A23 to A9 are compared for 8-bit access spac

A23 to A10 are compared for 16-bit access sp

10: Shifted by 10 bits

A23 to A10 are compared for 8-bit access spa

A23 to A11 are compared for 16-bit access sp

11: Shifted by 11 bits

A23 to A11 are compared for 8-bit access spa

A23 to A12 are compared for 16-bit access sp

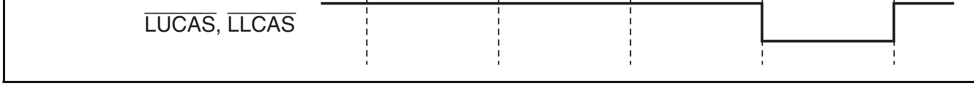


Figure 6.4 $\overline{RAS}$ Assertion Timing (Column Address Output for 2 cycles in Full Access Mode)

6.2.14 DRAM Access Control Register (DRACCR)

DRACCR specifies the settings for the DRAM/SDRAM interface. Rewrite this register w
DRAM/SDRAM is not accessed.

Bit	15	14	13	12	11	10	9
Bit Name	—	—	TPC1	TPC0	—	—	RCD1
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W

Bit	7	6	5	4	3	2	1
Bit Name	—	—	—	—	—	—	—
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

				11: Four cycles
11, 10	—	All 0	R	Reserved The initial value should not be changed.
9	RCD1	0	R/W	$\overline{RAS}$-$\overline{CAS}$ Wait Control
8	RCD0	0	R/W	Select the number of wait cycles inserted betw and $\overline{CAS}$ cycles. 00: No wait cycle inserted 01: One wait cycle inserted 10: Two wait cycles inserted 11: Three wait cycles inserted
7 to 0	—	All 0	R	Reserved The initial value should not be changed.

Bit	7	6	5	4	3	2	1
Bit Name	CKSPE	—	—	—	—	—	—
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	MRSE	0	R/W	Mode Register Set Enable Enables the setting in the SDRAM mode regist section 6.11.14, Setting SDRAM Mode Registe 0: Disables to set the SDRAM mode register 1: Enables to set the SDRAM mode register
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The initial val should not be changed.
11, 10	—	0	R/W	Reserved The initial value should not be changed.
9	—	0	R	Reserved
8	—	0	R/W	The initial value should not be changed.
7	CKSPE	0	R/W	Clock Suspend Enable Enables the clock suspend mode in which read output cycles are extended. Setting this bit to 1 cycles in which read data is output from SDRA 0: Disables the clock suspend mode 1: Enables the clock suspend mode

6.2.16 Refresh Control Register (REFCR)

REFCR specifies the refresh type for the DRAM/SDRAM interface.

Bit	15	14	13	12	11	10	9
Bit Name	CMF	CMIE	RCW1	RCW0	—	RTCK2	RTCK1
Initial Value	0	0	0	0	0	0	0
R/W	R/(W)*	R/W	R/W	R/W	R	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	RFSHE	RLW2	RLW1	RLW0	SLFRF	TPCS2	TPCS1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

				• When RTCNT matches RTCOR
14	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables an interrupt request (CMIF the CMF flag is set to 1. This bit is effective when refresh control is not performed (RFSHE = 0). When refresh control performed (RFSHE = 1), this bit is always clea This bit cannot be modified.
13 to 12	RCW1	0	R/W	$\overline{\text{CAS}}$-$\overline{\text{RAS}}$ Wait Control
	RCW0	0	R/W	Select the number of wait cycles inserted betw $\overline{\text{CAS}}$ asserted cycle and $\overline{\text{CAS}}$ asserted cycle d DRAM refresh. When the SDRAM space is selected, these bits affect operations although they can be read fro written to. 00: No wait cycle inserted 01: One wait cycle inserted 10: Two wait cycles inserted 11: Three wait cycles inserted
11	—	0	R	Reserved The initial value should not be changed.

				001: Counts on Pϕ/32 001: Counts on Pϕ/128 001: Counts on Pϕ/512 001: Counts on Pϕ/2048 001: Counts on Pϕ/4096
7	RFSHE	0	R/W	Refresh Control Enables or disables refresh control. When ref control is disabled, the refresh timer can be u interval timer. In single-chip activation mode, the setting of t should be made after setting the EXPE bit in S 1. For SYSCR, see section 3, MCU Operating 0: Refresh control enabled 1: Refresh control disabled
6 5 4	RLW2 RLW1 RLW0	0 0 0	R/W R/W R/W	Refresh Cycle Wait Control Select the number of wait cycles during a CA RAS refresh cycle for the DRAM interface and refresh cycle for the SDRAM interface. 000: No wait cycle inserted 001: One wait cycle inserted 010: Two wait cycles inserted 010: Three wait cycles inserted 010: Four wait cycles inserted 010: Five wait cycles inserted 010: Six wait cycles inserted 010: Seven wait cycles inserted

				0: Disables self-refresh 1: Enables self-refresh
2	TPS2	0	R/W	Precharge Cycle Control during Self-Refresh
1	TPS1	0	R/W	Selects the number of precharge cycles immed after a self-refresh cycle. The number of actual of precharge cycles is the sum of the numbers by these bits and bits TPC1 and TPC0.
0	TPS0	0	R/W	000: No wait cycle inserted 001: One wait cycle inserted 010: Two wait cycles inserted 010: Three wait cycles inserted 010: Four wait cycles inserted 010: Five wait cycles inserted 010: Six wait cycles inserted 010: Seven wait cycles inserted

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

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

6.2.18 Refresh Time Constant Register (RTCOR)

RTCOR specifies intervals at which a compare match for RTCOR and RTCNT is gener

The RTCOR value is always compared with the RTCNT value. When they match, the C in REFCR is set to 1 and RTCNT is initialized to H'00.

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

registers of peripheral modules such as SCI and timer.

- External access cycle

  A bus that accesses external devices via the external bus interface.

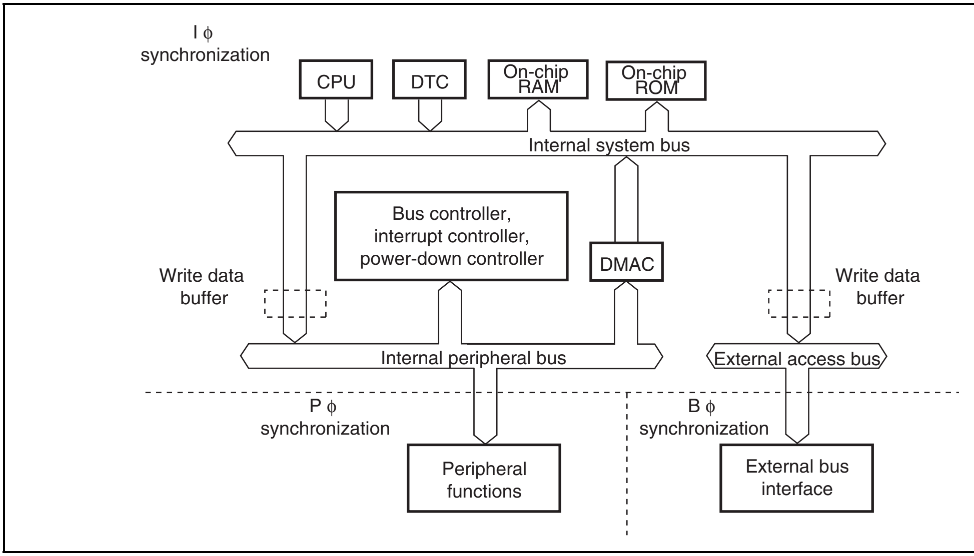


Figure 6.5 Internal Bus Configuration

	Bus controller CPU DTC DMAC Internal memory Clock pulse generator Power down control
Pϕ	I/O ports TPU PPG TMR WDT SCI A/D D/A IIC2 USB
Bϕ	External bus interface

The frequency of each synchronization clock (Iϕ, Pϕ, and Bϕ) is specified by the system control register (SCKCR) independently. For further details, see section 22, Clock Pulse Generator.

There will be cases when Pϕ and Bϕ are equal to Iϕ and when Pϕ and Bϕ are different fr according to the SCKCR specifications. In any case, access cycles for internal periphera and external space is performed synchronously with Pϕ and Bϕ, respectively.

the frequency rate of Iϕ and Pϕ is m : 1, 0 to m-1 cycles of Tsy may be inserted.

Figure 6.6 shows the external 2-state access timing when the frequency rate of Iϕ and Bϕ
Figure 6.7 shows the external 3-state access timing when the frequency rate of Iϕ and Bϕ

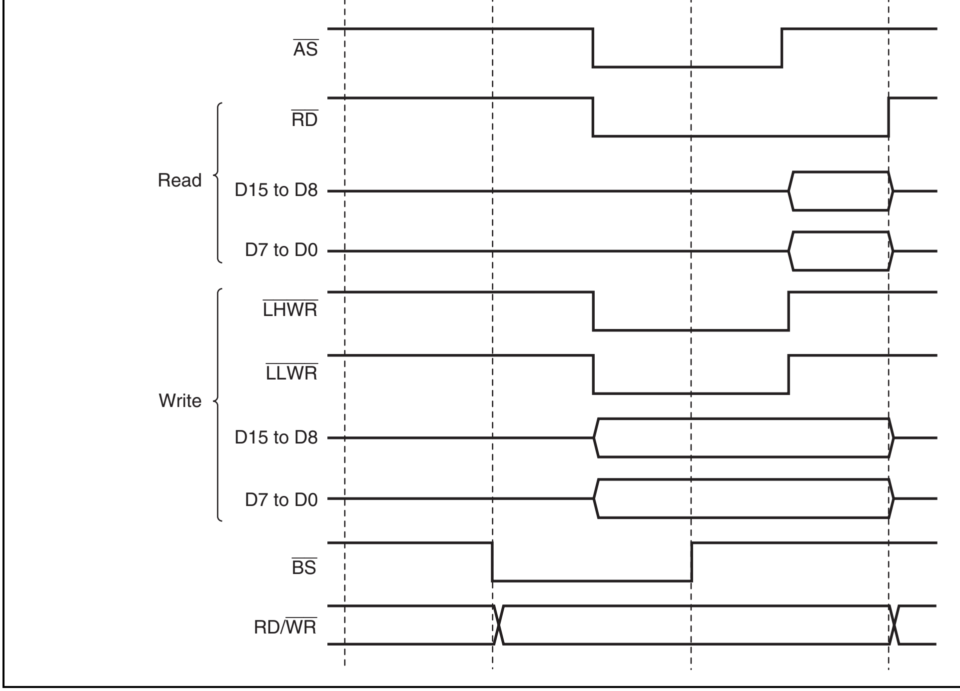


Figure 6.6 System Clock: External Bus Clock = 4:1, External 2-State Acce

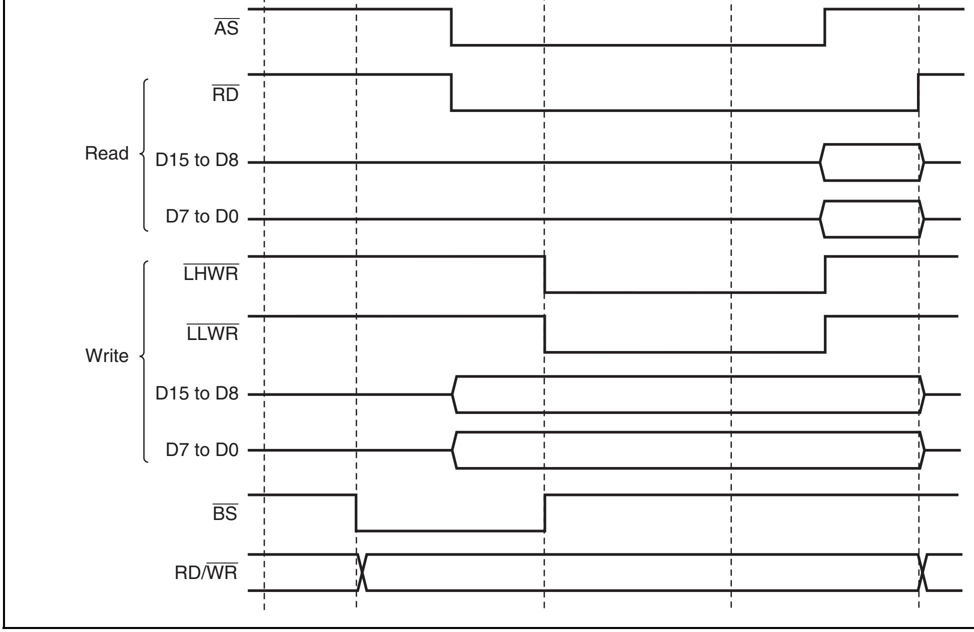


Figure 6.7 System Clock: External Bus Clock = 2:1, External 3-State Acces

Bus cycle start	$\overline{BS}$	Output	Signal indicating that the bus cycle has started
Address strobe/ address hold	$\overline{AS}$/$\overline{AH}$	Output	• Strobe signal indicating that the basic bus, SRAM, or burst ROM space is accessed ar output on address bus is enabled • Signal to hold the address during access to address/data multiplexed I/O interface
Read strobe	$\overline{RD}$	Output	Strobe signal indicating that the basic bus, byte SRAM, burst ROM, or address/data multiplexe is being read
Read/write	RD/$\overline{WR}$	Output	• Signal indicating the input or output directio • Write enable signal of the SRAM during ac byte control SRAM space
Low-high write/ lower-upper byte select	$\overline{LHWR}$/$\overline{LUB}$	Output	• Strobe signal indicating that the basic bus, or address/data multiplexed I/O space is w and the upper byte (D15 to D8) of data bus • Strobe signal indicating that the byte contro space is accessed, and the upper byte (D1 data bus is enabled
Low-low write/ lower-lower byte select	$\overline{LLWR}$/$\overline{LLB}$	Output	• Strobe signal indicating that the basic bus, or address/data multiplexed I/O space is w and the lower byte (D7 to D0) of data bus i • Strobe signal indicating that the byte contro space is accessed, and the lower byte (D7 data bus is enabled

Row address strobe	RAS	Output	• Row address strobe signal when area 2 specified as DRAM space • Row address strobe signal when area 2 specified as SDRAM space
Column address strobe	$\overline{\text{CAS}}$	Output	Column address strobe signal when area 2 specified as SDRAM space
Write enable	$\overline{\text{WE}}$	Output	• Write enable signal for DRAM • Write enable signal when area 2 is spec SDRAM space
Lower-upper-column address strobe/lower-upper-data mask enable	$\overline{\text{LUCAS}}$/DQMLU	Output	• Lower-upper-column address strobe sig bit DRAM • Upper-column address strobe signal for DRAM • Lower-upper-data mask enable signal fo SDRAM • Upper-data mask enable signal for 16-b
Lower-lower-column address strobe/lower-lower-data mask enable	$\overline{\text{LLCAS}}$/DQMLL	Output	• Lower-lower-column address strobe sig bit DRAM • Lower-column address strobe signal for DRAM • Column address strobe signal for 8-bit D • Lower-lower-data mask enable signal fo SDRAM • Lower-data mask enable signal for 16-b • Data mask enable signal for 8-bit SDRA

Bus request output	$\overline{\text{BREQO}}$	Output	External bus request signal used when inte master accesses external address space i external-bus released state
Data transfer acknowledge 3 (DMAC_3)	$\overline{\text{DACK3}}$	Output	Data acknowledge signal for DMAC_3 sing transfer
Data transfer acknowledge 2 (DMAC_2	$\overline{\text{DACK2}}$	Output	Data acknowledge signal for DMAC_2 sing transfer
Data transfer acknowledge 1 (DMAC_1)	$\overline{\text{DACK1}}$	Output	Data acknowledge signal for DMAC_1 sing transfer
Data transfer acknowledge 0 (DMAC_0)	$\overline{\text{DACK0}}$	Output	Data acknowledge signal for DMAC_0 sing transfer
External bus clock	Bϕ	Output	External bus clock

$\overline{CS3}$	—	—	—	O	O	O	—	—	O	O	
$\overline{CS4}$	—	—	—	O	O	O	—	—	O	O	
$\overline{CS5}$	—	—	—	O	O	O	—	—	O	O	
$\overline{CS6}$	—	—	—	O	O	O	—	—	O	O	
$\overline{CS7}$	—	—	—	O	O	O	—	—	O	O	
BS	—	—	—	O	O	O	O	O	O	O	
RD/$\overline{WR}$	—	—	—	O	O	O	O	O	O	O	
$\overline{AS}$	Output	Output	—	O	O	O	O	O	—	—	
$\overline{AH}$	—	—	—	—	—	—	—	—	O	O	
$\overline{RD}$	Output	Output	—	O	O	O	O	O	O	O	
$\overline{LHWR}$/$\overline{LUB}$	Output	Output	—	O	—	O	O	—	O	—	
$\overline{LLWR}$/$\overline{LLB}$	Output	Output	—	O	O	O	O	O	O	O	
$\overline{WAIT}$	—	—	—	O	O	O	O	O	O	O	Controlle WAITE

[Legend]

O: Used as a bus control signal

—: Not used as a bus control signal (used as a port input when initialized)

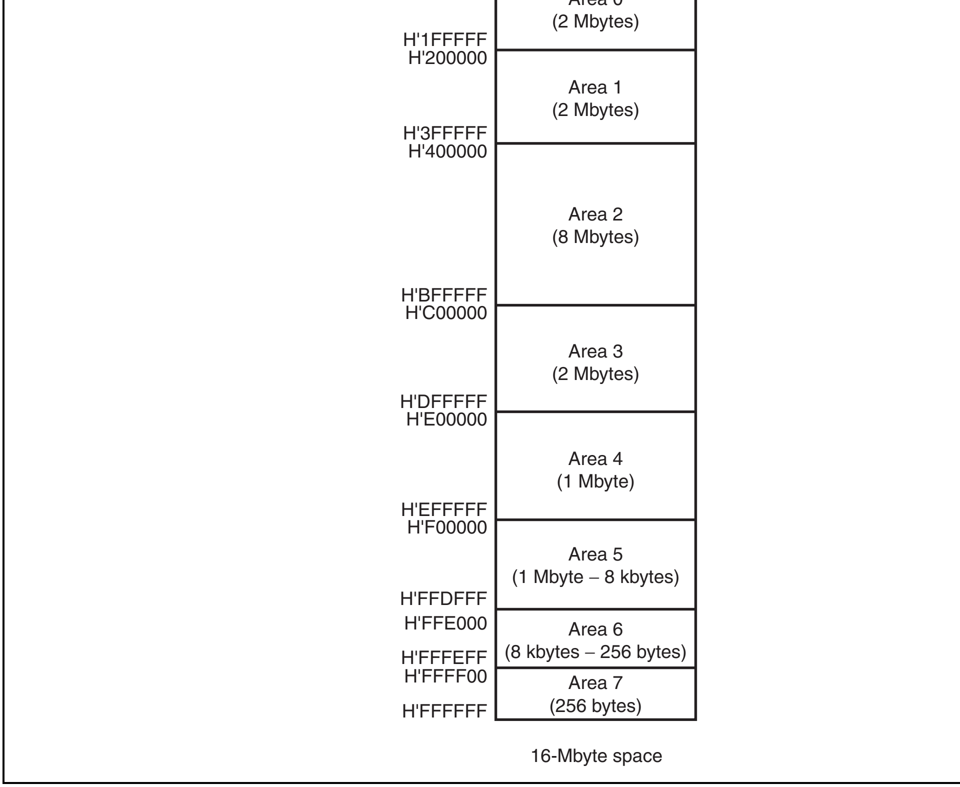


Figure 6.8 Address Space Area Division

be set to 1 when outputting signals $\overline{\mathrm{CS1}}$ to $\overline{\mathrm{CS7}}$.

In on-chip ROM enabled extended mode, pins $\overline{\mathrm{CS0}}$ to $\overline{\mathrm{CS7}}$ are all placed in the input state
reset and so the corresponding PFCR bits should be set to 1 when outputting signals $\overline{\mathrm{CS0}}$

The PFCR can specify multiple $\overline{\mathrm{CS}}$ outputs for a pin. If multiple $\overline{\mathrm{CSn}}$ outputs are specifie
single pin by the PFCR, $\overline{\mathrm{CS}}$ to be output are generated by mixing all the $\overline{\mathrm{CS}}$ signals. In th
the settings for the external bus interface areas in which the $\overline{\mathrm{CSn}}$ signals are output to a si
should be the same.

Figure 6.10 shows the signal output timing when the $\overline{\mathrm{CS}}$ signals to be output to areas 5 an
output to the same pin.

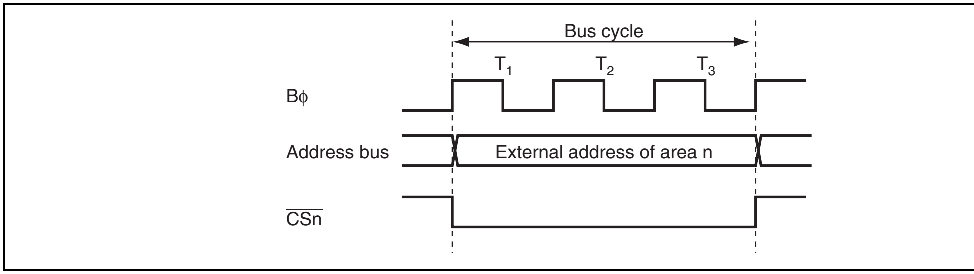


Figure 6.9 $\overline{\mathrm{CSn}}$ Signal Output Timing (n = 0 to 7)

Figure 6.10 Timing When $\overline{CS}$ Signal is Output to the Same Pin

6.5.4 External Bus Interface

The type of the external bus interfaces, bus width, endian format, number of access cycl
strobe assert/negate timings can be set for each area in the external address space. The b
and the number of access cycles for both on-chip memory and internal I/O registers are
are not affected by the external bus settings.

(1) Type of External Bus Interface

Four types of external bus interfaces are provided and can be selected in area units. Tabl
shows each interface name, description, area name to be set for each interface. Table 6.5
areas that can be specified for each interface. The initial state of each area is a basic bus

Table 6.4 Interface Names and Area Names

Interface	Description	Area Name
Basic interface	Directly connected to ROM and RAM	Basic bus space
Byte control SRAM interface	Directly connected to byte SRAM with byte control pin	Byte control SRAM s
Burst ROM interface	Directly connected to the ROM that allows page access	Burst ROM space
Address/data multiplexed I/O interface	Directly connected to the peripheral LSI that requires address and data multiplexing	Address/data multipl space

(2) Bus Width

A bus width of 8 or 16 bits can be selected with ABWCR. An area for which an 8-bit bus selected functions as an 8-bit access space and an area for which a 16-bit bus is selected f as a 16-bit access space. In addition, the bus width of address/data multiplexed I/O space or 16 bits, and the bus width for the byte control SRAM space is 16 bits.

The initial state of the bus width is specified by the operating mode.

If all areas are designated as 8-bit access space, 8-bit bus mode is set; if any area is desig 16-bit access space, 16-bit bus mode is set.

(3) Endian Format

Though the endian format of this LSI is big endian, data can be converted into little endia when reading or writing to the external address space.

Areas 7 to 2 can be specified as either big endian or little endian format by the LE7 to LE ENDIANCR.

The initial state of each area is the big endian format.

Note that the data format for the areas used as a program area or a stack area should be bi

Number of access cycles in the basic bus interface
= number of basic cycles (2, 3) + number of program wait cycles (0 to 7)
+ number of $\overline{\text{CS}}$ extension cycles (0, 1, 2)
[+ number of external wait cycles by the $\overline{\text{WAIT}}$ pin]

Assertion period of the chip select signal can be extended by CSACR.

(b) Byte Control SRAM Interface

The number of access cycles in the byte control SRAM interface is the same as that in th
bus interface.

Number of access cycles in byte control SRAM interface
= number of basic cycles (2, 3) + number of program wait cycles (0 to 7)
+ number of $\overline{\text{CS}}$ extension cycles (0, 1, 2)
[+ number of external wait cycles by the $\overline{\text{WAIT}}$ pin]

(c) Burst ROM Interface

The number of access cycles at full access in the burst ROM interface is the same as tha
basic bus interface. The number of access cycles in the burst access can be specified as o
eight cycles by the BSTS bit in BROMCR.

Number of access cycles in the burst ROM interface
= number of basic cycles (2, 3) + number of program wait cycles (0 to 7)
+ number of $\overline{\text{CS}}$ extension cycles (0, 1)
[+number of external wait cycles by the $\overline{\text{WAIT}}$ pin]
+ number of burst access cycles (1 to 8) $\times$ number of burst accesses (0 to 63)

(e) DRAM Interface

In the DRAM interface, the numbers of precharge cycles, row address output cycles, and address output cycles can be specified.

The number of precharge cycles can be specified as one to four cycles by bits TPC1 and DRACCR. The number of row address output cycles can be specified as one to four cycle RCD1 and RCD0 in DRACCR. The number of column address output cycles can be spec two or three cycles by the CAST bit in DRAMCR. For the column address output cycle, wait (0 to 7 cycles) specified by WTCRB or external wait by $\overline{\text{WAIT}}$ can be inserted.

Number of access cycles in the DRAM interface
= number of precharge cycles (1 to 4) + number of row address output cycles (1
+ number of column address output cycles (2 or 3)
+ number of program wait cycles (0 to 7)
[+number of external wait cycles by the WAIT pin]

(f) SDRAM Interface

In the SDRAM interface, the numbers of precharge cycles, row address output cycles, an address output cycles, as well as clock suspend and write-precharge delay, can be specifi DRACCR and WTCRB.

The number of precharge cycles can be specified as one to four cycles by bits TPC1 and DRACCR. The number of row address output cycles can be specified as one to four cycle RCD1 and RCD0 in DRACCR. The number of column address output cycles during read can be specified as two to four cycles by bits W21 and W20 in WTCRB.

The cycles for clock suspend and write-precharge delay can be inserted by bits CKSPE a TRWL in SDCR.

			[0,1]	[1]	[1]	[0 to 7]	[n]	[1]	[0,1]			
Byte-control SRAM interface		=	Th [0,1]	+T1 [1]	+T2 [1]				+Tt [0,1]			
		=	Th [0,1]	+T1 [1]	+T2 [1]	+Tpw [0 to 7]	+Ttw [n]	+T3 [1]	+Tt [0,1]			
Burst ROM interface		=	Th [0,1]	+T1 [1]	+T2 [1]					+Tb [(1 to 8) x m]		[(2 t
		=	Th [0,1]	+T1 [1]	+T2 [1]	+Tpw [0 to 7]	+Ttw [n]	+T3 [1]		+Tb [(1 to 8) x m]		[(2 to 11
Address/data multiplexed I/O interface		=Tma [2,3]	+Th [0,1]	+T1 [1]	+T2 [1]				+Tt [0,1]			
		=Tma [2,3]	+Th [0,1]	+T1 [1]	+T2 [1]	+Tpw [0 to 7]	+Ttw [n]	+T3 [1]	+Tt [0,1]			
DRAM interface	Full access	=Tp [1 to 4]	+Tr [1]	+Trw [0 to 3]	+TC1 [1]	+Tpw [0 to 7]	+Ttw [n]	+Tc2 [1]	+Tc3 [0,1]			
	Fast page	=			TC1 [1]	+Tpw [0 to 7]	+Ttw [n]	+Tc2 [1]	+Tc3 [0,1]			
	Refresh	=TRp [1 to 4]	+TRrw [0 to 3]	+TRr [1]	+TRc1 [1]	+TRcw [0 to 7]	+TRc2 [1]					
	Self-refresh	=TRp [1 to 4]	+TRrw [0 to 3]	+TRr [1]	+	Software standby mode [1+s]		+TRc3 [1]	+TRc4 [1]	+TRp [0 to 7]		
SDRAM interface	Setting mode register	=	Tp [1 to 4]	+Tr [1]	+Trw [0 to 3]	+Tc1 [1]				+Tc2 [1]	+Trwl [0,1]	
	Full access (read)	=	Tp [1 to 4]	+Tr [1]	+Trw [0 to 3]	+Tc1 [1]		+Tcl [1 to 3]	+Tsp [0,1]	+Tc2 [1]		
	Full access (write)	=	Tp [1 to 4]	+Tr [1]	+Trw [0 to 3]	+Tc1 [1]				+Tc2 [1]	+Trwl [0,1]	
	Page access (read)	=				Tc1 [1]		+Tcl [1 to 3]	+Tsp [0,1]	[1] +Tc2		
	Page access (write)	=				Tc1 [1]				[1] +Tc2		
	Cluster transfer (read)	=	Tp [1 to 4]	+Tr [1]	+Trw [0 to 3]	+Tc1 [1]	+Tcb [0 to 31]	+Tcl [1 to 3]		[1] +Tc2		+T [(
		=				Tc1 [1]	+Tcb [0 to 31]	+Tcl [1 to 3]		[1] +Tc2		
	Cluster transfer (write)	=	Tp [1 to 4]	+Tr [1]	+Trw [0 to 3]	+Tc1 [1]				[1] +Tc2	+Tcb [0 to 31]	
		=				Tc1 [1]				[1] +Tc2	+Tcb [0 to 31]	
	Refresh	=	TRp [1 to 4]	+TRr [1]	+TRc1 [1]	+TRcw [0 to 7]	+TRc2 [1]			[1]		
	Self-refresh	=	TRp [1 to 4]	+TRr [1]	+	Software standby mode [1+s]	+TRc2 [1]	+TRc3 [1]	+TRp [0 to 7]			

[Legend]
Number enclosed by bracket: Number of access cycles
n: Pin wait (0 to ∞)
m: Number of burst accesses (0 to 63)
s: Time for a transition to or from software standby mode

(1) Area 0

Area 0 includes on-chip ROM. All of area 0 is used as external address space in on-chip I
disabled extended mode, and the space excluding on-chip ROM is external address space
chip ROM enabled extended mode.

When area 0 external address space is accessed, the $\overline{CS0}$ signal can be output.

Either of the basic bus interface, byte control SRAM interface, or burst ROM interface ca
selected for area 0 by bit BSRM0 in BROMCR and bit BCSEL0 in SRAMCR. Table 6.7
the external interface of area 0.

Table 6.7 Area 0 External Interface

	Register Setting	
Interface	**BSRM0 of BROMCR**	**BCSEL0 of SRAMCR**
Basic bus interface	0	0
Byte control SRAM interface	0	1
Burst ROM interface	1	0
Setting prohibited	1	1

Interface	Register Setting	
	BSRM1 of BROMCR	BCSEL1 of SRAMCR
Basic bus interface	0	0
Byte control SRAM interface	0	1
Burst ROM interface	1	0
Setting prohibited	1	1

(3) Area 2

In externally extended mode, all of area 2 is external address space.

When area 2 external address space is accessed, the $\overline{CS2}$ signal can be output.

Either the basic bus interface, byte-control SRAM interface, DRAM interface, or SDRA interface can be selected for area 2 by the DRAME and DTYPE bits in DRAMCR and b BCSEL2 in SRAMCR. Table 6.9 shows the external interface of area 2.

Table 6.9 Area 2 External Interface

Interface	Register Setting		
	DRAME in DRAMCR	DTYPE in DRAMCR	BCSEL2 SRAMCR
Basic bus interface	0	Don't care	0
Byte-control SRAM interface	0	Don't care	1
DRAM interface	1	0	0
SDRAM interface	1	1	0
Setting prohibited	1	Don't care	1

Interface	Register Setting	
	MPXE3 of MPXCR	BCSEL3 of SRAMCR
Basic bus interface	0	0
Byte control SRAM interface	0	1
Address/data multiplexed I/O interface	1	0
Setting prohibited	1	1

(5) Area 4

In externally extended mode, all of area 4 is external address space.

When area 4 external address space is accessed, the $\overline{\text{CS4}}$ signal can be output.

Either of the basic bus interface, byte control SRAM interface, or address/data multiplexe interface can be selected for area 4 by bit MPXE4 in MPXCR and bit BCSEL4 in SRAM Table 6.11 shows the external interface of area 4.

Table 6.11 Area 4 External Interface

Interface	Register Setting	
	MPXE4 of MPXCR	BCSEL4 of SRAMCR
Basic bus interface	0	0
Byte control SRAM interface	0	1
Address/data multiplexed I/O interface	1	0
Setting prohibited	1	1

interface can be selected for area 5 by the MPXE5 bit in MPXCR and the BCSEL5 bit in SRAMCR. Table 6.12 shows the external interface of area 5.

Table 6.12 Area 5 External Interface

	Register Setting	
Interface	**MPXE5 of MPXCR**	**BCSEL5 of SRAMCR**
Basic bus interface	0	0
Byte control SRAM interface	0	1
Address/data multiplexed I/O interface	1	0
Setting prohibited	1	1

	Register Setting	
Interface	MPXE6 of MPXCR	BCSEL6 of SRAMCR
Basic bus interface	0	0
Byte control SRAM interface	0	1
Address/data multiplexed I/O interface	1	0
Setting prohibited	1	1

(8) Area 7

Area 7 includes internal I/O registers. In external extended mode, area 7 other than intern register area is external address space.

When area 7 external address space is accessed, the $\overline{\text{CS7}}$ signal can be output.

Either of the basic bus interface, byte control SRAM interface, or address/data multiplexe interface can be selected for area 7 by the MPXE7 bit in MPXCR and the BCSEL7 bit in SRAMCR. Table 6.14 shows the external interface of area 7.

Table 6.14 Area 7 External Interface

	Register Setting	
Interface	MPXE7 of MPXCR	BCSEL7 of SRAMCR
Basic bus interface	0	0
Byte control SRAM interface	0	1
Address/data multiplexed I/O interface	1	0
Setting prohibited	1	1

amount of data that can be accessed at one time is one byte: a word access is performed
byte accesses, and a longword access, as four byte accesses.

Figures 6.11 and 6.12 illustrate data alignment control for the 8-bit access space. Figure
shows the data alignment when the data endian format is specified as big endian. Figure
shows the data alignment when the data endian format is specified as little endian.

Data Size	Access Address	Access Count	Bus Cycle	Data Size	Strobe si LHWR/LUB RI Data bu D15 D8
Byte	n	1	1st	Byte	
Word	n	2	1st	Byte	
			2nd	Byte	
Longword	n	4	1st	Byte	
			2nd	Byte	
			3rd	Byte	
			4th	Byte	

Figure 6.11 Access Sizes and Data Alignment Control for 8-Bit Access Space (Bi

			2nd	Byte	15
			3rd	Byte	23
			4th	Byte	31

Figure 6.12 Access Sizes and Data Alignment Control for 8-Bit Access Spac (Little Endian)

(2) 16-Bit Access Space

With the 16-bit access space, the upper byte data bus (D15 to D8) and lower byte data bu D0) are used for accesses. The amount of data that can be accessed at one time is one byt word.

Figures 6.13 and 6.14 illustrate data alignment control for the 16-bit access space. Figure shows the data alignment when the data endian format is specified as big endian. Figure 6 shows the data alignment when the data endian format is specified as little endian.

In big endian, byte access for an even address is performed by using the upper byte data b byte access for an odd address is performed by using the lower byte data bus.

In little endian, byte access for an even address is performed by using the lower byte data byte access for an odd address is performed by using the third byte data bus.

Access Size	Access Address	Access Count	Bus Cycle	Data Size	D15–D8
Longword	Even (2n)	2	1st	Word	31–24
			2nd	Word	15–8
	Odd (2n+1)	3	1st	Byte	
			2nd	Word	23–16
			3rd	Byte	7–0

Figure 6.13 Access Sizes and Data Alignment Control for 16-Bit Access Space (Bi

Access Size	Access Address	Access Count	Bus Cycle	Data Size	Strobe s… LHWR/LUB … Data … D15 D8
Byte	Even (2n)	1	1st	Byte	
	Odd (2n+1)	1	1st	Byte	7–0
Word	Even (2n)	1	1st	Word	15–8
	Odd (2n+1)	2	1st	Byte	7–0
			2nd	Byte	
Longword	Even (2n)	2	1st	Word	15–8
			2nd	Word	31–24
	Odd (2n+1)	3	1st	Byte	7–0
			2nd	Word	23–16
			3rd	Byte	

Figure 6.14 Access Sizes and Data Alignment Control for 16-Bit Access Sp
(Little Endian)

or lower byte data bus (D7 to D0) is used according to the bus specifications for the area accessed (8-bit access space or 16-bit access space), the data size, and endian format whe accessing external address space,. For details, see section 6.5.6, Endian and Data Alignm

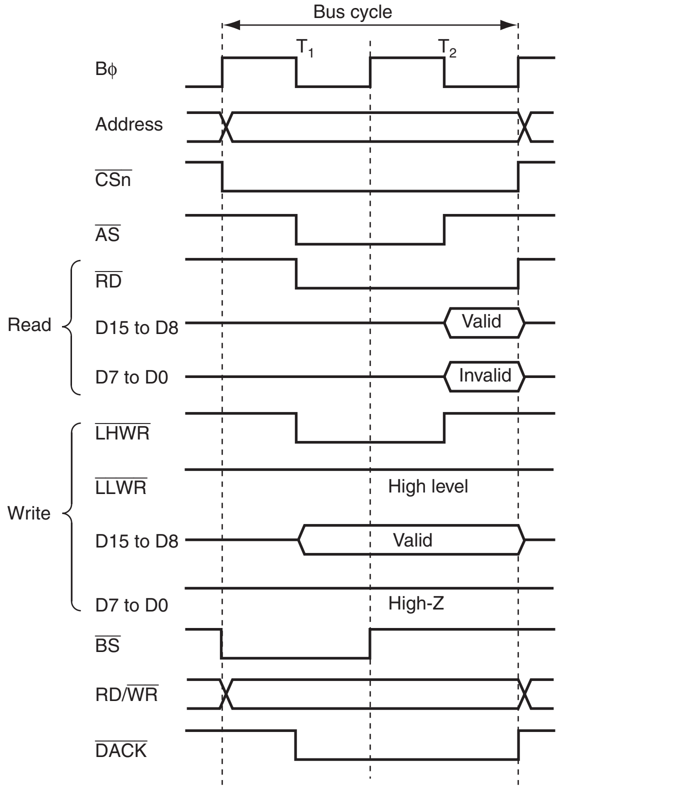
6.6.2 I/O Pins Used for Basic Bus Interface

Table 6.15 shows the pins used for basic bus interface.

Table 6.15 I/O Pins for Basic Bus Interface

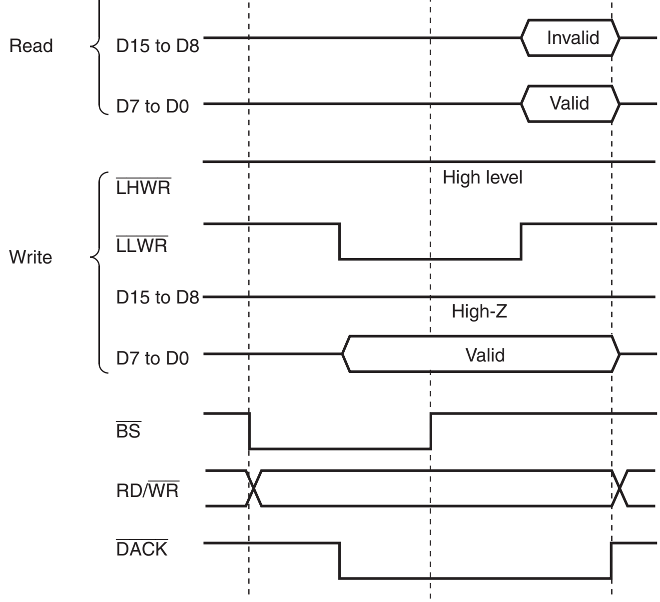
Name	Symbol	I/O	Function
Bus cycle start	$\overline{\text{BS}}$	Output	Signal indicating that the bus cycle has start
Address strobe	$\overline{\text{AS}}$*	Output	Strobe signal indicating that an address outp address bus is valid during access
Read strobe	$\overline{\text{RD}}$	Output	Strobe signal indicating the read access
Read/write	RD/$\overline{\text{WR}}$	Output	Signal indicating the data bus input or outpu direction
Low-high write	$\overline{\text{LHWR}}$	Output	Strobe signal indicating that the upper byte (D8) is valid during write access
Low-low write	$\overline{\text{LLWR}}$	Output	Strobe signal indicating that the lower byte (D0) is valid during write access
Chip select 0 to 7	$\overline{\text{CS0}}$ to $\overline{\text{CS7}}$	Output	Strobe signal indicating that the area is sele
Wait	$\overline{\text{WAIT}}$	Input	Wait request signal used when an external a space is accessed

Note: * When the address/data multiplexed I/O is selected, this pin only functions as th output and does not function as the $\overline{\text{AS}}$ output.



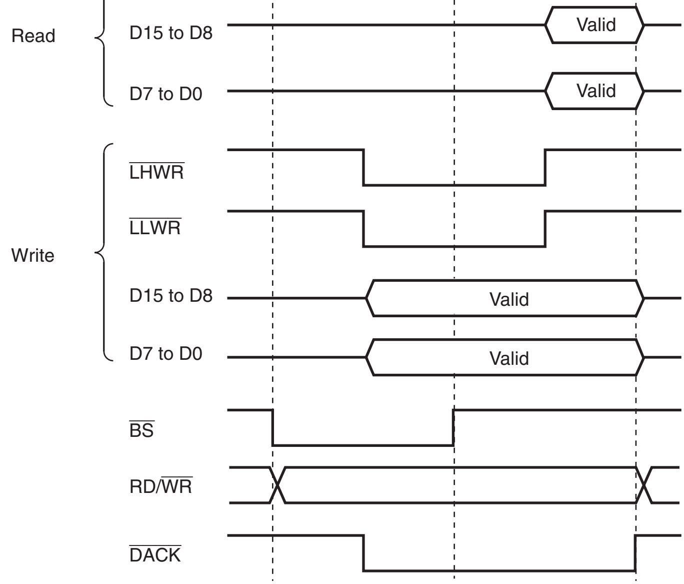
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

Figure 6.15 16-Bit 2-State Access Space Bus Timing (Byte Access for Even Ad



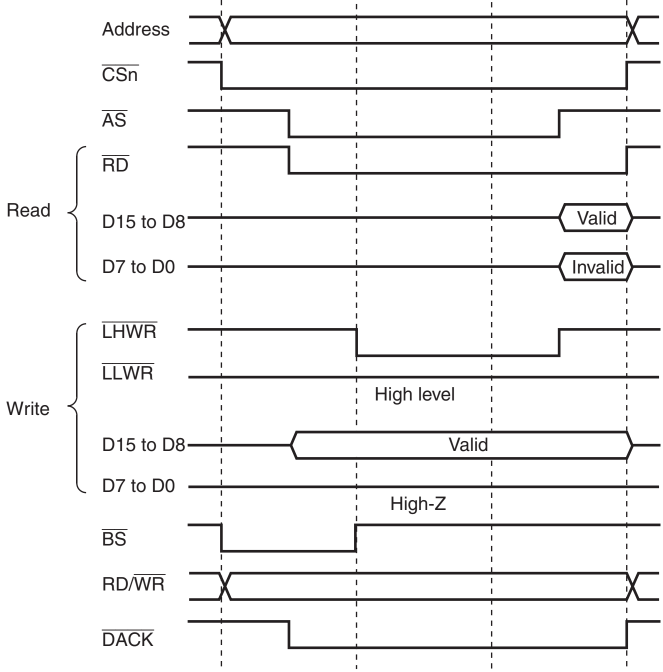
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

Figure 6.16 16-Bit 2-State Access Space Bus Timing (Byte Access for Odd Add



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

Figure 6.17 16-Bit 2-State Access Space Bus Timing (Word Access for Even A



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

Figure 6.18 16-Bit 3-State Access Space Bus Timing (Byte Access for Even Add

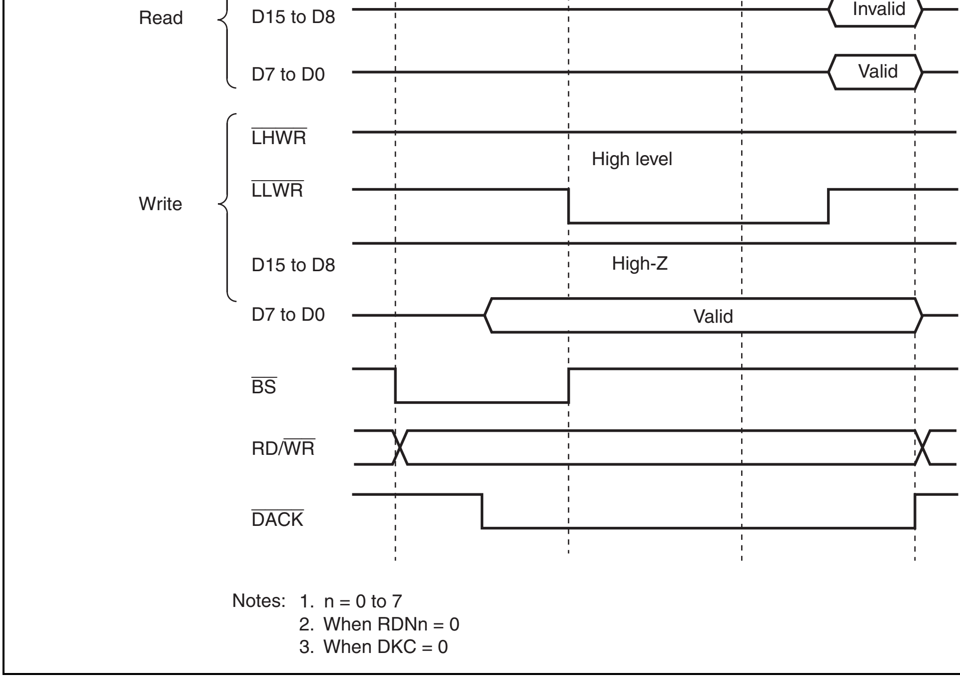
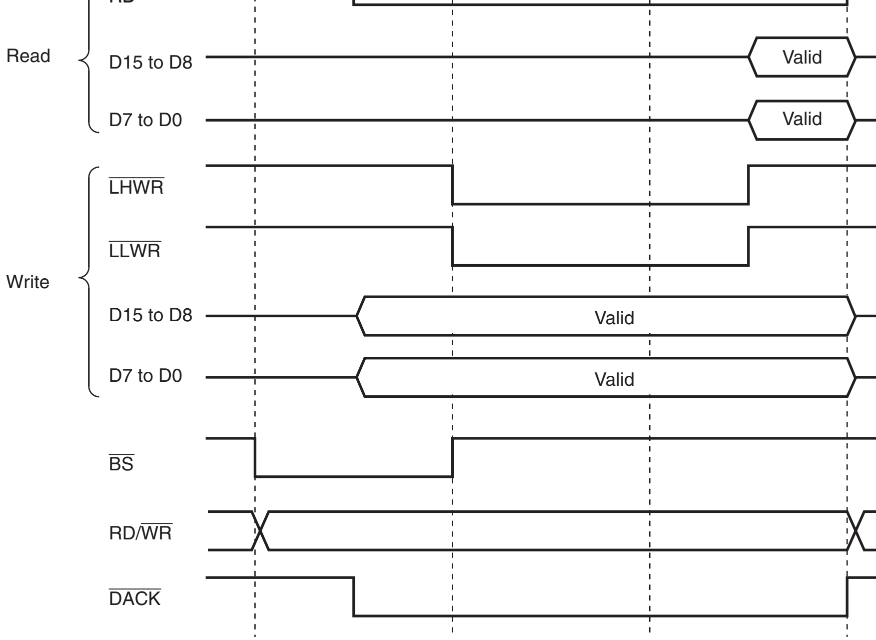


Figure 6.19 16-Bit 3-State Access Space Bus Timing (Word Access for Odd Ad

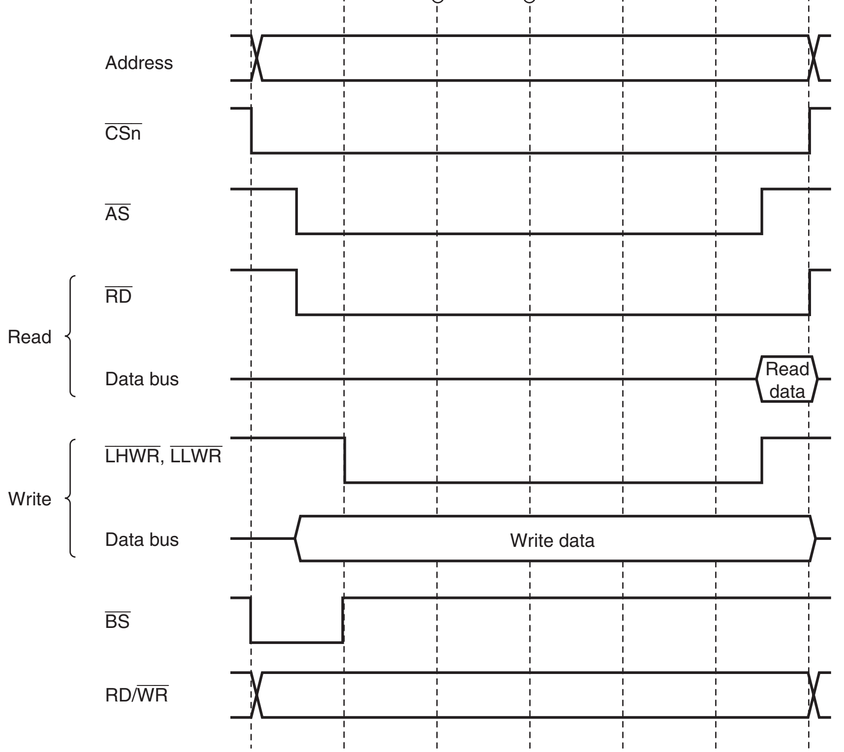


Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

Figure 6.20 16-Bit 3-State Access Space Bus Timing (Word Access for Even Ad

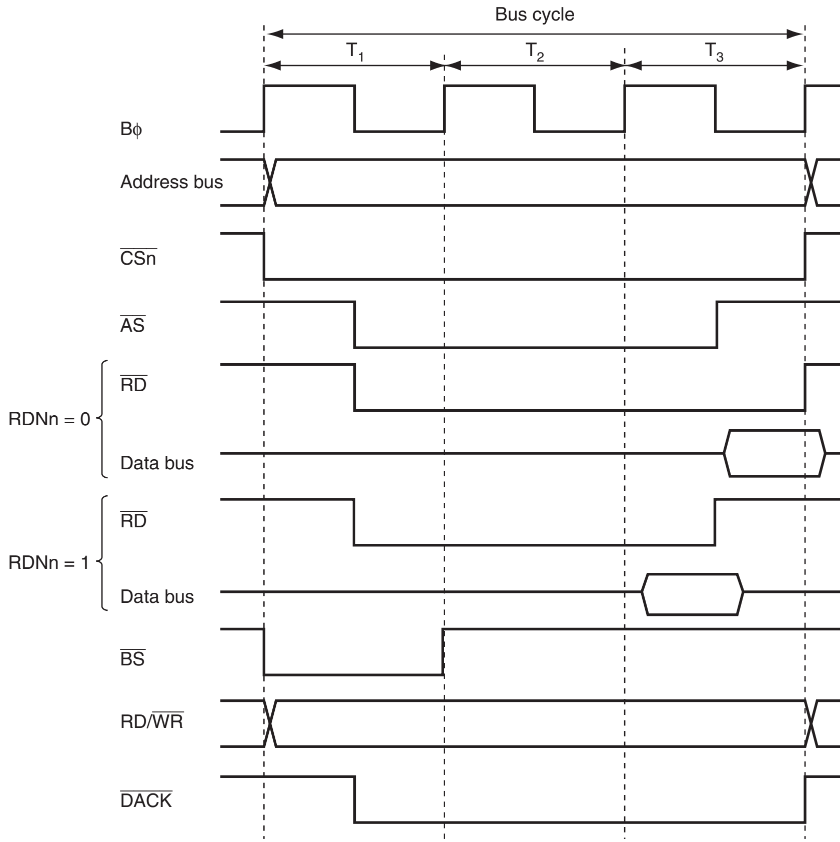
(2) Pin Wait Insertion

For 3-state access space, when the WAITE bit in BCR1 is set to 1 and the corresponding
is set to 1, wait input by means of the $\overline{\text{WAIT}}$ pin is enabled. When the external address s
accessed in this state, a program wait (Tpw) is first inserted according to the WTCRA ar
WTCRB settings. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of Bϕ in the last T2 or Tpw
another Ttw cycle is inserted until the $\overline{\text{WAIT}}$ pin is brought high. The pin wait insertion
effective when the Tw cycles are inserted to seven cycles or more, or when the number c
cycles to be inserted is changed according to the external devices. The WAITE bit is cor
all areas. For details on ICR, see section 9, I/O Ports.



Notes: 1. Upward arrows indicate the timing of WAIT pin sampling.
2. n = 0 to 7
3. When RDNn = 0

Figure 6.21 Example of Wait Cycle Insertion Timing



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

Figure 6.22 Example of Read Strobe Timing

3-state access space.

Both extension cycle Th inserted before the basic bus cycle and extension cycle Tt inserte the basic bus cycle, or only one of these, can be specified for individual areas. Insertion o insertion can be specified for the Th cycle with the upper eight bits (CSXH7 to CSXH0) i CSACR, and for the Tt cycle with the lower eight bits (CSXT7 to CSXT0).

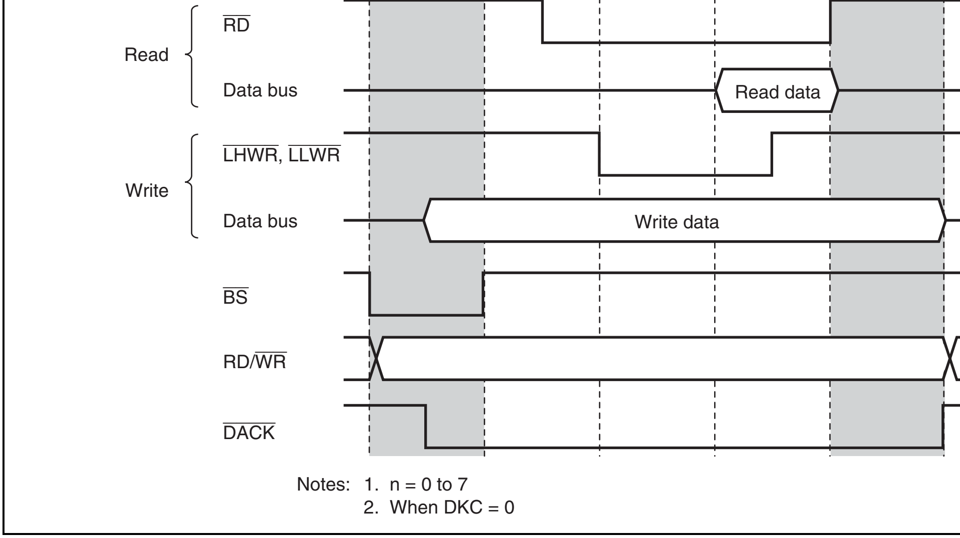


Figure 6.23 Example of Timing when Chip Select Assertion Period is Exten

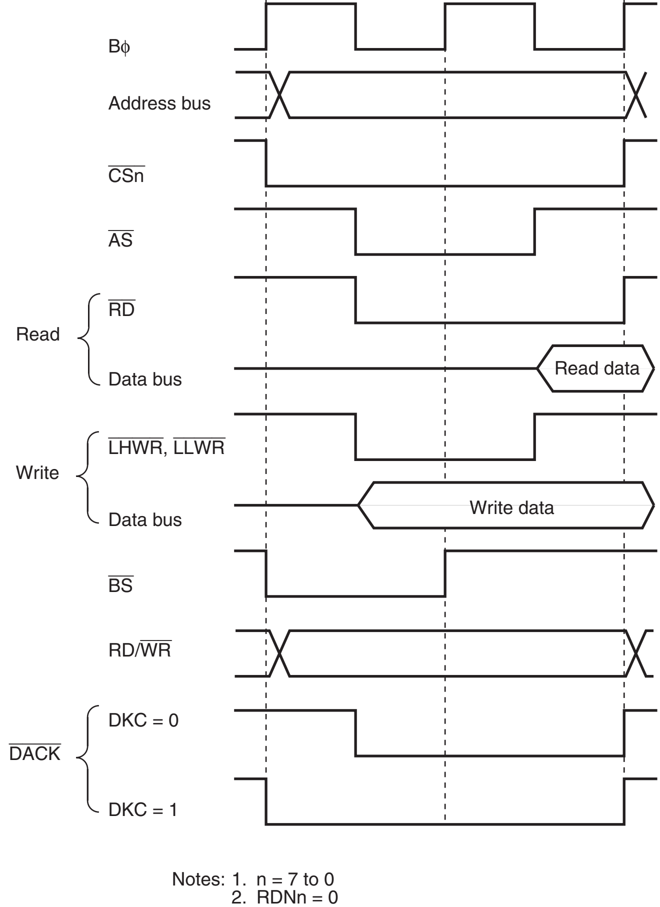


Figure 6.24 DACK Signal Output Timing

6.7.1 Byte Control SRAM Space Setting

Byte control SRAM interface can be specified for areas 0 to 7. Each area can be specifie control SRAM interface by setting bits BCSELn (n = 0 to 7) in SRAMCR. For the area as burst ROM interface or address/data multiplexed I/O interface, the SRAMCR setting and byte control SRAM interface cannot be used.

6.7.2 Data Bus

The bus width of the byte control SRAM space can be specified as 16-bit byte control S space according to bits ABWHn and ABWLn (n = 0 to 7) in ABWCR. The area specifie access space cannot be specified as the byte control SRAM space.

For the 16-bit byte control SRAM space, data bus (D15 to D0) is valid.

Access size and data alignment are the same as the basic bus interface. For details, see s 6.5.6, Endian and Data Alignment.

AS/AH	AS	Address strobe	Output	Strobe signal indicating that the ad output on the address bus is valid w basic bus interface space or byte c SRAM space is accessed
$\overline{CSn}$	$\overline{CSn}$	Chip select	Output	Strobe signal indicating that area n selected
$\overline{RD}$	$\overline{RD}$	Read strobe	Output	Output enable for the SRAM when control SRAM space is accessed
RD/$\overline{WR}$	RD/$\overline{WR}$	Read/write	Output	Write enable signal for the SRAM w byte control SRAM space is access
$\overline{LHWR}$/$\overline{LUB}$	$\overline{LUB}$	Lower-upper byte select	Output	Upper byte select when the 16-bit b control SRAM space is accessed
$\overline{LLWR}$/$\overline{LLB}$	$\overline{LLB}$	Lower-lower byte select	Output	Lower byte select when the 16-bit b control SRAM space is accessed
$\overline{WAIT}$	$\overline{WAIT}$	Wait	Input	Wait request signal used when an e address space is accessed
A23 to A0	A23 to A0	Address pin	Output	Address output pin
D15 to D0	D15 to D0	Data pin	Input/ output	Data input/output pin

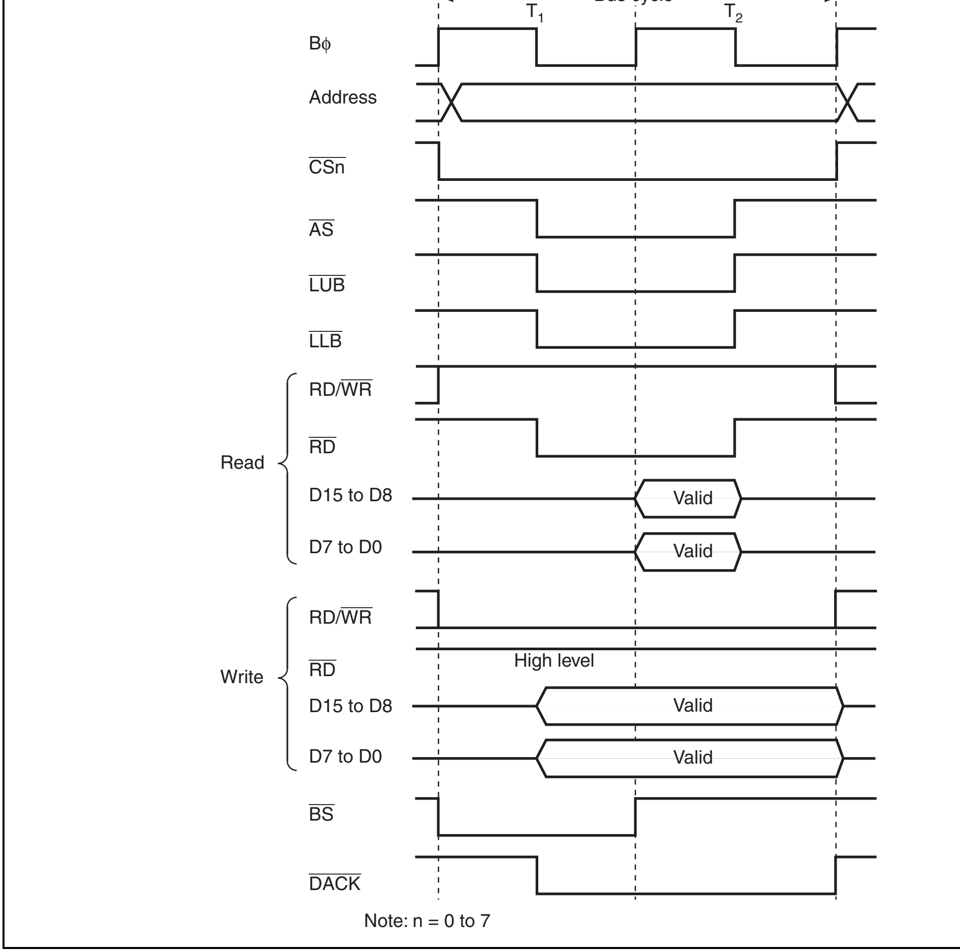


Figure 6.25 16-Bit 2-State Access Space Bus Timing

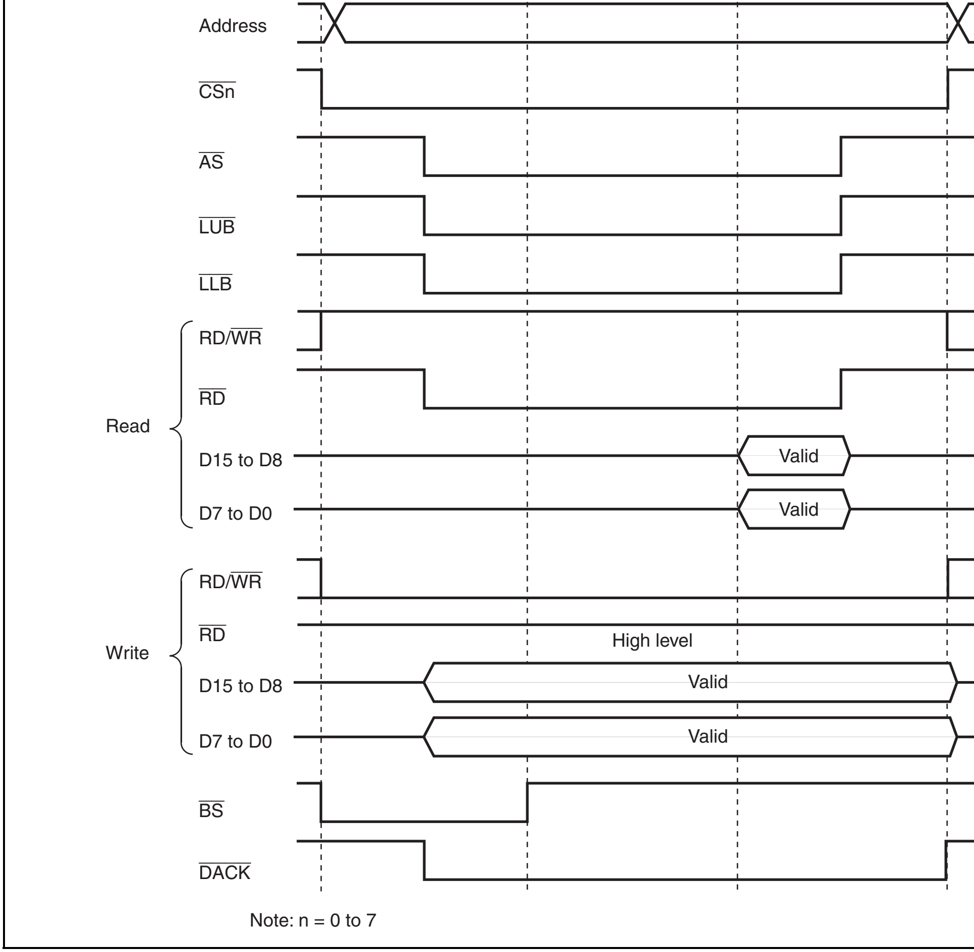
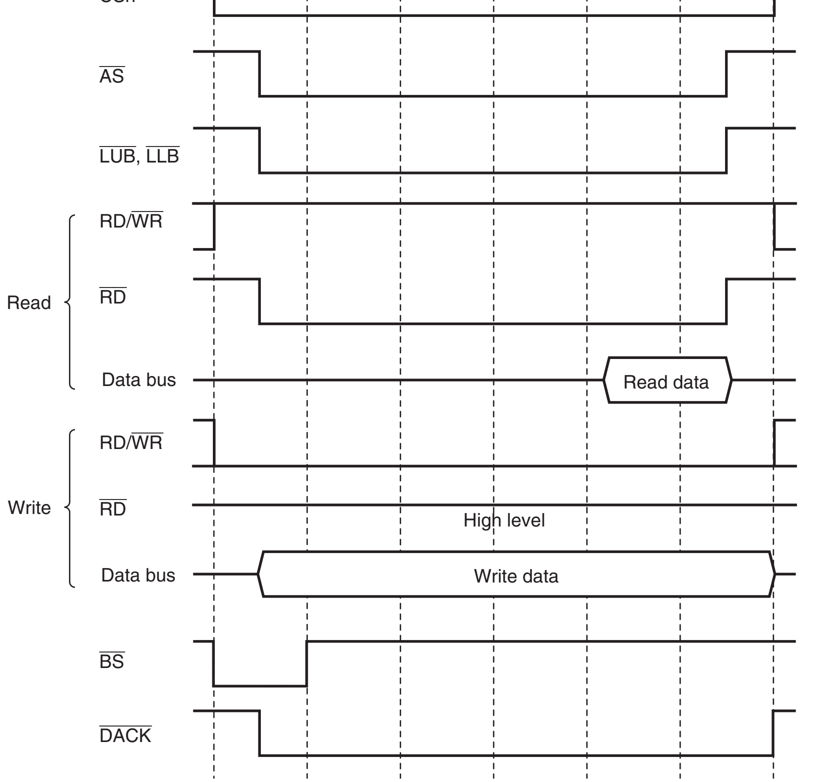


Figure 6.26 16-Bit 3-State Access Space Bus Timing

For 3-state access space, when the WAITE bit in BCR1 is set to 1, the corresponding DI
cleared to 0, and the ICR bit is set to 1, wait input by means of the $\overline{\text{WAIT}}$ pin is enabled
details on DDR and ICR, refer to section 9, I/O Ports.

Figure 6.27 shows an example of wait cycle insertion timing.



Notes: 1. Upward arrows indicate the timing of WAIT pin sampling.
2. n = 0 to 7

Figure 6.27 Example of Wait Cycle Insertion Timing

In the byte control SRAM interface, the extension cycles can be inserted before and afte
cycle in the same way as the basic bus interface. For details, refer to section 6.6.6, Exter
Chip Select ($\overline{CS}$) Assertion Period.

6.7.8 $\overline{DACK}$ Signal Output Timing

For DMAC single address transfers, the $\overline{DACK}$ signal assert timing can be modified by
DKC bit in BCR1.

Figure 6.28 shows the $\overline{DACK}$ signal output timing. Setting the DKC bit to 1 asserts the
signal a half cycle earlier.

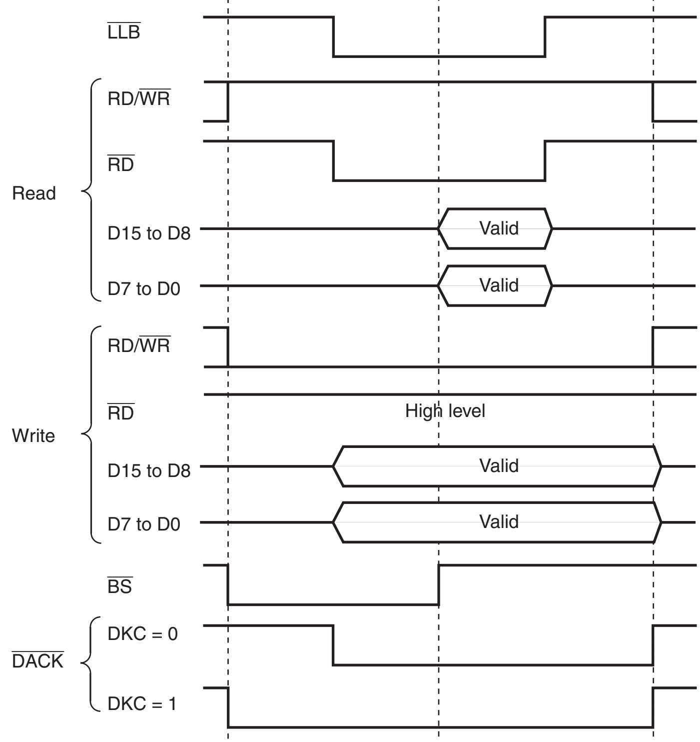


Figure 6.28 DACK Signal Output Timing

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

In the burst ROM interface, the burst access covers only CPU read accesses. Other acces
performed with the similar method to the basic bus interface.

6.8.1 Burst ROM Space Setting

Burst ROM interface can be specified for areas 0 and 1. Areas 0 and 1 can be specified a
ROM space by setting bits BSRMn (n = 0, 1) in BROMCR.

6.8.2 Data Bus

The bus width of the burst ROM space can be specified as 8-bit or 16-bit burst ROM int
space according to the ABWHn and ABWLn bits (n = 0, 1) in ABWCR.

For the 8-bit bus width, data bus (D7 to D0) is valid. For the 16-bit bus width, data bus (
D0) is valid.

Access size and data alignment are the same as the basic bus interface. For details, see s
6.5.6, Endian and Data Alignment.

Read strobe	$\overline{RD}$	Output	Strobe signal indicating the read access
Read/write	RD/$\overline{WR}$	Output	Signal indicating the data bus input or outpu direction
Low-high write	$\overline{LHWR}$	Output	Strobe signal indicating that the upper byte (D8) is valid during write access
Low-low write	$\overline{LLWR}$	Output	Strobe signal indicating that the lower byte (D0) is valid during write access
Chip select 0 and 1	$\overline{CS0}$, $\overline{CS1}$	Output	Strobe signal indicating that the area is selec
Wait	$\overline{WAIT}$	Input	Wait request signal used when an external a space is accessed

The basic access timing for burst ROM space is shown in figures 6.29 and 6.30.

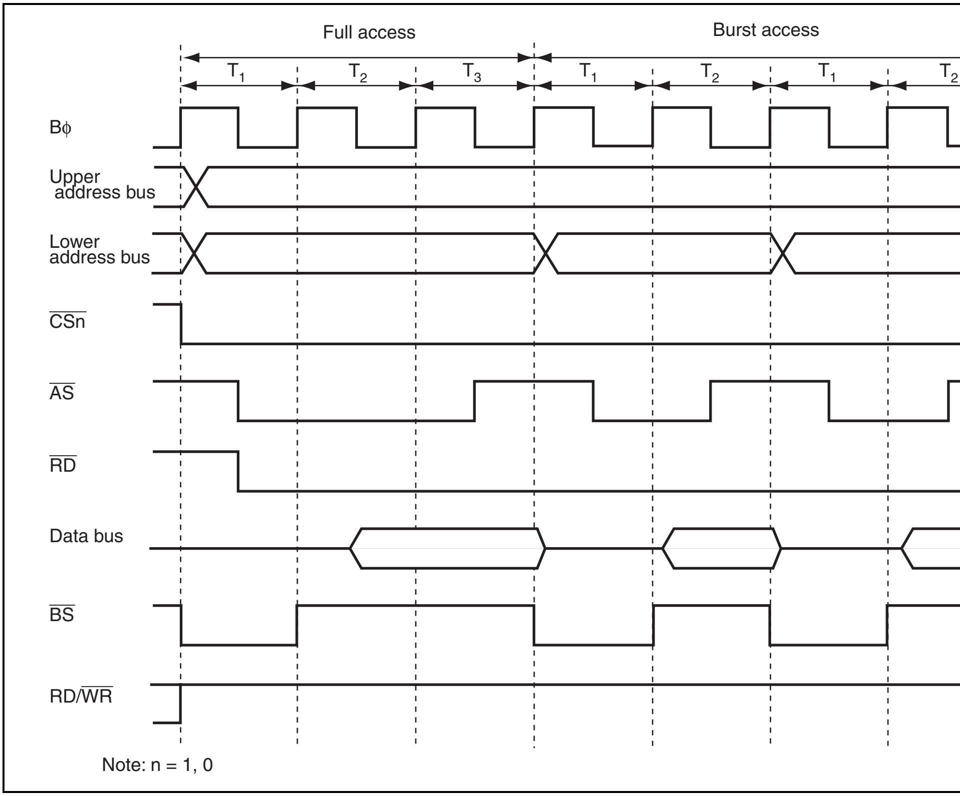


Figure 6.29 Example of Burst ROM Access Timing (ASTn = 1, Two Burst Cy

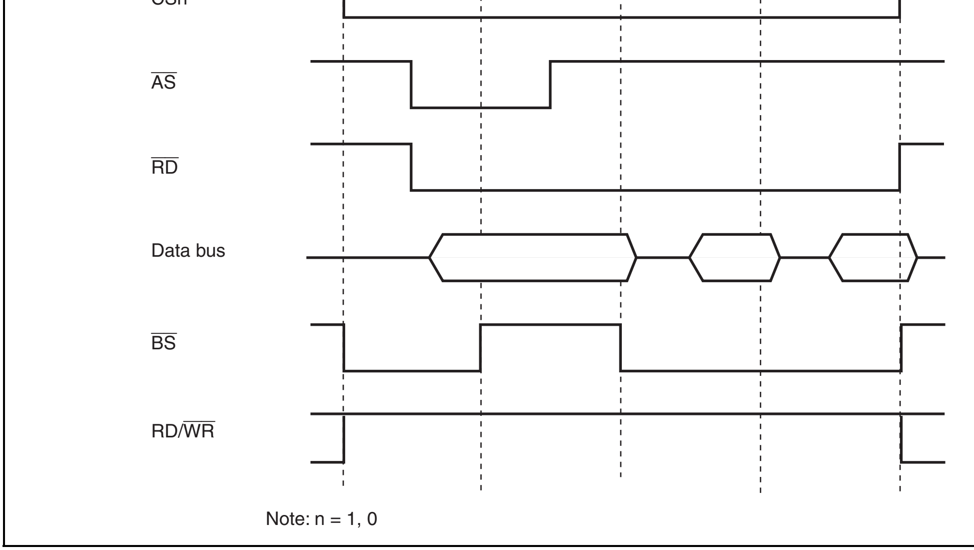


Figure 6.30 Example of Burst ROM Access Timing (ASTn = 0, One Burst Cy

The read strobe negation timing is the same timing as when RDNn = 0 in the basic bus i

6.8.7 Extension of Chip Select ($\overline{CS}$) Assertion Period

In the burst ROM interface, the extension cycles can be inserted in the same way as the
interface.

For the burst ROM space, the burst access can be enabled only in read access by the CP
case, the setting of the corresponding CSXTn bit in CSACR is ignored and an extension
be inserted only before the full access cycle. Note that no extension cycle can be inserte
after the burst access cycles.

In accesses other than read accesses by the CPU, the burst ROM space is equivalent to t
bus interface space. Accordingly, extension cycles can be inserted before and after the b
cycles.

specified as the address/data multiplexed I/O space by setting bits MPXEn (n = 3 to 7) in MPXCR.

6.9.2 Address/Data Multiplex

In the address/data multiplexed I/O space, data bus is multiplexed with address bus. Tabl shows the relationship between the bus width and address output.

Table 6.18 Address/Data Multiplex

Bus Width	Cycle	Data Pins													
		PI7	PI6	PI5	PI4	PI3	PI2	PI1	PI0	PH7	PH6	PH5	PH4	PH3	PH2
8 bits	Address	-	-	-	-	-	-	-	-	A7	A6	A5	A4	A3	A2
	Data	-	-	-	-	-	-	-	-	D7	D6	D5	D4	D3	D2
16 bits	Address	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2
	Data	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2

6.9.3 Data Bus

The bus width of the address/data multiplexed I/O space can be specified for either 8-bit a space or 16-bit access space by the ABWHn and ABWLn bits (n = 3 to 7) in ABWCR.

For the 8-bit access space, D7 to D0 are valid for both address and data. For 16-bit access D15 to D0 are valid for both address and data. If the address/data multiplexed I/O space i accessed, the corresponding address will be output to the address bus.

For details on access size and data alignment, see section 6.5.6, Endian and Data Alignme

				address/data multiplexed I/O space
$\overline{AS}$/$\overline{AH}$	$\overline{AH}$*	Address hold	Output	Signal to hold an address when the addres multiplexed I/O space is specified
$\overline{RD}$	$\overline{RD}$	Read strobe	Output	Signal indicating that the address/data mult space is being read
$\overline{LHWR}$/$\overline{LUB}$	$\overline{LHWR}$	Low-high write	Output	Strobe signal indicating that the upper byte D8) is valid when the address/data multiple space is written
$\overline{LLWR}$/$\overline{LLB}$	$\overline{LLWR}$	Low-low write	Output	Strobe signal indicating that the lower byte is valid when the address/data multiplexed written
D15 to D0	D15 to D0	Address/data	Input/ output	Address and data multiplexed pins for the address/data multiplexed I/O space. Only D7 to D0 are valid when the 8-bit spac specified. D15 to D0 are valid when the 16- specified.
A23 to A0	A23 to A0	Address	Output	Address output pin
$\overline{WAIT}$	$\overline{WAIT}$	Wait	Input	Wait request signal used when the external space is accessed
$\overline{BS}$	$\overline{BS}$	Bus cycle start	Output	Signal to indicate the bus cycle start
RD/$\overline{WR}$	RD/$\overline{WR}$	Read/write	Output	Signal indicating the data bus input or outp

Note: * The $\overline{AH}$ output is multiplexed with the $\overline{AS}$ output. At the timing that an area is as address/data multiplexed I/O, this pin starts to function as the $\overline{AH}$ output m that this pin cannot be used as the $\overline{AS}$ output. At this time, when other areas basic bus interface is accessed, this pin does not function as the $\overline{AS}$ output. U area is specified as address/data multiplexed I/O, be aware that this pin funct the $\overline{AS}$ output.

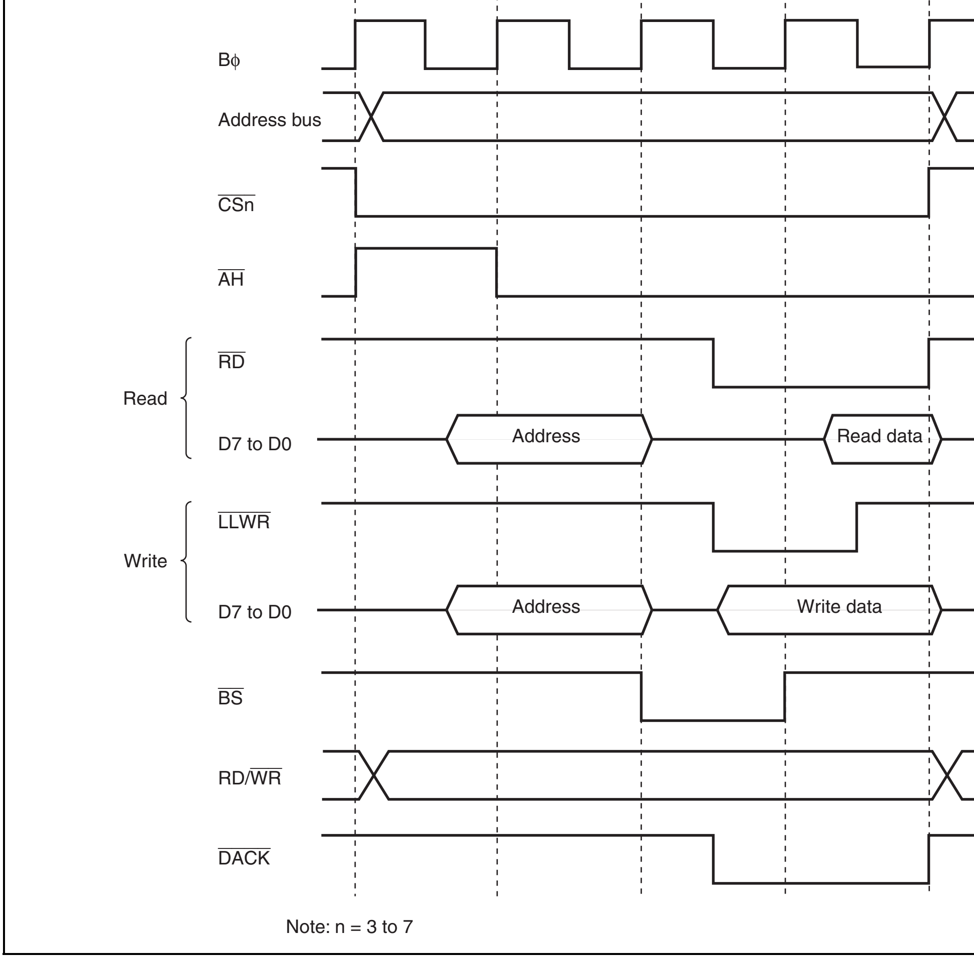


Figure 6.31 8-Bit Access Space Access Timing (ABWHn = 1, ABWLn = 1)

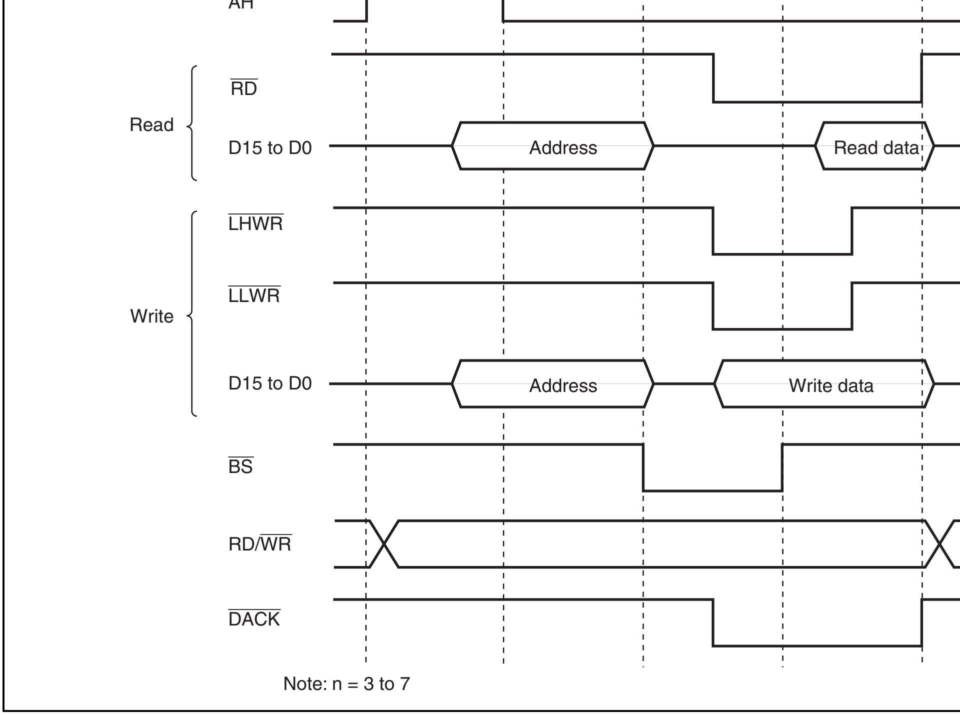


Figure 6.32 16-Bit Access Space Access Timing (ABWHn = 0, ABWLn =

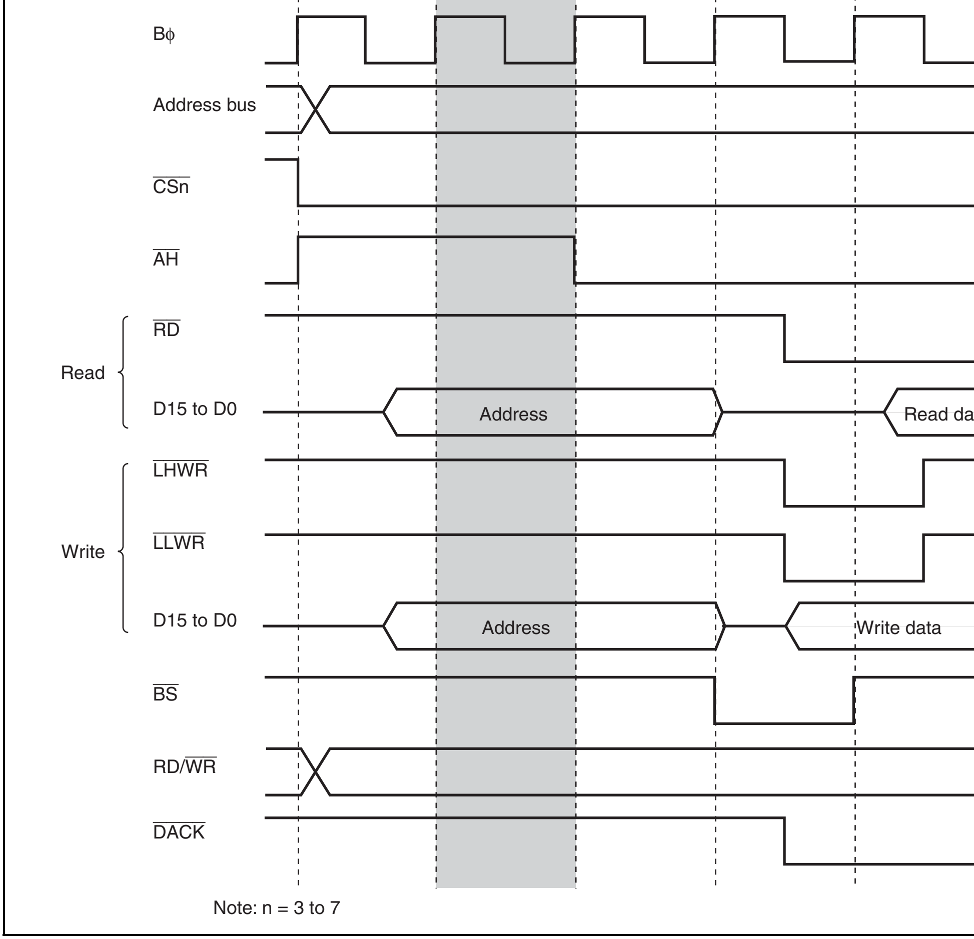


Figure 6.33 Access Timing of 3 Address Cycles (ADDEX = 1)

In the address/data multiplexed I/O interface, the read strobe timing of data cycles can b
in the same way as in basic bus interface. For details, refer to section 6.6.5, Read Strobe
Timing.

Figure 6.34 shows an example when the read strobe timing is modified.

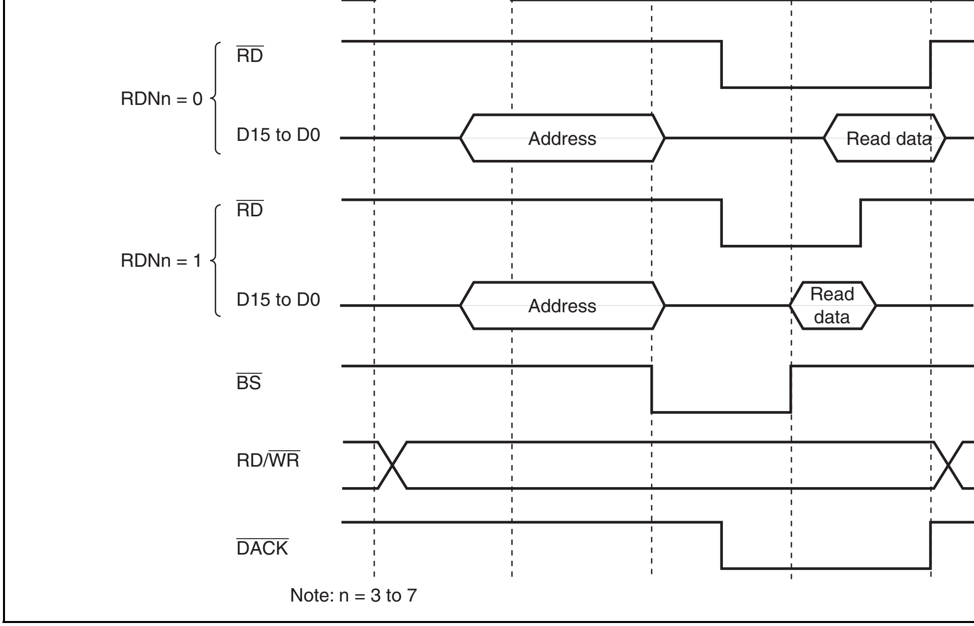


Figure 6.34 Read Strobe Timing

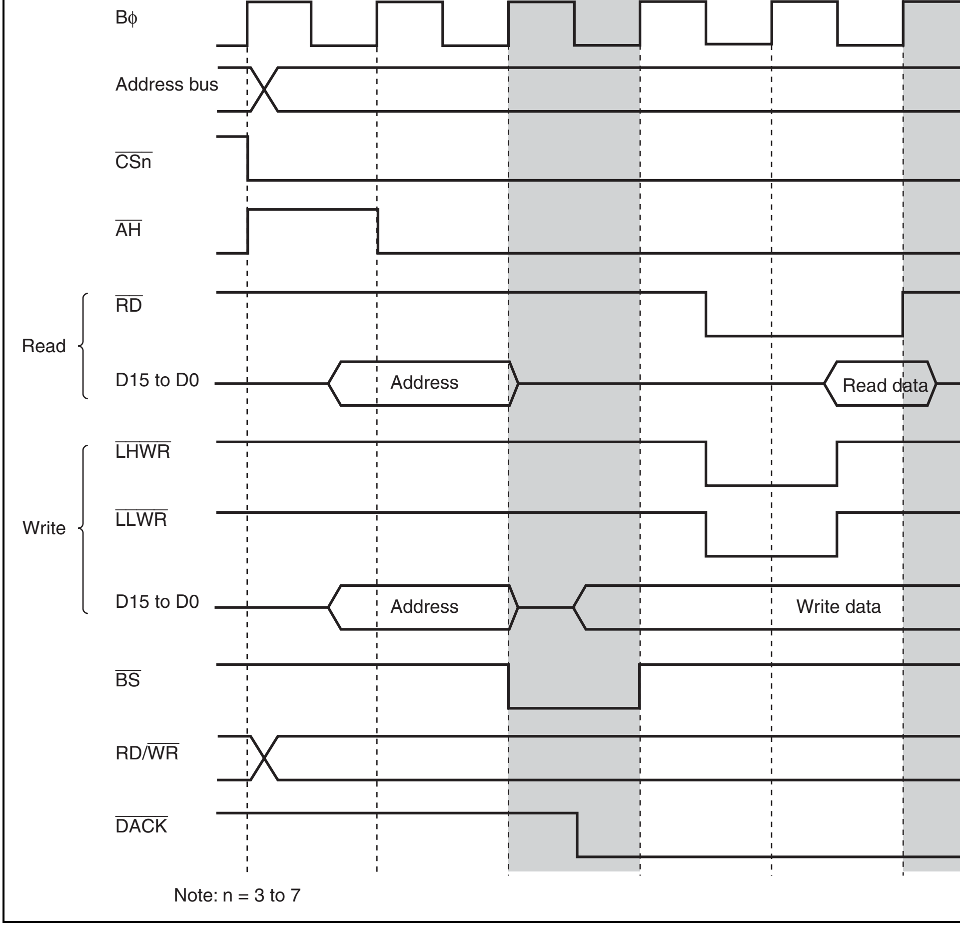


Figure 6.35 Chip Select ($\overline{CS}$) Assertion Period Extension Timing in Data Cy

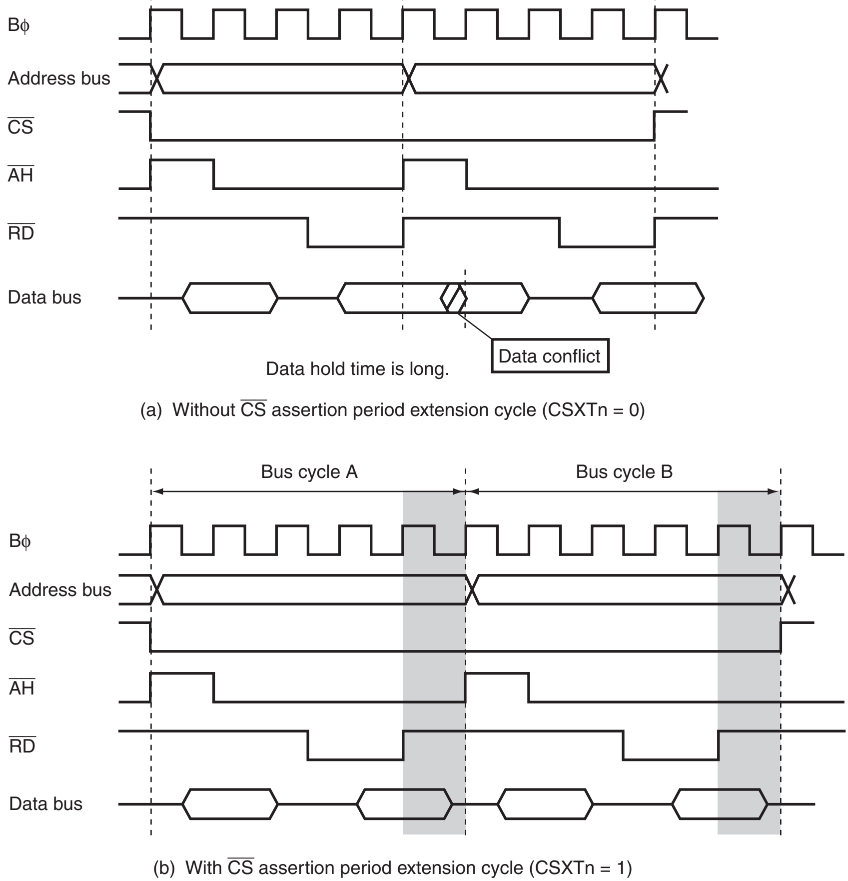


Figure 6.36 Consecutive Read Accesses to Same Area (Address/Data Multiplexed I/O Space)

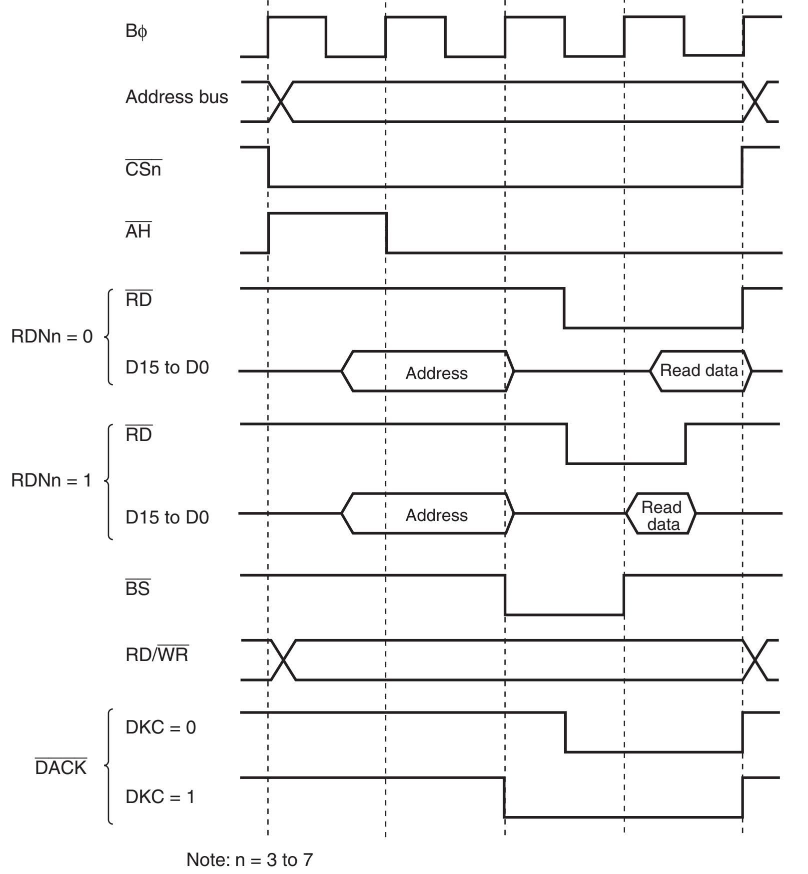


Figure 6.37 $\overline{\text{DACK}}$ Signal Output Timing

settings.

Table 6.20 Relationship Among DRAME and DTYPE and Area 2 Interfaces

DRAME	DTYPE	Area 2 Interface
0	×	Basic bus space (initial state)/byte-control SRAM space
1	0	DRAM space
1	1	SDRAM space

[Legend]

×: Don't care

6.10.2 Address Multiplexing

A Row address and a column address are multiplexed in the DRAM space. Select the nur
row address bits to be shifted with bits MXC1 and MXC0 in DRAMCR. Table 6.21 lists
relationship among bits MXC1 and MXC0 and shifted bit number.

Table 6.21 Relationship Among MXC1 and MXC0 and Shifted Bit Count

DRAMCR		Shit Bit Count	Data Bus Width	Address	External Address Pin															
MXC1	MXC0				A27 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3
0	0	8 bits	8/16 bits	Row address	A23 to A18	A17	-	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
				Column address	A23 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3
0	1	9 bits	8/16 bits	Row address	A23 to A18	A17	-	-	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
				Column address	A23 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3
1	0	10 bits	8/16 bits	Row address	A23 to A18	A17	-	-	-	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
				Column address	A23 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3
1	1	11 bits	8/16 bits	Row address	A23 to A18	A17	-	-	-	-	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
				Column address	A23 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3

6.10.4 I/O Pins Used for DRAM Interface

Table 6.22 shows the pins used for the DRAM interface.

Table 6.22 I/O Pins for DRAM Interface

Pin	DRAM Selected	Name	I/O	Function
$\overline{\text{WE}}$	$\overline{\text{WE}}$	Write enable	Output	Write enable signal for accessing DRAM interface
$\overline{\text{RAS}}$	$\overline{\text{RAS}}$	Row address strobe	Output	Row address strobe when the DR space is specified as area 2
$\overline{\text{LUCAS}}$/ DQMLU	$\overline{\text{LUCAS}}$	Lower-upper column address strobe	Output	• Lower-upper column address when the 32-bit DRAM space accessed • Upper column address strobe 16-bit DRAM space is access
$\overline{\text{LLCAS}}$/ DQMLL	$\overline{\text{LLCAS}}$	Lower-lower column address strobe	Output	• Lower-lower column address when the 32-bit DRAM space accessed • Lower column address strobe 16-bit DRAM space is access
$\overline{\text{OE}}$	$\overline{\text{OE}}$	Output enable	Output	Output enable signal when the D space is accessed
$\overline{\text{WAIT}}$	$\overline{\text{WAIT}}$	Wait	Input	Wait request signal used when a address space is accessed
A17 to A0	A17 to A0	Address pin	Output	Multiplexed address/data output
D15 to D0	D15 to D0	Data pin	Input/ output	Data input/output pin

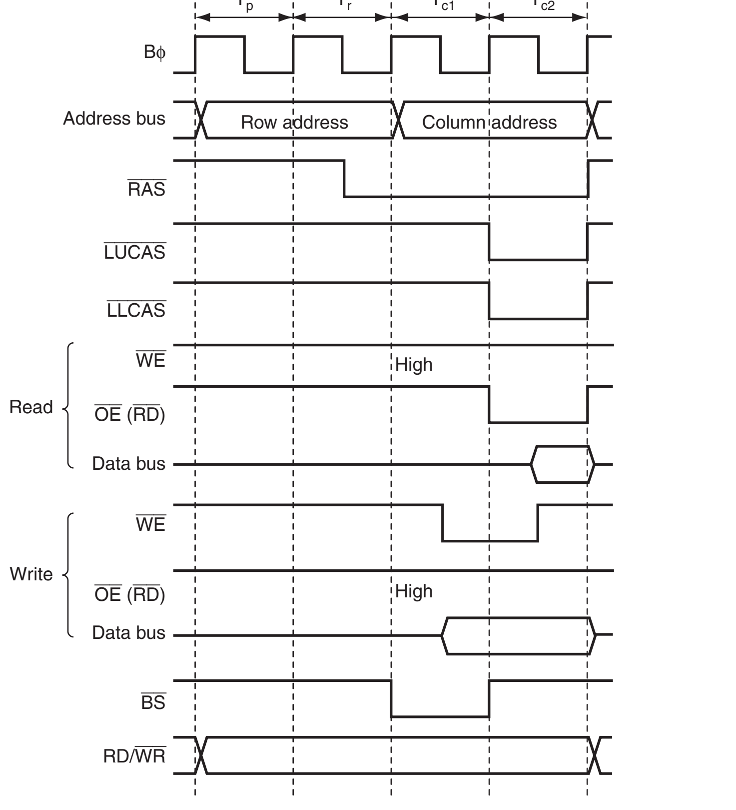


Figure 6.38 DRAM Basic Access Timing (RAS = 0 and CAST = 0)

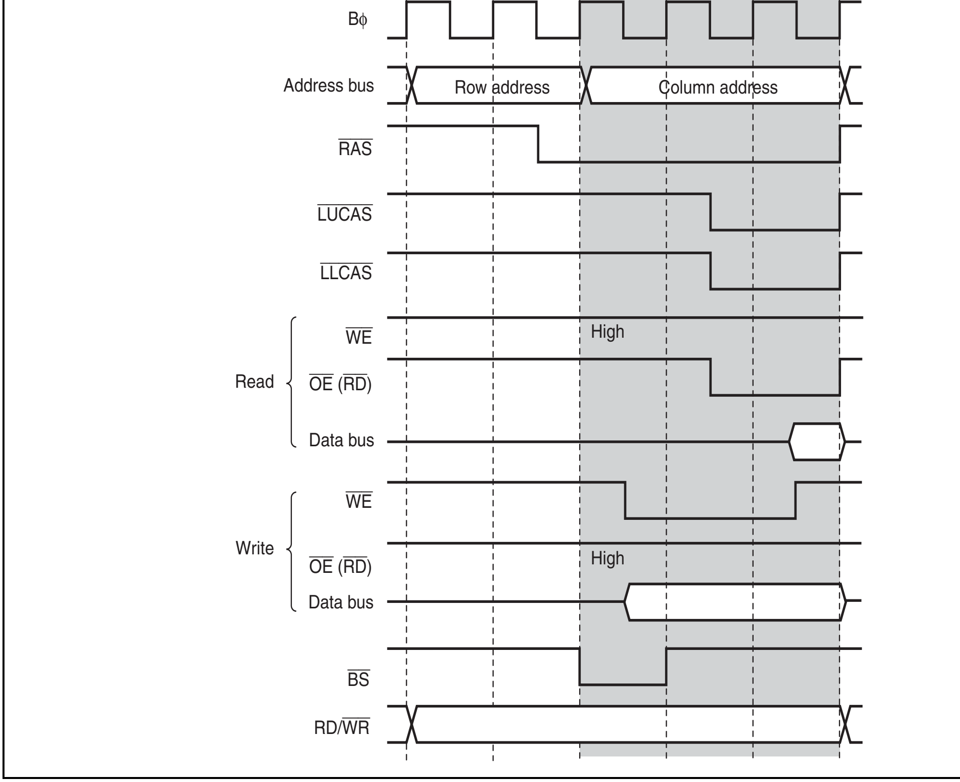


Figure 6.39 Access Timing Example of Column Address Output Cycles for 3 Clo
(RAST = 0)

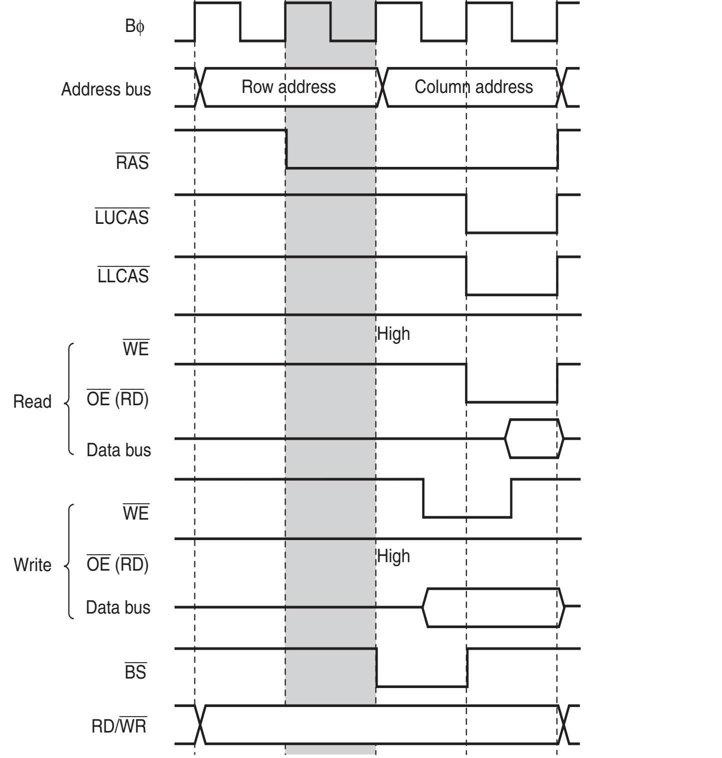


Figure 6.40 Access Timing Example of $\overline{RAS}$ Signal Driven Low at Start of Tr C
(CAST = 0)

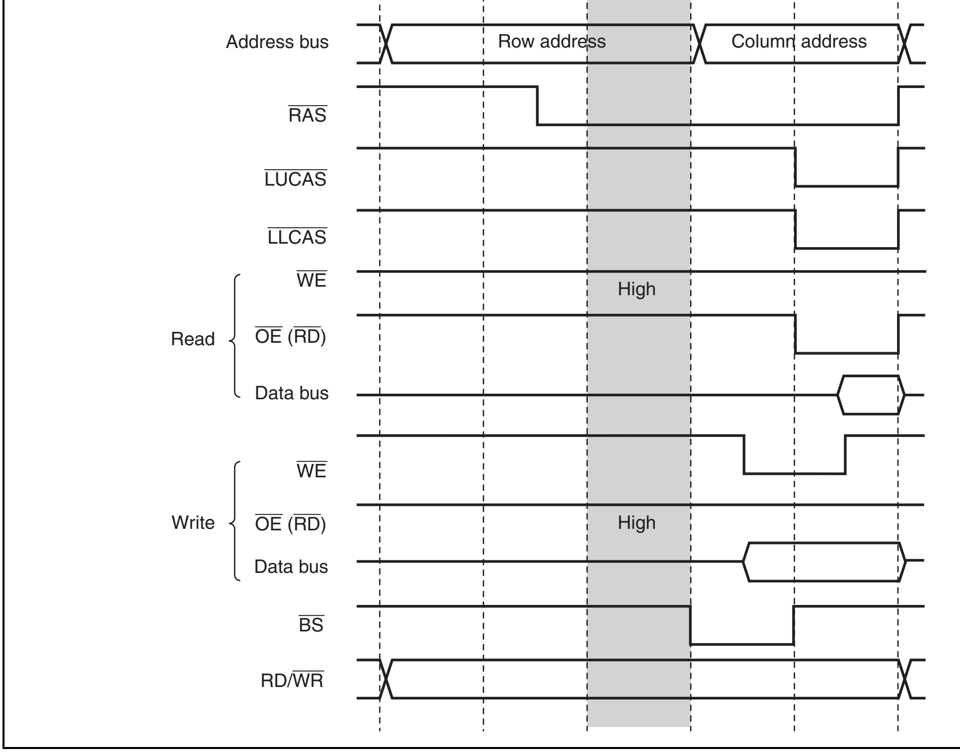


Figure 6.41 Access Timing Example when One Trw Cycle is Specified

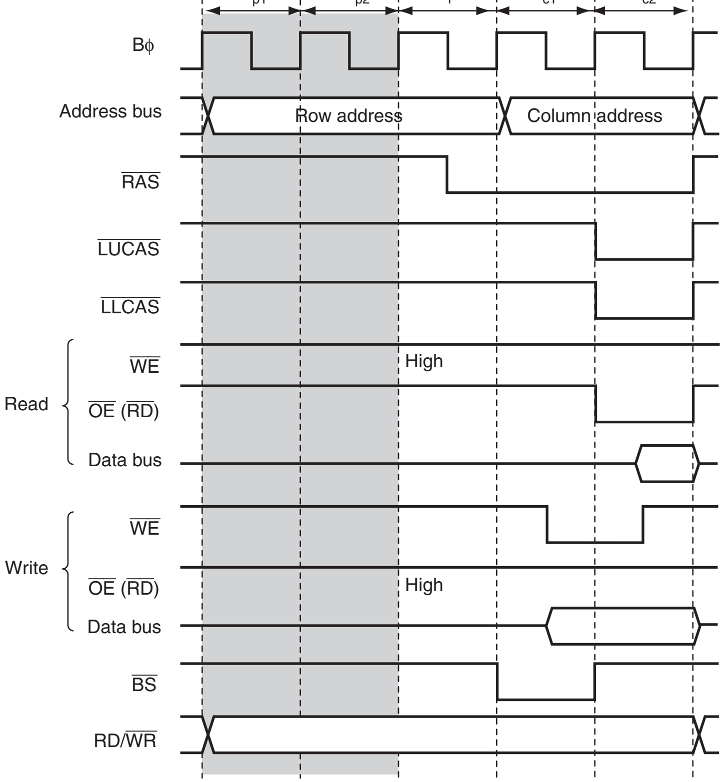


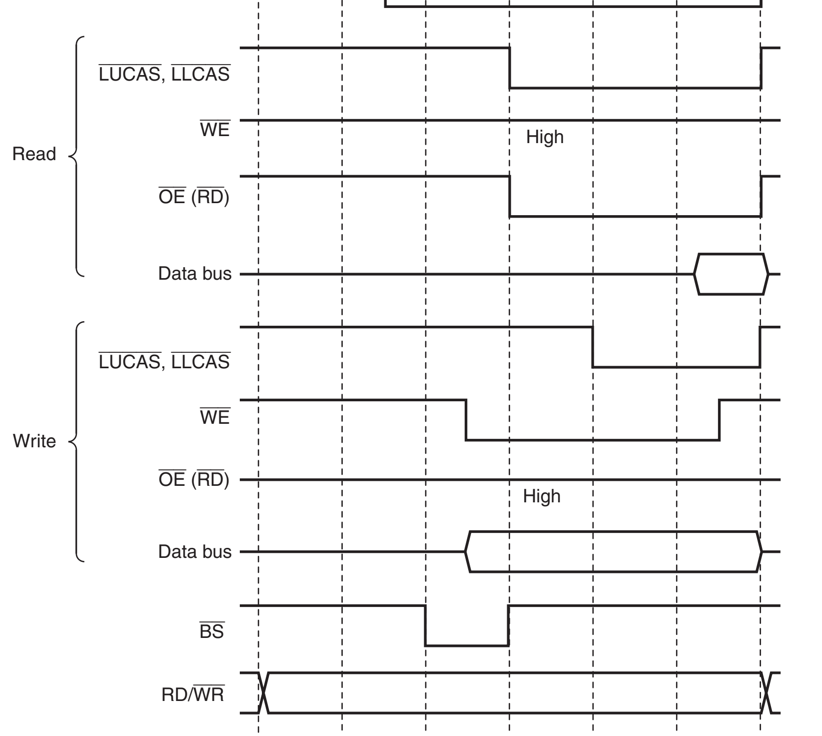
Figure 6.42 Access Timing Example of Two Precharge Cycles (RAST = 0 and CA

between the Tc1 and Tc2 cycles. The number of wait cycles is selected by bits W22 to W
WTCRB.

(2) Pin Wait Insertion

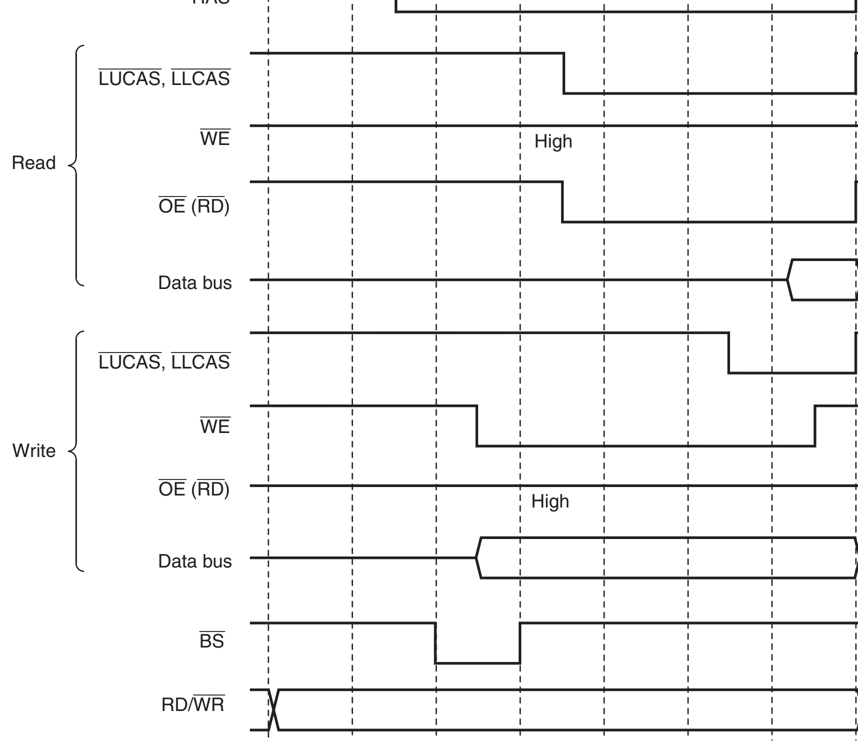
When the WAITE bit in BCR1 is set to 1, and the AST2 bit in ASTCR is set to 1, settin
bit for the corresponding pin to 1 enables wait input by the $\overline{\text{WAIT}}$ pin. When the DRAM
accessed in this state, a program wait (Tpw) is first inserted. If the $\overline{\text{WAIT}}$ pin is low at th
edge of Bϕ in the last Tc1 or Tpw cycle, another Ttw cycle is inserted until the $\overline{\text{WAIT}}$ pi
high. For details on ICR, see section 9, I/O Ports.

Figure 6.43 shows an example of wait cycle insertion timing for 2-cycle column address
Figure 6.44 shows an example of wait cycle insertion timing for 3-cycle column address



Note: Upward arrows indicate the timing of WAIT pin sampling.

Figure 6.43 Example of Wait Cycle Insertion Timing for 2-Cycle Column Address



Note: Upward arrows indicate the timing of WAIT pin sampling.

Figure 6.44 Example of Wait Cycle Insertion Timing for 3-Cycle Column Addres

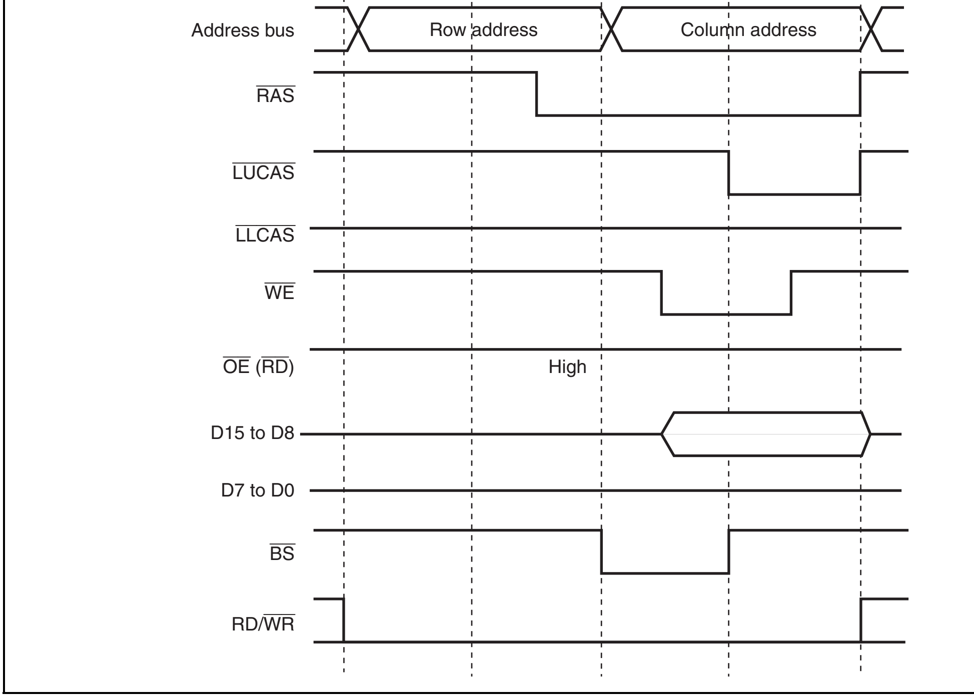


Figure 6.45 Timing Example of Byte Control with Use of Two CAS Signals (Write Access with Lowest Bit of Address = B'0, RAST = 0, CAST = 0)

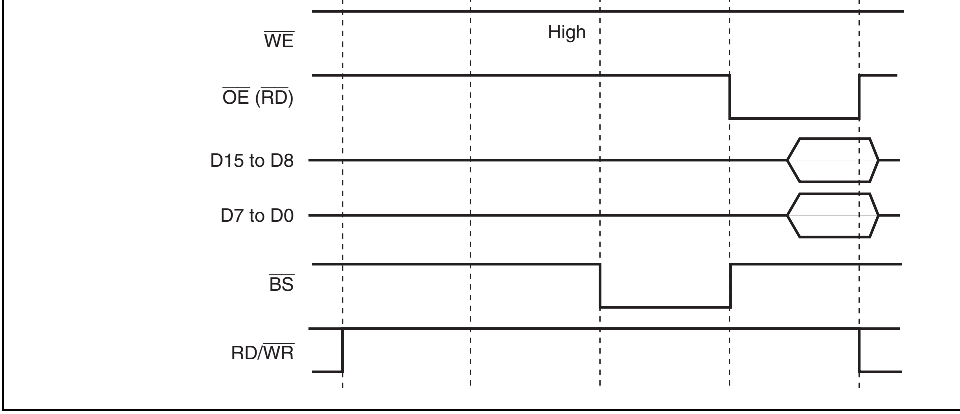


Figure 6.46 Timing Example of Word Control with Use of Two CAS Signa
(Read Access with Lowest Bit of Address = B'0, RAST = 0, CAST = 0)

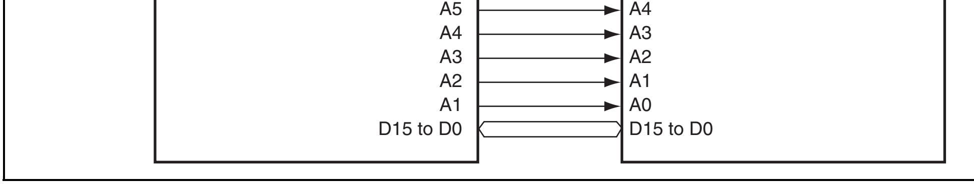


Figure 6.47 Example of Connection for Control with Two CAS Signals

6.10.11 Burst Access Operation

Besides an accessing method in which this LSI outputs a row address every time it access DRAM (called full access or normal access), some DRAMs have a fast-page mode functi which fast speed access can be achieved by modifying only a column address with the sa address output (burst access) when consecutive accesses are made to the same row addres

(1) Burst Access (Fast-Page Mode) Operation Timing

Figures 6.48 and 6.49 show operation timing of the fast-page mode.

When access cycles to the DRAM space are continued and the row addresses of the conse two cycles are the same, output cycles of the CAS and column address signals follow. Th address bits to be compared are decided by bits MXC1 and MXC0 in DRAMCR.

Wait cycles can be inserted during a burst access. The method and timing of the wait inse the same as that of full access mode. For details, see section 6.10.9, Wait Control.

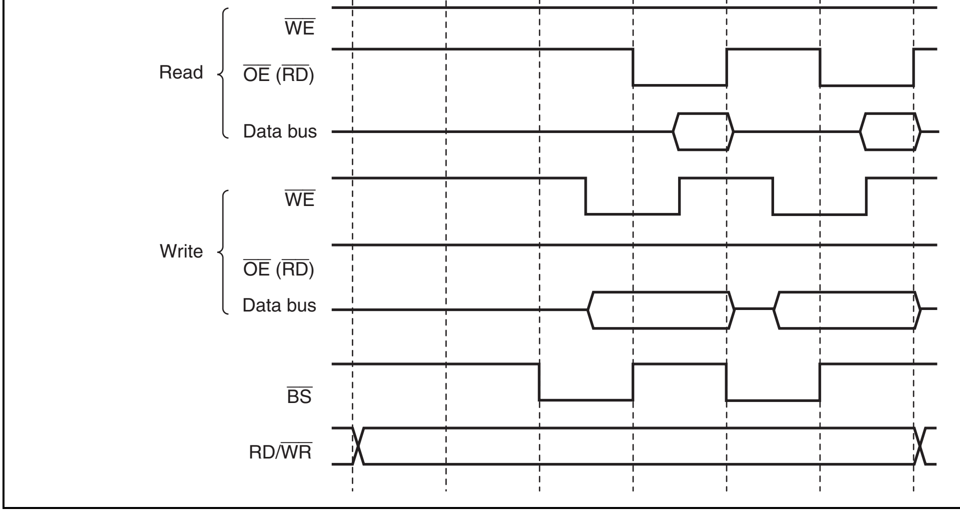


Figure 6.48 Operation Timing of Fast-Page Mode (RAST = 0, CAST = 0

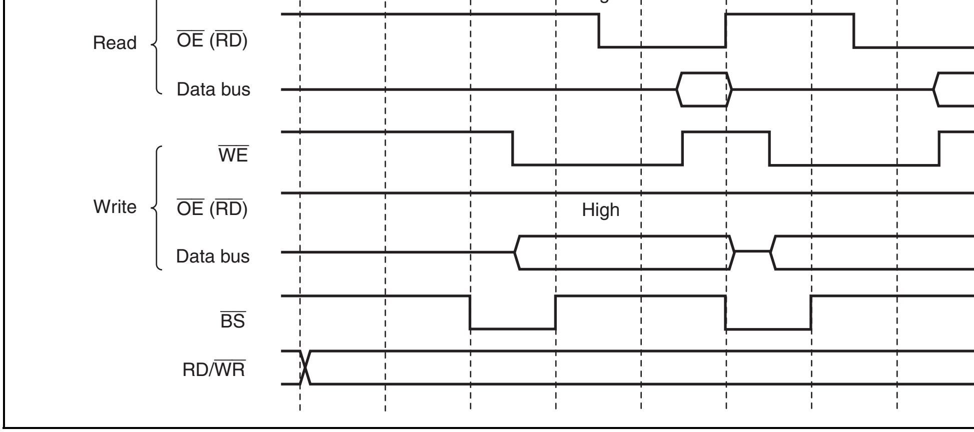


Figure 6.49 Operation Timing of Fast-Page Mode (RAST = 0, CAST = 1)

The fast-page mode access (burst access) is resumed when the row addresses of the curr and previous cycle are the same. While other spaces are accessed when the DRAM spac halted, the $\overline{RAS}$ signal must be low. Figure 6.50 shows a timing example of RAS down

The $\overline{RAS}$ signal goes high under the following conditions.

- When a refresh cycle is performed during RAS down mode
- When a self-refresh is performed
- When a transition to software standby mode is made
- When the external bus requested by the BREQ signal is released
- When either the RCDM or BE bit is cleared to 0

If a transition to the all-module clock-stop mode is made during RAS down mode, clock stopped with the $\overline{RAS}$ signal driven low. To make a transition with the $\overline{RAS}$ signal drive clear the RCDM bit to 0 before execution of the SLEEP instruction.

Clear the RCDM bit to 0 for write access to SCKCR to set the clock frequencies. For SC section 22, Clock Pulse Generator.

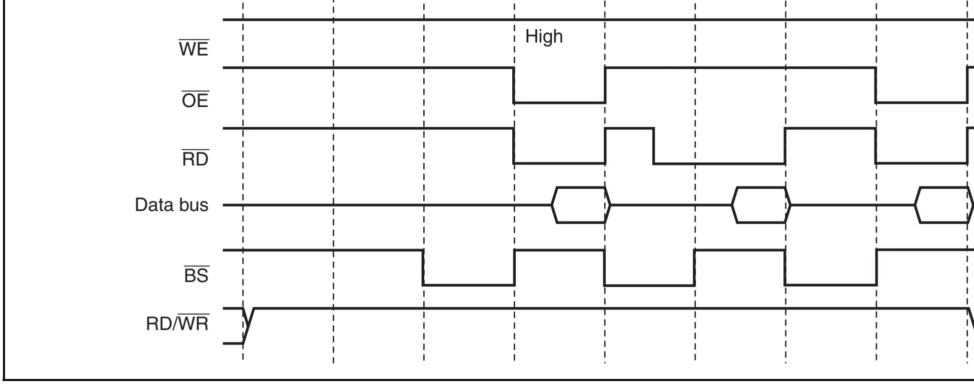


Figure 6.50 Timing Example of RAS Down Mode (RAST = 0, CAST = 0)

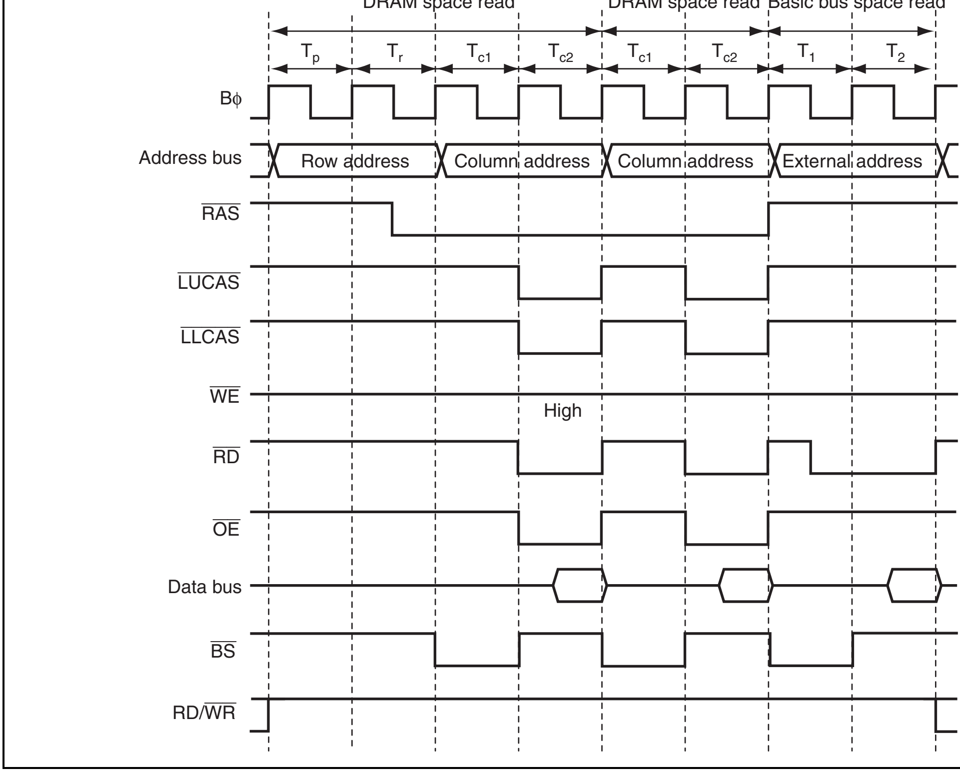


Figure 6.51 Timing Example of RAS Up Mode (RAST = 0, CAST = 0)

A CBR refresh cycle is performed when the value set in RTCOR matches the RTCNT va (compare match). RTCNT is an up-counter operated on the input clock specified by bits I to RTCK0 in REFCR. RTCNT is initialized upon the compare match and restarts to coun H'00. Accordingly, a CBR refresh cycle is repeated at intervals specified by bits RTCK2 RTCK0 in RTCOR. Set the bits so that the required refresh intervals of the DRAM must satisfied.

Since setting bits RTCK2 to RTCK0 starts RTCNT to count up, set RTCNT and RTCOR setting bits RTCK2 to RTCK0. When changing RTCNT and RTCOR, the counting opera should be halted. When changing bits RTCK2 to RTCK0, change them only after disablin external bus release, and if the write data buffer function is in use, disabling the write dat function and reading the external space.

The external space cannot be accessed in CBR refresh mode.

Figure 6.52 shows RTCNT operation, figure 6.53 shows compare match timing, and figu shows CBR refresh timing. Table 6.23 lists the pin states during a CBR refresh cycle.

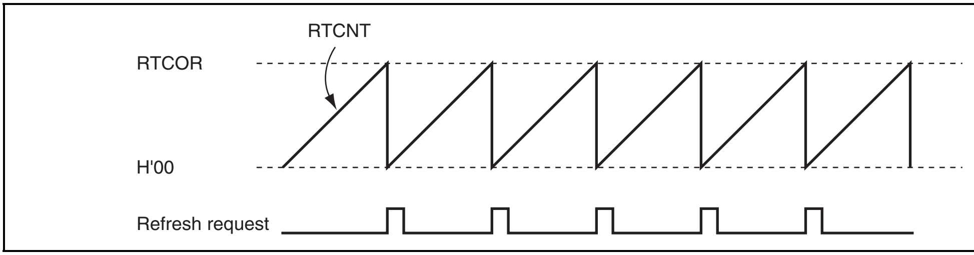


Figure 6.52 RTCNT Operation

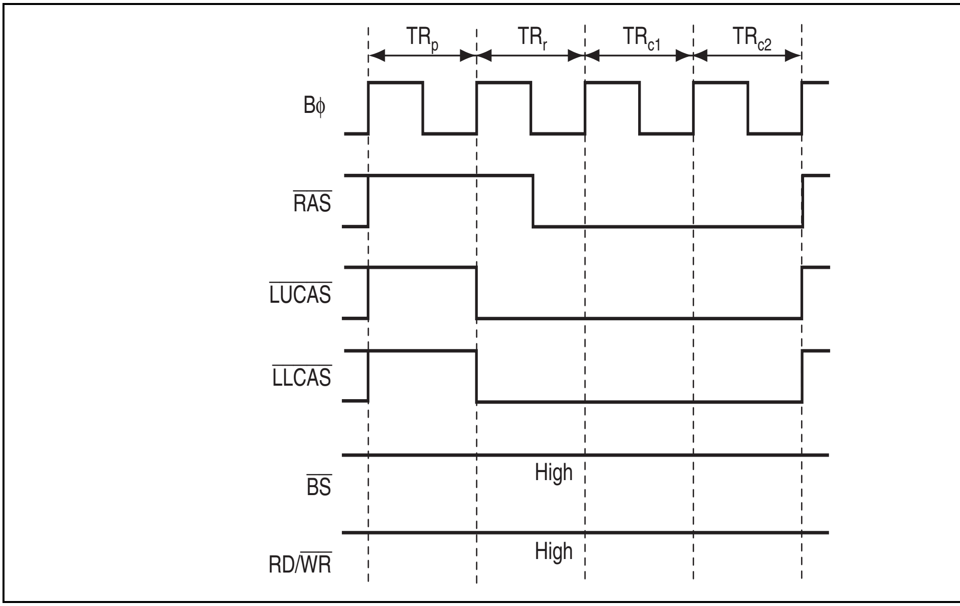


Figure 6.54 CBR Refresh Timing

$\overline{\text{BS}}$	High
RD/$\overline{\text{WR}}$	High

The $\overline{\text{RAS}}$ signal can be delayed for one to three clock cycles by setting bits RCW1 and R
REFCR. The pulse width of the $\overline{\text{RAS}}$ signal is changed by bits RLW2 to RLW0 in REFC
settings of bits RCW1, RCW0, and RLW2 to RLW0 are effective only for a refresh cycle
precharge time set by bit TPC1 and TPC0 is effective for a refresh cycle.

Figure 5.55 shows a timing for setting bits RCW1 and RCW0

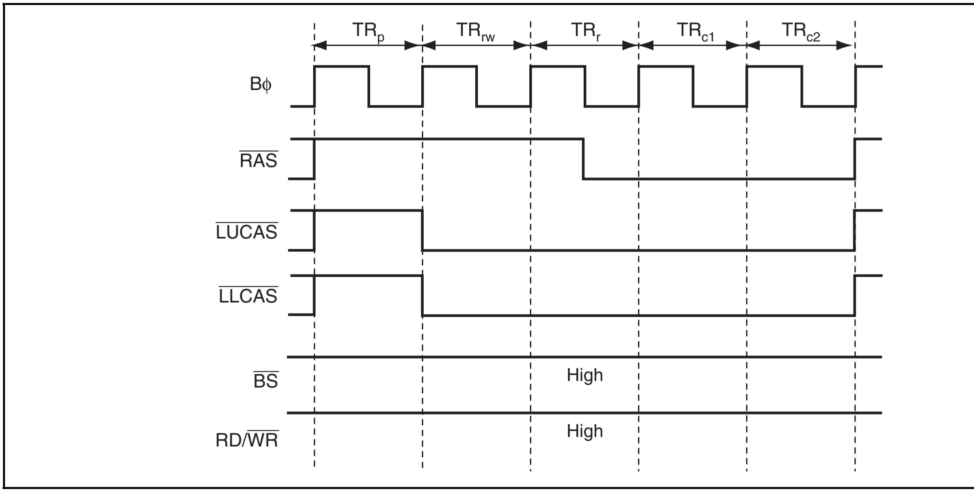


Figure 6.55 CBR Refresh Timing
(RCW1 = 0, RCW0 = 1, RLW2 = 0, RLW1 = 0, RLW0 = 0)

When the self-refresh mode is used, do not clear the OPE bit in SBYCR to 0.

For details, see section 23.2.1, Standby Control Register (SBYCR).

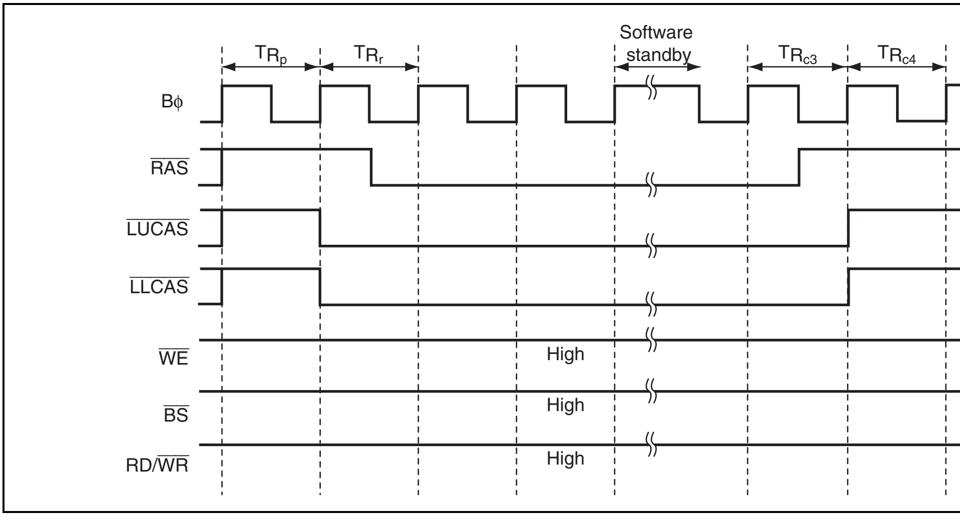


Figure 6.56 Self-Refresh Timing

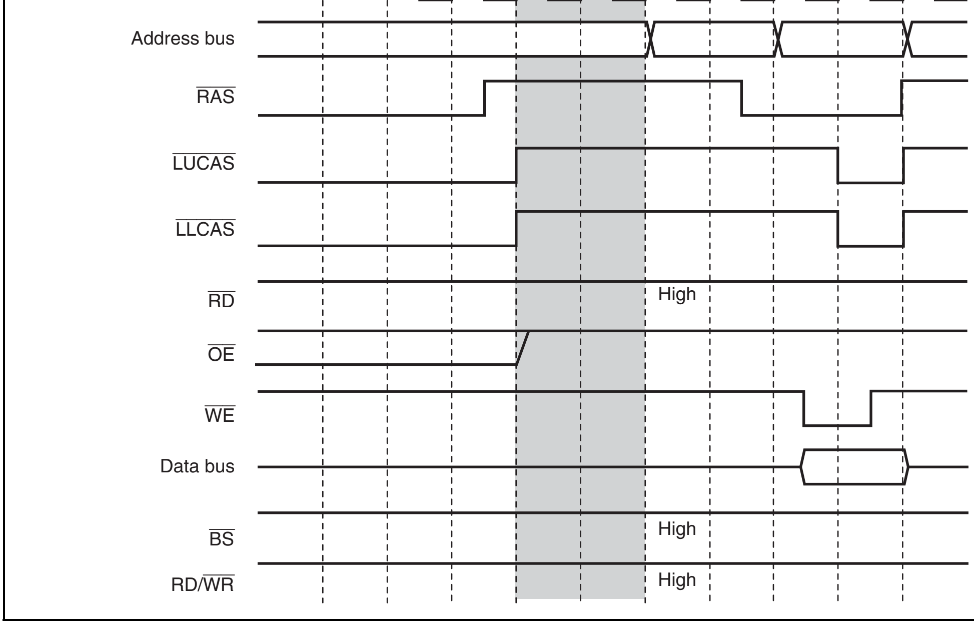


Figure 6.57 Timing Example when 1 Precharge Cycle Added

For details, see section 23.2.2, Module Stop Control Registers A and B (MSTPCR and MSTPCRB).

6.10.13 DRAM Interface and Single Address Transfer by DMAC

When fast-page mode (BE = 1) is set for the DRAM space, either fast-page access or ful can be selected, by the setting of bit DDS in DRAMCR, for the single address transfer b DMAC where the DRAM space is specified as the transfer source or destination. At the time, the output timing of the $\overline{\text{DACK}}$ and $\overline{\text{BS}}$ signals is changed. When BE = 0, full acce DRAM space is performed by single address transfer regardless of the setting of bit DD However, the output timing of the $\overline{\text{DACK}}$ and $\overline{\text{BS}}$ signals can be changed by the setting DDS.

The assertion timing of the $\overline{\text{DACK}}$ signal can be changed by bit DKC in BCR1.

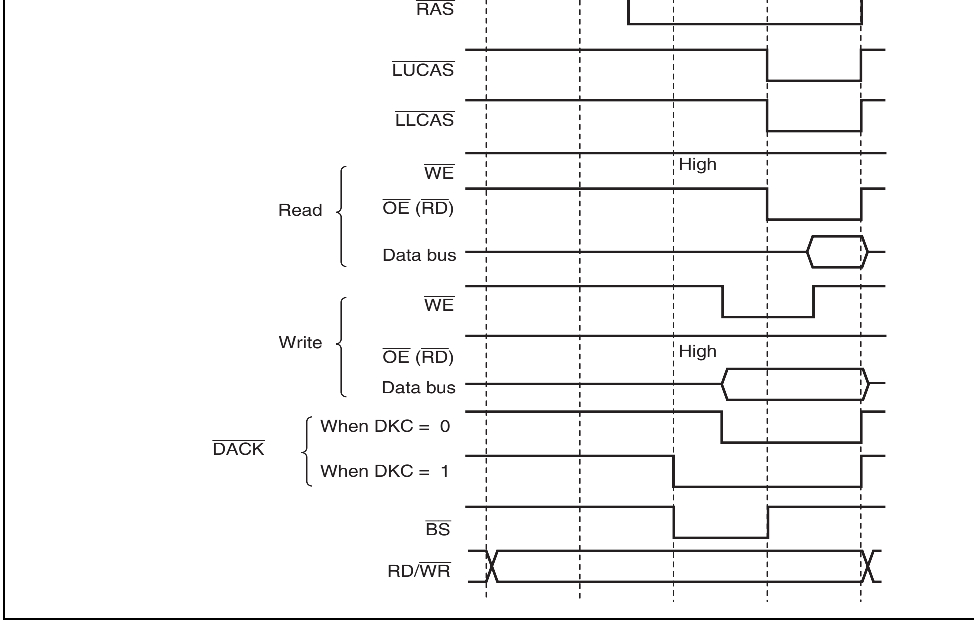


Figure 6.58 Output Timing Example of $\overline{\text{DACK}}$ when DDS = 1 (RAST = 0, CAST

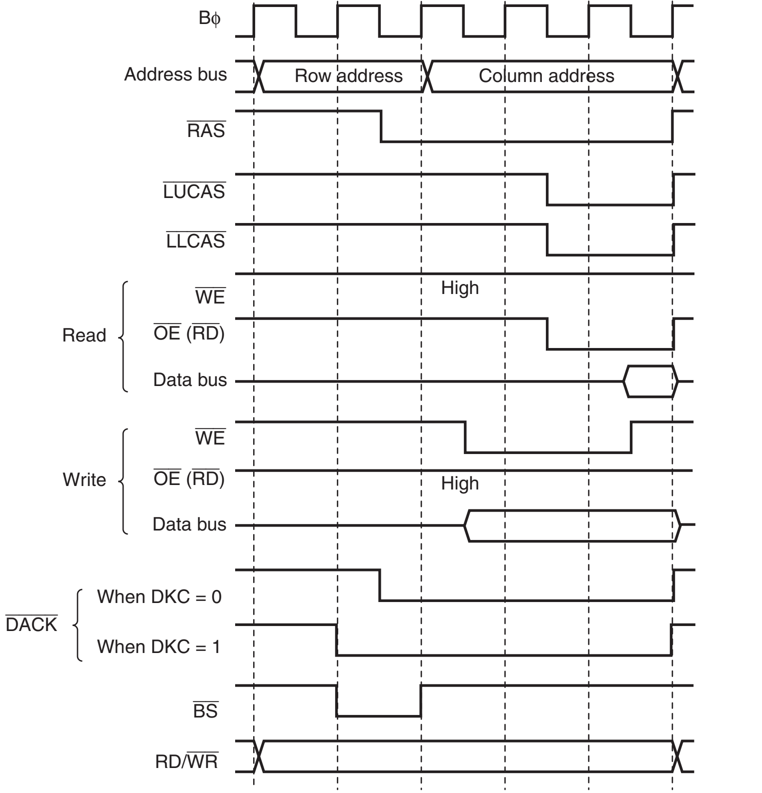


Figure 6.59 Output Timing Example of $\overline{\text{DACK}}$ when DDS = 0 (RAST = 0, CAS

In the SDRAM space, pins PB2, PB3, and PB4 are used as the $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and $\overline{\text{WE}}$ signa PB1 pin is used as the $\overline{\text{CS2}}$ signal by the PFCR setting, and the PB5 pin is used as the CK by setting the OEE bit in DRAMCR to 1. The bus settings of the SDRAM space depend c settings. The pin wait and program wait for the SDRAM space are not available. For PFC section 9, I/O Ports.

An SDRAM command is designated by the combination of the $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and $\overline{\text{WE}}$ signa the precharge-sel command (Precharge-sel) output on the upper column address.

This LSI supports the following commands: the NOP, auto-refresh (REF), self-refresh (S bank-precharge (PALL), bank active (ACTV), read (READ), write (WRIT), and mode re setting (MRS). Commands controlling a bank are not supported.

Table 6.24 Relationship among DRAME and DTYPE and Area 2 Interfaces

DRAME	DTYPE	Area 2 Interface
0	X	Basic bus space (initial state)/byte-control SRAM space
1	0	DRAM space
1	1	SDRAM space

[Legend]
X: Don't care

				Column address	A23 to A18	-	-	A23	A22	A21	A20	A19	P	A9	A8	A7	A6	A5	A4	A3
			16 bits	Row address	A23 to A18	-	-	A23	A22	A21	A20	P/A19*	A18	A17	A16	A15	A14	A13	A12	A11
				Column address	A23 to A18	-	-	A23	A22	A21	A20	P	A10	A9	A8	A7	A6	A5	A4	A3
0	1	9 bits	8 bits	Row address	A23 to A18	A17	-	-	A23	A22	A21	A20	P/A19*	A18	A17	A16	A15	A14	A13	A12
				Column address	A23 to A18	A17	-	-	A23	A22	A21	A20	P	A9	A8	A7	A6	A5	A4	A3
			16 bits	Row address	A23 to A18	A17	-	-	A23	A22	A21	P/A20*	A19	A18	A17	A16	A15	A14	A13	A12
				Column address	A23 to A18	A17	-	-	A23	A22	A21	P	A10	A9	A8	A7	A6	A5	A4	A3
1	0	10 bits	8 bits	Row address	A23 to A18	-	-	-	-	A23	A22	A21	P/A20*	A19	A18	A17	A16	A15	A14	A13
				Column address	A23 to A18	-	-	-	-	A23	A22	A21	P	A9	A8	A7	A6	A5	A4	A3
			16 bits	Row address	A23 to A18	-	-	-	-	A23	A22	P/A21*	A20	A19	A18	A17	A16	A15	A14	A13
				Column address	A23 to A18	-	-	-	-	A23	A22	P	A10	A9	A8	A7	A6	A5	A4	A3
1	1	11 bits	8 bits	Row address	A23 to A18	A17	-	-	-	-	A23	A22	P/A21*	A20	A19	A18	A17	A16	A15	A14
				Column address	A23 to A18	A17	-	-	-	-	A23	A10	P	A9	A8	A7	A6	A5	A4	A3
			16 bits	Row address	A23 to A18	A17	-	-	-	-	A23	P/A22*	A21	A20	A19	A18	A17	A16	A15	A14
				Column address	A23 to A18	A17	-	-	-	-	A11	P	A10	A9	A8	A7	A6	A5	A4	A3

Note: * When issuing the PALL command, precharge-sel = 1 is output and when issuing the ACTIV command, a corresponding address is output.

6.11.3 Data Bus

Either 8 or 16 bits can be selected as the data bus width of the SDRAM space by bits AI ABWL2 in ABWCR. SDRAM with 16-bit words can be connected directly to 16-bit bu space.

D7 to D0 are valid in 8-bit SDRAM space and D15 to D0 are valid in 16-bit SDRAM sp

The data endian format can be selected by bit LE2 in ENDIANCR. For details on the ac and alignment, see section 6.5.6, Endian and Data Alignment.

Pin	DRAM Selected	Name	I/O	Function
$\overline{\text{RAS}}$	$\overline{\text{RAS}}$	Row address strobe	Output	Row address strobe when the S space is specified as area 2
$\overline{\text{CAS}}$	$\overline{\text{CAS}}$	Column address strobe	Output	Column address strobe when t SDRAM space is specified as a
$\overline{\text{WE}}$	$\overline{\text{WE}}$	Write enable	Output	Write enable signal for accessi SDRAM interface
$\overline{\text{OE}}$/CKE	CKE	Clock enable	Output	Clock enable signal when the S space is specified as area 2.
$\overline{\text{LLCAS}}$/ DQMLU	DQMLU	Lower-upper data mask enable	Output	Upper data mask enable when bit SDRAM space is accessed
$\overline{\text{LLCAS}}$/ DQMLL	DQMLL	Lower-lower data mask enable	Output	• Lower data mask enable w 16-bit SDRAM space is acc • Data mask enable when th SDRAM is accessed
A17 to A0	A17 to A0	Address pin	Output	Multiplexed row/column-addres pin
D15 to D0	D15 to D0	Data pin	Input/ output	Data input/output pin
(PA7) PB7	SDϕ	Clock	Output	SDRAM clock
$\overline{\text{CS2}}$	$\overline{\text{CS}}$	Chip select	Output	Strobe signal indicating that SD selected

DRAMCR, bits RCW1 and RCW0 in REFCR are ignored.

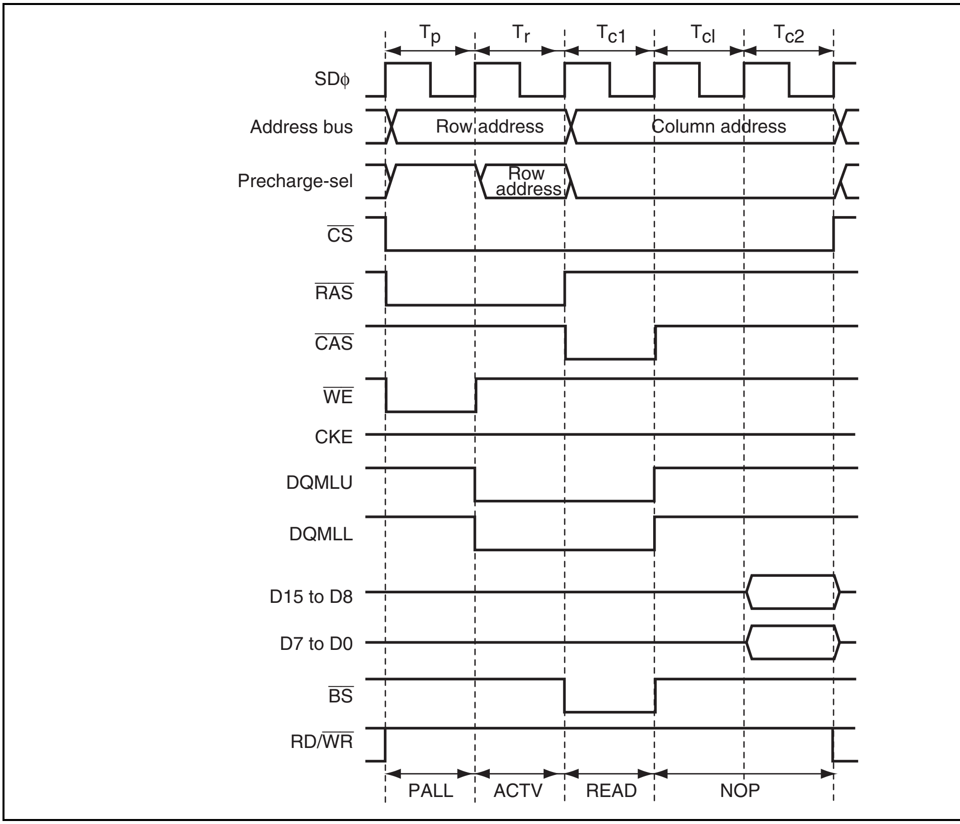


Figure 6.60 SDRAM Basic Read Access Timing (CAS Latency = 2)

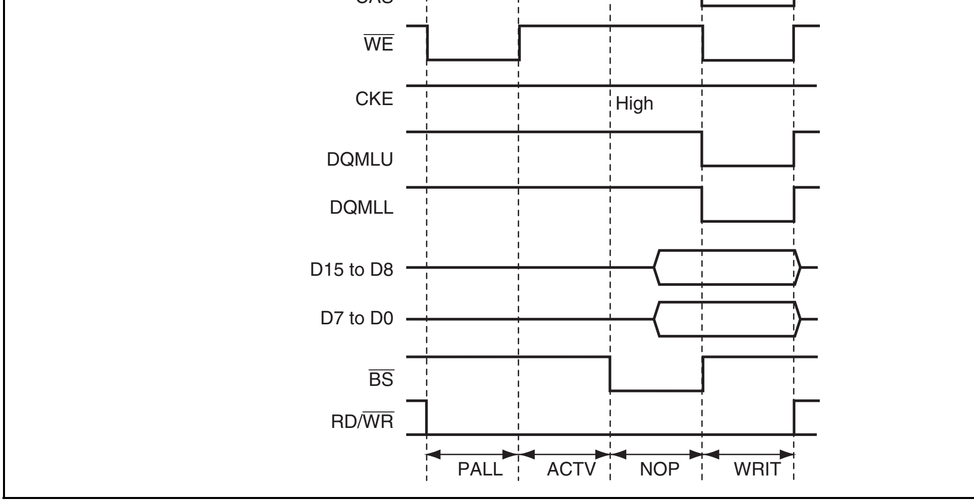


Figure 6.61 SDRAM Basic Write Access Timing

W21	W20	Description	Number of Latency (
0	0	Setting prohibited	—
	1	SDRAM with CAS latency of 2 is in use	1
1	0	SDRAM with CAS latency of 3 is in use	2
	1	SDRAM with CAS latency of 4 is in use	3

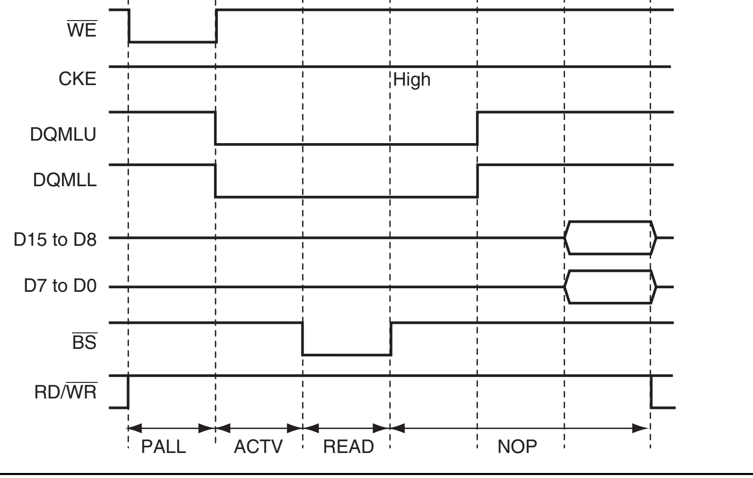


Figure 6.62 Timing Example of CAS Latency (CAS Latency = 3)

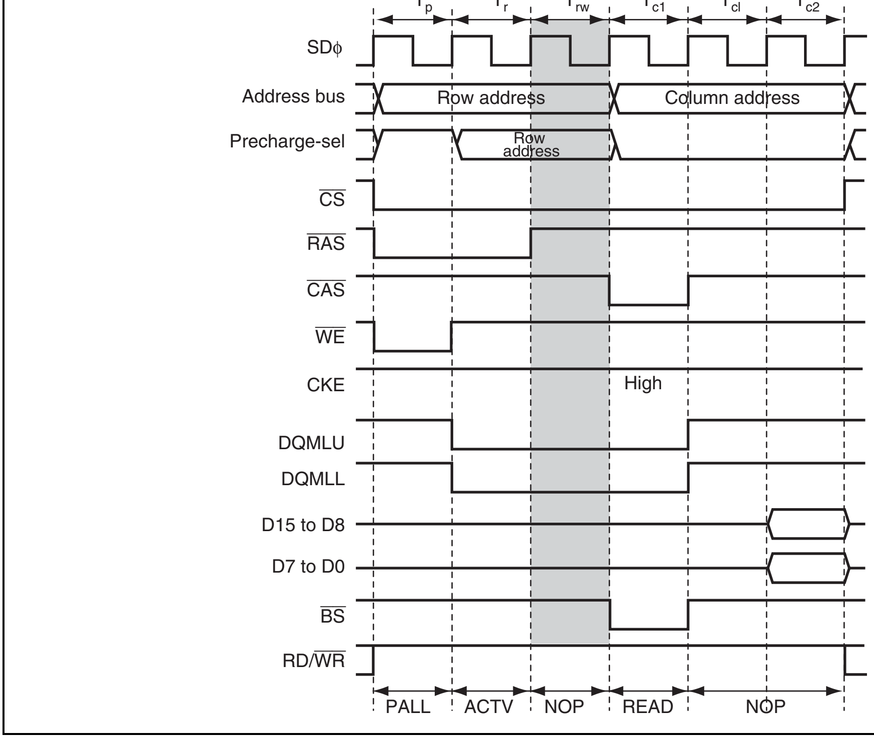


Figure 6.63 Read Timing Example of Row Address Output Retained for 1 Cloc
(RCD1 = 0, RCD0 = 1, CAS Latency = 2)

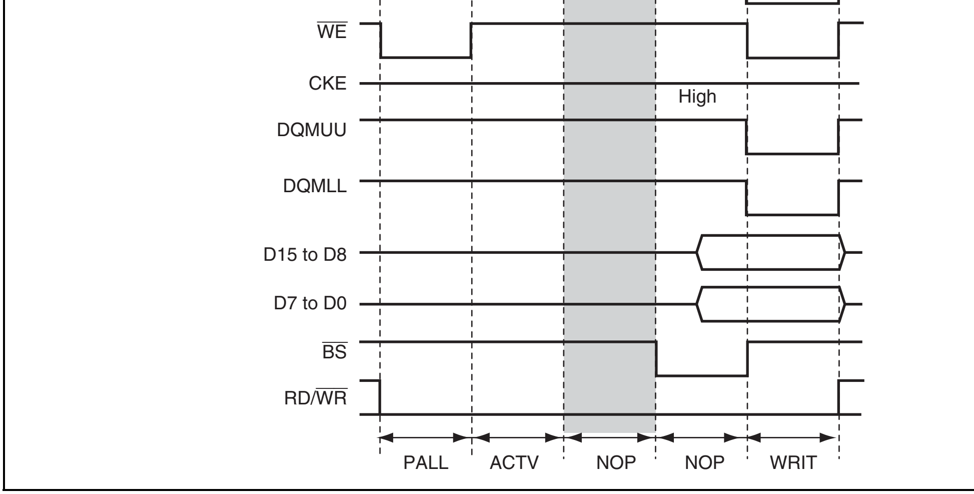


Figure 6.64 Write Timing Example of Row Address Output Retained for 1 Clock (RCD1 = 0, RCD0 = 1)

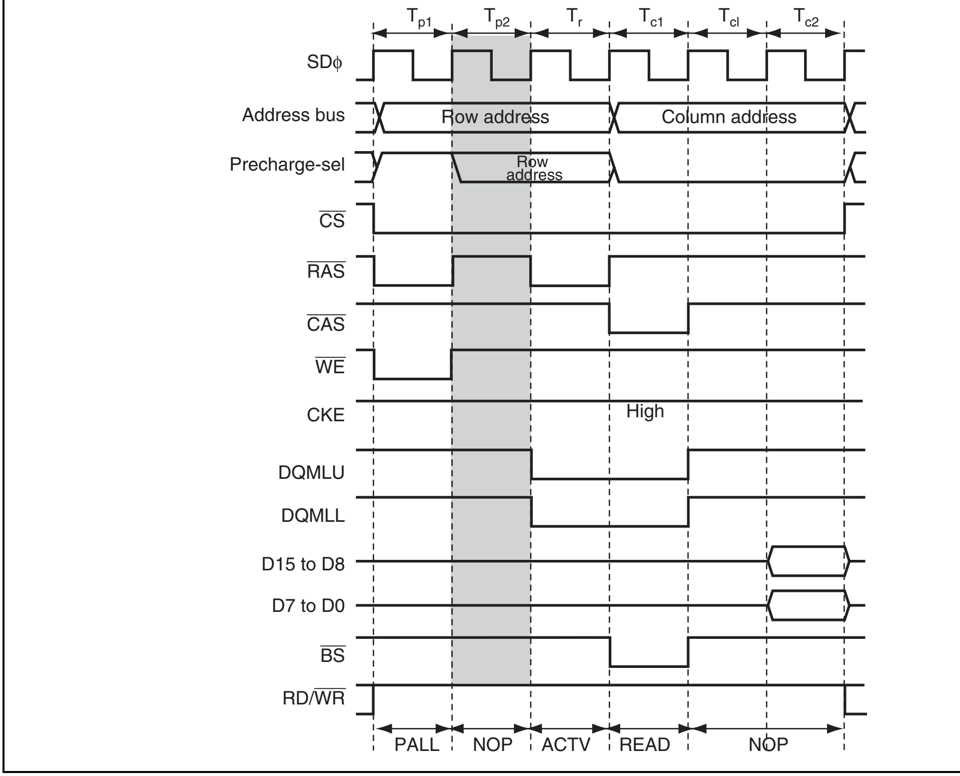


Figure 6.65 Read Timing Example of Two Precharge Cycles (TPC1 = 0, TPC0 = 1, CAS Latency = 2)

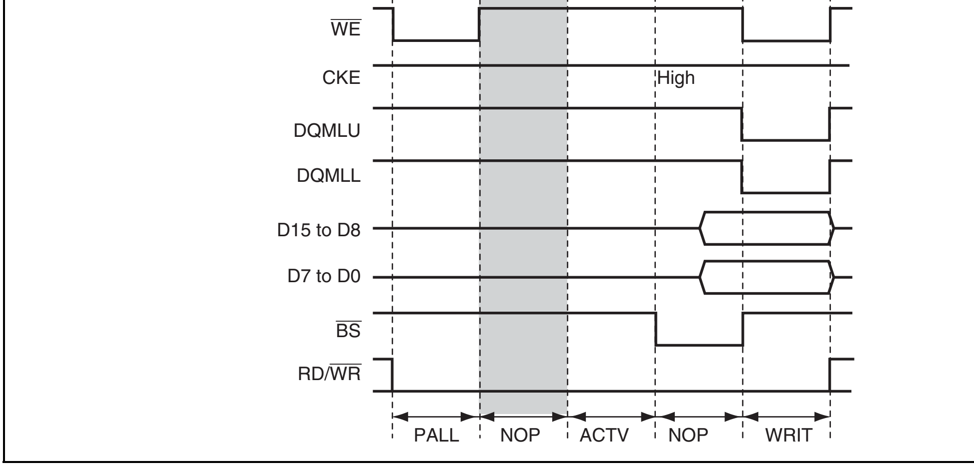


Figure 6.66 Write Timing Example of Two Precharge Cycles (TPC1 = 0, TPC0

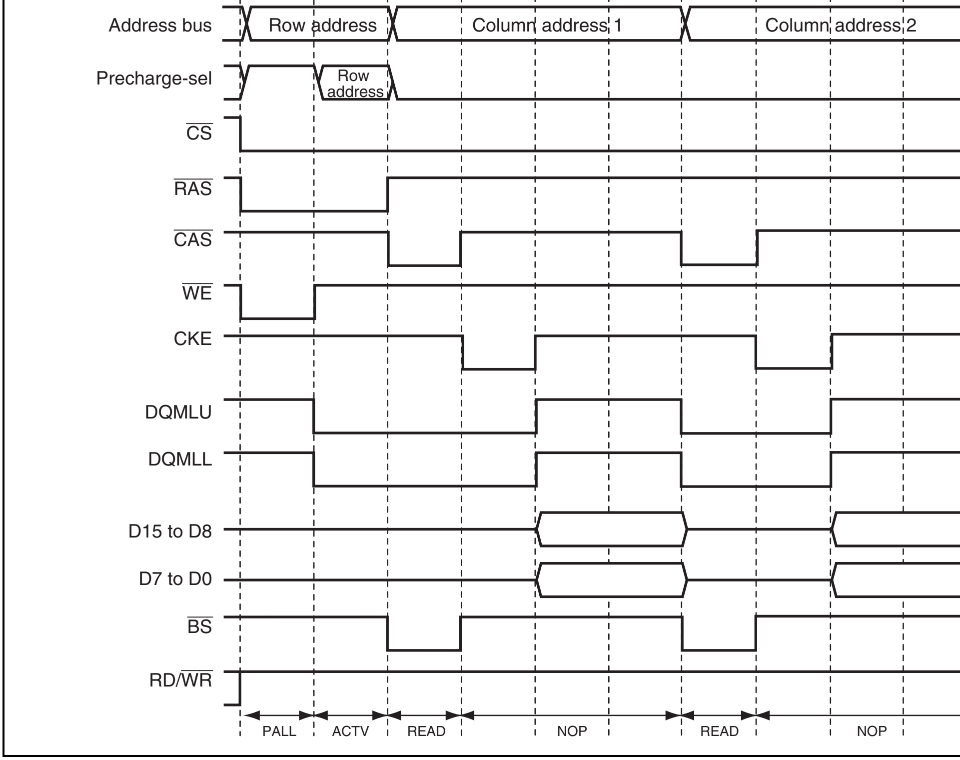


Figure 6.67 Read Timing Example when CKSPE = 1 (CAS Latency = 2)

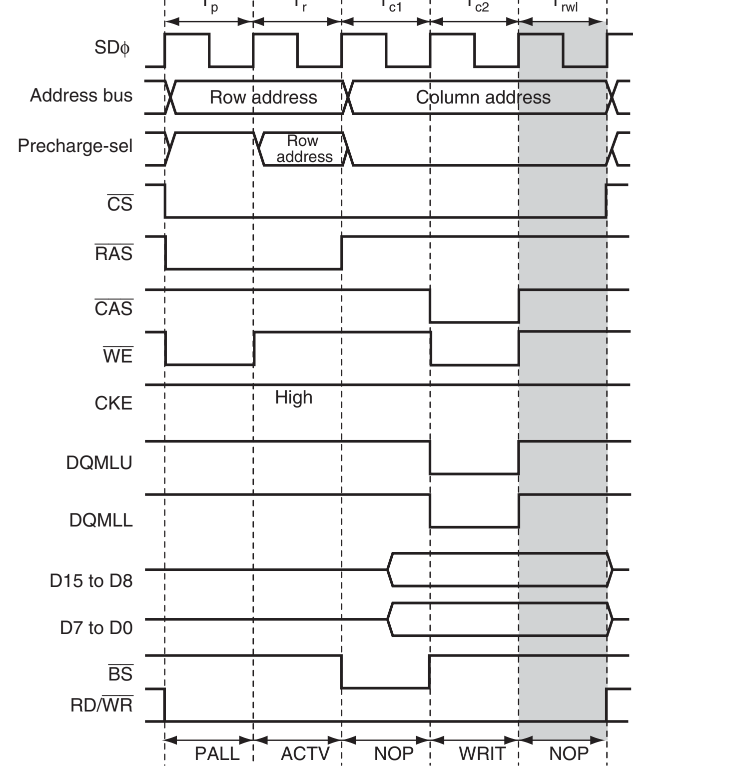


Figure 6.68 Write Timing Example when Write-Precharge Delay Cycle Insert
(TRWL = 1)

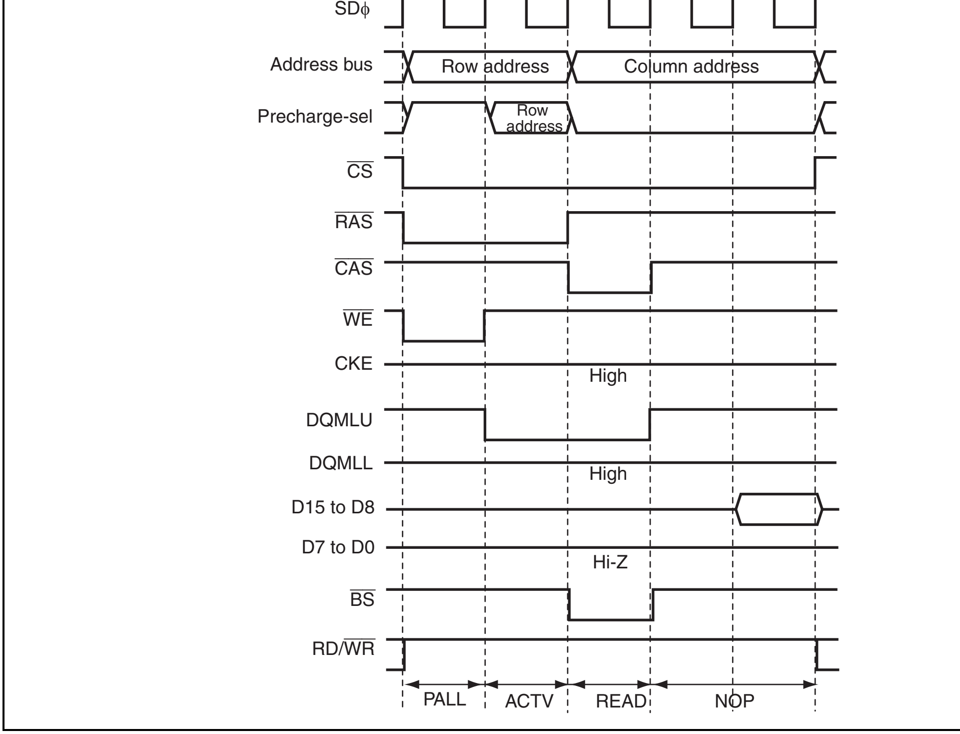


Figure 6.69 Control Timing Example of Byte Control by DQM in 16-Bit Acces
(Read Access with Lowest Bit of Address = B'0)

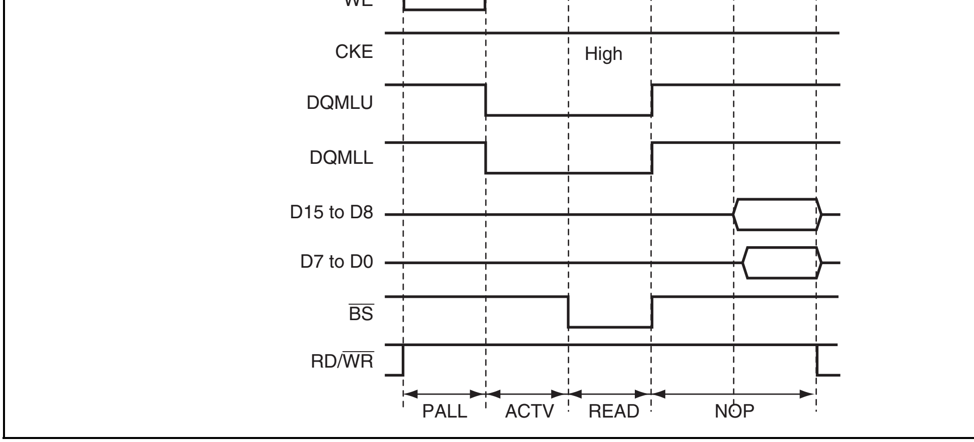


Figure 6.70 Control Timing Example of Word Control by DQM in 16-Bit Access (Read Access with Lowest Bit of Address = B'0, CAS Latency = 2)

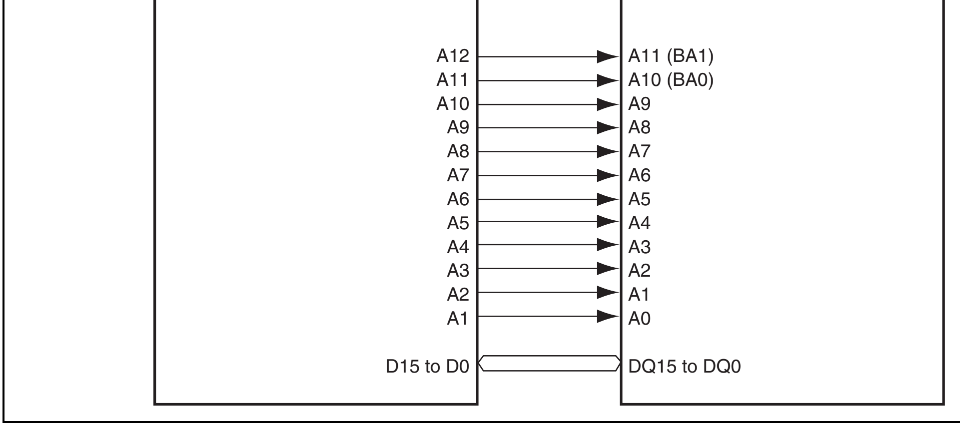


Figure 6.71 Connection Example of DQM Byte/Word Control

6.11.12 Fast-Page Access Operation

Besides an accessing method in which this LSI outputs a row address every time it acces SDRAM (called full access or normal access), some SDRAMs have a fast-page mode fu which fast speed access can be achieved by modifying only a column address with the s address output when consecutive accesses are made to the same row address.

The fast-page mode can be used by setting the BE bit in DRAMCR to 1.

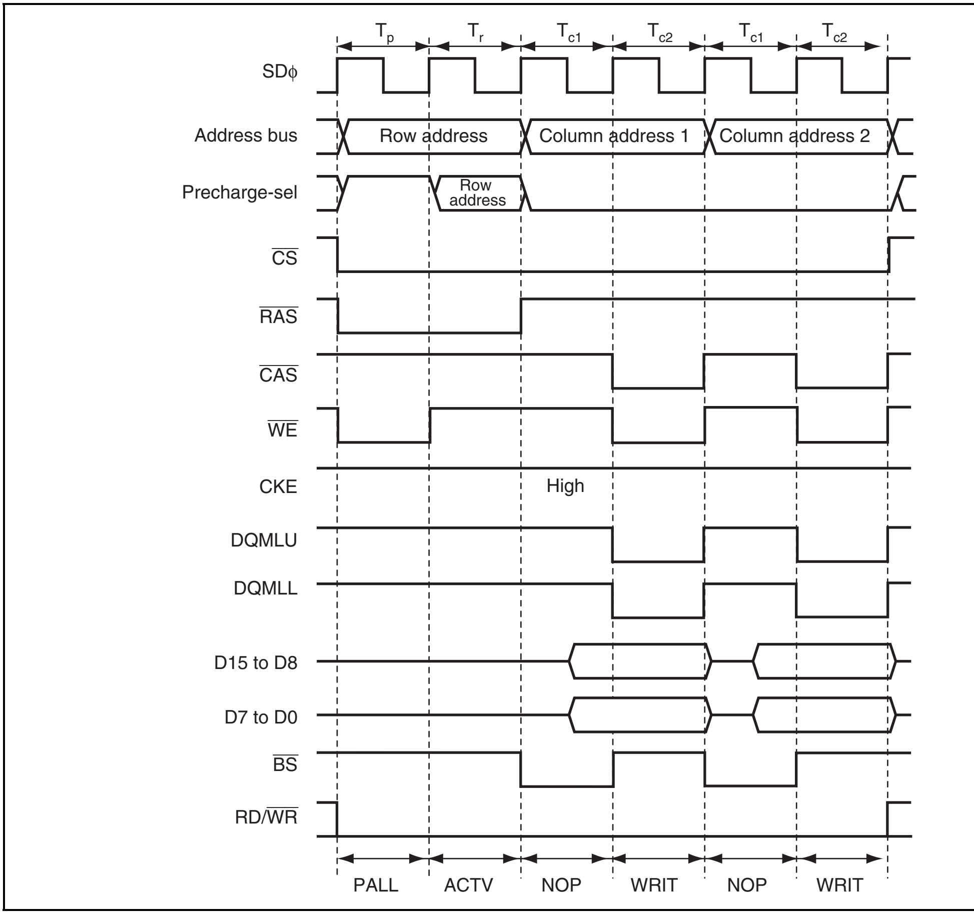


Figure 6.72 Longword Write Timing in 16-Bit Access Space (BE = 1, RCDM =

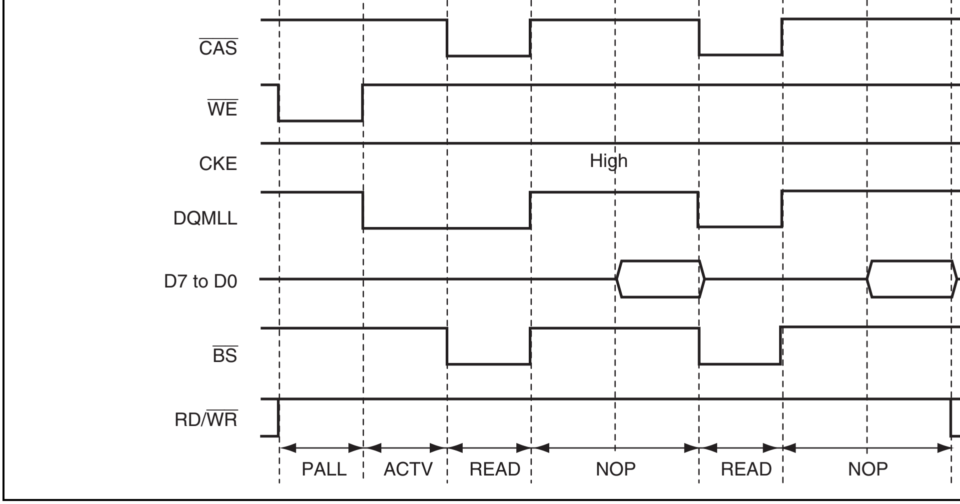


Figure 6.73 Word Read Timing in 8-Bit Access Space (BE = 1, RCDM = 0, CAS Latency = 2)

The next cycle after one of the following conditions is satisfied is a full access cycle.

- When a refresh cycle is performed during RAS down mode
- When a self-refresh is performed
- When a transition to software standby mode is made
- When the external bus requested by the $\overline{\text{BREQ}}$ signal is released
- When either the RCDM or BE bit is cleared to 0
- When setting the SDRAM mode register

Some SDRAMs have a limitation on the time to hold each bank active. When such SDRA use, if the user program cannot control the time (such as software standby or sleep mode) the auto-refresh or self-refresh so that the given specification can be satisfied. If a refresh not used, the user program must control the time.

Clear the RCDM bit to 0 for write access to SCKCR to set the clock frequencies. For SCI section 22, Clock Pulse Generator.

(3) RAS Up Mode

Clear the RCDM bit in DRAMCR to 0 to set the RAS up mode.

Whenever a SDRAM space access is halted and other spaces are accessed, the next cycle PALL command cycle. Only when the SDRAM space continues to be accessed, the fast-p mode access is performed.

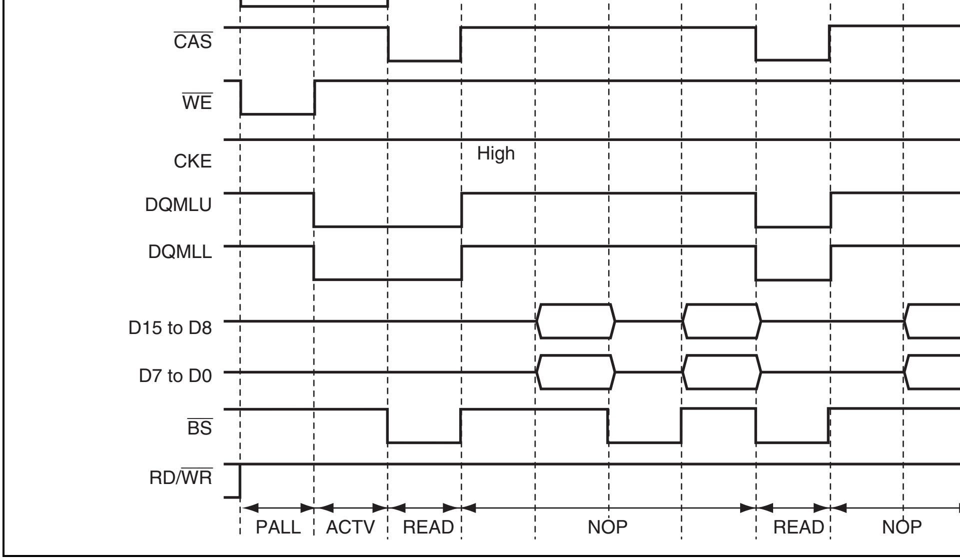


Figure 6.74 Timing Example of RAS Down Mode (BE = 1, RCDM = 1, CAS Lat

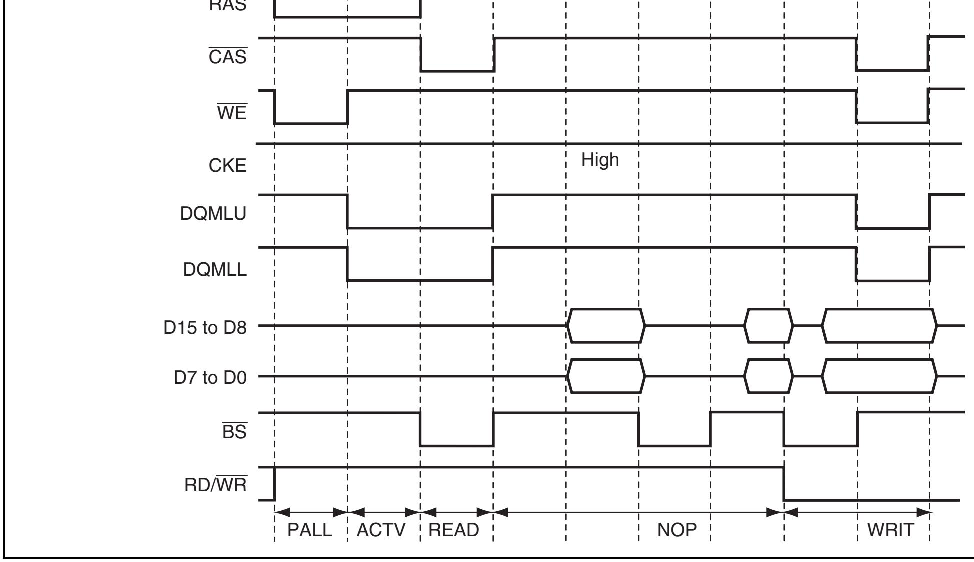


Figure 6.75 Timing Example of RAS Down Mode (BE = 1, RCDM = 1, CAS Later

RENESAS

An auto-refresh cycle is performed when the value set in RTCOR matches the RTCNT v
(compare match). RTCNT is an up-counter operated on the input clock specified bits RT
RTCK0 in REFCR. RTCNT is initialized upon the compare match and restarts to count
H'00. Accordingly, an auto-refresh cycle is repeated at intervals specified by bits RTCK
RTCK0 in RTCOR. Set the bits so that the required refresh intervals of the DRAM mus
satisfied.

Since setting bits RTCK2 to RTCK0 starts RTCNT to count up, set RTCNT and RTCO
setting bits RTCK2 to RTCK0. When changing RTCNT and RTCOR, the count operati
be halted. When changing bits RTCK2 to RTCK0, change them only after disabling ext
release and, if the write data buffer function is in use, disabling the write data buffer fun
reading the external space.

The external space cannot be accessed during auto-refresh.

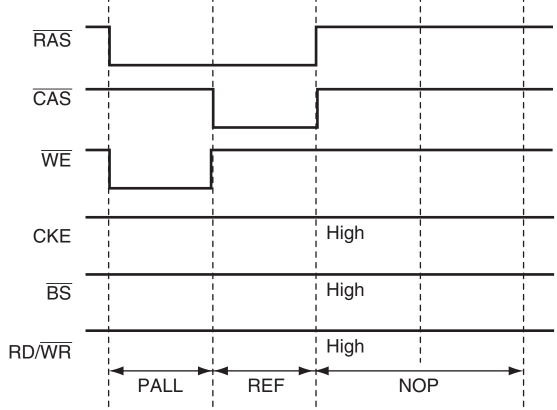


Figure 6.76 Auto-Refresh Operation

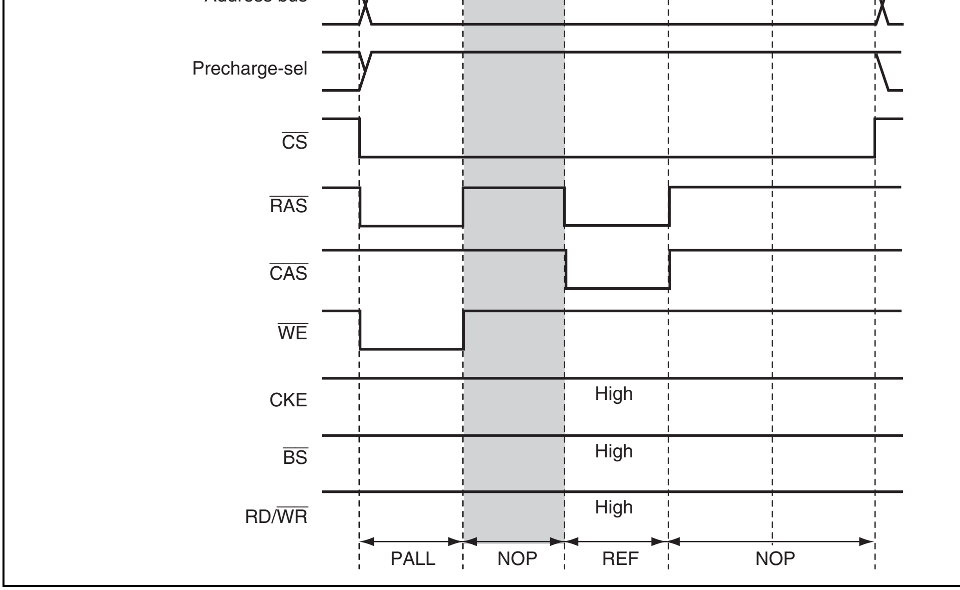


Figure 6.77 Auto-Refresh Timing (TPC1 = 0, TPC0 = 1)

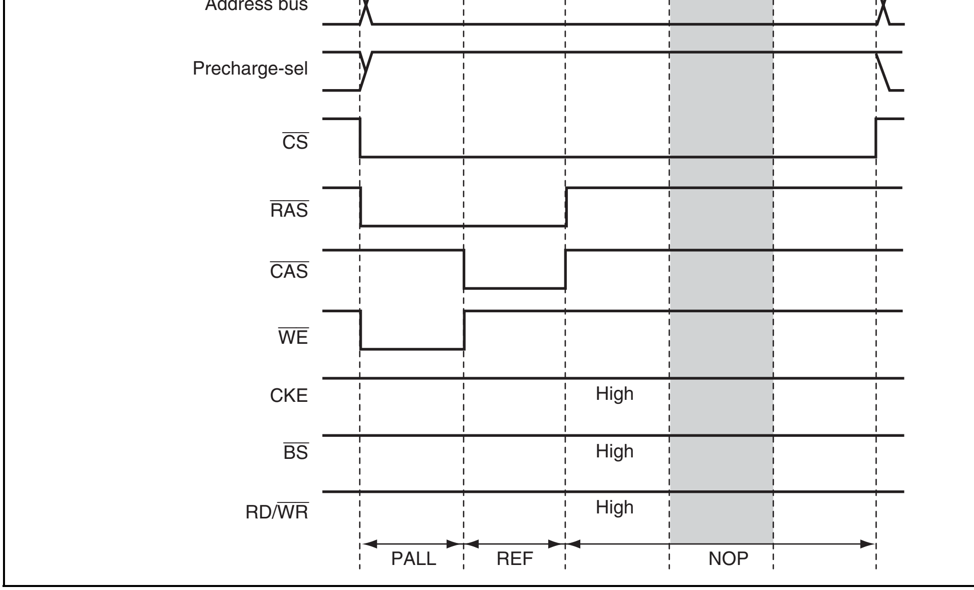


Figure 6.78 Auto-Refresh Timing (TPC1 = 0, TPC0 = 0, RLW2 = 0, RLW1 = 0, RL

When making a transition to the self-refresh mode, set the OEE bit in SBYCR to 1 and c
CKE pin.

When the self-refresh mode is used, do not clear the OPE bit in SBYCR to 0.

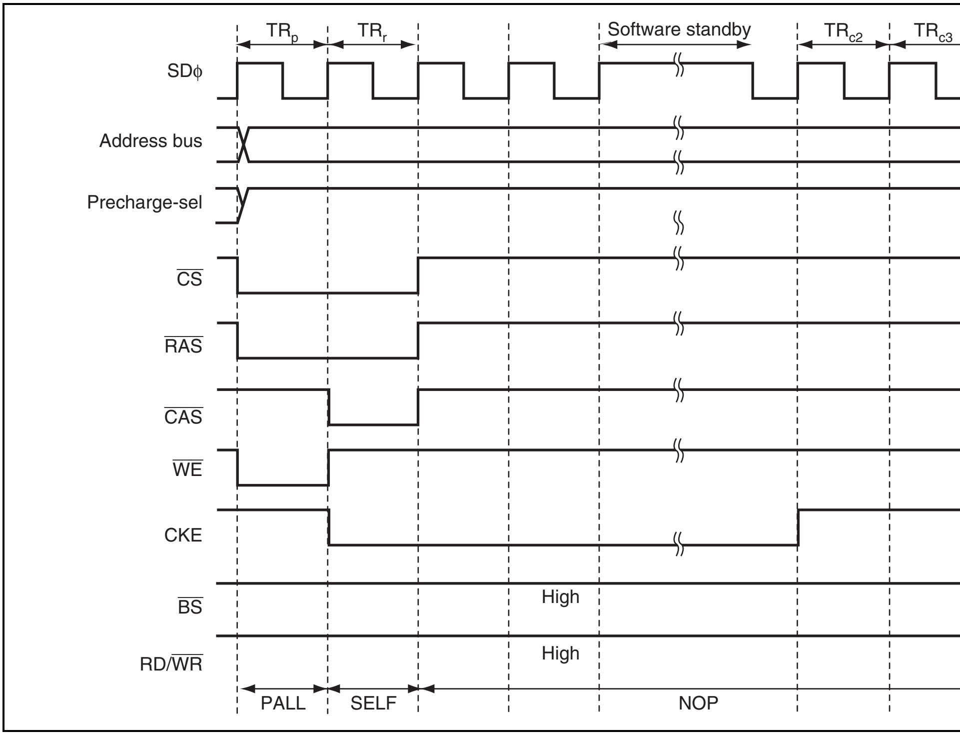


Figure 6.79 Self-Refresh Timing
(TPC1 = 0, TPC0 = 0, RCW1 = 0, RCW0 = 0, RLW1 = 0, RLW0 = 0)

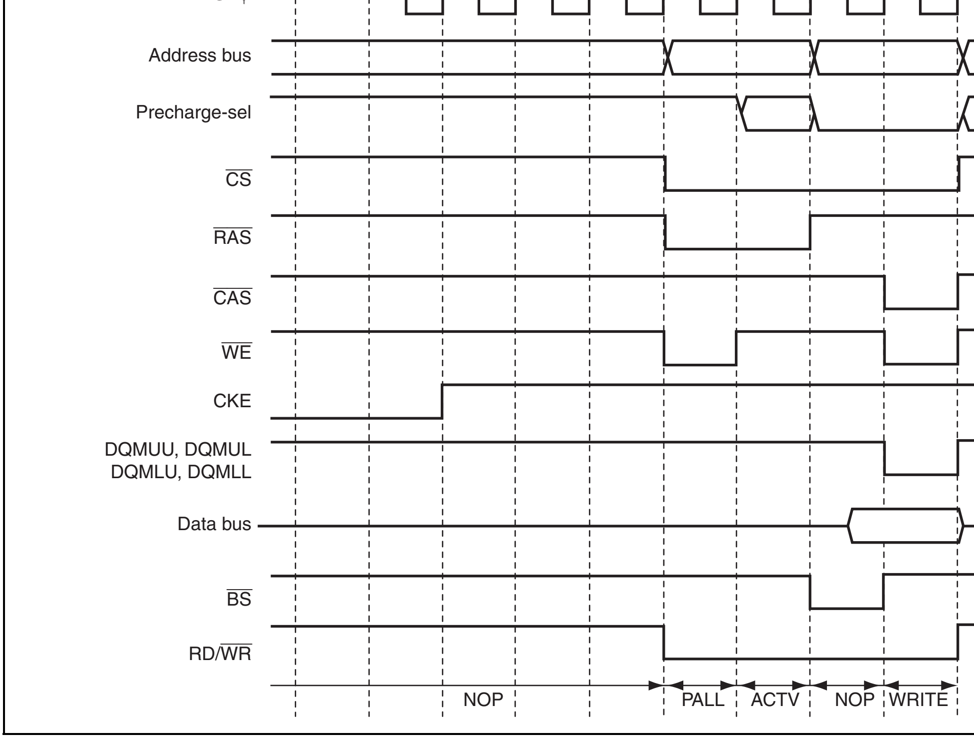


Figure 6.80 Timing Example when 1 Precharge Cycle Added (TPC2 to TPC0 = H'1, TPC1 = 0, TPC0 = 0)

For details, see section 23.2.2, Module Stop Control Registers A and B (MSTPCR and MSTPCRB).

6.11.14 Setting SDRAM Mode Register

To use SDRAM, the mode register must be specified after a power-on reset.

Setting the MRSE bit in SDCR to 1 enables the SDRAM mode register setting. After thi
the SDRAM space in bytes.

When the value to be set in the SDRAM mode register is x, write to the following memo
location (address). The value of x is written to the SDRAM mode register.

- H'4000000/H'400000 + x for 8-bit bus SDRAM
- H'4000000/H'400000 + 2x for 16-bit bus SDRAM

The SDRAM mode register latches the address signals when the MRS command is issue

This LSI does not support the burst read/burst write mode of SDRAM. When setting the
mode register, use the burst read/single write mode and set the burst length to 1. Setting
SDRAM mode register must be consistent with that in the bus controller.

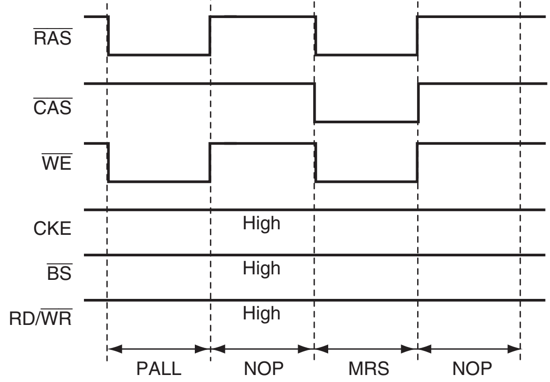


Figure 6.81 Timing of Setting SDRAM Mode Register

6.11.15 SDRAM Interface and Single Address Transfer by DMAC

When fast-page mode (BE = 1) is set for the SDRAM space, either fast-page access or fu can be selected, by the setting of bit DDS in DRAMCR, for the single address transfer by DMAC where the SDRAM space is specified as the transfer source or destination. At the time, the output timing of the $\overline{\text{DACK}}$ and $\overline{\text{BS}}$ signals can be changed. When BE = 0, a ful to the SDRAM space is performed with a single address transfer regardless of the setting DDS. However, the output timing of the $\overline{\text{DACK}}$ and $\overline{\text{BS}}$ signals can be changed by the se bit DDS.

The assertion timing of the $\overline{\text{DACK}}$ signals can be changed by the bit DKC in BCR1.

The output timing of the $\overline{\text{DACK}}$ signal can be independently set by the bits TRWL and C SDCR and bit DCK in BCR1 regardless of the setting of bit DDS.

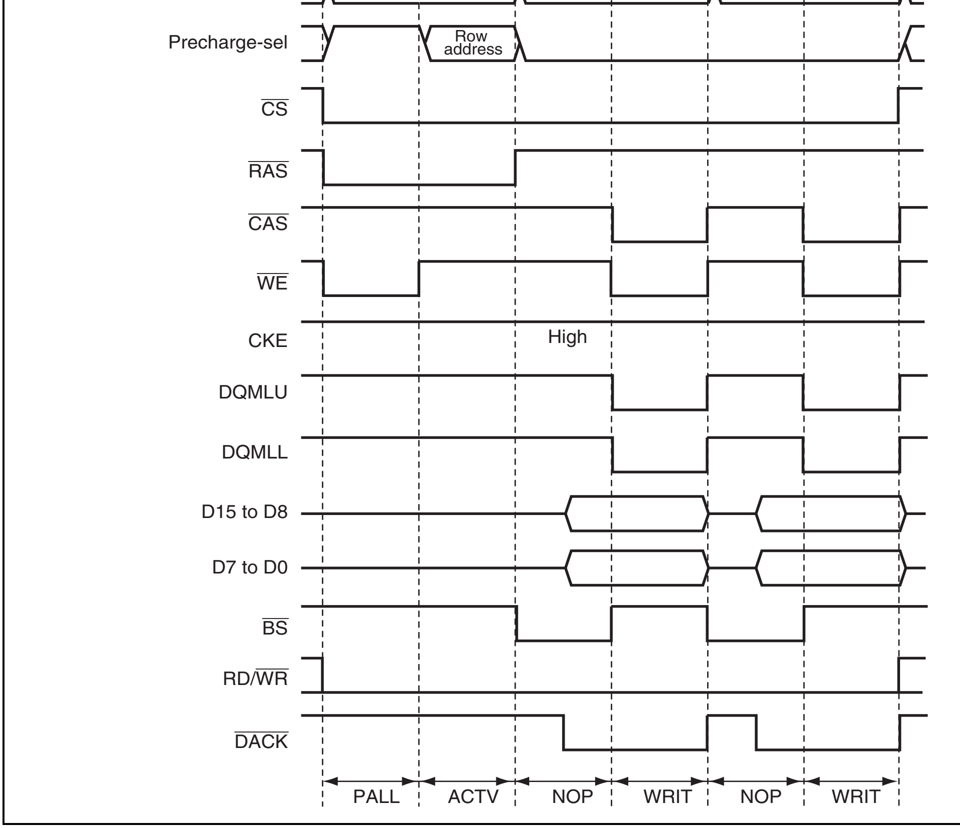


Figure 6.82 Output Timing Example of $\overline{\text{DACK}}$ when DDS = 1 (Write)

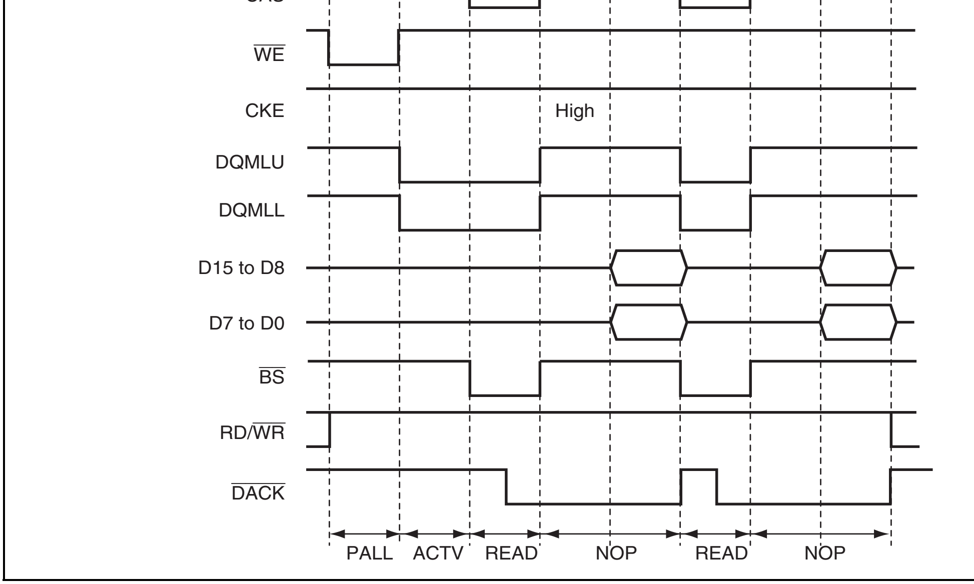


Figure 6.83 Output Timing Example of $\overline{\text{DACK}}$ when DDS = 1 (Read, CAS Laten

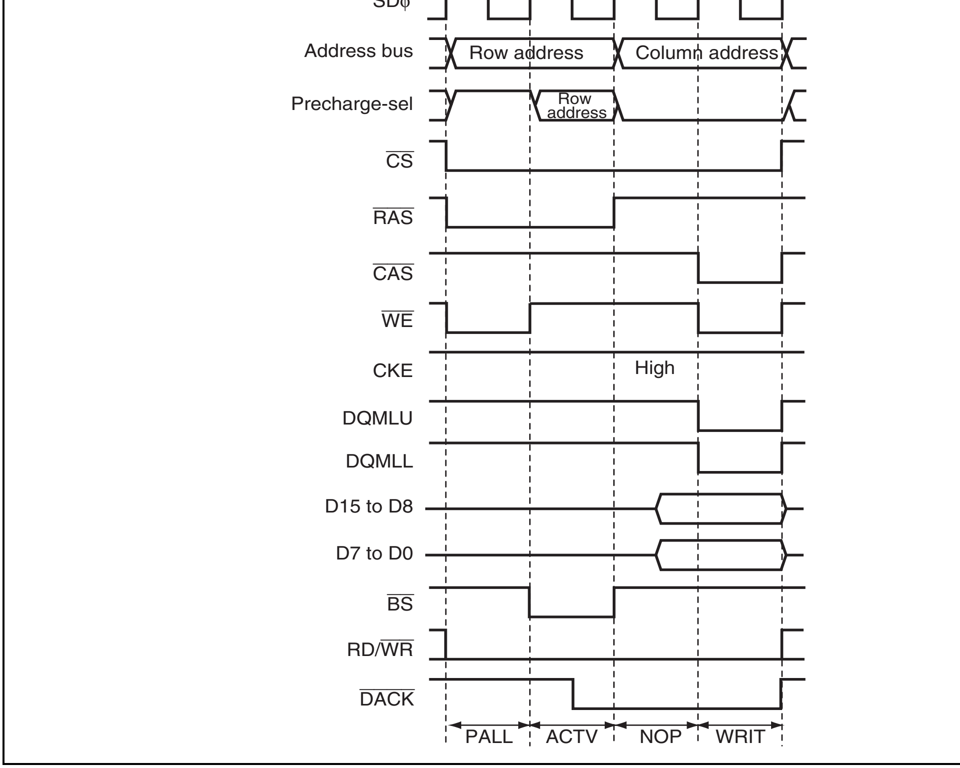


Figure 6.84 Output Timing Example of $\overline{\text{DACK}}$ when DDS = 0 (Write)

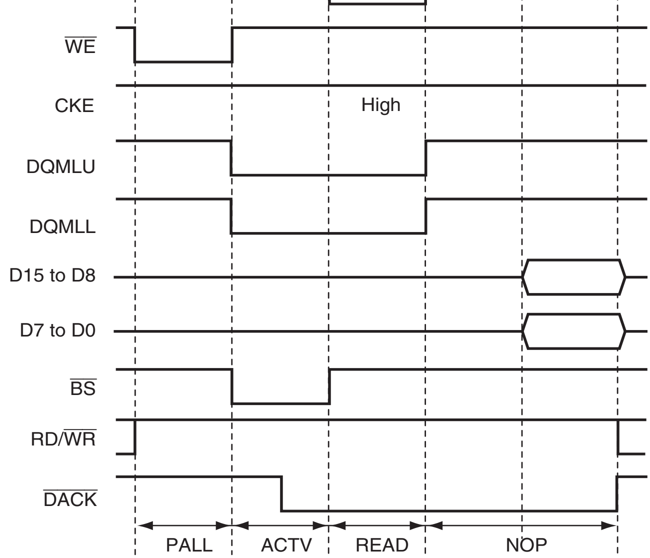


Figure 6.85 Output Timing Example of $\overline{\text{DACK}}$ when DDS = 0 (Read, CAS Laten

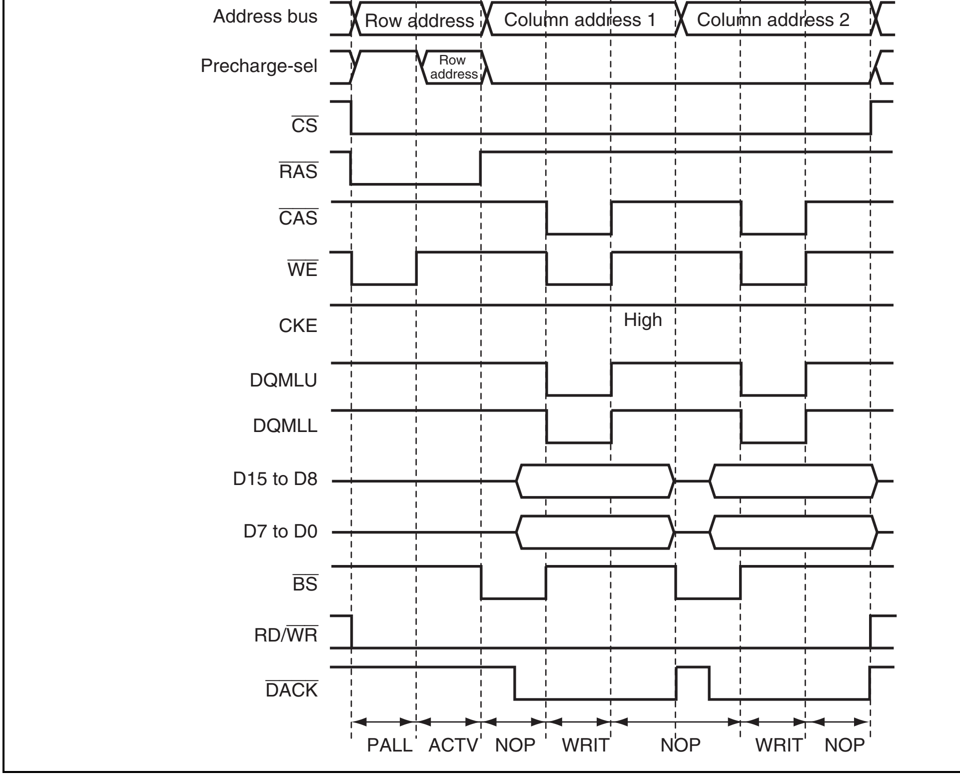


Figure 6.86 Output Timing Example of $\overline{\text{DACK}}$ when TRWL = 1 (Write)

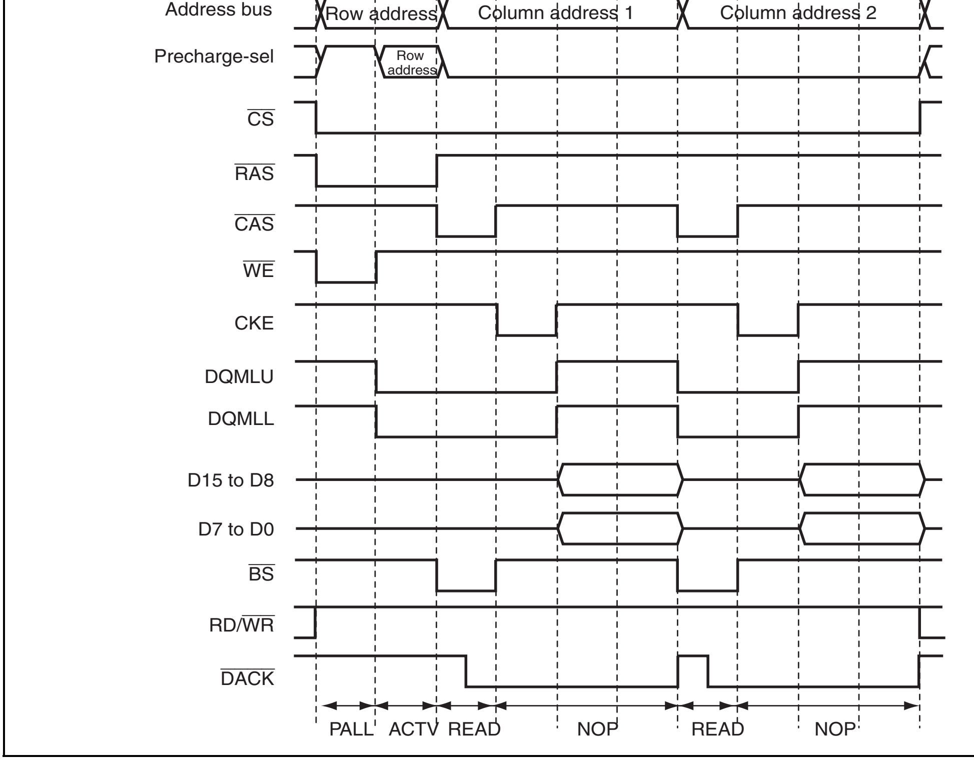


Figure 6.87 Output Timing Example of $\overline{\text{DACK}}$ when CKSPE = 1 (Read, CAS Latency = 2)

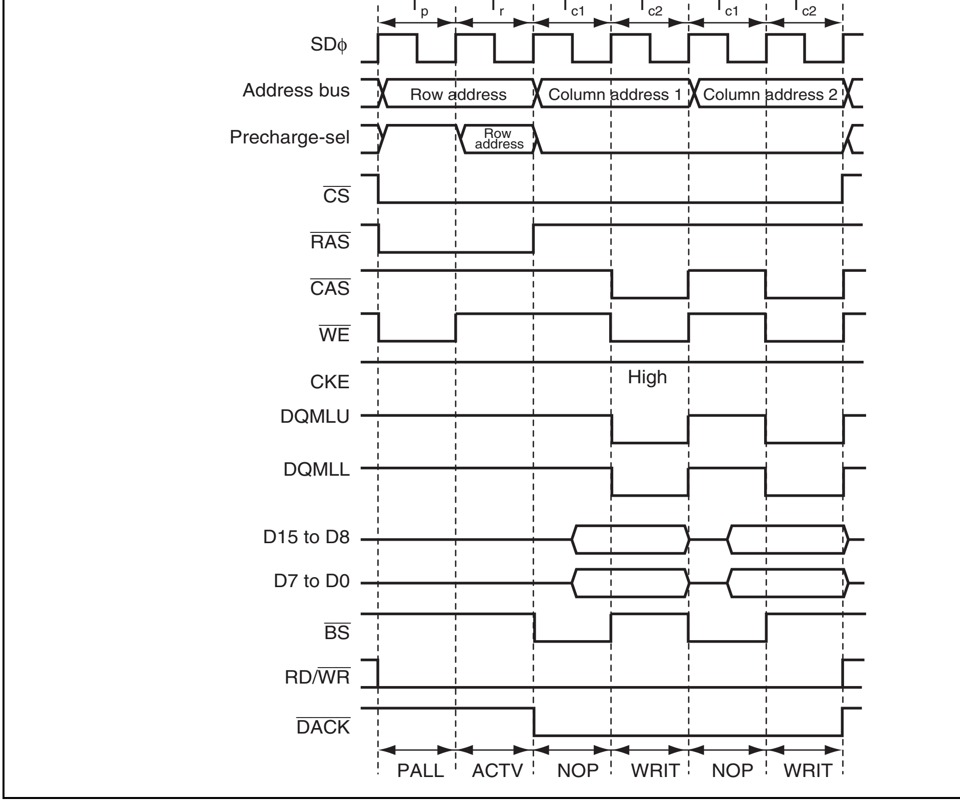


Figure 6.88 Output Timing Example of $\overline{\text{DACK}}$ when DKC = 1 and DDS = 1 (V

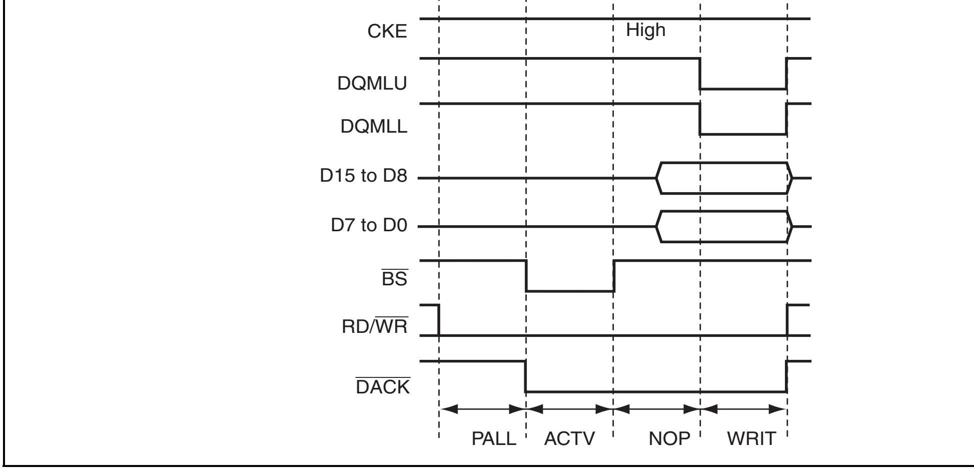


Figure 6.89 Output Timing Example of $\overline{\text{DACK}}$ when DKC = 1 and DDS = 0 (W

and write and previously accessed area.

1. When read cycles of different areas in the external address space occur consecutively
2. When an external write cycle occurs immediately after an external read cycle
3. When an external read cycle occurs immediately after an external write cycle
4. When an external access occurs immediately after a DMAC single address transfer (
   cycle)

Up to four idle cycles can be inserted under the conditions shown above. The number of
cycles to be inserted should be specified to prevent data conflicts between the output dat
previously accessed device and data from a subsequently accessed device.

Under conditions 1 and 2, which are the conditions to insert idle cycles after read, the nu
idle cycles can be selected from setting A specified by bits IDLCA1 and IDLCA0 in IDL
setting B specified by bits IDLCB1 and IDLCB0 in IDLCR: Setting A can be selected fr
four cycles, and setting B can be selected from one or two to four cycles. Setting A or B
specified for each area by setting bits IDLSEL7 to IDLSEL0 in IDLCR. Note that bits II
to IDLSEL0 correspond to the previously accessed area of the consecutive accesses.

The number of idle cycles to be inserted under conditions 3 and 4, which are conditions
idle cycles after write, can be determined by setting A as described above.

After the reset release, IDLCR is initialized to four idle cycle insertion under all conditio
shown above.

Table 6.28 shows the correspondence between conditions 1 to 4 and number of idle cycl
inserted for each area. Table 6.29 shows the correspondence between the number of idle
be inserted specified by settings A and B, and number of cycles to be inserted.

			1	B	B	B	B	B	B
Read after write	2	0	—	Invalid					
		1		A					
External access after single address transfer	3	0	—	Invalid					
		1		A					

[Legend]

A: Number of idle cycle insertion A is selected.

B: Number of idle cycle insertion B is selected.

Invalid: No idle cycle is inserted for the corresponding condition.

Table 6.29 Number of Idle Cycles Inserted

Bit Settings				
A		B		
IDLCA1	IDLCA0	IDLCB1	IDLCB0	Number of Cyc
—	—	0	0	0
0	0	—	—	1
0	1	0	1	2
1	0	1	0	3
1	1	1	1	4

and a data conflict is prevented.

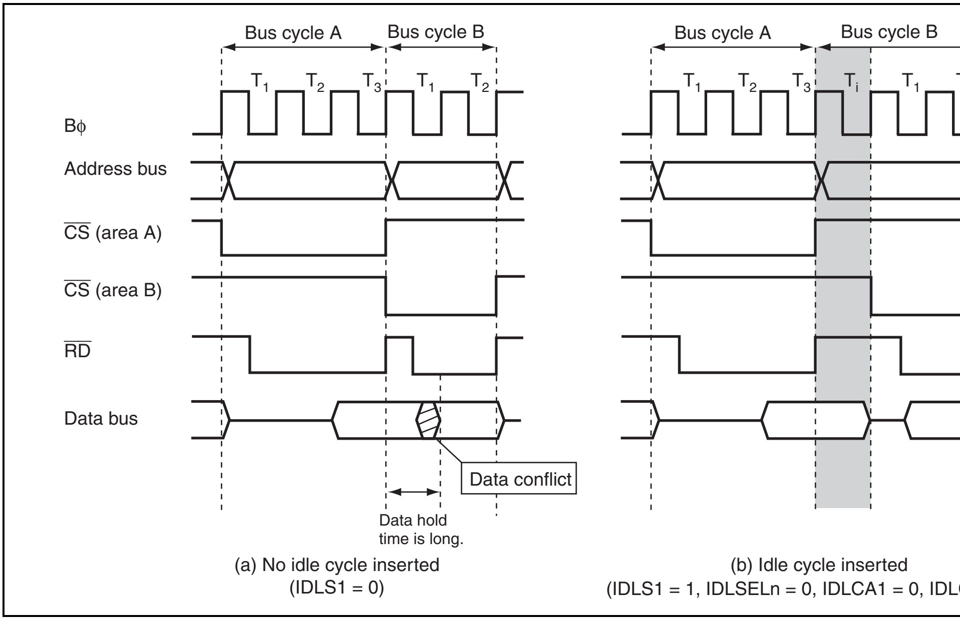


Figure 6.90 Example of Idle Cycle Operation (Consecutive Reads in Different

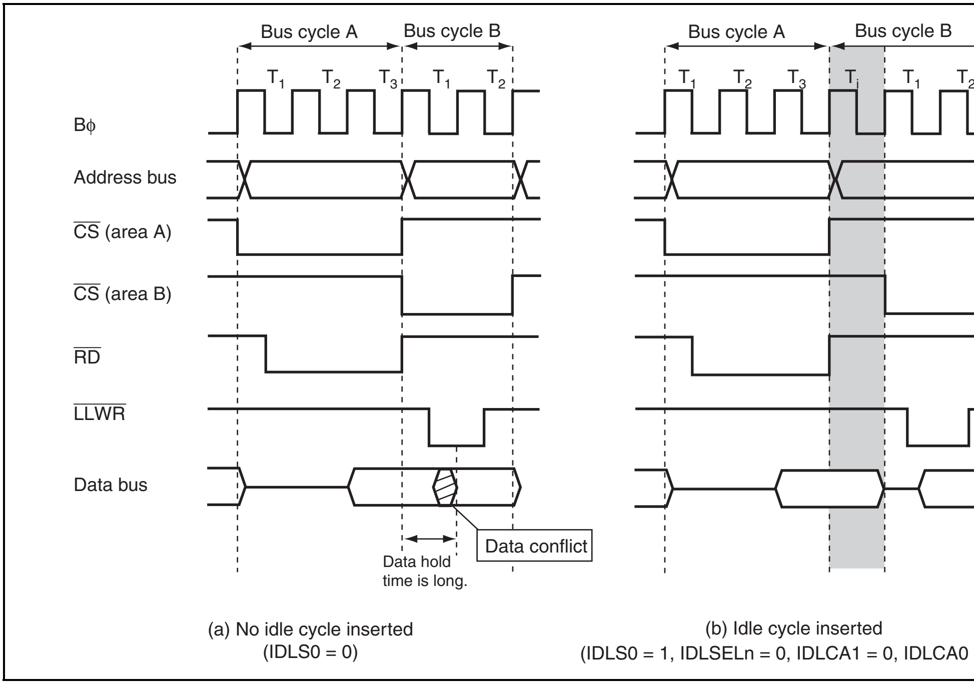


Figure 6.91 Example of Idle Cycle Operation (Write after Read)

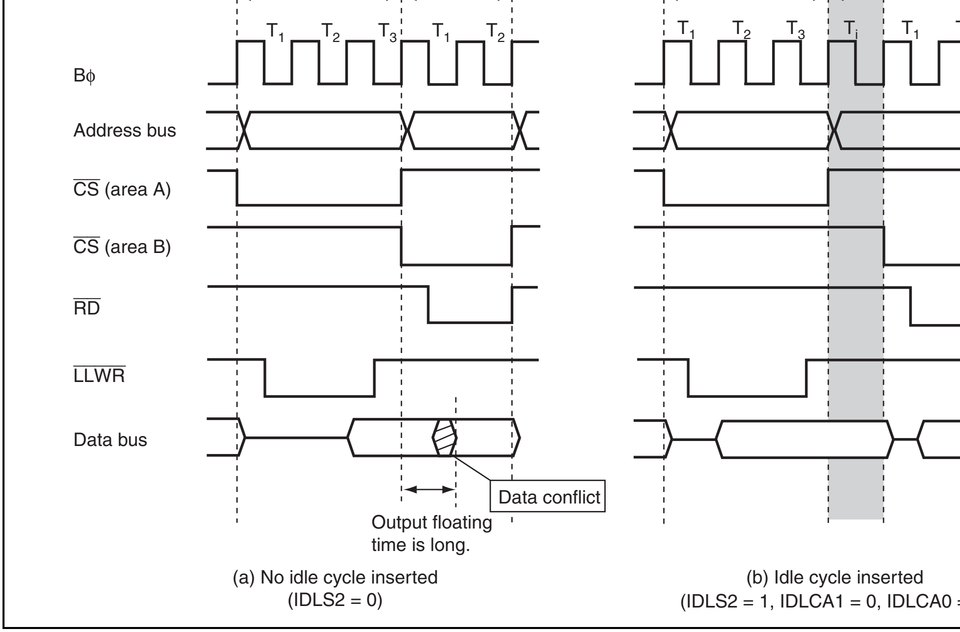


Figure 6.92 Example of Idle Cycle Operation (Read after Write)

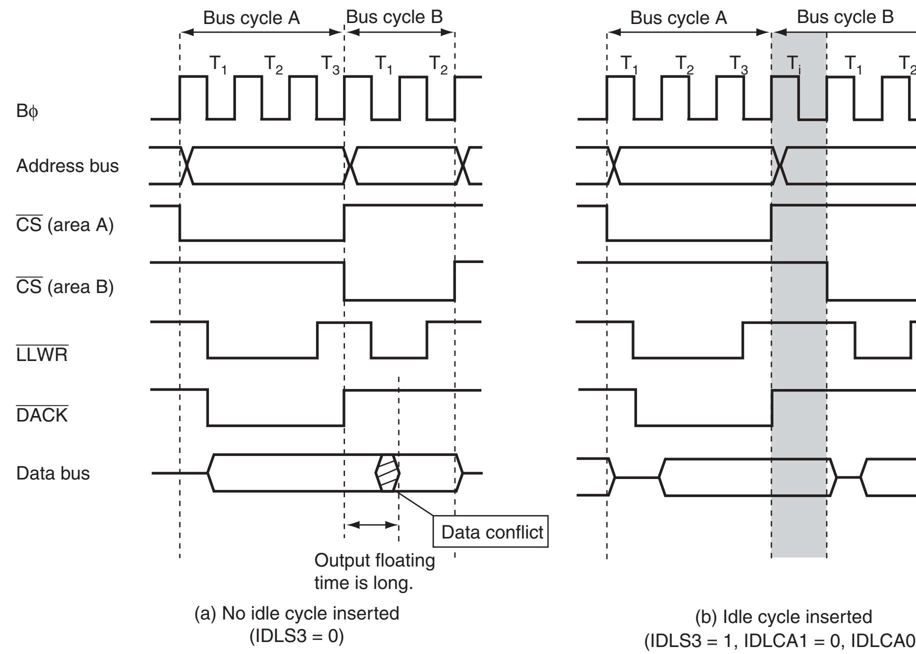


Figure 6.93 Example of Idle Cycle Operation (Write after Single Address Transfe

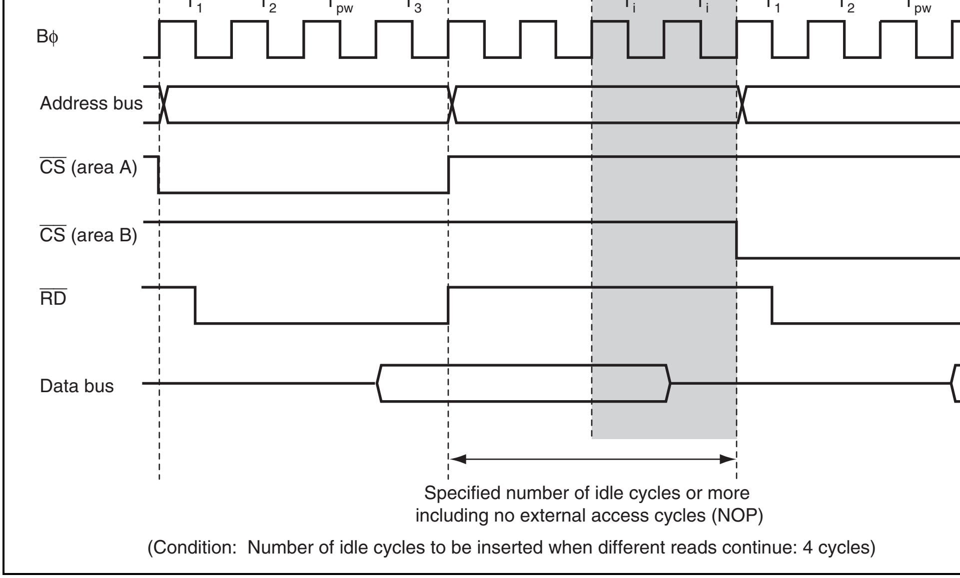


Figure 6.94 Idle Cycle Insertion Example

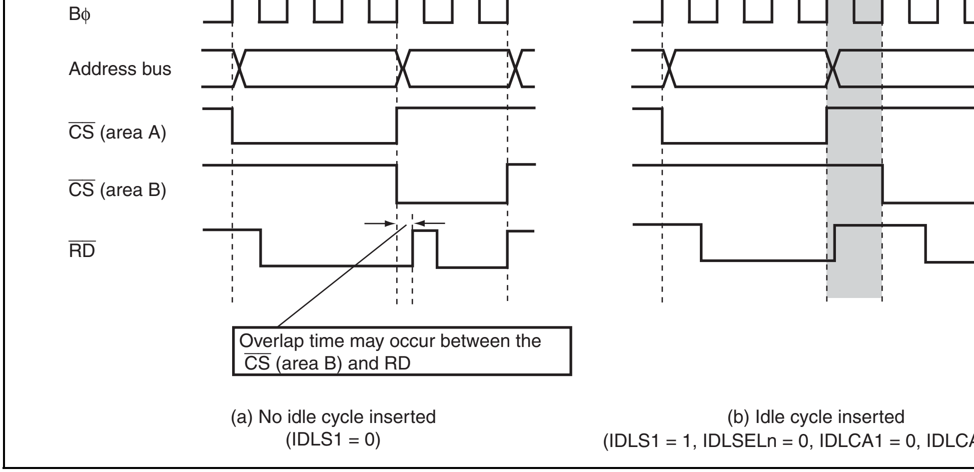


Figure 6.95 Relationship between Chip Select ($\overline{CS}$) and Read ($\overline{RD}$)

While the SDRAM space is accessed in a full access, the $\overline{CS2}$ signal is driven low even
cycle.

The idle cycle insertion is enabled even in a fast-page access in RAS down mode. The s
number of idle cycles is inserted. Figure 6.98 shows a timing example of the idle cycle i
RAS down mode.

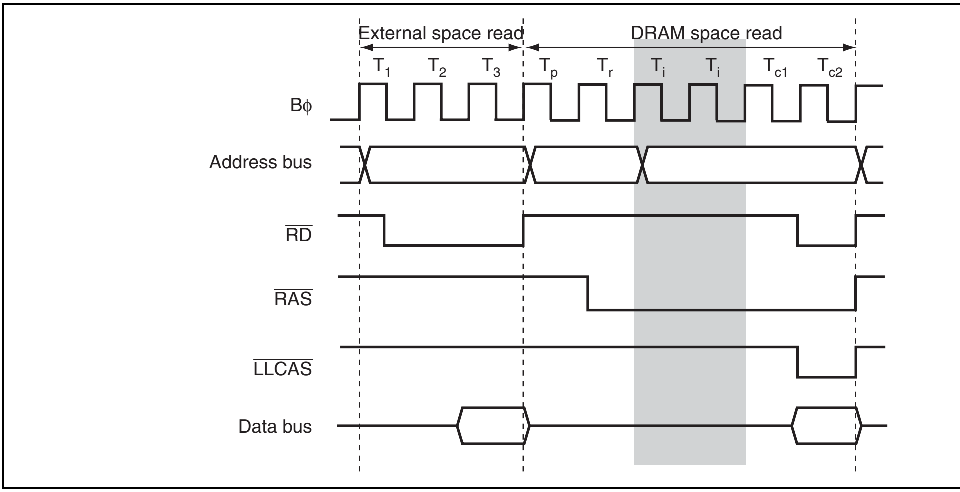


Figure 6.96 Example of DRAM Full Access after External Read (CAST =

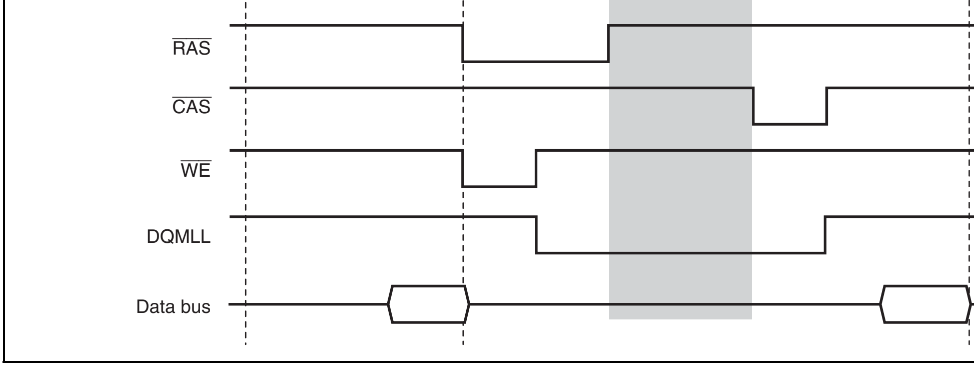


Figure 6.97 Example of SDRAM Full Access after External Read (CAS Latency

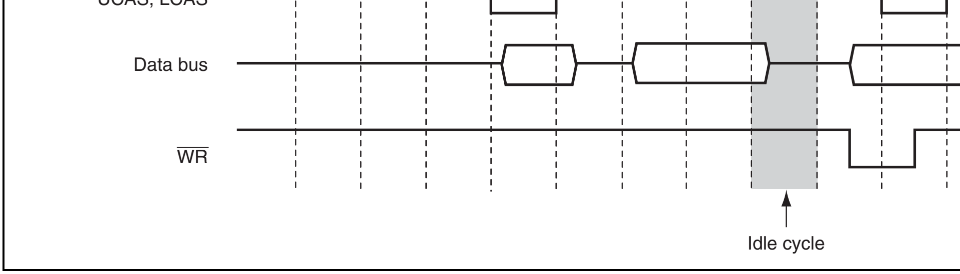


Figure 6.98 Example of Idle Cycles in RAS Down Mode (Write after Rea

									0	1	2 cycles i
									1	0	3 cycles i
									1	1	4 cycles i
Normal/DRAM/ SDRAM space read	Normal/DRAM/ SDRAM space read	—	—	—	0	—	—	—	—	—	Disabled
		—	—	—	1	0	0	0	—	—	1 cycle ir
							0	1			2 cycles i
							1	0			3 cycles i
							1	1			4 cycles i
						1			0	0	0 cycle ir
									0	1	2 cycles i
									1	0	3 cycles i
									1	1	4 cycles i
Normal/DRAM/ SDRAM space write	Normal/DRAM/ SDRAM space read	—	0	—	—	—	—	—	—	—	Disabled
		—	1	—	—	—	0	0	—	—	1 cycle ir
							0	1			2 cycles i
							1	0			3 cycles i
							1	1			4 cycles i
Single address write	Normal/DRAM/ SDRAM space write	0	—	—	—	—	—	—	—	—	Disabled
		1	—	—	—	—	0	0	—	—	1 cycle ir
							0	1			2 cycles i
							1	0			3 cycles i
							1	1			4 cycles i

RENESAS

LUCAS, LLCAS	High
DQMLU, DQMLL	High*[2]
$\overline{\text{AS}}$	High
$\overline{\text{RD}}$	High
$\overline{\text{BS}}$	High
RD/$\overline{\text{WR}}$	High*[3]
$\overline{\text{AH}}$	low
$\overline{\text{LHWR}}$, $\overline{\text{LLWR}}$	High
$\overline{\text{LUB}}$, $\overline{\text{LLB}}$	High
CKE	High
$\overline{\text{OE}}$	High
$\overline{\text{RAS}}$	High/Low*[4]
$\overline{\text{CAS}}$	High
$\overline{\text{WE}}$	High
$\overline{\text{DACKn}}$ (n = 3 to 0)	High

Notes: 1. Low when accessing the SDRAM in full access cycle

2. Low when reading the SDRAM in full access cycle

3. Low when accessing or writing to the DRAM/SDRAM in full access cycle

4. The pin state varies depending on the DRAM space access/ area access othe DRAM space, or RAS up mode/RAS down mode. For details, see figures 6.9 6.98.

In external extended mode, when the BRLE bit in BCR1 is set to 1, and the ICR bit for th corresponding pin is set to 1, the bus can be released to the external. Driving the $\overline{\text{BREQ}}$ p issues an external bus request to this LSI. When the $\overline{\text{BREQ}}$ pin is sampled, at the prescrib timing, the $\overline{\text{BACK}}$ pin is driven low, and the address bus, data bus, and bus control signal placed in the high-impedance state, establishing the external bus released state. For ICR, section 9, I/O Ports.

In the external bus released state, the CPU, DTC, and DMAC can access the internal spac the internal bus. When any one of the CPU, DTC, and DMAC attempts to accesses the ex address space, it temporarily defers initiation of the bus cycle, and waits for the bus reque the external bus master to be canceled.

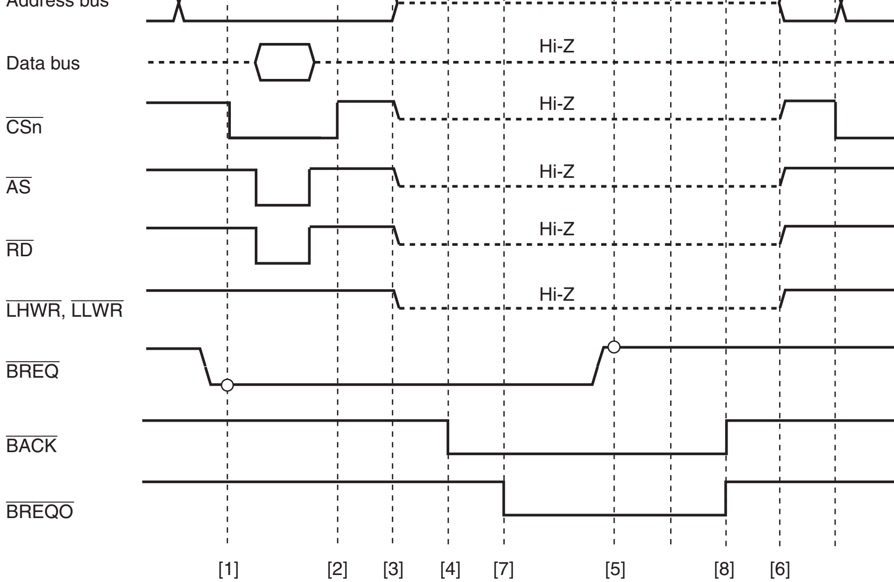
In the external bus released state, certain operations are suspended as follows until the bu from the external bus master is canceled:

- When a refresh is requested, refresh control is suspended.
- When the SLEEP instruction is executed to enter software standby mode or all-modul stop mode, control for software standby mode or all-module clock-stop mode is suspe
- When SCKCR is written to set the clock frequencies, changing of clock frequencies is suspended. For SCKCR, see section 22, Clock Pulse Generator.

If the BREQOE bit in BCR1is set to 1, the $\overline{\text{BREQO}}$ pin can be driven low to request canc of the bus request when any of the following requests are issued.

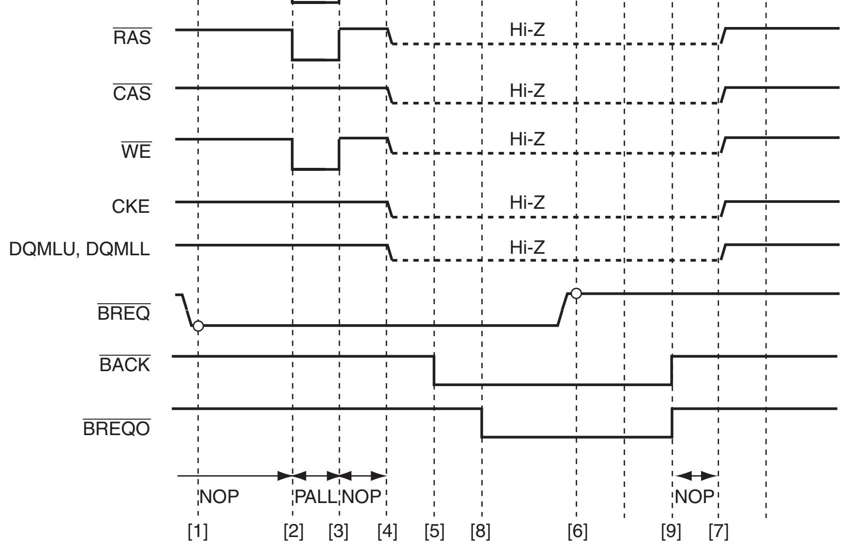
- When any one of the CPU, DTC, and DMAC attempts to access the external address s
- When a refresh is requested
- When a SLEEP instruction is executed to place the chip in software standby mode or module-clock-stop mode
- When SCKCR is written to set the clock frequencies

A23 to A0	High impedance
D15 to D0	High impedance
$\overline{\text{BS}}$	High impedance
$\overline{\text{CSn}}$ (n = 7 to 0)	High impedance
$\overline{\text{AS}}$	High impedance
$\overline{\text{AH}}$	High impedance
RD/$\overline{\text{WR}}$	High impedance
$\overline{\text{LUCAS}}$, $\overline{\text{LLCAS}}$	High impedance
$\overline{\text{RD}}$	High impedance
$\overline{\text{RAS}}$	High impedance
$\overline{\text{CAS}}$	High impedance
$\overline{\text{WE}}$	High impedance
DQMLU, DQMLL	High impedance
CKE	High impedance
$\overline{\text{OE}}$	High impedance
$\overline{\text{LUB}}$, $\overline{\text{LLB}}$	High impedance
$\overline{\text{LHWR}}$, $\overline{\text{LLWR}}$	High impedance
$\overline{\text{DACKn}}$ (n = 3 to 0)	High



[1] A low level of the BREQ signal is sampled at the rising edge of the Bϕ signal.
[2] The bus control signals are driven high at the end of the external space access cycle. It takes two cycles or more after the low level of the BREQ signal is sampled.
[3] The BACK signal is driven low, releasing bus to the external bus master.
[4] The BREQ signal state sampling is continued in the external bus released state.
[5] A high level of the BREQ signal is sampled.
[6] The external bus released cycles are ended one cycle after the BREQ signal is driven high.
[7] When the external space is accessed by an internal bus master during external bus released while the BR bit is set to 1, the BREQO signal goes low.
[8] Normally the BREQO signal goes high at the rising edge of the BACK signal.

Figure 6.99 Bus Released State Transition Timing (SRAM Interface is Not Us



[1] A low level of the $\overline{\text{BREQ}}$ signal is sampled at the rising edge of the Bϕ signal.
[2] The PALL command is issued.
[3] The bus control signals are driven high at the end of the external access cycle. It takes two cycles or more after the low level of the $\overline{\text{BREQ}}$ signal is sampled.
[4] The $\overline{\text{BACK}}$ signal is driven low, releasing bus to the external bus master.
[5] The $\overline{\text{BREQ}}$ signal state sampling is continued in the external bus released state.
[6] A high level of the $\overline{\text{BREQ}}$ signal is sampled.
[7] The $\overline{\text{BACK}}$ signal is driven high, ending external bus release cycle after one cycle.
[8] When the external space is accessed by an internal bus master or a refresh cycle is requested during e bus released while the BREQOE bit is set to 1, the $\overline{\text{BREQO}}$ signal goes low.
[9] Normally the $\overline{\text{BREQO}}$ signal goes high at the rising edge of the $\overline{\text{BACK}}$ signal.

Figure 6.100 Bus Released State Transition Timing (SRAM Interface is Us

Table 6.33 Number of Access Cycles for On-Chip Memory Spaces

Access Space	Access	Number of Access C
On-chip ROM space	Read	One Iϕ cycle
	Write	Three Iϕ cycles
On-chip RAM space	Read	One Iϕ cycle
	Write	One Iϕ cycle

In access to the registers for on-chip peripheral modules, the number of access cycles diff according to the register to be accessed. When the dividing ratio of the operating clock of master and that of a peripheral module is 1 : n, synchronization cycles using a clock divid to n-1 are inserted for register access in the same way as for external bus clock division.

Table 6.34 lists the number of access cycles for registers of on-chip peripheral modules.

Table 6.34 Number of Access Cycles for Registers of On-Chip Peripheral Modules

	Number of Cycles		
Module to be Accessed	**Read**	**Write**	**Write Data Buffer F**
DMAC registers	Two Iϕ		Disabled
MCU operating mode, clock pulse generator, power-down control registers, interrupt controller, bus controller, and DTC registers	Two Iϕ	Three Iϕ	Disabled
I/O port registers of PFCR and WDT	Two Pϕ	Three Pϕ	Disabled
I/O port registers other than PFCR and PORTM, TPU, PPG, TMR, SCI, SCI0 to SCI2, SCI4, A/D, and D/A registers	Two Pϕ		Enabled
I/O port registers of PORTM, USB, SCI5, and SCI6	Three Pϕ		Enabled

continues for two cycles or longer, and there is an internal access next, an external write
executed in the first two cycles. However, from the next cycle onward, internal accesses
memory or internal I/O register read/write) and the external address space write rather th
waiting until it ends are executed in parallel.

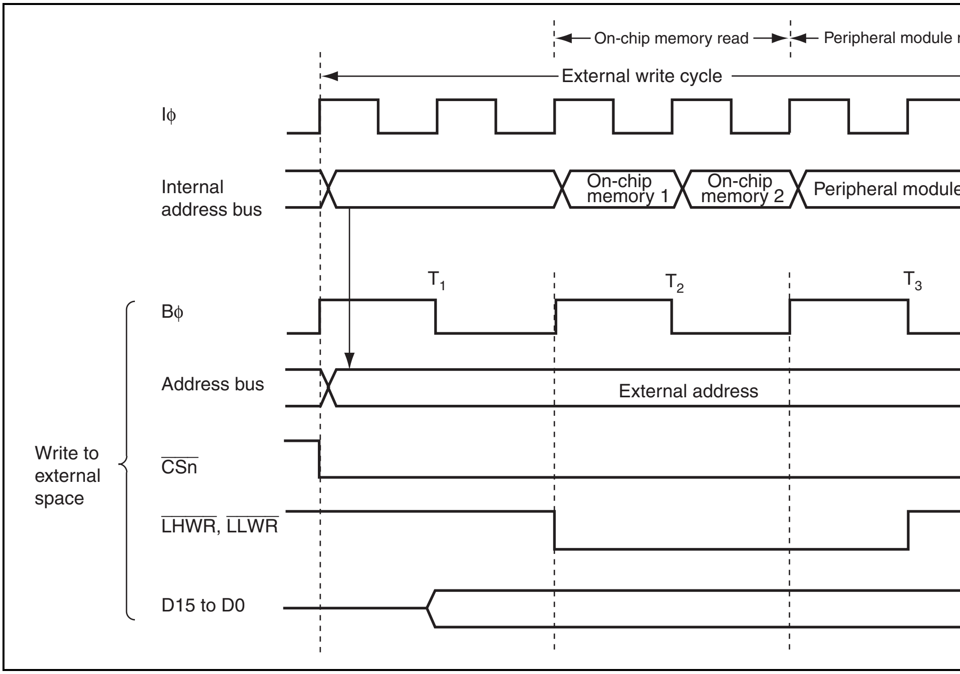


Figure 6.101 Example of Timing when Write Data Buffer Function is Use

is performed in the first two cycles. However, from the next cycle onward an internal me
an external access and internal I/O register write are executed in parallel rather than waiti
it ends.

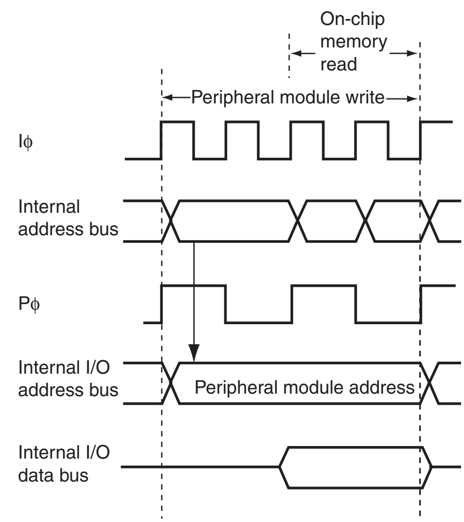


Figure 6.102 Example of Timing when Peripheral Module Write Data Buffer Function is Used

6.16.1 Operation

The bus arbiter detects the bus masters' bus request signals, and if the bus is requested, s
request acknowledge signal to the bus master. If there are bus requests from more than o
master, the bus request acknowledge signal is sent to the one with the highest priority. W
master receives the bus request acknowledge signal, it takes possession of the bus until t
is canceled.

The priority of the internal bus arbitration:

DMAC > DTC > CPU

The priority of the external bus arbitration:

Refresh > External bus release request > External access by the CPU, DTC, or DM

If the DMAC or DTC accesses continue, the CPU can be given priority over the DMAC
to execute the bus cycles alternatively between them by setting the IBCCS bit in BCR2.
case, the priority between the DMAC and DTC does not change.

An internal bus access by the CPU, DTC, or DMAC, an external bus release, and the re
be executed in parallel.

The timing for transfer of the bus is at the end of the bus cycle. In sleep mode, the bus is transferred synchronously with the clock.

Note, however, that the bus cannot be transferred in the following cases.

- The word or longword access is performed in some divisions.
- Stack handling is performed in multiple bus cycles.
- Transfer data read or write by memory transfer instructions, block transfer instruction instruction.

  (In the block transfer instructions, the bus can be transferred in the write cycle and the following transfer data read cycle.)
- From the target read to write in the bit manipulation instructions or memory operation instructions.

  (In an instruction that performs no write operation according to the instruction conditi a cycle corresponding the write cycle)

(2) DTC

The DTC sends the internal bus arbiter a request for the bus when an activation request is generated. When the DTC accesses an external bus space, the DTC first takes control of t from the internal bus arbiter and then requests a bus to the external bus arbiter.

Once the DTC takes control of the bus, the DTC continues the transfer processing cycles. master whose priority is higher than the DTC requests the bus, the DTC transfers the bus higher priority bus master. If the IBCCS bit in BCR2 is set to 1, the DTC transfers the bu CPU.

Note, however, that the bus cannot be transferred in the following cases.

After the DMAC takes control of the bus, it may continue the transfer processing cycles
the bus at the end of every bus cycle depending on the conditions.

The DMAC continues transfers without releasing the bus in the following case:

- Between the read cycle in the dual-address mode and the write cycle corresponding t
  cycle

If no bus master of a higher priority than the DMAC requests the bus and the IBCCS bit
is cleared to 0, the DMAC continues transfers without releasing the bus in the following

- During 1-block transfers in the block transfer mode
- During transfers in the burst mode

In other cases, the DMAC transfers the bus at the end of the bus cycle.

(4) External Bus Release

When the $\overline{\text{BREQ}}$ pin goes low and an external bus release request is issued while the BR
BCR1 is set to 1 with the corresponding ICR bit set to 1, a bus request is sent to th

(5) Refresh

When area 2 is specified as the DRAM space or SDRAM space with the RFSHE bit in R
to 1, RTCNT starts to count up. When the RTCOR value matches RTCNT, a bus reques
the bus arbiter.

A refresh cycle is inserted on completion of the external bus cycle. A refresh cycle is no
consecutively inserted. Once a refresh cycle is inserted, the bus is passed to another bus
When the bus is passed, if there is no bus request from other bus masters, NOP cycles ar

other than an instruction fetch access.

(2) Mode Settings

The burst read-burst write mode of synchronous DRAM is not supported.

When setting the mode register of synchronous DRAM, the burst read-single write mode selected and the burst length must be 1.

(3) External Bus Release Function and All-Module-Clock-Stop Mode

In this LSI, if the ACSE bit in MSTPCRA is set to 1 and a SLEEP instruction is executed the sleep state after shutting off the clocks to all peripheral modules (MSTPCRA and MS = H'FFFFFFFF) or allowing operation of the 8-bit timer module alone (MSTPCRA and M = H'F[C to F]FFFFFF), the all-module-clock-stop mode is entered in which the clock for controller and I/O ports is also stopped. For details, see section 23, Power-Down Modes.

In this state, the external bus release function is halted. To use the external bus release fu sleep mode, the ACSE bit in MSTPCR must be cleared to 0. Conversely, if a SLEEP inst place the chip in all-module-clock-stop mode is executed in the external bus released stat transition to all-module-clock-stop mode is deferred and performed until after the bus is recovered.

(4) External Bus Release Function and Software Standby Mode

In this LSI, internal bus master operation does not stop even while the bus is released, 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 released, the transition to software standby mode is deferred and performed after the bus recovered.

setting the BREQOE bit in BCR1 to 1.

(6) $\overline{\text{BREQO}}$ Output Timing

When the BREQOE bit is set to 1 and the $\overline{\text{BREQO}}$ signal is output, both the $\overline{\text{BREQO}}$ an
signals may go low simultaneously.

This will occur if the next external access request occurs while internal bus arbitration is
progress after the chip samples a low level of the $\overline{\text{BREQ}}$ signal.

(7) Refresh Settings

In single-chip activation mode, the setting of the RFSHE bit in REFCR should be made
setting the EXPE bit in SYSCR to 1. For SYSCR, see section 3, MCU Operating Modes

(8) Refresh Timer Settings

The setting of bits RTCK2 to RTCK0 in REFCR should be made after RTCNT and RTC
been set. When changing RTCNT and RTCOR, the counter operation should be halted.
changing bits RTCK2 to RTCK0, change them only after disabling external bus release
write data buffer function is in use, disabling the write data buffer function and reading
external space.

SBYCR, see section 23, Power-Down Modes.

(11) RAS Down Mode and Clock Frequencies Setting for DRAM/SDRAM

Write access to SCKCR for setting the clock frequencies should be performed in RAS up (RCDM = 0). When RAS down mode (RCDM = 1) is used, set the RCDM bit to 0 before to SCKCR. RAS down mode should be set again after clock frequencies are set. For SCK section 22, Clock Pulse Generator.

(12) Cluster Transfer to SDRAM Space

Cluster transfer mode is available for the SDRAM with CAS latency of 2. When the SDR used in cluster transfer mode, the SDRAM with CAS latency of 2 should be used. In clus transfer mode, the write-precharge output delay function by the TRWL bit is not availabl TRWL bit must be cleared to 0.

- DMAC activation methods are auto-request, on-chip module interrupt, and external r

  Auto request: CPU activates (cycle stealing or burst access can be sele

  On-chip module interrupt: Interrupt requests from on-chip peripheral modules can b as an activation source

  External request: Low level or falling edge detection of the $\overline{\text{DREQ}}$ signal selected. External request is available for all four chann

  In block transfer mode, low level detection is only avail

- Dual or single address mode can be selected as address mode

  Dual address mode: Both source and destination are specified by addresses

  Single address mode: Either source or destination is specified by the $\overline{\text{DREQ}}$ signal an other is specified by address

- Normal, repeat, or block transfer can be selected as transfer mode

  Normal transfer mode: One byte, one word, or one longword data is transferred single transfer request

  Repeat transfer mode: One byte, one word, or one longword data is transferred single transfer request

  Repeat size of data is transferred and then a transfer add returns to the transfer start address

  Up to 65536 transfers (65,536 bytes/words/longwords) as repeat size

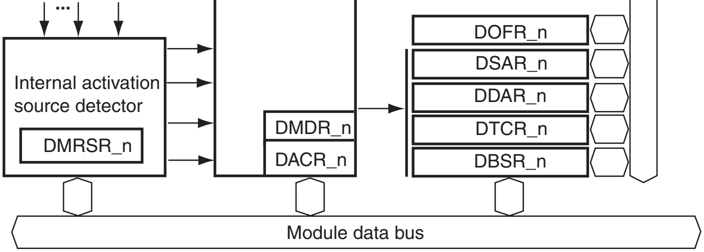
  Block transfer mode: One block data is transferred at a single transfer request

  Up to 65,536 bytes/words/longwords can be set as block respective boundary

Data is divided according to its address (byte or word) when it is transferred

- Two types of interrupts can be requested to the CPU

  A transfer end interrupt is generated after the number of data specified by the transfer is transferred. A transfer escape end interrupt is generated when the remaining total tr size is less than the transfer data size at a single transfer request, when the repeat size transfer is completed, or when the extended repeat area overflows.



[Legend]

DSAR_n:	DMA source address register	$\overline{DREQn}$: DMA transfer request
DDAR_n:	DMA destination address register	$\overline{DACKn}$: DMA transfer acknowledge
DOFR_n:	DMA offset register	$\overline{TENDn}$: DMA transfer end
DTCR_n:	DMA transfer count register	n = 0 to 3
DBSR_n:	DMA block size register	
DMDR_n:	DMA mode control register	
DACR_n:	DMA address control register	
DMRSR_n:	DMA module request select register	

Figure 7.1 Block Diagram of DMAC

1	DMA transfer request 1	$\overline{\text{DREQ1}}$	Input	Channel 1 external reque
	DMA transfer acknowledge 1	$\overline{\text{DACK1}}$	Output	Channel 1 single address acknowledge
	DMA transfer end 1	$\overline{\text{TEND1}}$	Output	Channel 1 transfer end
2	DMA transfer request 2	$\overline{\text{DREQ2}}$	Input	Channel 2 external reque
	DMA transfer acknowledge 2	$\overline{\text{DACK2}}$	Output	Channel 2 single address acknowledge
	DMA transfer end 2	$\overline{\text{TEND2}}$	Output	Channel 2 transfer end
3	DMA transfer request 3	$\overline{\text{DREQ3}}$	Input	Channel 3 external reque
	DMA transfer acknowledge 3	$\overline{\text{DACK3}}$	Output	Channel 3 single address acknowledge
	DMA transfer end 3	$\overline{\text{TEND3}}$	Output	Channel 3 transfer end

- DMA block size register_0 (DBSR_0)
- DMA mode control register_0 (DMDR_0)
- DMA address control register_0 (DACR_0)
- DMA module request select register_0 (DMRSR_0)

Channel 1:

- DMA source address register_1 (DSAR_1)
- DMA destination address register_1 (DDAR_1)
- DMA offset register_1 (DOFR_1)
- DMA transfer count register_1 (DTCR_1)
- DMA block size register_1 (DBSR_1)
- DMA mode control register_1 (DMDR_1)
- DMA address control register_1 (DACR_1)
- DMA module request select register_1 (DMRSR_1)

Channel 2:

- DMA source address register_2 (DSAR_2)
- DMA destination address register_2 (DDAR_2)
- DMA offset register_2 (DOFR_2)
- DMA transfer count register_2 (DTCR_2)
- DMA block size register_2 (DBSR_2)
- DMA mode control register_2 (DMDR_2)
- DMA address control register_2 (DACR_2)
- DMA module request select register_2 (DMRSR_2)

7.3.1 DMA Source Address Register (DSAR)

DSAR is a 32-bit readable/writable register that specifies the transfer source address. DS updates the transfer source address every time data is transferred. When DDAR is specifi destination address (the DIRS bit in DACR is 1) in single address mode, DSAR is ignorec

Although DSAR can always be read from by the CPU, it must be read from in longwords must not be written to while data for the channel is being transferred.

Bit	31	30	29	28	27	26	25
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	23	22	21	20	19	18	17
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	23	22	21	20	19	18	17
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	23	22	21	20	19	18	17
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Although DTCR can always be read from by the CPU, it must be read from in longword
must not be written to while data for the channel is being transferred.

Bit	31	30	29	28	27	26	25
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	23	22	21	20	19	18	17
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9
Bit Name	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	BKSZH31 to BKSZH16	Undefined	R/W	Specify the repeat size or block size. When H'0001 is set, the repeat or block size is one word, or one longword. When H'0000 is se means the maximum value (refer to table 7.1). DMA is in operation, the setting is fixed.
15 to 0	BKSZ15 to BKSZ0	Undefined	R/W	Indicate the remaining repeat or block size whil DMA is in operation. The value is decremented every time data is transferred. When the remair becomes 0, the value of the BKSZH bits is load the same value as the BKSZH bits.

DMDR controls the DMAC operation.

- DMDR_0

Bit	31	30	29	28	27	26	25
Bit Name	DTE	DACKE	TENDE	—	DREQS	NRD	—
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	23	22	21	20	19	18	17
Bit Name	ACT	—	—	—	ERRF	—	ESIF
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R/(W)*	R	R/(W)*

Bit	15	14	13	12	11	10	9
Bit Name	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	7	6	5	4	3	2	1
Bit Name	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R/W	R/W

Note: * Only 0 can be written to this bit after having been read as 1, to clear the flag.

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	7	6	5	4	3	2	1
Bit Name	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R/W	R/W

Note: * Only 0 can be written to this bit after having been read as 1, to clear the flag.

transfer.

In block transfer mode, if writing 0 to this bit w
being transferred, this bit is cleared to 0 after t
1-block size data transfer.

If an event which stops (sustains) a transfer oc
externally, this bit is automatically cleared to 0
the transfer.

Operating modes and transfer methods must n
changed while this bit is set to 1.

0: Disables a data transfer

1: Enables a data transfer (DMA is in operatio

[Clearing conditions]

- When the specified total transfer size of tra completed
- When a transfer is stopped by an overflow by a repeat size end
- When a transfer is stopped by an overflow by an extended repeat size end
- When a transfer is stopped by a transfer s interrupt
- When clearing this bit to 0 to stop a transfe

In block transfer mode, this bit changes after t
block transfer.

- When an address error or an NMI interrupt requested
- In the reset state or hardware standby mod

				1: Disables TEND signal output
28	—	0	R/W	Reserved Initial value should not be changed.
27	DREQS	0	R/W	$\overline{\text{DREQ}}$ Select Selects whether a low level or the falling edge o $\overline{\text{DREQ}}$ signal used in external request mode is When a block transfer is performed in external mode, clear this bit to 0. 0: Low level detection 1: Falling edge detection (the first transfer after transfer enabled is detected on a low level)
26	NRD	0	R/W	Next Request Delay Selects the accepting timing of the next transfe 0: Starts accepting the next transfer request aft completion of the current transfer 1: Starts accepting the next transfer request on after completion of the current transfer
25, 24	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
23	ACT	0	R	Active State Indicates the operating state for the channel. 0: Waiting for a transfer request or a transfer di state by clearing the DTE bit to 0 1: Active state
22 to 20	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
				generated [Clearing condition] • When clearing to 0 after reading ERRF = 1 [Setting condition] • When an address error or an NMI interrupt generated However, when an address error or an NMI in been generated in DMAC module stop mode, not set to 1.
18	—	0	R	Reserved This bit is always read as 0 and cannot be mo
17	ESIF	0	R/(W)*	Transfer Escape Interrupt Flag Indicates that a transfer escape end interrupt requested. A transfer escape end means that is terminated before the transfer counter reach 0: A transfer escape end interrupt has not bee requested 1: A transfer escape end interrupt has been re [Clearing conditions] • When setting the DTE bit to 1 • When clearing to 0 before reading ESIF = [Setting conditions] • When a transfer size error interrupt is requ • When a repeat size end interrupt is reques • When a transfer end interrupt by an extend area overflow is requested

				• When setting the DTE bit to 1 • When clearing to 0 after reading DTIF = 1 [Setting condition] • When DTCR reaches 0 and the transfer is completed
15 14	DTSZ1 DTSZ0	0 0	R/W R/W	Data Access Size 1 and 0 Select the data access size for a transfer. 00: Byte size (eight bits) 01: Word size (16 bits) 10: Longword size (32 bits) 11: Setting prohibited
13 12	MDS1 MDS0	0 0	R/W R/W	Transfer Mode Select 1 and 0 Select the transfer mode. 00: Normal transfer mode 01: Block transfer mode 10: Repeat transfer mode 11: Setting prohibited

				• In normal or repeat transfer mode, the tota size set in DTCR is less than the data acc • In block transfer mode, the total transfer si DTCR is less than the block size 0: Disables a transfer size error interrupt reque 1: Enables a transfer size error interrupt reque
10	—	0	R	Reserved This bit is always read as 0 and cannot be mo
9	ESIE	0	R/W	Transfer Escape Interrupt Enable Enables/disables a transfer escape end interr request. When the ESIF bit is set to 1 with this 1, a transfer escape end interrupt is requested CPU or DTC. The transfer end interrupt reque cleared by clearing this bit or the ESIF bit to 0 0: Disables a transfer escape end interrupt 1: Enables a transfer escape end interrupt
8	DTIE	0	R/W	Data Transfer End Interrupt Enable Enables/disables a transfer end interrupt requ transfer counter. When the DTIF bit is set to 1 bit set to 1, a transfer end interrupt is requeste CPU or DTC. The transfer end interrupt reque cleared by clearing this bit or the DTIF bit to 0 0: Disables a transfer end interrupt 1: Enables a transfer end interrupt

				11: External request
5	DTA	0	R/W	Data Transfer Acknowledge This bit is valid in DMA transfer by the on-chip r interrupt source. This bit enables or disables to source flag selected by DMRSR. 0: To clear the source in DMA transfer is disabl Since the on-chip module interrupt source is cleared in DMA transfer, it should be cleared CPU or DTC transfer. 1: To clear the source in DMA transfer is enable Since the on-chip module interrupt source is in DMA transfer, it does not require an interru the CPU or DTC transfer.
4, 3	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

000: Priority level 0 (low)

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 (high)

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

R/W	R	R	R/W	R/W	R	R	R/W	F

Bit	15	14	13	12	11	10	9	
Bit Name	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SA
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R	R	R/W	R/W	R/W	R/W	F

Bit	7	6	5	4	3	2	1	
Bit Name	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DA
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R	R	R/W	R/W	R/W	R/W	F

Bit	Bit Name	Initial Value	R/W	Description
31	AMS	0	R/W	Address Mode Select Selects address mode from single or dual addr mode. In single address mode, the $\overline{\text{DACK}}$ pin is according to the DACKE bit. 0: Dual address mode 1: Single address mode
30	DIRS	0	R/W	Single Address Direction Select Specifies the data transfer direction in single ad mode. This bit s ignored in dual address mode. 0: Specifies DSAR as source address 1: Specifies DDAR as destination address
29 to 27	—	0	R/W	Reserved These bits are always read as 0 and cannot be modified.

				transfer is requested after 1-block data transfe this bit is set to 1, the DTE bit in DMDR is clea At this time, the ESIF bit in DMDR is set to 1 t that a repeat size end interrupt is requested. 0: Disables a repeat size end interrupt 1: Enables a repeat size end interrupt
25	ARS1	0	R/W	Area Select 1 and 0
24	ARS0	0	R/W	Specify the block area or repeat area in block transfer mode. 00: Specify the block area or repeat area on th address 01: Specify the block area or repeat area on th destination address 10: Do not specify the block area or repeat are 11: Setting prohibited
23, 22	—	All 0	R	Reserved These bits are always read as 0 and cannot b modified.
21	SAT1	0	R/W	Source Address Update Mode 1 and 0
20	SAT0	0	R/W	Select the update method of the source addre (DSAR). When DSAR is not specified as the tr source in single address mode, this bit is igno 00: Source address is fixed 01: Source address is updated by adding the 10: Source address is updated by adding 1, 2, according to the data access size 11: Source address is updated by subtracting according to the data access size

				10: Destination address is updated by adding 1 according to the data access size 11: Destination address is updated by subtracti or 4 according to the data access size
15	SARIE	0	R/W	Interrupt Enable for Source Address Extended Overflow Enables/disables an interrupt request for an ex area overflow on the source address. When an extended repeat area overflow on the address occurs while this bit is set to 1, the DT DMDR is cleared to 0. At this time, the ESIF bit DMDR is set to 1 to indicate an interrupt by an repeat area overflow on the source address is requested. When block transfer mode is used with the exte repeat area function, an interrupt is requested a completion of a 1-block size transfer. When set DTE bit in DMDR of the channel for which a tra been stopped to 1, the transfer is resumed from state when the transfer is stopped. When the extended repeat area is not specified is ignored. 0: Disables an interrupt request for an extended overflow on the source address 1: Enables an interrupt request for an extended overflow on the source address
14, 13	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

				When an overflow in the extended repeat area with the SARIE bit set to 1, an interrupt can be requested. Table 7.3 shows the settings and a the extended repeat area.
7	DARIE	0	R/W	Destination Address Extended Repeat Area C Interrupt Enable Enables/disables an interrupt request for an e area overflow on the destination address. When an extended repeat area overflow on th destination address occurs while this bit is set DTE bit in DMDR is cleared to 0. At this time, bit in DMDR is set to 1 to indicate an interrupt extended repeat area overflow on the destinat address is requested. When block transfer mode is used with the ex repeat area function, an interrupt is requested completion of a 1-block size transfer. When se DTE bit in DMDR of the channel for which the has been stopped to 1, the transfer is resume state when the transfer is stopped. When the extended repeat area is not specifie is ignored. 0: Disables an interrupt request for an extende overflow on the destination address 1: Enables an interrupt request for an extende overflow on the destination address
6, 5	—	All 0	R	Reserved These bits are always read as 0 and cannot b modified.

becomes the start address and the end address
area for address addition and subtraction, respe

When an overflow in the extended repeat area
with the DARIE bit set to 1, an interrupt can be
requested. Table 7.3 shows the settings and ar
the extended repeat area.

00101	32 bytes specified as extended repeat area by the lower 5 bits of the addres
00110	64 bytes specified as extended repeat area by the lower 6 bits of the addres
00111	128 bytes specified as extended repeat area by the lower 7 bits of the addre
01000	256 bytes specified as extended repeat area by the lower 8 bits of the addre
01001	512 bytes specified as extended repeat area by the lower 9 bits of the addre
01010	1 kbyte specified as extended repeat area by the lower 10 bits of the addres
01011	2 kbytes specified as extended repeat area by the lower 11 bits of the addre
01100	4 kbytes specified as extended repeat area by the lower 12 bits of the addre
01101	8 kbytes specified as extended repeat area by the lower 13 bits of the addre
01110	16 kbytes specified as extended repeat area by the lower 14 bits of the addr
01111	32 kbytes specified as extended repeat area by the lower 15 bits of the addr
10000	64 kbytes specified as extended repeat area by the lower 16 bits of the addr
10001	128 kbytes specified as extended repeat area by the lower 17 bits of the ad
10010	256 kbytes specified as extended repeat area by the lower 18 bits of the ad
10011	512 kbytes specified as extended repeat area by the lower 19 bits of the ad
10100	1 Mbyte specified as extended repeat area by the lower 20 bits of the addre
10101	2 Mbytes specified as extended repeat area by the lower 21 bits of the addr
10110	4 Mbytes specified as extended repeat area by the lower 22 bits of the addr
10111	8 Mbytes specified as extended repeat area by the lower 23 bits of the addr
11000	16 Mbytes specified as extended repeat area by the lower 24 bits of the ad
11001	32 Mbytes specified as extended repeat area by the lower 25 bits of the ad
11010	64 Mbytes specified as extended repeat area by the lower 26 bits of the ad
11011	128 Mbytes specified as extended repeat area by the lower 27 bits of the ad
111××	Setting prohibited

[Legend]

×: Don't care

7.4 Transfer Modes

Table 7.4 shows the DMAC transfer modes. The transfer modes can be specified to the in
channels.

Table 7.4 Transfer Modes

				Address R
Address Mode	**Transfer mode**	**Activation Source**	**Common Function**	**Source**
Dual address	• Normal transfer • Repeat transfer • Block transfer Repeat or block size = 1 to 65,536 bytes, 1 to 65,536 words, or 1 to 65,536 longwords	• Auto request (activated by CPU) • On-chip module interrupt • External request	• Total transfer size: 1 to 4 Gbytes or not specified • Offset addition • Extended repeat area function	DSAR
Single address	• Instead of specifying the source or destination address registers, data is directly transferred from/to the external device using the $\overline{\text{DACK}}$ pin • The same settings as above are available other than address register setting (e.g., above transfer modes can be specified) • One transfer can be performed in one bus cycle (the types of transfer modes are the same as those of dual address modes)			DSAR/ $\overline{\text{DACK}}$

address is specified in DDAR. A transfer at a time is performed in two bus cycles (wher bus width is less than the data access size or the access address is not aligned with the b the data access size, the number of bus cycles are needed more than two because one bu divided into multiple bus cycles).

In the first bus cycle, data at the transfer source address is read and in the next cycle, the is written to the transfer destination address.

The read and write cycles are not separated. Other bus cycles (bus cycle by other bus ma refresh cycle, and external bus release cycle) are not generated between read and write c

The $\overline{\text{TEND}}$ signal output is enabled or disabled by the TENDE bit in DMDR. The $\overline{\text{TEN}}$ output in two bus cycles. When an idle cycle is inserted before the bus cycle, the $\overline{\text{TEND}}$ also output in the idle cycle. The $\overline{\text{DACK}}$ signal is not output.

Figure 7.2 shows an example of the signal timing in dual address mode and figure 7.3 sh operation in dual address mode.

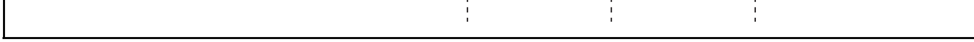


Figure 7.2 Example of Signal Timing in Dual Address Mode

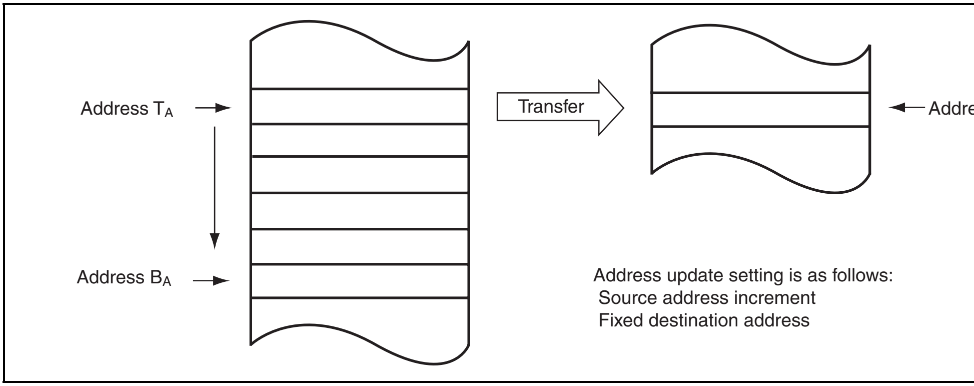


Figure 7.3 Operations in Dual Address Mode

(2) Single Address Mode

In single address mode, data between an external device and an external memory is direct transferred using the $\overline{\text{DACK}}$ pin instead of DSAR or DDAR. A transfer at a time is perfor one bus cycle. In this mode, the data bus width must be the same as the data access size. I details on the data bus width, see section 6, Bus Controller (BSC).

The DMAC accesses an external device as the transfer source or destination by outputting strobe signal ($\overline{\text{DACK}}$) to the external device with $\overline{\text{DACK}}$ and accesses the other transfer t outputting the address. Accordingly, the DMA transfer is performed in one bus cycle. Fig shows an example of a transfer between an external memory and an external device with $\overline{\text{DACK}}$ pin. In this example, the external device outputs data on the data bus and the data to the external memory in the same bus cycle.

also output in the idle cycle.

Figure 7.5 shows an example of timing charts in single address mode and figure 7.6 sho
example of operation in single address mode.

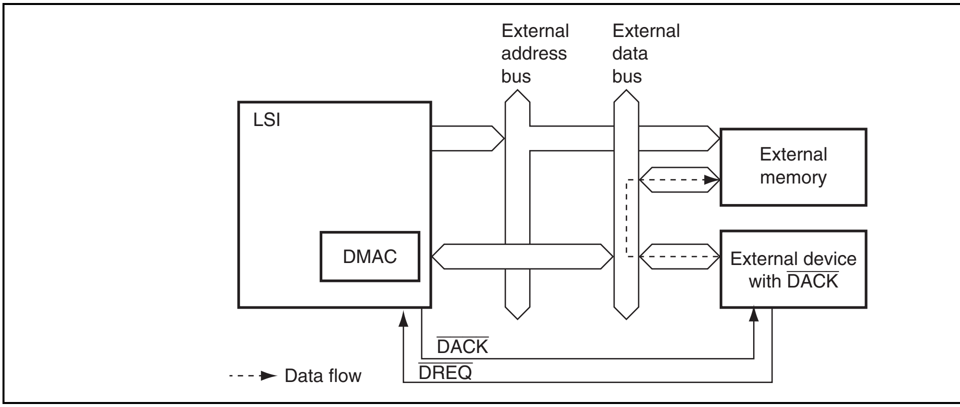


Figure 7.4 Data Flow in Single Address Mode

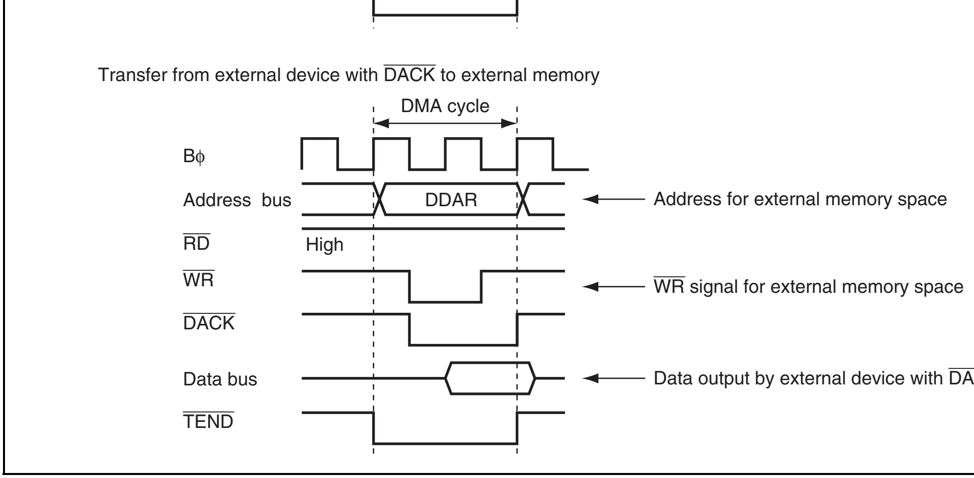


Figure 7.5 Example of Signal Timing in Single Address Mode

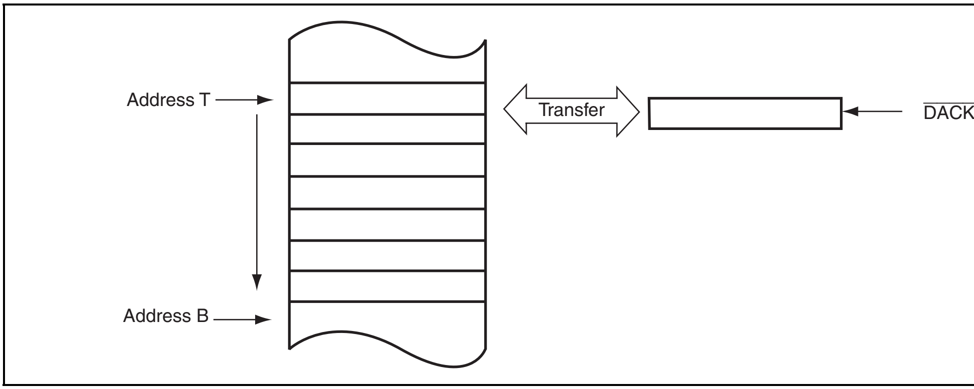


Figure 7.6 Operations in Single Address Mode

Figure 7.7 shows an example of the signal timing in normal transfer mode and figure 7.8 ... the operation in normal transfer mode.

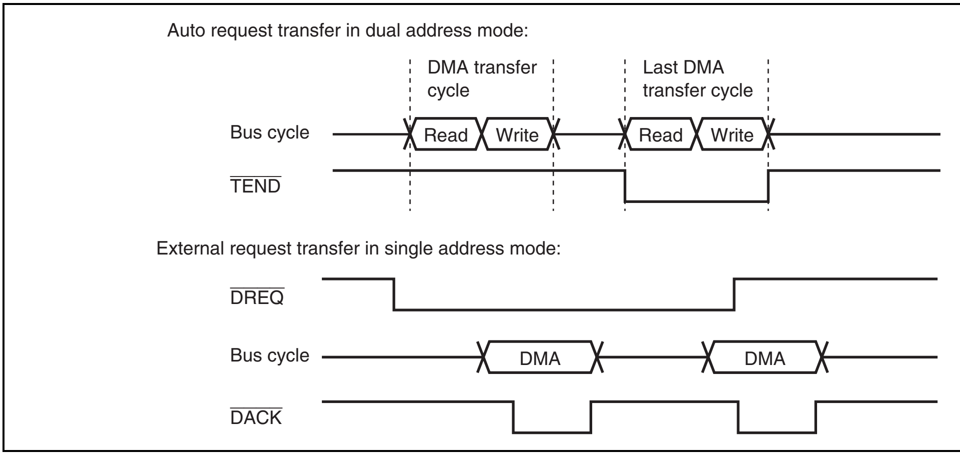


Figure 7.7 Example of Signal Timing in Normal Transfer Mode

Figure 7.8 Operations in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data access size of data is transferred at a single transfer requ
to 4 Gbytes can be specified as a total transfer size by DTCR. The repeat size can be spec
DBSR up to 65536 × data access size.

The repeat area can be specified for the source or destination address side by bits ARS1 a
in DACR. The address specified as the repeat area returns to the transfer start address wh
repeat size of transfers is completed. This operation is repeated until the total transfer size
specified in DTCR is completed. When H'00000000 is specified in DTCR, it is regarded a
free running mode and repeat transfer is continued until the DTE bit in DMDR is cleared

In addition, a DMA transfer can be stopped and a repeat size end interrupt can be requeste
CPU or DTC when the repeat size of transfers is completed. When the next transfer is req
after completion of a 1-repeat size data transfer while the RPTIE bit is set to 1, the DTE b
DMDR is cleared to 0 and the ESIF bit in DMDR is set to 1 to complete the transfer. At t
an interrupt is requested to the CPU or DTC when the ESIE bit in DMDR is set to 1.

The timings of the $\overline{\text{TEND}}$ and $\overline{\text{DACK}}$ signals are the same as in normal transfer mode.

Figure 7.9 shows the operation in repeat transfer mode while dual address mode is set.

When the repeat area is specified as neither source nor destination address side, the opera
the same as the normal transfer mode operation shown in figure 7.8. In this case, a repeat
interrupt can also be requested to the CPU when the repeat size of transfers is completed.

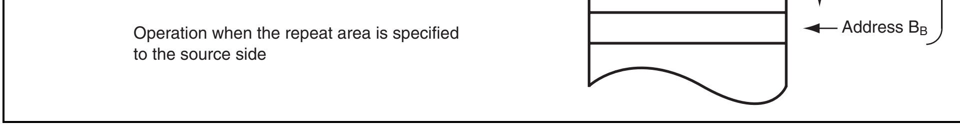


Figure 7.9 Operations in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, one block size of data is transferred at a single transfer request. U
Gbytes can be specified as total transfer size by DTCR. The block size can be specified
up to 65536 × data access size.

While one block of data is being transferred, transfer requests from other channels are su
When the transfer is completed, the bus is released to the other bus master.

The block area can be specified for the source or destination address side by bits ARS1 a
in DACR. The address specified as the block area returns to the transfer start address wh
block size of data is completed. When the block area is specified as neither source nor d
address side, the operation continues without returning the address to the transfer start a
repeat size end interrupt can be requested.

The $\overline{\text{TEND}}$ signal is output every time 1-block data is transferred in the last DMA transf
When the external request is selected as an activation source, the low level detection of t
signal (DREQS = 0) should be selected.

When an interrupt request by an extended repeat area overflow is used in block transfer
settings should be selected carefully. For details, see section 7.5.5, Extended Repeat Are
Function.

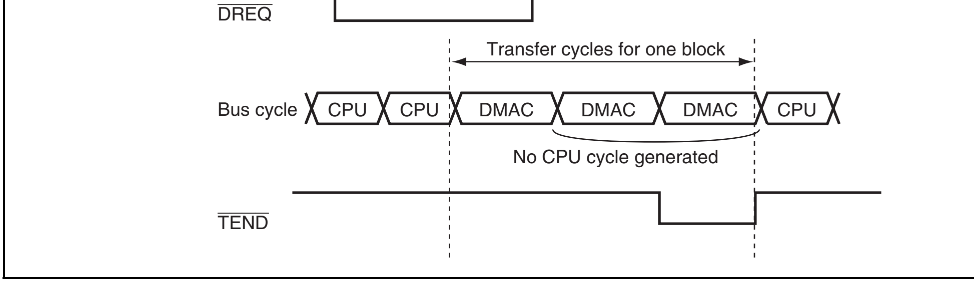


Figure 7.10 Operations in Block Transfer Mode

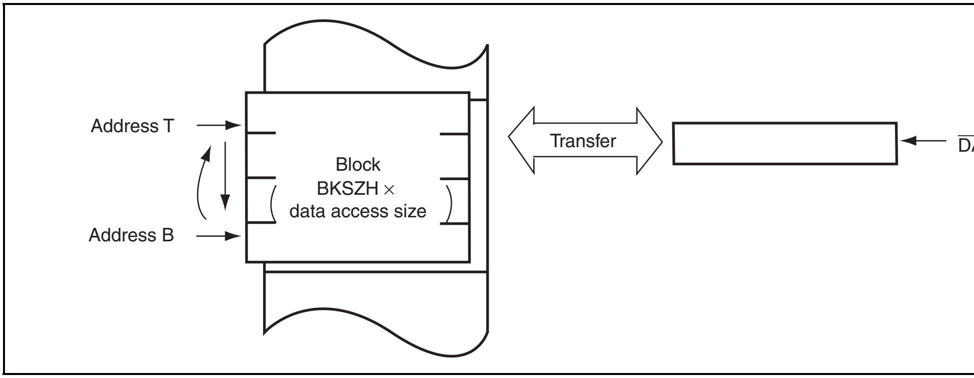


Figure 7.11 Operation in Single Address Mode in Block Transfer Mode (Block Area Specified)

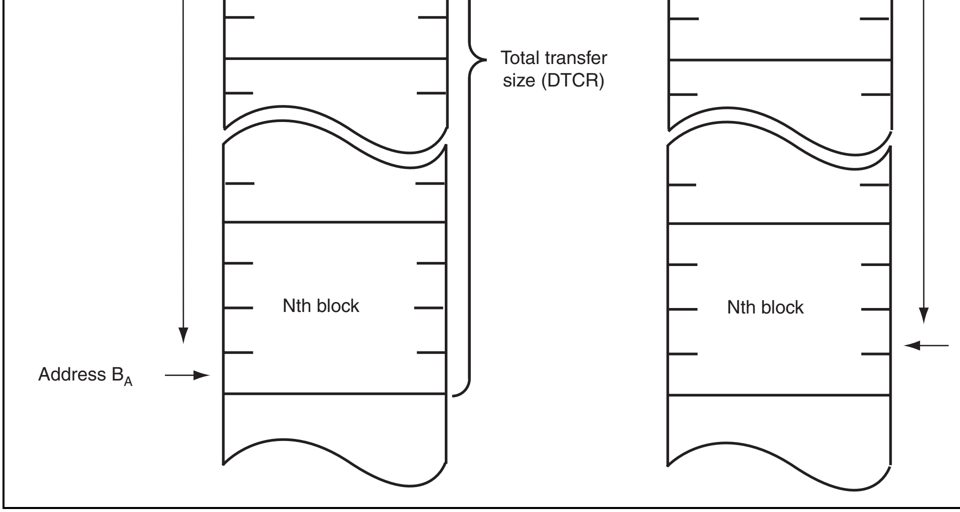


Figure 7.12 Operation in Dual Address Mode in Block Transfer Mode (Block Area Not Specified)

DMDR starts a transfer. The bus mode can be selected from cycle stealing and burst mod

(2) Activation by On-Chip Module Interrupt

An interrupt request from an on-chip peripheral module (on-chip peripheral module interr
used as a transfer request. When a DMA transfer is enabled (DTE = 1), the DMA transfer
started by an on-chip module interrupt.

The activation source of the on-chip module interrupt is selected by the DMA module req
select register (DMRSR). The activation sources are specified to the individual channels.
7.5 is a list of on-chip module interrupts for the DMAC. The interrupt request selected as
activation source can generate an interrupt request simultaneously to the CPU or DTC. Fo
refer to section 5, Interrupt Controller.

The DMAC receives interrupt requests by on-chip peripheral modules independent of the
controller. Therefore, the DMAC is not affected by priority given in the interrupt controll

When the DMAC is activated while DTA = 1, the interrupt request flag is automatically c
a DMA transfer. If multiple channels use a single transfer request as an activation source,
the channel having priority is activated, the interrupt request flag is cleared. In this case, o
channels may not be activated because the transfer request is not held in the DMAC.

When the DMAC is activated while DTA = 0, the interrupt request flag is not cleared by
DMAC and should be cleared by the CPU or DTC transfer.

When an activation source is selected while DTE = 0, the activation source does not requ
transfer to the DMAC. It requests an interrupt to the CPU or DTC.

In addition, make sure that an interrupt request flag as an on-chip module interrupt source
cleared to 0 before writing 1 to the DTE bit.

TGI5A (TGI5A input capture/compare match)	TPU_5	1
RXI0 (receive data full interrupt for SCI channel 0)	SCI_0	1
TXI0 (transmit data empty interrupt for SCI channel 0)	SCI_0	1
RXI1 (receive data full interrupt for SCI channel 1)	SCI_1	1
TXI1 (transmit data empty interrupt for SCI channel 1)	SCI_1	1
RXI2 (receive data full interrupt for SCI channel 2)	SCI_2	1
TXI2 (transmit data empty interrupt for SCI channel 2)	SCI_2	1
RXI4 (receive data full interrupt for SCI channel 4)	SCI_4	1
TXI4 (transmit data empty interrupt for SCI channel 4)	SCI_4	1
RXI5 (receive data full interrupt for SCI channel 5)	SCI_5	2
TXI5 (transmit data empty interrupt for SCI channel 5)	SCI_5	2
RXI6 (receive data full interrupt for SCI channel 6)	SCI_6	2
TXI6 (transmit data empty interrupt for SCI channel 6)	SCI_6	2
USBINTN0 (EP1FIFO full interrupt)	USB	2
USBINTN1 (EP2FIFO empty interrupt)	USB	2

When an external request is selected as an activation source, clear the DDR bit to 0 and s
ICR bit to 1 for the corresponding pin. For details, see section 9, I/O Ports.

7.5.4 Bus Access Modes

There are two types of bus access modes: cycle stealing and burst.

When an activation source is the auto request, the cycle stealing or burst mode is selected
DTF0 in DMDR. When an activation source is the on-chip module interrupt or external r
the cycle stealing mode is selected.

(1) Cycle Stealing Mode

In cycle stealing mode, the DMAC releases the bus every time one unit of transfers (byte
longword, or 1-block size) is completed. After that, when a transfer is requested, the DM
obtains the bus to transfer 1-unit data and then releases the bus on completion of the trans
operation is continued until the transfer end condition is satisfied.

When a transfer is requested to another channel during a DMA transfer, the DMAC relea
bus and then transfers data for the requested channel. For details on operations when a tra
requested to multiple channels, see section 7.5.8, Priority of Channels.

Bus released temporarily for the CPU

Figure 7.13 Example of Timing in Cycle Stealing Mode

(2) Burst Access Mode

In burst mode, once it takes the bus, the DMAC continues a transfer without releasing th
the transfer end condition is satisfied. Even if a transfer is requested from another chann
priority, the transfer is not stopped once it is started. The DMAC releases the bus in the
after the transfer for the channel in burst mode is completed. This is similarly to operatio
stealing mode. However, setting the IBCCS bit in IBCR of the bus controller makes the
release the bus to pass the bus to another bus master.

In block transfer mode, the burst mode setting is ignored (operation is the same as that in
mode during one block of transfers). The DMAC is always operated in cycle stealing mo

Clearing the DTE bit in DMDR stops a DMA transfer. A transfer requested before the D
cleared to 0 by the DMAC is executed. When an interrupt by a transfer size error, a repe
end, or an extended repeat area overflow occurs, the DTE bit is cleared to 0 and the tran

Figure 7.14 shows an example of timing in burst mode.

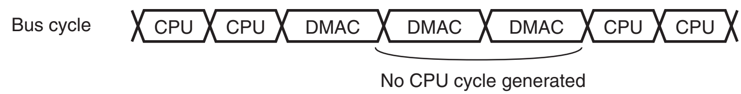


Figure 7.14 Example of Timing in Burst Mode

The extended repeat area on the source address is specified by bits SARA4 to SARA0 in
The extended repeat area on the destination address is specified by bits DARA4 to DARA
DACR. The extended repeat area sizes for each side can be specified independently.

A DMA transfer is stopped and an interrupt by an extended repeat area overflow can be r
to the CPU when the contents of the address register reach the end address of the extende
area. When an overflow on the extended repeat area set in DSAR occurs while the SARIE
DACR is set to 1, the ESIF bit in DMDR is set to 1 and the DTE bit in DMDR is cleared
stop the transfer. At this time, if the ESIE bit in DMDR is set to 1, an interrupt by an exte
repeat area overflow is requested to the CPU. When the DARIE bit in DACR is set to 1, a
overflow on the extended repeat area set in DDAR occurs, meaning that the destination si
target. During the interrupt handling, setting the DTE bit in DMDR resumes the transfer.

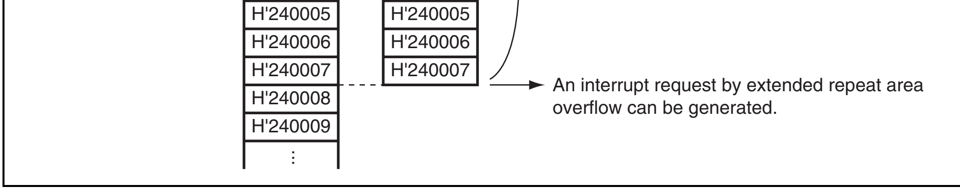


Figure 7.15 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, th following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the addr register must be set so that the block size is a power of 2 or the block size boundary is al the extended repeat area boundary. When an overflow on the extended repeat area occur transfer of one block, the interrupt by the overflow is suspended and the transfer overrur

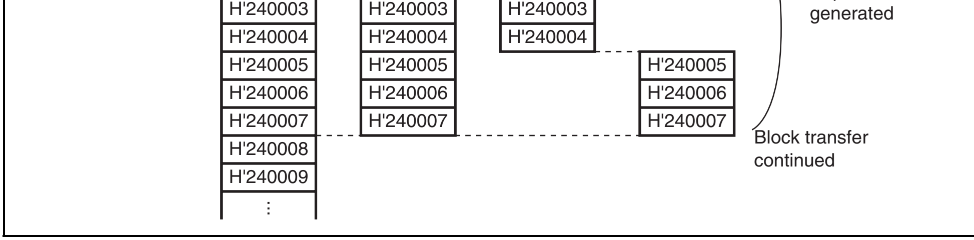


Figure 7.16 Example of Extended Repeat Area Function in Block Transfer M

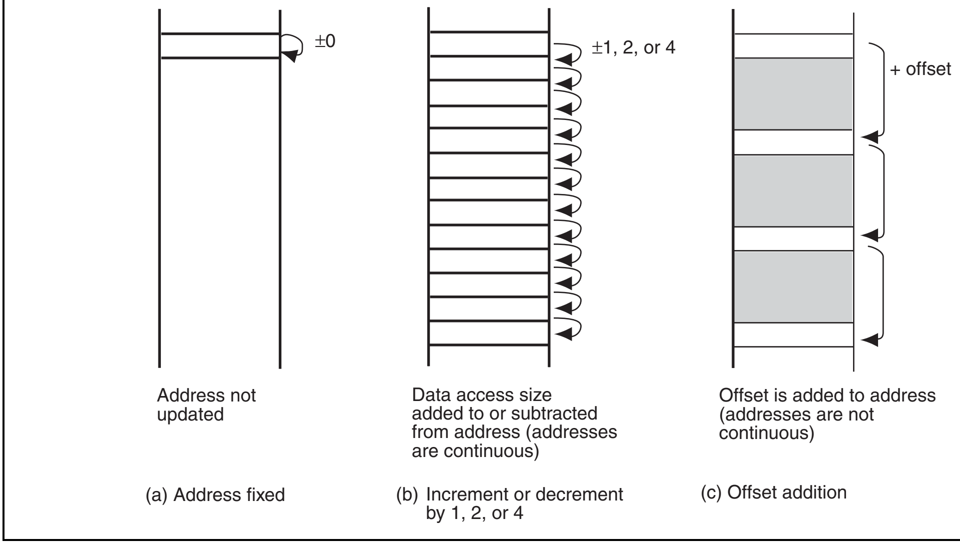


Figure 7.17 Address Update Method

In item (a), Address fixed, the transfer source or destination address is not updated indic
same address.

In item (b), Increment or decrement by 1, 2, or 4, the transfer source or destination addre
incremented or decremented by the value according to the data access size at each transf
word, or longword can be specified as the data access size. The value of 1 for byte, 2 for
4 for longword is used for updating the address. This operation realizes the data transfer
consecutive areas.

In item (c), Offset addition, the address update does not depend on the data access size.
specified by DOFR is added to the address every time the DMAC transfers data of the d
size.

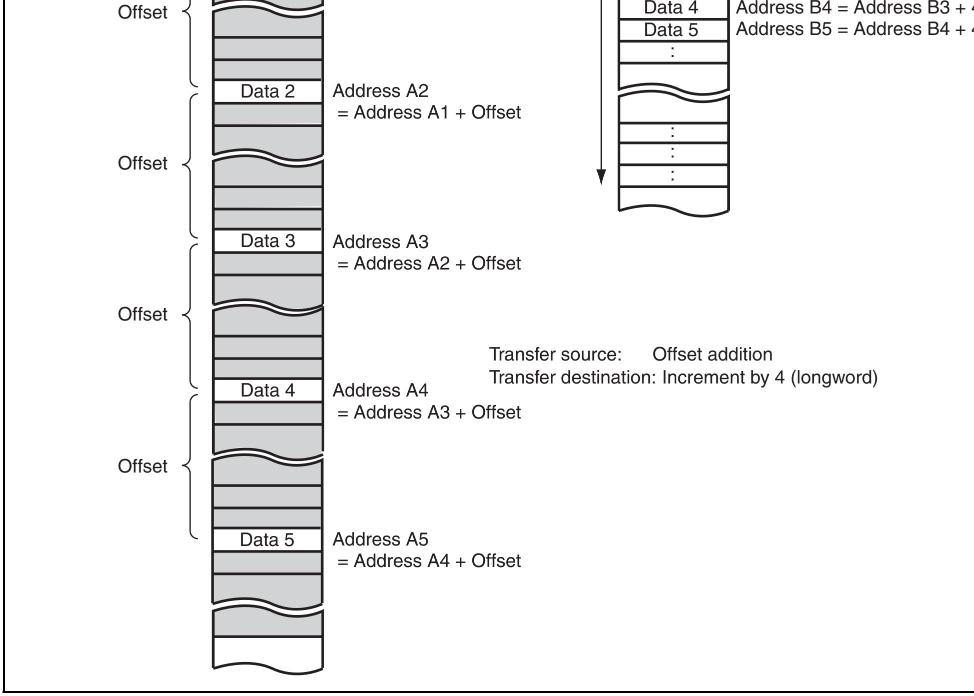


Figure 7.18 Operation of Offset Addition

In figure 7.18, the offset addition is selected as the transfer source address update and inc decrement by 1, 2, or 4 is selected as the transfer destination address. The address update that data at the address which is away from the previous transfer source address by the of read from. The data read from the address away from the previous address is written to th consecutive area in the destination side.

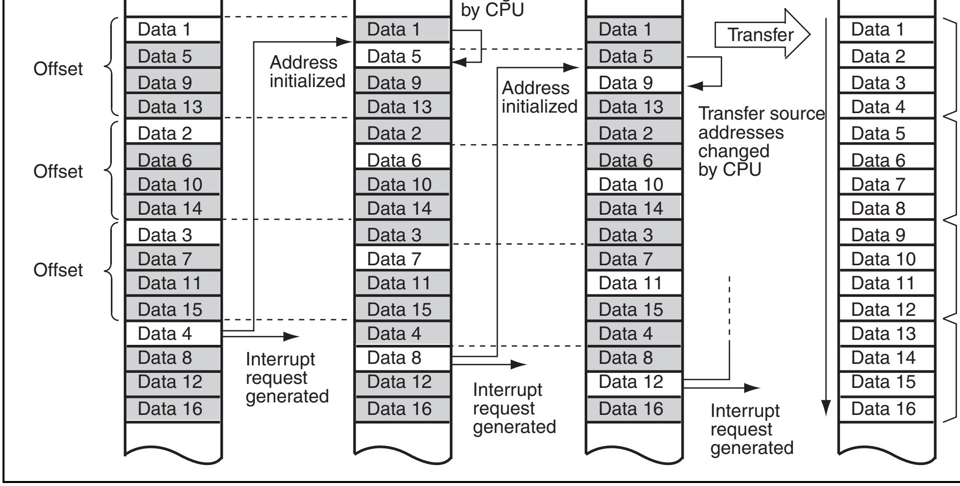


Figure 7.19 XY Conversion Operation Using Offset Addition in Repeat Transfe

In figure 7.19, the source address side is specified to the repeat area by DACR and the o addition is selected. The offset value is set to 4 × data access size (when the data access longword, H'00000010 is set in DOFR, as an example). The repeat size is set to 4 × data size (when the data access size is longword, the repeat size is set to 4 × 4 = 16 bytes, as example). The increment or decrement by 1, 2, or 4 is specified as the transfer destinatio A repeat size end interrupt is requested when the repeat size of transfers is completed.

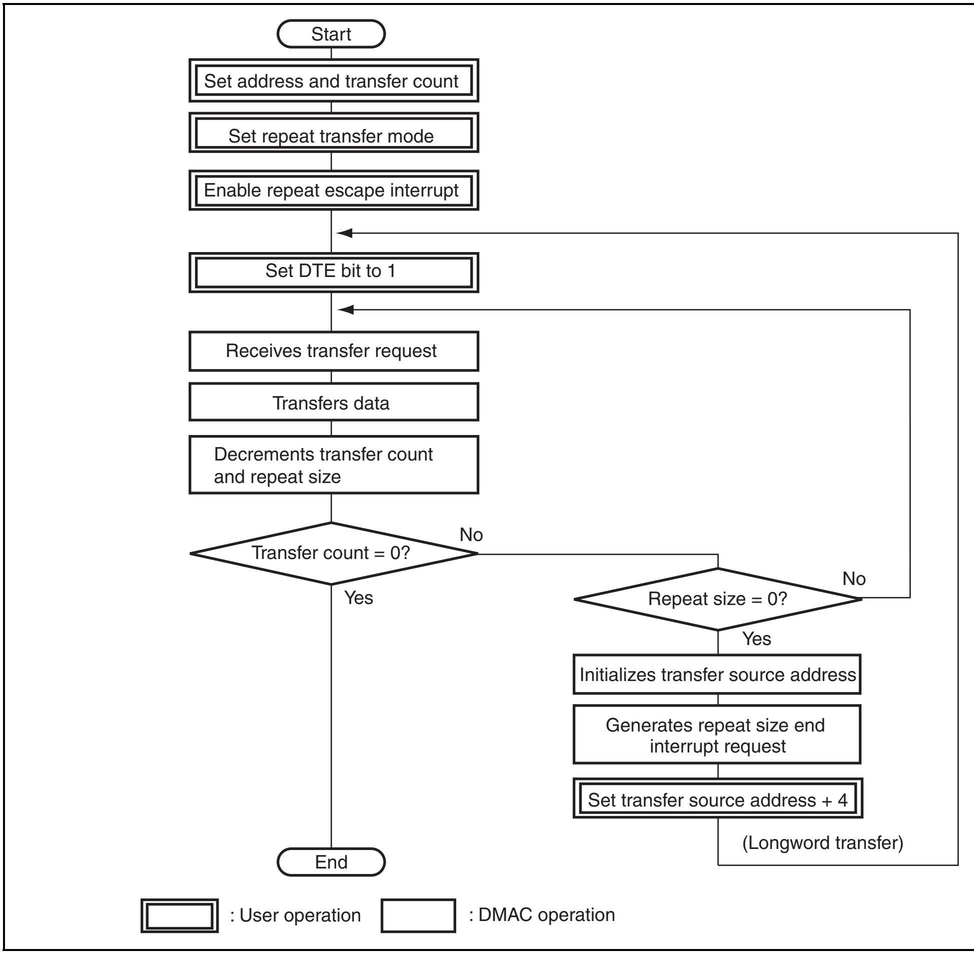


Figure 7.20 XY Conversion Flowchart Using Offset Addition in Repeat Transfer

7.5.7 Register during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs ac
the other settings and transfer state. The registers to be updated are DSAR, DDAR, DTC
BKSZH and BKSZ in DBSR, and the DTE, ACT, ERRF, ESIF, and DTIF bits in DMDI

(1) DMA Source Address Register

When the transfer source address set in DSAR is accessed, the contents of DSAR are ou
then are updated to the next address.

The increment or decrement can be specified by bits SAT1 and SAT0 in DACR. When S
SAT0 = B'00, the address is fixed. When SAT1 and SAT0 = B'01, the address is added w
offset. When SAT1 and SAT0 = B'10, the address is incremented. When SAT1 and SAT
the address is decremented. The size of increment or decrement depends on the data acc

The data access size is specified by bits DTSZ1 and DTSZ0 in DMDR. When DTSZ1 an
= B'00, the data access size is byte and the address is incremented or decremented by 1.
DTSZ1 and DTSZ0 = B'01, the data access size is word and the address is incremented o
decremented by 2. When DTSZ1 and DTSZ0 = B'10, the data access size is longword an
address is incremented or decremented by 4. Even if the access data size of the source ad
word or longword, when the source address is not aligned with the word or longword bo
the read bus cycle is divided into byte or word cycles. While data of one word or one lon
being read, the size of increment or decrement is changing according to the actual data a
for example, +1 or +2 for byte or word data. After one word or one longword of data is r
address when the read cycle is started is incremented or decremented by the value accor
bits SAT1 and SAT0.

(2) DMA Destination Address Register

When the transfer destination address set in DDAR is accessed, the contents of DDAR ar
and then are updated to the next address.

The increment or decrement can be specified by bits DAT1 and DAT0 in DACR. When D
and DAT0 = B'00, the address is fixed. When DAT1 and DAT0 = B'01, the address is ad
the offset. When DAT1 and DAT0 = B'10, the address is incremented. When DAT1 and D
B'11, the address is decremented. The incrementing or decrementing size depends on the
access size.

The data access size is specified by bits DTSZ1 and DTSZ0 in DMDR. When DTSZ1 an
= B'00, the data access size is byte and the address is incremented or decremented by 1. W
DTSZ1 and DTSZ0 = B'01, the data access size is word and the address is incremented o
decremented by 2. When DTSZ1 and DTSZ0 = B'10, the data access size is longword an
address is incremented or decremented by 4. Even if the access data size of the destinatio
is word or longword, when the destination address is not aligned with the word or longwo
boundary, the write bus cycle is divided into byte and word cycles. While one word or on
longword of data is being written, the incrementing or decrementing size is changing acc
the actual data access size, for example, +1 or +2 for byte or word data. After the one wo
longword of data is written, the address when the write cycle is started is incremented or
decremented by the value according to bits SAT1 and SAT0.

In block or repeat transfer mode, when the block or repeat size of data transfers is comple
the block or repeat area is specified to the destination address side, the destination addres
to the transfer start address and is not affected by the address update.

When the extended repeat area is specified to the destination address side, operation follo
setting. The upper address bits are fixed and is not affected by the address update.

While data is being transferred, all the bits of DTCR may be changed. DTCR must be ac longwords. If the upper word and lower word are read separately, incorrect data may be since the contents of DTCR during the transfer may be updated regardless of the access CPU. Moreover, DTCR for the channel being transferred must not be written to.

When a conflict occurs between the address update by DMA transfer and write access by the CPU has priority. When a conflict occurs between change from 1, 2, or 4 to 0 in DTC write access by the CPU (other than 0), the CPU has priority in writing to DTCR. Howe transfer is stopped.

(4) DMA Block Size Register (DBSR)

DBSR is enabled in block or repeat transfer mode. Bits 31 to 16 in DBSR function as BI bits 15 to 0 in DBSR function as BKSZ. The BKSZH bits (16 bits) store the block size a size and its value is not changed. The BKSZ bits (16 bits) function as a counter for the b and repeat size and its value is decremented every transfer by 1. When the BKSZ value change from 1 to 0 by a DMA transfer, 0 is not stored but the BKSZH value is loaded in BKSZ bits.

Since the upper 16 bits of DBSR are not updated, DBSR can be accessed in words.

DBSR for the channel being transferred must not be written to.

- When a transfer is stopped by an NMI interrupt
- When a transfer is stopped by and address error
- Reset state
- Hardware standby mode
- When a transfer is stopped by writing 0 to the DTE bit

Writing to the registers for the channels when the corresponding DTE bit is set to 1 is pro (except for the DTE bit). When changing the register settings after writing 0 to the DTE b confirm that the DTE bit has been cleared to 0.

Figure 7.21 show the procedure for changing the register settings for the channel being transferred.

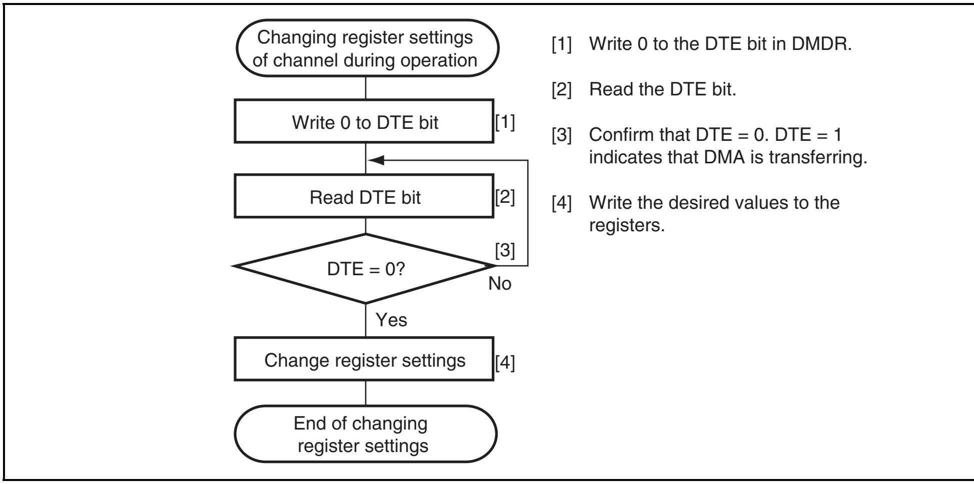


Figure 7.21 Procedure for Changing Register Setting For Channel being Transf

In burst mode, up to three times of DMA transfer are performed from the cycle in which
bit is written to 0. The ACT bit retains 1 from writing 0 to the DTE bit to completion of
transfer.

(7) ERRF Bit in DMDR

When an address error or an NMI interrupt occur, the DMAC clears the DTE bits for all
channels to stop a transfer. In addition, it sets the ERRF bit in DMDR_0 to 1 to indicate
address error or an NMI interrupt has occurred regardless of whether or not the DMAC
operation.

(8) ESIF Bit in DMDR

When an interrupt by an transfer size error, a repeat size end, or an extended repeat area
is requested, the ESIF bit in DMDR is set to 1. When both the ESIF and ESIE bits are se
transfer escape interrupt is requested to the CPU or DTC.

The ESIF bit is set to 1 when the ACT bit in DMDR is cleared to 0 to stop a transfer afte
cycle of the interrupt source is completed.

The ESIF bit is automatically cleared to 0 and a transfer request is cleared if the transfer
resumed by setting the DTE bit to 1 during interrupt handling.

For details on interrupts, see section 7.8, Interrupt Sources.

For details on interrupts, see section 7.8, Interrupt Sources.

7.5.8 Priority of Channels

The channels of the DMAC are given following priority levels: channel 0 > channel 1 > channel 2 > channel3. Table 7.6 shows the priority levels among the DMAC channels.

Table 7.6 Priority among DMAC Channels

Channel	Prio
Channel 0	High
Channel 1	↑
Channel 2	
Channel 3	Low

The channel having highest priority other than the channel being transferred is selected w transfer is requested from other channels. The selected channel starts the transfer after the being transferred releases the bus. At this time, when a bus master other than the DMAC the bus, the cycle for the bus master is inserted.

In a burst transfer or a block transfer, channels are not switched.

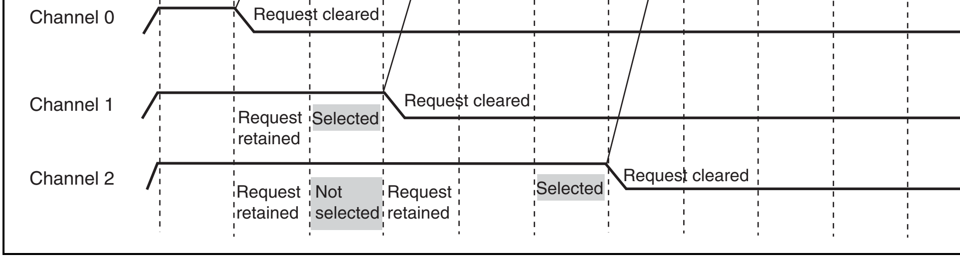


Figure 7.22 Example of Timing for Channel Priority

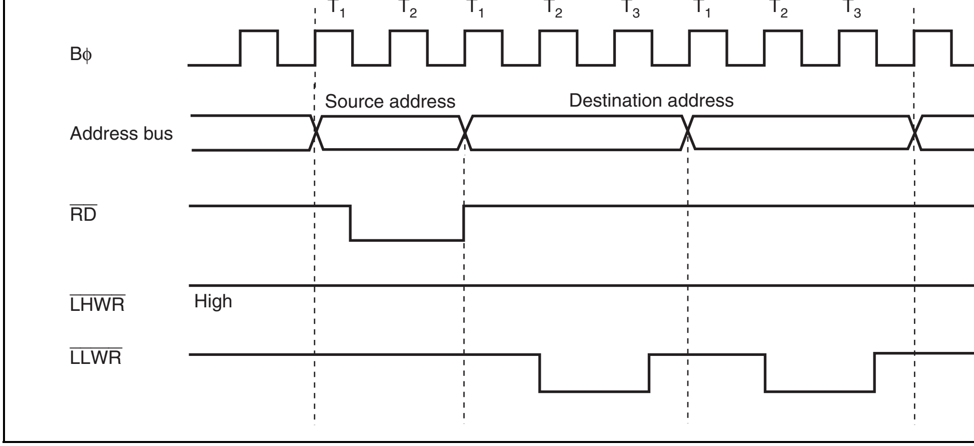


Figure 7.23 Example of Bus Timing of DMA Transfer

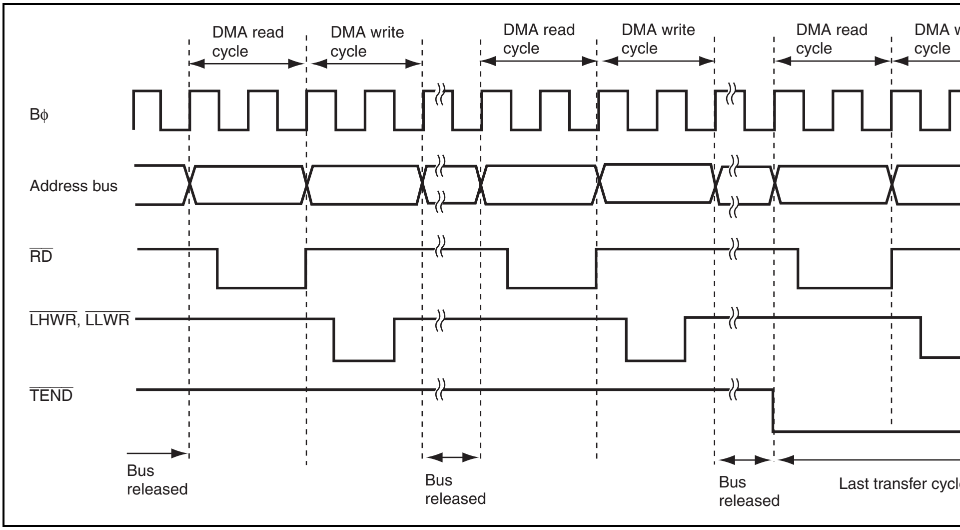


Figure 7.24 Example of Transfer in Normal Transfer Mode by Cycle Steal

In figures 7.25 and 7.26, the $\overline{\text{TEND}}$ signal output is enabled and data is transferred in lo from the external 16-bit 2-state access space to the 16-bit 2-state access space in normal mode by cycle stealing.

In figure 7.25, the transfer source (DSAR) is not aligned with a longword boundary and transfer destination (DDAR) is aligned with a longword boundary.

In figure 7.26, the transfer source (DSAR) is aligned with a longword boundary and the destination (DDAR) is not aligned with a longword boundary.

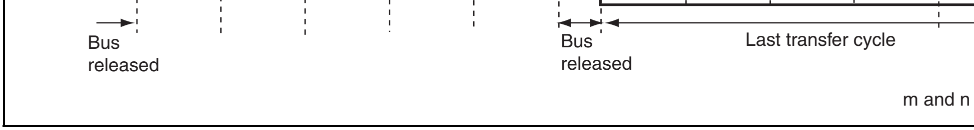


Figure 7.25 Example of Transfer in Normal Transfer Mode by Cycle Stealin
(Transfer Source DSAR = Odd Address and Source Address Increment)

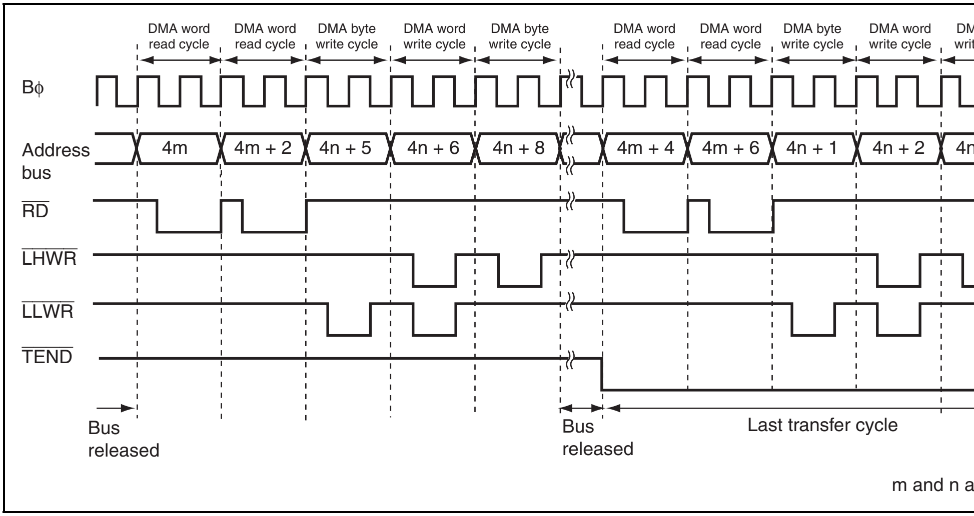


Figure 7.26 Example of Transfer in Normal Transfer Mode by Cycle Stealin
(Transfer Destination DDAR = Odd Address and Destination Address Decrem

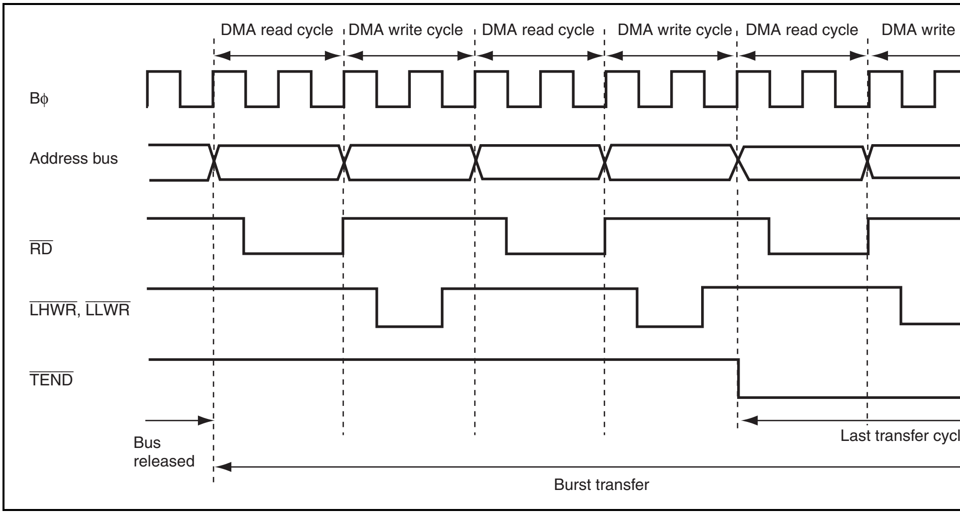


Figure 7.27 Example of Transfer in Normal Transfer Mode by Burst Acce

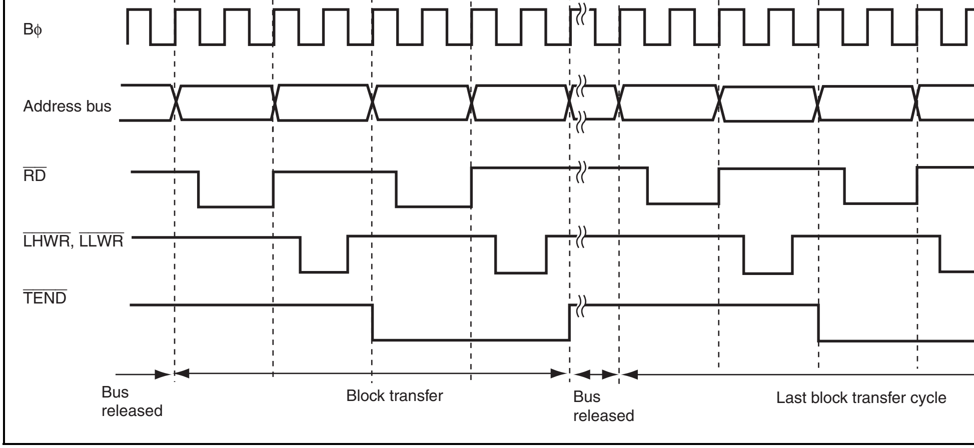
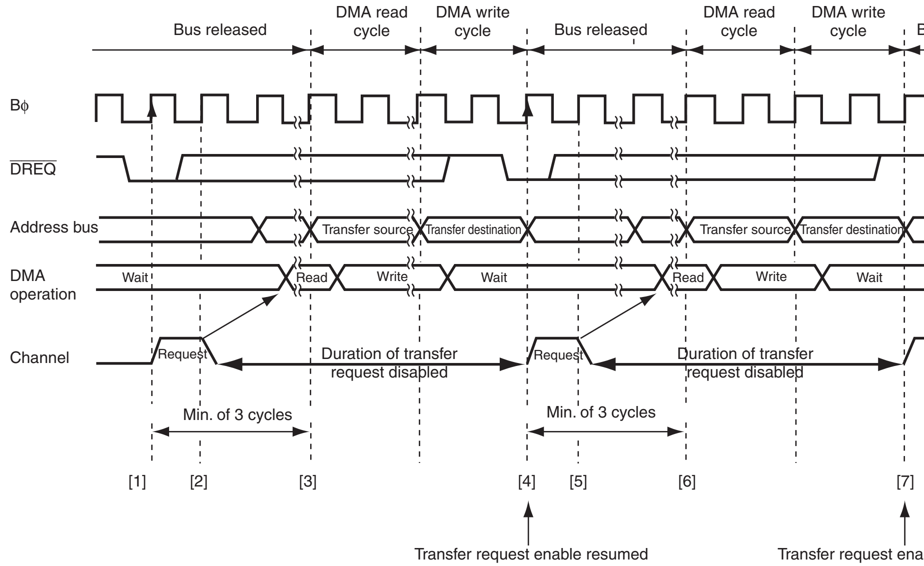


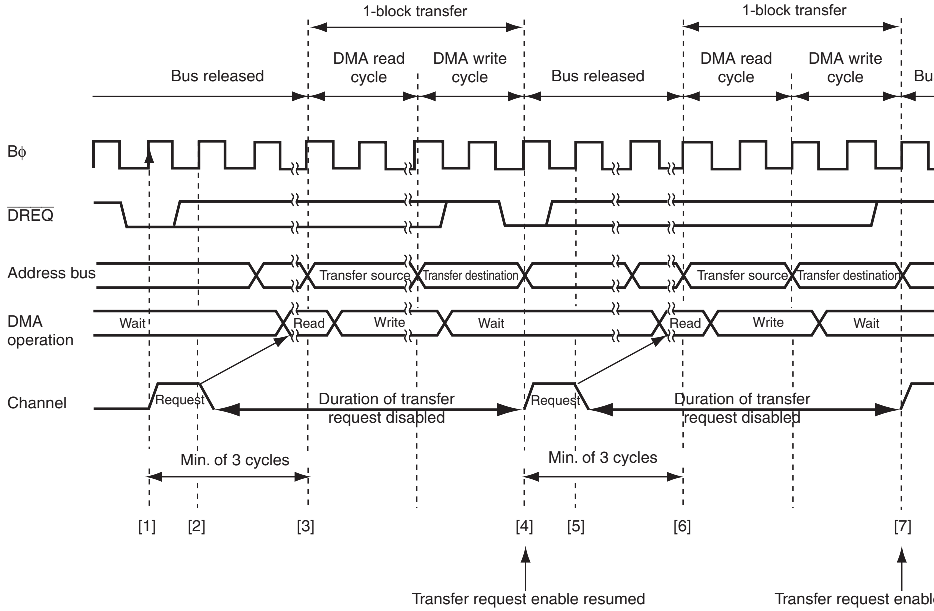
Figure 7.28 Example of Transfer in Block Transfer Mode

receiving the next transfer request resumes and then a low level of the DREQ signal is d
This operation is repeated until the transfer is completed.



[1] After DMA transfer request is enabled, a low level of the DREQ signal is detected at the rising edge of the Bφ signal and request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started and sampling the DREQ signal at the rising edge of the Bφ signal is started to detect a high leve DREQ signal.
[4][7] When a high level of the DREQ signal has been detected, transfer request enable is resumed after completion of the wr (A low level of the DREQ signal is detected at the rising edge of the Bφ signal and a transfer request is held. This is the s

Figure 7.29 Example of Transfer in Normal Transfer Mode Activated by DREQ Falling Edge



[1] After DMA transfer request is enabled, a low level of the $\overline{DREQ}$ signal is detected at the rising edge of the Bφ signal and a request is held.

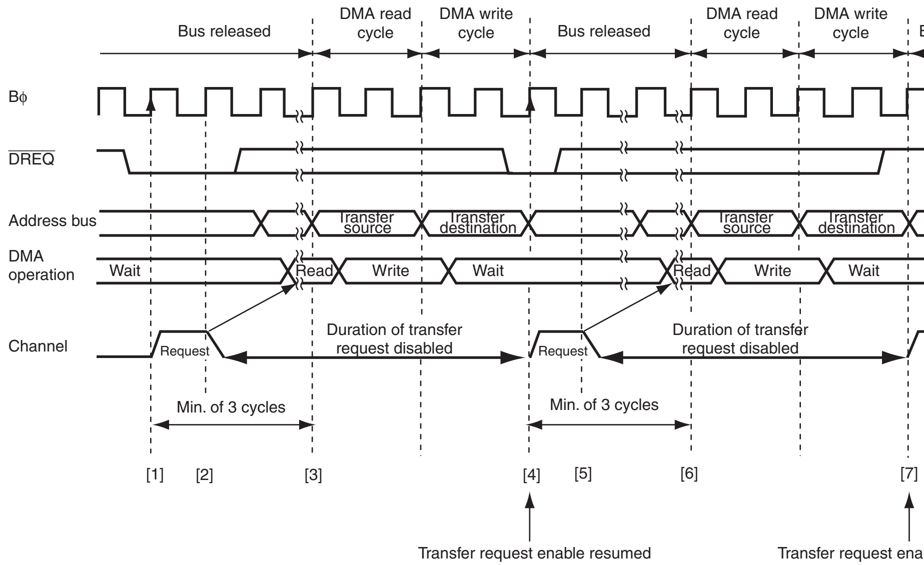
[2][5] The DMAC is activated and the transfer request is cleared.

[3][6] A DMA cycle is started and sampling the $\overline{DREQ}$ signal at the rising edge of the Bφ signal is started to detect a high level o $\overline{DREQ}$ signal.

[4][7] When a high level of the $\overline{DREQ}$ signal has been detected, transfer request enable is resumed after completion of the write (A low level of the $\overline{DREQ}$ signal is detected at the rising edge of the Bφ signal and a transfer request is held. This is the sa

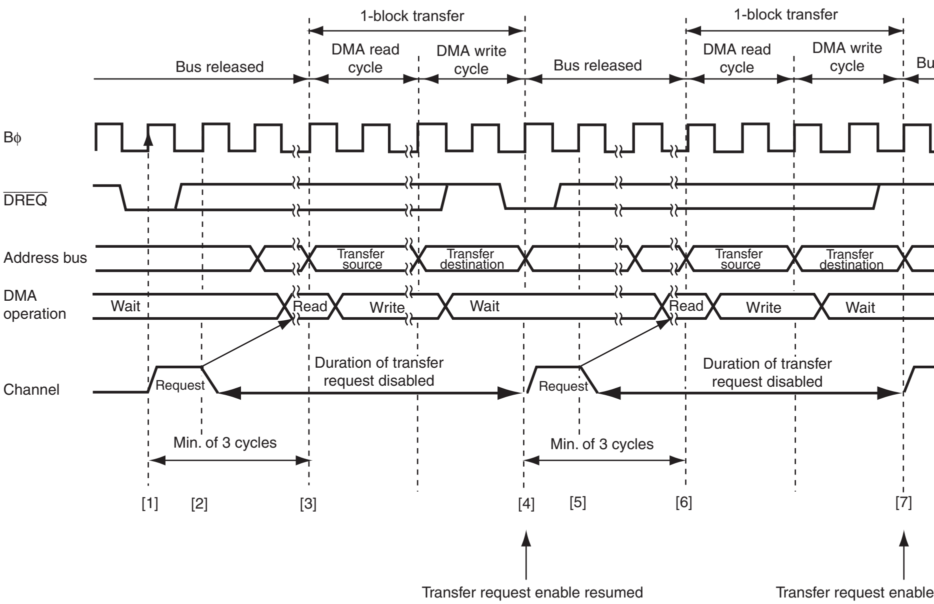
Figure 7.30 Example of Transfer in Block Transfer Mode Activated by $\overline{DREQ}$ Falling Edge

completed.



[1] After DMA transfer request is enabled, a low level of the $\overline{DREQ}$ signal is detected at the rising edge of the Bφ signal anc request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started.
[4][7] Transfer request enable is resumed after completion of the write cycle.
(A low level of the $\overline{DREQ}$ signal is detected at the rising edge of the Bφ signal and a transfer request is held. This is the

Figure 7.31 Example of Transfer in Normal Transfer Mode Activated by $\overline{DREQ}$ Low Level



[1] After DMA transfer request is enabled, a low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bφ signal and a transf request is held.

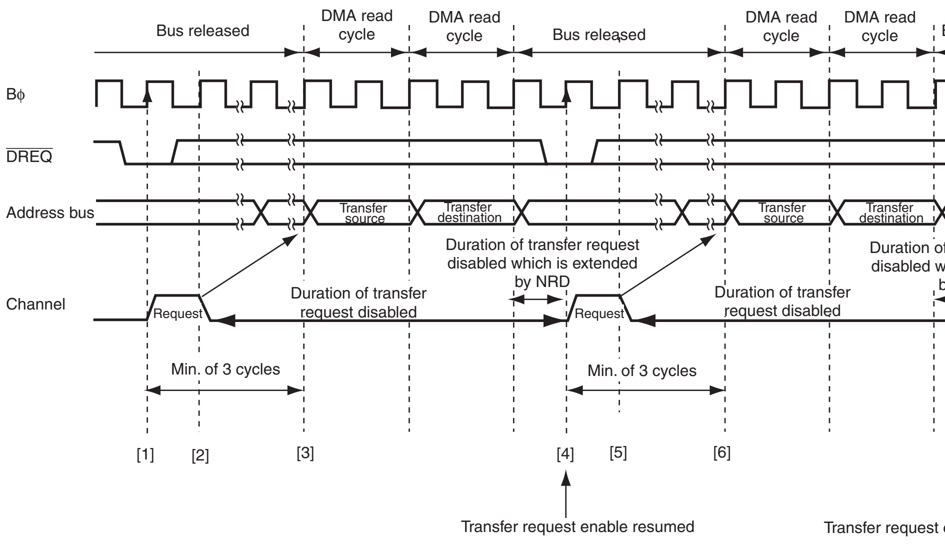
[2][5] The DMAC is activated and the transfer request is cleared.

[3][6] A DMA cycle is started.

[4][7] Transfer request enable is resumed after completion of the write cycle. (A low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bφ signal and a transfer request is held. This is the same as

Figure 7.32 Example of Transfer in Block Transfer Mode Activated by $\overline{\text{DREQ}}$ Low Level

enabled, a transfer request is held in the DMAC. When the DMAC is activated, the trans request is cleared. Receiving the next transfer request resumes after completion of the w and then a low level of the $\overline{DREQ}$ signal is detected. This operation is repeated until the completed.



[1] After DMA transfer request is enabled, a low level of the $\overline{DREQ}$ signal is detected at the rising edge of the Bφ signal and request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started.
[4][7] Transfer request enable is resumed one cycle after completion of the write cycle.
(A low level of the $\overline{DREQ}$ signal is detected at the rising edge of the Bφ signal and a transfer request is held. This is the s

Figure 7.33 Example of Transfer in Normal Transfer Mode Activated by $\overline{DREQ}$ Low Level with NRD = 1

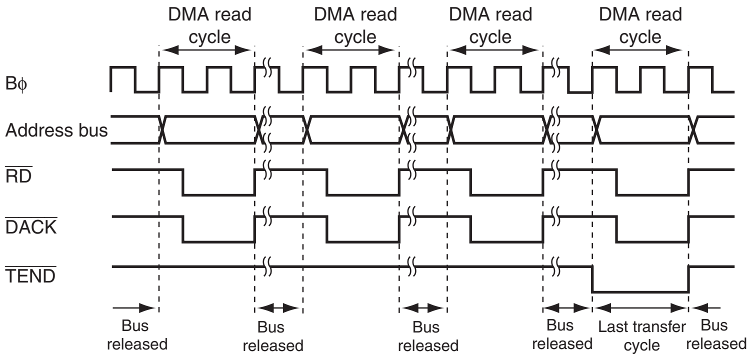


Figure 7.34 Example of Transfer in Single Address Mode (Byte Read)

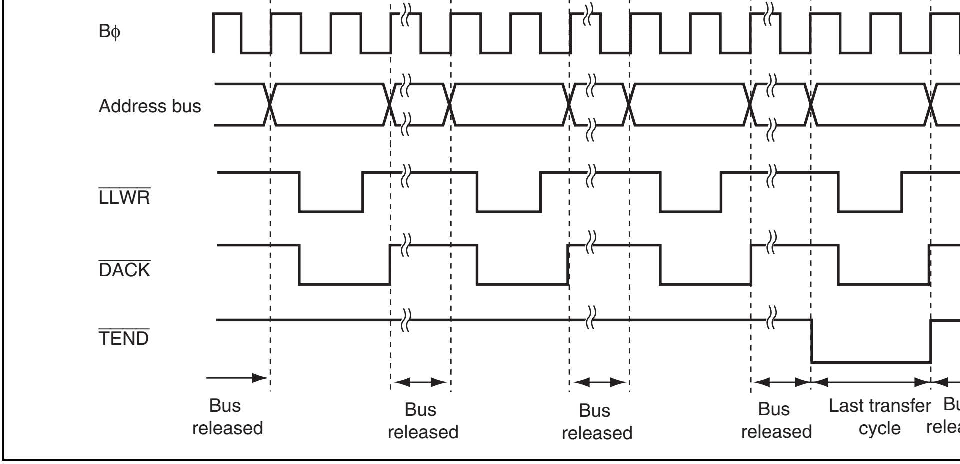
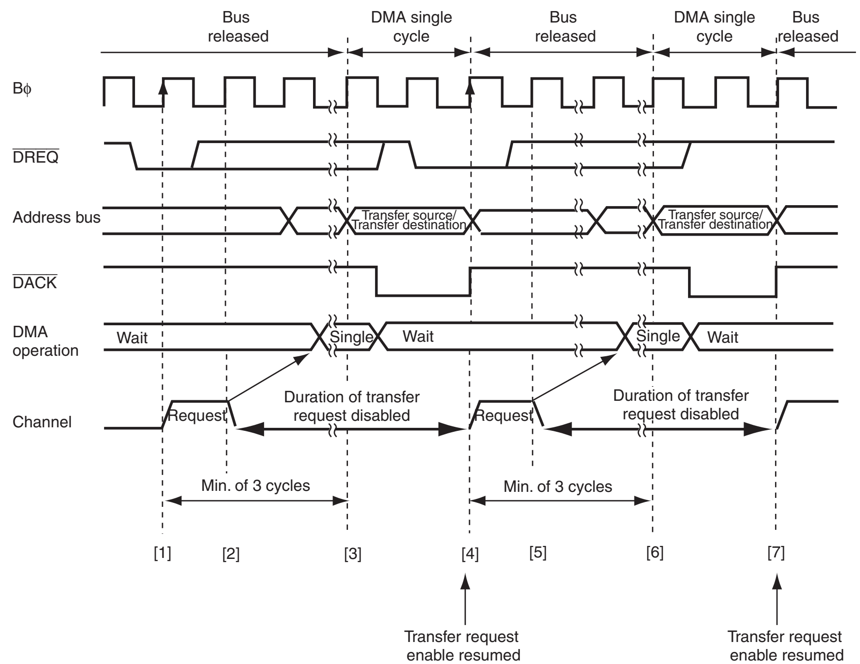


Figure 7.35 Example of Transfer in Single Address Mode (Byte Write)

operation is repeated until the transfer is completed.



[1] After DMA transfer request is enabled, a low level of the $\overline{DREQ}$ signal is detected at the rising edge of the Bφ signal and a tr request is held.

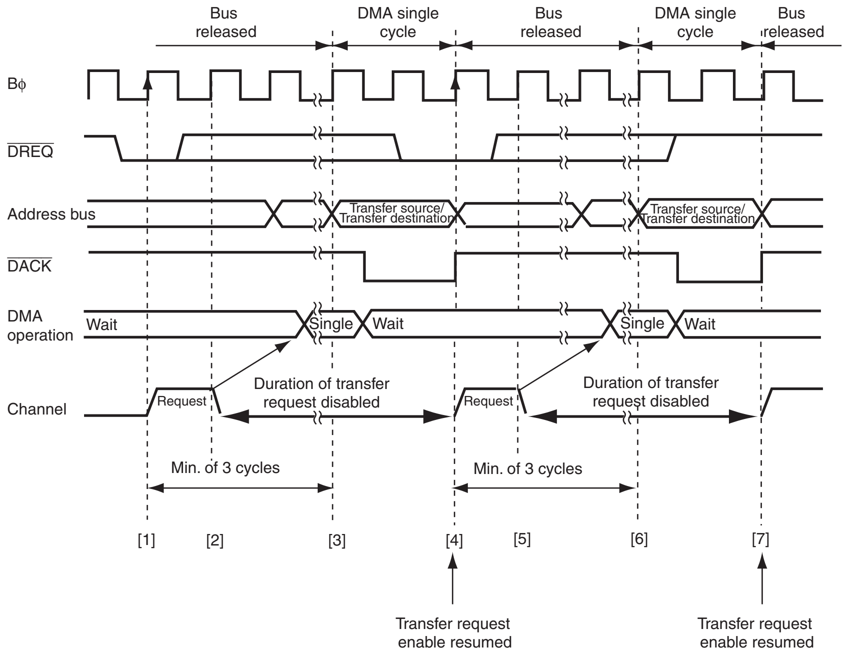
[2][5] The DMAC is activated and the transfer request is cleared.

[3][6] A DMA cycle is started and sampling the $\overline{DREQ}$ signal at the rising edge of the Bφ signal is started to detect a high level of t $\overline{DREQ}$ signal.

[4][7] When a high level of the $\overline{DREQ}$ signal has been detected, transfer enable is resumed after completion of the write cycle. (A low level of the $\overline{DREQ}$ signal is detected at the rising edge of the Bφ signal and a transfer request is held. This is the sam

Figure 7.36 Example of Transfer in Single Address Mode Activated by $\overline{DREQ}$ Falling Edge

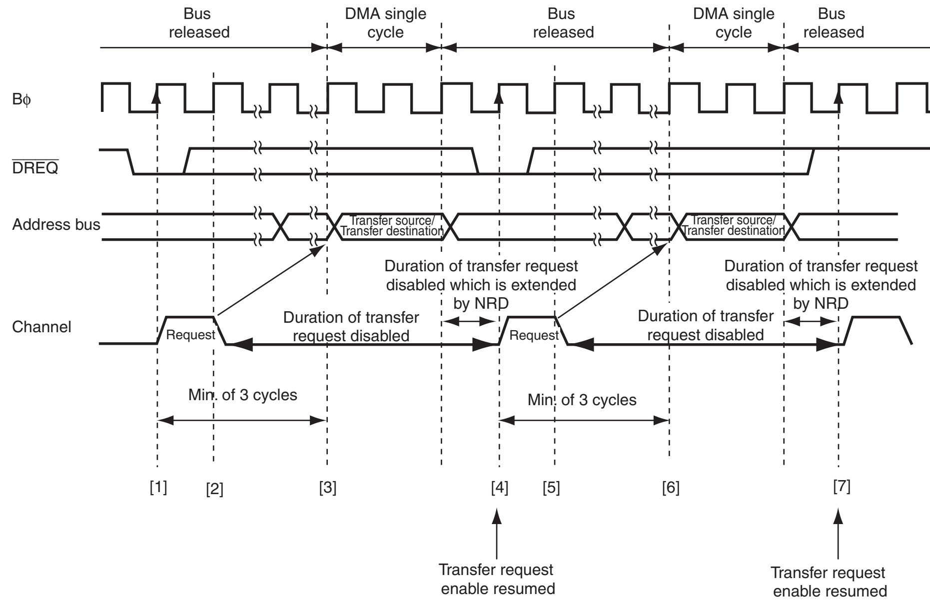
completed.



[1] After DMA transfer request is enabled, a low level of the DREQ signal is detected at the rising edge of the Bφ signal and a request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started.
[4][7] Transfer request enable is resumed after completion of the single cycle.
(A low level of the DREQ signal is detected at the rising edge of the Bφ signal and a transfer request is held. This is the sa

Figure 7.37 Example of Transfer in Single Address Mode Activated by DREQ Low Level

enabled, a transfer request is held in the DMAC. When the DMAC is activated, the transf request is cleared. Receiving the next transfer request resumes after one cycle of the trans request duration inserted by NRD = 1 on completion of the single cycle and then a low le $\overline{DREQ}$ signal is detected. This operation is repeated until the transfer is completed.



[1] After DMA transfer request is enabled, a low level of the $\overline{DREQ}$ signal is detected at the rising edge of the Bφ signal and a t request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started.
[4][7] Transfer request enable is resumed one cycle after completion of the single cycle.
(A low level of the $\overline{DREQ}$ signal is detected at the rising edge of the Bφ signal and a transfer request is held. This is the sar

Figure 7.38 Example of Transfer in Single Address Mode Activated by $\overline{DREQ}$ Low Level with NRD = 1

(2) Transfer End by Transfer Size Error Interrupt

When the following conditions are satisfied while the TSEIE bit in DMDR is set to 1, a
size error occurs and a DMA transfer is terminated. At this time, the DTE bit in DMR is
0 and the ESIF bit in DMDR is set to 1.

- In normal transfer mode and repeat transfer mode, when the next transfer is requeste
  transfer is disabled due to the DTCR value less than the data access size
- In block transfer mode, when the next transfer is requested while a transfer is disable
  the DTCR value less than the block size

When the TSEIE bit in DMDR is cleared to 0, data is transferred until the DTCR value r
A transfer size error is not generated. Operation in each transfer mode is shown below.

- In normal transfer mode and repeat transfer mode, when the DTCR value is less than
  access size, data is transferred in bytes
- In block transfer mode, when the DTCR value is less than the block size, the specifie
  data in DTCR is transferred instead of transferring the block size of data. The transfe
  performed in bytes.

When an overflow on the extended repeat area occurs while the extended repeat area is sp
and the SARIE or DARIE bit in DACR is set to 1, an interrupt by an extended repeat area
overflow is requested. When the interrupt is requested, the DMA transfer is terminated, th
bit in DMDR is cleared to 0, and the ESIF bit in DMDR is set to 1.

In dual address mode, even if an interrupt by an extended repeat area overflow occurs dur
read cycle, the following write cycle is performed.

In block transfer mode, even if an interrupt by an extended repeat area overflow occurs d
block transfer, the remaining data is transferred. The transfer is not terminated by an exte
repeat area overflow interrupt unless the current transfer is complete.

(5) Transfer End by Clearing DTE Bit in DMDR

When the DTE bit in DMDR is cleared to 0 by the CPU, a transfer is completed after the
DMA cycle and a DMA cycle in which the transfer request is accepted are completed.

In block transfer mode, a DMA transfer is completed after 1-block data is transferred.

transfer unit.

In single address mode, a DMA transfer is completed after completion of the bus cycle f
transfer unit.

(b) Block Transfer Mode

A DMA transfer is forced to stop. Since a 1-block size of transfers is not completed, ope
not guaranteed.

In dual address mode, the write cycle corresponding to the read cycle is performed. This
to (a) in normal transfer mode.

(7) Transfer End by Address Error

When an address error occurs, the DTE bits for all the channels are cleared to 0 and the
in DMDR_0 is set to 1. When an address error occurs during a DMA transfer, the transf
forced to stop. To perform a DMA transfer after an address error occurs, clear the ERRF
and then set the DTE bits for the channels.

The transfer end timing after an address error is the same as that after an NMI interrupt.

(8) Transfer End by Hardware Standby Mode or Reset

The DMAC is initialized by a reset and a transition to the hardware standby mode. A DM
transfer is not guaranteed.

The priority level of the CPU is specified by bits CPUP2 to CPUP0. The value of bits CP
CPUP0 is updated according to the exception handling priority.

If the CPU priority control is enabled by the CPUPCE bit in CPUPCR, when the CPU ha
over the DMAC, a transfer request for the corresponding channel is masked and the trans
activated. When another channel has priority over or the same as the CPU, a transfer requ
received regardless of the priority between channels and the transfer is activated.

The transfer request masked by the CPU priority control function is suspended. When the
channel is given priority over the CPU by changing priority levels of the CPU or channel
transfer request is received and the transfer is resumed. Writing 0 to the DTE bit clears th
suspended transfer request.

When the CPUPCE bit is cleared to 0, it is regarded as the lowest priority.

a DMA transfer.

In block transfer mode and an auto request transfer by burst access, bus cycles of the DM
transfer are consecutively performed. For this duration, since the DMAC has priority ov
CPU and DTC, accesses to the external space is suspended (the IBCCS bit in the bus co
register 2 (BCR2) is cleared to 0).

When the bus is passed to another channel or an auto request transfer by cycle stealing,
of the DMAC and on-chip bus master are performed alternatively.

When the arbitration function among the DMAC and on-chip bus masters is enabled by
IBCCS bit in BCR2, the bus is used alternatively except the bus cycles which are not se
For details, see section 6, Bus Controller (BSC).

A conflict may occur between external space access of the DMAC and an external bus r
cycle. Even if a burst or block transfer is performed by the DMAC, the transfer is stopp
temporarily and a cycle of external bus release is inserted by the BSC according to the e
bus priority (when the CPU external access and the DTC external access do not have pri
a DMAC transfer, the transfers are not operated until the DMAC releases the bus).

In dual address mode, the DMAC releases the external bus after the external space write
Since the read and write cycles are not separated, the bus is not released.

An internal space (on-chip memory and internal I/O registers) access of the DMAC and
external bus release cycle may be performed at the same time.

DMTEND2	Transfer end interrupt by channel 2 transfer counter	
DMTEND3	Transfer end interrupt by channel 3 transfer counter	
DMEEND0	Interrupt by channel 0 transfer size error Interrupt by channel 0 repeat size end Interrupt by channel 0 extended repeat area overflow on source address Interrupt by channel 0 extended repeat area overflow on destination address	
DMEEND1	Interrupt by channel 1 transfer size error Interrupt by channel 1 repeat size end Interrupt by channel 1 extended repeat area overflow on source address Interrupt by channel 1 extended repeat area overflow on destination address	
DMEEND2	Interrupt by channel 2 transfer size error Interrupt by channel 2 repeat size end Interrupt by channel 2 extended repeat area overflow on source address Interrupt by channel 2 extended repeat area overflow on destination address	
DMEEND3	Interrupt by channel 3 transfer size error Interrupt by channel 3 repeat size end Interrupt by channel 3 extended repeat area overflow on source address Interrupt by channel 3 extended repeat area overflow on destination address	Lo

Each interrupt is enabled or disabled by the DTIE and ESIE bits in DMDR for the corresp channel. A DMTEND interrupt is generated by the combination of the DTIF and DTIE bi DMDR. A DMEEND interrupt is generated by the combination of the ESIF and ESIE bit DMDR. The DMEEND interrupt sources are not distinguished. The priority among chann decided by the interrupt controller and it is shown in table 7.7. For details, see section 5, Controller.

An interrupt other than the transfer end interrupt by the transfer counter is generated wh
ESIF bit in DMDR is set to 1. The ESIF bit is set to 1 when the conditions are satisfied b
transfer while the enable bit is set to 1.

A transfer size error interrupt is generated when the next transfer cannot be performed b
DTCR value is less than the data access size, meaning that the data access size of transfe
be performed. In block transfer mode, the block size is compared with the DTCR value f
transfer error decision.

A repeat size end interrupt is generated when the next transfer is requested after complet
repeat size of transfers in repeat transfer mode. Even when the repeat area is not specifie
address register, the transfer can be stopped periodically according to the repeat size. At
when a transfer end interrupt by the transfer counter is generated, the ESIF bit is set to 1

An interrupt by an extended repeat area overflow on the source and destination addresse
generated when the address exceeds the extended repeat area (overflow). At this time, w
transfer end interrupt by the transfer counter, the ESIF bit is set to 1.

Figure 7.39 is a block diagram of interrupts and interrupt flags. To clear an interrupt, cle
DTIF or ESIF bit in DMDR to 0 in the interrupt handling routine or continue the transfe
setting the DTE bit in DMDR after setting the register. Figure 7.40 shows procedure to r
transfer by clearing a interrupt.

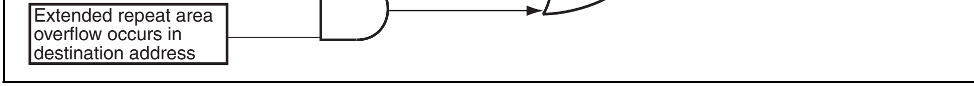
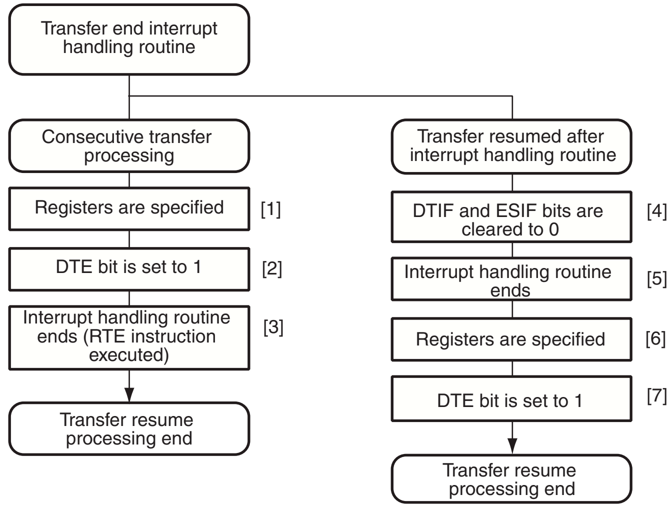


Figure 7.39 Interrupt and Interrupt Sources



[1] Specify the values in the registers such as transfer counter and address register.
[2] Set the DTE bit in DMDR to 1 to resume DMA operation. Setting the DTE bit to 1 automatically clears the D ESIF bit in DMDR to 0 and an interrupt source is cleared.
[3] End the interrupt handling routine by the RTE instruction.
[4] Read that the DTIF or the ESIF bit in DMDR = 1 and then write 0 to the bit.
[5] Complete the interrupt handling routine and clear the interrupt mask.
[6] Specify the values in the registers such as transfer counter and address register.
[7] Set the DTE bit to 1 to resume DMA operation.

Figure 7.40 Procedure Example of Resuming Transfer by Clearing Interrupt S

enters the module stop state. However, when a transfer for a channel is enabled or w
interrupt is being requested, bit MSTPA13 cannot be set to 1. Clear the DTE bit to 0
DTIF or DTIE bit in DMDR to 0, and then set bit MSTPA13.

When the clock is stopped, the DMAC registers cannot be accessed. However, the fo
register settings are valid in the module stop state. Disable them before entering the
stop state, if necessary.

— TENDE bit in DMDR is 1 (the TEND signal output enabled)

— DACKE bit in DMDR is 1 (the DACK signal output enabled)

3. Activation by $\overline{\text{DREQ}}$ Falling Edge

The $\overline{\text{DREQ}}$ falling edge detection is synchronized with the DMAC internal operatio

A. Activation request waiting state: Waiting for detecting the $\overline{\text{DREQ}}$ low level. A tr
2. is made.

B. Transfer waiting state: Waiting for a DMAC transfer. A transition to 3. is made.

C. Transfer prohibited state: Waiting for detecting the $\overline{\text{DREQ}}$ high level. A transitio
made.

After a DMAC transfer enabled, a transition to 1. is made. Therefore, the $\overline{\text{DREQ}}$ sig
sampled by low level detection at the first activation after a DMAC transfer enabled

4. Acceptation of Activation Source

At the beginning of an activation source reception, a low level is detected regardless
setting of $\overline{\text{DREQ}}$ falling edge or low level detection. Therefore, if the $\overline{\text{DREQ}}$ signal i
low before setting DMDR, the low level is received as a transfer request.

When the DMAC is activated, clear the $\overline{\text{DREQ}}$ signal of the previous transfer.

- Three transfer modes

  Normal/repeat/block transfer modes selectable

  Transfer source and destination addresses can be selected from increment/decrement
- Short address mode or full address mode selectable
  - Short address mode

    Transfer information is located on a 3-longword boundary

    The transfer source and destination addresses can be specified by 24 bits to selec
    Mbyte address space directly
  - Full address mode

    Transfer information is located on a 4-longword boundary

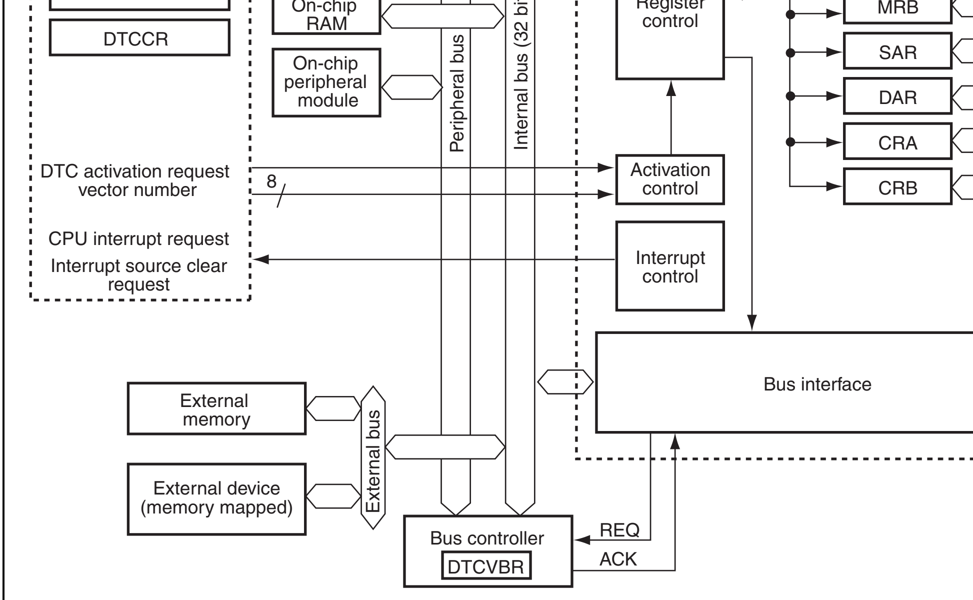
    The transfer source and destination addresses can be specified by 32 bits to selec
    Gbyte address space directly
- Size of data for data transfer can be specified as byte, word, or longword

  The bus cycle is divided if an odd address is specified for a word or longword transf

  The bus cycle is divided if address 4n + 2 is specified for a longword transfer.
- A CPU interrupt can be requested for the interrupt that activated the DTC

  A CPU interrupt can be requested after one data transfer completion

  A CPU interrupt can be requested after the specified data transfer completion
- Read skip of the transfer information specifiable
- Writeback skip executed for the fixed transfer source and destination addresses
- Module stop state specifiable



[Legend]

MRA, MRB:	DTC mode registers A, B
SAR:	DTC source address register
DAR:	DTC destination address register
CRA, CRB:	DTC transfer count registers A, B
DTCERA to DTCERE, DTCERG, and DTCERH:	DTC enable registers A to E, G, and sH
DTCCR:	DTC control register
DTCVBR:	DTC vector base register

Figure 8.1 Block Diagram of DTC

These six registers MRA, MRB, SAR, DAR, CRA, and CRB cannot be directly accesse
CPU. The contents of these registers are stored in the data area as transfer information. V
DTC activation request occurs, the DTC reads a start address of transfer information tha
in the data area according to the vector address, reads the transfer information, and trans
After the data transfer, it writes a set of updated transfer information back to the data are

- DTC enable registers A to E, G, and H (DTCERA to DTCERE, DTCERG, and DTC
- DTC control register (DTCCR)
- DTC vector base register (DTCVBR)

Bit	Bit Name	Value	R/W	Description
7	MD1	Undefined	—	DTC Mode 1 and 0
6	MD0	Undefined	—	Specify DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: Setting prohibited
5	Sz1	Undefined	—	DTC Data Transfer Size 1 and 0
4	Sz0	Undefined	—	Specify the size of data to be transferred. 00: Byte-size transfer 01: Word-size transfer 10: Longword-size transfer 11: Setting prohibited
3	SM1	Undefined	—	Source Address Mode 1 and 0
2	SM0	Undefined	—	Specify an SAR operation after a data transfer 0x: SAR is fixed (SAR writeback is skipped) 10: SAR is incremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10 11: SAR is decremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10
1, 0	—	Undefined	—	Reserved The write value should always be 0.

[Legend]

X: Don't care

Bit	Bit Name	Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable Specifies the chain transfer. For details, see s 8.5.7, Chain Transfer. The chain transfer cond selected by the CHNS bit. 0: Disables the chain transfer 1: Enables the chain transfer
6	CHNS	Undefined	—	DTC Chain Transfer Select Specifies the chain transfer condition. If the fo transfer is a chain transfer, the completion ch specified transfer count is not performed and source flag or DTCER is not cleared. 0: Chain transfer every time 1: Chain transfer only when transfer counter =
5	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, a CPU interrupt requ generated every time after a data transfer end this bit is set to 0, a CPU interrupt request is c generated when the specified number of data ends.
4	DTS	Undefined	—	DTC Transfer Mode Select Specifies either the source or destination as r block area during repeat or block transfer mod 0: Specifies the destination as repeat or block 1: Specifies the source as repeat or block are

				(by 1 when Sz1 and Sz0 = B'00; by 2 when Sz1 and Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10
1, 0	—	Undefined	—	Reserved The write value should always be 0.

[Legend]
X: Don't care

8.2.3 DTC Source Address Register (SAR)

SAR is a 32-bit register that designates the source address of data to be transferred by the

In full address mode, 32 bits of SAR are valid. In short address mode, the lower 24 bits o valid and bits 31 to 24 are ignored. At this time, the upper eight bits are filled with the va bit 23.

If a word or longword access is performed while an odd address is specified in SAR or if longword access is performed while address 4n + 2 is specified in SAR, the bus cycle is c into multiple cycles to transfer data. For details, see section 8.5.1, Bus Cycle Division.

SAR cannot be accessed directly from the CPU.

into multiple cycles to transfer data. For details, see section 8.5.1, Bus Cycle Division.

DAR cannot be accessed directly from the CPU.

8.2.5 DTC Transfer Count Register A (CRA)

CRA is a 16-bit register that designates the number of times data is to be transferred by t

In normal transfer mode, CRA functions as a 16-bit transfer counter (1 to 65,536). It is decremented by 1 every time data is transferred, and bit DTCEn (n = 15 to 0) correspond activation source is cleared and then an interrupt is requested to the CPU when the coun H'0000. The transfer count is 1 when CRA = H'0001, 65,535 when CRA = H'FFFF, and when CRA = H'0000.

In repeat transfer mode, CRA is divided into two parts: the upper eight bits (CRAH) and eight bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8- transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, an contents of CRAH are sent to CRAL when the count reaches H'00. The transfer count is CRAH = CRAL = H'01, 255 when CRAH = CRAL = H'FF, and 256 when CRAH = CR H'00.

In block transfer mode, CRA is divided into two parts: the upper eight bits (CRAH) and eight bits (CRAL). CRAH holds the block size while CRAL functions as an 8-bit block- counter (1 to 256 for byte, word, or longword). CRAL is decremented by 1 every time a (word or longword) data is transferred, and the contents of CRAH are sent to CRAL whe count reaches H'00. The block size is 1 byte (word or longword) when CRAH = CRAL = 255 bytes (words or longwords) when CRAH = CRAL = H'FF, and 256 bytes (words or longwords) when CRAH = CRAL =H'00.

CRA cannot be accessed directly from the CPU.

8.2.7 DTC enable registers A to E, G, and H (DTCERA to DTCERE, DTCERG, and DTCERH)

DTCER which is comprised of eight registers, DTCERA to DTCERE, DTCERG, and DT
is a register that specifies DTC activation interrupt sources. The correspondence between
sources and DTCE bits is shown in table 8.1. Use bit manipulation instructions such as B
BCLR to read or write a DTCE bit. If all interrupts are masked, multiple activation sourc
set at one time (only at the initial setting) by writing data after executing a dummy read o
relevant register.

Bit	15	14	13	12	11	10	9	
Bit Name	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	D
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	7	6	5	4	3	2	1	
Bit Name	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	D
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

6	DTCE6	0	R/W
5	DTCE5	0	R/W
4	DTCE4	0	R/W
3	DTCE3	0	R/W
2	DTCE2	0	R/W
1	DTCE1	0	R/W
0	DTCE0	0	R/W

8.2.8 DTC Control Register (DTCCR)

DTCCR specifies transfer information read skip.

Bit	7	6	5	4	3	2	1
Bit Name	—	—	—	RRS	RCHNE	—	—
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R/W	Reserved These bits are always read as 0. The write va always be 0.

				0: Transfer read skip is not performed. 1: Transfer read skip is performed when the ve numbers match.
3	RCHNE	0	R/W	Chain Transfer Enable After DTC Repeat Tran Enables/disables the chain transfer while trans counter (CRAL) is 0 in repeat transfer mode. In repeat transfer mode, the CRAH value is wri CRAL when CRAL is 0. Accordingly, chain trar not occur when CRAL is 0. If this bit is set to 1, chain transfer is enabled when CRAH is writter CRAL. 0: Disables the chain transfer after repeat trans 1: Enables the chain transfer after repeat trans
2, 1	—	All 0	R	Reserved These are read-only bits and cannot be modifie
0	ERR	0	R/(W)*	Transfer Stop Flag Indicates that an address error or an NMI interr occurs. If an address error or an NMI interrupt the DTC stops. 0: No interrupt occurs 1: An interrupt occurs [Clearing condition] • When writing 0 after reading 1

Note: * Only 0 can be written to clear this flag.

Bit Name														
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R

8.3 Activation Sources

The DTC is activated by an interrupt request. The interrupt source is selected by DTCER activation source can be selected by setting the corresponding bit in DTCER; the CPU in source can be selected by clearing the corresponding bit in DTCER. At the end of a data (or the last consecutive transfer in the case of chain transfer), the activation source interr corresponding DTCER bit is cleared.

8.4 Location of Transfer Information and DTC Vector Table

Locate the transfer information in the data area. The start address of transfer information located at the address that is a multiple of four (4n). Otherwise, the lower two bits are ig during access ([1:0] = B'00.) Transfer information can be located in either short address (three longwords) or full address mode (four longwords). The DTCMD bit in SYSCR sp either short address mode (DTCMD = 1) or full address mode (DTCMD = 0). For detail section 3.2.2, System Control Register (SYSCR). Transfer information located in the da shown in figure 8.2

The DTC reads the start address of transfer information from the vector table according activation source, and then reads the transfer information from the start address. Figure 8 correspondences between the DTC vector address and transfer information.

Figure 8.2 Transfer Information on Data Area

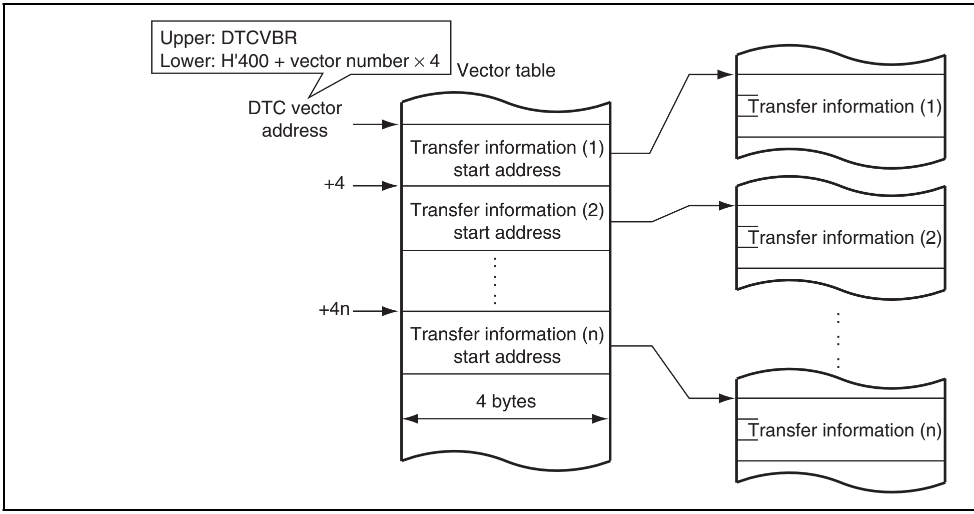


Figure 8.3 Correspondence between DTC Vector Address and Transfer Inform

	IRQ5	69	H'514	DTCEA10	
	IRQ6	70	H'518	DTCEA9	
	IRQ7	71	H'51C	DTCEA8	
	IRQ8	72	H'520	DTCEA7	
	IRQ9	73	H'524	DTCEA6	
	IRQ10	74	H'528	DTCEA5	
	IRQ11	75	H'52C	DTCEA4	
A/D	ADI	86	H'558	DTCEB15	
TPU_0	TGI0A	88	H'560	DTCEB13	
	TGI0B	89	H'564	DTCEB12	
	TGI0C	90	H'568	DTCEB11	
	TGI0D	91	H'56C	DTCEB10	
TPU_1	TGI1A	93	H'574	DTCEB9	
	TGI1B	94	H'578	DTCEB8	
TPU_2	TGI2A	97	H'584	DTCEB7	
	TGI2B	98	H'588	DTCEB6	
TPU_3	TGI3A	101	H'594	DTCEB5	
	TGI3B	102	H'598	DTCEB4	
	TGI3C	103	H'59C	DTCEB3	
	TGI3D	104	H'5A0	DTCEB2	
TPU_4	TGI4A	106	H'5A8	DTCEB1	
	TGI4B	107	H'5AC	DTCEB0	
TPU_5	TGI5A	110	H'5B8	DTCEC15	
	TGI5B	111	H'5BC	DTCEC14	L

DMAC	DMTEND0	128	H'600	DTCEC5	
	DMTEND1	129	H'604	DTCEC4	
	DMTEND2	130	H'608	DTCEC3	
	DMTEND3	131	H'60C	DTCEC2	
DMAC	DMEEND0	136	H'620	DTCED13	
	DMEEND1	137	H'624	DTCED12	
	DMEEND2	138	H'628	DTCED11	
	DMEEND3	139	H'62C	DTCED10	
SCI_0	RXI0	145	H'644	DTCED5	
	TXI0	146	H'648	DTCED4	
SCI_1	RXI1	149	H'654	DTCED3	
	TXI1	150	H'658	DTCED2	
SCI_2	RXI2	153	H'664	DTCED1	
	TXI2	154	H'668	DTCED0	
SCI_4	RXI4	161	H'684	DTCEE13	
	TXI4	162	H'688	DTCEE12	Lo

Note: * The DTCE bits with no corresponding interrupt are reserved, and the write valu always be 0. To leave software standby mode or all-module-clock-stop mode w interrupt, write 0 to the corresponding DTCE bit.

Table 8.2 shows the DTC transfer modes.

Table 8.2 DTC Transfer Modes

Transfer Mode	Size of Data Transferred at One Transfer Request	Memory Address Increment or Decrement	T C
Normal	1 byte/word/longword	Incremented/decremented by 1, 2, or 4, or fixed	1
Repeat*[1]	1 byte/word/longword	Incremented/decremented by 1, 2, or 4, or fixed	1
Block*[2]	Block size specified by CRAH (1 to 256 bytes/words/longwords)	Incremented/decremented by 1, 2, or 4, or fixed	1

Notes: 1. Either source or destination is specified to repeat area.
2. Either source or destination is specified to block area.
3. After transfer of the specified transfer count, initial state is recovered to contin operation.

Setting the CHNE bit in MRB to 1 makes it possible to perform a number of transfers w single activation (chain transfer). Setting the CHNS bit in MRB to 1 can also be made to chain transfer performed only when the transfer counter value is 0.

Figure 8.4 shows a flowchart of DTC operation, and table 8.3 summarizes the chain tran conditions (combinations for performing the second and third transfers are omitted).

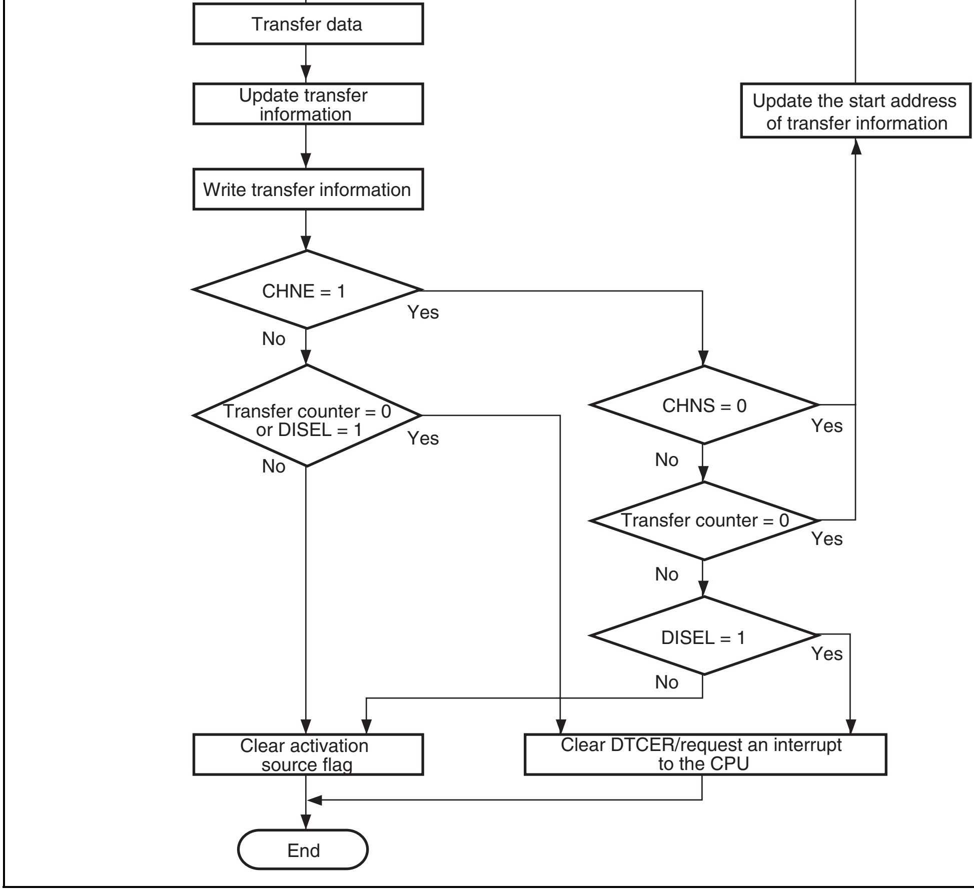


Figure 8.4 Flowchart of DTC Operation

1	1	0	Not 0	—	—		—	Ends at 1st tra
1	1	—	0*2	0	—	0	Not 0	Ends at 2nd tr
				0	—	0	0*2	Ends at 2nd tr
				0	—	1		Interrupt reque
1	1	1	Not 0	—	—	—	—	Ends at 1st tra
								Interrupt reque

Notes: 1. CRA in normal mode transfer, CRAL in repeat transfer mode, or CRB in block mode

2. When the contents of the CRAH is written to the CRAL in repeat transfer mod

8.5.1 Bus Cycle Division

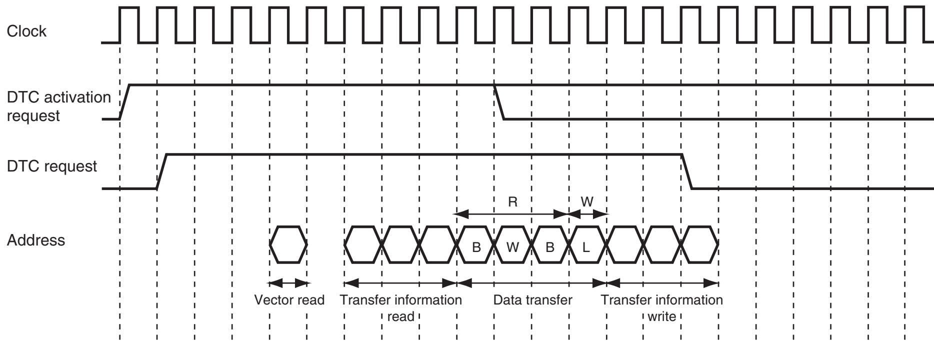
When the transfer data size is word and the SAR and DAR values are not a multiple of 2 cycle is divided and the transfer data is read from or written to in bytes. Similarly, when transfer data size is longword and the SAR and DAR values are not a multiple of 4, the is divided and the transfer data is read from or written to in words.

Table 8.4 shows the relationship among, SAR, DAR, transfer data size, bus cycle divisi access data size. Figure 8.5 shows the bus cycle division example.

Table 8.4 Number of Bus Cycle Divisions and Access Size

	Specified Data Size		
SAR and DAR Values	**Byte (B)**	**Word (W)**	**Longword (L**
Address 4n	1 (B)	1 (W)	1 (LW)
Address 2n + 1	1 (B)	2 (B-B)	3 (B-W-B)
Address 4n + 2	1 (B)	1 (W)	2 (W-W)

[Example 2: When an odd address and address 4n are specified in SAR and DAR, respectively, and when the data size of transfer is specified



[Example 3: When address 4n + 2 and address 4n are specified in SAR and DAR, respectively, and when the data size of transfer is specified

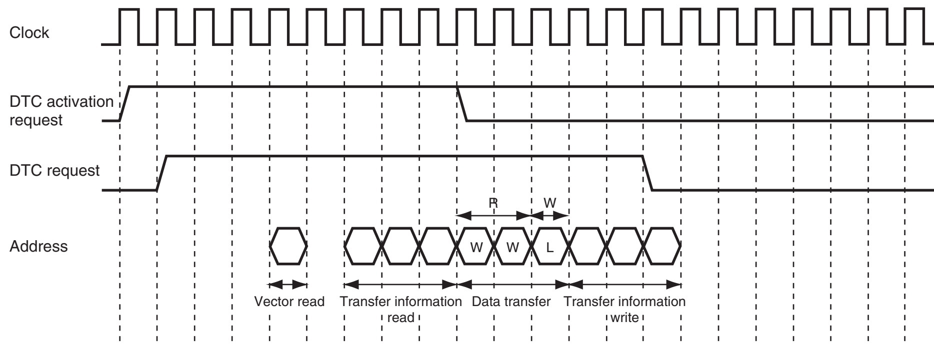


Figure 8.5 Bus Cycle Division Example

cleared to 0, the stored vector number is deleted, and the updated vector table and transf
information are read at the next activation.

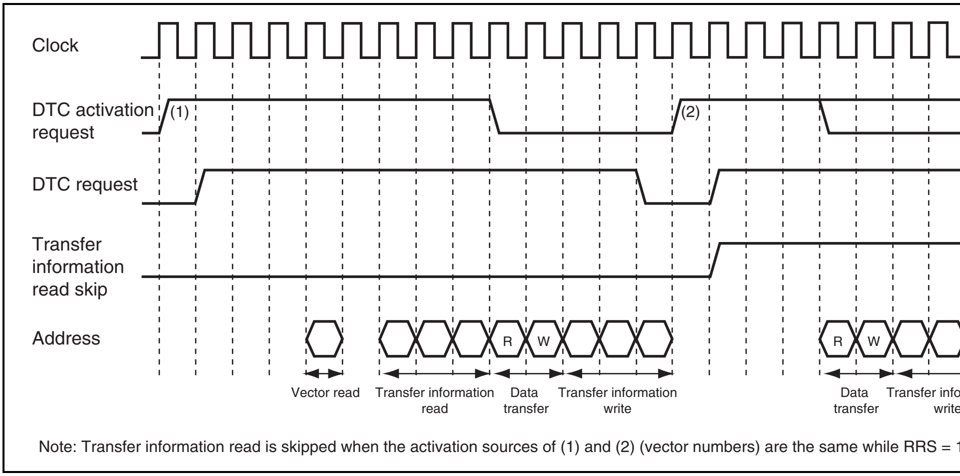


Figure 8.6 Transfer Information Read Skip Timing

SM1	DM1	SAR	DAR
0	0	Skipped	Skipped
0	1	Skipped	Written back
1	0	Written back	Skipped
1	1	Written back	Written back

8.5.4 Normal Transfer Mode

In normal transfer mode, one operation transfers one byte, one word, or one longword of
From 1 to 65,536 transfers can be specified. The transfer source and destination addresses
specified as incremented, decremented, or fixed. When the specified number of transfers
interrupt can be requested to the CPU.

Table 8.6 lists the register function in normal transfer mode. Figure 8.7 shows the memor
normal transfer mode.

Table 8.6 Register Function in Normal Transfer Mode

Register	Function	Written Back Value
SAR	Source address	Incremented/decremented/fixed
DAR	Destination address	Incremented/decremented/fixed
CRA	Transfer count A	CRA – 1
CRB	Transfer count B	Not updated

Note: * Transfer information writeback is skipped.

Figure 8.7 Memory Map in Normal Transfer Mode

8.5.5 Repeat Transfer Mode

In repeat transfer mode, one operation transfers one byte, one word, or one longword of
the DTS bit in MRB, either the source or destination can be specified as a repeat area. F
256 transfers can be specified. When the specified number of transfers ends, the transfer
and address register specified as the repeat area is restored to the initial state, and transfe
repeated. The other address register is then incremented, decremented, or left fixed. In re
transfer mode, the transfer counter (CRAL) is updated to the value specified in CRAH w
CRAL becomes H'00. Thus the transfer counter value does not reach H'00, and therefore
interrupt cannot be requested when DISEL = 0.

Table 8.7 lists the register function in repeat transfer mode. Figure 8.8 shows the memor
repeat transfer mode.

CRAH	Transfer count storage	CRAH	CRAH
CRAL	Transfer count A	CRAL – 1	CRAH
CRB	Transfer count B	Not updated	Not updated

Note: * Transfer information writeback is skipped.

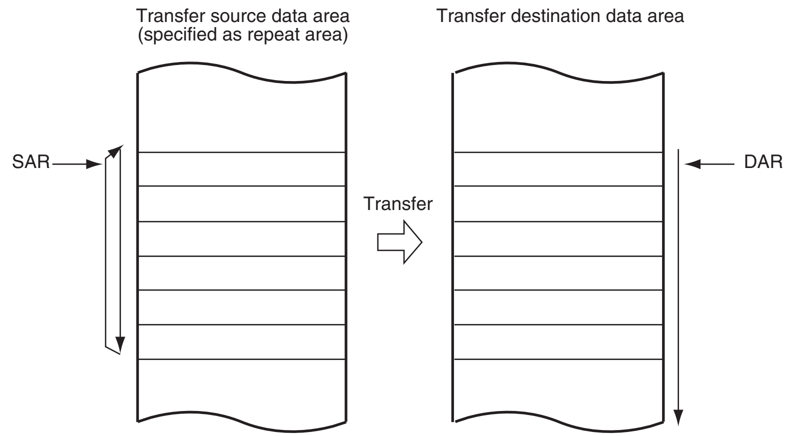


Figure 8.8 Memory Map in Repeat Transfer Mode (When Transfer Source is Specified as Repeat Area)

Table 8.8 lists the register function in block transfer mode. Figure 8.9 shows the memor
block transfer mode.

Table 8.8 Register Function in Block Transfer Mode

Register	Function	Written Back Value
SAR	Source address	DTS =0: Incremented/decremented/fixed*
		DTS = 1: SAR initial value
DAR	Destination address	DTS = 0: DAR initial value
		DTS =1: Incremented/decremented/fixed*
CRAH	Block size storage	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB – 1

Note: * Transfer information writeback is skipped.

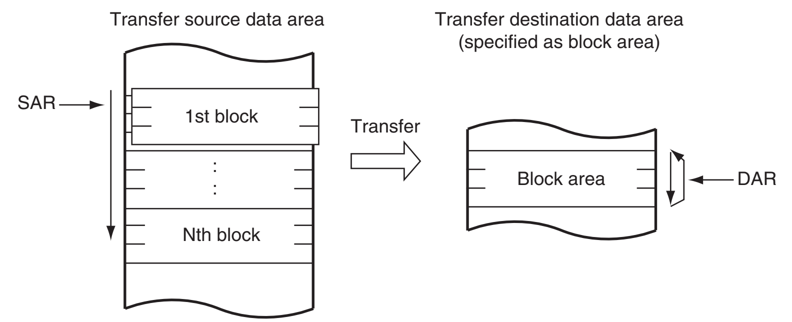


Figure 8.9 Memory Map in Block Transfer Mode
(When Transfer Destination is Specified as Block Area)

In repeat transfer mode, setting the RCHNE bit in DTCCR and the CHNE and CHNS bits to 1 enables a chain transfer after transfer with transfer counter = 1 has been completed.

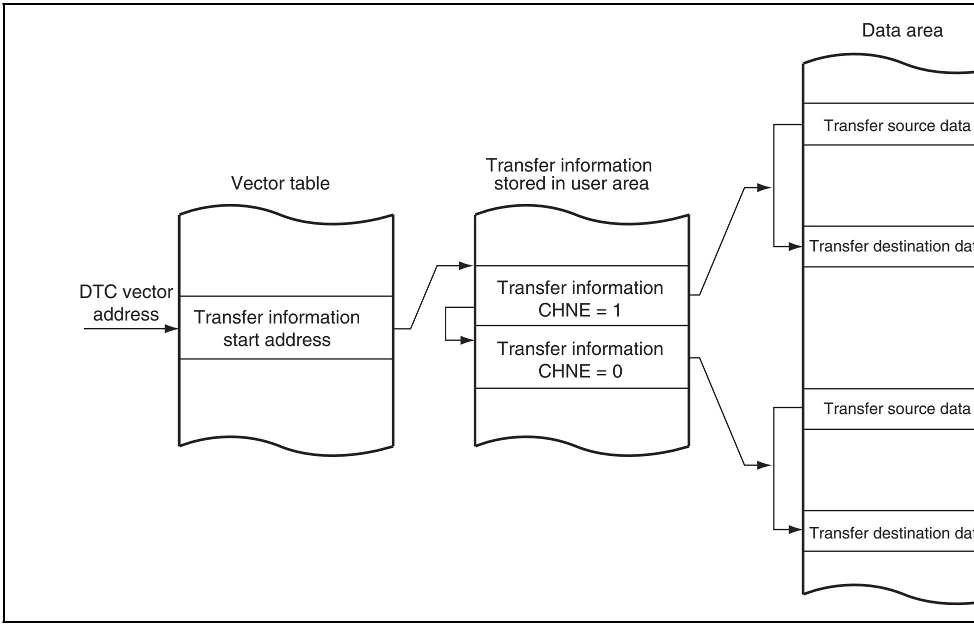


Figure 8.10 Operation of Chain Transfer

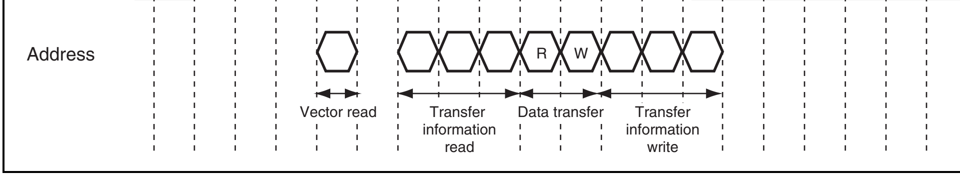


Figure 8.11 DTC Operation Timing
(Example of Short Address Mode in Normal Transfer Mode or Repeat Transfer

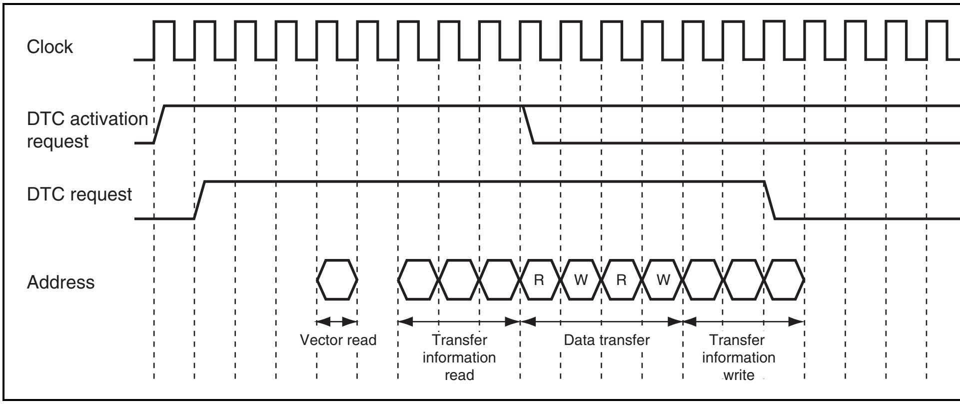


Figure 8.12 DTC Operation Timing
(Example of Short Address Mode in Block Transfer Mode with Block Size o

Figure 8.13   DTC Operation Timing (Example of Short Address Mode in Chain Tr

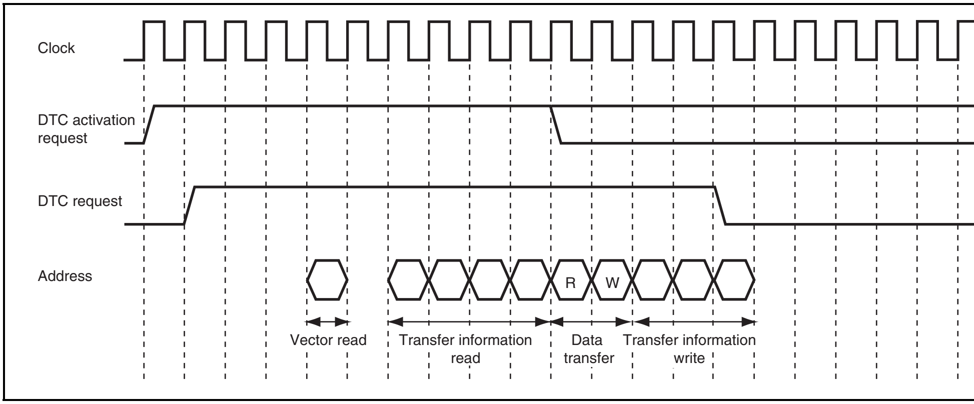


Figure 8.14   DTC Operation Timing
(Example of Full Address Mode in Normal Transfer Mode or Repeat Transfer M

Normal	1	0*[1]	4*[2]	3*[3]	0*[1]	3*[2,3]	2*[4]	1*[5]	3*[6]	2*[7]	1	3*[6]	2*[7]	1
Repeat	1	0*[1]	4*[2]	3*[3]	0*[1]	3*[2,3]	2*[4]	1*[5]	3*[6]	2*[7]	1	3*[6]	2*[7]	1
Block transfer	1	0*[1]	4*[2]	3*[3]	0*[1]	3*[2,3]	2*[4]	1*[5]	3•P *[6]	2•P*[7]	1•P	3•P *[6]	2•P*[7]	1•P

[Legend]

P: Block size (CRAH and CRAL value)

Note: 1. When transfer information read is skipped
2. In full address mode operation
3. In short address mode operation
4. When the SAR or DAR is in fixed mode
5. When the SAR and DAR are in fixed mode
6. When a longword is transferred while an odd address is specified in the addre register
7. When a word is transferred while an odd address is specified in the address when a longword is transferred while address 4n + 2 is specified

Word data read S_L	1	1	4	2	2	4	4 + 2m	2
Longword data read S_L	1	1	8	4	2	8	12 + 4m	4
Byte data write S_M	1	1	2	2	2	2	3 + m	2
Word data write S_M	1	1	4	2	2	4	4 + 2m	2
Longword data write S_M	1	1	8	4	2	8	12 + 4m	4
Internal operation S_N						1		

[Legend]

m: Number of wait cycles 0 to 7 (For details, see section 6, Bus Controller (BSC).)

The number of execution cycles is calculated from the formula below. Note that Σ means
of all transfers activated by one activation event (the number in which the CHNE bit is se
plus 1).

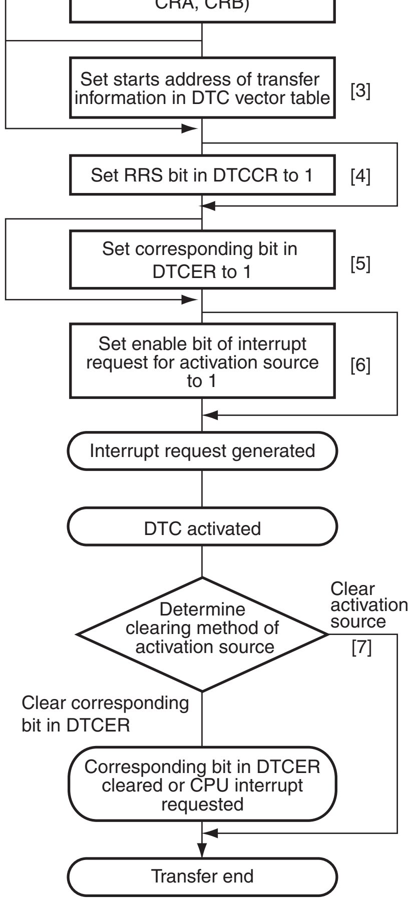
$$\text{Number of execution cycles} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L + M \cdot S_M) + N \cdot S$$

8.5.10 DTC Bus Release Timing

The DTC requests the bus mastership to the bus arbiter when an activation request occurs
DTC releases the bus after a vector read, transfer information read, a single data transfer,
transfer information writeback. The DTC does not release the bus during transfer informa
read, single data transfer, or transfer information writeback.

8.5.11 DTC Priority Level Control to the CPU

The priority of the DTC activation sources over the CPU can be controlled by the CPU pr
level specified by bits CPUP2 to CPUP0 in CPUPCR and the DTC priority level specifie
DTCP2 to DTCP0. For details, see section 5, Interrupt Controller.



information in the data area. For details on setting tra
information, see section 8.2, Register Descriptions. Fo
on location of transfer information, see section 8.4, Lo
Transfer Information and DTC Vector Table.

[3] Set the start address of the transfer information in the
vector table. For details on setting DTC vector table, s
8.4, Location of Transfer Information and DTC Vector

[4] Setting the RRS bit to 1 performs a read skip of secor
later transfer information when the DTC is activated c
tively by the same interrupt source. Setting the RRS b
always allowed. However, the value set during transfe
valid from the next transfer.

[5] Set the bit in DTCER corresponding to the DTC activa
interrupt source to 1. For the correspondence of interr
DTCER, refer to table 8.1. The bit in DTCER may be s
the second or later transfer. In this case, setting the bi
needed.

[6] Set the enable bits for the interrupt sources to be use
activation sources to 1. The DTC is activated when ar
used as an activation source is generated. For details
settings of the interrupt enable bits, see the correspor
descriptions of the corresponding module.

[7] After the end of one data transfer, the DTC clears the
source flag or clears the corresponding bit in DTCER
requests an interrupt to the CPU. The operation after
depends on the transfer information. For details, see s
8.2, Register Descriptions and figure 8.4.

Figure 8.15 DTC with Interrupt Activation

the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any v

2. Set the start address of the transfer information for an RXI interrupt at the DTC vecto
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 end (RXI) interrupt. Since the generation of a receive error during the SCI reception c 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 RXI interrupt is generated, and the DTC is activated. The receive data is transferred fr to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. Terminatio processing should be performed in the interrupt handling routine.

8.7.2 Chain Transfer

An example of DTC chain transfer is shown in which pulse output is performed using the Chain transfer can be used to perform pulse output data transfer and PPG output trigger c updating. Repeat mode transfer to the PPG's NDR is performed in the first half of the cha transfer, and normal mode transfer to the TPU's TGR in the second half. This is because c of the activation source and interrupt generation at the end of the specified number of tra restricted to the second half of the chain transfer (transfer when CHNE = 0).

3. Locate the TPU transfer information consecutively after the NDR transfer informatio

4. Set the start address of the NDR transfer information to the DTC vector address.
5. Set the bit corresponding to the TGIA interrupt in DTCER to 1.
6. Set TGRA as an output compare register (output disabled) with TIOR, and enable th interrupt with TIER.
7. Set the initial output value in PODR, and the next output value in NDR. Set bits in D NDER for which output is to be performed to 1. Using PCR, select the TPU compare be used as the output trigger.
8. Set the CST bit in TSTR to 1, and start the TCNT count operation.
9. Each time a TGRA compare match occurs, the next output value is transferred to ND set value of the next output trigger period is transferred to TGRA. The activation sou flag is cleared.
10. When the specified number of transfers are completed (the TPU transfer CRA value TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is CPU. Termination processing should be performed in the interrupt handling routine.

8.7.3 Chain Transfer when Counter = 0

By executing a second data transfer and performing re-setting of the first data transfer o 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 to have been set to start at lower address H'0000. Figure 8.16 shows the chain transfer w counter value is 0.

for the first data transfer reaches 0, the second data transfer is started. Set the upper ei
of the transfer source address for the first data transfer to H'21. The lower 16 bits of th
transfer destination address of the first data transfer and the transfer counter are H'000

5. Next, execute the first data transfer the 65536 times specified for the first data transfe
means of interrupts. When the transfer counter for the first data transfer reaches 0, the
data transfer is started. Set the upper eight bits of the transfer source address for the fi
transfer to H'20. The lower 16 bits of the transfer destination address of the first data t
and the transfer counter are H'0000.

6. Steps 4 and 5 are repeated endlessly. As repeat mode is specified for the second data t
no interrupt request is sent to the CPU.

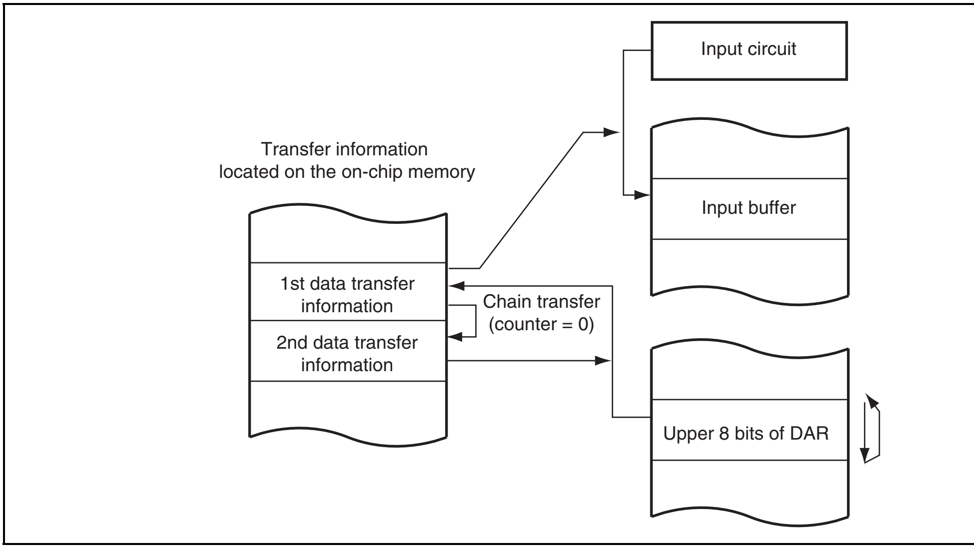


Figure 8.16 Chain Transfer when Counter = 0

Operation of the DTC can be disabled or enabled using the module stop control register. initial setting is for operation of the DTC to be enabled. Register access is disabled by se module stop state. The module stop state cannot be set while the DTC is activated. For d refer to section 23, Power-Down Modes.

8.9.2 On-Chip RAM

Transfer information can be located in on-chip RAM. In this case, the RAME bit in SYS not be cleared to 0.

8.9.3 DMAC Transfer End Interrupt

When the DTC is activated by a DMAC transfer end interrupt, the DTE bit of DMDR is controlled by the DTC but its value is modified with the write data regardless of the tran counter value and DISEL bit setting. Accordingly, even if the DTC transfer counter valu becomes 0, no interrupt request may be sent to the CPU in some cases.

8.9.4 DTCE Bit Setting

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all are disabled, multiple activation sources can be set at one time (only at the initial setting writing data after executing a dummy read on the relevant register.

8.9.6 Transfer Information Start Address, Source Address, and Destination Add

The transfer information start address to be specified in the vector table should be address
address other than address 4n is specified, the lower 2 bits of the address are regarded as

The source and destination addresses specified in SAR and DAR, respectively, will be tra
in the divided bus cycles depending on the address and data size.

8.9.7 Transfer Information Modification

When IBCCS = 1 and the DMAC is used, clear the IBCCS bit to 0 and then set to 1 agai
modifying the DTC transfer information in the CPU exception handling routine initiated
transfer end interrupt.

8.9.8 Endian Format

The DTC supports big and little endian formats. The endian formats used when transfer information is written to and when transfer information is read from by the DTC must be same.

Ports 2 and F include an open-drain control register (ODR) that controls on/off of the ou buffer PMOSs.

All of the I/O ports can drive a single TTL load and capacitive loads up to 30 pF.

All of the I/O ports can drive Darlington transistors when functioning as output ports.

Port 2 and 3 are Schmitt-trigger input. Schmitt-trigger inputs for other ports are enabled used as the $\overline{\text{IRQ}}$, TPU, TMR, or IIC2 inputs.

Table 9.1 Port Functions

			Function				Input
Port	**Description**	**Bit**	**I/O**	**Input**	**Output**	**Schmitt-Trigger Input*1**	**Pull-up MOS Function**
Port 1	General I/O port also functioning as interrupt inputs, SCI I/Os, DMAC I/Os, A/D converter inputs, TPU inputs, and IIC2 I/Os	7	P17/SCL0	$\overline{\text{IRQ7}}$-A/ TCLKD-B	—	$\overline{\text{IRQ7}}$-A, TCLKD-B, SCL0	—
		6	P16/SDA0	$\overline{\text{IRQ6}}$-A/ TCLKC-B	$\overline{\text{DACK1}}$-A	$\overline{\text{IRQ6}}$-A, TCLKC-B, SDA0	
		5	P15/SCL1	$\overline{\text{IRQ5}}$-A/ TCLKB-B/ RxD5/ IrRXD	$\overline{\text{TEND1}}$-A	$\overline{\text{IRQ5}}$-A, TCLKB-B, SCL1	
		4	P14/SDA1	$\overline{\text{DREQ1}}$-A/ $\overline{\text{IRQ4}}$-A/ TCLKA-B	TxD5/ IrTxD	$\overline{\text{IRQ4}}$-A, TCLKA-B, SDA1	
		3	P13	$\overline{\text{ADTRG0}}$/ $\overline{\text{IRQ3}}$-A	—	$\overline{\text{IRQ3}}$-A	

Port 2	General I/O port also functioning as interrupt inputs, PPG outputs, TPU I/Os, TMR I/Os, and SCI I/Os	7	P27/ TIOCB5	TIOCA5	PO7	P27, TIOCB5, TIOCA5	—
		6	P26/ TIOCA5	—	PO6/TMO1/ TxD1	All input functions	
		5	P25/ TIOCA4	TMCI1/ RxD1	PO5	P25, TIOCA4, TMCI1	
		4	P24/ TIOCB4/ SCK1	TIOCA4/ TMRI1	PO4	P24, TIOCB4, TIOCA4, TMRI1	
		3	P23/ TIOCD3	$\overline{\text{IRQ11}}$-A/ TIOCC3	PO3	P23, TIOCD3, $\overline{\text{IRQ11}}$-A	
		2	P22/ TIOCC3	$\overline{\text{IRQ10}}$-A	PO2/TMO0/ TxD0	All input functions	
		1	P21/ TIOCA3	TMCI0/ RxD0/ $\overline{\text{IRQ9}}$-A	PO1	P21, $\overline{\text{IRQ9}}$-A, TIOCA3, TMCI0	
		0	P20/ TIOCB3/ SCK0	TIOCA3/ TMRI0/ $\overline{\text{IRQ8}}$-A	PO0	P20, $\overline{\text{IRQ8}}$-A, TIOCB3, TIOCA3, TMRI0	

			TIOCA1		TEND1-B	functions	
		3	P33/ TIOCD0	TIOCC0/ TCLKB-A/ DREQ1-B	PO11	All input functions	
		2	P32/ TIOCC0	TCLKA-A	PO10/ DACK0-B	All input functions	
		1	P31/ TIOCB0	TIOCA0	PO9/ TEND0-B	All input functions	
		0	P30/ TIOCA0	DREQ0-B	PO8	All input functions	
Port 5	General input port also functioning as interrupt inputs, A/D converter inputs, and D/A converter outputs	7	—	P57/AN7/ IRQ7-B	DA1	IRQ7-B	—
		6	—	P56/AN6/ IRQ6-B	DA0	IRQ6-B	
		5	—	P55/AN5/ IRQ5-B	—	IRQ5-B	
		4	—	P54/AN4/ IRQ4-B	—	IRQ4-B	
		3	—	P53/AN3/ IRQ3-B	—	IRQ3-B	
		2	—	P52/AN2/ IRQ2-B	—	IRQ2-B	
		1	—	P51/AN1/ IRQ1-B	—	IRQ1-B	
		0	—	P50/AN0/ IRQ0-B	—	IRQ0-B	

				$\overline{\text{IRQ11}}$-B/ TMS			
		2	P62/SCK4	$\overline{\text{IRQ10}}$-B/ $\overline{\text{TRST}}$	TMO2/ $\overline{\text{DACK2}}$	$\overline{\text{IRQ10}}$-B	
		1	P61	TMCI2/ RxD4/ $\overline{\text{IRQ9}}$-B	$\overline{\text{TEND2}}$	TMCI2, $\overline{\text{IRQ9}}$-B	
		0	P60	TMRI2/ $\overline{\text{DREQ2}}$/ $\overline{\text{IRQ8}}$-B	TxD4	TMRI2, $\overline{\text{IRQ8}}$-B	
Port A	General I/O port also functioning as system clock output and bus control I/Os	7	—	PA7	Bϕ	—	—
		6	PA6	—	$\overline{\text{AS}}$/$\overline{\text{AH}}$/ $\overline{\text{BS}}$-B		
		5	PA5	—	$\overline{\text{RD}}$		
		4	PA4	—	$\overline{\text{LHWR}}$/$\overline{\text{LUB}}$		
		3	PA3	—	$\overline{\text{LLWR}}$/$\overline{\text{LLB}}$		
		2	PA2	$\overline{\text{BREQ}}$/ $\overline{\text{WAIT}}$	—		
		1	PA1	—	$\overline{\text{BACK}}$/ (RD/$\overline{\text{WR}}$)		
		0	PA0	—	$\overline{\text{BREQO}}$/ $\overline{\text{BS}}$-A		

					$\overline{\text{CS7}}$-A/ $\overline{\text{CAS}}$		
		2	PB2	—	$\overline{\text{CS2}}$-A/ $\overline{\text{CS6}}$-A/ $\overline{\text{RAS}}$		
		1	PB1	—	$\overline{\text{CS1}}$/ $\overline{\text{CS2}}$-B/ $\overline{\text{CS5}}$-A/ $\overline{\text{CS6}}$-B/ $\overline{\text{CS7}}$-B		
		0	PB0	—	$\overline{\text{CS0}}$/ $\overline{\text{CS4}}$-A/ $\overline{\text{CS5}}$-B		
Port C	General I/O port also functioning as bus control I/Os and A/D converter inputs	7	—	—	—	—	—
		6	—	—	—		
		5	—	—	—		
		4	—	—	—		
		3	PC3	—	$\overline{\text{LLCAS}}$/ DQMLL		
		2	PC2	—	$\overline{\text{LUCAS}}$/ DQMLU		
		1	—	—	—		
		0	—	—	—		

		1	PD1	—	A1		
		0	PD0	—	A0		
Port E	General I/O port also functioning as address outputs	7	PE7	—	A15	—	O
		6	PE6	—	A14		
		5	PE5	—	A13		
		4	PE4	—	A12		
		3	PE3	—	A11		
		2	PE2	—	A10		
		1	PE1	—	A9		
		0	PE0	—	A8		
Port F	General I/O port also functioning as address outputs	7	PF7	—	A23	—	O
		6	PF6	—	A22		
		5	PF5	—	A21		
		4	PF4	—	A20		
		3	PF3	—	A19		
		2	PF2	—	A18		
		1	PF1	—	A17		
		0	PF0	—	A16		

		1	PH1/D1*[2]	—	—		
		0	PH0/D0*[2]	—	—		
Port I	General I/O port also functioning as bi-directional data bus	7	PI7/D15*[2]	—	—	—	O
		6	PI6/D14*[2]	—	—		
		5	PI5/D13*[2]	—	—		
		4	PI4/D12*[2]	—	—		
		3	PI3/D11*[2]	—	—		
		2	PI2/D10*[2]	—	—		
		1	PI1/D9*[2]	—	—		
		0	PI0/D8*[2]	—	—		
Port M	General I/O port also functioning as SCI I/Os	7	—	—	—	—	—
		6	—	—	—		
		5	—	—	—		
		4	PM4	—	—		
		3	PM3	—	—		
		2	PM2	—	—		
		1	PM1	RxD6	—		
		0	PM0	—	TxD6		

Notes: 1. Pins without Schmitt-trigger input buffer have CMOS input buffer.

2. Addresses are also output when accessing to the address/data multiplexed I/

Port 3	8	O	O	O	O	—	—
Port 5	8	—	—	O	O	—	—
Port 6	6	O	O	O	O	—	—
Port A	8	O	O	O	O	—	—
Port B	4	O	O	O	O	—	—
Port C*	2	O	O	O	O	—	—
Port D	8	O	O	O	O	O	—
Port E	8	O	O	O	O	O	—
Port F	8	O	O	O	O	O	O
Port H	8	O	O	O	O	O	—
Port I	8	O	O	O	O	O	—
Port M	5	O	O	O	O	—	—

[Legend]

O: Register exists

—: No register exists

Note: * The write value should always be the initial value.

Bit	7	6	5	4	3	2	1
Bit Name	Pn7DDR	Pn6DDR	Pn5DDR	Pn4DDR	Pn3DDR	Pn2DDR	Pn1DDR
Initial Value	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W

Note: The lower six bits are valid and the upper two bits are reserved for port 6 registers.
The lower five bits are valid and the upper three bits are reserved for port M registers.
Bits 2 and 3 are valid and the other bits are reserved for port C registers.

Table 9.3 Startup Mode and Initial Value

	Startup Mode	
Port	**External Extended Mode**	**Single-Chip Mode**
Port A	H'80	H'00
Other ports	H'00	

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note: The lower six bits are valid and the upper two bits are reserved for port 6 registers.
The lower five bits are valid and the upper three bits are reserved for port M registers.
Bits 2 and 3 are valid and the other bits are reserved for port C registers.

9.1.3 Port Register (PORTn) (n = 1, 2, 3, 5, 6, A to F, H, I, and M)

PORT is an 8-bit read-only register that reflects the port pin state. A write to PORT is inv
When PORT is read, the DR bits that correspond to the respective DDR bits set to 1 are r
the status of each pin whose corresponding DDR bit is cleared to 0 is also read regardless
ICR value.

The initial value of PORT is undefined and is determined based on the port pin state.

Bit	7	6	5	4	3	2	1	
Bit Name	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Un
R/W	R	R	R	R	R	R	R	

Note: The lower six bits are valid and the upper two bits are reserved for port 6 registers.
The lower five bits are valid and the upper three bits are reserved for port M registers.
Bits 2 and 3 are valid and the other bits are reserved for port C registers.

When PORT is read, the pin state is always read regardless of the ICR value. When the I
is cleared to 0 at this time, the read pin state is not reflected in a corresponding on-chip p
module.

If ICR is modified, an internal edge may occur depending on the pin state. Accordingly,
should be modified when the corresponding input pins are not used. For example, an IRQ
modify ICR while the corresponding interrupt is disabled, clear the IRQF flag in ISR of
interrupt controller to 0, and then enable the corresponding interrupt. If an edge occurs a
ICR setting, the edge should be cancelled.

The initial value of ICR is H'00.

Bit	7	6	5	4	3	2	1
Bit Name	Pn7ICR	Pn6ICR	Pn5ICR	Pn4ICR	Pn3ICR	Pn2ICR	Pn1ICR
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: The lower six bits are valid and the upper two bits are reserved for port 6 registers.
The lower five bits are valid and the upper three bits are reserved for port M registers.
Bits 2 and 3 are valid and the other bits are reserved for port C registers.

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 9.4 Input Pull-Up MOS State

Port	Pin State	Reset	Hardware Standby Mode	Software Standby Mode	Ot Op
Port D	Address output	OFF			
	Port output	OFF			
	Port input	OFF		ON/OFF	
Port E	Address output	OFF			
	Port output	OFF			
	Port input	OFF		ON/OFF	
Port F	Address output	OFF			
	Port output	OFF			
	Port input	OFF		ON/OFF	
Port H	Data input/output	OFF			
	Port output	OFF			
	Port input	OFF		ON/OFF	
Port I	Data input/output	OFF			
	Port output	OFF			
	Port input	OFF		ON/OFF	

[Legend]

OFF: The input pull-up MOS is always off.

ON/OFF: If PCR is set to 1, the input pull-up MOS is on; if PCR is cleared to 0, the input MOS is off.

Bit Name	Pn7ODR	Pn6ODR	Pn5ODR	Pn4ODR	Pn3ODR	Pn2ODR	Pn1ODR
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

9.2 Output Buffer Control

This section describes the output priority of each pin.

The name of each peripheral module pin is followed by "_OE". This (for example: TIOC indicates whether the output of the corresponding function is valid (1) or if another settin specified (0). Table 9.5 lists each port output signal's valid setting. For details on the corresponding output signals, see the register description of each peripheral module. If t of each peripheral module pin is followed by A or B, the pin function can be modified b function control register (PFCR). For details, see section 9.3, Port Function Controller.

For a pin whose initial value changes according to the activation mode, "Initial value E" the initial value when the LSI is started up in external extended mode and "Initial value indicates the initial value when the LSI is started in single-chip mode.

IIC2	SCL0 input/output	1	
I/O port	P17 output	0	1
	P17 input (initial setting)	0	0

Note: * When pin functions as I/O: 1

(2) P16/$\overline{\text{DACK1}}$-A/$\overline{\text{IRQ6}}$-A/TCLKC-B/SDA0

The pin function is switched as shown below according to the combination of the DMAC register setting and P16DDR bit setting.

		Setting		
		DMAC	**IIC2**	**I/O Port**
Module Name	**Pin Function**	**$\overline{\text{DACK1}}$A_OE***	**SDA0_OE***	**P16DDR**
DMAC	$\overline{\text{DACK1}}$-A output	1	—	—
IIC2	SDA0 input/output	0	1	—
I/O port	P16 output	0	0	1
	P16 input (initial setting)	0	0	0

Note: * When pin functions as I/O: 1

	P15 input (initial setting)	0	0	0

Note: * When pin functions as I/O: 1

(4) P14/TxD5/IrTXD/$\overline{DREQ1}$-A/$\overline{IRQ4}$-A/TCLKA-B/SDA1

The pin function is switched as shown below according to the combination of the SCI, I
IIC2 register setting and P14DDR bit setting.

		Setting			
		SCI	IrDA	IIC2	I/O
Module Name	Pin Function	TxD5_OE	IrTXD_OE	SDA1_OE*	P1
SCI	TxD5 output	1	—	—	—
IrDA	IrTXD output	0	1	—	—
IIC2	SDA1 input/output	0	0	1	—
I/O port	P14 output	0	0	0	1
	P14 input (initial setting)	0	0	0	0

Note: * When pin functions as I/O: 1

(6) P12/SCK2/$\overline{\text{DACK0}}$-A/$\overline{\text{IRQ2}}$-A

The pin function is switched as shown below according to the combination of the DMAC register settings and P12DDR bit setting.

		Setting		
		DMAC	SCI	I/O Port
Module Name	Pin Function	$\overline{\text{DACK0A}}$_OE	SCK2_OE	P12DDR
DMAC	$\overline{\text{DACK0}}$-A output	1	—	—
SCI	SCK2 output	0	1	—
I/O port	P12 output	0	0	1
	P12 input (initial setting)	0	0	0

	P11 input (initial setting)	0	0

(8) P10/TxD2/$\overline{\text{DREQ0}}$-A/$\overline{\text{IRQ0}}$-A:

The pin function is switched as shown below according to the combination of the SCI re
setting and P10DDR bit setting.

		Setting	
		SCI	I/O Port
Module Name	Pin Function	TxD2_OE	P10DDR
SCI	TxD2 output	1	—
I/O port	P10 output	0	1
	P10 input (initial setting)	0	0

TPU	TIOCB5 output	1	—	—
PPG	PO7 output	0	1	—
I/O port	P27 output	0	0	1
	P27 input (initial setting)	0	0	0

(2) P26/PO6/TIOCA5/TMO1/TxD1

The pin function is switched as shown below according to the combination of the TPU, T
SCI, and PPG register settings and P26DDR bit setting.

		Setting				
		TPU	TMR	SCI	PPG	I/
Module Name	Pin Function	TIOCA5_OE	TMO1_OE	TxD1_OE	PO6_OE	P
TPU	TIOCA5 output	1	—	—	—	—
TMR	TMO1 output	0	1	—	—	—
SCI	TxD1 output	0	0	1	—	—
PPG	PO6 output	0	0	0	1	—
I/O port	P26 output	0	0	0	0	1
	P26 input (initial setting)	0	0	0	0	0

I/O port	P25 output	0	0	1
	P25 input (initial setting)	0	0	0

(4) P24/PO4/TIOCA4/TIOCB4/TMRI1/SCK1

The pin function is switched as shown below according to the combination of the TPU, PPG register settings and P24DDR bit setting.

		Setting			
		TPU	**SCI**	**PPG**	**I/O**
Module Name	**Pin Function**	**TIOCB4_OE**	**SCK1_OE**	**PO4_OE**	**P2**
TPU	TIOCB4 output	1	—	—	—
SCI	SCK1 output	0	1	—	—
PPG	PO4 output	0	0	1	—
I/O port	P24 output	0	0	0	1
	P24 input (initial setting)	0	0	0	0

I/O port	P23 output	0	0	1
	P23 input (initial setting)	0	0	0

(6) P22 /PO2/TIOCC3/TMO0/TxD0/$\overline{\text{IRQ10}}$-A

The pin function is switched as shown below according to the combination of the TPU, T
SCI, and PPG register settings and P22DDR bit setting.

		Setting				
		TPU	**TMR**	**SCI**	**PPG**	**I/**
Module Name	**Pin Function**	**TIOCC3_OE**	**TMO0_OE**	**TxD0_OE**	**PO2_OE**	**P**
TPU	TIOCC3 output	1	—	—	—	—
TMR	TMO0 output	0	1	—	—	—
SCI	TxD0 output	0	0	1	—	—
PPG	PO2 output	0	0	0	1	—
I/O port	P22 output	0	0	0	0	1
	P22 input (initial setting)	0	0	0	0	0

I/O port	P21 output	0	0	1
	P21 input (initial setting)	0	0	0

(8) P20/PO0/TIOCA3/TIOCB3/TMRI0/SCK0/$\overline{\text{IRQ8}}$-A

The pin function is switched as shown below according to the combination of the TPU, PPG register settings and P20DDR bit setting.

		Setting			
		TPU	SCI	PPG	I/O
Module Name	Pin Function	TIOCB3_OE	SCK0_OE	PO0_OE	P2
TPU	TIOCB3 output	1	—	—	—
SCI	SCK0 output	0	1	—	—
PPG	PO0 output	0	0	1	—
I/O port	P20 output	0	0	0	1
	P20 input (initial setting)	0	0	0	0

TPU	TIOCB2 output	1	—	—
PPG	PO15 output	0	1	—
I/O port	P37 output	0	0	1
	P37 input (initial setting)	0	0	0

(2) P36/PO14/TIOCA2

The pin function is switched as shown below according to the combination of the TPU an register settings and P36DDR bit setting.

		Setting		
		TPU	PPG	I/O Port
Module Name	Pin Function	TIOCA2_OE	PO14_OE	P36DDR
TPU	TIOCA2 output	1	—	—
PPG	PO14 output	0	1	—
I/O port	P36 output	0	0	1
	P36 input (initial setting)	0	0	0

PPG	PO13 output	0	0	1	—
I/O port	P35 output	0	0	0	1
	P35 input (initial setting)	0	0	0	0

(4) P34/PO12/TIOCA1/$\overline{\text{TEND1}}$-B

The pin function is switched as shown below according to the combination of the DMA
and PPG register settings and P34DDR bit setting.

		Setting			
		DMAC	**TPU**	**PPG**	**I/O**
Module Name	**Pin Function**	**$\overline{\text{TEND1B}}$_OE**	**TIOCA1_OE**	**PO12_OE**	**P3**
DMAC	$\overline{\text{TEND1}}$-B output	1	—	—	—
TPU	TIOCA1 output	0	1	—	—
PPG	PO12 output	0	0	1	—
I/O port	P34 output	0	0	0	1
	P34 input (initial setting)	0	0	0	0

I/O port	P33 output	0	0	1
	P33 input (initial setting)	0	0	0

(6) P32/PO10/TIOCC0/TCLKA-A/$\overline{\text{DACK0}}$-B

The pin function is switched as shown below according to the combination of the DMAC and PPG register settings and P32DDR bit setting.

		Setting			
		DMAC	TPU	PPG	I/O P
Module Name	Pin Function	DACK0B_OE	TIOCC0_OE	PO10_OE	P32
DMAC	$\overline{\text{DACK0}}$-B output	1	—	—	—
TPU	TIOCC0 output	0	1	—	—
PPG	PO10 output	0	0	1	—
I/O port	P32 output	0	0	0	1
	P32 input (initial setting)	0	0	0	0

PPG	PO9 output	0	0	1	—
I/O port	P31 output	0	0	0	1
	P31 input (initial setting)	0	0	0	0

(8) P30/PO8/TIOCA0/$\overline{\text{DREQ0}}$-B

The pin function is switched as shown below according to the combination of the TPU a
register settings and P33DDR bit setting.

		Setting		
		TPU	PPG	I/O Port
Module Name	Pin Function	TIOCA0_OE	PO8_OE	P30DDI
TPU	TIOCA0 output	1	—	—
PPG	PO8 output	0	1	—
I/O port	P30 output	0	0	1
	P30 input (initial setting)	0	0	0

D/A converter	DA0 output			

9.2.5 Port 6

(1) P65/TMO3/$\overline{\text{DACK3}}$/TCK

The pin function is switched as shown below according to the combination of the DMAC TMR register settings and P65DDR bit setting.

		Setting		
		DMAC	**TMR**	**I/O Port**
Module Name	**Pin Function**	**$\overline{\text{DACK3}}$_OE**	**TMO3_OE**	**P65DDR**
DMAC	$\overline{\text{DACK3}}$ output	1	—	—
TMR	TMO3 output	0	1	—
I/O port	P65 output	0	0	1
	P65 input (initial setting)	0	0	0

(initial setting)

(3) P63/TMRI3/$\overline{\text{DREQ3}}$/$\overline{\text{IRQ11}}$-B/TMS

The pin function is switched as shown below according to the P63DDR bit setting.

		Setting
		I/O Port
Module Name	Pin Function	P63DDR
I/O port	P63 output	1
	P63 input (initial setting)	0

SCI	SCK4 output	0	0	1	—
I/O port	P62 output	0	0	0	1
	P62 input (initial setting)	0	0	0	0

(5) P61/TMCI2/RxD4/$\overline{\text{TEND2}}$/$\overline{\text{IRQ9}}$-B

The pin function is switched as shown below according to the combination of the DMAC setting and P61DDR bit setting.

		Setting	
		DMAC	**I/O Port**
Module Name	**Pin Function**	**$\overline{\text{TEND2}}$_OE**	**P61DDR**
DMAC	$\overline{\text{TEND2}}$ output	1	—
I/O port	P61 output	0	1
	P61 input (initial setting)	0	0

	P60 input (initial setting)	0	0

9.2.6 Port A

(1) PA7/Bϕ

The pin function is switched as shown below according to the PA7DDR bit setting.

		Setting
		I/O Port
Module Name	Pin Function	PA7DDR
I/O port	Bϕ output* (initial setting E)	1
	PA7 input (initial setting S)	0

[Legend]

Initial setting E: Initial setting in external extended mode

Initial setting S: Initial setting in single-chip mode

Note: * The type of ϕ to be output switches according to the POSEL1 bit in SCKCR. F
see section 22.1.1, System Clock Control Register (SCKCR).

	BS-B output*	0	1	—	—
	$\overline{\text{AS}}$ output* (initial setting E)	0	0	1	—
I/O port	PA6 output	0	0	0	1
	PA6 input (initial setting S)	0	0	0	0

[Legend]

Initial setting E: Initial setting in external extended mode

Initial setting S: Initial setting in single-chip mode

Note: * Valid in external extended mode (EXPE = 1)

(3) PA5/$\overline{\text{RD}}$

The pin function is switched as shown below according to the combination of operating m
EXPE bit, and the PA5DDR bit settings.

		Setting	
		MCU Operating Mode	**I/O Port**
Module Name	**Pin Function**	**EXPE**	**PA5DDR**
Bus controller	$\overline{\text{RD}}$ output* (Initial setting E)	1	—
I/O port	PA5 output	0	1
	PA5 input (initial setting S)	0	0

[Legend]

Initial setting E: Initial setting in external extended mode

Initial setting S: Initial setting in single-chip mode

Note: * Valid in external extended mode (EXPE = 1)

	(initial setting E)			
I/O port	PA4 output	0	0	1
	PA4 input (initial setting S)	0	0	0

[Legend]

Initial setting E: Initial setting in external extended mode

Initial setting S: Initial setting in single-chip mode

Notes: 1. Valid in external extended mode (EXPE = 1)

2. When the byte control SRAM space is accessed while the byte control SRAM
specified or while LHWROE = 1, this pin functions as the $\overline{\text{LUB}}$ output; otherw
$\overline{\text{LHWR}}$ output.

I/O port	PA3 output	0	0	1
	PA3 input (initial setting S)	0	0	0

[Legend]

Initial setting E: Initial setting in external extended mode

Initial setting S: Initial setting in single-chip mode

Notes: 1. Valid in external extended mode (EXPE = 1)

2. If the byte control SRAM space is accessed, this pin functions as the $\overline{\text{LLB}}$ outp otherwise, the $\overline{\text{LLWR}}$.

(6) PA2/$\overline{\text{BREQ}}$/$\overline{\text{WAIT}}$

The pin function is switched as shown below according to the combination of the bus con register setting and the PA2DDR bit setting.

		Setting		
		Bus Controller		I/O Port
Module Name	Pin Function	BCR_BRLE	BCR_WAITE	PA2DDR
Bus controller	$\overline{\text{BREQ}}$ input	1	—	—
	$\overline{\text{WAIT}}$ input	0	1	—
I/O port	PA2 output	0	0	1
	PA2 input (initial setting)	0	0	0

Bus controller	BACK output *	1	—	—	—
	RD/WR output *	0	1	—	—
		0	0	1	—
I/O port	PA1 output	0	0	0	1
	PA1 input (initial setting)	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

(8) PA0/BREQO/BS-A

The pin function is switched as shown below according to the combination of operating
EXPE bit, bus controller register, port function control register (PFCR), and the PA0DD
settings.

		Setting		
		I/O Port	Bus Controller	I/O Port
Module Name	Pin Function	BSA_OE	BREQO_OE	PA0DDI
Bus controller	BS-A output*	1	—	—
	BREQO output*	0	1	—
I/O port	PA0 output	0	0	1
	PA0 input (initial setting)	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

Clock pulse generator	SDφ output*	1	—
I/O port	PB7 output	0	1
	PB7 input (initial setting)	0	0

Note: * Valid in SDRAM mode

(2) PB6/$\overline{\text{CS6}}$-D/(RD/$\overline{\text{WR}}$-B)

The pin function is switched as shown below according to the combination of operating r
EXPE bit, bus controller register, port function control register (PFCR), and the PB6DDF
settings.

		Setting			
		I/O Port			
Module Name	**Pin Function**	**Byte control SRAM Selection**	**(RD/$\overline{\text{WR}}$)-B_OE**	**$\overline{\text{CS6D}}$_OE**	**PB6I**
Bus controller	RD/$\overline{\text{WR}}$-B output*	1	—	—	—
		0	1	—	—
	$\overline{\text{CS6}}$-D output*	0	0	1	—
I/O port	PB6 output	0	0	0	1
	PB6 input (initial setting)	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

	$\overline{\text{OE}}$ output	0	1	—	—
	$\overline{\text{CS5}}$-D output*	0	0	1	—
I/O port	PB5 output	0	0	0	1
	PB5 input (initial setting)	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

(4) PB4/$\overline{\text{CS4}}$-B/$\overline{\text{WE}}$

The pin function is switched as shown below according to the combination of operating EXPE bit, bus controller register, port function control register (PFCR), and the PB4DD settings.

		Setting		
		Bus Controller		I/O Port
Module Name	Pin Function	$\overline{\text{WE}}$_OE	$\overline{\text{CS4B}}$_OE	PB4DD
Bus controller	$\overline{\text{WE}}$ output*	1	—	—
	$\overline{\text{CS4}}$-B output*	0	1	—
I/O port	PB4 output	0	0	1
	PB4 input (initial setting)	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

	$\overline{\text{CS3}}$-A output	0	1		
	$\overline{\text{CS7}}$-A output*	0	—	1	—
I/O port	PB3 output	0	0	0	1
	PB3 input (initial setting)	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

(6) PB2/$\overline{\text{CS2}}$-A/$\overline{\text{CS6}}$-A/$\overline{\text{RAS}}$

The pin function is switched as shown below according to the combination of operating n EXPE bit, bus controller register, port function control register (PFCR), and the PB2DDF settings.

		Setting			
		Bus Controller			I/O Port
Module Name	Pin Function	$\overline{\text{RAS}}$_OE	$\overline{\text{CS2A}}$_OE	$\overline{\text{CS6A}}$_OE	PB2
Bus controller	$\overline{\text{RAS}}$ output*	1	—	—	—
	$\overline{\text{CS2}}$-A output*	0	1	—	—
	$\overline{\text{CS6}}$-A output*	0	—	1	—
I/O port	PB2 output	0	0	0	1
	PB2 input (initial setting)	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

	$\overline{CS5}$-A output*	—	—	1	—	—
	$\overline{CS6}$-B output*	—	—	—	1	—
	$\overline{CS7}$-B output*	—	—	—	—	1
I/O port	PB1 output	0	0	0	0	0
	PB1 input (initial setting)	0	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

(8) PB0/$\overline{CS0}$/$\overline{CS4}$/$\overline{CS5}$-B

The pin function is switched as shown below according to the combination of operating EXPE bit, port function control register (PFCR), and the PB0DDR bit settings.

		Setting			
		I/O Port			
Module Name	**Pin Function**	**$\overline{CS0}$_OE**	**$\overline{CS4}$_OE**	**$\overline{CS5B}$_OE**	**PE**
Bus controller	$\overline{CS0}$ output (initial setting E)	1	—	—	—
	$\overline{CS4}$ output	—	1	—	—
	$\overline{CS5}$-B output	—	—	1	—
I/O port	PB0 output	0	0	0	1
	PB0 input (initial setting S)	0	0	0	0

[Legend]

Initial setting E: Initial setting in on-chip ROM disabled external extended mode

Initial setting S: Initial setting in other modes

Bus controller	LLCAS output*	1	—	—
	DQMLL output*	—	1	—
I/O port	PC3 output	0	0	1
	PC3 input (initial setting)	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

(2) PC2/$\overline{\text{LUCAS}}$/DQMLU

The pin function is switched as shown below according to the combination of operating r
EXPE bit, bus controller register, and the PC2DDR bit settings.

		Setting		
		Bus Controller		I/O Port
Module Name	Pin Function	$\overline{\text{LUCAS}}$_OE	DQMLU_OE	PC2DDR
Bus controller	$\overline{\text{LUCAS}}$ output*	1	—	—
	DQMLU output*	—	1	—
I/O port	PC2 output	0	0	1
	PC2 input (initial setting)	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

		On-chip ROM enabled extended mode	1
I/O port	PDn output	Single-chip mode*	1
	PDn input (initial setting)	Modes other than on-chip ROM disabled extended mode	0

[Legend]

n: 0 to 7

Note: * Address output is enabled by setting PDnDDR = 1 in external extended mode (EXPE = 1)

9.2.10 Port E

(1) PE7/A15, PE6/A14, PE5/A13, PE4/A12, PE3/A11, PE2/A10, PE1/A9, PE0/A8

The pin function is switched as shown below according to the combination of operating EXPE bit, and the PEnDDR bit settings.

		Setting	
			I/O Por
Module Name	Pin Function	MCU Operating Mode	PEnDD
Bus controller	Address output	On-chip ROM disabled extended mode	—
		On-chip ROM enabled extended mode	1
I/O port	PEn output	Single-chip mode*	1
	PEn input (initial setting)	Modes other than on-chip ROM disabled extended mode	0

[Legend]

n: 0 to 7

Note: * Address output is enabled by setting PDnDDR = 1 in external extended mode (EXPE = 1)

I/O port	PF7 output	0	1
	PF7 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

(2) PF6/A22

The pin function is switched as shown below according to the combination of operating m
EXPE bit, port function control register (PFCR), and the PF6DDR bit settings.

MCU Operating Mode	Pin Function	Setting	
		I/O Port	I/O Port
		A22_OE	PF6DDR
Bus controller	A22 output*	1	—
I/O port	PF6 output	0	1
	PF6 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

(initial setting)

Note: * Valid in external extended mode (EXPE = 1)

(4) PF4/A20

The pin function is switched as shown below according to the combination of operating
EXPE bit, port function control register (PFCR), and the PF4DDR bit settings.

			Setting	
			I/O Port	I/O Port
MCU Operating Mode	Module Name	Pin Function	A20_OE	PF4DDR
On-chip ROM disabled extended mode	Bus controller	A20 output	—	—
Modes other than on-chip ROM disabled extended mode	Bus controller	A20 output*	1	—
	I/O port	PF4 output	0	1
		PF4 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

Modes other than on-chip ROM disabled extended mode	Bus controller	A19 output*	1	—
	I/O port	PF3 output	0	1
		PF3 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

(6) PF2/A18

The pin function is switched as shown below according to the combination of operating m EXPE bit, port function control register (PFCR), and the PF2DDR bit settings.

			Setting	
			I/O Port	I/O Port
MCU Operating Mode	Module Name	Pin Function	A18_OE	PF2DDR
On-chip ROM disabled extended mode	Bus controller	A18 output	—	—
Modes other than on-chip ROM disabled extended mode	Bus controller	A18 output*	1	—
	I/O port	PF2 output	0	1
		PF2 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

Modes other than on-chip ROM disabled extended mode	Bus controller	A17 output*	1	—
	I/O port	PF1 output	0	1
		PF1 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

(8) PF0/A16

The pin function is switched as shown below according to the combination of operating EXPE bit, port function control register (PFCR), and the PF0DDR bit settings.

			Setting	
			I/O Port	I/O Port
MCU Operating Mode	Module Name	Pin Function	A16_OE	PF0DDR
On-chip ROM disabled extended mode	Bus controller	A16 output	—	—
Modes other than on-chip ROM disabled extended mode	Bus controller	A16 output*	1	—
	I/O port	PF0 output	0	1
		PF0 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

Bus controller	Data I/O* (initial setting E)	1	—
I/O port	PHn output	0	1
	PHn input (initial setting S)	0	0

[Legend]

Initial setting E: Initial setting in external extended mode

Initial setting S: Initial setting in single-chip mode

n: 0 to 7

Note: * Valid in external extended mode (EXPE = 1)

	(initial setting E)		
I/O port	PIn output	0	1
	PIn input (initial setting S)	0	0

[Legend]

Initial setting E: Initial setting in external extended mode

Initial setting S: Initial setting in single-chip mode

n: 0 to 7

Note: * Valid in external extended mode (EXPE = 1)

USB	PULLUP control output	1	—
I/O port	PM4 output	0	1
	PM4 input (initial setting)	0	0

(2) PM3

The pin function is switched as shown below according to the combination of the PM3DI
setting.

		Setting
		I/O Port
Module Name	Pin Function	PM3DDR
I/O port	PM3 output	1
	PM3 input (initial setting)	0

(3) PM2

The pin function is switched as shown below according to the combination of the PM2DI
setting.

		Setting
		I/O Port
Module Name	Pin Function	PM2DDR
I/O port	PM2 output	1
	PM2 input (initial setting)	0

(5) PM0/TxD6

The pin function is switched as shown below according to the combination of the SCI re
setting and PM0DDR bit setting.

		Setting	
		SCI	I/O Port
Module Name	Pin Function	TxD6_OE	PM0DDR
SCI	TxD6 output	1	—
I/O port	PM0 output	0	1
	PM0 input (initial setting)	0	0

	4	TxD5_OE	TxD5		SCR.TE = 1, IrCR.IrE = 0
		IrTxD5_OE	IrTxD5		SCR.TE = 1, IrCR.IrE = 1
		SDA1_OE	SDA1		ICCRA.ICE = 1
	3	—	—	—	—
	2	$\overline{\text{DACK0A}}$_OE	$\overline{\text{DACK0}}$	PFCR7.DMAS0[A,B] = 00	DACR.AMS = 1, DMDR.DACKE =
		SCK2_OE	SCK2		When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 while SMR.GM = 0, SCR.CKE [1, 0] = 0 SMR.GM = 1 When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 while SMR.C/A = 0, SCR.CKE [1, 0] = 0 SMR.C/A = 1, SCR.CKE 1 = 0
	1	$\overline{\text{TEND0A}}$_OE	$\overline{\text{TEND0}}$	PFCR7.DMAS0[A,B] = 00	DMDR.TENDE = 1
	0	TxD2_OE	TxD2		SCR.TE = 1
P2	7	TIOCB5_OE	TIOCB5		TPU.TIOR5.IOB3 = 0, TPU.TIOR5.IOB[1,0] = 01/10/11
		PO7_OE	PO7		NDERL.NDER7 = 1
	6	TIOCA5_OE	TIOCA5		TPU.TIOR5.IOA3 = 0, TPU.TIOR5.IOA[1,0] = 01/10/11
		TMO1_OE	TMO1		TCSR.OS3,2 = 01/10/11 or TCSR.OS[1,0] = 01/10/11
		TxD1_OE	TxD1		SCR.TE = 1
		PO6_OE	PO6		NDERL.NDER6 = 1
	5	TIOCA4_OE	TIOCA4		TPU.TIOR4.IOA3 = 0, TPU.TIOR4.IOA[1,0] = 01/10/11
		PO5_OE	PO5		NDERL.NDER5 = 1

			SMR.C/A = 1, SCR.CKE 1 = 0
	PO4_OE	PO4	NDERL.NDER4 = 1
3	TIOCD3_OE	TIOCD3	TPU.TMDR.BFB = 0, TPU.TIORL3.IOD3 = 0, TPU.TIORL3.IOD[1,0] = 01/10/11
	PO3_OE	PO3	NDERL.NDER3 = 1
2	TIOCC3_OE	TIOCC3	TPU.TMDR.BFA = 0, TPU.TIORL3.IOC3 = 0, TPU.TIORL3.IOD[1,0] = 01/10/11
	TMO0_OE	TMO0	TCSR.OS[3,2] = 01/10/11 or TCSR.OS[1,0] = 01/10/11
	TxD0_OE	TxD0	SCR.TE = 1
	PO2_OE	PO2	NDERL.NDER2 = 1
1	TIOCA3_OE	TIOCA3	TPU.TIORH3.IOA3 = 0, TPU.TIORH3.IOA[1,0] = 01/10/11
	PO1_OE	PO1	NDERL.NDER1 = 1
0	TIOCB3_OE	TIOCB3	TPU.TIORH3.IOB3 = 0, TPU.TIORH3.IOB[1,0] = 01/10/11
	SCK0_OE	SCK0	When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 while SMR.GM = 0, SCR.CKE [1, 0] = SMR.GM = 1 When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 while SMR.C/A = 0, SCR.CKE [1, 0] = SMR.C/A = 1, SCR.CKE 1 = 0
	PO0_OE	PO0	NDERL.NDER0 = 1

	TIOCB1_OE	TIOCB1		TPU.TIOR1.IOB3 = 0, TPU.TIOR1.IOB[1,0] = 01/10/11
	PO13_OE	PO13		NDERH.NDER13 = 1
4	$\overline{\text{TEND1B}}$_OE	$\overline{\text{TEND1}}$	PFCR7.DMAS1[A,B] = 01	DMDR.TENDE = 1
	TIOCA1_OE	TIOCA1		TPU.TIOR1.IOA3 = 0, TPU.TIOR1.IOA[1,0] = 01/10/11
	PO12_OE	PO12		NDERH.NDER12 = 1
3	TIOCD0_OE	TIOCD0		TPU.TMDR.BFB = 0, TPU.TIORL0.IOD3 = 0, TPU.TIORL0.IOD[1,0] = 01/10/11
	PO11_OE	PO11		NDERH.NDER11 = 1
2	$\overline{\text{DACK0B}}$_OE	$\overline{\text{DACK0}}$	PFCR7.DMAS0[A,B] = 01	DACR.AMS = 1,DMDR.DACKE =
	TIOCC0_OE	TIOCC0		TPU.TMDR.BFA = 0, TPU.TIORL0.IOC3 = 0, TPU.TIORL0.IOD[1,0] = 01/10/11
	PO10_OE	PO10		NDERH.NDER10 = 1
1	$\overline{\text{TEND0B}}$_OE	$\overline{\text{TEND0}}$	PFCR7.DMAS0[A,B] = 01	DMDR.TENDE = 1
	TIOCB0_OE	TIOCB0		TPU.TIORH0.IOB3 = 0, TPU.TIORH0.IOB[1,0] = 01/10/11
	PO9_OE	PO9		NDERH.NDER9 = 1
0	TIOCA0_OE	TIOCA0		TPU.TIORH0.IOA3 = 0, TPU.TIOH0.IOA[1,0] = 01/10/11
	PO8_OE	PO8		NDERH.NDER8 = 1

		SCK4_OE	SCK4		When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 while SMR.GM = 0, SCR.CKE [1, 0] = SMR.GM = 1 When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 while SMR.C/A = 0, SCR.CKE [1, 0] = SMR.C/A = 1, SCR.CKE 1 = 0
	1	$\overline{\text{TEND2}}$_OE	$\overline{\text{TEND2}}$	PFCR7.DMAS2[A,B] = 01	DMDR.TENDE = 1
	0	TxD4_OE	TxD4		SCR.TE = 1
PA	7	Bϕ_OE	Bϕ		PADDR.PA7DDR = 1, SCKCR.P
	6	$\overline{\text{AH}}$_OE	$\overline{\text{AH}}$		SYSCR.EXPE = 1, MPXCR.MPXEn (n = 7 to 3) = 1
		$\overline{\text{BSB}}$_OE	$\overline{\text{BS}}$	PFCR2.BSS = 1	SYSCR.EXPE = 1, PFCR2.BSE
		$\overline{\text{AS}}$_OE	$\overline{\text{AS}}$		SYSCR.EXPE = 1, PFCR2.ASOE
	5	$\overline{\text{RD}}$_OE	$\overline{\text{RD}}$		SYSCR.EXPE = 1
	4	$\overline{\text{LUB}}$_OE	$\overline{\text{LUB}}$		SYSCR.EXPE = 1, PFCR6.LHWI SRAMCR.BCSELn = 1
		$\overline{\text{LHWR}}$_OE	$\overline{\text{LHWR}}$		SYSCR.EXPE = 1, PFCR6.LHWI
	3	$\overline{\text{LLB}}$_OE	$\overline{\text{LLB}}$		SYSCR.EXPE = 1, SRAMCR.BC
		$\overline{\text{LLWR}}$_OE	$\overline{\text{LLWR}}$		SYSCR.EXPE = 1
	1	$\overline{\text{BACK}}$_OE	$\overline{\text{BACK}}$		SYSCR.EXPE = 1,BCR1.BRLE =
		(RD/$\overline{\text{WR}}$)_OE	RD/$\overline{\text{WR}}$		SYSCR.EXPE = 1, PFCR2.REW SRAMCR.BCSELn = 1
	0	$\overline{\text{BSA}}$_OE	$\overline{\text{BS}}$	PFCR2.BSS = 0	SYSCR.EXPE = 1, PFCR2.BSE
		$\overline{\text{BREQO}}$_OE	$\overline{\text{BREQO}}$		SYSCR.EXPE = 1, BCR1.BRLE BCR1.BREQOE = 1

	CS5D_OE	CS5	PFCR1.CS5S[A,B] = 11	SYSCR.EXPE = 1, PFCR0.CS5E
4	WE_OE	WE		SYSCR.EXPE = 1, DRAMCR.DRA
	CS4B_OE	CS4	PFCR1.CS4S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS4E
3	CAS_OE	CAS		SYSCR.EXPE = 1, DRAMCR.DRA DRAMCR.DTYPE = 1
	CS3A_OE	CS3	PFCR2.CS3S = 0	SYSCR.EXPE = 1, PFCR0.CS3E
	CS7A_OE	CS7	PFCR1.CS7S[A,B] = 00	SYSCR.EXPE = 1, PFCR0.CS7E
2	RAS_OE	RAS		SYSCR.EXPE = 1, DRAMCR.DRA
	CS2A_OE	CS2	PFCR2.CS2S = 0	SYSCR.EXPE = 1, PFCR0.CS2E
	CS6A_OE	CS6	PFCR1.CS6S[A,B] = 00	SYSCR.EXPE = 1, PFCR0.CS6E
1	CS1_OE	CS1		SYSCR.EXPE = 1, PFCR0.CS1E
	CS2B_OE	CS2	PFCR2.CS2S = 1	SYSCR.EXPE = 1, PFCR0.CS2E
	CS5A_OE	CS5	PFCR1.CS5S[A,B] = 00	SYSCR.EXPE = 1, PFCR0.CS5E
	CS6B_OE	CS6	PFCR1.CS6S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS6E
	CS7B_OE	CS7	PFCR1.CS7S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS7E
0	CS0_OE	CS0		SYSCR.EXPE = 1, PFCR0.CS0E
	CS4_OE	CS4		SYSCR.EXPE = 1, PFCR0.CS4E
	CS5B_OE	CS5	PFCR1.CS5S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS5E

		DQMLU_OE	DQMLU	SYSCR.EXPE = 1, ABWCR.[ABWH2,ABWL2] = x0/0 DRAMCR.DRAME = 1, DRAMCR.DTYPE = 1
PD	7	A7_OE	A7	SYSCR.EXPE = 1, PDDDR.PD7
	6	A6_OE	A6	SYSCR.EXPE = 1, PDDDR.PD6
	5	A5_OE	A5	SYSCR.EXPE = 1, PDDDR.PD5
	4	A4_OE	A4	SYSCR.EXPE = 1, PDDDR.PD4
	3	A3_OE	A3	SYSCR.EXPE = 1, PDDDR.PD3
	2	A2_OE	A2	SYSCR.EXPE = 1, PDDDR.PD2
	1	A1_OE	A1	SYSCR.EXPE = 1, PDDDR.PD1
	0	A0_OE	A0	SYSCR.EXPE = 1, PDDDR.PD0
PE	7	A15_OE	A15	SYSCR.EXPE = 1, PDDDR.PE7
	6	A14_OE	A14	SYSCR.EXPE = 1, PDDDR.PE6
	5	A13_OE	A13	SYSCR.EXPE = 1, PDDDR.PE5
	4	A12_OE	A12	SYSCR.EXPE = 1, PDDDR.PE4
	3	A11_OE	A11	SYSCR.EXPE = 1, PDDDR.PE3
	2	A10_OE	A10	SYSCR.EXPE = 1, PDDDR.PE2
	1	A9_OE	A9	SYSCR.EXPE = 1, PDDDR.PE1
	0	A8_OE	A8	SYSCR.EXPE = 1, PDDDR.PE0

	0	A16_OE	A16		SYSCR.EXPE = 1, PFCR4.A16E =
PH	7	D7_E	D7		SYSCR.EXPE = 1
	6	D6_E	D6		SYSCR.EXPE = 1
	5	D5_E	D5		SYSCR.EXPE = 1
	4	D4_E	D4		SYSCR.EXPE = 1
	3	D3_E	D3		SYSCR.EXPE = 1
	2	D2_E	D2		SYSCR.EXPE = 1
	1	D1_E	D1		SYSCR.EXPE = 1
	0	D0_E	D0		SYSCR.EXPE = 1
PI	7	D15_E	D15		SYSCR.EXPE = 1, ABWCR.ABW[
	6	D14_E	D14		SYSCR.EXPE = 1, ABWCR.ABW[
	5	D13_E	D13		SYSCR.EXPE = 1, ABWCR.ABW[
	4	D12_E	D12		SYSCR.EXPE = 1, ABWCR.ABW[
	3	D11_E	D11		SYSCR.EXPE = 1, ABWCR.ABW[
	2	D10_E	D10		SYSCR.EXPE = 1, ABWCR.ABW[
	1	D9_E	D9		SYSCR.EXPE = 1, ABWCR.ABW[
	0	D8_E	D8		SYSCR.EXPE = 1, ABWCR.ABW[
PM	4	—	—	—	—
	3	—	—	—	—
	2	—	—	—	—
	1	—	—	—	—
	0	TxD6_OE	TxD6		SCR.TE = 1

- Port function control register 6 (PFCR6)
- Port function control register 7 (PFCR7)
- Port function control register 9 (PFCR9)
- Port function control register B (PFCRB)
- Port function control register C (PFCRC)

9.3.1 Port Function Control Register 0 (PFCR0)

PFCR0 enables/disables the $\overline{CS}$ output.

Bit	7	6	5	4	3	2	1	
Bit Name	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	
Initial Value	0	0	0	0	0	0	0	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note: * 1 in external extended mode; 0 in other modes.

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/disable the corresponding output.
5	CS5E	0	R/W	0: Pin functions as I/O port
4	CS4E	0	R/W	1: Pin functions as $\overline{CSn}$ output pin
3	CS3E	0	R/W	(n = 7 to 0)
2	CS2E	0	R/W	
1	CS1E	0	R/W	
0	CS0E	Undefined*	R/W	

Note: * 1 in external extended mode, 0 in other modes.

Bit	Bit Name	Value	R/W	Description
7	CS7SA*	0	R/W	$\overline{\text{CS7}}$ Output Pin Select
6	CS7SB*	0	R/W	Selects the output pin for $\overline{\text{CS7}}$ when $\overline{\text{CS7}}$ outpu enabled (CS7E = 1) 00: Specifies pin PB3 as $\overline{\text{CS7}}$-A output 01: Specifies pin PB1 as $\overline{\text{CS7}}$-B output 10: Setting prohibited 11: Setting prohibited
5	CS6SA*	0	R/W	$\overline{\text{CS6}}$ Output Pin Select
4	CS6SB*	0	R/W	Selects the output pin for $\overline{\text{CS6}}$ when $\overline{\text{CS6}}$ outpu enabled (CS6E = 1) 00: Specifies pin PB2 as $\overline{\text{CS6}}$-A output 01: Specifies pin PB1 as $\overline{\text{CS6}}$-B output 10: Setting prohibited 11: Specifies pin PB6 as $\overline{\text{CS6}}$-D output
3	CS5SA*	0	R/W	$\overline{\text{CS5}}$ Output Pin Select
2	CS5SB*	0	R/W	Selects the output pin for $\overline{\text{CS5}}$ when $\overline{\text{CS5}}$ outpu enabled (CS5E = 1) 00: Specifies pin PB1 as $\overline{\text{CS5}}$-A output 01: Specifies pin PB0 as $\overline{\text{CS5}}$-B output 10: Setting prohibited 11: Specifies pin PB5 as $\overline{\text{CS5}}$-D output

section 6.5.3, Chip Select Signals.

9.3.3 Port Function Control Register 2 (PFCR2)

PFCR2 selects the $\overline{CS}$ output pin, enables/disables bus control I/O, and selects the bus c
pins.

Bit	7	6	5	4	3	2	1
Bit Name	—	CS2S	BSS	BSE	RDWRS	RDWRE	ASOE
Initial Value	0	0	0	0	0	0	1
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
6	CS2S*[1]	0	R/W	$\overline{CS2}$ Output Pin Select Selects the output pin for $\overline{CS2}$ when $\overline{CS2}$ outp enabled (CS2E = 1) 0: Specifies pin PB2 as $\overline{CS2}$-A output pin 1: Specifies pin PB1 as $\overline{CS2}$-B output pin

3	RDWRS*[2]	0	R/W	RD/$\overline{\text{WR}}$ Output Pin Select Selects the output pin for RD/$\overline{\text{WR}}$ 0: Specifies pin PA1 as RD/$\overline{\text{WR}}$-A output pin 1: Specifies pin PB6 as RD/$\overline{\text{WR}}$-B output pin
2	RDWRE*[2]	0	R/W	RD/$\overline{\text{WR}}$ Output Enable Enables/disables the RD/$\overline{\text{WR}}$ output 0: Disables the RD/$\overline{\text{WR}}$ output 1: Enables the RD/$\overline{\text{WR}}$ output
1	ASOE	1	R/W	$\overline{\text{AS}}$ Output Enable Enables/disables the $\overline{\text{AS}}$ output 0: Specifies pin PA6 as I/O port 1: Specifies pin PA6 as $\overline{\text{AS}}$ output pin
0	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.

Notes: 1. If multiple $\overline{\text{CS}}$ outputs are specified to a single pin according to the $\overline{\text{CSn}}$ output select bit (n = 2), multiple $\overline{\text{CS}}$ signals are output from the pin. For details, see s 6.5.3, Chip Select Signals.

2. If an area is specified as a byte control SDRAM space, the pin functions as RD output regardless of the RDWRE bit value.

Bit	Bit Name	Value	R/W	Description
7	A23E	0	R/W	Address A23 Enable Enables/disables the address output (A23) 0: Disables the A23 output 1: Enables the A23 output
6	A22E	0	R/W	Address A22 Enable Enables/disables the address output (A22) 0: Disables the A22 output 1: Enables the A22 output
5	A21E	0	R/W	Address A21 Enable Enables/disables the address output (A21) 0: Disables the A21 output 1: Enables the A21 output
4	A20E	0/1*	R/W	Address A20 Enable Enables/disables the address output (A20) 0: Disables the A20 output 1: Enables the A20 output
3	A19E	0/1*	R/W	Address A19 Enable Enables/disables the address output (A19) 0: Disables the A19 output 1: Enables the A19 output

0	A16E	0/1*	R/W	Address A16 Enable Enables/disables the address output (A16) 0: Disables the A16 output 1: Enables the A16 output

Note: * When external extended mode: Initial value is 1, reserved. This bit is always read as 1. The write value always be 1.
When other modes: Initial value is 1, enable setting.

Bit	Bit Name	Value	R/W	Description
7	—	1	R/W	Reserved This bit is always read as 1. The write value s always be 1.
6	LHWROE	1	R/W	$\overline{LHWR}$ Output Enable Enables/disables $\overline{LHWR}$ output (valid in exter extended mode). 0: Specifies pin PA4 as I/O port 1: Specifies pin PA4 as $\overline{LHWR}$ output pin
5	—	1	R/W	Reserved This bit is always read as 1. The write value s always be 1.
4	—	0	R	Reserved This is a read-only bit and cannot be modifiec
3	TCLKS	0	R/W	TPU External Clock Input Pin Select Selects the TPU external clock input pins. 0: Specifies pins P32, P33, P35, and P37 as clock input pins. 1: Specifies pins P14 to P17 as external clock pins.
2 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write va always be 0.

Bit	Bit Name	Value	R/W	Description
7	DMAS3A	0	R/W	DMAC control pin select
6	DMAS3B	0	R/W	Selects the I/O port to control DMAC_3. 00: Setting prohibited 01: Specifies pins P63 to P65 as DMAC contro 10: Setting prohibited 11: Setting prohibited
5	DMAS2A	0	R/W	DMAC control pin select
4	DMAS2B	0	R/W	Selects the I/O port to control DMAC_2. 00: Setting prohibited 01: Specifies pins P60 to P62 as DMAC contro 10: Setting prohibited 11: Setting prohibited
3	DMAS1A	0	R/W	DMAC control pin select
2	DMAS1B	0	R/W	Selects the I/O port to control DMAC_1. 00: Specifies pins P14 to P16 as DMAC contro 01: Specifies pins P33 to P35 as DMAC contro 10: Setting prohibited 11: Setting prohibited
1	DMAS0A	0	R/W	DMAC control pin select
0	DMAS0B	0	R/W	Selects the I/O port to control DMAC_0. 00: Specifies pins P10 to P12 as DMAC contro 01: Specifies pins P30 to P32 as DMAC contro 10: Setting prohibited 11: Setting prohibited

Bit	Bit Name	Value	R/W	Description
7	TPUMS5	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA5 function 0: Specifies pin P26 as output compare output capture 1: Specifies P27 as input capture input and P2 output compare
6	TPUMS4	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA4 function 0: Specifies P25 as output compare output and capture 1: Specifies P24 as input capture input and P2 output compare
5	TPUMS3A	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA3 function 0: Specifies P21 as output compare output and capture 1: Specifies P20 as input capture input and P2 output compare
4	TPUMS3B	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCC3 function 0: Specifies P22 as output compare output and capture 1: Specifies P23 as input capture input and P2 output compare

				capture 1: Specifies P35 as input capture input and P3 output compare
1	TPUMS0A	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA0 function 0: Specifies P30 as output compare output and capture 1: Specifies P31 as input capture input and P3 output compare
0	TPUMS0B	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCC0 function 0: Specifies P32 as output compare output and capture 1: Specifies P33 as input capture input and P3 output compare

Bit	Bit Name	Value	R/W	Description
7 to 4	—	All 0	R/W	Reserved These bits are always read as 0. The write va always be 0.
3	ITS11	0	R/W	$\overline{IRQ11}$ Pin Select Selects an input pin for $\overline{IRQ11}$. 0: Selects pin P23 as $\overline{IRQ11}$-A input 1: Selects pin P63 as $\overline{IRQ11}$-B input
2	ITS10	0	R/W	$\overline{IRQ10}$ Pin Select Selects an input pin for $\overline{IRQ10}$. 0: Selects pin P22 as $\overline{IRQ10}$-A input 1: Selects pin P62 as $\overline{IRQ10}$-B input
1	ITS9	0	R/W	$\overline{IRQ9}$ Pin Select Selects an input pin for $\overline{IRQ9}$. 0: Selects pin P21 as $\overline{IRQ9}$-A input 1: Selects pin P61 as $\overline{IRQ9}$-B input
0	ITS8	0	R/W	$\overline{IRQ8}$ Pin Select Selects an input pin for $\overline{IRQ8}$. 0: Selects pin P20 as $\overline{IRQ8}$-A input 1: Selects pin P60 as $\overline{IRQ8}$-B input

Bit	Bit Name	Value	R/W	Description
7	ITS7	0	R/W	$\overline{IRQ7}$ Pin Select Selects an input pin for $\overline{IRQ7}$. 0: Selects pin P17 as $\overline{IRQ7}$-A input 1: Selects pin P57 as $\overline{IRQ7}$-B input
6	ITS6	0	R/W	$\overline{IRQ6}$ Pin Select Selects an input pin for $\overline{IRQ6}$. 0: Selects pin P16 as $\overline{IRQ6}$-A input 1: Selects pin P56 as $\overline{IRQ6}$-B input
5	ITS5	0	R/W	$\overline{IRQ5}$ Pin Select Selects an input pin for $\overline{IRQ5}$. 0: Selects pin P15 as $\overline{IRQ5}$-A input 1: Selects pin P55 as $\overline{IRQ5}$-B input
4	ITS4	0	R/W	$\overline{IRQ4}$ Pin Select Selects an input pin for $\overline{IRQ4}$. 0: Selects pin P14 as $\overline{IRQ4}$-A input 1: Selects pin P54 as $\overline{IRQ4}$-B input
3	ITS3	0	R/W	$\overline{IRQ3}$ Pin Select Selects an input pin for $\overline{IRQ3}$. 0: Selects pin P13 as $\overline{IRQ3}$-A input 1: Selects pin P53 as $\overline{IRQ3}$-B input

0	ITS0	0	R/W	$\overline{IRQ0}$ Pin Select Selects an input pin for $\overline{IRQ0}$. 0: Selects pin P10 as $\overline{IRQ0}$-A input 1: Selects pin P50 as $\overline{IRQ0}$-B input

3. When a pin is used as an output, data to be output from the pin will be latched as the p
if the input function corresponding to the pin is enabled. To use the pin as an output, d
the input function for the pin by setting ICR.

9.4.2 Notes on Port Function Control Register (PFCR) Settings

1. Port function controller controls the I/O port.

   Before enabling a port function, select the input/output destination.

2. When changing input pins, this LSI may malfunction due to the internal edge generate
pin level difference before and after the change.

- To change input pins, the following procedure must be performed.

   A. Disable the input function by the corresponding on-chip peripheral module setting

   B. Select another input pin by PFCR

   C. Enable its input function by the corresponding on-chip peripheral module settings

3. If a pin function has both a select bit that modifies the input/output destination and an
bit that enables the pin function, first specify the input/output destination by the select
and then enable the pin function by the enable bit.

- 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
    - Simultaneous input/output for registers possible by counter synchronous oper
    - Maximum of 15-phase PWM output possible by combination with synchrono 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
- Conversion start trigger for the A/D converter can be generated
- Module stop state specifiable

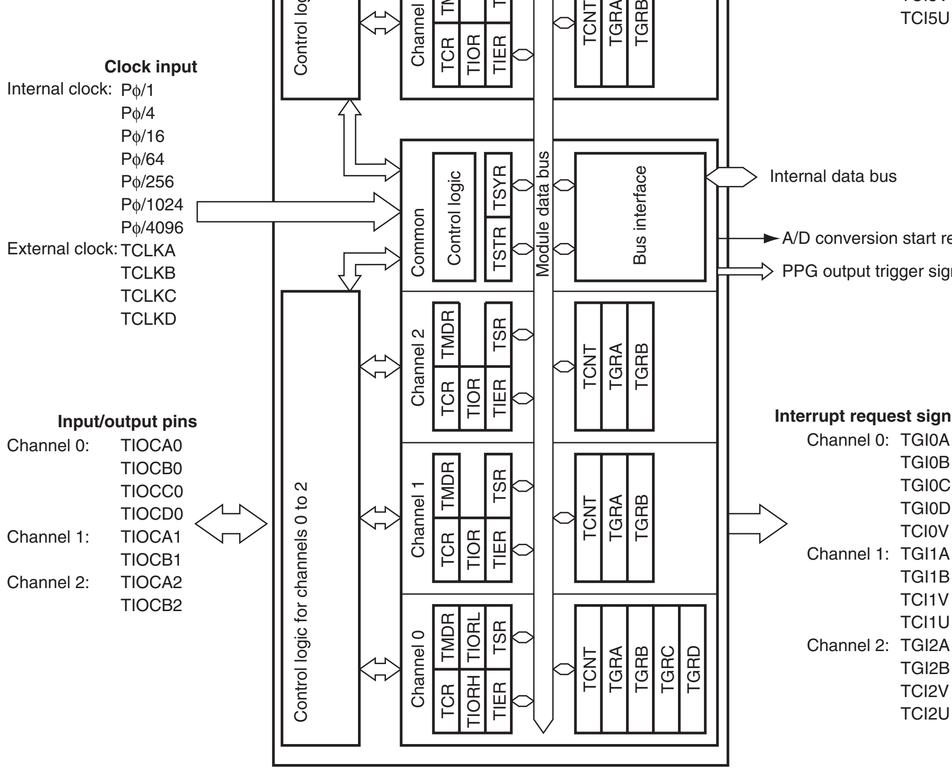
(TGR)		TGRB_0	TGRB_1	TGRB_2	TGRB_3	TGRB_4	TC
General registers/ buffer registers		TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TI TI
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TC co ma inp ca
Compare match output	0 output	O	O	O	O	O	O
	1 output	O	O	O	O	O	O
	Toggle output	O	O	O	O	O	O
Input capture function		O	O	O	O	O	O
Synchronous operation		O	O	O	O	O	O
PWM mode		O	O	O	O	O	O
Phase counting mode		—	O	O	—	O	O
Buffer operation		O	—	—	O	—	—
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	TC co ma inp ca

PPG trigger	TGRA_0/ TGRB_0 compare match or input capture	TGRA_1/ TGRB_1 compare match or input capture	TGRA_2/ TGRB_2 compare match or input capture	TGRA_3/ TGRB_3 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 C m in c C m in c O U

[Legend]

O : 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 10.1 Block Diagram of TPU

			(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 o
	TIOCB0	I/O	TGRB_0 input capture input/output compare output/PWM o
	TIOCC0	I/O	TGRC_0 input capture input/output compare output/PWM o
	TIOCD0	I/O	TGRD_0 input capture input/output compare output/PWM o
1	TIOCA1	I/O	TGRA_1 input capture input/output compare output/PWM o
	TIOCB1	I/O	TGRB_1 input capture input/output compare output/PWM o
2	TIOCA2	I/O	TGRA_2 input capture input/output compare output/PWM o
	TIOCB2	I/O	TGRB_2 input capture input/output compare output/PWM o
3	TIOCA3	I/O	TGRA_3 input capture input/output compare output/PWM o
	TIOCB3	I/O	TGRB_3 input capture input/output compare output/PWM o
	TIOCC3	I/O	TGRC_3 input capture input/output compare output/PWM o
	TIOCD3	I/O	TGRD_3 input capture input/output compare output/PWM o
4	TIOCA4	I/O	TGRA_4 input capture input/output compare output/PWM o
	TIOCB4	I/O	TGRB_4 input capture input/output compare output/PWM o
5	TIOCA5	I/O	TGRA_5 input capture input/output compare output/PWM o
	TIOCB5	I/O	TGRB_5 input capture input/output compare output/PWM o

— 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)

- Channel 1
  — Timer control register_1 (TCR_1)
  — Timer mode register_1 (TMDR_1)
  — Timer I/O control register _1 (TIOR_1)
  — Timer interrupt enable register_1 (TIER_1)
  — Timer status register_1 (TSR_1)
  — Timer counter_1 (TCNT_1)
  — Timer general register A_1 (TGRA_1)
  — Timer general register B_1 (TGRB_1)
- Channel 2
  — 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 general register B_3 (TGRB_3)

— Timer general register C_3 (TGRC_3)

— Timer general register D_3 (TGRD_3)

- Channel 4

— 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)

- Channel 5

— 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)

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 sou
5	CCLR0	0	R/W	tables 10.3 and 10.4 for details.
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the input clock edge. For detail table 10.5. When the input clock is counted usin edges, the input clock period is halved (e.g. Pϕ/4 edges = Pϕ/2 rising edge). If phase counting mo used on channels 1, 2, 4, and 5, this setting is ig and the phase counting mode setting has priorit clock edge selection is valid when the input clocl or slower. This setting is ignored if the input cloc or when overflow/underflow of another channel i selected.
2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The c source can be selected independently for each c
0	TPSC0	0	R/W	See tables 10.6 to 10.11 for details. To select the clock as the clock source, the DDR bit and ICR b corresponding pin should be set to 0 and 1, resp For details, see section 9, I/O Ports.

				synchronous operation*1
	1	0	0	TCNT clearing disabled
	1	0	1	TCNT cleared by TGRC compare mat capture*2
	1	1	0	TCNT cleared by TGRD compare mat capture*2
	1	1	1	TCNT cleared by counter clearing for a channel performing synchronous clear synchronous operation*1

Notes: 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.

2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared beca buffer register setting has priority, and compare match/input capture does not

Table 10.4 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
	0	0	1	TCNT cleared by TGRA compare mat capture
	0	1	0	TCNT cleared by TGRB compare mat capture
	0	1	1	TCNT cleared by counter clearing for a channel performing synchronous clear synchronous operation*1

Notes: 1. Synchronous operation is selected 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 modified.

Table 10.6 TPSC2 to TPSC0 (Channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on Pϕ/1
	0	0	1	Internal clock: counts on Pϕ/4
	0	1	0	Internal clock: counts on Pϕ/16
	0	1	1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin inp
	1	0	1	External clock: counts on TCLKB pin inp
	1	1	0	External clock: counts on TCLKC pin inp
	1	1	1	External clock: counts on TCLKD pin inp

Table 10.7 TPSC2 to TPSC0 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on Pϕ/1
	0	0	1	Internal clock: counts on Pϕ/4
	0	1	0	Internal clock: counts on Pϕ/16
	0	1	1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin inp
	1	0	1	External clock: counts on TCLKB pin inp
	1	1	0	Internal clock: counts on Pϕ/256
	1	1	1	Counts on TCNT2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

	1	1	0	External clock: counts on TCLKC pin i
	1	1	1	Internal clock: counts on Pϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 10.9 TPSC2 to TPSC0 (Channel 3)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3	0	0	0	Internal clock: counts on Pϕ/1
	0	0	1	Internal clock: counts on Pϕ/4
	0	1	0	Internal clock: counts on Pϕ/16
	0	1	1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin i
	1	0	1	Internal clock: counts on Pϕ/1024
	1	1	0	Internal clock: counts on Pϕ/256
	1	1	1	Internal clock: counts on Pϕ/4096

	1	1	0	Internal clock: counts on Pϕ/1024
	1	1	1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

Table 10.11 TPSC2 to TPSC0 (Channel 5)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	0	Internal clock: counts on Pϕ/1
	0	0	1	Internal clock: counts on Pϕ/4
	0	1	0	Internal clock: counts on Pϕ/16
	0	1	1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin inp
	1	0	1	External clock: counts on TCLKC pin inp
	1	1	0	Internal clock: counts on Pϕ/256
	1	1	1	External clock: counts on TCLKD pin inp

Note: This setting is ignored when channel 5 is in phase counting mode.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	R	Reserved These are read-only bits and cannot be modifie
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to normally operate and TGRD are to be used together for buffer op When TGRD is used as a buffer register, TGRD capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRD reserved. It is always read as 0 and cannot be 0: TGRB operates normally 1: TGRB and TGRD used together for buffer op
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to normally operate and TGRC are to be used together for buffer op When TGRC is used as a buffer register, TGRC capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRC reserved. It is always read as 0 and cannot be 0: TGRA operates normally 1: TGRA and TGRC used together for buffer op
3	MD3	0	R/W	Modes 3 to 0
2	MD2	0	R/W	Set the timer operating mode.
1	MD1	0	R/W	MD3 is a reserved bit. The write value should a 0. See table 10.12 for details.
0	MD0	0	R/W	

0	1	1	0	Phase counting mode 3
0	1	1	1	Phase counting mode 4
1	X	X	X	—

[Legend]

X: Don't care

Notes: 1. MD3 is a reserved bit. The write value should always be 0.

2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 shoul be written to MD2.

10.3.3 Timer I/O Control Register (TIOR)

TIOR controls TGR. The TPU has eight TIOR registers, two each for channels 0 and 3, a each for channels 1, 2, 4, and 5. Care is required since TIOR is affected by the TMDR se

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counte cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the re operates as a buffer register.

To designate the input capture pin in TIOR, the DDR bit and ICR bit for the correspondi should be set to 0 and 1, respectively. For details, see section 9, I/O Ports.

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- 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.
5	IOB1	0	R/W	For details, see tables 10.13, 10.15, 10.16, 10.1
4	IOB0	0	R/W	and 10.20.
3	IOA3	0	R/W	I/O Control A3 to A0
2	IOA2	0	R/W	Specify the function of TGRA.
1	IOA1	0	R/W	For details, see tables 10.21, 10.23, 10.24, 10.2
0	IOA0	0	R/W	and 10.28.

- 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.
5	IOD1	0	R/W	For details, see tables 10.14 and 10.18.
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.
1	IOC1	0	R/W	For details, see tables 10.22 and 10.26.
0	IOC0	0	R/W	

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCB0 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCB0 pin Input capture at both edges
1	1	x	x		Capture input source is channel 1/count Input capture at TCNT_1 count-up/count

[Legend]

X: Don't care

Note: When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and Pϕ/1 is used as the TC count clock, this setting is invalid and input capture is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCD0 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCD0 pin Input capture at both edges
1	1	X	X		Capture input source is channel 1/coun Input capture at TCNT_1 count-up/cour

[Legend]

X: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and Pϕ/1 is used as t 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 reg setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCB1 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCB1 pin Input capture at both edges
1	1	X	X		TGRC_0 compare match/input capture Input capture at generation of TGRC_0 c match/input capture

[Legend]

X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	X	0	1		Capture input source is TIOCB2 pin Input capture at falling edge
1	X	1	X		Capture input source is TIOCB2 pin Input capture at both edges

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCB3 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCB3 pin Input capture at both edges
1	1	x	x		Capture input source is channel 4/count Input capture at TCNT_4 count-up/count

[Legend]

X: Don't care

Note: When bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and Pϕ/1 is used as the TC count clock, this setting is invalid and input capture is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD3 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCD3 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCD3 pin Input capture at both edges
1	1	x	x		Capture input source is channel 4/coun Input capture at TCNT_4 count-up/cour

[Legend]

X: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and Pϕ/1 is used as tl TCNT_4 count clock, this setting is invalid and input capture is not generated

2. When the BFB bit in TMDR_3 is set to 1 and TGRD_3 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCB4 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCB4 pin Input capture at both edges
1	1	x	x		Capture input source is TGRC_3 compa match/input capture Input capture at generation of TGRC_3 c match/input capture

[Legend]

X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	x	0	1		Capture input source is TIOCB5 pin Input capture at falling edge
1	x	1	x		Capture input source is TIOCB5 pin Input capture at both edges

[Legend]

X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCA0 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA0 pin Input capture at both edges
1	1	X	X		Capture input source is channel 1/count Input capture at TCNT_1 count-up/count

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCC0 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCC0 pin Input capture at both edges
1	1	X	X		Capture input source is channel 1/coun Input capture at TCNT_1 count-up/coun

[Legend]

X: Don't care

Note: 1. When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCA1 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA1 pin Input capture at both edges
1	1	X	X		Capture input source is TGRA_0 compa match/input capture Input capture at generation of channel 0/ compare match/input capture

[Legend]

X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	X	0	1		Capture input source is TIOCA2 pin Input capture at falling edge
1	X	1	X		Capture input source is TIOCA2 pin Input capture at both edges

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCA3 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA3 pin Input capture at both edges
1	1	X	X		Capture input source is channel 4/count Input capture at TCNT_4 count-up/count

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCC3 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCC3 pin Input capture at both edges
1	1	X	X		Capture input source is channel 4/coun Input capture at TCNT_4 count-up/cour

[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 reg setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCA4 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA4 pin Input capture at both edges
1	1	X	X		Capture input source is TGRA_3 compa match/input capture Input capture at generation of TGRA_3 c match/input capture

[Legend]

X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	X	0	1		Input capture source is TIOCA5 pin Input capture at falling edge
1	X	1	X		Input capture source is TIOCA5 pin Input capture at both edges

[Legend]
X: Don't care

Bit	Bit Name	Initial value	R/W	Description
7	TTGE	0	R/W	A/D Conversion Start Request Enable Enables/disables generation of A/D conversion s requests by TGRA input capture/compare match 0: A/D conversion start request generation disab 1: A/D conversion start request generation enabl
6	—	1	R	Reserved This is a read-only bit and cannot be modified.
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables/disables interrupt requests (TCIU) by th flag when the TCFU flag in TSR is set to 1 in cha 2, 4, and 5. In channels 0 and 3, bit 5 is reserved. It is alway 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/disables interrupt requests (TCIV) by th 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

2	TGIEC	0	R/W	TGR Interrupt Enable C Enables/disables interrupt requests (TGIC) by t bit when the TGFC bit in TSR is set to 1 in char and 3. In channels 1, 2, 4, and 5, bit 2 is reserved. It is read as 0 and cannot be modified. 0: Interrupt requests (TGIC) by TGFC bit disabl 1: Interrupt requests (TGIC) by TGFC bit enable
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables/disables interrupt requests (TGIB) by t bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabl 1: Interrupt requests (TGIB) by TGFB bit enable
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables/disables interrupt requests (TGIA) by t bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabl 1: Interrupt requests (TGIA) by TGFA bit enable

Bit	Bit Name	Initial value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCN in channels 1, 2, 4, and 5. In channels 0 and 3, bit 7 is reserved. It is alway 1 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	1	R	Reserved This is a read-only bit and cannot be modified.
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that a TCNT underflow occurred when channels 1, 2, 4, and 5 are set to counting mode. In channels 0 and 3, bit 5 is reserved. It is alway 0 and cannot be modified. [Setting condition] When the TCNT value underflows (changes from to H'FFFF) [Clearing condition] When a 0 is written to TCFU after reading TCFU (When the CPU is used to clear this flag by writin while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)

				read the flag after writing 0 to it.)
3	TGFD	0	R/(W)*	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of TGR capture or compare match in channels 0 and 3. In channels 1, 2, 4, and 5, bit 3 is reserved. It is read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRD while TGRD is functio output compare register • When TCNT value is transferred to TGRD b capture signal while TGRD is functioning as capture register [Clearing conditions] • When DTC is activated by a TGID interrupt DISEL bit in MRB of DTC is 0 • When 0 is written to TGFD after reading TG (When the CPU is used to clear this flag by while the corresponding interrupt is enabled to read the flag after writing 0 to it.)

				• When TCNT value is transferred to TGRC by capture signal while TGRC is functioning as i capture register [Clearing conditions] • When DTC is activated by a TGIC interrupt w DISEL bit in MRB of DTC is 0 • When 0 is written to TGFC after reading TGF (When the CPU is used to clear this flag by w while the corresponding interrupt is enabled, to read the flag after writing 0 to it.)
1	TGFB	0	R/(W)*	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGR capture or compare match. [Setting conditions] • When TCNT = TGRB while TGRB is function output compare register • When TCNT value is transferred to TGRB by capture signal while TGRB is functioning as i capture register [Clearing conditions] • When DTC is activated by a TGIB interrupt w DISEL bit in MRB of DTC is 0 • When 0 is written to TGFB after reading TGF (When the CPU is used to clear this flag by w while the corresponding interrupt is enabled, to read the flag after writing 0 to it.)

capture register

[Clearing conditions]

- When DTC is activated by a TGIA interrupt DISEL bit in MRB of DTC is 0
- When DMAC is activated by a TGIA interrup the DTA bit in DMDR of DMAC is 1
- When 0 is written to TGFA after reading TG

(When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.)

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

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.3.7 Timer General Register (TGR)

TGR is a 16-bit readable/writable register with a dual function as output compare and inp
capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two
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 m
always be accessed in 16-bit units. TGR and buffer register combinations during buffer o
are TGRA–TGRC and TGRB–TGRD.

Bit	15	14	13	12	11	10	9
Bit Name							
Initial Value	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

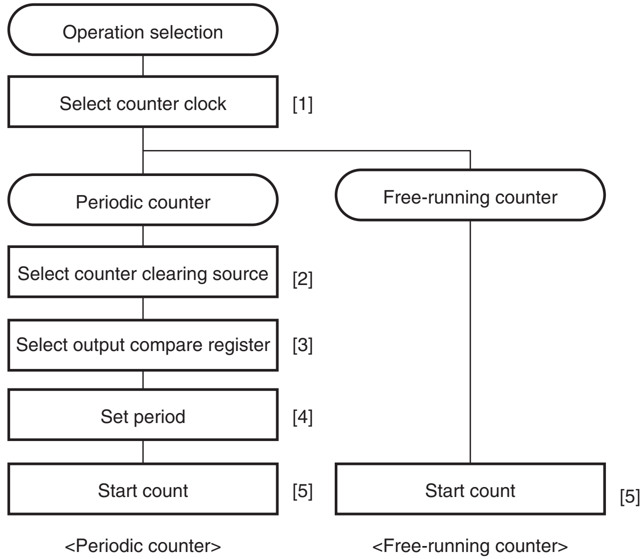
Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.
5	CST5	0	R/W	Counter Start 5 to 0
4	CST4	0	R/W	These bits select operation or stoppage for TCN
3	CST3	0	R/W	If 0 is written to the CST bit during operation wi TIOC pin designated for output, the counter sto TIOC pin output compare output level is retaine is written to when the CST bit is cleared to 0, th output level will be changed to the set initial out 0: TCNT_5 to TCNT_0 count operation is stopp 1: TCNT_5 to TCNT_0 performs count operatio
2	CST2	0	R/W	
1	CST1	0	R/W	
0	CST0	0	R/W	

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.
5 4 3 2 1 0	SYNC5 SYNC4 SYNC3 SYNC2 SYNC1 SYNC0	0 0 0 0 0 0	R/W R/W R/W R/W R/W R/W	Timer Synchronization 5 to 0 These bits select whether operation is independe synchronized with other channels. When synchronous operation is selected, synchr presetting of multiple channels, and synchronous through counter clearing on another channel are To set synchronous operation, the SYNC bits for two channels must be set to 1. To set synchrono clearing, in addition to the SYNC bit, the TCNT c source must also be set by means of bits CCLR2 CCLR0 in TCR. 0: TCNT_5 to TCNT_0 operate independently (T presetting/clearing is unrelated to other chann 1: TCNT_5 to TCNT_0 perform synchronous ope (TCNT synchronous presetting/synchronous c is possible)

When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the correspo
channel starts counting. TCNT can operate as a free-running counter, periodic counter, a

(a) Example of count operation setting procedure

Figure 10.2 shows an example of the count operation setting procedure.



[1] Select the counter clock w
bits TPSC2 to TPSC0 in T
At the same time, select t
input clock edge with bits
CKEG1 and CKEG0 in TC

[2] For periodic counter oper
select the TGR to be use
the TCNT clearing source
bits CCLR2 to CCLR0 in

[3] Designate the TGR selec
[2] as an output compare
register by means of TIO

[4] Set the periodic counter c
the TGR selected in [2].

[5] Set the CST bit in TSTR t
start the counter operatio

Figure 10.2 Example of Counter Operation Setting Procedure

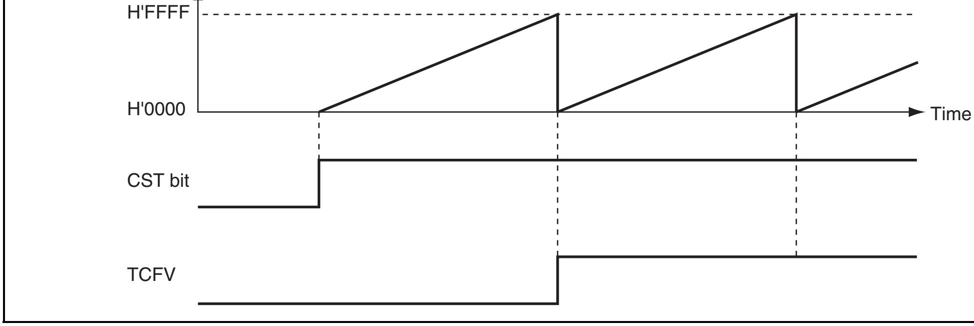


Figure 10.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the
channel performs periodic count operation. The TGR register for setting the period is des
as an output compare register, and counter clearing by compare match is selected by mea
CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts count-up ope
a periodic counter when the corresponding bit in TSTR is set to 1. When the count value
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
After a compare match, TCNT starts counting up again from H'0000.

Figure 10.4 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform 0, 1, or toggle output from the corresponding output pin using a c
match.

(a) Example of setting procedure for waveform output by compare match

Figure 10.5 shows an example of the setting procedure for waveform output by a compa

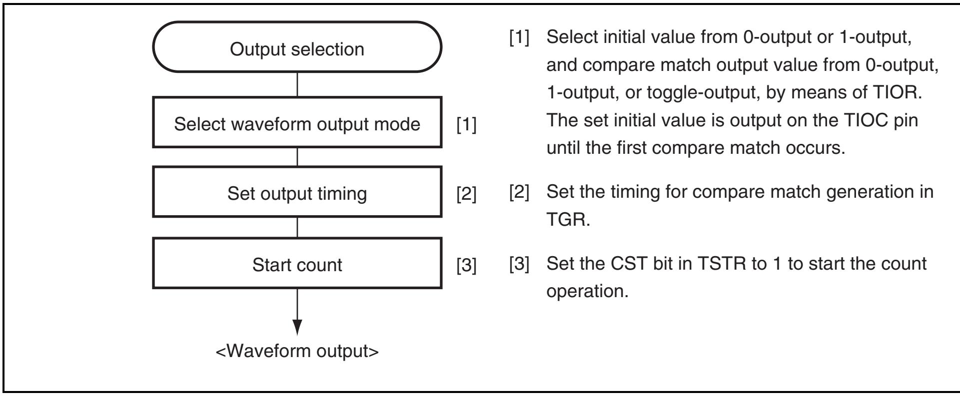


Figure 10.5 Example of Setting Procedure for Waveform Output by Compare

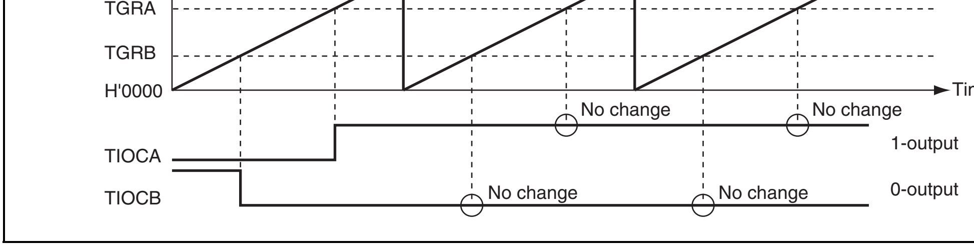


Figure 10.6 Example of 0-Output/1-Output Operation

Figure 10.7 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing p
by compare match B), and settings have been made so that output is toggled by both com
match A and compare match B.

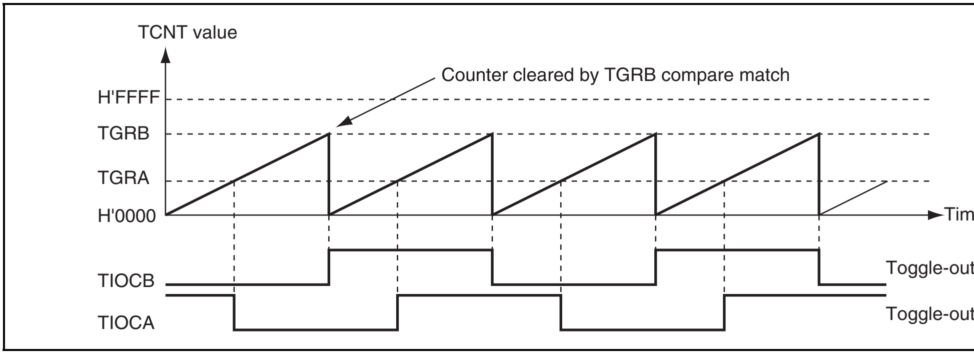
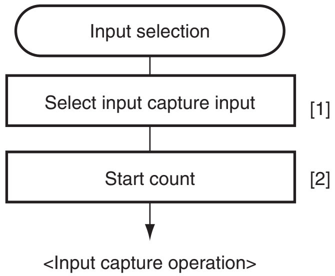


Figure 10.7 Example of Toggle Output Operation

(a) Example of setting procedure for input capture operation

Figure 10.8 shows an example of the setting procedure for input capture operation.



[1] Designate TGR as an input capture register by means of TIOR, and select the input capture source and input signal edge (rising edge, falling edge, or both edges).

[2] Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.8 Example of Setting Procedure for Input Capture Operation

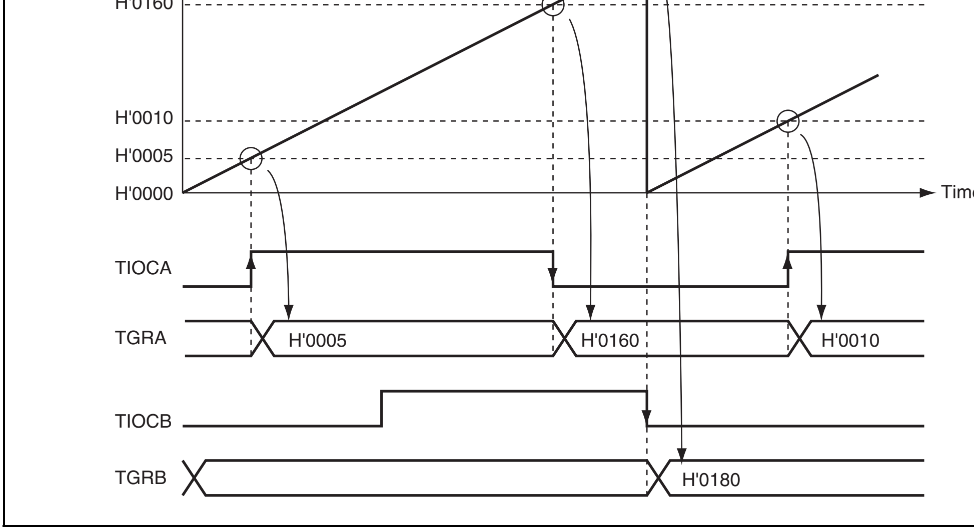
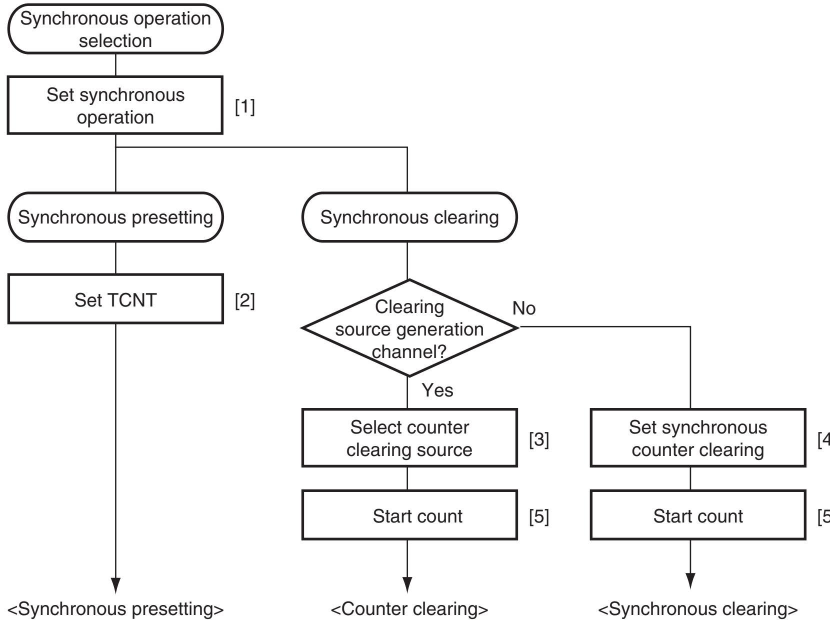


Figure 10.9 Example of Input Capture Operation

Figure 10.10 shows an example of the synchronous operation setting procedure.



[1] Set the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation t

[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 the CST bits in TSTR for the relevant channels to 1, to start the count operation.

Figure 10.10 Example of Synchronous Operation Setting Procedure

For details on PWM modes, see section 10.4.5, PWM Modes.

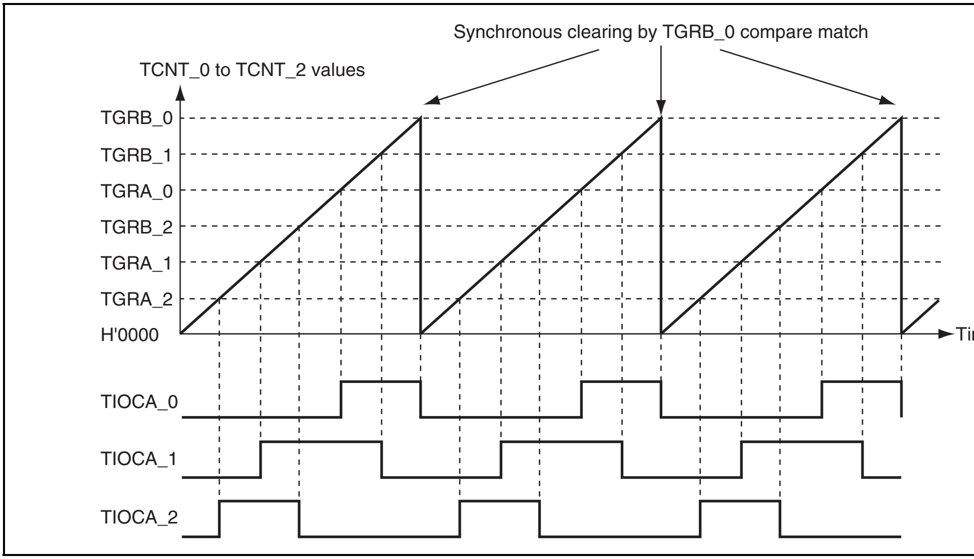


Figure 10.11 Example of Synchronous Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding
  transferred to the timer general register.

  This operation is illustrated in figure 10.12.

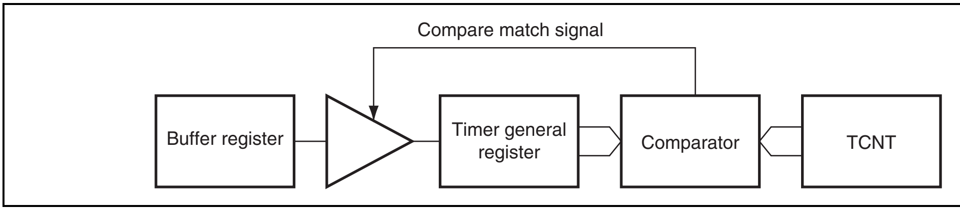
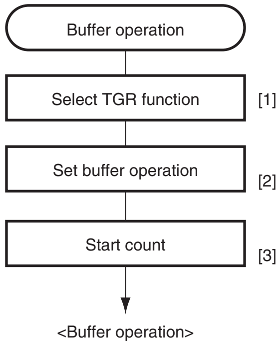


Figure 10.12 Compare Match Buffer Operation

Figure 10.13 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 10.14 shows an example of the buffer operation setting procedure.



[1] Designate TGR as an input capture register or output compare register by means of TIOR.

[2] Designate TGR for buffer operation with bits BFA and BFB in TMDR.

[3] Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.14 Example of Buffer Operation Setting Procedure

For details on PWM modes, see section 10.4.5, PWM Modes.

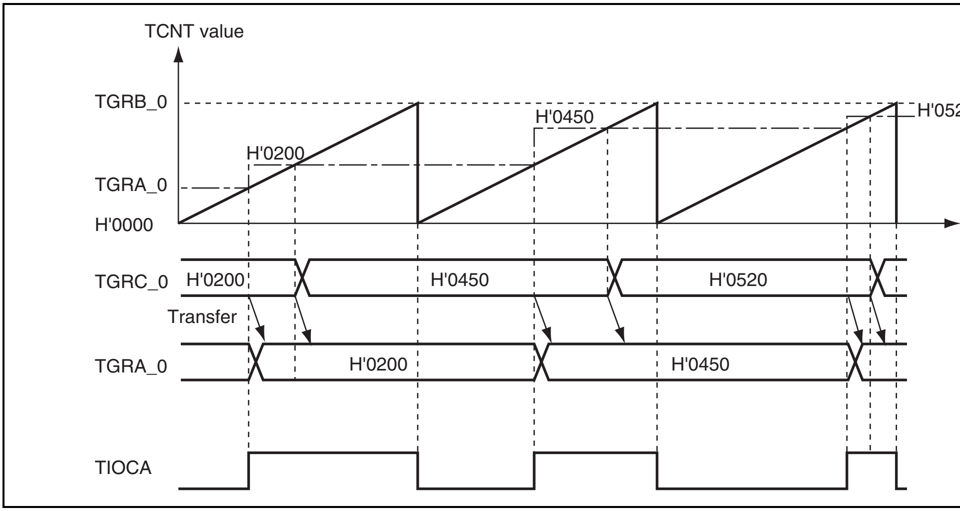


Figure 10.15 Example of Buffer Operation (1)

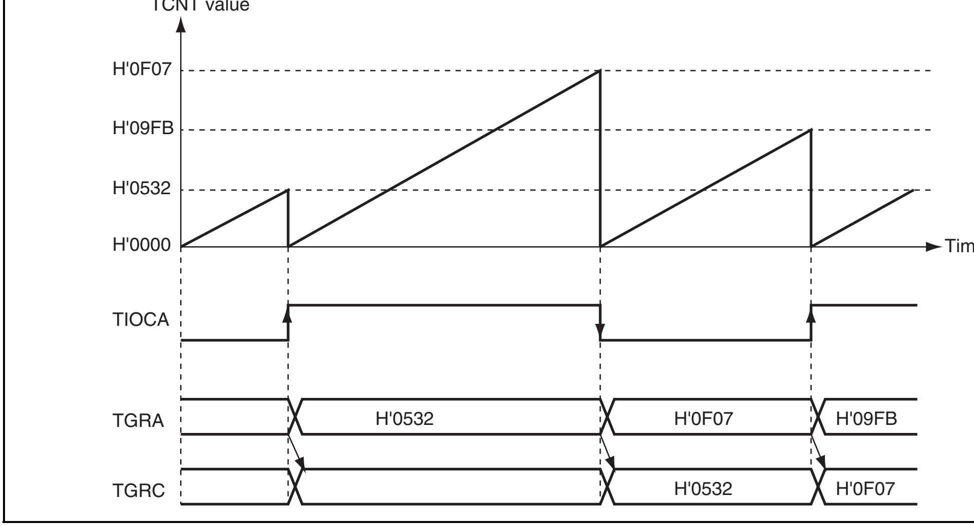


Figure 10.16 Example of Buffer Operation (2)

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is
and the counter operates independently in phase counting mode.

Table 10.30 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

(1) Example of Cascaded Operation Setting Procedure

Figure 10.17 shows an example of the setting procedure for cascaded operation.

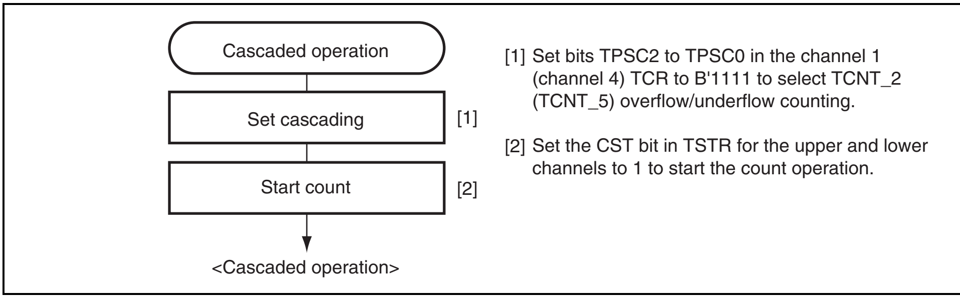


Figure 10.17 Example of Cascaded Operation Setting Procedure

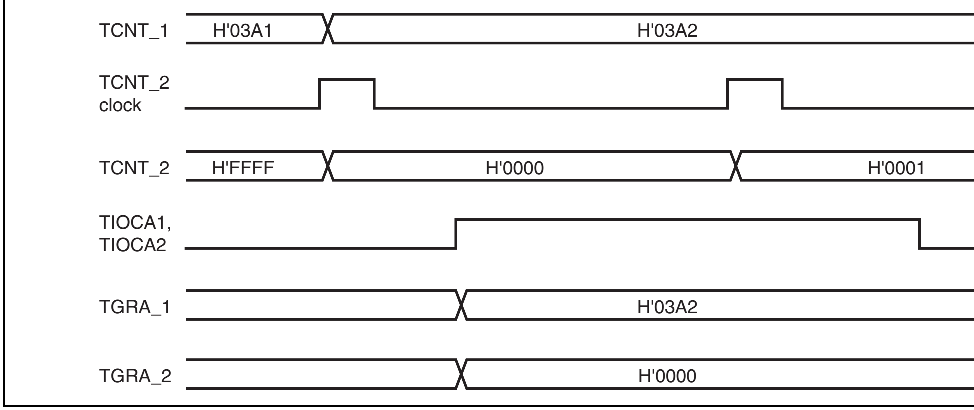


Figure 10.18 Example of Cascaded Operation (1)

Figure 10.19 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.

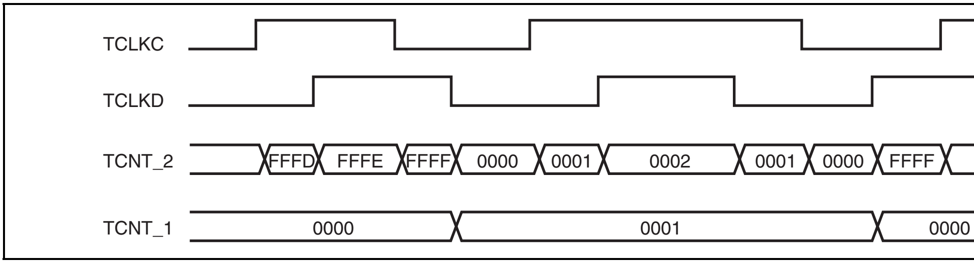


Figure 10.19 Example of Cascaded Operation (2)

There are two PWM modes, as described below.

1. PWM mode 1

   PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with T
   TGRC with TGRD. The outputs specified by bits IOA3 to IOA0 and IOC3 to IOC0
   are output from the TIOCA and TIOCC pins at compare matches A and C, respectiv
   outputs specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR are output at co
   matches B and D, respectively. The initial output value is the value set in TGRA or T
   the set values of paired TGRs are identical, the output value does not change when a
   match occurs.

   In PWM mode 1, a maximum 8-phase PWM output is possible.

2. PWM mode 2

   PWM output is generated using one TGR as the cycle register and the others as duty
   registers. The output specified in TIOR is performed by means of compare matches.
   counter clearing by a cycle register compare match, the output value of each pin is th
   value set in TIOR. If the set values of the cycle and duty cycle registers are identical
   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 w
   synchronous operation.

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 cyc

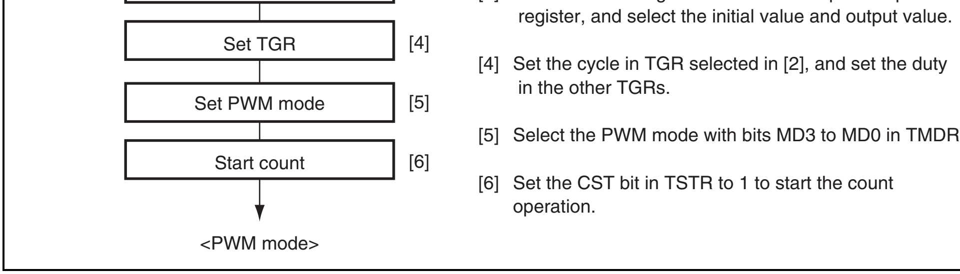


Figure 10.20 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 10.21 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 th 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 value set in TGRB regis duty cycle.

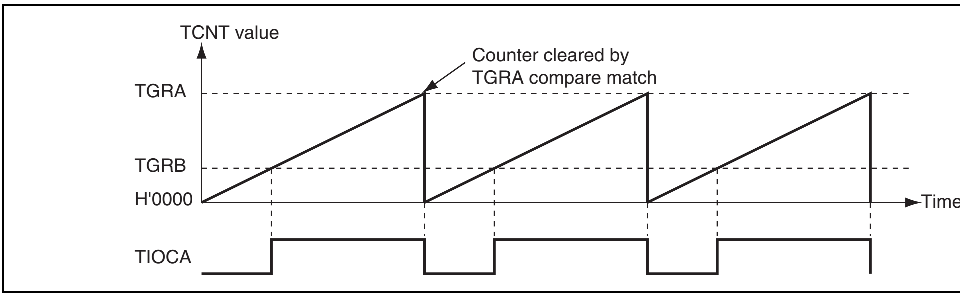


Figure 10.21 Example of PWM Mode Operation (1)

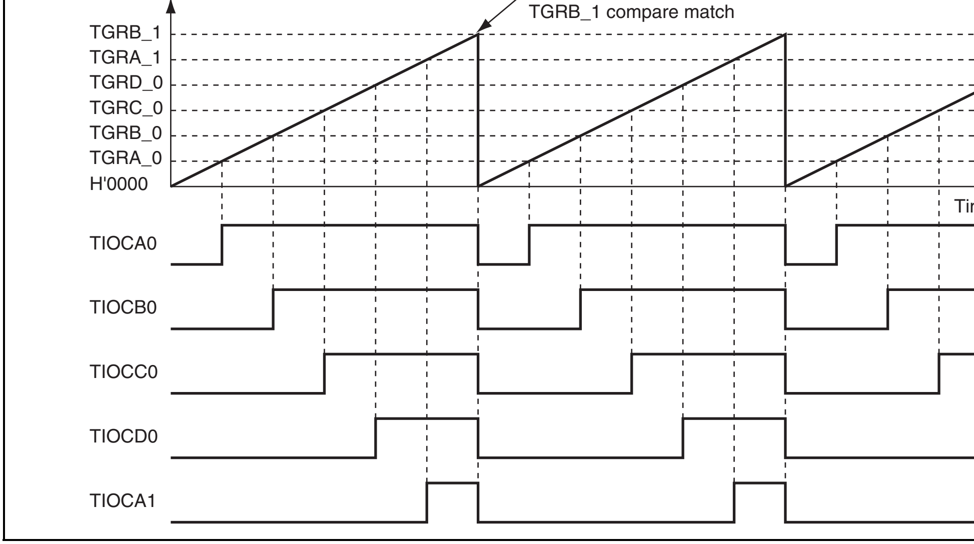


Figure 10.22 Example of PWM Mode Operation (2)

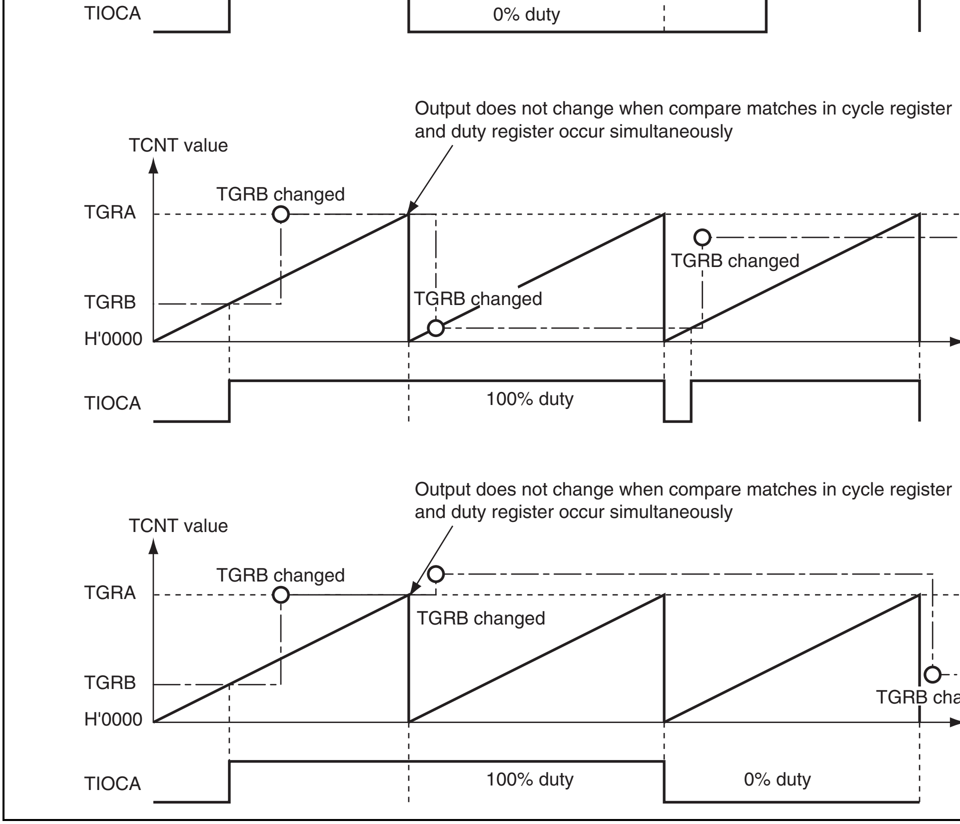


Figure 10.23 Example of PWM Mode Operation (3)

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 un
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 ind
whether TCNT is counting up or down.

Table 10.32 shows the correspondence between external clock pins and channels.

Table 10.32 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

Figure 10.24 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT counts up or down according to the phase difference bet external clocks. There are four modes, according to the count conditions.

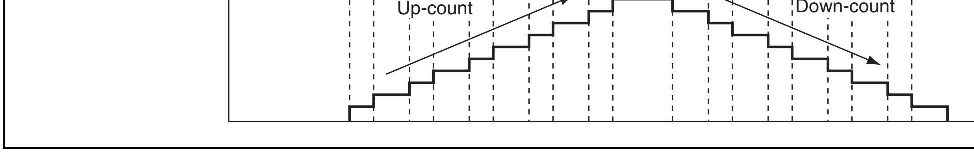


Figure 10.25 Example of Phase Counting Mode 1 Operation

Table 10.33 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

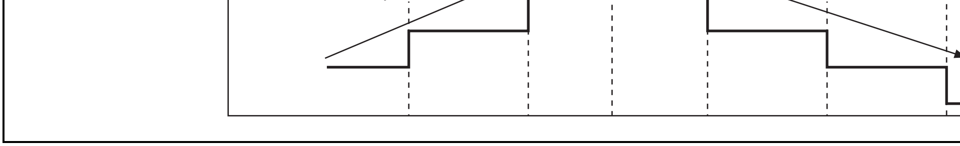


Figure 10.26 Example of Phase Counting Mode 2 Operation

Table 10.34 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

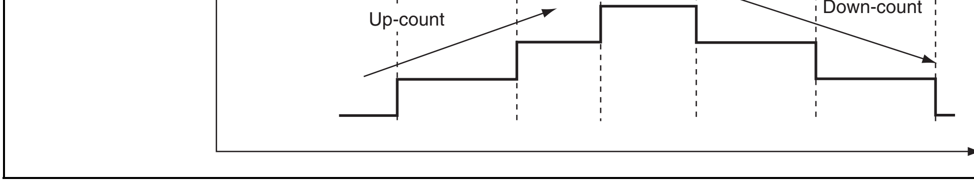


Figure 10.27 Example of Phase Counting Mode 3 Operation

Table 10.35 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

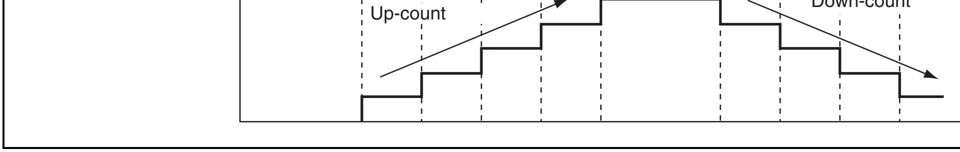


Figure 10.28 Example of Phase Counting Mode 4 Operation

Table 10.36 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
Low level	↑	
↑	High level	Don't care
↓	Low level	

[Legend]
↑ : Rising edge
↓ : Falling edge

position control cycle. TGRB_0 is used for input capture, with TGRB_0 and TGRD_0 op
in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input cap
source, and the pulse width of 2-phase encoder 4-multiplication pulses is detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, channel 0 TGRA_0
TGRC_0 compare matches are selected as the input capture source, and the up/down-cou
values for the control cycles are stored.

This procedure enables accurate position/speed detection to be achieved.

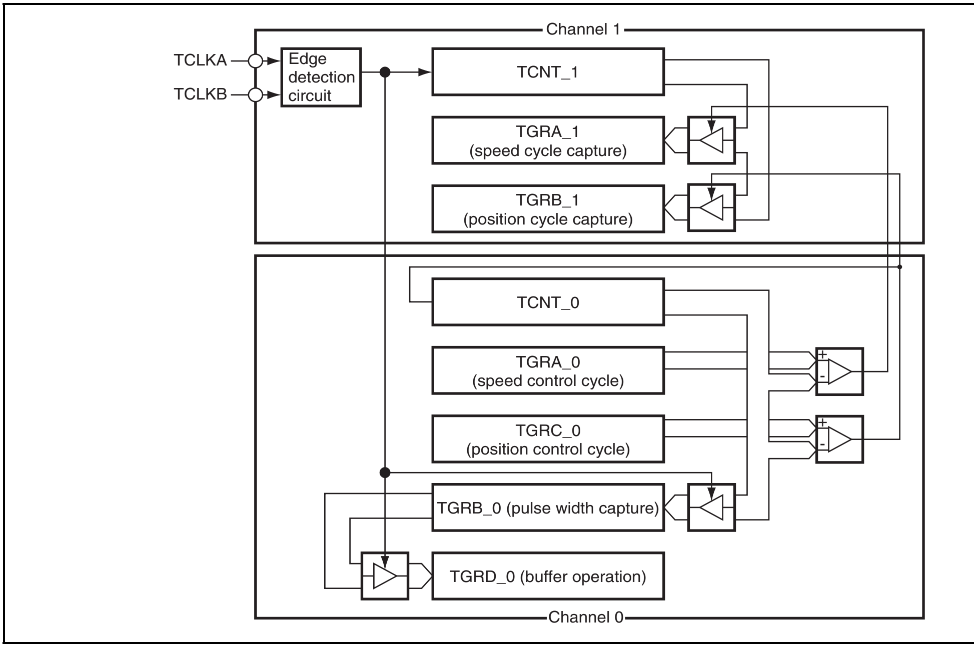


Figure 10.29 Phase Counting Mode Application Example

channel is fixed. For details, see section 5, Interrupt Controller.

Table 10.37 lists the TPU interrupt sources.

Table 10.37 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activa-tion
0	TGI0A	TGRA_0 input capture/compare match	TGFA_0	O
	TGI0B	TGRB_0 input capture/compare match	TGFB_0	O
	TGI0C	TGRC_0 input capture/compare match	TGFC_0	O
	TGI0D	TGRD_0 input capture/compare match	TGFD_0	O
	TCI0V	TCNT_0 overflow	TCFV_0	—
1	TGI1A	TGRA_1 input capture/compare match	TGFA_1	O
	TGI1B	TGRB_1 input capture/compare match	TGFB_1	O
	TCI1V	TCNT_1 overflow	TCFV_1	—
	TCI1U	TCNT_1 underflow	TCFU_1	—
2	TGI2A	TGRA_2 input capture/compare match	TGFA_2	O
	TGI2B	TGRB_2 input capture/compare match	TGFB_2	O
	TCI2V	TCNT_2 overflow	TCFV_2	—
	TCI2U	TCNT_2 underflow	TCFU_2	—
3	TGI3A	TGRA_3 input capture/compare match	TGFA_3	O
	TGI3B	TGRB_3 input capture/compare match	TGFB_3	O
	TGI3C	TGRC_3 input capture/compare match	TGFC_3	O
	TGI3D	TGRD_3 input capture/compare match	TGFD_3	O
	TCI3V	TCNT_3 overflow	TCFV_3	—

TCI5U	TCNT_5 underflow	TCFU_5	—

[Legend]

O : Possible

— : Not possible

Note: This table shows the initial state immediately after a reset. The relative channel pri
levels can be changed by the interrupt controller.

(1) 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
by the occurrence of a TGR input capture/compare match on a channel. The interrupt req
cleared by clearing the TGF flag to 0. The TPU has 16 input capture/compare match inte
four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

(2) Overflow Interrupt

An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TS
1 by the occurrence of a TCNT overflow on a channel. The interrupt request is cleared by
the TCFV flag to 0. The TPU has six overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TS
1 by the occurrence of a TCNT underflow on a channel. The interrupt request is cleared b
clearing the TCFU flag to 0. The TPU has four underflow interrupts, one each for channe
and 5.

The DMAC can be activated by the TGRA input capture/compare match interrupt for a
For details, see section 7, DMA Controller (DMAC).

A total of six TPU input capture/compare match interrupts can be used as DMAC activa
sources, one for each channel.

10.8 A/D Converter Activation

The TGRA input capture/compare match for each channel can activate the A/D convert

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurr
TGRA input capture/compare match on a particular channel, a request to start A/D conv
sent to the A/D converter. If the TPU conversion start trigger has been selected on the A
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
converter conversion start sources, one for each channel.

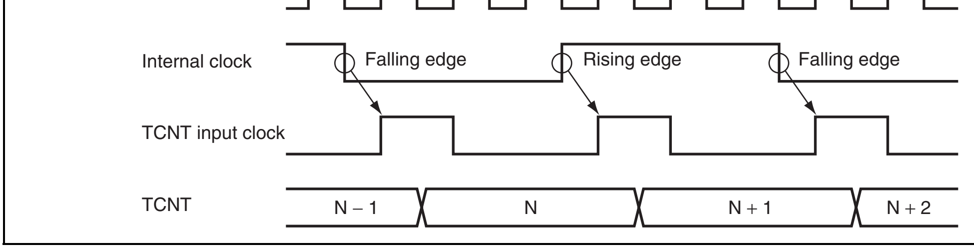


Figure 10.30 Count Timing in Internal Clock Operation

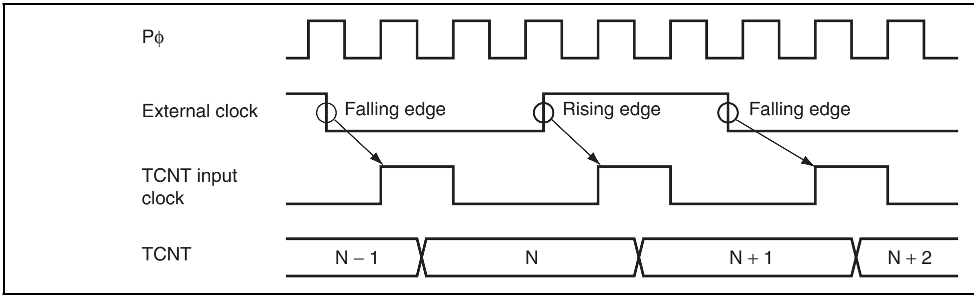


Figure 10.31 Count Timing in External Clock Operation

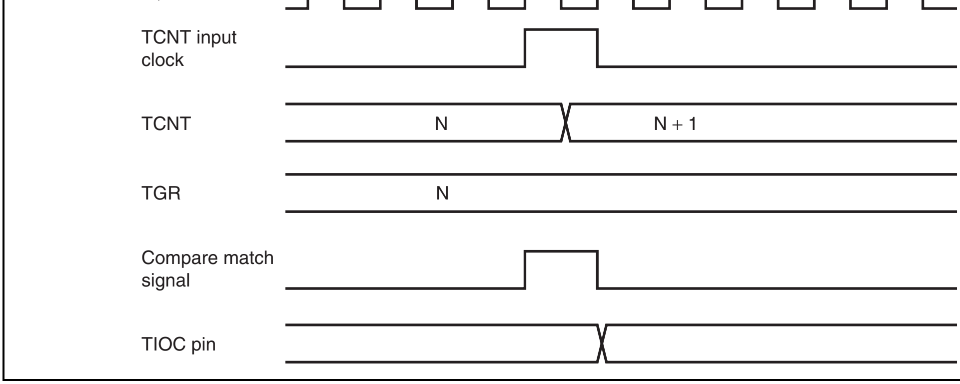


Figure 10.32 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 10.33 shows input capture signal timing.

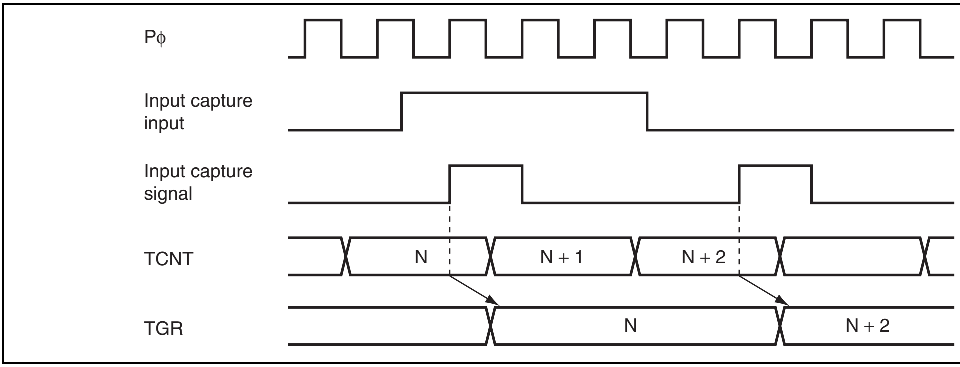


Figure 10.33 Input Capture Input Signal Timing

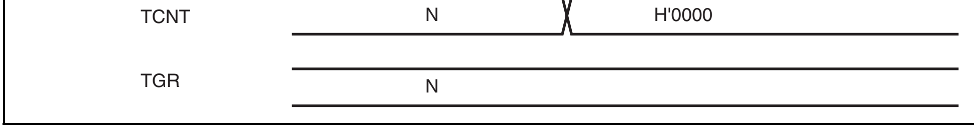


Figure 10.34 Counter Clear Timing (Compare Match)

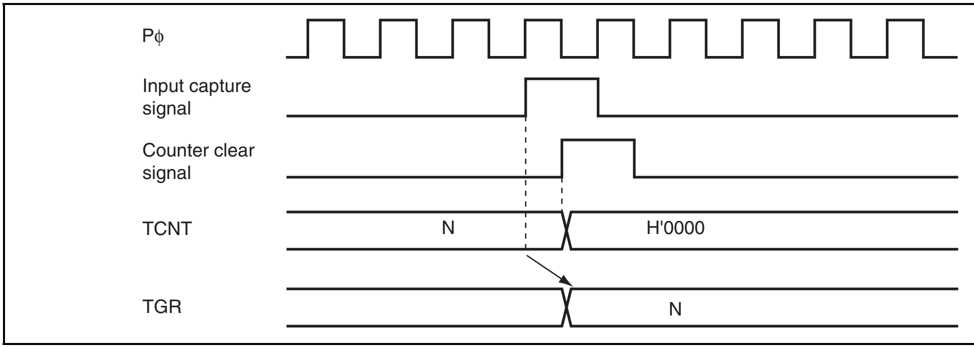


Figure 10.35 Counter Clear Timing (Input Capture)

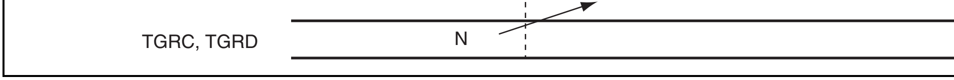


Figure 10.36 Buffer Operation Timing (Compare Match)

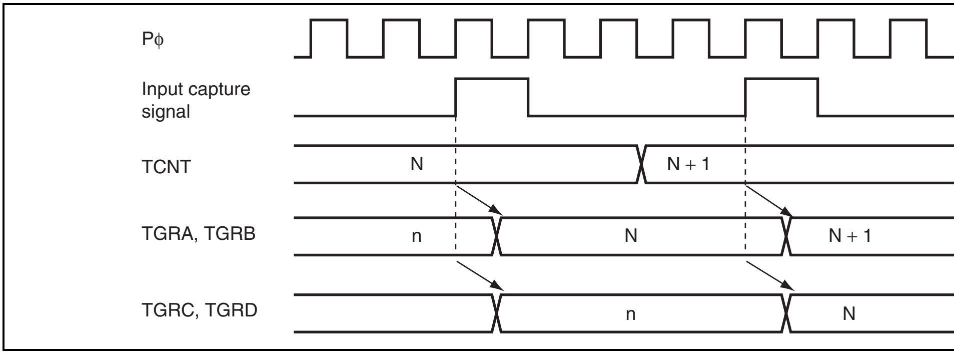


Figure 10.37 Buffer Operation Timing (Input Capture)

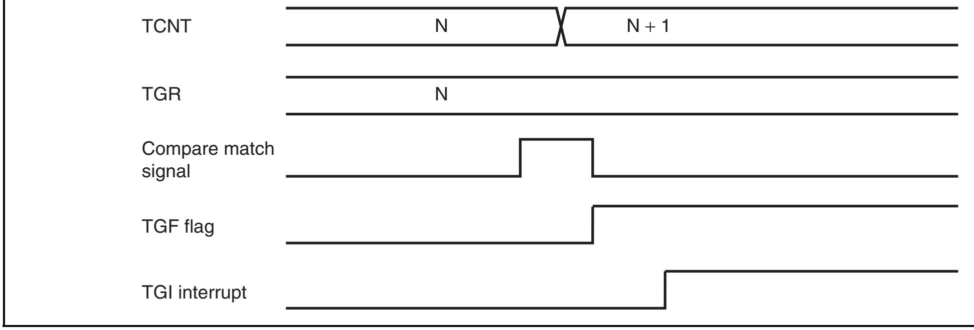


Figure 10.38 TGI Interrupt Timing (Compare Match)

(2) TGF Flag Setting Timing in Case of Input Capture

Figure 10.39 shows the timing for setting of the TGF flag in TSR by input capture occurr
the TGI interrupt request signal timing.

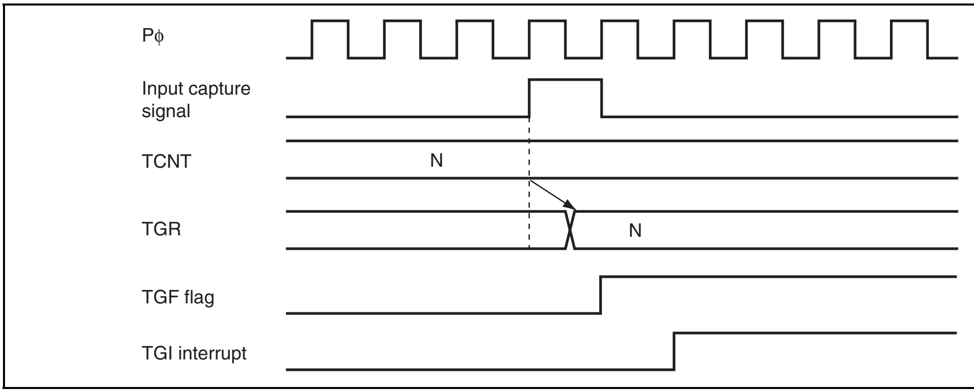


Figure 10.39 TGI Interrupt Timing (Input Capture)

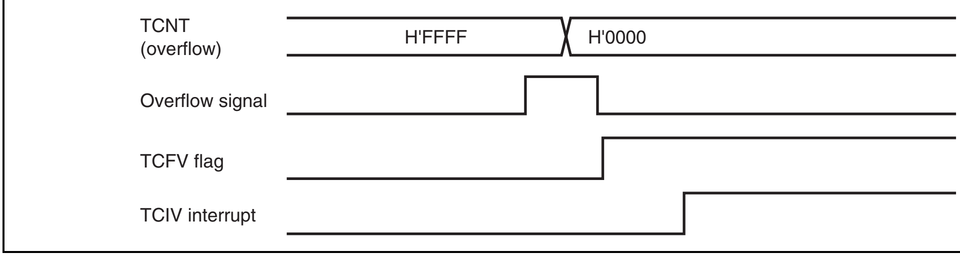


Figure 10.40 TCIV Interrupt Setting Timing

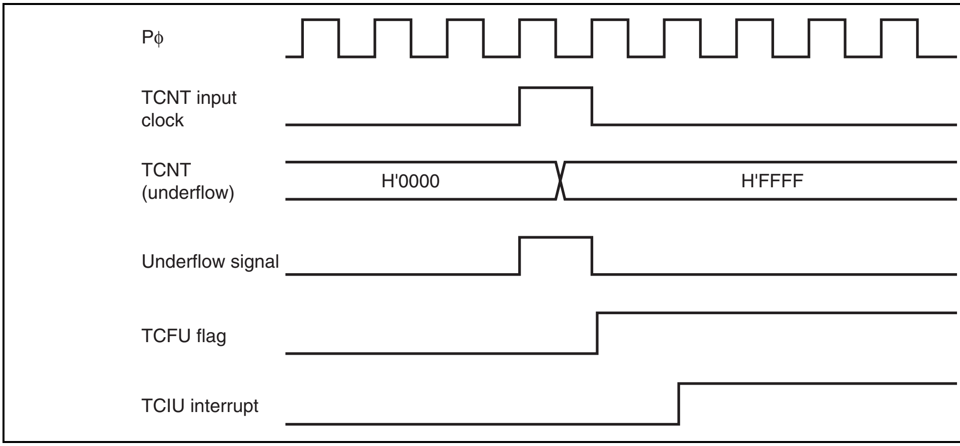


Figure 10.41 TCIU Interrupt Setting Timing

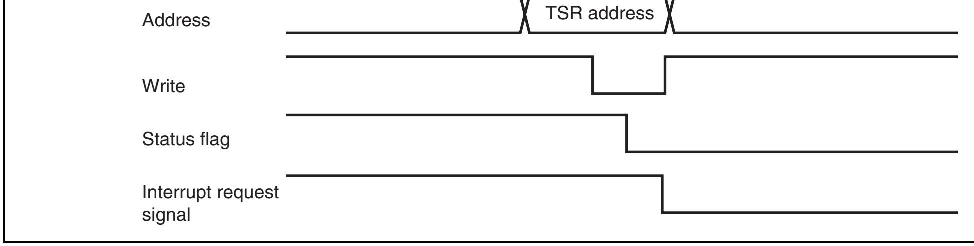


Figure 10.42 Timing for Status Flag Clearing by CPU

The status flag and interrupt request signal are cleared in synchronization with Pϕ after th DMAC transfer has started, as shown in figure 10.43. If conflict occurs for clearing the st and interrupt request signal due to activation of multiple DTC or DMAC transfers, it will to five clock cycles (Pϕ) for clearing them, as shown in figure 10.44. The next transfer re masked for a longer period of either a period until the current transfer ends or a period fo clock cycles (Pϕ) from the beginning of the transfer. Note that in the DTC transfer, the st may be cleared during outputting the destination address.

Figure 10.43 Timing for Status Flag Clearing by DTC or DMAC Activation

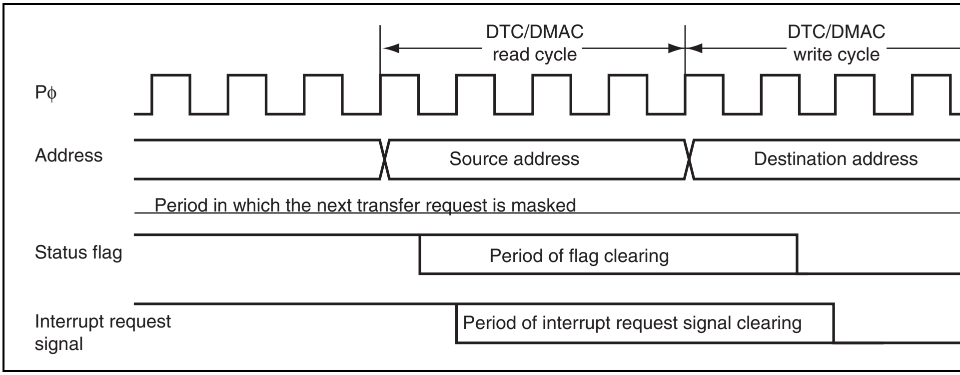


Figure 10.44 Timing for Status Flag Clearing by DTC or DMAC Activation

The input clock pulse width must be at least 1.5 states in the case of single-edge detectio
least 2.5 states in the case of both-edge detection. The TPU will not operate properly with
narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks n
least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.45 shows the inp
conditions in phase counting mode.

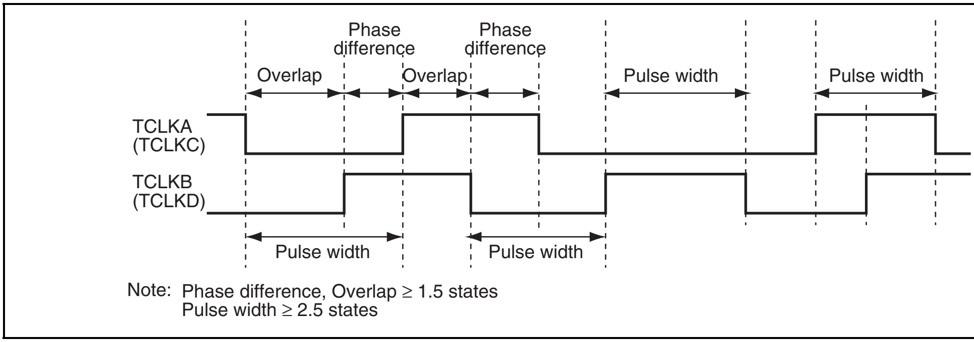


Figure 10.45 Phase Difference, Overlap, and Pulse Width in Phase Counting M

N: TGR set value

10.10.4 Conflict between TCNT Write and Clear Operations

If the counter clearing signal is generated in the T2 state of a TCNT write cycle, TCNT
takes precedence and the TCNT write is not performed. Figure 10.46 shows the timing i
case.

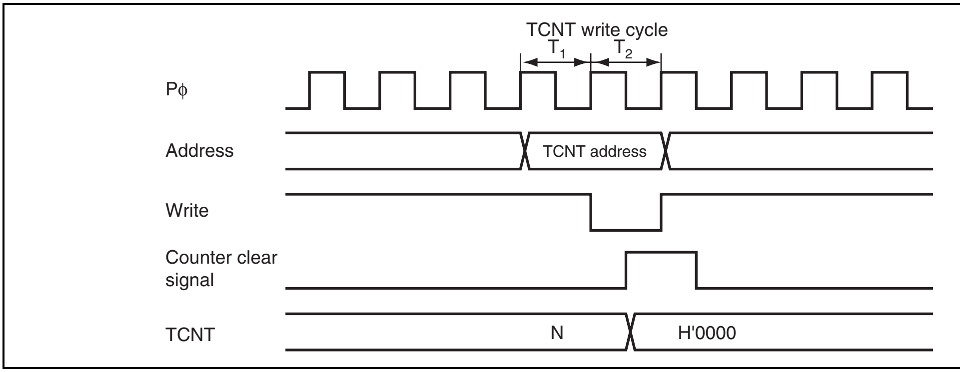


Figure 10.46 Conflict between TCNT Write and Clear Operations

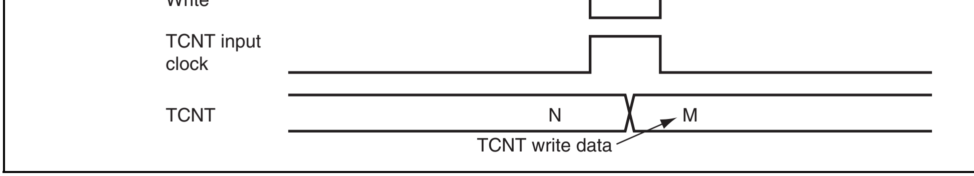


Figure 10.47 Conflict between TCNT Write and Increment Operations

10.10.6 Conflict between TGR Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write takes prec and the compare match signal is disabled. A compare match also does not occur when the value as before is written.

Figure 10.48 shows the timing in this case.

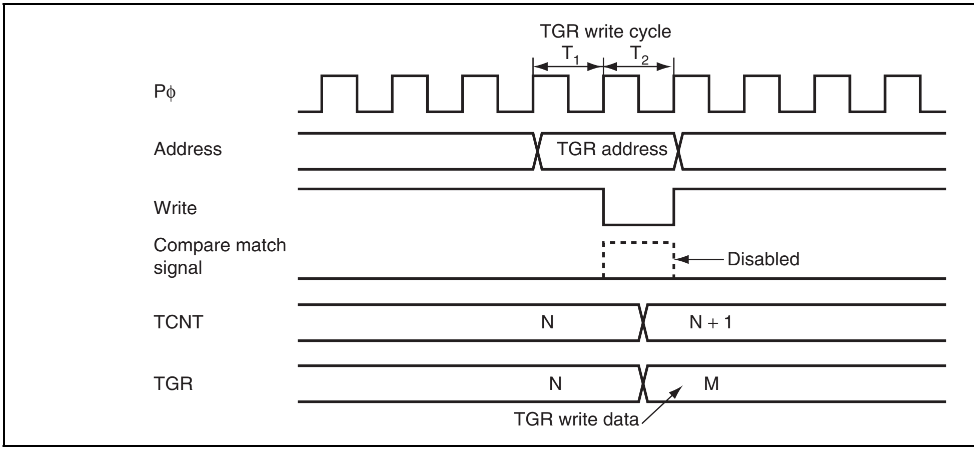


Figure 10.48 Conflict between TGR Write and Compare Match

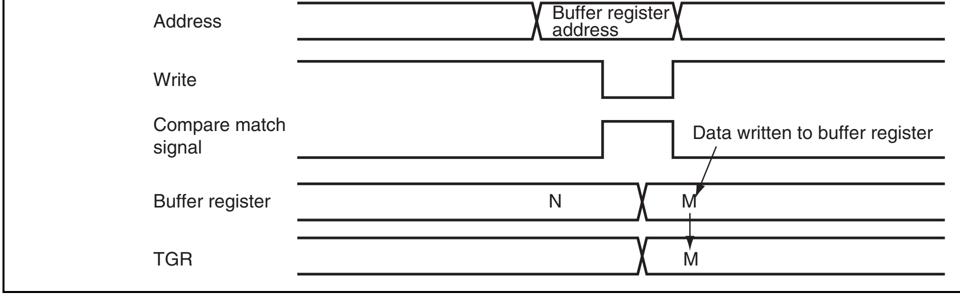


Figure 10.49 Conflict between Buffer Register Write and Compare Matc

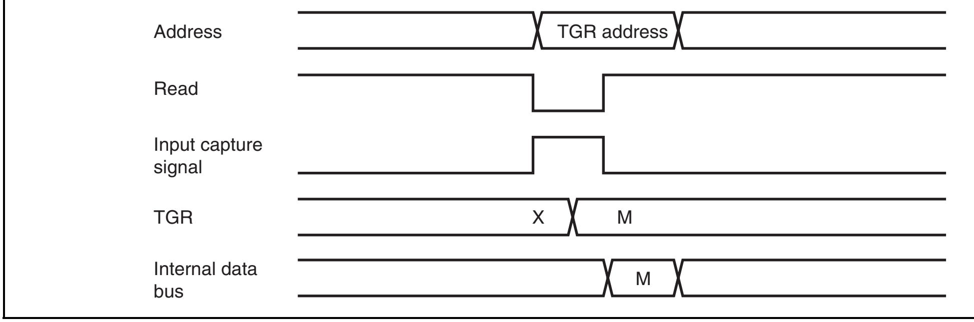


Figure 10.50 Conflict between TGR Read and Input Capture

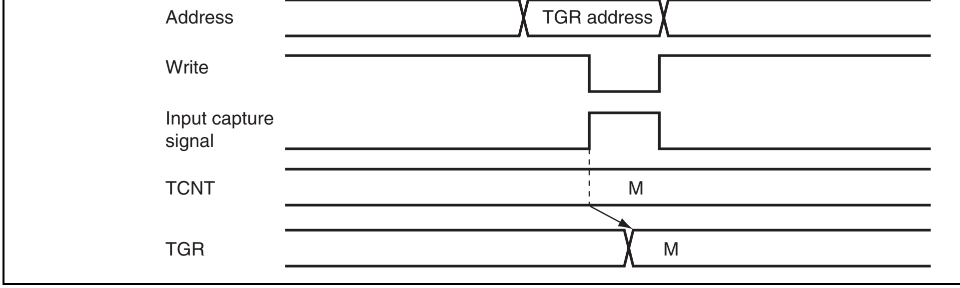


Figure 10.51 Conflict between TGR Write and Input Capture

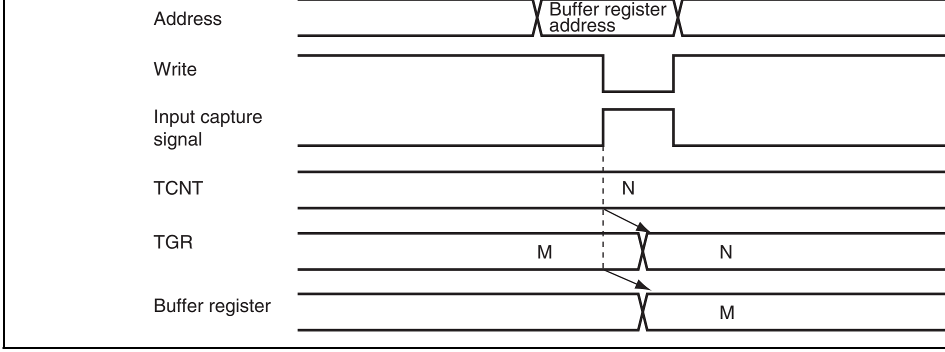


Figure 10.52 Conflict between Buffer Register Write and Input Capture

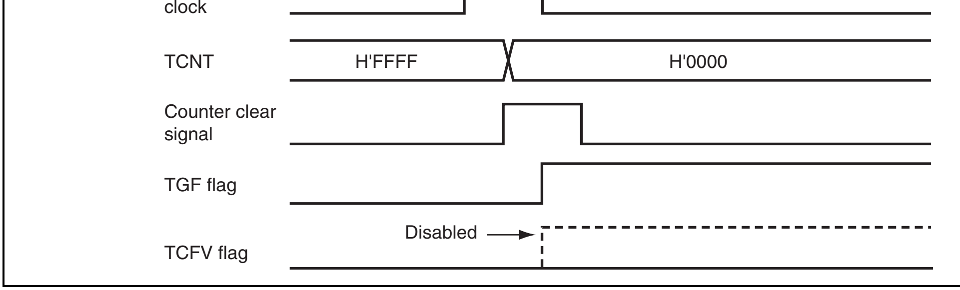


Figure 10.53 Conflict between Overflow and Counter Clearing

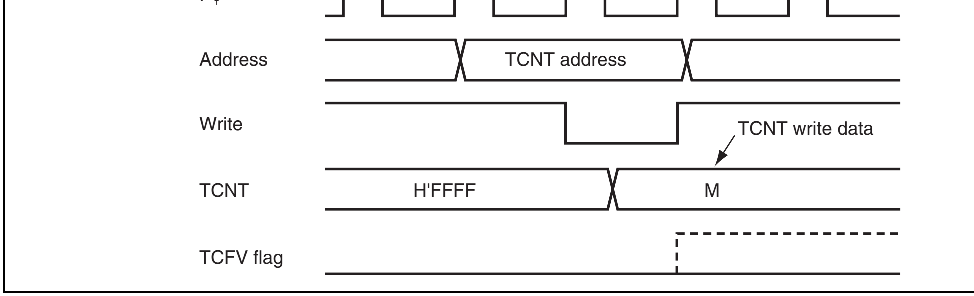


Figure 10.54 Conflict between TCNT Write and Overflow

10.10.13 Multiplexing of I/O Pins

In this LSI, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB in with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLK pin with the TIOCB2 I/O pin. When an external clock is input, compare match output sho be performed from a multiplexed pin.

10.10.14 Interrupts and Module Stop Mode

If module stop state is entered when an interrupt has been requested, it will not be possibl the CPU interrupt source or the DMAC or DTC activation source. Interrupts should there disabled before entering module stop state.

- Four output groups
- Selectable output trigger signals
- Non-overlapping mode
- Can operate together with the data transfer controller (DTC) and DMA controller (D
- Inverted output can be set
- Module stop state specifiable

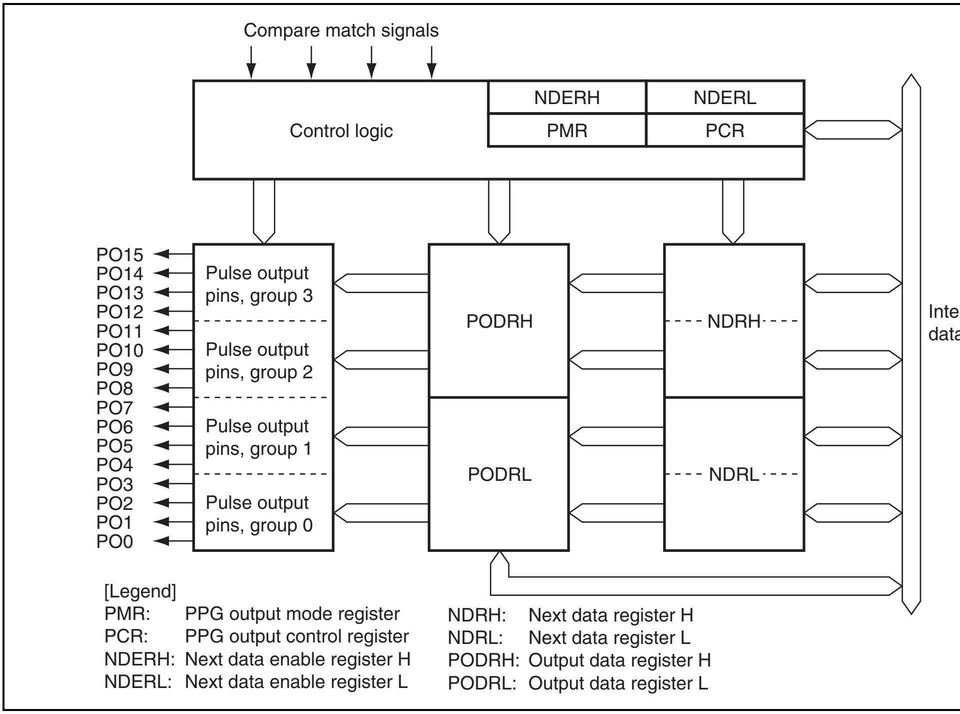


Figure 11.1 Block Diagram of PPG

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	

- PPG output control register (PCR)
- PPG output mode register (PMR)

11.3.1 Next Data Enable Registers H, L (NDERH, NDERL)

NDERH and NDERL enable/disable pulse output on a bit-by-bit basis.

- NDERH

Bit	7	6	5	4	3	2	1
Bit Name	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- NDERL

Bit	7	6	5	4	3	2	1
Bit Name	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 correspon NDRL bit is transferred to the PODRL bit by the output trigger. Values are not transferred from N 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	

- PODRL

Bit	7	6	5	4	3	2	1
Bit Name	POD7	POD6	POD5	POD4	POD3	POD2	POD1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- 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
5	POD13	0	R/W	the output trigger transfers NDRH values to this
4	POD12	0	R/W	during PPG operation. While NDERH is set to 1
3	POD11	0	R/W	cannot write to this register. While NDERH is cl
2	POD10	0	R/W	initial output value of the pulse can be set.
1	POD9	0	R/W	
0	POD8	0	R/W	

0	POD0	0	R/W

11.3.3 Next Data Registers H, L (NDRH, NDRL)

NDRH and NDRL store the next data for pulse output. The NDR addresses differ depend
whether pulse output groups have the same output trigger or different output triggers.

- NDRH

Bit	7	6	5	4	3	2	1
Bit Name	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- NDRL

Bit	7	6	5	4	3	2	1
Bit Name	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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, the upper four bits and
bits are mapped to 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 trigge 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

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be
3	NDR11	0	R/W	Next Data Register 11 to 8
2	NDR10	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigge with PCR.
1	NDR9	0	R/W	
0	NDR8	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, the upper four bits and lo
bits are mapped to different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 7 to 4
6	NDR6	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger with PCR.
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 r

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be r
3	NDR3	0	R/W	Next Data Register 3 to 0
2	NDR2	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger with PCR.
1	NDR1	0	R/W	
0	NDR0	0	R/W	

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	These bits select output trigger of pulse output 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	These bits select output trigger of pulse output 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	These bits select output trigger of pulse output 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
1	G0CMS1	1	R/W	Group 0 Compare Match Select 1 and 0
0	G0CMS0	1	R/W	These bits select output trigger of pulse output 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

Initial Value	1	1	1	1	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	G3INV	1	R/W	Group 3 Inversion Selects direct output or inverted output for pulse group 3. 0: Inverted output 1: Direct output
6	G2INV	1	R/W	Group 2 Inversion Selects direct output or inverted output for pulse group 2. 0: Inverted output 1: Direct output
5	G1INV	1	R/W	Group 1 Inversion Selects direct output or inverted output for pulse group 1. 0: Inverted output 1: Direct output
4	G0INV	1	R/W	Group 0 Inversion Selects direct output or inverted output for pulse group 0. 0: Inverted output 1: Direct output

				Selects normal or non-overlapping operation fo output group 2. 0: Normal operation (output values updated at c match A in the selected TPU channel) 1: Non-overlapping operation (output values up compare match A or B in the selected TPU c
1	G1NOV	0	R/W	Group 1 Non-Overlap Selects normal or non-overlapping operation fo output group 1. 0: Normal operation (output values updated at c match A in the selected TPU channel) 1: Non-overlapping operation (output values up compare match A or B in the selected TPU c
0	G0NOV	0	R/W	Group 0 Non-Overlap Selects normal or non-overlapping operation fo output group 0. 0: Normal operation (output values updated at c match A in the selected TPU channel) 1: Non-overlapping operation (output values up compare match A or B in the selected TPU c

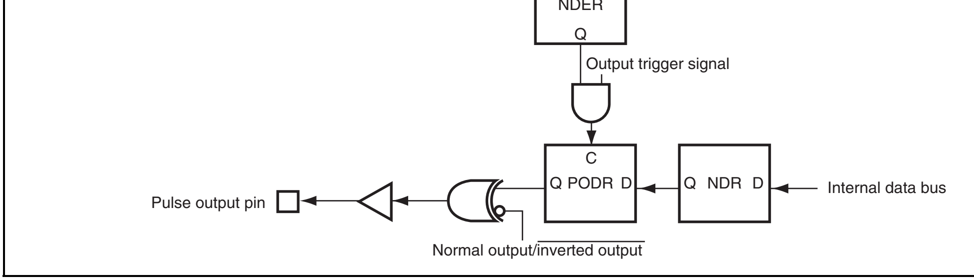


Figure 11.2 Schematic Diagram of PPG

11.4.1 Output Timing

If pulse output is enabled, the NDR contents are transferred to PODR and output when th specified compare match event occurs. Figure 11.3 shows the timing of these operations f case of normal output in groups 2 and 3, triggered by compare match A.

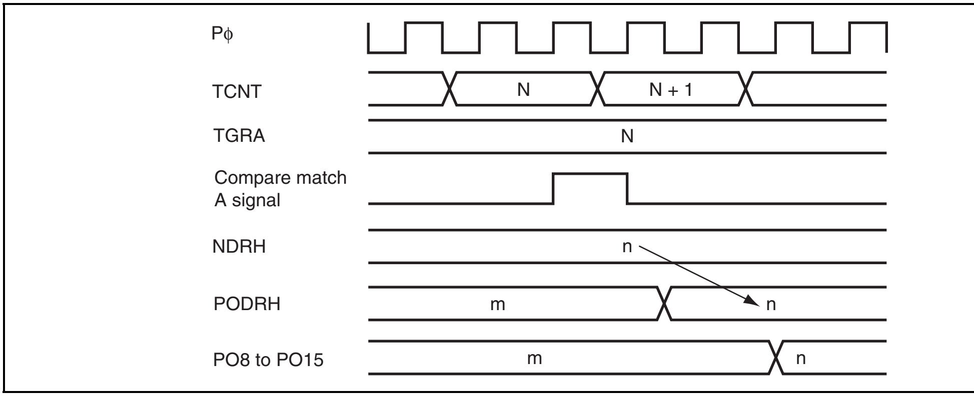
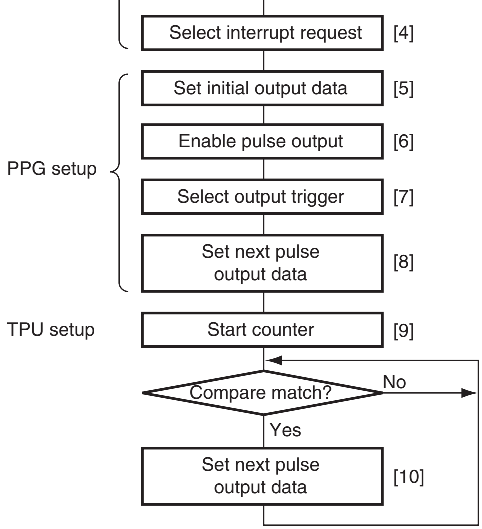


Figure 11.3 Timing of Transfer and Output of NDR Contents (Example)



[4] Enable the TGIA interrupt in TIER. Th
DTC or DMAC can also be set up to transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the bits in NDER for the pins to be used for pulse output to 1.

[7] Select the TPU compare match event be used as the output trigger in PCR.

[8] Set the next pulse output values in ND

[9] Set the CST bit in TSTR to 1 to start th
TCNT counter.

[10] At each TGIA interrupt, set the next output values in NDR.

Figure 11.4 Setup Procedure for Normal Pulse Output (Example)

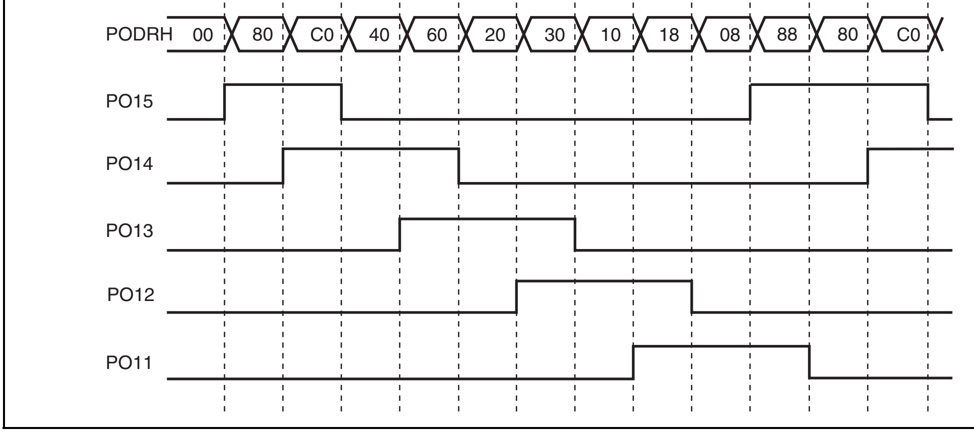


Figure 11.5 Normal Pulse Output Example (5-Phase Pulse Output)

1. Set up TGRA in TPU which is used as the output trigger to be an output compare regi a cycle in TGRA so the counter will be cleared by compare match A. Set the TGIEA TIER to 1 to enable the compare match/input capture A (TGIA) interrupt.
2. Write H'F8 to NDERH, and set bits G3CMS1, G3CMS0, G2CMS1, and G2CMS0 in select compare match in the TPU channel set up in the previous step to be the output Write output data H'80 in NDRH.
3. The timer counter in the TPU channel starts. When compare match A occurs, the ND contents are transferred to PODRH and output. The TGIA interrupt handling routine next output data (H'C0) in NDRH.
4. 5-phase pulse output (one or two phases active at a time) can be obtained subsequentl writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88... at successive TGIA interrupt

   If the DTC or DMAC is set for activation by the TGIA interrupt, pulse output can be without imposing a load on the CPU.

RENESAS

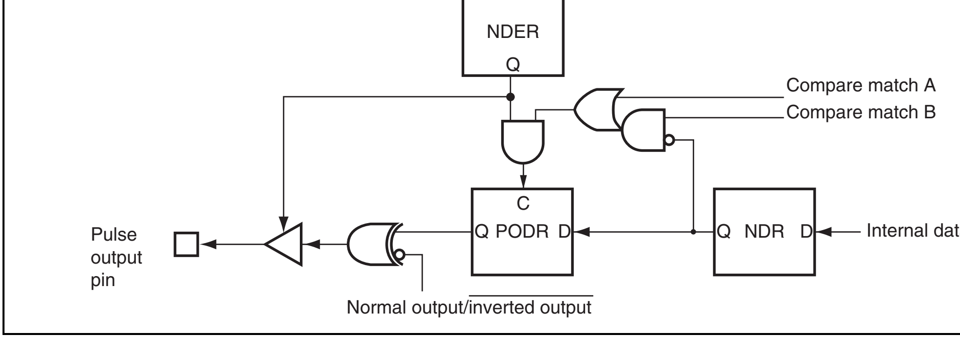


Figure 11.6 Non-Overlapping Pulse Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur b
compare match A.

The NDR contents should not be altered during the interval from compare match B to co
match A (the non-overlapping margin).

This can be accomplished by having the TGIA interrupt handling routine write the next
NDR, or by having the TGIA interrupt activate the DTC or DMAC. Note, however, that
data must be written before the next compare match B occurs.

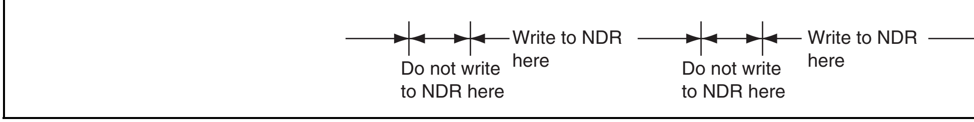
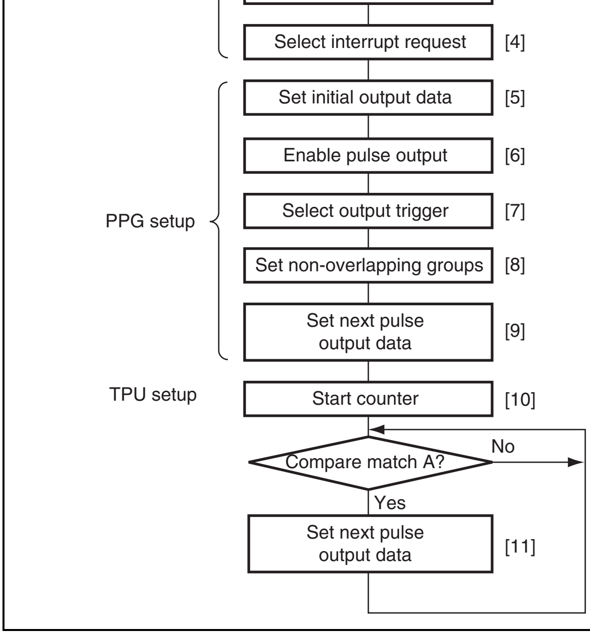


Figure 11.7 Non-Overlapping Operation and NDR Write Timing



counter clear source with bits CCLR1 and CCLR0.

[4] Enable the TGIA interrupt in TIER. Th DTC or DMAC can also be set up to transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the bits in NDER for the pins to be used for pulse output to 1.

[7] Select the TPU compare match event be used as the pulse output trigger in PCR.

[8] In PMR, select the groups that will operate in non-overlapping mode.

[9] Set the next pulse output values in ND

[10] Set the CST bit in TSTR to 1 to start t TCNT counter.

[11] At each TGIA interrupt, set the next output values in NDR.

Figure 11.8 Setup Procedure for Non-Overlapping Pulse Output (Exampl

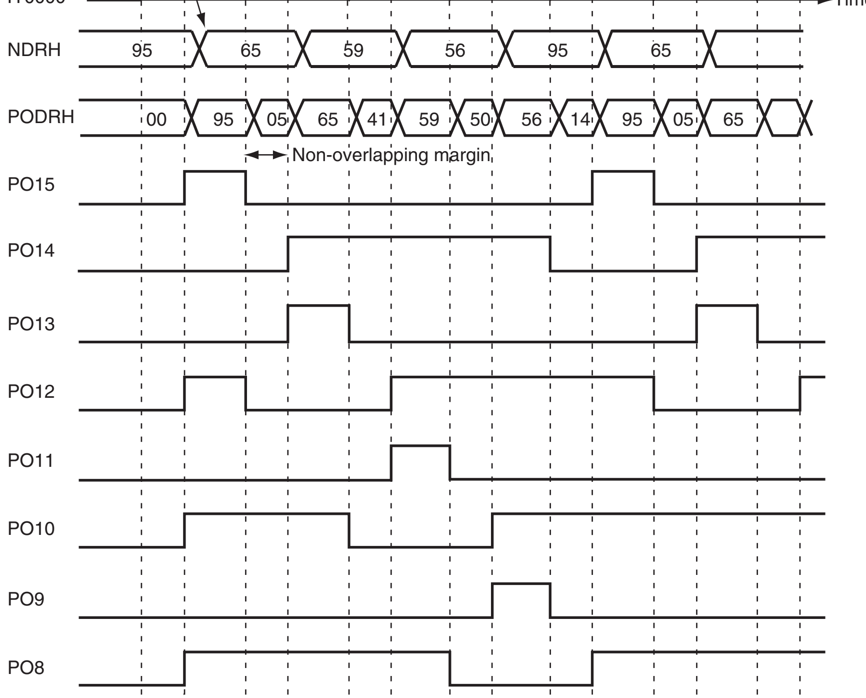


Figure 11.9 Non-Overlapping Pulse Output Example (4-Phase Complementa

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) to NDRH.

4. 4-phase complementary non-overlapping pulse output can be obtained subsequently H'59, H'56, H'95... at successive TGIA interrupts.

   If the DTC or DMAC is set for activation by a TGIA interrupt, pulse can be output v imposing a load on the CPU.

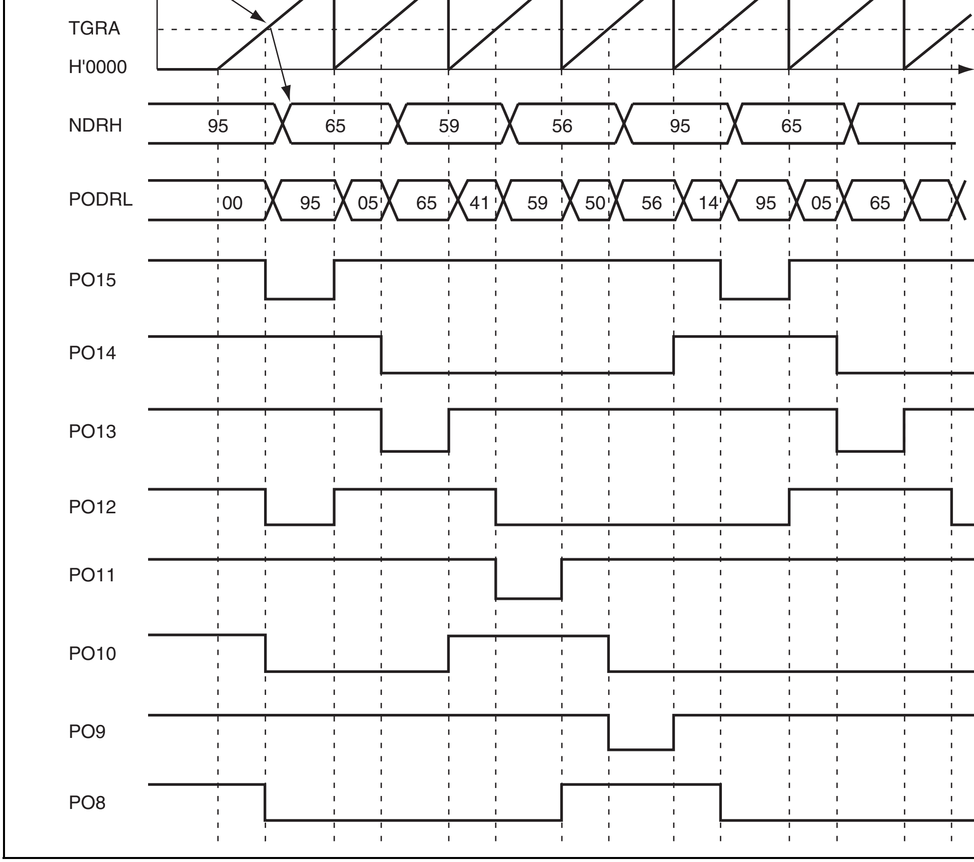


Figure 11.10 Inverted Pulse Output (Example)

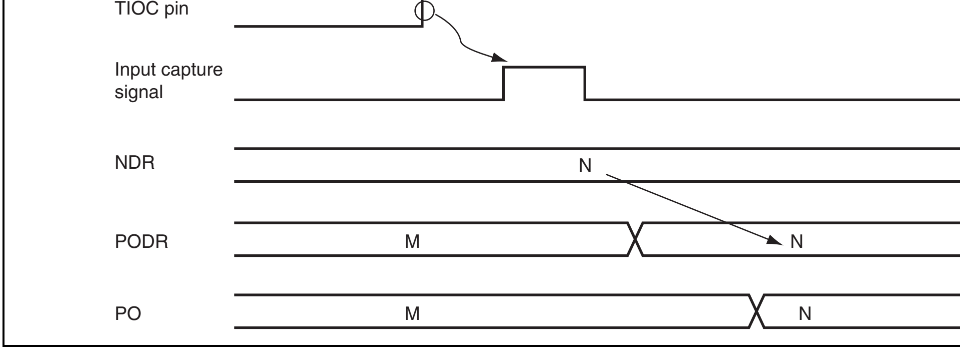


Figure 11.11 Pulse Output Triggered by Input Capture (Example)

Pins PO0 to PO15 are also used for other peripheral functions such as the TPU. When ou another peripheral function is enabled, the corresponding pins cannot be used for pulse ou Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of t of the pins.

Pin functions should be changed only under conditions in which the output trigger event v occur.

the same functions. Unit2 and unit 3 can generate baud rate clock for SCI and have the s
functions.

12.1 Features

- Selection of seven clock sources

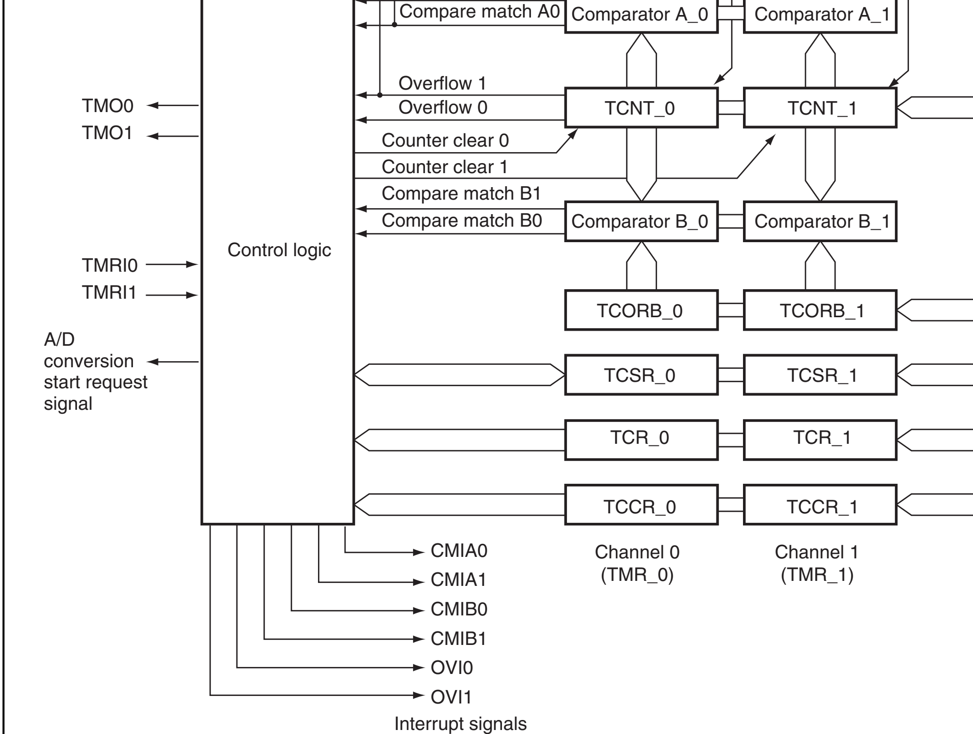
  The counters can be driven by one of six internal clock signals (Pϕ/2, Pϕ/8, Pϕ/32, P
  Pϕ/1024, or Pϕ/8192) or an external clock input (only internal clock available in unit
  Pϕ, Pϕ/2, Pϕ/8, Pϕ/32, Pϕ/64, Pϕ/1024, and Pϕ/8192).
- Selection of three ways to clear the counters

  The counters can be cleared on compare match A or B, or by an external reset signal
  available only in unit 0 and unit 1.)
- Timer output control by a combination of two compare match signals

  The timer output signal in each channel is controlled by a combination of two indepe
  compare match signals, enabling the timer to output pulses with a desired duty cycle
  output.
- Cascading of two channels

  Operation as a 16-bit timer is possible, using TMR_0 for the upper 8 bits and TMR_
  lower 8 bits (16-bit count mode).
  TMR_1 can be used to count TMR_0 compare matches (compare match count mode
- Three interrupt sources

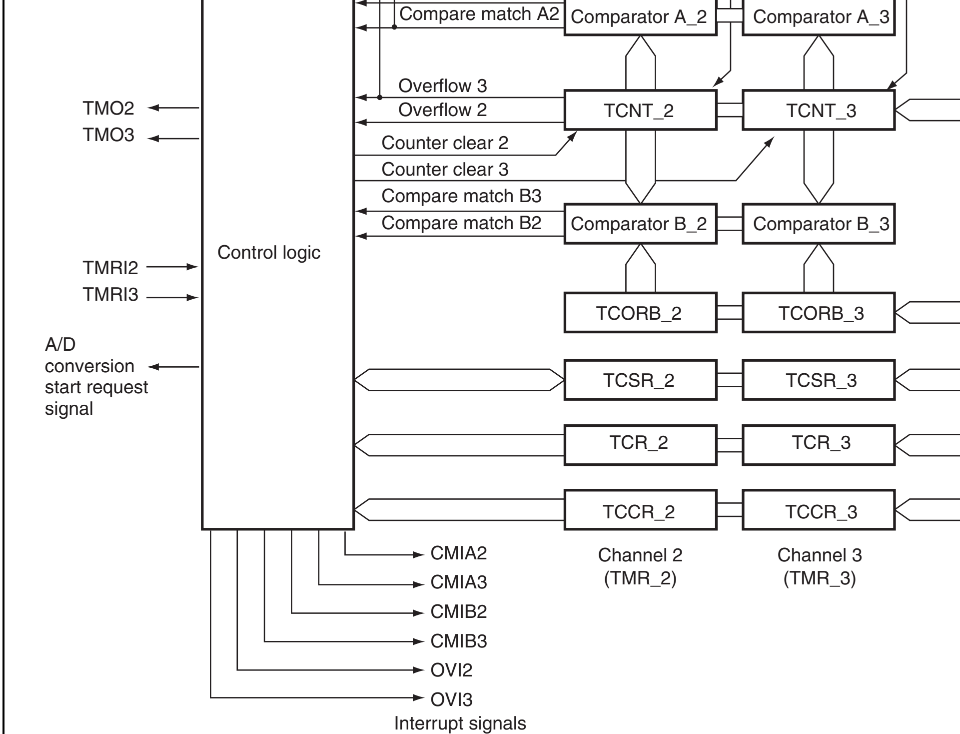
  Compare match A, compare match B, and overflow interrupts can be requested inde
  (This is available only in unit 0 and unit 1.)
- Generation of trigger to start A/D converter conversion (available in unit 0 and unit
- Capable of generating baud rate clock for SCI_5 and SCI_6. (This is available only i
  and unit 3). For details, see section 15, Serial Communication Interface (SCI, IrDA,
- Module stop state specifiable



[Legend]

TCORA_0:	Time constant register A_0	TCORA_1:	Time constant register A_1
TCNT_0:	Timer counter_0	TCNT_1:	Timer counter_1
TCORB_0:	Time constant register B_0	TCORB_1:	Time constant register B_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
TCCR_0:	Timer counter control register_0	TCCR_1:	Timer counter control register_1

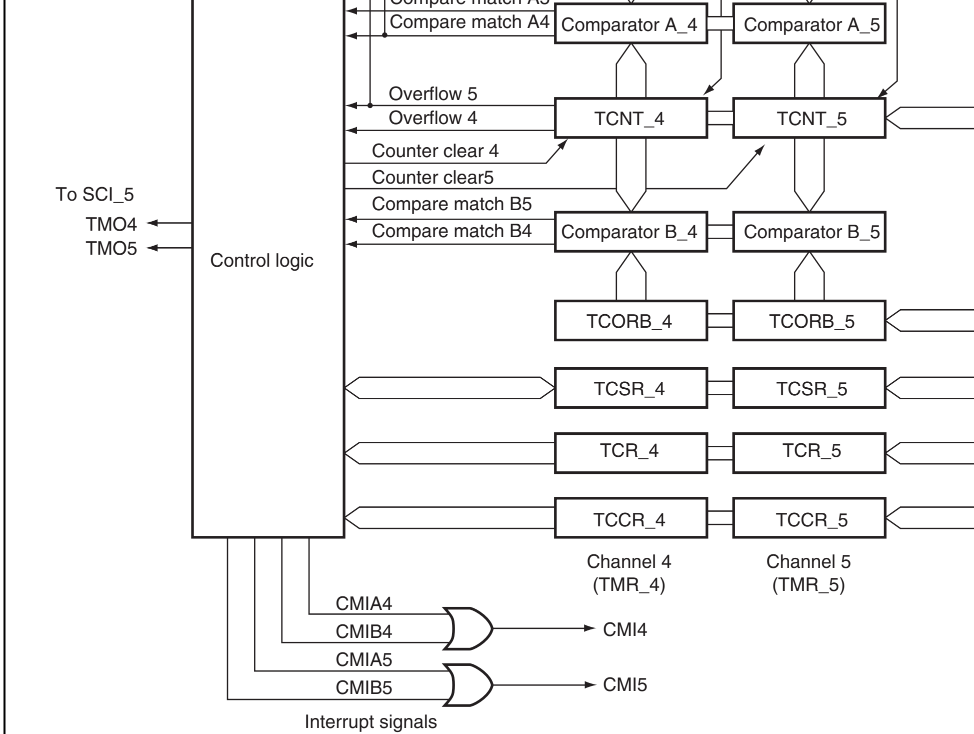
Figure 12.1 Block Diagram of 8-Bit Timer Module (Unit 0)



[Legend]

TCORA_2:	Time constant register A_2	TCORA_3:	Time constant register A_3
TCNT_2:	Timer counter_2	TCNT_3:	Timer counter_3
TCORB_2:	Time constant register B_2	TCORB_3:	Time constant register B_3
TCSR_2:	Timer control/status register_2	TCSR_3:	Timer control/status register_3
TCR_2:	Timer control register_2	TCR_3:	Timer control register_3
TCCR_2:	Timer counter control register_2	TCCR_3:	Timer counter control register_3

Figure 12.2 Block Diagram of 8-Bit Timer Module (Unit 1)

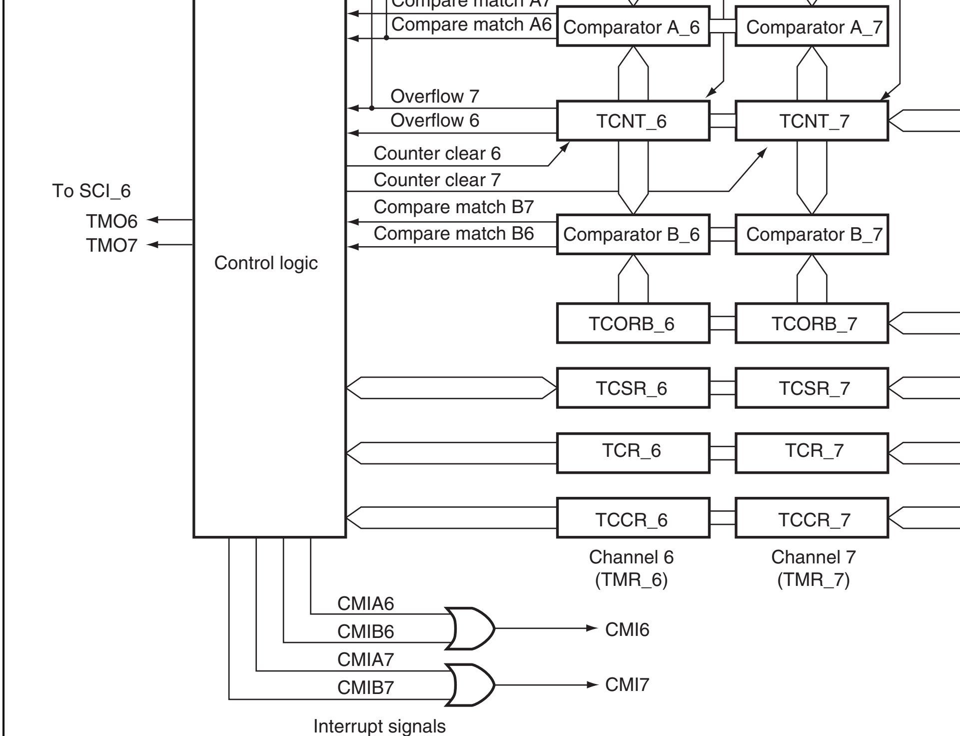


[Legend]

TCORA_4: Time constant register A_4
TCNT_4: Timer counter_4
TCORB_4: Time constant register B_4
TCSR_4: Timer control/status register_4
TCR_4: Timer control register_4
TCCR_4: Timer counter control register_4

TCORA_5: Time constant register A
TCNT_5: Timer counter_5
TCORB_5: Time constant register B
TCSR_5: Timer control/status reg
TCR_5: Timer control register_5
TCCR_5: Timer counter control re

Figure 12.3 Block Diagram of 8-Bit Timer Module (Unit 2)



[Legend]

TCORA_6:	Time constant register A_6	TCORA_7:	Time constant register
TCNT_6:	Timer counter_6	TCNT_7:	Timer counter_7
TCORB_6:	Time constant register B_6	TCORB_7:	Time constant register
TCSR_6:	Timer control/status register_6	TCSR_7:	Timer control/status re
TCR_6:	Timer control register_6	TCR_7:	Timer control register_
TCCR_6:	Timer counter control register_6	TCCR_7:	Timer counter control r

Figure 12.4 Block Diagram of 8-Bit Timer Module (Unit 3)

	1	Timer output pin	TMO1	Output	Outputs compare match
		Timer clock input pin	TMCI1	Input	Inputs external clock for co
		Timer reset input pin	TMRI1	Input	Inputs external reset to cou
1	2	Timer output pin	TMO2	Output	Outputs compare match
		Timer clock input pin	TMCI2	Input	Inputs external clock for co
		Timer reset input pin	TMRI2	Input	Inputs external reset to cou
	3	Timer output pin	TMO3	Output	Outputs compare match
		Timer clock input pin	TMCI3	Input	Inputs external clock for co
		Timer reset input pin	TMRI3	Input	Inputs external reset to cou
2	4	—	—	—	—
	5				
3	6				
	7				

— Timer counter control register_0 (TCCR_0)

— Timer control/status register_0 (TCSR_0)

- Channel 1 (TMR_1):

— 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 counter control register_1 (TCCR_1)

— Timer control/status register_1 (TCSR_1)

Unit 1:

- Channel 2 (TMR_2):

— Timer counter_2 (TCNT_2)

— Time constant register A_2 (TCORA_2)

— Time constant register B_2 (TCORB_2)

— Timer control register_2 (TCR_2)

— Timer counter control register_2 (TCCR_2)

— Timer control/status register_2 (TCSR_2)

- Channel 3 (TMR_3):

— Timer counter_3 (TCNT_3)

— Time constant register A_3 (TCORA_3)

— Time constant register B_3 (TCORB_3)

— Timer control register_3 (TCR_3)

— Timer counter control register_3 (TCCR_3)

— Timer control/status register_3 (TCSR_3)

— Timer counter_5 (TCNT_5)
— Time constant register A_5 (TCORA_5)
— Time constant register B_5 (TCORB_5)
— Timer control register_5 (TCR_5)
— Timer counter control register_5 (TCCR_5)
— Timer control/status register_5 (TCSR_5)

Unit 3:

- Channel 6 (TMR_6):
  — Timer counter_6 (TCNT_6)
  — Time constant register A_6 (TCORA_6)
  — Time constant register B_6 (TCORB_6)
  — Timer control register_6 (TCR_6)
  — Timer counter control register_6 (TCCR_6)
  — Timer control/status register_6 (TCSR_6)
- Channel 7 (TMR_7):
  — Timer counter_7 (TCNT_7)
  — Time constant register A_7 (TCORA_7)
  — Time constant register B_7 (TCORB_7)
  — Timer control register_7 (TCR_7)
  — Timer counter control register_7 (TCCR_7)
  — Timer control/status register_7 (TCSR_7)

Bit Name														
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.3.2 Time Constant Register A (TCORA)

TCORA is an 8-bit readable/writable register. TCORA_0 and TCORA_1 comprise a sin
register so they can be accessed together by a word transfer instruction. The value in TC
continually compared with the value in TCNT. When a match is detected, the correspon
CMFA flag in TCSR is set to 1. Note however that comparison is disabled during the T2
TCORA write cycle. The timer output from the TMO pin can be freely controlled by thi
match signal (compare match A) and the settings of bits OS1 and OS0 in TCSR. TCOR
initialized to H'FF.

	TCORA_0								TCORA_1					
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2
Bit Name														
Initial Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Name															
Initial Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/\

12.3.4 Timer Control Register (TCR)

TCR selects the TCNT clock source and the condition for clearing TCNT, and enables/di interrupt requests.

Bit	7	6	5	4	3	2	1
Bit Name	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare Match Interrupt Enable B Selects whether CMFB interrupt requests (CMIB enabled or disabled when the CMFB flag in TCS to 1. *2 0: CMFB interrupt requests (CMIB) are disabled 1: CMFB interrupt requests (CMIB) are enabled

				1:
				0: OVF interrupt requests (OVI) are disabled
				1: OVF interrupt requests (OVI) are enabled
4	CCLR1	0	R/W	Counter Clear 1 and 0*[1]
3	CCLR0	0	R/W	These bits select the method by which TCNT is 00: Clearing is disabled 01: Cleared by compare match A 10: Cleared by compare match B 11: Cleared at rising edge (TMRIS in TCCR is c 0) of the external reset input or when the ex reset input is high (TMRIS in TCCR is set to
2	CKS2	0	R/W	Clock Select 2 to 0*[1]
1	CKS1	0	R/W	These bits select the clock input to TCNT and c condition. See table 12.2.
0	CKS0	0	R/W	

Notes: 1. To use an external reset or external clock, the DDR and ICR bits in the corres pin should be set to 0 and 1, respectively. For details, see section 9, I/O Ports

2. In unit 2 and unit 3, one interrupt signal is used for CMIEB or CMIEA. For det section 12.7, Interrupt Sources.

3. Available only in unit 0 and unit 1

Bit	Bit Name	Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. It should not be
3	TMRIS	0	R/W	Timer Reset Input Select* Selects an external reset input when the CCLR1 CCLR0 bits in TCR are B'11. 0: Cleared at rising edge of the external reset 1: Cleared when the external reset is high
2	—	0	R	Reserved This bit is always read as 0. It should not be set
1	ICKS1	0	R/W	Internal Clock Select 1 and 0
0	ICKS0	0	R/W	These bits in combination with bits CKS2 to CKS select the internal clock. See table 12.2.

Note: * Available only in unit 0 and unit 1. The write value should always be 0 in unit 2 3.

				1	0	Uses internal clock. Counts at falling edge of
				1	1	Uses internal clock. Counts at falling edge of
	0	1	1	0	0	Uses internal clock. Counts at rising edge of
				0	1	Uses internal clock. Counts at rising edge of
				1	0	Uses internal clock. Counts at falling edge of
				1	1	Uses internal clock. Counts at falling edge of
	1	0	0	—	—	Counts at TCNT_1 overflow signal*1.
TMR_1	0	0	0	—	—	Clock input prohibited
	0	0	1	0	0	Uses internal clock. Counts at rising edge of
				0	1	Uses internal clock. Counts at rising edge of
				1	0	Uses internal clock. Counts at falling edge of
				1	1	Uses internal clock. Counts at falling edge of
	0	1	0	0	0	Uses internal clock. Counts at rising edge of
				0	1	Uses internal clock. Counts at rising edge of
				1	0	Uses internal clock. Counts at falling edge of
				1	1	Uses internal clock. Counts at falling edge of
	0	1	1	0	0	Uses internal clock. Counts at rising edge of
				0	1	Uses internal clock. Counts at rising edge of
				1	0	Uses internal clock. Counts at falling edge of
				1	1	Uses internal clock. Counts at falling edge of
	1	0	0	—	—	Counts at TCNT_0 compare match A*1.
All	1	0	1	—	—	Uses external clock. Counts at rising edge*2.
	1	1	0	—	—	Uses external clock. Counts at falling edge*2
	1	1	1	—	—	Uses external clock. Counts at both rising an edges*2.

Notes: 1. If the clock input of channel 0 is the TCNT_1 overflow signal and that of chan TCNT_0 compare match signal, no incrementing clock is generated. Do not u setting.

2. To use the external clock, the DDR and ICR bits in the corresponding pin sho to 0 and 1, respectively. For details, see section 9, I/O Ports.

				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of P
				1	1	Uses internal clock. Counts at falling edge of P
	0	1	1	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at rising edge of P
				1	1	Uses internal clock. Counts at falling edge of P
	1	0	0	—	—	Counts at TCNT_1 overflow signal*.
TMR_5	0	0	0	—	—	Clock input prohibited
	0	0	1	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of P
				1	1	Uses internal clock. Counts at falling edge of P
	0	1	0	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of P
				1	1	Uses internal clock. Counts at falling edge of P
	0	1	1	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at rising edge of P
				1	1	Uses internal clock. Counts at falling edge of P
	1	0	0	—	—	Counts at TCNT_0 compare match A*.
All	1	0	1	—	—	Setting prohibited
	1	1	0	—	—	Setting prohibited
	1	1	1	—	—	Setting prohibited

Note: * If the clock input of channel 4 is the TCNT_1 overflow signal and that of chann TCNT_0 compare match signal, no incrementing clock is generated. Do not us setting.

• TCSR_1

Bit	7	6	5	4	3	2	1
Bit Name	CMFB	CMFA	OVF	—	OS3	OS2	OS1
Initial Value	0	0	0	1	0	0	0
R/W	R/(W)*	R/(W)*	R/(W)*	R	R/W	R/W	R/W

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

- TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*1	Compare Match Flag B [Setting condition] • When TCNT matches TCORB [Clearing conditions] • When writing 0 after reading CMFB = 1 (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.) • When the DTC is activated by a CMIB interr the DISEL bit in MRB of the DTC is 0*3

				• When the DTC is activated by a CMIA interru the DISEL bit in MRB in the DTC is 0*[3]
5	OVF	0	R/(W)*[1]	Timer Overflow Flag [Setting condition] When TCNT overflows from H'FF to H'00 [Clearing condition] When writing 0 after reading OVF = 1 (When the CPU is used to clear this flag by writin while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)
4	ADTE	0	R/W	A/D Trigger Enable*[3] Selects enabling or disabling of A/D converter st requests by compare match A. 0: A/D converter start requests by compare matc disabled 1: A/D converter start requests by compare matc enabled
3	OS3	0	R/W	Output Select 3 and 2*[2]
2	OS2	0	R/W	These bits select a method of TMO pin output wh 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 oc (toggle output)

Notes: 1. Only 0 can be written to bits 7 to 5, to clear these flags.
2. Timer output is disabled when bits OS3 to OS0 are all 0. Timer output is 0 un
compare match occurs after a reset.
3. Available in unit 0 and unit 1 only.

- TCSR_1

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*[1]	Compare Match Flag B [Setting condition] • When TCNT matches TCORB [Clearing conditions] • When writing 0 after reading CMFB = 1 (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.) • When the DTC is activated by a CMIB interr the DISEL bit in MRB of the DTC is 0*[3]

				• When the DTC is activated by a CMIA interru the DISEL bit in MRB of the DTC is 0*[3]
5	OVF	0	R/(W)*[1]	Timer Overflow Flag [Setting condition] When TCNT overflows from H'FF to H'00 [Clearing condition] Cleared by reading OVF when OVF = 1, then wr OVF (When the CPU is used to clear this flag by writi while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)
4	—	1	R	Reserved This bit is always read as 1 and cannot be modif
3	OS3	0	R/W	Output Select 3 and 2*[2]
2	OS2	0	R/W	These bits select a method of TMO pin output w 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 o (toggle output)

Notes: 1. Only 0 can be written to bits 7 to 5, to clear these flags.
2. Timer output is disabled when bits OS3 to OS0 are all 0. Timer output is 0 un
compare match occurs after a reset.
3. Available only in unit 0 and unit 1.

12.4 Operation

12.4.1 Pulse Output

Figure 12.5 shows an example of the 8-bit timer being used to generate a pulse output w
desired duty cycle. The control bits are set as follows:

1. Clear the bit CCLR1 in TCR to 0 and set the bit CCLR0 in TCR to 1 so that TCNT i
at a TCORA compare match.
2. Set the bits OS3 to OS0 in TCSR to B'0110, causing the output to change to 1 at a T
compare match and to 0 at a TCORB compare match.

With these settings, the 8-bit timer provides pulses output at a cycle determined by TCC
pulse width determined by TCORB. No software intervention is required. The timer out
until the first compare match occurs after a reset.

12.4.2 Reset Input

Figure 12.6 shows an example of the 8-bit timer being used to generate a pulse which is o
after a desired delay time from a TMRI input. The control bits are set as follows:

1. Set both bits CCLR1 and CCLR0 in TCR to 1 and set the TMRIS bit in TCCR to 1 so
   TCNT is cleared at the high level input of the TMRI signal.
2. In TCSR, set bits OS3 to OS0 to B'0110, causing the output to change to 1 at a TCOR
   compare match and to 0 at a TCORB compare match.

With these settings, the 8-bit timer provides pulses output at a desired delay time from a T
input determined by TCORA and with a pulse width determined by TCORB and TCORA

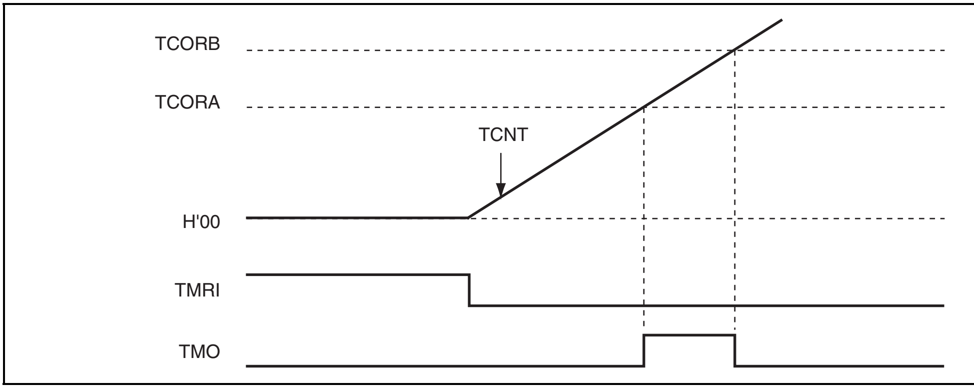


Figure 12.6 Example of Reset Input

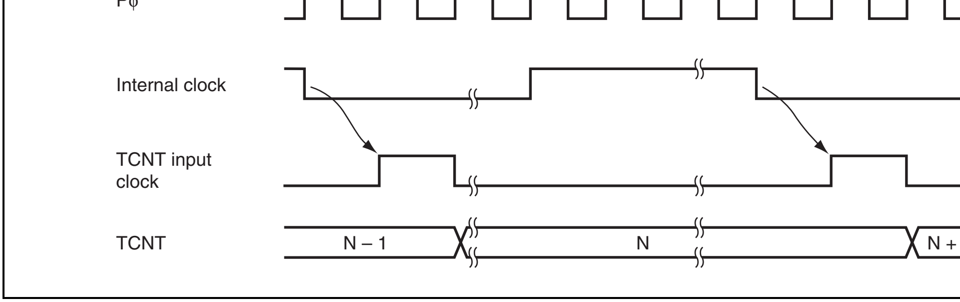


Figure 12.7 Count Timing for Internal Clock Input

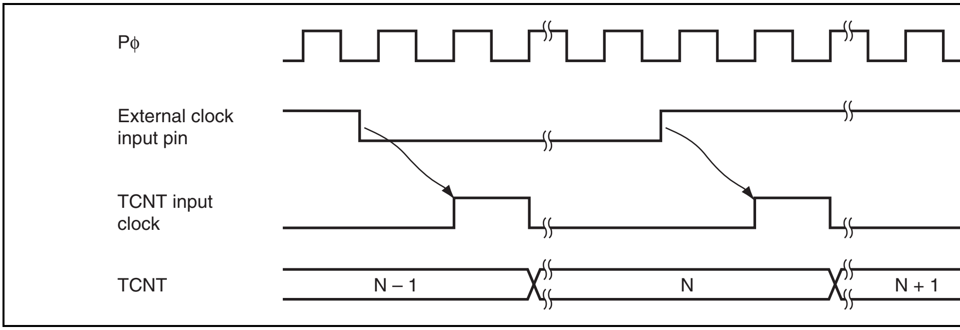


Figure 12.8 Count Timing for External Clock Input

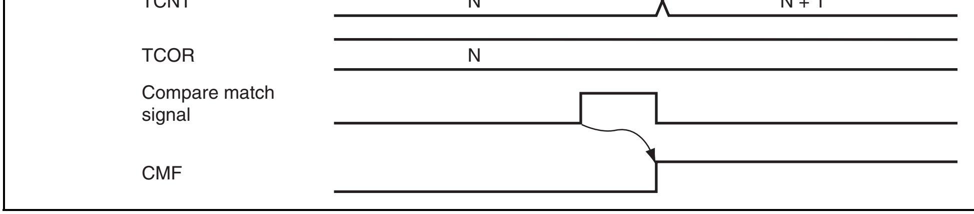


Figure 12.9 Timing of CMF Setting at Compare Match

12.5.3 Timing of Timer Output at Compare Match

When a compare match signal is generated, the timer output changes as specified by the b
to OS0 in TCSR. Figure 12.10 shows the timing when the timer output is toggled by the c
match A signal.

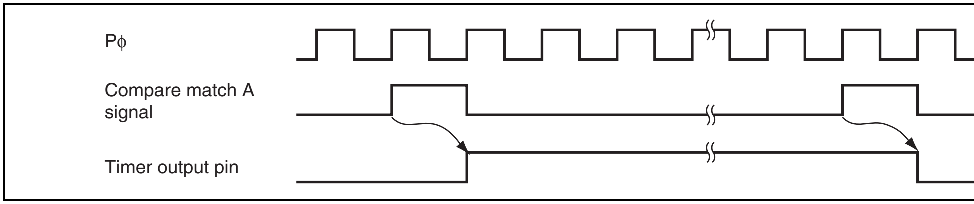


Figure 12.10 Timing of Toggled Timer Output at Compare Match A

Figure 12.11 Timing of Counter Clear by Compare Match

12.5.5 Timing of TCNT External Reset*

TCNT is cleared at the rising edge or high level of an external reset input, depending on
settings of bits CCLR1 and CCLR0 in TCR. The clear pulse width must be at least 2 sta
12.12 and Figure 12.13 shows the timing of this operation.

Note: * Clearing by an external reset is available only in units 0 and 1.

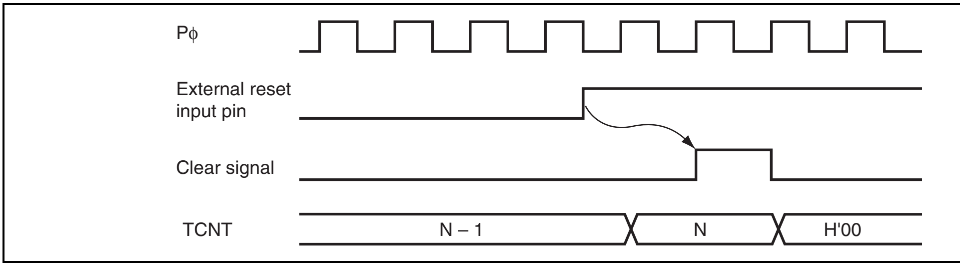


Figure 12.12 Timing of Clearance by External Reset (Rising Edge)

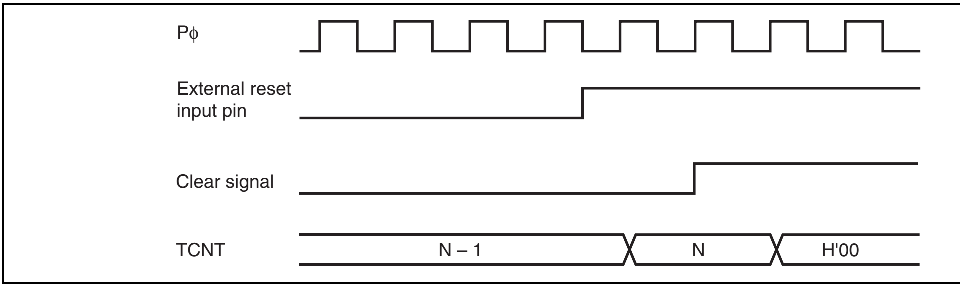


Figure 12.13 Timing of Clearance by External Reset (High Level)

Figure 12.14 Timing of OVF Setting

12.6 Operation with Cascaded Connection

If the bits CKS2 to CKS0 in either TCR_0 or TCR_1 are set to B'100, the 8-bit timers of
channels are cascaded. With this configuration, a single 16-bit timer could be used (16-bi
mode) or compare matches of the 8-bit channel 0 could be counted by the timer of channe
(compare match count mode).

12.6.1 16-Bit Counter Mode

When the bits CKS2 to CKS0 in TCR_0 are set to B'100, the timer functions as a single 1
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 n
16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit compare ma
occurs. The 16-bit counter (TCNT0 and TCNT1 together) is cleared even if counter cl
the TMRI0 pin has been set.
- The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bits c
cleared independently.

flag, generation of interrupts, output from the TMO pin, and counter clear are in accorda
the settings for each channel.

12.7 Interrupt Sources

12.7.1 Interrupt Sources and DTC Activation

- Interrupt in unit 0 and unit 1

There are three interrupt sources for the 8-bit timer (TMR_0 or TMR_1): CMIA, CMIB
Their interrupt sources and priorities are shown in table 12.4. Each interrupt source is en
disabled by the corresponding interrupt enable bit in TCR or TCSR, and independent int
requests are sent for each to the interrupt controller. It is also possible to activate the DT
means of CMIA and CMIB interrupts (This is available in unit 0 and unit 1 only).

Table 12.4 8-Bit Timer (TMR_0 or TMR_1) Interrupt Sources (in Unit 0 and Uni

Signal Name	Name	Interrupt Source	Interrupt Flag	DTC Activation	Pr
CMIA0	CMIA0	TCORA_0 compare match	CMFA	Possible	Hi
CMIB0	CMIB0	TCORB_0 compare match	CMFB	Possible	↑
OVI0	OVI0	TCNT_0 overflow	OVF	Not possible	Lo
CMIA1	CMIA1	TCORA_1 compare match	CMFA	Possible	Hi
CMIB1	CMIB1	TCORB_1 compare match	CMFB	Possible	↑
OVI1	OVI1	TCNT_1 overflow	OVF	Not possible	Lo

Name	Name	Interrupt Source	Flag	Activation	Pri
CMI4	CMIA4	TCORA_4 compare match	CMFA	Not possible	—
	CMIB4	TCORB_4 compare match	CMFB		
CMI5	CMIA5	TCORA_5 compare match	CMFA	Not possible	—
	CMIB5	TCORB_5 compare match	CMFB		

12.7.2 A/D Converter Activation

The A/D converter can be activated only by TMR_0 compare match A.*

If the ADTE bit in TCSR_0 is set to 1 when the CMFA flag in TCSR_0 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 converter. If the 8-bit timer conversion start trigger has been selected on the A/D converte this time, A/D conversion is started.

Note: * Available only in unit 0 and unit 1.

f: Counter frequency
ϕ: Operating frequency
N: TCOR value

12.8.2 Conflict between TCNT Write and Counter Clear

If a counter clear signal is generated during the T_2 state of a TCNT write cycle, the clear
priority and the write is not performed as shown in figure 12.15.

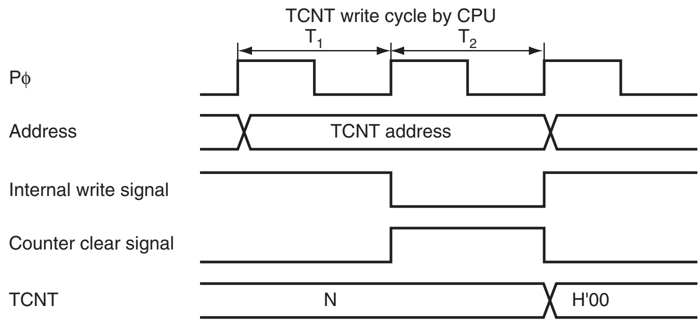


Figure 12.15 Conflict between TCNT Write and Clear

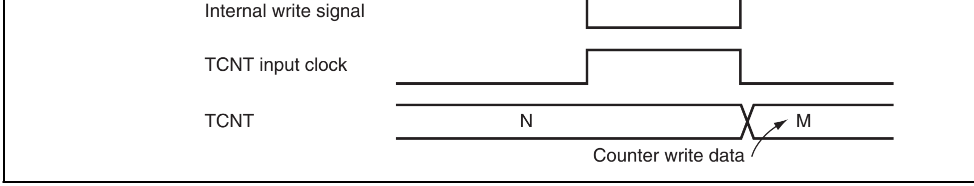


Figure 12.16 Conflict between TCNT Write and Increment

12.8.4 Conflict between TCOR Write and Compare Match

If a compare match event occurs during the T_2 state of a TCOR write cycle, the TCOR wr
priority and the compare match signal is inhibited as shown in figure 12.17.

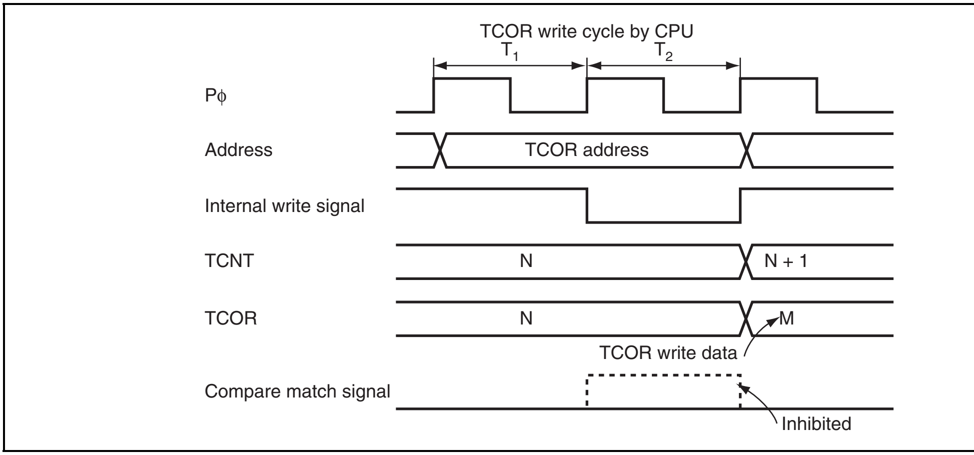


Figure 12.17 Conflict between TCOR Write and Compare Match

0-output	
No change	Lo

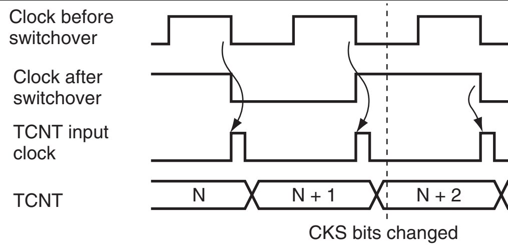
12.8.6 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the internal clock is switche 12.7 shows the relationship between the timing at which the internal clock is switched (b to the bits CKS1 and CKS0) and the TCNT operation.

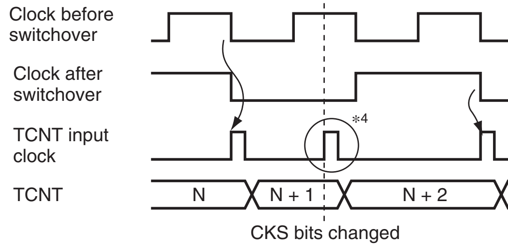
When the TCNT clock is generated from an internal clock, the rising or falling edge of t clock pulse are always monitored. Table 12.7 assumes that the falling edge is selected. I signal levels of the clocks before and after switching change from high to low as shown the change is considered as the falling edge. Therefore, a TCNT clock pulse is generated TCNT is incremented. This is similar to when the rising edge is selected.

The erroneous increment of TCNT can also happen when switching between rising and edges of the internal clock, and when switching between internal and external clocks.

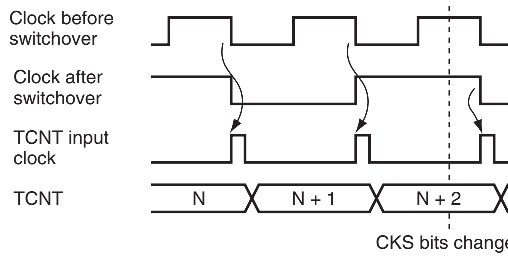
2 Switching from low to high*[2]



3 Switching from high to low*[3]



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 because the change of the signal levels is considered as a falling e TCNT is incremented.

module stop state. For details, see section 23, Power-Down Modes.

12.8.9 Interrupts in Module Stop State

If the module stop state is entered when an interrupt has been requested, it will not be po
clear the CPU interrupt source or the DTC activation source. Interrupts should therefore
disabled before entering the module stop state.

- Counter operational except in hardware standby mode or the reset state

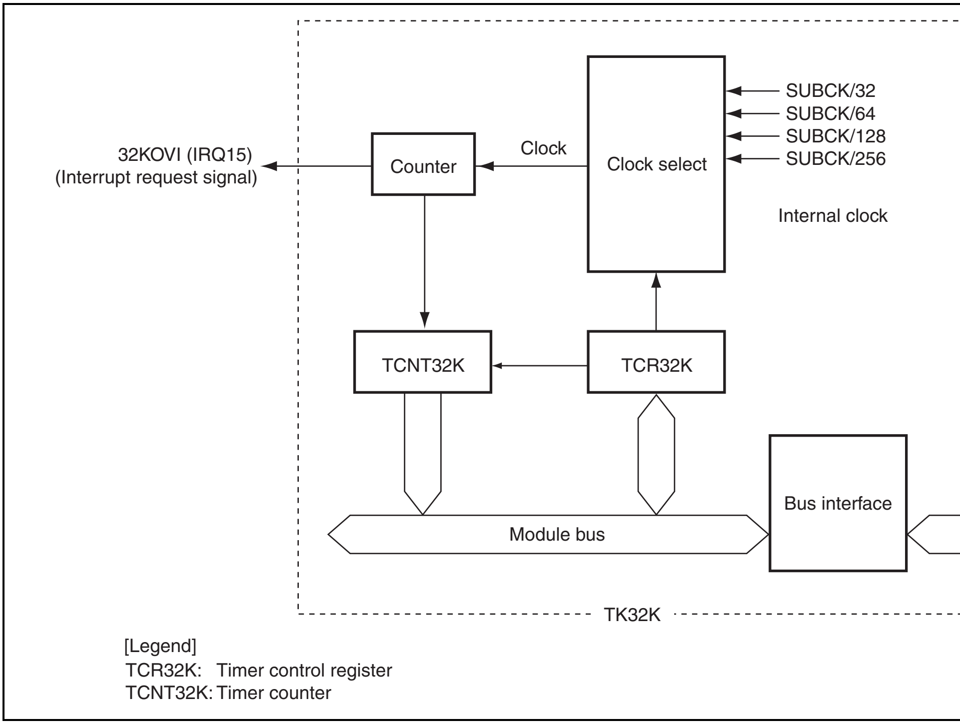


Figure 13.1 Block Diagram of TM32K

(TCR32K) is 0, TCNT32K is initialized to H'00.

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

Note: A correct value cannot be read if the counter is read while the 32-kHz oscillator i operation (OSC32STP = 1).

13.2.2 Time Control Register (TCR32K)

TCR32K enables the timer, stops the 32K oscillator, and selects the clock source to be in TCNT32K.

Bit:	7	6	5	4	3	2	1
BIt Name:	—	—	TME	—	—	OSC32STP	CKS1
Initial Value:	1	1	0	1	1	0	0
R/W:	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	R	Reserved
6	—	1	R	These bits are always read as 1 and cannot be r
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT32K starts countir this bit is cleared, TCNT32K stops counting and initialized to H'00.

00: Clock SUBCK/32 (cycle: 250 ms)

01: Clock SUBCK/64 (cycle: 500 ms)

10: Clock SUBCK/128 (cycle: 1 s)

11: Clock SUBCK/512 (cycle: 2 s)

Note: * When the CK32K bit in SUBCKCR is 1, 1 cannot be written to this bit.

13.3 Operation

Setting 1 to the TME bit in TCR32K starts the count-up operation.

A 32K timer interrupt (32KOVI) is generated each time TCNT32K overflows. Therefor
interrupt can be generated at intervals with a cycle determined by the clock select bits 0

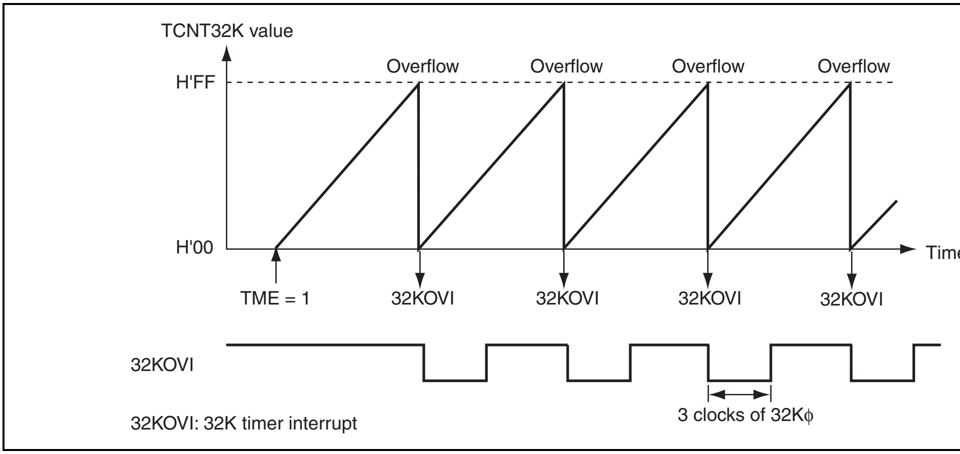


Figure 13.2 32K Timer Operation

13.5 Usage Notes

13.5.1 Changing Values of Bits CKS1 and CKS0

If bits CKS1 and CKS0 in TCR32K are written to while the TM32K is operating, errors c occur in the incrementation. The TM32K must be stopped (the TME bit is set to 0) befor values of bits CKS1 and CKS0 are changed.

13.5.2 Usage Notes on 32K Timer

- The 32K timer does not operate when the OSC32STP bit is set to 1. Always set the OSC32STP bit to 0 when starting the 32K timer.
- When the OSC32STP bit has been changed from 1 to 0, allow enough time to ensure of the oscillation by the 32-kHz oscillator.

13.5.3 Note on Reading Timer Counter

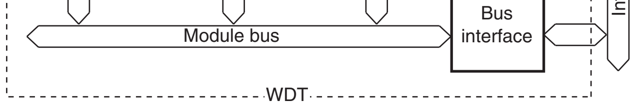
A counter read value is undefined during one clock of 32 kHz immediately after returnin software standby. Wait one clock of 32 kHz when reading the timer counter.

13.5.4 Note on Register Initialization

TCR32K and TCNT32K of the 32K timer are initialized in hardware standby mode or in reset state. These registers are not initialized by a reset caused by a watchdog timer overf

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 $\overline{\text{WDTOVF}}$. It is possible to select whe
    not the entire LSI is reset at the same time.
  - In interval timer mode
    If the counter overflows, the WDT generates an interval timer interrupt (WOVI).



[Legend]
TCSR: Timer control/status register
TCNT: Timer counter
RSTCSR: Reset control/status register

Note: * An internal reset signal can be generated by the RSTCSR setting.

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	$\overline{\text{WDTOVF}}$	Output	Outputs a counter overflow signal in watchdog timer mode

14.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT is initialized to H'00 when the TM
TCSR is cleared to 0.

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.3.2 Timer Control/Status Register (TCSR)

TCSR selects the clock source to be input to TCNT, and the timer mode.

Bit	7	6	5	4	3	2	1
Bit Name	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1
Initial Value	0	0	0	1	1	0	0
R/W	R/(W)*	R/W	R/W	R	R	R/W	R/W

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

				Cleared by reading TCSR when OVF = 1, then w to OVF (When the CPU is used to clear this flag by writin while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)
6	WT/$\overline{\text{IT}}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog interval timer. 0: Interval timer mode When TCNT overflows, an interval timer interr (WOVI) is requested. 1: Watchdog timer mode When TCNT overflows, the $\overline{\text{WDTOVF}}$ signal is
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. W bit is cleared, TCNT stops counting and is initiali H'00.
4, 3	—	All 1	R	Reserved These are read-only bits and cannot be modified
2 1 0	CKS2 CKS1 CKS0	0 0 0	R/W R/W R/W	Clock Select 2 to 0 Select the clock source to be input to TCNT. The cycle for Pϕ = 20 MHz is indicated in parenthese 000: Clock Pϕ/2 (cycle: 25.6 μs) 001: Clock Pϕ/64 (cycle: 819.2 μs) 010: Clock Pϕ/128 (cycle: 1.6 ms) 011: Clock Pϕ/512 (cycle: 6.6 ms) 100: Clock Pϕ/2048 (cycle: 26.2 ms) 101: Clock Pϕ/8192 (cycle: 104.9 ms) 110: Clock Pϕ/32768 (cycle: 419.4 ms) 111: Clock Pϕ/131072 (cycle: 1.68 s)

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

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

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 watchdo mode. This bit cannot be set in interval timer mo only 0 can be written. [Setting condition] When TCNT overflows (changed from H'FF to H watchdog timer mode [Clearing condition] Reading RSTCSR when WOVF = 1, and then w WOVF (When the CPU is used to clear this flag by writ while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)
6	RSTE	0	R/W	Reset Enable Specifies whether or not this LSI is internally re TCNT overflows during watchdog timer operatio 0: LSI is not reset even if TCNT overflows (Tho LSI is not reset, TCNT and TCSR in WDT ar 1: LSI is reset if TCNT overflows

14.4 Operation

14.4.1 Watchdog Timer Mode

To use the WDT in watchdog timer mode, set both the WT/$\overline{\text{IT}}$ and TME bits in TCSR to

During watchdog timer operation, if TCNT overflows without being rewritten because of crash or other error, the $\overline{\text{WDTOVF}}$ signal is output. This ensures that TCNT does not ove while the system is operating normally. Software must prevent TCNT overflows by rewri TCNT value (normally H'00 is written) before overflow occurs. This $\overline{\text{WDTOVF}}$ signal ca to reset the LSI internally in watchdog timer mode.

If TCNT overflows when the RSTE bit in RSTCSR is set to 1, a signal that resets this LS internally is generated at the same time as the $\overline{\text{WDTOVF}}$ signal. If a reset caused by a sig to the $\overline{\text{RES}}$ pin occurs at the same time as a reset caused by a WDT overflow, the $\overline{\text{RES}}$ pi has priority and the WOVF bit in RSTCSR is cleared to 0.

The $\overline{\text{WDTOVF}}$ signal is output for 133 cycles of Pϕ when RSTE = 1 in RSTCSR, and for cycles of Pϕ when RSTE = 0 in RSTCSR. The internal reset signal is output for 519 cycl

When RSTE = 1, an internal reset signal is generated. Since the system clock control regi (SCKCR) is initialized, the multiplication ratio of Pϕ becomes the initial value.

When RSTE = 0, an internal reset signal is not generated. Neither SCKCR nor the multip ratio of Pϕ is changed.

When TCNT overflows in watchdog timer mode, the WOVF bit in RSTCSR is set to 1. I overflows when the RSTE bit in RSTCSR is set to 1, an internal reset signal is generated entire LSI.

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

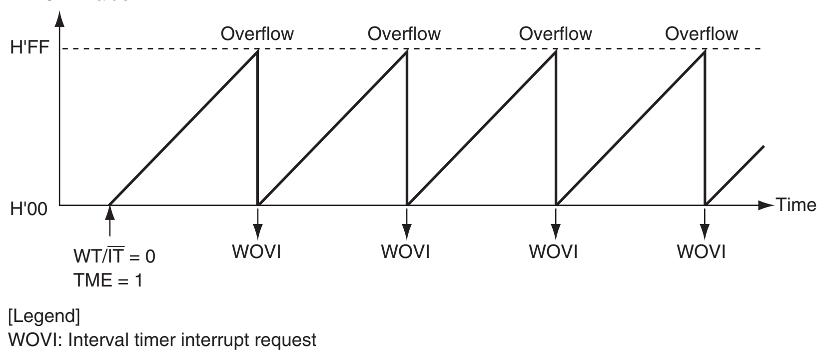


Figure 14.3 Operation in Interval Timer Mode

14.5 Interrupt Source

During interval timer mode operation, an overflow generates an interval timer interrupt (W
The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. The
must be cleared to 0 in the interrupt handling routine.

Table 14.2 WDT Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activatio
WOVI	TCNT overflow	OVF	Impossible

TCNT and TCSR must be written to by a word transfer instruction. They cannot be writ
byte transfer instruction.

For writing, TCNT and TCSR are assigned to the same address. Accordingly, perform d
transfer as shown in figure 14.4. The transfer instruction writes the lower byte data to T
TCSR.

To write to RSTCSR, execute a word transfer instruction for address H'FFA6. A byte tra
instruction cannot be used to write to RSTCSR.

The method of writing 0 to the WOVF bit in RSTCSR differs from that of writing to the
in RSTCSR. Perform data transfer as shown in figure 14.4.

At data transfer, the transfer instruction clears the WOVF bit to 0, but has no effect on t
bit. To write to the RSTE bit, perform data transfer as shown in figure 14.4. In this case,
transfer instruction writes the value in bit 6 of the lower byte to the RSTE bit, but has no
the WOVF bit.

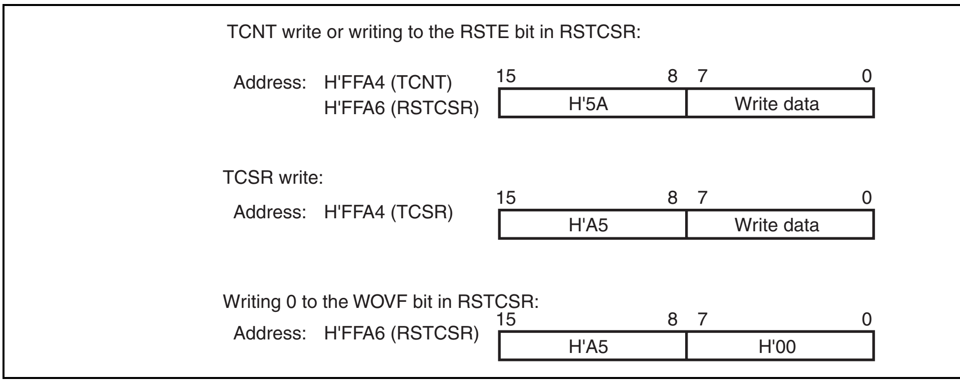


Figure 14.4 Writing to TCNT, TCSR, and RSTCSR

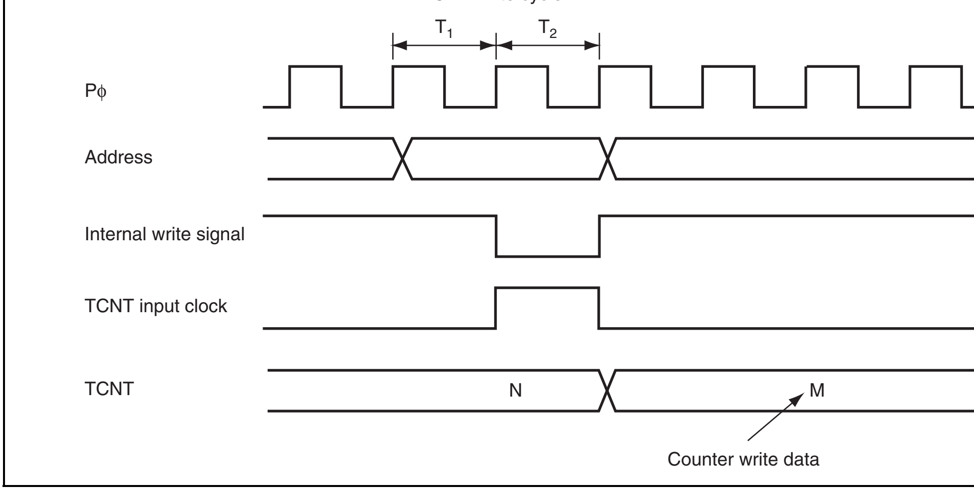


Figure 14.5 Conflict between TCNT Write and Increment

14.6.3 Changing Values of Bits CKS2 to CKS0

If bits CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors could o the incrementation. The watchdog timer must be stopped (by clearing the TME bit to 0) b values of bits CKS2 to CKS0 are changed.

14.6.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the timer mode is switched from watchdog timer mode to interval timer mode while th operating, errors could occur in the incrementation. The watchdog timer must be stopped clearing the TME bit to 0) before switching the timer mode.

If the $\overline{\text{WDTOVF}}$ signal is input to the $\overline{\text{RES}}$ pin, this LSI will not be initialized correctly.
sure that the $\overline{\text{WDTOVF}}$ signal is not input logically to the $\overline{\text{RES}}$ pin. To reset the entire sy
means of the $\overline{\text{WDTOVF}}$ signal, use a circuit like that shown in figure 14.6.

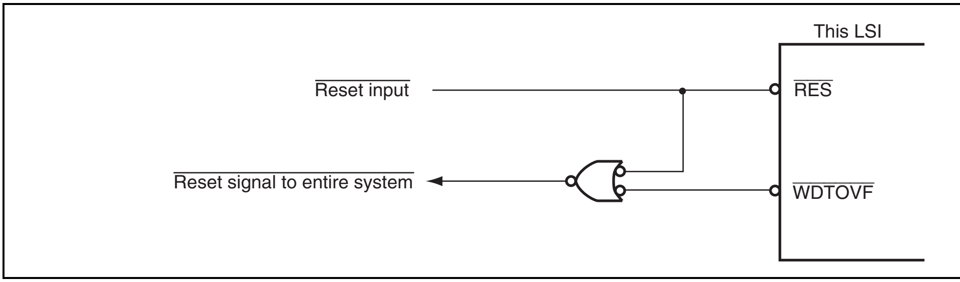


Figure 14.6 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example)

14.6.7 Transition to Watchdog Timer Mode or Software Standby Mode

When the WDT operates in watchdog timer mode, a transition to software standby mod
made even when the SLEEP instruction is executed when the SSBY bit in SBYCR is se
Instead, a transition to sleep mode is made.

To transit to software standby mode, the SLEEP instruction must be executed after halti
WDT (clearing the TME bit to 0).

When the WDT operates in interval timer mode, a transition to software standby mode i
through execution of the SLEEP instruction when the SSBY bit in SBYCR is set to 1.

communication mode. SCI_5 enables transmitting and receiving IrDA communication w
based on the IrDA Specifications version 1.0. This LSI incorporates the on-chip CRC (C
Redundancy Check) computing unit that realizes high reliability of high-speed data tran
the CRC computing unit is not connected to SCI, operation is executed by writing data t
registers.

Figure 15.1 shows a block diagram of the SCI_0 to SCI_4. Figure 15.2 shows a block di
the SCI_5 and SCI_6.

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 re
  be executed simultaneously. Double-buffering is used in both the transmitter and the
  enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected

  The external clock can be selected as a transfer clock source (except for the smart ca
  interface).
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode
- Four interrupt sources

  The interrupt sources are transmit-end, transmit-data-empty, receive-data-full, and re
  error. The transmit-data-empty and receive-data-full interrupt sources can activate th
  DMAC.
- Module stop state specifiable

16-MHz operation: 115.192 kbps, 460.784 kbps, or 720 kbps can be selected
32-MHz operation: 720 kbps

- Average transfer rate generator (SCI_5, SCI_6)
  8-MHz operation: 460.784 kbps can be selected
  10.667-MHz operation: 115.152 kbps or 460.606 kbps can be selected
  12-MHz operation: 230.263 kbps or 460.526 kbps can be selected
  16-MHz operation: 115.196 kbps, 460.784 kbps, 720 kbps, or 921.569 kbps can be se
  24-MHz operation: 115.132 kbps, 460.526 kbps, 720 kbps, or 921.053 kbps can be se
  32-MHz operation: 720 kbps can be selected

Clocked Synchronous Mode (SCI_0, 1, 2, and 4):

- Data length: 8 bits
- Receive error detection: Overrun errors

Smart Card Interface:

- An error signal can be automatically transmitted on detection of a parity error during
- Data can be automatically re-transmitted on receiving an error signal during transmiss
- Both direct convention and inverse convention are supported

		115.192 kbps	115.1
$P\phi$ = 12 Hz	—	—	460.52
			230.26
$P\phi$ = 16 Hz	—	720 kbps	921.56
		460 784kbps	720 kb
		115.192 kbps	460.78
			115.19
$P\phi$ = 24 Hz	—	—	921.05
			720 kb
			460.52
			115.13
$P\phi$ = 32 Hz	—	720 kbps	720 kb

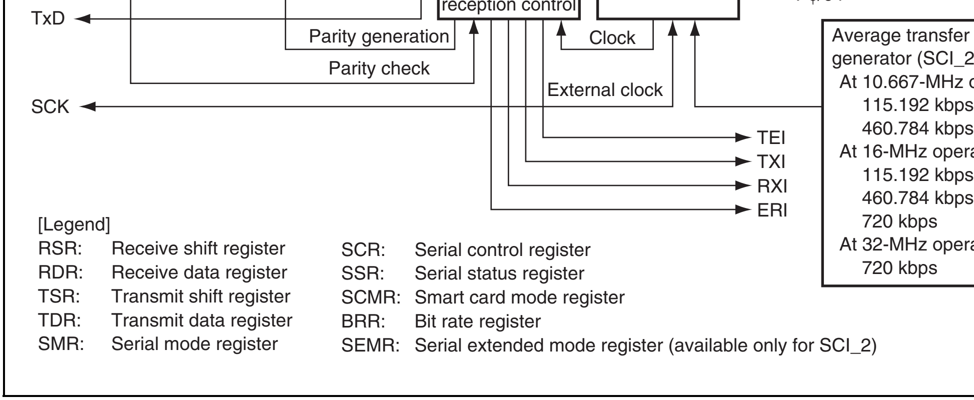


Figure 15.1 Block Diagram of SCI_0, 1, 2, and 4

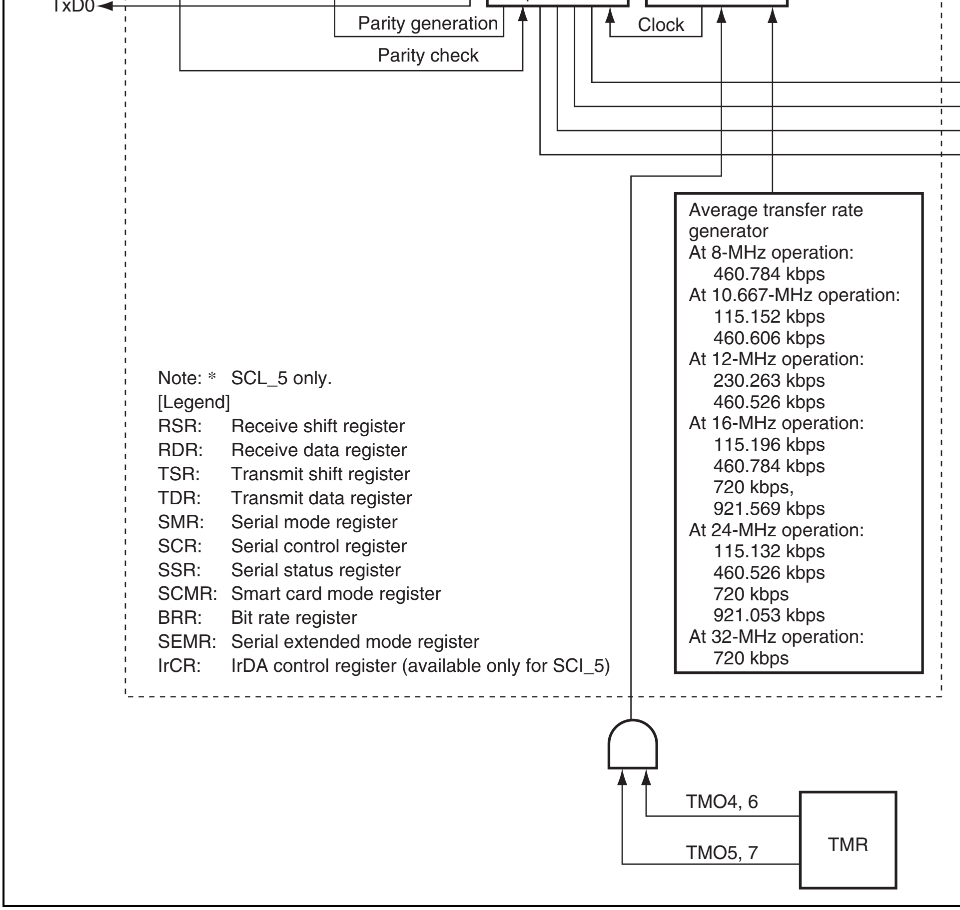


Figure 15.2 Block Diagram of SCI_5 and SCI_6

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
3	SCK3	I/O	Channel 3 clock input/output
	RxD3	Input	Channel 3 receive data input
	TxD3	Output	Channel 3 transmit data output
4	SCK4	I/O	Channel 4 clock input/output
	RxD4	Input	Channel 4 receive data input
	TxD4	Output	Channel 4 transmit data output
5	RxD5/IrRxD	Input	Channel 5 receive data input
	TxD5/IrTxD	Output	Channel 5 transmit data output
6	RxD6	Input	Channel 6 receive data input
	TxD6	Output	Channel 6 transmit data output

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting th channel designation.

- 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)

Channel 1:

- 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)

- Serial extended mode register_2 (SEMR_2)

Channel 4:

- Receive shift register_4 (RSR_4)
- Transmit shift register_4 (TSR_4)
- Receive data register_4 (RDR_4)
- Transmit data register_4 (TDR_4)
- Serial mode register_4 (SMR_4)
- Serial control register_4 (SCR_4)
- Serial status register_4 (SSR_4)
- Smart card mode register_4 (SCMR_4)
- Bit rate register_4 (BRR_4)

Channel 5:

- Receive shift register_5 (RSR_5)
- Transmit shift register_5 (TSR_5)
- Receive data register_5 (RDR_5)
- Transmit data register_5 (TDR_5)
- Serial mode register_5 (SMR_5)
- Serial control register_5 (SCR_5)
- Serial status register_5 (SSR_5)
- Smart card mode register_5 (SCMR_5)
- Bit rate register_5 (BRR_5)
- Serial extended mode register_5 (SEMR_5)
- IrDA control register_5 (IrCR)

- Serial extended mode register_6 (SEMR_6)
- Bit rate register_6 (BRR_6)

15.3.1 Receive Shift Register (RSR)

RSR is a shift register which is used to receive serial data input from the RxD pin and co
into parallel data. When one frame 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 frame o
data, it transfers the received serial data from RSR to RDR where it is stored. This allow
receive the next data. Since RSR and RDR function as a double buffer in this way, conti
receive operations can be performed. After confirming that the RDRF bit in SSR is set t
RDR only once. RDR cannot be written to by the CPU.

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the S
automatically transfers transmit data from TDR to TSR, and then sends the data to the Tx
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 baud rate generator clocl
Some bits in SMR have different functions in normal mode and smart card interface mod

- When SMIF in SCMR = 0

Bit	7	6	5	4	3	2	1
Bit Name	C/$\overline{\text{A}}$	CHR	PE	O/$\overline{\text{E}}$	STOP	MP	CKS1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- When SMIF in SCMR = 1

Bit	7	6	5	4	3	2	1
Bit Name	GM	BLK	PE	O/$\overline{\text{E}}$	BCP1	BCP0	CKS1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

				the MSB (bit 7) in TDR is not transmitted in transmission. In clocked synchronous mode, a fixed data leng bits is used.
5	PE	0	R/W	Parity Enable (valid only in asynchronous mode When this bit is set to 1, the parity bit is added t data before transmission, and the parity bit is ch reception. For a multiprocessor format, parity b and checking are not performed regardless of t setting.
4	O/$\overline{E}$	0	R/W	Parity Mode (valid 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 (valid only in asynchronous mo Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked. If second stop bit is 0, it is treated as the start bit transmit frame.
2	MP	0	R/W	Multiprocessor Mode (valid only in asynchronou When this bit is set to 1, the multiprocessor fun enabled. The PE bit and O/$\overline{E}$ bit settings are inv multiprocessor mode.

				baud rate, see section 15.3.9, Bit Rate Register is the decimal display of the value of n in BRR (s section 15.3.9, Bit Rate Register (BRR)).

Note: * Available in SCI_0, 1, 2, and 4 only. Setting is prohibited in SCI_5 and SCI_6.

Bit Functions in Smart Card Interface Mode (When SMIF in SCMR = 1):

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	GSM Mode Setting this bit to 1 allows GSM mode operation. mode, the TEND set timing is put forward to 11.0 the start and the clock output control function is appended. For details, see sections 15.7.6, Data Transmission (Except in Block Transfer Mode) a 15.7.8, Clock Output Control.
6	BLK	0	R/W	Setting this bit to 1 allows block transfer mode op For details, see section 15.7.3, Block Transfer M
5	PE	0	R/W	Parity Enable (valid only in asynchronous mode) When this bit is set to 1, the parity bit is added to data before transmission, and the parity bit is che reception. Set this bit to 1 in smart card interface
4	O/$\overline{E}$	0	R/W	Parity Mode (valid only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity For details on the usage of this bit in smart card mode, see section 15.7.2, Data Format (Except i Transfer Mode).

				15.3.9, Bit Rate Register (BRR).
1	CKS1	0	R/W	Clock Select 1, 0
0	CKS0	0	R/W	These bits select the clock source for the baud generator. 00: Pφ clock (n = 0) 01: Pφ/4 clock (n = 1) 10: Pφ/16 clock (n = 2) 11: Pφ/64 clock (n = 3) For the relation between the settings of these b baud rate, see section 15.3.9, Bit Rate Registe is the decimal display of the value of n in BRR (section 15.3.9, Bit Rate Register (BRR)).

Note: etu (Elementary Time Unit): 1-bit transfer time

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- When SMIF in SCMR = 1

Bit	7	6	5	4	3	2	1
Bit Name	TIE	RIE	TE	RE	MPIE	TEIE	CKE1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Functions in Normal Serial Communication Interface Mode (When SMIF in SC

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a TXI interrupt request i enabled. A TXI interrupt request can be cancelled by read from the TDRE flag and then clearing the flag to clearing the TIE bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt re are enabled. RXI and ERI interrupt requests can be cancelled reading 1 from the RDRF, FER, PER, or ORER then clearing the flag to 0, or by clearing the RIE

				When this bit is set to 1, reception is enabled. U condition, serial reception is started by detectin bit in asynchronous mode or the synchronous c in clocked synchronous mode. Note that SMR s set prior to setting the RE bit to 1 in order to de the reception format. Even if reception is halted by clearing this bit to RDRF, FER, PER, and ORER flags are not affe the previous value is retained.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (valid only whe bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which t multiprocessor bit is 0 is skipped, and setting of RDRF, FER, and ORER status flags in SSR is On receiving data in which the multiprocessor b bit is automatically cleared and normal receptio resumed. For details, see section 15.5, Multipro Communication Function. When receive data including MPB = 0 in SSR is received, transfer of the received data from RS detection of reception errors, and the settings o FER, and ORER flags in SSR are not performe receive data including MPB = 1 is received, the in SSR is set to 1, the MPIE bit is automatically 0, and RXI and ERI interrupt requests (in the ca the TIE and RIE bits in SCR are set to 1) and s the FER and ORER flags are enabled.

00: On-chip baud rate generator

The SCK pin functions as I/O port.

01: On-chip baud rate generator

The clock with the same frequency as the bi
output from the SCK pin.

1X: External clock

The clock with a frequency 16 times the bit r
should be input from the SCK pin.

- Clocked synchronous mode

0X: Internal clock

The SCK pin functions as the clock output p

1X: External clock

The SCK pin functions as the clock input pin

				1X: External clock or average transfer rate gen
				When an external clock is used, the clock w frequency 16 times the bit rate should be in the SCK pin.
				When an average transfer rate generator is
				• Clocked synchronous mode
				0X: Internal clock
				The SCK pin functions as the clock output p
				1X: External clock
				The SCK pin functions as the clock input pi
1	CKE1	0	R/W	Clock Enable 1, 0 (for SCI_5 and SCI_6)
0	CKE0	0	R/W	These bits select the clock source.
				• Asynchronous mode
				00: On-chip baud rate generator
				1X: TMR clock input or average transfer rate ge
				When an average transfer rate generator is
				When TMR clock input is used.
				• Clocked synchronous mode
				Not available

[Legend]

X: Don't care

				When this bit is set to 1, RXI and ERI interrupt re are enabled. RXI and ERI interrupt requests can be cancelled reading 1 from the RDRF, FER, PER, or ORER then clearing the flag to 0, or by clearing the RIE
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled this condition, serial transmission is started by w transmit data to TDR, and clearing the TDRE fla to 0. Note that SMR should be set prior to setting bit to 1 in order to designate the transmission fo If transmission is halted by clearing this bit to 0, TDRE flag in SSR is fixed 1.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. U condition, serial reception is started by detecting bit in asynchronous mode or the synchronous cl in clocked synchronous mode. Note that SMR sh set prior to setting the RE bit to 1 in order to des the reception format. Even if reception is halted by clearing this bit to RDRF, FER, PER, and ORER flags are not affe the previous value is retained.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (valid only when 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.

- When GM in SMR = 1

00: Output fixed low

01: Clock output

10: Output fixed high

11: Clock output

Note: * No SCK pins exist in SCI_5 and SCI_6.

15.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. T
RDRF, ORER, PER, and FER can only be cleared. Some bits in SSR have different func
normal mode and smart card interface mode.

- When SMIF in SCMR = 0

Bit	7	6	5	4	3	2	1
Bit Name	TDRE	RDRF	ORER	FRE	PER	TEND	MPB
Initial Value	1	0	0	0	0	1	0
R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R

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

Bit	Bit Name	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 TDF (When the CPU is used to clear this flag by w while the corresponding interrupt is enabled, to read the flag after writing 0 to it.) • When a TXI interrupt request is issued allowi DMAC or DTC to write data to TDR

				while the corresponding interrupt is enablec to read the flag after writing 0 to it.) • When an RXI interrupt request is issued allo DMAC or DTC to read data from RDR The RDRF flag is not affected and retains its pr value when the RE bit in SCR is cleared to 0. Note that when the next serial reception is com while the RDRF flag is being set to 1, an overru occurs and the received data is lost.
5	ORER	0	R/(W)*	Overrun Error Indicates that an overrun error has occurred du reception and the reception ends abnormally. [Setting condition] • When the next serial reception is completec RDRF = 1 In RDR, receive data prior to an overrun err occurrence is retained, but data received af overrun error occurrence is lost. When the (is set to 1, subsequent serial reception canr performed. Note that, in clocked synchronou serial transmission also cannot continue. [Clearing condition] • When 0 is written to ORER after reading OF (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.) Even when the RE bit in SCR is cleared, the flag is not affected and retains its previous v

is transferred to RDR, however, the RDRF fla
set. In addition, when the FER flag is being s
the subsequent serial reception cannot be pe
In clocked synchronous mode, serial transmi
also cannot continue.

[Clearing condition]

- When 0 is written to FER after reading FER =

  (When the CPU is used to clear this flag by w
  while the corresponding interrupt is enabled,
  to read the flag after writing 0 to it.)

  Even when the RE bit in SCR is cleared, the
  is not affected and retains its previous value.

				subsequent serial reception cannot be perfo clocked synchronous mode, serial transmis cannot continue. [Clearing condition] • When 0 is written to PER after reading PER (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.) Even when the RE bit in SCR is cleared, the is not affected and retains its previous value
2	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last transmit character [Clearing conditions] • When 0 is written to TDRE after reading TD • When a TXI interrupt request is issued allov DMAC or DTC to write data to TDR
1	MPB	0	R	Multiprocessor Bit Stores the multiprocessor bit value in the receiv When the RE bit in SCR is cleared to 0 its prev is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Sets the multiprocessor bit value to be added to transmit frame.

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

				• When 0 is written to TDRE after reading TDF (When the CPU is used to clear this flag by w while the corresponding interrupt is enabled, to read the flag after writing 0 to it.) • When a TXI interrupt request is issued allowi DMAC or DTC to write data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates whether receive data is stored in RDR. [Setting condition] • When serial reception ends normally and rec is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDF (When the CPU is used to clear this flag by w while the corresponding interrupt is enabled, to read the flag after writing 0 to it.) • When an RXI interrupt request is issued allov DMAC or DTC to read data from RDR The RDRF flag is not affected and retains its pre value even when the RE bit in SCR is cleared to Note that when the next reception is completed v RDRF flag is being set to 1, an overrun error occ the received data is lost.

				is set to 1, subsequent serial reception cann performed. Note that, in clocked synchronou serial transmission also cannot continue. [Clearing condition] • When 0 is written to ORER after reading OR (When the CPU is used to clear this flag by while the corresponding interrupt is enabled to read the flag after writing 0 to it.) Even when the RE bit in SCR is cleared, the flag is not affected and retains its previous v
4	ERS	0	R/(W)*	Error Signal Status [Setting condition] • When a low error signal is sampled [Clearing condition] • When 0 is written to ERS after reading ERS

subsequent serial reception cannot be perfor
clocked synchronous mode, serial transmissi
cannot continue.

[Clearing condition]

- When 0 is written to PER after reading PER =

  (When the CPU is used to clear this flag by w
  while the corresponding interrupt is enabled,
  to read the flag after writing 0 to it.)

  Even when the RE bit in SCR is cleared, the
  is not affected and retains its previous value.

				set timing depends on the register setting a
				When GM = 0 and BLK = 0, 2.5 etu after tra start
				When GM = 0 and BLK = 1, 1.5 etu after tra start
				When GM = 1 and BLK = 0, 1.0 etu after tra start
				When GM = 1 and BLK = 1, 1.0 etu after tra start
				[Clearing conditions]
				• When 0 is written to TEND after reading TE
				• When a TXI interrupt request is issued allov DMAC or DTC to write the next data to TDF
1	MPB	0	R	Multiprocessor Bit 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.

Bit	Bit Name	Value	R/W	Description
7 to 4	—	All 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: Transfer with LSB-first 1: Transfer with MSB-first This bit is valid only when the 8-bit data format is transmission/reception; when the 7-bit data form used, data is always transmitted/received with L
2	SINV	0	R/W	Smart Card Data Invert Inverts the transmit/receive data logic level. This not affect the logic level of the parity bit. To inver parity bit, invert the $O/\overline{E}$ bit in SMR. 0: TDR contents are transmitted as they are. Re data is stored as it is in RDR. 1: TDR contents are inverted before being trans Receive data is stored in inverted form in RDF
1	—	1	—	Reserved This bit is always read as 1.
0	SMIF	0	R/W	Smart Card Interface Mode Select When this bit is set to 1, smart card interface mo selected. 0: Normal asynchronous or clocked synchronous 1: Smart card interface mode

Asynchronous mode	0	$B = \frac{P\phi \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{ \frac{P\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)}$
	1	$B = \frac{P\phi \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{ \frac{P\phi \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)}$
Clocked synchronous mode		$N = \frac{P\phi \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface mode		$N = \frac{P\phi \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$\text{Error (\%)} = \{ \frac{P\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)}$

[Legend]

B: Bit rate (bit/s)

N: BRR setting for baud rate generator ($0 \leq N \leq 255$)

Pφ: Operating frequency (MHz)

n and S: Determined by the SMR settings shown in the following table.

SMR Setting		
CKS1	**CKS0**	**n**
0	0	0
0	1	1
1	0	2
1	1	3

SMR Setting		
BCP1	**BCP0**	**S**
0	0	3
0	1	6
1	0	3
1	1	2

Table 15.4 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

	Operating Frequency Pϕ (MHz)										
	8			9.8304			10			12	
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11
38400	—	—	—	0	7	0.00	0	7	1.73	0	9

4800	0	79	0.00	0	90	0.16	0	95	0.00	0	103
9600	0	39	0.00	0	45	–0.93	0	47	0.00	0	51
19200	0	19	0.00	0	22	–0.93	0	23	0.00	0	25
31250	0	11	2.40	0	13	0.00	0	14	–1.70	0	15
38400	0	9	0.00	—	—	—	0	11	0.00	0	12

Note: In SCI_2, 5, and 6, this is an example when the ABCS bit in SEMR_2, 5, and 6 is
When the ABCS bit is set to 1, the bit rate is two times.

Table 15.4 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode

	Operating Frequency Pϕ (MHz)										
	17.2032			18			19.6608			2	
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	3	75	0.48	3	79	–0.12	3	86	0.31	3	88
150	2	223	0.00	2	233	0.16	2	255	0.00	3	64
300	2	111	0.00	2	116	0.16	2	127	0.00	2	129
600	1	223	0.00	1	233	0.16	1	255	0.00	2	64
1200	1	111	0.00	1	116	0.16	1	127	0.00	1	129
2400	0	223	0.00	0	233	0.16	0	255	0.00	1	64
4800	0	111	0.00	0	116	0.16	0	127	0.00	0	129
9600	0	55	0.00	0	58	–0.69	0	63	0.00	0	64
19200	0	27	0.00	0	28	1.02	0	31	0.00	0	32
31250	0	16	1.20	0	17	0.00	0	19	–1.70	0	19
38400	0	13	0.00	0	14	–2.34	0	15	0.00	0	15

2400	1	80	–0.47	1	97	–0.35	1	106	0.39	1	113
4800	0	162	0.15	0	194	0.16	0	214	–0.07	0	227
9600	0	80	–0.47	0	97	–0.35	0	106	0.39	0	113
19200	0	40	–0.76	0	48	–0.35	0	53	–0.54	0	56
31250	0	24	0.00	0	29	0	0	32	0	0	34
38400	0	19	1.73	0	23	1.73	0	26	–0.54	0	28

Note: In SCI_2, 5, and 6, this is an example when the ABCS bit in SEMR_2, 5, and 6 is 0.
When the ABCS bit is set to 1, the bit rate is two times.

Table 15.5 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mod

Pϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0

Pϕ (MHz)	Maximum Bit Rate (bit/s)	n
17.2032	537600	0
18	562500	0
19.6608	614400	0
20	625000	0
25	781250	0
30	937500	0
33	1031250	0
35	1093750	0

14.7456	3.6864	230400		33	8.2500	5156
16	4.0000	250000		35	8.7500	5468

Note: In SCI_2, this is an example when the ABCS bit in SEMR_2 is 0.
When the ABCS bit is set to 1, the bit rate is two times.

5k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2
10k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	1
25k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1
50k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	0
100k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0
250k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0
500k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	—
1M	0	1			0	3	0	4	—	—	—	—	—	—	—
2.5M			0	0*[1]			0	1	—	—	0	2	—	—	—
5M							0	0*[1]	—	—	—	—	—	—	—

[Legend]

Space: Setting prohibited.

—: Can be set, but there will be error.

Notes: 1. Continuous transmission or reception is not possible.

2. No clocked synchronous mode exists in SCI_5 and SCI_6.

Table 15.8 Maximum Bit Rate with External Clock Input (Clocked Synchronous M

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	Pϕ (MHz)	External Input Clock (MHz)	Maximum Rate (bit
8	1.3333	1333333.3	20	3.3333	3333333.
10	1.6667	1666666.7	25	4.1667	4166666.
12	2.0000	2000000.0	30	5.0000	5000000.
14	2.3333	2333333.3	33	5.5000	5500000.
16	2.6667	2666666.7	35	5.8336	5833625.
18	3.0000	3000000.0			

Note * No clocked synchronous mode exists in SCI_5 and SCI_6.

Bit Rate (bit/sec)	14.2848			16.00			18.00			20	
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
9600	0	1	0.00	0	1	12.01	0	2	15.99	0	2

Bit Rate (bit/sec)	Operating Frequency Pϕ (MHz)										
	25.00			30.00			33.00			35	
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
9600	0	3	12.49	0	3	5.01	0	4	7.59	0	4

Table 15.10 Maximum Bit Rate for Each Operating Frequency (Smart Card Inter Mode, S = 372)

Pϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	Pϕ (MHz)	Maximum Bit Rate (bit/s)	n
7.1424	9600	0	0	18.00	24194	0
10.00	13441	0	0	20.00	26882	0
10.7136	14400	0	0	25.00	33602	0
13.00	17473	0	0	30.00	40323	0
14.2848	19200	0	0	33.00	44355	0
16.00	21505	0	0	35.00	47043	0

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	R	Reserved These bits are always read as undefined and c modified.
3	ABCS	0	R/W	Asynchronous Mode Base clock Select (valid c asynchronous mode) Selects the base clock for a 1-bit period. 0: The base clock has a frequency 16 times th rate 1: The base clock has a frequency 8 times the rate

base clock with a frequency 16 times the
rate)

010: 460.784 kbps of average transfer rate sp
Pϕ = 10.667 MHz is selected (operated u
base clock with a frequency 8 times the
rate)

011: 720 kbps of average transfer rate specifi
32 MHz is selected (operated using the
with a frequency 16 times the transfer ra

100: Setting prohibited

101: 115.192 kbps of average transfer rate sp
Pϕ = 16 MHz is selected (operated using
clock with a frequency 16 times the trans

110: 460.784 kbps of average transfer rate sp
Pϕ = 16 MHz is selected (operated using
clock with a frequency 16 times the trans

111: 720 kbps of average transfer rate specifi
16 MHz is selected (operated using the
with a frequency 8 times the transfer rate

The average transfer rate only supports opera
frequencies of 10.667 MHz, 16 MHz, and 32

Bit Name	—	—	—	ABCS	ACS3	ACS2	ACS1
Initial Value	Undefined	Undefined	Undefined	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	Undefined	R	Reserved These bits are always read as undefined and c modified.
4	ABCS	0	R/W	Asynchronous Mode Base Clock Select (valid c asynchronous mode) Selects the base clock for a 1-bit period. 0: The base clock has a frequency 16 times the rate 1: The base clock has a frequency 8 times the rate
3 2 1 0	ACS3 ACS2 ACS1 ACS0	0 0 0 0	R/W R/W R/W R/W	Asynchronous Mode Clock Source Select These bits select the clock source for the avera transfer rate function in the asynchronous mod the average transfer rate function is enabled, t clock is automatically specified regardless of th bit value. The average transfer rate only corres 8MHz, 10.667MHz, 12MHz, 16MHz, 24MHz, a 32MHz. No other clock is available. Setting of A ACS0 must be done in the asynchronous mode C/$\overline{A}$ bit in SMR = 0) and the external clock inpu (the CKE bit I SCR = 1). The setting examples figures 15.3 and 15.4. (Each number in the four-digit number below corresponds to the value in the bits ACS3 to A left to right respectively.)

average transfer rate specific to Pϕ = 8M
selected (operated using the base clock
frequency 8 times the transfer rate)

0100: TMR clock input
This setting allows the TMR compare ma
output to be used as the base clock. The
below shows the correspondence betwe
SCI channels and the compare match ou

SCI Channel	TMR Unit	Compare M Output
SCI_5	Unit 2	TMO4, TMC
SCI_6	Unit 3	TMO6, TMC

0101: 115.196 kbps of average transfer rate sp
Pϕ = 16 MHz is selected (operated using
clock with a frequency 16 times the trans

0110: 460.784 kbps of average transfer rate sp
Pϕ = 16 MHz is selected (operated using
clock with a frequency 16 times the trans

0111: 720 kbps of average transfer rate specifi
16 MHz is selected (operated using the
with a frequency 8 times the transfer rat

1011: 921.053 kbps of average transfer rate spe
Pϕ = 24 or MHz or 460.526 kbps of avera
transfer rate specific to Pϕ = 12MHz is se
(operated using the base clock with a fre
times the transfer rate)

1100: 720 kbps of average transfer rate specific
32 MHz is selected (operated using the b
with a frequency 16 times the transfer rat

1101: Reserved (setting prohibited)

111x: Reserved (setting prohibited)

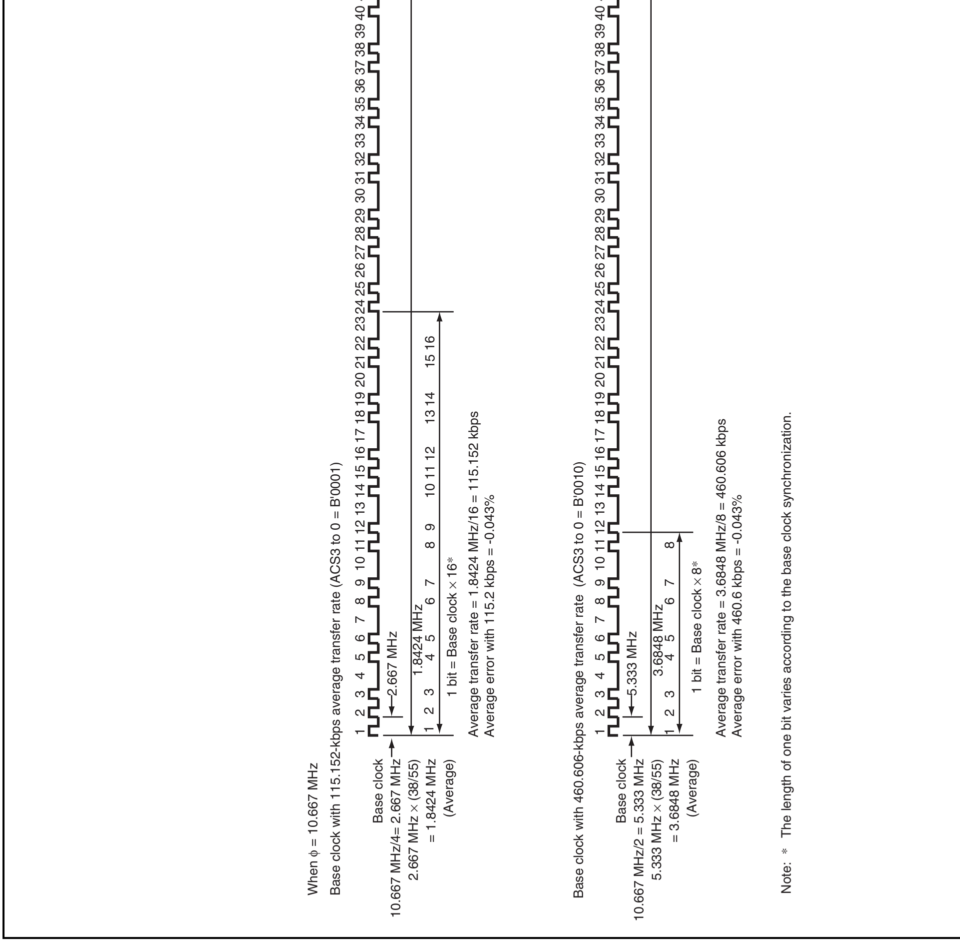


Figure 15.3 Examples of Base Clock when Average Transfer Rate Is Selecte

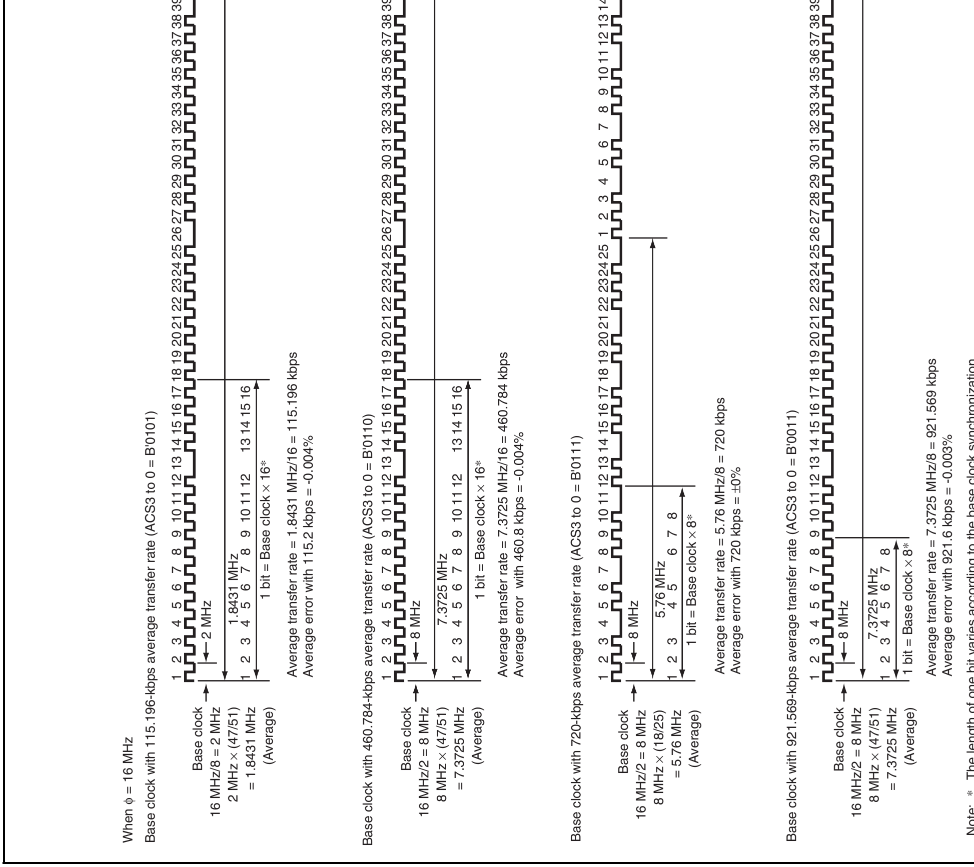


Figure 15.3 Examples of Base Clock when Average Transfer Rate Is Selected

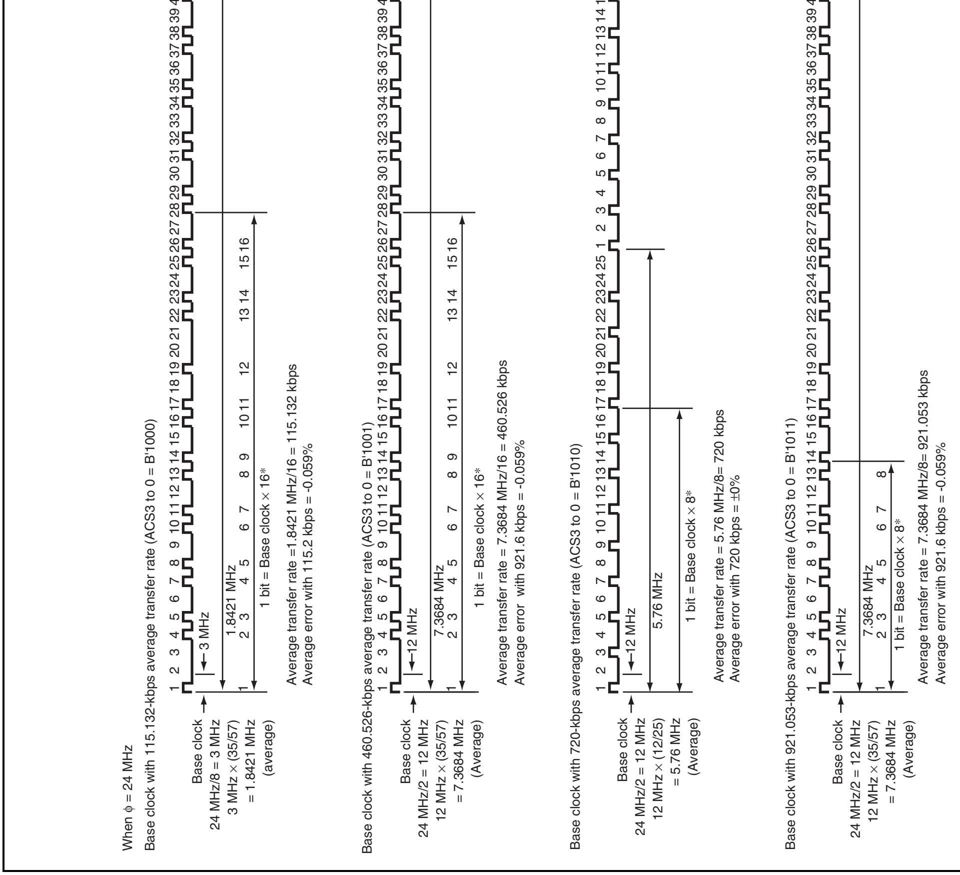


Figure 15.3 Examples of Base Clock when Average Transfer Rate Is Selecte

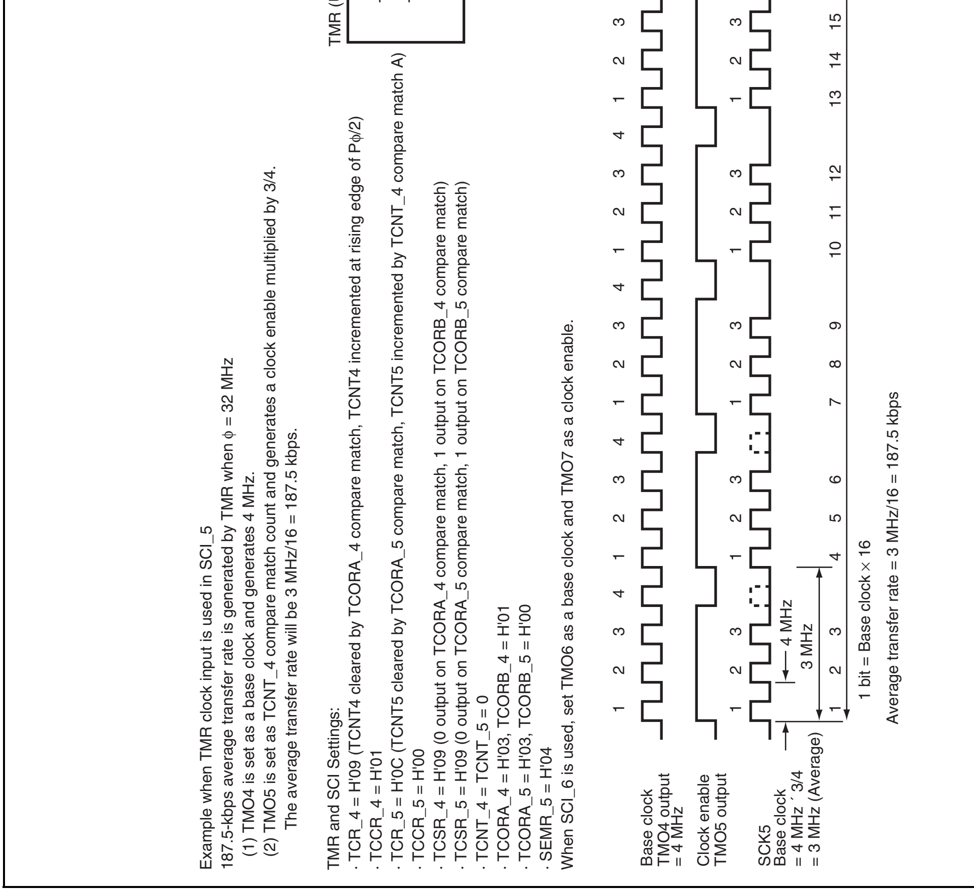


Figure 15.4 Example of Average Transfer Rate Setting when TMR Clock Is In

Bit	Bit Name	Value	R/W	Description
7	IrE	0	R/W	IrDA Enable Sets the SCI_5 I/O to normal SCI or IrDA. 0: TxD5/IrTxD and RxD5/IrRxD pins operate a TxD5 and RxD5. 1: TxD5/IrTxD and RxD5/IrRxD pins are opera IrTxD and IrRxD.
6 5 4	IrCK2 IrCK1 IrCK0	0 0 0	R/W R/W R/W	IrDA Clock Select 2 to 0 Sets the pulse width of high state at encoding output pulse when the IrDA function is enable 000: Pulse-width = B × 3/16 (Bit rate × 3/16) 001: Pulse-width = Pϕ/2 010: Pulse-width = Pϕ/4 011: Pulse-width = Pϕ/8 100: Pulse-width = Pϕ/16 101: Pulse-width = Pϕ/32 110: Pulse-width = Pϕ/64 111: Pulse-width = Pϕ/128
3	IrTxINV	0	R/W	IrTx Data Invert This bit specifies the inversion of the logic lev output. When inversion is done, the pulse wid state specified by the bits 6 to 4 becomes the width in low state. 0: Outputs the transmission data as it is as IrT 1: Outputs the inverted transmission data as I output

These bits are always read as 0. It should not
0.

15.4 Operation in Asynchronous Mode

Figure 15.5 shows the general format for asynchronous serial communication. One frame of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level). In asynchronous serial communication, the communication line is usually held in the mark s (high level). The SCI monitors the communication line, and when it goes to the space stat level), recognizes a start bit and starts serial communication. Inside the SCI, the transmitt receiver are independent units, enabling full-duplex communication. Both the transmitter receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

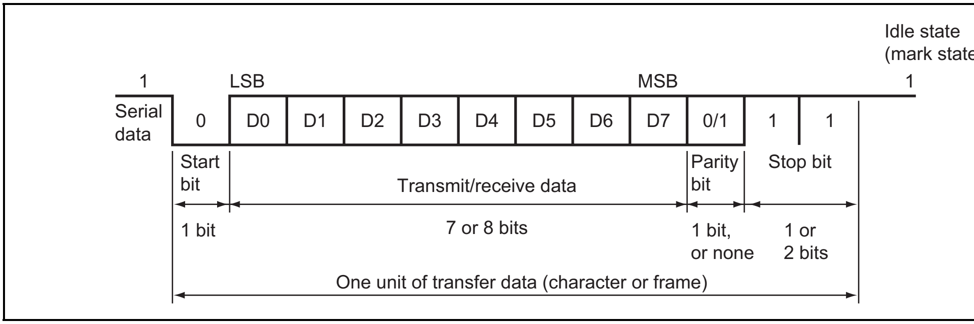


Figure 15.5 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

0	0	0	0	S | 8-bit data | STOP
0	0	0	1	S | 8-bit data | STOP | ST
0	1	0	0	S | 8-bit data | P | STO
0	1	0	1	S | 8-bit data | P | STO
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 | STO
0	–	1	0	S | 8-bit data | MPB | ST
0	–	1	1	S | 8-bit data | MPB | STO
1	–	1	0	S | 7-bit data | MPB | STOP
1	–	1	1	S | 7-bit data | MPB | STOP | STO

[Legend]
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

M: Reception margin
N: Ratio of bit rate to clock (When ABCS = 0, N = 16. When ABCS = 1, N = 8.)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined
formula below.

$$M = \left(0.5 - \frac{1}{2 \times 16} \right) \times 100 \quad [\%] \; = \; 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowe
system design.

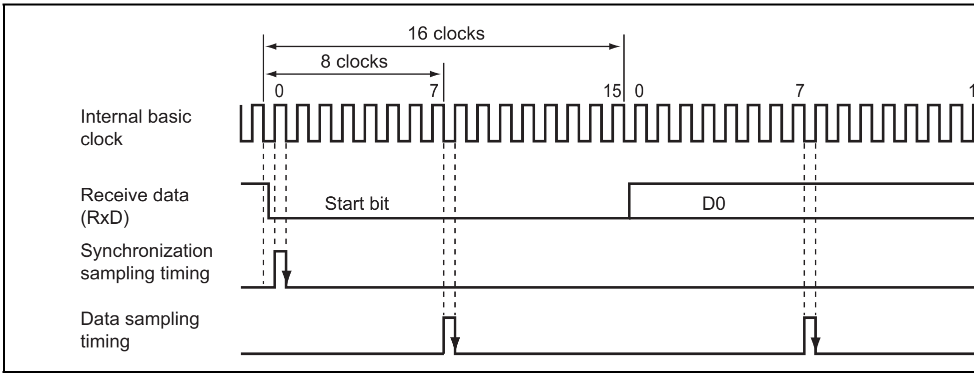


Figure 15.6 Receive Data Sampling Timing in Asynchronous Mode

Note: * This is an example when the ABCS bit in SEMR_2, 5, and 6 is 0. When the A
is 1, a frequency of 8 times the bit rate is used as a base clock and receive data
sampled at the rising edge of the 4th pulse of the base clock.

When the SCI is operated on an internal clock, the clock can be output from the SCK pi
frequency of the clock output in this case is equal to the bit rate, and the phase is such th
rising edge of the clock is in the middle of the transmit data, as shown in figure 15.7.

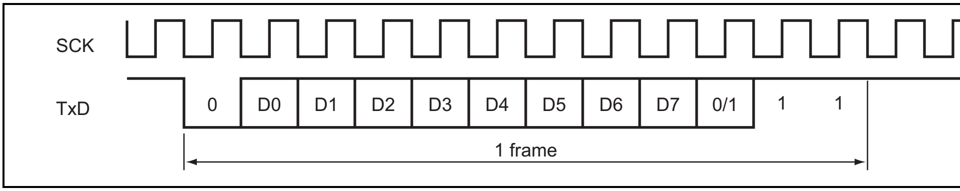
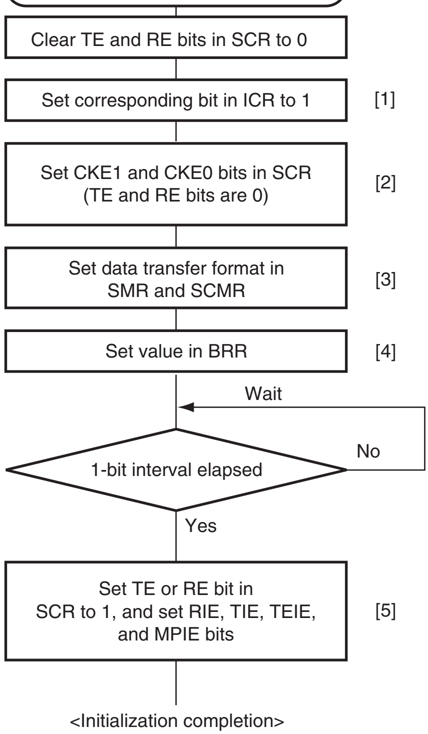


Figure 15.7 Phase Relation between Output Clock and Transmit Data (Asynchronous Mode)



[1] Set the bit in ICR for the corresponding pin when receiving data or using an external clock.

[2] 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 output is selected in asynchronous mode, the clock is output immediately after SCR settings are made.

[3] Set the data transfer format in SMR and SCMR.

[4] Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.

[5] 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 15.8 Sample SCI Initialization Flowchart

transmission of the current transmit data has finished, continuous transmission can b

3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity b 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 next transmit data is transferred from TDR to TSR, the sto 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 state is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a interrupt request is generated.

Figure 15.10 shows a sample flowchart for transmission in asynchronous mode.

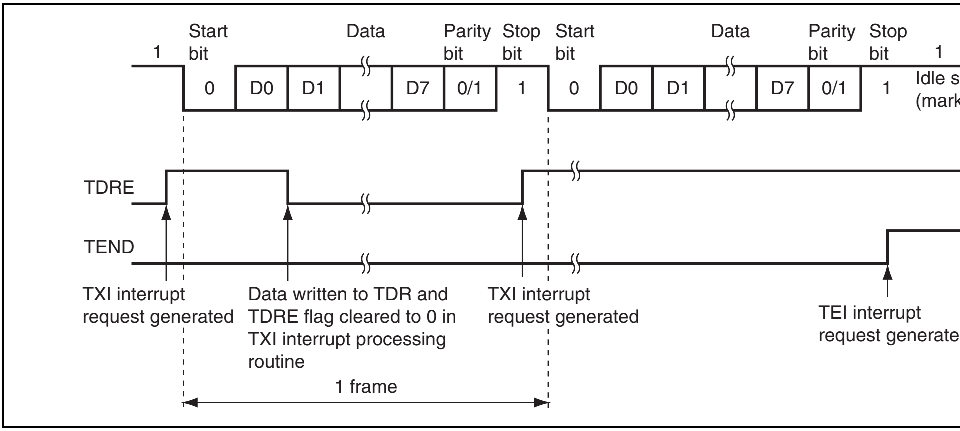
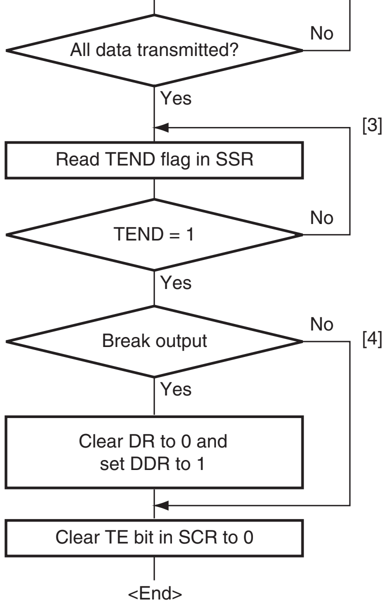


Figure 15.9 Example of Operation for Transmission in Asynchronous Mo (Example with 8-Bit Data, Parity, One Stop Bit)



To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DMAC or DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 15.10 Example of Serial Transmission Flowchart

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 transfer
RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is genera
4. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 a
data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI inter
request is generated.
5. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data i
transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt re
generated. Because the RXI interrupt processing routine reads the receive data transf
RDR before reception of the next receive data has finished, continuous reception ca
enabled.

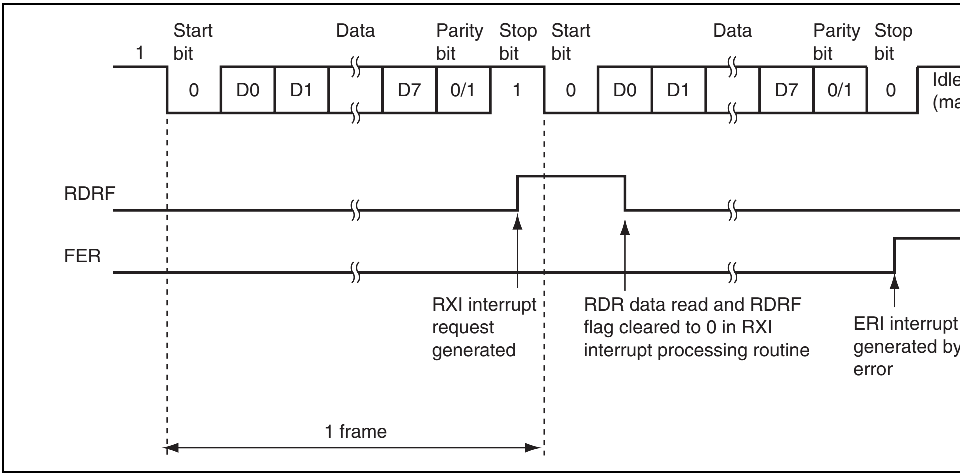
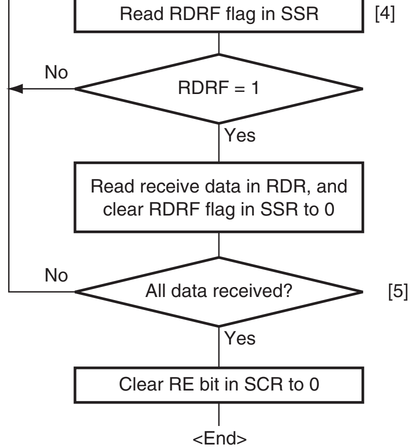


Figure 15.11 Example of SCI Operation for Reception (Example with 8-Bit Data, Parity, One Stop Bit)

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
1	1	0	1	Lost	Overrun error + parity e
0	0	1	1	Transferred to RDR	Framing error + parity e
1	1	1	1	Lost	Overrun error + framing parity error

Note: * The RDRF flag retains the state it had before data reception.



value of the input port corresponding to the RxD pin.

[4] SCI state 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 and RDR, and clear the RDRF flag to 0. However, the RDRF flag is cleared automatically when the DMAC or DTC is initiated by an RXI interrupt and reads data from RDR.

Figure 15.12 Sample Serial Reception Flowchart (1)

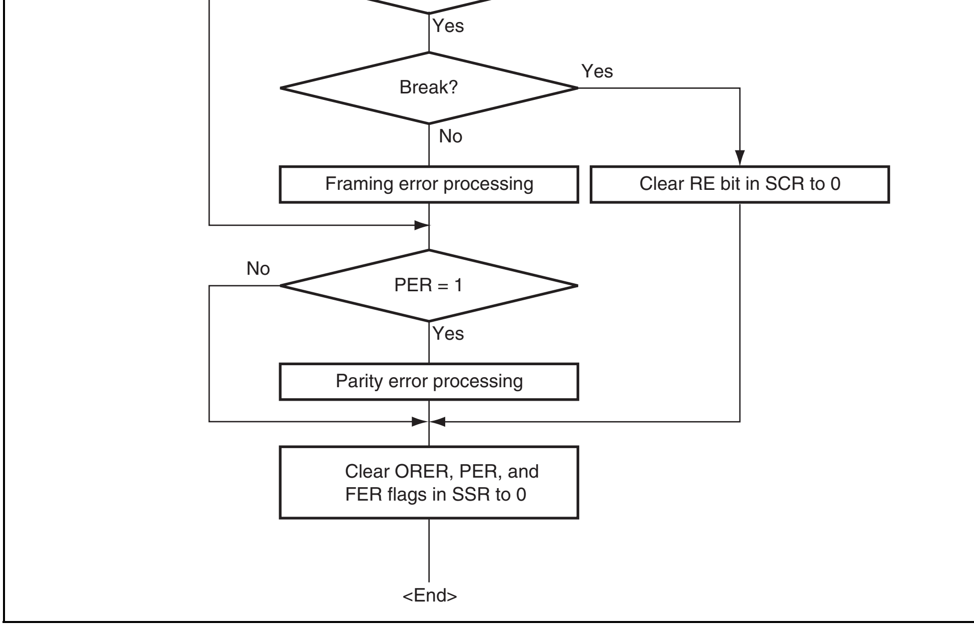


Figure 15.12 Sample Serial Reception Flowchart (2)

15.13 shows an example of inter-processor communication using the multiprocessor for
transmitting station first sends data which includes the ID code of the receiving station a
multiprocessor bit set to 1. It then transmits transmit data added with a multiprocessor bi
to 0. The receiving station skips data until data with a 1 multiprocessor bit is sent. When
a 1 multiprocessor bit is received, the receiving station compares that data with its own I
station whose ID matches then receives the data sent next. Stations whose ID does not m
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
transfer of receive data from RSR to RDR, error flag detection, and setting the SSR statu
RDRF, FER, and ORER in SSR to 1 are prohibited until data with a 1 multiprocessor bi
received. On reception of a receive character with a 1 multiprocessor bit, the MPB bit in
set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If t
in SCR is set to 1 at this time, an RXI interrupt is generated.

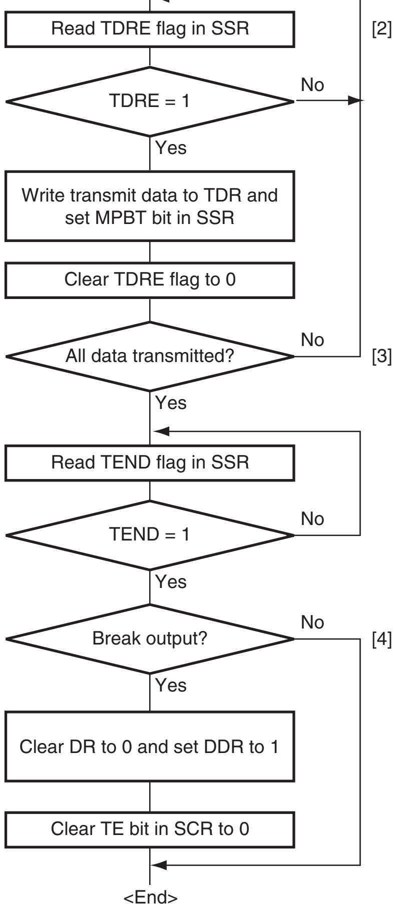
When the multiprocessor format is selected, the parity bit setting is invalid. All other bit
are the same as those in normal asynchronous mode. The clock used for multiprocessor
communication is the same as that in normal asynchronous mode.

ID transmission cycle = receiving station specification

Data transmission cycle = Data transmission to receiving station specified by ID

[Legend]
MPB: Multiprocessor bit

Figure 15.13 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)



to 1, a 1 is output for one frame, 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. However, the TDRE flag is checked and cleared automatically when the DMAC or DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port to 1, clear DR to 0, and then clear the TE bit in SCR to 0.

Figure 15.14 Sample Multiprocessor Serial Transmission Flowchart

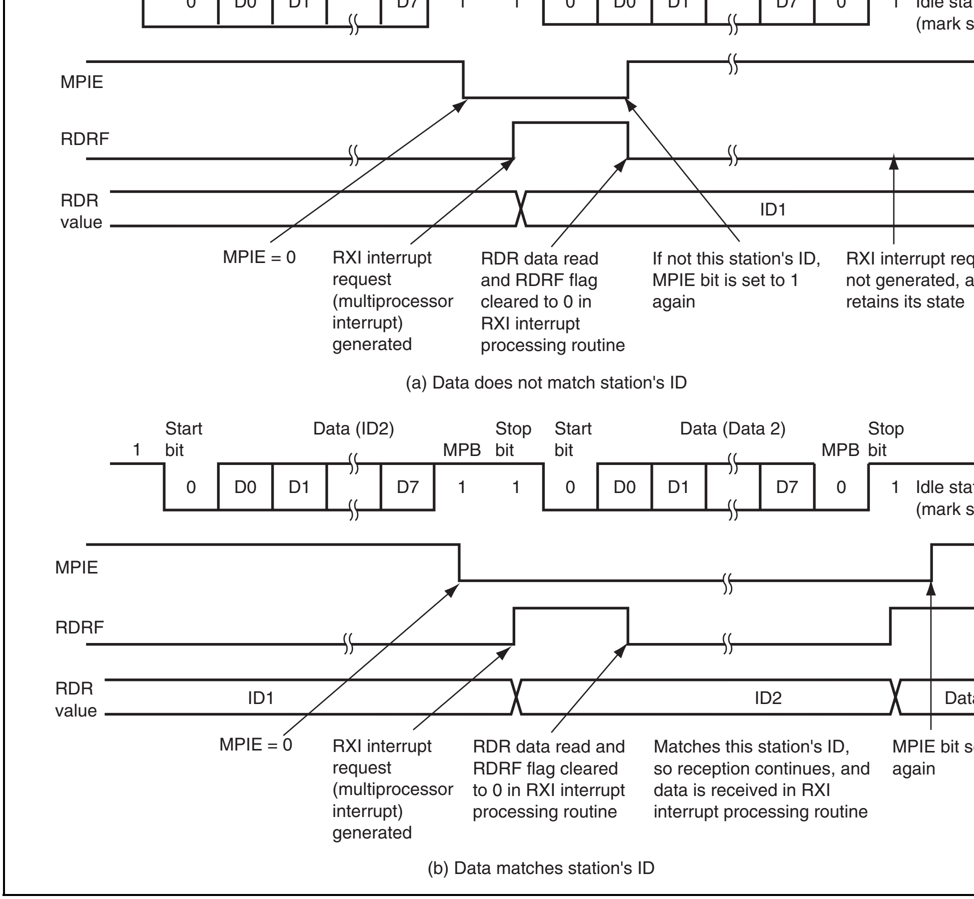
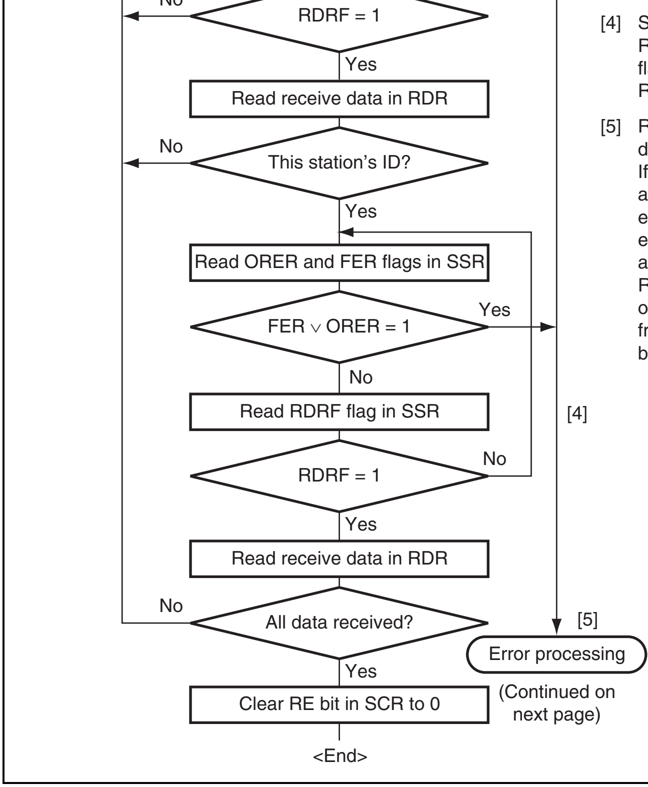


Figure 15.15 Example of SCI Operation for Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[4] SCI state 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 processing 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 processing, 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 framing error, a break can be detected by reading the RxD pin value.

Figure 15.16 Sample Multiprocessor Serial Reception Flowchart (1)

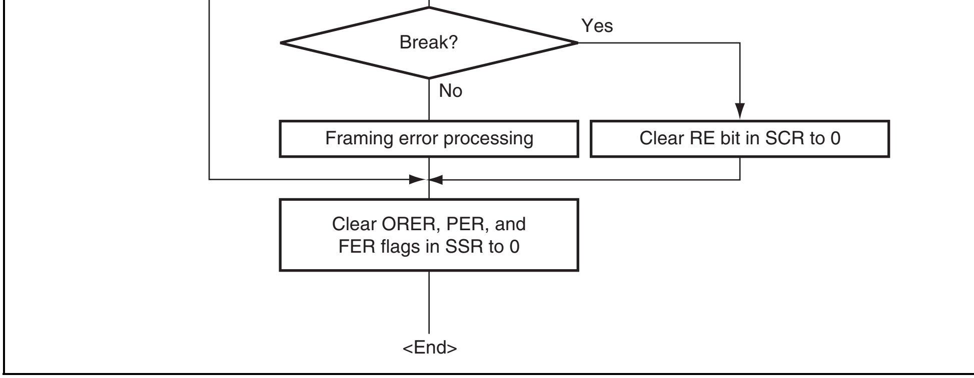


Figure 15.16 Sample Multiprocessor Serial Reception Flowchart (2)

receiver also have a double-buffered structure, so that the next transmit data can be writ
transmission or the previous receive data can be read during reception, enabling continu
transfer.

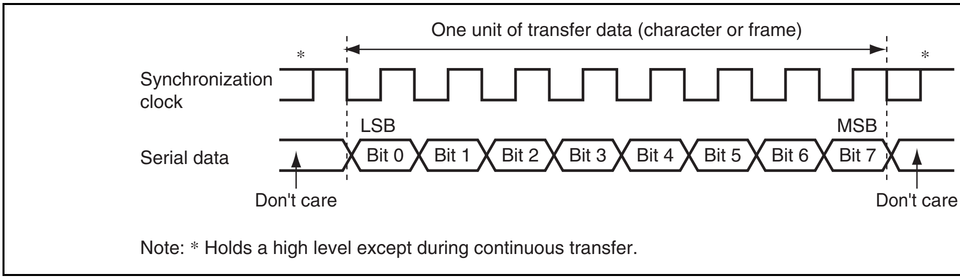
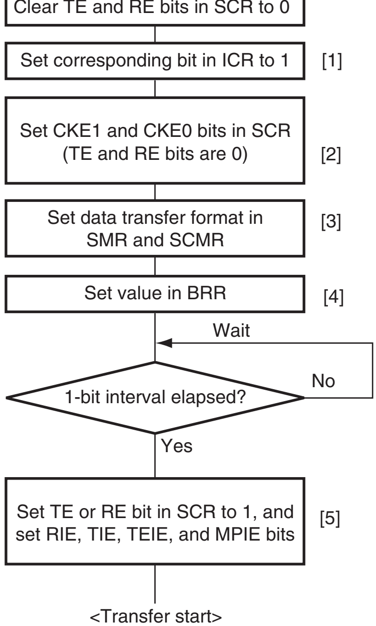


Figure 15.17 Data Format in Clocked Synchronous Communication (LSB-F

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 th
and CKE0 bits in SCR. When the SCI is operated on an internal clock, the synchronizati
is output from the SCK pin. Eight synchronization clock pulses are output in the transfer
character, and when no transfer is performed the clock is fixed high. Note that in the cas
reception only, the synchronization clock is output until an overrun error occurs or until
is cleared to 0. (Setting is prohibited in SCI_5 and SCI_6.)



pin when receiving data or using an external clock.

[2] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

[3] Set the data transfer format in SMR and SCMR.

[4] Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.

[5] 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.

Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 15.18 Sample SCI Initialization Flowchart

transmission of the current transmit data has finished, continuous transmission can s

3. 8-bit data is sent from the TxD pin synchronized with the output clock when clock o mode has been specified and synchronized with the input clock when use of an exter has been specified.
4. The SCI checks the TDRE flag at the timing for sending the last bit.
5. If the TDRE flag is cleared to 0, the next transmit data is transferred from TDR to T 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 retai output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interr is generated. The SCK pin is fixed high.

Figure 15.20 shows a sample flowchart for serial data transmission. Even if the TDRE f cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) Make sure to clear the receive error flags to 0 before starting transmission. Note that cle RE bit to 0 does not clear the receive error flags.

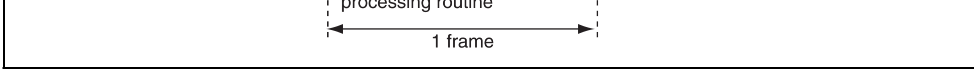
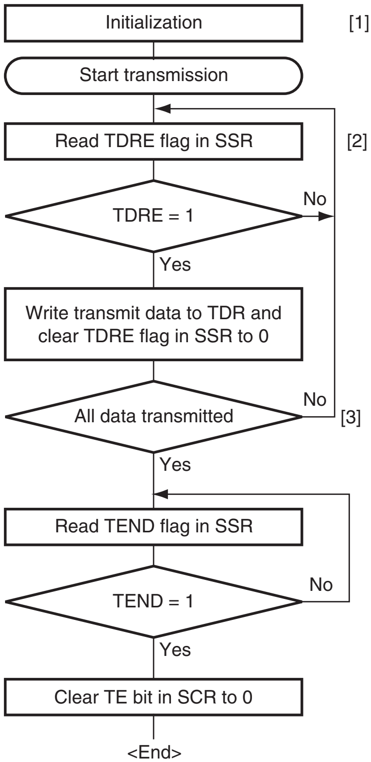


Figure 15.19 Example of Operation for Transmission in Clocked Synchronous



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[2] SCI state 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. However, the TDRE flag is checked and cleared automatically when the DMAC or DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

Figure 15.20 Sample Serial Transmission Flowchart

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 rec generated. Because the RXI interrupt processing routine reads the receive data transf RDR before reception of the next receive data has finished, continuous reception car enabled.

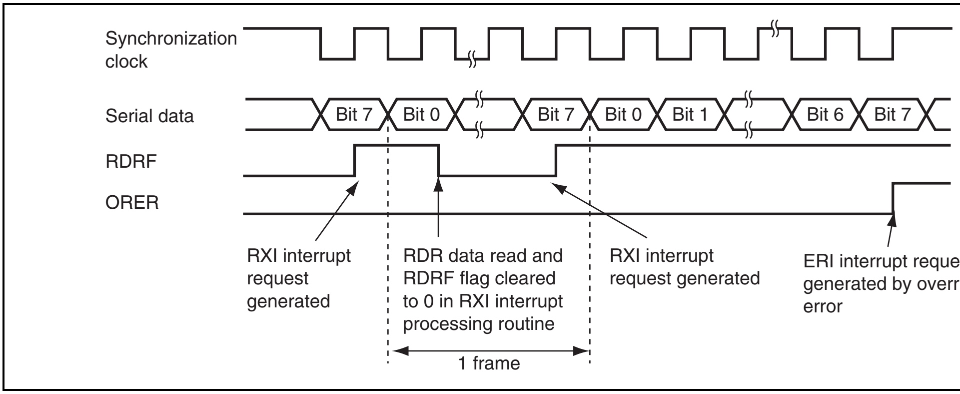
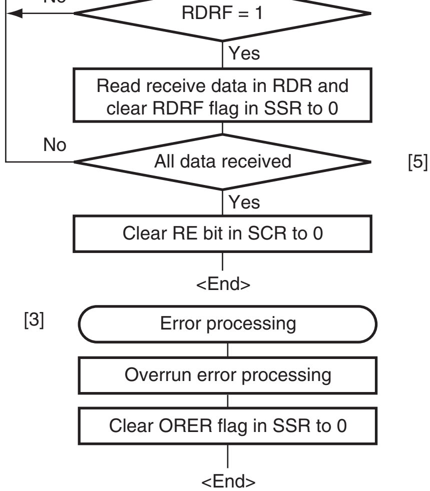


Figure 15.21 Example of Operation for Reception in Clocked Synchronous M

Transfer cannot be resumed while a receive error flag is set to 1. Accordingly, clear the FER, PER, and RDRF bits to 0 before resuming reception. Figure 15.22 shows a sample for serial data reception.



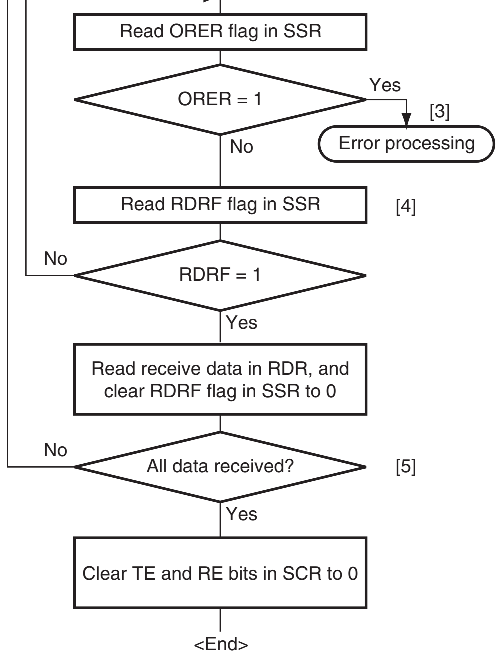
data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0 should be finished. However, the RDRF flag is cleared automatically when the DMAC or DTC is initiated by a receive data full interrupt (RXI) and reads data from RDR.

Figure 15.22 Sample Serial Reception Flowchart

15.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchron Mode) (SCI_0, 1, 2, and 4 only)

Figure 15.23 shows a sample flowchart for simultaneous serial transmit and receive opera After initializing the SCI, the following procedure should be used for simultaneous serial transmit and receive operations. To switch from transmit mode to simultaneous transmit a receive mode, after checking that the SCI has finished transmission and the TDRE and T flags are set to 1, clear the TE bit to 0. Then simultaneously set both the TE and RE bits t a single instruction. To switch from receive mode to simultaneous transmit and receive m after checking that the SCI has finished reception, clear the RE bit to 0. Then after checki the RDRF bit and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneo both the TE and RE bits to 1 with a single instruction.



Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

ORER flag in SSR, and after performing the appropriate error processing, clear the ORER flag to 0. Reception cannot be resumed i the ORER flag is set to 1.

[4] SCI state check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read th 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) o the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag t 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.
However, the TDRE flag is checke and cleared automatically when th DMAC or DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR. Similarly, the RDRF flag is cleared automatically when the DMAC or DTC is initiated by a receive data full interrupt (RXI) and reads data from RDR.

Figure 15.23 Sample Flowchart of Simultaneous Serial Transmission and Rec

TxD and RxD pins and pull up the data transmission line to V_{CC} using a resistor. Setting t
and TE bits to 1 with the smart card not connected enables closed transmission/reception
self diagnosis. To supply the smart card with the clock pulses generated by the SCI, input
pin output to the CLK pin of the smart card. A reset signal can be supplied via the output
this LSI. (In SCI_5 and SCI-6, the clock generated in SCI cannot be provided to smart ca

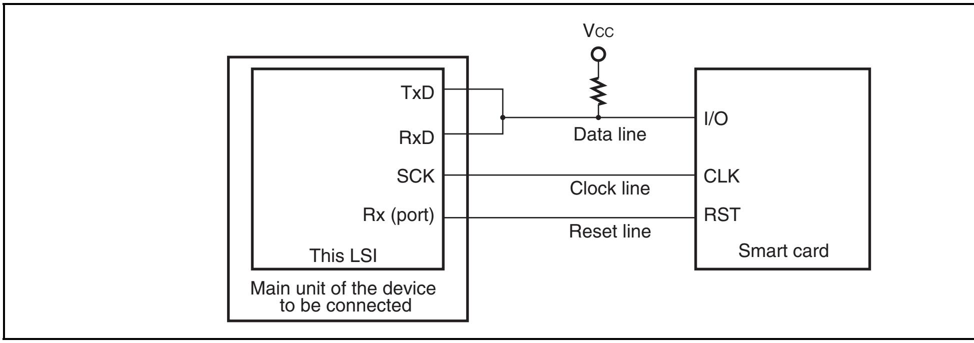


Figure 15.24 Pin Connection for Smart Card Interface

after at least 2 etu.

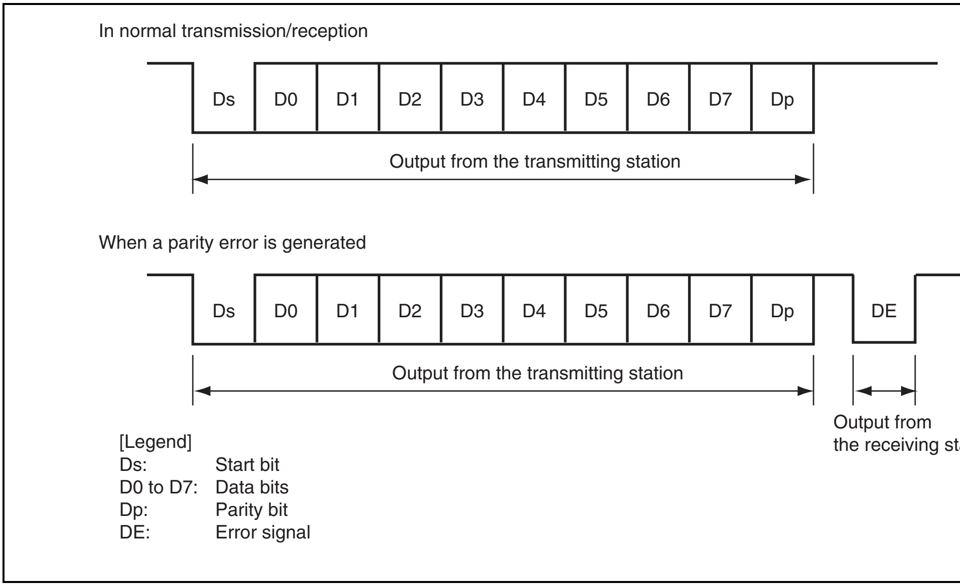


Figure 15.25 Data Formats in Normal Smart Card Interface Mode

For communication with the smart cards of the direct convention and inverse conventio
follow the procedure below.

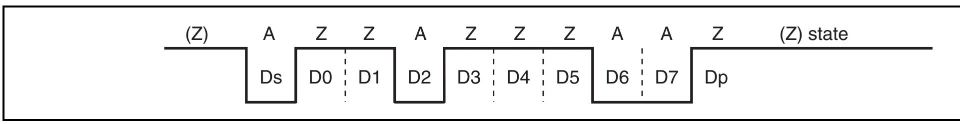


Figure 15.26 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respec
and data is transferred with MSB-first as the start character, as shown in figure 15.27. Th
data in the start character in the figure is H'3F. When using the inverse convention type, v
both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even pa
which is prescribed by the smart card standard, and corresponds to state Z. Since the SNI
this LSI only inverts data bits D7 to D0, write 1 to the O/$\overline{E}$ bit in SMR to invert the parity
both transmission and reception.

15.7.3 Block Transfer Mode

Block transfer mode is different from normal smart card interface mode in the following

- Even if a parity error is detected during reception, no error signal is output. Since the in SSR is set by error detection, clear the PER bit before receiving the parity bit of the frame.
- During transmission, at least 1 etu is secured as a guard time after the end of the parity before the start of the next frame.
- Since the same data is not re-transmitted during transmission, the TEND flag is set 11 after transmission start.
- Although the ERS flag in block transfer mode displays the error signal status as in no smart card interface mode, the flag is always read as 0 because no error signal is trans

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

[Legend]

M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, 256)

D: Duty cycle of clock (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 mar
determined by the formula below.

$$M = (0.5 - \frac{1}{2 \times 372}) \times 100\% = 49.866\%$$

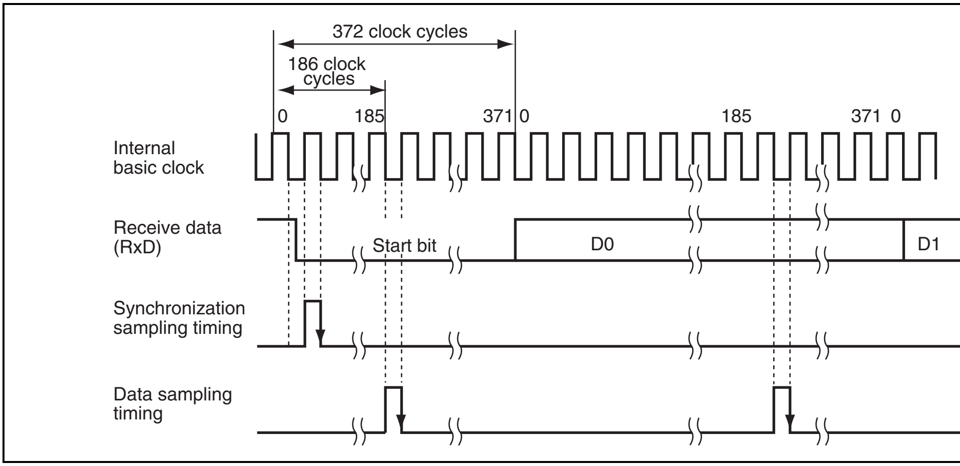


**Figure 15.28 Receive Data Sampling Timing in Smart Card Interface Mo
(When Clock Frequency is 372 Times the Bit Rate)**

5. Set the SMIF, SDIR, and SINV bits in SCMR appropriately. When the DDR correspo the TxD pin is cleared to 0, the TxD and RxD pins are changed from port pins to SCI placing the pins into high impedance state.
6. Set the value corresponding to the bit rate in BRR.
7. Set the CKE1 and CKE0 bits in SCR appropriately. Clear the TIE, RIE, TE, RE, MPI TEIE bits to 0 simultaneously.

   When the CKE0 bit is set to 1, the SCK pin is allowed to output clock pulses.
8. Set the TIE, RIE, TE, and RE bits in SCR appropriately after waiting for at least a 1-b interval. Setting the TE and RE bits to 1 simultaneously is prohibited except for self d

To switch from reception to transmission, first verify that reception has completed, then i the SCI. At the end of initialization, RE and TE should be set to 0 and 1, respectively. Re completion can be verified by reading the RDRF, PER, or ORER flag. To switch from transmission to reception, first verify that transmission has completed, then initialize the S the end of initialization, TE and RE should be set to 0 and 1, respectively. Transmission completion can be verified by reading the TEND flag.

re-transferred from TDR to TSR allowing automatic data retransmission.

3. If no error signal is returned from the receiving end, the ERS bit in SSR is not set to
4. In this case, one frame of data is determined to have been transmitted including re-tr the TEND bit in SSR is set to 1. Here, a TXI interrupt request is generated if the TIE SCR is set to 1. Writing transmit data to TDR starts transmission of the next data.

Figure 15.31 shows a sample flowchart for transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DTC or DMAC. In transmission, the TEND and TDRE flags in SSR are simultaneously set to 1, thus genera TXI interrupt request if the TIE bit in SCR has been set to 1. This activates the DTC or I a TXI request thus allowing transfer of transmit data if the TXI interrupt request is speci source of DTC or DMAC activation beforehand. The TDRE and TEND flags are autom cleared to 0 at data transfer by the DTC or DMAC. If an error occurs, the SCI automatic transmits the same data. During re-transmission, TEND remains as 0, thus not activating or DMAC. Therefore, the SCI and DTC or DMAC automatically transmit the specified bytes, including re-transmission in the case of error occurrence. However, the ERS flag automatically cleared; the ERS flag must be cleared by previously setting the RIE bit to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DTC or DMAC, be sure to set and enable th DMAC prior to making SCI settings. For DTC or DMAC settings, see section 8, Data T Controller (DTC) and section 7, DMA Controller (DMAC).

Figure 15.29 Data Re-Transfer Operation in SCI Transmission Mode

Note that the TEND flag is set in different timings depending on the GM bit setting in SM
Figure 15.30 shows the TEND flag set timing.

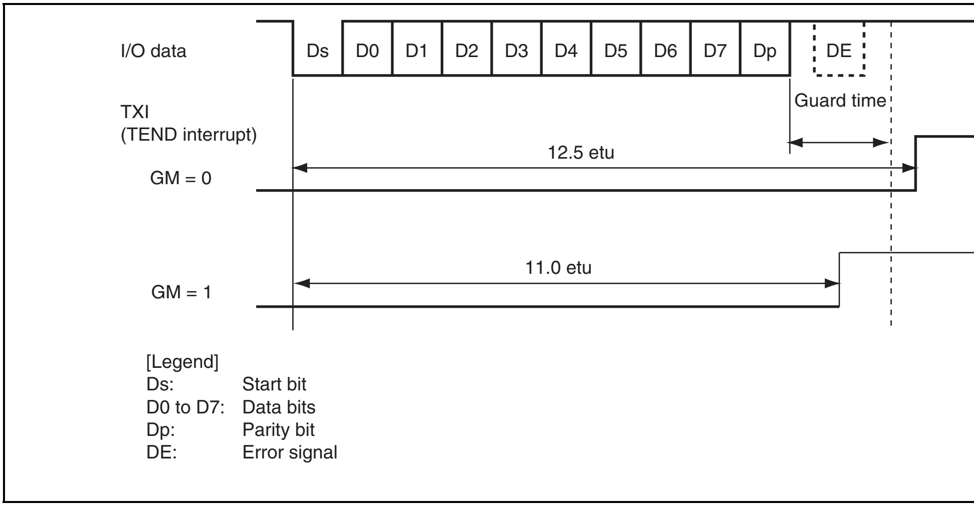


Figure 15.30 TEND Flag Set Timing during Transmission

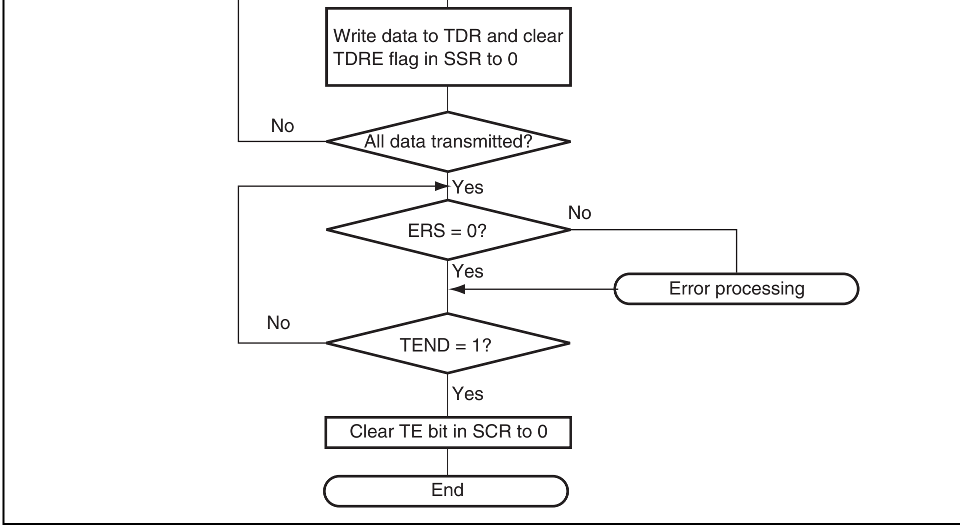


Figure 15.31 Sample Transmission Flowchart

4. In this case, data is determined to have been received successfully, and the RDRF bit set to 1. Here, an RXI interrupt request is generated if the RIE bit in SCR is set to 1.

Figure 15.33 shows a sample flowchart for reception. All the processing steps are automa performed using an RXI interrupt request to activate the DTC or DMAC. In reception, se RIE bit to 1 allows an RXI interrupt request to be generated when the RDRF flag is set to activates the DTC or DMAC by an RXI request thus allowing transfer of receive data if t interrupt request is specified as a source of DTC or DMAC activation beforehand. The R is automatically cleared to 0 at data transfer by the DTC or DMAC. If an error occurs dur reception, i.e., either the ORER or PER flag is set to 1, a transmit/receive error interrupt (request is generated and the error flag must be cleared. If an error occurs, the DTC or DM not activated and receive data is skipped, therefore, the number of bytes of receive data sp in the DTC or DMAC is transferred. Even if a parity error occurs and the PER bit is set to reception, receive data is transferred to RDR, thus allowing the data to be read.

Note: For operations in block transfer mode, see section 15.4, Operation in Asynchrono

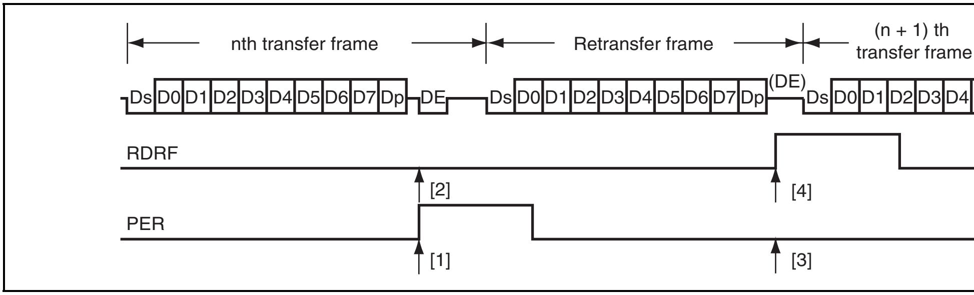


Figure 15.32 Data Re-Transfer Operation in SCI Reception Mode

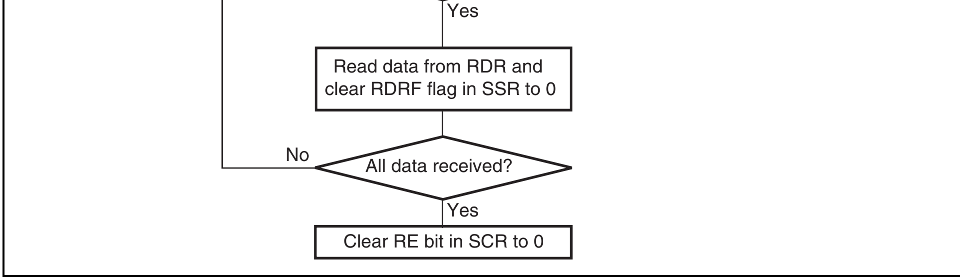


Figure 15.33 Sample Reception Flowchart

15.7.8 Clock Output Control

Clock output can be fixed using the CKE1 and CKE0 bits in SCR when the GM bit in S
to 1. Specifically, the minimum width of a clock pulse can be specified.

Figure 15.34 shows an example of clock output fixing timing when the CKE0 bit is cont
with GM = 1 and CKE1 = 0.

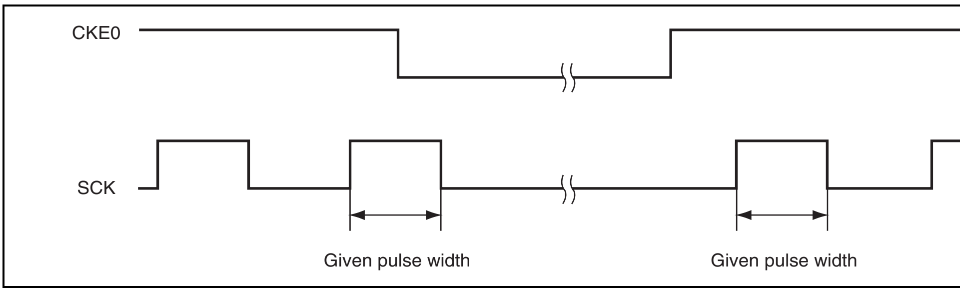


Figure 15.34 Clock Output Fixing Timing

Set the CKE0 bit in SCR to 1 to start clock output.

- At mode switching
  - — At transition from smart card interface mode to software standby mode
    1. Set the data register (DR) and data direction register (DDR) corresponding to t pin to the values for the output fixed state in software standby mode. (SCI_0, 1 4 only)
    2. Write 0 to the TE and RE bits in SCR to stop transmission/reception. Simultan set the CKE1 bit to the value for the output fixed state in software standby mo
    3. Write 0 to the CKE0 bit in SCR to stop the clock.
    4. Wait for one cycle of the serial clock. In the mean time, the clock output is fix specified level with the duty cycle retained.
    5. Make the transition to software standby mode.
  - — At transition from smart card interface mode to software standby mode
    1. Clear software standby mode.
    2. Write 1 to the CKE0 bit in SCR to start clock output. A clock signal with the appropriate duty cycle is then generated.

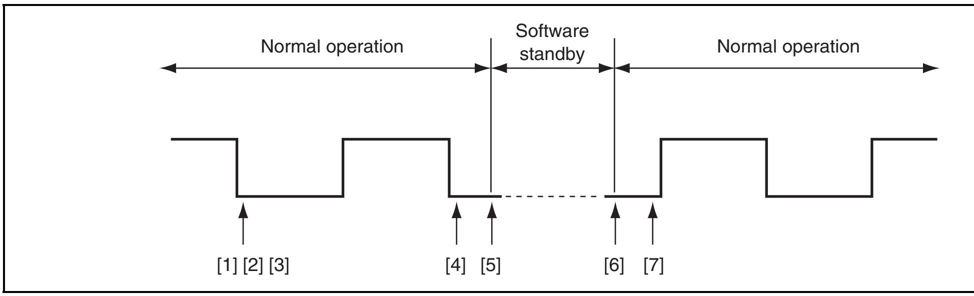


Figure 15.35 Clock Stop and Restart Procedure

rate, the transfer rate must be modified through programming.

Figure 15.36 is the IrDA block diagram.

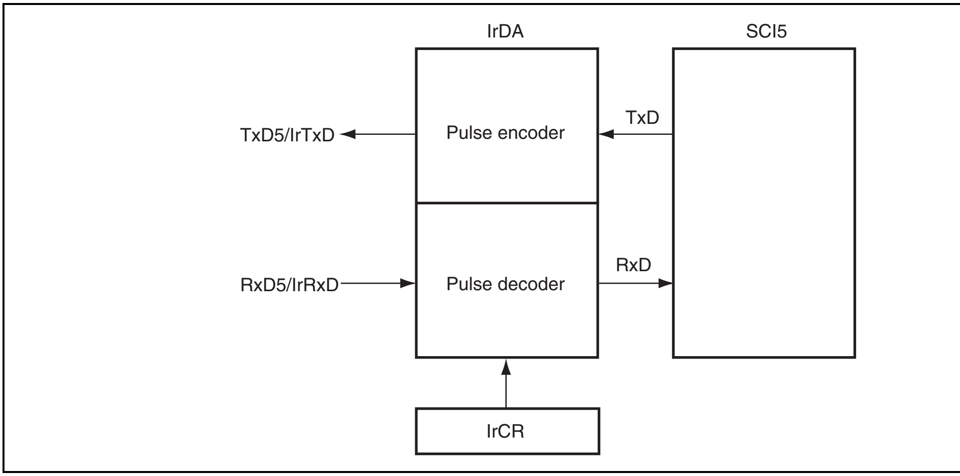


Figure 15.36 IrDA Block Diagram

MHz, a high-level pulse width of 1.6 μs can be specified because it is the smallest value i
range greater than 1.41 μs.

For serial data of level 1, no pulses are output.

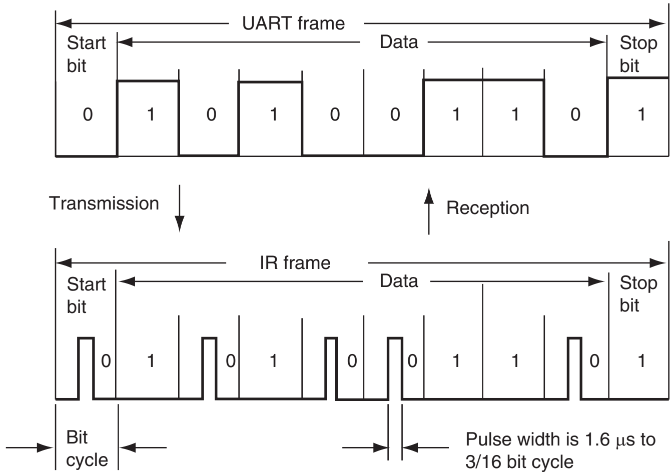


Figure 15.37 IrDA Transmission and Reception

(2) Reception

During reception, IR frames are converted to UART frames using the IrDA interface befo
inputting to SCI. 0 is output when the high level pulse is detected while 1 is output when
is detected during one bit period. Note that a pulse shorter than the minimum pulse width
μs is also regarded as a 0 signal.

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
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
30	110	110	110	110	110	110
33	110	110	110	110	110	110
35	110	110	110	110	110	110

DTC or DMAC to allow data transfer. The TDRE flag is automatically cleared to 0 at dat
by the DTC or DMAC.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the
PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interru
activate the DTC or DMAC to allow data transfer. The RDRF flag is automatically cleare
data transfer by the DTC or DMAC.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to 1
interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority
acceptance. However, note that if the TDRE and TEND flags are cleared to 0 simultaneou
the TXI interrupt processing routine, the SCI cannot branch to the TEI interrupt processin
later.

Note that the priority order for interrupts is different between the group of SCI_0, 1, 2, an
the group of SCI_5 and SCI_6.

Table 15.14 SCI Interrupt Sources (SCI_0, 1, 2, and 4)

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ERI	Receive error	ORER, FER, or PER	Not possible	Not possible
RXI	Receive data full	RDRF	Possible	Possible
TXI	Transmit data empty	TDRE	Possible	Possible
TEI	Transmit end	TEND	Not possible	Not possible

Table 15.16 shows the interrupt sources in smart card interface mode. A transmit end (T
interrupt request cannot be used in this mode.

Note that the priority order for interrupts is different between the group of SCI_0, 1, 2, a
the group of SCI_5 and SCI_6.

Table 15.16 SCI Interrupt Sources (SCI_0, 1, 2, and 4)

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	Not possible
RXI	Receive data full	RDRF	Possible	Possible
TXI	Transmit data empty	TEND	Possible	Possible

Table 15.17 SCI Interrupt Sources (SCI_5 and SCI_6)

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
RXI	Receive data full	RDRF	Not possible	Possible
TXI	Transmit data empty	TDRE	Not possible	Possible
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	Not possible

error occurrence.

When transmitting/receiving data using the DTC or DMAC, be sure to set and enable the DMAC prior to making SCI settings. For DTC or DMAC settings, see section 8, Data Tra Controller (DTC) and section 7, DMA Controller (DMAC).

In reception, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1 activates the DTC or DMAC by an RXI request thus allowing transfer of receive data if t request is specified as a source of DTC or DMAC activation beforehand. The RDRF flag automatically cleared to 0 at data transfer by the DTC or DMAC. If an error occurs, the R flag is not set but the error flag is set. Therefore, the DTC or DMAC is not activated and interrupt request is issued to the CPU instead; the error flag must be cleared.

When framing error detection is performed, a break can be detected by reading the RxD
directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag is se
PER flag may also be set. Note that, since the SCI continues the receive operation even
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 Detection

When the TE bit is 0, the TxD pin is used as an I/O port whose direction (input or outpu
level are determined by DR and DDR. This can be used to set the TxD pin to mark state
level) or send a break during serial data transmission. To maintain the communication li
state (the state of 1) until TE is set to 1, set both DDR and DR to 1. Since the TE bit is c
at this point, the TxD pin becomes an I/O port, and 1 is output from the TxD pin. To sen
during serial transmission, first set DDR to 1 and DR to 0, and then clear the TE bit to 0
TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission
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 Mo

Transmission cannot be started when a receive error flag (ORER, FER, or RER) is set to
the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting
transmission. Note also that the receive error flags cannot be cleared to 0 even if the RE
cleared to 0.

15.10.6 Restrictions on Using DTC or DMAC

- When the external clock source is used as a synchronization clock, update TDR by the or DTC and wait for at least five Pϕ clock cycles before allowing the transmit clock to input. If the transmit clock is input within four clock cycles after TDR modification, t may malfunction (see figure 15.38).
- When using the DMAC or DTC to read RDR, be sure to set the receive end interrupt the DTC or DMAC activation source.

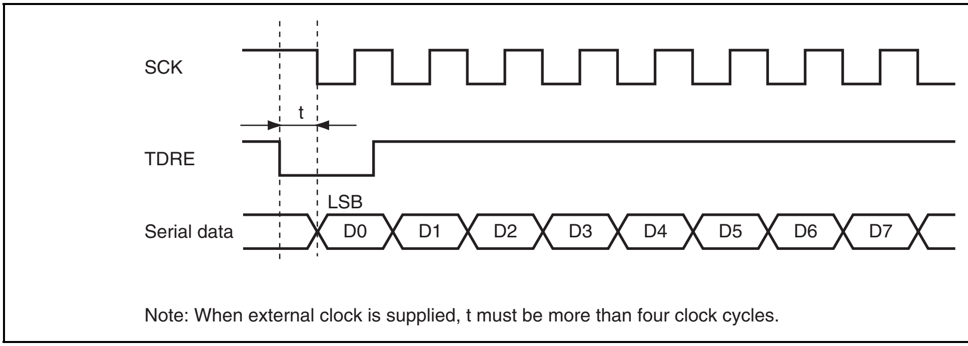


Figure 15.38 Sample Transmission using DTC in Clocked Synchronous Mo

- The DTC is not activated by the RXI or TXI request by SCI_5 or SCI6.

Figure 15.39 shows a sample flowchart for transition to software standby mode during transmission. Figures 15.40 and 15.41 show the port pin states during transition to softw standby mode.

Before specifying the module stop state or making a transition to software standby mode transmission mode using DTC transfer, stop all transmit operations (TE = TIE = TEIE = Setting the TE and TIE bits to 1 after cancellation sets the TXI flag to start transmission DTC.

Reception: Before specifying the module stop state or making a transition to software s mode, stop the receive operations (RE = 0). RSR, RDR, and SSR are reset. If transition during data reception, the data being received will be invalid.

To receive data in the same reception mode after cancellation of the power-down state, s bit to 1, and then start reception. To receive data in a different reception mode, initialize first.

For using the IrDA function, set the IrE bit in addition to setting the RE bit.

Figure 15.42 shows a sample flowchart for mode transition during reception.

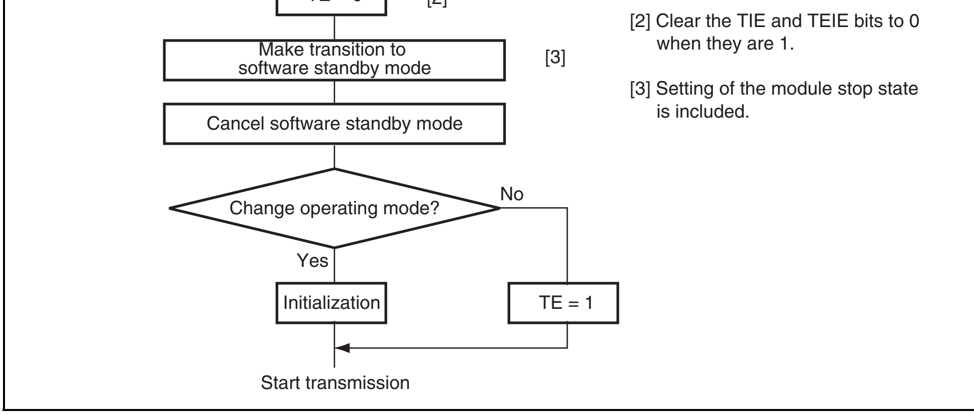


**Figure 15.39 Sample Flowchart for Software Standby Mode Transition duri
Transmission**

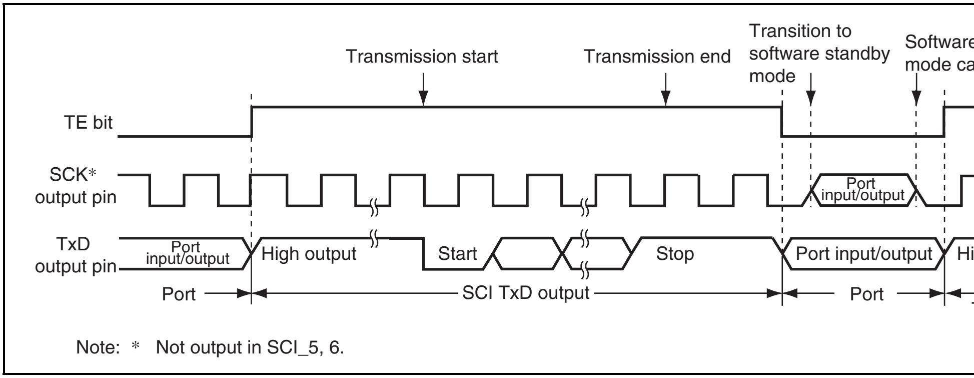


**Figure 15.40 Port Pin States during Software Standby Mode Transition
(Internal Clock, Asynchronous Transmission)**

Figure 15.41 Port Pin States during Software Standby Mode Transition
(Internal Clock, Clocked Synchronous Transmission)
(Setting is Prohibited in SCI_5 and SCI_6)

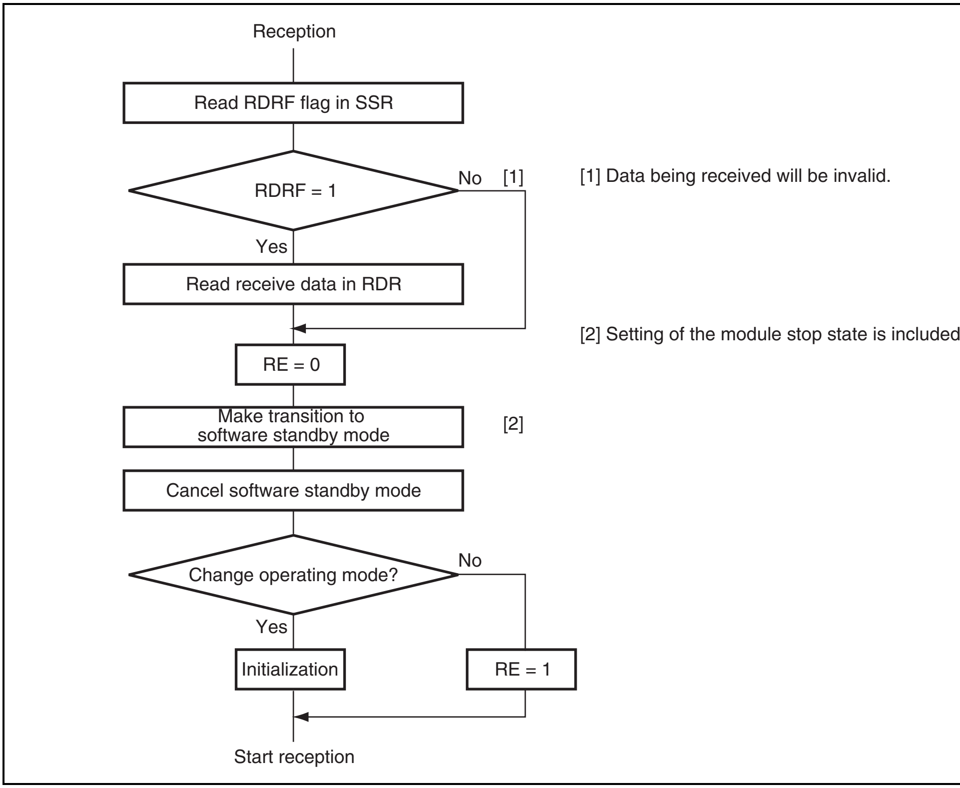


Figure 15.42 Sample Flowchart for Software Standby Mode Transition during R

- One of three generating polynomials selectable
- CRC code generation for LSB-first or MSB-first communication selectable

Figure 15.43 shows a block diagram of the CRC operation circuit.

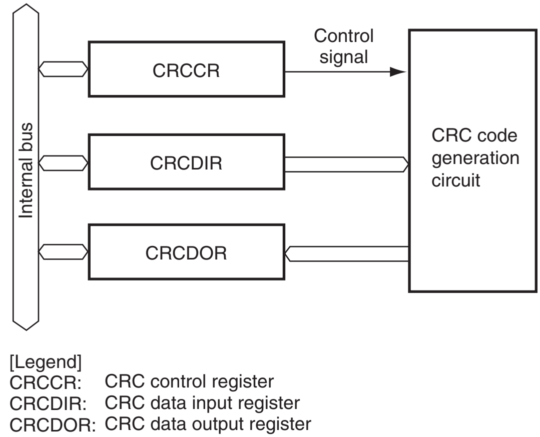


Figure 15.43 Block Diagram of CRC Operation Circuit

generating polynomial.

Bit	7	6	5	4	3	2	1
Bit Name	DORCLR	—	—	—	—	LMS	G1
Initial Value	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	DORCLR	0	W	CRCDOR Clear Setting this bit to 1 clears CRCDOR to H'000(
6 to 3	—	All 0	R	Reserved The initial value should not be changed.
2	LMS	0	R/W	CRC Operation Switch Selects CRC code generation for LSB-first or communication. 0: Performs CRC operation for LSB-first communication. The lower byte (bits 7 to 0 transmitted when CRCDOR contents (CRC are divided into two bytes to be transmitted parts. 1: Performs CRC operation for MSB-first communication. The upper byte (bits 15 to transmitted when CRCDOR contents (CRC are divided into two bytes to be transmitted parts.

CRCDIR is an 8-bit readable/writable register, to which the bytes to be CRC-operated are
The result is obtained in CRCDOR.

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(3) CRC Data Output Register (CRCDOR)

CRCDOR is a 16-bit readable/writable register that contains the result of CRC operation
bytes to be CRC-operated are written to CRCDIR after CRCDOR is cleared. When the C
operation result is additionally written to the bytes to which CRC operation is to be perfo
CRC operation result will be H'0000 if the data contains no CRC error. When bits 1 and 0
CRCCR (G1 and G0 bits) are set to 0 and 1, respectively, the lower byte of this register c
the result.

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

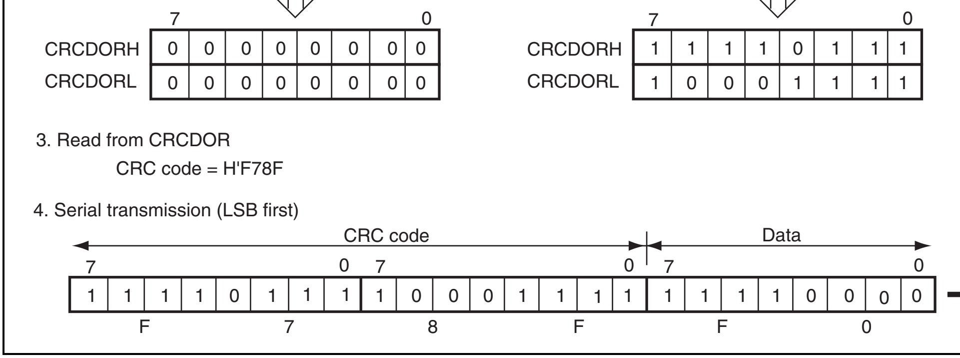


Figure 15.44 LSB-First Data Transmission

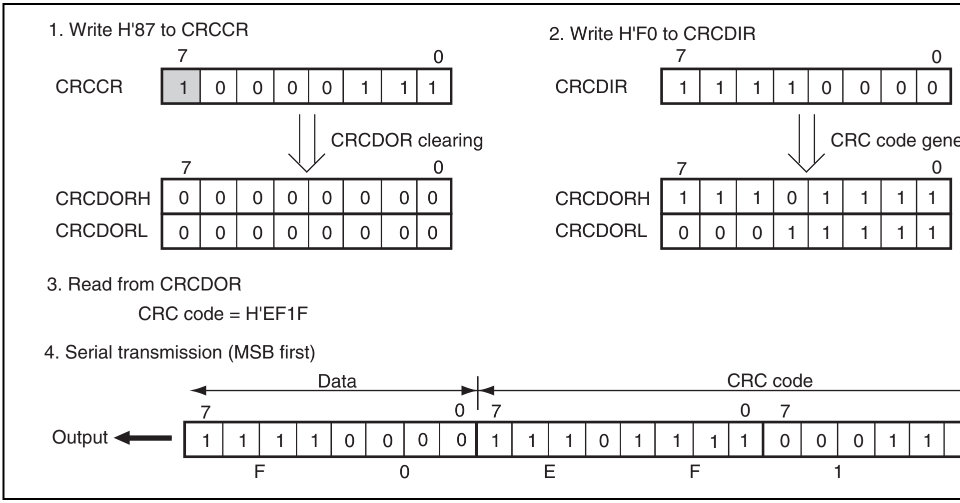


Figure 15.45 MSB-First Data Transmission

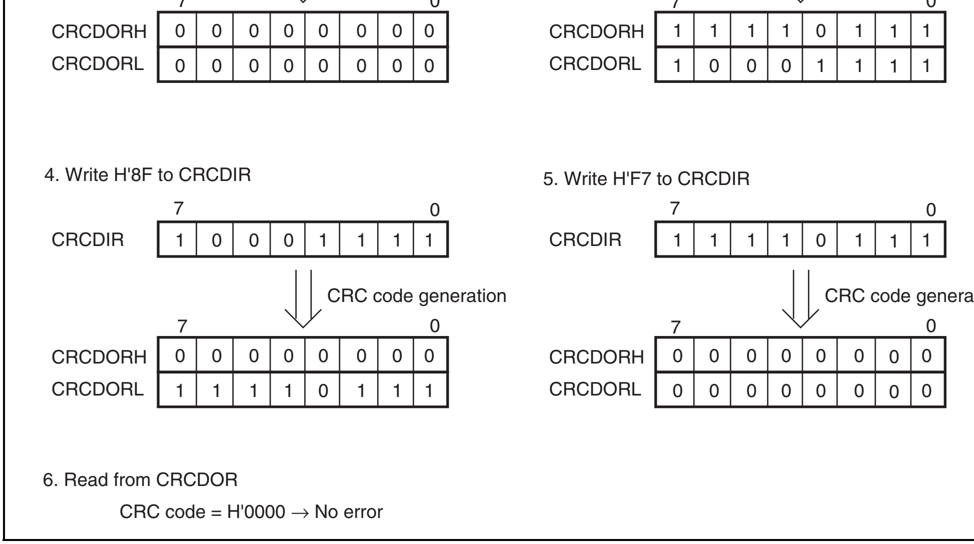


Figure 15.46 LSB-First Data Reception

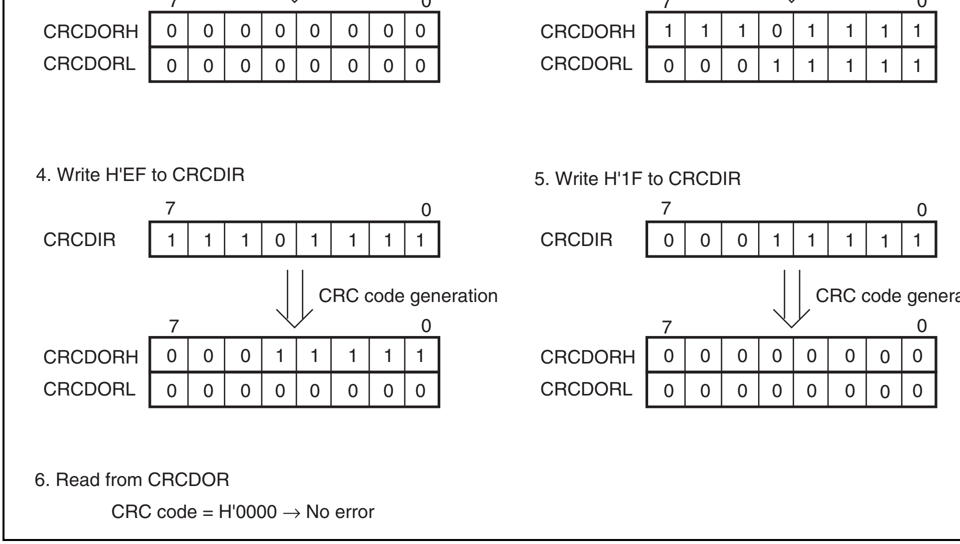


Figure 15.47 MSB-First Data Reception

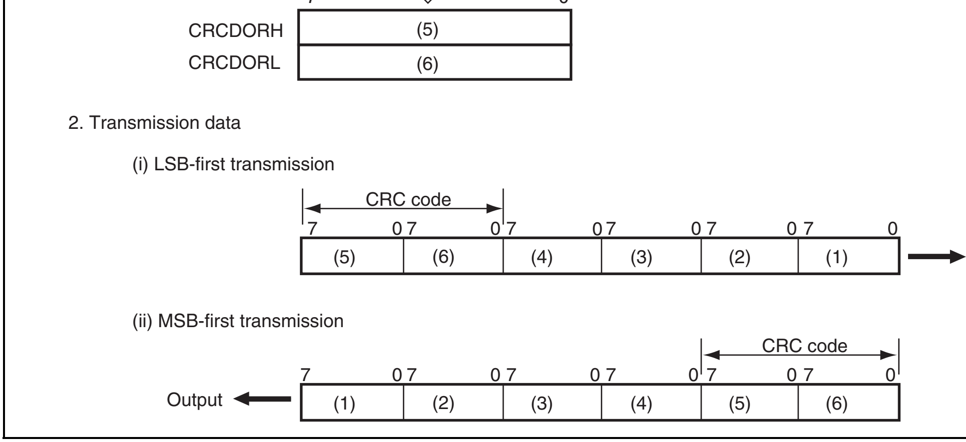


Figure 15.48 LSB-First and MSB-First Transmit Data

- Transfer speed: Supports full-speed (12 Mbps)
- Endpoint configuration:

Endpoint Name	Abbreviation	Transfer Type	Maximum Packet Size	FIFO Buffer Capacity (Byte)	DMA Tr
Endpoint 0	EP0s	Setup	8	8	—
	EP0i	Control-in	8	8	—
	EP0o	Control-out	8	8	—
Endpoint 1	EP1	Bulk-out	64	128	Possible
Endpoint 2	EP2	Bulk-in	64	128	Possible
Endpoint 3	EP3	Interrupt-in	8	8	—

Configuration1-Interface0-AlternateSetting0
- EndPoint1
- EndPoint2
- EndPoint3

- Interrupt requests: Generates various interrupt signals necessary for USB transmission/reception
- Power mode: Self power mode or bus power mode can be selected by the power mo (PWMD) in the control register (CTLR).

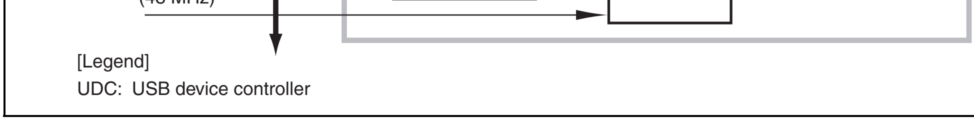


Figure 16.1 Block Diagram of USB

16.2 Input/Output Pins

Table 16.1 shows the USB pin configuration.

Table 16.1 Pin Configuration

Pin Name	I/O	Function
VBUS	Input	USB cable connection monitor pin
USD+	I/O	USB data I/O pin
USD-	I/O	USB data I/O pin
DrVcc	Input	Power supply pin for USB on-chip transceiver
DrVss	Input	Ground pin for USB on-chip transceiver

- Interrupt select register 2 (ISR2)
- Interrupt enable register 0 (IER0)
- Interrupt enable register 1 (IER1)
- Interrupt enable register 2 (IER2)
- EP0i data register (EPDR0i)
- EP0o data register (EPDR0o)
- EP0s data register (EPDR0s)
- EP1 data register (EPDR1)
- EP2 data register (EPDR2)
- EP3 data register (EPDR3)
- EP0o receive data size register (EPSZ0o)
- EP1 receive data size register (EPSZ1)
- Trigger register (TRG)
- Data status register (DASTS)
- FIFO clear register (FCLR)
- DMA transfer setting register (DMA)
- Endpoint stall register (EPSTL)
- Configuration value register (CVR)
- Control register (CTLR)
- Endpoint information register (EPIR)
- Transceiver test register 0 (TRNTREG0)
- Transceiver test register 1 (TRNTREG1)

Initial Value	0	0	0	1	0	0	0
R/W	R/W	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	BRST	0	R/W	Bus Reset This bit is set to 1 when a bus reset signal is det the USB bus. (When the CPU is used to clear this flag by writi while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)
6	EP1FULL	0	R	EP1 FIFO Full This bit is set when endpoint 1 receives one pac data successfully from the host, and holds a valu as long as there is valid data in the FIFO buffer. This is a status bit, and cannot be cleared.
5	EP2TR	0	R/W	EP2 Transfer Request This bit is set if there is no valid transmit data in buffer when an IN token for endpoint 2 is receive the host. A NACK handshake is returned to the data is written to the FIFO buffer and packet transmission is enabled. (When the CPU is used to clear this flag by writi while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)

				(When the CPU is used to clear this flag by wri while the corresponding interrupt is enabled, b read the flag after writing 0 to it.)
2	EP0oTS	0	R/W	EP0o Receive Complete This bit is set to 1 when endpoint 0 receives da the host successfully, stores the data in the FI and returns an ACK handshake to the host. (When the CPU is used to clear this flag by wri while the corresponding interrupt is enabled, b read the flag after writing 0 to it.)
1	EP0iTR	0	R/W	EP0i Transfer Request This bit is set if there is no valid transmit data i FIFO buffer when an IN token for endpoint 0 is from the host. A NACK handshake is returned host until data is written to the FIFO buffer and transmission is enabled. (When the CPU is used to clear this flag by wri while the corresponding interrupt is enabled, b read the flag after writing 0 to it.)
0	EP0iTS	0	R/W	EP0i Transmit Complete This bit is set when data is transmitted to the h endpoint 0 and an ACK handshake is returned (When the CPU is used to clear this flag by wri while the corresponding interrupt is enabled, b read the flag after writing 0 to it.)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write valu should always be 0.
5	—	0	R	
4	—	0	R	
3	VBUS MN	0	R	This is a status bit which monitors the state of t VBUS pin. This bit reflects the state of the VBUS pin and generates no interrupt request. This bit is alwa when the PULLUP_E bit in DMA is 0.
2	EP3 TR	0	R/W	EP3 Transfer Request This bit is set if there is no valid transmit data i FIFO buffer when an IN token for endpoint 3 is received from the host. A NACK handshake is to the host until data is written to the FIFO buff packet transmission is enabled. (When the CPU is used to clear this flag by wri while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)
1	EP3 TS	0	R/W	EP3 Transmit Complete This bit is set when data is transmitted to the h endpoint 3 and an ACK handshake is returned (When the CPU is used to clear this flag by wri while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)

16.3.3 Interrupt Flag Register 2 (IFR2)

IFR2, together with interrupt flag registers 0 and 1 (IFR0 and IFR1), indicates interrupt information required by the application. When an interrupt source is generated, the corre bit is set to 1. And then this bit, in combination with interrupt enable register 2 (IER2), g an interrupt request to the CPU. To clear, write 0 to the bit to be cleared and 1 to the oth

Bit	7	6	5	4	3	2	1
Bit Name	—	—	SURSS	SURSF	CFDN	—	SETC
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write va should always be 0.
5	SURSS	0	R	Suspend/Resume Status This is a status bit that describes bus state. 0: Normal state 1: Suspended state This bit is a status bit and generates no interr request.

				information register to the EPIR register ends (end). This module starts the USB operation aft endpoint information is completely set. (When the CPU is used to clear this flag by wri while the corresponding interrupt is enabled, b read the flag after writing 0 to it.)
2	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0. (When the CPU is used to clear this flag by wri while the corresponding interrupt is enabled, b read the flag after writing 0 to it.)
1	SETC	0	R/W	Set_Configuration Command Detection When the Set_Configuration command is dete bit is set to 1. (When the CPU is used to clear this flag by wri while the corresponding interrupt is enabled, b read the flag after writing 0 to it.)
0	SETI	0	R/W	Set_Interface Command Detection When the Set_Interface command is detected, is set to 1. (When the CPU is used to clear this flag by wri while the corresponding interrupt is enabled, b read the flag after writing 0 to it.)

R/W R/W R/W R/W R/W R/W R/W R/W

Bit	Bit Name	Initial Value	R/W	Description
7	BRST	0	R/W	Bus Reset
6	EP1 FULL	0	R/W	EP1 FIFO Full
5	EP2 TR	0	R/W	EP2 Transfer Request
4	EP2 EMPTY	0	R/W	EP2 FIFO Empty
3	SETUP TS	0	R/W	Setup Command Receive Complete
2	EP0o TS	0	R/W	EP0o Receive Complete
1	EP0i TR	0	R/W	EP0i Transfer Request
0	EP0i TS	0	R/W	EP0i Transmission Complete

R/W	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write valu should always be 0.
5	—	0	R	
4	—	0	R	
3	—	0	R	
2	EP3 TR	1	R/W	EP3 Transfer Request
1	EP3 TS	1	R/W	EP3 Transmission Complete
0	VBUSF	1	R/W	USB Bus Connect

R/W	R	R	R	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write va
5	—	0	R	should always be 0.
4	SURSE	1	R/W	Suspend/Resume Detection
3	CFDN	1	R/W	End Point Information Load End
2	—	1	R	Reserved This bit is always read as 1. The write value s always be 1.
1	SETCE	1	R/W	Set_Configuration Command Detection
0	SETIE	1	R/W	Set_Interface Command Detection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	BRST	0	R/W	Bus Reset
6	EP1 FULL	0	R/W	EP1 FIFO Full
5	EP2 TR	0	R/W	EP2 Transfer Request
4	EP2 EMPTY	0	R/W	EP2 FIFO Empty
3	SETUP TS	0	R/W	Setup Command Receive Complete
2	EP0o TS	0	R/W	EP0o Receive Complete
1	EP0i TR	0	R/W	EP0i Transfer Request
0	EP0i TS	0	R/W	EP0i Transmission Complete

R/W	R	R	R	R	R	R/W	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write va
5	—	0	R	should always be 0.
4	—	0	R	
3	—	0	R	
2	EP3 TR	0	R/W	EP3 Transfer Request
1	EP3 TS	0	R/W	EP3 Transmission Complete
0	VBUSF	0	R/W	USB Bus Connect

16.3.9 Interrupt Enable Register 2 (IER2)

IER2 enables the interrupt requests of interrupt flag register 2 (IFR2). When an interrupt to 1 while the corresponding bit of each interrupt is set to 1, an interrupt request is sent t CPU. The interrupt vector number is determined by the contents of interrupt select regis (ISR2).

Bit	7	6	5	4	3	2	1
Bit Name	SSRSME	—	—	SURSE	CFDN	—	SETCE
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R	R	R/W	R/W	R	R/W

				Suspend and Resume Operations.
3	CFDN	0	R/W	End Point Information Load End
2	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
1	SETCE	0	R/W	Set_Configuration Command Detection
0	SETIE	0	R/W	Set_Interface Command Detection

16.3.10 EP0i Data Register (EPDR0i)

EPDR0i is an 8-byte transmit FIFO buffer for endpoint 0. EPDR0i holds one packet of tra data for control-in. Transmit data is fixed by writing one packet of data and setting EP0iP the trigger register. When an ACK handshake is returned from the host after the data has transmitted, EP0iTS in interrupt flag register 0 is set. This FIFO buffer can be initialized of EP0iCLR in the FCLR register.

Bit	7	6	5	4	3	2	1	
Bit Name	D7	D6	D5	D4	D3	D2	D1	
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Ur
R/W	W	W	W	W	W	W	W	

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for control-in transfer

Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	R	Data register for control-out transfer

16.3.12 EP0s Data Register (EPDR0s)

EPDR0s is an 8-byte FIFO buffer specifically for receiving endpoint 0 setup commands setup command to be processed by the application is received. When command data is r successfully, the SETUPTS bit in interrupt flag register 0 is set.

As a latest setup command must be received in high priority, if data is left in this buffer, overwritten with new data. If reception of the next command is started while the current is being read, command reception has priority, the read by the application is forcibly sto the read data is invalid.

Bit	7	6	5	4	3	2	1
Bit Name	D7	D6	D5	D4	D3	D2	D1
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	R	Data register for storing the setup command a control-out transfer

Bit	7	6	5	4	3	2	1
Bit Name	D7	D6	D5	D4	D3	D2	D1
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	R	Data register for endpoint 1 transfer

16.3.14 EP2 Data Register (EPDR2)

EPDR2 is a 128-byte transmit FIFO buffer for endpoint 2. EPDR2 has a dual-buffer conf and has a capacity of twice the maximum packet size. When transmit data is written to thi buffer and EP2PKTE in the trigger register is set, one packet of transmit data is fixed, and dual-FIFO buffer is switched over. The transmit data for this FIFO buffer can be transferr DMA. This FIFO buffer can be initialized by means of EP2CLR in the FCLR register.

Bit	7	6	5	4	3	2	1	
Bit Name	D7	D6	D5	D4	D3	D2	D1	
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Ur
R/W	W	W	W	W	W	W	W	

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for endpoint 2 transfer

Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	U
R/W	W	W	W	W	W	W	W	

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for endpoint 3 transfer

16.3.16 EP0o Receive Data Size Register (EPSZ0o)

EPSZ0o indicates the number of bytes received at endpoint 0 from the host.

Bit	7	6	5	4	3	2	1
Bit Name	—	—	—	—	—	—	—
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Number of receive data for endpoint 0

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Number of received bytes for endpoint 1

16.3.18 Trigger Register (TRG)

TRG generates one-shot triggers to control the transfer sequence for each endpoint.

Bit	7	6	5	4	3	2	1	
Bit Name	—	EP3 PKTE	EP1 RDFN	EP2 PKTE	—	EP0s RDFN	EP0o RDFN	EP
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Ur
R/W	—	W	W	W	—	W	W	

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved The write value should always be 0.
6	EP3 PKTE	Undefined	W	EP3 Packet Enable After one packet of data has been written to t endpoint 3 transmit FIFO buffer, the transmit fixed by writing 1 to this bit.

				fixed by writing 1 to this bit.
3	—	Undefined	—	Reserved The write value should always be 0.
2	EP0s RDFN	Undefined	W	EP0s Read Complete Write 1 to this bit after data for the EP0s con FIFO has been read. Writing 1 to this bit ena transfer of data in the following data stage. A handshake is returned in response to transfe requests from the host in the data stage unti written to this bit.
1	EP0o RDFN	Undefined	W	EP0o Read Complete Writing 1 to this bit after one packet of data h read from the endpoint 0 transmit FIFO buffe initializes the FIFO buffer, enabling the next be received.
0	EP0i PKTE	Undefined	W	EP0i Packet Enable After one packet of data has been written to endpoint 0 transmit FIFO buffer, the transmi fixed by writing 1 to this bit.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write va should always be 0.
5	EP3 DE	0	R	EP3 Data Present This bit is set when the endpoint 3 FIFO buffe contains valid data.
4	EP2 DE	0	R	EP2 Data Present This bit is set when the endpoint 2 FIFO buffe contains valid data.
3	—	0	R	Reserved
2	—	0	R	These bits are always read as 0.
1	—	0	R	
0	EP0i DE	0	R	EP0i Data Present This bit is set when the endpoint 0 FIFO buffe contains valid data.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved The write value should always be 0.
6	EP3 CLR	Undefined	W	EP3 Clear Writing 1 to this bit initializes the endpoint 3 FIFO buffer.
5	EP1 CLR	Undefined	W	EP1 Clear Writing 1 to this bit initializes both sides of th endpoint 1 receive FIFO buffer.
4	EP2 CLR	Undefined	W	EP2 Clear Writing 1 to this bit initializes both sides of th endpoint 2 transmit FIFO buffer.
3	—	Undefined	—	Reserved
2	—		—	The write value should always be 0.
1	EP0o CLR	Undefined	W	EP0o Clear Writing 1 to this bit initializes the endpoint 0 FIFO buffer.
0	EP0i CLR	Undefined	W	EP0i Clear Writing 1 to this bit initializes the endpoint 0 FIFO buffer.

R/W	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write va should always be 0.
5	—	0	R	
4	—	0	R	
3	—	0	R	
2	PULLUP_E	0	R/W	PULLUP Enable This pin performs the pull-up control for the D with using PM4 as the pull-up control pin. 0: D+ is not pulled up. 1: D+ is pulled up.

(USBINTN1) is asserted again. However, if t
the data packet to be transmitted is less thar
bytes, the EP2 packet enable bit is not set
automatically, and so should be set by the C
DMA transfer end interrupt.

As EP2-related interrupt requests to the CPU
automatically masked, interrupt requests sho
masked as necessary in the interrupt enable

- Operating procedure

1. Write of 1 to the EP2 DMAE bit in DMAR
2. Set the DMAC to activate through USBIN
3. Transfer count setting in the DMAC
4. DMAC activation
5. DMA transfer
6. DMA transfer end interrupt generated

See section 16.8.3, DMA Transfer for Endpo

EP1-related interrupt requests to the CPU are
automatically masked.

- Operating procedure:

1. Write of 1 to the EP1 DMAE bit in DMA
2. Set the DMAC to activate through USBIN
3. Transfer count setting in the DMAC
4. DMAC activation
5. DMA transfer
6. DMA transfer end interrupt generated

See section 16.8.2, DMA Transfer for Endpoi

Bit Name	—	—	—	—	EP3STL	EP2STL	EP1STL
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write v
5	—	0	R	should always be 0.
4	—	0	R	
3	EP3STL	0	R/W	EP3 Stall When this bit is set to 1, endpoint 3 is placed stall state.
2	EP2STL	0	R/W	EP2 Stall When this bit is set to 1, endpoint 2 is placed stall state.
1	EP1STL	0	R/W	EP1 Stall When this bit is set to 1, endpoint 1 is placed stall state.
0	EP0STL	0	R/W	EP0 Stall When this bit is set to 1, endpoint 0 is placed stall state.

Bit	Bit Name	Initial Value	R/W	Description
7 6	CNFV1 CNFV0	All 0	R	These bits store Configuration Setting value w they receive Set Configuration command. CN updated when the SETC bit in IFR2 is set to 1
5 4	INTV1 INTV0	All 0	R	These bits store Interface Setting value when receive Set Interface command. INTV is upda when the SETI bit in IFR2 is set to 1.
3	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
2	ALTV2	0	R	These bits store Alternate Setting value when receive Set Interface command. ALTV2 to AL updated when the SETI bit in IFR2 is set to 1.
1	ALTV1	0	R	
0	ALTV0	0	R	

16.3.24 Control Register (CTLR)

This register sets functions for bits ASCE, PWMD, RSME, and, PWUPS.

Bit	7	6	5	4	3	2	1
Bit Name	—	—	—	RWUPS	RSME	PWMD	ASCE
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W

				Feature request. This bit is set to 1 when ren wakeup command is enabled.
3	RSME	0	R/W	Resume Enable This bit releases the suspend state (or execu remote wakeup). When RSME is set to 1, re request starts. If RSME is once set to 1, clea to 0 again afterwards. In this case, the value RSME must be kept for at least one clock pe 12-MHz clock.
2	PWMD	0	R/W	Bus Power Mode This bit specifies the USB power mode. Whe is set to 0, the self-power mode is selected f module. When set to 1, the bus-power mode selected.
1	ASCE	0	R/W	Automatic Stall Clear Enable Setting the ASCE bit to 1 automatically clear setting bit (the EPxSTL (x = 1, 2, or 3) bit in or EPSTR1) of the end point that has returne stall handshake to the host. The automatic s enable is common to the all end points. Thus individual control of the end point is not poss When the ASCE bit is set to 0, the stall settin not automatically cleared. This bit must be re by the users. To enable this bit, make sure th ASCE bit should be set to 1 before the EPxS 2, or 3) bit in EPSTL is set to 1.
0	—	0	R	Reserved This bit is always read as 0. The write value always be 0.

Bit	7	6	5	4	3	2	1
Bit Name	D7	D6	D5	D4	D3	D2	D1
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	W	W	W	W	W	W	W

• EPIR00

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	D7 to D4	Undefined	W	Endpoint Number [Enable setting range] 0 to 3
3, 2	D3, D2	Undefined	W	Endpoint Configuration Number [Enable setting range] 0 or 1
1, 0	D1, D0	Undefined	W	Endpoint Interface Number [Enable setting range] 0 to 3

				2: Bulk 3: Interrupt
3	D3	Undefined	W	Endpoint Transmission Direction [Possible setting range] 0: Out 1: In
2 to 0	D2 to D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

- EPIR02

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	D7 to D1	Undefined	W	Endpoint Maximum Packet Size [Possible setting range] 0 to 64
0	D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

- EPIR03

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

described below.

Since each endpoint FIFO number is optimized by the exclusive software that correspond transfer system, direction, and the maximum packet size, make sure to set the endpoint FI number to the data described in table 16.2.

1. The endpoint FIFO number 1 cannot designate other than the maximum packed size o control transfer method, and out transfer direction.
2. The endpoint number 0 and the endpoint FIFO number must have one-on one relation
3. The maximum packet size for the endpoint FIFO number 0 is 8 bytes only.
4. The endpoint FIFO number 0 can specify only the maximum packet size and the data rest should be all 0.
5. The maximum packet size for the endpoint FIFO numbers 1 and 2 is limited to 64 byt
6. The maximum packet size for the endpoint FIFO numbers 3 is limited to 8 bytes.
7. The maximum number of endpoint information setting is ten.
8. Up to ten endpoint information setting should be made.
9. Write 0 to the endpoints not in use.

Table 16.2 shows the example of limitations for the maximum packet size, the transfer m and the transfer direction.

Table 16.2 Example of Limitations for Setting Values

Endpoint FIFO Number	Maximum Packet Size	Transfer Method	Transfer Dire
0	8 bytes	Control	—
1	64 bytes	Bulk	Out
2	64 bytes	Bulk	In
3	8 bytes	Interrupt	In

—	1	1	1	—	—	—	—

N	EPIR[N]0	EPIR[N]1	EPIR[N]2	EPIR[N]3	EPIR[
0	00	00	10	00	00
1	14	20	80	00	01
2	24	28	80	00	02
3	34	38	10	00	03
4	00	00	00	00	00
5	00	00	00	00	00
6	00	00	00	00	00
7	00	00	00	00	00
8	00	00	00	00	00
9	00	00	00	00	00

Configuration	Interface	Alternate Setting	Endpoint Number	Endpoint FIFO Number	A
—	—	—	0	0	C
1	0	0	1	1	B
			2	2	B
			3	3	In

Bit	Bit Name	Initial Value	R/W	Description
7	PTSTE	0	R/W	Pin Test Enable Enables the test control for the on-chip transc output pins (USD+ and USD-).
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
3	SUSPEND	0	R/W	On-Chip Transceiver Output Signal Setting SUSPEND: Sets the (SUSPEND) signal of th transceiver. txenl: Sets the output enable (txenl) sig on-chip transceiver. txse0: Sets the Signal-ended 0 (txse0) s the on-chip transceiver. txdata: Sets the (txdata) signal of the on- transceiver.
2	txenl	0	R/W	
1	txse0	0	R/W	
0	txdata	0	R/W	

1	1	1	X	X	Hi-Z	Hi-Z

[Legend]

X: Don't care.

—: Cannot be controlled. Indicates state in normal operation according to the USB o and port settings.

R/W R R R R R R R

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
2	xver_data	—*	R	On-Chip Transceiver Input Signal Monitor xver_data: Monitors the differential input level (xver_data) signal of the on-chip transceiver. dpls: Monitors the USD+ (dpls) signal of t chip transceiver. dmns: Monitors the USD- (dmns) signal of chip transceiver.
1	dpls	—*	R	
0	dmns	—*	R	

Note: * Determined by the state of pins, VBUS, USD+, and USD-

1	0	1	1	0	1	1	0	
1	0	1	1	1	X	1	1	
1	1	1	0	0	0	0	0	
1	1	1	0	1	0	0	1	
1	1	1	1	0	0	1	0	
1	1	1	1	1	0	1	1	
1	X	0	X	X	0	1	1	Can be m when VBU

[Legend]

X: Don't care.

		transfer (EP0)		complete	USBINTN3		
	1		EP0i_TR*	EP0i transfer request	USBINTN2 or USBINTN3	×	×
	2		EP0o_TS*	EP0o receive complete	USBINTN2 or USBINTN3	×	×
	3		SETUP_TS*	Setup command receive complete	USBINTN2 or USBINTN3	×	×
	4	Bulk_in transfer (EP2)	EP2_EMPTY	EP2 FIFO empty	USBINTN2 or USBINTN3	×	US
	5		EP2_TR	EP2 transfer request	USBINTN2 or USBINTN3	×	×
	6	Bulk_out transfer (EP1)	EP1_FULL	EP1 FIFO Full	USBINTN2 or USBINTN3	×	US
	7	Status	BRST	Bus reset	USBINTN2 or USBINTN3	×	×
IFR1	0	Status	VBUSF	USB disconnection detection	USBINTN2 or USBINTN3	×	×
	1	Interrupt_in transfer (EP3)	EP3_TS	EP3 transfer complete	USBINTN2 or USBINTN3	×	×
	2		EP3_TR	EP3 transfer request	USBINTN2 or USBINTN3	×	×
	3	Status	VBUSMN	VBUS connection status	—	×	×
	4 5 6 7	—	Reserved	—	—	—	—

				USBINTN3, or RESUME		
5		SURSS	Suspend/resume status	—	×	×
6	—	Reserved	—	—	—	—
7						

Note: * EP0 interrupts must be assigned to the same interrupt request signal.

- USBINTN0 signal

  DMAC start interrupt signal only EP1. See section 16.8, DMA Transfer.
- USBINTN1 signal

  DMAC start interrupt signal only EP1. See section 16.8, DMA Transfer.
- USBINTN2 signal

  The USBINTN2 signal requests interrupt sources for which the corresponding bits in select registers 0 to 2 (ISR0 to ISR2) are cleared to 0. The USBINTN2 is driven low corresponding bit in the interrupt flag register is set to 1.
- USBINTN3 signal

  The USBINTN3 signal requests interrupt sources for which the corresponding bits in select registers 0 to 2 (ISR0 to ISR2) are cleared to 0. The USBINTN3 is driven low corresponding bit in the interrupt flag register is set to 1.
- RESUME signal

  The RESUME signal is a resume interrupt signal for canceling software standby mode RESUME signal is driven low at the transition to the resume state for canceling software standby mode.

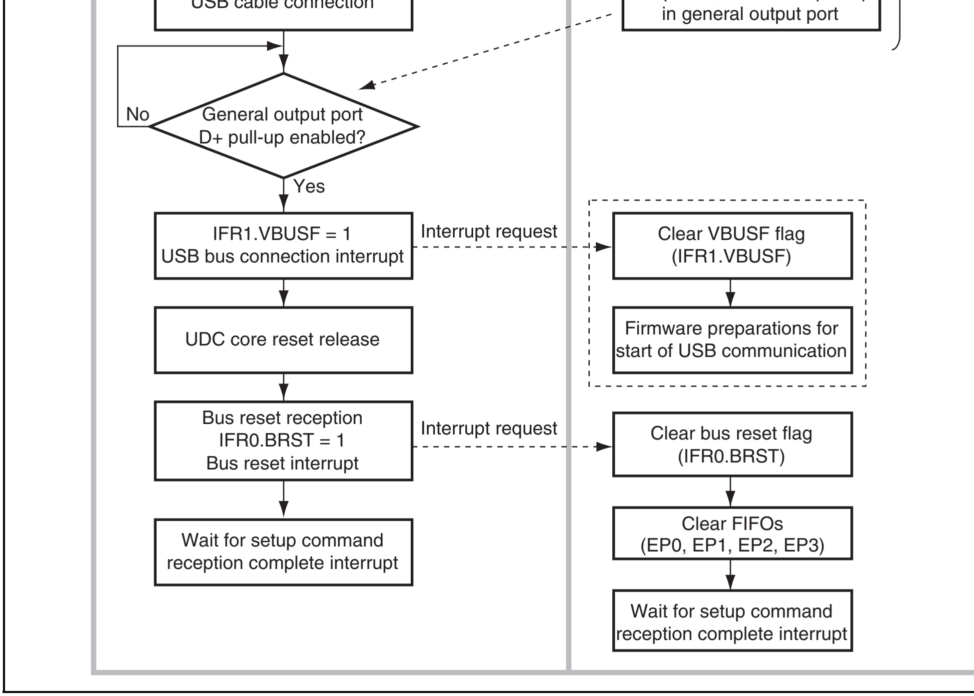


Figure 16.2 Cable Connection Operation

The above flowchart shows the operation in the case of in section 16.9, Example of USB
Circuitry.

In applications that do not require USB cable connection to be detected, processing by th
bus connection interrupt is not necessary. Preparations should be made with the bus-reset
interrupt.

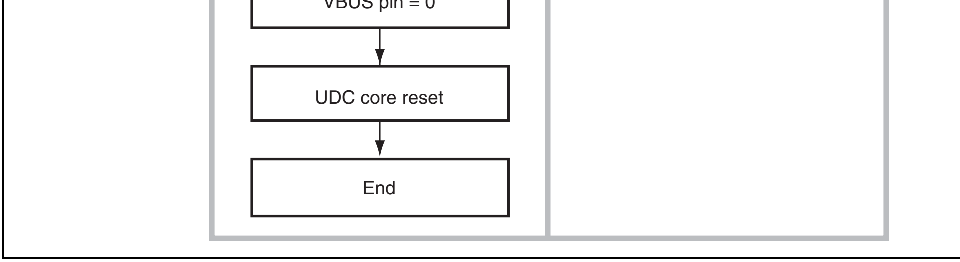


Figure 16.3 Cable Disconnection Operation

The above flowchart shows the operation in section 16.9, Example of USB External Cir

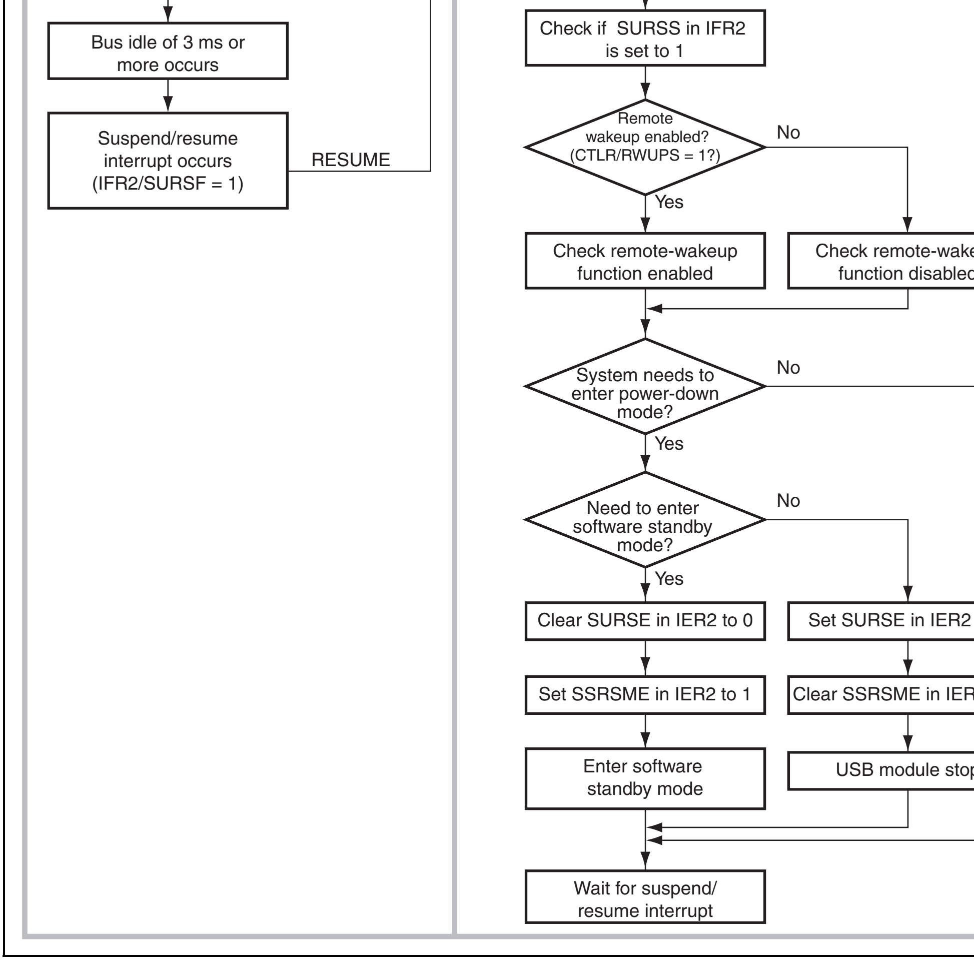


Figure 16.4 Suspend Operation

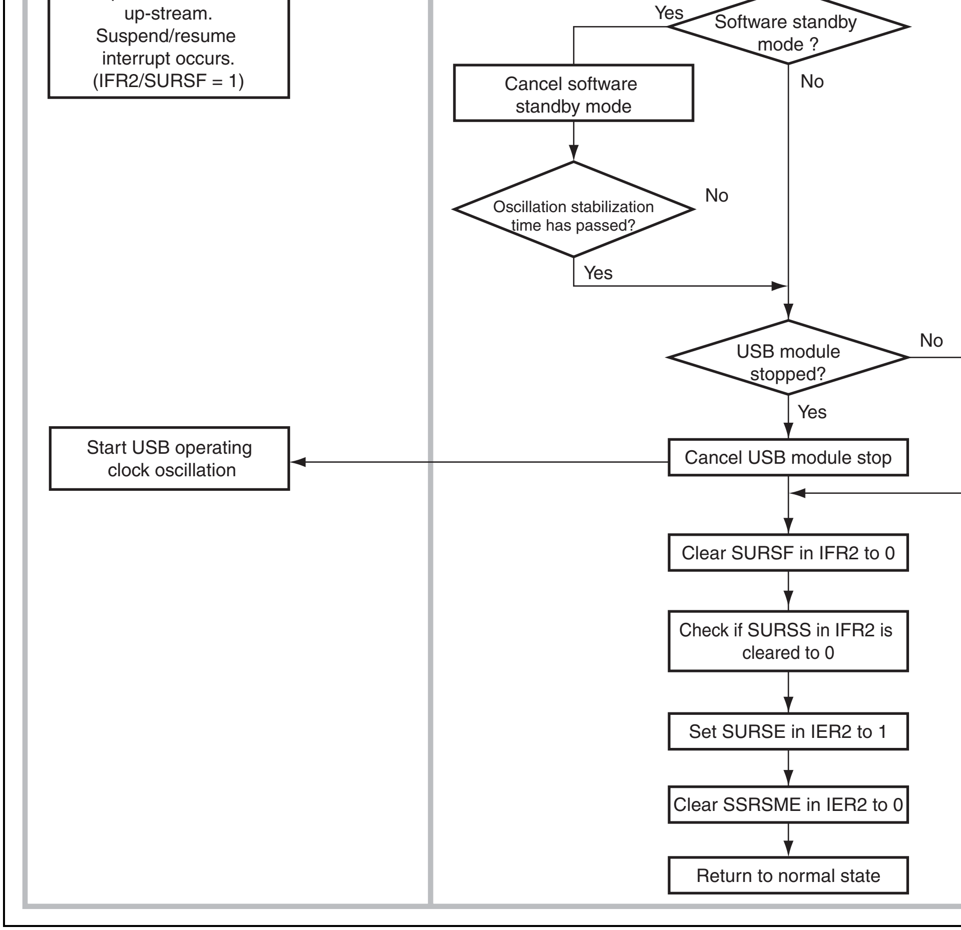


Figure 16.5 Resume Operation from Up-Stream

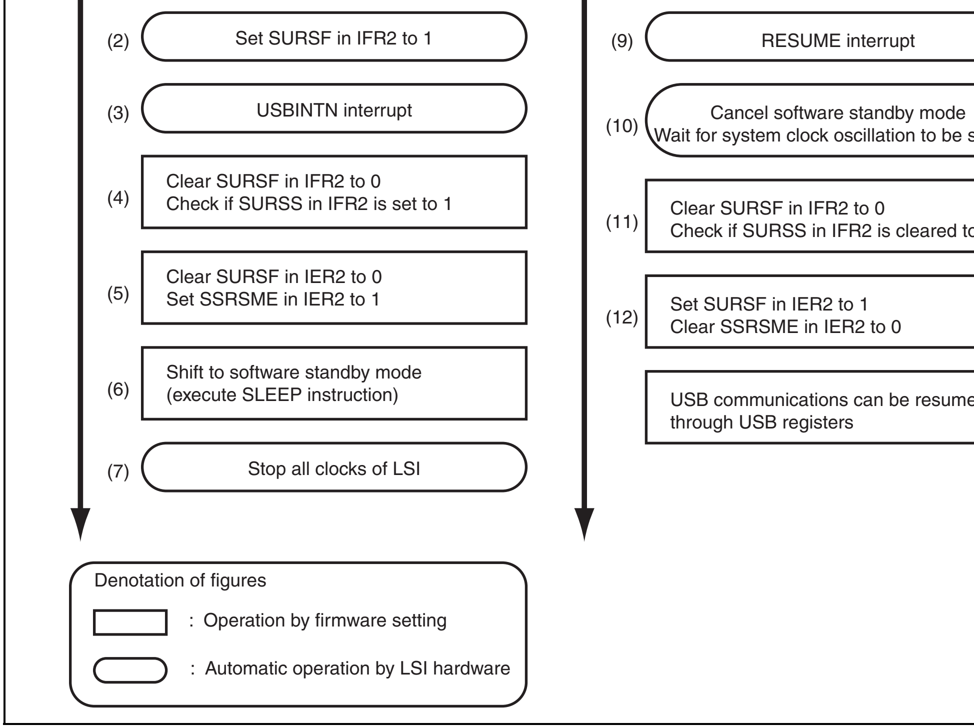


Figure 16.6 Flow of Transition to and Canceling Software Standby Mode

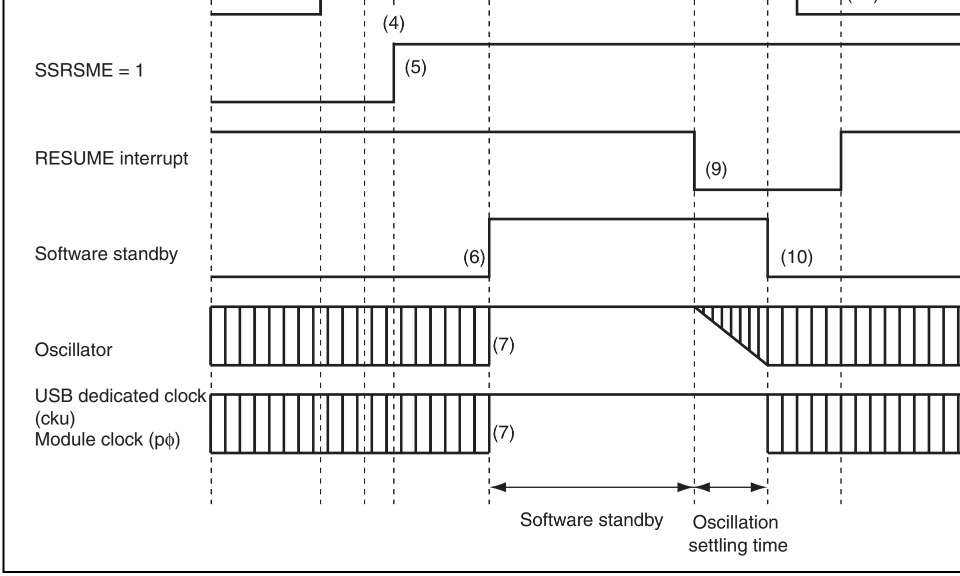


Figure 16.7 Timing of Transition to and Canceling Software Standby Mo

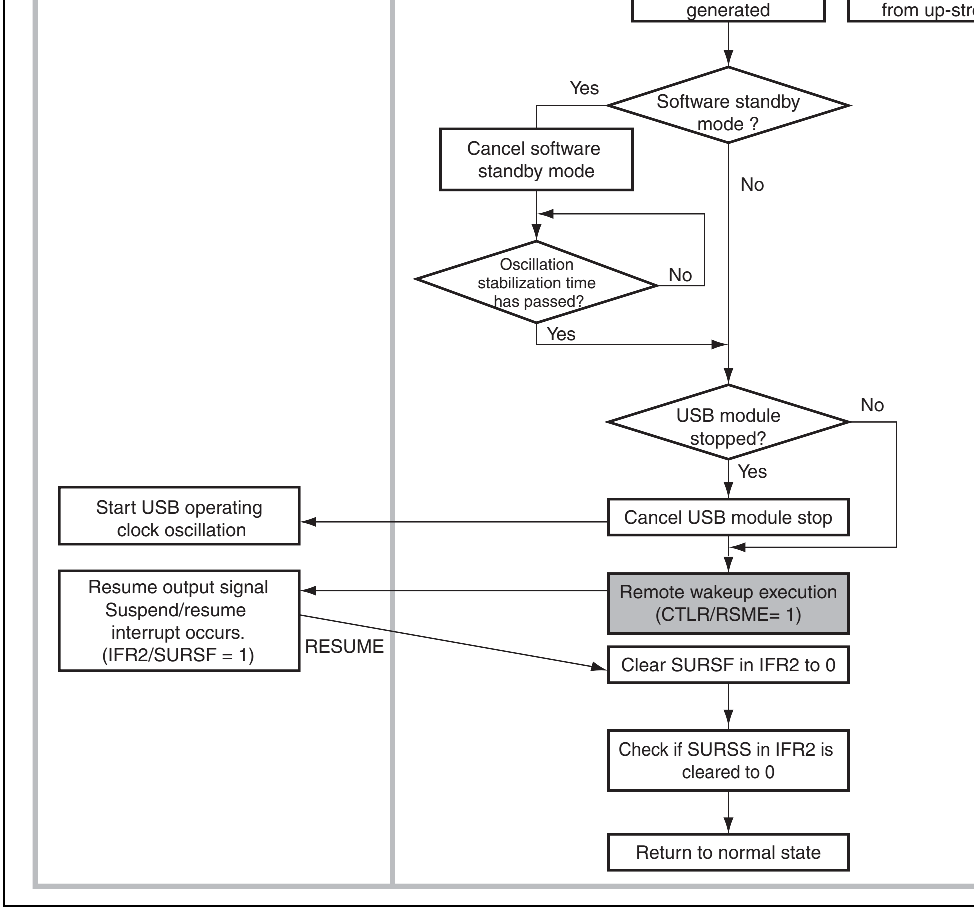


Figure 16.8 Remote-Wakeup

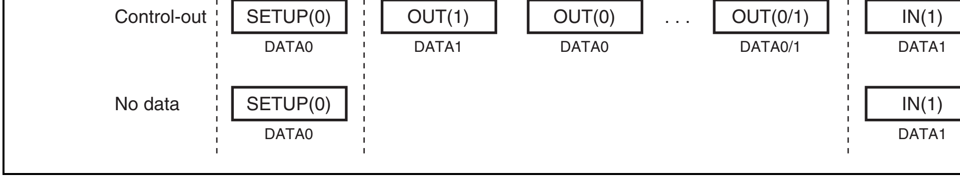
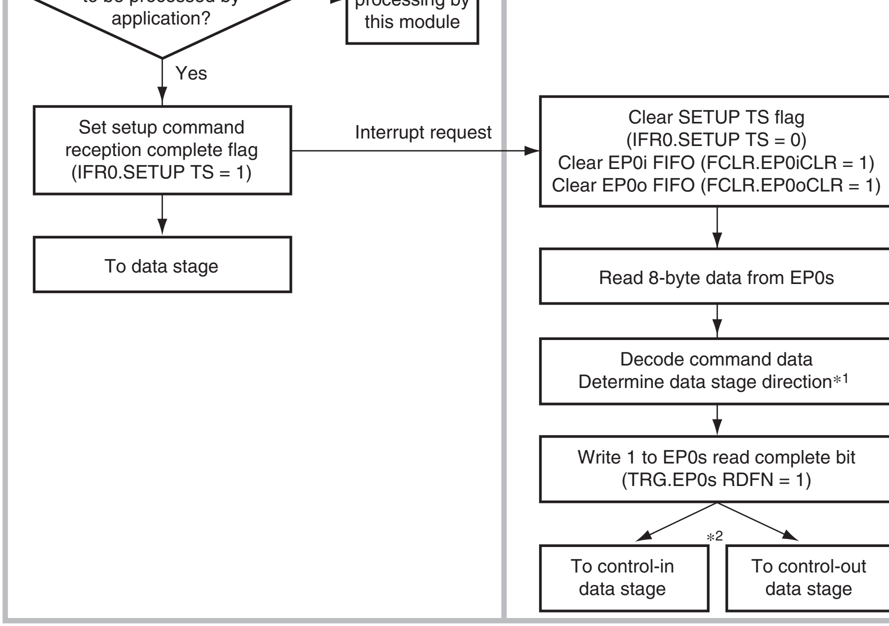


Figure 16.9 Transfer Stages in Control Transfer



Notes: 1. In the setup stage, the application analyzes command data from the host requiring processing by the application, and determines the subsequent processing (for example, data stage direction, et

2. When the transfer direction is control-out, the EP0i transfer request interrupt required in the statu stage should be enabled here. When the transfer direction is control-in, this interrupt is not requi and should be disabled.

Figure 16.10 Setup Stage Operation

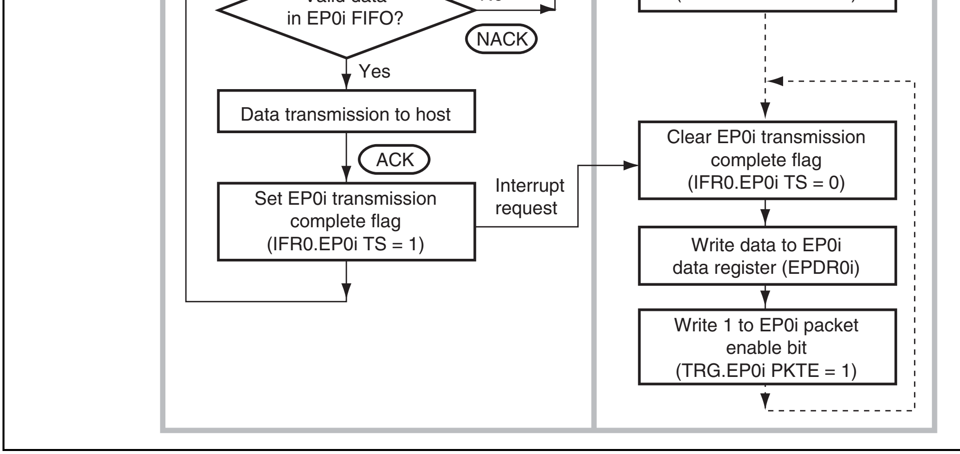


Figure 16.11 Data Stage (Control-In) Operation

The application first analyzes command data from the host in the setup stage, and determ subsequent data stage direction. If the result of command data analysis is that the data st transfer, one packet of data to be sent to the host is written to the FIFO. If there is more sent, this data is written to the FIFO after the data written first has been sent to the host bit in IFR0 = 1).

The end of the data stage is identified when the host transmits an OUT token and the sta is entered.

Note: If the size of the data transmitted by the function is smaller than the data size re the host, the function indicates the end of the data stage by returning to the host shorter than the maximum packet size. If the size of the data transmitted by the f an integral multiple of the maximum packet size, the function indicates the end stage by transmitting a zero-length packet.

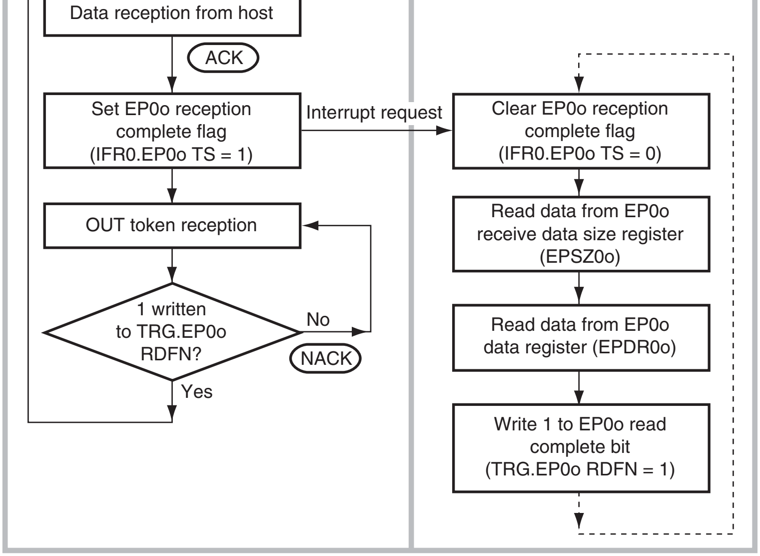


Figure 16.12 Data Stage (Control-Out) Operation

The application first analyzes command data from the host in the setup stage, and determi subsequent data stage direction. If the result of command data analysis is that the data sta transfer, the application waits for data from the host, and after data is received (EP0oTS b IFR0 = 1), reads data from the FIFO. Next, the application writes 1 to the EP0o read com empties the receive FIFO, and waits for reception of the next data.

The end of the data stage is identified when the host transmits an IN token and the status entered.

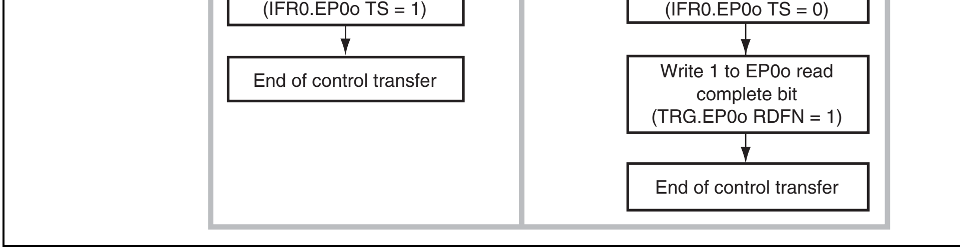


Figure 16.13 Status Stage (Control-In) Operation

The control-in status stage starts with an OUT token from the host. The application rece
byte data from the host, and ends control transfer.

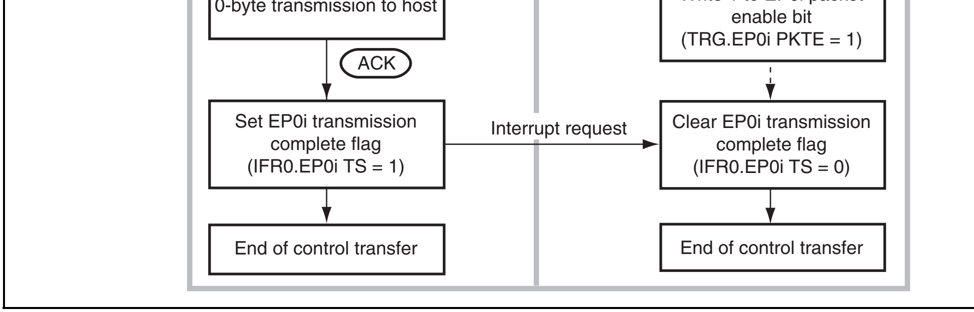


Figure 16.14 Status Stage (Control-Out) Operation

The control-out status stage starts with an IN token from the host. When an IN-token is re the start of the status stage, there is not yet any data in the EP0i FIFO, and so an EP0i tra request interrupt is generated. The application recognizes from this interrupt that the statu has started. Next, in order to transmit 0-byte data to the host, 1 is written to the EP0i pack bit but no data is written to the EP0i FIFO. As a result, the next IN token causes 0-byte d transmitted to the host, and control transfer ends.

After the application has finished all processing relating to the data stage, 1 should be wri the EP0i packet enable bit.

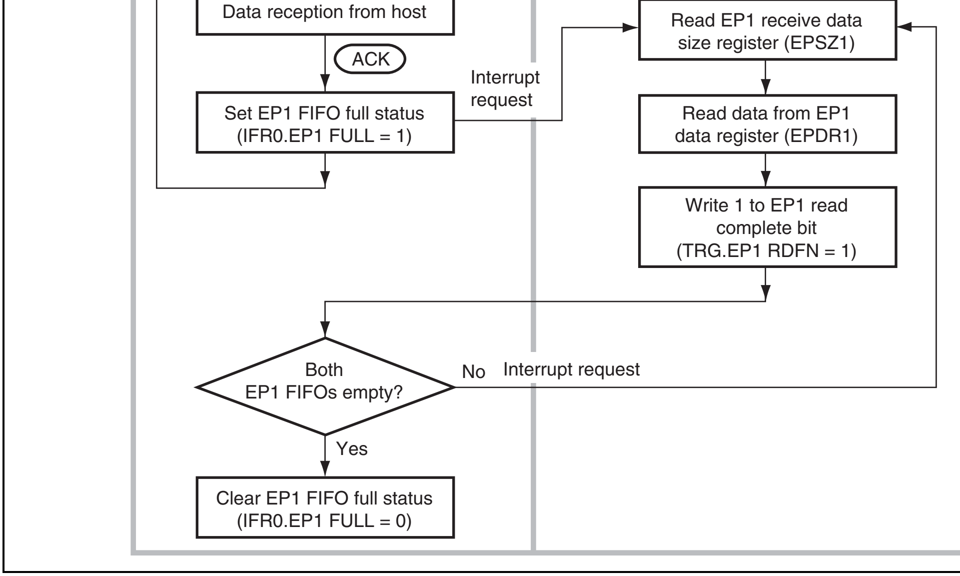


Figure 16.15 EP1 Bulk-Out Transfer Operation

EP1 has two 64-byte FIFOs, but the user can receive data and read receive data without
aware of this dual-FIFO configuration.

When one FIFO is full after reception is completed, the EP1FULL bit in IFR0 is set. Aft
receive operation into one of the FIFOs when both FIFOs are empty, the other FIFO is e
so the next packet can be received immediately. When both FIFOs are full, NACK is ret
the host automatically. When reading of the receive data is completed following data rec
is written to the EP1RDFN bit in TRG. This operation empties the FIFO that has just be
and makes it ready to receive the next packet.

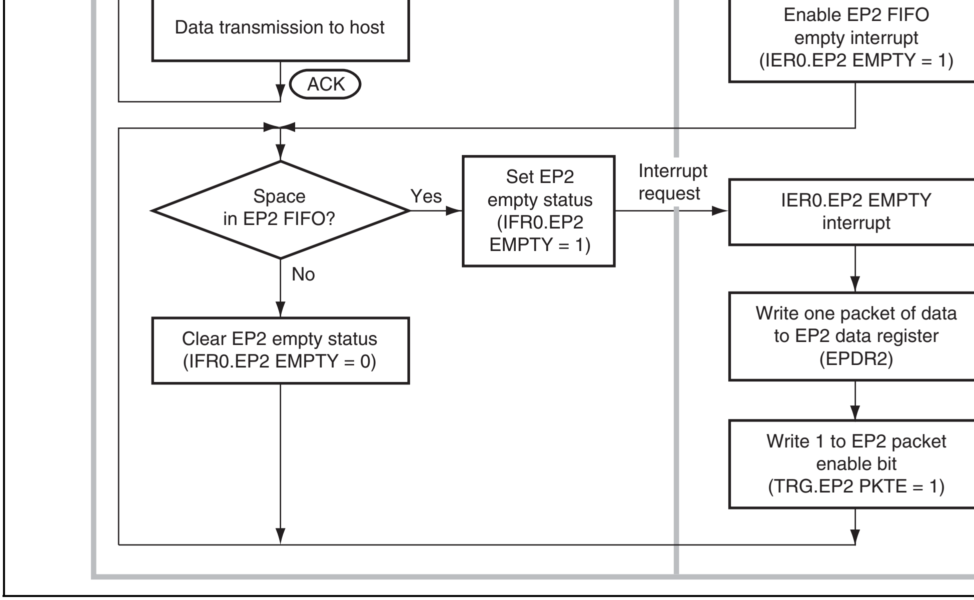
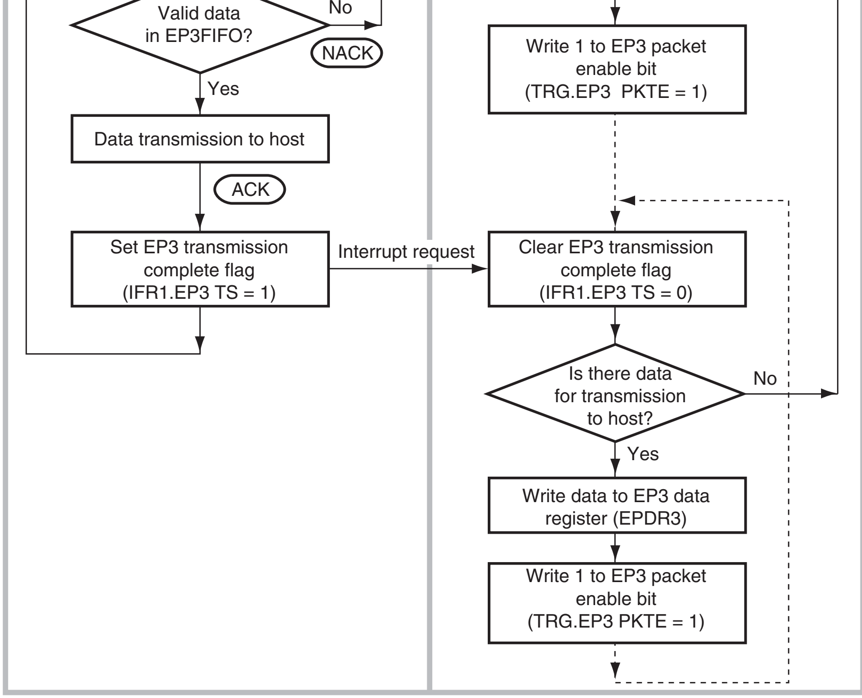


Figure 16.16 EP2 Bulk-In Transfer Operation

EP2 has two 64-byte FIFOs, but the user can transmit data and write transmit data withou aware of this dual-FIFO configuration. However, one data write is performed for one FIF example, even if both FIFOs are empty, it is not possible to perform EP2PKTE at one tim consecutively writing 128 bytes of data. EP2PKTE must be performed for each 64-byte w

When performing bulk-in transfer, as there is no valid data in the FIFOs on reception of t IN token, an EP2TR bit interrupt in IFR0 is requested. With this interrupt, 1 is written to EP2EMPTY bit in IER0, and the EP2 FIFO empty interrupt is enabled. At first, both EP2 are empty, and so an EP2 FIFO empty interrupt is generated immediately.



Note: This flowchart shows just one example of interrupt transfer processing. Other possibilities include an operation flow in which, if there is data to be transferred, the EP3 DE bit in the data status register is referenced to confirm that the FIFO is empty, and then data is written to the FIFO.

Figure 16.17 Operation of EP3 Interrupt-In Transfer

Decoding not Necessary on Application Side	Decoding Necessary on Applicati
Clear Feature	Get Descriptor
Get Configuration	Class/Vendor command
Get Interface	Set Descriptor
Get Status	Sync Frame
Set Address	
Set Configuration	
Set Feature	
Set Interface	

If decoding is not necessary on the application side, command decoding and data stage a
stage processing are performed automatically. No processing is necessary by the user. A
is not generated in this case.

If decoding is necessary on the application side, this module stores the command in the
FIFO. After reception is completed successfully, the IFR0/SETUP TS flag is set and an
request is generated. In the interrupt routine, eight bytes of data must be read from the E
register (EPDR0s) and decoded by firmware. The necessary data stage and status stage p
should then be carried out according to the result of the decoding operation.

The USB function module has internal status bits that hold the status (stall or non-stall) o
endpoint. When a transaction is sent from the host, the module references these internal s
and determines whether to return a stall to the host. These bits cannot be cleared by the
application; they must be cleared with a Clear Feature command from the host.

However, the internal status bit for EP0 is automatically cleared only when the setup com
received.

16.7.2 Forcible Stall by Application

The application uses the EPSTL register to issue a stall request for the USB function mod
When the application wishes to stall a specific endpoint, it sets the corresponding bit in E
1 in figure 16.18). The internal status bits are not changed at this time. When a transactio
from the host for the endpoint for which the EPSTL bit was set, the USB function module
references the internal status bit, and if this is not set, references the corresponding bit in
(1-2 in figure 16.18). If the corresponding bit in EPSTL is set, the USB function module
internal status bit and returns a stall handshake to the host (1-3 in figure 16.18). If the
corresponding bit in EPSTL is not set, the internal status bit is not changed and the transa
accepted.

Once an internal status bit is set, it remains set until cleared by a Clear Feature command
host, without regard to the EPSTL register. Even after a bit is cleared by the Clear Featur
command (3-1 in figure 16.18), the USB function module continues to return a stall hand
while the bit in EPSTL is set, since the internal status bit is set each time a transaction is
for the corresponding endpoint (1-2 in figure 16.18). To clear a stall, therefore, it is neces
the corresponding bit in EPSTL to be cleared by the application, and also for the internal
to be cleared with a Clear Feature command (2-1, 2-2, and 2-3 in figure 16.18).

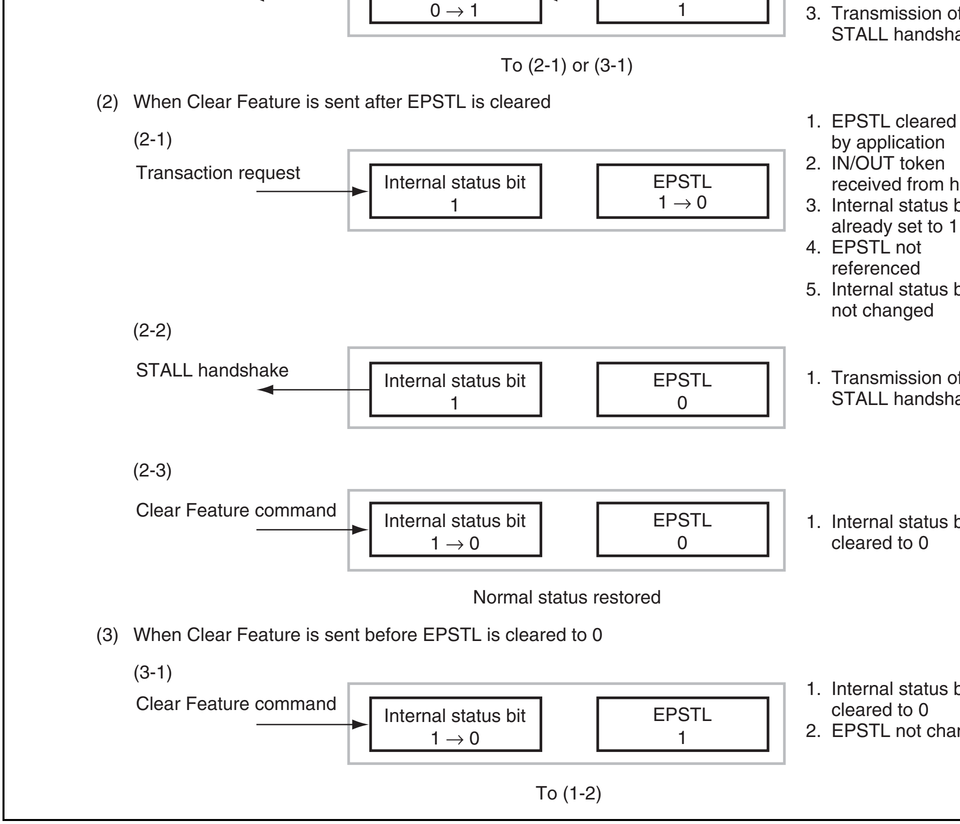


Figure 16.18 Forcible Stall by Application

the internal status bit must be cleared with a Clear Feature command (3-1 in figure 16.19) by the application, EPSTL should also be cleared (2-1 in figure 16.19).

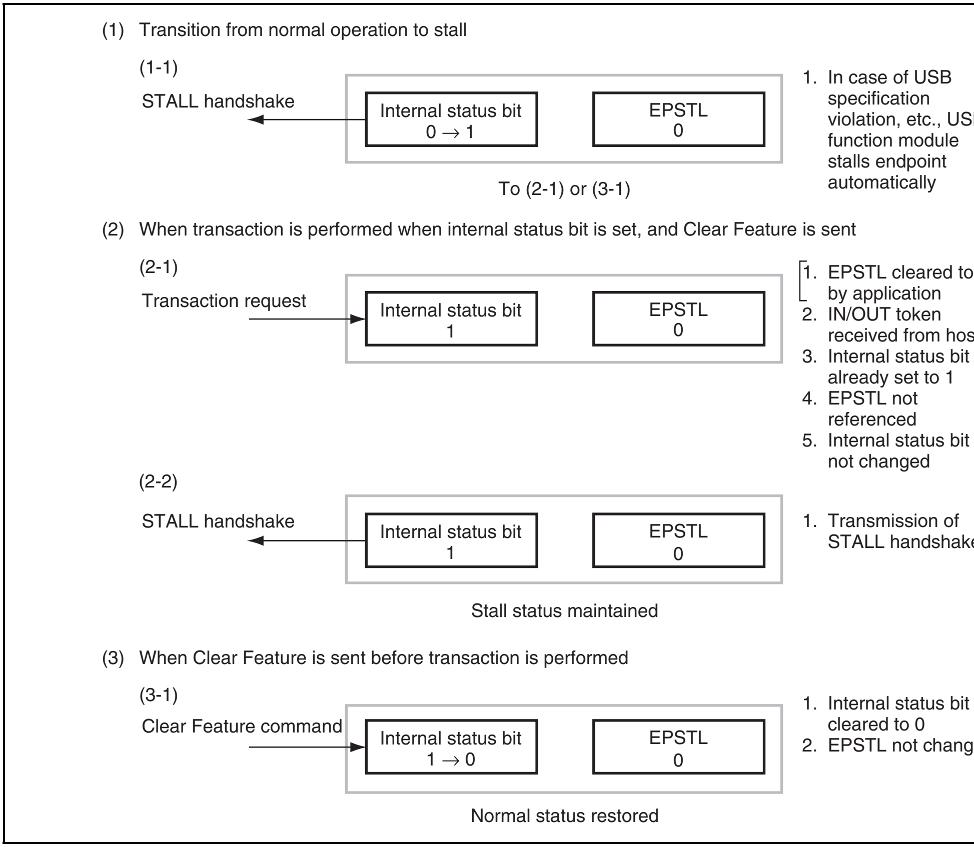


Figure 16.19 Automatic Stall by USB Function Module

If the DMA transfer is enabled by setting the EP1DMAE bit in the DMA transfer setting
to 1, zero-length data reception at endpoint 1 is ignored. When the DMA transfer is enab
RDFN bit for EP1 and PKTE bit for EP2 do not need to be set to 1 in TRG (note that the
must be set to 1 when the transfer data is less than the maximum number of bytes). Whe
data received at EP1 is read, the FIFO automatically enters the EMPTY state. When the
number of bytes (64 bytes) are written to the EP2 FIFO, the FIFO automatically enters t
state, and the data in the FIFO can be transmitted (see figures 16.20 and 16.21).

16.8.2 DMA Transfer for Endpoint 1

When the data received at EP1 is transferred by the DMAC, the USB function module automatically performs the same processing as writing 1 to the RDFN bit in TRG if the selected FIFO becomes empty. Accordingly, in DMA transfer, do not write 1 to the RDI
TRG. If the user writes 1 to the RDFN bit in DMA transfer, correct operation cannot be guaranteed.

Figure 16.20 shows an example of receiving 150 bytes of data from the host. In this case
processing which is the same as writing 1 to the RDFN bit in TRG is automatically perf
three times. This internal processing is performed when the currently selected data FIFO
empty. Accordingly, this processing is automatically performed both when 64-byte data
and when data less than 64 bytes is sent.

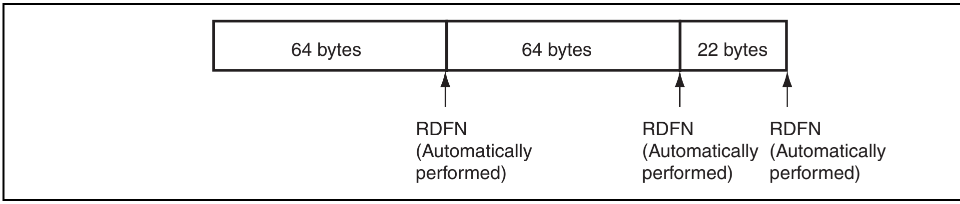


Figure 16.20 RDFN Bit Operation for EP1

processing which is the same as writing 1 to the PKTE bit in TRG is automatically perfor
twice. This internal processing is performed when the currently selected data FIFO becon
Accordingly, this processing is automatically performed only when 64-byte data is sent.

When the last 22 bytes are sent, the internal processing for writing 1 to the PKTE bit is no performed, and the user must write 1 to the PKTE bit by software. In this case, the applic no more data to transfer but the USB function module continues to output DMA requests as long as the FIFO has an empty space. When all data has been transferred, write 0 to the EP2DMAE bit in DMAR to cancel DMA requests for EP2.

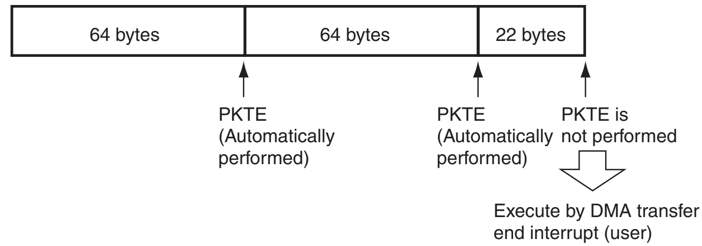
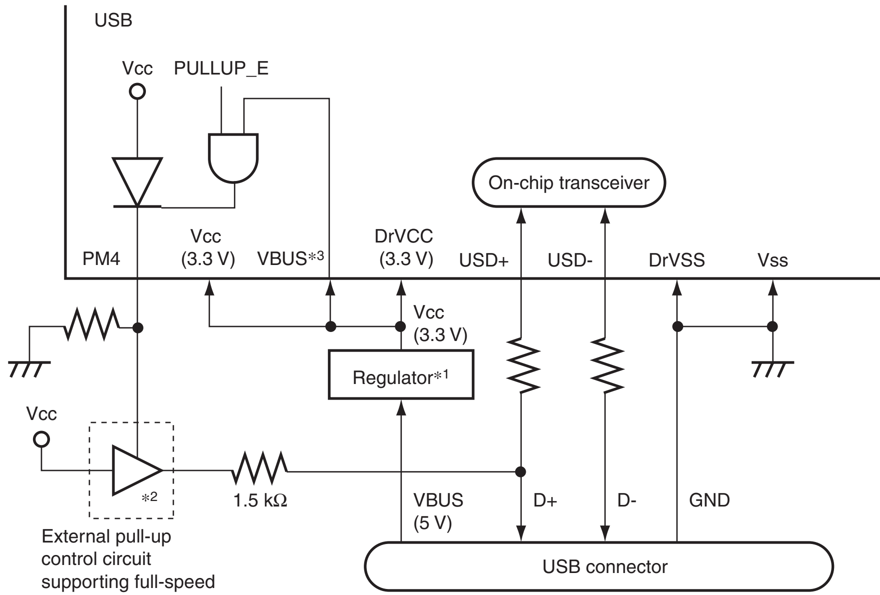


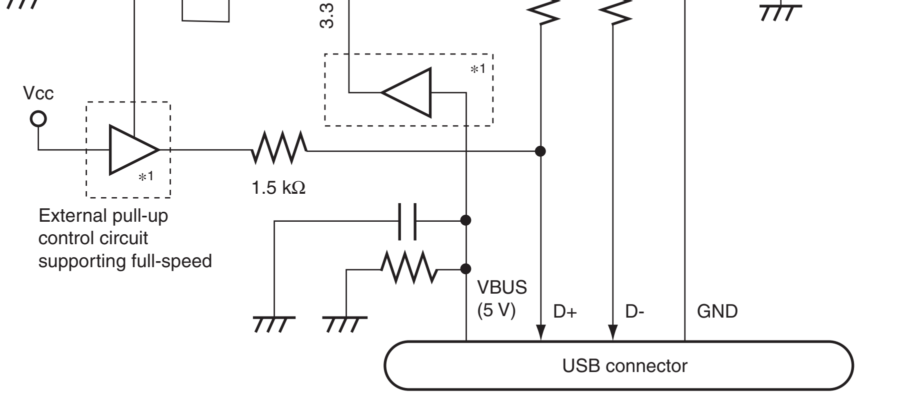
Figure 16.21 PKTE Bit Operation for EP2

As USB states, etc., are managed by hardware in this module, a VBUS signal that re
connection/disconnection is necessary. The power supply signal (VBUS) in the USB
used for this purpose. However, if the cable is connected to the USB host/hub when
function (system installing this LSI) power is off, a voltage (5 V) will be applied fro
host/hub. Therefore, an IC (such as an HD74LV1G08A or 2G08A) that allows volta
application when the system power is off should be connected externally.



Notes: 1. Reduce voltage to the operating voltage of this LSI (3.3 V).
2. To protect this LSI from being damaged, use the IC (such as HD74LV-A Series) which c
be applied voltage even when the system power is turned off.
3. Prevent noise from the VBUS pin while the USB is performing communication.

Figure 16.22 Example of Circuitry in Bus Power Mode



Notes: 1. To protect this LSI from being damaged, use the IC (such as HD74LV-A Series) which can be applied voltage even when the system power is turned off.
2. Prevent noise from the VBUS pin while the USB is performing communication.

Figure 16.23 Example of Circuitry in Self Power Mode

2. EPDR0s must always be read in 8-byte units. If the read is terminated at a midpoint, received at the next setup cannot be read correctly.

16.10.2 Clearing the FIFO

If a USB cable is disconnected during data transfer, the data being received or transmitt remain in the FIFO. When disconnecting a USB cable, clear the FIFO.

While a FIFO is transferring data, it must not be cleared.

16.10.3 Overreading and Overwriting the Data Registers

Note the following when reading or writing to a data register of this module.

(1) Receive data registers

The receive data registers must not be read exceeding the valid amount of receive data, t number of bytes indicated by the receive data size register. Even for EPDR1 which has FIFO buffers, the maximum data to be read at one time is 64 bytes. After the data is rea current valid FIFO buffer, be sure to write 1 to EP1RDFN in TRG, which switches the buffer, updates the receive data size to the new number of bytes, and enables the next da received.

(2) Transmit data registers

The transmit data registers must not be written to exceeding the maximum packet size. EPDR2 which has double FIFO buffers, write data within the maximum packet size at o After the data is written, write 1 to PKTE in TRG to switch the valid buffer and enable t data to be written. Data must not be continuously written to the two FIFO buffers.

16.10.6 Notes on TR Interrupt

Note the following when using the transfer request interrupt (TR interrupt) for IN transfer EP2, or EP3.

The TR interrupt flag is set if the FIFO for the target EP has no data when the IN token is from the USB host. However, at the timing shown in figure 16.24, multiple TR interrupts successively. Take appropriate measures against malfunction in such a case.

Note: This module determines whether to return NAKC if the FIFO of the target EP has when receiving the IN token, but the TR interrupt flag is set after a NAKC handsl sent. If the next IN token is sent before PKTE of TRG is written to, the TR interru set again.

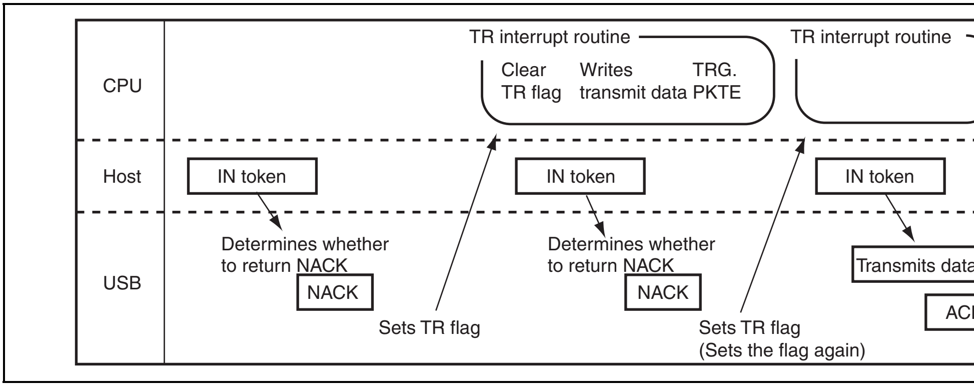


Figure 16.24 TR Interrupt Flag Set Timing

1	16 MHz	EXTAL × 3	EXTAL × 1 (16 M
			EXTAL × 2 (32 M

17.1 Features

- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independe
  each other, the continuous transmission/reception can be performed.
- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronize
  automatically. If transmission or reception is not yet possible, drive the SCL signal l
  preparations are completed
- Six interrupt sources

  Transmit-data-empty (including slave-address match), transmit-end, receive-data-ful
  (including slave-address match), arbitration lost, NACK detection, and stop conditio
  detection
- Direct bus drive

  Two pins, the SCL and SDA pins function as NMOS open-drain outputs.

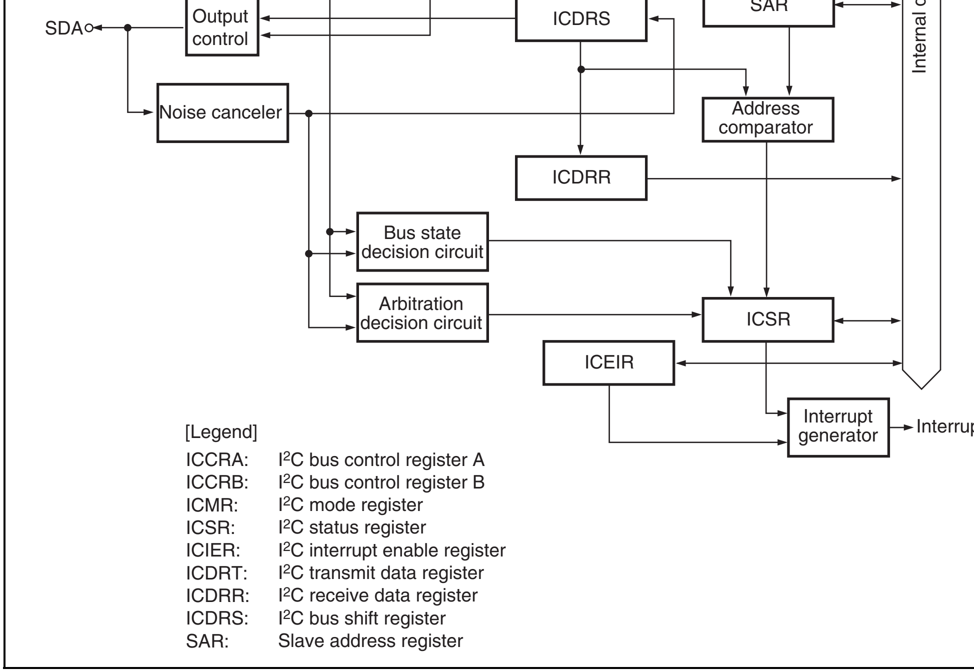


Figure 17.1 Block Diagram of I²C Bus Interface 2

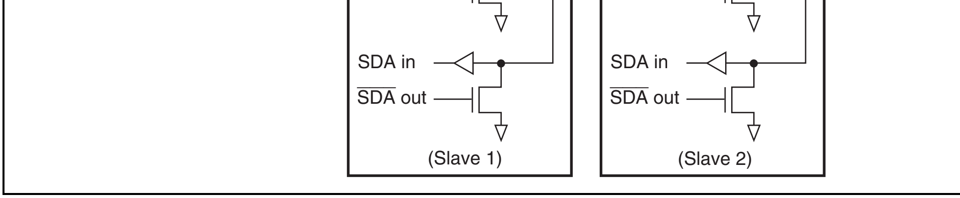


Figure 17.2 Connections to the External Circuit by the I/O Pins

17.2 Input/Output Pins

Table 17.1 shows the pin configuration of the I^{2}C bus interface 2.

Table 17.1 Pin Configuration of the I^{2}C Bus Interface 2

Channel	Abbreviation	I/O	Function
0	SCL0	I/O	Channel 0 serial clock I/O pin
	SDA0	I/O	Channel 0 serial data I/O pin
1	SCL1	I/O	Channel 1 serial clock I/O pin
	SDA1	I/O	Channel 1 serial data I/O pin

Note: The pin symbols are represented as SCL and SDA; channel numbers are omitted manual.

- I^2C bus status register_0 (ICSR_0)
- Slave address register_0 (SAR_0)
- I^2C bus transmit data register_0 (ICDRT_0)
- I^2C bus receive data register_0 (ICDRR_0)
- I^2C bus shift register_0 (ICDRS_0)

Channel 1:

- I^2C bus control register A_1 (ICCRA_1)
- I^2C bus control register B_1 (ICCRB_1)
- I^2C bus mode register_1 (ICMR_1)
- I^2C bus interrupt enable register_1 (ICIER_1)
- I^2C bus status register_1 (ICSR_1)
- Slave address register_1 (SAR_1)
- I^2C bus transmit data register_1 (ICDRT_1)
- I^2C bus receive data register_1 (ICDRR_1)
- I^2C bus shift register_1 (ICDRS_1)

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface Enable 0: This module is halted 1: This bit is enabled for transfer operations SDA pins are bus drive state)
6	RCVD	0	R/W	Reception Disable This bit enables or disables the next operatio TRS is 0 and ICDRR is read. 0: Enables next reception 1: Disables next reception
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select When arbitration is lost in master mode, MS TRS are both reset by hardware, causing a t to slave receive mode. Modification of the TI should be made between transfer frames. Operating modes are described below accor MST and TRS combination. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3	CKS3	0	R/W	Transfer Clock Select 3 to 0
2	CKS2	0	R/W	These bits are valid only in master mode. Ma setting according to the required transfer rat details on the transfer rate, see table 17.2.
1	CKS1	0	R/W	
0	CKS0	0	R/W	

			1	Pϕ/100	80.0 kHz	100 kHz	200 kHz	250 kHz	3
		1	0	Pϕ/112	71.4 kHz	89.3 kHz	179 kHz	223 kHz	2
			1	Pϕ/128	62.5 kHz	78.1 kHz	156 kHz	195 kHz	2
1	0	0	0	Pϕ/56	143 kHz	179 kHz	357 kHz	446 kHz	5
			1	Pϕ/80	100 kHz	125 kHz	250 kHz	313 kHz	4
		1	0	Pϕ/96	83.3 kHz	104 kHz	208 kHz	260 kHz	3
			1	Pϕ/128	62.5 kHz	78.1 kHz	156 kHz	195 kHz	2
	1	0	0	Pϕ/336	23.8 kHz	29.8 kHz	59.5 kHz	74.4 kHz	9
			1	Pϕ/200	40.0 kHz	50.0 kHz	100 kHz	125 kHz	1
		1	0	Pϕ/224	35.7 kHz	44.6 kHz	89.3 kHz	112 kHz	1
			1	Pϕ/256	31.3 kHz	39.1 kHz	78.1 kHz	97.7 kHz	1

17.3.2 I²C Bus Control Register B (ICCRB)

ICCRB issues start/stop condition, manipulates the SDA pin, monitors the SCL pin, and c
reset in the I²C control module.

Bit	7	6	5	4	3	2	1
Bit Name	BBSY	SCP	SDAO	—	SCLO	—	IICRST
Initial Value	0	1	1	1	1	1	0
R/W	R/W	R/W	R	R/W	R	—	R/W

				a start condition. To issue a start or stop cond the MOV instruction.
6	SCP	1	R/W	Start/Stop Condition Issue This bit controls the issuance of start or stop c in master mode. To issue a start condition, write 1 to BBSY an SCP. A re-transmit start condition is issued in way. To issue a stop condition, write 0 to BBS to SCP. This bit is always read as 1. If 1 is wr data is not stored.
5	SDAO	1	R	This bit monitors the output level of SDA. 0: When reading, the SDA pin outputs a low le 1: When reading the SDA pin outputs a high l
4	—	1	R/W	Reserved The write value should always be 1.
3	SCLO	1	R	This bit monitors the SCL output level. When reading and SCLO is 1, the SCL pin ou high level. When reading and SCLO is 0, the outputs a low level.
2	—	1	—	Reserved This bit is always read as 0.
1	IICRST	0	R/W	IIC Control Module Reset This bit reset the IIC control module except th registers. If hang-up occurs because of comm failure during I^2C operation, by setting this bit
0	—	1	—	Reserved This bit is always read as 1.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The write value should always be 0.
6	WAIT	0	R/W	Wait Insertion This bit selects whether to insert a wait after da transfer except for the acknowledge bit. When set to 1, after the falling of the clock for the last the low period is extended for two transfer cloc When this bit is cleared to 0, data and the ackn bit are transferred consecutively with no waits i The setting of this bit is invalid in slave mode.
5	—	1	—	Reserved
4	—	1	—	These bits are always read as 1.
3	BCWP	1	R/W	BC Write Protect This bit controls the modification of the BC2 to bits. When modifying, this bit should be cleared and the MOV instruction should be used. 0: When writing, the values of BC2 to BC0 are 1: When reading, 1 is always read When writing, the settings of BC2 to BC0 are in

001: 2

010: 3

011: 4

100: 5

101: 6

110: 7

111: 8

I^2C control module can be reset without settin
ports and initializing the registers.

17.3.4 I^2C Bus Interrupt Enable Register (ICIER)

ICIER enables or disables interrupt sources and the acknowledge bits, sets the acknowle
be transferred, and confirms the acknowledge bit to be received.

Bit	7	6	5	4	3	2	1
Bit Name	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

				This bit enables or disables the transmit end in (TEI) request at the rising of the ninth clock wh TDRE bit in ICSR is set to 1. The TEI request c canceled by clearing the TEND bit or the TEIE 0: Transmit end interrupt (TEI) request is disab 1: Transmit end interrupt (TEI) request is enab
5	RIE	0	R/W	Receive Interrupt Enable This bit enables or disables the receive full inte (RXI) request when receive data is transferred ICDRS to ICDRR and the RDRF bit in ICSR is The RXI request can be canceled by clearing t RDRF or RIE bit to 0. 0: Receive data full interrupt (RXI) request is d 1: Receive data full interrupt (RXI) request is e
4	NAKIE	0	R/W	NACK Receive Interrupt Enable This bit enables or disables the NACK receive (NAKI) request when the NACKF and AL bits i are set to 1. The NAKI request can be cancele clearing the NACKF or AL bit, or the NAKIE bit 0: NACK receive interrupt (NAKI) request is dis 1: NACK receive interrupt (NAKI) request is en

				1: If the acknowledge bit is 1, continuous tran suspended
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowle that are returned by the receive device. This b be modified. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be the acknowledge timing. 0: 0 is sent at the acknowledge timing 1: 1 is sent at the acknowledge timing

Bit	Bit Name	Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Setting condition] • When data is transferred from ICDRT to IC and ICDRT becomes empty [Clearing conditions] • When 0 is written to this bit after reading TD (When the CPU is used to clear this flag by 0 while the corresponding interrupt is enabl sure to read the flag after writing 0 to it.) • When data is written to ICDRT
6	TEND	0	R/W	Transmit End [Setting condition] • When the ninth clock of SCL rises while the flag is 1 [Clearing conditions] • When 0 is written to this bit after reading TE (When the CPU is used to clear this flag by 0 while the corresponding interrupt is enabl sure to read the flag after writing 0 to it.) • When data is written to ICDRT

				• When data is read from ICDRR
4	NACKF	0	R/W	No Acknowledge Detection Flag [Setting condition] • When no acknowledge is detected from th device in transmission while the ACKE bit is set to 1 [Clearing condition] • When 0 is written to this bit after reading N 1 (When the CPU is used to clear this flag b 0 while the corresponding interrupt is enal sure to read the flag after writing 0 to it.)
3	STOP	0	R/W	Stop Condition Detection Flag [Setting condition] • When a stop condition is detected after fra transfer [Clearing condition] • When 0 is written to this bit after reading S (When the CPU is used to clear this flag b 0 while the corresponding interrupt is enal sure to read the flag after writing 0 to it.)

				disagree at the rising of SCL in master tran mode • When the SDA pin outputs a high level in m mode while a start condition is detected [Clearing condition] • When 0 is written to this bit after reading Al (When the CPU is used to clear this flag by 0 while the corresponding interrupt is enabl sure to read the flag after writing 0 to it.)
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 wher frame following a start condition matches bits S SVA0 in SAR. [Setting conditions] • When the slave address is detected in slav receive mode • When the general call address is detected receive mode [Clearing condition] • When 0 is written to this bit after reading A (When the CPU is used to clear this flag by 0 while the corresponding interrupt is enabl sure to read the flag after writing 0 to it.)

sure to read the flag after writing 0 to it.)

17.3.6 Slave Address Register (SAR)

SAR is sets the slave address. In slave mode, if the upper 7 bits of SAR match the upper the first frame received after a start condition, the LSI operates as the slave device.

Bit	7	6	5	4	3	2	1
Bit Name	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA6 to SVA0	0	R/W	Slave Address 6 to 0 These bits set a unique address differing from addresses of other slave devices connected t bus.
0	—	0	R/W	Reserved Although this bit is readable/writable, only 0 s written to.

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

17.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit read-only register that stores the receive data. When one byte of data h
received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data can
received. ICDRR is a receive-only register; therefore, this register cannot be written to by
CPU.

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

17.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is an 8-bit write-only register that is used to transmit/receive data. In transmission
transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, da
transferred from ICDRS to ICDRR after one by of data is received. This register cannot b
from the CPU.

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W

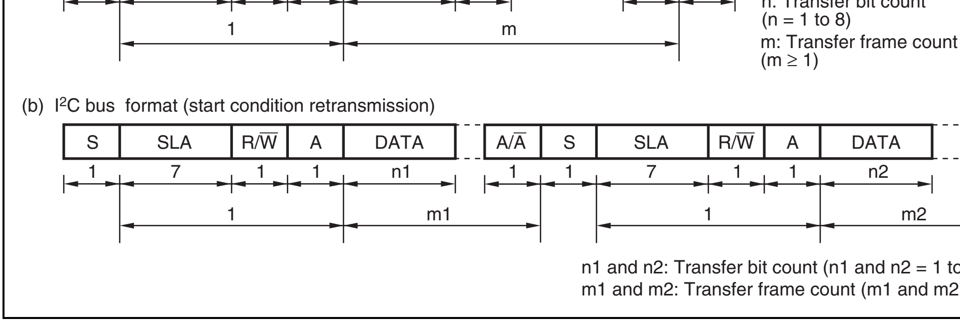


Figure 17.3 I²C Bus Formats

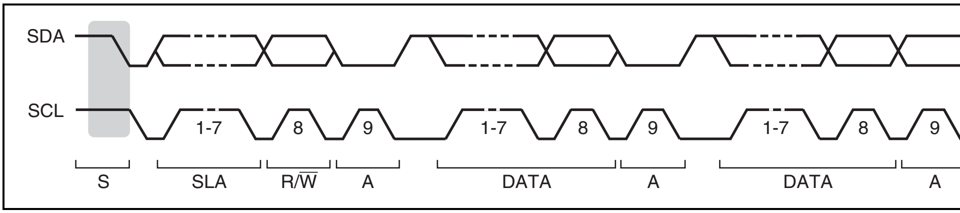


Figure 17.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is h

SLA: Slave address

R/$\overline{\text{W}}$: Indicates the direction of data transfer; from the slave device to the master devi R/$\overline{\text{W}}$ is 1, or from the master device to the slave device when R/$\overline{\text{W}}$ is 0.

A: Acknowledge. The receive device drives SDA low.

DATA: Transferred data

P: Stop condition. The master device drives SDA from low to high while SCL is h

ICCRA to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using th instruction. (The start condition is issued.) This generates the start condition.

3. After confirming that TDRE in ICSR has been set, write the transmit data (the first by the slave address and R/W) to ICDRT. After this, when TDRE is automatically cleare data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR is at the rising of the ninth transmit clock pulse. Read the ACKBR bit in ICIER to confir the slave device has been selected. Then, write the second byte data to ICDRT. When is 1, the slave device has not been acknowledged, so issue a stop condition. To issue t condition, write 0 to BBSY and SCP using the MOV instruction. SCL is fixed to a lov until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the en byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR is 1) fr receive device while CKE in ICIER is 1. Then, issue the stop condition to clear TENI NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode

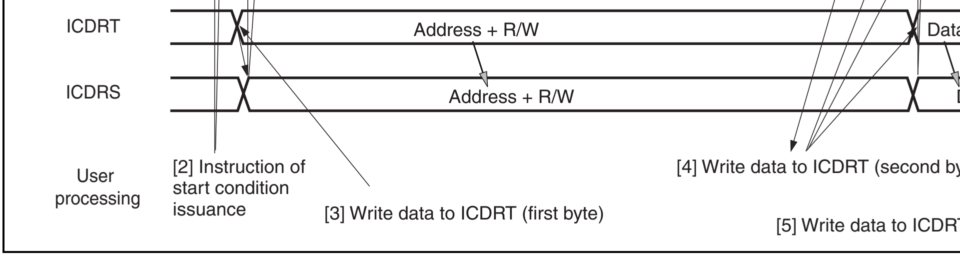


Figure 17.5 Master Transmit Mode Operation Timing 1

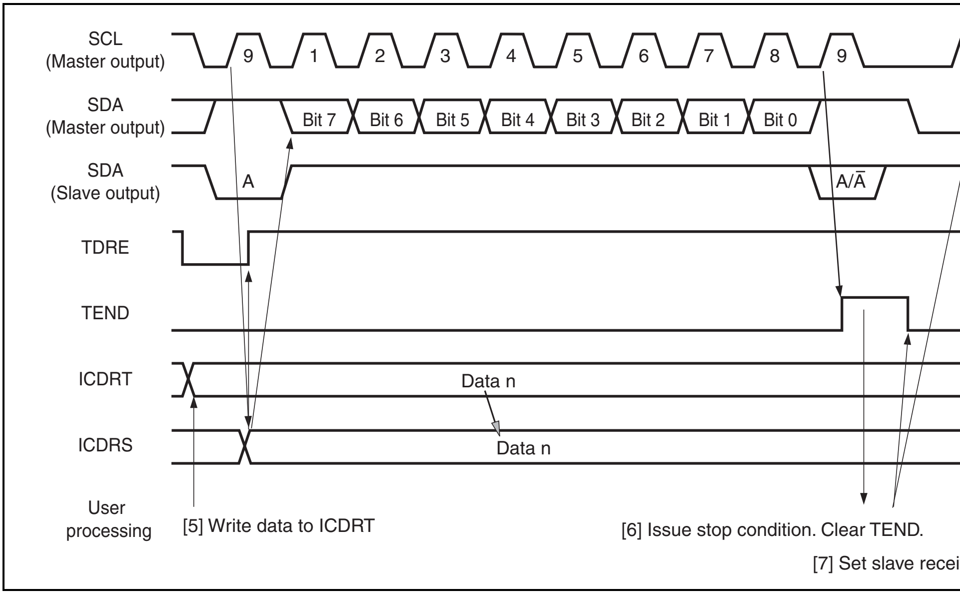


Figure 17.6 Master Transmit Mode Operation Timing 2

specified by the ACKBT in ICIER to SDA, at the ninth receive clock pulse.

3. After the reception of the first frame data is completed, the RDRF bit in ICSR is set to rising of the ninth receive clock pulse. At this time, the received data is read by readin ICDRR. At the same time, RDRF is cleared.
4. The continuous reception is performed by reading ICDRR and clearing RDRF to 0 ev RDRF is set. If the eighth receive clock pulse falls after reading ICDRR by other proc while RDRF is 1, SCL is fixed to a low level until ICDRR is read.
5. If the next frame is the last receive data, set the RCVD bit in ICCR1 before reading IC This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at the rising of the ninth receive clock pulse, the stop co is issued.
7. When the STOP bit in ICSR is set to 1, read ICDRR and clear RCVD to 0.
8. The operation returns to the slave receive mode.

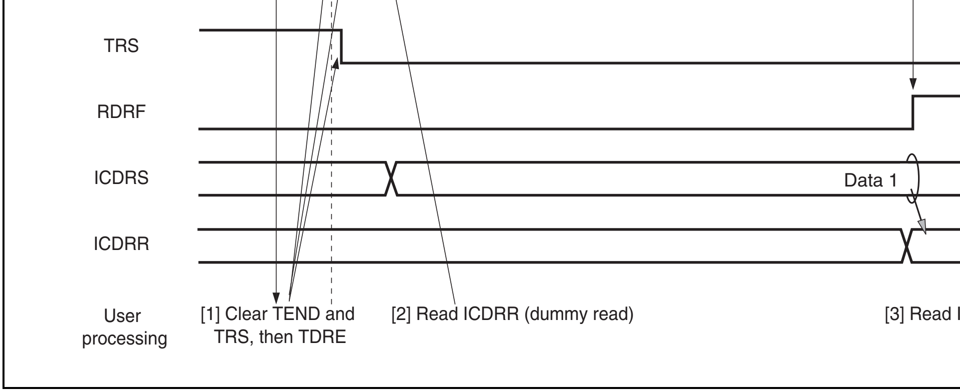


Figure 17.7 Master Receive Mode Operation Timing 1

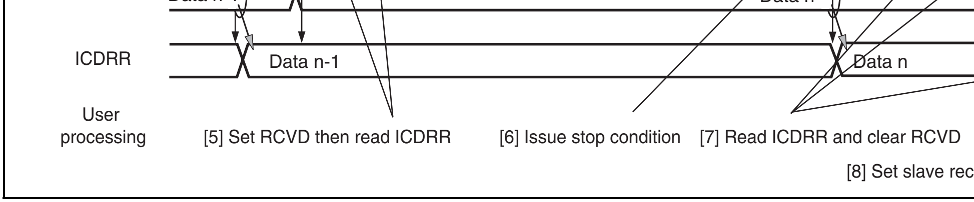


Figure 17.8 Master Receive Mode Operation Timing 2

17.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, and the master device o the receive clock pulse and returns an acknowledge signal. Figures 17.9 and 17.10 show t operation timings in slave transmit mode. The transmission procedure and operations in s transmit mode are described below.

1. Set the ICR bit in the corresponding register to 1, then set the ICE bit in ICCRA to 1. ACKBIT in ICIER, and perform other initial settings. Set the MST and TRS bits in IC select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following the detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, a rising of the ninth clock pulse. At this time, if the eighth bit data (R/$\overline{W}$) is 1, TRS in IC and TDRE in ICSR are set to 1, and the mode changes to slave transmit mode automa The continuous transmission is performed by writing the transmit data to ICDRT ever TDRE is set.
3. If TDRE is set after writing the last transmit data to ICDRT, wait until TEND in ICSR 1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for end processing, and read ICDRR (dummy read) to free SCL.
5. Clear TDRE.

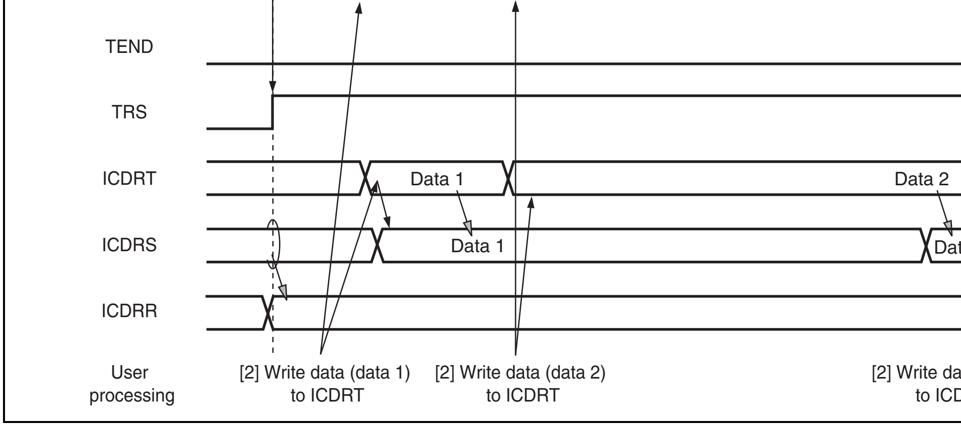


Figure 17.9 Slave Transmit Mode Operation Timing 1

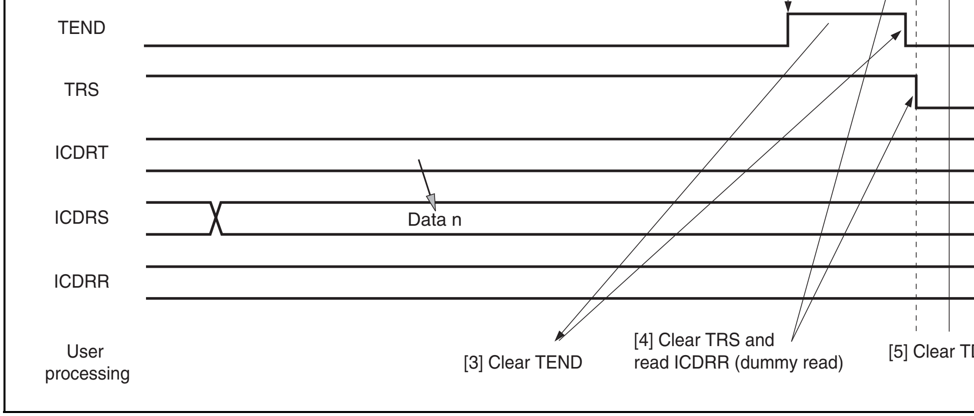


Figure 17.10 Slave Transmit Mode Operation Timing 2

2. When the slave address matches in the first frame following detection of the start co[...]
the slave address outputs the level specified by ACKBT in ICIER to SDA, at the risi[...]
ninth clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy re[...]
(Since the read data shows the slave address and R/$\overline{W}$, it is not used).
3. Read ICDRR every time RDRF is set. If the eighth clock pulse falls while RDRF is [...]
fixed to a low level until ICDRR is read. The change of the acknowledge (ACKBT) [...]
before reading ICDRR to be returned to the master device is reflected in the next tra[...]
frame.
4. The last byte data is read by reading ICDRR.

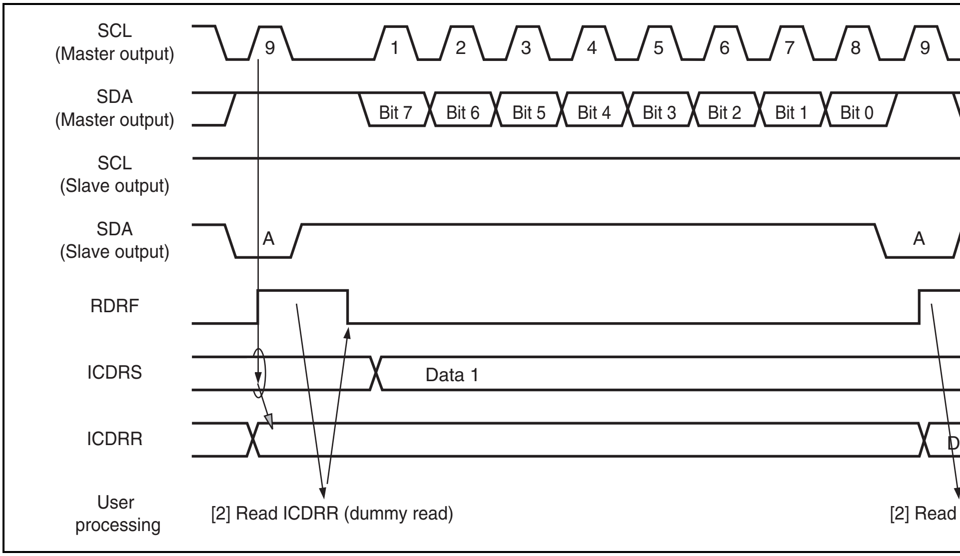


Figure 17.11 Slave Receive Mode Operation Timing 1

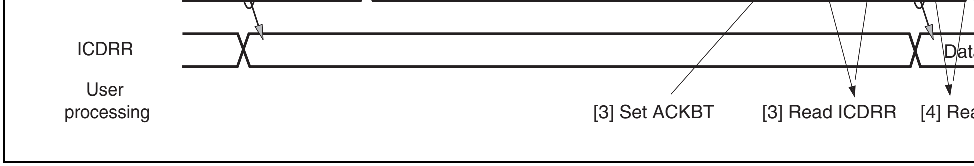


Figure 17.12 Slave Receive Mode Operation Timing 2

17.4.6 Noise Canceler

The logic levels at the SCL and SDA pins are routed through the noise cancelers before b latched internally. Figure 17.13 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The signal inpu (or SDA) is sampled on the system clock, but is not passed forward to the next circuit unl outputs of both latches agree. If they do not agree, the previous value is held.

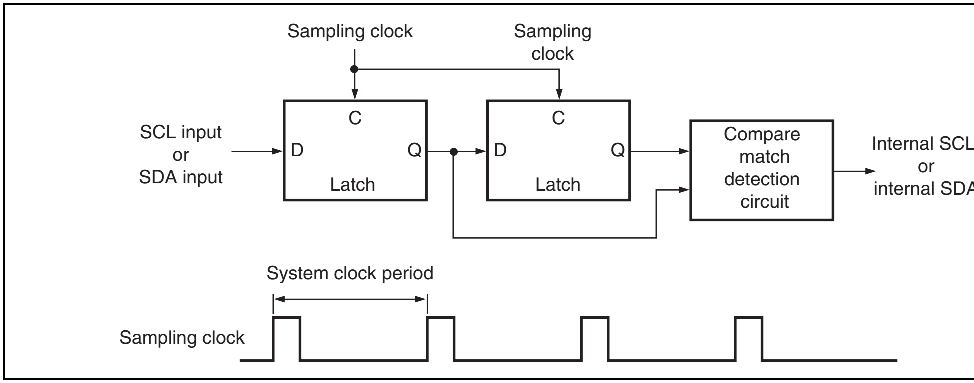
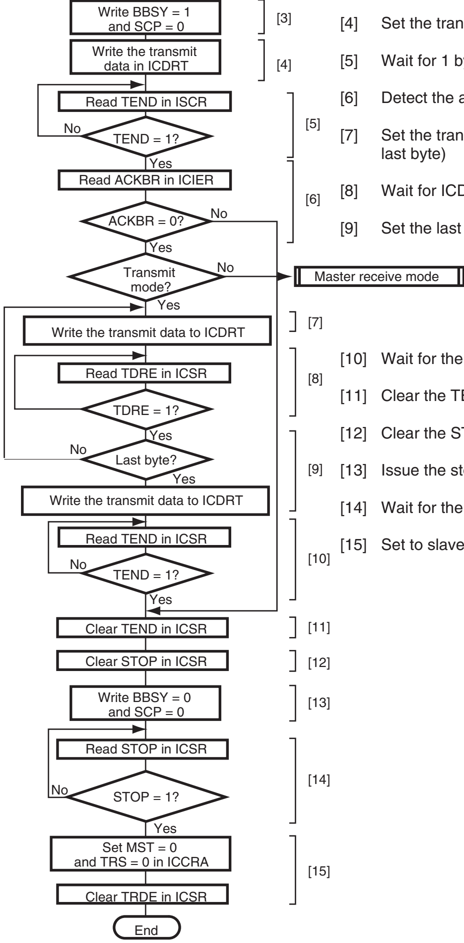


Figure 17.13 Block Diagram of Noise Canceler



[4] Set the transmit data for the first byte (slave address + R/W)

[5] Wait for 1 byte of data to be transmitted

[6] Detect the acknowledge bit, transferred from the specified slave dev

[7] Set the transmit data for the second and subsequent data (except fo last byte)

[8] Wait for ICDRT empty

[9] Set the last byte of transmit data

[10] Wait for the completion of transmission of the last byte

[11] Clear the TEND flag

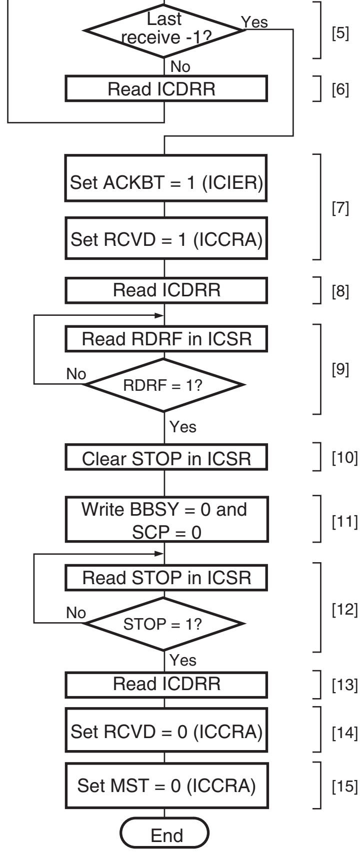
[12] Clear the STOP flag

[13] Issue the stop condition

[14] Wait for the creation of the stop condition

[15] Set to slave receive mode. Clear TDRE.

Figure 17.14 Sample Flowchart of Master Transmit Mode



[7] Set acknowledge of the last byte. Disable continuous rec (RCVD = 1).*2

[8] Read receive data of (last byte -1).*2

[9] Wait for the last byte to be received

[10] Clear the STOP flag

[11] Issue the stop condition

[12] Wait for the creation of stop condition

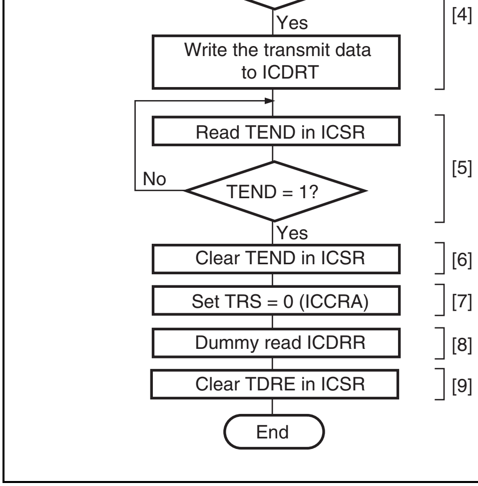
[13] Read the receive data of the last byte

[14] Clear RCVD to 0

[15] Set to slave receive mode

Note: 1. Do not generate an interrupt during steps [1] to [3].
2. For one-byte reception, steps [2] to [6] do not need to be executed. After step [1], execute step [7]. In step [8], read ICDRR (dummy read).

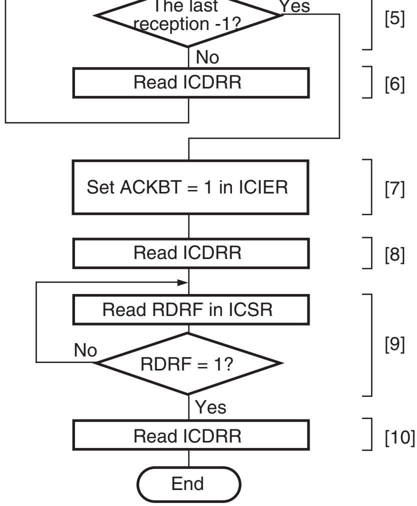
Figure 17.15 Sample Flowchart for Master Receive Mode



[8] Dummy read ICDRR to free the SCL line.

[9] Clear the TDRE flag.

Figure 17.16 Sample Flowchart for Slave Transmit Mode



[8] Read the receive data of (last byte -1).*

[9] Wait for the reception of the last byte to be comp

[10] Read the last byte of receive data.

Note: * For one-byte reception, steps [2] to [6] do not need to be executed. After step [1], execute step [7]. In step [8], read ICDRR (dummy read).

Figure 17.17 Sample Flowchart for Slave Receive Mode

Receive Data Full	RXI	(RDRF = 1) · (RIE = 1)
Stop Recognition	STPI	(STOP = 1) · (STIE = 1)
NACK Detection	MAKI	{(NACKF = 1) + (AL = 1)} · (NAKIE = 1)
Arbitration Lost		

17.6 Bit Synchronous Circuit

This module has a possibility that the high-level period is shortened in the two states des below.

In master mode,

- When SCL is driven low by the slave device
- When the rising speed of SCL is lowered by the load on the SCL line (load capacitan pull-up resistance)

Therefore, this module monitors SCL and communicates bit by bit in synchronization.

Figure 17.18 shows the timing of the bit synchronous circuit, and table 17.4 shows the ti SCL output changes from low to Hi-Z and the period which SCL is monitored.

Table 17.4 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL
0	0	7.5 tcyc
	1	19.5 tcyc
1	0	17.5 tcyc
	1	41.5 tcyc

17.7 Usage Notes

1. Confirm the ninth falling edge of the clock before issuing a stop or a repeated start co
   The ninth falling edge can be confirmed by monitoring the SCLO bit in the I^2C bus co register B (ICCRB).
   If a stop or a repeated start condition is issued at certain timing in either of the follow the stop or repeated start condition may be issued incorrectly.
   — The rising time of the SCL signal exceeds the time given in section 17.6, Bit Sync Circuit, because of the load on the SCL bus (load capacitance or pull-up resistance
   — The bit synchronous circuit is activated because a slave device holds the SCL bus during the eighth clock.
2. The WAIT bit in the I^2C bus mode register (ICMR) must be held 0.
   If the WAIT bit is set to 1, when a slave device holds the SCL signal low more than o transfer clock cycle during the eighth clock, the high level period of the ninth clock m shorter than a given period.

- Eight input channels
- Conversion time: 7.6 μs per channel (at 35-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
- Eight data registers

  A/D conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three types of conversion start

  Conversion can be started by software, a conversion start trigger by the 16-bit timer (TPU) or 8-bit timer (TMR), or an external trigger signal.
- Interrupt source

  A/D conversion end interrupt (ADI) request can be generated.
- Module stop state specifiable

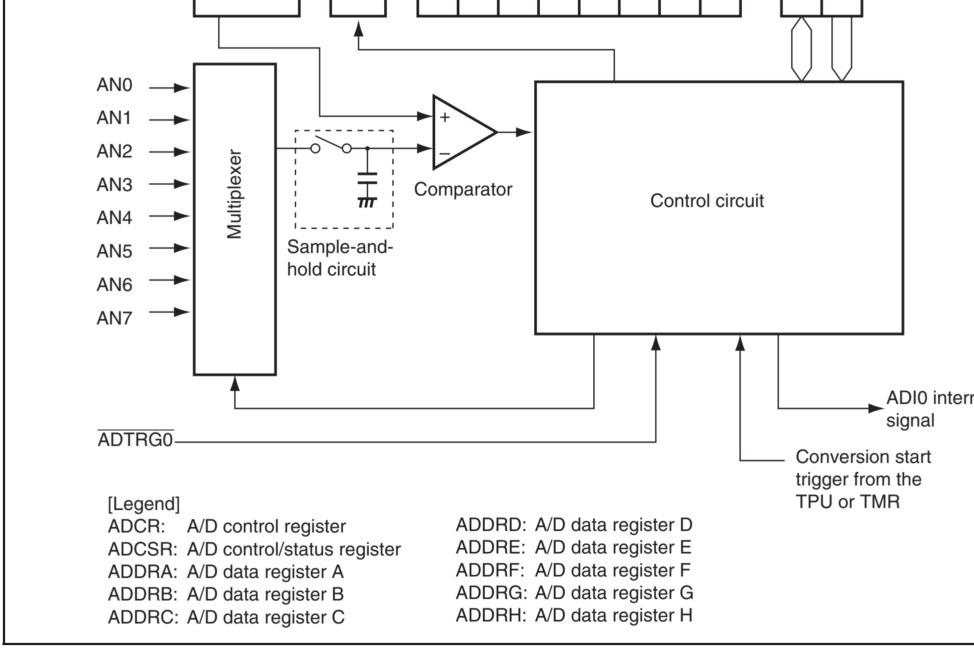


Figure 18.1 Block Diagram of A/D Converter

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	
A/D external trigger input pin	$\overline{\text{ADTRG0}}$	Input	External trigger input for starting A/D c
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

18.3 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)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Bit Name											—	—	—	—	—
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel	A/D Data Register Which Stores Conversion Resu
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD
AN4	ADDRE
AN5	ADDRF
AN6	ADDRG
AN7	ADDRH

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 conv [Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all specified in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.) • When the DMAC or DTC is activated by an interrupt and ADDR is read
6	ADIE	0	R/W	A/D Interrupt Enable When this bit is set to 1, ADI interrupts by ADF enabled.
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion, and converter enters wait state. Setting this bit to 1 starts A/D conversion. In sir this bit is cleared to 0 automatically when A/D c on the specified channel ends. In scan mode, A conversion continues sequentially on the specif channels until this bit is cleared to 0 by software or hardware standby mode.

0011: AN3
0100: AN4
0101: AN5
0110: AN6
0111: AN7
1XXX: Setting prohibited

- When SCANE = 1 and SCANS = 0

0000: AN0
0001: AN0 and AN1
0010: AN0 to AN2
0011: AN0 to AN3
0100: AN4
0101: AN4 and AN5
0110: AN4 to AN6
0111: AN4 to AN7
1XXX: Setting prohibited

- When SCANE = 1 and SCANS = 1

0000: AN0
0001: AN0 and AN1
0010: AN0 to AN2
0011: AN0 to AN3
0100: AN0 to AN4
0101: AN0 to AN5
0110: AN0 to AN6
0111: AN0 to AN7
1XXX: Setting prohibited

[Legend]
X: Don't care
Note: * Only 0 can be written to this bit, to clear the flag.

Bit	Bit Name	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 st conversion by a trigger signal. 00: A/D conversion start by external trigger is d 01: A/D conversion start by external trigger fron enabled 10: A/D conversion start by external trigger fron enabled 11: A/D conversion start by the $\overline{\text{ADTRG0}}$ pin is
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	These bits select the A/D conversion operating 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 and 0
2	CKS0	0	R/W	These bits set the A/D conversion time. Set bits and CKS0 only while A/D conversion is stoppec 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)

18.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has tw
operating modes: single mode and scan mode. When changing the operating mode or ana
channel, to prevent incorrect operation, first clear the ADST bit in ADCSR to 0 to halt A/
conversion. The ADST bit can be set to 1 at the same time as the operating mode or analo
channel is changed.

18.4.1 Single Mode

In single mode, A/D conversion is to be performed only once on the analog input of the s
single channel.

1. A/D conversion for the selected channel is started when the ADST bit in ADCSR is se
   software or an external trigger input.
2. When A/D conversion is completed, the A/D conversion result is transferred to the corresponding A/D data register of the channel.
3. When A/D conversion is completed, the ADF bit in ADCSR is set to 1. If the ADIE b
   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
   A/D conversion ends. The A/D converter enters wait state. If the ADST bit is cleared
   during A/D conversion, A/D conversion stops and the A/D converter enters wait state

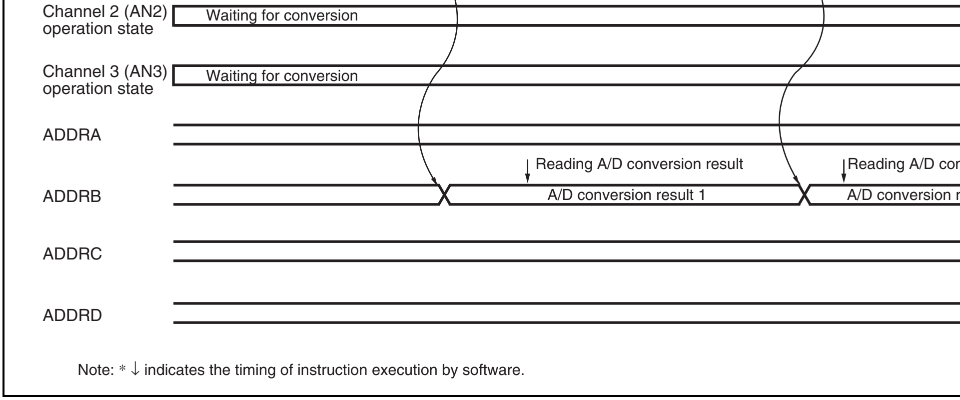
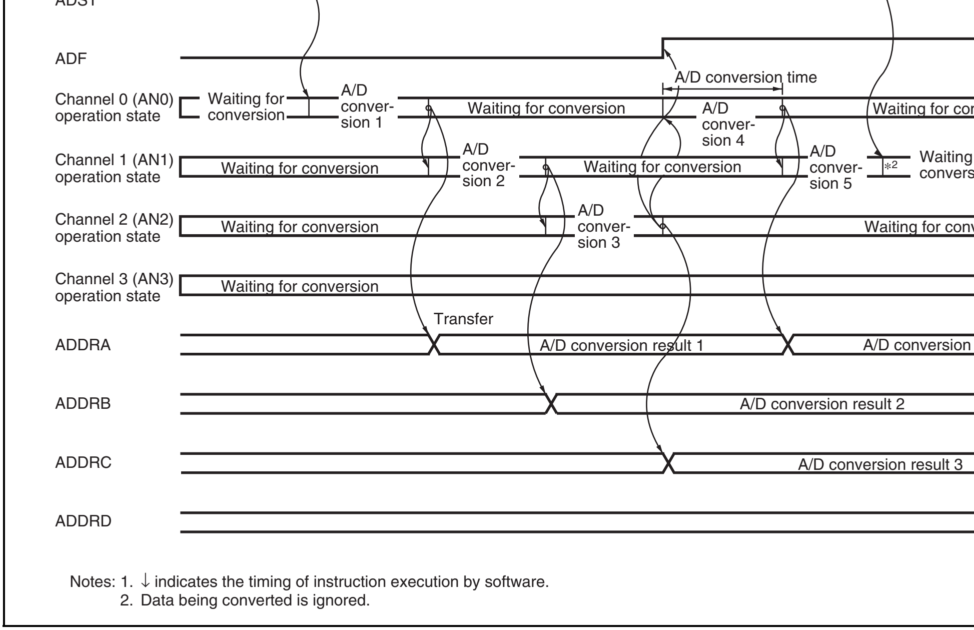


Figure 18.2 Example of A/D Converter Operation (Single Mode, Channel 1 Se

18.4.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the analog inputs of th
channels up to four or eight channels.

1. When the ADST bit in ADCSR is set to 1 by software, TPU, TMR, or an external tri input, A/D conversion starts on the first channel in the group. Consecutive A/D conv a maximum of four channels (SCANE and SCANS = B'10) or on a maximum of eigh channels (SCANE and SCANS = B'11) can be selected. When consecutive A/D conv performed on four channels, A/D conversion starts on AN4 when CH3 and CH2 = B consecutive A/D conversion is performed on eight channels, A/D conversion starts o when CH3 = B'0.
2. When A/D conversion for each channel is completed, the A/D conversion result is se transferred to the corresponding ADDR of each channel.



Notes: 1. ↓ indicates the timing of instruction execution by software.
2. Data being converted is ignored.

Figure 18.3 Example of A/D Conversion (Scan Mode, Three Channels (AN0 to AN2) Selected)

In scan mode, the values given in table 18.3 apply to the first conversion time. The valu table 18.4 apply to the second and subsequent conversions. In either case, bits CKS1 anc ADCR should be set so that the conversion time is within the ranges indicated by the A/ conversion characteristics.

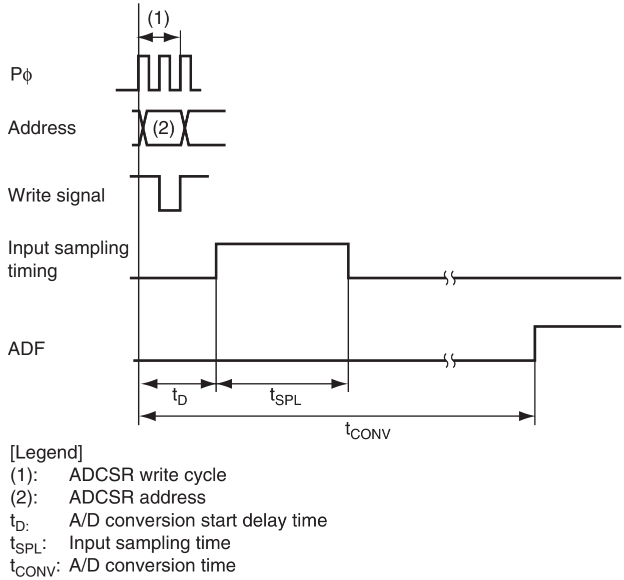


Figure 18.4 A/D Conversion Timing

Table 18.4 A/D Conversion Characteristics (Scan Mode)

CKS1	CKS0	Conversion Time (Number of States)
0	0	512 (Fixed)
	1	256 (Fixed)
1	0	128 (Fixed)
	1	64 (Fixed)

18.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set to ADCR, an external trigger is input from the $\overline{\text{ADTRG0}}$ pin. A/D conversion starts when th bit in ADCSR is set to 1 on the falling edge of the $\overline{\text{ADTRG0}}$ pin. Other operations, in bot and scan modes, are the same as when the ADST bit has been set to 1 by software. Figure shows the timing.

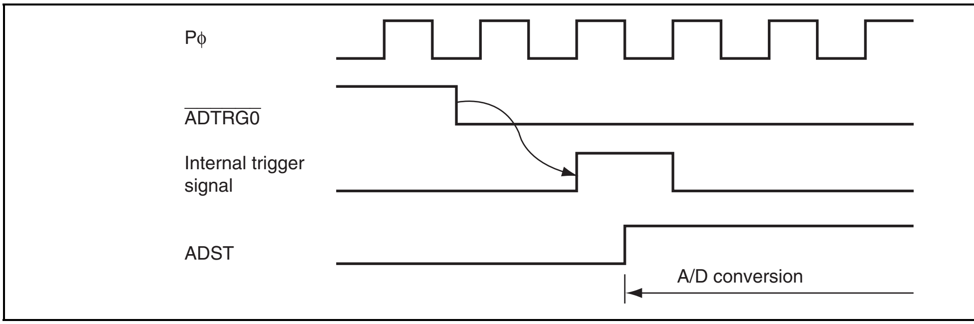


Figure 18.5 External Trigger Input Timing

ADI0	A/D conversion end	ADF	Possible	Possible

18.6 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 18.6).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion chara
  when the digital output changes from the minimum voltage value B'0000000000 (H'
  B'0000000001 (H'001) (see figure 18.7).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion chara
  when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H'3
  figure 18.7).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero v
  the full-scale voltage. Does not include the offset error, full-scale error, or quantizat
  (see figure 18.7).
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes the offs
  full-scale error, quantization error, and nonlinearity error.

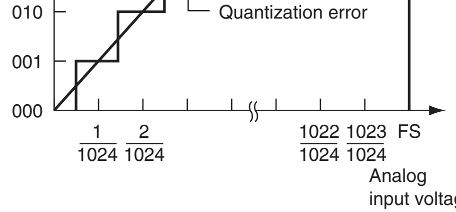


Figure 18.6 A/D Conversion Accuracy Definitions

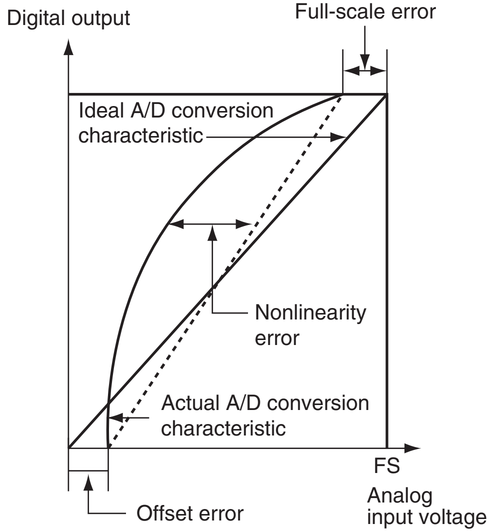


Figure 18.7 A/D Conversion Accuracy Definitions

This LSI's analog input is designed so that the conversion accuracy is guaranteed for an signal for which the signal source impedance is 10 kΩ or less. This specification is prov enable the A/D converter's sample-and-hold circuit input capacitance to be charged with sampling time; if the sensor output impedance exceeds 10 kΩ, charging may be insuffic may not be possible to guarantee the A/D conversion accuracy. However, if a large capa provided externally for conversion in single mode, the input load will essentially compri the internal input resistance of 10 kΩ, and the signal source impedance is ignored. Howe a low-pass filter effect is obtained in this case, it may not be possible to follow an analog with a large differential coefficient (e.g., 5 mV/μs or greater) (see figure 18.8). When co high-speed analog signal or conversion in scan mode, a low-impedance buffer should be

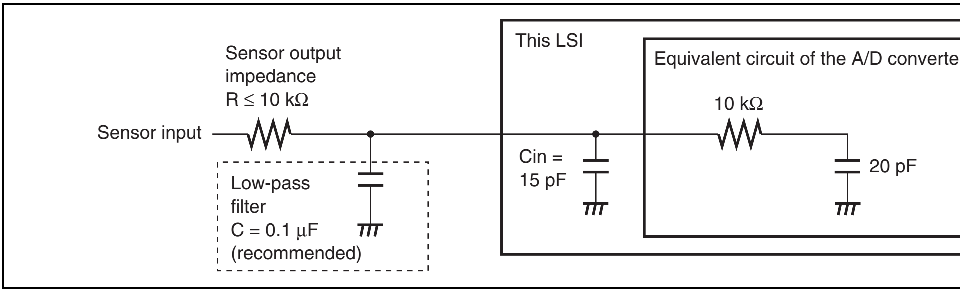


Figure 18.8 Example of Analog Input Circuit

18.7.3 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adv affect absolute accuracy. Be sure to make the connection to an electrically stable GND s AVss.

Care is also required to insure that digital signals on the board do not interfere with filte and filter circuits do not act as antennas.

- Vref setting range

  The reference voltage at the Vref pin should be set in the range Vref ≤ AVcc.

18.7.5 Notes on Board Design

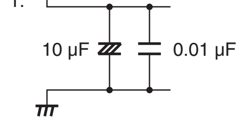
In board design, digital circuitry and analog circuitry should be as mutually isolated as po and layout in which digital circuit signal lines and analog circuit signal lines cross or are proximity should be avoided as far as possible. Failure to do so may result in incorrect op of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Digital circuitry must be isolated from the analog input pins (AN0 to AN7), analog refere power supply (Vref), and analog power supply (AVcc) by the analog ground (AVss). Als analog ground (AVss) should be connected at one point to a stable ground (Vss) on the bo

18.7.6 Notes on Noise Countermeasures

A protection circuit connected to prevent damage due to an abnormal voltage such as an e surge at the analog input pins (AN0 to AN7) should be connected between AVcc and AV shown in figure 18.9. Also, the bypass capacitors connected to AVcc and the filter capaci connected to the AN0 to AN7 pins must be connected to AVss.

If a filter capacitor is connected, the input currents at the AN0 to AN7 pins are averaged, error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if t current charged and discharged by the capacitance of the sample-and-hold circuit in the A converter exceeds the current input via the input impedance (R_{in}), an error will arise in the input pin voltage. Careful consideration is therefore required when deciding the circuit co

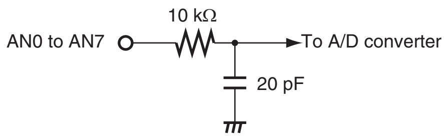


2. R_{in}: Input impedance

Figure 18.9 Example of Analog Input Protection Circuit

Table 18.6 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5	kΩ



Note: Values are reference values.

Figure 18.10 Analog Input Pin Equivalent Circuit

18.7.7 A/D Input Hold Function in Software Standby Mode

When this LSI enters software standby mode with A/D conversion enabled, the analog i
retained, and the analog power supply current is equal to as during A/D conversion. If th
power supply current needs to be reduced in software standby mode, clear the ADST, T
TRGS0 bits all to 0 to disable A/D conversion.

- Module stop state specifiable

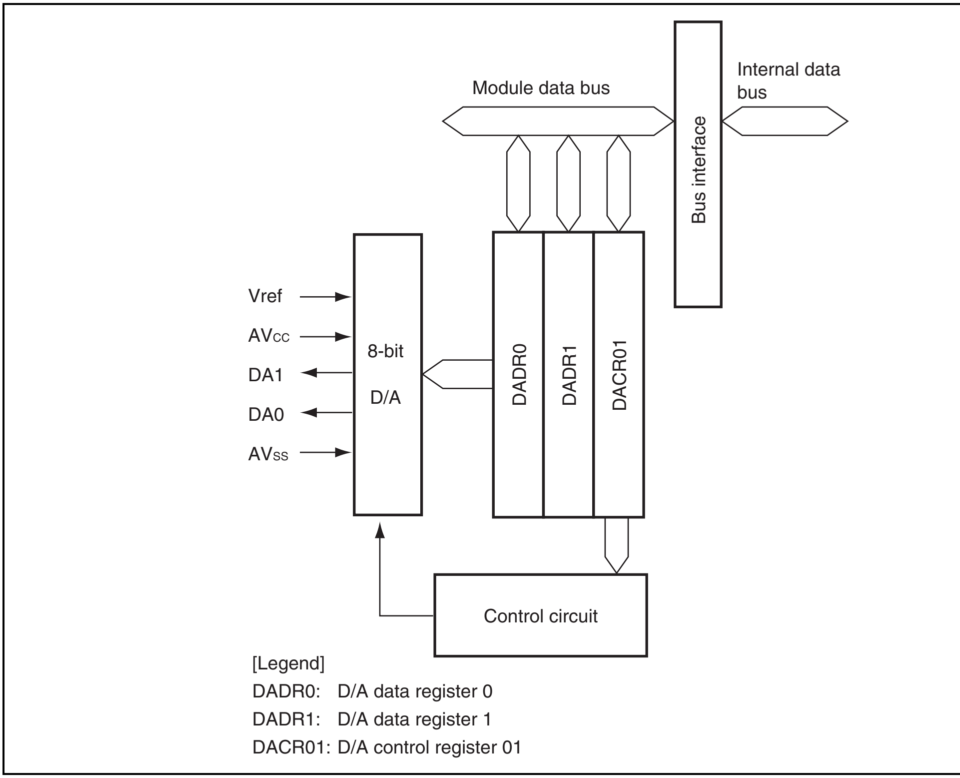


Figure 19.1 Block Diagram of D/A Converter

Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output

19.3 Register Descriptions

The D/A converter has the following registers.

- D/A data register 0 (DADR0)
- D/A data register 1 (DADR1)
- D/A control register 01 (DACR01)

19.3.1 D/A Data Registers 0 and 1 (DADR0 and DADR1)

DADR0 and DADR1 are 8-bit readable/writable registers that store data to which D/A co
is to be performed. Whenever an analog output is enabled, the values in DADR are conve
output to the analog output pins.

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	R/W	Description
7	DAOE1	0	R/W	D/A Output Enable 1 Controls D/A conversion and analog output. 0: Analog output of channel 1 (DA1) is disabled 1: D/A conversion of channel 1 is enabled. Ana of channel 1 (DA1) is enabled.
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output. 0: Analog output of channel 0 (DA0) is disabled 1: D/A conversion of channel 0 is enabled. Ana of channel 0 (DA0) is enabled.
5	DAE	0	R/W	D/A Enable Used together with the DAOE0 and DAOE1 bits D/A conversion. When this bit is cleared to 0, D conversion is controlled independently for chan 1. When this bit is set to 1, D/A conversion for c and 1 is controlled together. Output of conversion results is always controlle DAOE0 and DAOE1 bits. For details, see Table Control of D/A Conversion.
4 to 0	—	All 1	R	Reserved These are read-only bits and cannot be modifie

			Analog output of channel 0 (DA0) is disabled and an output of channel 1 (DA1) is enabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0 and DA1) is enabled.
1	0	0	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0 and DA1) is disabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channel 0 (DA0) is enabled and an output of channel 1 (DA1) is disabled.
	1	0	D/A conversion of channels 0 and 1 is enabled. Analog output of channel 0 (DA0) is disabled and an output of channel 1 (DA1) is enabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0 and DA1) is enabled.

from the analog output pin DA0 after the conversion time t_{DCONV} has elapsed. The cor
result continues to be output until DADR0 is written to again or the DAOE0 bit is cl
The output value is expressed by the following formula:

$$\text{Contents of DADR/256} \times V_{ref}$$

3. If DADR0 is written to again, the conversion is immediately started. The conversion
output after the conversion time t_{DCONV} has elapsed.
4. If the DAOE0 bit is cleared to 0, analog output is disabled.

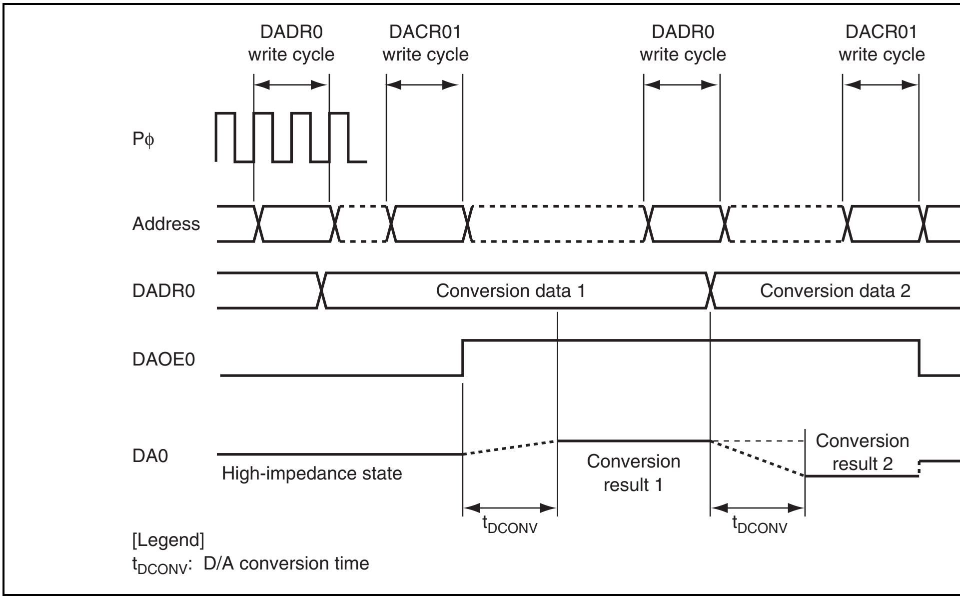


Figure 19.2 Example of D/A Converter Operation

When this LSI enters software standby mode with D/A conversion enabled, the D/A outp
retained, and the analog power supply current is equal to as during D/A conversion. If the
power supply current needs to be reduced in software standby mode, clear the ADST, TR
TRGS0 bits all to 0 to disable D/A conversion.

Flash memory version	H8SX/1663	40 kbytes	H'FF2000 to H'I
	H8SX/1664		

H8SX/1664	R5F61664	512 kbytes	H'000000 to H'07FFFF (modes 2, 3,

- Programming/erasing interface by the download of on-chip program

  This LSI has a programming/erasing program. After downloading this program to th
  RAM, programming/erasing can be performed by setting the parameters.
- Programming/erasing time

  Programming time: 3 ms (typ) for 128-byte simultaneous programming

  Erasing time: 2000 ms (typ) per 1 block (64 kbytes)
- Number of programming

  The number of programming can be up to 100 times at the minimum. (1 to 100 times
  guaranteed.)
- Three on-board programming modes

  SCI boot mode: Using the on-chip SCI_4, the user MAT can be programmed/er
  SCI boot mode, the bit rate between the host and this LSI can b
  automatically.

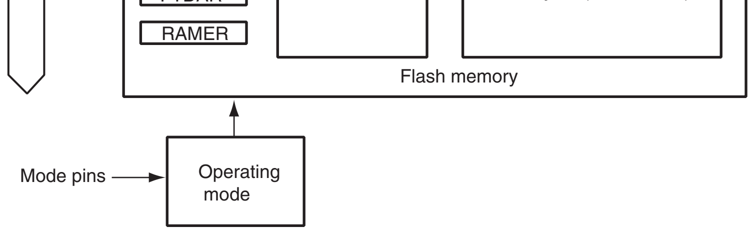
  USB boot mode: Using the on-chip USB, the user MAT can be programmed/era

  User program mode: Using a desired interface, the user MAT can be programmed/er
- Off-board programming mode

  Programmer mode: Using a PROM programmer, the user MAT can be programmed/
- Programming/erasing protection

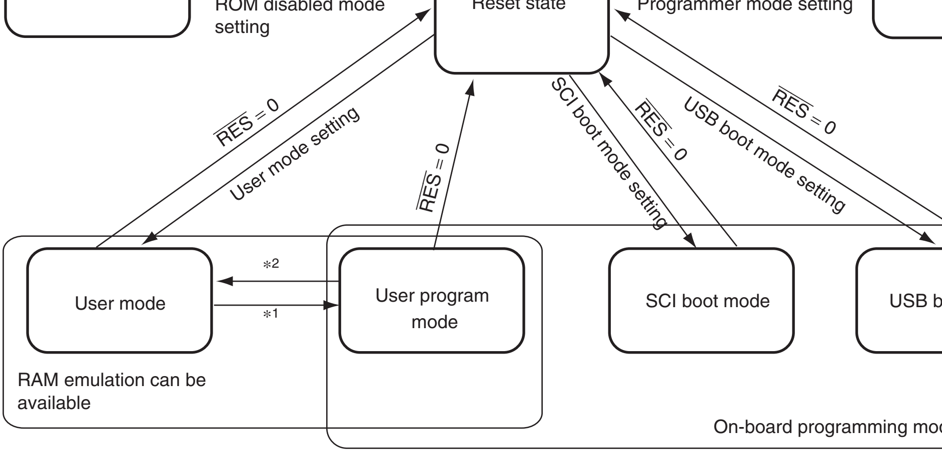
  Protection against programming/erasing of the flash memory can be set by hardware
  protection, software protection, or error protection.
- Flash memory emulation function using the on-chip RAM

  Realtime emulation of the flash memory programming can be performed by overlayi
  of the flash memory (user MAT) area and the on-chip RAM.



[Legend]
FCCS: Flash code control/status register
FPCS: Flash program code select register
FECS: Flash erase code select register
FKEY: Flash key code register
FTDAR: Flash transfer destination address register
RAMER: RAM emulation register

Figure 21.1 Block Diagram of Flash Memory



Notes: * In this LSI, the user program mode is defined as the period from the timing when a prog concerning programming and erasure is started in user mode to the timing when the pro completed.

1. Programming and erasure is started.
2. Programming and erasure is completed.

Figure 21.2 Mode Transition of Flash Memory

Block division erasure	O*[1]	O*[1]	O	×
Program data transfer	From host via SCI	From host via USB	From desired device via RAM	Via prog
RAM emulation	×	×	O	×
Reset initiation MAT	Embedded program storage area	Embedded program storage area	User MAT	—
Transition to user mode	Changing mode and reset	Changing mode and reset	Completing Programming/ erasure*[3]	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.

2. In this LSI, the user programming mode is defined as the period from the timin program concerning programming and erasure is started to the timing when th program is completed. For details on a program concerning programming and see section 21.7.3, User Program Mode.

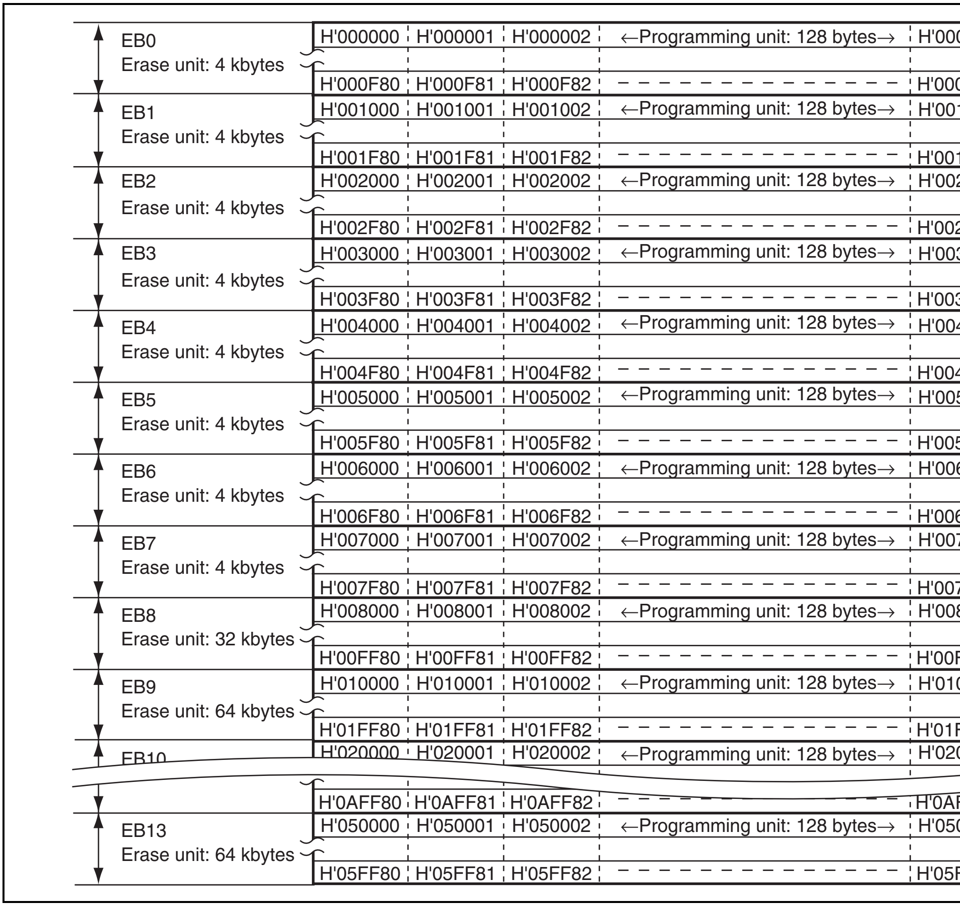


Figure 21.3 Block Structure of User MAT

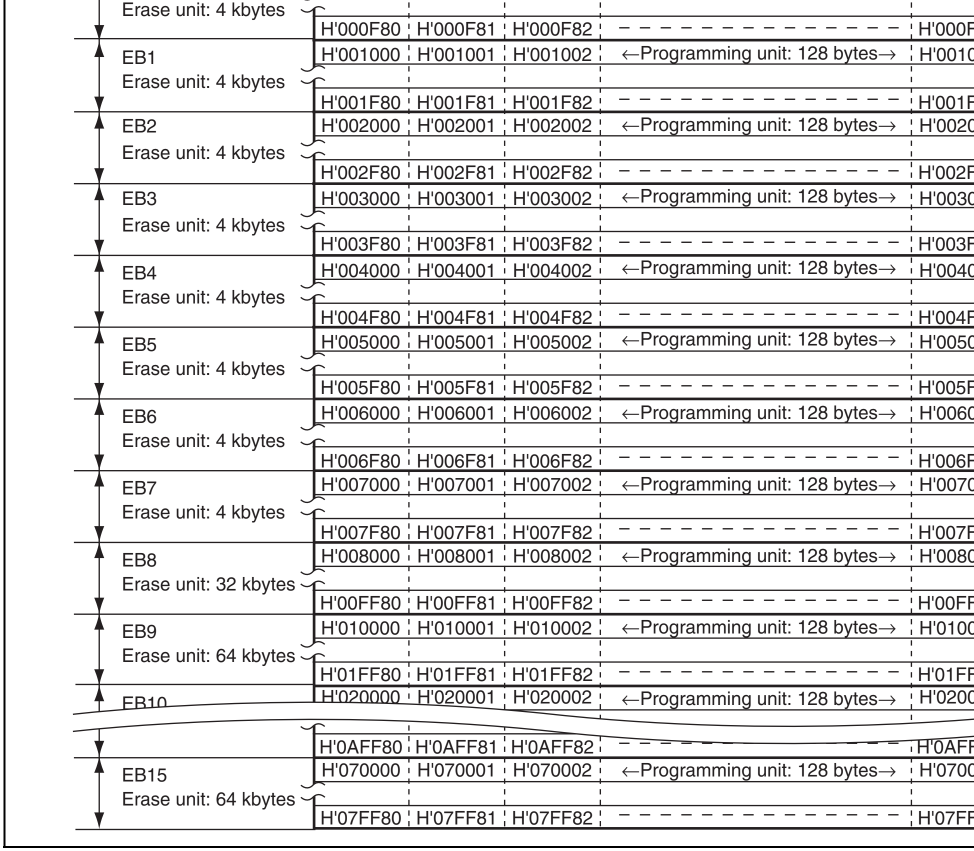


Figure 21.4 Block Structure of User MAT

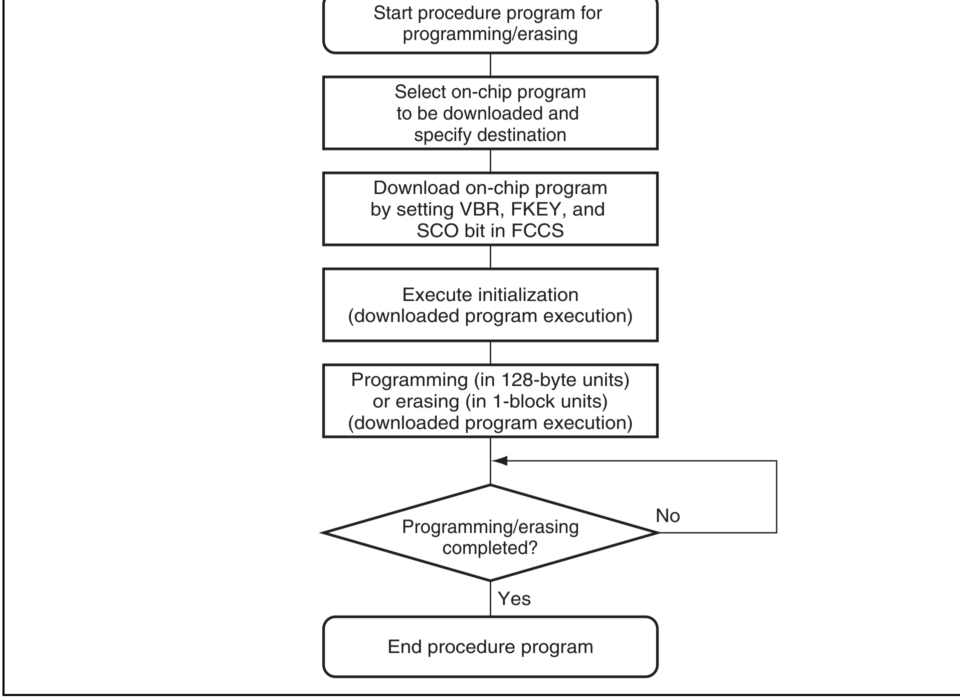


Figure 21.5 Procedure for Creating Procedure Program

(1) Selection of On-Chip Program to be Downloaded

This LSI has programming/erasing programs which can be downloaded to the on-chip R
on-chip program to be downloaded is selected by the programming/erasing interface reg
start address of the on-chip RAM where an on-chip program is downloaded is specified
flash transfer destination address register (FTDAR).

(3) Initialization of Programming/Erasing

A pulse with the specified period must be applied when programming or erasing. The spe
pulse width is made by the method in which wait loop is configured by the CPU instructio
Accordingly, the operating frequency of the CPU needs to be set before programming/era
operating frequency of the CPU is set by the programming/erasing interface parameter.

(4) Execution of Programming/Erasing

The start address of the programming destination and the program data are specified in 12
units when programming. The block to be erased is specified with the erase block numbe
erase-block units when erasing. Specifications of the start address of the programming de
program data, and erase block number are performed by the programming/erasing interfa
parameters, and the on-chip program is initiated. The on-chip program is executed by usin
JSR or BSR instruction and executing the subroutine call of the specified address in the o
RAM. The execution result is returned to the programming/erasing interface parameter.

The area to be programmed must be erased in advance when programming flash memory
interrupts are disabled during programming/erasing.

(5) When Programming/Erasing is Executed Consecutively

When processing does not end by 128-byte programming or 1-block erasure, consecutive
programming/erasing can be realized by updating the start address of the programming de
and program data, or the erase block number. Since the downloaded on-chip program is le
on-chip RAM even after programming/erasing completes, download and initialization are
required when the same processing is executed consecutively.

PM2	Input	SCI boot mode/USB boot mode setting (for boot mode setting by MD3 to MD0)
TxD4	Output	Serial transmit data output (used in SCI boot mode)
RxD4	Input	Serial receive data input (used in SCI boot mode)
USD+, USD−	I/O	USB data I/O (used in USB boot mode)
VBUS	Input	USB cable connection/disconnection detect (used in USB b
PM3	Input	USB bus power mode/self power mode setting (used in US mode)
PM4	Output	D+ pull-up control (used in USB boot mode)

- Flash transfer destination address register (FTDAR)

Programming/Erasing Interface Parameters:

- Download pass and fail result parameter (DPFR)
- Flash pass and fail result parameter (FPFR)
- Flash program/erase frequency parameter (FPEFEQ)
- Flash multipurpose address area parameter (FMPAR)
- Flash multipurpose data destination area parameter (FMPDR)
- Flash erase block select parameter (FEBS)

- RAM emulation register (RAMER)

There are several operating modes for accessing the flash memory. Respective operating registers, and parameters are assigned to the user MAT. The correspondence between ope modes and registers/parameters for use is shown in table 21.3.

erasing interface parameters	FPFR	—	O	O	O	—	—
	FPEFEQ	—	O	—	—	—	—
	FMPAR	—	—	O	—	—	—
	FMPDR	—	—	O	—	—	—
	FEBS	—	—	—	O	—	—
RAM emulation	RAMER	—	—	—	—	—	O

21.6.1 Programming/Erasing Interface Registers

The programming/erasing interface registers are 8-bit registers that can be accessed only
These registers are initialized by a power-on reset.

(1) Flash Code Control/Status Register (FCCS)

FCCS monitors errors during programming/erasing the flash memory and requests the o
program to be downloaded to the on-chip RAM.

Bit	7	6	5	4	3	2	1
Bit Name	—	—	—	FLER	—	—	—
Initial Value	1	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

				flash memory, the reset must be released after input period (period of $\overline{RES}$ = 0) of at least 100 0: Flash memory operates normally (Error pro invalid) [Clearing condition] • At a power-on reset 1: An error occurs during programming/erasin memory (Error protection is valid) [Setting conditions] • When an interrupt, such as NMI, occurs du programming/erasing. • When the flash memory is read during programming/erasing (including a vector re an instruction fetch). • When the SLEEP instruction is executed d programming/erasing (including software st mode). • When a bus master other than the CPU, su DMAC and DTC, obtains bus mastership d programming/erasing.
3 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modifie

Dummy read of FCCS must be executed twice immediately after setting this bit to 1. All interr be disabled during download. This bit is cleare when download is completed.

During program download initiated with this bi particular processing which accompanies ban switching of the program storage area is exec Before a download request, initialize the VBR to H'00000000. After download is completed, contents can be changed.

0: Download of the programming/erasing pro not requested.

[Clearing condition]

- When download is completed

1: Download of the programming/erasing pro requested.

[Setting conditions] (When all of the following are satisfied)

- Not in RAM emulation mode (the RAMS bi RAMER is cleared to 0)
- H'A5 is written to FKEY
- Setting of this bit is executed in the on-chi

Note: * This is a write-only bit. This bit is always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modifie
0	PPVS	0	R/W	Program Pulse Verify Selects the programming program to be downlo 0: Programming program is not selected. [Clearing condition] When transfer is completed 1: Programming program is selected.

(3) Flash Erase Code Select Register (FECS)

FECS selects the erasing program to be downloaded.

Bit	7	6	5	4	3	2	1	
Bit Name	—	—	—	—	—	—	—	E
Initial Value	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modifie
0	EPVB	0	R/W	Erase Pulse Verify Block Selects the erasing program to be downloaded 0: Erasing program is not selected. [Clearing condition] When transfer is completed 1: Erasing program is selected.

Bit	Bit Name	Value	R/W	Description
7	K7	0	R/W	Key Code
6	K6	0	R/W	When H'A5 is written to FKEY, writing to the S FCCS is enabled. When a value other than H' written, the SCO bit cannot be set to 1. Theref on-chip program cannot be downloaded to the RAM.
5	K5	0	R/W	
4	K4	0	R/W	
3	K3	0	R/W	
2	K2	0	R/W	Only when H'5A is written can programming/e the flash memory be executed. When a value H'5A is written, even if the programming/erasi program is executed, programming/erasing ca performed.
1	K1	0	R/W	H'A5: Writing to the SCO bit is enabled. (The cannot be set to 1 when FKEY is a valu than H'A5.)
0	K0	0	R/W	H'5A: Programming/erasing of the flash mem enabled. (When FKEY is a value other H'A5, the software protection state is e
				H'00: Initial value

Bit	Bit Name	Value	R/W	Description
7	TDER	0	R/W	Transfer Destination Address Setting Error This bit is set to 1 when an error has occurred i the start address specified by bits TDA6 to TD A start address error is determined by whether set in bits TDA6 to TDA0 is within the range of H'02 when download is executed by setting the in FCCS to 1. Make sure that this bit is cleared before setting the SCO bit to 1 and the value sp by bits TDA6 to TDA0 should be within the rang H'00 to H'02. 0: The value specified by bits TDA6 to TDA0 is the range. 1: The value specified by bits TDA6 to TDA0 is H'03 and H'FF and download has stopped.
6	TDA6	0	R/W	Transfer Destination Address
5	TDA5	0	R/W	Specifies the on-chip RAM start address of the download destination. A value between H'00 ar and up to 4 kbytes can be specified as the start of the on-chip RAM.
4	TDA4	0	R/W	H'00: H'FF9000 is specified as the star address.
3	TDA3	0	R/W	H'01: H'FFA000 is specified as the star address.
2	TDA2	0	R/W	H'02: H'FFB000 is specified as the star address.
1	TDA1	0	R/W	H'03 to H'7F: Setting prohibited. (Specifying a value from H'03 to the TDER bit to 1 and stops dow the on-chip program.)
0	TDA0	0	R/W	

is written in R0. The programming/erasing interface parameters are used in download co
initialization before programming or erasing, programming, and erasing. Table 21.4 sho
usable parameters and target modes. The meaning of the bits in the flash pass and fail re
parameter (FPFR) varies in initialization, programming, and erasure.

Table 21.4 Parameters and Target Modes

Parameter	Download	Initialization	Programming	Erasure	R/W	Initial Value	All
DPFR	O	—	—	—	R/W	Undefined	On
FPFR	O	O	O	O	R/W	Undefined	R0
FPEFEQ	—	O	—	—	R/W	Undefined	ER
FMPAR	—	—	O	—	R/W	Undefined	ER
FMPDR	—	—	O	—	R/W	Undefined	ER
FEBS	—	—	—	O	R/W	Undefined	ER

Note: * A single byte of the start address of the on-chip RAM specified by FTDAR

Download Control: The on-chip program is automatically downloaded by setting the S
FCCS to 1. The on-chip RAM area to download the on-chip program is the 4-kbyte area
from the start address specified by FTDAR. Download is set by the programming/erasin
registers, and the download pass and fail result parameter (DPFR) indicates the return va

register ER1. This parameter is called the flash multipurpose address area parameter (FM

The program data is always in 128-byte units. When the program data does not satisfy 12
128-byte program data is prepared by filling the dummy code (H'FF). The boundary of th
address of the programming destination on the user MAT is aligned at an address where t
eight bits (A7 to A0) are H'00 or H'80.

The program data for the user MAT must be prepared in consecutive areas. The program
must be in a consecutive space which can be accessed using the MOV.B instruction of th
and is not in the flash memory space.

The start address of the area that stores the data to be written in the user MAT must be se
general register ER0. This parameter is called the flash multipurpose data destination area
parameter (FMPDR).

For details on the programming procedure, see section 21.7.3, User Program Mode.

Erasure: When the flash memory is erased, the erase block number on the user MAT mu
passed to the erasing program which is downloaded.

The erase block number on the user MAT must be set in general register ER0. This paran
called the flash erase block select parameter (FEBS).

One block is selected from the block numbers of 0 to 13 as the erase block number.

For details on the erasing procedure, see section 21.7.3, User Program Mode.

7 to 3	—	—	—	Unused These bits return 0.
2	SS	—	R/W	Source Select Error Detect Only one type can be specified for the on-chip which can be downloaded. When the program downloaded is not selected, more than two ty programs are selected, or a program which is mapped is selected, an error occurs. 0: Download program selection is normal 1: Download program selection is abnormal
1	FK	—	R/W	Flash Key Register Error Detect Checks the FKEY value (H'A5) and returns th 0: FKEY setting is normal (H'A5) 1: FKEY setting is abnormal (value other tha
0	SF	—	R/W	Success/Fail Returns the download result. Reads back the downloaded to the on-chip RAM and determin whether it has been transferred to the on-chip 0: Download of the program has ended norm error) 1: Download of the program has ended abno (error occurs)

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	—	—	Unused These bits return 0.
1	FQ	—	R/W	Frequency Error Detect Compares the specified CPU operating frequer the operating frequencies supported by this LS returns the result. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	—	R/W	Success/Fail Returns the initialization result. 0: Initialization has ended normally (no error) 1: Initialization has ended abnormally (error occ

(b) Programming

FPFR indicates the return value of the programming result.

Bit	7	6	5	4	3	2	1
Bit Name	—	MD	EE	FK	—	WD	WA

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused Returns 0.

				performed (FLER = 1)
5	EE	—	R/W	Programming Execution Error Detect Writes 1 to this bit when the specified data cou written because the user MAT was not erased is set to 1, there is a high possibility that the u has been written to partially. In this case, afte the error factor, erase the user MAT. 0: Programming has ended normally 1: Programming has ended abnormally (prog result is not guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Checks the FKEY value (H'5A) before progran starts, and returns the result. 0: FKEY setting is normal (H'5A) 1: FKEY setting is abnormal (value other than
3	—	—	—	Unused Returns 0.
2	WD	—	R/W	Write Data Address Detect When an address not in the flash memory are specified as the start address of the storage d for the program data, an error occurs. 0: Setting of the start address of the storage destination for the program data is normal 1: Setting of the start address of the storage destination for the program data is abnorm

				1: Setting of the start address of the programm destination is abnormal
0	SF	—	R/W	Success/Fail Returns the programming result. 0: Programming has ended normally (no error) 1: Programming has ended abnormally (error o

(c) Erasure

FPFR indicates the return value of the erasure result.

Bit	7	6	5	4	3	2	1
Bit Name	—	MD	EE	FK	EB	—	—

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused Returns 0.
6	MD	—	R/W	Erasure Mode Related Setting Error Detect Detects the error protection state and returns th When the error protection state is entered, this to 1. Whether the error protection state is enter can be confirmed with the FLER bit in FCCS. F conditions to enter the error protection state, se 21.8.3, Error Protection. 0: Normal operation (FLER = 0) 1: Error protection state, and programming ca performed (FLER = 1)

4	FK	—	R/W	Flash Key Register Error Detect Checks the FKEY value (H'5A) before erasure and returns the result. 0: FKEY setting is normal (H'5A) 1: FKEY setting is abnormal (value other than
3	EB	—	R/W	Erase Block Select Error Detect Checks whether the specified erase block nun the block range of the user MAT, and returns t 0: Setting of erase block number is normal 1: Setting of erase block number is abnormal
2, 1	—	—	—	Unused These bits return 0.
0	SF	—	R/W	Success/Fail Indicates the erasure result. 0: Erasure has ended normally (no error) 1: Erasure has ended abnormally (error occur

Bit	15	14	13	12	11	10	9
Bit Name	F15	F14	F13	F12	F11	F10	F9

Bit	7	6	5	4	3	2	1
Bit Name	F7	F6	F5	F4	F3	F2	F1

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Unused These bits should be cleared to 0.
15 to 0	F15 to F0	—	R/W	Frequency Set These bits set the operating frequency of the C When the PLL multiplication function is used, s multiplied frequency. The setting value must be calculated as follows: 1. The operating frequency shown in MHz uni be rounded in a number of three decimal pl be shown in a number of two decimal place 2. The value multiplied by 100 is converted to binary digit and is written to FPEFEQ (gene register ER0). For example, when the operating frequency of is 35.000 MHz, the value is as follows: 1. The number of three decimal places of 35.0 rounded. 2. The formula of 35.00 × 100 = 3500 is conve the binary digit and B'0000 1101 1010 1100 (H'0DAC) is set to ER0.

Bit	23	22	21	20	19	18	17	
Bit Name	MOA23	MOA22	MOA21	MOA20	MOA19	MOA18	MOA17	M

Bit	15	14	13	12	11	10	9
Bit Name	MOA15	MOA14	MOA13	MOA12	MOA11	MOA10	MOA9

Bit	7	6	5	4	3	2	1
Bit Name	MOA7	MOA6	MOA5	MOA4	MOA3	MOA2	MOA1

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOA31 to MOA0	—	R/W	These bits store the start address of the progr destination on the user MAT. Consecutive 128 programming is executed starting from the spe start address of the user MAT. Therefore, the start address of the programming destination I 128-byte boundary, and MOA6 to MOA0 are a cleared to 0.

Bit	23	22	21	20	19	18	17	
Bit Name	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18	MOD17	MO

Bit	15	14	13	12	11	10	9	
Bit Name	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9	M

Bit	7	6	5	4	3	2	1	
Bit Name	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1	M

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOD31 to MOD0	—	R/W	These bits store the start address of the area w stores the program data for the user MAT. Con 128-byte data is programmed to the user MAT from the specified start address.

Bit	31	30	29	28	27	26	25
Bit Name							
Initial Value	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	23	22	21	20	19	18	17
Bit Name							
Initial Value	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9
Bit Name							
Initial Value	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial Value	0	0	0	0	0	0	0	
R/W	R	R	R	R	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	0	R	Reserved These are read-only bits and cannot be modifie
3	RAMS	0	R/W	RAM Select Selects the function which emulates the flash m using the on-chip RAM. 0: Disables RAM emulation function 1: Enables RAM emulation function (all blocks c user MAT are protected against programmin erasing)
2 1 0	RAM2 RAM1 RAM0	0 0 0	R/W R/W R/W	Flash Memory Area Select These bits select the user MAT area overlaid w on-chip RAM when RAMS = 1. The following a correspond to the 4-kbyte erase blocks. 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)

Mode Setting	EMLE	MD3	MD2	MD1	MD0	P
SCI boot mode	0	0	0	1	0	0
USB boot mode						1
User program mode			1	1	—	—

21.7.1 SCI Boot Mode

SCI boot mode executes programming/erasing of the user MAT by means of the control and program data transmitted from the externally connected host via the on-chip SCI_4.

In SCI boot mode, the tool for transmitting the control command and program data, and program data must be prepared in the host. The serial communication mode is set to asy mode. The system configuration in SCI boot mode is shown in figure 21.6. Interrupts ar in SCI boot mode. Configure the user system so that interrupts do not occur.

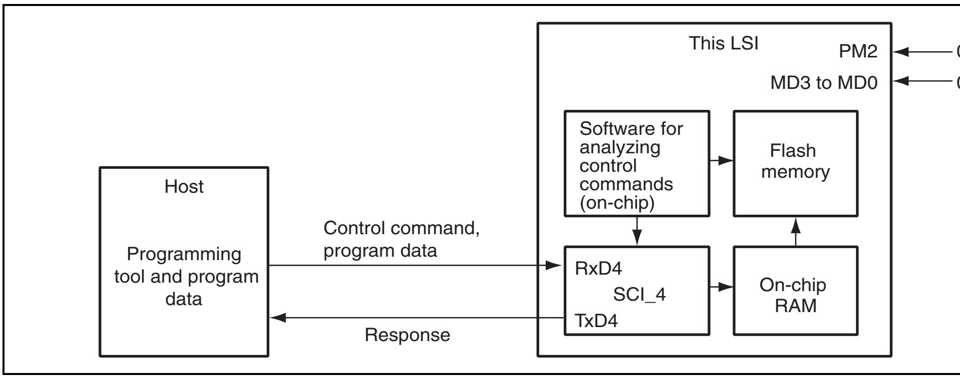


Figure 21.6 System Configuration in SCI Boot Mode

adjustment end sign. When the host receives this bit adjustment end sign normally, it tran
byte of H'55 to this LSI. When reception is not executed normally, initiate boot mode aga
bit rate may not be adjusted within the allowable range depending on the combination of
rate of the host and the system clock frequency of this LSI. Therefore, the transfer bit rate
host and the system clock frequency of this LSI must be as shown in table 21.6.

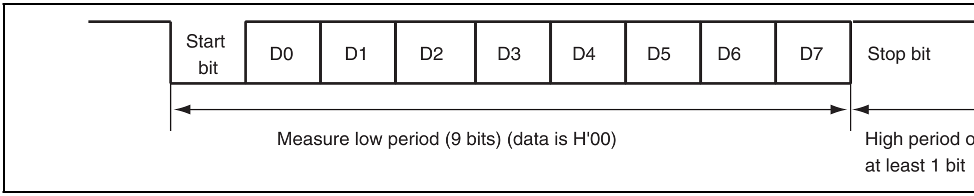


Figure 21.7 Automatic-Bit-Rate Adjustment Operation

Table 21.6 System Clock Frequency for Automatic-Bit-Rate Adjustment

Bit Rate of Host	System Clock Frequency of This LSI
9,600 bps	8 to 18 MHz
19,200 bps	8 to 18 MHz

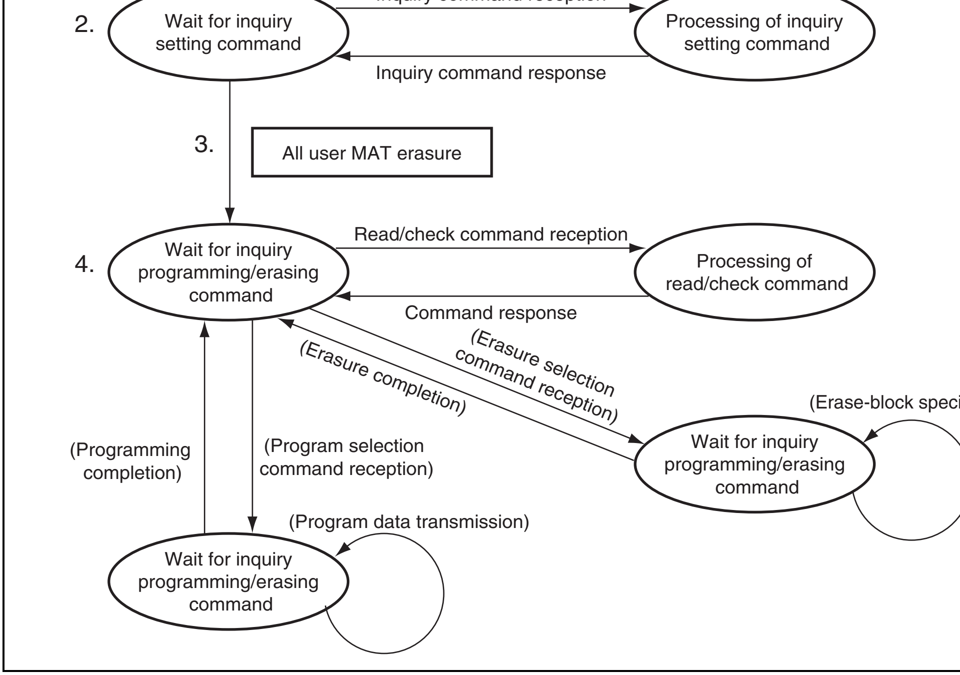


Figure 21.8 SCI Boot Mode State Transition Diagram

waiting for erase block data is entered. The erase block number must be transmitted a
erasing command is transmitted. When the erasure is finished, the erase block number
set to H'FF and transmitted. Then the state of waiting for erase block data is returned t
state of waiting for programming/erasing command. Erasure must be executed when t
specified block is programmed without a reset start after programming is executed in
mode. When programming can be executed by only one operation, all blocks are erase
entering the state of waiting for programming/erasing command or another command
this case, the erasing operation is not required. The commands other than the
programming/erasing command perform sum check, blank check (erasure check), and
read of the user MAT and acquisition of current status information.

Memory read of the user MAT can only read the data programmed after all user MAT ha
automatically been erased. No other data can be read.

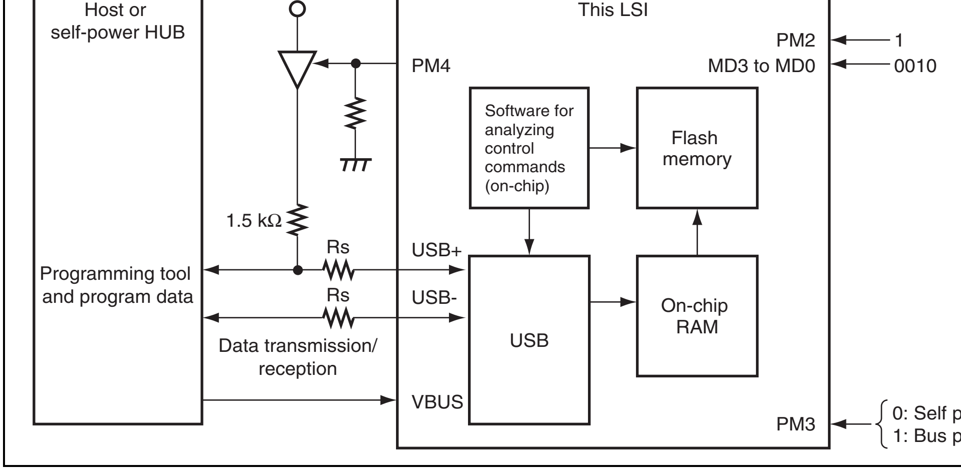


Figure 21.9 System Configuration in USB Boot Mode

	For bus power mode (PM3 = 1)	500
Endpoint configuration	EP0 Control (in out) 8 bytes Configuration 1 └ InterfaceNumber0 └ AlternateSetting0 ├ EP1 Bulk (out) 64 bytes └ EP2 Bulk (in) 64 bytes	

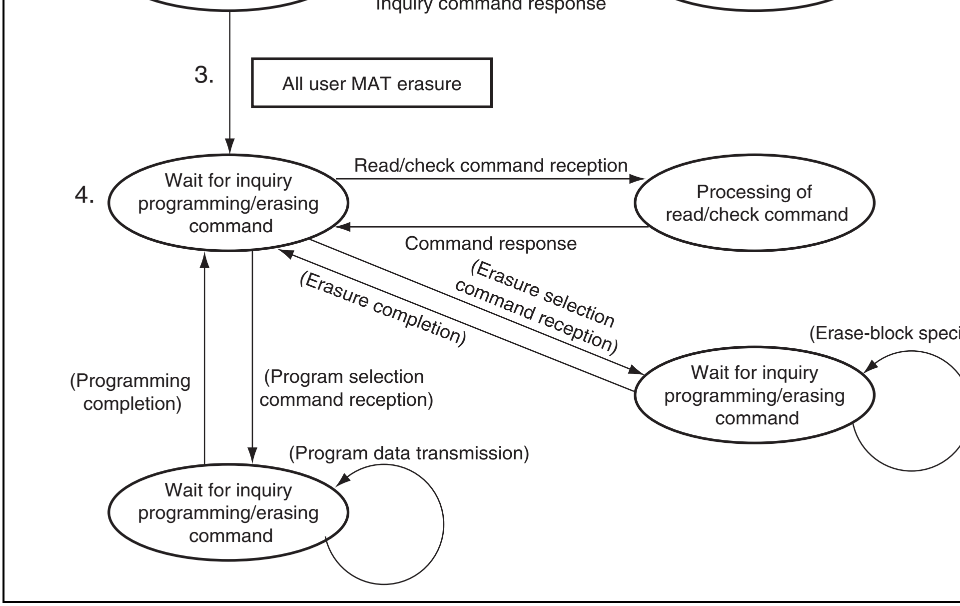
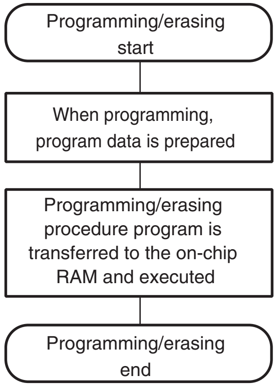


Figure 21.10 USB Boot Mode State Transition Diagram

check), and memory read of the user MAT, and acquiring the current status informatio

(3) Notes on USB Boot Mode Execution

- The clock of 48 MHz needs to be supplied to the USB module. Set the external clock frequency and clock pulse generator so as to supply 48 MHz as the clock for the USB For details, refer to section 22, Clock Pulse Generator.
- Use the PM4 pin for the D+ pull-up control connection.
- For the stable supply of the power during the flash memory programming and erasing cable should not be connected via the bus powered HUB.
- If the bus powered HUB is disconnected during the flash memory programming and e permanent damage to the LSI may result.
- If the USB bus in the bus power mode enters the suspend mode, this does not make th transition to the software standby mode in the power-down state.



1. Exit RAM emulation mode beforehand. Download is not allow in emulation mode.
2. When the program data is adjusted in emulation mode, selec the download destination specified by FTDAR carefully. Make sure that the download area does not overlap the emulation area.
3. Programming/erasing is executed only in the on-chip RAM.
4. After programming/erasing is finished, protect the flash memo by the hardware protection.

Figure 21.11 Programming/Erasing Flow

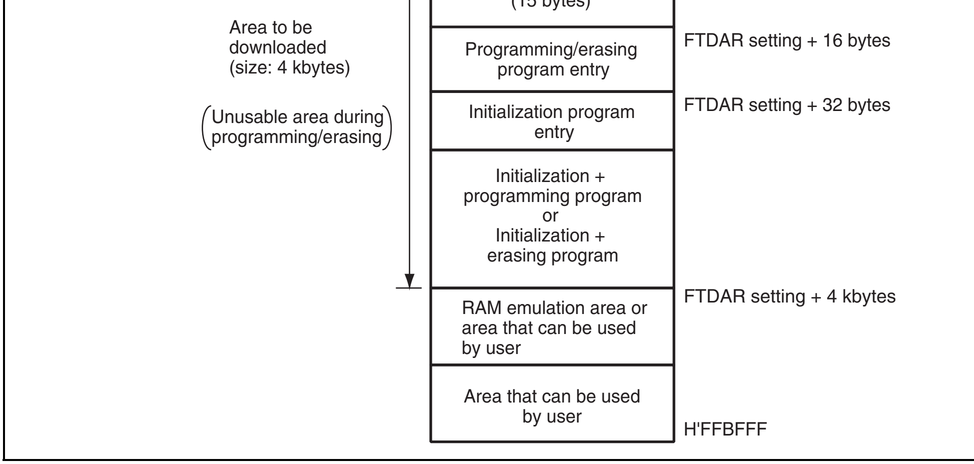


Figure 21.12 RAM Map when Programming/Erasing is Executed

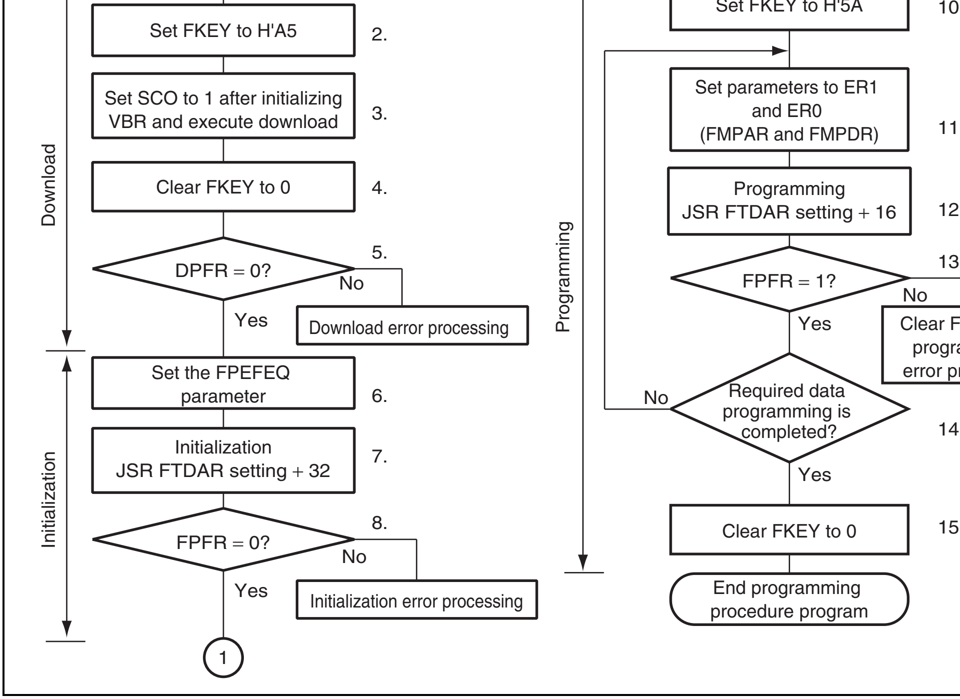


Figure 21.13   Programming Procedure in User Program Mode

H'FF, the program processing time can be shortened.

1. Select the on-chip program to be downloaded and the download destination. When th bit in FPCS is set to 1, the programming program is selected. Several programming/e programs cannot be selected at one time. If several programs are selected, a downloa returned to the SS bit in the DPFR parameter. The on-chip RAM start address of the d destination is specified by FTDAR.
2. Write H'A5 in FKEY. If H'A5 is not written to FKEY, the SCO bit in FCCS cannot b to request download of the on-chip program.
3. After initializing VBR to H'00000000, set the SCO bit to 1 to execute download. To s SCO bit to 1, all of the following conditions must be satisfied.
   - RAM emulation mode has been canceled.
   - H'A5 is written to FKEY.
   - Setting the SCO bit is executed in the on-chip RAM.

   When the SCO bit is set to 1, download is started automatically. Since the SCO bit is to 0 when the procedure program is resumed, the SCO bit cannot be confirmed to be procedure program. The download result can be confirmed by the return value of the parameter. To prevent incorrect decision, before setting the SCO bit to 1, set one byte on-chip RAM start address specified by FTDAR, which becomes the DPFR paramete value other than the return value (e.g. H'FF). Since particular processing that is accom by bank switching as described below is performed when download is executed, initi VBR contents to H'00000000. Dummy read of FCCS must be performed twice imme after the SCO bit is set to 1.
   - The user-MAT space is switched to the on-chip program storage area.
   - After the program to be downloaded and the on-chip RAM start address specified FTDAR are checked, they are transferred to the on-chip RAM.
   - FPCS, FECS, and the SCO bit in FCCS are cleared to 0.

— If access to the flash memory is requested by the DMAC or DTC during downloa operation cannot be guaranteed. Make sure that an access request by the DMAC not generated.

4. FKEY is cleared to H'00 for protection.
5. The download result must be confirmed by the value of the DPFR parameter. Check of the DPFR parameter (one byte of start address of the download destination specif FTDAR). If the value of the DPFR parameter is H'00, download has been performed If the value is not H'00, the source that caused download to fail can be investigated b description below.
   — If the value of the DPFR parameter is the same as that before downloading, the s the start address of the download destination in FTDAR may be abnormal. In thi confirm the setting of the TDER bit in FTDAR.
   — If the value of the DPFR parameter is different from that before downloading, ch bit or FK bit in the DPFR parameter to confirm the download program selection setting, respectively.
6. The operating frequency of the CPU is set in the FPEFEQ parameter for initializatio settable operating frequency of the FPEFEQ parameter ranges from 8 to 50 MHz. W frequency is set otherwise, an error is returned to the FPFR parameter of the initializ program and initialization is not performed. For details on setting the frequency, see 21.6.2 (3), Flash Program/Erase Frequency Parameter (FPEFEQ: General Register E CPU).

— Since the stack area is used in the initialization program, a stack area of 128 bytes maximum must be allocated in RAM.

— Interrupts can be accepted during execution of the initialization program. Make su program storage area and stack area in the on-chip RAM and register values are no overwritten.

8. The return value in the initialization program, the FPFR parameter is determined.
9. All interrupts and the use of a bus master other than the CPU are disabled during programming/erasing. The specified voltage is applied for the specified time when programming or erasing. If interrupts occur or the bus mastership is moved to other th CPU during programming/erasing, causing a voltage exceeding the specifications to b applied, the flash memory may be damaged. Therefore, interrupts are disabled by sett (I bit) in the condition code register (CCR) to B'1 in interrupt control mode 0 and by s bits 2 to 0 (I2 to I0 bits) in the extend register (EXR) to B'111 in interrupt control mod Accordingly, interrupts other than NMI are held and not executed. Configure the user so that NMI interrupts do not occur. The interrupts that are held must be executed afte programming completes. When the bus mastership is moved to other than the CPU, su the DMAC or DTC, the error protection state is entered. Therefore, make sure the DM not acquire the bus.
10. FKEY must be set to H'5A and the user MAT must be prepared for programming.

executed and an error is returned to the FPFR parameter. In this case, the program must be transferred to the on-chip RAM and then programming must be executed

12. Programming is executed. The entry point of the programming program is at the add is 16 bytes after #DLTOP (start address of the download destination specified by FT Call the subroutine to execute programming by using the following steps.

```
MOV.L     #DLTOP+16,ER2    ; Set entry address to ER2
JSR       @ER2             ; Call programming routine
NOP
```

— The general registers other than ER0 or ER1 are held in the programming progra
— R0L is a return value of the FPFR parameter.
— Since the stack area is used in the programming program, a stack area of 128 byt maximum must be allocated in RAM.

13. The return value in the programming program, the FPFR parameter is determined.
14. Determine whether programming of the necessary data has finished. If more than 12 data are to be programmed, update the FMPAR and FMPDR parameters in 128-byte repeat steps 11 to 14. Increment the programming destination address by 128 bytes a the programming data pointer correctly. If an address which has already been progra written to again, not only will a programming error occur, but also flash memory wil damaged.
15. After programming finishes, clear FKEY and specify software protection. If this LSI restarted by a reset immediately after programming has finished, secure the reset inp (period of $\overline{\text{RES}} = 0$) of at least 100 μs.

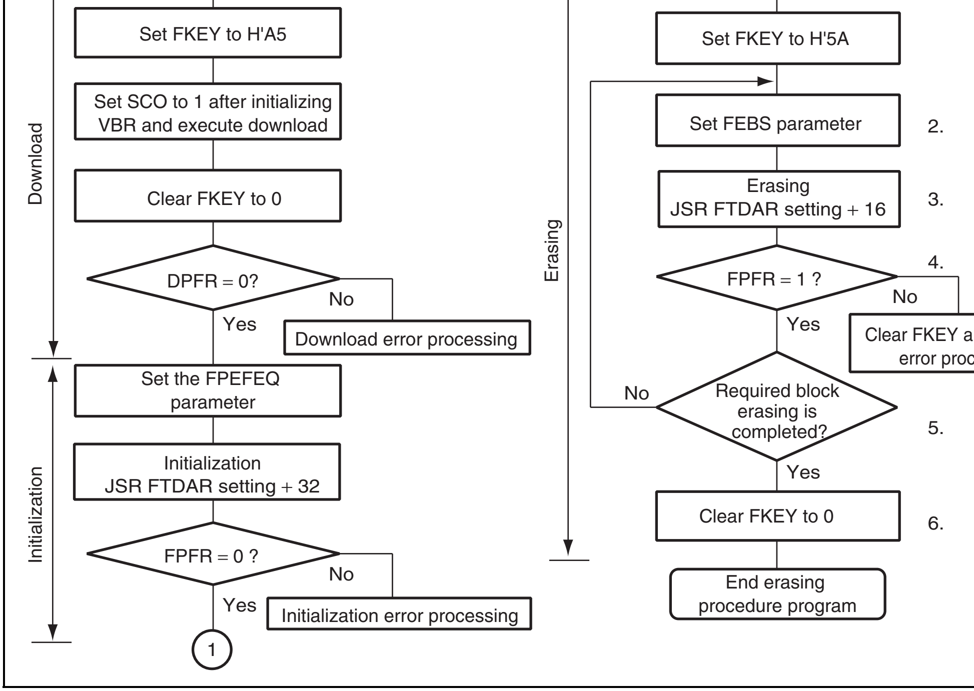


Figure 21.14 Erasing Procedure in User Program Mode

bit in FPCS is set to 1, the programming program is selected. Several programming/
programs cannot be selected at one time. If several programs are selected, a downloa
returned to the SS bit in the DPFR parameter. The on-chip RAM start address of the
destination is specified by FTDAR.

For the procedures to be carried out after setting FKEY, see section 21.7.3 (2), Progr
Procedure in User Program Mode.

2. Set the FEBS parameter necessary for erasure. Set the erase block number (FEBS pa
of the user MAT in general register ER0. If a value other than an erase block number
user MAT is set, no block is erased even though the erasing program is executed, and
is returned to the FPFR parameter.

3. Erasure is executed. Similar to as in programming, the entry point of the erasing prog
the address which is 16 bytes after #DLTOP (start address of the download destinati
specified by FTDAR). Call the subroutine to execute erasure by using the following

```
MOV.L #DLTOP+16, ER2      ; Set entry address to ER2
JSR   @ER2                ; Call erasing routine
NOP
```

- The general registers other than ER0 or ER1 are held in the erasing program.
- R0L is a return value of the FPFR parameter.
- Since the stack area is used in the erasing program, a stack area of 128 bytes at th
maximum must be allocated in RAM.

4. The return value in the erasing program, the FPFR parameter is determined.

5. Determine whether erasure of the necessary blocks has finished. If more than one blo
be erased, update the FEBS parameter and repeat steps 2 to 5.

6. After erasure completes, clear FKEY and specify software protection. If this LSI is r
a power-on reset immediately after erasure has finished, secure the reset input period
of $\overline{\text{RES}}$ = 0) of at least 100 μs.

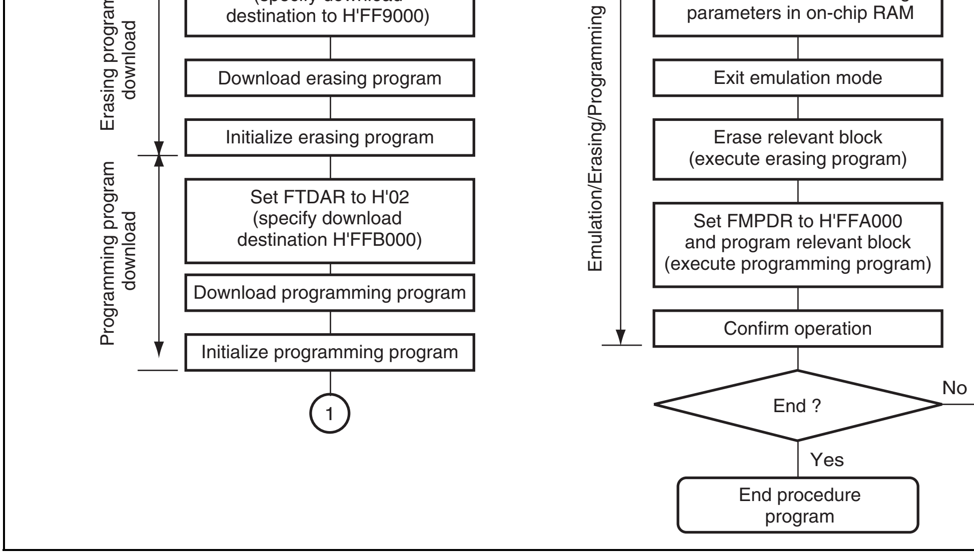


Figure 21.15 Repeating Procedure of Erasing, Programming, and RAM Emulation in User Program Mode

In Figure 21.15, since RAM emulation is performed, the erasing/programming program i
downloaded to avoid the 4-kbyte on-chip RAM area (H'FFA000 to H'FFAFFF). Downloa
initialization are performed only once at the beginning. Note the following when executi
procedure program.

- Be careful not to overwrite data in the on-chip RAM with overlay settings. In additio
programming program area, erasing program area, and RAM emulation area, areas fo
procedure programs, work area, and stack area are reserved in the on-chip RAM. Do
settings that will overwrite data in these areas.

- The on-chip program is downloaded to and executed in the on-chip RAM specified b
FTDAR. Therefore, this on-chip RAM area is not available for use.
- Since the on-chip program uses a stack area, allocate 128 bytes at the maximum as a
area.
- Download requested by setting the SCO bit in FCCS to 1 should be executed from th
RAM because it will require switching of the memory MATs.
- In an operating mode in which the external address space is not accessible, such as s
mode, the required procedure programs, NMI handling vector table, and NMI handli
should be transferred to the on-chip RAM before programming/erasing starts (downl
is determined).
- The flash memory is not accessible during programming/erasing. Programming/erasi
executed by the program downloaded to the on-chip RAM. Therefore, the procedure
that initiates operation, the NMI handling vector table, and the NMI handling routine
stored in the on-chip RAM other than the flash memory.
- After programming/erasing starts, access to the flash memory should be inhibited un
is cleared. The reset input state (period of $\overline{RES}$ = 0) must be set to at least 100 μs wh
operating mode is changed and the reset start executed on completion of programmin
Transitions to the reset state are inhibited during programming/erasing. When the res
is input, a reset input state (period of $\overline{RES}$ = 0) of at least 100 μs is needed before the
signal is released.
- When the program data storage area is within the flash memory area, an error will oc
when the data stored is normal program data. Therefore, the data should be transferre
on-chip RAM to place the address that the FMPDR parameter indicates in an area ot
the flash memory.

Table 21.9 Usable Area for Programming in User Program Mode

Item	Storable/Executable Area		Selected M	
	On-Chip RAM	User MAT	User MAT	Embe Progr Stora
Storage area for program data	O	×*	—	—
Operation for selecting on-chip program to be downloaded	O	O	O	
Operation for writing H'A5 to FKEY	O	O	O	
Execution of writing 1 to SCO bit in FCCS (download)	O	×		O
Operation for clearing FKEY	O	O	O	
Decision of download result	O	O	O	
Operation for download error	O	O	O	
Operation for setting initialization parameter	O	O	O	
Execution of initialization	O	×	O	
Decision of initialization result	O	O	O	
Operation for initialization error	O	O	O	
NMI handling routine	O	×	O	
Operation for disabling interrupts	O	O	O	
Operation for writing H'5A to FKEY	O	O	O	
Operation for setting programming parameter	O	×	O	
Execution of programming	O	×	O	
Decision of programming result	O	×	O	
Operation for programming error	O	×	O	
Operation for clearing FKEY	O	×	O	

Note: * Transferring the program data to the on-chip RAM beforehand enables this are used.

RENESAS

Operation for clearing FKEY	O	O	O
Decision of download result	O	O	O
Operation for download error	O	O	O
Operation for setting initialization parameter	O	O	O
Execution of initialization	O	×	O
Decision of initialization result	O	O	O
Operation for initialization error	O	O	O
NMI handling routine	O	×	O
Operation for disabling interrupts	O	O	O
Operation for writing H'5A to FKEY	O	O	O
Operation for setting erasure parameter	O	×	O
Execution of erasure	O	×	O
Decision of erasure result	O	×	O
Operation for erasure error	O	×	O
Operation for clearing FKEY	O	×	O

program is initiated, and the error in programming/erasing is indicated by the FPFR para

Table 21.11 Hardware Protection

		Function to be Pro	
Item	**Description**	**Download**	**Progra Erasin**
Reset protection	• The programming/erasing interface registers are initialized in the reset state (including a reset by the WDT) and the programming/erasing protection state is entered. • The reset state will not be entered by a reset using the $\overline{RES}$ pin unless the $\overline{RES}$ pin is held low until oscillation has settled after a power is initially supplied. In the case of a reset during operation, hold the $\overline{RES}$ pin low for the $\overline{RES}$ pulse width given in the AC characteristics. If a reset is input during programming or erasure, data in the flash memory is not guaranteed. In this case, execute erasure and then execute programming again.	O	O

by SCO bit	entered when the SCO bit in FCCS is cleared to 0 to disable download of the programming/erasing programs.		
Protection by FKEY	The programming/erasing protection state is entered because download and programming/erasing are disabled unless the required key code is written in FKEY.	O	O
Emulation protection	The programming/erasing protection state is entered when the RAMS bit in the RAM emulation register (RAMER) is set to 1.	O	O

21.8.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when a CPU runa occurs or operations not according to the programming/erasing procedures are detected programming/erasing of the flash memory. Aborting programming or erasure in such ca prevents damage to the flash memory due to excessive programming or erasing.

If an error occurs during programming/erasing of the flash memory, the FLER bit in FC to 1 and the error protection state is entered.

- When an interrupt request, such as NMI, occurs during programming/erasing.
- When the flash memory is read from during programming/erasing (including a vecto an instruction fetch).
- When a SLEEP instruction is executed (including software-standby mode) during programming/erasing.
- When a bus master other than the CPU, such as the DMAC and DTC, obtains bus m during programming/erasing.

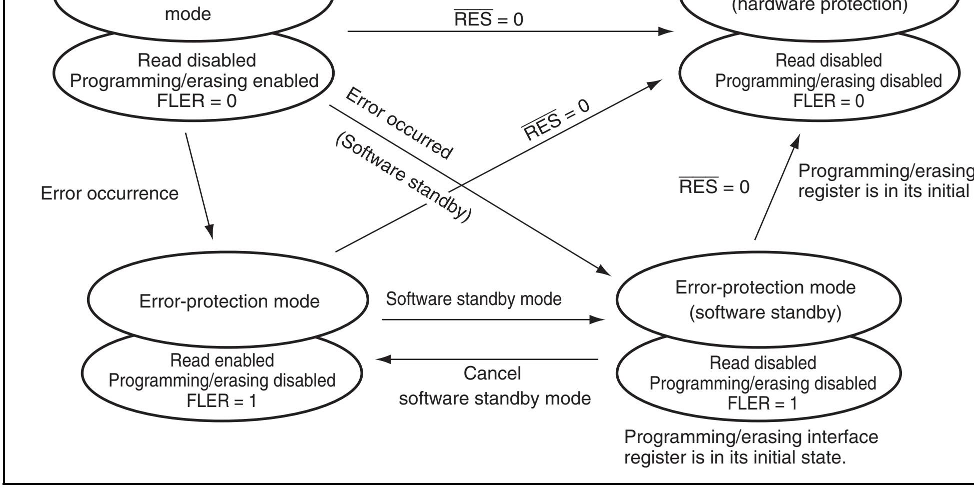


Figure 21.16 Transitions to Error Protection State

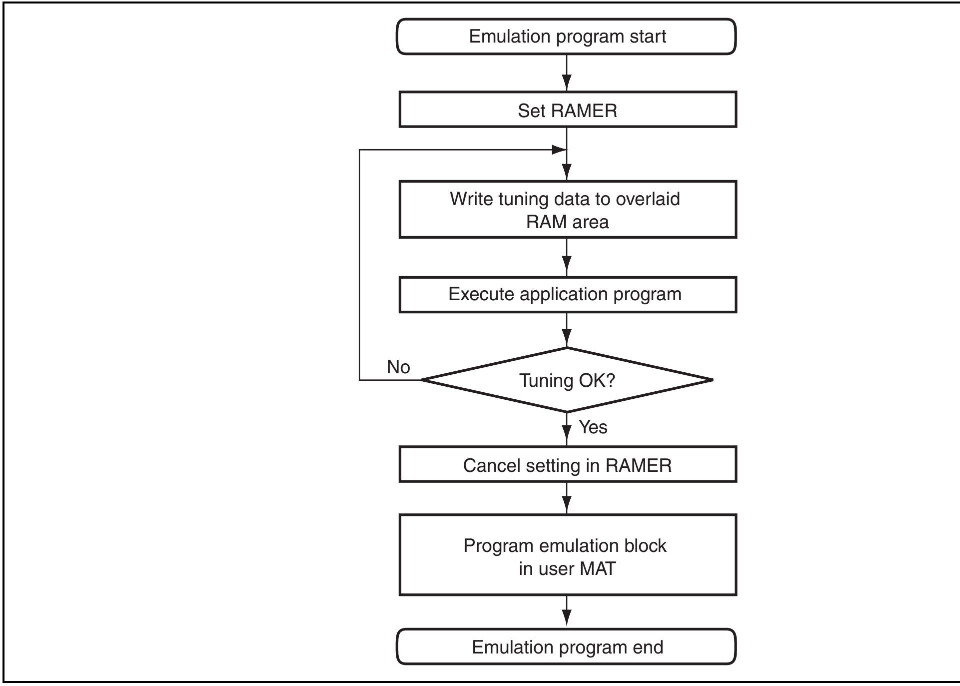
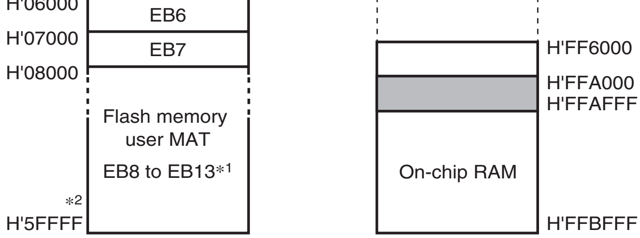


Figure 21.17 RAM Emulation Flow



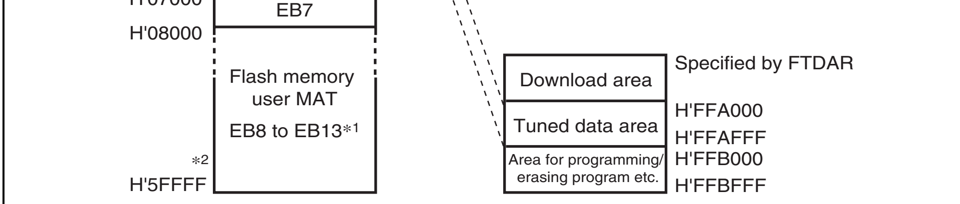
Notes: 1. EB8 to EB15 in the H8SX/1664.
2. H'7FFFF in the H8SX/1664.

Figure 21.18 Address Map of Overlaid RAM Area (H8SX/1663)

The flash memory area that can be emulated is the one area selected by bits RAM2 to RA
RAMER from among the eight blocks, EB0 to EB7, of the user MAT.

To overlay a part of the on-chip RAM with block EB0 for realtime emulation, set the RA
RAMER to 1 and bits RAM2 to RAM0 to B'000.

For programming/erasing the user MAT, the procedure programs including a download p
of the on-chip program must be executed. At this time, the download area should be speci
that the overlaid RAM area is not overwritten by downloading the on-chip program. Sinc
in which the tuned data is stored is overlaid with the download area when FTDAR = H'01
tuned data must be saved in an unused area beforehand.



Notes: 1. EB8 to EB15 in the H8SX/1664.
2. H'7FFFF in the H8SX/1664.

Figure 21.19 Programming Tuned Data (H8SX/1663)

1. After tuning program data is completed, clear the RAMS bit in RAMER to 0 to canc
   overlaid RAM.
2. Transfer the user-created procedure program to the on-chip RAM.
3. Start the procedure program and download the on-chip program to the on-chip RAM
   address of the download destination should be specified by FTDAR so that the tuned
   does not overlay the download area.
4. When block EB0 of the user MAT has not been erased, the programming program m
   downloaded after block EB0 is erased. Specify the tuned data saved in the FMPAR a
   FMPDR parameters and then execute programming.

Note: Setting the RAMS bit to 1 makes all the blocks of the user MAT enter the programming/erasing protection state (emulation protection state) regardless of t of the RAM2 to RAM0 bits. Under this condition, the on-chip program cannot b downloaded. When data is to be actually programmed and erased, clear the RAM to 0.

User MAT	H8SX/1663	384 kbytes	FZTAT512V3A
	H8SX/1664	512 kbytes	

21.11 Standard Serial Communication Interface Specifications for B Mode

The boot program initiated in boot mode performs serial communication using the host ar chip SCI_4. The serial communication interface specifications are shown below.

The boot program has three states.

1. Bit-rate-adjustment state

   In this state, the boot program adjusts the bit rate to achieve serial communication wit host. Initiating boot mode enables starting of the boot program and entry to the bit-rat adjustment state. The program receives the command from the host to adjust the bit ra adjusting the bit rate, the program enters the inquiry/selection state.

2. Inquiry/selection state

   In this state, the boot program responds to inquiry commands from the host. The devic clock mode, and bit rate are selected. After selection of these settings, the program is enter the programming/erasing state by the command for a transition to the programming/erasing state. The program transfers the libraries required for erasure to chip RAM and erases the user MATs before the transition.

3. Programming/erasing state

   Programming and erasure by the boot program take place in this state. The boot progr made to transfer the programming/erasing programs to the on-chip RAM by comman the host. Sum checks and blank checks are executed by sending these commands from host.

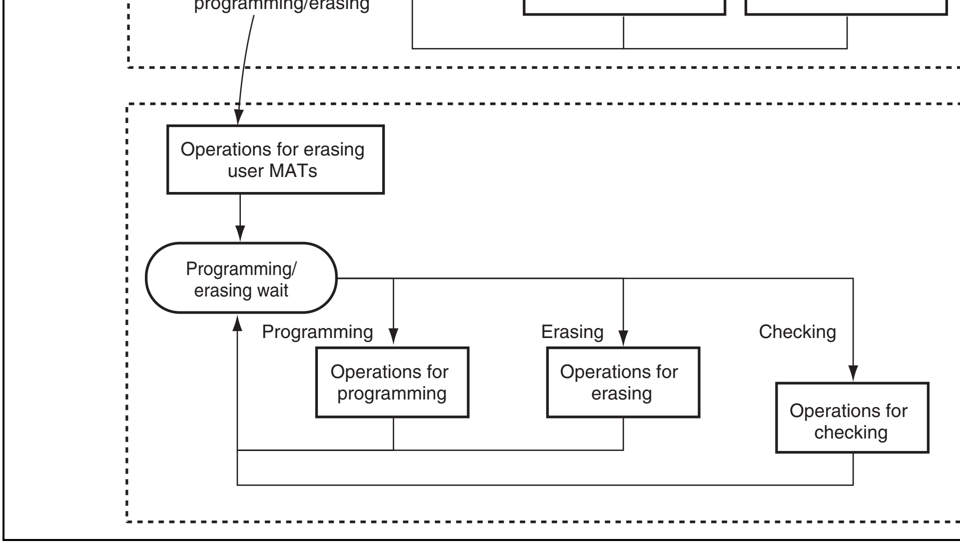


Figure 21.20 Boot Program States

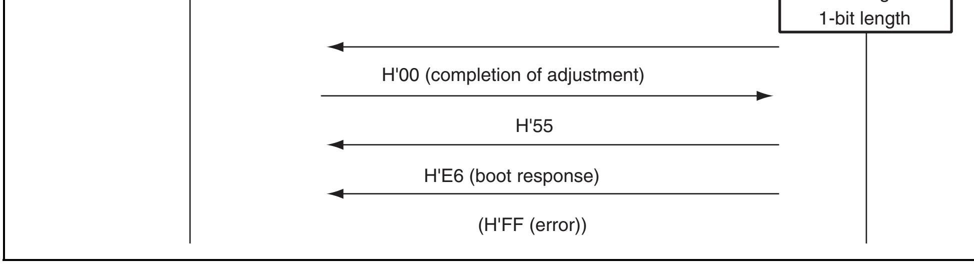


Figure 21.21 Bit-Rate-Adjustment Sequence

(2) Communications Protocol

After adjustment of the bit rate, the protocol for serial communications between the host a
boot program is as shown below.

1. One-byte commands and one-byte responses

   These one-byte commands and one-byte responses consist of the inquiries and the AC
   successful completion.

2. n-byte commands or n-byte responses

   These commands and responses are comprised of n bytes of data. These are selections
   responses to inquiries.

   The program data size is not included under this heading because it is determined in a
   command.

3. Error response

   The error response is a response to inquiries. It consists of an error response and an er
   and comes two bytes.

4. Programming of 128 bytes

   The size is not specified in commands. The size of n is indicated in response to the
   programming unit inquiry.

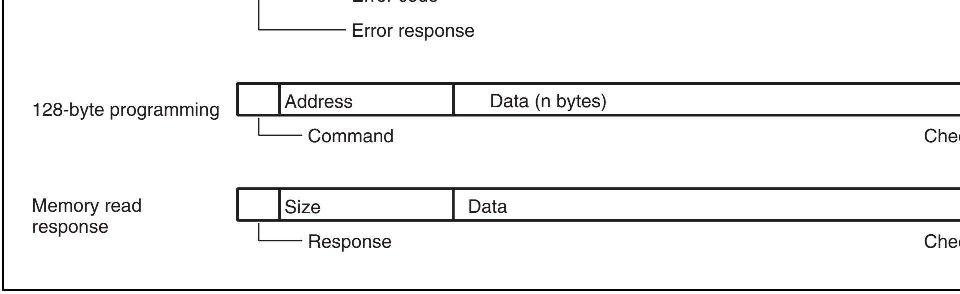


Figure 21.22 Communication Protocol Format

- Command (one byte): Commands including inquiries, selection, programming, erasi checking
- Response (one byte): Response to an inquiry
- Size (one byte): The amount of data for transmission excluding the command, amou and checksum
- Checksum (one byte): The checksum is calculated so that the total of all values from command byte to the SUM byte becomes H'00.
- Data (n bytes): Detailed data of a command or response
- Error response (one byte): Error response to a command
- Error code (one byte): Type of the error
- Address (four bytes): Address for programming
- Data (n bytes): Data to be programmed (the size is indicated in the response to the programming unit inquiry.)
- Size (four bytes): Four-byte response to a memory read

H'10	Device selection	Selection of device code
H'21	Clock mode inquiry	Inquiry regarding numbers of clock modes values of each mode
H'11	Clock mode selection	Indication of the selected clock mode
H'22	Multiplication ratio inquiry	Inquiry regarding the number of frequency-multiplied clock types, the number of multip ratios, and the values of each multiple
H'23	Operating clock frequency inquiry	Inquiry regarding the maximum and minimu values of the main clock and peripheral clo
H'25	User MAT information inquiry	Inquiry regarding the a number of user MA the start and last addresses of each MAT
H'26	Block for erasing information Inquiry	Inquiry regarding the number of blocks and and last addresses of each block
H'27	Programming unit inquiry	Inquiry regarding the unit of program data
H'3F	New bit rate selection	Selection of new bit rate
H'40	Transition to programming/erasing state	Erasing of user MAT, and entry to program erasing state
H'4F	Boot program status inquiry	Inquiry into the operated status of the boot

response to the supported device inquiry.

Command | H'20

- Command, H'20, (one byte): Inquiry regarding supported devices

Response

H'30	Size	Number of devices	
Number of characters	Device code		Product name
...			
SUM			

- Response, H'30, (one byte): Response to the supported device inquiry
- Size (one byte): Number of bytes to be transmitted, excluding the command, size, an checksum, that is, the amount of data contributes by the number of devices, characte codes and product names
- Number of devices (one byte): The number of device types supported by the boot pr
- Number of characters (one byte): The number of characters in the device codes and program's name
- Device code (four bytes): ASCII code of the supporting product
- Product name (n bytes): Type name of the boot program in ASCII-coded characters
- SUM (one byte): Checksum

  The checksum is calculated so that the total number of all values from the command the SUM byte becomes H'00.

- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to the device selection command
  ACK will be returned when the device code matches.

Error response	H'90	ERROR

- Error response, H'90, (one byte): Error response to the device selection command
  ERROR : (one byte): Error code
  H'11: Sum check error
  H'21: Device code error, that is, the device code does not match

(c) Clock Mode Inquiry

The boot program will return the supported clock modes in response to the clock mode in

Command	H'21

- Command, H'21, (one byte): Inquiry regarding clock mode

Response	H'31	Size	Mode	···	SUM

- Response, H'31, (one byte): Response to the clock-mode inquiry
- Size (one byte): Amount of data that represents the modes
- Mode (one byte): Values of the supported clock modes (i.e. H'01 means clock mode 1
- SUM (one byte): Checksum

- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to the clock mode selection command
  ACK will be returned when the clock mode matches.

Error Response	H'91	ERROR

- Error response, H'91, (one byte): Error response to the clock mode selection comma
- ERROR : (one byte): Error code
  H'11: Checksum error
  H'22: Clock mode error, that is, the clock mode does not match.

Even if the clock mode numbers are H'00 and H'01 by a clock mode inquiry, the clock n
be selected using these respective values.

...						
SUM						

- Response, H'32, (one byte): Response to the multiplication ratio inquiry
- Size (one byte): The amount of data that represents the number of clock sources and multiplication ratios and the multiplication ratios
- Number of types (one byte): The number of supported multiplied clock types (e.g. when there are two multiplied clock types, which are the main and peripheral clo number of types will be H'02.)
- Number of multiplication ratios (one byte): The number of multiplication ratios for ea (e.g. the number of multiplication ratios to which the main clock can be set and the pe clock can be set.)
- Multiplication ratio (one byte)

  Multiplication ratio: The value of the multiplication ratio (e.g. when the clock-frequen multiplier is four, the value of multiplication ratio will be H'04.)

  Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the divided by two, the value of division ratio will be H'FE. H'FE = D'-2)

  The number of multiplication ratios returned is the same as the number of multiplicati and as many groups of data are returned as there are types.
- SUM (one byte): Checksum

operating clock frequency	frequency
...	
SUM	

- Response, H'33, (one byte): Response to operating clock frequency inquiry
- Size (one byte): The number of bytes that represents the minimum values, maximum and the number of frequencies.
- Number of operating clock frequencies (one byte): The number of supported operati frequency types
  (e.g. when there are two operating clock frequency types, which are the main and pe clocks, the number of types will be H'02.)
- Minimum value of operating clock frequency (two bytes): The minimum value of th multiplied or divided clock frequency.

  The minimum and maximum values of the operating clock frequency represent the v MHz, valid to the hundredths place of MHz, and multiplied by 100. (e.g. when the v 17.00 MHz, it will be 2000, which is H'07D0.)
- Maximum value (two bytes): Maximum value among the multiplied or divided clock frequencies.

  There are as many pairs of minimum and maximum values as there are operating clo frequencies.
- SUM (one byte): Checksum

- Response, H'35, (one byte): Response to the user MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start ac and area-last address
- Number of areas (one byte): The number of consecutive user MAT areas
  When the user MAT areas are consecutive, the number of areas is H'01.
- Area-start address (four bytes): Start address of the area
- Area-last address (four bytes): Last address of the area
  There are as many groups of data representing the start and last addresses as there are
- SUM (one byte): Checksum

(h) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses.

Command

H'26

- Command, H'26, (two bytes): Inquiry regarding erased block information

Response

H'36	Size	Number of blocks	
Block start address			Block last address
...			
SUM			

- Response, H'36, (one byte): Response to the number of erased blocks and addresses
- Size (three bytes): The number of bytes that represents the number of blocks, block-st addresses, and block-last addresses.
- Number of blocks (one byte): The number of erased blocks
- Block start address (four bytes): Start address of a block

Response	H'37	Size	Programming unit	SUM

- Response, H'37, (one byte): Response to programming unit inquiry
- Size (one byte): The number of bytes that indicate the programming unit, which is fi
- Programming unit (two bytes): A unit for programming
  This is the unit for reception of programming.
- SUM (one byte): Checksum

(j) New Bit-Rate Selection

The boot program will set a new bit rate and return the new bit rate.

This selection should be sent after sending the clock mode selection command.

Command	H'3F	Size	Bit rate	Input frequency
	Number of multiplication ratios	Multiplication ratio 1	Multiplication ratio 2	
	SUM			

- Command, H'3F, (one byte): Selection of new bit rate
- Size (one byte): The number of bytes that represents the bit rate, input frequency, nu multiplication ratios, and multiplication ratio
- Bit rate (two bytes): New bit rate
  One hundredth of the value (e.g. when the value is 19200 bps, it will be 192, which i
- Input frequency (two bytes): Frequency of the clock input to the boot program
  This is valid to the hundredths place and represents the value in MHz multiplied by when the value is 20.00 MHz, it will be 2000, which is H'07D0.)
- Number of multiplication ratios (one byte): The number of multiplication ratios to w device can be set.

divided by two, the value of division ratio will be H'FE. H'FE = D'-2)

- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to selection of a new bit rate
  When it is possible to set the bit rate, the response will be ACK.

Error Response	H'BF	ERROR

- Error response, H'BF, (one byte): Error response to selection of new bit rate
- ERROR: (one byte): Error code

H'11: Sum checking error
H'24: Bit-rate selection error
The rate is not available.
H'25: Error in input frequency
This input frequency is not within the specified range.
H'26: Multiplication-ratio error
The ratio does not match an available ratio.
H'27: Operating frequency error
The frequency is not within the specified range.

frequency error is generated.

3. Operating frequency error

   Operating frequency is calculated from the received value of the input frequency and multiplication or division ratio. The input frequency is input to the LSI and the LSI i at the operating frequency. The expression is given below.

   Operating frequency = Input frequency × Multiplication ratio, or
   Operating frequency = Input frequency ÷ Division ratio

   The calculated operating frequency should be checked to ensure that it is within the minimum to maximum frequencies which are available with the clock modes of the device. When it is out of this range, an operating frequency error is generated.

4. Bit rate

   To facilitate error checking, the value (n) of clock select (CKS) in the serial mode re (SMR), and the value (N) in the bit rate register (BRR), which are found from the pe operating clock frequency (ϕ) and bit rate (B), are used to calculate the error rate to e it is less than 4%. If the error is more than 4%, a bit rate error is generated. The error calculated using the following expression:

$$\text{Error (\%)} = \{[\frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{(2\times n - 1)}}] - 1\} \times 100$$

When the new bit rate is selectable, the rate will be set in the register after sending ACK response. The host will send an ACK with the new bit rate for confirmation and the boo will response with that rate.

Confirmation	H'06

- Confirmation, H'06, (one byte): Confirmation of a new bit rate

Response	H'06

- Response, H'06, (one byte): Response to confirmation of a new bit rate

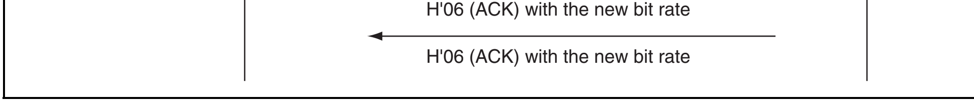


Figure 21.23 New Bit-Rate Selection Sequence

(5) Transition to Programming/Erasing State

The boot program will transfer the erasing program and erase the user MATs. On comple this erasure, ACK will be returned and the program will enter the programming/erasing s

The host should select the device code, clock mode, and new bit rate with device selectio mode selection, and new bit-rate selection commands, and then send the command for the transition to programming/erasing state. These procedures should be carried out before se the programming selection command or program data.

Command	H'40

- Command, H'40, (one byte): Transition to programming/erasing state

Response	H'06

- Response, H'06, (one byte): Response to transition to programming/erasing state
  The boot program will send ACK when the user MATs have been erased by the trans erasing program.

Error Response	H'C0	H'51

- Error code, H'51, (one byte): Erasing error
  An error occurred and erasure was not completed.

The order for commands in the inquiry selection state is shown below.

1. A supported device inquiry (H'20) should be made to inquire about the supported de
2. The device should be selected from among those described by the returned informati with a device-selection (H'10) command.
3. A clock-mode inquiry (H'21) should be made to inquire about the supported clock m
4. The clock mode should be selected from among those described by the returned info and set.
5. After selection of the device and clock mode, inquiries for other required informatio be made, such as the multiplication-ratio inquiry (H'22) or operating frequency inqui which are needed for a new bit-rate selection.
6. A new bit rate should be selected with the new bit-rate selection (H'3F) command, ac to the returned information on multiplication ratios and operating frequencies.
7. After selection of the device and clock mode, the information of the user MAT shoul to inquire about the user MATs information inquiry (H'25), erased block information (H'26), and programming unit inquiry (H'27).
8. After making inquiries and selecting a new bit rate, issue the transition to programming/erasing state command (H'40). The boot program will then enter the programming/erasing state.

H'50	128-byte programming	Programs 128 bytes of data
H'48	Erasing selection	Transfers the erasing program
H'58	Block erasing	Erases a block of data
H'52	Memory read	Reads the contents of memory
H'4B	User MAT sum check	Checks the checksum of the user MAT
H'4D	User MAT blank check	Checks the blank data of the user MAT
H'4F	Boot program status inquiry	Inquires into the boot program's status

- Programming

  Programming is executed by the programming selection and 128-byte programming commands.

  Firstly, the host should send the programming selection command

  After issuing the programming selection command, the host should send the 128-byte programming command. The 128-byte programming command that follows the select command represents the data programmed according to the method specified by the se command. When more than 128-byte data is programmed, 128-byte commands should repeatedly be executed. Sending a 128-byte programming command with H'FFFFFFF address will stop the programming. On completion of programming, the boot program wait for selection of programming or erasing.

  Where the sequence of programming operations that is executed includes programmin another method or of another MAT, the procedure must be repeated from the program selection command.

  The sequence for the programming selection and 128-byte programming commands is in figure 21.24.

Figure 21.24 Programming Sequence

- Erasure

  Erasure is executed by the erasure selection and block erasure commands.

  Firstly, erasure is selected by the erasure selection command and the boot program th
  the specified block. The command should be repeatedly executed if two or more blo
  be erased. Sending a block erasure command from the host with the block number H
  stop the erasure operating. On completion of erasing, the boot program will wait for
  of programming or erasing.

  The sequence for the erasure selection and block erasure commands is shown in figu

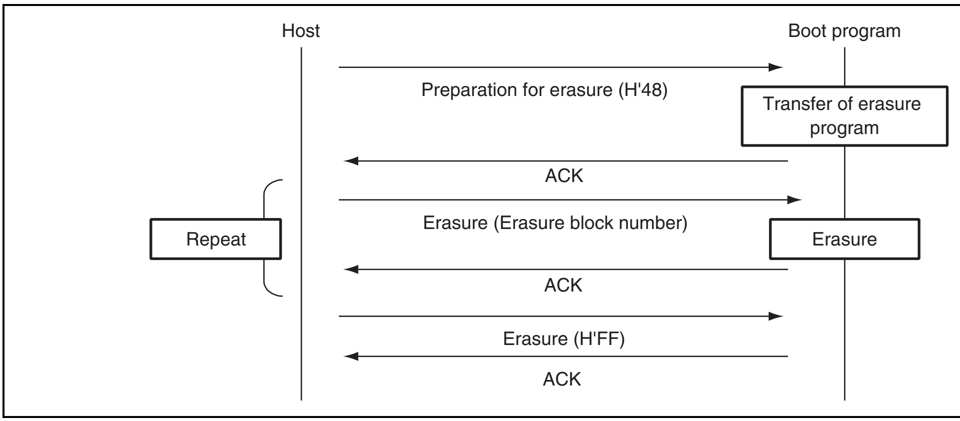


Figure 21.25 Erasure Sequence

Error Response	H'C3	ERROR

- Error response : H'C3 (1 byte): Error response to user-program programming selectio
- ERROR : (1 byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not complete

(b) 128-Byte Programming

The boot program will use the programming program transferred by the programming sel program the user MATs in response to 128-byte programming.

Command							
H'50	Address						
Data	···						
···							
SUM							

- Command, H'50, (one byte): 128-byte programming
- Programming Address (four bytes): Start address for programming
  Multiple of the size specified in response to the programming unit inquiry
  (i.e. H'00, H'01, H'00, H'00 : H'01000000)
- Program data (128 bytes): Data to be programmed
  The size is specified in the response to the programming unit inquiry.
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, the boot program will return ACK.

programming is in 128-byte units, the lower eight bits of the address should be H'00 or H
When there are less than 128 bytes of data to be programmed, the host should fill the res
H'FF.

Sending the 128-byte programming command with the address of H'FFFFFFFF will stop
programming operation. The boot program will interpret this as the end of the programm
wait for selection of programming or erasing.

Command

H'50	Address	SUM

- Command, H'50, (one byte): 128-byte programming
- Programming Address (four bytes): End code is H'FF, H'FF, H'FF, H'FF.
- SUM (one byte): Checksum

Response

H'06

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, the boot program will return ACK.

Error Response

H'D0	ERROR

- Error Response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code

  H'11: Checksum error
  H'53: Programming error
  An error has occurred in programming and programming cannot be co

Error Response	H'C8	ERROR

- ERROR: (one byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not complet

(d) Block Erasure

The boot program will erase the contents of the specified block.

Command	H'58	Size	Block number	SUM

- Command, H'58, (one byte): Erasure
- Size (one byte): The number of bytes that represents the erase block number
  This is fixed to 1.
- Block number (one byte): Number of the block to be erased
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to Erasure
  After erasure has been completed, the boot program will return ACK.

Error Response	H'D8	ERROR

- Error Response, H'D8, (one byte): Response to Erasure
- ERROR (one byte): Error code

  H'11: Sum check error
  H'29: Block number error
  Block number is incorrect.
  H'51: Erasure error
  An error has occurred during erasure.

Response	H'06

- Response, H'06, (one byte): Response to end of erasure (ACK)
  When erasure is to be performed after the block number H'FF has been sent, the proc
  should be executed from the erasure selection command.

(e) Memory Read

The boot program will return the data in the specified address.

Command	H'52	Size	Area	Read address
	Read size			SUM

- Command: H'52 (1 byte): Memory read
- Size (1 byte): Amount of data that represents the area, read address, and read size (fi
- Area (1 byte)
  H'01: User MAT
  An address error occurs when the area setting is incorrect.
- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response	H'52	Read size						
	Data	···						
	SUM							

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (1 byte): Checksum

The boot program will return the byte-by-byte total of the contents of the bytes of the use program.

Command	H'4B

- Command, H'4B, (one byte): Sum check for user program

Response	H'5B	Size	Checksum of user program	SUM

- Response, H'5B, (one byte): Response to the sum check of the user program
- Size (one byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (four bytes): Checksum of user MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

(g) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the resul

Command	H'4D

- Command, H'4D, (one byte): Blank check for user MATs

Response	H'06

- Response, H'06, (one byte): Response to the blank check for user MATs
  If the contents of all user MATs are blank (H'FF), the boot program will return ACK.

Error Response	H'CD	H'52

- Error Response, H'CD, (one byte): Error response to the blank check of user MATs.
- Error code, H'52, (one byte): Erasure has not been completed.

- Status (one byte): State of the boot program
- ERROR (one byte): Error status

  ERROR = 0 indicates normal operation.
  ERROR = 1 indicates error has occurred.

- SUM (one byte): Sum check

Table 21.16 Status Code

Code	Description
H'11	Device selection wait
H'12	Clock mode selection wait
H'13	Bit rate selection wait
H'1F	Programming/erasing state transition wait (bit rate selection is completed)
H'31	Programming state for erasure
H'3F	Programming/erasing selection wait (erasure is completed)
H'4F	Program data receive wait
H'5F	Erase block specification wait (erasure is completed)

H'26	Multiplication ratio error
H'27	Operating frequency error
H'29	Block number error
H'2A	Address error
H'2B	Data length error
H'51	Erasure error
H'52	Erasure incomplete error
H'53	Programming error
H'54	Selection processing error
H'80	Command error
H'FF	Bit-rate-adjustment confirmation error

3.3-V programming voltage. Use only the specified socket adapter.

5. Do not remove the chip from the PROM programmer nor input a reset signal during programming/erasing in which a high voltage is applied to the flash memory. Doing damage the flash memory permanently. If a reset is input accidentally, the reset mus released after the reset input period of at least 100μs.
6. The flash memory is not accessible until FKEY is cleared after programming/erasing the operating mode is changed and this LSI is restarted by a reset immediately after programming/erasing has finished, secure the reset input period (period of $\overline{RES}$ = 0) 100μs. Transition to the reset state during programming/erasing is inhibited. If a rese accidentally, the reset must be released after the reset input period of at least 100μs.
7. At powering on or off the Vcc power supply, fix the $\overline{RES}$ pin to low and set the flash to hardware protection state. This power on/off timing must also be satisfied at a pov power-on caused by a power failure and other factors.
8. In on-board programming mode or programmer mode, programming of the 128-byte programming-unit block must be performed only once. Perform programming in the where the programming-unit block is fully erased.
9. When the chip is to be reprogrammed with the programmer after execution of progra erasure in on-board programming mode, it is recommended that automatic programm performed after execution of automatic erasure.
10. To program the flash memory, the program data and program must be allocated to ac which are higher than those of the external interrupt vector table and H'FF must be w all the system reserved areas in the exception handling vector table.
11. The programming program that includes the initialization routine and the erasing pro includes the initialization routine are each 4 kbytes or less. Accordingly, when the C frequency is 35 MHz, the download for each program takes approximately 60 μs at t maximum.

after immediately setting it to 1. Otherwise, download cannot be performed normally.
Immediately after executing the instruction to set the SCO bit to 1, dummy read of the
must be executed twice.

15. The contents of some registers are not saved in a programming/erasing program. Whe
needed, save registers in the procedure program.

changes the frequency through the setting of the system clock control register (SCKCR) subclock control register (SUBCKCR).

This LSI supports five clocks: a system clock provided to the CPU and bus masters, a pe module clock provided to the peripheral modules, an external bus clock provided to the bus, a 32K timer clock, and a USB clock provided to the USB module. Frequencies of th peripheral module clock, the external bus clock, and the system clock can be set indepen although the peripheral module clock and the external bus clock operate with the freque than the system clock frequency.

The system clock, peripheral module clock, and external bus clock can be uniformly set 32.768 kHz subclock.

The USB module requires the 48-MHz clock. Set the external clock frequency and the M pin so that the USB clock (cku) frequency becomes 48 MHz.

Note that the MD_CLK pin setting also changes the frequencies of the peripheral modul the external bus clock, and the system clock.

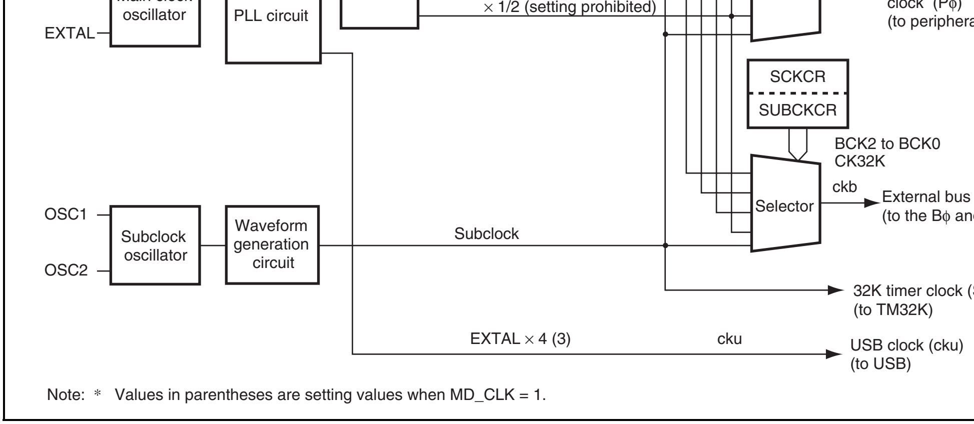


Figure 22.1 Block Diagram of Clock Pulse Generator

Table 22.1 Selection of Clock Pulse Generator

MD_CLK	EXTAL Input Clock Frequencies	Iϕ/Pϕ/Bϕ	USB Clock (cku)
0	8 MHz to 18 MHz	EXTAL ×4, ×2, ×1, ×1/2	EXTAL ×4
1	16 MHz	EXTAL ×2, ×1, ×1/2	EXTAL ×3

bus clocks.

Bit	15	14	13	12	11	10	9
Bit Name	PSTOP1	PSTOP0	—	—	—	ICK2	ICK1
Initial Value	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	—	PCK2	PCK1	PCK0	—	BCK2	BCK1
Initial Value	0	0	1	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PSTOP1	0	R/W	Bϕ Clock Output Enable Controls ϕ output on PA7. • Normal operation 0: ϕ output 1: Fixed high
14	PSTOP0	0	R/W	ϕ Clock Output Enable Controls ϕ output (SDϕ) on PB7. • Normal operation 0: ϕ output 1: Fixed high

				000: × 4 × 2 001: × 2 × 1 010: × 1 × 1/2 011: × 1/2 Setting prohibite 1XX: Setting prohibited The frequencies of the peripheral module clock external bus clock change to the same frequen system clock if the frequency of the system clo lower than that of the two clocks.
7	—	0	R/W	Reserved Although this bit is readable/writable, only 0 sh written to.
6 5 4	PCK2 PCK1 PCK0	0 1 0	R/W R/W R/W	Peripheral Module Clock (Pϕ) Select These bits select the frequency of the peripher module clock. The ratio to the input clock is as PCK (2:0) MD_CLK = 0 MD_CLK = 1 000: × 4 × 2 001: × 2 × 1 010: × 1 × 1/2 011: × 1/2 Setting prohibite 1XX: Setting prohibited The frequency of the peripheral module clock s lower than that of the system clock. Though the can be set so as to make the frequency of the peripheral module clock higher than that of the clock, the clocks will have the same frequency

001:	× 2	× 1
010:	× 1	× 1/2
011:	× 1/2	Setting prohibite
1XX:	Setting prohibited	

The frequency of the external bus clock shoul than that of the system clock. Though these b set so as to make the frequency of the extern clock higher than that of the system clock, the will have the same frequency in reality.

Note: X: Don't care

22.1.2 Subclock Control Register (SUBCKCR)

SUBCKCR stops the main clock oscillator, selects the operating clock of the system clo selects the operating clock after a transition from software standby mode.

Bit	7	6	5	4	3	2	1
Bit Name	—	—	—	—	—	EXSTP	WAKE32K
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

				standby mode. 1: The main clock oscillator and PLL are stop during subclock operation.
1	WAKE32K	0	R/W	Wakeup Clock Select Selects the operating clock for use as the syste after the transition from the subclock operation software standby mode has been initiated by a interrupt. 0: On leaving software standby mode, the ma is the operating clock. 1: On leaving software standby mode, the su the operating clock. This setting is valid wh (CK32K) is set to 1.
0	CK32K	0	R/W	Subclock Select 0: The system clock (Iϕ), peripheral module c (Pϕ), and external bus clock (Bϕ) operate c main clock. 1: The system clock (Iϕ), peripheral module c (Pϕ), and external bus clock (Bϕ) operate c subclock. When the OSC32STP bit in TCR32K is 1, 1 ca written to this bit. This bit is cleared to 0 when c software standby mode while the value of WAK is 0. Dummy read of this bit must be performed immediately after this bit is written to.

16 MHz.

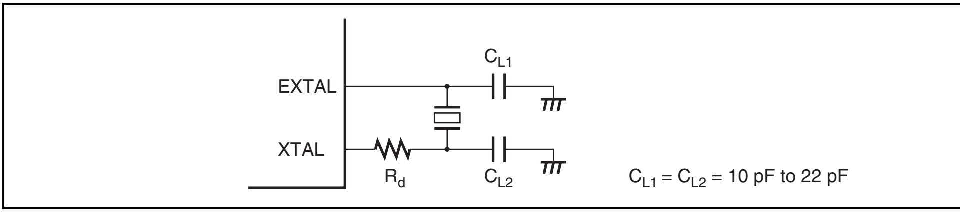


Figure 22.2 Connection of Crystal Resonator (Example)

Table 22.2 Damping Resistance Value

Frequency (MHz)	8	12	16	18
R_d (Ω)	200	0	0	0

Table 22.3 Crystal Resonator Characteristics

Frequency (MHz)	8	12	16	18
R_S Max. (Ω)	80	60	50	40
C_0 Max. (pF)			7	

22.2.2 External Clock Input

An external clock signal can be input as the examples in Figure 22.4. When the XTAL pi open, make the parasitic capacitance less than 10 pF. When the counter clock is input to t pin, put the external clock in high level during standby mode.

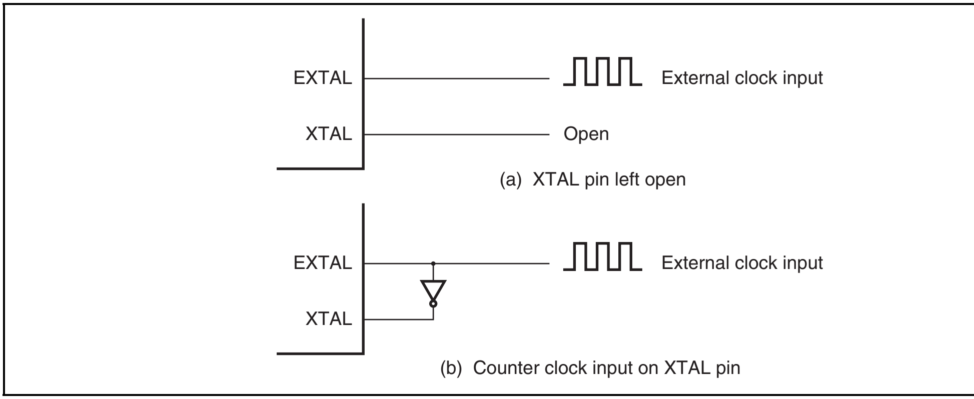


Figure 22.4 External Clock Input (Examples)

The PLL circuit has the function of multiplying the frequency of the clock from the osci
factor of 4. The frequency multiplication rate is fixed. The phase difference is controlled
the timing of the rising edge of the internal clock is the same as that of the EXTAL pin s

22.4 Frequency Divider

The frequency divider divides the PLL clock to generate a 1/2, 1/4, or 1/8 clock. After th
ICK2 to ICK0, PCK 2 to PCK0, and BCK2 to BCK0 are updated, this LSI operates with
updated frequency.

22.5 Subclock Oscillator

22.5.1 Connecting 32.768 kHz Crystal Resonator

To supply a clock to the subclock oscillator, connect a 32.768-kHz crystal resonator, as
figure 22.6. The usage notes given in section 22.6.3, Notes on Board Design, apply to th
connection of this crystal resonator.

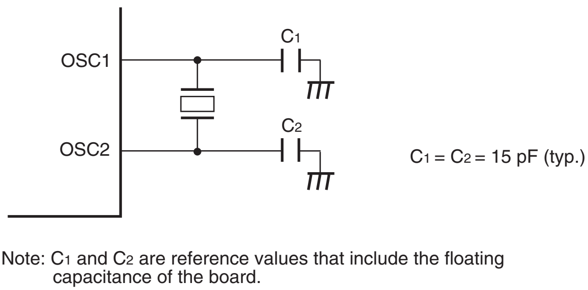


Figure 22.6 Connection Example of 32.768-kHz Crystal Resonator

Figure 22.7 Equivalent Circuit for 32.768-kHz Crystal Resonator

22.5.2 Handling of Pins when the Subclock is Not to be Used

If the subclock is not required, connect the OSC1 pin to Vss and leave the OSC2 pin open, as shown in figure 22.8.

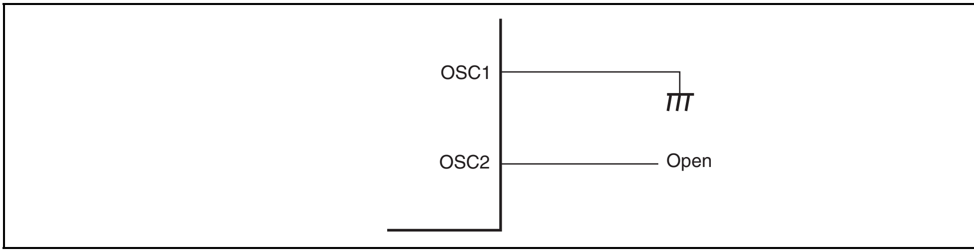


Figure 22.8 Pin Handling when Subclock is not Used

MHz, and 8 MHz ≤ Bϕ ≤ 50 MHz.

2. All the on-chip peripheral modules (except for the DMAC and DTC) operate on the therefore that the time processing of modules such as a timer and SCI differs before changing the clock division ratio.

   In addition, wait time for clearing software standby mode differs by changing the clo division ratio. For details, see section 23.7.3, Setting Oscillation Settling Time after Software Standby Mode.

3. The relationship among the system clock, peripheral module clock, and external bus ≥ Pϕ and Iϕ ≥ Bϕ. In addition, the system clock setting has the highest priority. Acco Pϕ or Bϕ may have the frequency set by bits ICK2 to ICK0 regardless of the settings PCK2 to PCK0 or BCK2 to BCK0.

4. Note that the frequency of ϕ will be changed in the middle of a bus cycle when settin while executing the external bus cycle with the write-data-buffer function.

5. Figure 22.9 shows the clock modification timing. After a value is written to SCKCR waits for the current bus cycle to complete. After the current bus cycle completes, ea frequency will be modified within one cycle (worst case) of the external input clock

Figure 22.9 Clock Modification Timing

22.6.2 Notes on Resonator

Since various characteristics related to the resonator are closely linked to the user's board
thorough evaluation is necessary on the user's part, using the resonator connection examp
shown in this section as a reference. As the parameters for the resonator will depend on th
floating capacitance of the resonator and the mounting circuit, the parameters should be
determined in consultation with the resonator manufacturer. The design must ensure that
exceeding the maximum rating is not applied to the resonator pin.

22.6.3 Notes on Board Design

When using the crystal resonator, place the crystal resonator and its load capacitors as clo
XTAL and EXTAL pins as possible. Other signal lines should be routed away from the o
circuit as shown in Figure 22.10 to prevent induction from interfering with correct oscilla

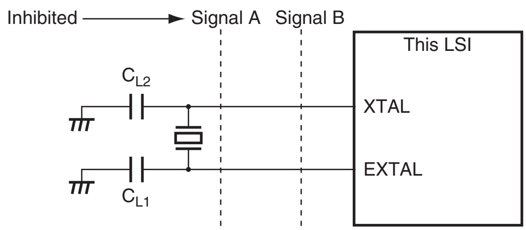


Figure 22.10 Note on Board Design for Oscillation Circuit

Note: * CB and CPB are laminated ceramic capacitors.

Figure 22.11 Recommended External Circuitry for PLL Circuit

- The system clock, peripheral module clock, and external bus clock can be uniformly
  32.768 kHz subclock.
- Module stop function

  The functions for each peripheral modules can be stopped to make a transition to a p
  down mode.
- Transition function to power-down mode

  Transition to a power-down mode is possible to stop the CPU, peripheral modules, a
  oscillator.
- Four power-down modes

  Sleep mode

  All-module-clock-stop mode

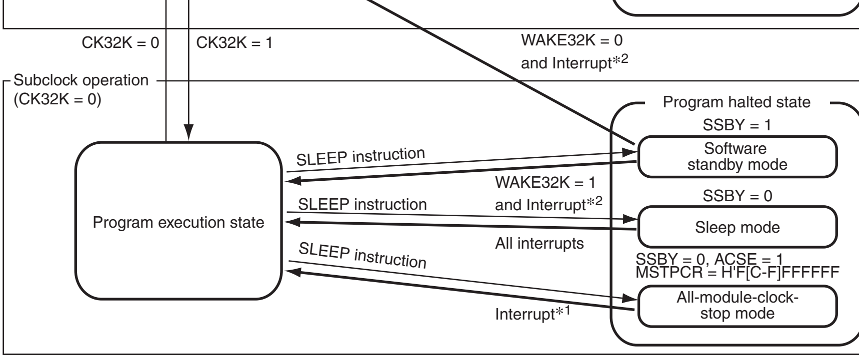
  Software standby mode

  Hardware standby mode

method				
Oscillator	Functioning	Functioning	Halted	Halted
Subclock oscillator	Functioning*[6]	Functioning*[6]	Functioning*[6]	Halted
CPU	Halted (retained)	Halted (retained)	Halted (retained)	Halted
Watchdog timer	Functioning	Functioning	Halted (retained)	Halted
8-bit timer	Functioning	Functioning*[4]	Halted (retained)	Halted
32K timer	Functioning	Functioning	Functioning	Halted
Peripheral modules	Functioning	Halted*[1]	Halted*[1]	Halted*
I/O port	Functioning	Retained	Retained	Hi-Z

Notes: "Halted (retained)" in the table means that the internal register values are retained
internal operations are suspended.

1. SCI enters the reset state, and other peripheral modules retain their states.
2. External interrupt and some internal interrupts (8-bit timer, watchdog timer, anc timer)
3. All peripheral modules enter the reset state.
4. "Functioning" or "Halted" is selectable through the setting of bits MSTPA9 and in MSTPCRA. However, pin output is disabled even when "Functioning" is sele
5. External interrupt and 32K timer interrupt
6. "Functioning" or "Halted" is selectable through the setting of bit OSC32STP in



[Legend] ⟶ Transition after exception handling

Notes: 1. NMI, IRQ0 to IRQ11, 8-bit timer interrupts, watchdog timer interrupts, and 32K timer interrupts. Note that the 8-bit timer interrupt is valid when the MSTPCRA9 or MSTPCRA8 bit is cleared to 0
2. NMI, IRQ0 to IRQ11, and 32K timer interrupts. Note that IRQ is valid only when the corresponding bit in SSIER is set to 1.
3. The SLPIE bit is 0.

Notes:
- From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ is driven low.
- From any state except hardware syandby mode, a transition to the reset state occurs when $\overline{RES}$ is driven low.

Figure 23.1 Mode Transitions

23.2 Register Descriptions

The registers related to the power-down modes are shown below. For details on the syst control register (SCKCR), refer to section 22.1.1, System Clock Control Register (SCK

- Standby control register (SBYCR)
- Module stop control register A (MSTPCRA)
- Module stop control register B (MSTPCRB)
- Module stop control register C (MSTPCRC)

Bit Name	SLPIE	—	—	—	—	—	—
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	SSBY	0	R/W	Software Standby Specifies the transition mode after executing the instruction 0: Shifts to sleep mode after the SLEEP instructi executed 1: Shifts to software standby mode after the SLE instruction is executed This bit does not change when clearing the softw standby mode by using external interrupts and s normal operation. For clearing, write 0 to this bit. the WDT is used as the watchdog timer, the sett bit is disabled. In this case, a transition is always sleep mode or all-module-clock-stop mode after SLEEP instruction is executed. When the SLPIE to 1, this bit should be cleared to 0.
14	OPE	1	R/W	Output Port Enable Specifies whether the output of the address bus control signals ($\overline{CS0}$ to $\overline{CS7}$, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, and retained or set to the high-impedance state in so standby mode. 0: In software standby mode, address bus and b control signals are high-impedance 1: In software standby mode, address bus and b control signals retain output state

according to the operating frequency so that the
time is at least equal to the oscillation settling ti
an external clock, a PLL circuit settling time is n
Refer to table 23.2 to set the standby time.

While oscillation is being settled, the timer is co
the Pϕ clock frequency. Careful consideration is
in multi-clock mode.

00000: Reserved

00001: Reserved

00010: Reserved

00011: Reserved

00100: Reserved

00101: Standby time = 64 states

00110: Standby time = 512 states

00111: Standby time = 1024 states

01000: Standby time = 2048 states

01001: Standby time = 4096 states

01010: Standby time = 16384 states

01011: Standby time = 32768 states

01100: Standby time = 65536 states

01101: Standby time = 131072 states

01110: Standby time = 262144 states

01111: Standby time = 524288 states

1XXXX: Reserved

				set to 1. Writing 0 clears this bit.
6 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.

[Legend] X: Don't care

Note: With the F-ZTAT version, the flash memory settling time must be reserved.

23.2.2 Module Stop Control Registers A and B (MSTPCRA and MSTPCRB)

MSTPCRA and MSTPCRB set the module stop function. Setting a bit to 1 makes the corresponding module enter the module stop state, while clearing the bit to 0 clears the m stop state.

- MSTPCRA

Bit	15	14	13	12	11	10	9	
Bit Name	ACSE	MSTPA14	MSTPA13	MSTPA12	MSTPA11	MSTPA10	MSTPA9	M
Initial Value	0	0	0	0	1	1	1	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	7	6	5	4	3	2	1	
Bit Name	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	M
Initial Value	1	1	1	1	1	1	1	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

• MSTPCRA

Bit	Bit Name	Initial Value	R/W	Module
15	ACSE	0	R/W	All-Module-Clock-Stop Mode Enable Enables/disables all-module-clock-stop mode fo reducing current consumption by stopping the b controller and I/O ports operations when the CP executes the SLEEP instruction after the modu state has been set for all the on-chip peripheral controlled by MSTPCR. 0: All-module-clock-stop mode disabled 1: All-module-clock-stop mode enabled
14	MSTPA14	0	R/W	Reserved
13	MSTPA13	0	R/W	DMA controller (DMAC)
12	MSTPA12	0	R/W	Data transfer controller (DTC)
11 10	MSTPA11 MSTPA10	1 1	R/W R/W	Reserved These bits are always read as 1. The write valu always be 1.
9	MSTPA9	1	R/W	8-bit timer (TMR_3 and TMR_2)
8	MSTPA8	1	R/W	8-bit timer (TMR_1 and TMR_0)
7 6	MSTPA7 MSTPA6	1 1	R/W R/W	Reserved These bits are always read as 1. The write valu always be 1.

0	MSTPA0	1	R/W	16-bit timer pulse unit (TPU channels 5 to 0)

- MSTPCRB

Bit	Bit Name	Initial Value	R/W	Module
15	MSTPB15	1	R/W	Programmable pulse generator (PPG)
14	MSTPB14	1	R/W	Reserved
13	MSTPB13	1	R/W	These bits are always read as 1. The write value always be 1.
12	MSTPB12	1	R/W	Serial communication interface_4 (SCI_4)
11	MSTPB11	1	R/W	Reserved This bit is always read as 1. The write value shou always be 1.
10	MSTPB10	1	R/W	Serial communication interface_2 (SCI_2)
9	MSTPB9	1	R/W	Serial communication interface_1 (SCI_1)
8	MSTPB8	1	R/W	Serial communication interface_0 (SCI_0)
7	MSTPB7	1	R/W	I^2C bus Interface 1 (IIC_1)
6	MSTPB6	1	R/W	I^2C bus Interface 0 (IIC_0)
5	MSTPB5	1	R/W	Reserved
4	MSTPB4	1	R/W	These bits are always read as 1. The write value always be 1.
3	MSTPB3	1	R/W	
2	MSTPB2	1	R/W	
1	MSTPB1	1	R/W	
0	MSTPB0	1	R/W	

Bit	7	6	5	4	3	2	1
Bit Name	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Module
15	MSTPC15	1	R/W	Serial communication interface_5 (SCI_5), (IrD
14	MSTPC14	1	R/W	Serial communication interface_6 (SCI_6)
13	MSTPC13	1	R/W	8-bit timer (TMR_4, TMR_5)
12	MSTPC12	1	R/W	8-bit timer (TMR_6, TMR_7)
11	MSTPC11	1	R/W	Universal serial bus interface (USB)
10	MSTPC10	1	R/W	Cyclic redundancy check
9	MSTPC9	1	R/W	Reserved
8	MSTPC8	1	R/W	These bits are always read as 1. The write valu always be 1.
7	MSTPC7	0	R/W	Reserved
6	MSTPC6	0	R/W	These bits are always read as 0. The write valu always be 0.
5	MSTPC5	0	R/W	
4	MSTPC4	0	R/W	On-chip RAM_4 (H'FF2000 to H'FF3FFF)
3	MSTPC3	0	R/W	On-chip RAM_3 (H'FF4000 to H'FF5FFF)
2	MSTPC2	0	R/W	On-chip RAM_2 (H'FF6000 to H'FF7FFF)
1	MSTPC1	0	R/W	On-chip RAM_1 (H'FF8000 to H'FF9FFF)
0	MSTPC0	0	R/W	On-chip RAM_0 (H'FFA000 to H'FFBFFF)

reflected in the peripheral module and external bus clocks. The peripheral module and ex
clocks are restricted to the operating clock specified by bits ICK2 to ICK0.

23.3.2 Switching to Subclock

When the CK32K bit in SUBCKCR is set to 1, a transition from the main clock operation subclock operation is made at the end of the bus cycle regardless of the SCKCR setting. I subclock operation, the CPU, bus masters, peripheral modules, and all external buses ope the 32.768-kHz subclock.

When the CK32K bit in SUBCKCR is set to 0 in the subclock operation, a transition to th clock operation is made at the end of the bus cycle. Since a transition from the subclock o to the main clock operation is made via software standby mode, the oscillation settling tin main clock must elapse. Set the oscillation settling time of the main clock with bits STS4 in SBYCR.

The main clock oscillator can be operated or stopped by the EXSTP bit in SUBCKCR in subclock operation. When a transition is made from the subclock operation to the main cl operation with the main clock oscillator operating, the wait for the oscillation settling tim main clock oscillator is not necessary. A transition to the main clock operation can be ma minimum setting time with the setting of bits STS4 to STS0 in SBYCR.

In the same way as in the main clock operation, if a SLEEP instruction is executed in the operation while the SSBY bit in SBYCR is set to 1, this LSI enters software standby mod a transition is made to software standby mode in the subclock operation, the operating clo system clock after clearing of software standby mode can be selected with the WAKE32K SUBCKCR. This LSI is placed in the subclock operation if the WAKE32K bit is 1, or pla the main clock operation if the WAKE32K bit is 0.

After the reset state is cleared, all modules other than the DMAC, DTC, and on-chip RA
the module stop state.

The registers of the module for which the module stop state is selected cannot be read fr
written to.

23.5 Sleep Mode

23.5.1 Transition to Sleep Mode

When the SLEEP instruction is executed when the SSBY bit in SBYCR is 0, the CPU en
mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registe
retained. Other peripheral functions do not stop.

23.5.2 Clearing Sleep Mode

Sleep mode is exited by any interrupt, signals on the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin, and a reset caus
watchdog timer overflow.

1. Clearing by interrupt

   When an interrupt occurs, sleep mode is exited and interrupt exception processing st
   mode is not exited if the interrupt is disabled, or interrupts other than NMI are maske
   CPU.
2. Clearing by $\overline{\text{RES}}$ pin

   Setting the $\overline{\text{RES}}$ pin level low selects the reset state. After the stipulated reset input d
   driving the $\overline{\text{RES}}$ pin high makes the CPU start the reset exception processing.

bit in SBYCR cleared to 0 will cause all modules (except for the 8-bit timer*, watchdog t
32K timer), the bus controller, and the I/O ports to stop operating, and to make a transitio
module-clock-stop mode at the end of the bus cycle.

When further reduction in power consumption is necessary in all-module-clock-stop mod
the modules controlled by MSTPCRC (MSTPCRC[15 to 8] = H'FFFF).

All-module-clock-stop mode is cleared by an external interrupt (NMI or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ11}}$ p
$\overline{\text{RES}}$ pin, or an internal interrupt (8-bit timer*, watchdog timer, or 32K timer), and the CP
returns to the normal program execution state via the exception handling state. All-modul
stop mode is not cleared if interrupts are disabled, if interrupts other than NMI are maske
CPU side, or if the relevant interrupt is designated as a DTC activation source.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Note: * Operation or halting of the 8-bit timer can be selected by bits MSTPA9 and M
in MSTPCRA.

consumption to be significantly reduced.

If the WDT is used as a watchdog timer, it is impossible to make a transition to software
mode. The WDT should be stopped before the SLEEP instruction execution.

23.7.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to IR
internal interrupt (32K timer) or by means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

1. Clearing by interrupt

   When an NMI or IRQ0 to IRQ11* interrupt request signal is input, clock oscillation
   after the elapse of the time set in bits STS4 to STS0 in SBYCR, stable clocks are sup
   the entire LSI, software standby mode is cleared, and interrupt exception handling is

   When clearing software standby mode with an IRQ0 to IRQ11* interrupt, set the
   corresponding enable bit to 1 and ensure that no interrupt with a higher priority than
   IRQ0 to IRQ11* is generated. Software standby mode cannot be cleared if the interr
   been masked on the CPU side or has been designated as a DTC activation source.

   Note: * By setting the SSIn bit in SSIER to 1, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ11}}$ can be used as a sof
   standby mode clearing source.

2. Clearing by $\overline{\text{RES}}$ pin

   When the $\overline{\text{RES}}$ pin is driven low, clock oscillation is started. At the same time as clo
   oscillation starts, clocks are supplied to the entire LSI. Note that the $\overline{\text{RES}}$ pin must b
   until clock oscillation settles. When the $\overline{\text{RES}}$ pin goes high, the CPU begins reset exc
   handling.

3. Clearing by $\overline{\text{STBY}}$ pin

   When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Table 23.2 Oscillation Settling Time Settings

STS4	STS3	STS2	STS1	STS0	Standby Time	Pφ* [MHz] 35	25	20
0	0	0	0	0	Reserved	—	—	—
				1	Reserved	—	—	—
			1	0	Reserved	—	—	—
				1	Reserved	—	—	—
		1	0	0	Reserved	—	—	—
				1	64	1.8	2.6	3.2
			1	0	512	14.6	20.5	25.6
				1	1024	29.3	41.0	51.2
	1	0	0	0	2048	58.5	81.9	102.4
				1	4096	0.12	0.16	0.20
			1	0	16384	0.47	0.66	0.82
				1	32768	0.94	1.31	1.64
		1	0	0	65536	1.87	2.62	3.28
				1	131072	3.74	5.24	6.55
			1	0	262144	7.49	10.49	13.11
				1	524288	14.98	20.97	26.21
1	0	0	0	0	Reserved	—	—	—

[light shading] : Recommended time setting when using an external clock.

[dark shading] : Recommended time setting when using a crystal resonator.

Note: * Pφ is the output from the peripheral module frequency divider.

				1	1024	78.8	102.4	128.0
	1	0	0	0	2048	157.5	204.8	256.0
				1	4096	0.32	0.41	0.51
			1	0	16384	1.26	1.64	2.05
				1	32768	2.52	3.28	4.10
		1	0	0	65536	5.04	6.55	8.19
				1	131072	10.08	13.11	16.38
			1	0	262144	20.16	26.21	32.77
				1	524288	40.33	52.43	65.54
1	0	0	0	0	Reserved	—	—	—

[light shading]: Recommended time setting when using an external clock.

[dark shading]: Recommended time setting when using a crystal resonator.

Note: * ϕ is the output from the peripheral module frequency divider.

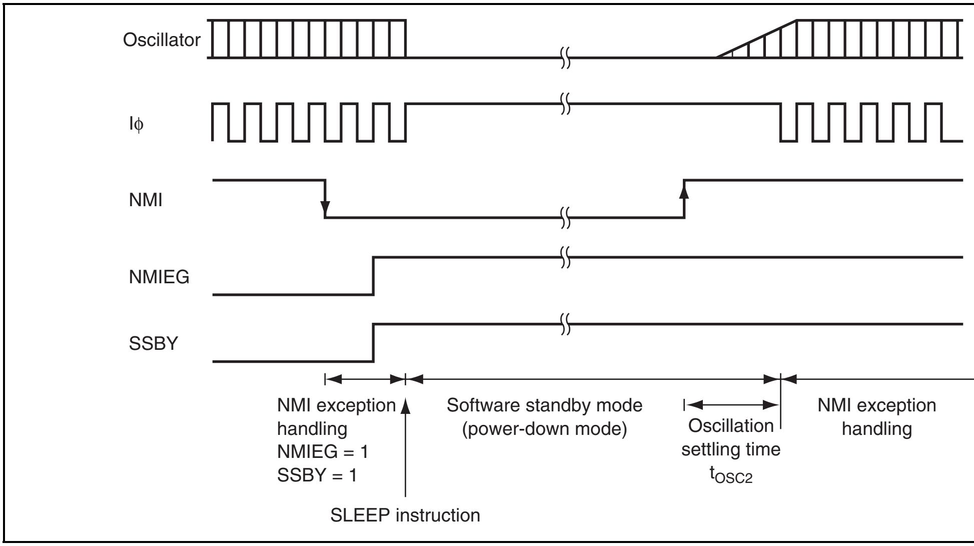


Figure 23.2 Software Standby Mode Application Example

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 be
driving the STBY pin low. Do not change the state of the mode pins (MD2 to MD0) wh
LSI is in hardware standby mode.

23.8.2 Clearing Hardware Standby Mode

Hardware standby mode is cleared by means of the STBY pin and the RES pin. When th
pin is driven high while the RES pin is low, the reset state is entered and clock oscillatio
started. Ensure that the RES pin is held low until clock oscillation settles (for details on
oscillation settling time, refer to table 23.2). When the RES pin is subsequently driven h
transition is made to the program execution state via the reset exception handling state.

23.8.3 Hardware Standby Mode Timing

Figure 23.3 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
hardware standby mode. Hardware standby mode is cleared by driving the STBY pin hi
waiting for the oscillation settling time, then changing the RES pin from low to high.

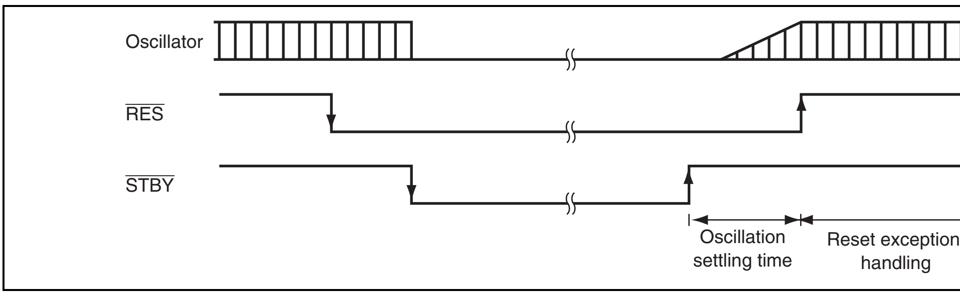


Figure 23.3 Hardware Standby Mode Timing

Timing.

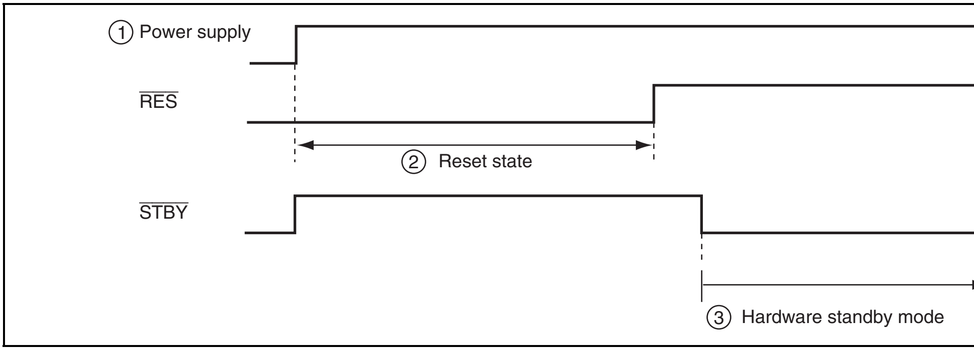


Figure 23.4 Timing Sequence at Power-On

Transitions to the power-down state are inhibited when sleep instruction exception hand initiated, and the CPU immediately starts sleep instruction exception handling.

When a SLEEP instruction is executed while the SLPIE bit is cleared to 0, a transition is the power-down state. The power-down state is canceled by a canceling factor interrupt 23.5).

When a canceling factor interrupt is generated immediately before the execution of a SL instruction, exception handling for the interrupt starts. When execution returns from the handling routine, the SLEEP instruction is executed to enter the power-down state. In th the power-down state is not canceled until the next canceling factor interrupt is generate figure 23.6).

When the SLPIE bit is set to 1 in the handling routine for a canceling factor interrupt so execution of a SLEEP instruction will produce sleep instruction exception handling, the of the system is as shown in figure 23.7. Even if a canceling factor interrupt is generated immediately before the SLEEP instruction is executed, sleep instruction exception hand initiated by execution of the SLEEP instruction. Therefore, the CPU executes the instruc follows the SLEEP instruction after sleep instruction exception and exception service ro without shifting to the power-down state.

When the SLPIE bit is set to 1 to start sleep exception handling, clear the SSBY bit in S to 0.

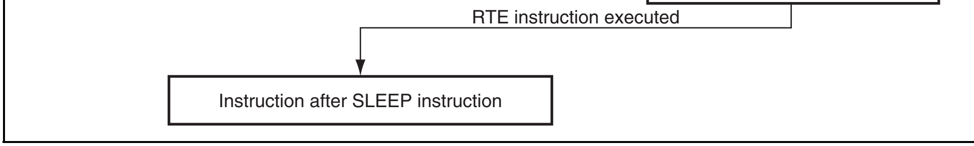


Figure 23.5 When Canceling Factor Interrupt is Generated after SLEEP Instruction Execution

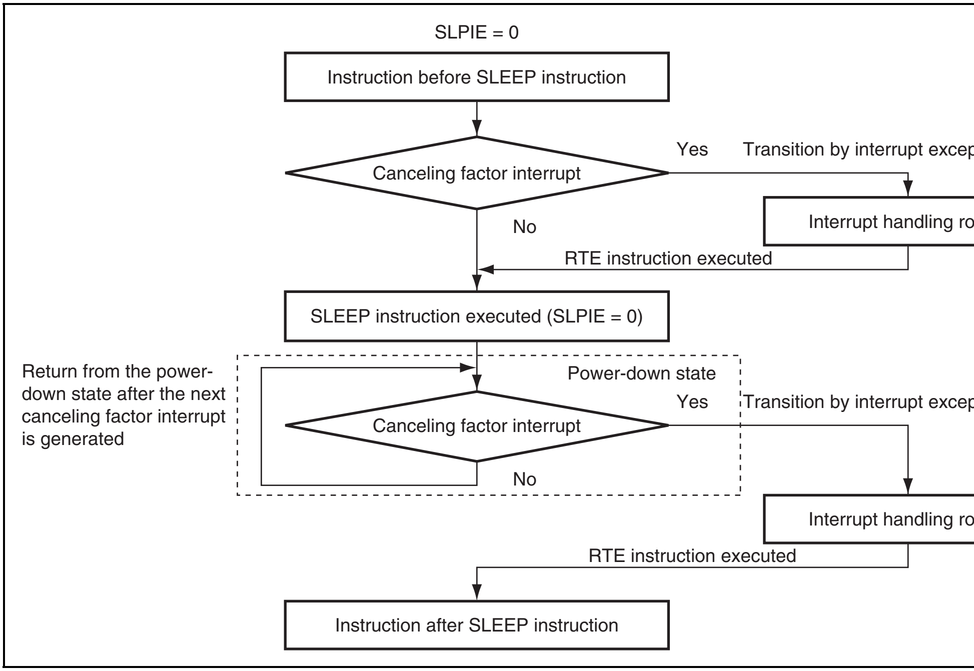


Figure 23.6 When Canceling Factor Interrupt is Generated Immediately befo SLEEP Instruction Execution (Sleep Instruction Exception Handling Not Initia

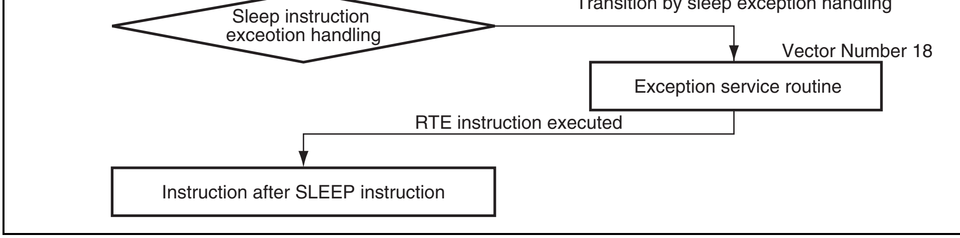


Figure 23.7 When Canceling Factor Interrupt is Generated Immediately be SLEEP Instruction Execution (Sleep Instruction Exception Handling Initiat

Table 23.3 φ Pin (PA7) State in Each Processing State

Register Setting Value		Normal Operating State	Sleep Mode	All-Module-Clock-Stop Mode	Software Standby Mode	
DDR	PSTOP1				OPE = 0	OPE = 1
0	X	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
1	0	Bφ output	Bφ output	Bφ output	High	High
1	1	High	High	High	High	High

Table 23.4 φ Pin (PB7) State in Each Processing State (SDRAM Interface Enabled)

Register Setting Value	Normal Operating State	Sleep Mode	All-Module-Clock-Stop Mode	Software Standby Mode	
PSTOP0				OPE = 0	OPE = 1
0	SDφ output	SDφ output	SDφ output	High	High
1	High	High	High	High	High

23.11.3 Module Stop Mode of DMAC or DTC

Depending on the operating state of the DMAC and DTC, bits MSTPA13 and MSTPA1
be set to 1, respectively. The module stop state setting for the DMAC or DTC should be
performed only when the DMAC or DTC is not activated.

For details, refer to section 7, DMA Controller (DMAC), and section 8, Data Transfer C
(DTC).

23.11.4 On-Chip Peripheral Module Interrupts

Relevant interrupt operations cannot be performed in the module stop state. Consequent
module stop state is entered when an interrupt has been requested, it will not be possible
the CPU interrupt source or the DMAC or DTC activation source. Interrupts should ther
disabled before entering the module stop state.

23.11.5 Writing to MSTPCRA, MSTPCRB, and MSTPCRC

MSTPCRA, MSTPCRB, and MSTPCRC should only be written to by the CPU.

clock. For details, refer to section 6.5.4, External Bus Interface.

- Among the internal I/O register area, addresses not listed in the list of registers are u or reserved addresses. Undefined and reserved addresses cannot be accessed. Do not these addresses; otherwise, the operation when accessing these bits and subsequent o cannot be guaranteed.

2. Register bits

- Bit configurations of the registers are listed in the same order as the register addresse
- Reserved bits are indicated by — in the bit name column.
- Space in the bit name field indicates that the entire register is allocated to either the c data.
- For the registers of 16 or 32 bits, the MSB is listed first.
- Byte configuration description order is subject to big endian.

3. Register states in each operating mode

- Register states are listed in the same order as the register addresses.
- For the initialized state of each bit, refer to the register description in the correspondi section.
- The register states shown here are for the basic operating modes. If there is a specific an on-chip peripheral module, refer to the section on that on-chip peripheral module.

Time constant registerB_4	TCORB_4	8	H'FEA46	TMR_4	16	3P
Time constant registerB_5	TCORB_5	8	H'FEA47	TMR_5	16	3P
Timer counter_4	TCNT_4	8	H'FEA48	TMR_4	16	3P
Timer counter_5	TCNT_5	8	H'FEA49	TMR_5	16	3P
Timer counter control register_4	TCCR_4	8	H'FEA4A	TMR_4	16	3P
Timer counter control register_5	TCCR_5	8	H'FEA4B	TMR_5	16	3P
CRC control register	CRCCR	8	H'FEA4C	CRC	16	3P
CRC data input register	CRCDIR	8	H'FEA4D	CRC	16	3P
CRC data output register	CRCDOR	16	H'FEA4E	CRC	16	3P
Timer control register_6	TCR_6	8	H'FEA50	TMR_6	16	3P
Timer control register_7	TCR_7	8	H'FEA51	TMR_7	16	3P
Timer control/status register_6	TCSR_6	8	H'FEA52	TMR_6	16	3P
Timer control/status register_7	TCSR_7	8	H'FEA53	TMR_7	16	3P
Time constant registerA_6	TCORA_6	8	H'FEA54	TMR_6	16	3P
Time constant registerA_7	TCORA_7	8	H'FEA55	TMR_7	16	3P
Time constant registerB_6	TCORB_6	8	H'FEA56	TMR_6	16	3P
Time constant registerB_7	TCORB_7	8	H'FEA57	TMR_7	16	3P
Timer counter_6	TCNT_6	8	H'FEA58	TMR_6	16	3P
Timer counter_7	TCNT_7	8	H'FEA59	TMR_7	16	3P
Timer counter control register_6	TCCR_6	8	H'FEA5A	TMR_6	16	3P
Timer counter control register_7	TCCR_7	8	H'FEA5B	TMR_7	16	3P
Interrupt flag register 0	IFR0	8	H'FEE00	USB	8	3P
Interrupt flag register 1	IFR1	8	H'FEE01	USB	8	3P
Interrupt flag register 2	IFR2	8	H'FEE02	USB	8	3P
Interrupt enable register 0	IER0	8	H'FEE04	USB	8	3P
Interrupt enable register 1	IER1	8	H'FEE05	USB	8	3P
Interrupt enable register 2	IER2	8	H'FEE06	USB	8	3P

EP3 data register	EPDR3	8	H'FEE18	USB	8	3
EP0o receive data size register	EPSZ0o	8	H'FEE24	USB	8	3
EP1 receive data size register	EPSZ1	8	H'FEE25	USB	8	3
Data status register	DASTS	8	H'FEE27	USB	8	3
FIFO clear register	FCLR	8	H'FEE28	USB	8	3
Endpoint stall register	EPSTL	8	H'FEE2A	USB	8	3
Trigger register	TRG	8	H'FEE2C	USB	8	3
DMA transfer setting register	DMA	8	H'FEE2D	USB	8	3
Configuration value register	CVR	8	H'FEE2E	USB	8	3
Control register	CTLR	8	H'FEE2F	USB	8	3
Endpoint information register	EPIR	8	H'FEE32	USB	8	3
Transceiver testregister0	TRNTREG00	8	H'FEE44	USB	8	3
Transceiver testregister1	TRNTREG1	8	H'FEE45	USB	8	3
Port M data direction register	PMDDR	8	H'FEE50	I/O port	8	3
Port M data register	PMDR	8	H'FEE51	I/O port	8	3
Port M register	PORTM	8	H'FEE52	I/O port	8	3
Port M input buffer control register	PMICR	8	H'FEE53	I/O port	8	3
Serial mode register_5	SMR_5	8	H'FF600	SCI_5	8	3
Bit rate register_5	BRR_5	8	H'FF601	SCI_5	8	3
Serial control register_5	SCR_5	8	H'FF602	SCI_5	8	3
Transmit data register_5	TDR_5	8	H'FF603	SCI_5	8	3
Serial status register_5	SSR_5	8	H'FF604	SCI_5	8	3
Receive data register_5	RDR_5	8	H'FF605	SCI_5	8	3
Smart card mode register_5	SCMR_5	8	H'FF606	SCI_5	8	3
Serial extended mode register_5	SEMR_5	8	H'FF608	SCI_5	8	3
IrDA control register	IrCR	8	H'FF60C	SCI_5	8	3
Serial mode register_6	SMR_6	8	H'FF610	SCI_6	8	3

Timer counter	TCNT32K	8	H'FFABD	TM32K	8	2P
Port 1 data direction register	P1DDR	8	H'FFB80	I/O port	8	2P
Port 2 data direction register	P2DDR	8	H'FFB81	I/O port	8	2P
Port 3 data direction register	P3DDR	8	H'FFB82	I/O port	8	2P
Port 6 data direction register	P6DDR	8	H'FFB85	I/O port	8	2P
Port A data direction register	PADDR	8	H'FFB89	I/O port	8	2P
Port B data direction register	PBDDR	8	H'FFB8A	I/O port	8	2P
Port C data direction register	PCDDR	8	H'FFB8B	I/O port	8	2P
Port D data direction register	PDDDR	8	H'FFB8C	I/O port	8	2P
Port E data direction register	PEDDR	8	H'FFB8D	I/O port	8	2P
Port F data direction register	PFDDR	8	H'FFB8E	I/O port	8	2P
Port 1 input buffer control register	P1ICR	8	H'FFB90	I/O port	8	2P
Port 2 input buffer control register	P2ICR	8	H'FFB91	I/O port	8	2P
Port 3 input buffer control register	P3ICR	8	H'FFB92	I/O port	8	2P
Port 5 input buffer control register	P5ICR	8	H'FFB94	I/O port	8	2P
Port 6 input buffer control register	P6ICR	8	H'FFB95	I/O port	8	2P
Port A input buffer control register	PAICR	8	H'FFB99	I/O port	8	2P
Port B input buffer control register	PBICR	8	H'FFB9A	I/O port	8	2P
Port C input buffer control register	PCICR	8	H'FFB9B	I/O port	8	2P
Port D input buffer control register	PDICR	8	H'FFB9C	I/O port	8	2P
Port E input buffer control register	PEICR	8	H'FFB9D	I/O port	8	2P
Port F input buffer control register	PFICR	8	H'FFB9E	I/O port	8	2P
Port H register	PORTH	8	H'FFBA0	I/O port	8	2P
Port I register	PORTI	8	H'FFBA1	I/O port	8	2P
Port H data register	PHDR	8	H'FFBA4	I/O port	8	2P
Port I data register	PIDR	8	H'FFBA5	I/O port	8	2P
Port H data direction register	PHDDR	8	H'FFBA8	I/O port	8	2P
Port I data direction register	PIDDR	8	H'FFBA9	I/O port	8	2P

Port F open-drain control register	PFODR	8	H'FFBBD	I/O port	8	2I
Port function control register 0	PFCR0	8	H'FFBC0	I/O port	8	2I
Port function control register 1	PFCR1	8	H'FFBC1	I/O port	8	2I
Port function control register 2	PFCR2	8	H'FFBC2	I/O port	8	2I
Port function control register 4	PFCR4	8	H'FFBC4	I/O port	8	2I
Port function control register 6	PFCR6	8	H'FFBC6	I/O port	8	2I
Port function control register 7	PFCR7	8	H'FFBC7	I/O port	8	2I
Port function control register 9	PFCR9	8	H'FFBC9	I/O port	8	2I
Port function control register B	PFCRB	8	H'FFBCB	I/O port	8	2I
Port function control register C	PFCRC	8	H'FFBCC	I/O port	8	2I
Software standby release IRQ enable register	SSIER	16	H'FFBCE	INTC	8	2I
DMA source address register_0	DSAR_0	32	H'FFC00	DMAC_0	16	2I
DMA destination address register_0	DDAR_0	32	H'FFC04	DMAC_0	16	2I
DMA offset register_0	DOFR_0	32	H'FFC08	DMAC_0	16	2I
DMA transfer count register_0	DTCR_0	32	H'FFC0C	DMAC_0	16	2I
DMA block size register_0	DBSR_0	32	H'FFC10	DMAC_0	16	2I
DMA mode control register_0	DMDR_0	32	H'FFC14	DMAC_0	16	2I
DMA address control register_0	DACR_0	32	H'FFC18	DMAC_0	16	2I
DMA source address register_1	DSAR_1	32	H'FFC20	DMAC_1	16	2I
DMA destination address register_1	DDAR_1	32	H'FFC24	DMAC_1	16	2I
DMA offset register_1	DOFR_1	32	H'FFC28	DMAC_1	16	2I
DMA transfer count register_1	DTCR_1	32	H'FFC2C	DMAC_1	16	2I
DMA block size register_1	DBSR_1	32	H'FFC30	DMAC_1	16	2I
DMA mode control register_1	DMDR_1	32	H'FFC34	DMAC_1	16	2I
DMA address control register_1	DACR_1	32	H'FFC38	DMAC_1	16	2I
DMA source address register_2	DSAR_2	32	H'FFC40	DMAC_2	16	2I
DMA destination address register_2	DDAR_2	32	H'FFC44	DMAC_2	16	2I

DMA transfer count register_3	DTCR_3	32	H'FFC6C	DMAC_3	16	2Iφ
DMA block size register_3	DBSR_3	32	H'FFC70	DMAC_3	16	2Iφ
DMA mode control register_3	DMDR_3	32	H'FFC74	DMAC_3	16	2Iφ
DMA address control register_3	DACR_3	32	H'FFC78	DMAC_3	16	2Iφ
DMA module request select register_0	DMRSR_0	8	H'FFD20	DMAC_0	16	2Iφ
DMA module request select register_1	DMRSR_1	8	H'FFD21	DMAC_1	16	2Iφ
DMA module request select register_2	DMRSR_2	8	H'FFD22	DMAC_2	16	2Iφ
DMA module request select register_3	DMRSR_3	8	H'FFD23	DMAC_3	16	2Iφ
Interrupt priority register A	IPRA	16	H'FFD40	INTC	16	2Iφ
Interrupt priority register B	IPRB	16	H'FFD42	INTC	16	2Iφ
Interrupt priority register C	IPRC	16	H'FFD44	INTC	16	2Iφ
Interrupt priority register D	IPRD	16	H'FFD46	INTC	16	2Iφ
Interrupt priority register E	IPRE	16	H'FFD48	INTC	16	2Iφ
Interrupt priority register F	IPRF	16	H'FFD4A	INTC	16	2Iφ
Interrupt priority register G	IPRG	16	H'FFD4C	INTC	16	2Iφ
Interrupt priority register H	IPRH	16	H'FFD4E	INTC	16	2Iφ
Interrupt priority register I	IPRI	16	H'FFD50	INTC	16	2Iφ
Interrupt priority register K	IPRK	16	H'FFD54	INTC	16	2Iφ
Interrupt priority register L	IPRL	16	H'FFD56	INTC	16	2Iφ
Interrupt priority register Q	IPRQ	16	H'FFD60	INTC	16	2Iφ
Interrupt priority register R	IPRR	16	H'FFD62	INTC	16	2Iφ
IRQ sense control register H	ISCRH	16	H'FFD68	INTC	16	2Iφ
IRQ sense control register L	ISCRL	16	H'FFD6A	INTC	16	2Iφ
DTC vector base register	DTCVBR	32	H'FFD80	BSC	16	2Iφ
Bus width control register	ABWCR	16	H'FFD84	BSC	16	2Iφ
Access state control register	ASTCR	16	H'FFD86	BSC	16	2Iφ
Wait control register A	WTCRA	16	H'FFD88	BSC	16	2Iφ
Wait control register B	WTCRB	16	H'FFD8A	BSC	16	2Iφ

Address/data multiplexed I/O control register	MPXCR	16	H'FFD9C	BSC	16	2I
DRAM control register	DRAMCR	16	H'FFDA0	BSC	16	2I
DRAM access control register	DRACCR	16	H'FFDA2	BSC	16	2I
Synchronous DRAM control register	SDCR	16	H'FFDA4	BSC	16	2I
Refresh control register	REFCR	16	H'FFDA6	BSC	16	2I
Refresh timer counter	RTCNT	8	H'FFDA8	BSC	16	2I
Refresh time constant register	RTCOR	8	H'FFDA9	BSC	16	2I
RAM emulation register	RAMER	8	H'FFD9E	BSC	16	2I
Mode control register	MDCR	16	H'FFDC0	SYSTEM	16	2I
System control register	SYSCR	16	H'FFDC2	SYSTEM	16	2I
System clock control register	SCKCR	16	H'FFDC4	SYSTEM	16	2I
Standby control register	SBYCR	16	H'FFDC6	SYSTEM	16	2I
Module stop control register A	MSTPCRA	16	H'FFDC8	SYSTEM	16	2I
Module stop control register B	MSTPCRB	16	H'FFDCA	SYSTEM	16	2I
Module stop control register C	MSTPCRC	16	H'FFDCC	SYSTEM	16	2I
Subclock control register	SUBCKCR	8	H'FFDCF	SYSTEM	8	2I
Serial extended mode register_2	SEMR_2	8	H'FFE84	SCI_2	8	2I
Serial mode register_4	SMR_4	8	H'FFE90	SCI_4	8	2I
Bit rate register_4	BRR_4	8	H'FFE91	SCI_4	8	2I
Serial control register_4	SCR_4	8	H'FFE92	SCI_4	8	2I
Transmit data register_4	TDR_4	8	H'FFE93	SCI_4	8	2I
Serial status register_4	SSR_4	8	H'FFE94	SCI_4	8	2I
Receive data register_4	RDR_4	8	H'FFE95	SCI_4	8	2I
Smart card mode register_4	SCMR_4	8	H'FFE96	SCI_4	8	2I

I^{2}C bus interrupt enable register_0	ICIER_0	8	H'FFEB3	IIC2_0	8	2P
I^{2}C bus status register_0	ICSR_0	8	H'FFEB4	IIC2_0	8	2P
Slave address register_0	SAR_0	8	H'FFEB5	IIC2_0	8	2P
I^{2}C bus transmit data register_0	ICDRT_0	8	H'FFEB6	IIC2_0	8	2P
I^{2}C bus receive data register_0	ICDRR_0	8	H'FFEB7	IIC2_0	8	2P
I^{2}C bus control register A_1	ICCRA_1	8	H'FFEB8	IIC2_1	8	2P
I^{2}C bus control register B_1	ICCRB_1	8	H'FFEB9	IIC2_1	8	2P
I^{2}C bus mode register_1	ICMR_1	8	H'FFEBA	IIC2_1	8	2P
I^{2}C bus interrupt enable register_1	ICIER_1	8	H'FFEBB	IIC2_1	8	2P
I^{2}C bus status register_1	ICSR_1	8	H'FFEBC	IIC2_1	8	2P
Slave address register_1	SAR_1	8	H'FFEBD	IIC2_1	8	2P
I^{2}C bus transmit data register_1	ICDRT_1	8	H'FFEBE	IIC2_1	8	2P
I^{2}C bus receive data register_1	ICDRR_1	8	H'FFEBF	IIC2_1	8	2P
Timer control register_2	TCR_2	8	H'FFEC0	TMR_2	16	2P
Timer control register_3	TCR_3	8	H'FFEC1	TMR_3	16	2P
Timer control/status register_2	TCSR_2	8	H'FFEC2	TMR_2	16	2P
Timer control/status register_3	TCSR_3	8	H'FFEC3	TMR_3	16	2P
Time constant register A_2	TCORA_2	8	H'FFEC4	TMR_2	16	2P
Time constant register A_3	TCORA_3	8	H'FFEC5	TMR_3	16	2P
Time constant register B_2	TCORB_2	8	H'FFEC6	TMR_2	16	2P
Time constant register B_3	TCORB_3	8	H'FFEC7	TMR_3	16	2P
Timer counter_2	TCNT_2	8	H'FFEC8	TMR_2	16	2P
Timer counter_3	TCNT_3	8	H'FFEC9	TMR_3	16	2P
Timer counter control register_2	TCCR_2	8	H'FFECA	TMR_2	16	2P
Timer counter control register_3	TCCR_3	8	H'FFECB	TMR_3	16	2P

Timer control register_5	TCR_5	8	H'FFEF0	TPU_5	16	2[illegible]
Timer mode register_5	TMDR_5	8	H'FFEF1	TPU_5	16	2[illegible]
Timer I/O control register_5	TIOR_5	8	H'FFEF2	TPU_5	16	2[illegible]
Timer interrupt enable register_5	TIER_5	8	H'FFEF4	TPU_5	16	2[illegible]
Timer status register_5	TSR_5	8	H'FFEF5	TPU_5	16	2[illegible]
Timer counter_5	TCNT_5	16	H'FFEF6	TPU_5	16	2[illegible]
Timer general register A_5	TGRA_5	16	H'FFEF8	TPU_5	16	2[illegible]
Timer general register B_5	TGRB_5	16	H'FFEFA	TPU_5	16	2[illegible]
DTC enable register A	DTCERA	16	H'FFF20	INTC	16	2[illegible]
DTC enable register B	DTCERB	16	H'FFF22	INTC	16	2[illegible]
DTC enable register C	DTCERC	16	H'FFF24	INTC	16	2[illegible]
DTC enable register D	DTCERD	16	H'FFF26	INTC	16	2[illegible]
DTC enable register E	DTCERE	16	H'FFF28	INTC	16	2[illegible]
DTC enable register G	DTCERG	16	H'FFF2C	INTC	16	2[illegible]
DTC enable register H	DTCERH	16	H'FFF2E	INTC	16	2[illegible]
DTC control register	DTCCR	8	H'FFF30	INTC	16	2[illegible]
Interrupt control register	INTCR	8	H'FFF32	INTC	16	2[illegible]
CPU priority control register	CPUPCR	8	H'FFF33	INTC	16	2[illegible]
IRQ enable register	IER	16	H'FFF34	INTC	16	2[illegible]
IRQ status register	ISR	16	H'FFF36	INTC	16	2[illegible]
Port 1 register	PORT1	8	H'FFF40	I/O port	8	2[illegible]
Port 2 register	PORT2	8	H'FFF41	I/O port	8	2[illegible]
Port 3 register	PORT3	8	H'FFF42	I/O port	8	2[illegible]
Port 5 register	PORT5	8	H'FFF44	I/O port	8	2[illegible]
Port 6 register	PORT6	8	H'FFF45	I/O port	8	2[illegible]
Port A register	PORTA	8	H'FFF49	I/O port	8	2[illegible]
Port B register	PORTB	8	H'FFF4A	I/O port	8	2[illegible]

Port A data register	PADR	8	H'FFF59	I/O port	8	2P
Port B data register	PBDR	8	H'FFF5A	I/O port	8	2P
Port C data register	PCDR	8	H'FFF5B	I/O port	8	2P
Port D data register	PDDR	8	H'FFF5C	I/O port	8	2P
Port E data register	PEDR	8	H'FFF5D	I/O port	8	2P
Port F data register	PFDR	8	H'FFF5E	I/O port	8	2P
Serial mode register_2	SMR_2	8	H'FFF60	SCI_2	8	2P
Bit rate register_2	BRR_2	8	H'FFF61	SCI_2	8	2P
Serial control register_2	SCR_2	8	H'FFF62	SCI_2	8	2P
Transmit data register_2	TDR_2	8	H'FFF63	SCI_2	8	2P
Serial status register_2	SSR_2	8	H'FFF64	SCI_2	8	2P
Receive data register_2	RDR_2	8	H'FFF65	SCI_2	8	2P
Smart card mode register_2	SCMR_2	8	H'FFF66	SCI_2	8	2P
D/A data register 0	DADR0	8	H'FFF68	D/A	8	2P
D/A data register 1	DADR1	8	H'FFF69	D/A	8	2P
D/A control register 01	DACR01	8	H'FFF6A	D/A	8	2P
PPG output control register	PCR	8	H'FFF76	PPG	8	2P
PPG output mode register	PMR	8	H'FFF77	PPG	8	2P
Next data enable register H	NDERH	8	H'FFF78	PPG	8	2P
Next data enable register L	NDERL	8	H'FFF79	PPG	8	2P
Output data register H	PODRH	8	H'FFF7A	PPG	8	2P
Output data register L	PODRL	8	H'FFF7B	PPG	8	2P
Next data register H*	NDRH	8	H'FFF7C	PPG	8	2P
Next data register L*	NDRL	8	H'FFF7D	PPG	8	2P
Next data register H*	NDRH	8	H'FFF7E	PPG	8	2P
Next data register L*	NDRL	8	H'FFF7F	PPG	8	2P

Bit rate register_1	BRR_1	8	H'FFF89	SCI_1	8	2I
Serial control register_1	SCR_1	8	H'FFF8A	SCI_1	8	2I
Transmit data register_1	TDR_1	8	H'FFF8B	SCI_1	8	2I
Serial status register_1	SSR_1	8	H'FFF8C	SCI_1	8	2I
Receive data register_1	RDR_1	8	H'FFF8D	SCI_1	8	2I
Smart card mode register_1	SCMR_1	8	H'FFF8E	SCI_1	8	2I
A/D data register A	ADDRA	16	H'FFF90	A/D	16	2I
A/D data register B	ADDRB	16	H'FFF92	A/D	16	2I
A/D data register C	ADDRC	16	H'FFF94	A/D	16	2I
A/D data register D	ADDRD	16	H'FFF96	A/D	16	2I
A/D data register E	ADDRE	16	H'FFF98	A/D	16	2I
A/D data register F	ADDRF	16	H'FFF9A	A/D	16	2I
A/D data register G	ADDRG	16	H'FFF9C	A/D	16	2I
A/D data register H	ADDRH	16	H'FFF9E	A/D	16	2I
A/D control/status register	ADCSR	8	H'FFFA0	A/D	16	2I
A/D control register	ADCR	8	H'FFFA1	A/D	16	2I
Timer control/status register	TCSR	8	H'FFFA4	WDT		2I
Timer counter	TCNT	8	H'FFFA5	WDT		2I
Reset control/status register	RSTCSR	8	H'FFFA7	WDT		2I
Timer control register_0	TCR_0	8	H'FFFB0	TMR_0	16	2I
Timer control register_1	TCR_1	8	H'FFFB1	TMR_1	16	2I
Timer control/status register_0	TCSR_0	8	H'FFFB2	TMR_0	16	2I
Timer control/status register_1	TCSR_1	8	H'FFFB3	TMR_1	16	2I
Time constant register A_0	TCORA_0	8	H'FFFB4	TMR_0	16	2I
Time constant register A_1	TCORA_1	8	H'FFFB5	TMR_1	16	2I
Time constant register B_0	TCORB_0	8	H'FFFB6	TMR_0	16	2I
Time constant register B_1	TCORB_1	8	H'FFFB7	TMR_1	16	2I

Timer I/O control register H_0	TIORH_0	8	H'FFFC2	TPU_0	16	2P
Timer I/O control register L_0	TIORL_0	8	H'FFFC3	TPU_0	16	2P
Timer interrupt enable register_0	TIER_0	8	H'FFFC4	TPU_0	16	2P
Timer status register_0	TSR_0	8	H'FFFC5	TPU_0	16	2P
Timer counter_0	TCNT_0	16	H'FFFC6	TPU_0	16	2P
Timer general register A_0	TGRA_0	16	H'FFFC8	TPU_0	16	2P
Timer general register B_0	TGRB_0	16	H'FFFCA	TPU_0	16	2P
Timer general register C_0	TGRC_0	16	H'FFFCC	TPU_0	16	2P
Timer general register D_0	TGRD_0	16	H'FFFCE	TPU_0	16	2P
Timer control register_1	TCR_1	8	H'FFFD0	TPU_1	16	2P
Timer mode register_1	TMDR_1	8	H'FFFD1	TPU_1	16	2P
Timer I/O control register_1	TIOR_1	8	H'FFFD2	TPU_1	16	2P
Timer interrupt enable register_1	TIER_1	8	H'FFFD4	TPU_1	16	2P
Timer status register_1	TSR_1	8	H'FFFD5	TPU_1	16	2P
Timer counter_1	TCNT_1	16	H'FFFD6	TPU_1	16	2P
Timer general register A_1	TGRA_1	16	H'FFFD8	TPU_1	16	2P
Timer general register B_1	TGRB_1	16	H'FFFDA	TPU_1	16	2P
Timer control register_2	TCR_2	8	H'FFFE0	TPU_2	16	2P
Timer mode register_2	TMDR_2	8	H'FFFE1	TPU_2	16	2P
Timer I/O control register_2	TIOR_2	8	H'FFFE2	TPU_2	16	2P
Timer interrupt enable register_2	TIER_2	8	H'FFFE4	TPU_2	16	2P
Timer status register_2	TSR_2	8	H'FFFE5	TPU_2	16	2P
Timer counter_2	TCNT_2	16	H'FFFE6	TPU_2	16	2P
Timer general register A_2	TGRA_2	16	H'FFFE8	TPU_2	16	2P
Timer general register B_2	TGRB_2	16	H'FFFEA	TPU_2	16	2P
Timer control register_3	TCR_3	8	H'FFFF0	TPU_3	16	2P
Timer mode register_3	TMDR_3	8	H'FFFF1	TPU_3	16	2P

Timer general register D_3	TGRD_3	16	H'FFFFE	TPU_3	16	2I

Note: * When the same output trigger is specified for pulse output groups 2 and 3 by setting, the NDRH address is H'FFF7C. When different output triggers are sp NDRH addresses for pulse output groups 2 and 3 are H'FFF7E and H'FFF7C respectively. Similarly, When the same output trigger is specified for pulse ou groups 0 and 1 by the PCR setting, the NDRL address is H'FFF7D. When dif output triggers are specified, the NDRL addresses for pulse output groups 0 a H'FFF7F and H'FFF7D, respectively.

TCSR_5	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
TCORA_4								
TCORA_5								
TCORB_4								
TCORB_5								
TCNT_4								
TCNT_5								
TCCR_4	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCCR_5	—	—	—	—	TMRIS	—	ICKS1	ICKS0
CRCCR	DORCLR	—	—	—	—	LMS	G1	G0
CRCDIR								
CRCDOR								
TCR_6	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCR_7	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCSR_6	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
TCSR_7	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
TCORA_6								
TCORA_7								
TCORB_6								
TCORB_7								
TCNT_6								
TCNT_7								
TCCR_6	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCCR_7	—	—	—	—	TMRIS	—	ICKS1	ICKS0
IFR0	BRST	EP1 FULL	EP2 TR	EP2 EMPTY	SETUP TS	EP0o TS	EP0i TR	EP0i TS
IFR1	—	—	—	—	VBUS MN	EP3 TR	EP3 TS	VBUSF
IFR2	—	—	SURSS	SURSF	CFDN	—	SETC	SETI
IER0	BRST	EP1 FULL	EP2 TR	EP2 EMPTY	SETUP TS	EP0o TS	EP0i TR	EP0i TS

EPDR2	D7	D6	D5	D4	D3	D2	D1	D0
EPDR3	D7	D6	D5	D4	D3	D2	D1	D0
EPSZ0o	—	—	—	—	—	—	—	—
EPSZ1	—	—	—	—	—	—	—	—
DASTS	—	—	EP3 DE	EP2 DE	—	—	—	EP0i D
FCLR	—	EP3 CLR	EP1 CLR	EP2 CLR	—	—	EP0o CLR	EP0i C
EPSTL	—	—	—	—	EP3STL	EP2STL	EP1STL	EP0ST
TRG	—	EP3 PKTE	EP1 RDFN	EP2 PKTE	—	EP0s RDFN	EP0o RDFN	EP0i PK
DMA	—	—	—	—	—	PULLUP_E	EP2DMAE	EP1DM
CVR	CNFV1	CNFV0	INTV1	INTV0	—	ALTV2	ALTV1	ALTV0
CTLR	—	—	—	RWUPS	RSME	RWMD	ASCE	—
EPIR	D7	D6	D5	D4	D3	D2	D1	D0
TRNTREG0	PTSTE	—	—	—	SUSPEND	txenl	txse0	txdata
TRNTREG1	—	—	—	—	—	xver_data	dpls	dmns
PMDDR	—	—	—	PM4DDR	PM3DDR	PM2DDR	PM1DDR	PM0DD
PMDR	—	—	—	PM4DR	PM3DR	PM2DR	PM1DR	PM0DF
PORTM	—	—	—	PM4	PM3	PM2	PM1	PM0
PMICR	—	—	—	PM4ICR	PM3ICR	PM2ICR	PM1ICR	PM0ICI
SMR_5*	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
	(GM)	(BLK)	(PE)	(O/$\overline{E}$)	(BCP1)	(BCP0)		
BRR_5								
SCR_5*	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_5								
SSR_5*	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
				(ERS)				

SSR_6*	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_6								
SCMR_6	—	—	—	—	SDIR	SINV	—	SMIF
SEMR_6	—	—	—	ABCS	ACS3	ACS2	ACS1	ACS0
TCNT32K								
TCR32K	—	—	TME	—	—	OSC32STP	CKS1	CKS0
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
P3DDR	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P320DDR
P6DDR	—	—	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR
PCDDR	—	—	—	—	PC3DDR	PC2DDR	—	—
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
PFDDR	PF7DDR	PF6DDR	PF5IDDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR
P1ICR	P17ICR	P16ICR	P15ICR	P14ICR	P13ICR	P12ICR	P11ICR	P10ICR
P2ICR	P27ICR	P26ICR	P25ICR	P24ICR	P23ICR	P22ICR	P21ICR	P20ICR
P3ICR	P37ICR	P36ICR	P35ICR	P34ICR	P33ICR	P32ICR	P31ICR	P30ICR
P5ICR	P57ICR	P56ICR	P55ICR	P54ICR	P53ICR	P52ICR	P51ICR	P50ICR
P6ICR	—	—	P65ICR	P64ICR	P63ICR	P62ICR	P61ICR	P60ICR
PAICR	PA7ICR	PA6ICR	PA5ICR	PA4ICR	PA3ICR	PA2ICR	PA1ICR	PA0ICR
PBICR	PB7ICR	PB6ICR	PB5ICR	PB4ICR	PB3ICR	PB2ICR	PB1ICR	PB0ICR
PCICR	—	—	—	—	PC3ICR	PC2ICR	—	—
PDICR	PD7ICR	PD6ICR	PD5ICR	PD4ICR	PD3ICR	PD2ICR	PD1ICR	PD0ICR
PEICR	PE7ICR	PE6ICR	PE5ICR	PE4ICR	PE3ICR	PE2ICR	PE1ICR	PE0ICR

PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PC
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PC
PFPCR	PF7PCR	PF6PCR	PF5PCR	PF4PCR	PF3PCR	PF2PCR	PF1PCR	PF0PC
PHPCR	PH7PCR	PH6PCR	PH5PCR	PH4PCR	PH3PCR	PH2PCR	PH1PCR	PH0PC
PIPCR	PI7PCR	PI6PCR	PI5PCR	PI4PCR	PI3PCR	PI2PCR	PI1PCR	PI0PCI
P2ODR	P27ODR	P26ODR	P25ODR	P24ODR	P23ODR	P22ODR	P21ODR	P20OD
PFODR	PF7ODR	PF6ODR	PF5ODR	PF4ODR	PF3ODR	PF2ODR	PF1ODR	PF0OD
PFCR0	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E
PFCR1	CS7SA	CS7SB	CS6SA	CS6SB	CS5SA	CS5SB	CS4SA	CS4SB
PFCR2	—	CS2S	BSS	BSE	RDWRS	RDWRE	ASOE	—
PFCR4	A23E	A22E	A21E	A20E	A19E	A18E	A17E	A16E
PFCR6	—	LHWROE	—	—	TCLKS	—	—	—
PFCR7	DMAS3A	DMAS3B	DMAS2A	DMAS2B	DMAS1A	DMAS1B	DMAS0A	DMAS0
PFCR9	TPUMS5	TPUMS4	TPUMS3A	TPUMS3B	TPUMS2A	TPUMS2B	TPUMS1A	TPUMS
PFCRB	—	—	—	—	ITS11	ITS10	ITS9	ITS8
PFCRC	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	ITS0
SSIER	SSI15	—	—	—	SSI11	SSI10	SSI9	SSI8
	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	SSI0
DSAR_0								
DDAR_0								

	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH1
	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8
	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0
DMDR_0	DTE	DACKE	TENDE	—	DREQS	NRD	—	—
	ACT	—	—	—	ERRF	—	ESIF	DTIF
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0
DACR_0	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0
	—	—	SAT1	SAT0	—	—	DAT1	DAT0
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0
DSAR_1								
DDAR_1								
DOFR_1								
DTCR_1								

	—	—	SAT1	SAT0	—	—	DAT1	DAT0
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0
DSAR_2								
DDAR_2								
DOFR_2								
DTCR_2								
DBSR_2	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH
	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8
	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0
DMDR_2	DTE	DACKE	TENDE	—	DREQS	NRD	—	—
	ACT	—	—	—	—	—	ESIF	DTIF
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0

DOFR_3								
DTCR_3								
DBSR_3	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH2
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH1
	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8
	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0
DMDR_3	DTE	DACKE	TENDE	—	DREQS	NRD	—	—
	ACT	—	—	—	—	—	ESIF	DTIF
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0
DACR_3	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0
	—	—	SAT1	SAT0	—	—	DAT1	DAT0
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0
DMRSR_0								
DMRSR_1								
DMRSR_2								
DMRSR_3								

	—	—	—	—	—	—	—	—
IPRF	—	—	—	—	—	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
IPRK	—	IPRK14	IPRK13	IPRK12	—	—	—	—
	—	IPRK6	IPRK5	IPRK4	—	IPRK2	IPRK1	IPRK0
IPRL	—	IPRL14	IPRL13	IPRL12	—	—	—	—
	—	IPRL6	IPRL5	IPRL4	—	—	—	—
IPRQ	—	—	—	—	—	—	—	—
	—	IPRQ6	IPRQ5	IPRQ4	—	IPRQ2	IPRQ1	IPRQ0
IPRR	—	IPRR14	IPRR13	IPRR12	—	IPRR10	IPRR9	IPRR8
	—	IPRR6	IPRR5	IPRR4	—	IPRR2	IPRR1	IPRR0
ISCRH	IRQ15SR	IRQ15SF	—	—	—	—	—	—
	IRQ11SR	IRQ11SF	IRQ10SR	IRQ10SF	IRQ9SR	IRQ9SF	IRQ8SR	IRQ8SF
ISCRL	IRQ7SR	IRQ7SF	IRQ6SR	IRQ6SF	IRQ5SR	IRQ5SF	IRQ4SR	IRQ4SF
	IRQ3SR	IRQ3SF	IRQ2SR	IRQ2SF	IRQ1SR	IRQ1SF	IRQ0SR	IRQ0SF
DTCVBR								
ABWCR	ABWH7	ABWH6	ABWH5	ABWH4	ABWH3	ABWH2	ABWH1	ABWH0
	ABWL7	ABWL6	ABWL5	ABWL4	ABWL3	ABWL2	ABWL1	ABWL0

	CSXT7	CSXT6	CSXT5	CSXT4	CSXT3	CSXT2	CSXT1	CSXT0
IDLCR	IDLS3	IDLS2	IDLS1	IDLS0	IDLCB1	IDLCB0	IDLCA1	IDLCA0
	IDLSEL7	IDLSEL6	IDLSEL5	IDLSEL4	IDLSEL3	IDLSEL2	IDLSEL1	IDLSEL0
BCR1	BRLE	BREQOE	—	—	—	—	WDBE	WAITE
	DKC	—	—	—	—	—	—	—
BCR2	—	—	—	IBCCS	—	—	—	PWDBE
ENDIANCR	LE7	LE6	LE5	LE4	LE3	LE2	—	—
SRAMCR	BCSEL7	BCSEL6	BCSEL5	BCSEL4	BCSEL3	BCSEL2	BCSEL1	BCSEL0
	—	—	—	—	—	—	—	—
BROMCR	BSRM0	BSTS02	BSTS01	BSTS00	—	—	BSWD01	BSWD00
	BSRM1	BSTS12	BSTS11	BSTS10	—	—	BSWD11	BSWD10
MPXCR	MPXE7	MPXE6	MPXE5	MPXE4	MPXE3	—	—	—
	—	—	—	—	—	—	—	ADDEX
DRAMCR	DRAME	DTYPE	—	—	OEE	RAST	—	CAST
	BE	RCDM	DDS	—	—	—	MXC1	MXC0
DRACCR	—	—	TPC1	TPC0	—	—	RCD1	RCD0
	—	—	—	—	—	—	—	—
SDCR	MRSE	—	—	—	—	—	—	—
	CKSPE	—	—	—	—	—	—	TRWL
REFCR	CMF	CMIE	RCW1	RCW0	—	RTCK2	RTCK1	RTCK0
	RFSHE	RLW2	RLW1	RLW0	SLFRF	TPCS2	TPCS1	TPCS0
RTCNT								
RTCOR								
RAMER	—	—	—	—	RAMS	RAM2	RAM1	RAM0

	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA
MSTPCRB	MSTPB15	MSTPB14	MSTPB13	MSTPB12	MSTPB11	MSTPB10	MSTPB9	MSTPB
	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB
MSTPCRC	MSTPC15	MSTPC14	MSTPC13	MSTPC12	MSTPC11	MSTPC10	MSTPC9	MSTPC
	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC
SUBCKCR	—	—	—	—	—	EXSTP	WAKE32K	CS32K
SEMR_2	—	—	—	—	ABCS	ACS2	ACS1	ACS0
SMR_4*	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_4								
SCR_4*	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_4								
SSR_4*	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_4								
SCMR_4	—	—	—	—	SDIR	SINV	—	SMIF
FCCS	—	—	—	FLER	—	—	—	SCO
FPCS	—	—	—	—	—	—	—	PPVS
FECS	—	—	—	—	—	—	—	EPVB
FKEY	K7	K6	K5	K4	K3	K2	K1	K0
FTDAR	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0
ICCRA_0	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0
ICCRB_0	BBSY	SCP	SDAO	—	SCLO	—	IICRST	—
ICMR_0	—	WAIT	—	—	BCWP	BC2	BC1	BC0
ICIER_0	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
ICSR_0	TDRE	TEND	RDRF	NACKF	STOP	AL	AAS	ADZ
SAR_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	—

ICDRR_1								
TCR_2	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCR_3	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCSR_2	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
TCSR_3	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
TCORA_2								
TCORA_3								
TCORB_2								
TCORB_3								
TCNT_2								
TCNT_3								
TCCR_2	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCCR_3	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCR_4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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								
TGRA_4								
TGRB_4								

TGRB_5								
DTCERA	DTCEA15	DTCEA14	DTCEA13	DTCEA12	DTCEA11	DTCEA10	DTCEA9	DTCEA
	DTCEA7	DTCEA6	DTCEA5	DTCEA4	—	—	—	—
DTCERB	DTCEB15	—	DTCEB13	DTCEB12	DTCEB11	DTCEB10	DTCEB9	DTCEB
	DTCEB7	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB
DTCERC	DTCEC15	DTCEC14	DTCEC13	DTCEC12	DTCEC11	DTCEC10	DTCEC9	DTCEC
	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	—	—
DTCERD	—	—	DTCED13	DTCED12	DTCED11	DTCED10	—	—
	—	—	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED
DTCERE	—	—	DTCEE13	DTCEE12	—	—	—	—
	—	—	—	—	—	—	—	—
DTCERG	—	—	—	—	DTCEG11	DTCEG10	—	—
	DTCEG7	DTCEG6	—	—	—	—	—	—
DTCERH	DTCEH15	DTCEH14	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
DTCCR	—	—	—	RRS	RCHNE	—	—	ERR
INTCR	—	—	INTM1	INTM0	NMIEG	—	—	—
CPUPCR	CPUPCE	DTCP2	DTCP1	DTCP0	IPSETE	CPUP2	CPUP1	CPUP0
IER	IRQ15E	—	—	—	IRQ11E	IRQ10E	IRQ9E	IRQ8E
	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
ISR	IRQ15F	—	—	—	IRQ11F	IRQ10F	IRQ9F	IRQ8F
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F

PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR
P3DR	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
P6DR	—	—	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR
PADR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
PCDR	—	—	—	—	PC3DR	PC2DR	—	—
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
SMR_2*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_2								
SCR_2*[1]	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_2								
SSR_2*[1]	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_2								
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF
DADR0								
DADR1								
DACR01	DAOE1	DAOE0	DAE	—	—	—	—	—

NDRL*[2]	—	—	—	—	NDR3	NDR2	NDR1	NDR0
SMR_0*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_0								
SCR_0*[1]	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_0								
SSR_0*[1]	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_0								
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF
SMR_1*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_1								
SCR_1*[1]	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_1								
SSR_1*[1]	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_1								
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF
ADDRA								
ADDRB								
ADDRC								
ADDRD								

ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	—	—
TCSR	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
TCNT								
RSTCSR	WOVF	RSTE	—	—	—	—	—	—
TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCSR_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
TCSR_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
TCORA_0								
TCORA_1								
TCORB_0								
TCORB_1								
TCNT_0								
TCNT_1								
TCCR_0	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCCR_1	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0
TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0
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								

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	TGFD	—	TGFB	TGFA
TCNT_1								
TGRA_1								
TGRB_1								
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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								
TGRA_2								
TGRB_2								
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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

Notes: 1. Parts of the bit functions differ in normal mode and the smart card interface.

2. When the same output trigger is specified for pulse output groups 2 and 3 by th setting, the NDRH address is H'FFF7C. When different output triggers are spe NDRH addresses for pulse output groups 2 and 3 are H'FFF7E and H'FFF7C, respectively. Similarly, When the same output trigger is specified for pulse outp groups 0 and 1 by the PCR setting, the NDRL address is H'FFF7D. When diffe output triggers are specified, the NDRL addresses for pulse output groups 0 ar H'FFF7F and H'FFF7D, respectively.

TCORA_5	Initialized	—	—	—	—	Initialized
TCORB_4	Initialized	—	—	—	—	Initialized
TCORB_5	Initialized	—	—	—	—	Initialized
TCNT_4	Initialized	—	—	—	—	Initialized
TCNT_5	Initialized	—	—	—	—	Initialized
TCCR_4	Initialized	—	—	—	—	Initialized
TCCR_5	Initialized	—	—	—	—	Initialized
CRCCR	Initialized	—	—	—	—	Initialized
CRCDIR	Initialized	—	—	—	—	Initialized
CRCDOR	Initialized	—	—	—	—	Initialized
TCR_6	Initialized	—	—	—	—	Initialized
TCR_7	Initialized	—	—	—	—	Initialized
TCSR_6	Initialized	—	—	—	—	Initialized
TCSR_7	Initialized	—	—	—	—	Initialized
TCORA_6	Initialized	—	—	—	—	Initialized
TCORA_7	Initialized	—	—	—	—	Initialized
TCORB_6	Initialized	—	—	—	—	Initialized
TCORB_7	Initialized	—	—	—	—	Initialized
TCNT_6	Initialized	—	—	—	—	Initialized
TCNT_7	Initialized	—	—	—	—	Initialized
TCCR_6	Initialized	—	—	—	—	Initialized
TCCR_7	Initialized	—	—	—	—	Initialized

ISR1	Initialized	—	—	—	—	Initialized	
ISR2	Initialized	—	—	—	—	Initialized	
EPDR0i	Initialized	—	—	—	—	Initialized	
EPDR0o	Initialized	—	—	—	—	Initialized	
EPDR0s	Initialized	—	—	—	—	Initialized	
EPDR1	Initialized	—	—	—	—	Initialized	
EPDR2	Initialized	—	—	—	—	Initialized	
EPDR3	Initialized	—	—	—	—	Initialized	
EPSZ0o	Initialized	—	—	—	—	Initialized	
EPSZ1	Initialized	—	—	—	—	Initialized	
DASTS	Initialized	—	—	—	—	Initialized	
FCLR	Initialized	—	—	—	—	Initialized	
EPSTL	Initialized	—	—	—	—	Initialized	
TRG	Initialized	—	—	—	—	Initialized	
DMA	Initialized	—	—	—	—	Initialized	
CVR	Initialized	—	—	—	—	Initialized	
CTLR	Initialized	—	—	—	—	Initialized	
EPIR	Initialized	—	—	—	—	Initialized	
TRNTREG0	Initialized	—	—	—	—	Initialized	
TRNTREG1	Initialized	—	—	—	—	Initialized	
PMDDR	Initialized	—	—	—	—	Initialized	I
PMDR	Initialized	—	—	—	—	Initialized	
PORTM	—	—	—	—	—	—	
PMICR	Initialized	—	—	—	—	Initialized	

SEMR_5	Initialized	—	—	—	—	Initialized
IrCR	Initialized	—	—	—	—	Initialized
SMR_6	Initialized	—	—	—	—	Initialized
BRR_6	Initialized	—	—	—	—	Initialized
SCR_6	Initialized	—	—	—	—	Initialized
TDR_6	Initialized	Initialized	—	Initialized	Initialized	Initialized
SSR_6	Initialized	Initialized	—	Initialized	Initialized	Initialized
RDR_6	Initialized	Initialized	—	Initialized	Initialized	Initialized
SCMR_6	Initialized	—	—	—	—	Initialized
SEMR_6	Initialized	—	—	—	—	Initialized
TCNT32K	Initialized	—	—	—	—	Initialized
TCR32K	Initialized	—	—	—	—	Initialized
P1DDR	Initialized	—	—	—	—	Initialized
P2DDR	Initialized	—	—	—	—	Initialized
P3DDR	Initialized	—	—	—	—	Initialized
P6DDR	Initialized	—	—	—	—	Initialized
PADDR	Initialized	—	—	—	—	Initialized
PBDDR	Initialized	—	—	—	—	Initialized
PCDDR	Initialized	—	—	—	—	Initialized
PDDDR	Initialized	—	—	—	—	Initialized
PEDDR	Initialized	—	—	—	—	Initialized
PFDDR	Initialized	—	—	—	—	Initialized

PCICR	Initialized	—	—	—	—	Initialized
PDICR	Initialized	—	—	—	—	Initialized
PEICR	Initialized	—	—	—	—	Initialized
PFICR	Initialized	—	—	—	—	Initialized
PORTH	—	—	—	—	—	—
PORTI	—	—	—	—	—	—
PHDR	Initialized	—	—	—	—	Initialized
PIDR	Initialized	—	—	—	—	Initialized
PHDDR	Initialized	—	—	—	—	Initialized
PIDDR	Initialized	—	—	—	—	Initialized
PHICR	Initialized	—	—	—	—	Initialized
PIICR	Initialized	—	—	—	—	Initialized
PDPCR	Initialized	—	—	—	—	Initialized
PEPCR	Initialized	—	—	—	—	Initialized
PFPCR	Initialized	—	—	—	—	Initialized
PHPCR	Initialized	—	—	—	—	Initialized
PIPCR	Initialized	—	—	—	—	Initialized
P2ODR	Initialized	—	—	—	—	Initialized
PFODR	Initialized	—	—	—	—	Initialized
PFCR0	Initialized	—	—	—	—	Initialized
PFCR1	Initialized	—	—	—	—	Initialized
PFCR2	Initialized	—	—	—	—	Initialized
PFCR4	Initialized	—	—	—	—	Initialized

DDAR_0	Initialized	—	—	—	—	Initialized
DOFR_0	Initialized	—	—	—	—	Initialized
DTCR_0	Initialized	—	—	—	—	Initialized
DBSR_0	Initialized	—	—	—	—	Initialized
DMDR_0	Initialized	—	—	—	—	Initialized
DACR_0	Initialized	—	—	—	—	Initialized
DSAR_1	Initialized	—	—	—	—	Initialized
DDAR_1	Initialized	—	—	—	—	Initialized
DOFR_1	Initialized	—	—	—	—	Initialized
DTCR_1	Initialized	—	—	—	—	Initialized
DBSR_1	Initialized	—	—	—	—	Initialized
DMDR_1	Initialized	—	—	—	—	Initialized
DACR_1	Initialized	—	—	—	—	Initialized
DSAR_2	Initialized	—	—	—	—	Initialized
DDAR_2	Initialized	—	—	—	—	Initialized
DOFR_2	Initialized	—	—	—	—	Initialized
DTCR_2	Initialized	—	—	—	—	Initialized
DBSR_2	Initialized	—	—	—	—	Initialized
DMDR_2	Initialized	—	—	—	—	Initialized
DACR_2	Initialized	—	—	—	—	Initialized

DMRSR_0	Initialized	—	—	—	—	Initialized
DMRSR_1	Initialized	—	—	—	—	Initialized
DMRSR_2	Initialized	—	—	—	—	Initialized
DMRSR_3	Initialized	—	—	—	—	Initialized
IPRA	Initialized	—	—	—	—	Initialized
IPRB	Initialized	—	—	—	—	Initialized
IPRC	Initialized	—	—	—	—	Initialized
IPRD	Initialized	—	—	—	—	Initialized
IPRE	Initialized	—	—	—	—	Initialized
IPRF	Initialized	—	—	—	—	Initialized
IPRG	Initialized	—	—	—	—	Initialized
IPRH	Initialized	—	—	—	—	Initialized
IPRI	Initialized	—	—	—	—	Initialized
IPRK	Initialized	—	—	—	—	Initialized
IPRL	Initialized	—	—	—	—	Initialized
IPRQ	Initialized	—	—	—	—	Initialized
IPRR	Initialized	—	—	—	—	Initialized
ISCRH	Initialized	—	—	—	—	Initialized
ISCRL	Initialized	—	—	—	—	Initialized
DTCVBR	Initialized	—	—	—	—	Initialized
ABWCR	Initialized	—	—	—	—	Initialized
ASTCR	Initialized	—	—	—	—	Initialized

ENDIANCR	Initialized	—	—	—	—	Initialized
SRAMCR	Initialized	—	—	—	—	Initialized
BROMCR	Initialized	—	—	—	—	Initialized
MPXCR	Initialized	—	—	—	—	Initialized
DRAMCR	Initialized	—	—	—	—	Initialized
DRACCR	Initialized	—	—	—	—	Initialized
SDCR	Initialized	—	—	—	—	Initialized
REFCR	Initialized	—	—	—	—	Initialized
RTCNT	Initialized	—	—	—	—	Initialized
RTCOR	Initialized	—	—	—	—	Initialized
RAMER	Initialized	—	—	—	—	Initialized
MDCR	Initialized	—	—	—	—	Initialized
SYSCR	Initialized	—	—	—	—	Initialized
SCKCR	Initialized	—	—	—	—	Initialized
SBYCR	Initialized	—	—	—	—	Initialized
MSTPCRA	Initialized	—	—	—	—	Initialized
MSTPCRB	Initialized	—	—	—	—	Initialized
MSTPCRC	Initialized	—	—	—	—	Initialized
SUBCKCR	Initialized	—	—	—	—	Initialized
SEMR_2	Initialized	—	—	—	—	Initialized

FCCS	Initialized	—	—	—	—	Initialized	I
FPCS	Initialized	—	—	—	—	Initialized	
FECS	Initialized	—	—	—	—	Initialized	
FKEY	Initialized	—	—	—	—	Initialized	
FTDAR	Initialized	—	—	—	—	Initialized	
ICCRA_0	Initialized	—	—	—	—	Initialized	I
ICCRB_0	Initialized	—	—	—	—	Initialized	
ICMR_0	Initialized	—	—	—	—	Initialized	
ICIER_0	Initialized	—	—	—	—	Initialized	
ICSR_0	Initialized	—	—	—	—	Initialized	
SAR_0	Initialized	—	—	—	—	Initialized	
ICDRT_0	Initialized	—	—	—	—	Initialized	
ICDRR_0	Initialized	—	—	—	—	Initialized	
ICCRA_1	Initialized	—	—	—	—	Initialized	I
ICCRB_1	Initialized	—	—	—	—	Initialized	
ICMR_1	Initialized	—	—	—	—	Initialized	
ICIER_1	Initialized	—	—	—	—	Initialized	
ICSR_1	Initialized	—	—	—	—	Initialized	
SAR_1	Initialized	—	—	—	—	Initialized	
ICDRT_1	Initialized	—	—	—	—	Initialized	
ICDRR_1	Initialized	—	—	—	—	Initialized	

TCORB_3	Initialized	—	—	—	—	Initialized
TCNT_2	Initialized	—	—	—	—	Initialized
TCNT_3	Initialized	—	—	—	—	Initialized
TCCR_2	Initialized	—	—	—	—	Initialized
TCCR_3	Initialized	—	—	—	—	Initialized
TCR_4	Initialized	—	—	—	—	Initialized
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
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

DTCCR	Initialized	—	—	—	—	Initialized	
INTCR	Initialized	—	—	—	—	Initialized	
CPUPCR	Initialized	—	—	—	—	Initialized	
IER	Initialized	—	—	—	—	Initialized	
ISR	Initialized	—	—	—	—	Initialized	
PORT1	—	—	—	—	—	—	I
PORT2	—	—	—	—	—	—	
PORT3	—	—	—	—	—	—	
PORT5	—	—	—	—	—	—	
PORT6	—	—	—	—	—	—	
PORTA	—	—	—	—	—	—	
PORTB	—	—	—	—	—	—	
PORTC	—	—	—	—	—	—	
PORTD	—	—	—	—	—	—	
PORTE	—	—	—	—	—	—	
PORTF	—	—	—	—	—	—	
P1DR	Initialized	—	—	—	—	Initialized	
P2DR	Initialized	—	—	—	—	Initialized	
P3DR	Initialized	—	—	—	—	Initialized	
P6DR	Initialized	—	—	—	—	Initialized	
PADR	Initialized	—	—	—	—	Initialized	
PBDR	Initialized	—	—	—	—	Initialized	
PCDR	Initialized	—	—	—	—	Initialized	

SSR_2	Initialized	Initialized	—	Initialized	Initialized	Initialized
RDR_2	Initialized	Initialized	—	Initialized	Initialized	Initialized
SCMR_2	Initialized	—	—	—	—	Initialized
DADR0	Initialized	—	—	—	—	Initialized
DADR1	Initialized	—	—	—	—	Initialized
DACR01	Initialized	—	—	—	—	Initialized
PCR	Initialized	—	—	—	—	Initialized
PMR	Initialized	—	—	—	—	Initialized
NDERH	Initialized	—	—	—	—	Initialized
NDERL	Initialized	—	—	—	—	Initialized
PODRH	Initialized	—	—	—	—	Initialized
PODRL	Initialized	—	—	—	—	Initialized
NDRH	Initialized	—	—	—	—	Initialized
NDRL	Initialized	—	—	—	—	Initialized
SMR_0	Initialized	—	—	—	—	Initialized
BRR_0	Initialized	—	—	—	—	Initialized
SCR_0	Initialized	—	—	—	—	Initialized
TDR_0	Initialized	Initialized	—	Initialized	Initialized	Initialized
SSR_0	Initialized	Initialized	—	Initialized	Initialized	Initialized
RDR_0	Initialized	Initialized	—	Initialized	Initialized	Initialized
SCMR_0	Initialized	—	—	—	—	Initialized

ADDRA	Initialized	—	—	—	—	Initialized	A
ADDRB	Initialized	—	—	—	—	Initialized	
ADDRC	Initialized	—	—	—	—	Initialized	
ADDRD	Initialized	—	—	—	—	Initialized	
ADDRE	Initialized	—	—	—	—	Initialized	
ADDRF	Initialized	—	—	—	—	Initialized	
ADDRG	Initialized	—	—	—	—	Initialized	
ADDRH	Initialized	—	—	—	—	Initialized	
ADCSR	Initialized	—	—	—	—	Initialized	
ADCR	Initialized	—	—	—	—	Initialized	
TCSR	Initialized	—	—	—	—	Initialized	W
TCNT	Initialized	—	—	—	—	Initialized	
RSTCSR	Initialized	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	Initialized	T
TCR_1	Initialized	—	—	—	—	Initialized	T
TCSR_0	Initialized	—	—	—	—	Initialized	T
TCSR_1	Initialized	—	—	—	—	Initialized	T
TCORA_0	Initialized	—	—	—	—	Initialized	T
TCORA_1	Initialized	—	—	—	—	Initialized	T
TCORB_0	Initialized	—	—	—	—	Initialized	T
TCORB_1	Initialized	—	—	—	—	Initialized	T

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
TMDR_1	Initialized	—	—	—	—	Initialized
TIOR_1	Initialized	—	—	—	—	Initialized
TIER_1	Initialized	—	—	—	—	Initialized
TSR_1	Initialized	—	—	—	—	Initialized
TCNT_1	Initialized	—	—	—	—	Initialized
TGRA_1	Initialized	—	—	—	—	Initialized
TGRB_1	Initialized	—	—	—	—	Initialized

TGRB_2	Initialized	—	—	—	—	Initialized
TCR_3	Initialized	—	—	—	—	Initialized
TMDR_3	Initialized	—	—	—	—	Initialized
TIORH_3	Initialized	—	—	—	—	Initialized
TIORL_3	Initialized	—	—	—	—	Initialized
TIER_3	Initialized	—	—	—	—	Initialized
TSR_3	Initialized	—	—	—	—	Initialized
TCNT_3	Initialized	—	—	—	—	Initialized
TGRA_3	Initialized	—	—	—	—	Initialized
TGRB_3	Initialized	—	—	—	—	Initialized
TGRC_3	Initialized	—	—	—	—	Initialized
TGRD_3	Initialized	—	—	—	—	Initialized

Input voltage (port 5)	V_{in}	–0.3 to AV_{CC} +0.3
Reference power supply voltage	V_{ref}	–0.3 to AV_{CC} +0.3
Analog power supply voltage	AV_{CC}	–0.3 to +4.6
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3
Operating temperature	T_{opr}	Regular specifications: –20 to +75*
		Wide-range specifications: –40 to +85*
Storage temperature	T_{stg}	–55 to +125

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exc

Note: * The operating temperature range during programming/erasing of the flash me 0°C to +75°C for regular specifications and 0°C to +85°C for wide-range spec

trigger input voltage	TMR input pin, port 2, port 3	VT^+	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.06$	—	—	V	
	Port 5*[2]	VT^-	$AV_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$AV_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$AV_{CC} \times 0.06$	—	—	V	
Input high voltage (except Schmitt trigger input pin)	MD, $\overline{\text{RES}}$, $\overline{\text{STBY}}$, EMLE, NMI	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL Other input pins		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$		
	Port 5		$AV_{CC} \times 0.7$	—	$AV_{CC} + 0.3$		
Input low voltage (except Schmitt trigger input pin)	MD, $\overline{\text{RES}}$, $\overline{\text{STBY}}$, EMLE	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL, NMI		–0.3	—	$V_{CC} \times 0.2$		
	Other input pins		–0.3	—	$V_{CC} \times 0.2$		
Output high voltage	All output pins	V_{OH}	$V_{CC} - 0.5$	—	—	V	I_{OH} =
			$V_{CC} - 1.0$	—	—		I_{OH} =
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} =
	Port 3		—	—	1.0		I_{OL} =
Input leakage current	$\overline{\text{RES}}$	$\lvert I_{in} \rvert$	—	—	10.0	μA	V_{in} = V_{CC} –
	MD, $\overline{\text{STBY}}$, EMLE, NMI		—	—	1.0		
	Port 5		—	—	1.0		V_{in} = AV_{CC}

RENESAS

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input pull-up MOS current	Ports D to F, H, I	$-I_p$	10		300	μA	V_{CC} 3.6 V_{in} =
Input capacitance	All input pins	C_{in}	—	—	15	pF	V_{in} = f = T_a =
Current consumption*3	Normal operation	I_{CC}*5	—	75	125	mA	f = 5
	Sleep mode		—	70	90		
	Subclock operation		—	5.0	10		32.7 crys reso use
	Standby mode*4		—	50	100	μA	T_a ≤
			—	—	300		50°
	All-module-clock-stop mode*6		—	33	45	mA	
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	1.0 (3.0 V)	2.0	mA	
	Standby for A/D and D/A conversion		—	1.0	20	μA	
Reference power supply current	During A/D and D/A conversion	AI_{CC}	—	1.5 (3.0 V)	3.0	mA	
	Standby for A/D and D/A conversion		—	1.5	5.0	μA	
RAM standby voltage		V_{RAM}	2.5	—	—	V	

$I_{CC}max$ = 35 (mA) + 1.8 (mA/MHz) × f (normal operation)

$I_{CC}max$ = 30 (mA) + 1.2 (mA/MHz) × f (sleep mode)

6. The values are for reference.
7. This can be applied when the $\overline{RES}$ pin is held low at power-on.

Table 25.3 Permissible Output Currents

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V
V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V*, T_a = –20°C to +75°C (regular specific
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.
Permissible output low current (per pin)	Output pins except port 3	I_{OL}	—	—	2.0
Permissible output low current (per pin)	Port 3	I_{OL}	—	—	10
Permissible output low current (total)	Total of all output pins	ΣI_{OL}	—	—	80
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	40

Caution: To protect the LSI's reliability, do not exceed the output current values in table

Note: * When the A/D and D/A converters are not used, the AV_{CC}, V_{ref}, and AV_{SS} pins sl be open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

Figure 25.1 Output Load Circuit

25.3.1 Clock Timing

Table 25.4 Clock Timing

Conditions: V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0
V_{SS} = PLLV_{SS} = DrV_{SS} = AV_{SS} = 0 V, Iϕ = 8 MHz to 50 MHz,
Bϕ = 8 MHz to 50 MHz, Pϕ = 8 MHz to 35 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 Co
Clock cycle time	t_{cyc}	20	125	ns	Figure 2
Clock high pulse width	t_{CH}	5	—	ns	
Clock low pulse width	t_{CL}	5	—	ns	
Clock rising time	t_{Cr}	—	5	ns	
Clock falling time	t_{Cf}	—	5	ns	
Oscillation settling time after reset (crystal)	t_{OSC1}	10	—	ms	Figure 2
Oscillation settling time after leaving software standby mode (crystal)	t_{OSC2}	10	—	ms	Figure 2

Subclock cycle time	t_{SUB}	30.5	30.5	μs

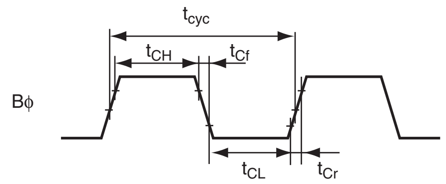


Figure 25.2 External Bus Clock Timing

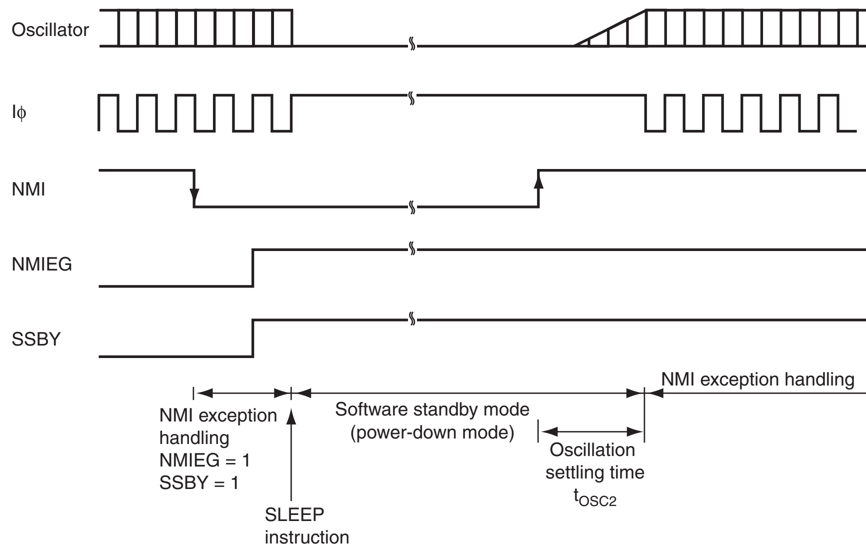


Figure 25.3 Oscillation Settling Timing after Software Standby Mode

Figure 25.4 Oscillation Settling Timing

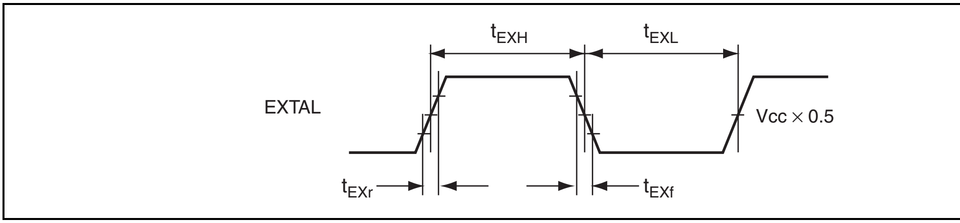


Figure 25.5 External Input Clock Timing

$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 25
NMI hold time	t_{NMIH}	10	—	ns	
NMI pulse width (after leaving software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (after leaving software standby mode)	t_{IRQW}	200	—	ns	

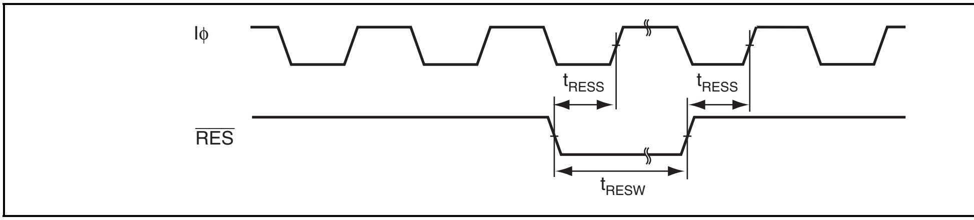


Figure 25.6 Reset Input Timing

Note: * SSIER must be set to cancel software standby mode.

Figure 25.7 Interrupt Input Timing

25.3.3 Bus Timing

Table 25.6 Bus Timing (1)

Conditions: V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0
V_{SS} = PLLV_{SS} = DrV_{SS} = AV_{SS} = 0 V, Bϕ = 8 MHz to 50 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 Cond
Address delay time	t_{AD}	—	18	ns	Figure 25.34
Address setup time 1	t_{AS1}	$0.5 \times t_{cyc} - 8$	—	ns	
Address setup time 2	t_{AS2}	$1.0 \times t_{cyc} - 8$	—	ns	
Address setup time 3	t_{AS3}	$1.5 \times t_{cyc} - 8$	—	ns	
Address setup time 4	t_{AS4}	$2.0 \times t_{cyc} - 8$	—	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	

Read data hold time 2	t_{RDH2}	0	—	ns
Read data access time 2	t_{AC2}	—	$1.5 \times t_{cyc} - 30$	ns
Read data access time 4	t_{AC4}	—	$2.5 \times t_{cyc} - 30$	ns
Read data access time 5	t_{AC5}	—	$1.0 \times t_{cyc} - 30$	ns
Read data access time 6	t_{AC6}	—	$2.0 \times t_{cyc} - 30$	ns
Read data access time (from address) 1	t_{AA1}	—	$1.0 \times t_{cyc} - 30$	ns
Read data access time (from address) 2	t_{AA2}	—	$1.5 \times t_{cyc} - 30$	ns
Read data access time (from address) 3	t_{AA3}	—	$2.0 \times t_{cyc} - 30$	ns
Read data access time (from address) 4	t_{AA4}	—	$2.5 \times t_{cyc} - 30$	ns
Read data access time (from address) 5	t_{AA5}	—	$3.0 \times t_{cyc} - 30$	ns

$\overline{WR}$ 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 3	t_{WDH3}	$1.5 \times t_{cyc} - 8$	—	ns	
Byte control delay time	t_{UBD}	—	15	ns	Figure 25.14
Byte control pulse width 1	t_{UBW1}	—	$1.0 \times t_{cyc} - 15$	ns	Figure
Byte control pulse width 2	t_{UBW2}	—	$2.0 \times t_{cyc} - 15$	ns	Figure
Multiplexed address delay time 1	t_{MAD1}	—	18	ns	Figure 25.18
Multiplexed address hold time	t_{MAH}	$1.0 \times t_{cyc} - 15$	—	ns	
Multiplexed address setup time 1	t_{MAS1}	$0.5 \times t_{cyc} - 15$	—	ns	
Multiplexed address setup time 2	t_{MAS2}	$1.5 \times t_{cyc} - 15$	—	ns	
Address hold delay time	t_{AHD}	—	15	ns	
Address hold pulse width 1	t_{AHW1}	$1.0 \times t_{cyc} - 15$	—	ns	
Address hold pulse width 2	t_{AHW2}	$2.0 \times t_{cyc} - 15$	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	15	—	ns	Figure 25.18
$\overline{WAIT}$ hold time	t_{WTH}	5.0	—	ns	
$\overline{BREQ}$ setup time	t_{BREQS}	20	—	ns	Figure
$\overline{BACK}$ delay time	t_{BACD}	—	15	ns	
Bus floating time	t_{BZD}	—	30	ns	
$\overline{BREQO}$ delay time	t_{BRQOD}	—	15	ns	Figure
$\overline{BS}$ delay time	t_{BSD}	1.0	15	ns	Figure 25.9, 25.14
RD/$\overline{WR}$ delay time	t_{RWD}	—	15	ns	

Read data access time 3	t_{AC3}	—	$2.0 \times t_{cyc} - 20$	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
Write data hold time 2	t_{WDH2}	$1.0 \times t_{cyc} - 8$	—	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
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
$\overline{\text{CAS}}$ delay time 1	t_{CASD1}	—	15	ns
$\overline{\text{CAS}}$ delay time 2	t_{CASD2}	—	15	ns
$\overline{\text{CAS}}$ setup time 1	t_{CSR1}	$0.5 \times t_{cyc} - 10$	—	ns
$\overline{\text{CAS}}$ setup time 2	t_{CSR2}	$1.5 \times t_{cyc} - 10$	—	ns
$\overline{\text{CAS}}$ pulse width 1	t_{CASW1}	$1.0 \times t_{cyc} - 15$	—	ns
$\overline{\text{CAS}}$ pulse width 2	t_{CASW2}	$1.5 \times t_{cyc} - 15$	—	ns
$\overline{\text{CAS}}$ precharge time 1	t_{CPW1}	$1.0 \times t_{cyc} - 15$	—	ns
$\overline{\text{CAS}}$ precharge time 2	t_{CPW2}	$1.5 \times t_{cyc} - 15$	—	ns
$\overline{\text{OE}}$ delay time 1	t_{OED1}	—	15	ns
$\overline{\text{OE}}$ 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

Conditions: $V_{CC} = PLLV_{CC} = DrV_{CC}$ = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0
$V_{SS} = PLLV_{SS} = DrV_{SS} = AV_{SS}$ = 0 V, Bϕ = 8 MHz to 50 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 Cond
Read data setup time 3	t_{RDS3}	12	—	ns	Figure to 25.
Read data hold time 3	t_{RDH3}	0	—	ns	
Read data setup time 4	t_{RDS4}	12	—	ns	
Read data hold time 4	t_{RDH4}	0	—	ns	
Write data delay time 2	t_{WDD2}	—	15	ns	
Write data hold time 4	t_{WDH4}	1	—	ns	

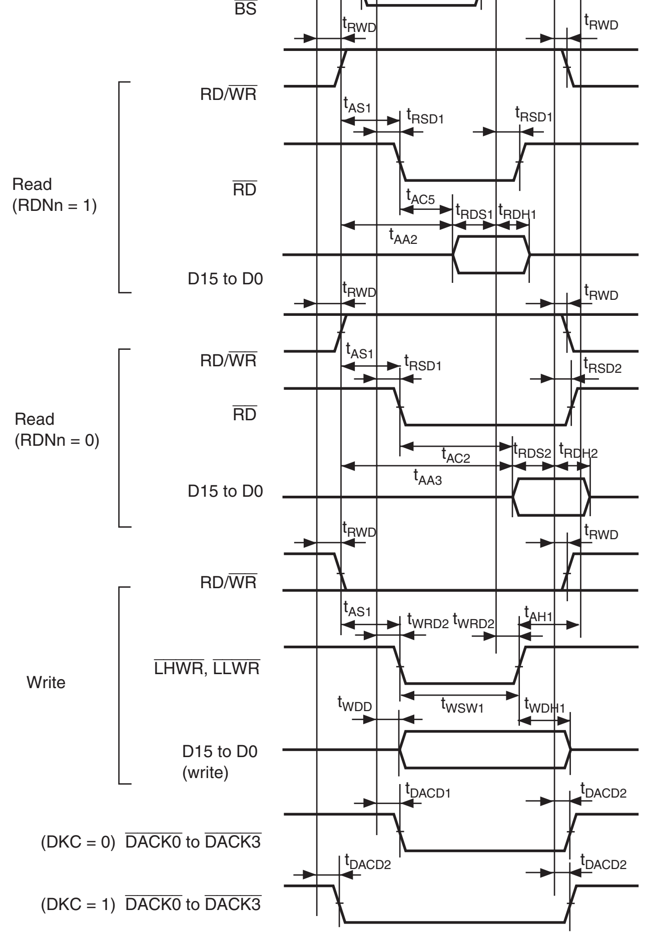


Figure 25.8 Basic Bus Timing: Two-State Access

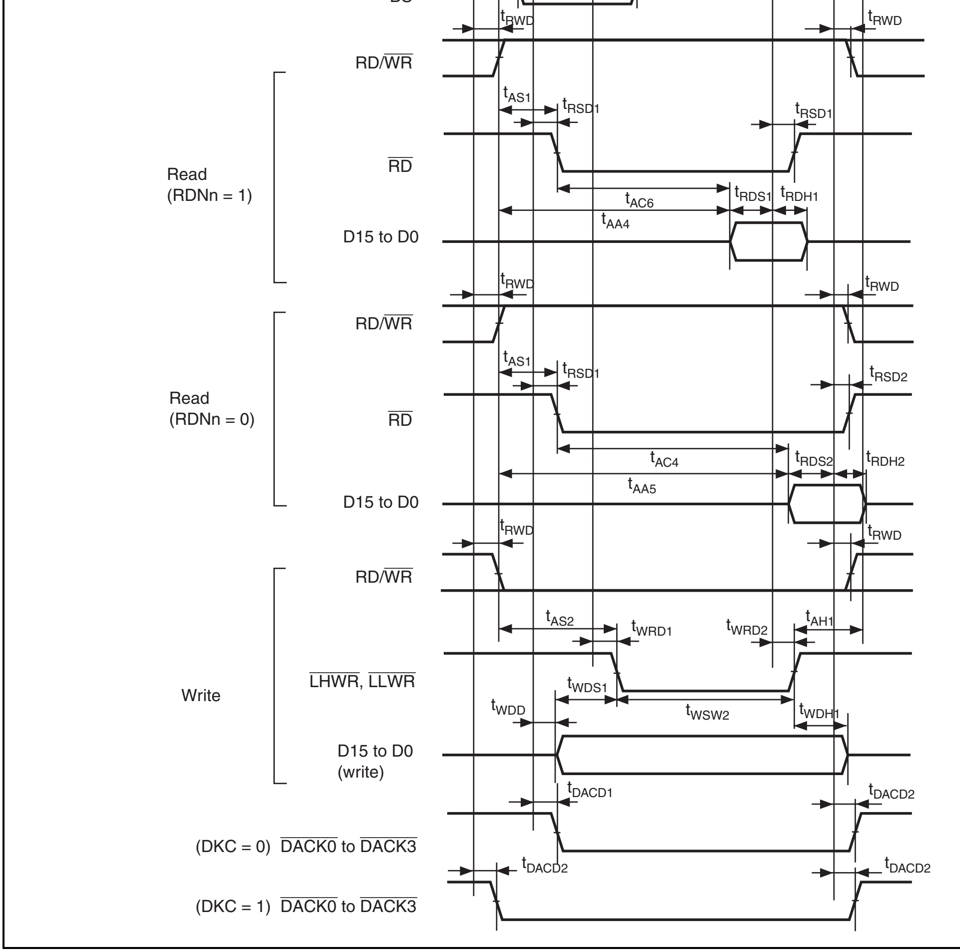


Figure 25.9 Basic Bus Timing: Three-State Access

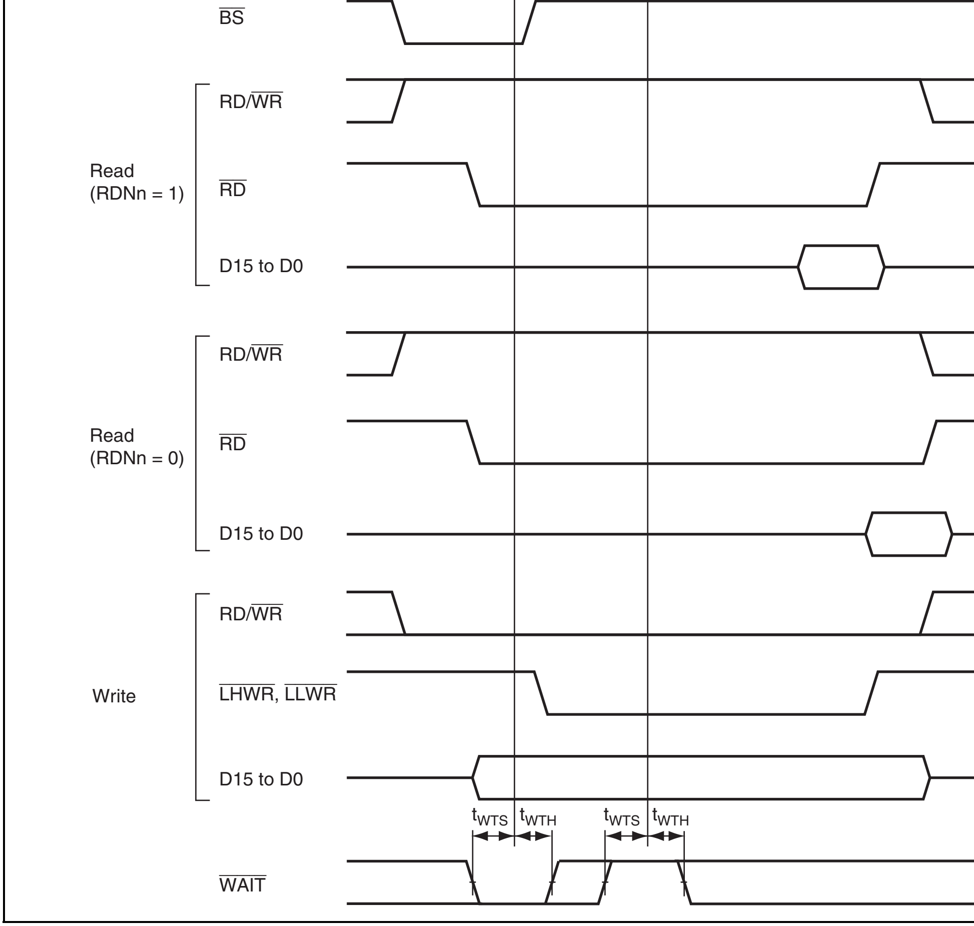


Figure 25.10 Basic Bus Timing: Three-State Access, One Wait

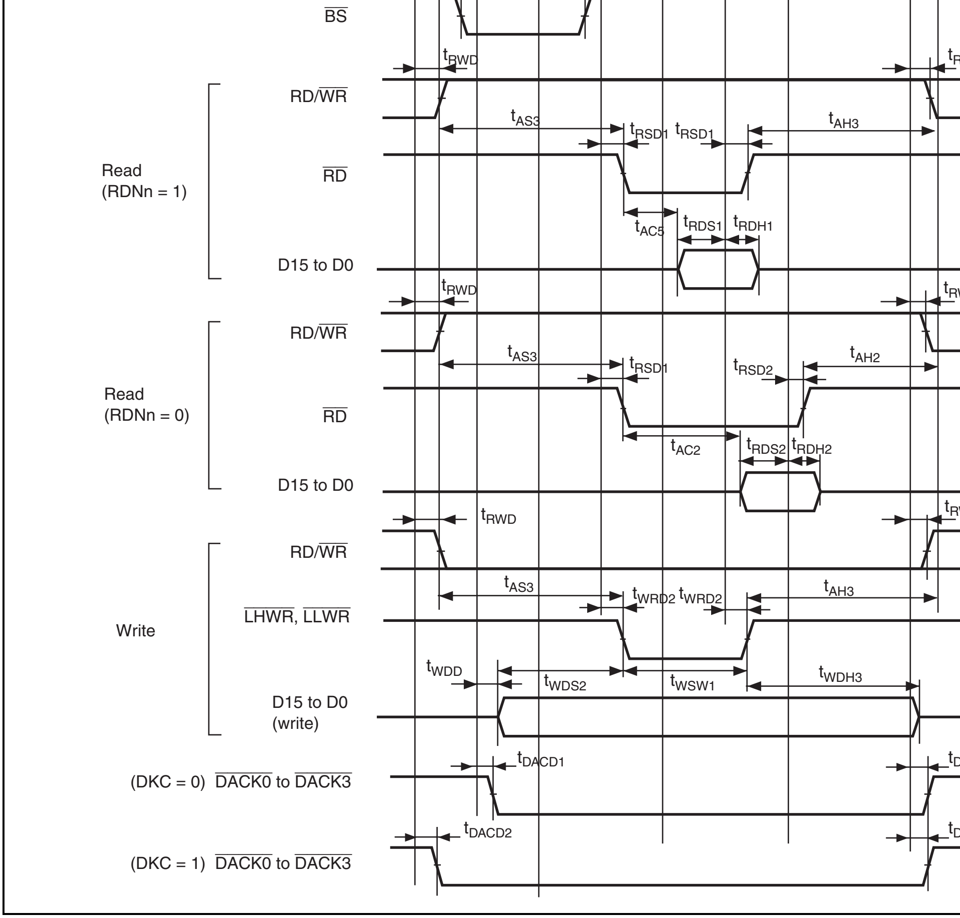


Figure 25.11 Basic Bus Timing: Two-State Access ($\overline{CS}$ Assertion Period Exter

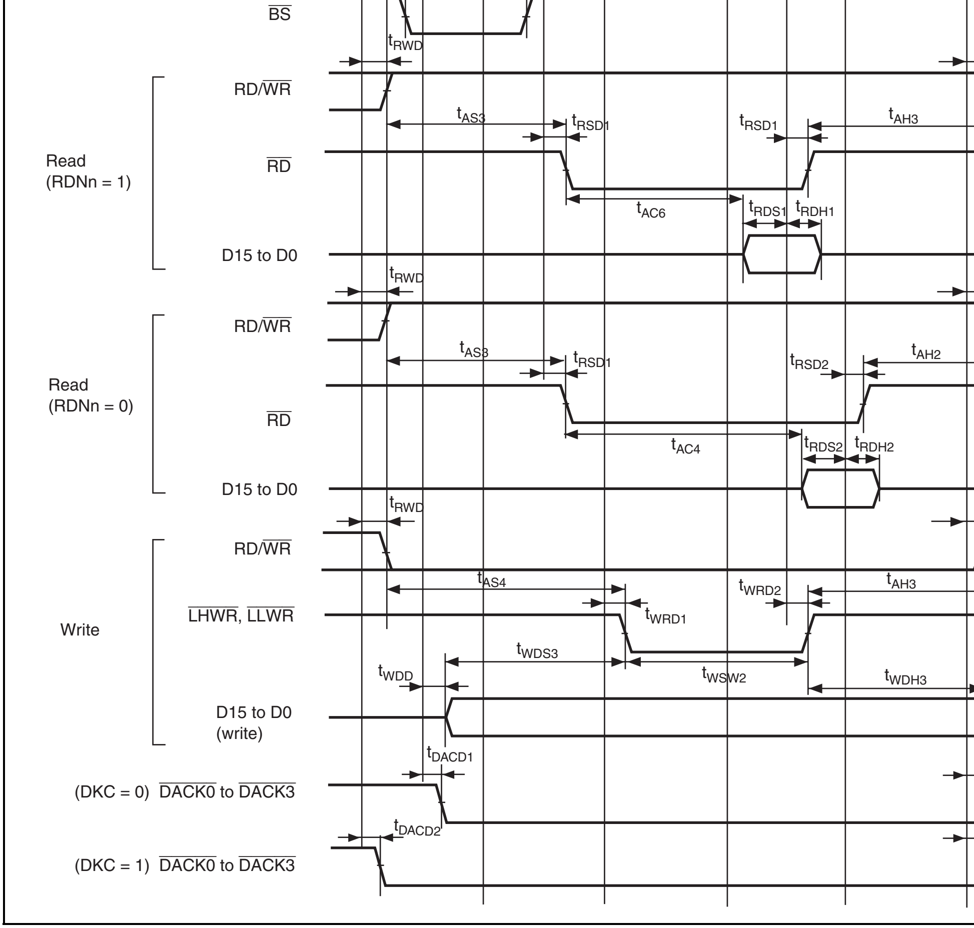


Figure 25.12 Basic Bus Timing: Three-State Access ($\overline{CS}$ Assertion Period Exter

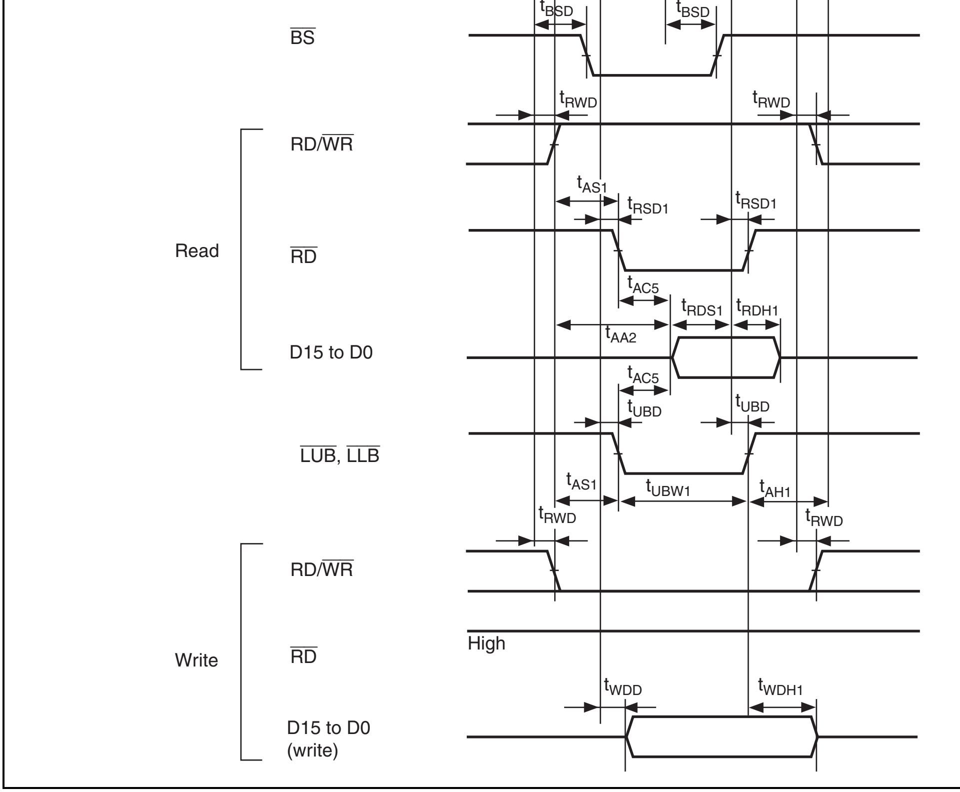


Figure 25.13 Byte Control SRAM: Two-State Read/Write Access

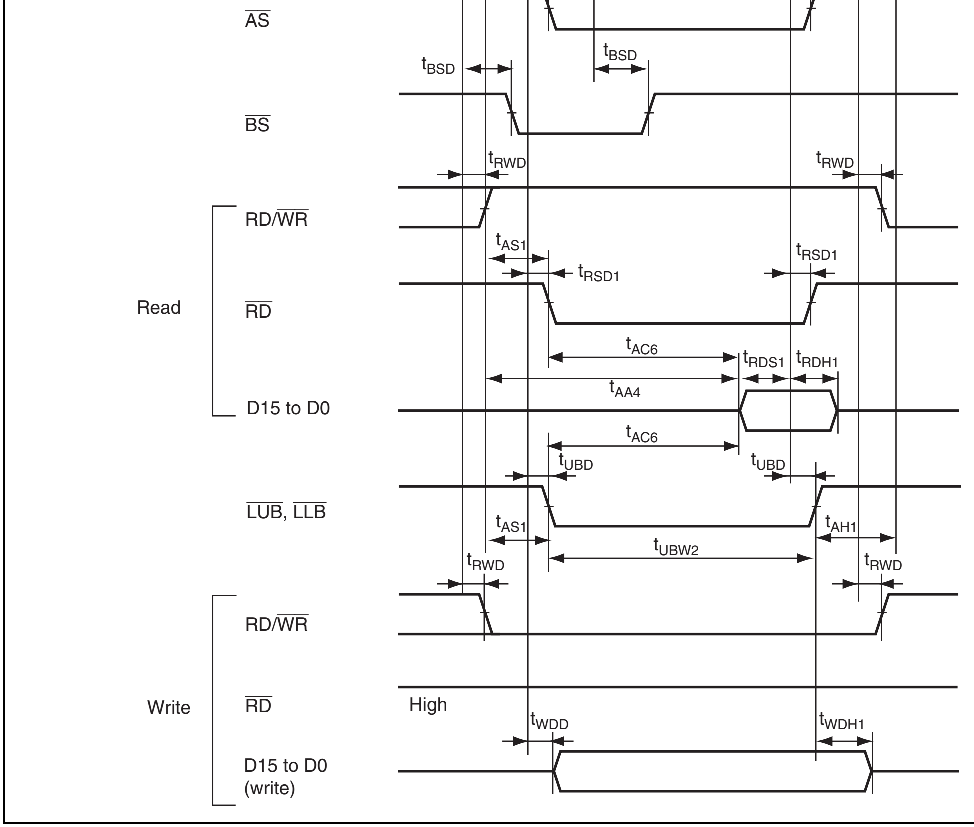


Figure 25.14 Byte Control SRAM: Three-State Read/Write Access

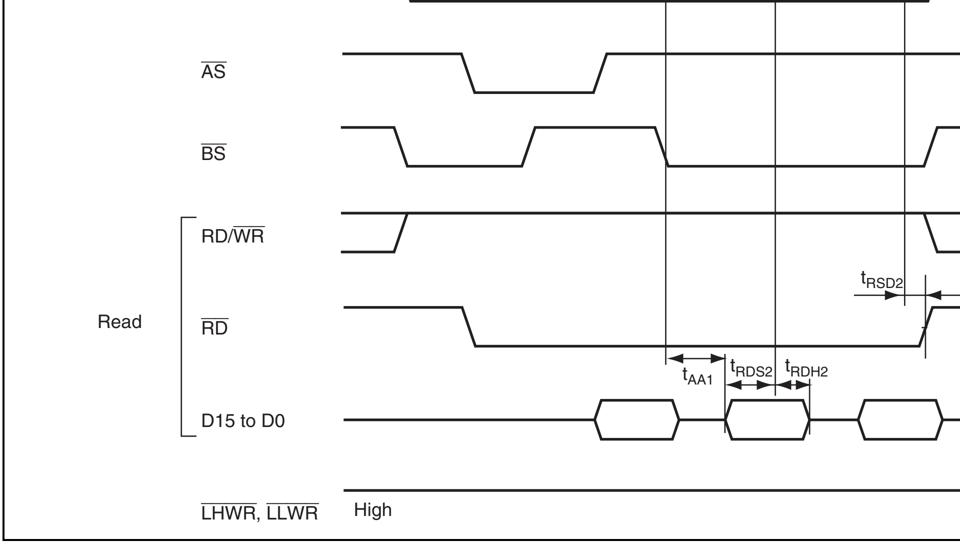


Figure 25.15 Burst ROM Access Timing: One-State Burst Access

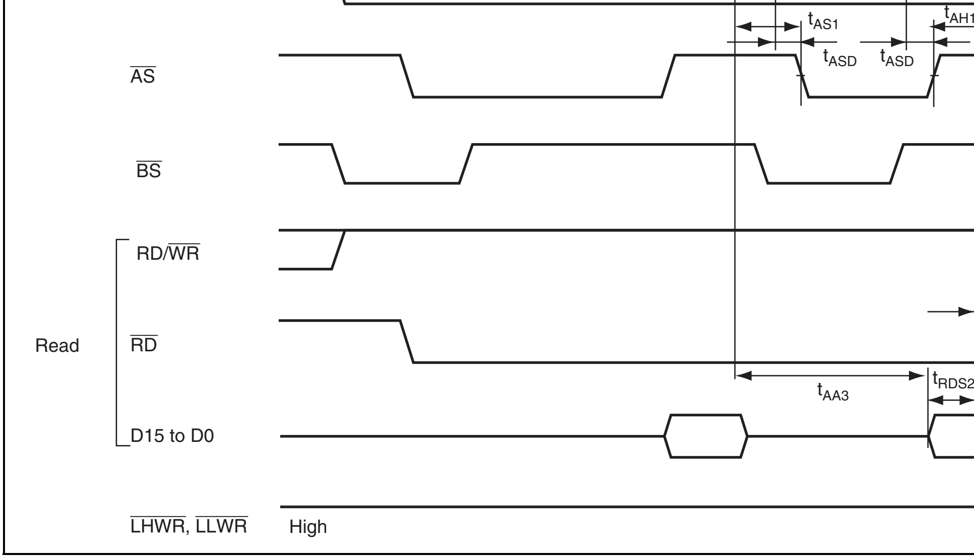


Figure 25.16 Burst ROM Access Timing: Two-State Burst Access

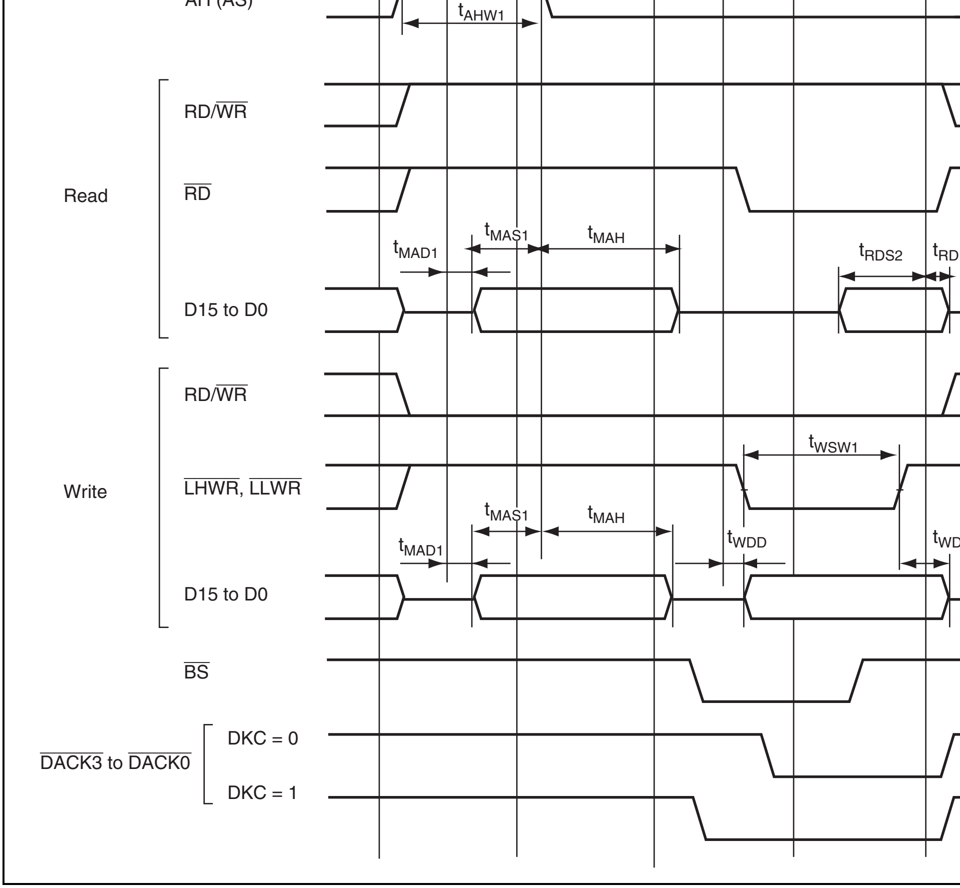


Figure 25.17 Address/Data Multiplexed Access Timing (No Wait) (Basic, Four-State Access)

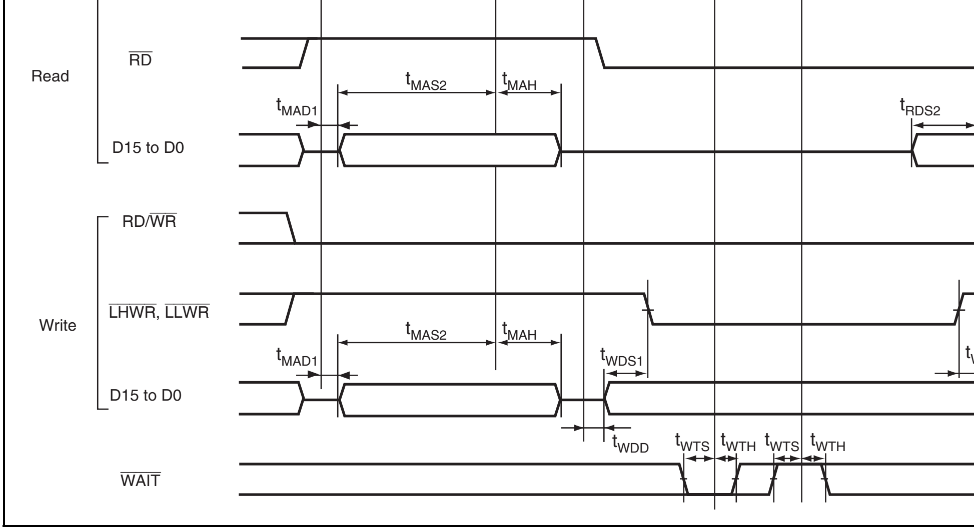


Figure 25.18 Address/Data Multiplexed Access Timing (Wait Control) (Address Cycle Program Wait × 1 + Data Cycle Program Wait × 1 + Data Cycle Pin Wait × 1)

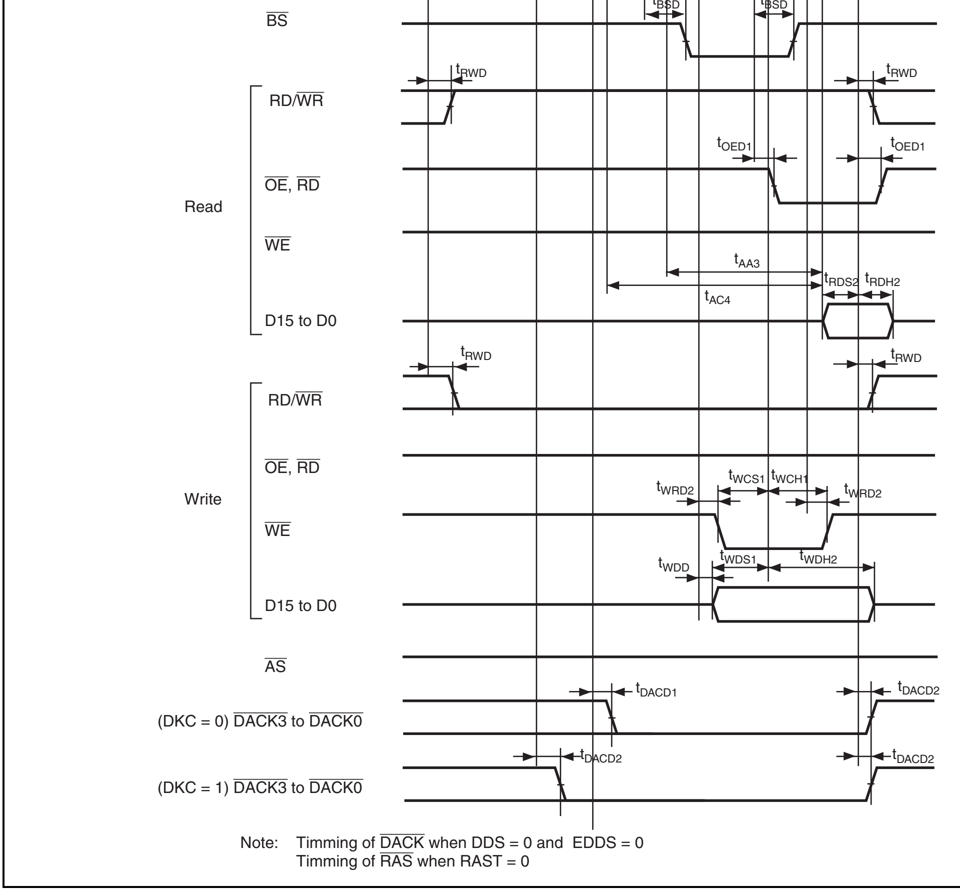


Figure 25.19 DRAM Access Timing: Two-State Access

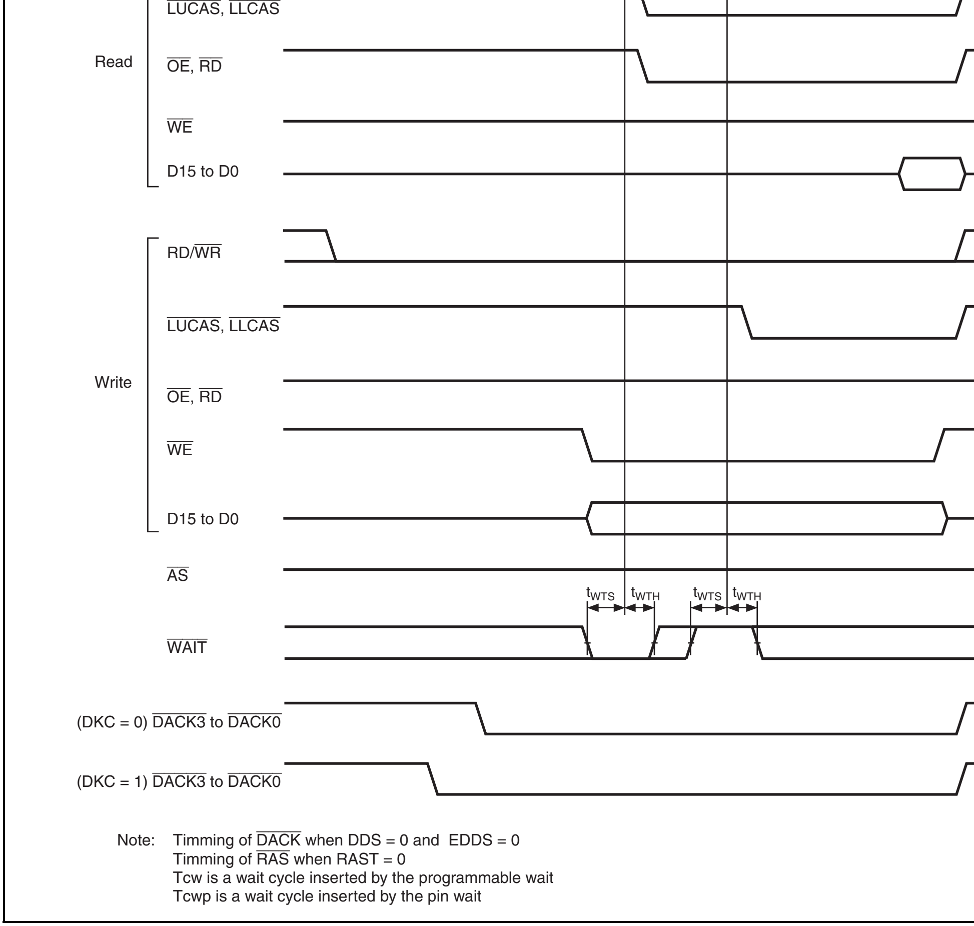
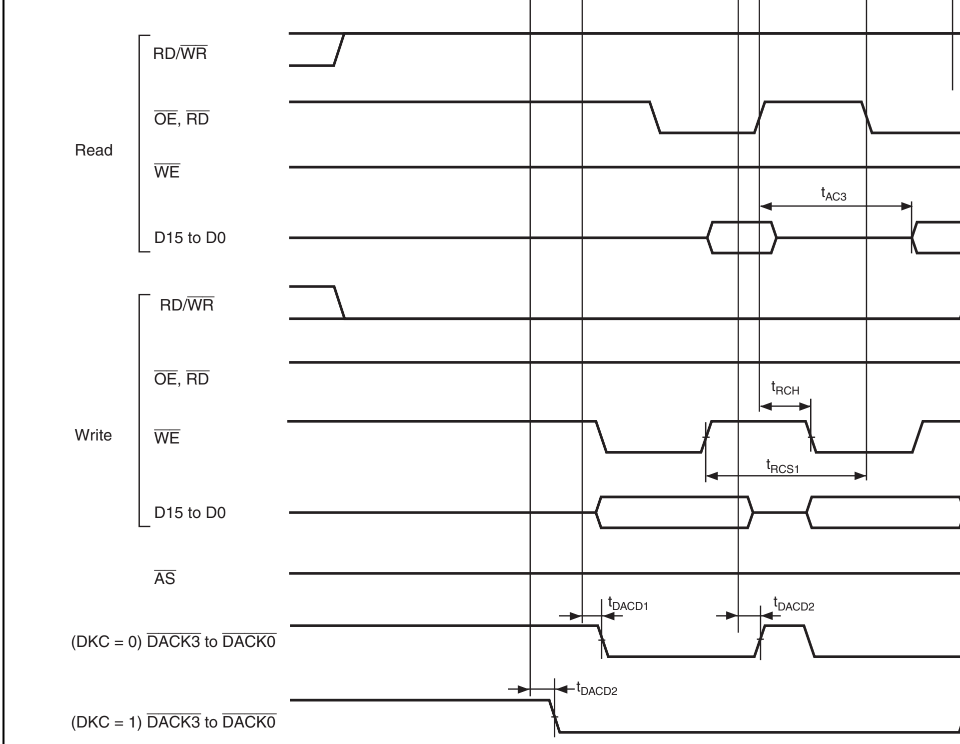
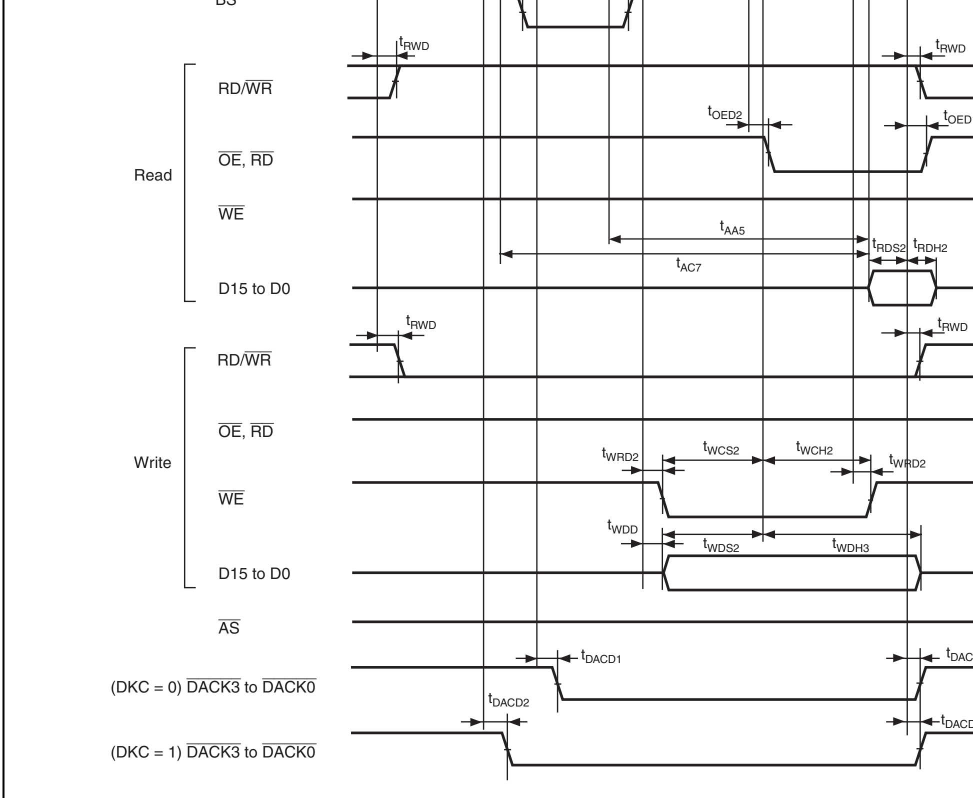


Figure 25.20 DRAM Access Timing: Two-State Access, One Wait



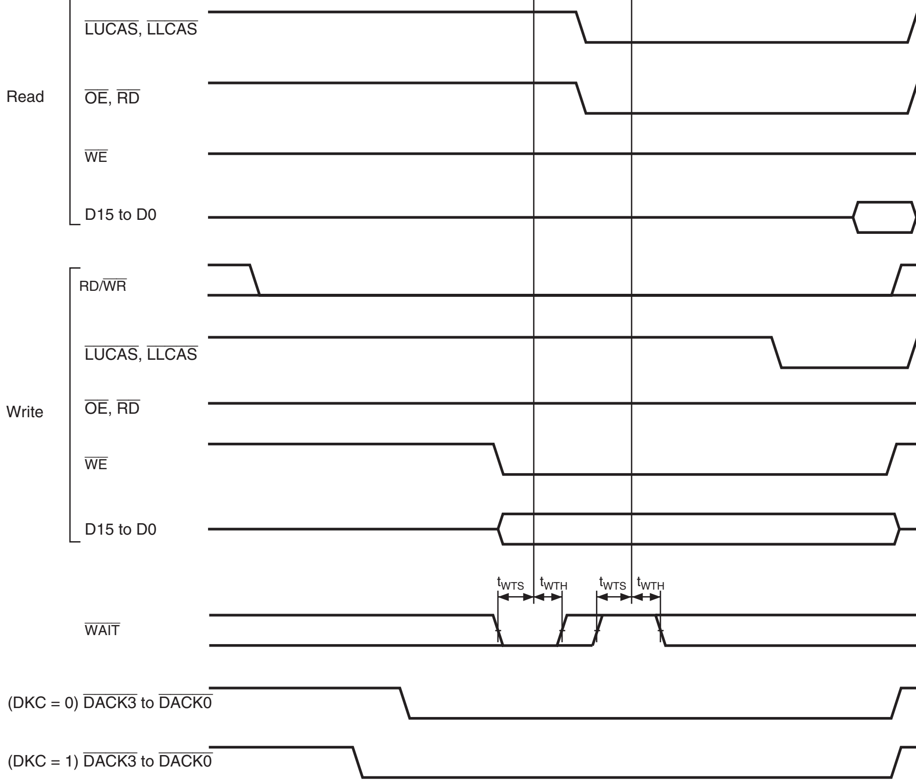
Note: Timming of $\overline{DACK}$ when DDS = 0 and EDDS = 0
Timming of $\overline{RAS}$ when RAST = 0

Figure 25.21 DRAM Access Timing: Two-State Burst Access



Note: Timming of $\overline{DACK}$ when DDS = 0 and EDDS = 0
Timming of $\overline{RAS}$ when RAST = 1

Figure 25.22 DRAM Access Timing: Three-State Access (RAST = 1)



Note: Timming of $\overline{DACK}$ when DDS = 0 and EDDS = 0
Timming of $\overline{RAS}$ when RAST = 0
Tcw is a wait cycle inserted by the programmable wait
Tcwp is a wait cycle inserted by the pin wait

Figure 25.23 DRAM Access Timing: Three-State Access, One Wait

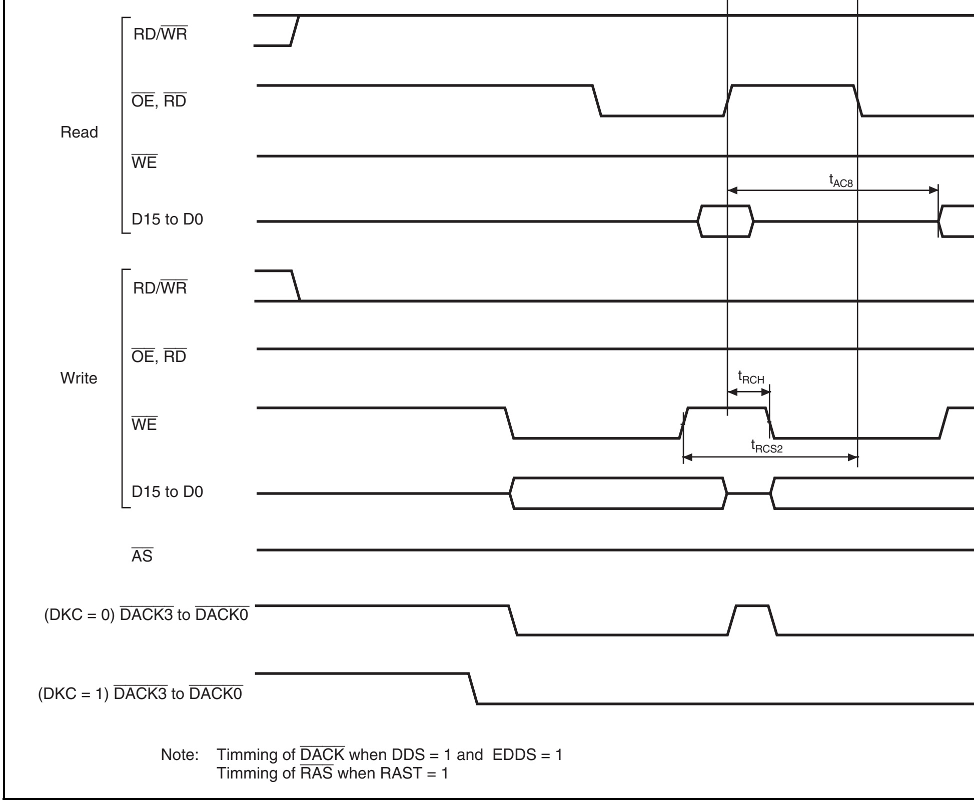


Figure 25.24 DRAM Access Timing: Three-State Burst Access

Figure 25.25   CAS Before RAS Refresh Timing

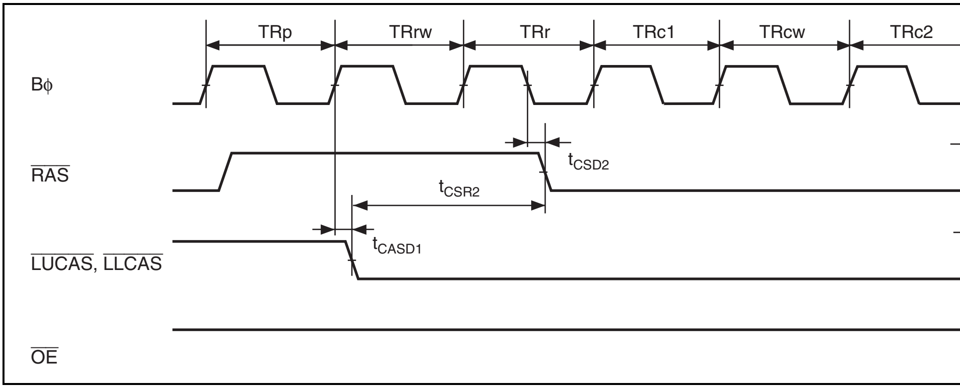


Figure 25.26   CAS Before RAS Refresh Timing (Wait Cycle Inserted)

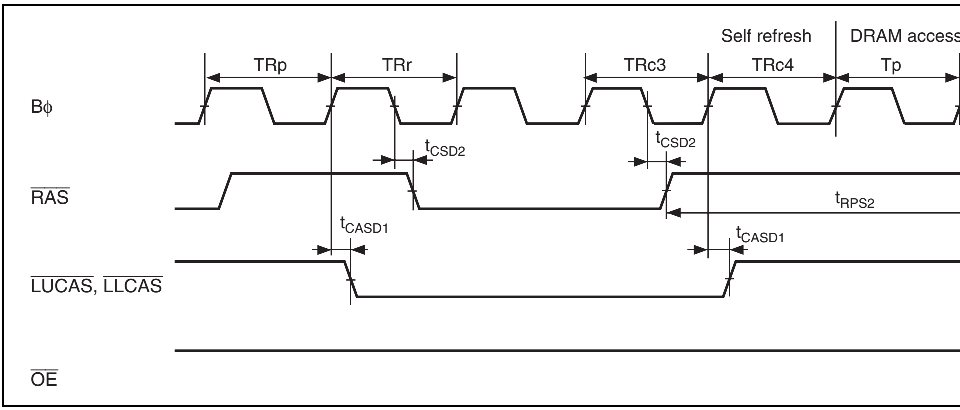


Figure 25.27   Self-Refresh Timing (After Leaving Software Standby: RAST

Figure 25.28 Self-Refresh Timing (After Leaving Software Standby: RAST =

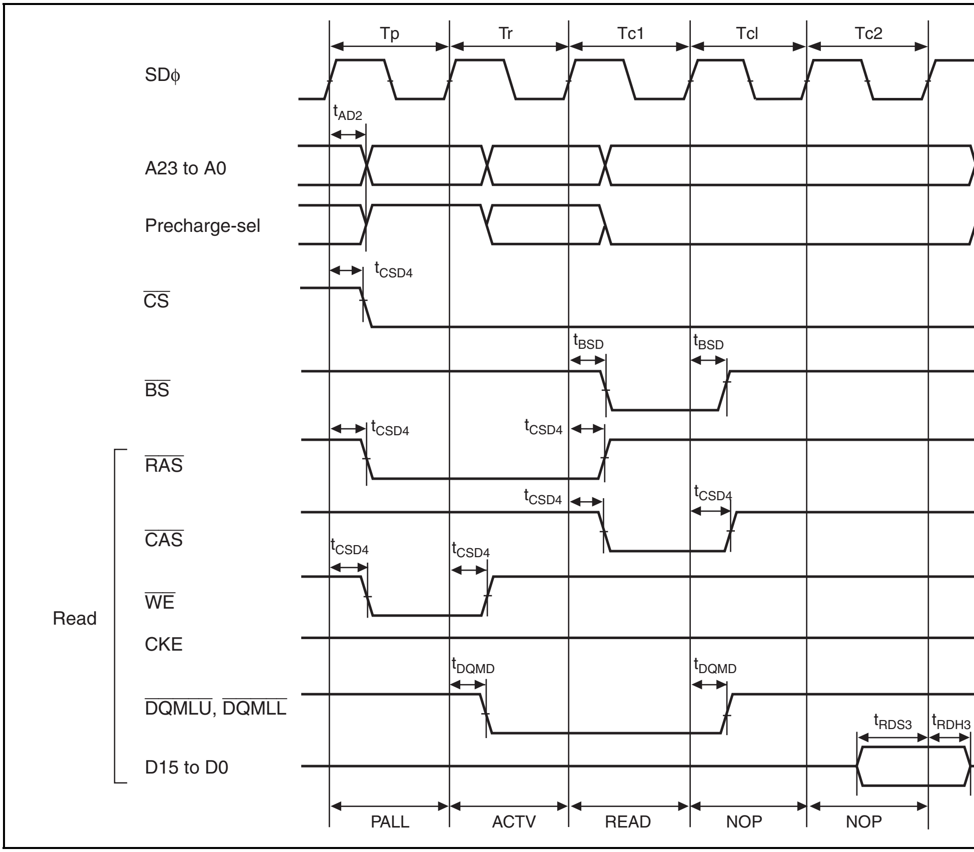


Figure 25.29 Synchronous DRAM Basic Read Access Timing (CAS Latency

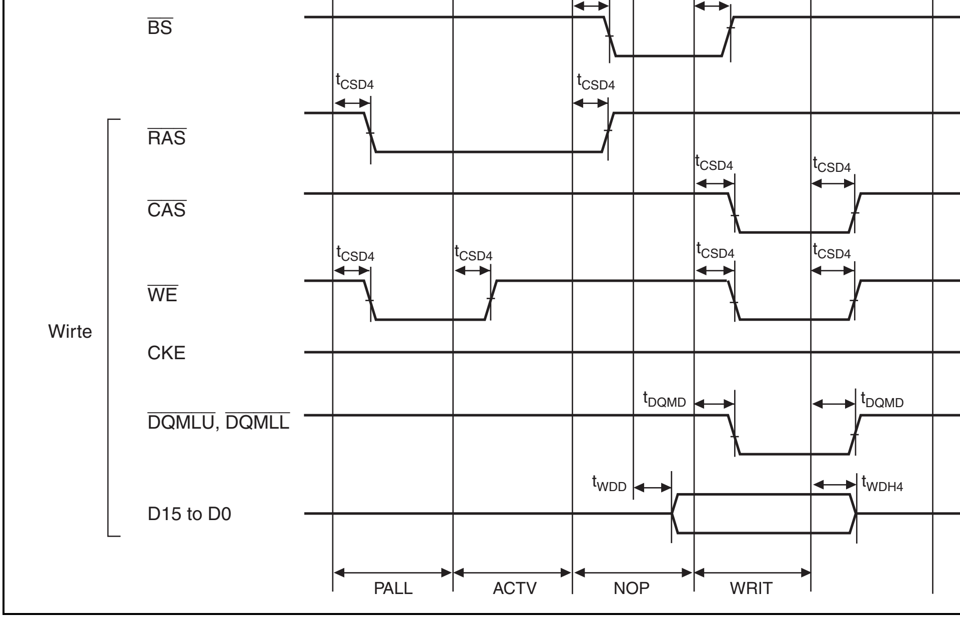


Figure 25.30 Synchronous DRAM Basic Write Access Timing (CAS Latenc

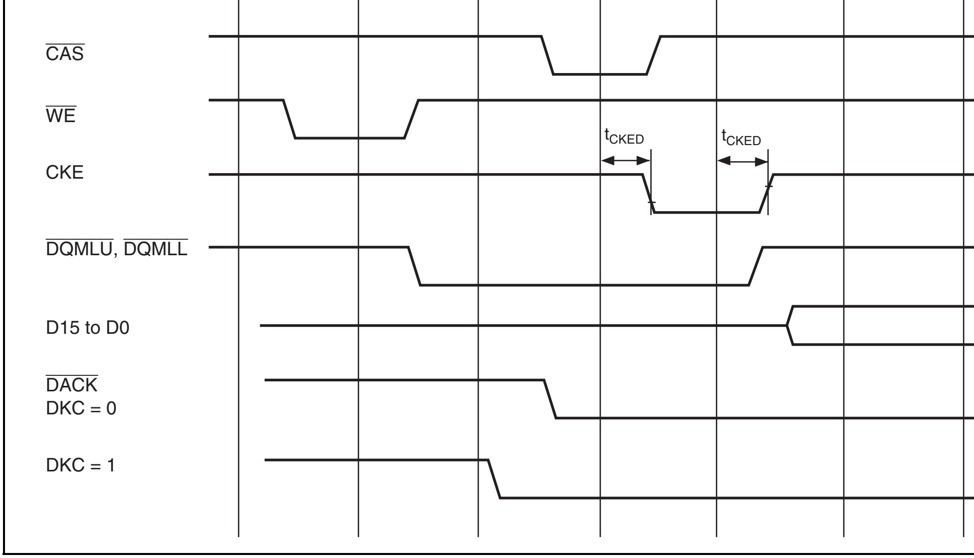


Figure 25.31 Extended Read Data Cycle (CAS Latency 2)

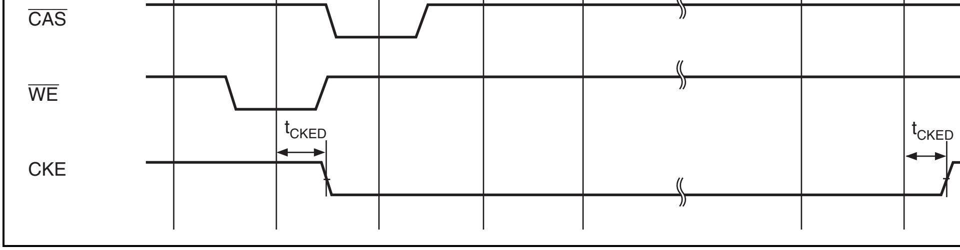


Figure 25.32 Synchronous DRAM Self-Refresh Timing

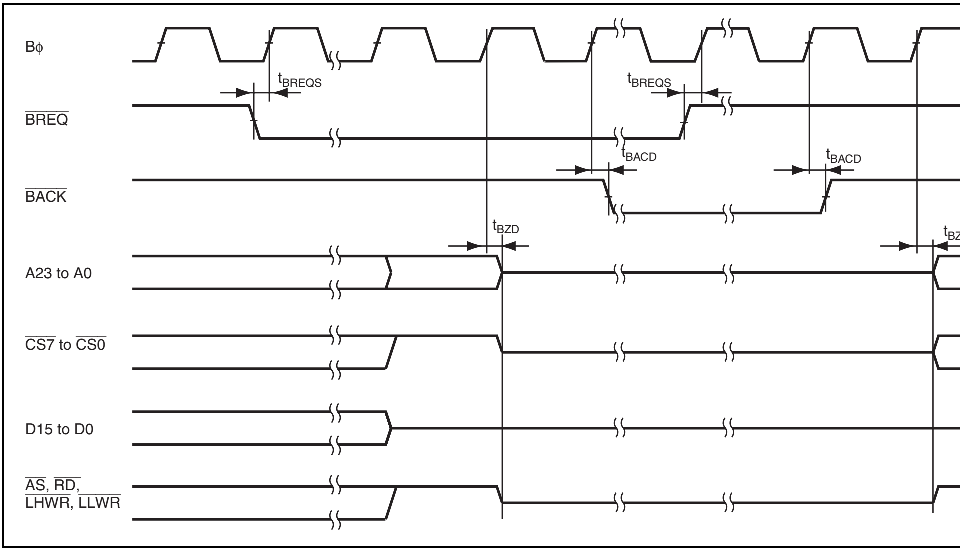


Figure 25.33 External Bus Release Timing

25.3.4 DMAC Timing

Table 25.7 DMAC Timing

Conditions: V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V
V_{SS} = PLLV_{SS} = DrV_{SS} = AV_{SS} = 0 V, Bϕ = 8 MHz to 50 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 Con
$\overline{\text{DREQ}}$ setup time	t_{DRQS}	20	—	ns	Figure 25
$\overline{\text{DREQ}}$ hold time	t_{DRQH}	5	—	ns	
$\overline{\text{TEND}}$ delay time	t_{TED}	—	15	ns	Figure 25
$\overline{\text{DACK}}$ delay time 1	t_{DACD1}	—	15	ns	Figures 25 25.38
$\overline{\text{DACK}}$ delay time 2	t_{DACD2}	—	15	ns	

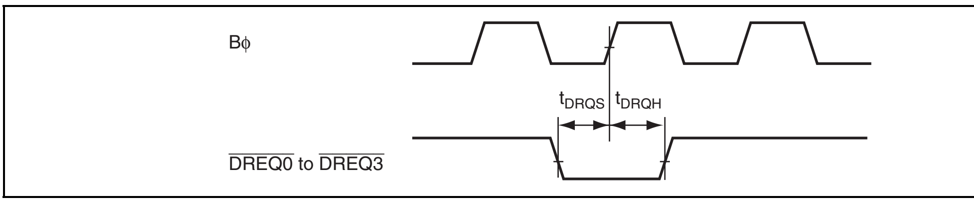


Figure 25.35 DMAC ($\overline{\text{DREQ}}$) Input Timing

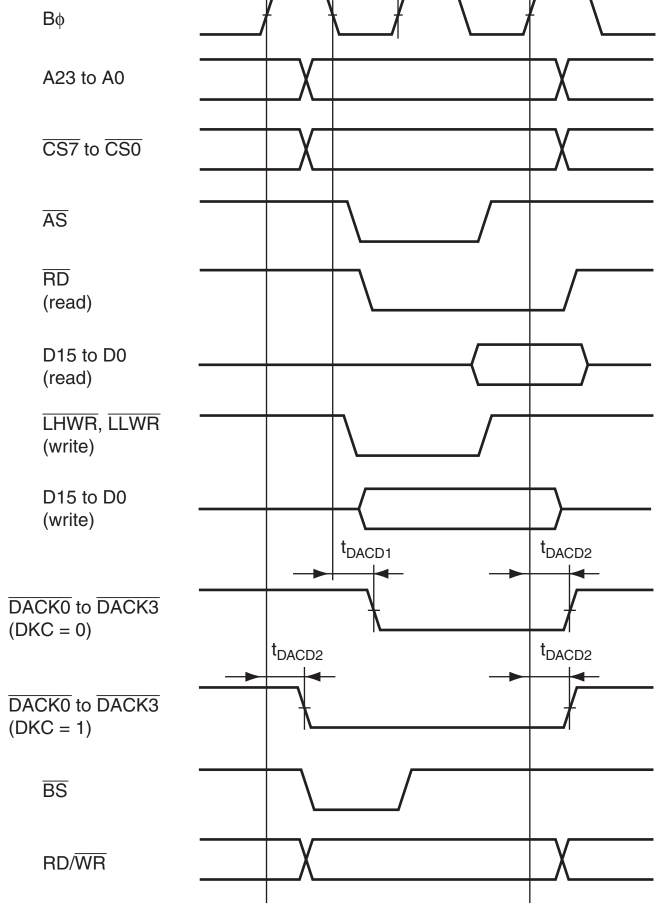


Figure 25.37 DMAC Single-Address Transfer Timing: Two-State Access

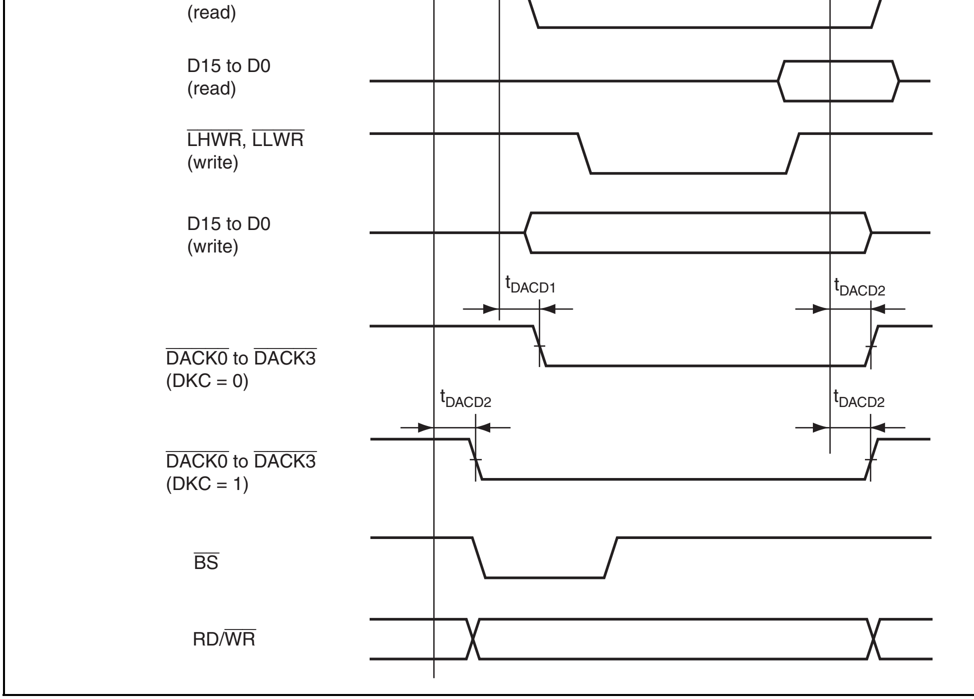


Figure 25.38 DMAC Single-Address Transfer Timing: Three-State Access

	Input data setup time		t_{PRS}	25	—	ns	
	Input data hold time		t_{PRH}	25	—	ns	
TPU	Timer output delay time		t_{TOCD}	—	40	ns	Figure
	Timer input setup time		t_{TICS}	25	—	ns	
	Timer clock input setup time		t_{TCKS}	25	—	ns	Figure
	Timer clock pulse width	Single-edge setting	t_{TCKWH}	1.5	—	t_{cyc}	
		Both-edge setting	t_{TCKWL}	2.5	—	t_{cyc}	
PPG	Pulse output delay time		t_{POD}	—	40	ns	Figure
8-bit timer	Timer output delay time		t_{TMOD}	—	40	ns	Figure
	Timer reset input setup time		t_{TMRS}	25	—	ns	Figure
	Timer clock input setup time		t_{TMCS}	25	—	ns	Figure
	Timer clock pulse width	Single-edge setting	t_{TMCWH}	1.5	—	t_{cyc}	
		Both-edge setting	t_{TMCWL}	2.5	—	t_{cyc}	
WDT	Overflow output delay time		t_{WOVD}	—	40	ns	Figure
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure
		Clocked 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	t_{cyc}	

SCL input high pulse width	t_{SCLH}	3 t_{cyc} + 300	—	ns
SCL input low pulse width	t_{SCLL}	5 t_{cyc} + 300	—	ns
SCL, SDA input falling time	t_{Sf}	—	300	ns
SCL, SDA input spike pulse removal time	t_{SP}	—	1 t_{cyc}	ns
SDA input bus free time	t_{BUF}	5 t_{cyc}	—	ns
Start condition input hold time	t_{STAH}	3 t_{cyc}	—	ns
Retransmit start condition input setup time	t_{STAS}	3 t_{cyc}	—	ns
Stop condition input setup time	t_{STOS}	1 t_{cyc} + 20	—	ns
Data input setup time	t_{SDAS}	0	—	ns
Data input hold time	t_{SDAH}	0	—	ns
SCL, SDA capacitive load	Cb	—	400	pF
SCL, SDA falling time	t_{Sf}	—	300	ns

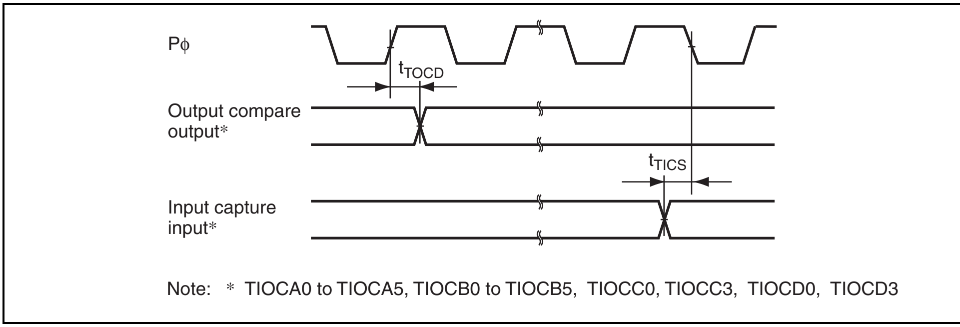


Figure 25.40 TPU Input/Output Timing

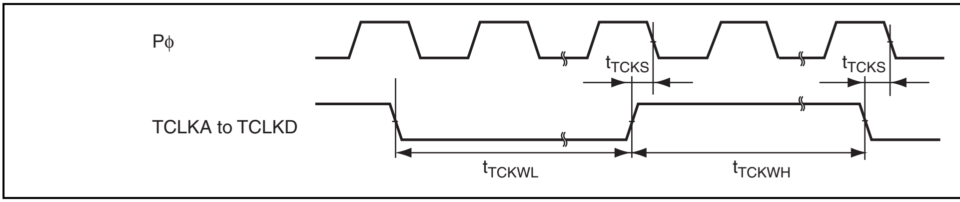


Figure 25.41 TPU Clock Input Timing

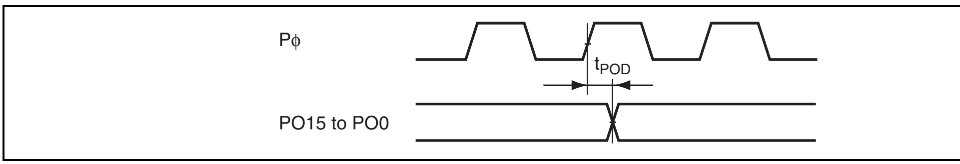


Figure 25.42 PPG Output Timing

Figure 25.44   8-Bit Timer Reset Input Timing

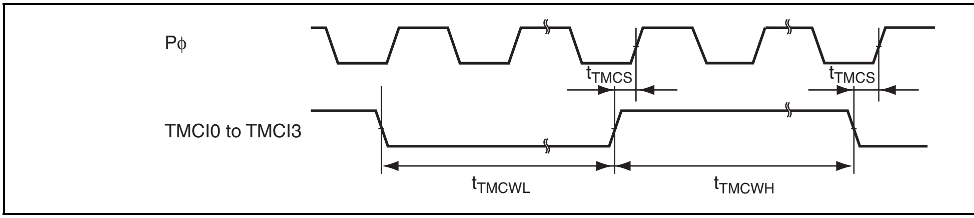


Figure 25.45   8-Bit Timer Clock Input Timing

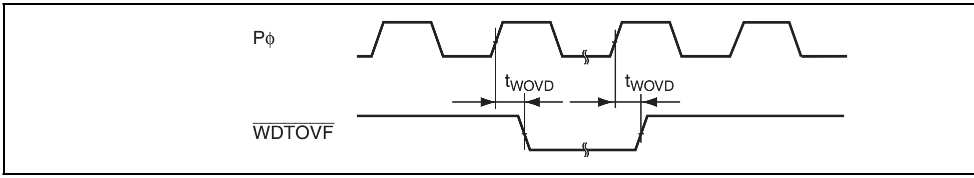


Figure 25.46   WDT Output Timing

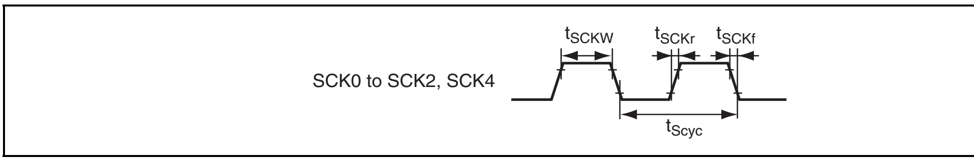


Figure 25.47   SCK Clock Input Timing

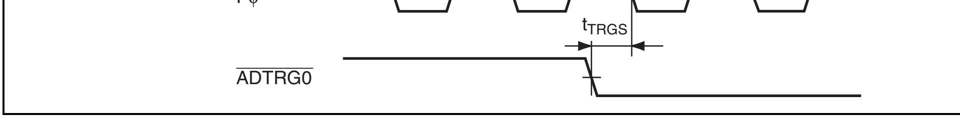


Figure 25.49 A/D Converter External Trigger Input Timing

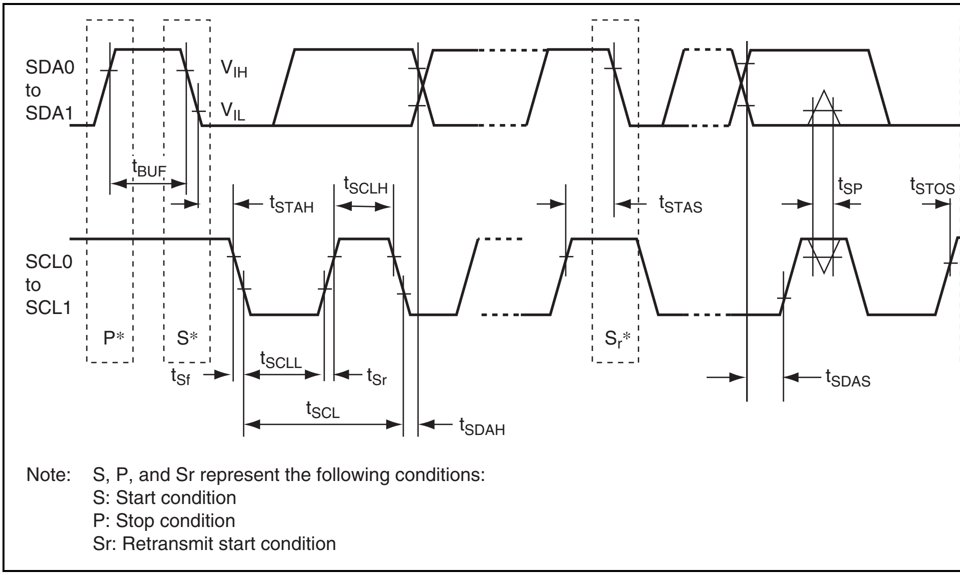


Figure 25.50 I²C Bus Interface2 Input/Output Timing (Option)

	Input low voltage	V_{IL}	—	0.8	V	
	Differential input sensitivity	V_{DI}	0.2	—	V	$\lvert (D+) - (D-) \rvert$
	Differential common mode range	V_{CM}	0.8	2.5	V	
Output	Output high voltage	V_{OH}	2.8	—	V	$I_{OH} = -200\ \mu A$
	Output low voltage	V_{OL}	—	0.3	V	$I_{OL} = 2$ mA
	Crossover voltage	V_{CRS}	1.3	2.0	V	
	Rising time	t_R	4	20	ns	
	Falling time	t_F	4	20	ns	
	Ratio of rising time to falling time	t_{RFM}	90	111.11	%	(T_R/T_F)
	Output resistance	Z_{DRV}	28	44	Ω	Including $R_S = 22\Omega$

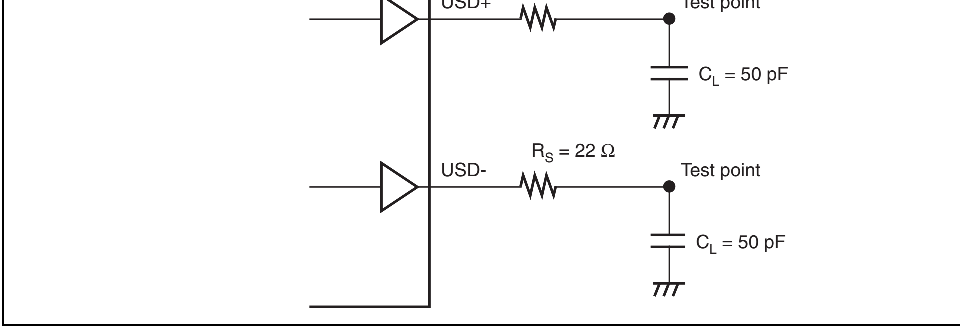


Figure 25.52 Load Condition

Conversion time	7.6	—	—	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

25.6 D/A Conversion Characteristics

Table 25.11 D/A Conversion Characteristics

Conditions: V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V
V_{SS} = PLLV_{SS} = DrV_{SS} = AV_{SS} = 0 V, Pϕ = 8 MHz to 35 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 Conditi
Resolution	8	8	8	Bit	
Conversion time	—	—	10	μs	20-pF capaci
Absolute accuracy	—	±2.0	±3.0	LSB	2-MΩ resistiv
	—	—	±2.0	LSB	4-MΩ resistiv

T_a = 0°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Typ.	Max.	Unit	Tes Cor
Programming time*[1], *[2], *[4]	t_P	—	3	30	ms/128 bytes	
Erasure time*[1], *[2], *[4]	t_E	—	160	800	ms/4-kbyte block	
		—	1000	5000	ms/32-kbyte block	
		—	2000	10000	ms/64-kbyte block	
Programming time (total)*[1], *[2], *[4]	Σ_{tP}	—	8	23	s/384 kbytes	T_a = for a
Erasure time (total)*[1], *[2], *[4]	Σ_{tE}	—	15	45	s/384 kbytes	T_a =
Programming, Erasure time (total)*[1], *[2], *[4]	Σ_{tPE}	—	23	68	s/384 kbytes	T_a =
Overwrite count	N_{WEC}	100*[3]	—	—	times	
Data save time*[5]	T_{DRP}	10	—	—	years	

Notes: 1. Programming time and erase time depend on data in the flash memory.

2. Programming time and erase time do not include time for data transfer.

3. All the characteristics after programming are guaranteed within this value (gu value is from 1 to Min. value).

4. Characteristics when programming is performed within the Min. value

Item	Symbol	Min.	Typ.	Max.	Unit	Cond
Programming time*1, *2, *4	t_P	—	3	30	ms/128 bytes	
Erasure time*1, *2, *4	t_E	—	160	800	ms/4-kbyte block	
		—	1000	5000	ms/32-kbyte block	
		—	2000	10000	ms/64-kbyte block	
Programming time (total)*1, *2, *4	Σ_{tP}	—	10	30	s/512 kbytes	T_a = 2 for al
Erasure time (total)*1, *2, *4	Σ_{tE}	—	20	60	s/512 kbytes	T_a = 2
Programming, Erasure time (total)*1, *2, *4	Σ_{tPE}	—	30	90	s/512 kbytes	T_a = 2
Overwrite count	N_{WEC}	100*3	—	—	times	
Data save time*5	T_{DRP}	10	—	—	years	

Notes: 1. Programming time and erase time depend on data in the flash memory.

2. Programming time and erase time do not include time for data transfer.

3. All the characteristics after programming are guaranteed within this value (gua value is from 1 to Min. value).

4. Characteristics when programming is performed within the Min. value

Port 3	All	Hi-Z	Hi-Z	Keep	Keep	Kee
P55 to P50	All	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Kee
P56/ AN6/ DA0/ $\overline{\text{IRQ6}}$-B	All	Hi-Z	Hi-Z	[DAOE0 = 1] Keep [DAOE0 = 0] Hi-Z	[DAOE0 = 1] Keep [DAOE0 = 0] Hi-Z	Kee
P57/ AN7/ DA1/ $\overline{\text{IRQ7}}$-B	All	Hi-Z	Hi-Z	[DAOE1 = 1] Keep [DAOE1 = 0] Hi-Z	[DAOE1 = 1] Keep [DAOE1 = 0] Hi-Z	Kee
P65 to P60	All	Hi-Z	Hi-Z	Keep	Keep	Kee
PA0/ $\overline{\text{BREQO}}$/ $\overline{\text{BS}}$-A	All	Hi-Z	Hi-Z	[$\overline{\text{BREQO}}$ output] Hi-Z [$\overline{\text{BS}}$ output] Keep [Other than above] Keep	[$\overline{\text{BREQO}}$ output] Hi-Z [$\overline{\text{BS}}$ output] Hi-Z [Other than above] Keep	[$\overline{\text{BR}}$ outp $\overline{\text{BRI}}$ [$\overline{\text{BS}}$ Hi-Z [Oth abo Kee
PA1/ $\overline{\text{BACK}}$/ (RD/$\overline{\text{WR}}$)	All	Hi-Z	Hi-Z	[$\overline{\text{BACK}}$ output] Hi-Z [RD/$\overline{\text{WR}}$ output] Keep [Other than above] Keep	[$\overline{\text{BACK}}$ output] Hi-Z [RD/$\overline{\text{WR}}$ output] Hi-Z [Other than above] Keep	[$\overline{\text{BA}}$ $\overline{\text{BAC}}$

	mode (EXPE = 1)					
PA4/ $\overline{LHWR}$/ $\overline{LUB}$	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Kee
	External extended mode (EXPE = 1)	H	Hi-Z	[$\overline{LHWR}$, $\overline{LUB}$ output] H [Other than above] Keep	[$\overline{LHWR}$, $\overline{LUB}$ output] Hi-Z [Other than above] Keep	[$\overline{LHW}$ outpu Hi-Z [Othe abov Keep
PA5/$\overline{RD}$	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep
	External extended mode (EXPE = 1)	H	Hi-Z	H	Hi-Z	Hi-Z
PA6/ $\overline{AS}$/ $\overline{AH}$/ $\overline{BS}$-B	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	[$\overline{AS}$, $\overline{BS}$ output] H [$\overline{AH}$ output] L [Other than above] Keep	[$\overline{AS}$, $\overline{AH}$, $\overline{BS}$ output] Hi-Z [Other than above] Keep	[$\overline{AS}$, outpu Hi-Z [Othe abov Keep
	External extended mode (EXPE = 1)	H				
PA7/Bϕ	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	[Clock output] H [Other than above] Keep	[Clock output] H [Other than above] Keep	[Cloc Clock [Othe abov Keep
	External extended mode (EXPE = 1)	Clock output	Hi-Z			
PB0/ $\overline{CS0}$/ $\overline{CS4}$/ $\overline{CS5}$-B	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	[$\overline{CS}$ output] H [Other than above] Keep	[$\overline{CS}$ output] Hi-Z [Other than above] Keep	[$\overline{CS}$ o Hi-Z [Othe abov Keep
	External extended mode (EXPE = 1)	H				

PB3/ $\overline{CS3}$/ $\overline{CS7}$-A	All	Hi-Z	Hi-Z	[$\overline{CS}$ output]	[$\overline{CS}$ output]	[$\overline{CS}$
				H	Hi-Z	Hi-Z
				[Other than above]	[Other than above]	[Otl abo
				Keep	Keep	Kee
PC2/ $\overline{LUCAS}$/ DQMLU	All	Hi-Z	Hi-Z	[$\overline{LUCAS}$, DQMLU output]	[$\overline{LUCAS}$, DQMLU output]	[$\overline{LU}$ DQ out
				H	Hi-Z	Hi-Z
				[Other than above]	[Other than above]	[Otl abo
				Keep	Keep	Kee
PC3/ $\overline{LLCAS}$/ DQMLL	All	Hi-Z	Hi-Z	[$\overline{LLCAS}$, DQMLL output]	[$\overline{LLCAS}$, DQMLL output]	[$\overline{LL}$ DQ
				H	Hi-Z	Hi-Z
				[Other than above]	[Other than above]	[Otl abo
				Keep	Keep	Kee
Port D	External extended mode (EXPE = 1)	L	Hi-Z	Keep	Hi-Z	Hi-
	ROM enabled extended mode	Hi-Z	Hi-Z	Keep	[Address output]	[Ac out
					Hi-Z	Hi-
					[Other than above]	[Ot abo
					Keep	Ke
	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Ke

PF7 to PF4	External extended mode (EXPE = 1)		L/ Hi-Z*	Hi-Z	Keep	[Address output] Hi-Z [Other than above] Keep	[Add outp Hi-Z [Oth abov Kee
	Single-chip mode (EXPE = 0)		Hi-Z	Hi-Z	Keep	Keep	Kee
Port H	Single-chip mode (EXPE = 0)		Hi-Z	Hi-Z	Keep	Keep	Kee
	External extended mode (EXPE = 1)		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
Port I	Single-chip mode (EXPE = 0)		Hi-Z	Hi-Z	Keep	Keep	Kee
	External extended mode (EXPE = 1)	8-bit bus mode	Hi-Z	Hi-Z	Keep	Keep	Kee
		16-bit bus mode	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
		32-bit bus mode	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
Port M	All		Hi-Z	Hi-Z	Keep	Keep	Kee

[Legend]

H: High-level output

L: Low-level output

Keep: Input pins become high-impedance, output pins retain their state.

Hi-Z: High impedance

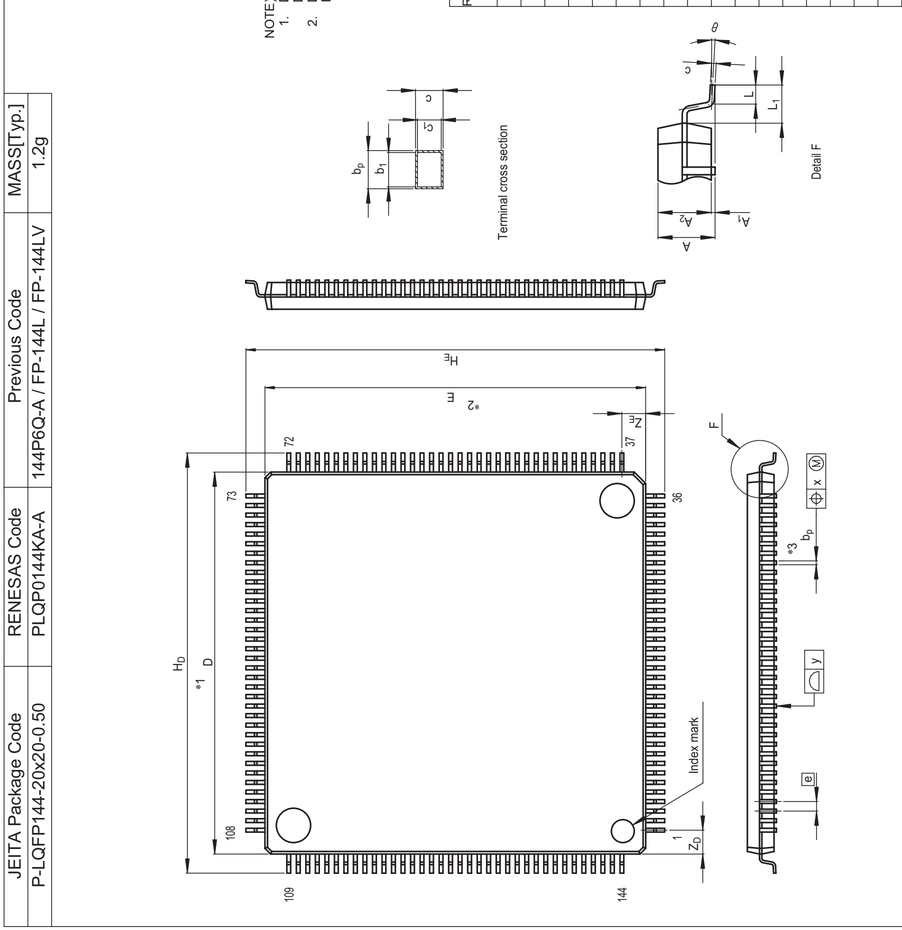


Figure C.1 Package Dimensions (FP-144LV)

MD_CLK	(Always used as a mode pin)
MD3, MD2, MD1, MD0	(Always used as mode pins)
NMI	• Connect this pin to VCC via a pull-up resistor
EXTAL	(Always used as a clock pin)
XTAL	• Leave this pin open
OSC1	• Connect this pin to VSS via a pull-down resistor
OSC2	• Leave this pin open
$\overline{WDTOVF}$	• Leave this pin open
USD+	• Leave this pin open
USD-	• Leave this pin open
VBUS	• Connect this pin to VSS via a pull-down resistor
Port 1	• Connect these pins to VCC via a pull-up resistor or to VSS via a pull-c resistor, respectively
Port 2	
Port 3	
Port 6	
PA2 to PA0	
PB7 to PB1	
Port C	
PF7 to PF5	
Port M	
Port 5	• Connect these pins to AVcc via a pull-up resistor or to AVss via a pull resistor, respectively

Port D	• These pins are left open in the initial state for the address output.	
Port E		
PF4 to PF0		
Port H	(Used as a data bus)	
Port I	(Used as a data bus)	• Connect these pins to VCC via a pull-up resistor or to VSS via a pull-down resistor, respectively, in the initial state for the general input.
Vref	• Connect this pin to AVcc	

Notes: 1. Do not change the initial value (input-buffer disabled) of PnICR, where n corres an unused pin.

2. When the pin function is changed from its initial state, use a pull-up or pull-dow resistor as needed.

L

M

P

S

T

U

Renesas 32-Bit CISC Microcomputer
Hardware Manual
H8SX/1663 Group

Publication Date:	Rev.1.00, Jun. 07, 2006
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Customer Support Department
	Global Strategic Communication Div.
	Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, J

RENESAS SALES OFFICES

http://www.rer

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan,
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8SX/1663Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0294-0100