

To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

"Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

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

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

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

H8SX/1648, H8SX/1648A, H8SX/1648L, H8SX/1648G, H8SX/1648H Group

Hardware Manual

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

H8SX/1648	R5F61648	H8SX/1642L	R5F61642L
H8SX/1644	R5F61644	H8SX/1648G	R5F61648G
H8SX/1642	R5F61642	H8SX/1644G	R5F61644G
H8SX/1648A	R5F61648A	H8SX/1642G	R5F61642G
H8SX/1644A	R5F61644A	H8SX/1648H	R5F61648H
H8SX/1642A	R5F61642A	H8SX/1644H	R5F61644H
H8SX/1648L	R5F61648L	H8SX/1642H	R5F61642H
H8SX/1644L	R5F61644L		

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

document, please confirm the latest product information with a Renesas sales office. Also, please pay attention and careful attention to additional and different information to be disclosed by Renesas such as that through our website. (<http://www.renesas.com>)

5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation, traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth in this document.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
 - (1) artificial life support devices or systems
 - (2) surgical implantations
 - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
 - (4) any other purposes that pose a direct threat to human lifeRenesas shall have no liability for damages arising out of the uses set forth in the above and purchases of Renesas products to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have special characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical damage, injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design of hardware and software including but not limited to redundancy, fire control and malfunction prevention measures, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final product system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is high. You should implement safety measures so that Renesas products may not be easily detached from the products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

vicinity of LSI, an associated shoot-through current flows internally, and malfunction due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

In a similar way, the states of pins in a product that is reset by an on-chip power-management function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stability of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

Before changing from one product to another, i.e. to one with a different type number, ensure that the change will not lead to problems.

- The characteristics of MPU/MCU in the same group but having different type numbers differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

When designing an application system that includes this LSI, take all points to note into account. Points to note are given in their contexts and at the final part of each section in the section giving usage notes.

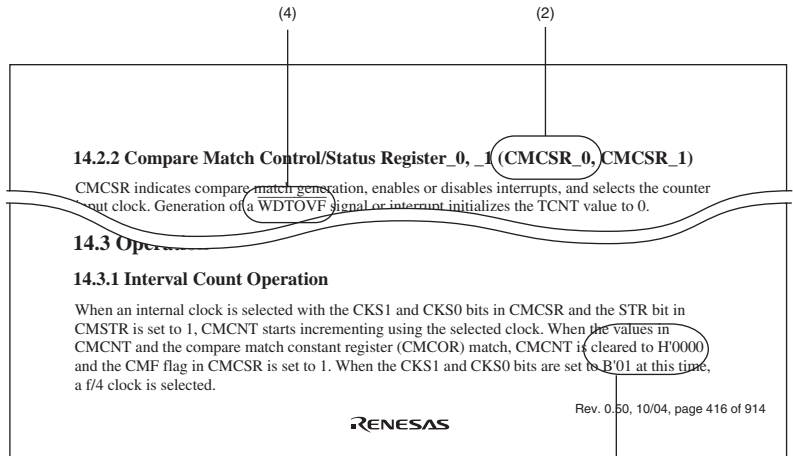
The list of revisions is a summary of major points of revision or addition for earlier versions. It does not cover all revised items. For details on the revised points, see the actual revisions in the manual.

The following documents have been prepared for the H8SX/1648A, H8SX/1648L, H8SX/1648G, H8SX/1648H Group. Before using any of the documents, please visit our web site to verify that you have the most up-to-date available version of the document.

Document Type	Contents	Document Title	Document ID
Data Sheet	Overview of hardware and electrical characteristics	—	—
Hardware Manual	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	H8SX/1648A, H8SX/1648L, H8SX/1648G, H8SX/1648H Group Hardware Manual	This manual
Software Manual	Detailed descriptions of the CPU and instruction set	H8SX Family Software Manual	REJ01G0001
Application Note	Examples of applications and sample programs	The latest versions are available from our web site.	
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.		

Binary numbers are given as B'nnnn (B' may be omitted if the number is obviously binary), hexadecimal numbers are given as H'nnnn or 0xn timer, and decimal numbers are given as nnnn.
 [Examples] Binary: B'11 or 11
 Hexadecimal: H'EFA0 or 0xEFA0
 Decimal: 1234

- (4) Notation for active-low
 An overbar on the name indicates that a signal or pin is active-low.
 [Example] WDT0VF



Note: The bit names and sentences in the above figure are examples and have nothing to do with the contents of this manual.

Bit	Bit Name	Initial Value	R/W	Description
15	-	0	R	Reserved
14	-	0	R	Reserved These bits are always read as 0.
13 to 11	ASID2 to ASID0	All 0	R/W	Address Identifier These bits enable or disable the pin function.
10	-	0	R	Reserved This bit is always read as 0.
9	-	1	R	Reserved This bit is always read as 1.
-	-	0	-	-

Note: The bit names and sentences in the above figure are examples, and have nothing to do with the content of the manual.

- (1) Bit
Indicates the bit number or numbers.
In the case of a 32-bit register, the bits are arranged in order from 31 to 0. In the case of a 16-bit register, the bits are arranged in order from 15 to 0.
- (2) Bit name
Indicates the name of the bit or bit field.
When the number of bits has to be clearly indicated in the field, appropriate notation is included (e.g., ASID[3:0]).
A reserved bit is indicated by "-".
Certain kinds of bits, such as those of timer counters, are not assigned bit names. In such cases, the entry under Bit Name is blank.
- (3) Initial value
Indicates the value of each bit immediately after a power-on reset, i.e., the initial value.
0: The initial value is 0
1: The initial value is 1
-: The initial value is undefined
- (4) R/W
For each bit and bit field, this entry indicates whether the bit or field is readable or writable or both writing to and reading from the bit or field are impossible.
The notation is as follows:
R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable.
However, writing is only performed to flag clearing.
R: The bit or field is readable.
"R" is indicated for all reserved bits. When writing to the register, write the value under Initial Value in the bit chart to reserved bits or fields.
W: The bit or field is writable.
- (5) Description
Describes the function of the bit or field and specifies the values for writing.

SCI	Serial communications interface
TMR	8-bit timer
TPU	16-bit timer pulse unit
WDT	Watchdog timer

- Abbreviations other than those listed above

Abbreviation	Description
ACIA	Asynchronous communications interface adapter
bps	Bits per second
CRC	Cyclic redundancy check
DMA	Direct memory access
DMAC	Direct memory access controller
GSM	Global System for Mobile Communications
Hi-Z	High impedance
IEBus	Inter Equipment Bus (IEBus is a trademark of NEC Electronics Corporation)
I/O	Input/output
IrDA	Infrared Data Association
LSB	Least significant bit
MSB	Most significant bit
NC	No connection
PLL	Phase-locked loop
PWM	Pulse width modulation
SFR	Special function register
SIM	Subscriber Identity Module
UART	Universal asynchronous receiver/transmitter
VCO	Voltage-controlled oscillator

All trademarks and registered trademarks are the property of their respective owners.

1.4.1	Pin Assignments
1.4.2	Correspondence between Pin Configuration and Operating Modes
1.4.3	Pin Functions
Section 2 CPU	
2.1	Features
2.2	CPU Operating Modes
2.2.1	Normal Mode
2.2.2	Middle Mode
2.2.3	Advanced Mode
2.2.4	Maximum Mode
2.3	Instruction Fetch
2.4	Address Space
2.5	Registers
2.5.1	General Registers
2.5.2	Program Counter (PC)
2.5.3	Condition-Code Register (CCR)
2.5.4	Extended Control Register (EXR)
2.5.5	Vector Base Register (VBR)
2.5.6	Short Address Base Register (SBR)
2.5.7	Multiply-Accumulate Register (MAC)
2.5.8	Initial Values of CPU Registers
2.6	Data Formats
2.6.1	General Register Data Formats
2.6.2	Memory Data Formats
2.7	Instruction Set
2.7.1	Instructions and Addressing Modes
2.7.2	Table of Instructions Classified by Function
2.7.3	Basic Instruction Formats

2.8.7	Immediate—#xx
2.8.8	Program-Counter Relative—@(d:8, PC) or @(d:16, PC)
2.8.9	Program-Counter Relative with Index Register—@(RnL.B, PC), @(Rn.W, PC), or @(ERn.L, PC).....
2.8.10	Memory Indirect—@@aa:8
2.8.11	Extended Memory Indirect—@@vec:7
2.8.12	Effective Address Calculation
2.8.13	MOVA Instruction
2.9	Processing States
Section 3 MCU Operating Modes	
3.1	Operating Mode Selection
3.2	Register Descriptions
3.2.1	Mode Control Register (MDCR)
3.2.2	System Control Register (SYSCR).....
3.3	Operating Mode Descriptions
3.3.1	Mode 1
3.3.2	Mode 2
3.3.3	Mode 3
3.3.4	Mode 4
3.3.5	Mode 5
3.3.6	Mode 6
3.3.7	Mode 7
3.3.8	Pin Functions
3.4	Address Map
3.4.1	Address Map

4.8	Watchdog Timer Reset
4.9	Determination of Reset Generation Source.....
Section 5 Voltage Detection Circuit (LVD)	
5.1	Features.....
5.2	Register Descriptions.....
5.2.1	Voltage Detection Control Register (LVDCR).....
5.2.2	Reset Status Register (RSTSR).....
5.3	Voltage Detection Circuit
5.3.1	Voltage Monitoring Reset.....
5.3.2	Voltage Monitoring Interrupt.....
5.3.3	Release from Deep Software Standby Mode by the Voltage-Detection Circuit
5.3.4	Voltage Monitor.....
Section 6 Exception Handling	
6.1	Exception Handling Types and Priority
6.2	Exception Sources and Exception Handling Vector Table
6.3	Reset
6.3.1	Reset Exception Handling.....
6.3.2	Interrupts after Reset.....
6.3.3	On-Chip Peripheral Functions after Reset Release
6.4	Traces.....
6.5	Address Error.....
6.5.1	Address Error Source.....
6.5.2	Address Error Exception Handling
6.6	Interrupts.....
6.6.1	Interrupt Sources.....
6.6.2	Interrupt Exception Handling

7.3.1	Interrupt Control Register (INTCR)
7.3.2	CPU Priority Control Register (CPUPCR)
7.3.3	Interrupt Priority Register H8SX/1648 Group, H8SX/1648A Group and H8SX/1648L Group: Interrupt Priority Registers A to I, K to O, Q, and R (IPRA to IPRI, IPRK to IPRO, IPRQ, and IPRR) H8SX/1648G Group H8SX/1648H Group: Interrupt Priority Registers A to O, O, and R (IPR IPRO, IPRQ, and IPRR)
7.3.4	IRQ Enable Register (IER)
7.3.5	IRQ Sense Control Registers H and L (ISCRH, ISCRL).....
7.3.6	IRQ Status Register (ISR).....
7.3.7	Software Standby Release IRQ Enable Register (SSIER)
7.4	Interrupt Sources
7.4.1	External Interrupts
7.4.2	Internal Interrupts
7.5	Interrupt Exception Handling Vector Table.....
7.6	Interrupt Control Modes and Interrupt Operation
7.6.1	Interrupt Control Mode 0
7.6.2	Interrupt Control Mode 2
7.6.3	Interrupt Exception Handling Sequence
7.6.4	Interrupt Response Times
7.6.5	DTC and DMAC Activation by Interrupt
7.7	CPU Priority Control Function Over DTC, DMAC, and EXDMAC
7.8	Usage Notes
7.8.1	Conflict between Interrupt Generation and Disabling
7.8.2	Instructions that Disable Interrupts
7.8.3	Times when Interrupts are Disabled
7.8.4	Interrupts during Execution of EEPMOV Instruction
7.8.5	Interrupts during Execution of MOVMD and MOVSD Instructions.....
7.8.6	Interrupts of Peripheral Modules

8.5	Usage Notes	
Section 9 Bus Controller (BSC).....		
9.1	Features.....	
9.2	Register Descriptions.....	
9.2.1	Bus Width Control Register (ABWCR).....	
9.2.2	Access State Control Register (ASTCR)	
9.2.3	Wait Control Registers A and B (WTCRA, WTCRB)	
9.2.4	Read Strobe Timing Control Register (RDNCR)	
9.2.5	\overline{CS} Assertion Period Control Registers (CSACR)	
9.2.6	Idle Control Register (IDLCR)	
9.2.7	Bus Control Register 1 (BCR1)	
9.2.8	Bus Control Register 2 (BCR2)	
9.2.9	Endian Control Register (ENDIANCR).....	
9.2.10	SRAM Mode Control Register (SRAMCR)	
9.2.11	Burst ROM Interface Control Register (BROMCR).....	
9.2.12	Address/Data Multiplexed I/O Control Register (MPXCR)	
9.2.13	DRAM Control Register (DRAMCR)	
9.2.14	DRAM Access Control Register (DRACCR).....	
9.2.15	Synchronous DRAM Control Register (SDCR)	
9.2.16	Refresh Control Register (REFCR)	
9.2.17	Refresh Timer Counter (RTCNT).....	
9.2.18	Refresh Time Constant Register (RTCOR)	
9.3	Bus Configuration.....	
9.4	Multi-Clock Function and Number of Access Cycles	
9.5	External Bus.....	
9.5.1	Input/Output Pins.....	
9.5.2	Area Division.....	
9.5.3	Chip Select Signals	

9.7	Byte Control SRAM Interface
9.7.1	Byte Control SRAM Space Setting.....
9.7.2	Data Bus
9.7.3	I/O Pins Used for Byte Control SRAM Interface
9.7.4	Basic Timing.....
9.7.5	Wait Control
9.7.6	Read Strobe (\overline{RD})
9.7.7	Extension of Chip Select (\overline{CS}) Assertion Period.....
9.7.8	\overline{DACK} and \overline{EDACK} Signal Output Timing.....
9.8	Burst ROM Interface
9.8.1	Burst ROM Space Setting.....
9.8.2	Data Bus
9.8.3	I/O Pins Used for Burst ROM Interface.....
9.8.4	Basic Timing.....
9.8.5	Wait Control
9.8.6	Read Strobe (\overline{RD}) Timing.....
9.8.7	Extension of Chip Select (\overline{CS}) Assertion Period.....
9.9	Address/Data Multiplexed I/O Interface.....
9.9.1	Address/Data Multiplexed I/O Space Setting
9.9.2	Address/Data Multiplex
9.9.3	Data Bus
9.9.4	I/O Pins Used for Address/Data Multiplexed I/O Interface.....
9.9.5	Basic Timing.....
9.9.6	Address Cycle Control.....
9.9.7	Wait Control
9.9.8	Read Strobe (\overline{RD}) Timing.....
9.9.9	Extension of Chip Select (\overline{CS}) Assertion Period.....
9.9.10	\overline{DACK} and \overline{EDACK} Signal Output Timing.....
9.10	DRAM Interface

	9.10.12	Refresh Control.....
	9.10.13	DRAM Interface and Single Address Transfer by DMAC and EXDMA.....
9.11		Synchronous DRAM Interface
	9.11.1	Setting SDRAM space
	9.11.2	Address Multiplexing.....
	9.11.3	Data Bus.....
	9.11.4	I/O Pins Used for DRAM Interface
	9.11.5	Basic Timing.....
	9.11.6	CAS Latency Control.....
	9.11.7	Controlling Row Address Output Cycle
	9.11.8	Controlling Precharge Cycle.....
	9.11.9	Controlling Clock Suspend Insertion.....
	9.11.10	Controlling Write-Precharge Delay
	9.11.11	Controlling Byte and Word Accesses
	9.11.12	Fast-Page Access Operation
	9.11.13	Refresh Control.....
	9.11.14	Setting SDRAM Mode Register
	9.11.15	SDRAM Interface and Single Address Transfer by DMAC and EXDMA.....
	9.11.16	EXDMAC Cluster Transfer
9.12		Idle Cycle.....
	9.12.1	Operation
	9.12.2	Pin States in Idle Cycle.....
9.13		Bus Release.....
	9.13.1	Operation
	9.13.2	Pin States in External Bus Released State.....
	9.13.3	Transition Timing
9.14		Internal Bus.....
	9.14.1	Access to Internal Address Space
9.15		Write Data Buffer Function

10.3	Register Descriptions
10.3.1	DMA Source Address Register (DSAR)
10.3.2	DMA Destination Address Register (DDAR)
10.3.3	DMA Offset Register (DOFR).....
10.3.4	DMA Transfer Count Register (DTCR)
10.3.5	DMA Block Size Register (DBSR)
10.3.6	DMA Mode Control Register (DMDR).....
10.3.7	DMA Address Control Register (DACR).....
10.3.8	DMA Module Request Select Register (DMRSR)
10.4	Transfer Modes
10.5	Operations.....
10.5.1	Address Modes
10.5.2	Transfer Modes
10.5.3	Activation Sources
10.5.4	Bus Access Modes
10.5.5	Extended Repeat Area Function
10.5.6	Address Update Function using Offset
10.5.7	Register during DMA Transfer
10.5.8	Priority of Channels
10.5.9	DMA Basic Bus Cycle.....
10.5.10	Bus Cycles in Dual Address Mode
10.5.11	Bus Cycles in Single Address Mode.....
10.6	DMA Transfer End
10.7	Relationship among DMAC and Other Bus Masters
10.7.1	CPU Priority Control Function Over DMAC
10.7.2	Bus Arbitration among DMAC and Other Bus Masters
10.8	Interrupt Sources
10.9	Usage Notes

	11.3.8	Cluster Buffer Registers 0 to 7 (CLSBR0 to CLSBR7).....
11.4		Transfer Modes
	11.4.1	Ordinary Modes
	11.4.2	Cluster Transfer Modes.....
11.5		Mode Operation
	11.5.1	Address Modes
	11.5.2	Transfer Modes
	11.5.3	Activation Sources
	11.5.4	Bus Mode.....
	11.5.5	Extended Repeat Area Function
	11.5.6	Address Update Function Using Offset
	11.5.7	Registers during EXDMA Transfer Operation
	11.5.8	Channel Priority Order.....
	11.5.9	Basic Bus Cycles
	11.5.10	Bus Cycles in Dual Address Mode
	11.5.11	Bus Cycles in Single Address Mode.....
	11.5.12	Operation Timing in Each Mode
11.6		Operation in Cluster Transfer Mode
	11.6.1	Address Mode
	11.6.2	Setting of Address Update Mode
	11.6.3	Caution for Combining with Extended Repeat Area Function
	11.6.4	Bus Cycles in Cluster Transfer Dual Address Mode
	11.6.5	Operation Timing in Cluster Transfer Mode
11.7		Ending EXDMA Transfer.....
11.8		Relationship among EXDMAC and Other Bus Masters.....
	11.8.1	CPU Priority Control Function Over EXDMAC
	11.8.2	Bus Arbitration with Another Bus Master
11.9		Interrupt Sources.....
11.10		Usage Notes.....

12.2.9	DTC Vector Base Register (DTCVBR).....
12.3	Activation Sources.....
12.4	Location of Transfer Information and DTC Vector Table.....
12.5	Operation
12.5.1	Bus Cycle Division
12.5.2	Transfer Information Read Skip Function
12.5.3	Transfer Information Writeback Skip Function.....
12.5.4	Normal Transfer Mode
12.5.5	Repeat Transfer Mode
12.5.6	Block Transfer Mode.....
12.5.7	Chain Transfer
12.5.8	Operation Timing.....
12.5.9	Number of DTC Execution Cycles
12.5.10	DTC Bus Release Timing
12.5.11	DTC Priority Level Control to the CPU
12.6	DTC Activation by Interrupt.....
12.7	Examples of Use of the DTC.....
12.7.1	Normal Transfer Mode
12.7.2	Chain Transfer
12.7.3	Chain Transfer when Counter = 0.....
12.8	Interrupt Sources.....
12.9	Usage Notes.....
12.9.1	Module Stop State Setting
12.9.2	On-Chip RAM
12.9.3	DMAC Transfer End Interrupt.....
12.9.4	DTCE Bit Setting.....
12.9.5	Chain Transfer
12.9.6	Transfer Information Start Address, Source Address, and Destination Address

	(n = 1 to 6, A to F, H to K, M, and N)
13.1.5	Pull-Up MOS Control Register (PnPCR) (n = D to F and H to K)
13.1.6	Open-Drain Control Register (PnODR) (n = 2 and F)
13.2	Output Buffer Control
13.2.1	Port 1
13.2.2	Port 2
13.2.3	Port 3
13.2.4	Port 5
13.2.5	Port 6
13.2.6	Port A
13.2.7	Port B
13.2.8	Port C
13.2.9	Port D
13.2.10	Port E
13.2.11	Port F
13.2.12	Port H
13.2.13	Port I
13.2.14	Port J
13.2.15	Port K
13.2.16	Port M
13.2.17	Port N
13.3	Port Function Controller
13.3.1	Port Function Control Register 0 (PFCR0)
13.3.2	Port Function Control Register 1 (PFCR1)
13.3.3	Port Function Control Register 2 (PFCR2)
13.3.4	Port Function Control Register 4 (PFCR4)
13.3.5	Port Function Control Register 6 (PFCR6)
13.3.6	Port Function Control Register 7 (PFCR7)
13.3.7	Port Function Control Register 8 (PFCR8)

14.2	Input/Output Pins.....	
14.3	Register Descriptions.....	
14.3.1	Timer Control Register (TCR).....	
14.3.2	Timer Mode Register (TMDR).....	
14.3.3	Timer I/O Control Register (TIOR).....	
14.3.4	Timer Interrupt Enable Register (TIER).....	
14.3.5	Timer Status Register (TSR).....	
14.3.6	Timer Counter (TCNT).....	
14.3.7	Timer General Register (TGR).....	
14.3.8	Timer Start Register (TSTR).....	
14.3.9	Timer Synchronous Register (TSYR).....	
14.4	Operation.....	
14.4.1	Basic Functions.....	
14.4.2	Synchronous Operation.....	
14.4.3	Buffer Operation.....	
14.4.4	Cascaded Operation.....	
14.4.5	PWM Modes.....	
14.4.6	Phase Counting Mode.....	
14.5	Interrupt Sources.....	
14.6	DTC Activation.....	
14.7	DMAC Activation.....	
14.8	A/D Converter Activation.....	
14.9	Operation Timing.....	
14.9.1	Input/Output Timing.....	
14.9.2	Interrupt Signal Timing.....	
14.10	Usage Notes.....	
14.10.1	Module Stop Function Setting.....	
14.10.2	Input Clock Restrictions.....	
14.10.3	Caution on Cycle Setting.....	

Section 15 Programmable Pulse Generator (PPG)

- 15.1 Features.....
- 15.2 Input/Output Pins.....
- 15.3 Register Descriptions.....
 - 15.3.1 Next Data Enable Registers H, L (NDERH, NDERL)
 - 15.3.2 Output Data Registers H, L (PODRH, PODRL).....
 - 15.3.3 Next Data Registers H, L (NDRH, NDRL)
 - 15.3.4 PPG Output Control Register (PCR)
 - 15.3.5 PPG Output Mode Register (PMR)
- 15.4 Operation
- 15.4.1 Output Timing.....
- 15.4.2 Sample Setup Procedure for Normal Pulse Output.....
- 15.4.3 Example of Normal Pulse Output (Example of 5-Phase Pulse Output).....
- 15.4.4 Non-Overlapping Pulse Output.....
- 15.4.5 Sample Setup Procedure for Non-Overlapping Pulse Output.....
- 15.4.6 Example of Non-Overlapping Pulse Output (Example of 4-Phase Complementary Non-Overlapping Pulse Output).....
- 15.4.7 Inverted Pulse Output
- 15.4.8 Pulse Output Triggered by Input Capture
- 15.5 Usage Notes
- 15.5.1 Module Stop State Setting
- 15.5.2 Operation of Pulse Output Pins.....
- 15.5.3 TPU Setting when PPG1 is in Use.....

Section 16 8-Bit Timers (TMR).....

- 16.1 Features.....
- 16.2 Input/Output Pins.....

16.5.1	TCNT Count Timing
16.5.2	Timing of CMFA and CMFB Setting at Compare Match
16.5.3	Timing of Timer Output at Compare Match
16.5.4	Timing of Counter Clear by Compare Match
16.5.5	Timing of TCNT External Reset.....
16.5.6	Timing of Overflow Flag (OVF) Setting
16.6	Operation with Cascaded Connection.....
16.6.1	16-Bit Counter Mode.....
16.6.2	Compare Match Count Mode.....
16.7	Interrupt Sources.....
16.7.1	Interrupt Sources and DTC Activation
16.7.2	A/D Converter Activation.....
16.8	Usage Notes
16.8.1	Notes on Setting Cycle
16.8.2	Conflict between TCNT Write and Counter Clear
16.8.3	Conflict between TCNT Write and Increment.....
16.8.4	Conflict between TCOR Write and Compare Match.....
16.8.5	Conflict between Compare Matches A and B.....
16.8.6	Switching of Internal Clocks and TCNT Operation
16.8.7	Mode Setting with Cascaded Connection
16.8.8	Module Stop State Setting
16.8.9	Interrupts in Module Stop State
Section 17 32K Timer (TM32K).....	
17.1	Features.....
17.2	Register Descriptions.....
17.2.1	Timer Control Register (TCR32K).....
17.2.2	Timer Counter (TCNT32K1, TCNT32K2, TCNT32K3).....
17.3	Operation

- 18.2 Input/Output Pin
- 18.3 Register Descriptions.....
 - 18.3.1 Timer Counter (TCNT).....
 - 18.3.2 Timer Control/Status Register (TCSR).....
 - 18.3.3 Reset Control/Status Register (RSTCSR).....
- 18.4 Operation
- 18.4.1 Watchdog Timer Mode.....
- 18.4.2 Interval Timer Mode.....
- 18.5 Interrupt Source
- 18.6 Usage Notes.....
 - 18.6.1 Notes on Register Access.....
 - 18.6.2 Conflict between Timer Counter (TCNT) Write and Increment.....
 - 18.6.3 Changing Values of Bits CKS2 to CKS0.....
 - 18.6.4 Switching between Watchdog Timer Mode and Interval Timer Mode.....
 - 18.6.5 Internal Reset in Watchdog Timer Mode.....
 - 18.6.6 System Reset by $\overline{\text{WDTOVF}}$ Signal.....
 - 18.6.7 Transition to Watchdog Timer Mode or Software Standby Mode.....

Section 19 Serial Communication Interface (SCI, IrDA, CRC).....

- 19.1 Features.....
- 19.2 Input/Output Pins.....
- 19.3 Register Descriptions.....
 - 19.3.1 Receive Shift Register (RSR)
 - 19.3.2 Receive Data Register (RDR).....
 - 19.3.3 Transmit Data Register (TDR).....
 - 19.3.4 Transmit Shift Register (TSR)
 - 19.3.5 Serial Mode Register (SMR)
 - 19.3.6 Serial Control Register (SCR).....
 - 19.3.7 Serial Status Register (SSR)



	19.4.5	Serial Data Transmission (Asynchronous Mode)
	19.4.6	Serial Data Reception (Asynchronous Mode)
19.5		Multiprocessor Communication Function
	19.5.1	Multiprocessor Serial Data Transmission
	19.5.2	Multiprocessor Serial Data Reception
19.6		Operation in Clock Synchronous Mode.....
	19.6.1	Clock.....
	19.6.2	SCI Initialization (Clock Synchronous Mode).....
	19.6.3	Serial Data Transmission (Clock Synchronous Mode).....
	19.6.4	Serial Data Reception (Clock Synchronous Mode)
	19.6.5	Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)
19.7		Operation in Smart Card Interface Mode.....
	19.7.1	Sample Connection
	19.7.2	Data Format (Except in Block Transfer Mode)
	19.7.3	Block Transfer Mode
	19.7.4	Receive Data Sampling Timing and Reception Margin
	19.7.5	Initialization.....
	19.7.6	Data Transmission (Except in Block Transfer Mode)
	19.7.7	Serial Data Reception (Except in Block Transfer Mode)
	19.7.8	Clock Output Control.....
19.8		IrDA Operation.....
19.9		Interrupt Sources.....
	19.9.1	Interrupts in Normal Serial Communication Interface Mode
	19.9.2	Interrupts in Smart Card Interface Mode
19.10		Usage Notes
	19.10.1	Module Stop State Setting
	19.10.2	Break Detection and Processing
	19.10.3	Mark State and Break Detection.....

Section 20	I ² C Bus Interface 2 (IIC2)
20.1	Features
20.2	Input/Output Pins
20.3	Register Descriptions
20.3.1	I ² C Bus Control Register A (ICCRA)
20.3.2	I ² C Bus Control Register B (ICCRB)
20.3.3	I ² C Bus Mode Register (ICMR)
20.3.4	I ² C Bus Interrupt Enable Register (ICIER)
20.3.5	I ² C Bus Status Register (ICSR)
20.3.6	Slave Address Register (SAR)
20.3.7	I ² C Bus Transmit Data Register (ICDRT)
20.3.8	I ² C Bus Receive Data Register (ICDRR)
20.3.9	I ² C Bus Shift Register (ICDRS)
20.4	Operation
20.4.1	I ² C Bus Format
20.4.2	Master Transmit Operation
20.4.3	Master Receive Operation
20.4.4	Slave Transmit Operation
20.4.5	Slave Receive Operation
20.4.6	Noise Canceller
20.4.7	Example of Use
20.5	Interrupt Request
20.6	Bit Synchronous Circuit
20.7	Usage Notes
Section 21	A/D Converter
21.1	Features
21.2	Input/Output Pins
21.3	Register Descriptions

21.4.4	External Trigger Input Timing.....
21.5	Interrupt Source
21.6	A/D Conversion Accuracy Definitions
21.7	Usage Notes
21.7.1	Module Stop Function Setting
21.7.2	A/D Input Hold Function in Software Standby Mode
21.7.3	Notes on A/D Activation by an External Trigger
21.7.4	Permissible Signal Source Impedance
21.7.5	Influences on Absolute Accuracy
21.7.6	Setting Range of Analog Power Supply and Other Pins
21.7.7	Notes on Board Design
21.7.8	Notes on Noise Countermeasures
Section 22	D/A Converter
22.1	Features.....
22.2	Input/Output Pins.....
22.3	Register Descriptions.....
22.3.1	D/A Data Registers 0 and 1 (DADR0 and DADR1).....
22.3.2	D/A Control Register 01 (DACR01)
22.4	Operation
22.5	Usage Notes
22.5.1	Module Stop State Setting
22.5.2	D/A Output Hold Function in Software Standby Mode.....
22.5.3	Notes on Deep Software Standby Mode
Section 23	RAM.....

24.7	Register Descriptions
24.7.1	Programming/Erasing Interface Registers
24.7.2	Programming/Erasing Interface Parameters
24.7.3	RAM Emulation Register (RAMER).....
24.8	On-Board Programming Mode
24.8.1	Boot Mode
24.8.2	User Program Mode.....
24.8.3	User Boot Mode.....
24.8.4	On-Chip Program and Storable Area for Program Data
24.9	Protection
24.9.1	Hardware Protection
24.9.2	Software Protection.....
24.9.3	Error Protection.....
24.10	Flash Memory Emulation Using RAM.....
24.11	Switching between User MAT and User Boot MAT
24.12	Programmer Mode
24.13	Standard Serial Communications Interface Specifications for Boot Mode.....
24.14	Usage Notes
	Section 25 Boundary Scan.....
25.1	Features.....
25.2	Block Diagram of Boundary Scan Function
25.3	Input/Output Pins
25.4	Register Descriptions.....
25.4.1	Instruction Register (JTIR)
25.4.2	Bypass Register (JTBPR)
25.4.3	Boundary Scan Register (JTBSR).....
25.4.4	IDCODE Register (JTID)
25.5	Operations.....

26.3	PLL Circuit
26.4	Frequency Divider
26.5	Subclock Oscillator.....
26.5.1	Connecting 32.768 kHz Resonator
26.5.2	Handling of Pins when the Subclock is Not to be Used.....
26.6	Usage Notes
26.6.1	Notes on Clock Pulse Generator
26.6.2	Notes on Resonator
26.6.3	Notes on Board Design
Section 27 Power-Down Modes	
27.1	Features.....
27.2	Register Descriptions.....
27.2.1	Standby Control Register (SBYCR)
27.2.2	Module Stop Control Registers A and B (MSTPCRA and MSTPCRB)
27.2.3	Module Stop Control Register C (MSTPCRC).....
27.2.4	Deep Standby Control Register (DPSBYCR).....
27.2.5	Deep Standby Wait Control Register (DPSWCR).....
27.2.6	Deep Standby Interrupt Enable Register (DPSIER)
27.2.7	Deep Standby Interrupt Flag Register (DPSIFR).....
27.2.8	Deep Standby Interrupt Edge Register (DPSIEGR)
27.2.9	Reset Status Register (RSTSR).....
27.2.10	Deep Standby Backup Register (DPSBKRn)
27.3	Multi-Clock Function
27.3.1	Switching of Main Clock Frequencies.....
27.3.2	Switching to Subclock
27.4	Module Stop State.....
27.5	Sleep Mode.....
27.5.1	Entry to Sleep Mode

27.8.4	B ϕ or SDRAM ϕ Operation after Exit from Deep Software Standby Mode
27.8.5	Setting Oscillation Settling Time after Exit from Deep Software Standby Mode
27.8.6	Deep Software Standby Mode Application Example
27.8.7	Flowchart of Deep Software Standby Mode Operation
27.9	Hardware Standby Mode
27.9.1	Transition to Hardware Standby Mode
27.9.2	Clearing Hardware Standby Mode.....
27.9.3	Hardware Standby Mode Timing.....
27.9.4	Timing Sequence at Power-On
27.10	Sleep Instruction Exception Handling
27.11	ϕ Clock Output Control.....
27.12	Usage Notes
27.12.1	I/O Port Status.....
27.12.2	Current Consumption during Oscillation Settling Standby Period
27.12.3	Module Stop State of DMAC, EXDMAC, or DTC
27.12.4	On-Chip Peripheral Module Interrupts
27.12.5	Writing to MSTPCRA, MSTPCRB, and MSTPCRC
27.12.6	Control of Input Buffers by DIRQnE (n = 3 to 0).....
27.12.7	Conflict between a transition to deep software standby mode and interrupts
27.12.8	B ϕ or SDRAM ϕ Output State
Section 28 List of Registers	
28.1	Register Addresses (Address Order).....
28.2	Register Bits.....
28.3	Register States in Each Operating Mode

29.5 A/D Conversion Characteristics
29.6 D/A Conversion Characteristics
29.7 Flash Memory Characteristics
29.8 Power-On Reset Circuit and Voltage-Detection Circuit Characteristics
(H8SX/1648L Group, H8SX/1648H Group)

Appendix
A. Port States in Each Pin State.....
B. Product Lineup.....
C. Package Dimensions.....
D. Treatment of Unused Pins.....

Main Revisions and Additions in this Edition.....

Index

As peripheral functions, each LSI of the Group includes a DMA controller and EXDMA controller*¹, which enable high-speed data transfer, and a bus-state controller, which enable direct connection to different kinds of memory. The LSI of the Group also includes serial communication interfaces, A/D and D/A converters, and a multi-function timer that makes control easy. Together, the modules realize low-cost configurations for end systems. The consumption of these modules is kept down dynamically by an on-chip power-management function. The on-chip ROM is a flash memory (F-ZTAT^{TM*}) with a capacity of 1024 Kbytes (H8SX/1648, H8SX/1648A, H8SX/1648L, H8SX/1648G, and H8SX/1648H), 512 Kbytes (H8SX/1644, H8SX/1644A, H8SX/1644L, H8SX/1644G, and H8SX/1644H), or 256 Kbytes (H8SX/1642, H8SX/1642A, H8SX/1642L, H8SX/1642G, and H8SX/1642H).

- Note:
1. Supported only by the H8SX/1648G Group and H8SX/1648H Group.
 2. F-ZTATTM is a trademark of Renesas Technology Corp.

1.1.1 Applications

Examples of the applications of this LSI include PC peripheral equipment, optical storage office automation equipment, and industrial equipment.

CPU	CPU	<ul style="list-style-type: none"> • 32-bit high-speed H8SX CPU (CISC type) Upwardly compatible for H8/300, H8/300H, and H8S C object level • General-register architecture (sixteen 16-bit general re • Eleven addressing modes • 4-Gbyte address space Program: 4 Gbytes available Data: 4 Gbytes available • 87 basic instructions, classifiable as bit arithmetic and instructions, multiply and divide instructions, bit manipu instructions, multiply-and-accumulate instructions, and • Minimum instruction execution time: 20.0 ns (for an AD instruction while system clock $f_{\phi} = 50$ MHz and $V_{CC} = 3.0$ to 3.6 V) • On-chip multiplier ($16 \times 16 \rightarrow 32$ bits) • Supports multiply-and-accumulate instructions ($16 \times 16 + 42 \rightarrow 42$ bits)
	Operating mode	<ul style="list-style-type: none"> • Advanced mode Normal, middle, or maximum mode is not supported.

and MD0 pins high)

Mode 4: On-chip ROM disabled external extended mode
(selected by driving the MD1 and MD0 pins low and the MD2 pin high)

Mode 5: On-chip ROM disabled external extended mode
(selected by driving the MD1 pin low and driving and MD0 pins high)

Mode 6: On-chip ROM enabled external extended mode
(selected by driving the MD0 pin low and driving and MD1 pins high)

Mode 7: Single-chip mode (can be externally extended)
(selected by driving the MD2, MD1, and MD0 pins high)

- Low power consumption state (transition driven by the instruction)

Power on reset (POR)* ³	<ul style="list-style-type: none">• At power-on or low power supply voltage, an internal reset signal is generated
Voltage detection circuit (LVD)* ³	<ul style="list-style-type: none">• At low power supply voltage, an internal reset signal and interrupt are generated.

		<ul style="list-style-type: none"> • Eight priority orders specifiable (by setting the interrupt register) • Independent vector addresses
	Break interrupt (UBC)	<ul style="list-style-type: none"> • Break point can be set for four channels • Address break can be set for CPU instruction fetch cycle
DMA	EXDMA controller (EXDMAC)* ¹	<ul style="list-style-type: none"> • Four-channel DMA transfer available • Two activation methods (auto-request and external request) • Four transfer modes (normal, repeat, block, and cluster) • Dual or single address mode selectable • Extended repeat area function
	DMA controller (DMAC)	<ul style="list-style-type: none"> • Four-channel DMA transfer available • Three activation methods (auto-request, on-chip module interrupt, and external request) • Three transfer modes (normal, repeat, and block transfer) • Dual or single address mode selectable • Extended repeat area function
	Data transfer controller (DTC)	<ul style="list-style-type: none"> • H8SX/1648, H8SX/1648A, and H8SX/1648L Groups: Allow DMA transfer over 76 channels (number of DTC sources) • H8SX/1648G and H8SX/1648H Groups: Allow DMA transfer over 84 channels (number of DTC sources) • Activated by interrupt sources (chain transfer enabled) • Three transfer modes (normal, repeat, and block transfer) • Short-address mode or full-address mode selectable

- Bus arbitration function (arbitrates bus mastership among internal CPU, DMAC, EXDMAC*¹, DTC, refresh*¹, and bus masters)

Bus formats

- External memory interfaces (for the connection of ROM, SRAM*¹, byte control SRAM, DRAM*¹, synchronous DRAM*¹)
- Address/data bus format: Support for both separate and multiplexed buses (8-bit access or 16-bit access)
- Endian conversion function for connecting devices in big-endian format

Clock

Clock pulse generator (CPG)

- One clock generation circuit available
- Separate clock signals are provided for each of functional modules (detailed below) and each is independently programmable (multi-clock function)
 - System-intended data transfer modules, i.e. the CPU, are synchronized with the system clock (I ϕ): 8 to 50 MHz
 - Internal peripheral functions run in synchronization with the peripheral module clock (P ϕ): 8 to 35 MHz
 - Modules in the external space are supplied with the bus clock (B ϕ): 8 to 50 MHz
- Includes a PLL frequency multiplication circuit and frequency divider, so the operating frequency is selectable
- Five low-power-consumption modes: sleep mode, all-clock-stop mode, software standby mode, deep software standby mode, and hardware standby mode

- Three ways to start A/D conversion:
 - Unit 0: Software, timer (TPU/TMR (units 0 and 1)) trigger, and external trigger
 - Unit 1: Software, TMR (units 2 and 3) trigger, and external trigger
 - Unit 2: Software, TMR (units 2 and 3) trigger, and external trigger
- Activation of DTC and DMAC by ADI interrupt:
 - Unit 0: DTC and DMAC can be activated by an ADI0 interrupt.
 - Unit 1: DMAC can be activated by an ADI1 interrupt.
 - Unit 2: DMAC can be activated by an ADI2 interrupt.

D/A converter	D/A converter (DAC)	<ul style="list-style-type: none"> • 8-bit resolution × two output channels • Output voltage: 0 V to Vref, maximum conversion time (with 20-pF load)
Timer	8-bit timer (TMR)	<ul style="list-style-type: none"> • 8 bits × eight channels (can be used as 16 bits × four channels) • Select from among seven clock sources (six internal clocks and one external clock) • Allows the output of pulse trains with a desired duty cycle and PWM signals

channels), and phase counting mode (two-phase encoder mode). The phase counting mode is not available for each channel

- Input capture function supported
- Output compare function (by the output of compare mode) supported

Note: * The pin function of unit 1 cannot be used in extended mode.

Program-
mable pulse
generator
(PPG)

- 32-bit*¹*² pulse output
- Four output groups, non-overlapping mode, and inverse mode can be set
- Selectable output trigger signals; the PPG can operate in conjunction with the data transfer controller (DTC) and the data transfer controller (DMAC)

Notes: 1. Pulse output pins PO31 to PO16 cannot be used for input capture.

2. Pulse of unit 1 cannot be output in external bus mode in extended mode.

Watchdog
timer

Watchdog
timer (WDT)

- 8 bits × one channels (selectable from eight counter clocks)
- Switchable between watchdog timer mode and interval timer mode

32K timer

32K timer
(TM32K)

- Eight counter clocks which divides the 32.768 Hz clock selected
 - 8 bits x one channel or 24 bits x 1 channel can be selected
 - Interrupts can be generated when the counter overflows
 - Eight overflow cycles selectable (250 msec, 500 msec, 1 sec, 30 sec, 60 sec, about 23 days, and about 48 days)
-

card/SIM

I²C bus
interface

I²C bus
interface 2
(IIC2)

- Four channels
- Bus can be directly driven (the SCL and SDA pins are open drains).
- IIC2 (unit 1) is open drain pins supporting 5 V input.

I/O ports

- Input-only pins:
H8SX/1648 Group, H8SX/1648A Group, H8SX/1648L pins each.
H8SX/1648G Group, H8SX/1648H Group: 13 pins each.
- Input/Output pins:
H8SX/1648 Group, H8SX/1648A Group, H8SX/1648L pins each.
H8SX/1648G Group, H8SX/1648H: 102 pins each.
- Eight large-current drive pins (port 3)
- 40 pull-up resistors
- 16 open drains
- Four open-drain I/O pins supporting 5 V input

Package

- LQFP-144 package*²
- LFBGA176 package*¹

- Notes:
1. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.
 2. Supported only by the H8SX/1648 Group, the H8SX/1648A Group, and the H8SX/1648L Group.
 3. Supported only by the H8SX/1648L Group and the H8SX/1648H Group.

IIC2		0	0	0	0
TMR		0	0	0	0
WDT		0	0	0	0
10-bit ADC		0	0	0	0
8-bit DAC		0	0	0	0
EXDMAC		—	—	0	0
SDRAM interface		—	—	0	0
32K timer		—	—	0	0
POR/LVD		—	0	—	0
Package	LQFP-144	0	0	—	—
	LFBGA-176	—	—	0	0

Note: * The setting method of Port 6 differs in the H8SX/1648 Group and the H8SX/1648 Group. For details, see section 13.2.5, Port 6.

	R5F61648D50FPV	1024 Kbytes	56 Kbytes	LQFP-144	Wi
	R5F61644D50FPV	512 Kbytes	40 Kbytes	LQFP-144	spe
	R5F61642D50FPV	256 Kbytes	24 Kbytes	LQFP-144	tior
H8SX/1648A	R5F61648AN50FPV	1024 Kbytes	56 Kbytes	LQFP-144	Re
	R5F61644AN50FPV	512 Kbytes	40 Kbytes	LQFP-144	spe
	R5F61642AN50FPV	256 Kbytes	24 Kbytes	LQFP-144	tior
	R5F61648AD50FPV	1024 Kbytes	56 Kbytes	LQFP-144	Wi
	R5F61644AD50FPV	512 Kbytes	40 Kbytes	LQFP-144	spe
	R5F61642AD50FPV	256 Kbytes	24 Kbytes	LQFP-144	tior
H8SX/1648G	R5F61648GN50BGV	1024 Kbytes	56 Kbytes	LFBGA-176	Re
	R5F61644GN50BGV	512 Kbytes	40 Kbytes	LFBGA-176	spe
	R5F61642GN50BGV	256 Kbytes	24 Kbytes	LFBGA-176	tior
	R5F61648GD50BGV	1024 Kbytes	56 Kbytes	LFBGA-176	Wi
	R5F61644GD50BGV	512 Kbytes	40 Kbytes	LFBGA-176	spe
	R5F61642GD50BGV	256 Kbytes	24 Kbytes	LFBGA-176	tior
H8SX/1648L	R5F61648LN50FPV	1024 Kbytes	56 Kbytes	LQFP-144	Re
	R5F61644LN50FPV	512 Kbytes	40 Kbytes	LQFP-144	spe
	R5F61642LN50FPV	256 Kbytes	24 Kbytes	LQFP-144	tior
	R5F61648LD50FPV	1024 Kbytes	56 Kbytes	LQFP-144	Wi
	R5F61644LD50FPV	512 Kbytes	40 Kbytes	LQFP-144	spe
	R5F61642LD50FPV	256 Kbytes	24 Kbytes	LQFP-144	tior

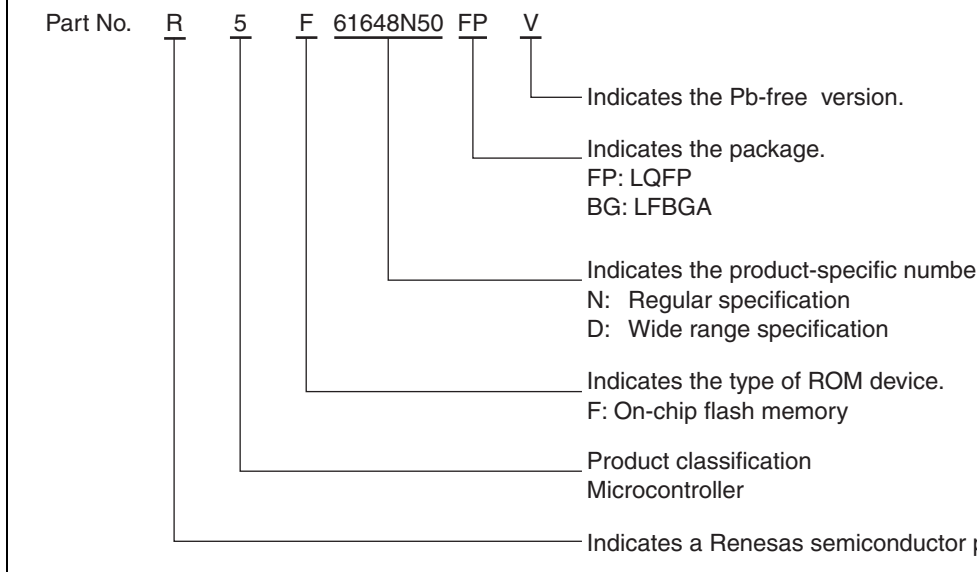
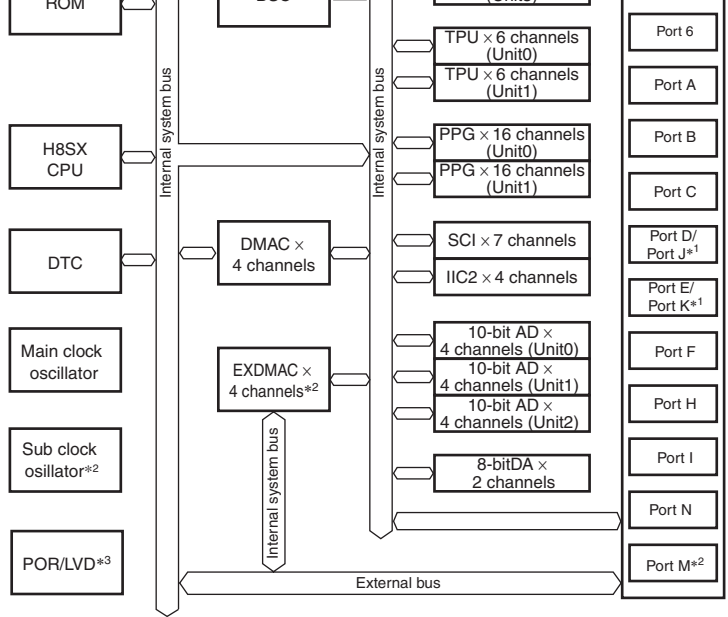


Figure 1.1 How to Read the Product Name Code

- Small Package

Package	Package Code	Body Size	Pin Pitch
LQFP-144	PLQP0144KA-A (FP-144LV)*	20.0 × 20.0 mm	0.50 mm
LFBGA-176	PLBG0176GA-A (BP-176V)*	13.0 × 13.0 mm	0.80 mm

Note: * Pb-free version



[Legend]

CPU:	Central processing unit	TMR:	8-bit timer
DTC:	Data transfer controller	TPU:	16-bit timer pulse unit
BSC:	Bus controller	PPG:	Programmable pulse generator
DMAC:	DMA controller	SCI:	Serial communications interface
EXDMAC*2:	EXDMA Controller	IIC2:	IIC bus interface 2
TM32K*2:	32K Timer	POR/LVD*3:	Power-on reset / Low voltage detection circuit
WDT:	Watchdog timer		

Note: *1 In single-chip mode, the port D and port E functions can be used in the initial state. Pin functions are selectable by setting the PCJKE bit in PFCRD. Ports D and E are enabled when PCJKE = 0 (initial value) and ports J and K are enabled when PCJKE = 1. In external extended mode, only ports D and E can be used.

*2 Supported only by the H8SX/1648G Group and H8SX/1648H Group.

*3 Supported only by the 8SX/1648L Group and H8SX/1648H Group.

Figure 1.2 Block Diagram

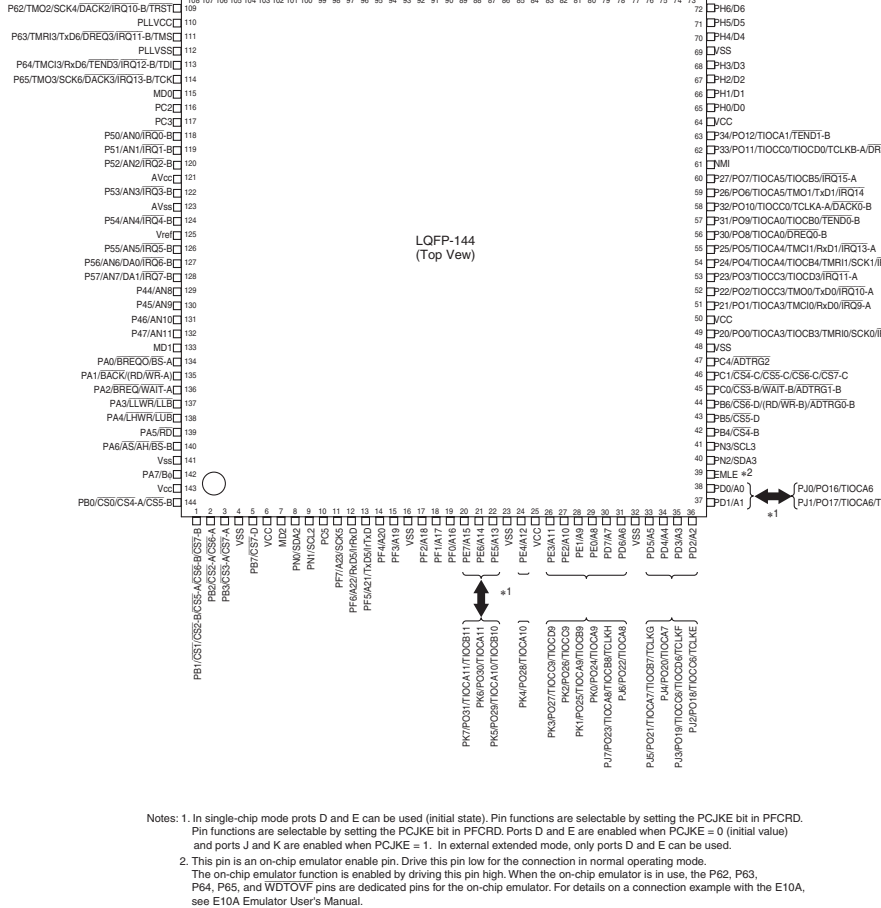


Figure 1.3 Pin Assignments (LQFP-144: H8SX/1648, 1648A, and 1648L) Group

F	PF6	PF5	PF7	PC5	LFBGA-176 (Top view)							WDTOVF	V _{SS}	EXTAL
G	V _{SS}	V _{SS}	PF3	PF4								RES	V _{CL}	P15
H	PF0	PE7*1	PF1	PF2								P67	NC*3	OSC1*
J	V _{SS}	PE4*1	PE5*1	PE6*1								P13	V _{SS}	P66
K	PE3*1	PE2*1	NC*3	V _{CC}								P17	P10	P12
L	PE0*1	PD7*1	PE1*1	PD6*1								P18	P14	P15
M	V _{SS}	V _{SS}	PD5*1	PD3*1	NC*3	PB4	NC*3	P20	P23	P31	V _{CC}	V _{SS}	P10	P12
N	PD4*1	PD2*1	PN2	PM4	NC*3	PB5	PC1	V _{CC}	P24	P32	NMI	PH2	V _{SS}	NC*3
P	PD1*1	PD0*1	PN3	NC*3	V _{SS}	PC0	V _{SS}	P22	P30	P27	P34	PH1	PH4	PH6
R	EMLE*2	PM3	V _{CC}	NC*3	NC*3	PB6	PC4	P21	P25	P26	P33	PH0	PH3	PH5
	1	2	3	4	5	6	7	8	9	10	11	12	13	14

- Notes: 1. In single-chip mode ports D and E can be used (initial state). Pin functions are selectable by setting the PCJKE bit in PFCRD. Pin functions are selectable by setting the PCJKE bit in PFCRD. Ports D and E are enabled when PCJKE = 0 (initial value) and ports J and K are enabled when PCJKE = 1. In external extended mode, only ports D and E can be used.
2. This pin is an on-chip emulator enable pin. Drive this pin low for the connection in normal operating mode. The on-chip emulator function is enabled by driving this pin high. When the on-chip emulator is in use, the P62, P63, P64, P65, and WDTOVF pins are dedicated pins for the on-chip emulator. For details on a connection example with the E10A, see E10A Emulator User's Manual.
3. NC should be left open.

Figure 1.4 Pin Assignments (LFBGA-176: H8SX/1648G and 1648H Group)

3	B1	PB3/ $\overline{\text{CS3-A}}/\overline{\text{CS7-A}}/\text{CAS}^{*3}$	PB3/ $\overline{\text{CS3-A}}/\overline{\text{CS7-A}}/\text{CAS}^{*3}$	PB3/ $\overline{\text{CS3-A}}/\overline{\text{CS7-A}}/\text{CAS}^{*3}$
4	C2	VSS	VSS	VSS
5	D3	PB7/ $\overline{\text{CS7-D}}/\text{SDRAM}\phi^{*3}$	PB7/ $\overline{\text{CS7-D}}/\text{SDRAM}\phi^{*3}$	PB7/ $\overline{\text{CS7-D}}/\text{SDRAM}\phi^{*3}$
6	C1	VCC	VCC	VCC
7	D2	MD2	MD2	MD2
—	E4	PM0 ^{*3}	PM0 ^{*3}	PM0 ^{*3}
8	D1	PN0/SDA2	PN0/SDA2	PN0/SDA2
—	E3	PM1 ^{*3}	PM1 ^{*3}	PM1 ^{*3}
9	E2	PN1/SCL2	PN1/SCL2	PN1/SCL2
—	E1	PM2 ^{*3}	PM2 ^{*3}	PM2 ^{*3}
10	F4	PC5	PC5	PC5
11	F3	PF7/A23/SCK5	PF7/A23/SCK5	PF7/A23/SCK5
12	F1	PF6/A22/RxD5/IrRxD	PF6/A22/RxD5/IrRxD	PF6/A22/RxD5/IrRxD
13	F2	PF5/A21/TxD5/IrTxD	PF5/A21/TxD5/IrTxD	PF5/A21/TxD5/IrTxD
14	G4	PF4/A20	PF4/A20	A20
15	G3	PF3/A19	PF3/A19	A19
16	G1	VSS	VSS	Vss
—	G2	VSS	VSS	VSS
17	H4	PF2/A18	PF2/A18	A18
18	H3	PF1/A17	PF1/A17	A17
19	H1	PF0/A16	PF0/A16	A16
20	H2	PE7/A15	PE7/A15	PK7/PO31/ TIOCA11/ TIOCB11 ^{*1}

25	K4	Vcc	Vcc	Vcc
—	K3	NC	NC	NC
26	K1	PE3/A11	PE3/A11	PK3/PO27/ TIOCC9/ TIOCD9* ¹
27	K2	PE2/A10	PE2/A10	PK2/PO26/ TIOCC9* ¹
28	L3	PE1/A9	PE1/A9	PK1/PO25/ TIOCA9/TIOCB9* ¹
29	L1	PE0/A8	PE0/A8	PK0/PO24/ TIOCA9* ¹
30	L2	PD7/A7	PD7/A7	PJ7/PO23/ TIOCA8/TIOCB8/ TCLKH* ¹
31	L4	PD6/A6	PD6/A6	PJ6/PO22/ TIOCA8* ¹
32	M1	Vss	Vss	Vss
—	M2	Vss	Vss	Vss
33	M3	PD5/A5	PD5/A5	PJ5/PO21/ TIOCA7/TIOCB7/ TCLKG* ¹
34	N1	PD4/A4	PD4/A4	PJ4/PO20/ TIOCA7* ¹
35	M4	PD3/A3	PD3/A3	PJ3/PO19/ TIOCC6/TIOCD6/ TCLKF* ¹
36	N2	PD2/A2	PD2/A2	PJ2/PO18/ TIOCC6/TCLK* ¹

41	P3	PN3/SCL3	PN3/SCL3	PN3/SCL3
—	N4	PM4* ³	PM4* ³	PM4* ³
—	R3	VCC	VCC	VCC
—	P4	NC	NC	NC
—	M5	NC	NC	NC
—	R4	NC	NC	NC
—	N5	NC	NC	NC
—	P5	V _{ss}	V _{ss}	V _{ss}
—	R5	NC	NC	NC
42	M6	PB4/ $\overline{\text{CS4-B}}$ / $\overline{\text{WE}}$ * ³	PB4/ $\overline{\text{CS4-B}}$ / $\overline{\text{WE}}$ * ³	PB4/ $\overline{\text{CS4-B}}$ / $\overline{\text{WE}}$ * ³
43	N6	PB5/ $\overline{\text{CS5-D}}$ / $\overline{\text{OE}}$ / $\overline{\text{CKE}}$ * ³	PB5/ $\overline{\text{CS5-D}}$ / $\overline{\text{OE}}$ / $\overline{\text{CKE}}$ * ³	PB5/ $\overline{\text{CS5-D}}$ / $\overline{\text{OE}}$ / $\overline{\text{CKE}}$ * ³
44	R6	PB6/ $\overline{\text{CS6-D}}$ / (RD/ $\overline{\text{WR-B}}$)/ $\overline{\text{ADTRG0-B}}$	PB6/ $\overline{\text{CS6-D}}$ / (RD/ $\overline{\text{WR-B}}$)/ $\overline{\text{ADTRG0-B}}$	PB6/ $\overline{\text{CS6-D}}$ / (RD/ $\overline{\text{WR-B}}$)/ $\overline{\text{ADTRG0-B}}$
45	P6	PC0/ $\overline{\text{CS3-B}}$ / $\overline{\text{WAIT-B}}$ / $\overline{\text{ADTRG1-B}}$	PC0/ $\overline{\text{CS3-B}}$ / $\overline{\text{WAIT-B}}$ / $\overline{\text{ADTRG1-B}}$	PC0/ $\overline{\text{CS3-B}}$ / $\overline{\text{WAIT-B}}$ / $\overline{\text{ADTRG1-B}}$
—	M7	NC	NC	NC
46	N7	PC1/ $\overline{\text{CS4-C}}$ / $\overline{\text{CS5-C}}$ / $\overline{\text{CS6-C}}$ / $\overline{\text{CS7-C}}$	PC1/ $\overline{\text{CS4-C}}$ / $\overline{\text{CS5-C}}$ / $\overline{\text{CS6-C}}$ / $\overline{\text{CS7-C}}$	PC1/ $\overline{\text{CS4-C}}$ / $\overline{\text{CS5-C}}$ / $\overline{\text{CS6-C}}$ / $\overline{\text{CS7-C}}$
47	R7	PC4/ $\overline{\text{ADTRG2}}$	PC4/ $\overline{\text{ADTRG2}}$	PC4/ $\overline{\text{ADTRG2}}$
48	P7	VSS	VSS	VSS
49	M8	P20/PO0/TIOCA3/TIOCB3/ TMRI0/SCK0/ $\overline{\text{IRQ8-A}}$	P20/PO0/TIOCA3/TIOCB3/ TMRI0/SCK0/ $\overline{\text{IRQ8-A}}$	P20/PO0/TIOCA3/ TMRI0/SCK0/ $\overline{\text{IRQ8-A}}$
50	N8	VCC	VCC	VCC
51	R8	P21/PO1/TIOCA3/TMCI0/ RxD0/ $\overline{\text{IRQ9-A}}$	P21/PO1/TIOCA3/TMCI0/ RxD0/ $\overline{\text{IRQ9-A}}$	P21/PO1/TIOCA3/ RxD0/ $\overline{\text{IRQ9-A}}$

56	P9	P30/PO8/TIOCA0/ DREQ0-B/EDREQ2* ³	P30/PO8/TIOCA0/ DREQ0-B/EDREQ2* ³	P30/PO8/TIOCA0/ DREQ0-B/EDREQ2* ³
57	M10	P31/PO9/TIOCA0/TIOCB0/ TEND0-B/ETEND2* ³	P31/PO9/TIOCA0/TIOCB0/ TEND0-B/ETEND2* ³	P31/PO9/TIOCA0/TIOCB0/ TEND0-B/ETEND2* ³
58	N10	P32/PO10/TIOCC0/TCLKA-A/ DACK0-B/EDACK2* ³	P32/PO10/TIOCC0/TCLKA-A/ DACK0-B/EDACK2* ³	P32/PO10/TIOCC0/TCLKA-A/ DACK0-B/EDACK2* ³
59	R10	P26/PO6/TIOCA5/TMO1/ TxD1/IRQ14	P26/PO6/TIOCA5/TMO1/ TxD1/IRQ14	P26/PO6/TIOCA5/TMO1/ TxD1/IRQ14
60	P10	P27/PO7/TIOCA5/TIOCB5/ IRQ15-A	P27/PO7/TIOCA5/TIOCB5/ IRQ15-A	P27/PO7/TIOCA5/TIOCB5/ IRQ15-A
61	N11	NMI	NMI	NMI
62	R11	P33/PO11/TIOCC0/TIOCD0/ TCLKB-A/DREQ1-B/ EDREQ3* ³	P33/PO11/TIOCC0/TIOCD0/ TCLKB-A/DREQ1-B/ EDREQ3* ³	P33/PO11/TIOCC0/TIOCD0/ TCLKB-A/DREQ1-B/ EDREQ3* ³
63	P11	P34/PO12/TIOCA1/ TEND1-B/ETEND3* ³	P34/PO12/TIOCA1/ TEND1-B/ETEND3* ³	P34/PO12/TIOCA1/ TEND1-B/ETEND3* ³
64	M11	VCC	VCC	VCC
65	R12	PH0/D0	PH0/D0	D0
66	P12	PH1/D1	PH1/D1	D1
67	N12	PH2/D2	PH2/D2	D2
68	R13	PH3/D3	PH3/D3	D3
69	M12	VSS	VSS	VSS
70	P13	PH4/D4	PH4/D4	D4
71	R14	PH5/D5	PH5/D5	D5
72	P14	PH6/D6	PH6/D6	D6

77	M14	P12/D10	P12/D10	P12/D10
78	L12	PI3/D11	PI3/D11	PI3/D11
79	M15	Vss	Vss	Vss
80	L13	PI4/D12	PI4/D12	PI4/D12
81	L14	PI5/D13	PI5/D13	PI5/D13
82	L15	PI6/D14	PI6/D14	PI6/D14
83	K12	PI7/D15	PI7/D15	PI7/D15
84	K13	P10/TxD2/DREQ0-A/ EDREQ0-A* ³ /IRQ0-A	P10/TxD2/DREQ0-A/ EDREQ0-A* ³ /IRQ0-A	P10/TxD2/DREQ0-A/ EDREQ0-A* ³ /IRQ0-A
85	K15	P11/RxD2/TEND0-A/ ETEND0-A* ³ /IRQ1-A	P11/RxD2/TEND0-A/ ETEND0-A* ³ /IRQ1-A	P11/RxD2/TEND0-A/ ETEND0-A* ³ /IRQ1-A
86	K14	P12/SCK2/DACK0-A/ EDACK0-A* ³ /IRQ2-A	P12/SCK2/DACK0-A/ EDACK0-A* ³ /IRQ2-A	P12/SCK2/DACK0-A/ EDACK0-A* ³ /IRQ2-A
87	J12	P13/ADTRG0-A/EDRAK0-A* ³ / IRQ3-A	P13/ADTRG0-A/EDRAK0-A* ³ / IRQ3-A	P13/ADTRG0-A/EDRAK0-A* ³ / EDRAK0-A* ³ /IRQ3-A
88	J13	Vss	Vss	Vss
—	J15	Vss	Vss	Vss
89	J14	P66/EDRAK0-B* ³	P66/EDRAK0-B* ³	P66/EDRAK0-B* ³
90	H12	P67/EDRAK1-B* ³ /IRQ15-B	P67/EDRAK1-B* ³ /IRQ15-B	P67/EDRAK1-B* ³ /IRQ15-B
—	H13	NC	NC	NC
—	H15	OSC2* ³	OSC2* ³	OSC2* ³
—	H14	OSC1* ³	OSC1* ³	OSC1* ³
91	G12	RES	RES	RES
92	G13	VCL	VCL	VCL

97	F15	XTAL	XTAL	XTAL
98	F14	EXTAL	EXTAL	EXTAL
—	E13	NC	NC	NC
99	E15	Vcc	Vcc	Vcc
100	E14	P16/TCLKC-B/ SCK3/SDA0/DACK1-A/ EDACK1-A* ³ /IRQ6-A	P16/TCLKC-B/ SCK3/SDA0/DACK1-A/ EDACK1-A* ³ /IRQ6-A	P16/TCLKC-B/ SCK3/SDA0/DA EDACK1-A* ³ /IR
101	E12	P17/TCLKD-B/ SCL0/ADTRG1-A/ EDRAK1-A* ³ /IRQ7-A	P17/TCLKD-B/ SCL0/ADTRG1-A/ EDRAK1-A* ³ /IRQ7-A	P17/TCLKD-B/ SCL0/ADTRG1- EDRAK1-A* ³ /IR
102	D15	STBY	STBY	STBY
103	D14	Vss	Vss	Vss
104	D13	P35/PO13/TIOCA1/TIOCB1/ TCLKC-A/DACK1-B/ EDACK3* ³	P35/PO13/TIOCA1/TIOCB1/ TCLKC-A/DACK1-B/ EDACK3* ³	P35/PO13/TIOCA TCLKC-A/DACK EDACK3* ³
105	C15	P36/PO14/TIOCA2/ EDRAK2* ³	P36/PO14/TIOCA2/ EDRAK2* ³	P36/PO14/TIOCA EDRAK2* ³
106	D12	P37/PO15/TIOCA2/TIOCB2/ TCLKD-A/EDRAK3* ³	P37/PO15/TIOCA2/TIOCB2/ TCLKD-A/EDRAK3* ³	P37/PO15/TIOCA TCLKD-A/EDRA
107	C14	P60/TMRI2/TxD4/DERQ2/ EDREQ0-B* ³ /IRQ8-B	P60/TMRI2/TxD4/DERQ2/ EDREQ0-B* ³ /IRQ8-B	P60/TMRI2/TxD EDREQ0-B* ³ /IR
108	B15	P61/TMC12/RxD4/TEND2/ET END0-B* ³ /IRQ9-B	P61/TMC12/RxD4/TEND2/ET END0-B* ³ /IRQ9-B	P61/TMC12/RxD ETEND0-B* ³ /IR
109	B14	P62/TMO2/SCK4/DACK2/ EDACK0-B* ³ / IRQ10-B	P62/TMO2/SCK4/DACK2/ EDACK0-B* ³ /TRST* ²	P62/TMO2/SCK EDACK0-B* ³
110	A15	PLLvcc	PLLvcc	PLLvcc

114	A13	P65/TMO3/SCK6/DACK3/ EDACK1-B* ³ / IRQ13-B	P65/TMO3/SCK6/DACK3/ EDACK1-B* ³ / IRQ13-B/TCK* ²	P65/TMO3/SCK6/ EDACK1-B* ³ / IRQ13-B
115	B12	MD0	MD0	MD0
116	D11	PC2/LUCAS/DQMLU* ³	PC2/LUCAS/DQMLU* ³	PC2/LUCAS/DQM
117	A12	PC3/LLCAS/DQMLL* ³	PC3/LLCAS/DQMLL* ³	PC3/LLCAS/DQM
—	C11	NC	NC	NC
118	B11	P50/AN0/IRQ0-B	P50/AN0/IRQ0-B	P50/AN0/IRQ0-B
119	A11	P51/AN1/IRQ1-B	P51/AN1/IRQ1-B	P51/AN1/IRQ1-B
120	D10	P52/AN2/IRQ2-B	P52/AN2/IRQ2-B	P52/AN2/IRQ2-B
—	C10	NC	NC	NC
121	A10	Avcc	Avcc	Avcc
122	B10	P53/AN3/IRQ3-B	P53/AN3/IRQ3-B	P53/AN3/IRQ3-B
123	D9	Avss	Avss	Avss
124	C9	P54/AN4/IRQ4-B	P54/AN4/IRQ4-B	P54/AN4/IRQ4-B
125	A9	Vref	Vref	Vref
—	B9	NC	NC	NC
126	D8	P55/AN5/IRQ5-B	P55/AN5/IRQ5-B	P55/AN5/IRQ5-B
127	C8	P56/AN6/DA0/IRQ6-B	P56/AN6/DA0/IRQ6-B	P56/AN6/DA0/IRQ
128	A8	P57/AN7/DA1/IRQ7-B	P57/AN7/DA1/IRQ7-B	P57/AN7/DA1/IRQ
129	B8	P44/AN8	P44/AN8	P44/AN8
130	D7	P45/AN9	P45/AN9	P45/AN9
131	C7	P46/AN10	P46/AN10	P46/AN10
132	A7	P47/AN11	P47/AN11	P47/AN11

135	A3	PA1/BAON($\overline{\text{RD}}$ /WTPA)	PA1/BAON($\overline{\text{RD}}$ /WTPA)	PA1/BAON($\overline{\text{RD}}$ /WTPA)
136	B5	PA2/BREQ/WAIT-A	PA2/BREQ/WAIT-A	PA2/BREQ/WAIT-A
137	D5	PA3/LLWR/LLB	PA3/LLWR/LLB	LLWR/LLB
138	A4	PA4/LHWR/LUB	PA4/LHWR/LUB	PA4/LHWR/LUB
139	B4	PA5/ $\overline{\text{RD}}$	PA5/ $\overline{\text{RD}}$	$\overline{\text{RD}}$
140	C4	PA6/AS/AH/BS-B	PA6/AS/AH/BS-B	PA6/AS/AH/BS-B
—	A3	Vcc	Vcc	Vcc
141	D4	Vss	Vss	Vss
142	B3	PA7/B ϕ	PA7/B ϕ	PA7/B ϕ
143	A2	Vcc	Vcc	Vcc
144	B2	PB0/CS0/CS4-A/ $\overline{\text{CS5}}$ -B	PB0/CS0/CS4-A/CS5-B	PB0/CS0/CS4-A/CS5-B

- Notes:
1. These pins can be used when the PCJKE bit in PFCRD is set to 1 in single-chip mode.
 2. Pins TDO, TRST, TMS, TDI, and TCK are enabled in mode 3.
 3. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.
 4. The $\overline{\text{IRQ15}}$ -A is not supported by the H8SX/1648G Group and the H8SX/1648H Group.

	PLL _{CC} V _{CC}	Input	Power supply pin for the PLL circuit.
	PLL _{SS} V _{SS}	Input	Ground pin for the PLL circuit.
Clock	XTAL	Input	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin. For an example connection, see section 26, Clock Pulse Generator.
	EXTAL	Input	
	OSC1* ³	Input	The 32.768 kHz crystal resonator is connected to this pin.
	OSC2* ³	Input	The 32.768 kHz crystal resonator is connected to this pin.
	B ϕ	Output	Outputs the system clock for external devices.
	SDRAM ϕ * ³	Output	When connecting the synchronous DRAM, connect this pin to the CLK pin of synchronous DRAM. For details, see section 9, Bus Controller (BSC).
Operating mode control	MD3 to MD0	Input	Pins for setting the operating mode. The signal level of these pins must not be changed during operation.
System control	$\overline{\text{RES}}$	Input	Reset signal input pin. This LSI enters the reset state when this signal goes low.
	$\overline{\text{STBY}}$	Input	This LSI enters hardware standby mode when this signal goes low.
	EMLE	Input	Input pin for the on-chip emulator enable signal. When the on-chip emulator is used, the signal level should be fixed high. If the on-chip emulator is not used, the signal level should be fixed low.
On-chip emulator	$\overline{\text{TRST}}$	Input	On-chip emulator pins or boundary scan pins. When the EMLE pin is driven high, these pins are dedicated for the on-chip emulator.
	TMS	Input	
	TDI	Input	
	TCK	Input	
	TDO	Output	
Address bus	A23 to A0	Output	Output pins for the address bits.

$\overline{\text{BS-A}}/\overline{\text{BS-B}}$	Output	Indicates the start of a bus cycle.
$\overline{\text{AS}}$	Output	Strobe signal which indicates that the output of the address bus is valid in access to the basic bus interface or byte control SRAM interface space.
$\overline{\text{AH}}$	Output	This signal is used to hold the address when the address-data multiplexed I/O interface space is accessed.
$\overline{\text{RD}}$	Output	Strobe signal which indicates that reading from the bus interface space is in progress.
$\overline{\text{RD}}/\overline{\text{WR}}$	Output	Indicates the direction (input or output) of the bus interface space.
$\overline{\text{LHWR}}$	Output	Strobe signal which indicates that the higher-order address (D15 to D8) is valid in access to the basic bus interface space.
$\overline{\text{LLWR}}$	Output	Strobe signal which indicates that the lower-order address (D7 to D0) is valid in access to the basic bus interface space.
$\overline{\text{LUB}}$	Output	Strobe signal which indicates that the higher-order address (D15 to D8) is valid in access to the byte control SRAM interface space.
$\overline{\text{LLB}}$	Output	Strobe signal which indicates that the lower-order address (D7 to D0) is valid in access to the byte control SRAM interface space.

WAIT-A/ WAIT-B	Input	Requests wait cycles in access to the external
$\overline{\text{RAS}}^{*3}$	Output	<ul style="list-style-type: none"> Row address strobe signal when area 2 is as DRAM interface space Row address strobe signal when area 2 is as synchronous DRAM space
$\overline{\text{CAS}}^{*3}$	Output	Column address strobe signal when area 2 is as synchronous DRAM interface space
$\overline{\text{WE}}^{*3}$	Output	<ul style="list-style-type: none"> Write enable signal for DRAM space Write enable signal when area 2 is specific synchronous DRAM interface space
$\overline{\text{OE/CKE}}^{*3}$	Output	<ul style="list-style-type: none"> Output enable signal for DRAM interface space Clock enable signal for synchronous DRAM interface space
$\overline{\text{LUCAS}}^{*3}$	Output	Upper column address strobe signal for 16-bit DRAM interface space
$\overline{\text{LLCAS}}^{*3}$	Output	<ul style="list-style-type: none"> Lower-column address strobe signal for 16-bit DRAM interface space Column address strobe signal for 8-bit DRAM interface space
$\overline{\text{DQMLU}}^{*3}$	Output	Upper-data mask enable signal for 16-bit synchronous DRAM interface space
$\overline{\text{DQMLL}}^{*3}$	Output	<ul style="list-style-type: none"> Lower-data mask enable signal for 16-bit synchronous DRAM interface space Data mask enable signal for 8-bit synchronous DRAM interface space

$\overline{\text{IRQ6-A}}/\overline{\text{IRQ6-B}}$
 $\overline{\text{IRQ5-A}}/\overline{\text{IRQ5-B}}$
 $\overline{\text{IRQ4-A}}/\overline{\text{IRQ4-B}}$
 $\overline{\text{IRQ3-A}}/\overline{\text{IRQ3-B}}$
 $\overline{\text{IRQ2-A}}/\overline{\text{IRQ2-B}}$
 $\overline{\text{IRQ1-A}}/\overline{\text{IRQ1-B}}$
 $\overline{\text{IRQ0-A}}/\overline{\text{IRQ0-B}}$

DMA controller (DMAC)	$\overline{\text{DREQ0-A}}/\overline{\text{DREQ0-B}}$ $\overline{\text{DREQ1-A}}/\overline{\text{DREQ1-B}}$ $\overline{\text{DREQ2}}$ $\overline{\text{DREQ3}}$	Input	Requests DMAC activation.
	$\overline{\text{DACK0-A}}/\overline{\text{DACK0-B}}$ $\overline{\text{DACK1-A}}/\overline{\text{DACK1-B}}$ $\overline{\text{DACK2}}$ $\overline{\text{DACK3}}$	Output	DMAC single address-transfer acknowledge
	$\overline{\text{TEND0-A}}/\overline{\text{TEND0-B}}$ $\overline{\text{TEND1-A}}/\overline{\text{TEND1-B}}$ $\overline{\text{TEND2}}$ $\overline{\text{TEND3}}$	Output	Indicates end of data transfer by the DMAC.
EXMDA controller (EXDMAC)*-3	$\overline{\text{EDREQ0-A}}/\overline{\text{EDREQ0-B}}$ $\overline{\text{EDREQ1-A}}/\overline{\text{EDREQ1-B}}$ $\overline{\text{EDREQ2}}$ $\overline{\text{EDREQ3}}$	Input	Requests EXDMAC activation.
	$\overline{\text{EDACK0-A}}/\overline{\text{EDACK0-B}}$ $\overline{\text{EDACK1-A}}/\overline{\text{EDACK1-B}}$ $\overline{\text{EDACK2}}$ $\overline{\text{EDACK3}}$	Output	EXDMAC single address-transfer acknowledge
	$\overline{\text{ETEND0-A}}/\overline{\text{ETEND0-B}}$ $\overline{\text{ETEND1-A}}/\overline{\text{ETEND1-B}}$ $\overline{\text{ETEND2}}$ $\overline{\text{ETEND3}}$	Output	Indicates end of data transfer by the EXDMAC.

TIOCC0		output.
TIOCD0		
TIOCA1 TIOCB1	Input/ output	Signals for TGRA_1 and TGRB_1. These pins as input capture inputs, output compare outputs.
TIOCA2 TIOCB2	Input/ output	Signals for TGRA_2 and TGRB_2. These pins as input capture inputs, output compare outputs.
TIOCA3 TIOCB3 TIOCC3 TIOCD3	Input/ output	Signals for TGRA_3 to TGRD_3. These pins as input capture inputs, output compare outputs, outputs.
TIOCA4 TIOCB4	Input/ output	Signals for TGRA_4 and TGRB_4. These pins as input capture inputs, output compare outputs.
TIOCA5 TIOCB5	Input/ output	Signals for TGRA_5 and TGRB_5. These pins as input capture inputs, output compare outputs.
TCLKE TCLKF TCLKG TCLKH	Input	Input pins for external clock signals.
TIOCA6 TIOCB6 TIOCC6 TIOCD6	Input/ output	Signals for TGRA_6 to TGRD_6. These pins as input capture inputs, output compare outputs, outputs.
TIOCA7 TIOCB7	Input/ output	Signals for TGRA_7 and TGRB_7. These pins as input capture inputs, output compare outputs.

	TIOCA11 TIOCB11	Input/ output	Signals for TGRA_11 and TGRB_11. These pins are used as input capture inputs, output compare inputs, and PWM outputs.
Programmable pulse generator (PPG)	PO31 to PO0	Output	Output pins for the pulse signals.
8-bit timer (TMR)	TMO0 to TMO3	Output	Output pins for the compare match signals.
	TMCI0 to TMCI3	Input	Input pins for the external clock signals that are used by the counters.
	TMRI0 to TMRI3	Input	Input pins for the counter-reset signals.
Watchdog timer (WDT)	WDTOVF	Output	Output pin for the counter-overflow signal in watchdog timer mode.
Serial communication interface (SCI)	TxD0	Output	Output pins for data transmission.
	TxD1		
	TxD2		
	TxD3		
	TxD4		
	TxD5		
	TxD6		
	RxD0	Input	Input pins for data reception.
	RxD1		
	RxD2		
	RxD3		
	RxD4		
	RxD5		
RxD6			

2 (IIC2)	SCL1	output	by the NMOS open drain output. SCL2 and SCL3
	SCL2 SCL3		5-V input.
	SDA0 SDA1 SDA2 SDA3	Input/ output	Input/output pin for IIC data. Bus can be direct to the NMOS open drain output. SDA2 and SDA3 are 5-V input.
A/D converter	AN11 to AN0	Input	Input pins for the analog signals to be processed by the A/D converter.
	ADTRG0-A/ ADTRG0-B ADTRG1-A/ ADTRG1-B ADTRG2	Input	Input pins for the external trigger signal that starts the A/D conversion.
D/A converter	DA1, DA0	Output	Output pins for the analog signals from the D/A converter.
A/D converter, D/A converter	AV _{CC}	Input	Analog power supply pin for the A/D and D/A converters. When the A/D and D/A converters are not in use, connect this pin to the system power supply.
	AV _{SS}	Input	Ground pin for the A/D and D/A converters. Connect this pin to the system power supply (0 V).
	Vref	Input	Reference power supply pin for the A/D and D/A converters. When the A/D and D/A converters are not in use, connect this pin to the system power supply.
	P17 to P10	Input/ output	8-bit input/output pins.
	P27 to P20	Input/ output	8-bit input/output pins.

PB7 to PB0	Input/ output	8-bit input/output pins.
PC5 to PC0	Input/ output	6-bit input/output pins.
PD7 to PD0	Input/ output	8-bit input/output pins.
PE7 to PE0	Input/ output	8-bit input/output pins.
PF7 to PF0	Input/ output	8-bit input/output pins.
PH7 to PH0	Input/ output	8-bit input/output pins.
PI7 to PI0	Input/ output	8-bit input/output pins.
PN3 to PN0	Input/ output	4-bit input/output (open drain) pins.
PM4 to PM0* ³	Input/ output	5-bit input/output pins
PJ7 to PJ0* ¹	Input/ output	8-bit input/output pins.
PK7 to PK0* ¹	Input/ output	8-bit input/output pins.

- Note:
1. These pins can be used when the PCJKE bit in PFCRD is set to 1 in single-chip mode.
 2. The $\overline{\text{IRQ15-A}}$ is not supported by the H8SX/1648G Group and the H8SX/1648H Group.
 3. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

- Upward-compatible with H8/500, H8/500H, and H8S CPUs
 - Can execute object programs of these CPUs
- 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, @-ERn, @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

- Notes:
1. Advanced mode is only supported as the CPU operating mode of the H8SX/1648 Group, the H8SX/1648A Group, the H8SX/1648L Group, the H8SX/1648G Group, and the H8SX/1648H Group. Normal, middle, and maximum modes are not supported.
 2. The multiplier and divider are supported by the H8SX/1648 Group, the H8SX/1648A Group, the H8SX/1648L Group, the H8SX/1648G Group, and the H8SX/1648H Group.

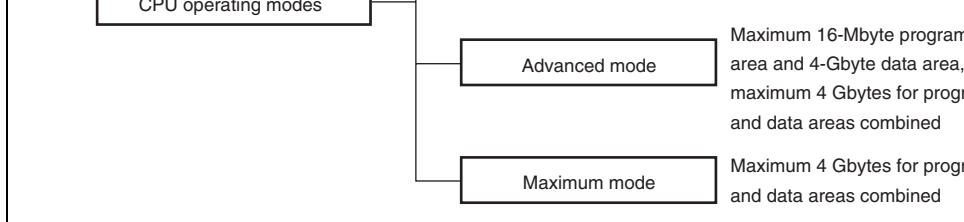


Figure 2.1 CPU Operating Modes

2.2.1 Normal Mode

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

- **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 register, it cannot contain any value, even when the corresponding general register Rn is used as an address register. (If the general register Rn is referenced in the register indirect addressing mode, pre-/post-increment or pre-/post-decrement and a carry or borrow occurs, however, the corresponding extended register En will be affected.)
- **Instruction Set**
All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid.

Figure 2.2 Exception Vector Table (Normal Mode)

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressing are used in the JMP and JSR instructions. An 8-bit absolute address included in the instruction 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 exception handling are shown in figure 2.3. The PC contents are saved or restored in 16-bit units

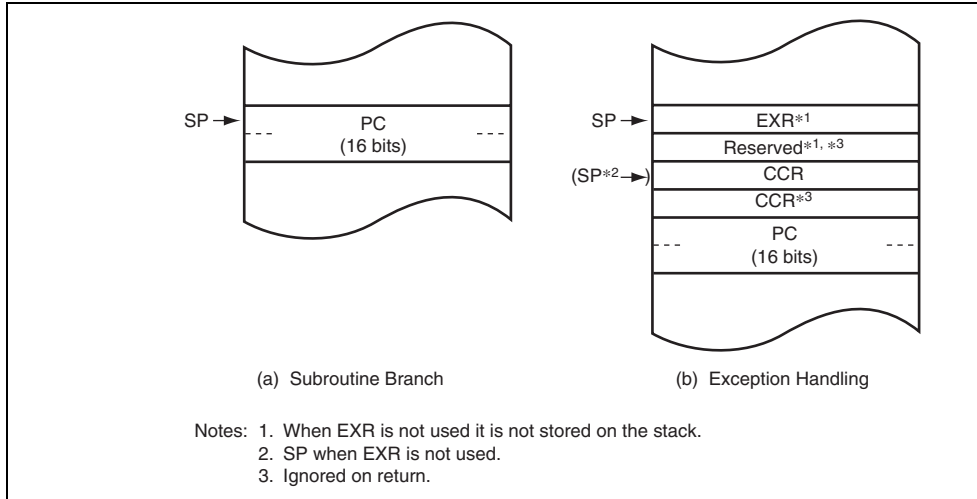


Figure 2.3 Stack Structure (Normal Mode)

The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers. When the extended register En is used as a 16-bit register (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 register referenced in the register indirect addressing mode with pre-/post-increment or pre-/post-decrement and a carry or borrow occurs, however, the value in the corresponding extended register En will be affected.)

- **Instruction Set**

All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid 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 vector table. One branch address is stored per 32 bits. The upper eight bits are ignored and the lower 24 bits 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) addressing modes are used in the JMP and JSR instructions. An 8-bit absolute address included in the instruction code specifies a memory location. Execution branches to the contents of the memory location.

In middle mode, an operand is a 32-bit (longword) operand, providing a 32-bit branch address. 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 exception handling are shown in figure 2.5. The PC contents are saved or restored in 24-bit units.

- Instruction Set

All instructions and addressing modes can be used.

- Exception Vector Table and Memory Indirect Branch Addresses

In advanced mode, the top area starting at H'00000000 is allocated to the exception vector table. One branch address is stored per 32 bits. The upper eight bits are ignored and the lower 24 bits are stored. The structure of the exception vector table is shown in figure 2.4.

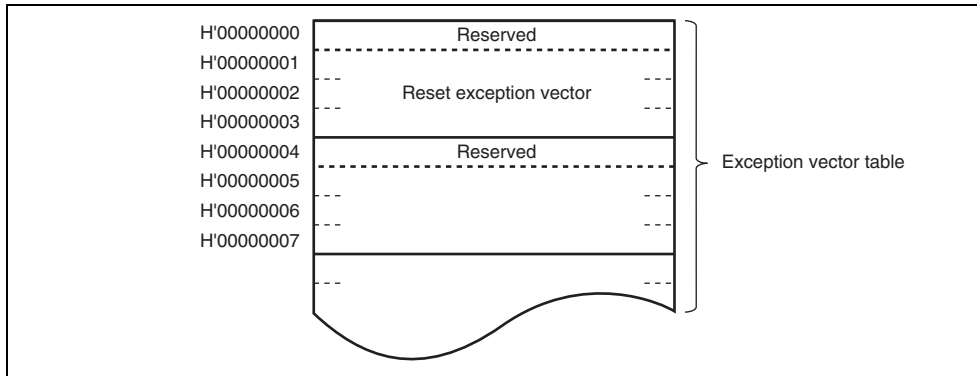


Figure 2.4 Exception Vector Table (Middle and Advanced Modes)

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

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

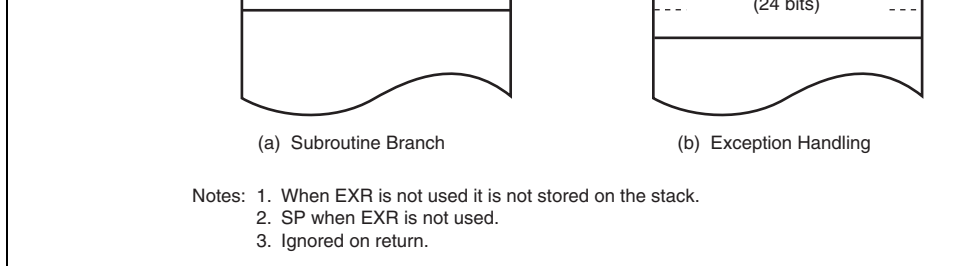


Figure 2.5 Stack Structure (Middle and Advanced Modes)

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-bit segments of 32-bit registers or address registers.

- Instruction Set

All instructions and addressing modes can be used.

- Exception Vector Table and Memory Indirect Branch Addresses

In 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 is shown in figure 2.6.

Figure 2.6 Exception Vector Table (Maximum Modes)

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressing are used in the JMP and JSR instructions. An 8-bit absolute address included in the instruction code specifies a memory location. Execution branches to the contents of the memory location. In maximum mode, an operand is a 32-bit (longword) operand, providing a 32-bit branch address.

- Stack Structure

The stack structure of PC at a subroutine branch and that of PC and CCR at an exception 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.

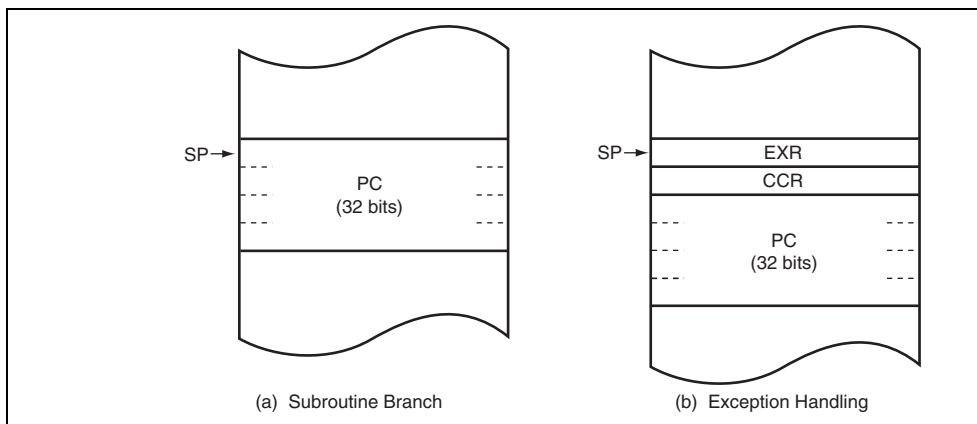


Figure 2.7 Stack Structure (Maximum Mode)

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

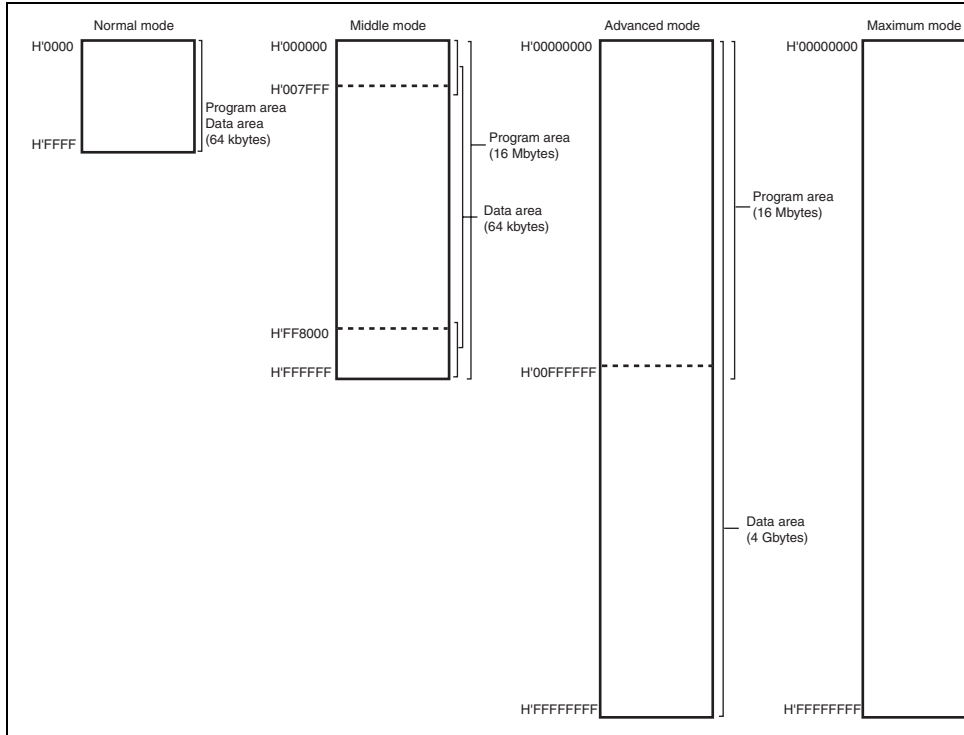
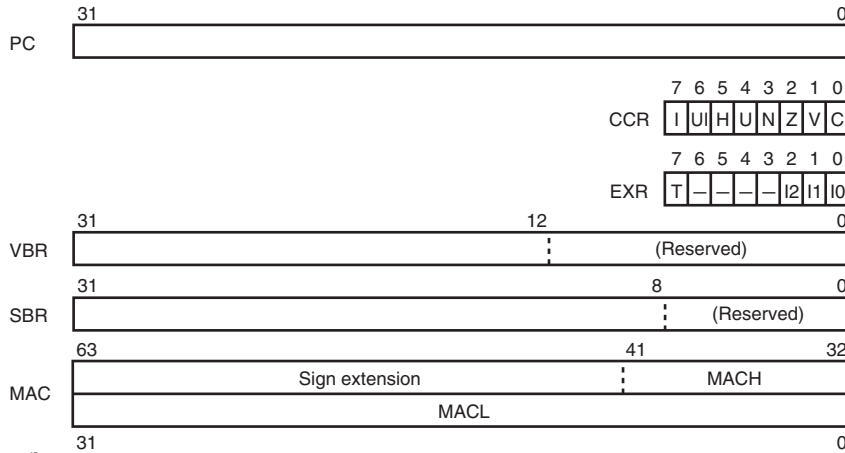


Figure 2.8 Memory Map

ER0	E0	R0H	R0L
ER1	E1	R1H	R1L
ER2	E2	R2H	R2L
ER3	E3	R3H	R3L
ER4	E4	R4H	R4L
ER5	E5	R5H	R5L
ER6	E6	R6H	R6L
ER7 (SP)	E7	R7H	R7L

Control Registers



[Legend]

- | | | |
|------------------------------------|--------------------------------|-----------------------------------|
| SP: Stack pointer | U: User bit | T: Trace bit |
| PC: Program counter | N: Negative flag | I2 to I0: Interrupt mask bits |
| CCR: Condition-code register | Z: Zero flag | VBR: Vector base register |
| I: Interrupt mask bit | V: Overflow flag | SBR: Short address base register |
| UI: User bit or interrupt mask bit | C: Carry flag | MAC: Multiply-accumulate register |
| H: Half-carry flag | EXR: Extended control register | |

Figure 2.9 CPU Registers

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

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

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

The usage of each register can be selected independently.

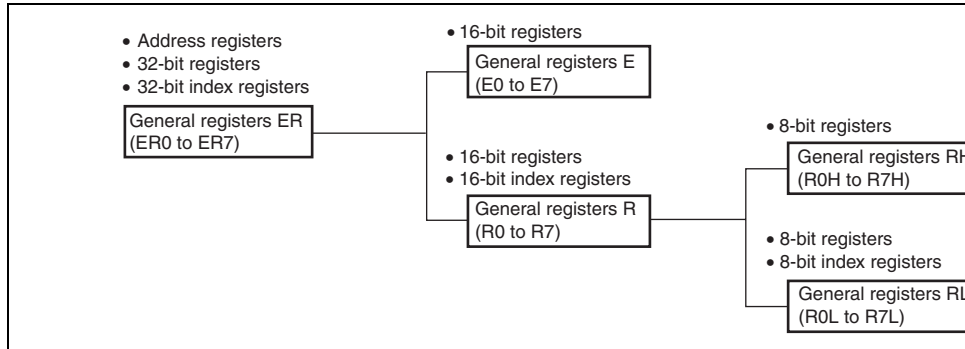


Figure 2.10 Usage of General Registers



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 execute. The length of all CPU instructions is 16 bits (one word) or a multiple of 16 bits, so the least significant bit is ignored. (When the instruction code is fetched, the least significant bit is regarded as a zero.)

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 at the start of an exception handling.
6	UI	Undefined	R/W	User Bit Can be written to and read from by software using LDC, STC, ANDC, ORC, and XORC instructions.
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 set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.
4	U	Undefined	R/W	User Bit Can be written to and read from by software using LDC, STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit (representing the 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 accumulator for bit 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 I0).

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

For details, see section 6, 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 generated each time an instruction is executed. When this bit is cleared to 0, instructions are executed in sequence.
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 (I2 to I0).
0	I0	1	R/W	

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

2.5.7 Multiply-Accumulate Register (MAC)

MAC is a 64-bit register that stores the results of multiply-and-accumulate operations. It is composed of two 32-bit registers denoted MACH and MACL. The lower 10 bits of MACH are valid, and the 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 the I bits in EXR to 0, and sets the I bits in CCR and EXR to 1. The general registers, MAC, and the lower 10 bits in CCR are not initialized. In particular, the initial value of the stack pointer (ER7) is 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.

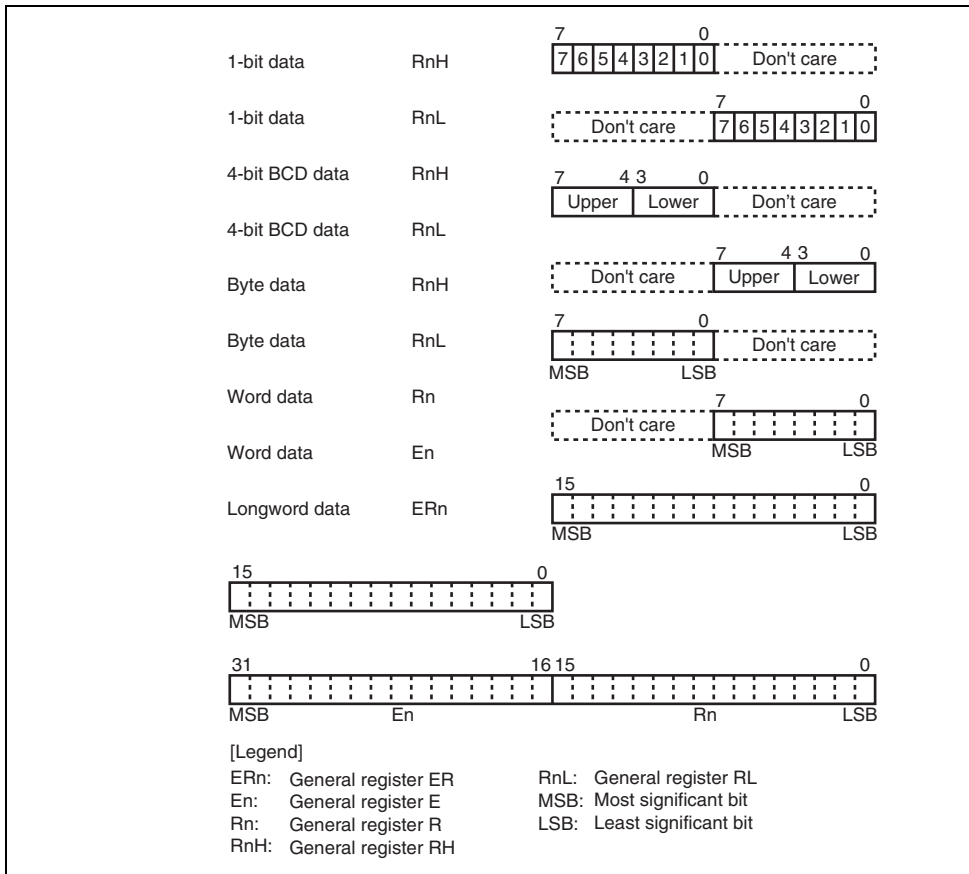


Figure 2.12 General Register Data Formats

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 should be byte size or longword size.

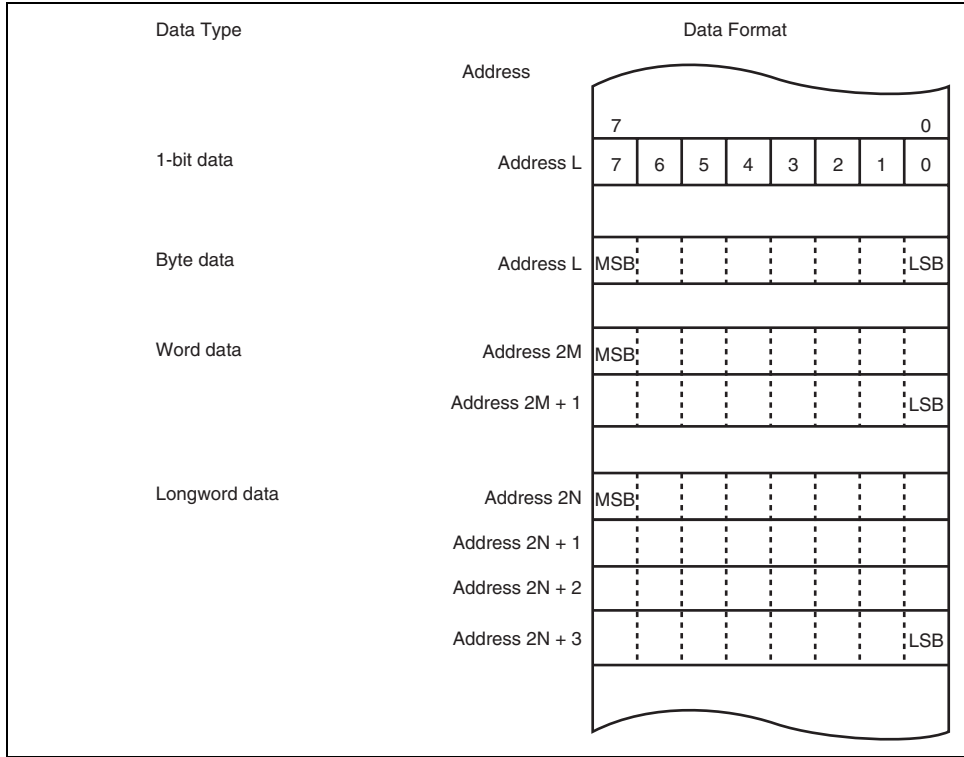


Figure 2.13 Memory Data Formats

	POP, PUSH* ¹	W/L
	LDM, STM	L
	MOVA	B/W* ²
Block transfer	EEMOV	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* ⁶ , MULS/U* ⁶	L
	EXTU, EXTS	W/L
	TAS	B
	MAC* ⁶	—
	LDMAC* ⁶ , STMAC* ⁶	—
	CLRMAC* ⁶	—
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, BAND, 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.L @-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. Only when the multiplier is available.

Data transfer	MOV	B/W/L	S	SD	SD	SD	SD	SD	SD
		B		S/D				S/D	
	MOVFP, MOVTP	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
		B		S	D	D	D	D	D
		B		D	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
		B		S	D	D	D	D	D
		B		D	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* ¹²	—							
	CLRMAC* ¹²	—							
	LDMAC* ¹²	—	S						
	STMAC* ¹²	—	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
		W/L* ⁶	D	D	D	D	D		D
		B/WL* 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 manipulation	BSET, BCLR, BNOT, BTST, BSET/cc, BCLR/cc	B	D	D				D	D
		BAND, BAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST, BSTZ, BISTZ	B	D	D				D

(VBR, SBR)						
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 data 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	—		0				
BRA	—		0	0			
BRA/S	—		0*				
JMP	—	0			0	0	0
BSR	—		0				
JSR	—	0			0	0	0
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 registers (R8 to R7), and 32-bit registers (ER0 to ER7).

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

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 specified by ER5 to a memory location specified by ER6. The number of longword 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	B/W/L	$Rd \pm 1 \rightarrow Rd, Rd \pm 2 \rightarrow Rd$
DEC		Increments or decrements a general register by 1 or 2. (Byte operations can be incremented or decremented by 1 only.)
ADDS	L	$Rd \pm 1 \rightarrow Rd, Rd \pm 2 \rightarrow Rd, Rd \pm 4 \rightarrow Rd$
SUBS		Adds or subtracts the value 1, 2, or 4 to or from data in a general register.
DAA	B	Rd (decimal adjust) $\rightarrow Rd$
DAS		Decimal-adjusts an addition or subtraction result in a general register 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 registers: 8 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 registers: 8 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 registers: 8 bits \times 32 bits \rightarrow upper 32 bits).
MULXS	B/W	$Rd \times Rs \rightarrow Rd$ Performs signed multiplication on data in two general registers: 8 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: 8 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: 8 bits \times 32 bits \rightarrow upper 32 bits).
DIVXU	B/W	$Rd \div Rs \rightarrow Rd$ Performs unsigned division on data in two general registers: either 8 bits \div 8 bits \rightarrow 8-bit quotient and 8-bit remainder, or 32 bits \div 16 bits \rightarrow 16-bit quotient and 16-bit remainder.

		Compares data between immediate data, general registers, and and stores the result in CCR.
NEG	B/W/L	$0 - (\text{EAd}) \rightarrow (\text{EAd})$ Takes the two's complement (arithmetic complement) of data in a register or the contents of a memory location.
EXTU	W/L	$(\text{EAd}) (\text{zero extension}) \rightarrow (\text{EAd})$ Performs zero-extension on the lower 8 or 16 bits of data in a general register or memory to word or longword size. The lower 8 bits to word or longword, or the lower 16 bits to longword, will be zero-extended.
EXTS	W/L	$(\text{EAd}) (\text{sign extension}) \rightarrow (\text{EAd})$ Performs sign-extension on the lower 8 or 16 bits of data in a general register or memory to word or longword size. The lower 8 bits to word or longword, or the lower 16 bits to longword, will be sign-extended.
TAS	B	$@\text{ERd} - 0, 1 \rightarrow (<\text{bit } 7> \text{ of } @\text{EAd})$ Tests memory contents, and sets the most significant bit (bit 7) to 1 if the contents are not zero.
MAC*	—	$(\text{EAs}) \times (\text{EAd}) + \text{MAC} \rightarrow \text{MAC}$ Performs signed multiplication on memory contents and adds the result to the MAC.
CLRMAC*	—	$0 \rightarrow \text{MAC}$ Clears MAC to zero.
LDMAC*	—	$\text{Rs} \rightarrow \text{MAC}$ Loads data from a general register to MAC.
STMAC*	—	$\text{MAC} \rightarrow \text{Rd}$ Stores data from MAC to a general register.

Note: * When supporting multiplier only

Performs a logical exclusive OR operation on data between immediate data, general registers, and memory.

NOT	B/W/L	\sim (EAd) \rightarrow (EAd) Takes the one's complement of the contents of a general register or a memory location.
-----	-------	--

Table 2.8 Shift Operation Instructions

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

BCLR	B	$0 \rightarrow (\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle)$ Clears a specified bit in the contents of a general register or a memory location to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BCLR/cc	B	if cc, $0 \rightarrow (\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle)$ If the specified condition is satisfied, this instruction clears a specified bit in a memory location to 0. The bit number can be specified by 3-bit immediate data, or by the lower three bits of a general register. The condition status can be specified as a condition.
BNOT	B	$\sim (\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle) \rightarrow (\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle)$ Inverts a specified bit in the contents of a general register or a memory location. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BTST	B	$\sim (\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle) \rightarrow Z$ Tests a specified bit in the contents of a general register or a memory location and sets or clears the Z flag accordingly. The bit number can be specified by 3-bit immediate data or the lower three bits of a general register.
BAND	B	$C \wedge (\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle) \rightarrow C$ ANDs the carry flag with a specified bit in the contents of a general register or a memory location and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BIAND	B	$C \wedge [\sim (\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle)] \rightarrow C$ ANDs the carry flag with the inverse of a specified bit in the contents of a general register or a memory location and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BOR	B	$C \vee (\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle) \rightarrow C$ ORs the carry flag with a specified bit in the contents of a general register or a memory location and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

Exclusive-ORs the carry flag with the inverse of a specified bit in the contents of a general register or a memory location and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

BLD	B	$(\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle) \rightarrow C$ Transfers a specified bit in the contents of a general register or a memory location to the carry flag. The bit number is specified by 3-bit immediate data.
BILD	B	$\sim (\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle) \rightarrow C$ Transfers the inverse of a specified bit in the contents of a general register or a memory location to the carry flag. The bit number is specified by 3-bit immediate data.
BST	B	$C \rightarrow (\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle)$ Transfers the carry flag value to a specified bit in the contents of a general register or a memory location. The bit number is specified by 3-bit immediate data.
BSTZ	B	$Z \rightarrow (\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle)$ Transfers the zero flag value to a specified bit in the contents of a general register or a memory location. The bit number is specified by 3-bit immediate data.
BIST	B	$\sim C \rightarrow (\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle)$ Transfers the inverse of the carry flag value to a specified bit in the contents of a general register or a memory location. The bit number is specified by 3-bit immediate data.

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 specified address.
Bcc	—	Branches to a specified address if the specified condition is satisfied.
BRA/S	—	Branches unconditionally to a specified address after executing the current instruction. The next instruction should be a 1-word instruction except for 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 multiple 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	$R_s \rightarrow VBR, R_s \rightarrow SBR$ Transfers the general register contents to VBR or SBR.
STC	B/W	$CCR \rightarrow (EAd), EXR \rightarrow (EAd)$ 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	$VBR \rightarrow Rd, SBR \rightarrow Rd$ Transfers the contents of VBR or SBR to a general register.
ANDC	B	$CCR \wedge \#IMM \rightarrow CCR, EXR \wedge \#IMM \rightarrow EXR$ Logically ANDs the CCR or EXR contents with immediate data.
ORC	B	$CCR \vee \#IMM \rightarrow CCR, EXR \vee \#IMM \rightarrow EXR$ Logically ORs the CCR or EXR contents with immediate data.
XORC	B	$CCR \oplus \#IMM \rightarrow CCR, EXR \oplus \#IMM \rightarrow EXR$ Logically exclusive-ORs the CCR or EXR contents with immedi
NOP	—	$PC + 2 \rightarrow PC$ Only increments the program counter.

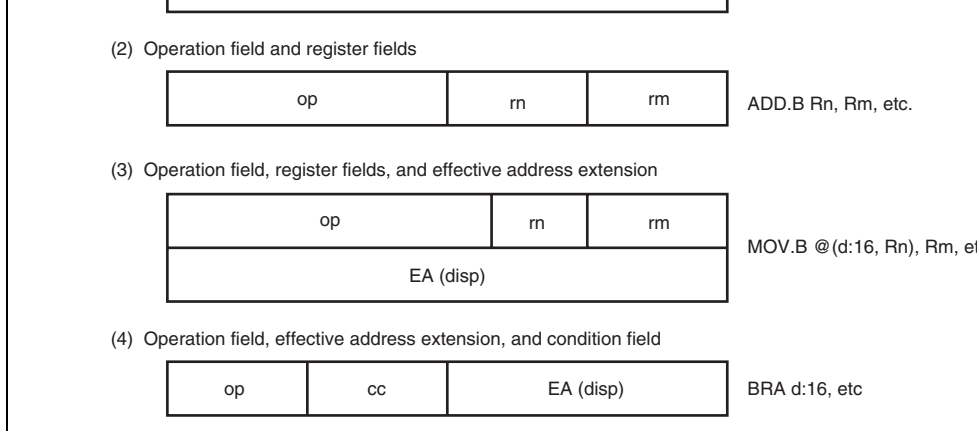


Figure 2.14 Instruction Formats

- **Operation Field**
Indicates the function of the instruction, and specifies the addressing mode and operation carried out on the operand. The operation field always includes the first four bits of the instruction. Some instructions have two operation fields.
- **Register Field**
Specifies a general register. Address registers are specified by 3 bits, data registers by 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,ERn)
4	Index register indirect with displacement	@(d:16, RnL.B)/@(d:16,Rn.W)/@(d:32, RnL.B)/@(d:32,Rn.W)/@(d:32,ERn)
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.W,PC)
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 specified by the 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 the contents of an address register (ERn) and a 16- or 32-bit displacement. ERn is specified by the register field of the instruction code. The displacement is included in the instruction code. The 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 when the displacement is 1, 2, or 3 and the operand is byte data, when the displacement is 2, 4, or 6 and the operand is word data, or when the displacement is 4, 8, or 12 and the operand is longword data.

2.8.4 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)

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

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

- Register indirect with post-decrement—@ERn-

The operand value is the contents of a memory location which is pointed to by the effective address. The operation result: the contents of an address register (ERn). ERn is specified by the register field of the instruction code. After the operand value is accessed, 1, 2, or 4 is subtracted from the address register contents. The remainder is stored in the address register. The value subtracted is 1 for byte access, 2 for 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 and multiple effective addresses are calculated, the contents of the general register after the first calculation of 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 bits (@aa:32) is used. For an 8-bit absolute address, the upper 24 bits are specified by SBR. For a 16-bit absolute address, the upper 16 bits are sign-extended. A 32-bit absolute address can access the entire address space.

To access the program area, the absolute address of 24 bits (@aa:24) or 32 bits (@aa:32) is used. 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	Maximum Mode	
Data area	8 bits (@aa:8)	A consecutive 256-byte area (the upper address is set in SBR)			
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF	H'00000000 to H'00007FFF, H'FFFFFF8000 to H'FFFFFF	
	32 bits (@aa:32)		H'FF8000 to H'FFFFFF	H'00000000 to H'FFFFFF	
Program area	24 bits (@aa:24)		H'000000 to H'FFFFFF	H'00000000 to H'00FFFFFF	
	32 bits (@aa:32)			H'00000000 to H'00007FFF, H'00FFFFFF to H'FFFFFF	

manipulation instructions contain 3-bit immediate data in the instruction code, for specifying a branch displacement. The BFLD and BFST instructions contain 8-bit immediate data in the instruction code, for specifying a bit field. The TRAPA instruction contains 2-bit immediate data in the instruction 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 address which is the sum of an 8- or 16-bit displacement in the instruction code and the 32-bit address of the PC contents. The 8-bit or 16-bit displacement is sign-extended to 32 bits when added to the PC contents. The PC contents to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is -126 to $+128$ bytes (-63 to $+64$ words) or -32766 to $+32768$ bytes (-16383 to $+16384$ words) from the branch instruction. The operand value should be an even number. In advanced mode, only the lower 24 bits of this branch address 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.V, PC) or @(ERn.L, PC)

This mode is used in the Bcc and BSR instructions. The operand value is a 32-bit branch address which is the sum of the following operation result and the 32-bit address of the PC contents. The operation result is the contents of an address register specified by the register field in the instruction code (RnL or ERn) is zero-extended and multiplied by 2. The PC contents to which the displacement is added is the address of the first byte of the next instruction. In advanced mode, only the lower 24 bits of 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 and the vector address of an exception handling other than a reset or a CPU address error can be specified by VBR.

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

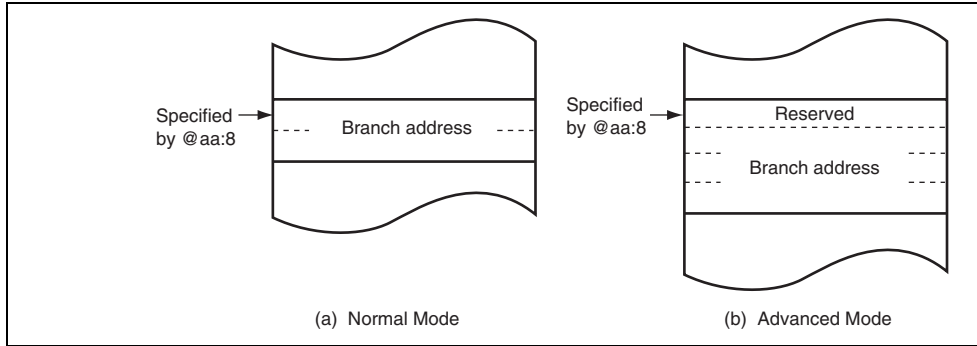


Figure 2.15 Branch Address Specification in Memory Indirect Mode

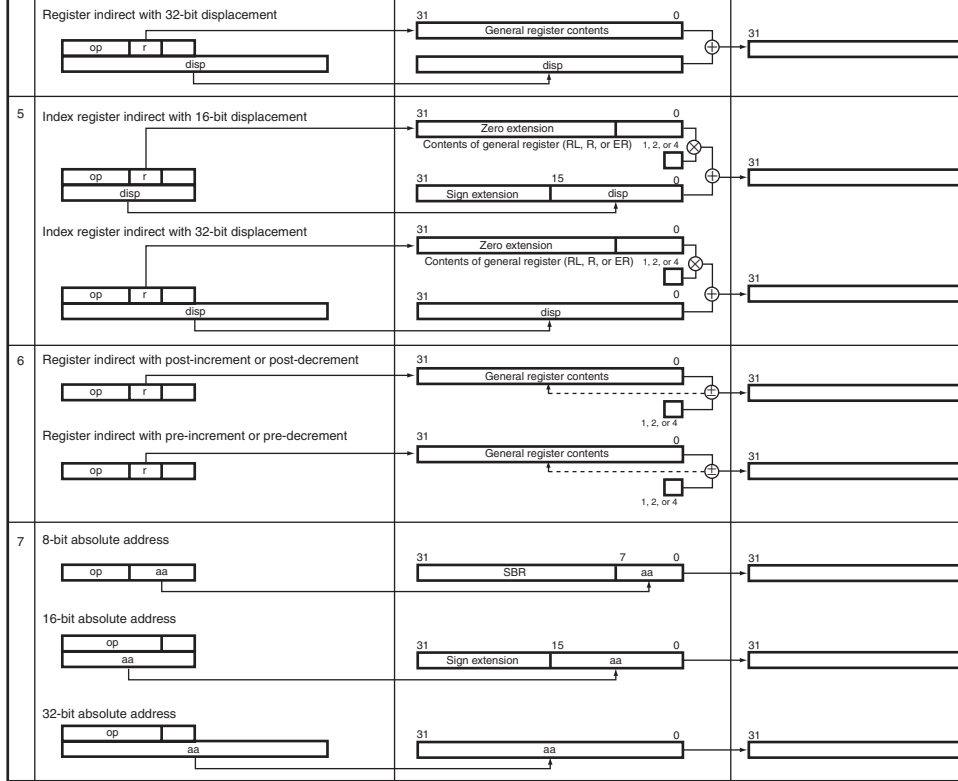
advanced mode, the first byte of the longword-size data is assumed to be all 0 (H'00).

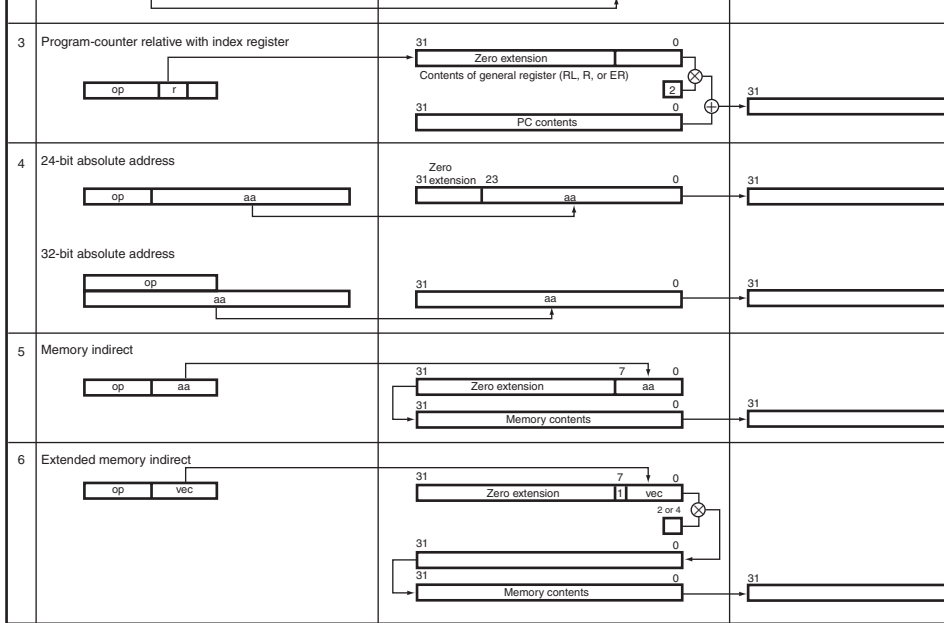
2.8.12 Effective Address Calculation

Tables 2.14 and 2.15 show how effective addresses are calculated in each addressing mode. The lower bits of the effective address are valid and the upper bits are ignored (zero extended or zero 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-extended for the transfer and operation instructions.
- The lower 24 bits of the effective address are valid and the upper eight bits are zero-extended for the branch instructions.





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 addressing mode shown in item 5 of table 2.14. The obtained data is used instead of the general register. The result is stored in a general register. For details, see H8SX Family Software Manual.

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 normal processing flow due to activation of an exception source, such as, a reset, trace, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception handling table and branches to that address. For further details, see section 6, Exception Handling.

- 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 request from a bus master other than the CPU. While the bus is released, the CPU halts operations.

- Program stop state

This is a power-down state in which the CPU stops operating. The program stop state occurs when a SLEEP instruction is executed or the CPU enters hardware standby mode. For further details, see section 27, Power-Down Modes.

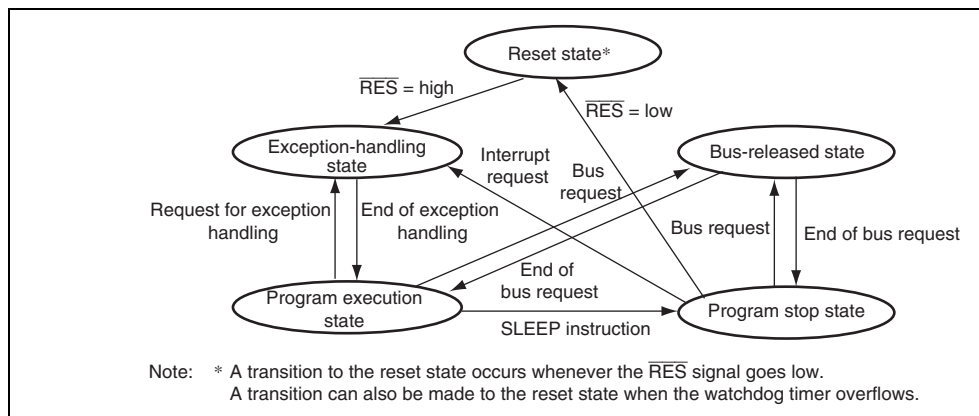


Figure 2.16 State Transitions

MCU Operating Mode	MD2	MD1	MD0	CPU Operating Mode	Address Space	LSI Initiation Mode	On-Chip ROM	Ext B Defa
1	0	0	1	Advanced mode	16 Mbytes	User boot mode	Enabled	—
2	0	1	0			Boot mode	Enabled	—
3	0	1	1			Boundary scan enabled single-chip mode	Enabled	—
4	1	0	0			On-chip ROM disabled extended mode	Disabled	16 bits
5	1	0	1			On-chip ROM disabled extended mode	Disabled	8 bits
6	1	1	0			On-chip ROM enabled extended mode	Enabled	8 bits
7	1	1	1			Single-chip mode	Enabled	—

Table 3.2 SDRAM Interface Setting for each 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 space are 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 can be selected.

Modes 1 and 2 are the user boot mode and the boot mode, respectively, in which the flash can be programmed and erased. For details on the user boot mode and boot mode, see section 4.2.2 Flash Memory.

Modes 4 to 6 are external extended modes, in which the external memory and devices can be accessed. In the external extended modes, the external address space can be designated as a 16-bit address space for each area by the bus controller after starting program execution.

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

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

MDS3 to MD0 are latched. Latching is released by a reset.

Bit	15	14	13	12	11	10	9
Bit Name	MDS3* ³	—	—	—	MDS3	MDS2	MDS1
Initial Value	0/Undefined* ^{2*3}	1	0	1	Undefined* ¹	Undefined* ¹	Undefined* ¹
R/W	R	R	R	R	R	R	R

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

- Notes: 1. Determined by the settings of pins MD2 to MD0
 2. Determined by the setting of MD3 pin.
 3. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Bit	Bit Name	Initial Value	R/W	Descriptions
15	—	0	R	<ul style="list-style-type: none"> H8SX/1648, H8SX/1648A, H8SX/1648B, H8SX/1648C, H8SX/1648D, H8SX/1648E, H8SX/1648F, H8SX/1648G, H8SX/1648H Group: Reserved These are read-only bits and cannot be modified.
	MDS7* ³	Undefined* ^{2*3}	R	<ul style="list-style-type: none"> H8SX/1648G, H8SX/1648H Group: This pin indicates a value set with the mode register (MD3). When MDCR is read, the signal levels in the mode register (MD3) pin is latched into this bit. This latch is released by a reset.

7	—	0/Undefined* ² * ³	R	Reserved
6	—	1	R	These are read-only bits and cannot be modified.
5	—	0	R	
4	—	1	R	
3	—	Undefined* ¹	R	
2	—	Undefined* ¹	R	
1	—	Undefined* ¹	R	
0	—	Undefined* ¹	R	

- Notes:
1. Determined by the settings of pins MD2 to MD0
 2. Determined by the setting of MD3 pin.
 3. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

Table 3.3 Settings of Bits MDS3 to MDS0

MCU Operating Mode	Mode Pins			MDCR			
	MD2	MD1	MD0	MDS3	MDS2	MDS1	MDS0
1	0	0	1	1	1	0	1
2	0	1	0	1	1	0	0
3	0	1	1	0	1	0	0
4	1	0	0	0	0	1	0
5	1	0	1	0	0	0	1
6	1	1	0	0	1	0	1
7	1	1	1	0	1	0	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 value should always be 1.
13	MACS	0	R/W	MAC Saturation Operation Control Selects either saturation operation or non-saturation 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 should always be 1.
11	FETCHMD	0	R/W	Instruction Fetch Mode Select This LSI can prefetch an instruction in units of 16 bits or 32 bits. Select the bus width for instruction prefetching depending on the used memory for the stored programs. 0: 32-bit mode 1: 16-bit mode

not write to this bit when P_ORGE = 1.
 When writing 0 to this bit after reading EXPE, the external bus cycle should not be executed.
 The external bus cycle may be carried out in parallel with the internal bus cycle depending on the timing of the write data buffer function, refresh control function and EXDMAC bus right release status and others.

0: External bus disabled
 1: External bus enabled

8	RAME	1	R/W	RAM Enable Enables or disables the on-chip RAM. This bit is initialized when the reset state is released. Do not write 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 value should 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 should always be 1.

- Notes: 1. The initial value depends on the LSI initiation mode.
 2. For details on the settings of the EXPE and PCJKE bits when the external address space is in use, see section 13.3.12, Port Function Control Register D (PFCRD).

This is the boot mode for the flash memory. The LSI operates in the same way as in mode 7 except for programming and erasing of the flash memory. For details, see section 24, Flash Memory.

3.3.3 Mode 3

This is the boundary scan function enabled single-chip activation mode. The operation is the same as mode 7 except for the boundary scan function. For details on the boundary scan function, see section 25, Boundary Scan.

3.3.4 Mode 4

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

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

3.3.6 Mode 6

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

The initial bus width mode immediately after a reset is eight bits, with 8-bit access to all I/O ports. Ports D, E, and F function as input ports, but they can be used as an address bus by specifying the data direction register (DDR) for each port. For details, see section 13, I/O Ports. Port H functions as a data bus, and parts of ports A and B function as bus control signals. However, if any port is designated as a 16-bit access space by the bus controller, the bus width mode switches to 16-bit and ports H and I function as a data bus.

3.3.7 Mode 7

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

All I/O ports can be used as general input/output ports. The external address space cannot be accessed in the initial state, but setting the EXPE bit in the system control register (SYSCR) enables the external address space. After the external address space is enabled, ports D, E, and F can be used as an address output bus and ports H and I as a data bus by specifying the data direction register (DDR) for each port. When the external address space is not in use, port K can be used by setting the PCJKE bit in the port function control register D (PFCRD) to 1. For details, see section 13, I/O Ports.

3	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C*1	P*/A	P*/A	P*/A	P*/A	P*/D
4	P/C*	P/C*	P*/C	P*/C	P/C*	P*/C	P*/C*1	A	A	A	P*/A	D
5	P/C*	P/C*	P*/C	P*/C	P/C*	P*/C	P*/C*1	A	A	A	P*/A	D
6	P/C*	P/C*	P*/C	P*/C	P*/C	P*/C	P*/C*1	P*/A	P*/A	P*/A	P*/A	D
7	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C*1	P*/A	P*/A	P*/A	P*/A	P*/D

[Legend]

P: I/O port

A: Address bus output

D: Data bus input/output

C: Control signals, clock input/output

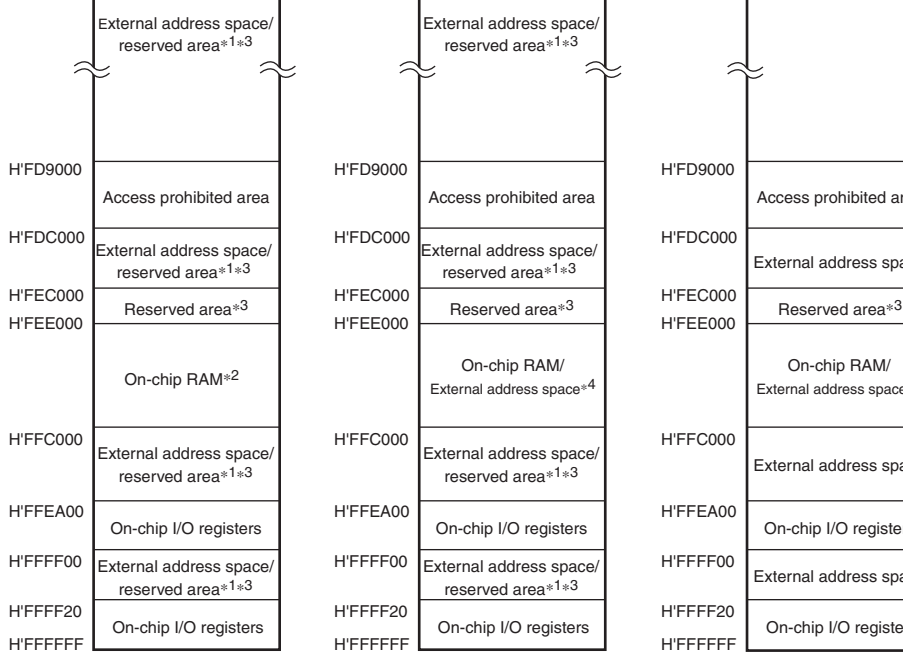
*: Immediately after a reset

Note: 1. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

3.4 Address Map

3.4.1 Address Map

Figures 3.1 to 3.3 show the address map in each operating mode.



- Notes:1. This area is specified as the external address space when EXPE = 1 and the reserved area when EXPE = 0.
 2. The on-chip RAM is used for flash memory programming. Do not clear the RAME bit in SYSCR 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/1648, 1648A, 1648L, and 1648H (1)

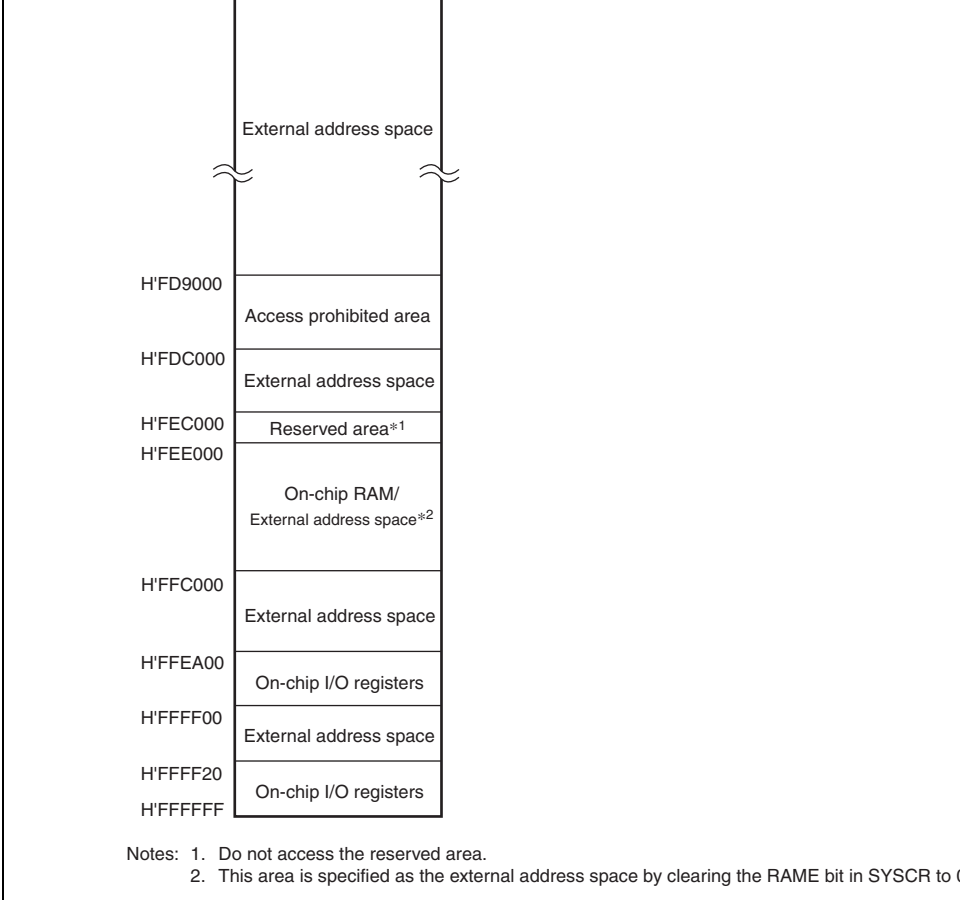


Figure 3.1 Address Map in Each Operating Mode of H8SX/1648, 1648A, 1648L and 1648H (2)

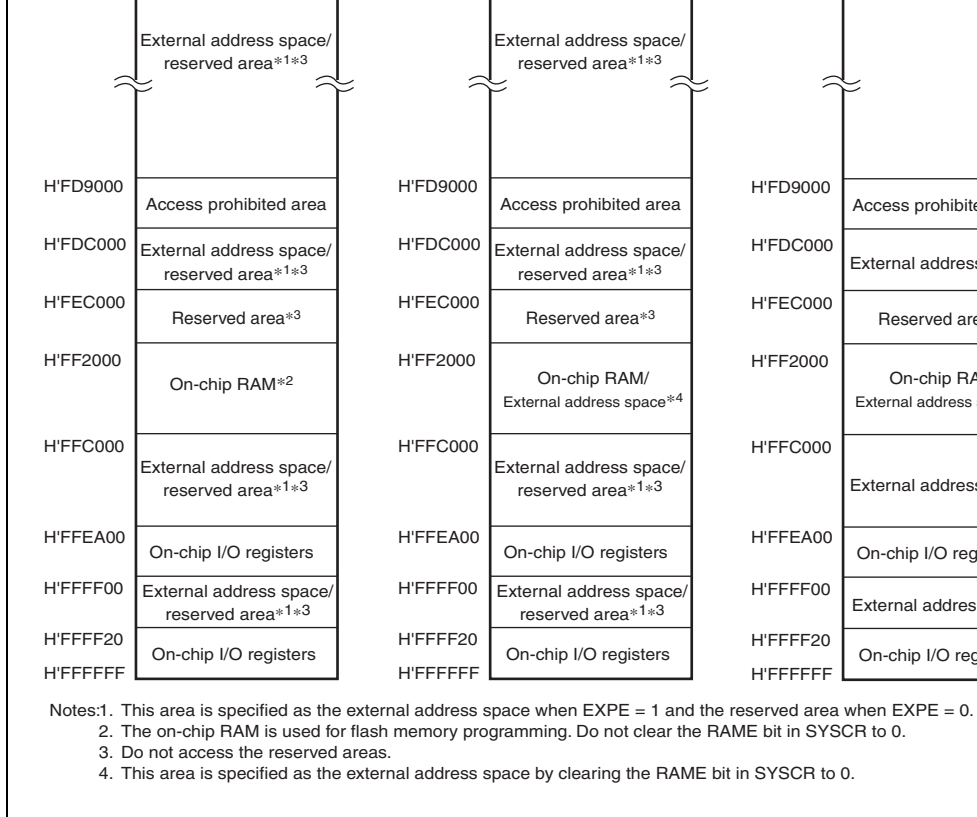
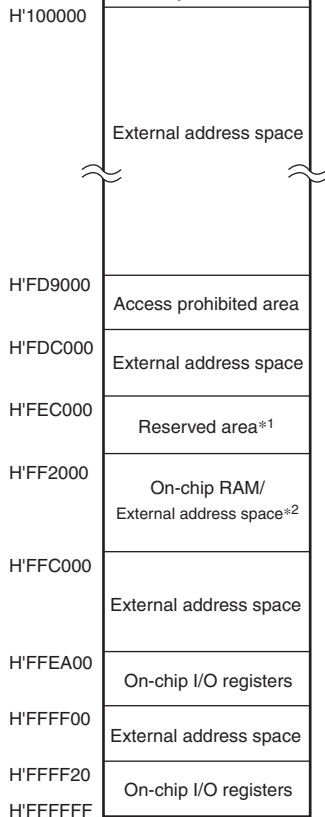


Figure 3.2 Address Map in Each Operating Mode of H8SX/1644, 1644A, 1644L, and 1644H (1)



Notes: 1. Do not access the reserved area.

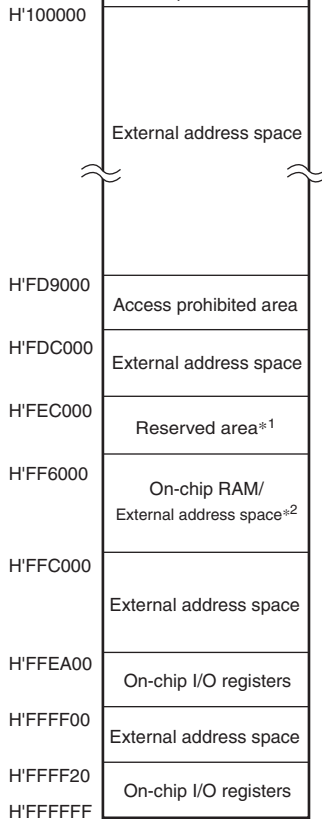
2. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.

Figure 3.2 Address Map in Each Operating Mode of H8SX/1644, 1644A, 1644L and 1644H (2)

	External address space/ reserved area*1*3	External address space/ reserved area*1*3	External address space/ reserved area*1*3
H'FD9000	Access prohibited area	H'FD9000	Access prohibited area
H'FDC000	External address space/ reserved area*1*3	H'FDC000	External address space/ reserved area*1*3
H'FEC000	Reserved area*3	H'FEC000	Reserved area*3
H'FF6000	On-chip RAM*2	H'FF6000	On-chip RAM/ External address space*4
H'FFC000	External address space/ reserved area*1*3	H'FFC000	External address space/ reserved area*1*3
H'FFEA00	On-chip I/O registers	H'FFEA00	On-chip I/O registers
H'FFFF00	External address space/ reserved area*1*3	H'FFFF00	External address space/ reserved area*1*3
H'FFFF20	On-chip I/O registers	H'FFFF20	On-chip I/O registers
H'FFFFFF		H'FFFFFF	

- Notes:1. This area is specified as the external address space when EXPE = 1 and the reserved area when EXPE = 0.
2. The on-chip RAM is used for flash memory programming. Do not clear the RAME bit in SYSCR 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 3.3 Address Map in Each Operating Mode of H8SX/1642, 1642A, 1642L, and 1642H (1)



Notes: 1. Do not access the reserved area.

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

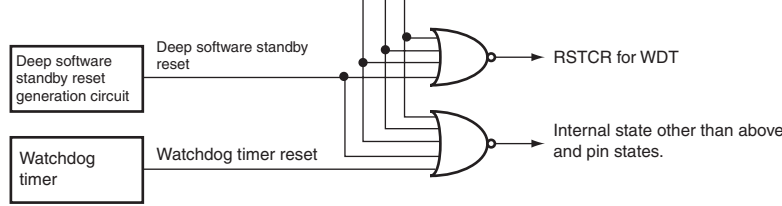
Figure 3.3 Address Map in Each Operating Mode of H8SX/1642, 1642A, 1642L and 1642H (2)

When using power-on reset* and voltage monitoring reset*, RES pin must be fixed high.

Table 4.1 Reset Names And Sources

Reset Name	Source
Pin reset	Voltage input to the $\overline{\text{RES}}$ pin is driven low.
Power-on reset*	Vcc rises or lowers
Voltage-monitoring reset*	Vcc falls (voltage-detection: Vdet)
Deep software standby reset	Deep software standby mode is canceled by interrupt.
Watchdog timer reset	The watchdog timer overflows.

Note: * Supported only by the H8SX/1648L Group and the H8SX/1648H Group.



Note: * Supported only by the H8SX/1648L Group and H8SX/1648H Group.

Figure 4.1 Block Diagram of Reset Circuit

When a reset is canceled, the reset exception handling is started. For the reset exception see section 6.3, Reset.

4.2 Input/Output Pin

Table 4.2 shows the pin related to reset.

Table 4.2 Pin Configuration

Pin Name	Symbol	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset input

Bit	7	6	5	4	3	2	1
Bit name	DPSRSTF	—	—	—	—	LVDF*2	—
Initial value:	0	0	0	0	0	0*3	0*3
R/W:	R/(W)*1	R/W	R/W	R/W	R/W	R/W*4	R/W

- Note:
1. Only 0 can be written to clear the flag.
 2. Supported only by the H8SX/1648L Group and H8SX/1648H Group.
 3. Initial value is undefined in the H8SX/1648L Group and H8SX/1648H Group.
 4. Only 0 can be written to clear the flag in the H8SX/1648L Group and H8SX/1648H Group.
 5. Only read is possible in the H8SX/1648L Group and H8SX/1648H Group.

Bit	Bit Name	Initial Value	R/W	Description
7	DPSRSTF	0	R/(W)*1	<p>Deep Software Standby Reset Flag</p> <p>Indicates that deep software standby mode is canceled by an interrupt source specified with DPSIER or DPSIEGR and an internal reset is generated.</p> <p>[Setting condition]</p> <p>When deep software standby mode is canceled by an interrupt source.</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> • When this bit is read as 1 and then written b • When a pin reset, power-on reset*2 and voltage monitoring reset*2 is generated.
6 to 3	—	All 0	R/W	<p>Reserved</p> <p>These bits are always read as 0. The write value always be 0.</p>

[Setting condition]

Vcc falling to or below Vdet.

[Clearing condition]

- After Vcc has exceeded Vdet and the specified stabilization period has elapsed, writing 0 to the flag after reading it as 1.
- Generation of a pin reset or power-on reset

1	—	Undefined	R/W	Reserved
---	---	-----------	-----	----------

The write value should always be 0.

0	PORF	Undefined	R	Power-on Reset Flag
---	------	-----------	---	---------------------

This bit indicates that a power-on reset has been generated.

[Setting condition]

Generation of a power-on reset

[Clearing condition]

Generation of a pin reset

- Note:
1. Only 0 can be written to clear the flag.
 2. Supported only by the H8SX/1648L Group and H8SX/1648H Group.

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

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)*	<p>Watchdog Timer Overflow Flag</p> <p>This bit is set when TCNT overflows in watchdog timer mode but not set in interval timer mode. Only 0 can be written to clear the flag.</p> <p>[Setting condition]</p> <p>When TCNT overflows (H'FF → H'00) in watchdog timer mode.</p> <p>[Clearing condition]</p> <p>When this bit is read as 1 and then written by 0.</p> <p>(The flag must be read after writing of 0, when this bit is set by the CPU using an interrupt.)</p>
6	RSTE	0	R/W	<p>Reset Enable</p> <p>Selects whether or not the LSI internal state is reset by overflow in watchdog timer mode.</p> <p>0: Internal state is not reset when TCNT overflows. (Although this LSI internal state is not reset, TCNT and TCSR counter and WDT are reset.)</p> <p>1: Internal state is reset when TCNT overflows.</p>
5	—	0	R/W	<p>Reserved</p> <p>Although this bit is readable/writable, operation is not affected by this bit.</p>
4 to 0	—	1	R	<p>Reserved</p> <p>These are read-only bits but cannot be modified.</p>

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

This is an internal reset generated by the power-on reset circuit.

If $\overline{\text{RES}}$ is in the high-level state when power is supplied, a power-on reset is generated. If $\overline{\text{RES}}$ has exceeded V_{por} and the specified period (power-on reset time) has elapsed, the chip is released from the power-on reset state. The power-on reset time is a period for stabilization of the power supply and the LSI circuit.

If $\overline{\text{RES}}$ is at the high-level when the power-supply voltage (V_{cc}) falls to or below V_{por} , a power-on reset is generated. The chip is released after V_{cc} has risen above V_{por} and the power-on reset time has elapsed.

After a power-on reset has been generated, the PORF bit in RSTSR is set to 1. The PORF bit is a read-only register and is only initialized by a pin reset. Figure 4.2 shows the operation of the power-on reset.

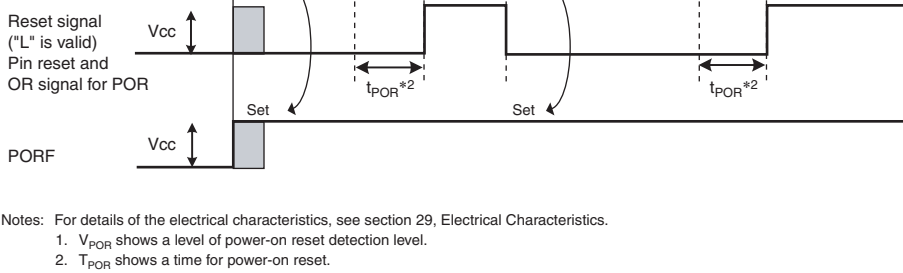


Figure 4.2 Operation of a Power on Reset

4.6 Power Supply Monitoring Reset (H8SX1648L Group, H8SX1648L Group)

This is an internal reset generated by the power-supply detection circuit.

When V_{cc} falls below V_{det} in the state where the LVDE bit in LVDCR has been set to 1 and the LVDR1 bit has been cleared to 0, a voltage-monitoring reset is generated. When V_{cc} subsequently rises above V_{det} , release from the voltage-monitoring reset proceeds after a specified time has elapsed.

For details of the voltage-monitoring reset, see section 5, Voltage Detection Circuit (LVDCR) in section 29, Electrical Characteristics.

This is an internal reset generated by the watchdog timer.

When the RSTE bit in RSTCSR is set to 1, a watchdog timer reset is generated by a TC overflow. After a certain time, the watchdog timer reset is canceled.

For details of the watchdog timer reset, see section 18, Watchdog Timer (WDT).

4.9 Determination of Reset Generation Source

Reading RSTCSR, RSTSR, and LVDCR* of the voltage detection circuit determines which source was used to execute the reset exception handling. Figure 4.2 shows an example of the flowchart to identify a reset generation source.

Note: * Supported only by the H8SX/1648L Group and H8SX/1648H Group.

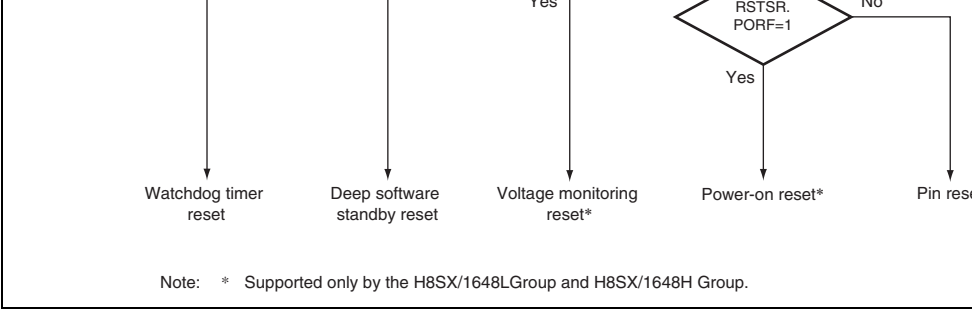


Figure 4.3 Example of Reset Generation Source Determination Flow

- voltage-detection circuit
- Capable of detecting the power-supply voltage (V_{cc}) becoming less than or equal to V_{det} .
- Capable of generating an internal reset or interrupt when a low voltage is detected.

A block diagram of the voltage detection circuit is shown in figure 5.1.

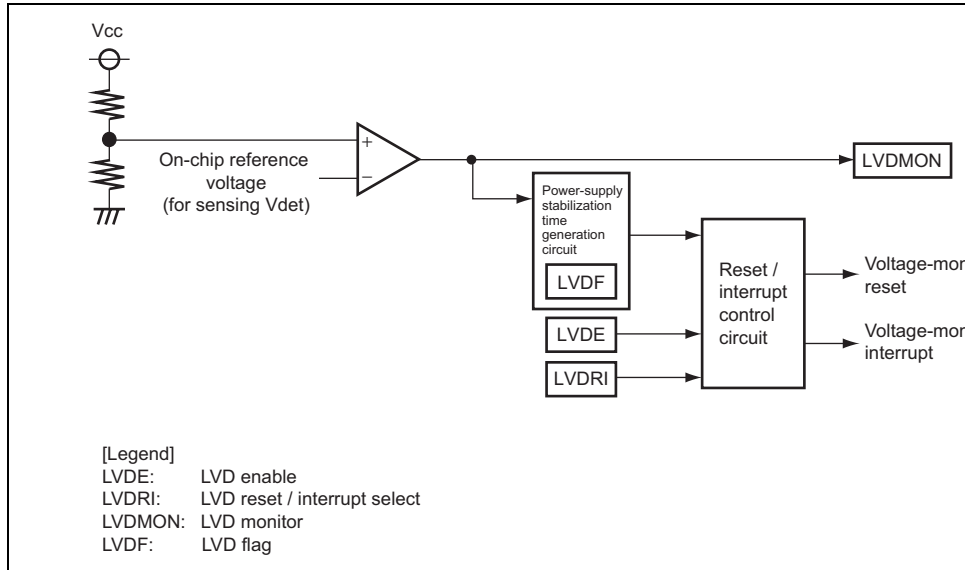


Figure 5.1 Block Diagram of Voltage-Detection Circuit

LVDE, LVDRI, and LVDMON are initialized by a pin reset or power-on reset

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

Bit	Bit Name	Initial Value	R/W	Description
7	LVDE	0	R/W	<p>LVD Enable</p> <p>This bit enables or disables issuing of a reset interrupt by the voltage-detection circuit.</p> <p>0: Disabled 1: Enabled</p>
6	LVDRI	0	R/W	<p>LVD Reset/Interrupt Select</p> <p>This bit selects whether an internal reset interrupt is generated when the voltage detection circuit detects a low voltage. When modifying the LVDRI bit, ensure that low-voltage detection is in the disabled state (the LVDE bit is cleared).</p> <p>0: A reset is generated when a voltage is detected. 1: An interrupt is generated when a low voltage is detected.</p>
5	—	0	R/W	<p>Reserved</p> <p>This bit is always read as 0 and the write data should always be 0.</p>

5.2.2 Reset Status Register (RSTSR)

RSTSR indicates the source of an internal reset or voltage monitoring interrupt.

Bit	7	6	5	4	3	2	1
Bit name	DPSRSTF	—	—	—	—	LVDF	—
Initial value:	0	0	0	0	0	Undefined	Undefined
R/W:	R/(W)*	R/W	R/W	R/W	R/W	R/(W)*	R/W

Note: * To clear the flag, only 0 should be written to.

Bit	Bit Name	Initial Value	R/W	Description
7	DPSRSTF	0	R/W*	<p>Deep Software Standby Reset Flag</p> <p>This bit indicates release from deep software standby mode due to the interrupt source selected by DPSIER and DPSIEGR, and the generation of an internal reset.</p> <p>[Setting condition]</p> <p>Release from deep software standby mode due to an interrupt source.</p> <p>[Clearing condition]</p> <ul style="list-style-type: none"> • Writing 0 to the bit after reading it as 1. • Generation of a pin reset, power-on reset, or voltage monitoring reset.

- [Clearing condition]
- After Vcc has exceeded Vdet and the specified stabilization period has elapsed, writing 0 to the bit after reading it as 1
- Generation of a pin reset or power-on

1	—	Undefined	R/W	Reserved The write value should always be 0.
0	PORF	Undefined	R	Power-on Reset Flag This bit indicates that a power-on reset has been generated. [Setting condition] Generation of a power-on reset [Clearing condition] Generation of a pin reset

Note: * To clear the flag, only 0 should be written to.

Reset, after Vcc has risen above Vdet, release from the voltage-monitoring reset takes place. The period for stabilization (t_{por}) has elapsed. The period for stabilization (t_{por}) is a time that is set by the voltage detection circuit in order to stabilize the Vcc and the internal circuit of the device.

When a voltage-monitoring reset is generated, the LVDF bit is set to 1.

For details, see section 29, Electrical Characteristics.

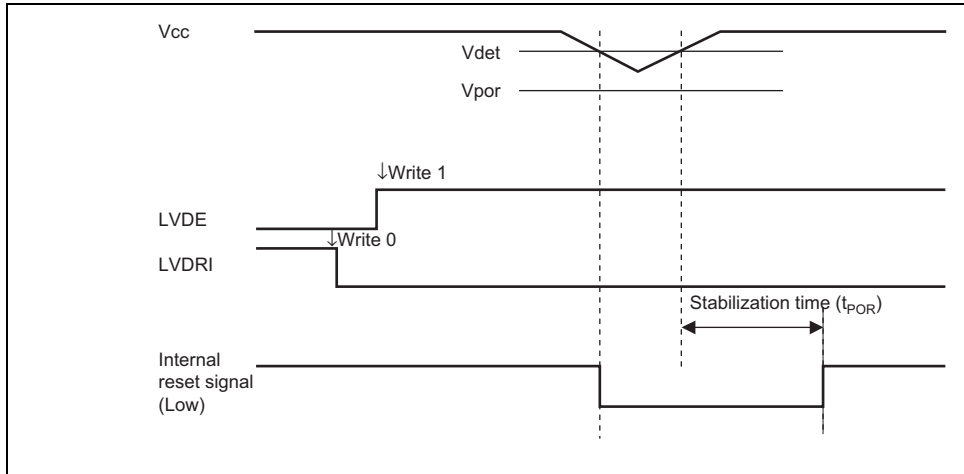


Figure 5.2 Timing of the Voltage-Monitoring Reset

the IRQ14SR and IRQ14SF bits in the ISCR to 01 (interrupt request on falling edge).

Figure 5.4 shows the procedure for setting the voltage-monitoring interrupt.

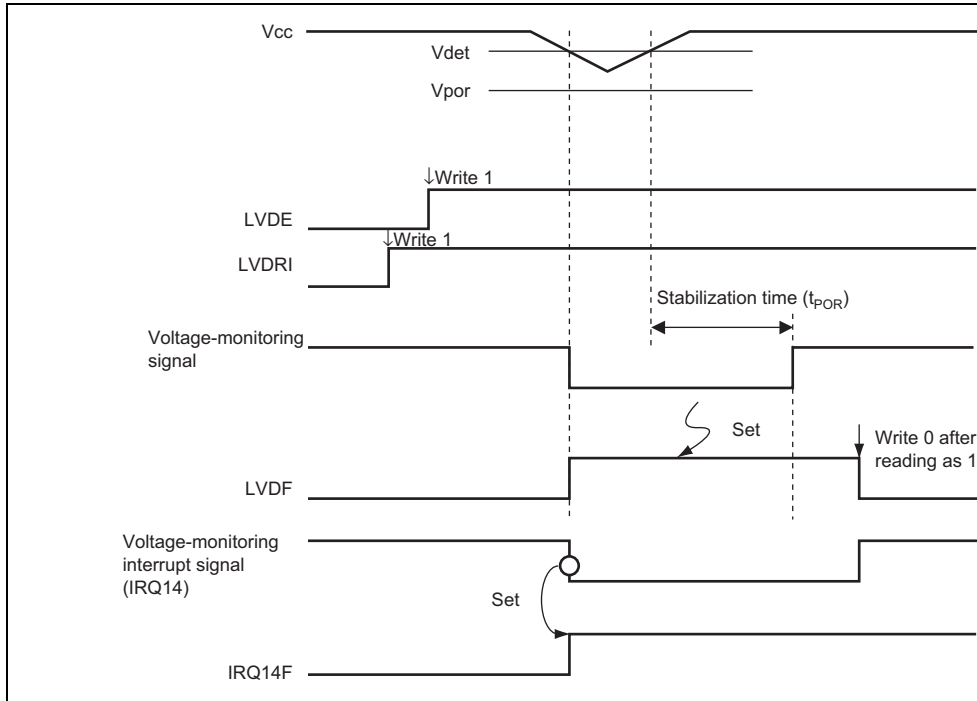


Figure 5.3 Timing of the Voltage-Monitoring Interrupt

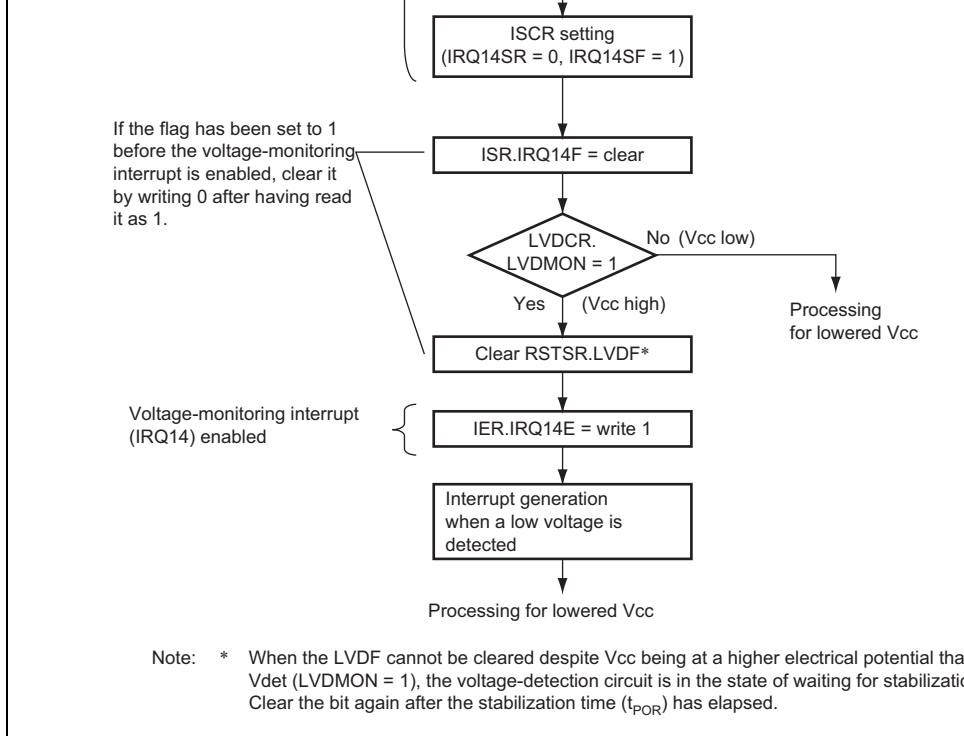



Figure 5.4 Example of the Procedure for Setting the Voltage-Monitoring Interrupt

The result of voltage detection by the voltage-detection circuit can be monitored by checking the value of the LVDMON bit in LVDCR. When the LVDMON bit has been enabled by setting the LVDE bit, 0 indicates that Vcc is at or below Vdet and 1 indicates that Vcc is above Vdet. The LVDMON bit should be read while the voltage-monitoring reset has been disabled by setting the LVDR to 1.

Before clearing the LVDF bit in RSTSR to 0, confirm that the LVDMON bit is set to 1 (indicating that Vcc is above Vdet). When it is impossible to clear the LVDF bit despite the LVDMON bit being 1, the voltage-detection circuit is in the state of waiting for stabilization. In such a case, read the bit again after the stabilization time (t_{por}) has elapsed.

Table 6.1 Exception Types and Priority

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

- Notes: 1. Traces are enabled only in interrupt control mode 2. Trace exception handling starts after execution of an RTE instruction.
2. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or other instructions, or on completion of reset exception handling.
3. Trap instruction exception handling requests and sleep instruction exception handling requests are accepted at all times in program execution state.

Exception Source	Vector Number	Vector Table Address Offset	
		Normal Mode* ²	Advanced, M* ² Maximum* ²
Reset	0	H'0000 to H'0001	H'0000 to H'0001
Reserved for system use	1	H'0002 to H'0003	H'0004 to H'0005
	2	H'0004 to H'0005	H'0008 to H'0009
	3	H'0006 to H'0007	H'000C to H'000D
Illegal instruction	4	H'0008 to H'0009	H'0010 to H'0011
Trace	5	H'000A to H'000B	H'0014 to H'0015
Reserved for system use	6	H'000C to H'000D	H'0018 to H'0019
Interrupt (NMI)	7	H'000E to H'000F	H'001C to H'001D
Trap instruction	(#0)	8	H'0010 to H'0011
	(#1)	9	H'0012 to H'0013
	(#2)	10	H'0014 to H'0015
	(#3)	11	H'0016 to H'0017
CPU address error	12	H'0018 to H'0019	H'0030 to H'0031
DMA address error* ³	13	H'001A to H'001B	H'0034 to H'0035
UBC break interrupt	14	H'001C to H'001D	H'0038 to H'0039
Reserved for system use	15	H'001E to H'001F	H'003C to H'003D
	17	H'0022 to H'0023	H'0044 to H'0045
Sleep interrupt	18	H'0024 to H'0025	H'0048 to H'0049

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
IRQ12	76	H'0098 to H'0099	H'0130 to H
IRQ13	77	H'009A to H'009B	H'0134 to H
IRQ14	78	H'009C to H'009D	H'0138 to H
IRQ15	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, DMAC, and EXDMAC*⁵.
 4. For details of internal interrupt vectors, see section 7.5, Interrupt Exception H Vector Table.
 5. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

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

The chip can also be reset by release of deep software standby mode or overflow of the watchdog timer. For details, see section 18, Watchdog Timer (WDT) and section 27, Power-Down Mode.

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

6.3.1 Reset Exception Handling

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

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

Figures 6.1 and 6.2 show examples of the reset sequence.

After the reset state is released, MSTPCRA and MSTPCRB are initialized to H'0FFF and H'0000, respectively, and all modules except for EXDMAC*, the DTC, and DMAC enter the module stop state.

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

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

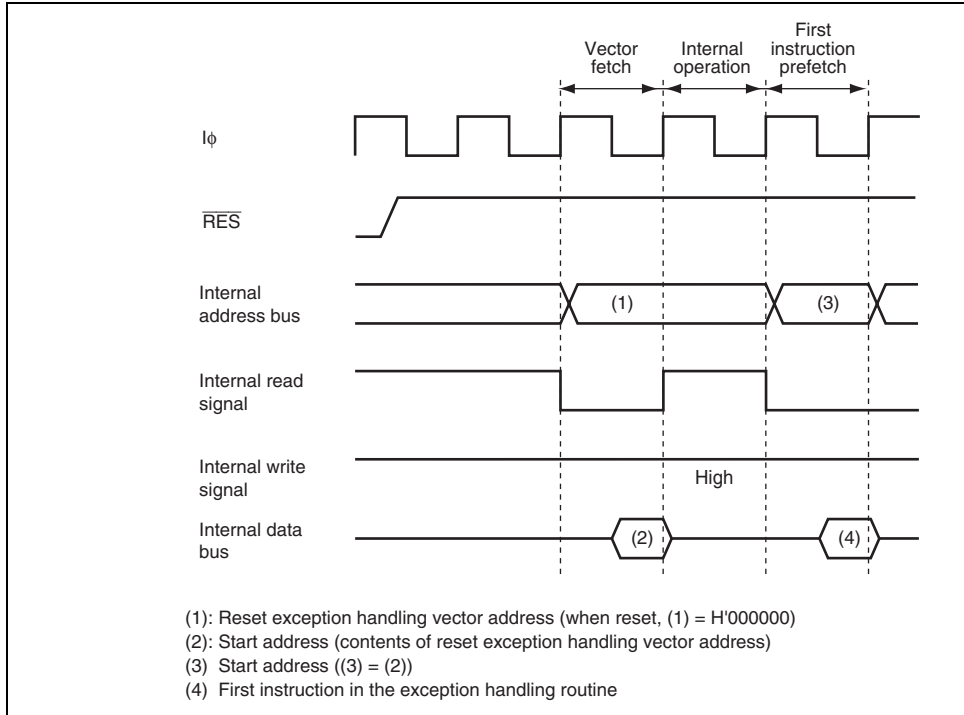
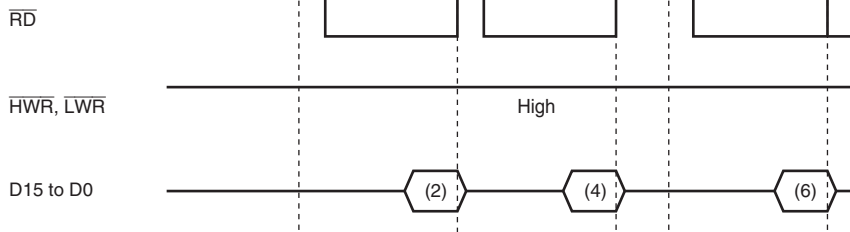


Figure 6.1 Reset Sequence (On-chip ROM Enabled Advanced Mode)



- (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 6.2 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 6.4 Status of CCR and EXR after Trace Exception Handling

Interrupt Control Mode	CCR			EXR
	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 (no
		Fetches instructions from odd addresses	Occurs
		Fetches instructions from areas other than on-chip peripheral module space* ¹	No (no
		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 (no
		Accesses stack when the stack pointer value is odd	Occurs
Data read/write	CPU	Accesses word data from even addresses	No (no
		Accesses word data from odd addresses	No (no
		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 (no
		Accesses word data from odd addresses	No (no
		Accesses external memory space in single-chip mode	Occurs
		Accesses to access prohibited area* ²	Occurs
Data read/write	EXDMAC* ³	Accesses word data from even addresses	No (no
		Accesses word data from odd addresses	No (no
		Accesses external memory space in single-chip mode	Occurs
		Accesses to access prohibited area* ²	Occurs
		Accesses to external memory space	No (no
		Accesses to space other than external memory space	Occurs

6.3.2 Address Error Exception Handling

When an address error occurs, address error exception handling starts after the bus cycle the address error ends and current instruction execution completes. The address error 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 address error is generated. The start address of the exception service routine is loaded from the vector table to PC, and program execution starts from that address.

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

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

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

handling.

Table 6.6 Status of CCR and EXR after Address Error Exception Handling

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

[Legend]

1: Set to 1

0: Cleared to 0

—: Retains the previous value.

UBC break interrupt	User break controller (UBC)	1
IRQ0 to IRQ15	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$ pin (external input)	16
Voltage-detection circuit* ²	Voltage-detection circuit (LVD) * ²	1
On-chip peripheral module	32K timer (TM32K)* ¹	1
	DMA controller (DMAC)	8
	EXDMA controller* ¹ (EXDMAC)* ¹	8
	Watchdog timer (WDT)	1
	A/D converter	3
	16-bit timer pulse unit (TPU)	52
	8-bit timer (TMR)	16
	Serial communications interface (SCI)	28
	I ² C bus interface 2 (IIC2)	4
	Refresh controller* ¹	1

- Notes: 1. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.
2. Supported only by the H8SX/1648L Group and the H8SX/1648H Group.

Different vector numbers and vector table offsets are assigned to different interrupt sources. For the vector number and vector table offset, refer to table 7.2, Interrupt Sources, Vector Address, and Vector Table Offsets, and Interrupt Priority in section 7, Interrupt Controller.

3. An exception handling vector table address corresponding to the interrupt source is generated. When the TRAPA instruction is executed, the start address of the exception service routine is loaded from the vector table to PC, and program execution starts from that address.

6.7 Instruction Exception Handling

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

6.7.1 Trap Instruction

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

1. The 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 specified in the TRAPA instruction is generated, the start address of the exception service routine is loaded 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 31 specified in the instruction code.

Table 6.8 shows the state of CCR and EXR after execution of trap instruction exception handling.

6.7.2 Sleep Instruction Exception Handling

The sleep instruction exception handling starts when a sleep instruction is executed with the SLEEP bit in SBYCR set to 0 and the SLPIE bit in SBYCR set to 1. The sleep instruction exception handling can always be executed in the program execution state. In the exception handling, the 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 specified by the SLEEP instruction is generated, the start address of the exception service routine is determined 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 been executed. In such cases the sleep instruction will be started when the transactions of a bus master other than the CPU has been completed and the CPU has gained the bus mastership.

Table 6.9 shows the state of CCR and EXR after execution of sleep instruction exception handling. For details, see section 27.10, Sleep Instruction Exception Handling.

6.7.3 Exception Handling by Illegal Instruction

The illegal instructions are general illegal instructions and slot illegal instructions. The exception handling by the general illegal instruction starts when an undefined code is executed. The 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 a delayed branch instruction) is executed. The exception handling by the general illegal instruction 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 is generated, the start address of the exception service routine is loaded from the vector table, PC, and program execution starts from that address.

Table 6.10 shows the state of CCR and EXR after execution of illegal instruction exception handling.

6.8 Stack Status after Exception Handling

Figure 6.3 shows the stack after completion of exception handling.

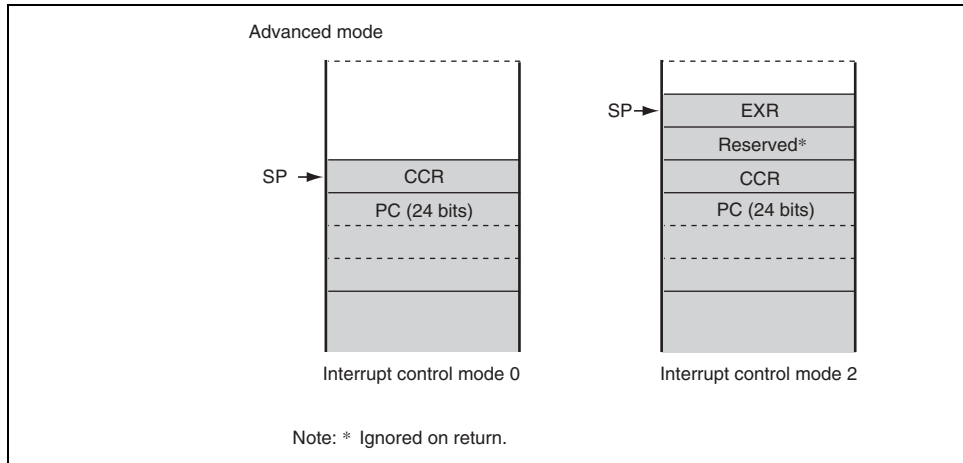


Figure 6.3 Stack Status after Exception Handling

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. Figure 6.4 shows an example of operation when the SP value is odd.

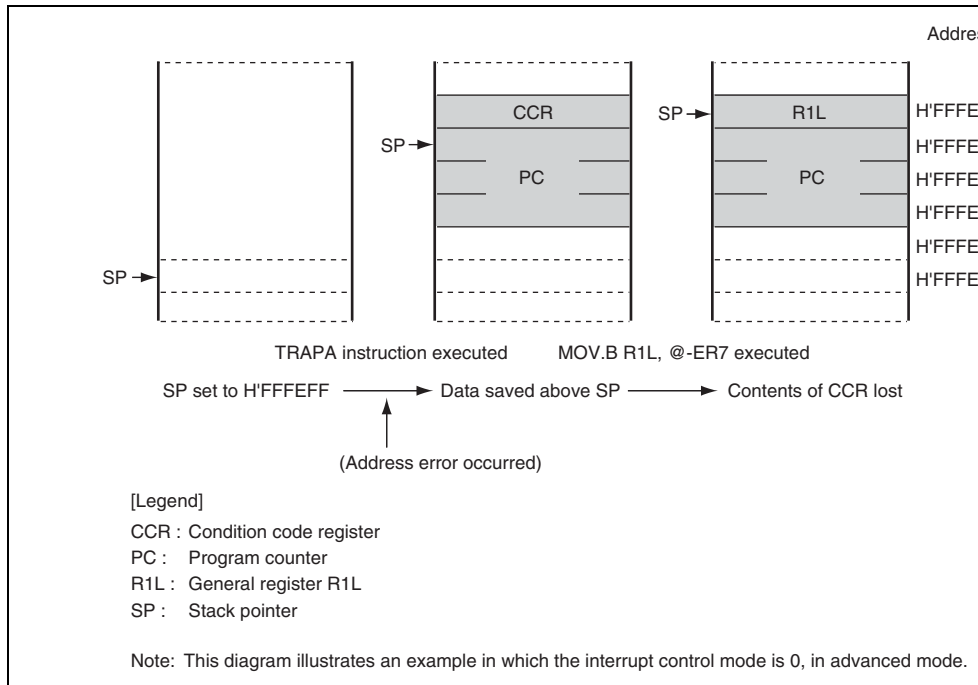


Figure 6.4 Operation when SP Value is Odd

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

- NMI
- Illegal instructions
- Trace
- Trap instructions
- CPU address error
- DMA address error*¹
- Sleep instruction
- UBC break interrupt
- Independent vector addresses
All interrupt sources are assigned independent vector addresses, making it unnecessary for the source to be identified in the interrupt handling routine.
- Seventeen external interrupts
NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge detection can be selected for $\overline{\text{NMI}}$. Falling edge, rising edge, or both edge detection, or level sensing, can be selected for $\overline{\text{IRQ15}}$ 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, DMAC, and EXDMAC*². The priority level of the CPU can be automatically assigned on an exception generation. Priority is given to the CPU interrupt exception handling over that of the DTC, DMAC, and EXDMAC transfer.

Notes: 1. DMA address error is occurred in the DTC, DMAC, and EXDMAC*².

2. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

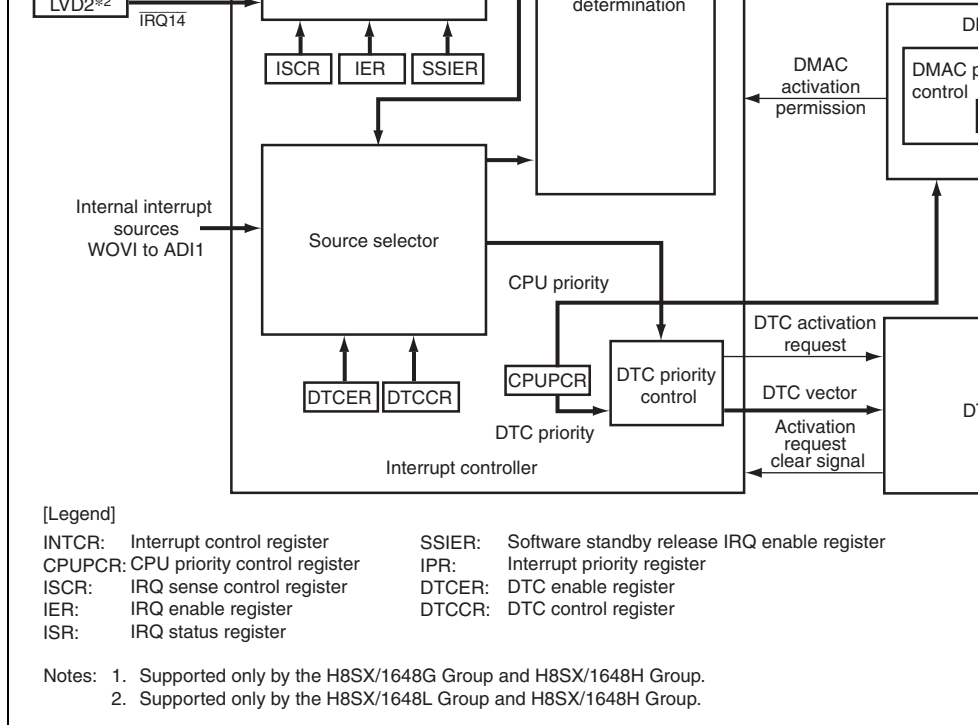


Figure 7.1 Block Diagram of Interrupt Controller

7.3 Register Descriptions

The interrupt controller has the following registers.

- Interrupt control register (INTCR)
- CPU priority control register (CPUPCR)
- Interrupt priority register
H8SX/1648 Group, H8SX/1648A Group, H8SX/1648L Group:
Interrupt priority registers A to I, K to O, Q, and R
(IPRA to IPRI, IPRK to IPRO, IPRQ, and IPRR)
H8SX/1648G Group, H8SX/1648H Group:
Interrupt priority registers A to O, Q, and R
(IPRA to IPRO, 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 modes for 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 EXIPR. 11: Setting prohibited.
3	NMIEG	0	R/W	NMI Edge Select Selects the input edge for the NMI pin. 0: Interrupt request generated at falling edge of NMI pin. 1: Interrupt request generated at rising edge of NMI pin.
2 to 0	—	All 0	R	Reserved These are read-only bits and cannot be modified

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	R/(W)*

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

000: 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	<p>Interrupt Priority Set Enable</p> <p>Controls the function which automatically assigns the interrupt priority level of the CPU. Setting this bit automatically sets bits CPUP2 to CPUP0 by the interrupt mask bit (I bit in CCR or bits I2 to I0 in ICR).</p> <p>0: Bits CPUP2 to CPUP0 are not updated automatically.</p> <p>1: The interrupt mask bit value is reflected in bits CPUP2 to CPUP0.</p>
---	--------	---	-----	---



011: Priority level 3

100: Priority level 4

101: Priority level 5

110: Priority level 6

111: Priority level 7 (highest)

-
- Notes:
1. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.
 2. When the IPSETE bit is set to 1, the CPU priority is automatically updated, so cannot be modified.

Setting a value in the range from B'000 to B'111 in the 3-bit groups of bits 14 to 12, 10 to 8 and 2 to 0 assigns a priority level to the corresponding interrupt. For the correspondence between the interrupt sources and the IPR settings, see table 7.2.

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

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This is a read-only bit and cannot be modified.
14	IPR14	1	R/W	Sets the priority level of the corresponding interrupt source.
13	IPR13	1	R/W	
12	IPR12	1	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)

				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	IPR6	1	R/W	Sets the priority level of the corresponding interrupt source.
5	IPR5	1	R/W	
4	IPR4	1	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	—	0	R	Reserved This is a read-only bit and cannot be modified.
2	IPR2	1	R/W	Sets the priority level of the corresponding interrupt source.
1	IPR1	1	R/W	
0	IPR0	1	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)

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ15E	0	R/W	IRQ15 Enable The IRQ15 interrupt request is enabled when th The 32KOVI interrupt in the TM32K* ¹ is also ena
14	IRQ14E	0	R/W	IRQ14 Enable The IRQ14 interrupt request is enabled when th The voltage-monitoring interrupt in the LVD* ² is
13	IRQ13E	0	R/W	IRQ13 Enable The IRQ13 interrupt request is enabled when th
12	IRQ12E	0	R/W	IRQ12 Enable The IRQ12 interrupt request is enabled when th
11	IRQ11E	0	R/W	IRQ11 Enable The IRQ11 interrupt request is enabled when th
10	IRQ10E	0	R/W	IRQ10 Enable The IRQ10 interrupt request is enabled when th
9	IRQ9E	0	R/W	IRQ9 Enable The IRQ9 interrupt request is enabled when this
8	IRQ8E	0	R/W	IRQ8 Enable The IRQ8 interrupt request is enabled when this
7	IRQ7E	0	R/W	IRQ7 Enable The IRQ7 interrupt request is enabled when this

2	IRQ2E	0	R/W	IRQ2 Enable The IRQ2 interrupt request is enabled when th
1	IRQ1E	0	R/W	IRQ1 Enable The IRQ1 interrupt request is enabled when th
0	IRQ0E	0	R/W	IRQ0 Enable The IRQ0 interrupt request is enabled when th

- Notes:
1. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.
 2. Supported only by the H8SX/1648L Group and the H8SX/1648H Group.

- ISCRH

Bit	15	14	13	12	11	10	9
Bit Name	IRQ15SR	IRQ15SF	IRQ14SR	IRQ14SF	IRQ13SR	IRQ13SF	IRQ12SR
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

- ISCR_L

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

edges of IRQ15

- TM32K*¹: When used as 32KOVI

IRQ15 is used as the 32KOVI interrupt in the

IRQ15 is generated at falling edge of $\overline{\text{IRQ15}}$.

00: Initial value

01: Interrupt request generated at falling edge

10: Setting prohibited

11: Setting prohibited

13	IRQ14SR	0	R/W
12	IRQ14SF	0	R/W

IRQ14 Sense Control Rise

IRQ14 Sense Control Fall

- When used as IRQ14

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 $\overline{\text{IRQ14}}$

- LVD*²: When used as a voltage-monitoring
interrupt

IRQ14 is used as the LVD voltage-monitoring

IRQ14 is generated at falling edge of $\overline{\text{IRQ14}}$.

00: Initial value

01: Interrupt request generated at falling edge

10: Setting prohibited

11: Setting prohibited

				00: Interrupt request generated by low level of
				01: Interrupt request generated at falling edge
				10: Interrupt request generated at rising edge
				11: Interrupt request generated at both falling and rising edges of IRQ12
7	IRQ11SR	0	R/W	IRQ11 Sense Control Rise
6	IRQ11SF	0	R/W	IRQ11 Sense Control Fall
				00: Interrupt request generated by low level of
				01: Interrupt request generated at falling edge
				10: Interrupt request generated at rising edge
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ11}}$
5	IRQ10SR	0	R/W	IRQ10 Sense Control Rise
4	IRQ10SF	0	R/W	IRQ10 Sense Control Fall
				00: Interrupt request generated by low level of
				01: Interrupt request generated at falling edge
				10: Interrupt request generated at rising edge
				11: Interrupt request generated at both falling and rising edges of IRQ10

00: Interrupt request generated by low level of
 01: Interrupt request generated at falling edge
 10: Interrupt request generated at rising edge
 11: Interrupt request generated at both falling
 edges of $\overline{\text{IRQ8}}$

Notes: 1. Supported only by the H8SX/1648G Group and H8SX/1648H Group.
 2. Supported only by the H8SX/1648L Group and H8SX/1648H Group.

- ISCR_L

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 01: Interrupt request generated at falling edge 10: Interrupt request generated at rising edge 11: Interrupt request generated at both falling edges of $\overline{\text{IRQ7}}$
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 01: Interrupt request generated at falling edge 10: Interrupt request generated at rising edge 11: Interrupt request generated at both falling edges of $\overline{\text{IRQ6}}$

				00: Interrupt request generated by low level of \overline{IF}
				01: Interrupt request generated at falling edge of \overline{IF}
				10: Interrupt request generated at rising edge of \overline{IF}
				11: Interrupt request generated at both falling and rising edges of \overline{IF}
7	IRQ3SR	0	R/W	IRQ3 Sense Control Rise
6	IRQ3SF	0	R/W	IRQ3 Sense Control Fall
				00: Interrupt request generated by low level of \overline{IF}
				01: Interrupt request generated at falling edge of \overline{IF}
				10: Interrupt request generated at rising edge of \overline{IF}
				11: Interrupt request generated at both falling and rising edges of \overline{IF}
5	IRQ2SR	0	R/W	IRQ2 Sense Control Rise
4	IRQ2SF	0	R/W	IRQ2 Sense Control Fall
				00: Interrupt request generated by low level of \overline{IF}
				01: Interrupt request generated at falling edge of \overline{IF}
				10: Interrupt request generated at rising edge of \overline{IF}
				11: Interrupt request generated at both falling and rising edges of \overline{IF}

0	IRQ0SF	0	R/W	IRQ0 Sense Control Flag
				00: Interrupt request generated by low level of I/O
				01: Interrupt request generated at falling edge of I/O
				10: Interrupt request generated at rising edge of I/O
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ0}}$

7.3.6 IRQ Status Register (ISR)

ISR is an IRQ15 to IRQ0 interrupt request register.

Bit	15	14	13	12	11	10	9	
Bit Name	IRQ15F	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	
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	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	
Initial Value	0	0	0	0	0	0	0	
R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	

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

- When IRQ15 interrupt exception handling is executed while falling-, rising-, or both-edge sensing is selected
- When the DTC is activated by an IRQ15 interrupt exception and the DISEL bit in MRB of the DTC is cleared

TM32K*²: When used as 32KOV1

[Setting condition]

- When the interrupt selected by ISCR occurs

[Clearing condition]

- Writing 0 after reading IRQ15F = 1
 - When IRQ15 interrupt exception handling is executed while falling-edge sensing is selected
-

- When IRQ14 interrupt exception handling is executed while falling-, rising-, or both-edge sensing is selected
- When the DTC is activated by an IRQ14 interrupt exception handling and the DIESEL bit in MRB of the DTC is cleared

LVD*³: When used as a voltage-monitoring interrupt exception handling
[Setting condition]

- When the interrupt selected by ISCR occurs
- [Clearing condition]

- Writing 0 after reading IRQ14F = 1
 - When IRQ14 interrupt exception handling is executed while falling-edge sensing is selected
-

5	IRQ5F	0	R/(W)* ¹	selected
4	IRQ4F	0	R/(W)* ¹	<ul style="list-style-type: none"> When the DTC is activated by an IRQn interrupt and the DISEL bit in MRB of the DTC is cleared
3	IRQ3F	0	R/(W)* ¹	
2	IRQ2F	0	R/(W)* ¹	
1	IRQ1F	0	R/(W)* ¹	
0	IRQ0F	0	R/(W)* ¹	

- Note:
1. Only 0 can be written, to clear the flag.
 2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.
 3. Supported only by the H8SX/1648L Group and H8SX/1648H Group.

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

Bit	Bit Name	Initial Value	R/W	Description
15	SSI15	0	R/W	Software Standby Release IRQ Setting
14	SSI14	0	R/W	These bits select the IRQn interrupt used to leave software standby mode (n = 15 to 0). 0: An IRQn request is not sampled in software standby mode 1: When an IRQn request occurs in software standby mode, this LSI leaves software standby mode when the oscillation settling time has elapsed
13	SSI13	0	R/W	
12	SSI12	0	R/W	
11	SSI11	0	R/W	
10	SSI10	0	R/W	
9	SSI9	0	R/W	
8	SSI8	0	R/W	
7	SSI7	0	R/W	
6	SSI6	0	R/W	
5	SSI5	0	R/W	
4	SSI4	0	R/W	
3	SSI3	0	R/W	
2	SSI2	0	R/W	
1	SSI1	0	R/W	
0	SSI0	0	R/W	

The NMIEN bit in INTCON selects whether an interrupt is generated by the rising or falling edge of the NMI pin. The NMIEG bit in INTCR selects whether an interrupt is requested at the rising or falling edge of the NMI pin.

When an NMI interrupt is generated, the interrupt controller determines that an error has occurred 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
- Sets the ERRF bit of DMDR_0 in the EXDMAC* to 1
- Clears the DTE bits of DMDRs for all channels in the DMAC to 0 to forcibly terminate data transfer
- Clears the DTE bits of DMDRs for all channels in the EXDMAC* to 0 to forcibly terminate data transfer

Note: * Supported only by the H8SX/1648G and H8SX/1648H groups.

(2) **IRQn Interrupts**

An IRQn interrupt is requested by a signal input on pins $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$. $\overline{\text{IRQn}}$ (n = 15 to 0) has the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, 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 0 by software. The bit manipulation instructions and memory operation instructions should be used to clear the flag.

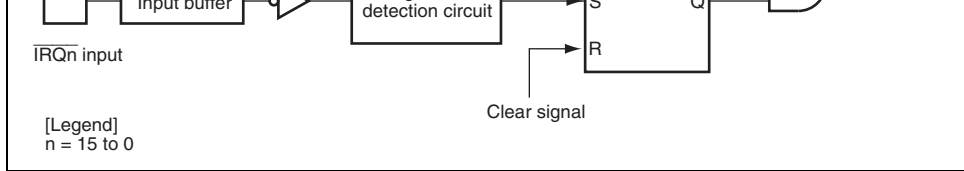


Figure 7.2 Block Diagram of Interrupts IRQn

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

7.4.2 Internal Interrupts

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

- For each on-chip peripheral module there are flags that indicate the interrupt request and enable bits that enable or disable these interrupts. They can be controlled independently. When the enable bit is set to 1, an interrupt request is issued to the interrupt controller.
- The interrupt priority 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 DMAC priority control functions.

Classification	Interrupt Source	Vector Number	Vector Address Offset*		Priority	DTC Activation
			Advanced Mode	Middle Mode Maximum Mode		
External pin	NMI	7	H'001C	—	High	—
UBC	UBC break interrupt	14	H'0038	—	↑	—
External pin	IRQ0	64	H'0100	IPRA6 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
	IRQ12	76	H'0130	IPRD14 to IPRD12		O
	IRQ13	77	H'0134	IPRD10 to IPRD8	O	
IRQ14	78	H'0138	IPRD6 to IPRD4	O		
LVD* ³	Voltage-monitoring interrupt				O	
External pin	IRQ15	79	H'013C	IPRD2 to IPRD0	O	
TM32K* ²	32KOV1				—	
—	Reserved for system use	80	H'0140	—	—	
WDT	WOV1	81	H'0144	IPRE10 to IPRE8	Low	—

A/D_0	AD10	86	H'0158	IPRF10 to IPRF8	O
—	Reserved for system use	87	H'015C	—	—
TPU_0	TGI0A	88	H'0160	IPRF6 to IPRF4	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
	TGI1B	94	H'0178		O
	TCI1V	95	H'017C		—
	TCI1U	96	H'0180		—
TPU_2	TGI2A	97	H'0184	IPRG14 to IPRG12	O
	TGI2B	98	H'0188		O
	TCI2V	99	H'018C		—
	TCI2U	100	H'0190		—
TPU_3	TGI3A	101	H'0194	IPRG10 to IPRG8	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
	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	—
EXDMAC* ²	EXDMTEND0	132	H'0210	IPRJ14 to IPRJ12	O	—
	EXDMTEND1	133	H'0214	IPRJ10 to IPRJ8	O	—
	EXDMTEND2	134	H'0218	IPRJ6 to IPRJ4	O	—
	EXDMTEND3	135	H'021C	IPRJ2 to IPRJ0	O	—
DMAC	DMEEND0	136	H'0220	IPRK14 to IPRK12	O	—
	DMEEND1	137	H'0224		O	—
	DMEEND2	138	H'0228		O	—
	DMEEND3	139	H'022C		O	—

Low

	RXI0	145	H'0244	
	TXI0	146	H'0248	
	TEI0	147	H'024C	
SCI_1	ERI1	148	H'0250	IPRK2 to IPRK0
	RXI1	149	H'0254	
	TXI1	150	H'0258	
	TEI1	151	H'025C	
SCI_2	ERI2	152	H'0260	IPRL14 to IPRL12
	RXI2	153	H'0264	
	TXI2	154	H'0268	
	TEI2	155	H'026C	
SCI_3	ERI3	156	H'0270	IPRL10 to IPRL8
	RXI3	157	H'0274	
	TXI3	158	H'0278	
	TEI3	159	H'027C	
SCI_4	ERI4	160	H'0280	IPRL6 to IPRL4
	RXI4	161	H'0284	
	TXI4	162	H'0288	
	TEI4	163	H'028C	

Low

	TCI7V	171	H'02AC	IPRM6 to IPRM4	—	—
	TCI7U	172	H'02B0		—	—
TPU_8	TGI8A	173	H'02B4	IPRM2 to IPRM0	0	0
	TGI8B	174	H'02B8		0	—
	TCI8V	175	H'02BC	IPRN14 to IPRN12	—	—
	TCI8U	176	H'02C0		—	—
TPU_9	TGI9A	177	H'02C4	IPRN10 to IPRN8	0	0
	TGI9B	178	H'02C8		0	—
	TGI9C	179	H'02CC		0	—
	TGI9D	180	H'02D0		0	—
	TCI9V	181	H'02D4	IPRN6 to IPRN4	—	—
TPU_10	TGI10A	182	H'02D8	IPRN2 to IPRN0	0	0
	TGI10B	183	H'02DC		0	—
	Reserved for system use	184	H'02E0	—	—	—
	Reserved for system use	185	H'02E4		—	—
	TCI10V	186	H'02E8	I PRO14 to I PRO12	0	—
	TCI10U	187	H'02EC		—	—
TPU_11	TGI11A	188	H'02F0	I PRO10 to I PRO8	0	0
	TGI11B	189	H'02F4		0	—
	TCI11V	190	H'02F8	I PRO6 to I PRO4	—	—
	TCI11U	191	H'02FC		—	—
—	Reserved for system use	192	H'0300	—	—	—
		215	H'035C		—	—
IIC2_0	IIC10	216	H'0360	I PRQ6 to I PRQ4	—	—
—	Reserved for system use	217	H'0364		—	—

Low

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		—
A/D_2	ADI2	232	H'03A0	IPRR6 to IPRR4	—
—	Reserved for system use	233	H'03A4		—
IIC2_2	IICI2	234	H'03A8		—
IIC2_3	IICI3	235	H'03AC		—
—	Reserved for system use	236	H'03B0	—	—
A/D_1	ADI1	237	H'03B4	IPRR2 to IPRR0	—
—	Reserved for system use	238	H'03B8	—	—
		255	H'03FC		—

Low

- Notes:
1. Lower 16 bits of the start address.
 2. Supported only by the H8SX/1648G and the H8SX/1648H groups.
 3. Supported only by the H8SX/1648L and the H8SX/1648H groups.

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

7.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests except for NMI are masked by the I bit in CCR when the CPU is in the interrupt exception handling routine. Figure 7.3 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, the 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 pending. When the I bit is cleared to 0, an interrupt request is accepted.
3. For multiple interrupt requests, the interrupt controller selects the interrupt request with the highest priority, sends the request to the CPU, and holds other interrupt requests pending.
4. When the CPU accepts the interrupt request, it starts interrupt exception handling after the execution of the current instruction has been completed.
5. The PC and CCR contents are saved to the stack area during the interrupt exception handling. The PC contents saved on the stack is the address of the first instruction to be executed after returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.

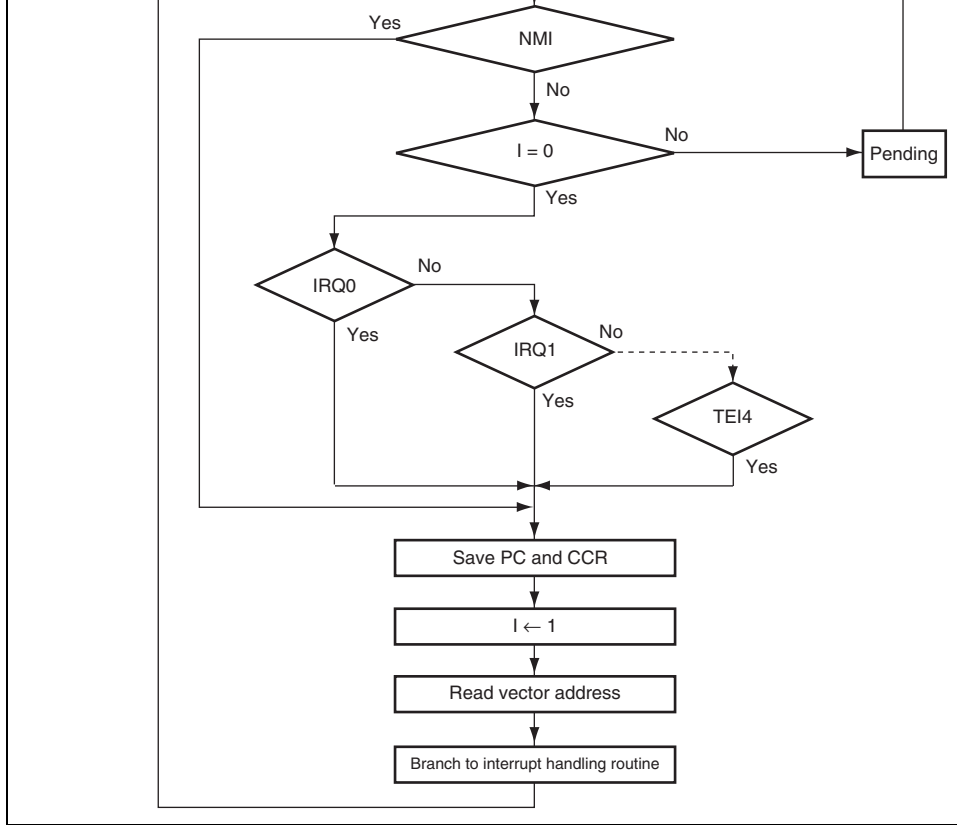


Figure 7.3 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

- the default setting shown in table 7.2.
3. Next, the priority of the selected interrupt request is compared with the interrupt mask level in EXR. When the interrupt request does not have priority over the mask level set, it is pending, and only an interrupt request with a priority over the interrupt mask level is accepted.
 4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
 5. The PC, CCR, and EXR contents are saved to the stack area during interrupt exception handling. The PC saved on the stack is the address of the first instruction to be executed after returning from the interrupt handling routine.
 6. The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority of the 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 the interrupt handling routine at the address indicated by the contents of the vector address table.

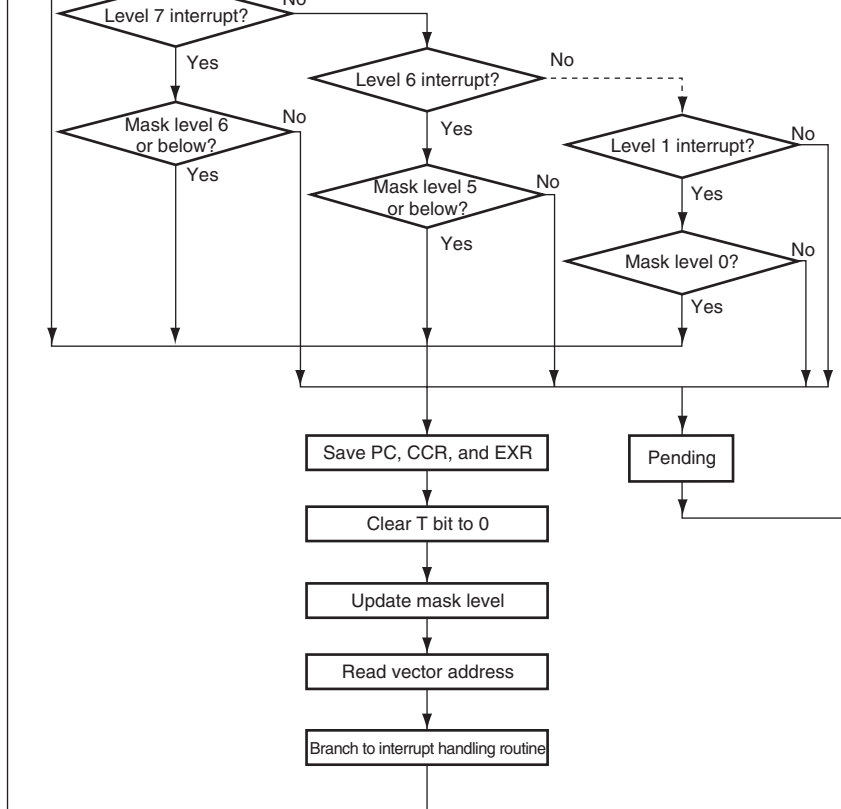
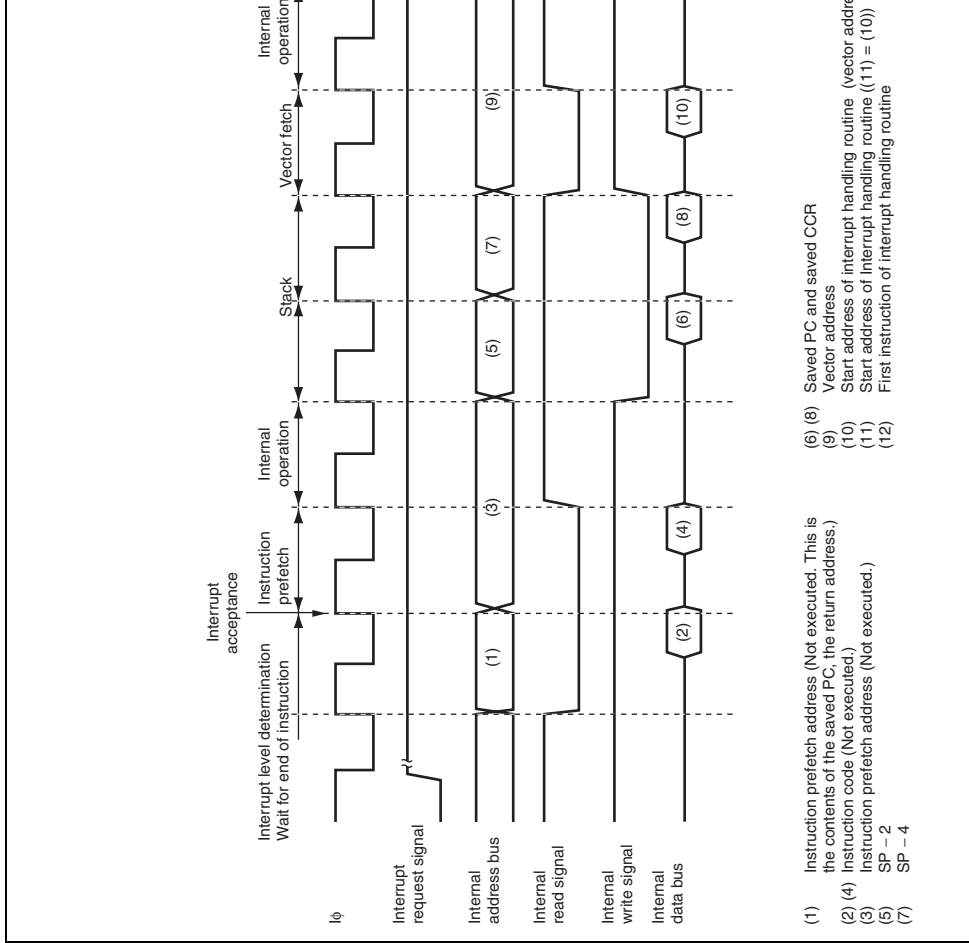


Figure 7.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2



- (6) (8) Saved PC and saved CCR
- (9) Vector address
- (10) Start address of interrupt handling routine (vector address)
- (11) Start address of interrupt handling routine ((11) = (10))
- (12) First instruction of interrupt handling routine

- (1) Instruction prefetch address (Not executed. This is the contents of the saved PC, the return address.)
- (2) (4) Instruction code (Not executed.)
- (3) Instruction prefetch address (Not executed.)
- (5) SP - 2
- (7) SP - 4

Figure 7.5 Interrupt Exception Handling



Execution State	Normal Mode		Advanced mode		Maximum
	Interrupt Control Mode 0	Interrupt Control Mode 2	Interrupt Control Mode 0	Interrupt Control Mode 2	Interrupt Control Mode 0
Interrupt priority determination* ¹				3	
Number of states until executing instruction ends* ²				1 to 19 + 2·S _I	
PC, CCR, EXR stacking	S _K to 2·S _K * ⁶	2·S _K	S _K to 2·S _K * ⁶	2·S _K	2·S _K
Vector fetch				S _h	
Instruction fetch* ³				2·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 handling
 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 setting to 2n, the interrupt response time is 2·S_K.

[Legend]

m: Number of wait cycles in an external device access.

7.6.5 DTC and DMAC Activation by Interrupt

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

- Interrupt request to 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 table 10-1, section 10, DMA Controller (DMAC), and section 12, Data Transfer Controller (DTC).

Figure 7.6 shows a block diagram of the DTC, DMAC, and interrupt controller.

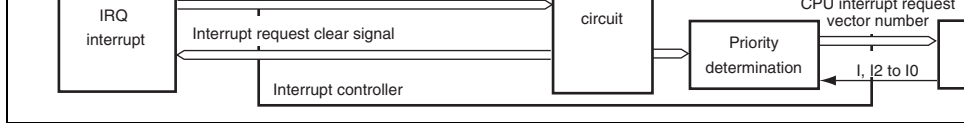


Figure 7.6 Block Diagram of DTC, DMAC, and Interrupt Controller

(1) Selection of Interrupt Sources

The activation source for each DMAC channel is selected by DMRSR. The selected activation source is input to the DMAC through the select circuit. When transfer by an on-chip module interrupt is enabled (DTF1 = 1, DTF0 = 0, and DTE = 1 in DMDR) and the DTA bit in DTCMCR is set to 1, the interrupt source selected for the DMAC activation source is controlled by the DMAC 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 or CPU 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 by 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 transfer counter indicates 0, an interrupt request is made to the CPU by clearing the DTCE bit to 0 after the DTC data transfer.

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

the peripheral independently.
 Table 7.6 lists the selection of interrupt sources and interrupt source clear control by setting the DTA bit in DMDR of the DMAC, the DTCE bit in DTCERA to DTCERH of the DTC, and the DISEL bit in MRB of the DTC.

Table 7.6 Interrupt Source Selection and Clear Control

DMAC Setting	DTC Setting		Interrupt Source Selection/Clear Control		
	DTA	DTCE	DISEL	DMAC	DTC
0	0	*	O	X	√
		1	0	O	√
			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 shown in table 7.6, when the DTC or DMAC reads/writes the prescribed register.

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

EXDMAC* is assigned by bits EDMAP2 to EDMAP0 in the EXDMA mode control register (EDMDR_0 to EDMDR_3) for each channel.

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

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

For the DMAC, the priority level can be specified for each channel. The DMAC activation source is controlled according to the priority level of each DMAC channel indicated by bits DMAP2 to DMAP0 and the priority level of the CPU. If the CPU has priority, the DMAC activation source is held. The DMAC is activated when the condition by which the activation source is held is cancelled (CPUPCE = 1 and value of bits CPUP2 to CPUP0 is greater than that of bits DMAP2 to DMAP0). If different priority levels are specified for channels, the channels of the higher priority levels continue transfer and the activation sources for the channels of lower priority level than that of the CPU are held.

mask bit of the CPU to the CPU priority level. Clearing the IPSETE bit to 0 disables the CPU to automatically assign the priority level. Therefore, the priority level is assigned directly by software rewriting bits CPUP2 to CPUP0. Even if the IPSETE bit is 1, the priority level of the CPU is software assignable by rewriting the interrupt mask bit of the CPU (I bit in CCR) and bits in EXR).

The priority level which is automatically assigned when the IPSETE bit is 1 differs according to the interrupt control mode.

In interrupt control mode 0, the I bit in CCR of the CPU is reflected in bit CPUP2. Bits CPUP1 and CPUP0 are fixed 0. In interrupt control mode 2, the values of bits I2 to I0 in EXR of the CPU are reflected in bits CPUP2 to CPUP0.

Table 7.7 shows the CPU priority control.

Table 7.7 CPU Priority Control

Interrupt Control Mode	Interrupt Priority	Interrupt Mask Bit	IPSETE in CPUPCR	Control Status	
				CPUP2 to CPUP0	Updating of I 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

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

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

be executed on completion of the instruction. However, if there is an interrupt request with priority over that interrupt, interrupt exception handling will be executed for the interrupt with priority and another interrupt will be ignored. The same also applies when an interrupt source flag is cleared to 0. Figure 7.7 shows an example in which the TCIEV bit in TIER of the TPU is cleared to 0. The above conflict will not occur if an enable bit or interrupt source flag is cleared to 0 when the interrupt is masked.

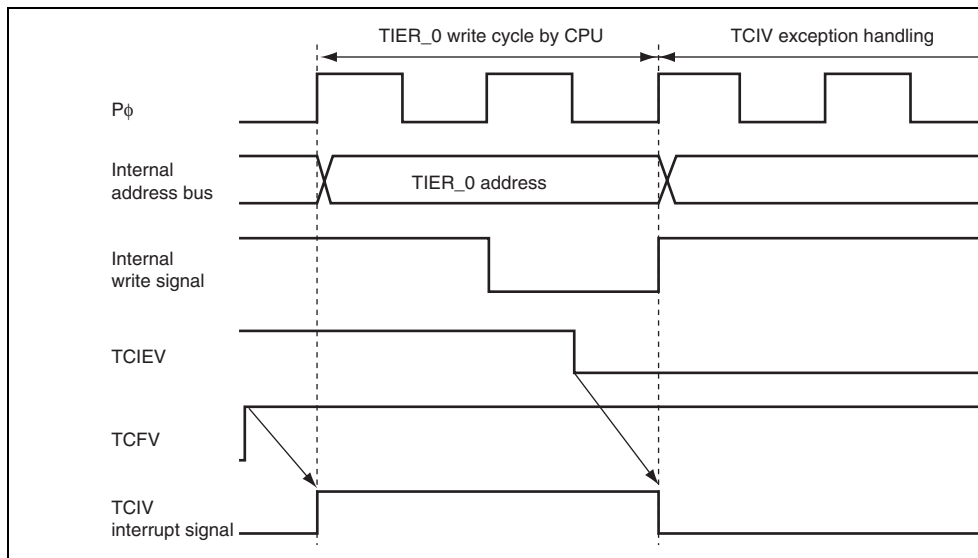


Figure 7.7 Conflict between Interrupt Generation and Disabling

Similarly, when an interrupt is requested immediately before the DTC enable bit is changed to activate the DTC, DTC activation and the interrupt exception handling by the CPU are both 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 has updated the mask level with an LDC, ANDC, ORC, or XORC instruction, and for a period after writing to the registers of the interrupt controller.

7.8.4 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 the transfer is not accepted until the transfer is completed.

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

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

7.8.5 Interrupts during Execution of MOVMD and MOVSD Instructions

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

8.1 Features

- Number of break channels: four (channels A, B, C, and D)
- Break comparison conditions (each channel)
 - Address
 - Bus master (CPU cycle)
 - Bus cycle (instruction execution (PC break))
- UBC break interrupt exception handling is executed immediately before execution of instruction fetched from the specified address (PC break).
- Module stop state can be set

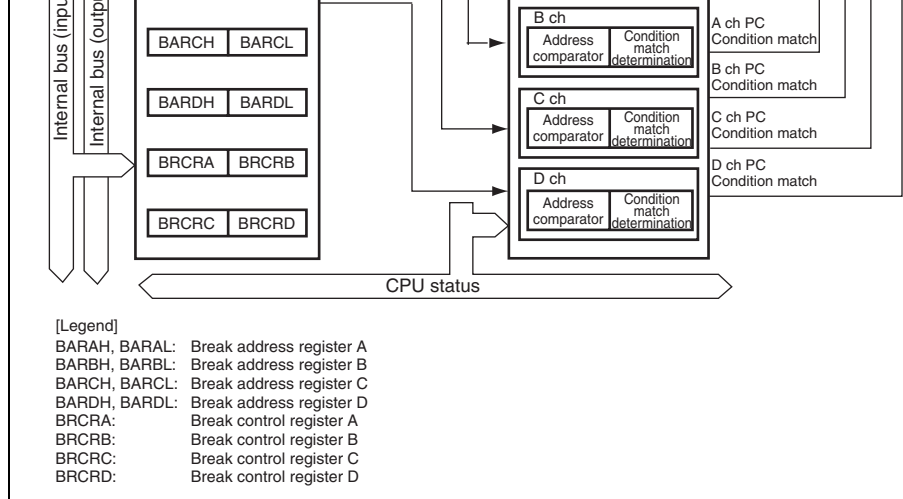


Figure 8.1 Block Diagram of UBC

	BAMRAL	R/W	H'0000	H'FFA06
Break address register B	BARBH	R/W	H'0000	H'FFA08
	BARBL	R/W	H'0000	H'FFA0A
Break address mask register B	BAMRBH	R/W	H'0000	H'FFA0C
	BAMRBL	R/W	H'0000	H'FFA0E
Break address register C	BARCH	R/W	H'0000	H'FFA10
	BARCL	R/W	H'0000	H'FFA12
Break address mask register C	BAMRCH	R/W	H'0000	H'FFA14
	BAMRCL	R/W	H'0000	H'FFA16
Break address register D	BARDH	R/W	H'0000	H'FFA18
	BARDL	R/W	H'0000	H'FFA1A
Break address mask register D	BAMRDH	R/W	H'0000	H'FFA1C
	BAMRDL	R/W	H'0000	H'FFA1E
Break control register A	BRCRA	R/W	H'0000	H'FFA28
Break control register B	BRCRB	R/W	H'0000	H'FFA2C
Break control register C	BRCRC	R/W	H'0000	H'FFA30
Break control register D	BRCRD	R/W	H'0000	H'FFA34

BARnL

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	BARn15	BARn14	BARn13	BARn12	BARn11	BARn10	BARn9	BARn8	BARn7	BARn6	BARn5	BARn4	BARn3	BARn2	BARn1
Initial Value:	0	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	R/W

- BARnH

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	BARn31 to BARn16	All 0	R/W	Break Address n31 to 16 These bits hold the upper bit values (bits 31 to 16) of the address break-condition on channel n.

[Legend]

n = Channels A to D

- BARnL

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	BARn15 to BARn0	All 0	R/W	Break Address n15 to 0 These bits hold the lower bit values (bits 15 to 0) of the address break-condition on channel n.

[Legend]

n = Channels A to D

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAMRn15	BAMRn14	BAMRn13	BAMRn12	BAMRn11	BAMRn10	BAMRn9	BAMRn8	BAMRn7	BAMRn6	BAMRn5	BAMRn4	BAMRn3	BAMRn2	BAMRn1	BAMRn0
Initial Value:	0	0	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	R/W	R/W

- BAMRnH

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	BAMRn31 to BAMRn16	All 0	R/W	Break Address Mask n31 to 16 Be sure to write H'FF00 here before setting a condition in the break control register.

[Legend]

n = Channels A to D

- BAMRnL

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	BAMRn15 to BAMRn0	All 0	R/W	Break Address Mask n15 to 0 Be sure to write H'0000 here before setting a condition in the break control register.

[Legend]

n = Channels A to D

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R/W	Reserved
14	—	0	R/W	These bits are always read as 0. The write value should always be 0.
13	CMFCPn	0	R/W	Condition Match CPU Flag UBC break source flag that indicates satisfactory specified CPU bus cycle condition. 0: The CPU cycle condition for channel n break requests has not been satisfied. 1: The CPU cycle condition for channel n break requests has been satisfied.
12	—	0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
11	CPn2	0	R/W	CPU Cycle Select
10	CPn1	0	R/W	These bits select CPU cycles as the bus cycle condition for the given channel.
9	CPn0	0	R/W	000: Break requests will not be generated. 001: The bus cycle break condition is CPU cycle 01x: Setting prohibited 1xx: Setting prohibited
8	—	0	R/W	Reserved
7	—	0	R/W	These bits are always read as 0. The write value should always be 0.
6	—	0	R/W	

condition for the given channel.

00: Break requests will not be generated.

01: The bus cycle break condition is read cycle.

1x: Setting prohibited

1	—	0	R/W	Reserved
0	—	0	R/W	These bits are always read as 0. The write value should always be 0.

[Legend]

n = Channels A to D

consist of CPU cycle, PC break, and reading. Condition comparison is not performed if CPU cycle setting is CPn = B'000, the PC break setting is IDn = B'00, or the read setting is RWn = B'00.

3. The condition match CPU flag (CMFCPn) is set in the event of a break condition match in the corresponding channel. These flags are set when the break condition matches but are cleared when it no longer does. To confirm setting of the same flag again, read the flag from the break interrupt handling routine, and then write 0 to it (the flag is cleared by writing 1 to it after reading it as 1).

[Legend]

n = Channels A to D

8.4.2 PC Break

1. When specifying a PC break, specify the address as the first address of the required instruction. If the address for a PC break condition is not the first address of an instruction, a break will never be generated.
2. The break occurs after fetching and execution of the target instruction have been completed. In cases of contention between a break before instruction execution and a user maskable interrupt, priority is given to the break before instruction execution.
3. A break will not be generated even if a break before instruction execution is set in a device.
4. The PC break condition is generated by specifying CPU cycles as the bus condition in the control register n (BRCRn.CPn0 = 1), PC break as the break condition (IDn0 = 1), and CPU cycles as the bus-cycle condition (RWn0 = 1).

[Legend]

n = Channels A to D

BRCRC	CMFCPC (bit 13)	for channel B Indicates that the condition matches in the CPU for channel C
BRCRD	CMFCPD (bit 13)	Indicates that the condition matches in the CPU for channel D

oscillation settling time has elapsed subsequent to the transition to software standby. When an interrupt is the canceling source, interrupt exception handling is executed after the SLEEP instruction, and the instruction following the SLEEP instruction is then executed.

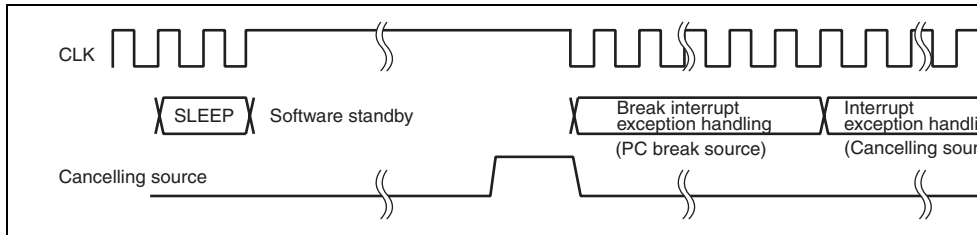


Figure 8.2 Contention between SLEEP Instruction (Software Standby) and PC Break

2. Prohibition on Setting of PC Break
 - Setting of a UBC break interrupt for program within the UBC break interrupt handling routine is prohibited.
3. The procedure for clearing a UBC flag bit (condition match flag) is shown below. A flag is cleared by writing 0 to it after reading it as 1. As the register that contains the flag bits is accessible in byte units, bit manipulation instructions can be used.

4. After setting break conditions for the UBC, an unexpected UBC break interrupt may occur after the execution of an illegal instruction. This depends on the value of the program counter and the internal bus cycle.

- Manages external address space in area units
 - Manages the external address space divided into eight areas
 - Chip select signals ($\overline{CS0}$ to $\overline{CS7}$) 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
 - DRAM*, synchronous DRAM*, burst ROM, byte control SRAM, or address/data multiplexed I/O interface can be set
 - 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{WAIT} pin.
 - Extension cycles can be inserted while \overline{CSn} is asserted for each area (n = 0 to 7)
 - The negation timing of the read strobe signal (\overline{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

High-speed data transfer is available using EXDMAC cluster transfer

- 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 access

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

Idle cycles can be inserted before the external access after a DMAC/EXDMAC* single transfer (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 executed 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, EXDMAC, DTC, refresh*, and external bus master

- EXDMAC external bus transfers and internal accesses can be executed in parallel*.

- Multi-clock function

The internal peripheral functions can be operated in synchronization with the peripheral module clock (P ϕ). Accesses to the external address space can be operated in synchronization with the external bus clock (B ϕ).

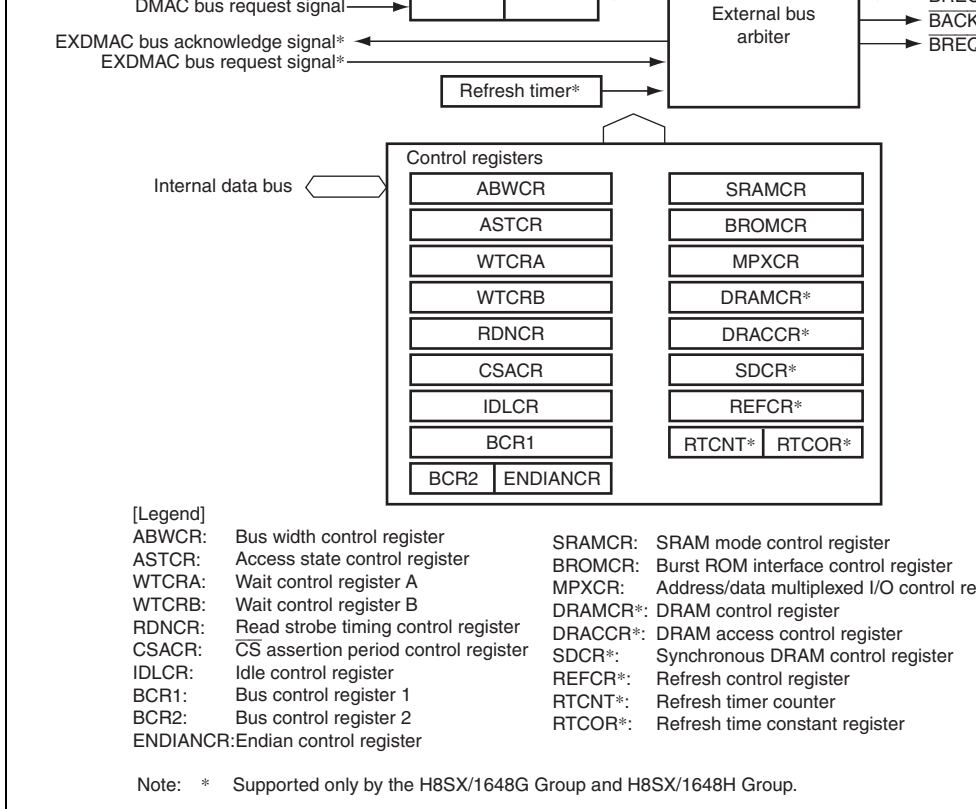


Figure 9.1 Block Diagram of Bus Controller

- 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)*

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

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 area is designated as 8-bit access space or 16-bit access space
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-bit access space
9	ABWH1	1	R/W	
8	ABWL0	1/0	R/W	1 1: Area n is designated as 8-bit access 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 H'FFFF.
2. An address space specified as byte control SRAM interface must not be specified as 8-bit access space.

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 area is designated as 2-state access space or 3-state access space. Wait cycle insertion is enabled or disabled at the same time. 0: Area n is designated as 2-state access space. Wait cycle insertion in area n access is disabled. 1: Area n is designated as 3-state access space. Wait cycle insertion in area n access is enabled. (n = 7 to 0)
13	AST5	1	R/W	
12	AST4	1	R/W	
11	AST3	1	R/W	
10	AST2	1	R/W	
9	AST1	1	R/W	
8	AST0	1	R/W	
7 to 0	—	All 0	R	

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	W62	1	R/W	Area 6 Wait Control 2 to 0
9	W61	1	R/W	These bits select the number of program wait cycles inserted when accessing area 6 while bit AST6 in ASTC6 is set.
8	W60	1	R/W	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	W42	1	R/W	Area 4 Wait Control 2 to 0
1	W41	1	R/W	These bits select the number of program wait cycles when accessing area 4 while bit AST4 in ASTCF
0	W40	1	R/W	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

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	W22	1	R/W	Area 2 Wait Control 2 to 0
9	W21	1	R/W	These bits select the number of program wait cycles when accessing area 2 while bit AST2 in ASTCR is specified. At this time, W22 is ignored. The CAS latency can be specified even if the wait cycle insertion is disabled by ASTCR.
8	W20	1	R/W	

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 connected
 10: SDRAM with a CAS latency of 3 is connected
 11: SDRAM with a CAS latency of 4 is connected

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	W02	1	R/W	Area 0 Wait Control 2 to 0
1	W01	1	R/W	These bits select the number of program wait cycles inserted when accessing area 0 while bit AST0 in ASTC0 is set to 1. 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
0	W00	1	R/W	

Note: * Supported only by the H8SX1648G Group and the H8SX1648H Group.

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 read strobe in a corresponding area read access.
13	RDN5	0	R/W	
12	RDN4	0	R/W	As shown in figure 9.2, the read strobe for an area for which the RDNn bit is set to 1 is negated one half-cycle earlier than that for an area for which the RDNn bit is cleared to 0. The read data setup and hold times 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 negated one half-cycle at the end of the read cycle. 1: In an area n read access, the \overline{RD} signal is negated one half-cycle before the end of the read cycle. (n = 7 to 0)
7 to 0	—	All 0	R	Reserved These are read-only bits and cannot be modified.

- Notes:
1. In an external address space which is specified as byte control SRAM interface, RDNCr setting is ignored and the same operation when RDNn = 1 is performed.
 2. In an external address space which is specified as the burst ROM interface, RDNCr setting is ignored and the same operation when RDNn = 0 is performed for read accesses by the CPU and EXDMAC* cluster transfer.

* Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

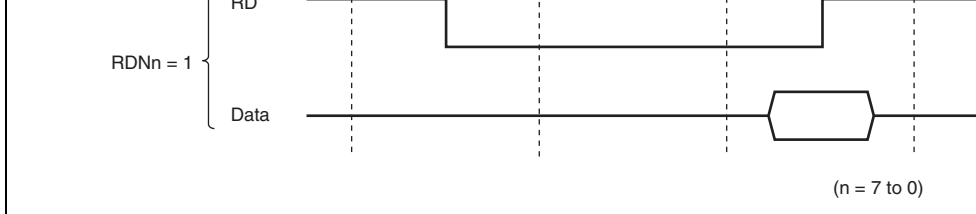


Figure 9.2 Read Strobe Negation Timing (Example of 3-State Access Space)

9.2.5 \overline{CS} Assertion Period Control Registers (CSACR)

CSACR selects whether or not the assertion periods of the chip select signals (\overline{CSn}) and address signals for the basic bus, byte-control SRAM, burst ROM, and address/data multiplexed bus interface are to be extended. Extending the assertion period of the \overline{CSn} and address signals extends the setup time and hold time of read strobe (\overline{RD}) and write strobe ($\overline{LHWR}/\overline{LLWR}$) to be flexible 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	C
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	C
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 (Tt) is extended
(n = 7 to 0)

7	CSXT7	0	R/W	\overline{CS} and Address Signal Assertion Period Cont
6	CSXT6	0	R/W	These bits specify whether or not the Tt cycle
5	CSXT5	0	R/W	inserted (see figure 9.3). When an area for wh
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 i
2	CSXT2	0	R/W	after the normal access cycle.
1	CSXT1	0	R/W	0: In access to area n, the \overline{CSn} and address a
0	CSXT0	0	R/W	period (Tt) is not extended
				1: In access to area n, the \overline{CSn} and address a
				period (Tt) is extended
				(n = 7 to 0)

Notes: In burst ROM interface, the CSXTn settings are ignored during read accesses by and EXDMAC* cluster transfer.

* Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

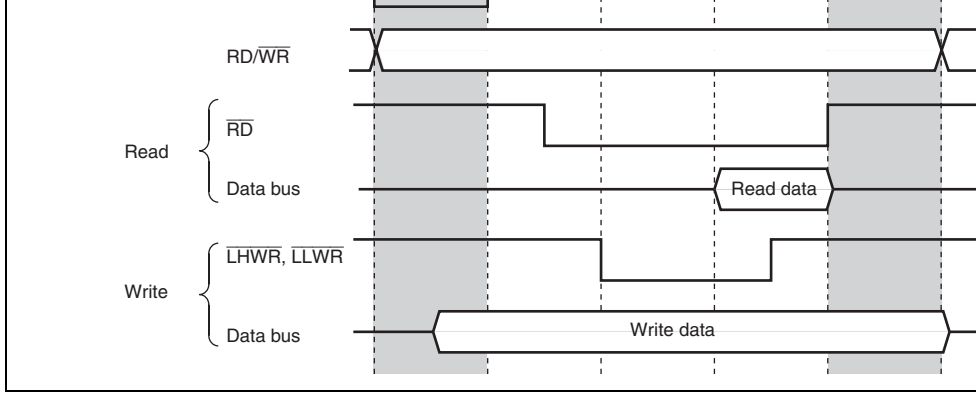


Figure 9.3 \overline{CS} and Address Assertion Period Extension
 (Example of Basic Bus Interface, 3-State Access Space, and RDNn = 0)

Bit Name	IDLSEL7	IDLSEL6	IDLSEL5	IDLSEL4	IDLSEL3	IDLSEL2	IDLSEL1
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	<p>Idle Cycle Insertion 3</p> <p>Inserts an idle cycle between the bus cycles w DMAC or EXDMAC* single address transfer (cycle) is followed by external access.</p> <p>0: No idle cycle is inserted 1: An idle cycle is inserted</p>
14	IDLS2	1	R/W	<p>Idle Cycle Insertion 2</p> <p>Inserts an idle cycle between the bus cycles w external write cycle is followed by external read cycle.</p> <p>0: No idle cycle is inserted 1: An idle cycle is inserted</p>
13	IDLS1	1	R/W	<p>Idle Cycle Insertion 1</p> <p>Inserts an idle cycle between the bus cycles w external read cycles of different areas continuously.</p> <p>0: No idle cycle is inserted 1: An idle cycle is inserted</p>

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 inserted in 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 inserted in each area for the idle insertion condition specified by IDLS1 and IDLS0.
5	IDLSEL5	0	R/W	0: Number of idle cycles to be inserted for area specified by IDLCA1 and IDLCA0.
4	IDLSEL4	0	R/W	1: Number of idle cycles to be inserted for area specified by IDLCB1 and IDLCB0.
3	IDLSEL3	0	R/W	
2	IDLSEL2	0	R/W	
1	IDLSEL1	0	R/W	
0	IDLSEL0	0	R/W	(n = 7 to 0)

Note: * Supported only by the groups H8SX/1648G Group and the H8SX/1648H Group

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

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Bit	Bit Name	Initial Value	R/W	Description
15	BRLE	0	R/W	<p>External Bus Release Enable</p> <p>Enables/disables external bus release.</p> <p>0: External bus release disabled $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$, and $\overline{\text{BREQO}}$ pins can be used as I/O ports</p> <p>1: External bus release enabled*</p> <p>For details, see section 13, I/O Ports.</p>
14	BREQOE	0	R/W	<p>$\overline{\text{BREQO}}$ Pin Enable</p> <p>Controls outputting the bus request signal ($\overline{\text{BREQ}}$) to the external bus master in the external bus request state when an internal bus master performs an external address space access.</p> <p>0: $\overline{\text{BREQO}}$ output disabled $\overline{\text{BREQO}}$ pin can be used as I/O port</p> <p>1: $\overline{\text{BREQO}}$ output enabled</p>
13, 12	—	All 0	R	<p>Reserved</p> <p>These are read-only bits and cannot be modified.</p>

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. When area 2 is specified as the synchronous DRAM space, the setting of this bit does not affect synchronous DRAM space access operation. 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 13, I/O Ports.
7	DKC	0	R/W	$\overline{\text{DACK}}$ Control Selects the timing of DMAC transfer acknowledge signal assertion. 0: $\overline{\text{DACK}}$ signal is asserted at the B ϕ falling edge 1: $\overline{\text{DACK}}$ signal is asserted at the B ϕ rising edge
6	EDKC*	0	R/W	$\overline{\text{EDACK}}$ Control Controls the assertion timing of an acknowledge signal for an EXDMAC transfer. 0: $\overline{\text{EDACK}}$ signal asserted at the falling edge of B ϕ 1: $\overline{\text{EDACK}}$ signal asserted at the rising edge of B ϕ
5 to 0	—	All 0	R	Reserved These are read-only bits and cannot be modified.

Notes: When external bus release is enabled or input by the $\overline{\text{WAIT}}$ pin is enabled, make sure the ICR bit to 1. For details, see section 13, I/O Ports.

* Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These are read-only bits and cannot be modified.
5	EBCCS*	0	R/W	External Bus Cycle Control Select Selects the method for external bus arbitration. 0: Releases the bus depending on the priority. 1: Executes the bus cycle alternatively when a conflict occurs between a bus request by the EXDTC external bus master or refresh bus and a refresh bus request by an external space access by the CPU, DMAC, or DTC.
4	IBCCS	0	R/W	Internal Bus Cycle Control Select Selects the internal bus arbiter function. 0: Releases the bus mastership according to the priority. 1: Executes the bus cycles alternatively when a conflict of bus mastership request conflicts with a DMAC or DTC bus mastership request.
3, 2	—	All 0	R	Reserved These are read-only bits and cannot be modified.
1	—	1	R/W	Reserved This bit is always read as 1. The write value should always be 1.

ENDIANCR selects the endian format for each area of the external address space. Though the format of this LSI is big endian, data can be transferred in the little endian format during address space access.

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

Bit	7	6	5	4	3	2	1
Bit Name	LE7	LE6	LE5	LE4	LE3	LE2	—
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	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 endi
4	LE4	0	R/W	1: Data format of area n is specified as little end
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 modified.

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	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 selected
12	BCSEL4	0	R/W	BROMCR, DRAMCR* and MPXCR must be cleared to
11	BCSEL3	0	R/W	0.
10	BCSEL2	0	R/W	0: Area n is basic bus interface
9	BCSEL1	0	R/W	1: Area n is byte control SRAM interface
8	BCSEL0	0	R/W	(n = 7 to 0)
7 to 0	—	All 0	R	Reserved

These are read-only bits and cannot be modified.

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

Bit Name	BSTRM1	BSTS12	BSTS11	BSTS10	—	—	BSWD11	B
Initial Value	0	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 bit, clear bit BCSEL0 in SRAMCR to 0. 0: Basic bus interface or byte-control SRAM interface 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 modified.

Specifies the area 1 bus interface as a basic interface or a burst ROM interface. To set this bit to 1, clear BCSEL1 in SRAMCR to 0.

0: Basic bus interface or byte-control SRAM interface

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 access
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 modified
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 area 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)

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

Bit Name	BE	RCDM	DDS	EDDS	—	—	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	<p>Area 2 DRAM Interface Select</p> <p>Selects whether or not area 2 is specified as DRAM/SDRAM interface. When this bit is set to 1, the BCS1 DTYPE bit. When this bit is set to 1, the BCS1 SRAMCR should be set to 0.</p> <p>0: Basic bus interface or byte-control SRAM interface 1: DRAM/SDRAM interface</p>
14	DTYPE	0	R/W	<p>DRAM Select</p> <p>Selects the type of DRAM to be used in area 2</p> <p>0: DRAM is used in area 2 1: SDRAM is used in area 2</p>
13, 12	—	All 0	R	<p>Reserved</p> <p>The initial value should not be changed.</p>

Selects whether the $\overline{\text{RAS}}$ signal is asserted at the rising edge or falling edge of the $\text{B}\phi$ signal in the Tr cycle during a DRAM access. The relationship between the $\overline{\text{RAS}}$ bit and $\overline{\text{RAS}}$ assertion timing is shown in figure 10-10.

When SDRAM is used, the setting of this bit does not affect operation.

0: RAS signal is asserted at the falling edge of the $\text{B}\phi$ signal in the Tr cycle

1: RAS signal is asserted at the rising edge of the $\text{B}\phi$ 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 address output cycles is two or three during a DRAM access. When SDRAM is used, the setting of this bit does not 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 accessed in high-speed page mode. When DRAM with the page mode is used, connect the $\overline{\text{OE}}$ signal of the DRAM to the $\overline{\text{OE}}$ signal of DRAM. 0: DRAM/SDRAM is accessed with full access mode 1: DRAM/SDRAM is accessed in high-speed page mode

interface, the READ/WRITE command is issued. The DRAM/SDRAM interface is enabled only after the issuance of the ACTV command when the same address is accessed consecutively.

0: RAS up mode when the DRAM/SDRAM is accessed

1: RAS down mode when the DRAM/SDRAM is accessed

5	DDS	0	R/W	<p>DMAC Single Address Transfer Option</p> <p>Selects whether a DMAC single address transfer through the DRAM/SDRAM interface is enabled in full access mode or is also enabled in fast-page mode.</p> <p>When clearing the BE bit to 0 to disable a burst access to the DRAM/SDRAM interface, a DMAC single address transfer is performed in full access mode regardless of the setting of this bit.</p> <p>This bit does not affect an external access by external masters or a DMAC dual address transfer. Setting this bit to 1 changes the \overline{DACK} output timing.</p> <p>0: DMAC single address transfer through the DRAM/SDRAM is enabled only in full access mode</p> <p>1: DMAC single address transfer through the DRAM/SDRAM is also enabled in fast-page mode</p>
---	-----	---	-----	--

This bit does not affect an external access by masters or an EXDMAC dual address transfer. this bit to 1 changes the EDACK output timing.

- 0: EXDMAC single address transfer through th
DRAM/SDRAM is enabled only in full acces
- 1: EXDMAC single address transfer through th
DRAM/SDRAM is also enabled in fast-page
mode

3	—	0	R	Reserved
2	—	0	R/W	The initial value should not be changed.



0: Shifted by 0 bits
 A23 to A9 are compared for 8-bit access space
 A23 to A10 are compared for 16-bit access space
 10: Shifted by 10 bits
 A23 to A10 are compared for 8-bit access space
 A23 to A11 are compared for 16-bit access space
 11: Shifted by 11 bits
 A23 to A11 are compared for 8-bit access space
 A23 to A12 are compared for 16-bit access space

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

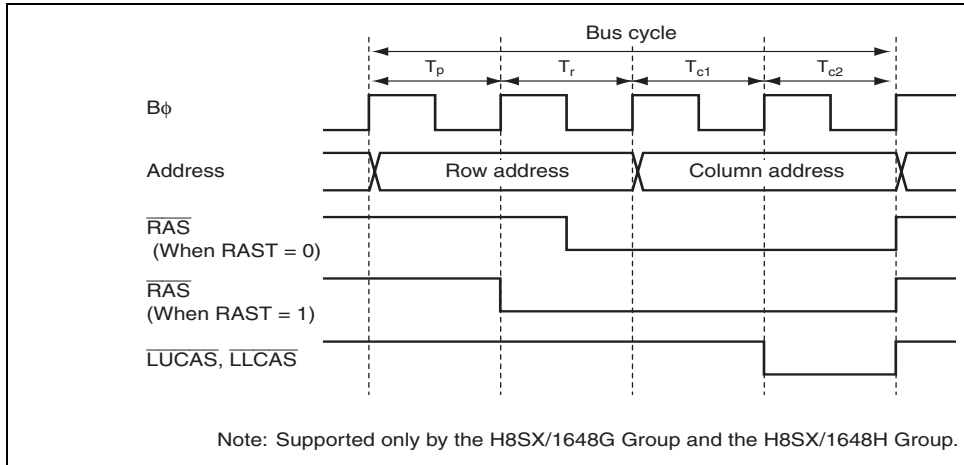


Figure 9.4 $\overline{\text{RAS}}$ Assertion Timing (Column Address Output for 2 states in Full Access Mode)

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, 14	—	All 0	R	Reserved The initial value should not be changed.
13	TPC1	0	R/W	Precharge Cycle Control
12	TPC0	0	R/W	Select the number of RAS precharge cycles or normal access and a refresh cycle. 00: One cycle 01: Two cycles 10: Three cycles 11: Four cycles
11, 10	—	All 0	R	Reserved The initial value should not be changed.
9	RCD1	0	R/W	RAS-CAS Wait Control
8	RCD0	0	R/W	Select the number of wait cycles inserted between RAS and 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.

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

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	<p>Mode Register Set Enable</p> <p>Enables the setting in the SDRAM mode register section 9.11.14, Setting SDRAM Mode Register.</p> <p>0: Disables to set the SDRAM mode register</p> <p>1: Enables to set the SDRAM mode register</p>
14 to 12	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The initial value should not be changed.</p>
11, 10	—	0	R/W	<p>Reserved</p> <p>The initial value should not be changed.</p>
9	—	0	R	Reserved
8	—	0	R/W	The initial value should not be changed.
7	CKSPE	0	R/W	<p>Clock Suspend Enable</p> <p>Enables the clock suspend mode in which read output cycles are extended. Setting this bit to 1 extends the output cycles in which read data is output from SDRAM.</p> <p>0: Disables the clock suspend mode</p> <p>1: Enables the clock suspend mode</p>

9.2.16 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	F
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	T
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.

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

011: Counts on Pφ/128
 100: Counts on Pφ/512
 110: Counts on Pφ/2048
 111: Counts on Pφ/4096

7	RFSHE	0	R/W	Refresh Control Enables or disables refresh control. When refresh control is disabled, the refresh timer can be used as an interval timer. In single-chip activation mode, the setting of this bit should be made after setting the EXPE bit in SYSCR. 1. For SYSCR, see section 3, MCU Operating Mode. 0: Refresh control enabled 1: Refresh control disabled
6	RLW2	0	R/W	Refresh Cycle Wait Control
5	RLW1	0	R/W	Select the number of wait cycles during a CAS refresh cycle for the DRAM interface and a refresh cycle for the SDRAM interface.
4	RLW0	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

0: Disables self-refresh

1: Enables self-refresh

2	TPCS2	0	R/W	Precharge Cycle Control during Self-Refresh
1	TPCS1	0	R/W	Selects the number of precharge cycles immediately after a self-refresh cycle. The number of actual precharge cycles is the sum of the number of precharge cycles selected by these bits and bits TPC1 and TPC0.
0	TPCS0	0	R/W	

000: No wait cycle inserted
001: One wait cycle inserted
010: Two wait cycles inserted
011: Three wait cycles inserted
100: Four wait cycles inserted
101: Five wait cycles inserted
110: Six wait cycles inserted
111: Seven wait cycles inserted

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

Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

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: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

9.2.18 Refresh Time Constant Register (RTCOR)

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

The RTCOR value is always compared with the RTCNT value. When they match, the CMIF bit 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

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

4. External access bus: A bus that accesses external devices via the external bus interface

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

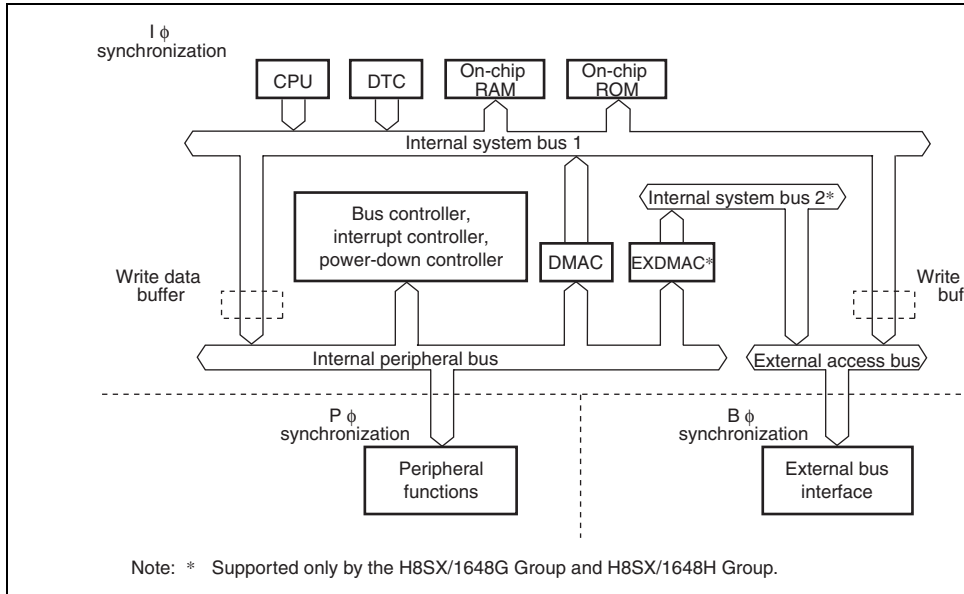


Figure 9.5 Internal Bus Configuration

	Bus controller CPU DTC DMAC EXDMAC* Internal memory Clock pulse generator Power down control
P ϕ	I/O ports TPU PPG TMR WDT SCI A/D D/A IIC2
B ϕ	External bus interface

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

The frequency of each synchronization clock (I ϕ , P ϕ , and B ϕ) is specified by the system control register (SCKCR) independently. For further details, see section 26, Clock Pulse Generator.

There will be cases when P ϕ and B ϕ are equal to I ϕ and when P ϕ and B ϕ are different from I ϕ according to the SCKCR specifications. In any case, access cycles for internal peripheral and external space is performed synchronously with P ϕ and B ϕ , respectively.

the frequency rate of $I\phi$ and $P\phi$ is $m : 1$, 0 to $m-1$ cycles of $1s_y$ may be inserted.

Figure 9.6 shows the external 2-state access timing when the frequency rate of $I\phi$ and $B\phi$

Figure 9.7 shows the external 3-state access timing when the frequency rate of $I\phi$ and $B\phi$

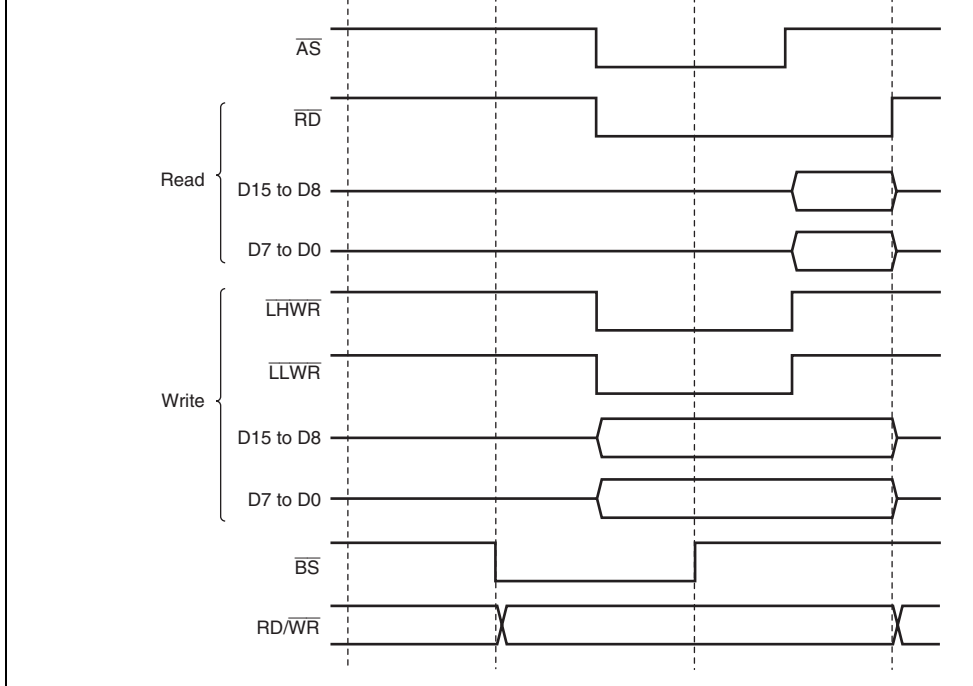


Figure 9.6 System Clock: External Bus Clock = 4:1, External 2-State Access

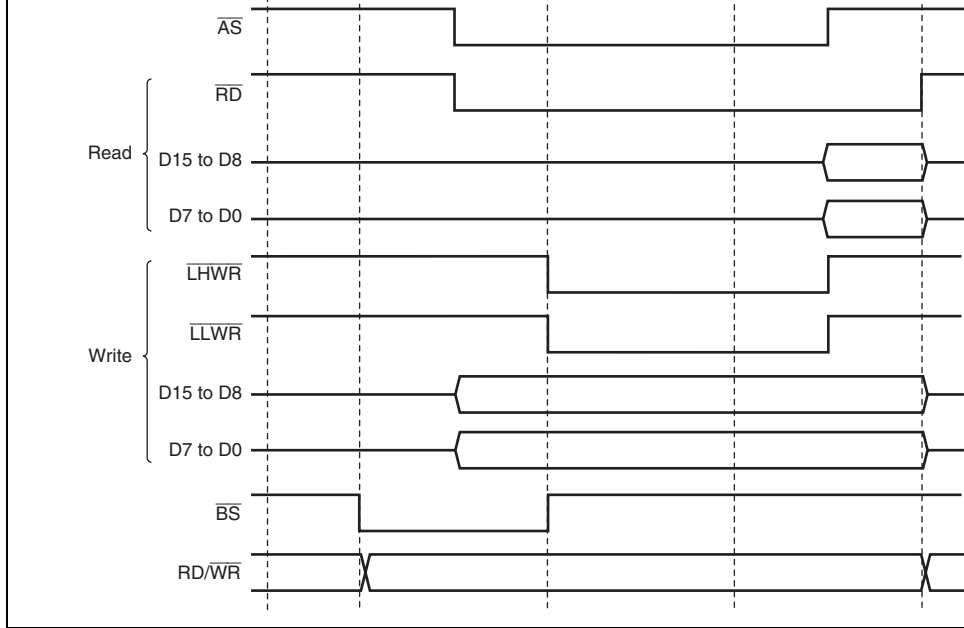


Figure 9.7 System Clock: External Bus Clock = 2:1, External 3-State Access

Bus cycle start	\overline{DS}	Output	<ul style="list-style-type: none"> Signal indicating that the bus cycle has started
Address strobe/ address hold	$\overline{AS/AH}$	Output	<ul style="list-style-type: none"> Strobe signal indicating that the basic bus, burst SRAM, or burst ROM space is accessed and the output on address bus is enabled Signal to hold the address during access to the address/data multiplexed I/O interface
Read strobe	\overline{RD}	Output	Strobe signal indicating that the basic bus, byte control SRAM, burst ROM, or address/data multiplexed I/O space is being read
Read/write	RD/\overline{WR}	Output	<ul style="list-style-type: none"> Signal indicating the input or output direction of the basic bus, burst SRAM, or address/data multiplexed I/O space Write enable signal of the SRAM during access to the byte control SRAM space
Low-high write/ lower-upper byte select	$\overline{LHWR/LUB}$	Output	<ul style="list-style-type: none"> Strobe signal indicating that the basic bus, burst SRAM, or address/data multiplexed I/O space is written, and the upper byte (D15 to D8) of data bus is enabled Strobe signal indicating that the byte control SRAM space is accessed, and the upper byte (D15 to D8) of data bus is enabled
Low-low write/ lower-lower byte select	$\overline{LLWR/LLB}$	Output	<ul style="list-style-type: none"> Strobe signal indicating that the basic bus, burst SRAM, or address/data multiplexed I/O space is written, and the lower byte (D7 to D0) of data bus is enabled Strobe signal indicating that the byte control SRAM space is accessed, and the lower byte (D7 to D0) of data bus is enabled

Row address strobe*	RAS	Output	<ul style="list-style-type: none"> Row address strobe signal when area 1 is specified as DRAM space Row address strobe signal when area 2 is specified as SDRAM space
Column address strobe*	$\overline{\text{CAS}}$	Output	Column address strobe signal when area 2 is specified as SDRAM space
Write enable*	$\overline{\text{WE}}$	Output	<ul style="list-style-type: none"> Write enable signal for DRAM Write enable signal when area 2 is specified as SDRAM space
Lower-upper-column address strobe/lower-upper-data mask enable*	$\overline{\text{LUCAS}}/\overline{\text{DQMLU}}$	Output	<ul style="list-style-type: none"> Lower-upper-column address strobe signal for 8-bit DRAM Upper-column address strobe signal for 16-bit DRAM Lower-upper-data mask enable signal for 8-bit SDRAM Upper-data mask enable signal for 16-bit SDRAM
Lower-lower-column address strobe/lower-lower-data mask enable*	$\overline{\text{LLCAS}}/\overline{\text{DQMLL}}$	Output	<ul style="list-style-type: none"> Lower-lower-column address strobe signal for 8-bit DRAM Lower-column address strobe signal for 16-bit DRAM Column address strobe signal for 8-bit SDRAM Lower-lower-data mask enable signal for 8-bit SDRAM Lower-data mask enable signal for 16-bit SDRAM Data mask enable signal for 8-bit SDRAM

Bus request output	BREQ0	Output	External bus request signal used when internal master accesses external address space in external-bus released state
Data transfer acknowledge 3 (DMAC_3)	$\overline{DACK3}$	Output	Data acknowledge signal for DMAC_3 single transfer
Data transfer acknowledge 2 (DMAC_2)	$\overline{DACK2}$	Output	Data acknowledge signal for DMAC_2 single transfer
Data transfer acknowledge 1 (DMAC_1)	$\overline{DACK1}$	Output	Data acknowledge signal for DMAC_1 single transfer
Data transfer acknowledge 0 (DMAC_0)	$\overline{DACK0}$	Output	Data acknowledge signal for DMAC_0 single transfer
Data transfer acknowledge 3 (EXDMAC_3)*	$\overline{EDACK3}$	Output	Data acknowledge signal for EXDMAC_3 single address transfer
Data transfer acknowledge 2 (EXDMAC_2)*	$\overline{EDACK2}$	Output	Data acknowledge signal for EXDMAC_2 single address transfer
Data transfer acknowledge 1 (EXDMAC_1)*	$\overline{EDACK1}$	Output	Data acknowledge signal for EXDMAC_1 single address transfer
Data transfer acknowledge 0 (EXDMAC_0)*	$\overline{EDACK0}$	Output	Data acknowledge signal for EXDMAC_0 single address transfer
External bus clock	B ϕ	Output	External bus clock

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

CS6	—	—	—	0	0	0	—	—	0	0	—	—	—	—
CS7	—	—	—	0	0	0	—	—	0	0	—	—	—	—
BS	—	—	—	0	0	0	0	0	0	0	0	0	0	0
RD/WR	—	—	—	0	0	0	0	0	0	0	0	0	0	0
AS	Output	Output	—	0	0	0	0	0	—	—	—	—	—	—
AH	—	—	—	—	—	—	—	—	0	0	—	—	—	—
RD	Output	Output	—	0	0	0	0	0	0	0	0	0	—	—
LHWR/LUB	Output	Output	—	0	—	0	0	—	0	—	—	—	—	—
LLWR/LLB	Output	Output	—	0	0	0	0	0	0	0	—	—	—	—
RAS*	—	—	—	—	—	—	—	—	—	—	0	0	0	0
CAS*	—	—	—	—	—	—	—	—	—	—	—	—	0	0
WE*	—	—	—	—	—	—	—	—	—	—	0	0	0	0
LUCAS/DQMLU*	—	—	—	—	—	—	—	—	—	—	0	—	0	—
LLCAS/DQMLL*	—	—	—	—	—	—	—	—	—	—	0	0	0	0
OE*	—	—	—	—	—	—	—	—	—	—	0	0	—	—
CKE*	—	—	—	—	—	—	—	—	—	—	—	—	0	0
WAIT	—	—	—	0	0	0	0	0	0	0	0	0	—	—

[Legend]

0: Used as bus control signal.

—: Not used as bus control signal (I/O port as initial state).

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

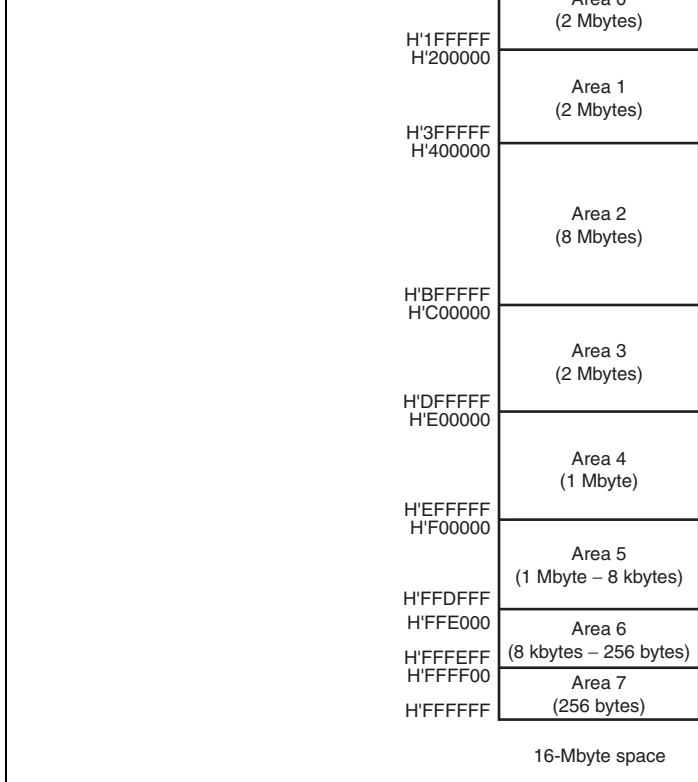


Figure 9.8 Address Space Area Division

be set to 1 when outputting signals CS1 to CS7.

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

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

Figure 9.10 shows the signal output timing when the \overline{CS} signals to be output to areas 5 and 6 are output to the same pin.

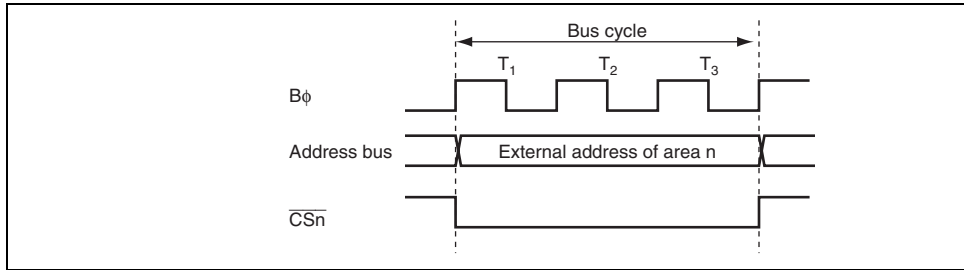


Figure 9.9 \overline{CSn} Signal Output Timing ($n = 0$ to 7)

9.5.4 External Bus Interface

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

(1) Type of External Bus Interface

Six types of external bus interfaces are provided and can be selected in area units. Table 9.4 shows each interface name, description, area name to be set for each interface. Table 9.5 shows the settings that can be specified for each interface. The initial state of each area is a basic bus interface.

	address and data multiplexing	
DRAM interface*	Directly connected to DRAM	DRAM space
Synchronous DRAM interface*	Directly connected to synchronous DRAM	Synchronous DRAM

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

Table 9.5 Areas Specifiable for Each Interface

Interface	Related Registers	Areas							
		0	1	2	3	4	5	6	
Basic interface	SRAMCR	0	0	0	0	0	0	0	0
Byte control SRAM interface		0	0	0	0	0	0	0	0
Burst ROM interface	BROMCR	0	0	—	—	—	—	—	—
Address/data multiplexed I/O interface	MPXCR	—	—	—	0	0	0	0	0
DRAM interface*	DRAMCR	—	—	0	—	—	—	—	—
Synchronous DRAM interface*		—	—	0	—	—	—	—	—

Note: * Supported only by the H8SX/1648L Group and the H8SX/1648H Group.

(3) Endian Format

Though the endian format of this LSI is big endian, data can be converted into little endian 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 LE2 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 big

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

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{CS} extension cycles (0, 1, 2)
[+ number of external wait cycles by the \overline{WAIT} pin]

(b) Byte Control SRAM Interface

The number of access cycles in the byte control SRAM interface is the same as that in the basic 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{CS} extension cycles (0, 1, 2)
[+ number of external wait cycles by the \overline{WAIT} pin]

(c) Burst ROM Interface

The number of access cycles at full access in the burst ROM interface is the same as that in the basic bus interface. The number of access cycles in the burst access can be specified as one to 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{CS} extension cycles (0, 1)
[+number of external wait cycles by the \overline{WAIT} pin]
+ number of burst access cycles (1 to 8) × 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 TPC0 in DRACCR. The number of row address output cycles can be specified as one to four cycles by bits RCD1 and RCD0 in DRACCR. The number of column address output cycles can be specified as two or three cycles by the CAST bit in DRAMCR. For the column address output cycle, program 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

$$\begin{aligned} &= \text{number of precharge cycles (1 to 4)} + \text{number of row address output cycles (1 to 4)} \\ &\quad + \text{number of column address output cycles (2 or 3)} \\ &\quad + \text{number of program wait cycles (0 to 7)} \\ &\quad [+ \text{number of external wait cycles by the WAIT pin}] \end{aligned}$$

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

(f) SDRAM Interface

In the SDRAM interface, the numbers of precharge cycles, row address output cycles, and column address output cycles, as well as clock suspend and write-precharge delay, can be specified by bits in DRACCR and WTCRB.

The number of precharge cycles can be specified as one to four cycles by bits TPC1 and TPC0 in DRACCR. The number of row address output cycles can be specified as one to four cycles by bits 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 and TRWL in SDCR.

		=	[0,1]	[1]	[1]	[1]	[0 to 7]	[n]	+T3	+Tt	[0,1]	
			Th	+T1	+T2	+Tpw	+Ttw	+T3	+Tt			
Byte-control SRAM interface		=	Th	+T1	+T2						+Tt	
		=	[0,1]	[1]	[1]						[0,1]	
			Th	+T1	+T2	+Tpw	+Ttw	+T3	+Tt			
		=	[0,1]	[1]	[1]	[0 to 7]	[n]	[1]			[0,1]	
Burst ROM interface		=	Th	+T1	+T2							+Tb
		=	[0,1]	[1]	[1]							[(1 to 8) x m] [(2 to 16) x m]
			Th	+T1	+T2	+Tpw	+Ttw	+T3	+Tt			+Tb
		=	[0,1]	[1]	[1]	[0 to 7]	[n]	[1]				[(1 to 8) x m] [(2 to 16) x m]
Address/data multiplexed I/O interface		=	Tma	+Th	+T1	+T2					+Tt	
		=	[2,3]	[0,1]	[1]	[1]					[0,1]	
			Tma	+Th	+T1	+T2	+Tpw	+Ttw	+T3	+Tt		
		=	[2,3]	[0,1]	[1]	[1]	[0 to 7]	[n]	[1]		[0,1]	
DRAM interface*	Full access	=	TP	+Tr	+Trw	+TC1	+Tpw	+Ttw	+Tc2	+Tc3		
		=	[1 to 4]	[1]	[0 to 3]	[1]	[0 to 7]	[n]	[1]	[0,1]	[0,1]	
	Fast page	=				TC1	+Tpw	+Ttw	+Tc2	+Tc3		
						[1]	[0 to 7]	[n]	[1]	[0,1]		
	Refresh	=	TRp	+TRw	+TRr	+TRc1	+TRcw	+TRc2				
			[1 to 4]	[0 to 3]	[1]	[1]	[0 to 7]	[1]				
	Self-refresh	=	TRp	+TRw	+TRr	+ Software standby mode [1+s]		+TRc3	+TRc4	+TRp		
			[1 to 4]	[0 to 3]	[1]			[1]	[1]	[0 to 7]		
SDRAM interface*	Setting mode register	=	TP	+Tr	+Trw	+Tc1				+Tc2	+Trwl	
			[1 to 4]	[1]	[0 to 3]	[1]				[1]	[0,1]	
	Full access (read)	=	TP	+Tr	+Trw	+Tc1		+Tcl	+Tsp	+Tc2		
			[1 to 4]	[1]	[0 to 3]	[1]		[1 to 3]	[0,1]	[1]		
	Full access (write)	=	TP	+Tr	+Trw	+Tc1				+Tc2	+Trwl	
			[1 to 4]	[1]	[0 to 3]	[1]				[1]	[0,1]	
	Page access (read)	=				Tc1		+Tcl	+Tsp	+Tc2		
						[1]		[1 to 3]	[0,1]	[1]		
	Page access (write)	=				Tc1				+Tc2		
						[1]				[1]		
	Cluster transfer (read)	=	TP	+Tr	+Trw	+Tc1	+Tcb	+Tcl		+Tc2		
			[1 to 4]	[1]	[0 to 3]	[1]	[0 to 31]	[1 to 3]		[1]		
		=				Tc1	+Tcb	+Tcl		+Tc2		
						[1]	[0 to 31]	[1 to 3]		[1]		
	Cluster transfer (write)	=	TP	+Tr	+Trw	+Tc1				+Tc2	+Tcb	
			[1 to 4]	[1]	[0 to 3]	[1]				[1]	[0 to 31]	
		=				Tc1				+Tc2	+Tcb	
						[1]				[1]	[0 to 31]	
	Refresh	=	TRp	+TRr	+TRc1	+TRcw	+TRc2					
			[1 to 4]	[1]	[1]	[0 to 7]	[1]					
	Self-refresh	=	TRp	+TRr	+ Software standby mode [1+s]	+TRc2	+TRc3	+TRp				
			[1 to 4]	[1]		[1]	[1]	[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

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

9.5.5 Area and External Bus Interface

(1) Area 0

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

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

Either of the basic bus interface, byte control SRAM interface, or burst ROM interface can be selected for area 0 by bit BSRM0 in BROMCR and bit BCSEL0 in SRAMCR. Table 9.7 shows the external interface of area 0.

Table 9.7 Area 0 External Interface

Interface	Register Setting	
	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 $\overline{\text{CS2}}$ signal can be output.

The basic bus interface, byte-control SRAM interface, DRAM interface*, or SDRAM interface can be selected for area 2 by the DRAME and DTYPE bits in DRAMCR* and bit BCSEL1 of SRAMCR. Table 9.9 shows the external interface of area 2.

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

(4) Area 3

In externally extended mode, all of area 3 is external address space.

When area 3 external address space is accessed, the $\overline{CS3}$ signal can be output.

Either of the basic bus interface, byte control SRAM interface, or address/data multiplexed interface can be selected for area 3 by bit MPXE3 in MPXCR and bit BCSEL3 in SRAMCR. Table 9.10 shows the external interface of area 3.

Table 9.10 Area 3 External Interface

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

Interface	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 9.12 shows the external interface of area 5.

Table 9.12 Area 5 External Interface

Interface	Register Setting	
	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

(7) Area 6

Area 6 includes internal I/O registers. In external extended mode, area 6 other than on-chip register area is external address space.

When area 6 external address space is accessed, the $\overline{CS6}$ signal can be output.

Either of the basic bus interface, byte control SRAM interface, or address/data multiplexed interface can be selected for area 6 by the MPXE6 bit in MPXCR and the BCSEL6 bit in SRAMCR. Table 9.13 shows the external interface of area 6.

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

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

Either of the basic bus interface, byte control SRAM interface, or address/data multiplexed interface can be selected for area 7 by the MPXE7 bit in MPXCR and the BCSEL7 bit in SRAMCR. Table 9.14 shows the external interface of area 7.

Table 9.14 Area 7 External Interface

Interface	Register Setting	
	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 as four byte accesses, and a longword access, as four byte accesses.

Figures 9.11 and 9.12 illustrate data alignment control for the 8-bit access space. Figure 9.11 shows the data alignment when the data endian format is specified as big endian. Figure 9.12 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 signal	
					LHWR/LUB	RD
					Data bus	
					D15	D8:D7
Byte	n	1	1st	Byte	7	7
Word	n	2	1st	Byte	15	7
			2nd	Byte	7	7
Longword	n	4	1st	Byte	31	7
			2nd	Byte	23	7
			3rd	Byte	15	7
			4th	Byte	7	7

Figure 9.11 Access Sizes and Data Alignment Control for 8-Bit Access Space (Big Endian)

		2nd	Byte
		3rd	Byte
		4th	Byte

Figure 9.12 Access Sizes and Data Alignment Control for 8-Bit Access Space (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 bus (D7 to D0) are used for accesses. The amount of data that can be accessed at one time is one word.

Figures 9.13 and 9.14 illustrate data alignment control for the 16-bit access space. Figure 9.13 shows the data alignment when the data endian format is specified as big endian. Figure 9.14 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 bus, and 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 bus, and byte access for an odd address is performed by using the third byte data bus.


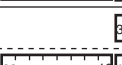

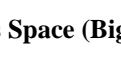
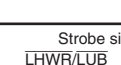
Longword	Even (2n)	2	1st	Word	
			2nd	Word	
	Odd (2n+1)	3	1st	Byte	
			2nd	Word	
			3rd	Byte	

Figure 9.13 Access Sizes and Data Alignment Control for 16-Bit Access Space (Big Endian)




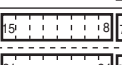
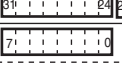

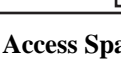


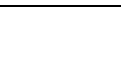
Access Size	Access Address	Access Count	Bus Cycle	Data Size	Strobe signal
					LHWR/LUB
					RD
					Data bus
					D15 D8 D7
Byte	Even (2n)	1	1st	Byte	
	Odd (2n+1)	1	1st	Byte	
Word	Even (2n)	1	1st	Word	
	Odd (2n+1)	2	1st	Byte	
			2nd	Byte	
Longword	Even (2n)	2	1st	Word	
			2nd	Word	
	Odd (2n+1)	3	1st	Byte	
			2nd	Word	
			3rd	Byte	

Figure 9.14 Access Sizes and Data Alignment Control for 16-Bit Access Space (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 when accessing external address space,. For details, see section 9.5.6, Endian and Data Alignment.

9.6.2 I/O Pins Used for Basic Bus Interface

Table 9.15 shows the pins used for basic bus interface.

Table 9.15 I/O Pins for Basic Bus Interface

Name	Symbol	I/O	Function
Bus cycle start	\overline{BS}	Output	Signal indicating that the bus cycle has started
Address strobe	\overline{AS}^*	Output	Strobe signal indicating that an address on the address bus is valid during access
Read strobe	\overline{RD}	Output	Strobe signal indicating the read access
Read/write	RD/\overline{WR}	Output	Signal indicating the data bus input or output direction
Low-high write	\overline{LHWR}	Output	Strobe signal indicating that the upper byte (D7 to D0) is valid during write access
Low-low write	\overline{LLWR}	Output	Strobe signal indicating that the lower byte (D7 to D0) is valid during write access
Chip select 0 to 7	$\overline{CS0}$ to $\overline{CS7}$	Output	Strobe signal indicating that the area is selected
Wait	\overline{WAIT}	Input	Wait request signal used when an external device is accessed

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

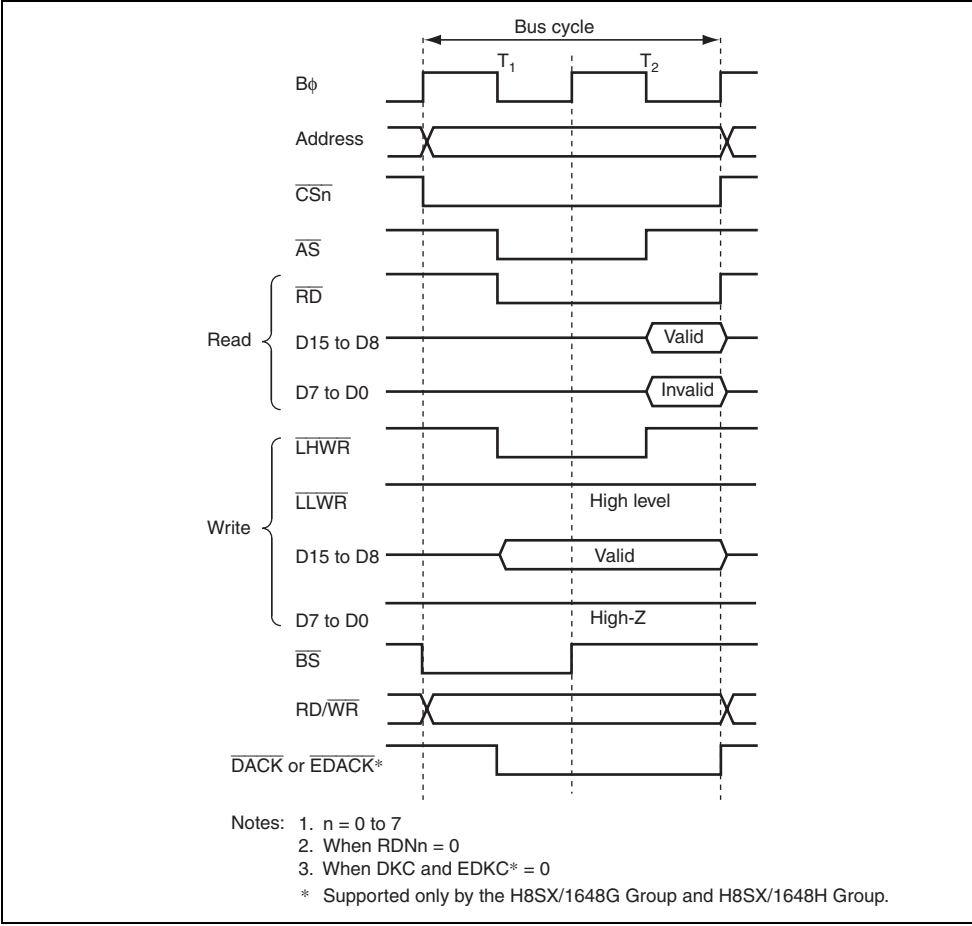
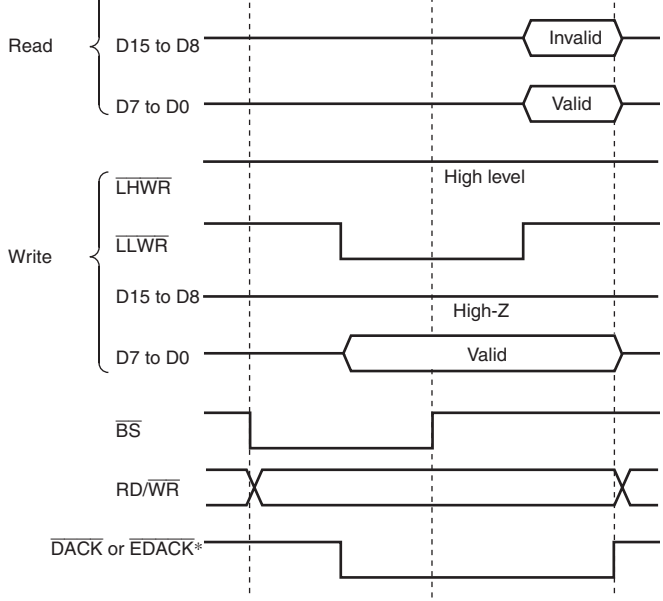
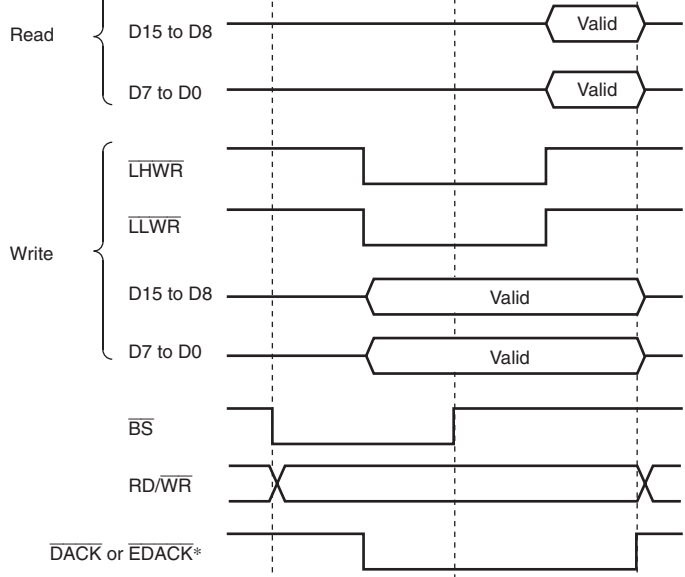


Figure 9.15 16-Bit 2-State Access Space Bus Timing (Byte Access for Even Address)



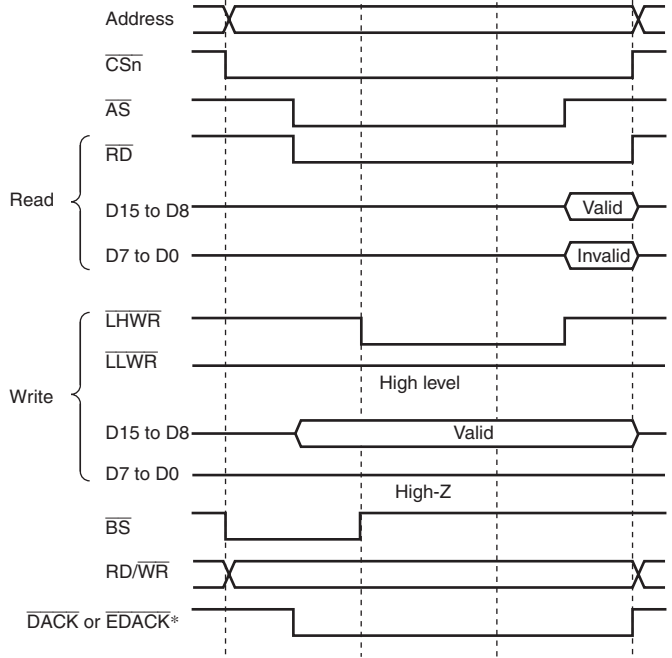
- Notes:
1. $n = 0$ to 7
 2. When $RDNn = 0$
 3. When DKC and $EDKC^* = 0$
- * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Figure 9.16 16-Bit 2-State Access Space Bus Timing (Byte Access for Odd Ad



- Notes:
1. $n = 0$ to 7
 2. When $RDNn = 0$
 3. When DKC and $EDKC^* = 0$
- * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Figure 9.17 16-Bit 2-State Access Space Bus Timing (Word Access for Even Ad



- Notes: 1. n = 0 to 7
 2. When RDn = 0
 3. When DKC and EDKC* = 0
 * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Figure 9.18 16-Bit 3-State Access Space Bus Timing (Byte Access for Even Address)

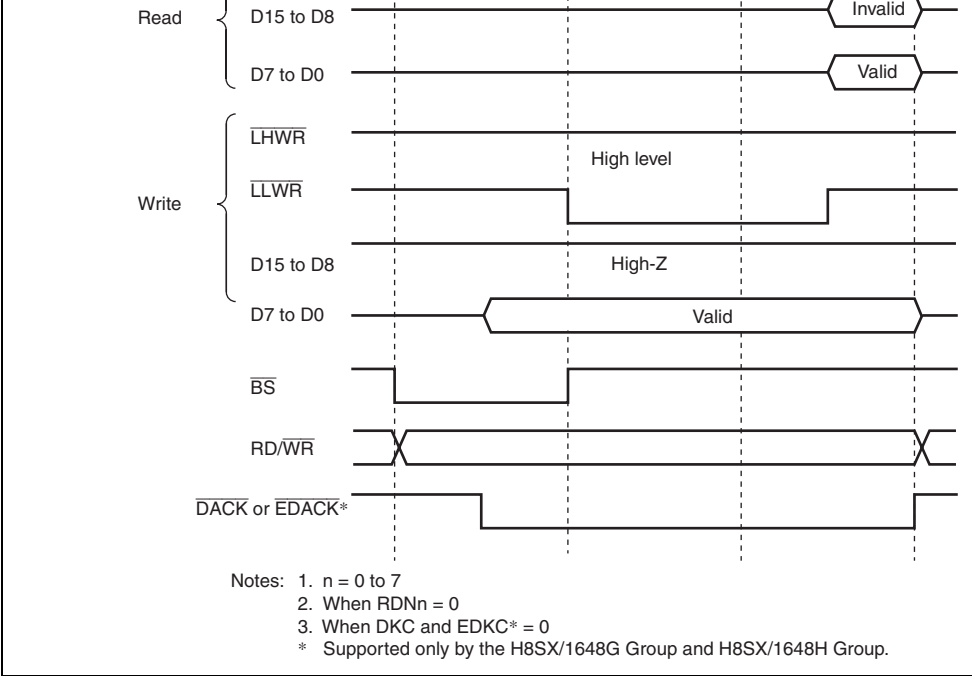


Figure 9.19 16-Bit 3-State Access Space Bus Timing (Word Access for Odd Ad

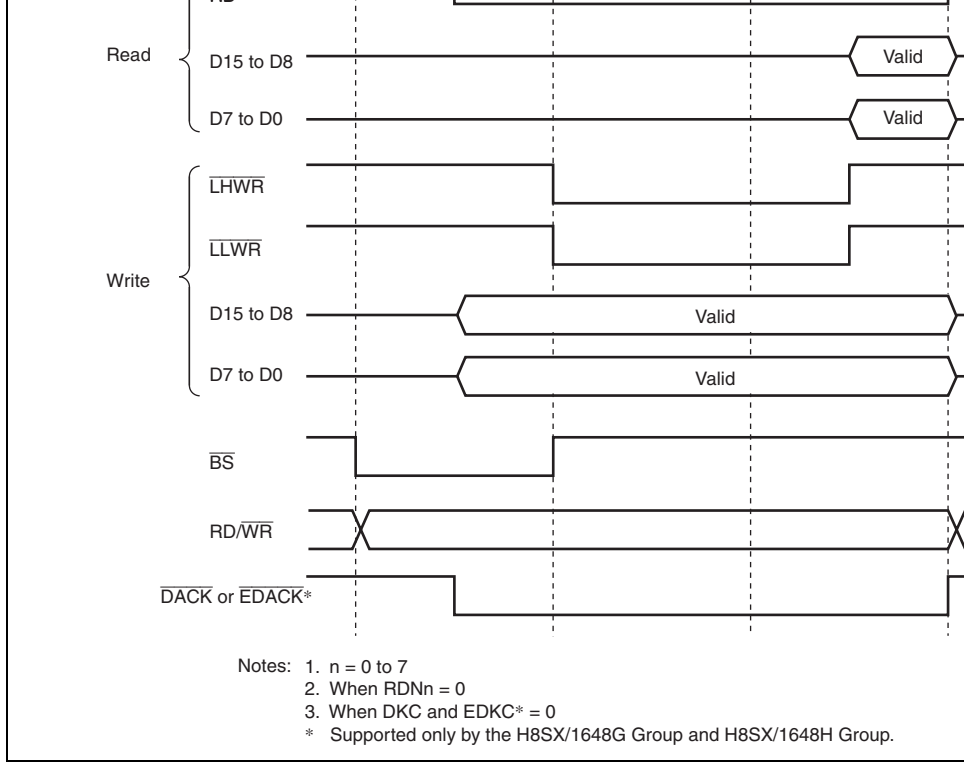
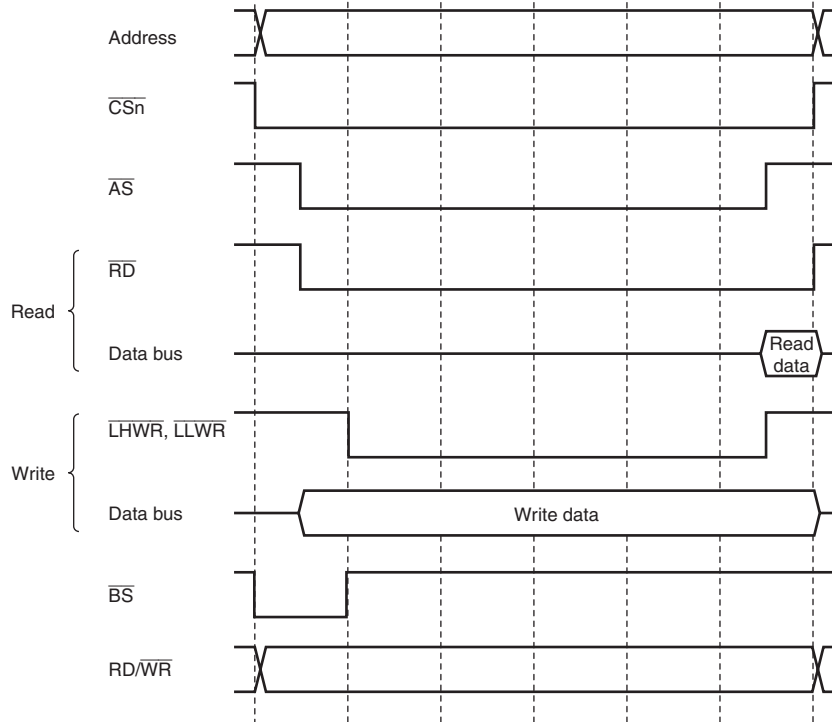


Figure 9.20 16-Bit 3-State Access Space Bus Timing (Word Access for Even A

(2) Pin Wait Insertion

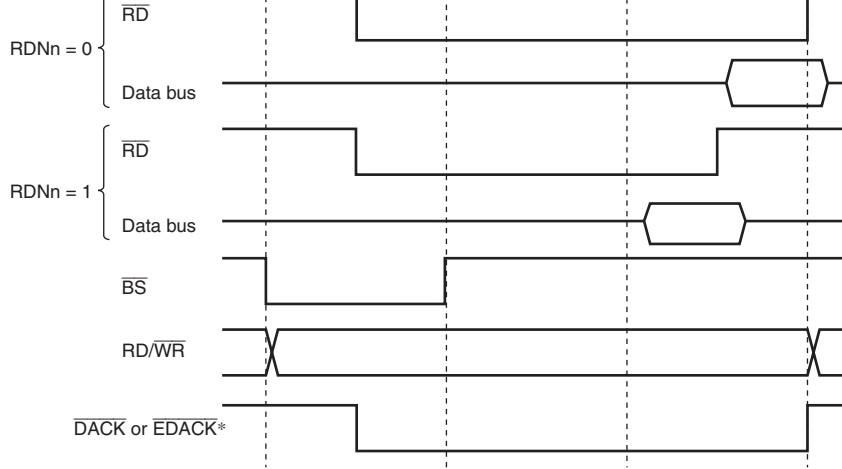
For 3-state access space, when the WAITE bit in BCR1 is set to 1 and the corresponding $\overline{\text{WAIT}}$ pin is set to 1, wait input by means of the $\overline{\text{WAIT}}$ pin is enabled. When the external address space is accessed in this state, a program wait (Tpw) is first inserted according to the WTCRA and WTCRB settings. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of B ϕ in the last T2 or Tpw cycle, another Ttw cycle is inserted until the $\overline{\text{WAIT}}$ pin is brought high. The pin wait insertion is effective when the Tw cycles are inserted to seven cycles or more, or when the number of cycles to be inserted is changed according to the external devices. The WAITE bit is common to all areas. For details on ICR, see section 13, I/O Ports.



- Notes: 1. Upward arrows indicate the timing of $\overline{\text{WAIT}}$ pin sampling.
 2. $n = 0$ to 7
 3. When $\text{RDn} = 0$

Figure 9.21 Example of Wait Cycle Insertion Timing

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.



- Notes: 1. n = 0 to 7
 2. When DKC and EDKC* = 0
 * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Figure 9.22 Example of Read Strobe Timing

3-state access space.

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

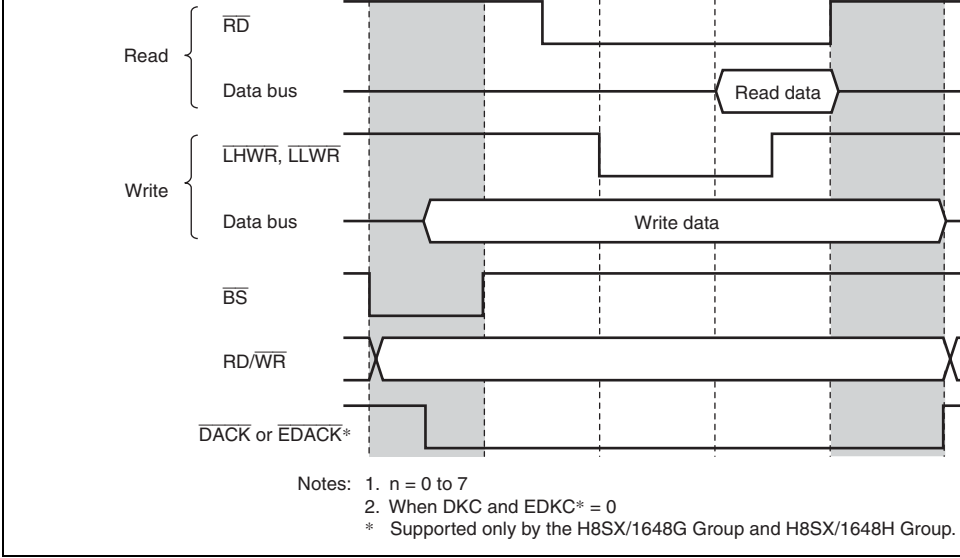
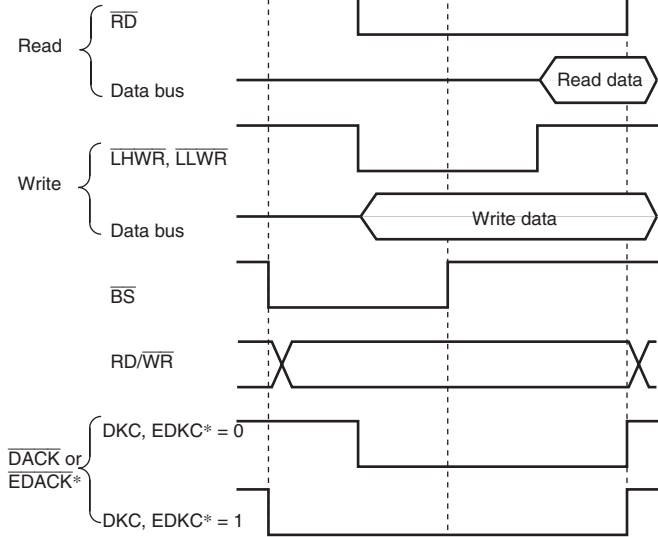


Figure 9.23 Example of Timing when Chip Select Assertion Period is Extended



Notes: 1. n = 7 to 0
 2. $\overline{RD}n = 0$
 * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Figure 9.24 \overline{DACK} and \overline{EDACK}^* Signal Output Timing

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

9.7.1 Byte Control SRAM Space Setting

Byte control SRAM interface can be specified for areas 0 to 7. Each area can be specified as burst ROM interface or address/data multiplexed I/O interface, the SRAMCR setting is used. For the area specified as burst ROM interface or address/data multiplexed I/O interface, the SRAMCR setting is used and byte control SRAM interface cannot be used.

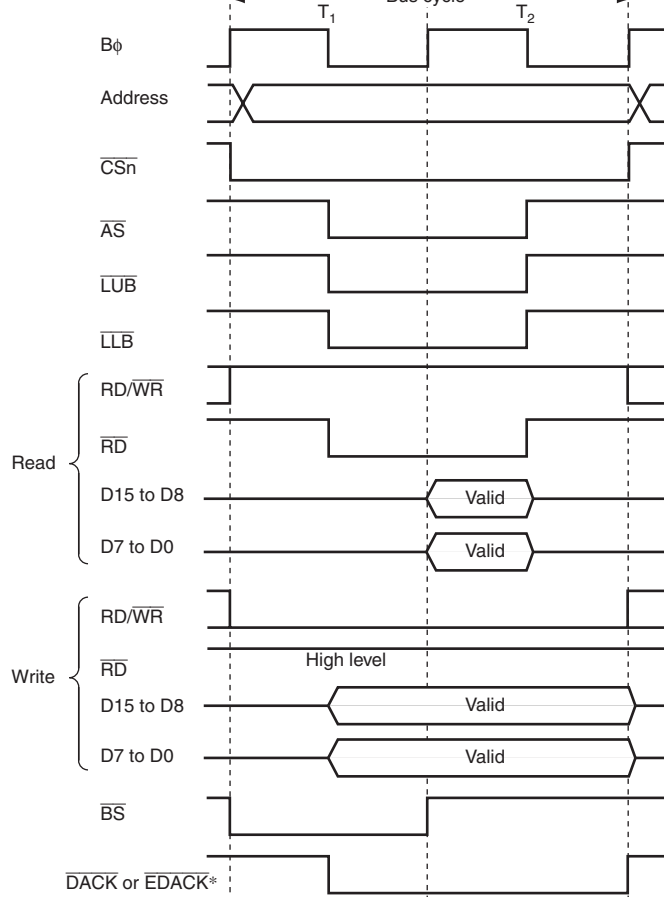
9.7.2 Data Bus

The bus width of the byte control SRAM space can be specified as 16-bit byte control SRAM space according to bits ABWH_n and ABWL_n (n = 0 to 7) in ABWCR. The area specified as burst ROM interface or address/data multiplexed I/O interface 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 section 9.5.6, Endian and Data Alignment.

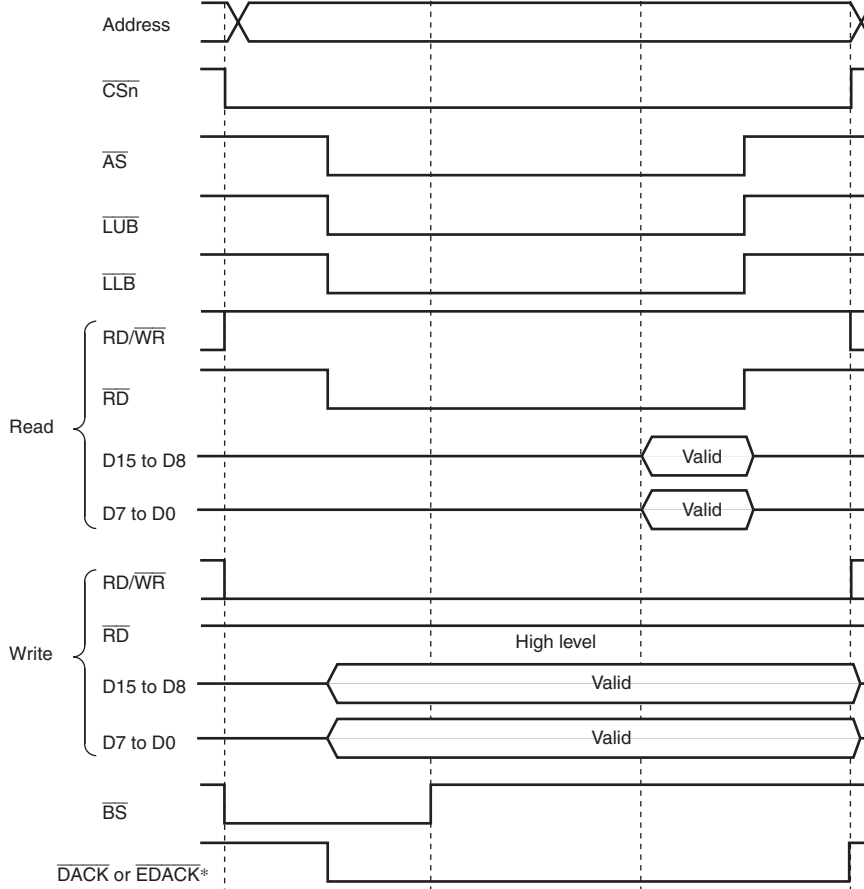
AS/AH	AS	Address strobe	Output	Strobe signal indicating that the address output on the address bus is valid. When the basic bus interface space or byte control SRAM space is accessed
$\overline{\text{CSn}}$	$\overline{\text{CSn}}$	Chip select	Output	Strobe signal indicating that area n is selected
$\overline{\text{RD}}$	$\overline{\text{RD}}$	Read strobe	Output	Output enable for the SRAM when the address control SRAM space is accessed
$\overline{\text{RD}}/\overline{\text{WR}}$	$\overline{\text{RD}}/\overline{\text{WR}}$	Read/write	Output	Write enable signal for the SRAM when the address byte control SRAM space is accessed
$\overline{\text{LHWR}}/\overline{\text{LUB}}$	$\overline{\text{LUB}}$	Lower-upper byte select	Output	Upper byte select when the 16-bit address control SRAM space is accessed
$\overline{\text{LLWR}}/\overline{\text{LLB}}$	$\overline{\text{LLB}}$	Lower-lower byte select	Output	Lower byte select when the 16-bit address control SRAM space is accessed
$\overline{\text{WAIT}}$	$\overline{\text{WAIT}}$	Wait	Input	Wait request signal used when the 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



Note: n = 0 to 7

* Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Figure 9.25 16-Bit 2-State Access Space Bus Timing



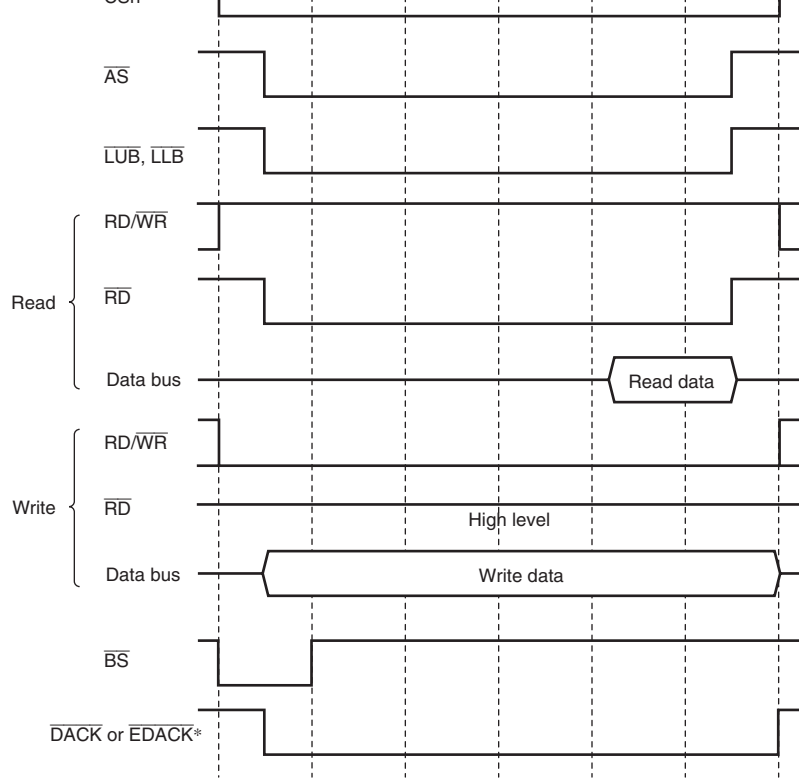
Note: n = 0 to 7

* Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Figure 9.26 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 DD is 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 13, I/O Ports.

Figure 9.27 shows an example of wait cycle insertion timing.



- Notes: 1. Upward arrows indicate the timing of $\overline{\text{WAIT}}$ pin sampling.
 2. $n = 0$ to 7
 * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

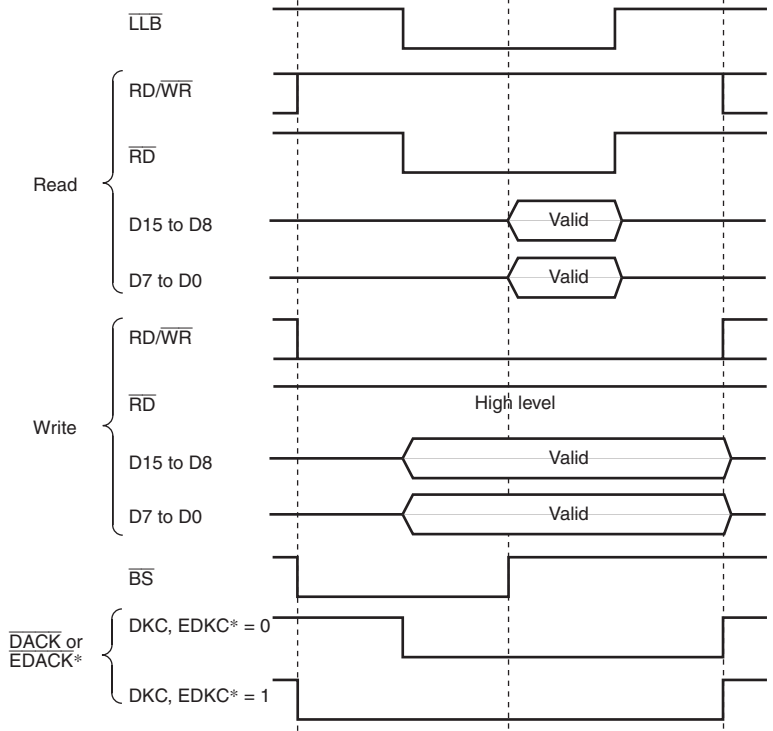
Figure 9.27 Example of Wait Cycle Insertion Timing

In the byte control DMA interface, the extension cycles can be inserted before and after the basic cycle in the same way as the basic bus interface. For details, refer to section 9.6.6, Extension Cycles, and Chip Select (\overline{CS}) Assertion Period.

9.7.8 \overline{DACK} and \overline{EDACK}^* Signal Output Timing

For DMAC or EXDMAC* single address transfers, the \overline{DACK} and \overline{EDACK}^* signal assertion can be modified by using the DKC and EDKC* bits in BCR1.

Figure 9.28 shows the \overline{DACK} and \overline{EDACK}^* signal output timing. Setting the DKC bit or EDKC* bit to 1 asserts the \overline{DACK} or \overline{EDACK}^* signal a half cycle earlier.



Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Figure 9.28 Output Timing for $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}^*$ Signals

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

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

In the burst ROM interface, the burst access covers only read accesses by the CPU and E. cluster transfer. Other accesses are performed with the similar method to the basic bus interface.

9.8.1 Burst ROM Space Setting

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

9.8.2 Data Bus

The bus width of the burst ROM space can be specified as 8-bit or 16-bit burst ROM interface space according to the ABWH_n and ABWL_n 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 (D15 to D0) is valid.

Access size and data alignment are the same as the basic bus interface. For details, see section 9.5.6, 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 output 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 selected
Wait	\overline{WAIT}	Input	Wait request signal used when an external memory space is accessed

according to the settings of BSTS01, BSTS00, BSTS11, and BSTS10 bits in BROMCR.

The basic access timing for burst ROM space is shown in figures 9.29 and 9.30.

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

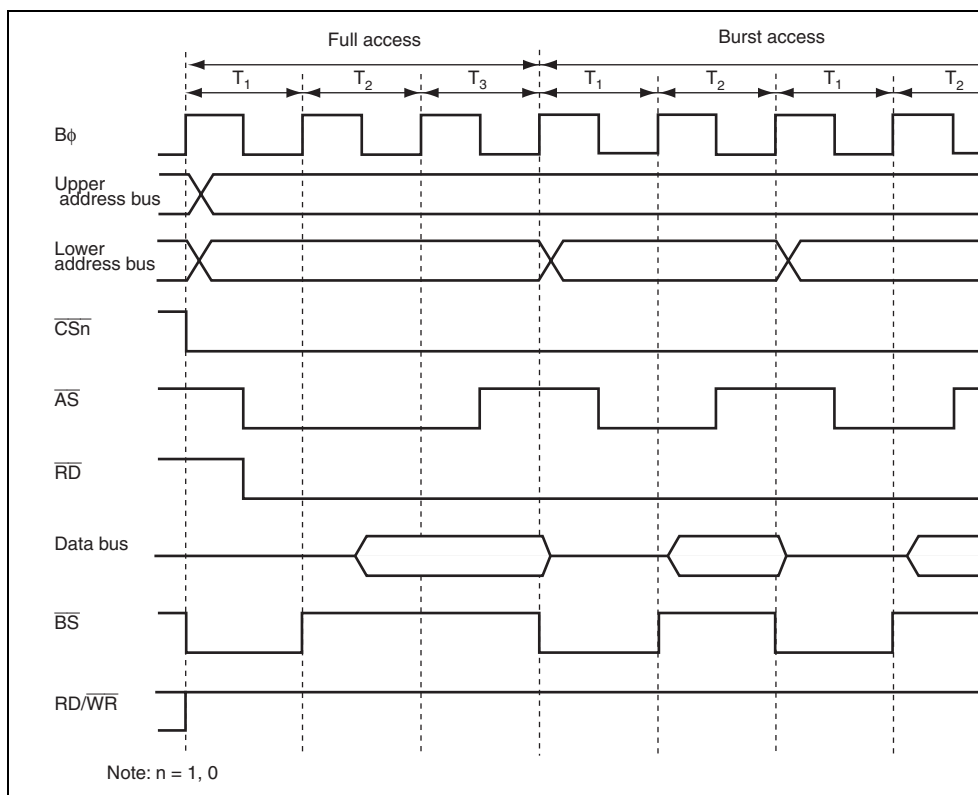
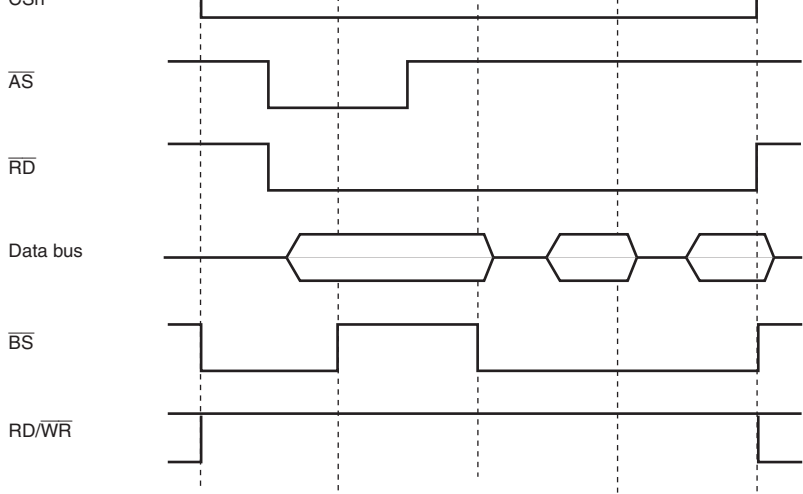


Figure 9.29 Example of Burst ROM Access Timing (ASTn = 1, Two Burst Cycles)



Note: n = 1, 0

Figure 9.30 Example of Burst ROM Access Timing (ASTn = 0, One Burst C

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

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

9.8.7 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 basic bus interface.

For the burst ROM space, the burst access can be enabled only in read access by the CPU and EXDMAC* cluster transfer. In this case, the setting of the corresponding CSXn bit in the burst control register is ignored and an extension cycle can be inserted only before the full access cycle. Note that an extension cycle can be inserted before or after the burst access cycles.

In accesses other than read accesses by the CPU and EXDMAC* cluster transfer, the burst ROM space is equivalent to the basic bus interface space. Accordingly, extension cycles can be inserted before and after the burst access cycles.

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

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

9.9.2 Address/Data Multiplex

In the address/data multiplexed I/O space, data bus is multiplexed with address bus. Table 9.18 shows the relationship between the bus width and address output.

Table 9.18 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	

9.9.3 Data Bus

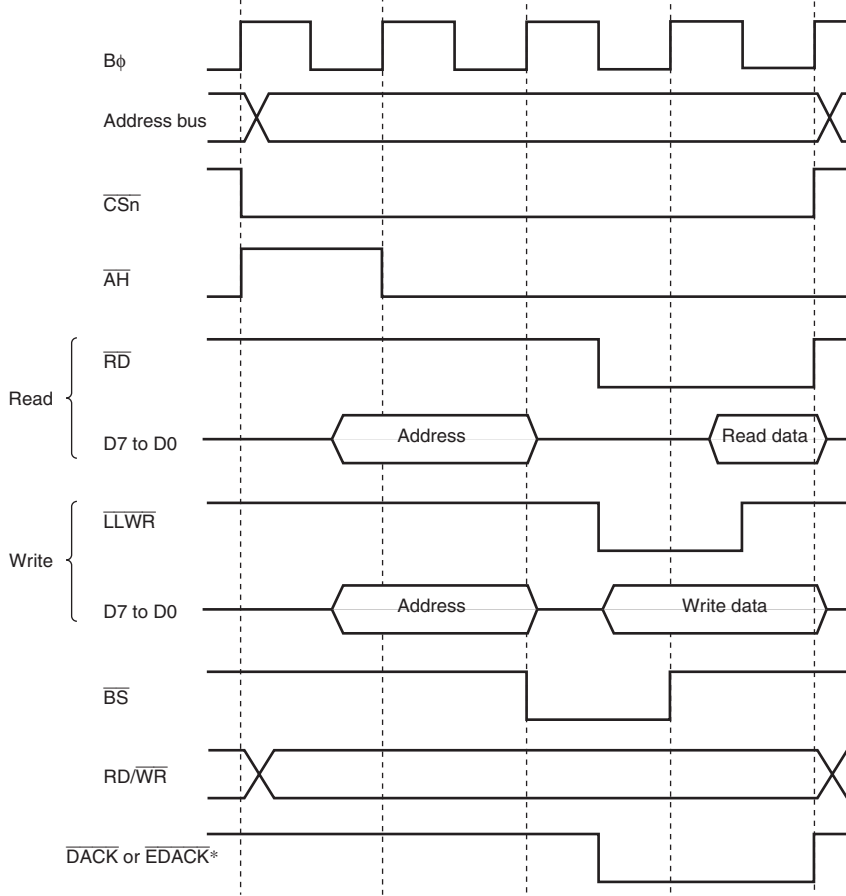
The bus width of the address/data multiplexed I/O space can be specified for either 8-bit access space or 16-bit access space by the ABWHn and ABWLn bits (n = 3 to 7) in ABWCR.

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

For details on access size and data alignment, see section 9.5.6, Endian and Data Alignment.

$\overline{AS}/\overline{AH}$	\overline{AH}^*	Address hold	Output	Signal to hold an address when the address/data multiplexed I/O space is specified
\overline{RD}	\overline{RD}	Read strobe	Output	Signal indicating that the address/data multiplexed I/O space is being read
$\overline{LHWR}/\overline{LUB}$	\overline{LHWR}	Low-high write	Output	Strobe signal indicating that the upper byte (D7 to D8) is valid when the address/data multiplexed I/O space is written
$\overline{LLWR}/\overline{LLB}$	\overline{LLWR}	Low-low write	Output	Strobe signal indicating that the lower byte (D0 to D7) is valid when the address/data multiplexed I/O space is 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 space is specified. D15 to D0 are valid when the 16-bit space is 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 I/O space is accessed
\overline{BS}	\overline{BS}	Bus cycle start	Output	Signal to indicate the bus cycle start
$\overline{RD}/\overline{WR}$	$\overline{RD}/\overline{WR}$	Read/write	Output	Signal indicating the data bus input or output

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



Notes: n = 3 to 7

* Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Figure 9.31 8-Bit Access Space Access Timing (ABWHn = 1, ABWLn = 1)

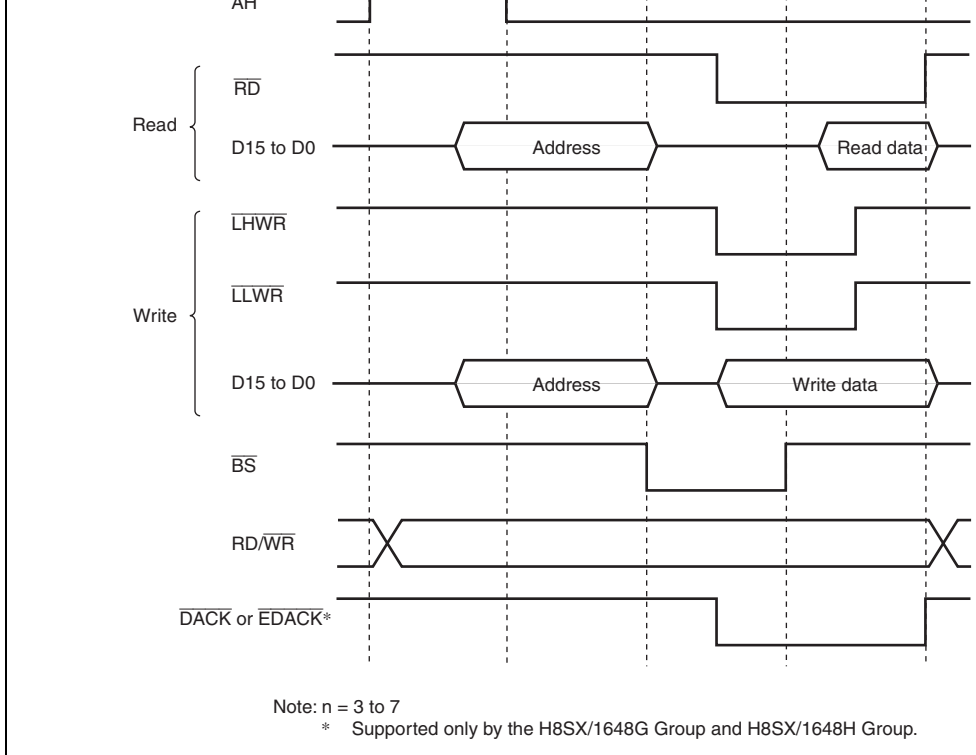


Figure 9.32 16-Bit Access Space Access Timing (ABWHn = 0, ABWLn = 1)

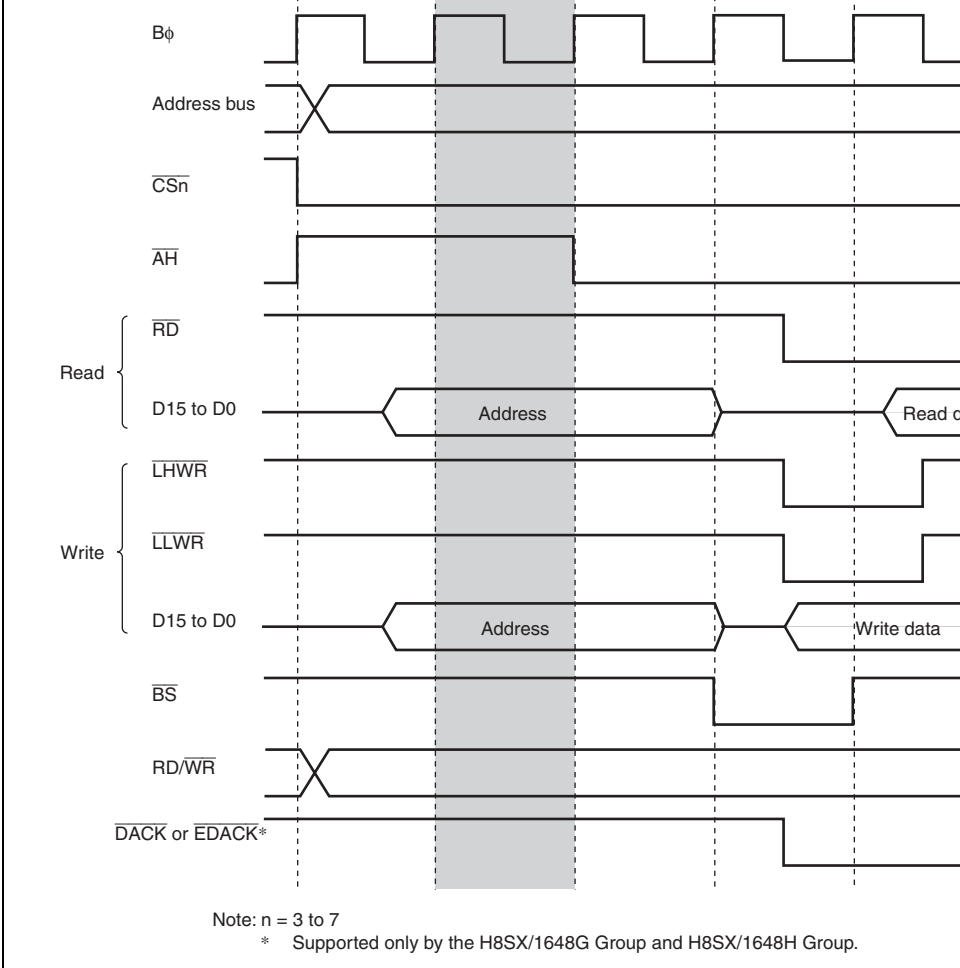


Figure 9.33 Access Timing of 3 Address Cycles (ADDEX = 1)

In the address/data multiplexed I/O interface, the read strobe timing of data bytes can be modified in the same way as in basic bus interface. For details, refer to section 9.6.5, Read Strobe Timing.

Figure 9.34 shows an example when the read strobe timing is modified.

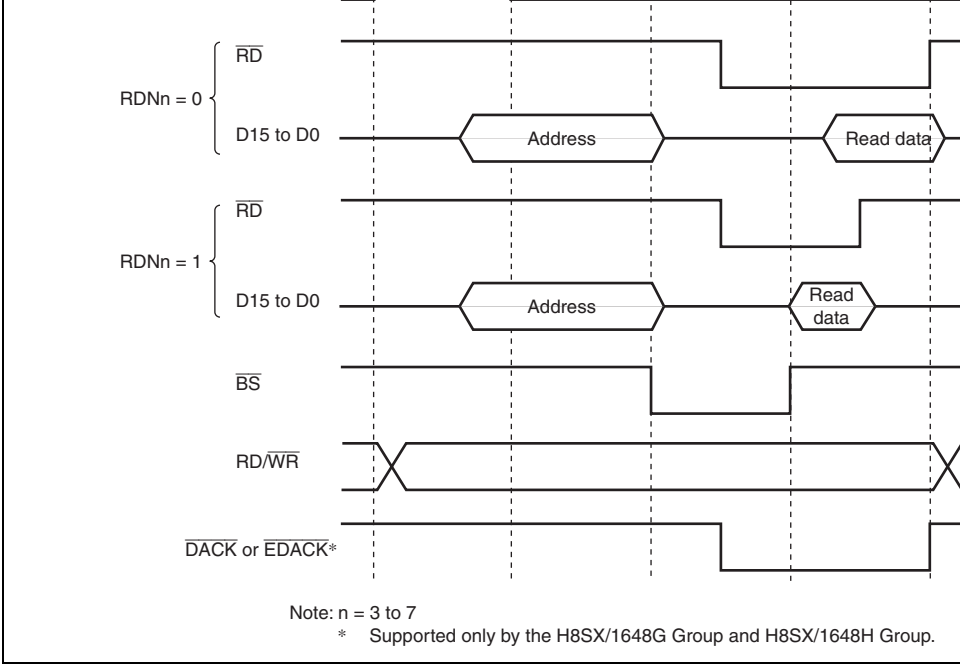
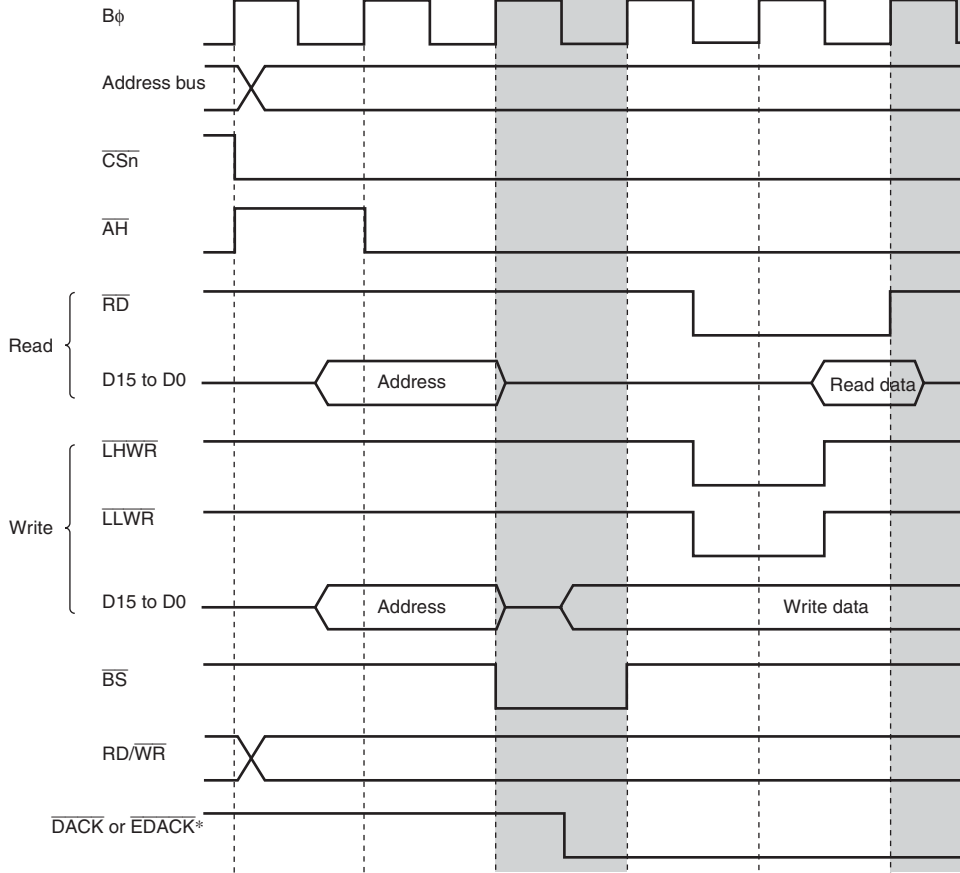


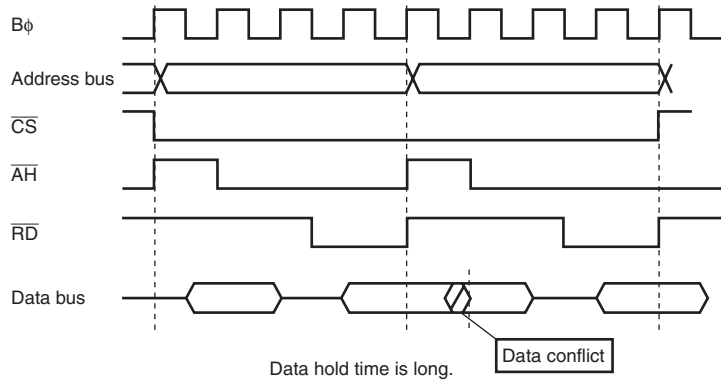
Figure 9.34 Read Strobe Timing



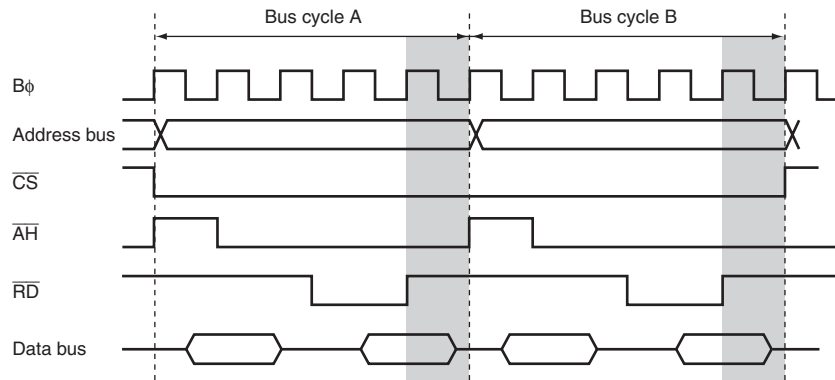
Note: n = 3 to 7

* Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Figure 9.35 Chip Select (\overline{CS}) Assertion Period Extension Timing in Data Cycle

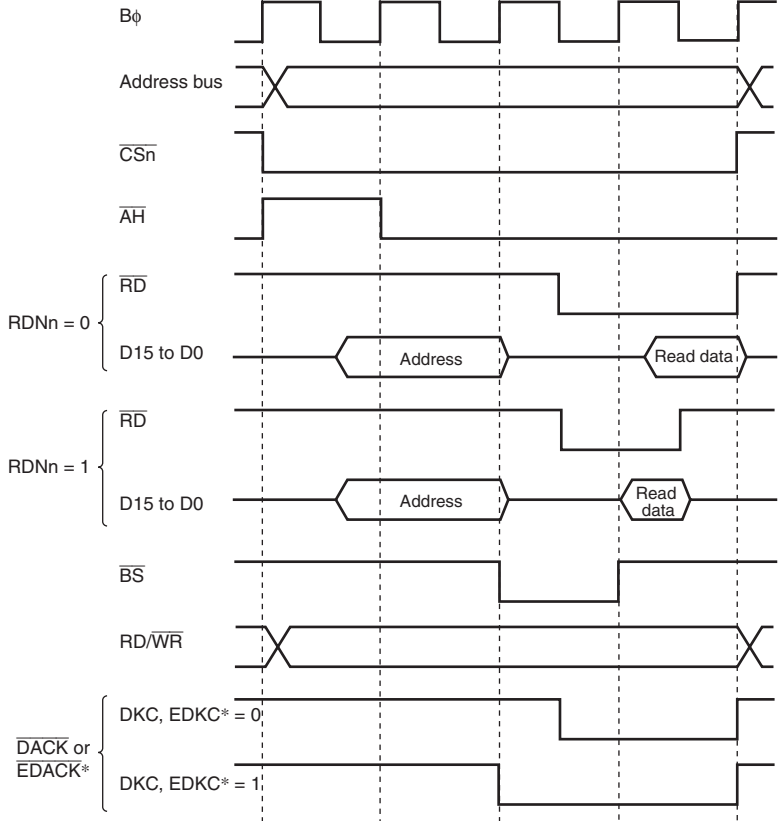


(a) Without \overline{CS} assertion period extension cycle ($CSXTn = 0$)



(b) With \overline{CS} assertion period extension cycle ($CSXTn = 1$)

**Figure 9.36 Consecutive Read Accesses to Same Area
(Address/Data Multiplexed I/O Space)**



Note: $n = 3$ to 7

* Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Figure 9.37 Output Timing for $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}^*$ Signals

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

The bus settings of the DRAM space such as bus width and wait cycle number depend on the settings.

Table 9.20 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

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	A13
				Column address	A23 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3

9.10.3 Data Bus

The data bus width of the DRAM space can be selected from 8 and 16 bits by bits ABWL1 and ABWL2 in ABWCR. DRAM with 16-bit words can be connected directly to 16-bit bus space.

D7 to D0 are valid in 8-bit DRAM space, and D15 to D0 are valid in 16-bit DRAM space.

The data endian format can be selected by bit LE2 in ENDIANCR. For details on the access and alignment, see section 9.5.6, Endian and Data Alignment.

		strobe		space is specified as area 2
$\overline{\text{LUCAS}}/\text{DQMLU}$	$\overline{\text{LUCAS}}$	Lower-upper column address strobe	Output	<ul style="list-style-type: none"> Lower-upper column address strobe when the 32-bit DRAM space is accessed Upper column address strobe when the 16-bit DRAM space is accessed
$\overline{\text{LLCAS}}/\text{DQMLL}$	$\overline{\text{LLCAS}}$	Lower-lower column address strobe	Output	<ul style="list-style-type: none"> Lower-lower column address strobe when the 32-bit DRAM space is accessed Lower column address strobe when the 16-bit DRAM space is accessed
$\overline{\text{OE}}$	$\overline{\text{OE}}$	Output enable	Output	Output enable signal when the DRAM space is accessed
$\overline{\text{WAIT}}$	$\overline{\text{WAIT}}$	Wait	Input	Wait request signal used when a DRAM 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

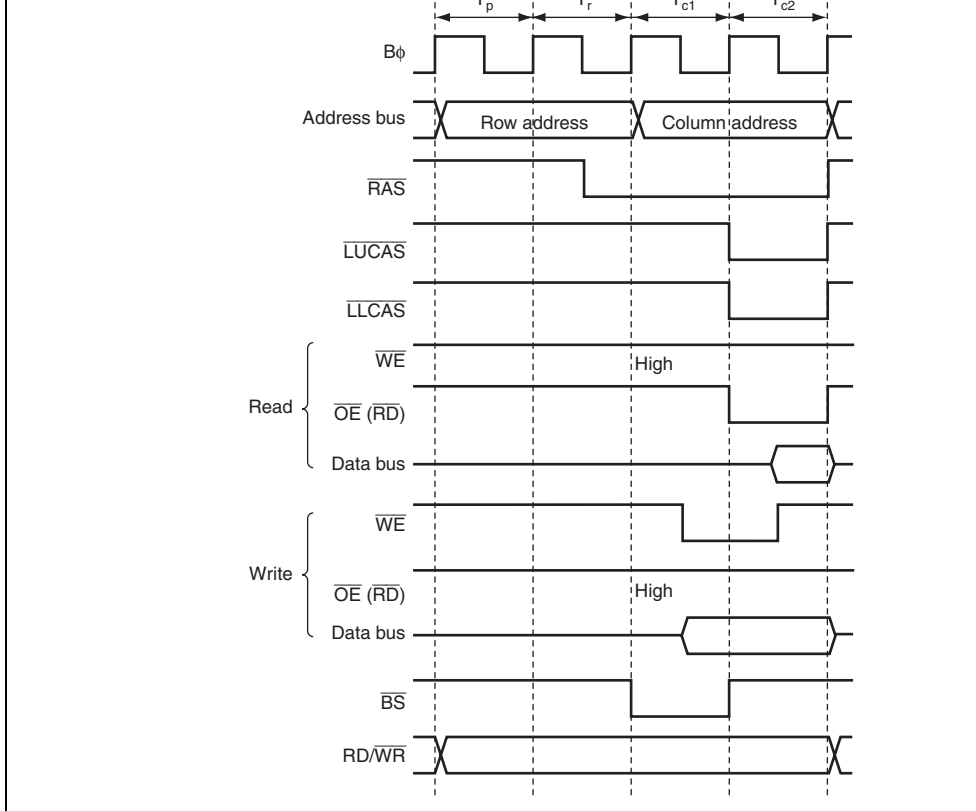


Figure 9.38 DRAM Basic Access Timing (RAS = 0 and CAST = 0)

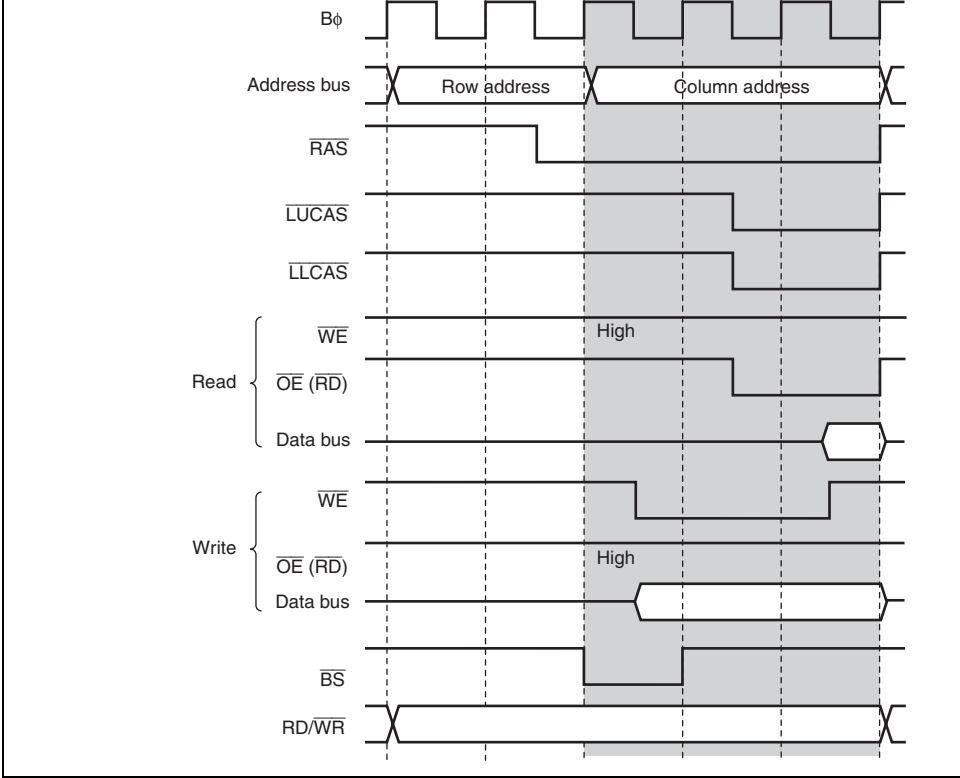


Figure 9.39 Access Timing Example of Column Address Output Cycles for 3 Cl
(RAST = 0)

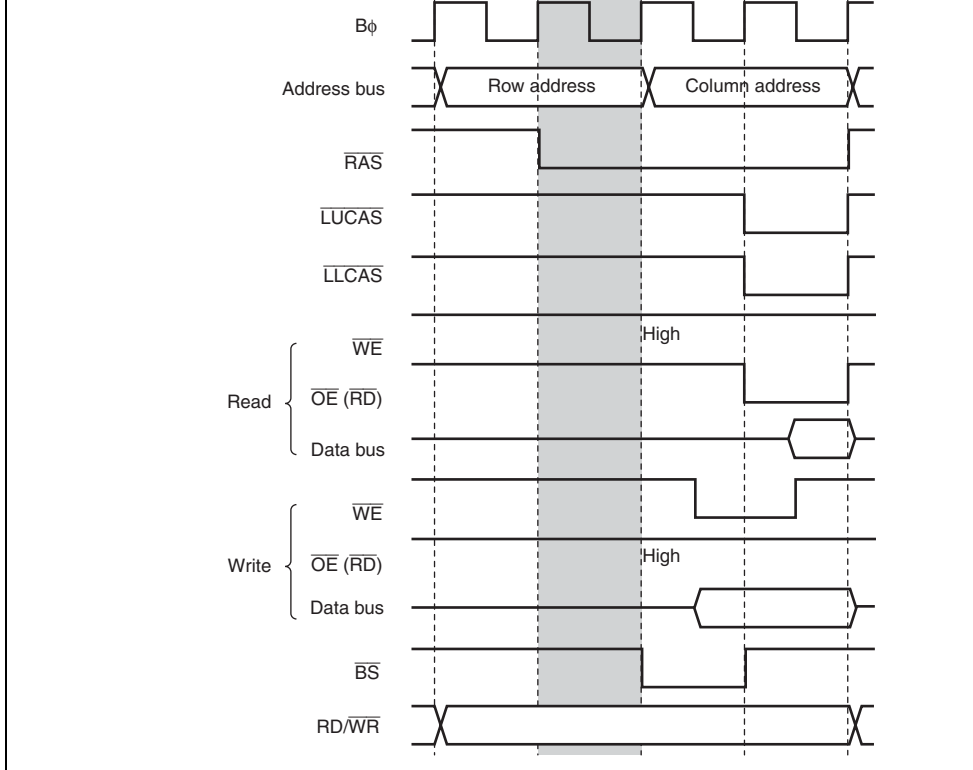


Figure 9.40 Access Timing Example of $\overline{\text{RAS}}$ Signal Driven Low at Start of Tr C (CAST = 0)

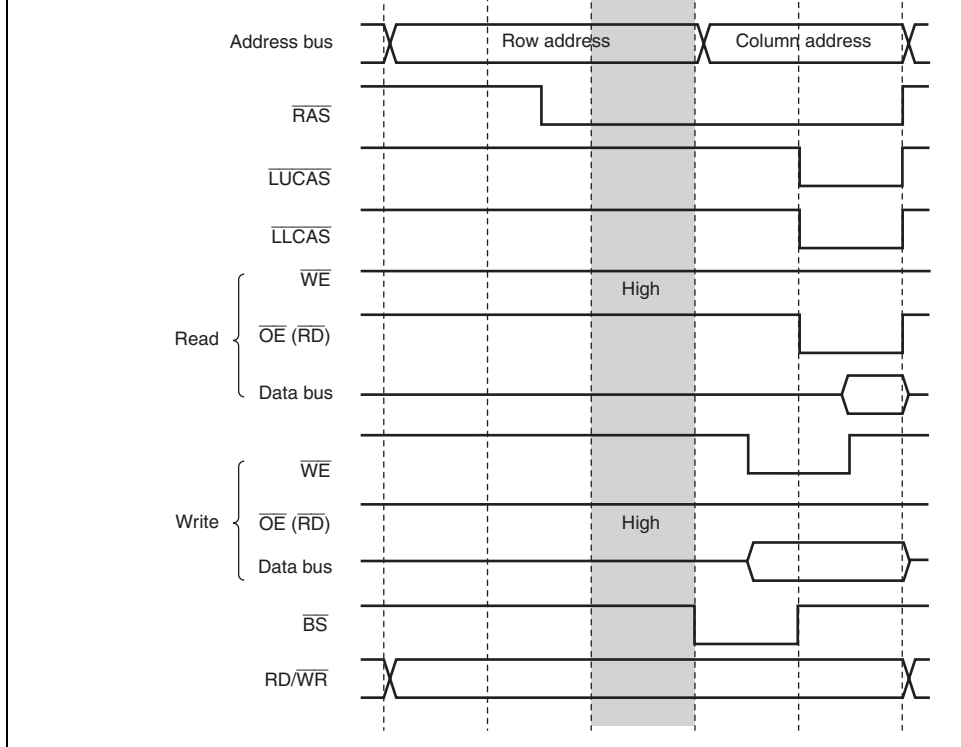


Figure 9.41 Access Timing Example when One Trw Cycle is Specified (RAST=0,

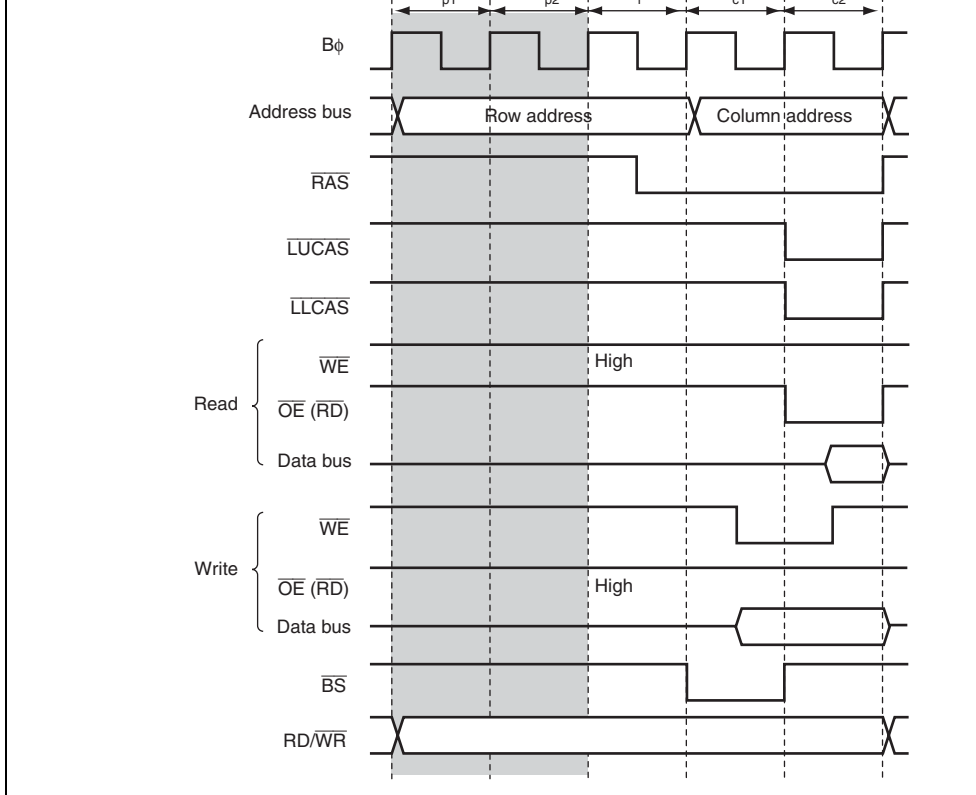


Figure 9.42 Access Timing Example of Two Precharge Cycles (RAST = 0 and CA

between the $Tc1$ and $Tc2$ cycles. The number of wait cycles is selected by bits $W22$ to $W20$ in the $WTCRB$.

(2) Pin Wait Insertion

When the $WAITE$ bit in $BCR1$ is set to 1, and the $AST2$ bit in $ASTCR$ is set to 1, setting the $WAITEN$ bit for the corresponding pin to 1 enables wait input by the \overline{WAIT} pin. When the DRAM is accessed in this state, a program wait (Tpw) is first inserted. If the \overline{WAIT} pin is low at the falling edge of $B\phi$ in the last $Tc1$ or Tpw cycle, another Ttw cycle is inserted until the \overline{WAIT} pin is high. For details on ICR, see section 13, I/O Ports.

Figure 9.43 shows an example of wait cycle insertion timing for 2-cycle column address access.

Figure 9.44 shows an example of wait cycle insertion timing for 3-cycle column address access.

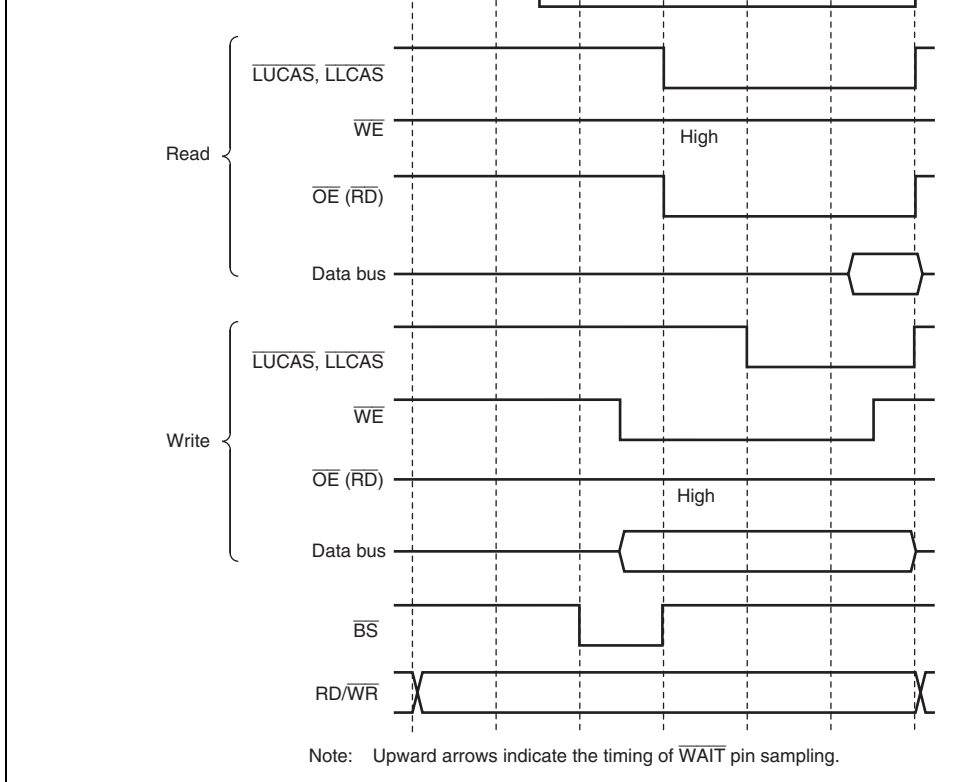


Figure 9.43 Example of Wait Cycle Insertion Timing for 2-Cycle Column Address

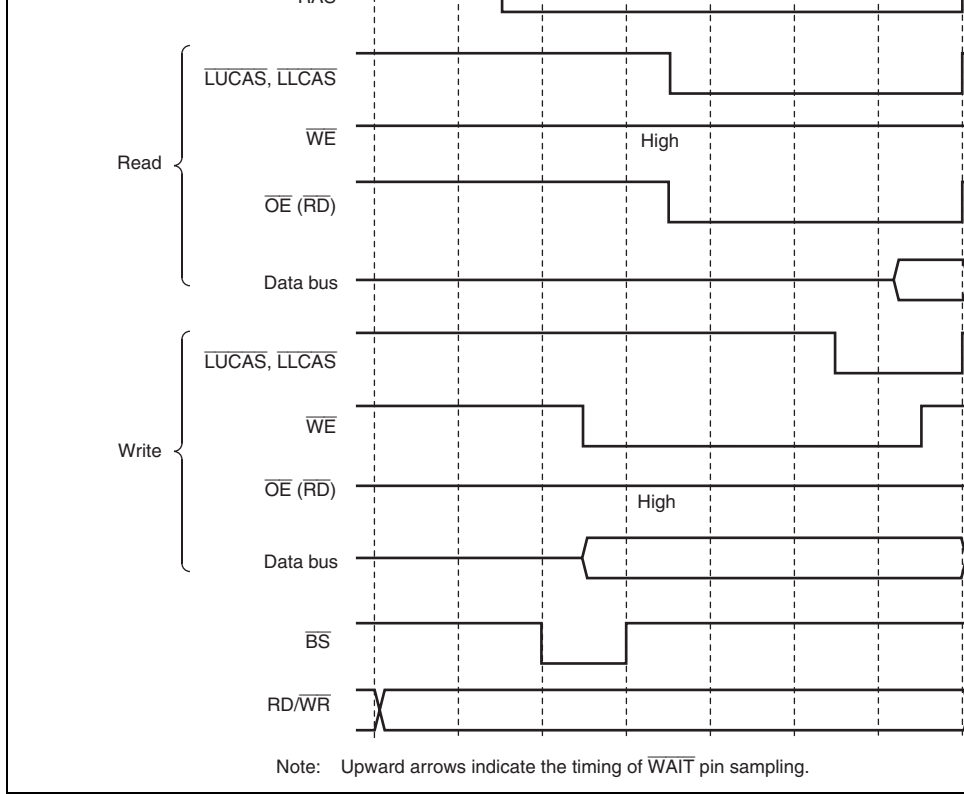
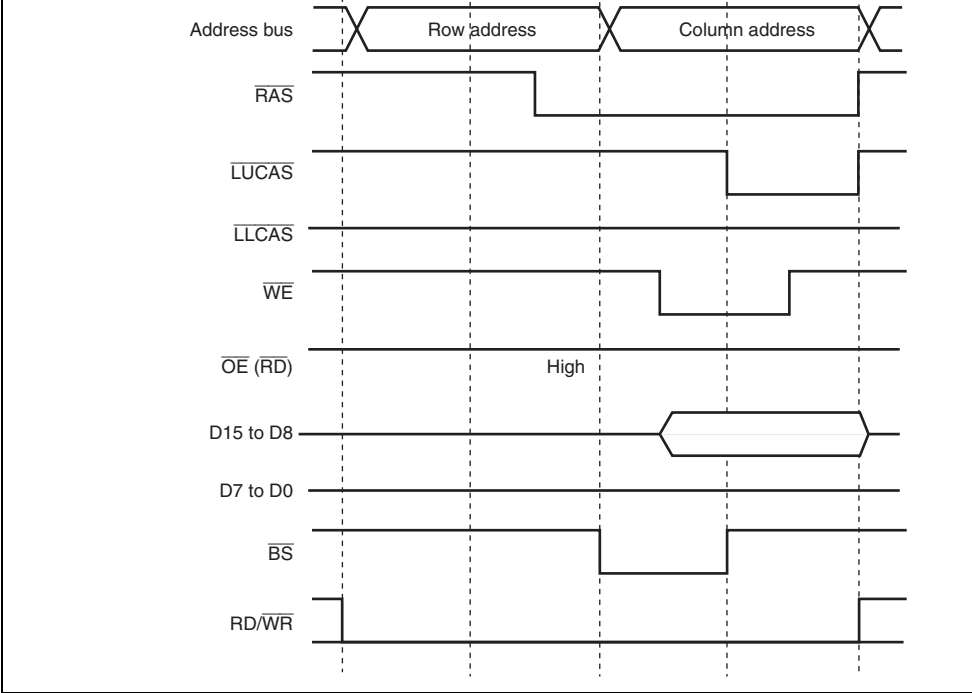


Figure 9.44 Example of Wait Cycle Insertion Timing for 3-Cycle Column Address



**Figure 9.45 Timing Example of Byte Control with Use of Two CAS Signals
(Write Access with Lowest Bit of Address = B'0, RAST = 0, CAST = 0)**

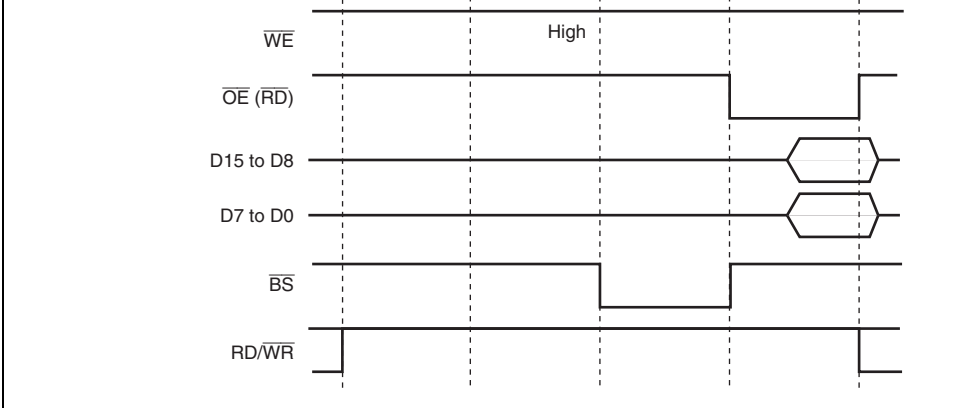


Figure 9.46 Timing Example of Word Control with Use of Two CAS Signals (Read Access with Lowest Bit of Address = B'0, RAST = 0, CAST = 0)

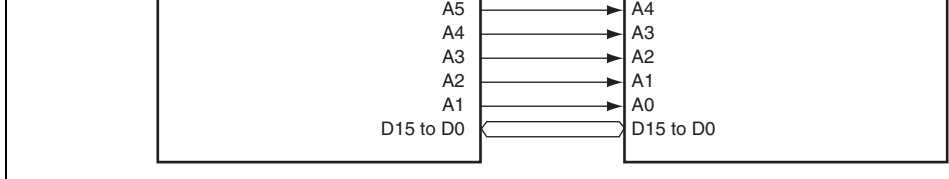


Figure 9.47 Example of Connection for Control with Two CAS Signals

9.10.11 Burst Access Operation

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

The fast-page mode (burst access) can be specified when the BE bit in DRAMCR is set to 1.

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

Figures 9.48 and 9.49 show operation timing of the fast-page mode.

When access cycles to the DRAM space are continued and the row addresses of the consecutive two cycles are the same, output cycles of the CAS and column address signals follow. The column 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 insertion are the same as that of full access mode. For details, see section 9.10.9, Wait Control.

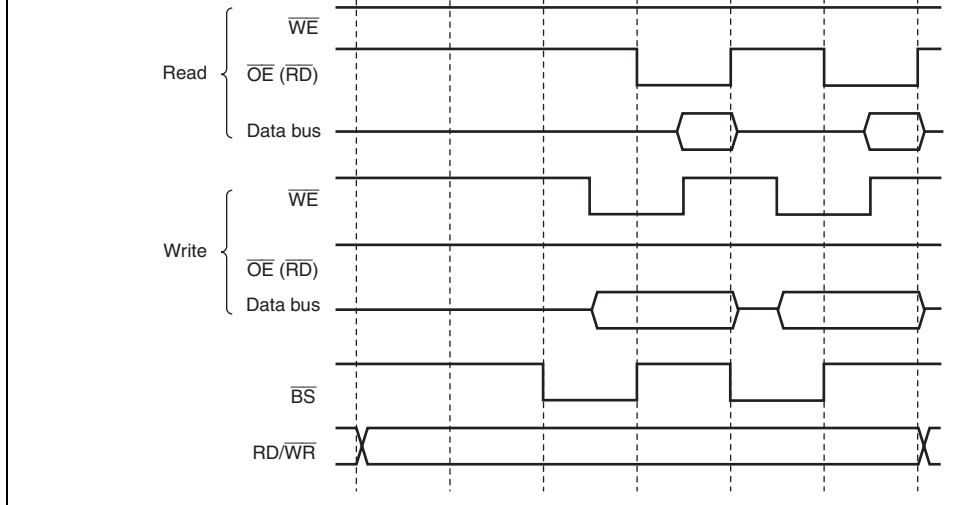


Figure 9.48 Operation Timing of Fast-Page Mode (RAST = 0, CAST = 0)

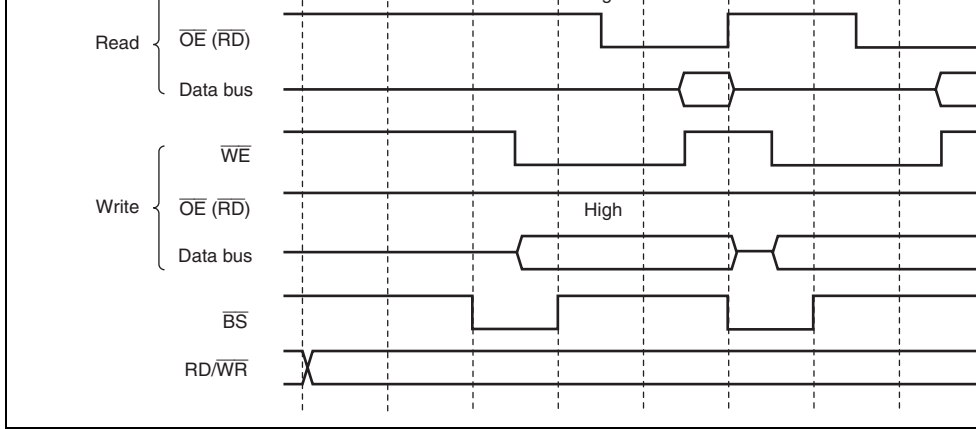


Figure 9.49 Operation Timing of Fast-Page Mode (RAST = 0, CAST = 1)

The fast-page mode access (burst access) is resumed when the row addresses of the current and previous cycle are the same. While other spaces are accessed when the DRAM space is halted, the $\overline{\text{RAS}}$ signal must be low. Figure 9.50 shows a timing example of RAS down mode.

The $\overline{\text{RAS}}$ 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 is stopped with the $\overline{\text{RAS}}$ signal driven low. To make a transition with the $\overline{\text{RAS}}$ signal driven high, 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 SCKCR, see section 26, Clock Pulse Generator.

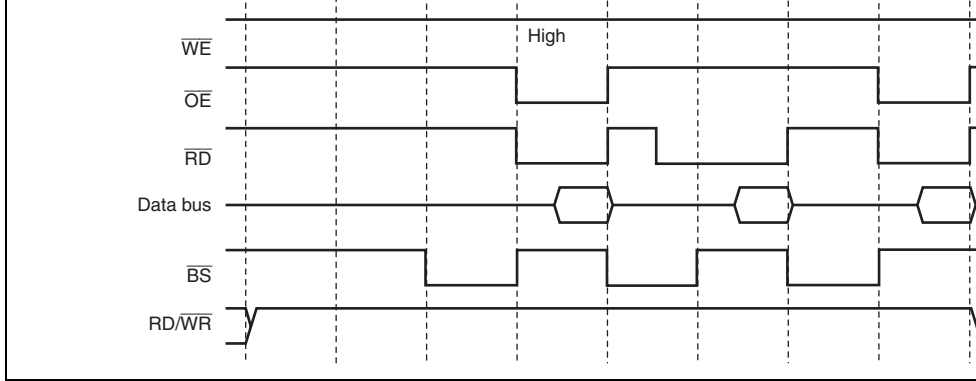


Figure 9.50 Timing Example of RAS Down Mode (RAST = 0, CAST = 0)

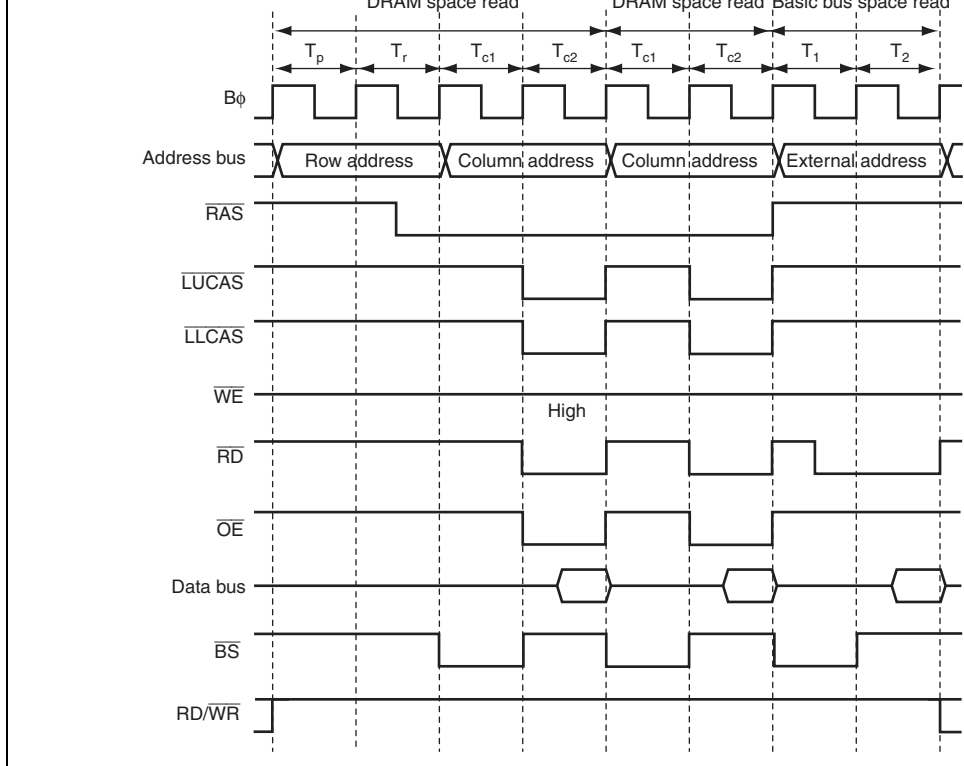


Figure 9.51 Timing Example of RAS Up Mode (RAST = 0, CAST = 0)

A CBR refresh cycle is performed when the value set in RTCOR matches the RTCNT value (compare match). RTCNT is an up-counter operated on the input clock specified by bits RTCK1 to RTCK0 in REFCR. RTCNT is initialized upon the compare match and restarts to count from H'00. Accordingly, a CBR refresh cycle is repeated at intervals specified by bits RTCK2 to RTCK0 in RTCOR. Set the bits so that the required refresh intervals of the DRAM must be satisfied.

Since setting bits RTCK2 to RTCK0 starts RTCNT to count up, set RTCNT and RTCOR before setting bits RTCK2 to RTCK0. When changing RTCNT and RTCOR, the counting operation should be halted. When changing bits RTCK2 to RTCK0, change them only after disabling external access and bus release by the EXDMAC, and if the write data buffer function is enabled, disabling the write data buffer function and reading the external space.

The external space cannot be accessed in CBR refresh mode.

Figure 9.52 shows RTCNT operation, figure 9.53 shows compare match timing, and figure 9.54 shows CBR refresh timing. Table 9.23 lists the pin states during a CBR refresh cycle.

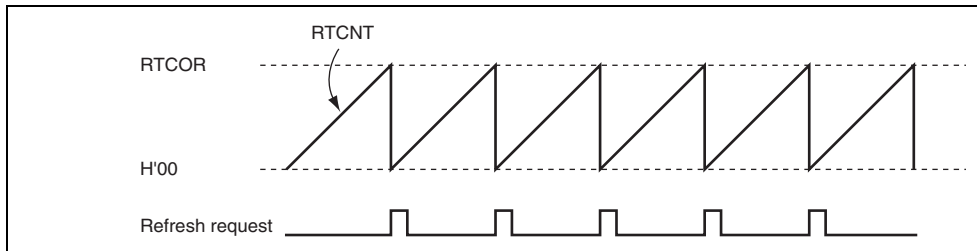


Figure 9.52 RTCNT Operation

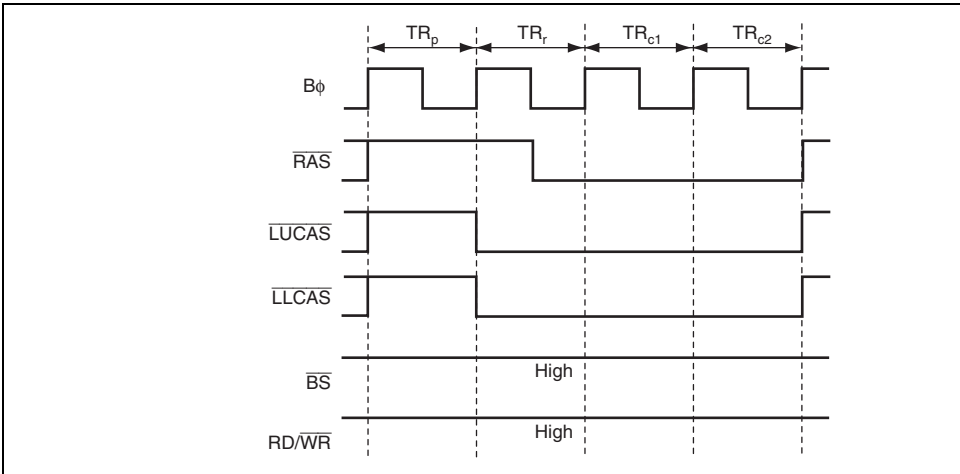


Figure 9.54 CBR Refresh Timing

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

The \overline{RAS} signal can be delayed for one to three clock cycles by setting bits RCW1 and RREFCR. The pulse width of the \overline{RAS} signal is changed by bits RLW2 to RLW0 in REFCR. 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

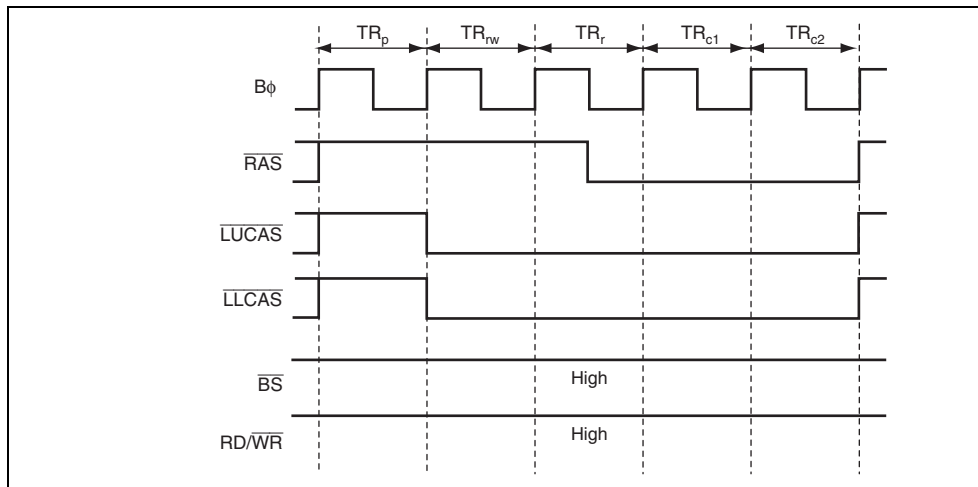


Figure 9.55 CBR Refresh Timing
(RCW1 = 0, RCW0 = 1, RLW2 = 0, RLW1 = 0, RLW0 = 0)

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

For details, see section 27.2.1, Standby Control Register (SBYCR).

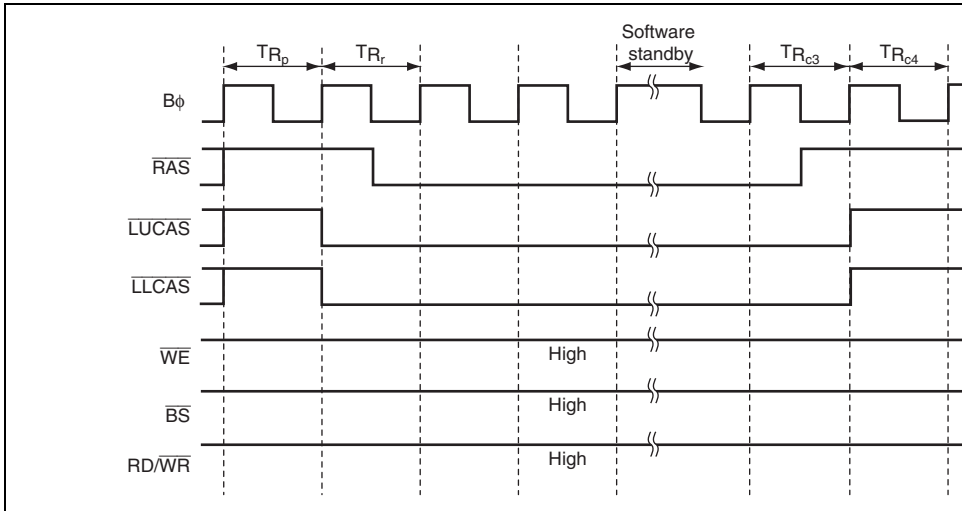


Figure 9.56 Self-Refresh Timing

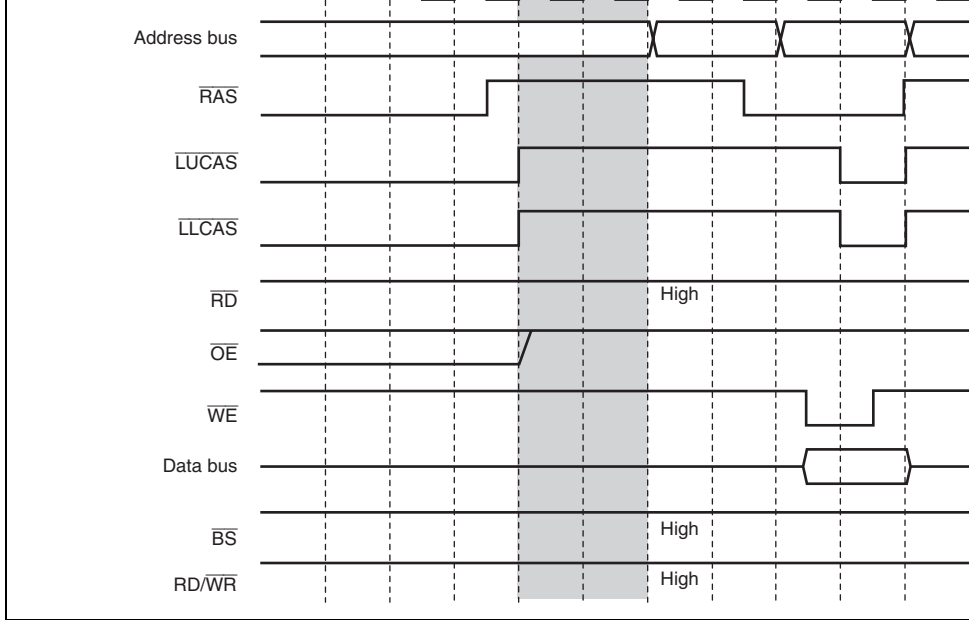


Figure 9.57 Timing Example when 1 Precharge Cycle Added

For details, see section 27.2.2, Module Stop Control Registers A and B (MSTPCRA and MSTPCRB).

9.10.13 DRAM Interface and Single Address Transfer by DMAC and EXDMAC

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

The assertion timing of the \overline{DACK} and \overline{EDACK} signal can be changed by bits DKC and BCR1.

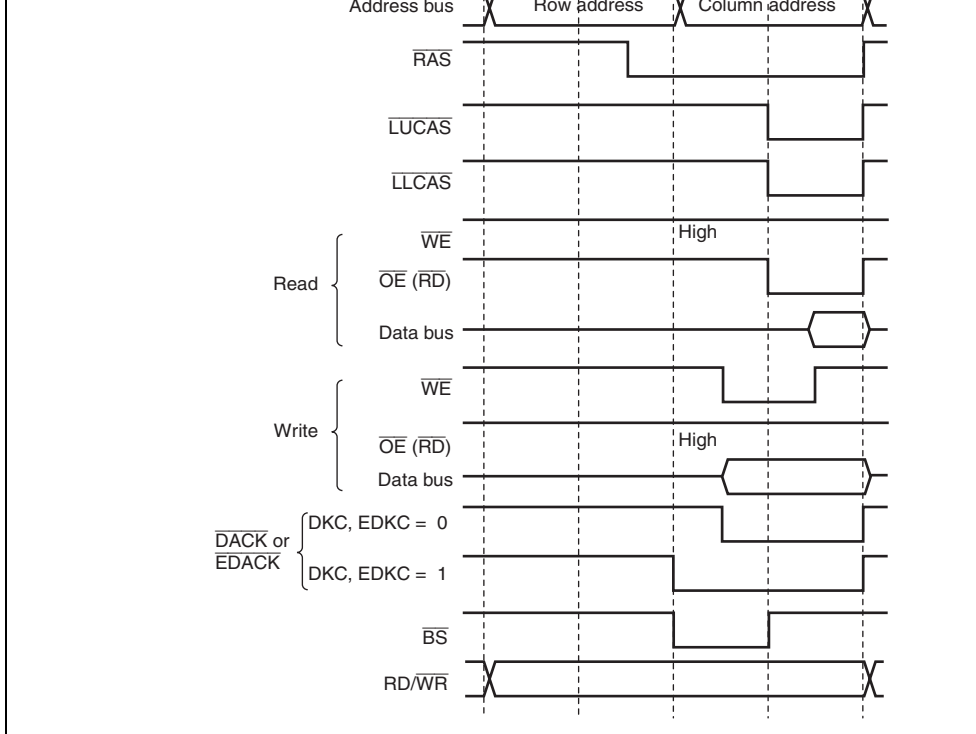


Figure 9.58 Output Timing Example of $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ when $\text{DDS} = 1$ or $\text{ED} = 1$ ($\text{RAST} = 0, \text{CAST} = 0$)

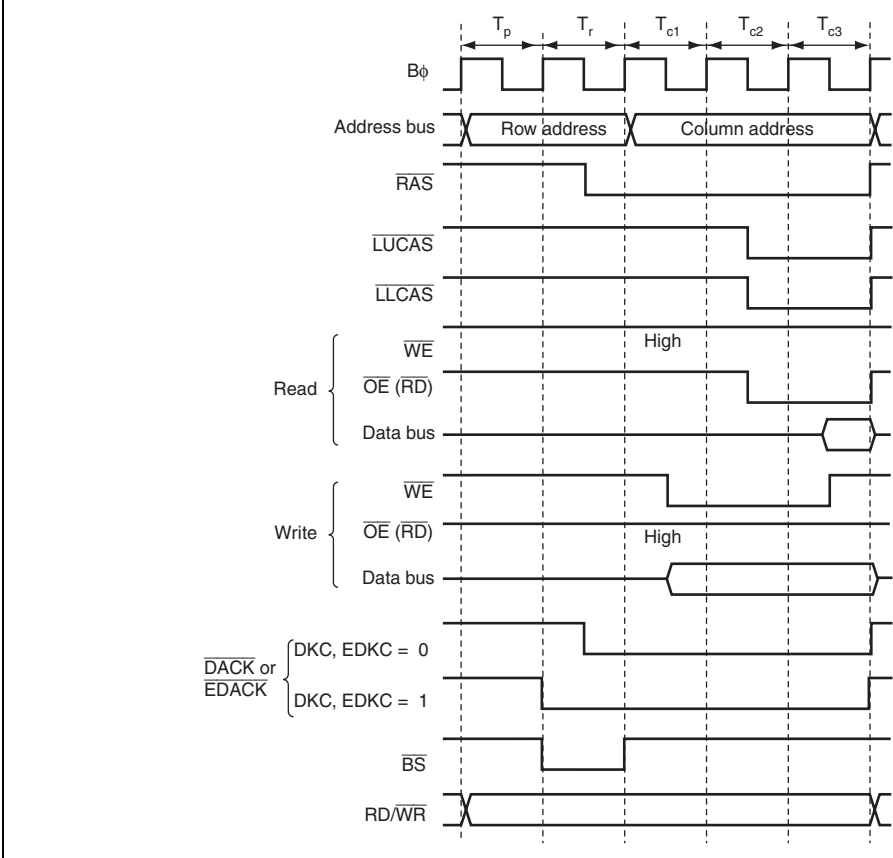


Figure 9.59 Output Timing Example of \overline{DACK} and \overline{EDACK} when $DDS = 0$ or E ($RAST = 0, CAST = 1$)

Table 9.24 lists the relationship among the DRAME and DTYPE bits and area 2 interface

In the SDRAM space, pins PB2, PB3, and PB4 are used as the $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and $\overline{\text{WE}}$ signals. 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 on the PFCR settings. The pin wait and program wait for the SDRAM space are not available. For PFCR settings, see section 13, I/O Ports.

An SDRAM command is designated by the combination of the $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and $\overline{\text{WE}}$ signals and 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 (SR), bank-precharge (PALL), bank active (ACTV), read (READ), write (WRIT), and mode register setting (MRS). Commands controlling a bank are not supported.

Table 9.24 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.

9.11.3 Data Bus

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

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

The data endian format can be selected by bit LE2 in ENDIANCR. For details on the address and alignment, see section 9.5.6, 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 SDRAM space is specified as area 2
$\overline{\text{CAS}}$	$\overline{\text{CAS}}$	Column address strobe	Output	Column address strobe when the SDRAM space is specified as area 2
$\overline{\text{WE}}$	$\overline{\text{WE}}$	Write enable	Output	Write enable signal for accessing SDRAM interface
$\overline{\text{OE/CKE}}$	CKE	Clock enable	Output	Clock enable signal when the SDRAM space is specified as area 2.
$\overline{\text{LLCAS}}/\text{DQMLU}$	DQMLU	Lower-upper data mask enable	Output	Upper data mask enable when the 16-bit SDRAM space is accessed
$\overline{\text{LLCAS}}/\text{DQMLL}$	DQMLL	Lower-lower data mask enable	Output	<ul style="list-style-type: none"> • Lower data mask enable when the 16-bit SDRAM space is accessed • Data mask enable when the SDRAM is accessed
A17 to A0	A17 to A0	Address pin	Output	Multiplexed row/column-address pin
D15 to D0	D15 to D0	Data pin	Input/output	Data input/output pin
PB7	SDRAM ϕ	Clock	Output	SDRAM clock
$\overline{\text{CS2}}$	$\overline{\text{CS}}$	Chip select	Output	Strobe signal indicating that SDRAM is selected

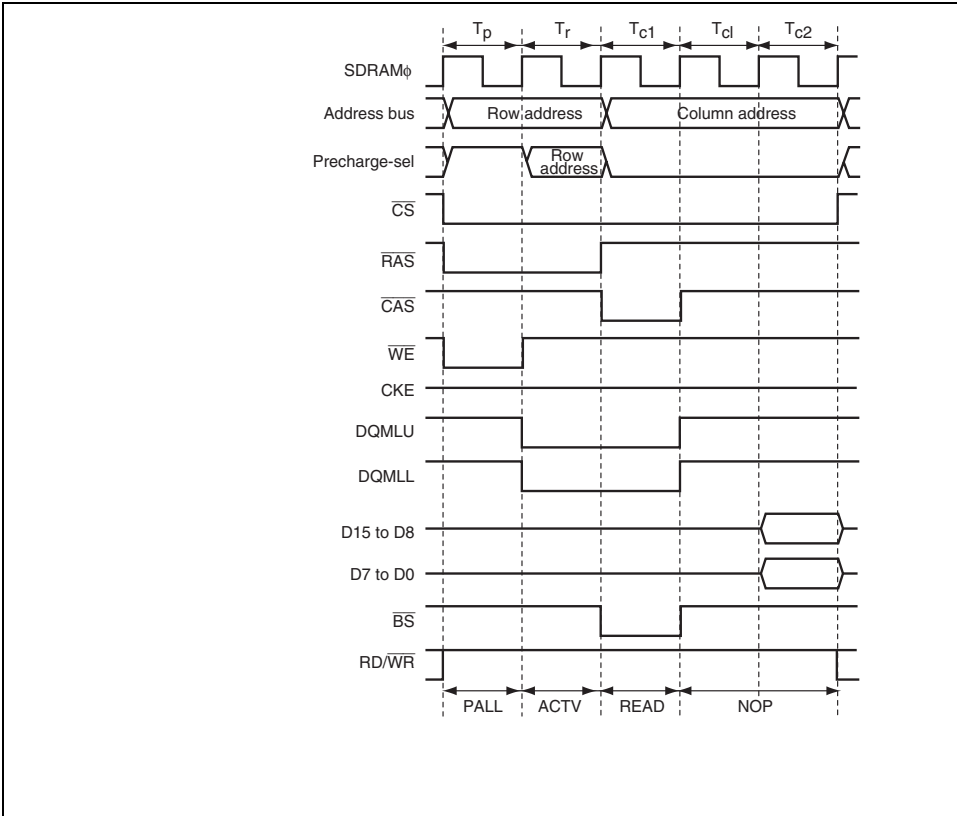


Figure 9.60 SDRAM Basic Read Access Timing (CAS Latency = 2)

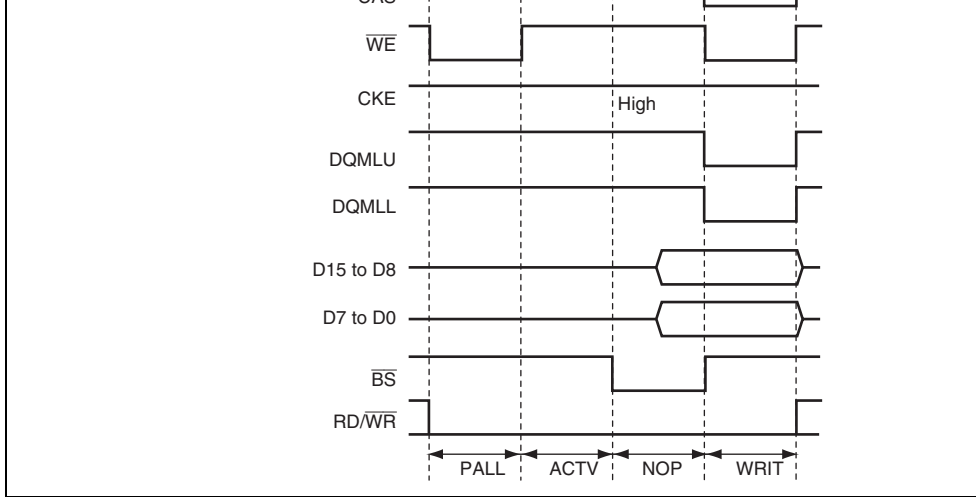


Figure 9.61 SDRAM Basic Write Access Timing

W21	W20	Description	Number of Latency (ns)
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

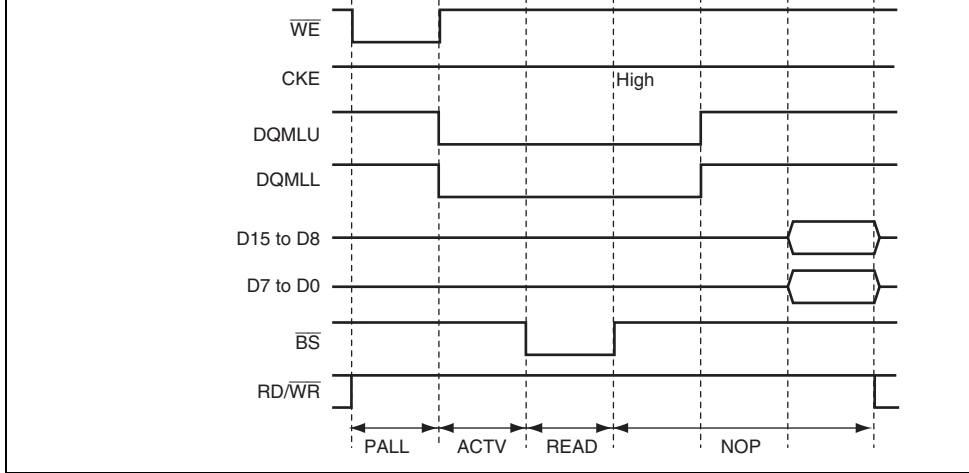


Figure 9.62 Timing Example of CAS Latency (CAS Latency = 3)

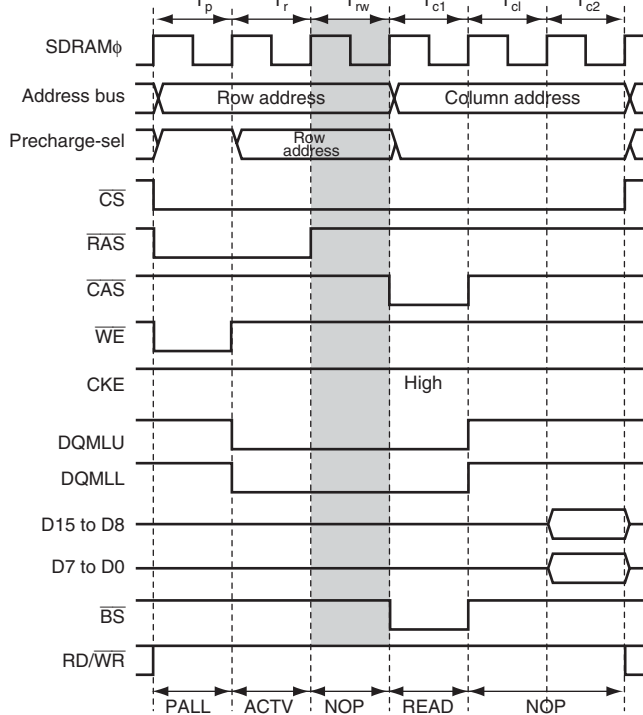


Figure 9.63 Read Timing Example of Row Address Output Retained for 1 Clock (RCD1 = 0, RCD0 = 1, CAS Latency = 2)

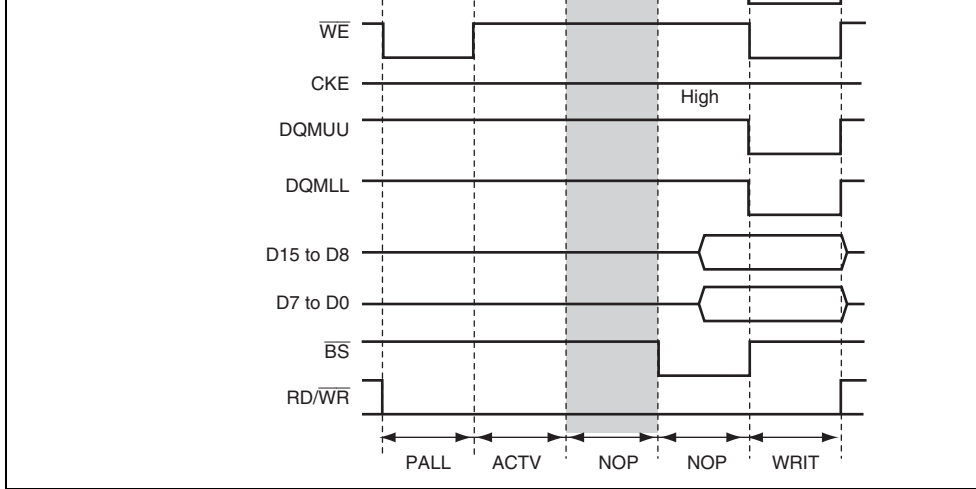


Figure 9.64 Write Timing Example of Row Address Output Retained for 1 Clock (RCD1 = 0, RCD0 = 1)

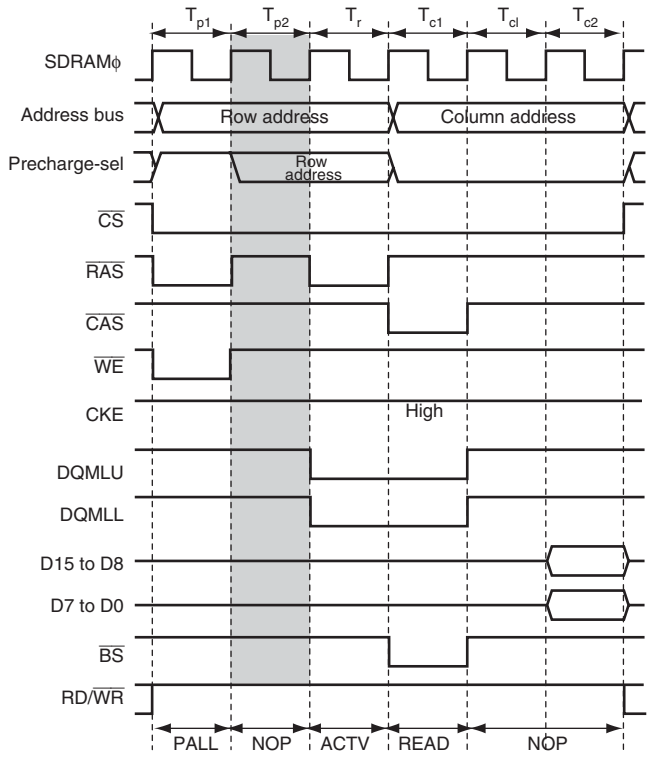


Figure 9.65 Read Timing Example of Two Precharge Cycles (TPC1 = 0, TPC0 = 1, CAS Latency = 2)

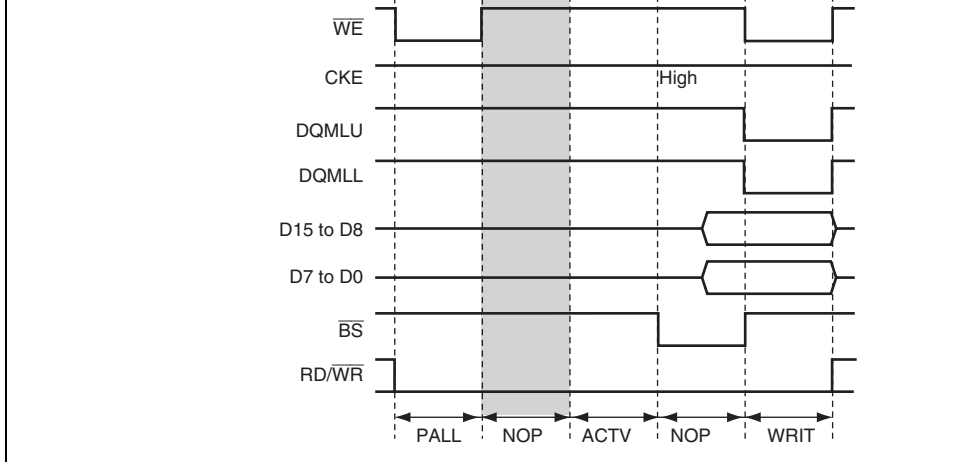


Figure 9.66 Write Timing Example of Two Precharge Cycles (TPC1 = 0, TPC0 = 1)

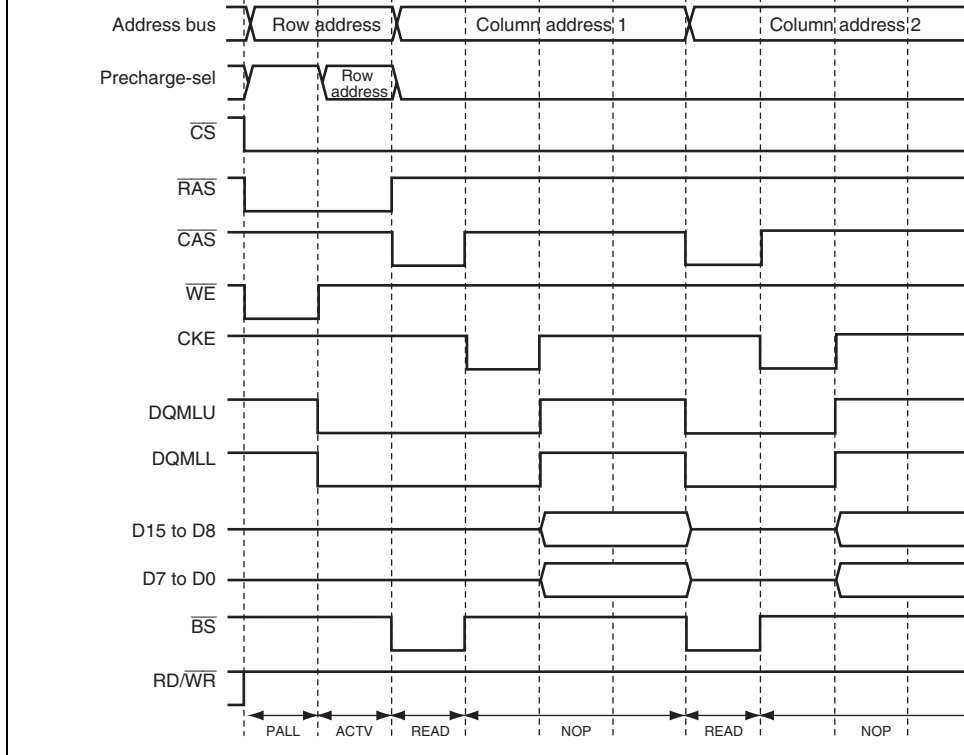


Figure 9.67 Read Timing Example when CKSPE = 1 (CAS Latency = 2)

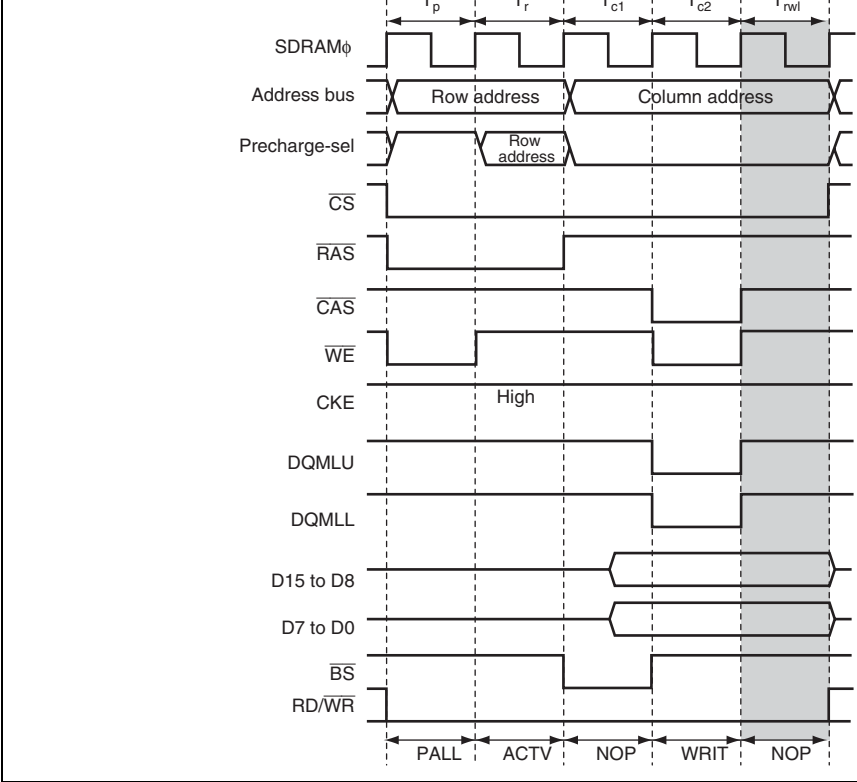


Figure 9.68 Write Timing Example when Write-Precharge Delay Cycle Inserted (TRWL = 1)

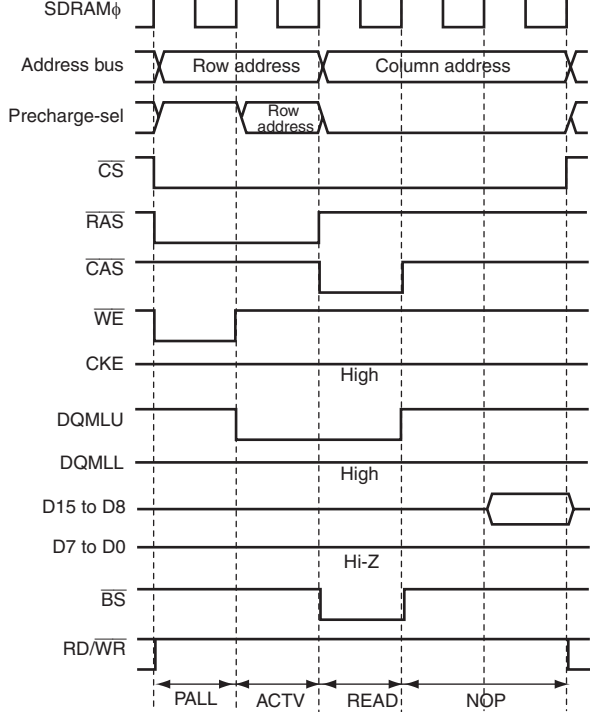


Figure 9.69 Control Timing Example of Byte Control by DQM in 16-Bit Access (Read Access with Lowest Bit of Address = B'0)

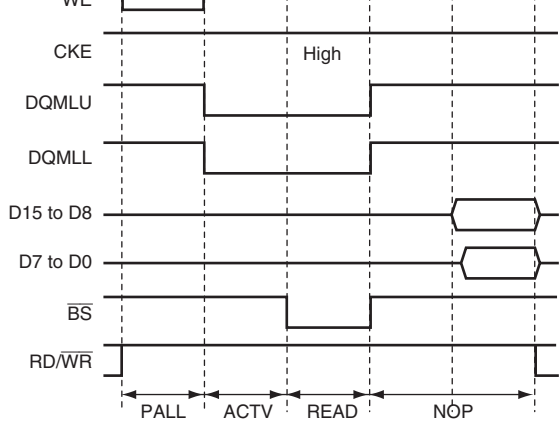


Figure 9.70 Control Timing Example of Word Control by DQM in 16-Bit Access (Read Access with Lowest Bit of Address = B'0, CAS Latency = 2)

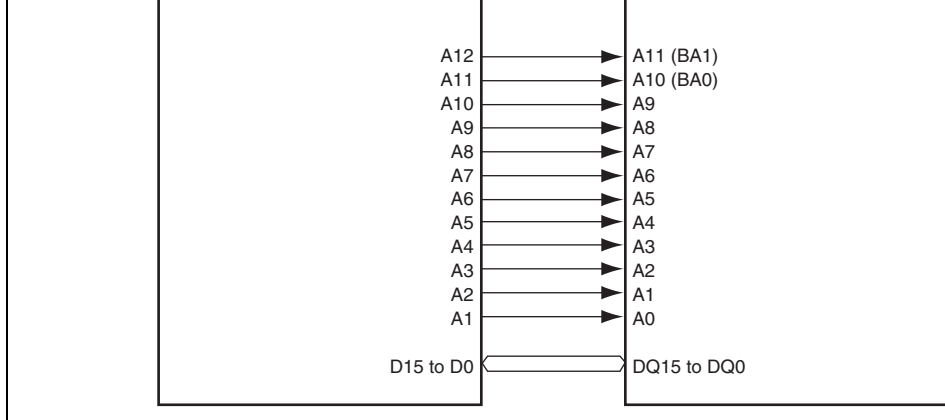


Figure 9.71 Connection Example of DQM Byte/Word Control

9.11.12 Fast-Page Access Operation

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

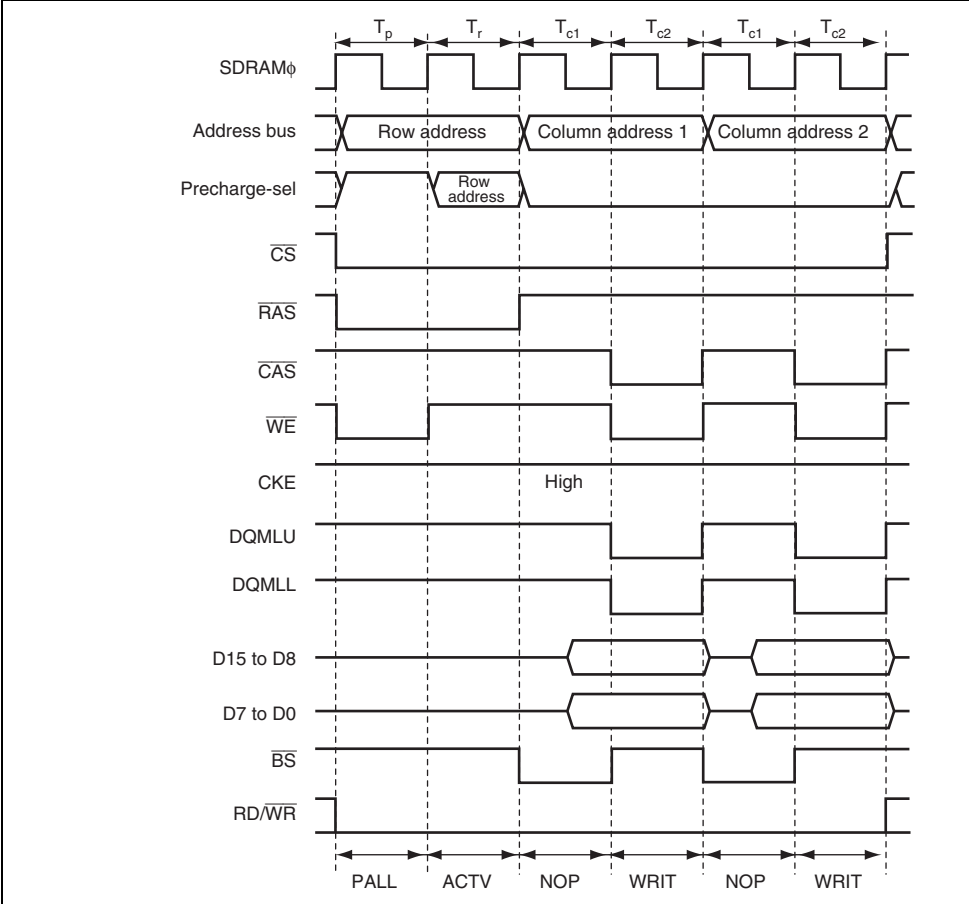
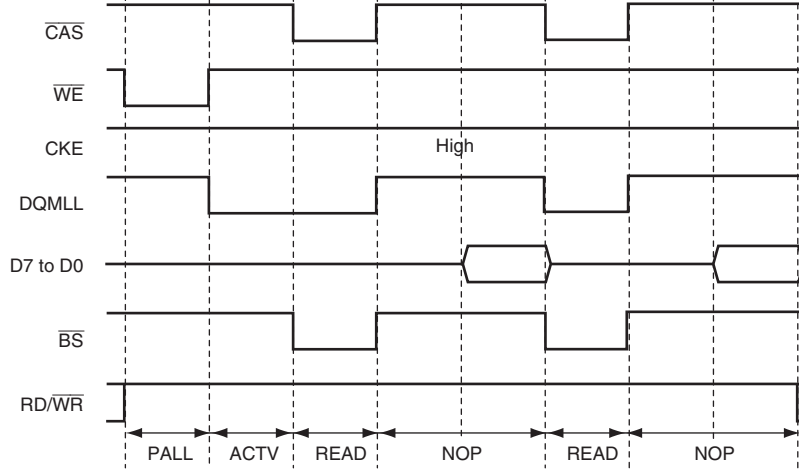


Figure 9.72 Longword Write Timing in 16-Bit Access Space (BE = 1, RCDM)



**Figure 9.73 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 SDRAMs are used, 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 is 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 SDRAM, see section 26, 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 is a fast-PALL command cycle. Only when the SDRAM space continues to be accessed, the fast-PALL mode access is performed.

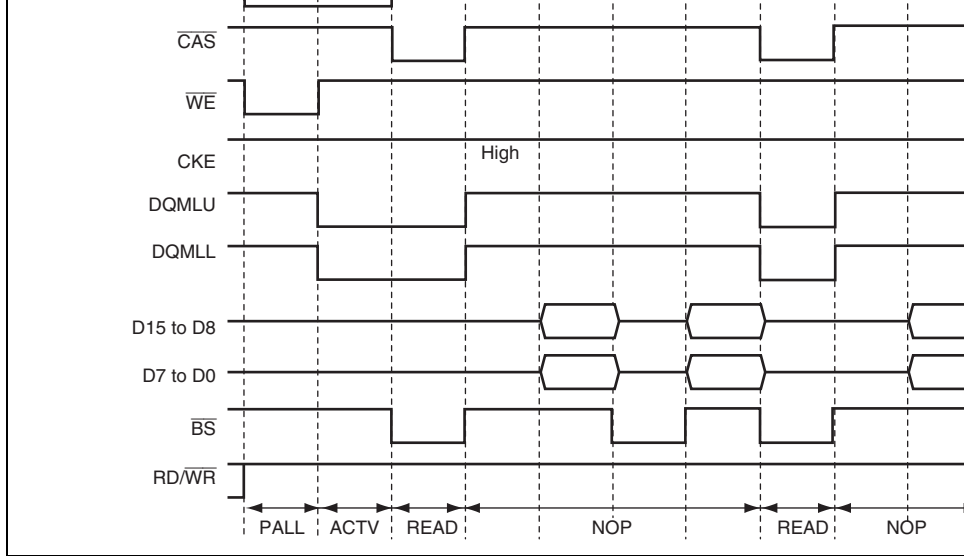


Figure 9.74 Timing Example of RAS Down Mode (BE = 1, RCDM = 1, CAS Latency = 3)

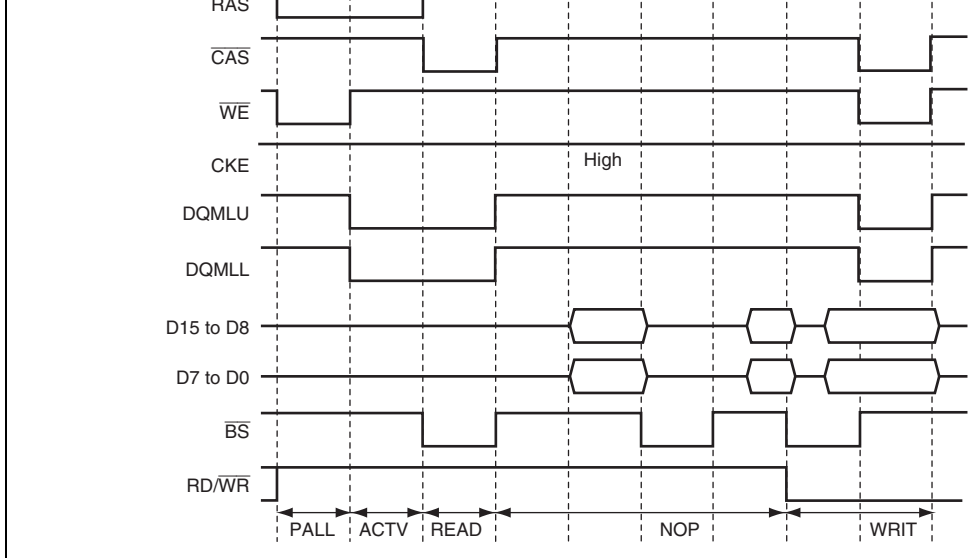


Figure 9.75 Timing Example of RAS Down Mode (BE = 1, RCDM = 1, CAS Later)

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

Since setting bits RTCK2 to RTCK0 starts RTCNT to count up, set RTCNT and RTCOR before setting bits RTCK2 to RTCK0. When changing RTCNT and RTCOR, the count operation must be halted. When changing bits RTCK2 to RTCK0, change them only after disabling the external access and external bus release by the EXDMAC, if the write data buffer function is in use. After disabling the write data buffer function and reading the external space.

The external space cannot be accessed during auto-refresh.

Figure 9.76 shows auto-refresh cycle timing. For details, see section 9.10.12, Refresh Cycle Timing.

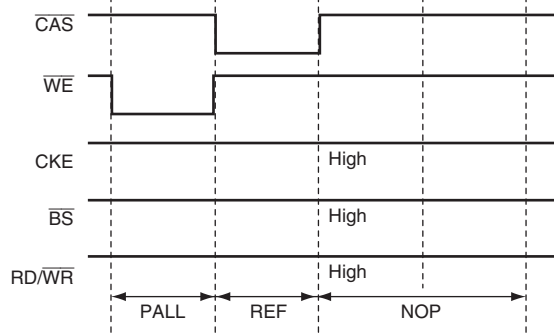


Figure 9.76 Auto-Refresh Operation

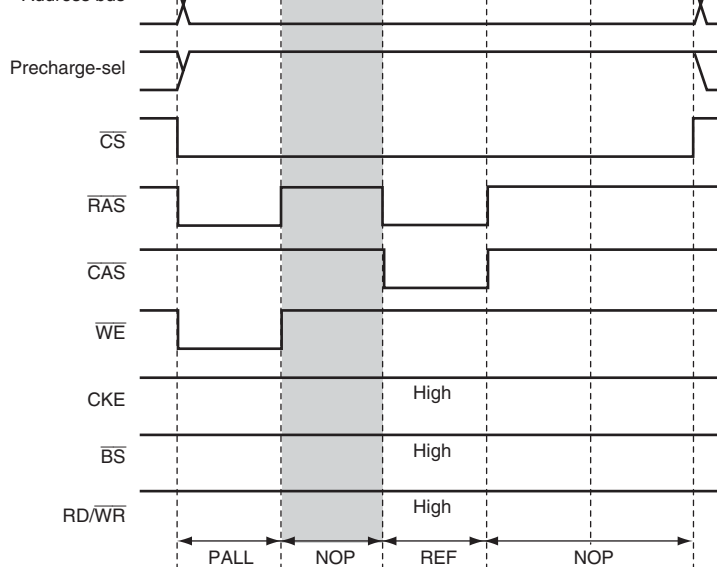


Figure 9.77 Auto-Refresh Timing (TPC1 = 0, TPC0 = 1)

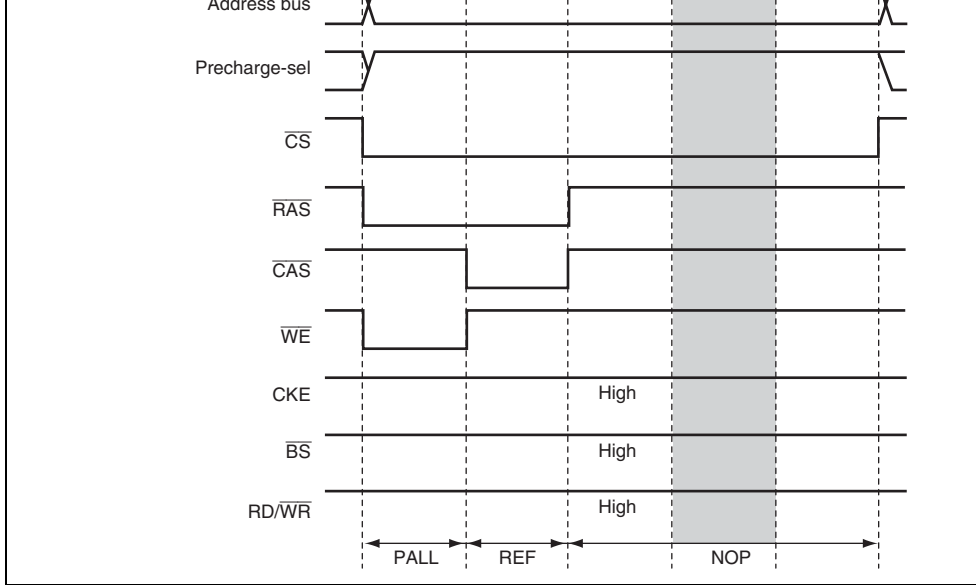


Figure 9.78 Auto-Refresh Timing (TPC1 = 0, TPC0 = 0, RLW2 = 0, RLW1 = 0, RLW0 = 0)

first performed and then the self refresh is entered.

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

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

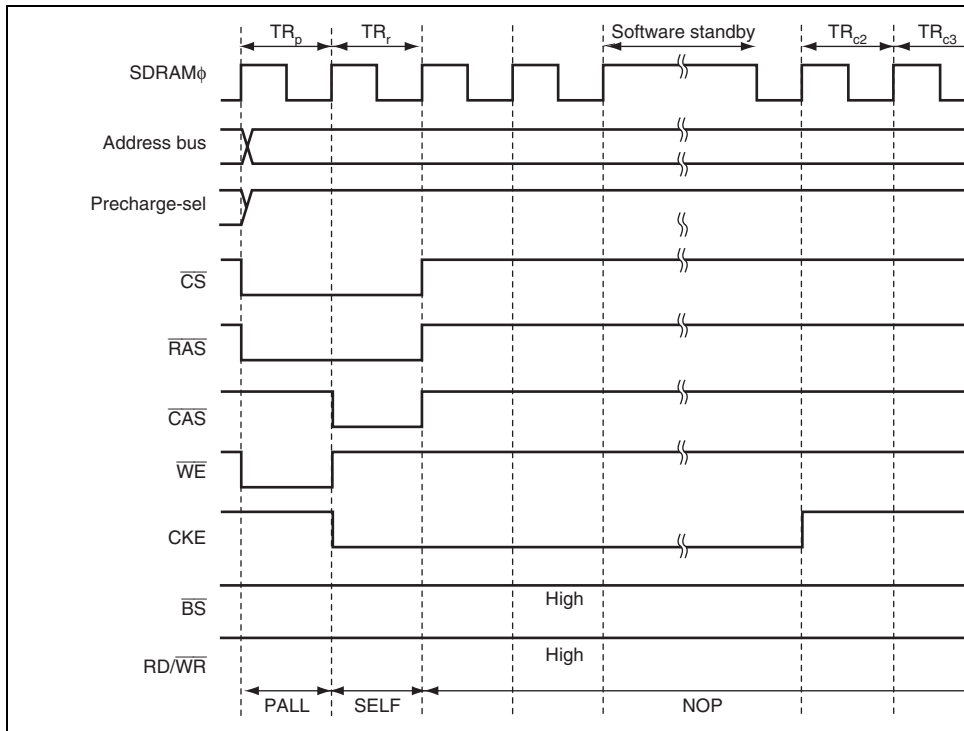


Figure 9.79 Self-Refresh Timing in software standby mode
 (TPC1 = 0, TPC0 = 0, RCW1 = 0, RCW0 = 0, RLW2 = 0, RLW1 = 0, RLW0 = 0)

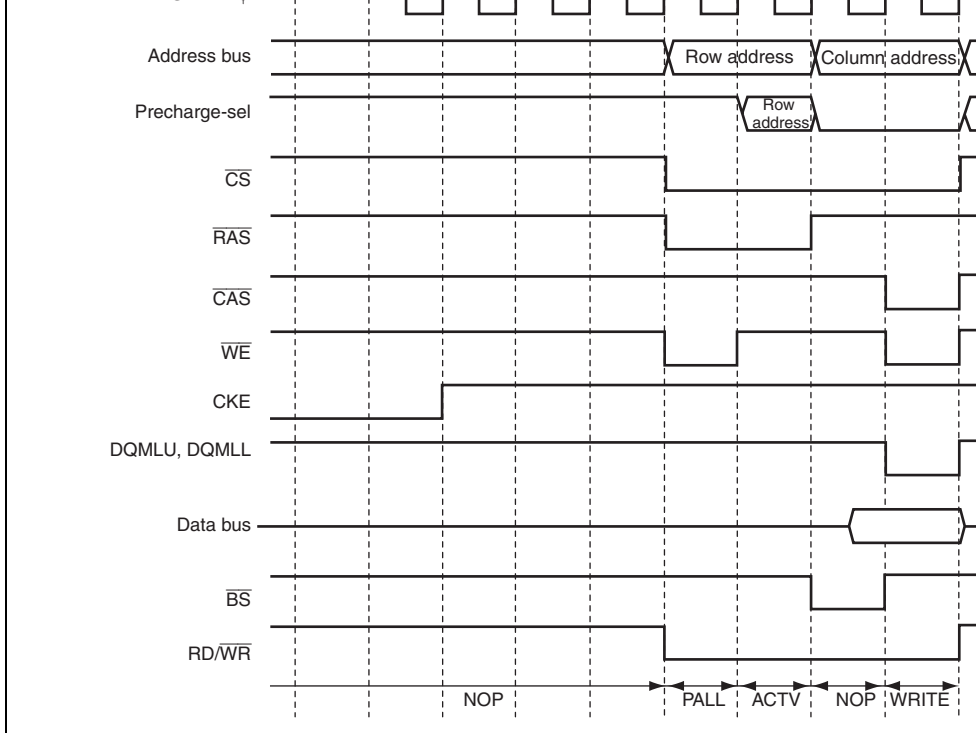


Figure 9.80 Timing Example when 1 Precharge Cycle Added in the Software Standby Mode (TPCS2 to TPCS0 = H'1, TPC1 = 0, TPC0 = 0)

1. In PBDDR/PBDR, set PB1 ($\overline{CS2}$) as a high-level output and PB5(CKE) as a low-level output.

Since the setting of the IOKEEP bit ensures retention of pin state at this time, the state of high-level output on $\overline{CS2}$ and low-level output on is retained.

2. Set the PSTOP0 bit in SCKCR to 1 and SDRAM ϕ as a high level output. Since the setting of the IOKEEP bit continues to ensure retention of pin state, the existing state of low-level output on SDRAM ϕ is retained.
3. Clear the IOKEEP bit in DPSBYCR.

This releases pin states from retention due to the setting of the IOKEEP bit, but the state of pins $\overline{CS2}$, CKE, and SDRAM ϕ as set in steps 1 and 2 do not change.

4. In the synchronous DRAM-related control registers that were initialized by the initialization sequence, reset that accompanied the transition to deep software standby mode, remake the settings to enable the synchronous DRAM interface. At this time, do not make settings in RTCNT, and RTCOR.

Once the synchronous DRAM interface has been enabled, the state of the CKE pin changes from low-level output to high-level output.

5. Restart output of the SDRAM ϕ clock signal by clearing the PSTOP0 bit in SCKCR. SDRAM ϕ restarts supply of SDRAM ϕ to the synchronous DRAM.
6. Set REFCR, RTCNT, and RTCOR and enable refreshing.
As the state of the CKE pin has been changed in the step 4, adjust the time between the state of change of the CKE pin and the next cycle of auto-refreshing in this procedure within the stipulated refreshing interval of the synchronous DRAM.
7. Resume access to the synchronous DRAM.
Pre-charging time after the termination of self-refresh will be secured by the timing setting in step 6.

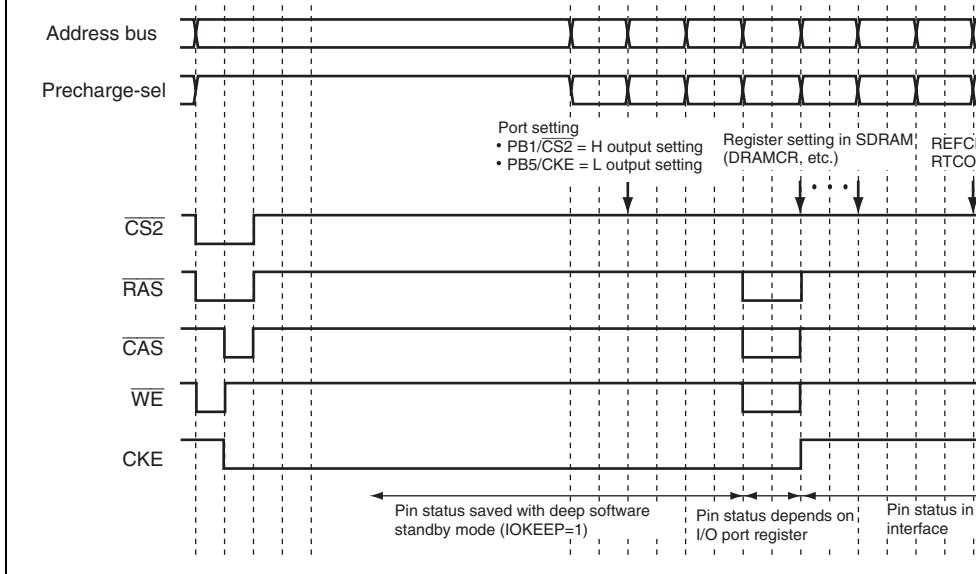


Figure 9.81 Self-Refresh Timing in Deep Software Standby Mode
 (TPC1 = 0, TPC0 = 0, RCW1 = 0, RCW0 = 0, RLW2 = 0, RLW1 = 0, RLW0 = 0)

For details, see section 27.2.2, Module Stop Control Registers A and B (MSTPCRA and MSTPCRB).

9.11.14 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 this, the SDRAM space in bytes.

When the value to be set in the SDRAM mode register is x , write to the following memory 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 issued.

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 the SDRAM mode register must be consistent with that in the bus controller.

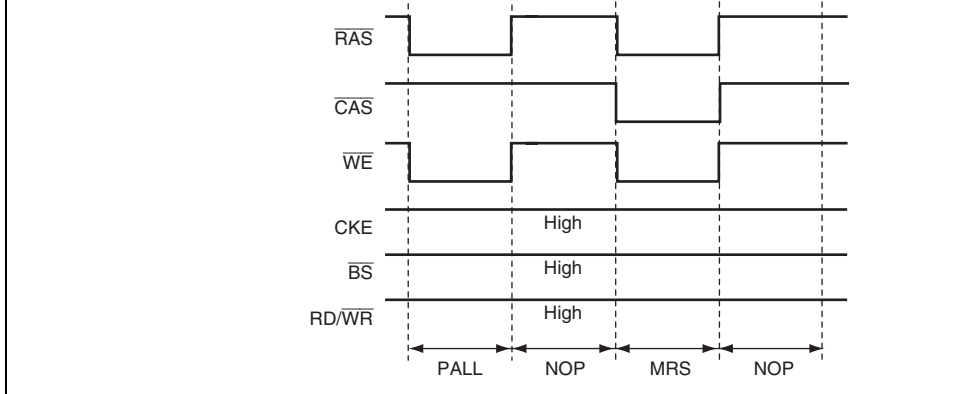


Figure 9.82 Timing of Setting SDRAM Mode Register

9.11.15 SDRAM Interface and Single Address Transfer by DMAC and EXDMAC

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

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

The output timing of the $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ signals can be independently set by the bits DKC and CKSPE in SDCR and bit DKC and EDKC in BCR1 regardless of the setting of bits DDS and EDDS.

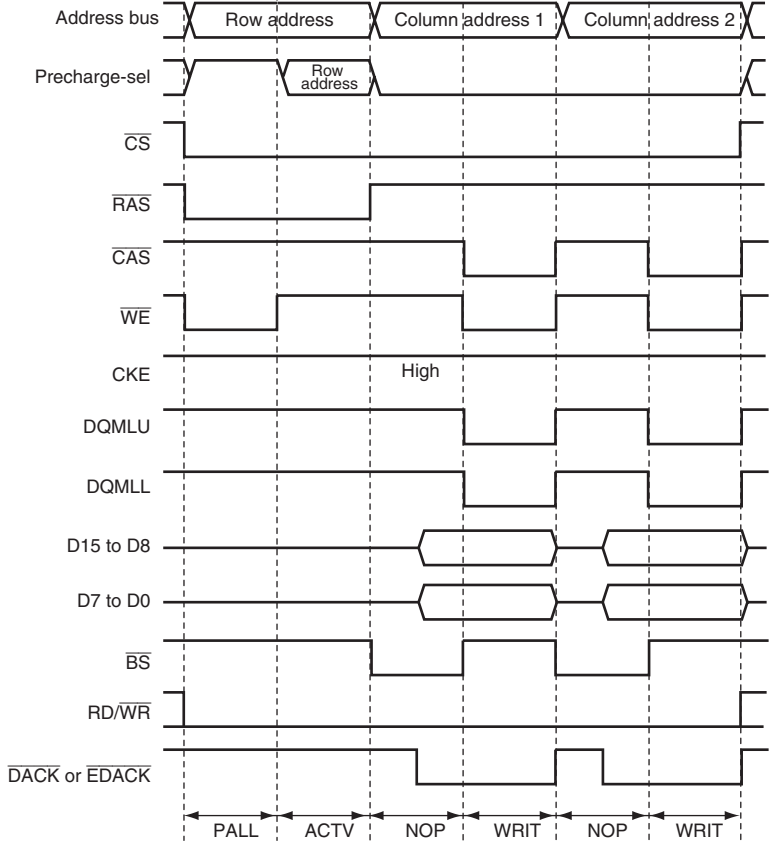


Figure 9.83 Output Timing Example of $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ when $\text{DDS} = 1$ or E (Write)

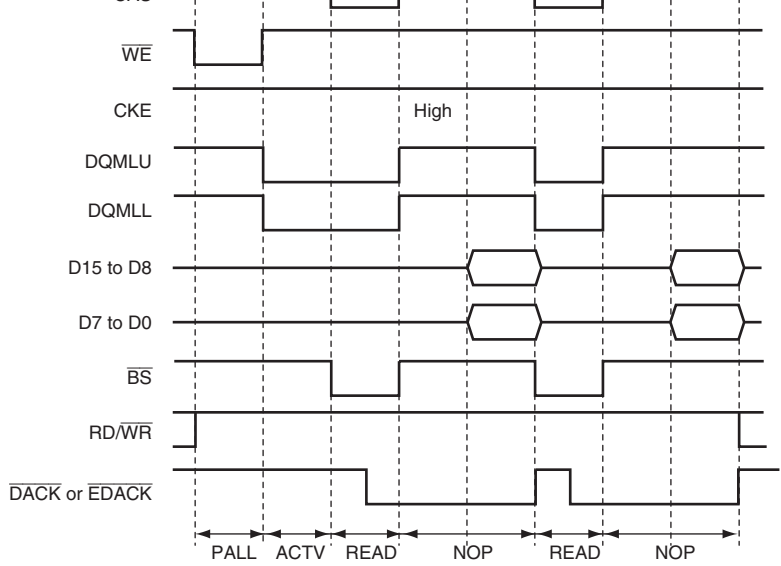


Figure 9.84 Output Timing Example of $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ when $\text{DDS} = 1$ or ED (Read, CAS Latency = 2)

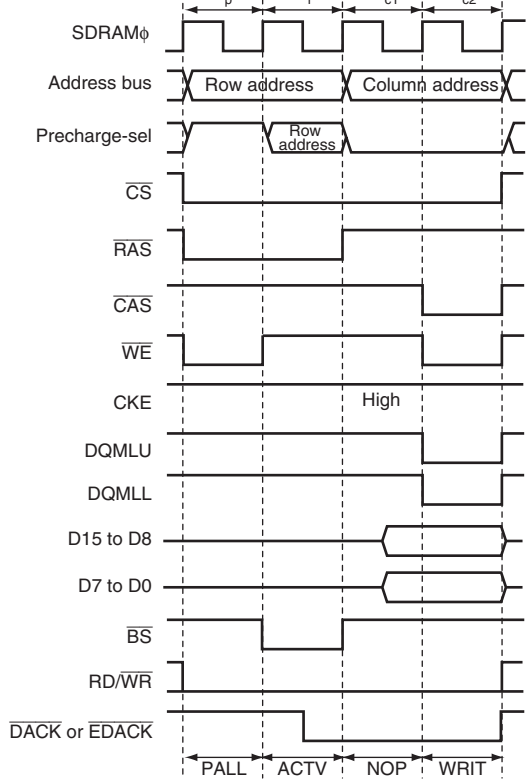


Figure 9.85 Output Timing Example of \overline{DACK} and \overline{EDACK} when $DDS = 0$ or E (Write)

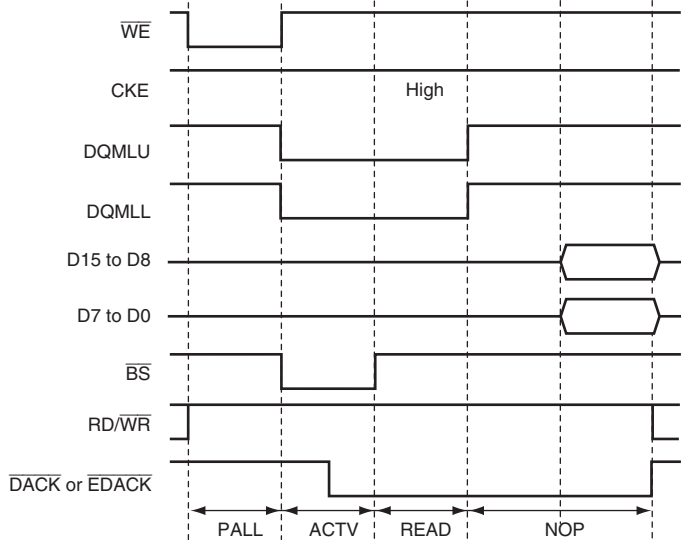


Figure 9.86 Output Timing Example of $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ when $\text{DDS} = 0$ or ED (Read, CAS Latency = 2)

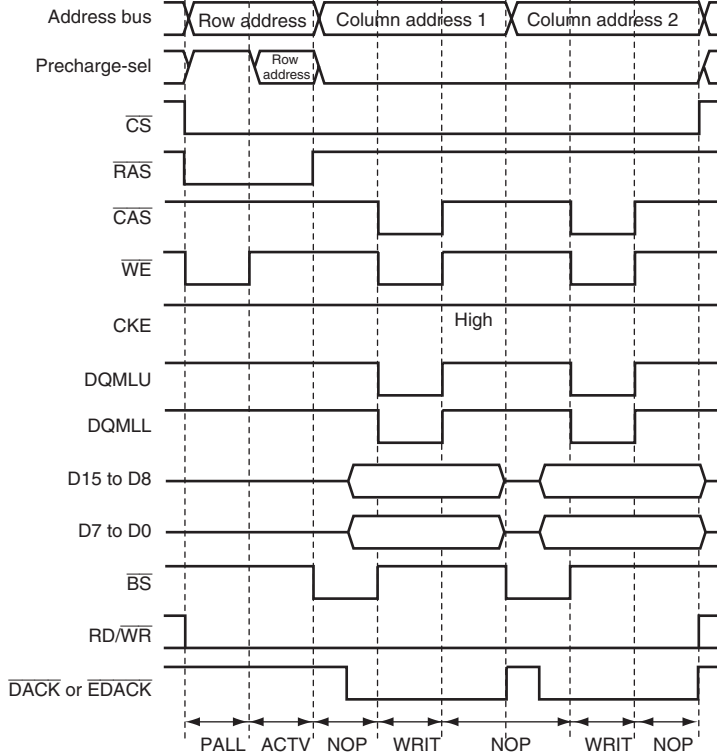


Figure 9.87 Output Timing Example of $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ when $\text{TRWL} = 1$ (

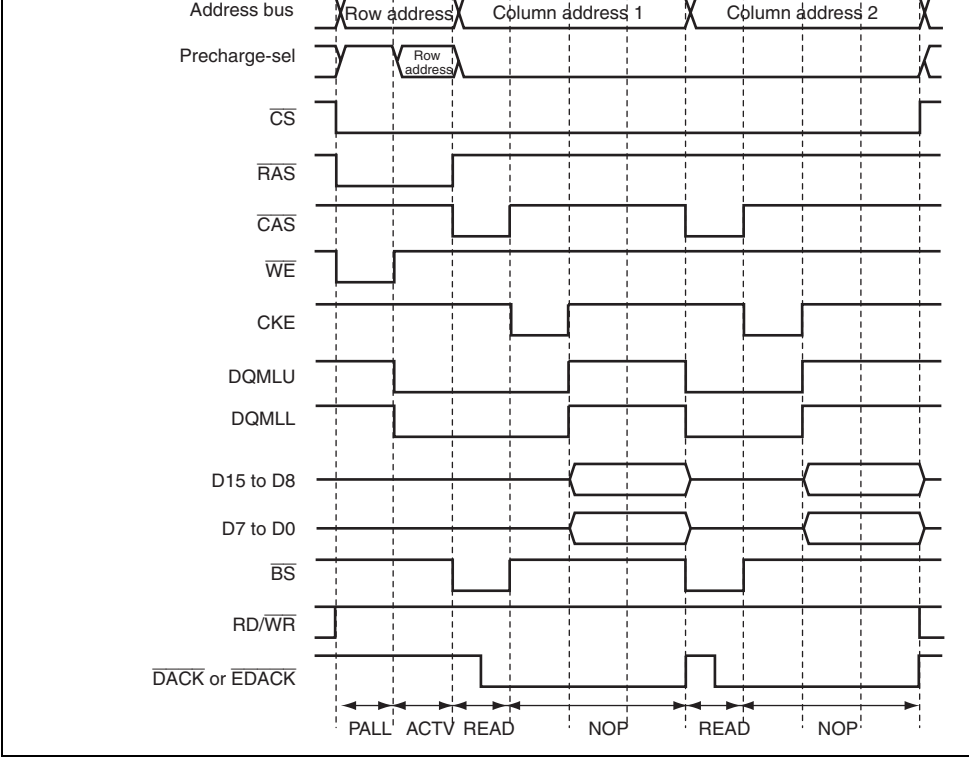


Figure 9.88 Output Timing Example of $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ when $\text{CKSPE} = \text{Read}$, CAS Latency = 2

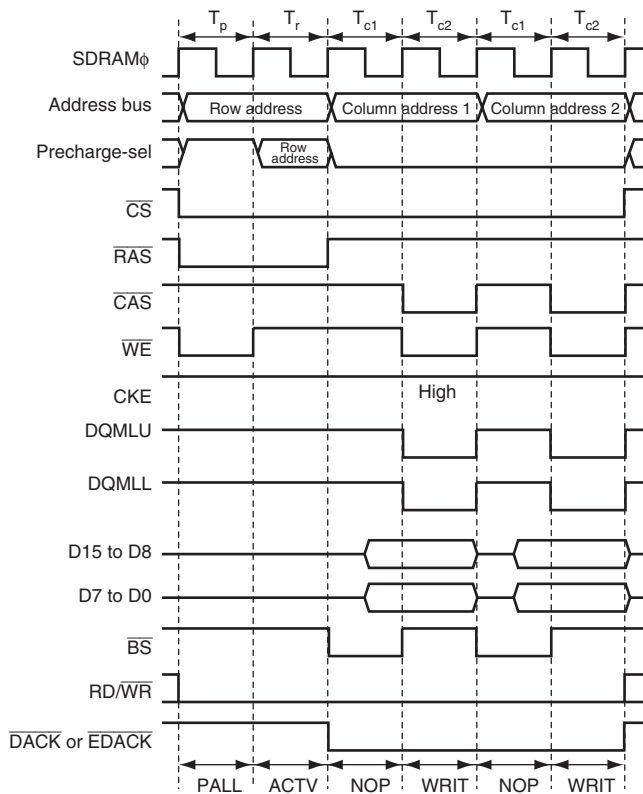


Figure 9.89 Output Timing Example of $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ when $\text{DKC} = 1$ or $\text{EDKC} = 1$ and $\text{DDS} = 1$ or $\text{EDDS} = 1$ (Write)

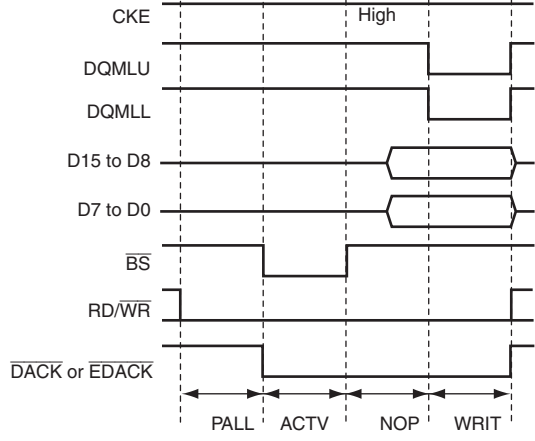


Figure 9.90 Output Timing Example of \overline{DACK} and \overline{EDACK} when $DKC = 1$ or $EDKC = 1$ and $DDS = 0$ or $EDDS = 0$ (Write)

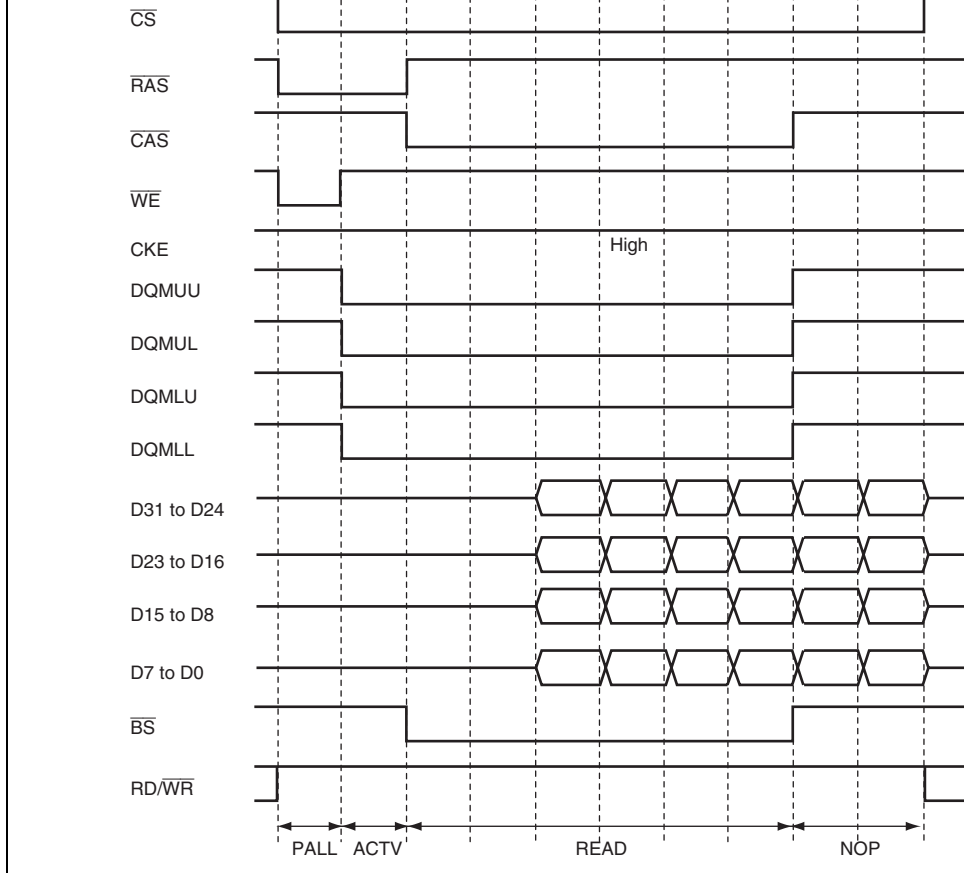
9.11.16 EXDMAC Cluster Transfer

Using an EXDMAC cluster transfer mode, data can be read from or written to consecutively. For details, see section 11, EXDMA Controller (EXDMAC).

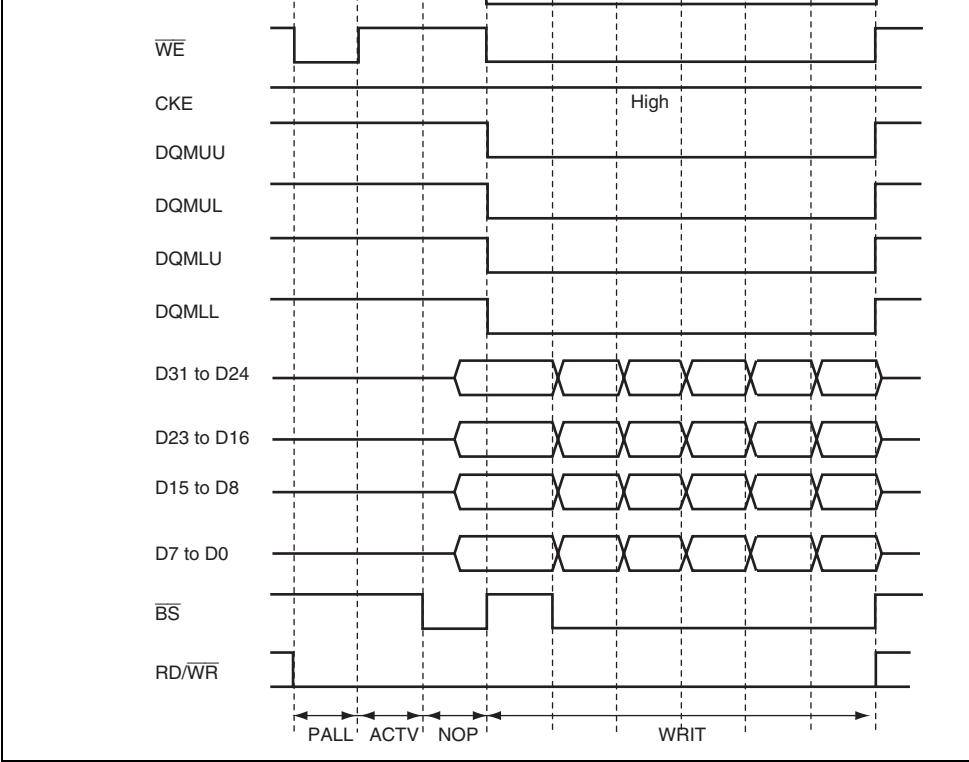
Figures 9.91 and 9.92 show a read/write timing using a cluster transfer.

For 1-cycle read or write, set the BE bit in DRAMCR to 1, clear the TRWL bit in SDCR, and set the CAS latency to 2. During a read cycle, the clock suspend mode cannot be used.

Do not change the bus controller register settings during a cluster transfer.



**Figure 9.91 Word-Size 6-Word Cluster Transfer
(Read, BE = 1, EDDS = 1, CAS Latency = 2)**



**Figure 9.92 Word-Size 6-Word Cluster Transfer
(Write, BE = 1, EDDS = 1)**

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 or EXDMAC* single access transfer (write 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 data of a 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 number of idle cycles can be selected from setting A specified by bits IDLCA1 and IDLCA0 in IDLCR, and setting B specified by bits IDLCB1 and IDLCB0 in IDLCR: Setting A can be selected from one to four cycles, and setting B can be selected from one or two to four cycles. Setting A or B can be specified for each area by setting bits IDLSEL7 to IDLSEL0 in IDLCR. Note that bits IDLSEL7 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 to insert 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 conditions shown above.

Table 9.28 shows the correspondence between conditions 1 to 4 and number of idle cycles to be inserted for each area. Table 9.29 shows the correspondence between the number of idle cycles to be inserted specified by settings A and B, and number of cycles to be inserted.

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

			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 9.29 Number of Idle Cycles Inserted

Bit Settings				
A		B		Number of Cycles
IDLCA1	IDLCA0	IDLCB1	IDLCB0	
—	—	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.

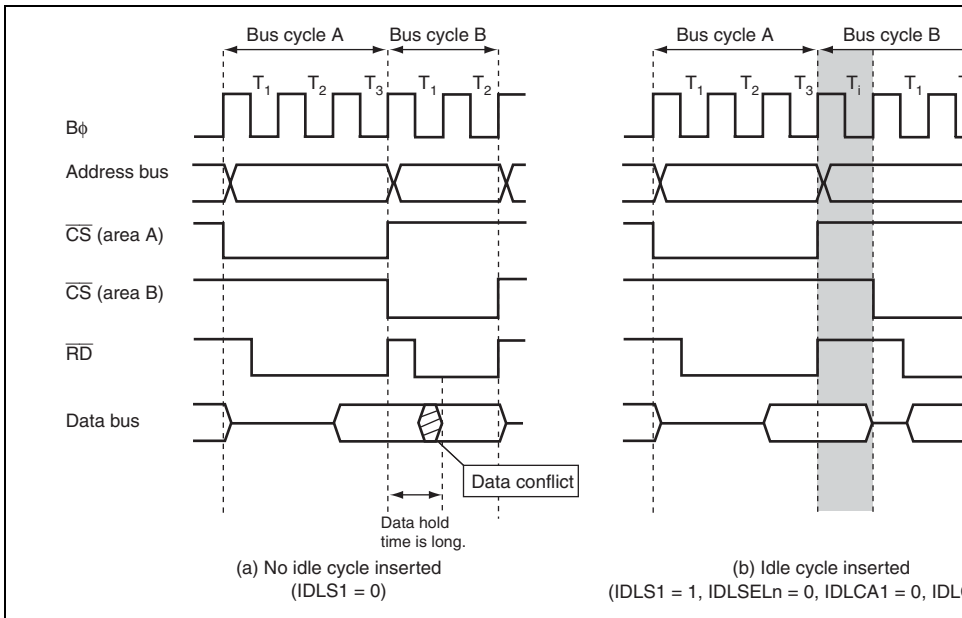


Figure 9.93 Example of Idle Cycle Operation (Consecutive Reads in Different

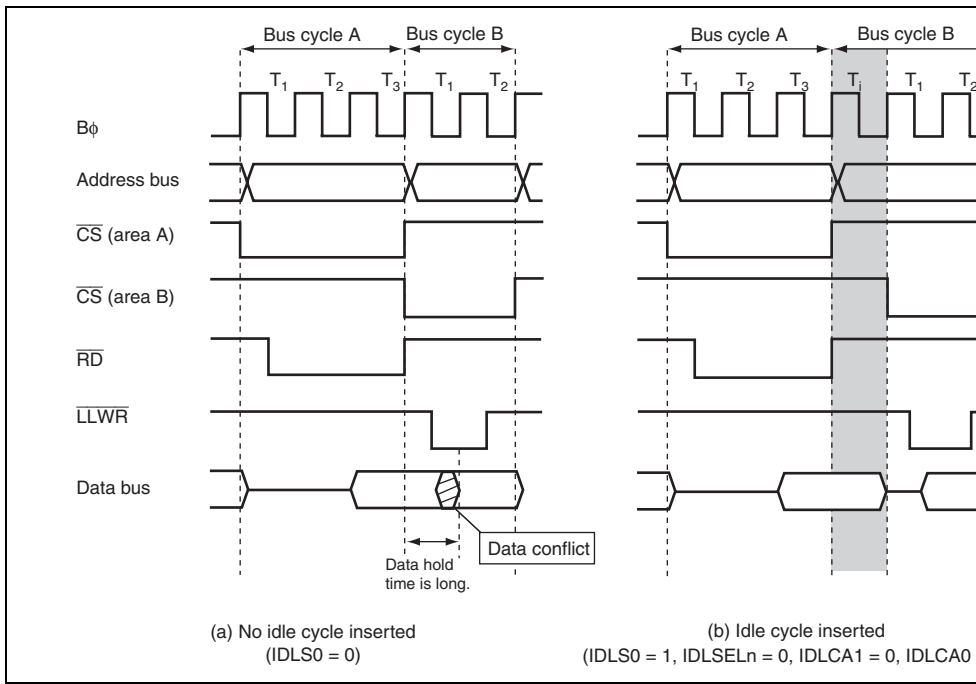


Figure 9.94 Example of Idle Cycle Operation (Write after Read)

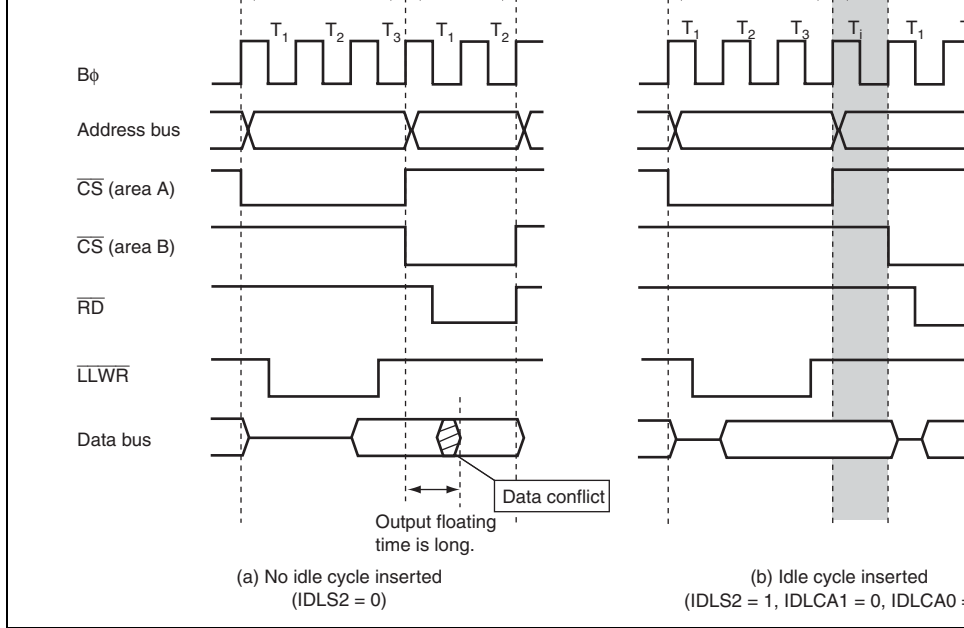


Figure 9.95 Example of Idle Cycle Operation (Read after Write)

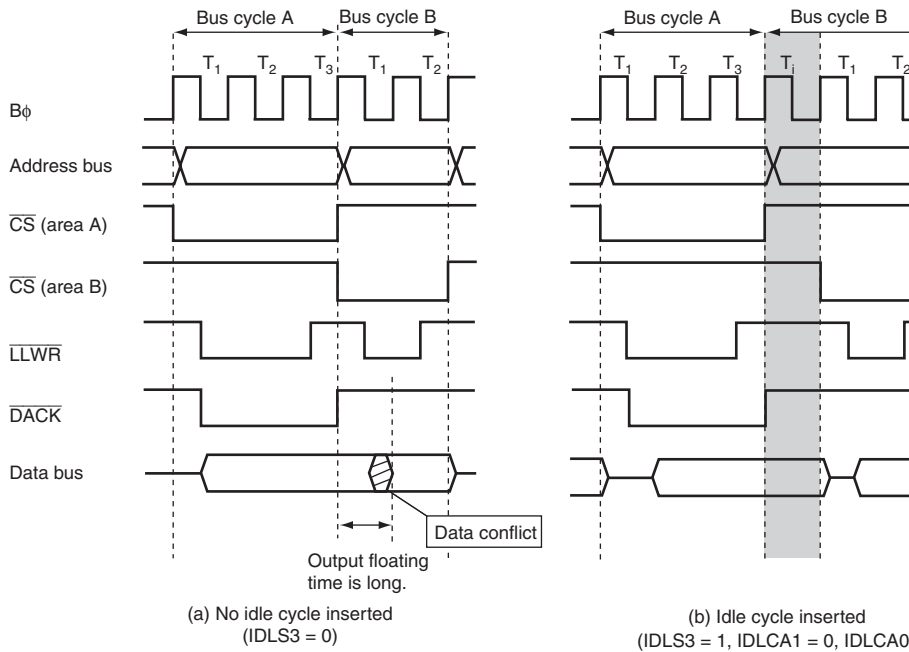


Figure 9.96 Example of Idle Cycle Operation (Write after Single Address Transfer)

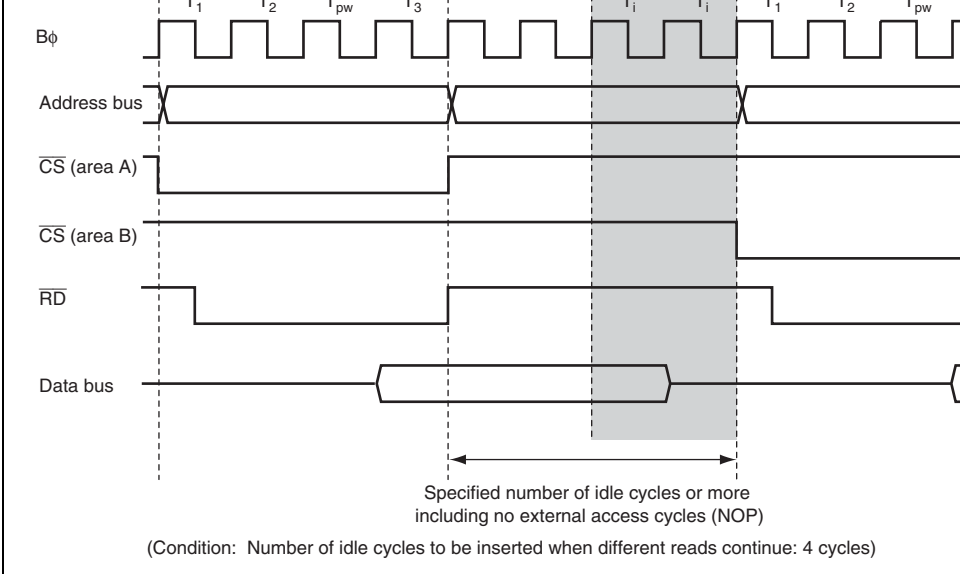


Figure 9.97 Idle Cycle Insertion Example

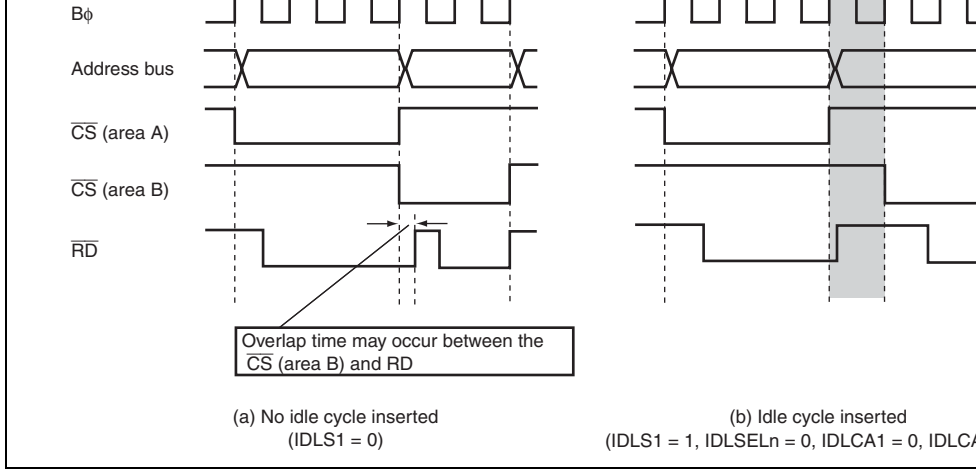


Figure 9.98 Relationship between Chip Select (\overline{CS}) and Read (\overline{RD})

While the SDRAM space is accessed in a full access, the CS2 signal is driven low even cycle.

The idle cycle insertion is enabled even in a fast-page access in RAS down mode. The number of idle cycles is inserted. Figure 9.101 shows a timing example of the idle cycle in RAS down mode.

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

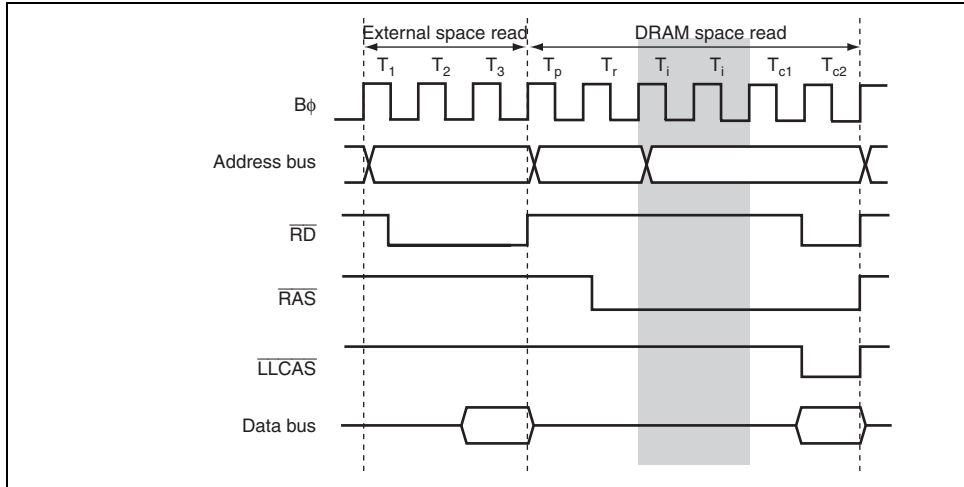


Figure 9.99 Example of DRAM Full Access after External Read (CAST = 0)

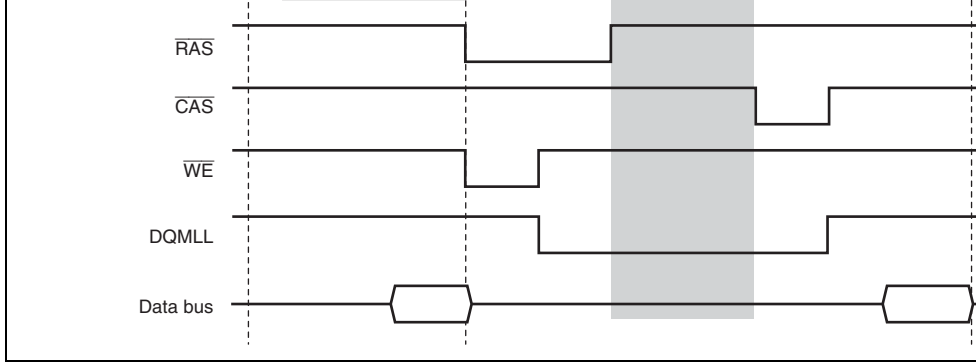


Figure 9.100 Example of SDRAM Full Access after External Read (CAS Latency)

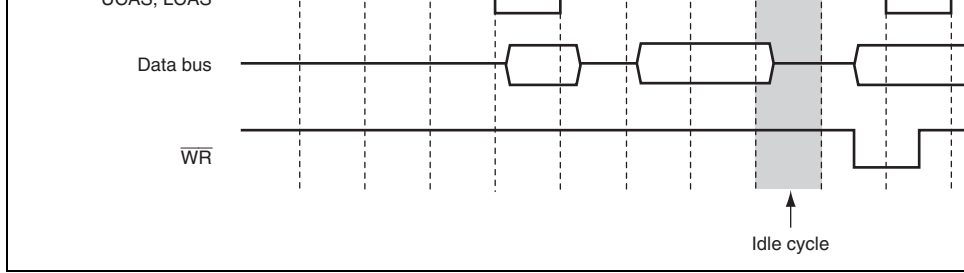


Figure 9.101 Example of Idle Cycles in RAS Down Mode (Write after Read)

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

$\overline{\text{LUCAS}}, \overline{\text{LLCAS}}^*$	High
$\overline{\text{DQMLU}}, \overline{\text{DQMLL}}^{*5}$	High ^{*2}
$\overline{\text{AS}}$	High
$\overline{\text{RD}}$	High
$\overline{\text{BS}}$	High
$\overline{\text{RD/WR}}$	High ^{*3}
$\overline{\text{AH}}$	low
$\overline{\text{LHWR}}, \overline{\text{LLWR}}$	High
$\overline{\text{LUB}}, \overline{\text{LLB}}$	High
$\overline{\text{CKE}}^{*5}$	High
$\overline{\text{OE}}^{*5}$	High
$\overline{\text{RAS}}^{*5}$	High/Low ^{*4}
$\overline{\text{CAS}}^{*5}$	High
$\overline{\text{WE}}^{*5}$	High
$\overline{\text{DACKn}}$ (n = 3 to 0)	High
$\overline{\text{EDACKn}}$ (n = 3 to 0) ^{*5}	High

- Notes:
1. Low when accessing the SDRAM^{*5} in full access cycle
 2. Low when reading the SDRAM^{*5} in full access cycle
 3. Low when accessing or writing to the DRAM/SDRAM^{*5} in full access cycle
 4. The pin state varies depending on the DRAM space access/ area access other DRAM space, or RAS up mode/RAS down mode. For details, see figures 9.9 and 9.100.
 5. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

9.13.1 Operation

In external extended mode, when the BRLE bit in BCR1 is set to 1, and the ICR bit for the corresponding pin is set to 1, the bus can be released to the external. Driving the $\overline{\text{BREQ}}$ pin issues an external bus request to this LSI. When the $\overline{\text{BREQ}}$ pin is sampled, at the prescribed timing, the $\overline{\text{BACK}}$ pin is driven low, and the address bus, data bus, and bus control signals are placed in the high-impedance state, establishing the external bus released state. For ICR, see section 13, I/O Ports.

In the external bus released state, the CPU, DTC, DMAC can access the internal space via the internal bus. When any one of the CPU, DTC, DMAC, and EXDMAC* attempts to access the external address space, it temporarily defers initiation of the bus cycle, and waits for the bus request from the external bus master to be canceled.

In the external bus released state, certain operations are suspended as follows until the bus request 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-module stop mode, control for software standby mode or all-module clock-stop mode is suspended.
- When SCKCR is written to set the clock frequencies, changing of clock frequencies is suspended. For SCKCR, see section 26, Clock Pulse Generator.

If the BREQOE bit in BCR1 is set to 1, the $\overline{\text{BREQO}}$ pin can be driven low to request cancel of the bus request when any of the following requests are issued.

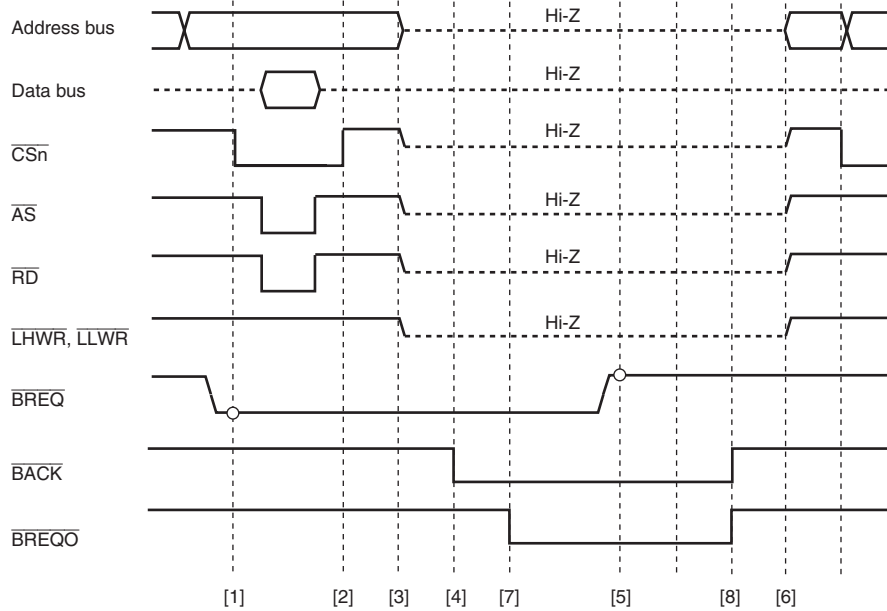
- When any one of the CPU, DTC, DMAC, and EXDMAC* attempts to access the external address space
- When a refresh* is requested

9.13.2 Pin States in External Bus Released State

Table 9.32 shows pin states in the external bus released state.

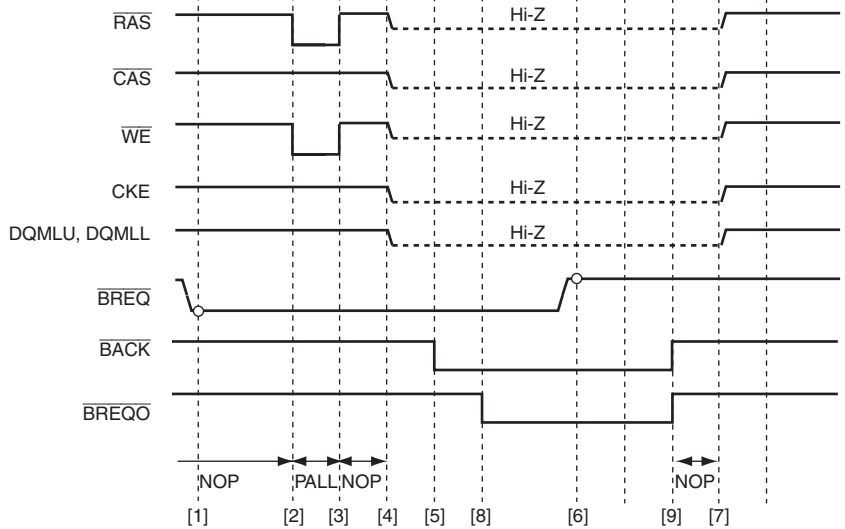
$\overline{\text{LUCAS}}$, $\overline{\text{LLCAS}}^*$	High impedance
$\overline{\text{RD}}$	High impedance
$\overline{\text{RAS}}^*$	High impedance
$\overline{\text{CAS}}^*$	High impedance
$\overline{\text{WE}}$	High impedance
$\overline{\text{DQMLU}}$, $\overline{\text{DQMLL}}^*$	High impedance
$\overline{\text{CKE}}^*$	High impedance
$\overline{\text{OE}}^*$	High impedance
$\overline{\text{LUB}}$, $\overline{\text{LLB}}$	High impedance
$\overline{\text{LHWR}}$, $\overline{\text{LLWR}}$	High impedance
$\overline{\text{DACK}}_n$ (n = 3 to 0)	High
$\overline{\text{EDACK}}_n$ (n = 3 to 0)*	High

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.



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

Figure 9.102 Bus Released State Transition Timing (SRAM Interface is Not T



- [1] A low level of the $\overline{\text{BREQ}}$ signal is sampled at the rising edge of the $\text{B}\phi$ 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 external bus released while the $\overline{\text{BREQOE}}$ 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 9.103 Bus Released State Transition Timing (SRAM Interface is Use

Table 9.34 Number of Access Cycles for On-Chip Memory Spaces

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

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

Table 9.34 lists the number of access cycles for registers of on-chip peripheral modules.

TPU, PPG0, TMR0, TMR1, SCI0 to SCI4, IIC2_0,
IIC2_1, A/D_0, and D/A registers

I/O port registers of PORTM* ¹ , N, TMR2, TMR3, SCI5, SCI6, IIC2_3, IIC2_4, A/D_1, AD_2, and PPG1 registers	Three P ϕ	Enabled
--	----------------	---------

- Notes: 1. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.
2. Supported only by the H8SX/1648L Group and the H8SX/1648H Group.

continues for two cycles or longer, and there is an internal access next, an external write is 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 than waiting until it ends are executed in parallel.

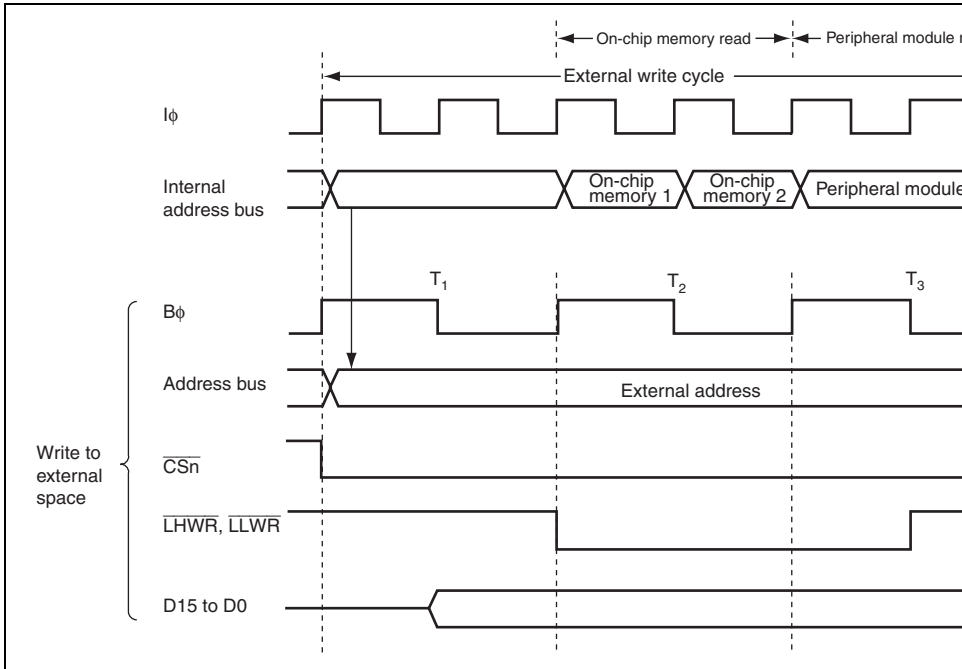


Figure 9.104 Example of Timing when Write Data Buffer Function is Used

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

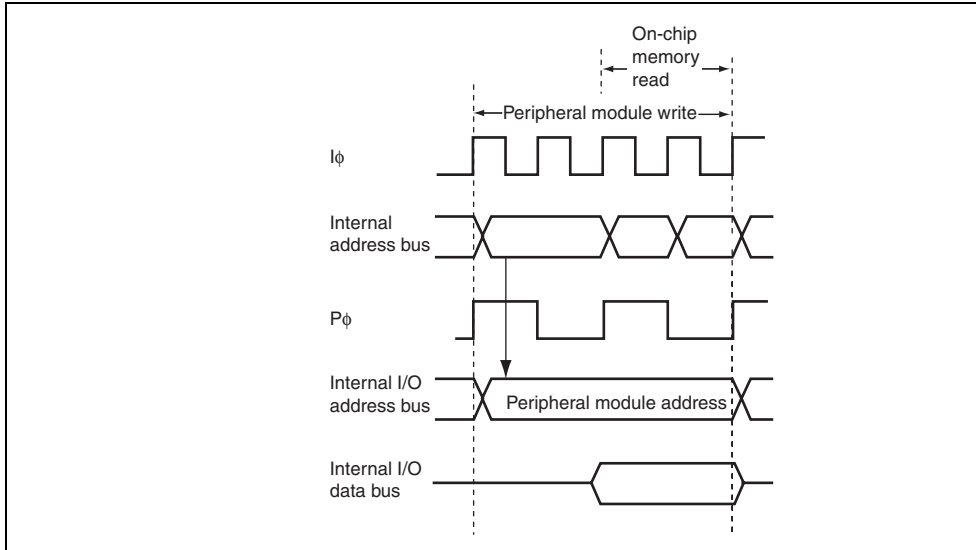


Figure 9.105 Example of Timing when Peripheral Module Write Data Buffer Function is Used

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

9.16.1 Operation

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

The priority of the internal bus arbitration:

DMAC > DTC > CPU

The priority of the external bus arbitration:

Refresh* > EXDMAC* > External bus release request > External access by the CPU
DMAC

If the DMAC or DTC accesses continue, the CPU can be given priority over the DMAC or DTC to execute the bus cycles alternatively between them by setting the IBCCS bit in BCR2. In this case, the priority between the DMAC and DTC does not change. If an external bus release request, an EXDMAC* access, and a refresh* cycle request continue, an external bus access by the CPU, DTC, and DMAC can be given priority to execute the bus cycles alternatively between them by setting the EBCCS bit in BCR2. In this case, the priorities among the refresh*, EXDMAC*, and external bus release request do not change.

An internal bus access by internal bus masters and an external bus access by an external bus master, a refresh* cycle, or an EXDMAC* access can be executed in parallel.

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

arbiter transfer the bus to the EXDMAC*.

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 condition, a cycle corresponding the write cycle)

Note: * Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

(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 the bus 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. If a bus master whose priority is higher than the DTC requests the bus, the DTC transfers the bus to the higher priority bus master. If the IBCCS bit in BCR2 is set to 1, the DTC transfers the bus to the CPU.

generated. When the DMAC accesses an external bus space, the DMAC first takes control of the bus from the internal bus arbiter and then requests a bus to the external bus arbiter.

After the DMAC takes control of the bus, it may continue the transfer processing cycles on 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 to the read 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 cases:

- 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) EXDMAC

The EXDMAC sends the external bus arbiter a request for the bus when an activation request is generated. If an internal bus master accesses the external space, the bus is passed to the internal bus master when the bus master can release the bus. Some EXDMAC transfers are continued once the internal bus master releases control of the bus. Some EXDMAC transfers are divided and it releases the bus for each bus cycle.

- Transfers are continued without bus release between a read cycle and the subsequent write cycle in dual address mode
- Transfers are continued without bus release in cluster transfer mode

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

(5) External Bus Release

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

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

(6) Refresh

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

A refresh cycle is inserted on completion of the external bus cycle. A refresh cycle is not consecutively inserted. Once a refresh cycle is inserted, the bus is passed to another bus master. When the bus is passed, if there is no bus request from other bus masters, NOP cycles are inserted.

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

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 is selected and the burst length must be 1.

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

(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 M = H'FFFFFFF) or allowing operation of the 8-bit timer module alone (MSTPCRA and M = H'F[C to F]FFFFFFF), the all-module-clock-stop mode is entered in which the clock for the controller and I/O ports is also stopped. For details, see section 27, Power-Down Modes.

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

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

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

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

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

(7) Refresh Settings

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

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

(8) Refresh Timer Settings

The setting of bits RTCK2 to RTCK0 in REFCR should be made after RTCNT and RTCOR have been set. When changing RTCNT and RTCOR, the counter operation should be halted. When changing bits RTCK2 to RTCK0, external access and external bus release by the EXDMR should be prohibited. The write data buffer function should be used after the write data buffer function is disabled and the external space is read.

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

RAS down mode (RCDM = 1) is used, execute the SLEEP instruction after setting the RAS down mode to 0. RAS down mode should be set again after recovery from software standby mode. For SBYCR, see section 27, Power-Down Modes.

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

(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 mode (RCDM = 0). When RAS down mode (RCDM = 1) is used, set the RCDM bit to 0 before writing to SCKCR. RAS down mode should be set again after clock frequencies are set. For SCKCR, see section 26, Clock Pulse Generator.

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

(12) Cluster Transfer to SDRAM Space

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

Note: Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

- DMAC activation methods are auto-request, on-chip module interrupt, and external request
 - Auto request: CPU activates (cycle stealing or burst access can be selected)
 - On-chip module interrupt: Interrupt requests from on-chip peripheral modules can be selected as an activation source
 - External request: Low level or falling edge detection of the $\overline{\text{DACK}}$ signal can be selected. External request is available for all four channels
- 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 $\overline{\text{DACK}}$ signal and the 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 per single transfer request
 - Repeat transfer mode: One byte, one word, or one longword data is transferred per single transfer request
 - Repeat size of data is transferred and then a transfer address is incremented and returns to the transfer start address
 - Up to 65536 transfers (65,536 bytes/words/longwords) can be set 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 size

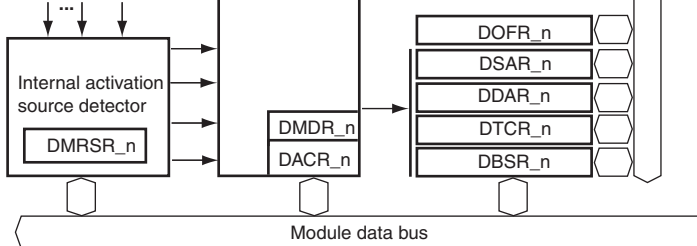
respective boundary

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 transfer 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.

- Module stop state can be set.



[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 | 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 10.1 Block Diagram of DMAC

1	DMA transfer request 1	$\overline{\text{DREQ1}}$	Input	Channel 1 external request
	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 request
	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 request
	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)

10.3.1 DMA Source Address Register (DSAR)

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

Although DSAR can always be read from by the CPU, it must be read from in longwords and 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 DTCCR 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	0
R/W	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	BKSZ8
Initial Value	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
Bit	7	6	5	4	3	2	1	
Bit Name	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0
Initial Value	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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	BKSZH31 to BKSZH16	All 0	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 set, the value means the maximum value (refer to table 10.1). When the DMA is in operation, the setting is fixed.
15 to 0	BKSZ15 to BKSZ0	All 0	R/W	Indicate the remaining repeat or block size while DMA is in operation. The value is decremented every time data is transferred. When the remaining value becomes 0, the value of the BKSZH bits is loaded with 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.

Bit Name	DTF27	DTF26	MD07	MD06	MD05	MD04	MD03	MD02	MD01
Initial Value	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	R/W
Bit	7	6	5	4	3	2	1		
Bit Name	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0	DMAP3
Initial Value	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R/W	R/W	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 while data is being transferred, this bit is cleared to 0 after the next 1-block size data transfer.

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

Operating modes and transfer methods must not be changed while this bit is set to 1.

0: Disables a data transfer

1: Enables a data transfer (DMA is in operation)

[Clearing conditions]

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

In block transfer mode, this bit changes after the next block transfer.

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

28	—	0	R/W	Reserved Initial value should not be changed.
27	DREQS	0	R/W	DREQ Select Selects whether a low level or the falling edge of the DREQ signal used in external request mode is used. 0: Low level detection 1: Falling edge detection (the first transfer after the transfer enabled is detected on a low level)
26	NRD	0	R/W	Next Request Delay Selects the accepting timing of the next transfer request. 0: Starts accepting the next transfer request after the completion of the current transfer 1: Starts accepting the next transfer request only 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 data 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.

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 interrupt has been generated in DMAC module stop mode, ERRF is not set to 1.

18	—	0	R	Reserved
----	---	---	---	----------

This bit is always read as 0 and cannot be modified.

17	ESIF	0	R/(W)*	Transfer Escape Interrupt Flag
----	------	---	--------	--------------------------------

Indicates that a transfer escape end interrupt has been requested. A transfer escape end means that a transfer is terminated before the transfer counter reaches 0.

0: A transfer escape end interrupt has not been requested
1: A transfer escape end interrupt has been requested

[Clearing conditions]

- When setting the DTE bit to 1
- When clearing to 0 before reading ESIF = 1

[Setting conditions]

- When a transfer size error interrupt is requested
- When a repeat size end interrupt is requested
- When a transfer end interrupt by an external interrupt or 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	DTSZ1	0	R/W	Data Access Size 1 and 0
14	DTSZ0	0	R/W	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	MDS1	0	R/W	Transfer Mode Select 1 and 0
12	MDS0	0	R/W	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 total transfer size set in DTCR is less than the data access size
 - In block transfer mode, the total transfer size set in DTCR is less than the block size
- 0: Disables a transfer size error interrupt request
1: Enables a transfer size error interrupt request

10	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
9	ESIE	0	R/W	Transfer Escape Interrupt Enable Enables/disables a transfer escape end interrupt request. When the ESIF bit is set to 1 with this bit set to 1, a transfer escape end interrupt is requested to the CPU or DTC. The transfer end interrupt request is 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 request to the CPU or DTC. When the DTIF bit is set to 1 with this bit set to 1, a transfer end interrupt is requested to the CPU or DTC. The transfer end interrupt request is 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	<p>Data Transfer Acknowledge</p> <p>This bit is valid in DMA transfer by the on-chip interrupt source. This bit enables or disables to source flag selected by DMRSR.</p> <p>0: To clear the source in DMA transfer is disabled. Since the on-chip module interrupt source is cleared in DMA transfer, it should be cleared by CPU or DTC transfer.</p> <p>1: To clear the source in DMA transfer is enabled. Since the on-chip module interrupt source is cleared in DMA transfer, it does not require an interrupt by the CPU or DTC transfer.</p>
4, 3	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0 and cannot be modified.</p>

- 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	R
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	R
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	R

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

transfer is requested after 1-block data transfer. When this bit is set to 1, the DTE bit in DMDR is cleared. At this time, the ESIF bit in DMDR is set to 1 to request 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 the source address 01: Specify the block area or repeat area on the destination address 10: Do not specify the block area or repeat area 11: Setting prohibited
23, 22	—	All 0	R	Reserved These bits are always read as 0 and cannot be 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 address (DSAR). When DSAR is not specified as the transfer source in single address mode, this bit is ignored. 00: Source address is fixed 01: Source address is updated by adding the data access size 10: Source address is updated by adding 1, 2, or 4 according to the data access size 11: Source address is updated by subtracting 1, 2, or 4 according to the data access size

10: Destination address is updated by adding
according to the data access size

11: Destination address is updated by subtracting
or 4 according to the data access size

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

When an overflow in the extended repeat area with the SARIE bit set to 1, an interrupt can be requested. Table 10.3 shows the settings and the extended repeat area.

7	DARIE	0	R/W	<p>Destination Address Extended Repeat Area Overflow Interrupt Enable</p> <p>Enables/disables an interrupt request for an extended repeat area overflow on the destination address.</p> <p>When an extended repeat area overflow on the destination address occurs while this bit is set to 1, the DTE bit in DMDR is cleared to 0. At this time, the SARIE bit in DMDR is set to 1 to indicate an interrupt request for an extended repeat area overflow on the destination address is requested.</p> <p>When block transfer mode is used with the extended repeat area function, an interrupt is requested at the completion of a 1-block size transfer. When the DTE bit in DMDR of the channel for which the transfer has been stopped to 1, the transfer is resumed from the state when the transfer is stopped.</p> <p>When the extended repeat area is not specified, this bit is ignored.</p> <p>0: Disables an interrupt request for an extended repeat area overflow on the destination address</p> <p>1: Enables an interrupt request for an extended repeat area overflow on the destination address</p>
6, 5	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0 and cannot be modified.</p>

area for address addition and subtraction, resp
When an overflow in the extended repeat area
with the DARIE bit set to 1, an interrupt can be
requested. Table 10.3 shows the settings and a
the extended repeat area.

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

[Legend]

x: Don't care

address	<ul style="list-style-type: none"> Repeat transfer (activated by CPU) Block transfer 	<ul style="list-style-type: none"> On-chip module interrupt External request 	<ul style="list-style-type: none"> size: 1 to 4 Gbytes or not specified Offset addition Extended repeat area function
	Repeat or block size = 1 to 65,536 bytes, 1 to 65,536 words, or 1 to 65,536 longwords		

Single address	<ul style="list-style-type: none"> 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/ $\overline{\text{DACK}}$
----------------	---	-----------------------------------

When the auto request setting is selected as the activation source, the cycle stealing or burst can be selected. When the total transfer size is not specified (DTCR = H'00000000), the counter is stopped and the transfer is continued without the limitation of the transfer counter.

divided into multiple bus cycles).

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

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

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

Figure 10.2 shows an example of the signal timing in dual address mode and figure 10.3 shows an example of operation in dual address mode.

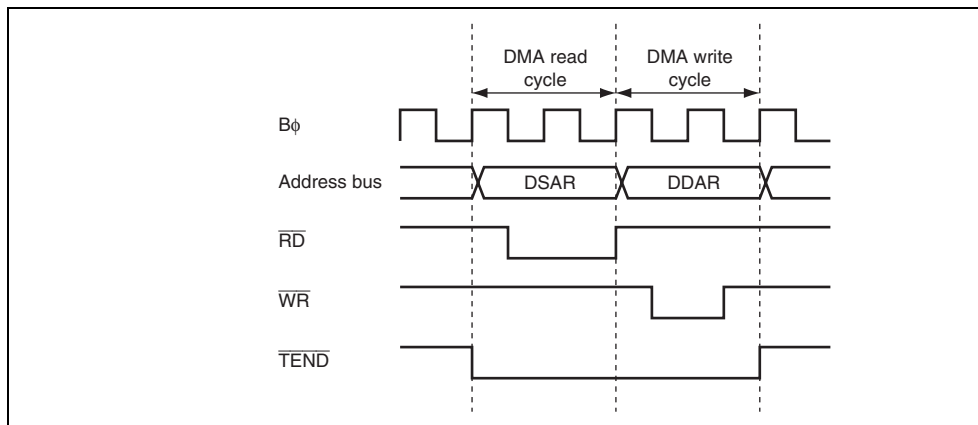


Figure 10.2 Example of Signal Timing in Dual Address Mode

(2) Single Address Mode

In single address mode, data between an external device and an external memory is directly transferred using the $\overline{\text{DACK}}$ pin instead of DSAR or DDAR. A transfer at a time is performed in one bus cycle. In this mode, the data bus width must be the same as the data access size. For details on the data bus width, see section 9, Bus Controller (BSC).

The DMAC accesses an external device as the transfer source or destination by outputting a strobe signal ($\overline{\text{DACK}}$) to the external device with $\overline{\text{DACK}}$ and accesses the other transfer source or destination by outputting the address. Accordingly, the DMA transfer is performed in one bus cycle. Figure 10.3 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 is transferred to the external memory in the same bus cycle.

The transfer direction is decided by the DIRS bit in DACR which specifies an external device or external memory as the transfer source or destination. When $\text{DIRS} = 0$, data is transferred from an external memory (DSAR) to an external device with the $\overline{\text{DACK}}$ pin. When $\text{DIRS} = 1$, data is transferred from an external device with the $\overline{\text{DACK}}$ pin to an external memory (DDAR). The settings of registers which are not used as the transfer source or destination are ignored.

The $\overline{\text{DACK}}$ signal output is enabled in single address mode by the DACKE bit in DMDR. When the $\overline{\text{DACK}}$ signal is low active.

The $\overline{\text{TEND}}$ signal output is enabled or disabled by the TENDE bit in DMDR. The $\overline{\text{TEND}}$ signal is output in one bus cycle. When an idle cycle is inserted before the bus cycle, the $\overline{\text{TEND}}$ signal is also output in the idle cycle.

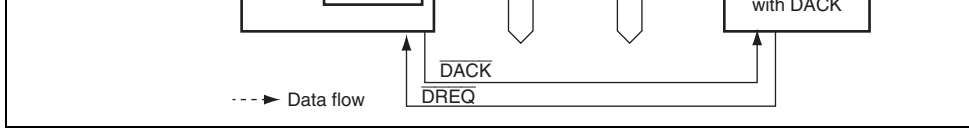


Figure 10.4 Data Flow in Single Address Mode

Transfer from external device with \overline{DACK} to external memory

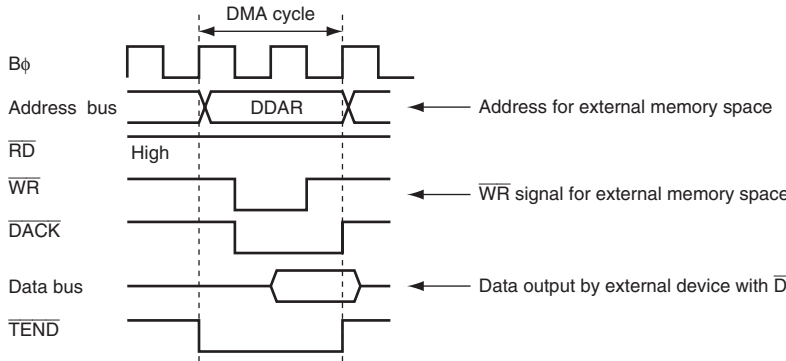


Figure 10.5 Example of Signal Timing in Single Address Mode

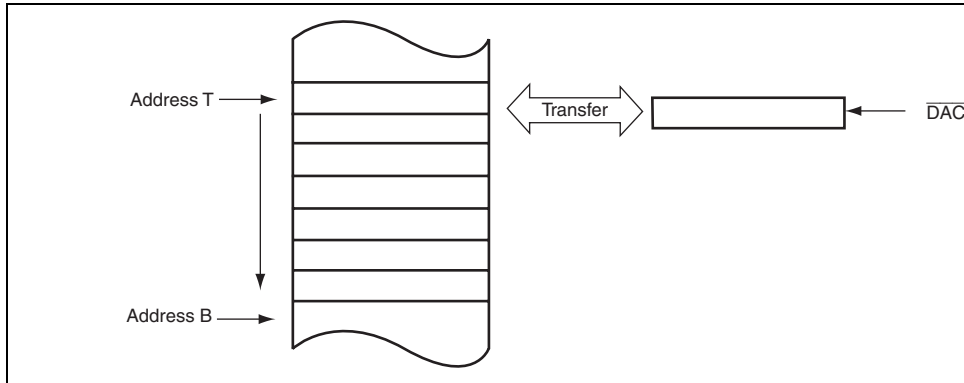


Figure 10.6 Operations in Single Address Mode

Figure 10.7 shows an example of the signal timing in normal transfer mode and figure 10.8 shows the operation in normal transfer mode.

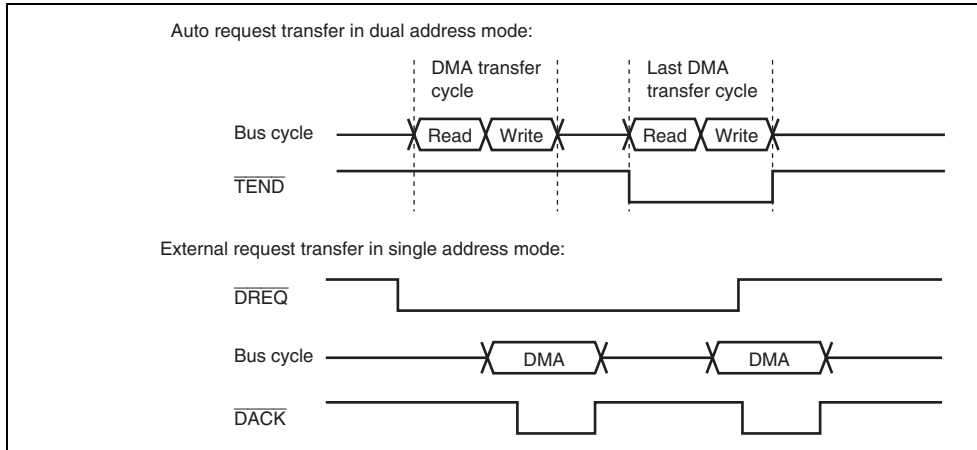


Figure 10.7 Example of Signal Timing in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data access size of data is transferred at a single transfer request. A total transfer size of up to 4 Gbytes can be specified as a total transfer size by DTCR. The repeat size can be specified by DBSR up to $65536 \times$ data access size.

The repeat area can be specified for the source or destination address side by bits ARS1 and ARS0 in DACR. The address specified as the repeat area returns to the transfer start address when the 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 as 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 requested to the CPU or DTC when the repeat size of transfers is completed. When the next transfer is requested after completion of a 1-repeat size data transfer while the RPTIE bit is set to 1, the DTE bit in DMDR is cleared to 0 and the ESIF bit in DMDR is set to 1 to complete the transfer. At this time, an interrupt is requested to the CPU or DTC when the ESIE bit in DMDR is set to 1.

The timings of the \overline{TEND} and \overline{DACK} signals are the same as in normal transfer mode.

Figure 10.9 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 operation is the same as the normal transfer mode operation shown in figure 10.8. In this case, a repeat size end interrupt can also be requested to the CPU when the repeat size of transfers is completed.

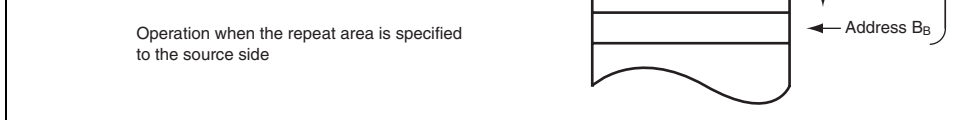


Figure 10.9 Operations in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, one block size of data is transferred at a single transfer request. Up to 65536 bytes can be specified as total transfer size by DTCR. The block size can be specified in units of 16 bytes up to $65536 \times$ data access size.

While one block of data is being transferred, transfer requests from other channels are suspended. 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 and ARS0 in DACR. The address specified as the block area returns to the transfer start address when the block size of data is completed. When the block area is specified as neither source nor destination address side, the operation continues without returning the address to the transfer start address. 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 transfer.

When an interrupt request by an extended repeat area overflow is used in block transfer mode, the settings should be selected carefully. For details, see section 10.5.5, Extended Repeat Area Function.

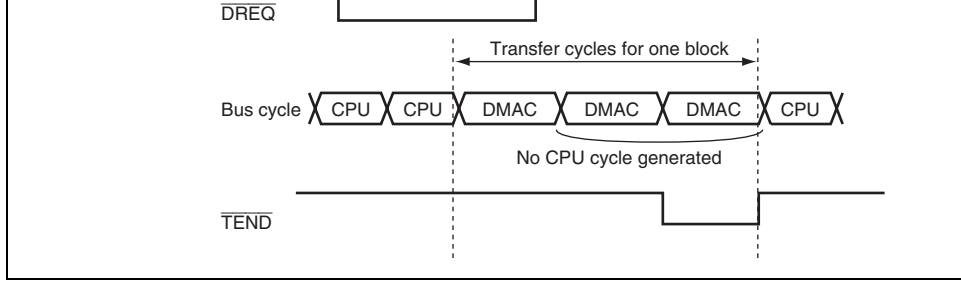


Figure 10.10 Operations in Block Transfer Mode

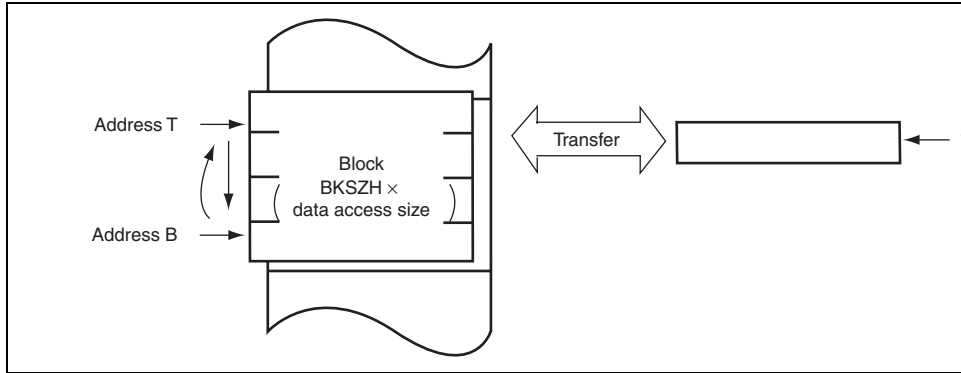
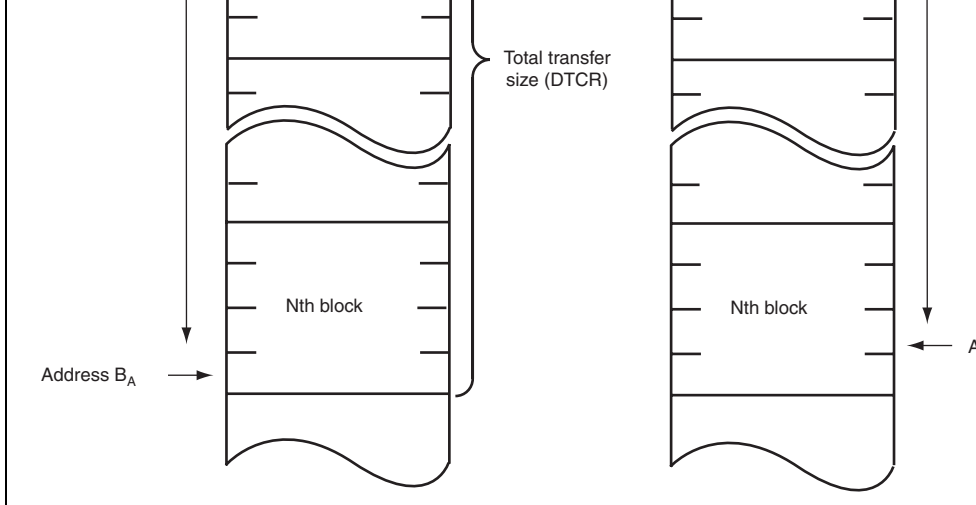


Figure 10.11 Operation in Single Address Mode in Block Transfer Mode (Block Area Specified)



**Figure 10.12 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 mo

(2) Activation by On-Chip Module Interrupt

An interrupt request from an on-chip peripheral module (on-chip peripheral module inte used as a transfer request. When a DMA transfer is enabled ($DTE = 1$), the DMA transfe started by an on-chip module interrupt.

The activation source of the on-chip module interrupt is selected by the DMA module re select register (DMRSR). The activation sources are specified to the individual channels 10.5 is a list of on-chip module interrupts for the DMAC. The interrupt request selected activation source can generate an interrupt request simultaneously to the CPU or DTC. F refer to section 7, Interrupt Controller.

The DMAC receives interrupt requests by on-chip peripheral modules independent of th controller. Therefore, the DMAC is not affected by priority given in the interrupt contro

When the DMAC is activated while $DTA = 1$, the interrupt request flag is automatically 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, 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 req 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 sour cleared to 0 before writing 1 to the DTE bit.

TGI5A (TGI5A input capture/compare match)	TPU_5	11
RXI0 (receive data full interrupt for SCI channel 0)	SCI_0	14
TXI0 (transmit data empty interrupt for SCI channel 0)	SCI_0	14
RXI1 (receive data full interrupt for SCI channel 1)	SCI_1	14
TXI1 (transmit data empty interrupt for SCI channel 1)	SCI_1	15
RXI2 (receive data full interrupt for SCI channel 2)	SCI_2	15
TXI2 (transmit data empty interrupt for SCI channel 2)	SCI_2	15
RXI3 (receive data full interrupt for SCI channel 3)	SCI_3	15
TXI3 (transmit data empty interrupt for SCI channel 3)	SCI_3	15
RXI4 (receive data full interrupt for SCI channel 4)	SCI_4	16
TXI4 (transmit data empty interrupt for SCI channel 4)	SCI_4	16
TGI6A (TGI6A input capture/compare match)	TPU_6	16
TGI7A (TGI7A input capture/compare match)	TPU_7	16
TGI8A (TGI8A input capture/compare match)	TPU_8	17
TGI9A (TGI9A input capture/compare match)	TPU_9	17
TGI10A (TGI10A input capture/compare match)	TPU_10	18
TGI11A (TGI11A input capture/compare match)	TPU_11	18
RXI5 (receive data full interrupt for SCI channel 5)	SCI_5	22
TXI5 (transmit data empty interrupt for SCI channel 5)	SCI_5	22
RXI6 (receive data full interrupt for SCI channel 6)	SCI_6	22
TXI6 (transmit data empty interrupt for SCI channel 6)	SCI_6	22
ADI2 (conversion end interrupt for A/D converter unit 2)	A/D_2	23
ADI1 (conversion end interrupt for A/D converter unit 1)	A/D_1	23

When an external request is selected as an activation source, clear the DEN bit to 0 and ICR bit to 1 for the corresponding pin. For details, see section 13, I/O Ports.

10.5.4 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 by DTF0 in DMDR. When an activation source is the on-chip module interrupt or external interrupt, 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, halfword, longword, or 1-block size) is completed. After that, when a transfer is requested, the DMAC obtains the bus to transfer 1-unit data and then releases the bus on completion of the transfer. The transfer operation is continued until the transfer end condition is satisfied.

When a transfer is requested to another channel during a DMA transfer, the DMAC releases the bus and then transfers data for the requested channel. For details on operations when a transfer is requested to multiple channels, see section 10.5.8, Priority of Channels.

Figure 10.13 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 the bus until the transfer end condition is satisfied. Even if a transfer is requested from another channel with higher priority, the transfer is not stopped once it is started. The DMAC releases the bus in the next cycle after the transfer for the channel in burst mode is completed. This is similar to operation in cycle stealing mode. However, setting the IBCCS bit in BCR2 of the bus controller makes the DMAC 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 cycle stealing mode during one block of transfers). The DMAC is always operated in cycle stealing mode.

Clearing the DTE bit in DMDR stops a DMA transfer. A transfer requested before the DTE bit is cleared to 0 by the DMAC is executed. When an interrupt by a transfer size error, a repeat area overflow, or an extended repeat area overflow occurs, the DTE bit is cleared to 0 and the transfer is stopped.

Figure 10.14 shows an example of timing in burst mode.

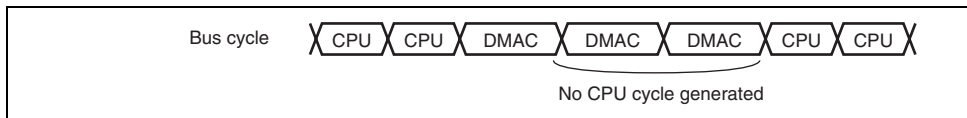


Figure 10.14 Example of Timing in Burst Mode

The extended repeat area on the source address is specified by bits SARA4 to SARA0 in DSAR. The extended repeat area on the destination address is specified by bits DARA4 to DARA0 in DDAR and 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 requested to the CPU when the contents of the address register reach the end address of the extended repeat area. When an overflow on the extended repeat area set in DSAR occurs while the SARA bit in DACR is set to 1, the ESIF bit in DMDR is set to 1 and the DTE bit in DMDR is cleared to stop the transfer. At this time, if the ESIE bit in DMDR is set to 1, an interrupt by an extended repeat area overflow is requested to the CPU. When the DARIE bit in DACR is set to 1, an overflow on the extended repeat area set in DDAR occurs, meaning that the destination address has reached the target. During the interrupt handling, setting the DTE bit in DMDR resumes the transfer.

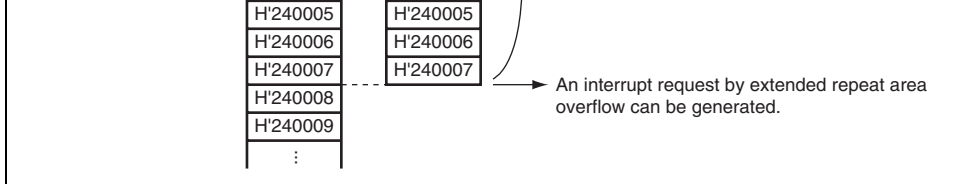


Figure 10.15 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during transfer of one block, the interrupt by the overflow is suspended and the transfer overruns.

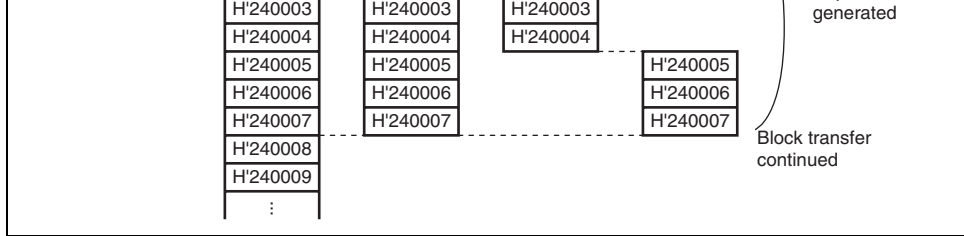


Figure 10.16 Example of Extended Repeat Area Function in Block Transfer

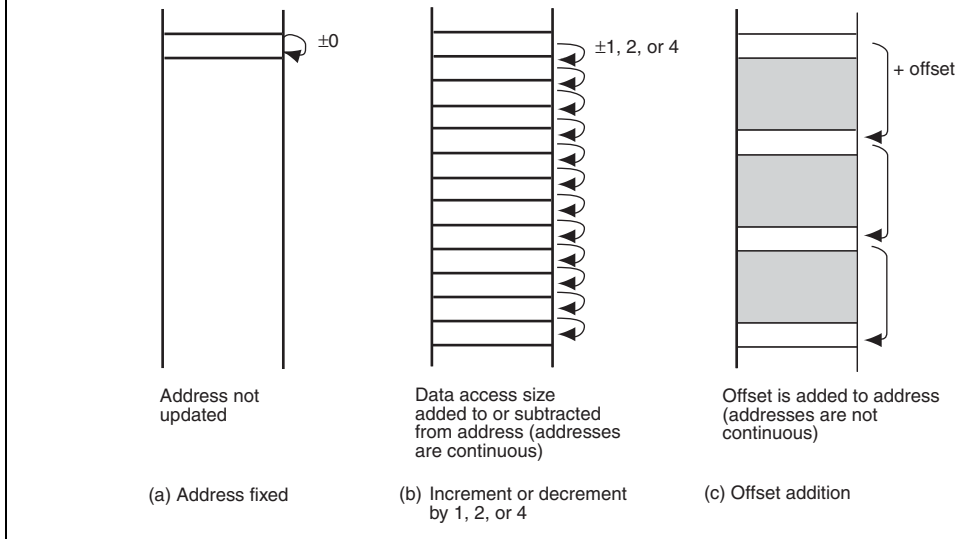


Figure 10.17 Address Update Method

In item (a), Address fixed, the transfer source or destination address is not updated indicating the same address.

In item (b), Increment or decrement by 1, 2, or 4, the transfer source or destination address is incremented or decremented by the value according to the data access size at each transfer. A byte, word, or longword can be specified as the data access size. The value of 1 for byte, 2 for word, or 4 for longword is used for updating the address. This operation realizes the data transfer in consecutive areas.

In item (c), Offset addition, the address update does not depend on the data access size. The value specified by DOFR is added to the address every time the DMAC transfers data of the data access size.

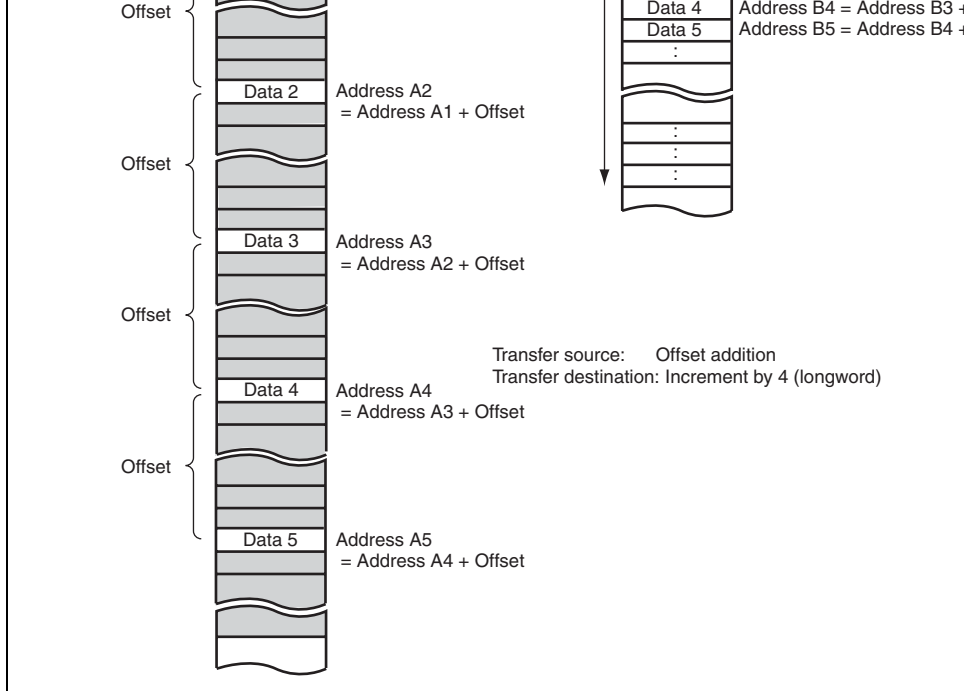


Figure 10.18 Operation of Offset Addition

In figure 10.18, the offset addition is selected as the transfer source address update and increment or decrement by 1, 2, or 4 is selected as the transfer destination address. The address update is performed so that data at the address which is away from the previous transfer source address by the offset is read from. The data read from the address away from the previous address is written to the next consecutive area in the destination side.

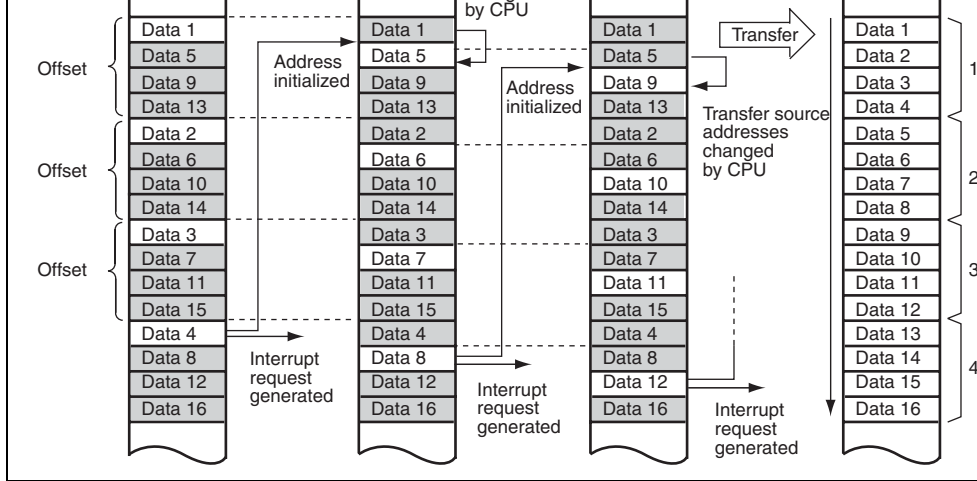


Figure 10.19 XY Conversion Operation Using Offset Addition in Repeat Transfer

In figure 10.19, the source address side is specified to the repeat area by DACR and the offset addition is selected. The offset value is set to $4 \times$ data access size (when the data access size is longword, H'00000010 is set in DOFR, as an example). The repeat size is set to $4 \times$ data access size (when the data access size is longword, the repeat size is set to $4 \times 4 = 16$ bytes, as an example). The increment by 1, 2, or 4 is specified as the transfer destination address. A repeat end interrupt is requested when the repeat size of transfers is completed.

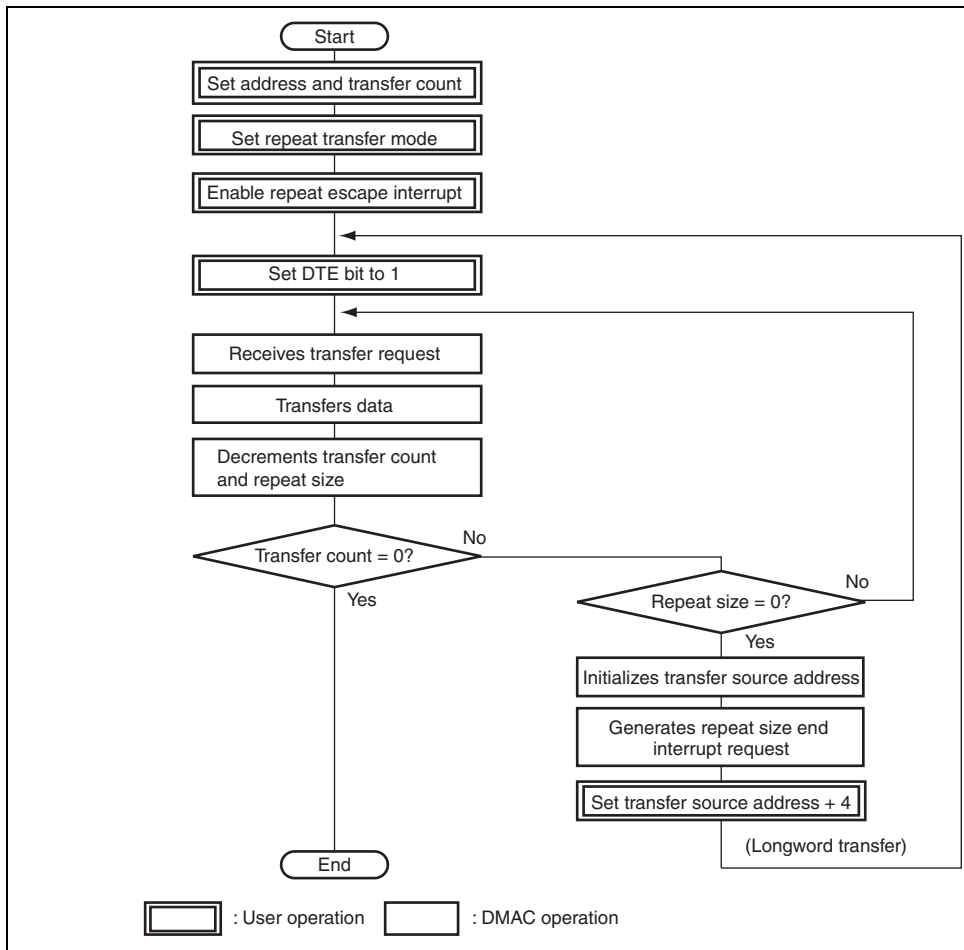


Figure 10.20 XY Conversion Flowchart Using Offset Addition in Repeat Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and transfer state. The registers to be updated are DSAR, DDAR, DTCL, BKSZH and BKSZ in DBSR, and the DTE, ACT, ERRF, ESIF, and DTIF bits in DMDR.

(1) DMA Source Address Register

When the transfer source address set in DSAR is accessed, the contents of DSAR are output, and then are updated to the next address.

The increment or decrement can be specified by bits SAT1 and SAT0 in DACR. When SAT1 and SAT0 = B'00, the address is fixed. When SAT1 and SAT0 = B'01, the address is added with the value of DSAR. When SAT1 and SAT0 = B'10, the address is incremented. When SAT1 and SAT0 = B'11, the address is decremented. The size of increment or decrement depends on the data access size.

The data access size is specified by bits DTSZ1 and DTSZ0 in DMDR. When DTSZ1 and DTSZ0 = B'00, the data access size is byte and the address is incremented or decremented by 1. When DTSZ1 and DTSZ0 = B'01, the data access size is word and the address is incremented or decremented by 2. When DTSZ1 and DTSZ0 = B'10, the data access size is longword and the address is incremented or decremented by 4. Even if the access data size of the source address is byte, word or longword, when the source address is not aligned with the word or longword boundary, the read bus cycle is divided into byte or word cycles. While data of one word or one longword is being read, the size of increment or decrement is changing according to the actual data access size. For example, +1 or +2 for byte or word data. After one word or one longword of data is read, the source address when the read cycle is started is incremented or decremented by the value according to bits SAT1 and SAT0.

(2) DMA Destination Address Register

When the transfer destination address set in DDAR is accessed, the contents of DDAR are updated and then are updated to the next address.

The increment or decrement can be specified by bits DAT1 and DAT0 in DACR. When DAT1 and DAT0 = B'00, the address is fixed. When DAT1 and DAT0 = B'01, the address is incremented by the offset. When DAT1 and DAT0 = B'10, the address is incremented. When DAT1 and DAT0 = 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 and DTSZ0 = B'00, the data access size is byte and the address is incremented or decremented by 1. When DTSZ1 and DTSZ0 = B'01, the data access size is word and the address is incremented or decremented by 2. When DTSZ1 and DTSZ0 = B'10, the data access size is longword and the address is incremented or decremented by 4. Even if the access data size of the destination is word or longword, when the destination address is not aligned with the word or longword boundary, the write bus cycle is divided into byte and word cycles. While one word or one longword of data is being written, the incrementing or decrementing size is changing according to the actual data access size, for example, +1 or +2 for byte or word data. After the one word or one 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 completed, the block or repeat area is specified to the destination address side, the destination address is updated 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 follows the 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 accessed in words. If the upper word and lower word are read separately, incorrect data may be read since the contents of DTCR during the transfer may be updated regardless of the access by the 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, the CPU has priority. When a conflict occurs between change from 1, 2, or 4 to 0 in DTCR and write access by the CPU (other than 0), the CPU has priority in writing to DTCR. However, DMA 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 BKSZ. Bits 15 to 0 in DBSR function as BKSZH. The BKSZH bits (16 bits) store the block size and its value is not changed. The BKSZ bits (16 bits) function as a counter for the block size and repeat size and its value is decremented every transfer by 1. When the BKSZ value is changed from 1 to 0 by a DMA transfer, 0 is not stored but the BKSZH value is loaded into the 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 an 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 prohibited (except for the DTE bit). When changing the register settings after writing 0 to the DTE bit, confirm that the DTE bit has been cleared to 0.

Figure 10.21 show the procedure for changing the register settings for the channel being transferred.

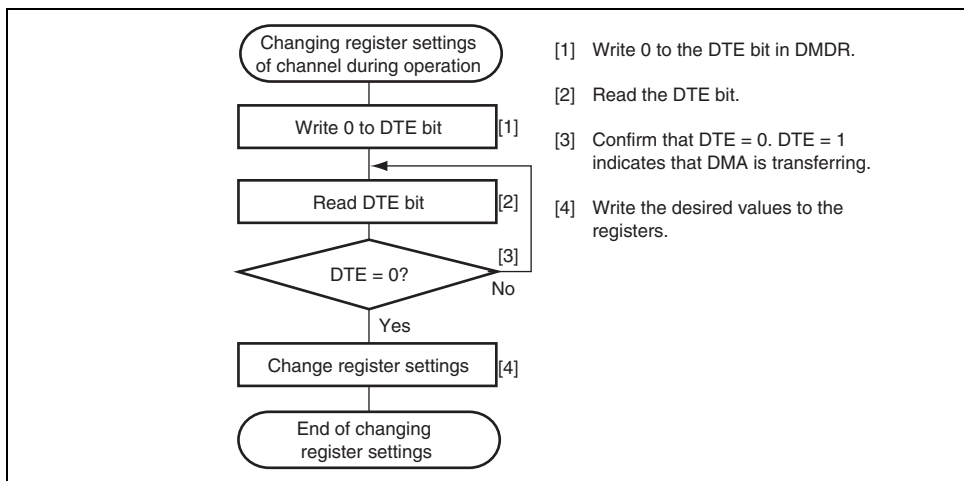


Figure 10.21 Procedure for Changing Register Setting For Channel being Transferred

In burst mode, up to three times of DMA transfer are performed from the cycle in which the ACT bit is written to 0. The ACT bit retains 1 from writing 0 to the DTE bit to completion of DMA transfer.

(7) ERRF Bit in DMDR

When an address error or an NMI interrupt occur, the DMAC clears the DTE bits for all transfer channels to stop a transfer. In addition, it sets the ERRF bit in DMDR_0 to 1 to indicate that an address error or an NMI interrupt has occurred regardless of whether or not the DMAC is in operation.

(8) ESIF Bit in DMDR

When an interrupt by a transfer size error, a repeat size end, or an extended repeat area completion is requested, the ESIF bit in DMDR is set to 1. When both the ESIF and ESIE bits are set to 1, a 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 after completion of a cycle of the interrupt source is completed.

The ESIF bit is automatically cleared to 0 and a transfer request is cleared if the transfer is resumed by setting the DTE bit to 1 during interrupt handling.

For details on interrupts, see section 10.8, Interrupt Sources.

10.5.8 Priority of Channels

The channels of the DMAC are given following priority levels: channel 0 > channel 1 > channel 2 > channel 3. Table 10.6 shows the priority levels among the DMAC channels.

Table 10.6 Priority among DMAC Channels

Channel	Priority
Channel 0	Highest
Channel 1	High
Channel 2	Low
Channel 3	Lowest

The channel having highest priority other than the channel being transferred is selected if a transfer is requested from other channels. The selected channel starts the transfer after the channel being transferred releases the bus. At this time, when a bus master other than the DMAC requests the bus, the cycle for the bus master is inserted.

In a burst transfer or a block transfer, channels are not switched.

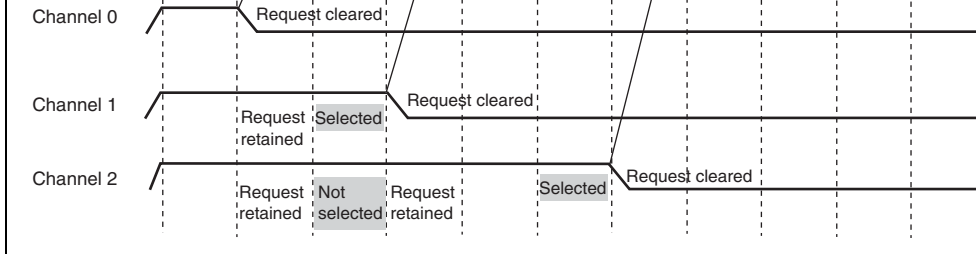


Figure 10.22 Example of Timing for Channel Priority

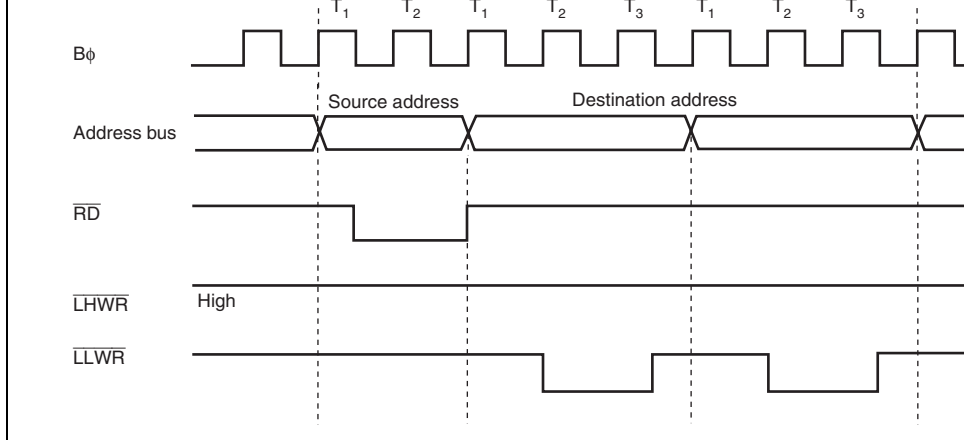


Figure 10.23 Example of Bus Timing of DMA Transfer

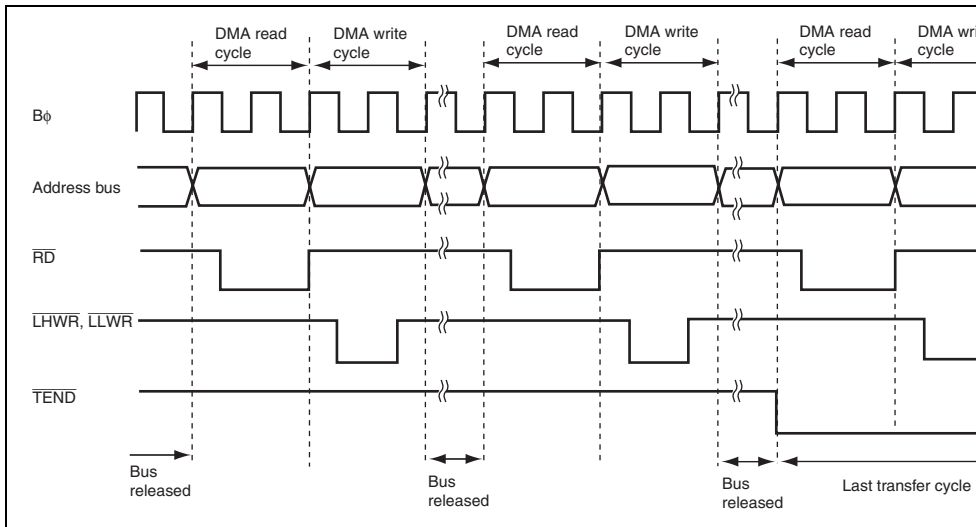


Figure 10.24 Example of Transfer in Normal Transfer Mode by Cycle Stealing

In figures 10.25 and 10.26, the \overline{TEND} signal output is enabled and data is transferred in longword mode from the external 16-bit 2-state access space to the 16-bit 2-state access space in normal transfer mode by cycle stealing.

In figure 10.25, the transfer source (DSAR) is not aligned with a longword boundary and the transfer destination (DDAR) is aligned with a longword boundary.

In figure 10.26, the transfer source (DSAR) is aligned with a longword boundary and the transfer destination (DDAR) is not aligned with a longword boundary.



Figure 10.25 Example of Transfer in Normal Transfer Mode by Cycle Stealing (Transfer Source DSAR = Odd Address and Source Address Increment)

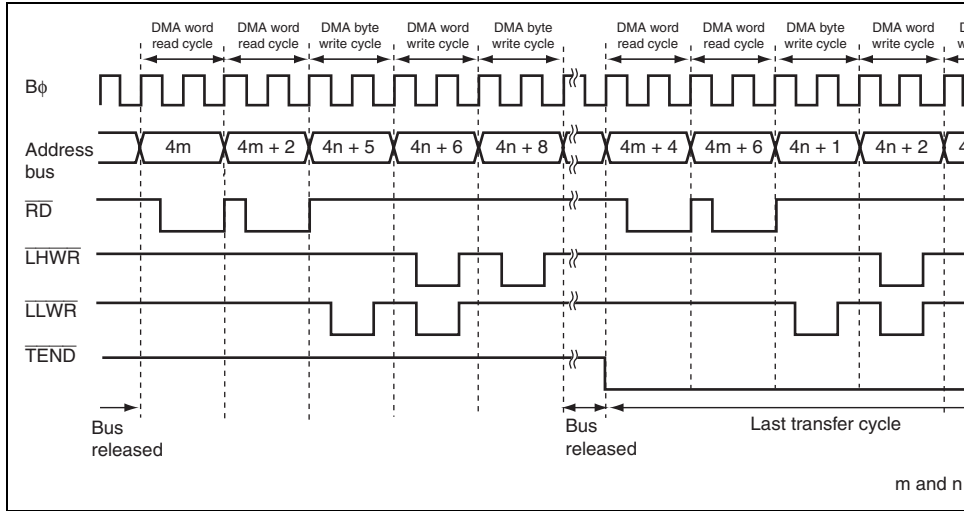


Figure 10.26 Example of Transfer in Normal Transfer Mode by Cycle Stealing (Transfer Destination DDAR = Odd Address and Destination Address Decrement)

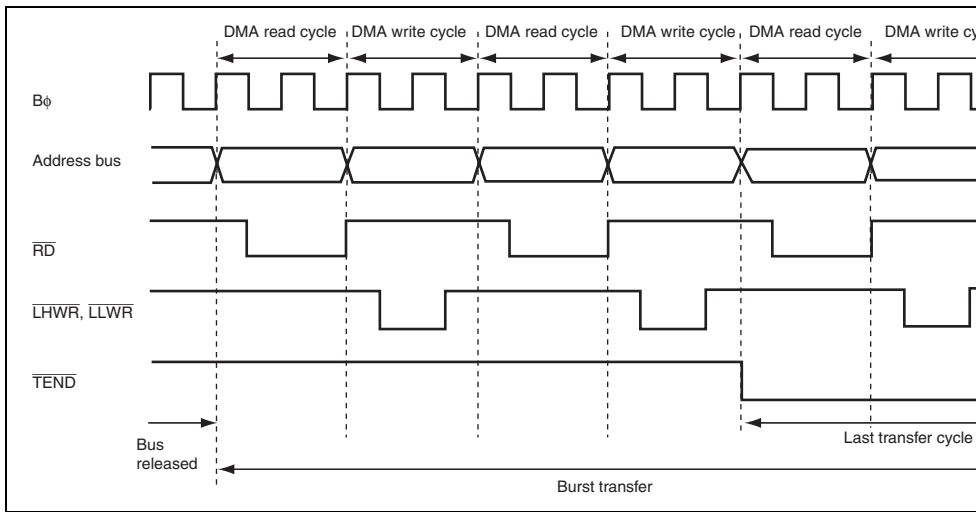


Figure 10.27 Example of Transfer in Normal Transfer Mode by Burst Access

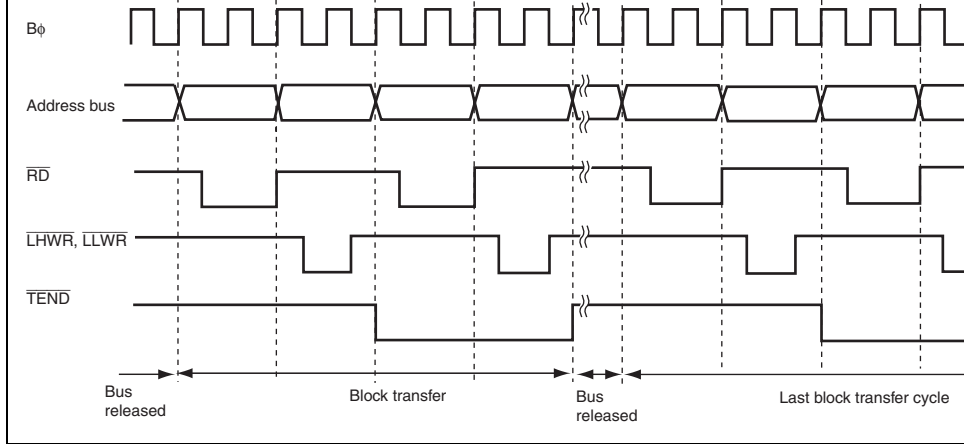


Figure 10.28 Example of Transfer in Block Transfer Mode

When a high level of the $\overline{\text{DREQ}}$ signal has been detected until completion of the DMA write cycle, receiving the next transfer request resumes and then a low level of the $\overline{\text{DREQ}}$ signal is detected. This operation is repeated until the transfer is completed.

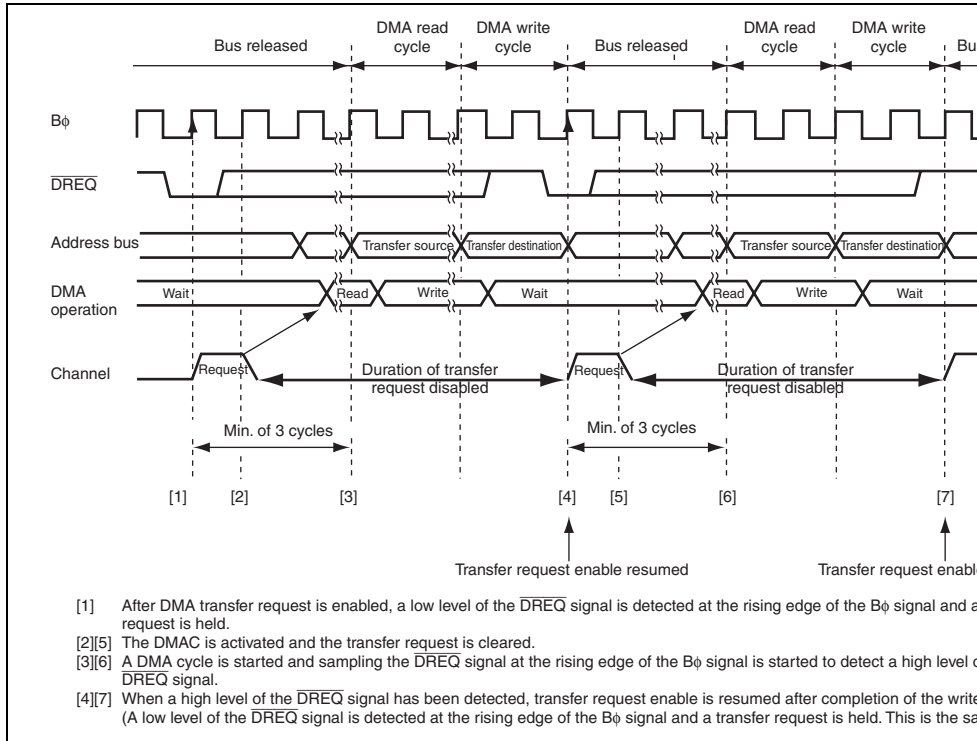


Figure 10.29 Example of Transfer in Normal Transfer Mode Activated by $\overline{\text{DREQ}}$ Falling Edge

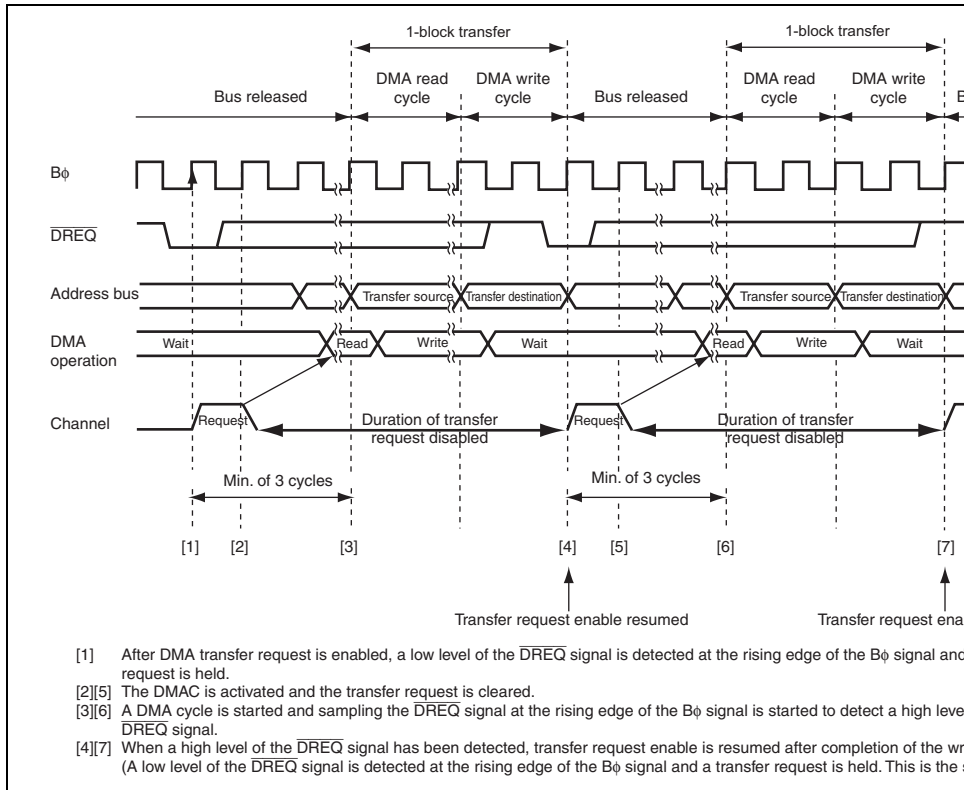


Figure 10.30 Example of Transfer in Block Transfer Mode Activated by \overline{DREQ} Falling Edge

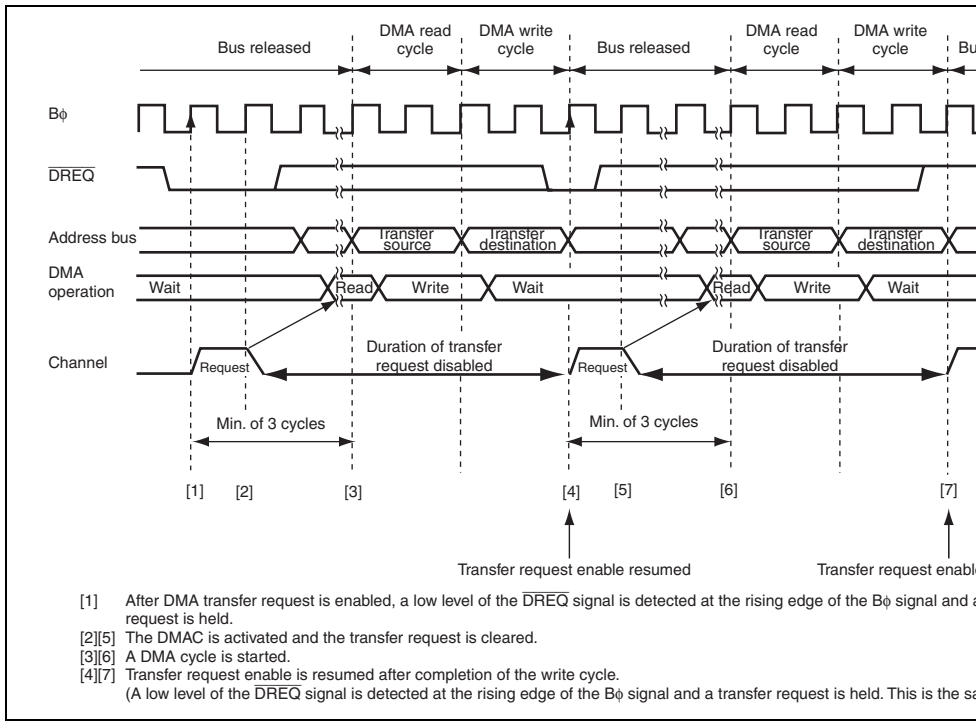
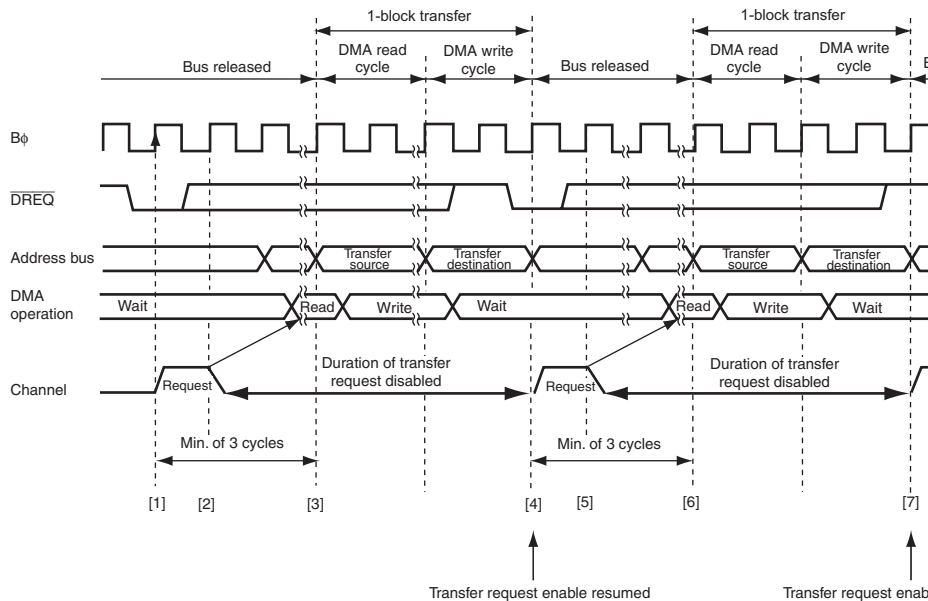


Figure 10.31 Example of Transfer in Normal Transfer Mode Activated by \overline{DREQ} Low Level



- [1] After DMA transfer request is enabled, a low level of the \overline{DREQ} signal is detected at the rising edge of the $B\phi$ signal and a transfer 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 $B\phi$ signal and a transfer request is held. This is the same as [1].)

Figure 10.32 Example of Transfer in Block Transfer Mode Activated by \overline{DREQ} Low Level

enabled, a transfer request is held in the DMAC. When the DMAC is activated, the transfer request is cleared. Receiving the next transfer request resumes after completion of the write cycle and then a low level of the $\overline{\text{DREQ}}$ signal is detected. This operation is repeated until the transfer is completed.

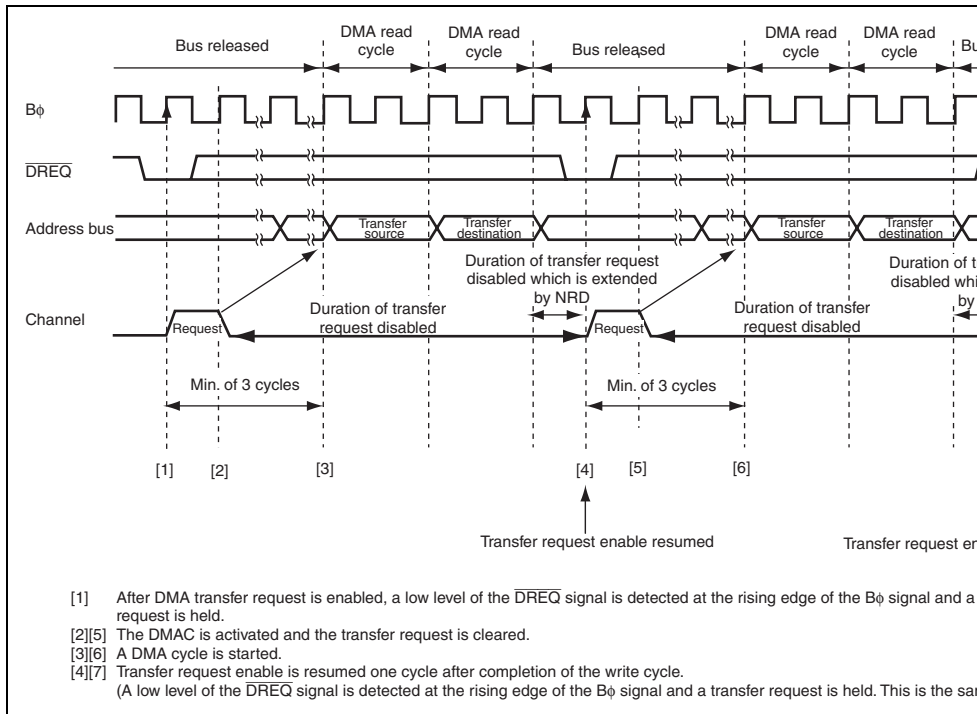


Figure 10.33 Example of Transfer in Normal Transfer Mode Activated by $\overline{\text{DREQ}}$ Low Level with $\text{NRD} = 1$

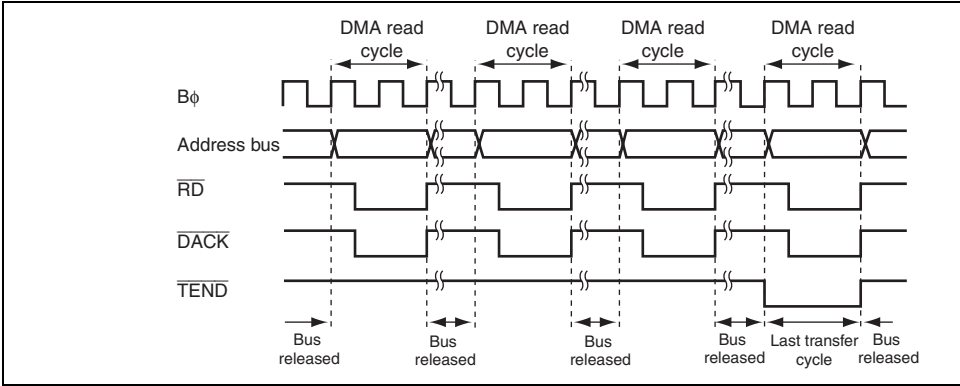


Figure 10.34 Example of Transfer in Single Address Mode (Byte Read)

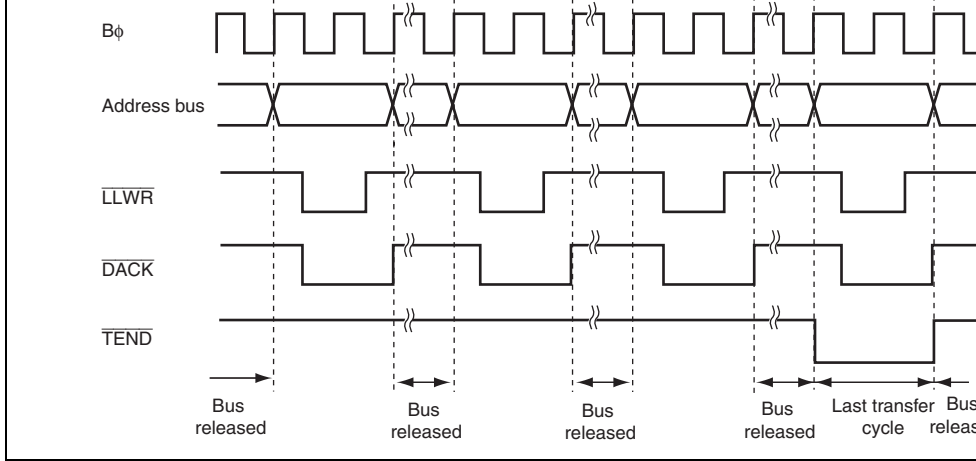


Figure 10.35 Example of Transfer in Single Address Mode (Byte Write)

operation is repeated until the transfer is completed.

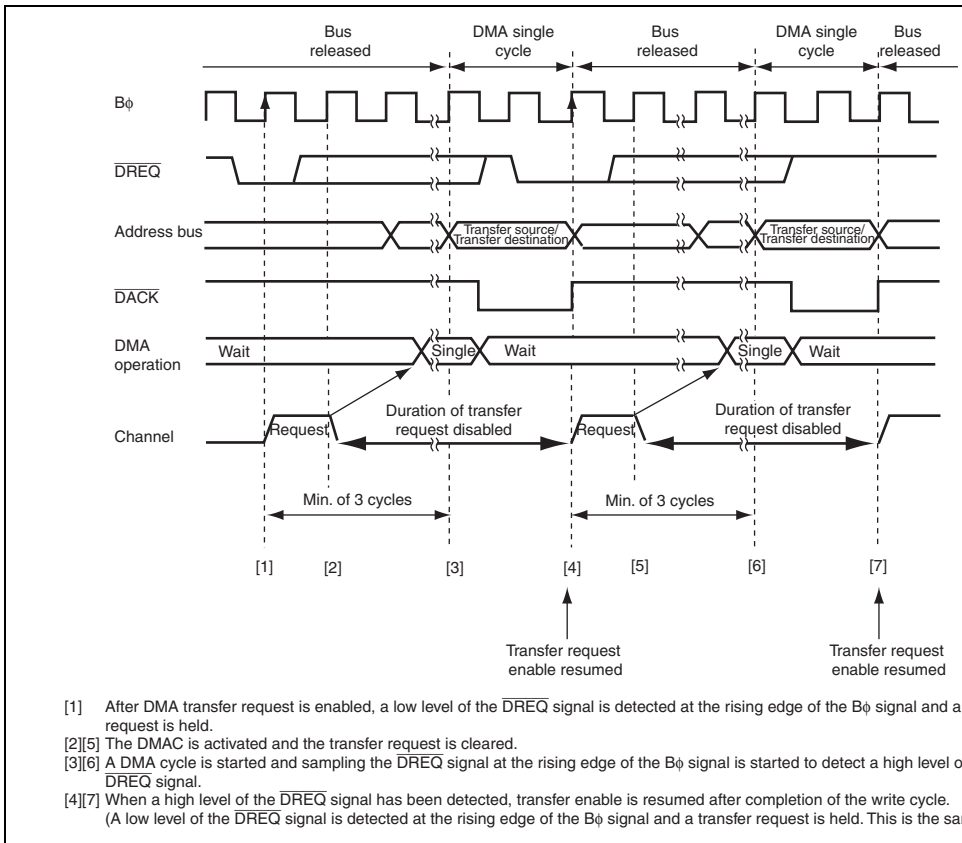
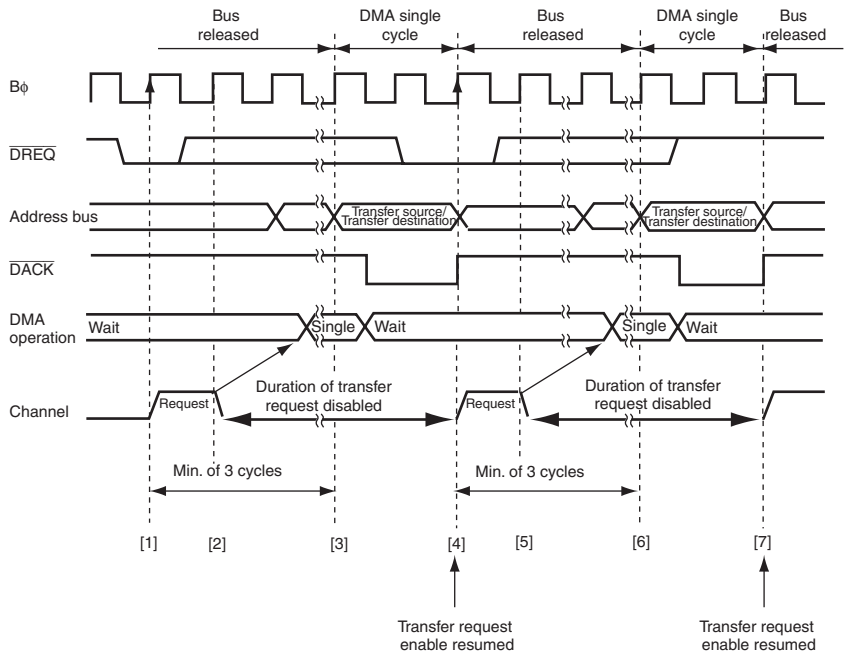


Figure 10.36 Example of Transfer in Single Address Mode Activated by \overline{DREQ} Falling Edge



- [1] After DMA transfer request is enabled, a low level of the \overline{DREQ} signal is detected at the rising edge of the $B\phi$ signal and a transfer 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 \overline{DREQ} signal is detected at the rising edge of the $B\phi$ signal and a transfer request is held. This is the same as [1].)

Figure 10.37 Example of Transfer in Single Address Mode Activated by \overline{DREQ} Low Level

enabled, a transfer request is held in the DMAC. When the DMAC is activated, the transfer request is cleared. Receiving the next transfer request resumes after one cycle of the transfer request duration inserted by $\overline{\text{NRD}} = 1$ on completion of the single cycle and then a low level $\overline{\text{DREQ}}$ signal is detected. This operation is repeated until the transfer is completed.

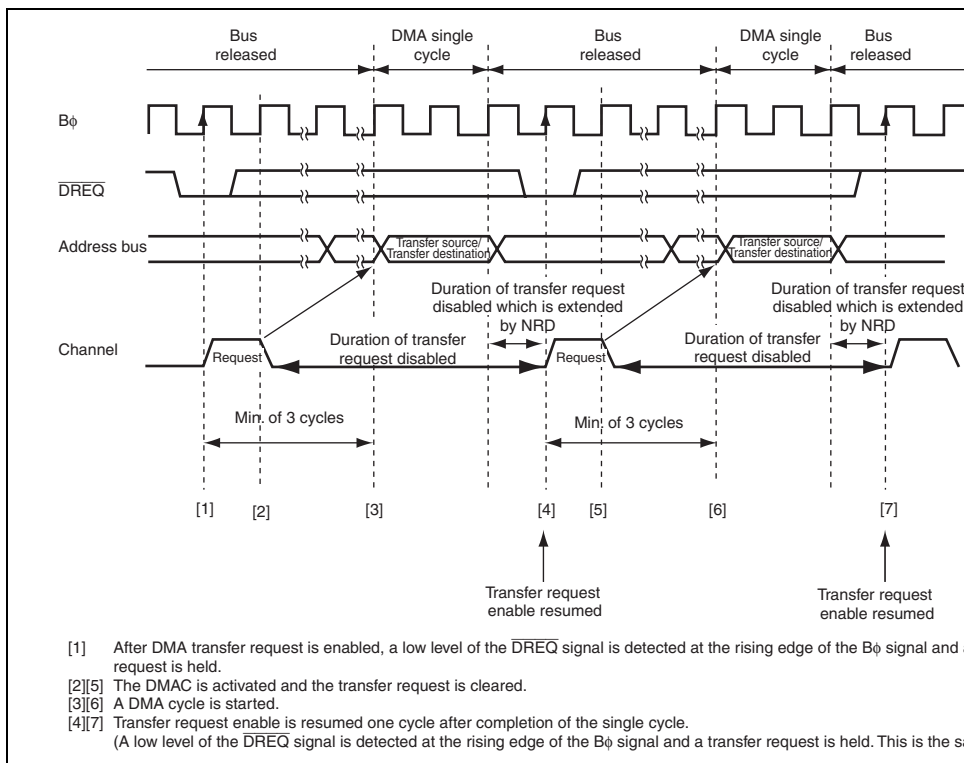


Figure 10.38 Example of Transfer in Single Address Mode Activated by $\overline{\text{DREQ}}$ Low Level with $\overline{\text{NRD}} = 1$

(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 transfer size error occurs and a DMA transfer is terminated. At this time, the DTE bit in DMDR is set to 0 and the ESIF bit in DMDR is set to 1.

- In normal transfer mode and repeat transfer mode, when the next transfer is requested while a 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 disabled due to 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 reaches 0. 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 the data access size, data is transferred in bytes
- In block transfer mode, when the DTCR value is less than the block size, the specified data in DTCR is transferred instead of transferring the block size of data. The transfer is performed in bytes.

When an overflow on the extended repeat area occurs while the extended repeat area is selected 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, the SARIE or DARIE 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 during a read cycle, the following write cycle is performed.

In block transfer mode, even if an interrupt by an extended repeat area overflow occurs during a block transfer, the remaining data is transferred. The transfer is not terminated by an extended 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 current 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 for the transfer unit.

(b) Block Transfer Mode

A DMA transfer is forced to stop. Since a 1-block size of transfers is not completed, operation is not guaranteed.

In dual address mode, the write cycle corresponding to the read cycle is performed. This is the same as 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 ERRF bit in DMDR_0 is set to 1. When an address error occurs during a DMA transfer, the transfer is forced to stop. To perform a DMA transfer after an address error occurs, clear the ERRF bit 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 DMA transfer is not guaranteed.

The priority level of the CPU is specified by bits CPUP2 to CPUP0. The value of bits CPUP2 to 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 has priority over the DMAC, a transfer request for the corresponding channel is masked and the transfer is not activated. When another channel has priority over or the same as the CPU, a transfer request is 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, a transfer request is received and the transfer is resumed. Writing 0 to the DTE bit clears the suspended transfer request.

When the CPUPCE bit is cleared to 0, it is regarded as the lowest priority.

write cycles of a DMA transfer.

In block transfer mode and an auto request transfer by burst access, bus cycles of the DMAC transfer are consecutively performed. For this duration, since the DMAC has priority over CPU and DTC, accesses to the external space are suspended (the IBCCS bit in the bus controller register 2 (BCR2) is cleared to 0).

When the bus is passed to another channel or an auto request transfer by cycle stealing, bus cycles 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 setting the IBCCS bit in BCR2, the bus is used alternatively except the bus cycles which are not separated. For details, see section 9, Bus Controller (BSC).

A conflict may occur among the external space access of a DMAC, refresh cycle*, an EXDMAC cycle*, and an external bus release cycle. Even if a burst or block transfer is performed by the DMAC, the transfer is stopped temporarily and the refresh cycle*, EXDMAC cycle*, and external bus release cycle are inserted by the BSC according to the external bus priority (when the external access and the DTC external access do not have priority over a DMAC transfer, DMAC 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 cycle. 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 an EXDMAC cycle, an EXDMAC cycle, and an external bus release cycle may be performed at the same time.

Note: * Supported only by the H8S/1648G Group and H8S/1648H Group.

DMTEND1	Transfer end interrupt by channel 1 transfer counter
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

Each interrupt is enabled or disabled by the DTIE and ESIE bits in DMDR for the corresponding channel. A DMTEND interrupt is generated by the combination of the DTIF and DTIE bits in DMDR. A DMEEND interrupt is generated by the combination of the ESIF and ESIE bits in DMDR. The DMEEND interrupt sources are not distinguished. The priority among channels is decided by the interrupt controller and it is shown in table 10.7. For details, see section 10.7.3. Interrupt Controller.

An interrupt other than the transfer end interrupt by the transfer counter is generated when the ESIF bit in DMDR is set to 1. The ESIF bit is set to 1 when the conditions are satisfied by transfer while the enable bit is set to 1.

A transfer size error interrupt is generated when the next transfer cannot be performed because the DTCR value is less than the data access size, meaning that the data access size of transfer cannot be performed. In block transfer mode, the block size is compared with the DTCR value for transfer error decision.

A repeat size end interrupt is generated when the next transfer is requested after completing the repeat size of transfers in repeat transfer mode. Even when the repeat area is not specified in the address register, the transfer can be stopped periodically according to the repeat size. At the time 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 addresses is generated when the address exceeds the extended repeat area (overflow). At this time, when a transfer end interrupt by the transfer counter, the ESIF bit is set to 1.

Figure 10.39 is a block diagram of interrupts and interrupt flags. To clear an interrupt, clear the DTIF or ESIF bit in DMDR to 0 in the interrupt handling routine or continue the transfer by setting the DTE bit in DMDR after setting the register. Figure 10.40 shows procedure to restart the transfer by clearing an interrupt.

Extended repeat area overflow occurs in destination address

Figure 10.39 Interrupt and Interrupt Sources

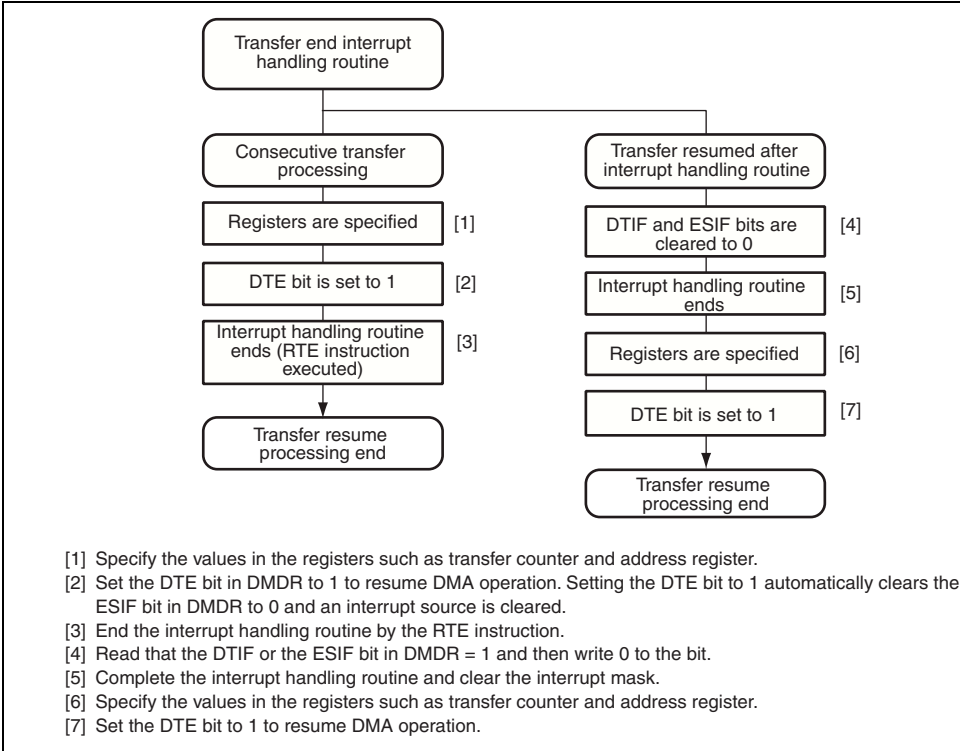


Figure 10.40 Procedure Example of Resuming Transfer by Clearing Interrupt

enters the module stop state. However, when a transfer for a channel is enabled or when an 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 register settings are valid in the module stop state. Disable them before entering the module stop state, if necessary.

- TEND bit in DMDR is 1 (the $\overline{\text{TEND}}$ signal output enabled)
- DACK bit in DMDR is 1 (the $\overline{\text{DACK}}$ signal output enabled)

3. Activation by $\overline{\text{DREQ}}$ Falling Edge

The $\overline{\text{DREQ}}$ falling edge detection is synchronized with the DMAC internal operation.

- A. Activation request waiting state: Waiting for detecting the $\overline{\text{DREQ}}$ low level. A transition to 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 transition to 1. is made.

After a DMAC transfer enabled, a transition to 1. is made. Therefore, the $\overline{\text{DREQ}}$ signal is 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 of the setting of $\overline{\text{DREQ}}$ falling edge or low level detection. Therefore, if the $\overline{\text{DREQ}}$ signal is high 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.

11.1 Features

- Up to 4-Gbyte address space accessible
- Selection of byte, word, or longword transfer data length
- Total transfer size of up to 4 Gbytes (4,294,967,295 bytes)
Selection of free-running mode (with no total transfer size specified)
- Selection of auto-requests or external requests for activating the EXDMAC
Auto-request: Activation from the CPU (Cycle steal mode or burst mode can be selected)
External request: Low level sensing or falling edge sensing for the $\overline{\text{EDREQ}}$ signal can be selected.
All of four channels can accept external requests.
- Selection of dual address mode or single address mode
Dual address mode: Both the transfer source and destination addresses are specified to transfer data.
Single address mode: The $\overline{\text{EDACK}}$ signal is used to access the transfer source or destination peripheral device and the address of the other device is specified to transfer data.
- Normal, repeat, block, or cluster transfer (only for the EXDMAC) can be selected as transfer mode
Normal transfer mode: One byte, one word, or one longword data is transferred at a single transfer request
Repeat transfer mode: One byte, one word, or one longword data is transferred at a single transfer request
Repeat size of data is transferred and then a transfer address is specified to start the transfer
the transfer start address
Up to 64-kbyte transfers can be set as repeat size (65,536 bytes/words/longwords)
Block transfer mode: One block data is transferred at a single transfer request

addition

When offset addition is used to update addresses, the mid-addresses can be skipped during transfer.

- Transfer of word or longword data to addresses beyond each data boundary
Data can be divided into an optimal data size (byte or word) according to addresses when transferring data.
- Two kinds of interrupts requested to the CPU
Transfer end interrupt: Requested after the number of data set by the transfer counter is completely transferred
Transfer escape end interrupt: Requested when the remaining transfer size is smaller than the size set for a single transfer request, after a repeat-size transfer is completed, or when an extended repeat area overflow occurs.
- Acceptance of a transfer request can be reported to an external device via the $\overline{\text{EDRACK}}$ (only for the EXDMAC).
- Operation of EXDMAC, connected to a dedicated bus, in parallel with a bus master such as CPU, DTC, or DMAC (only for the EXDMAC).
- Module stop state can be set.

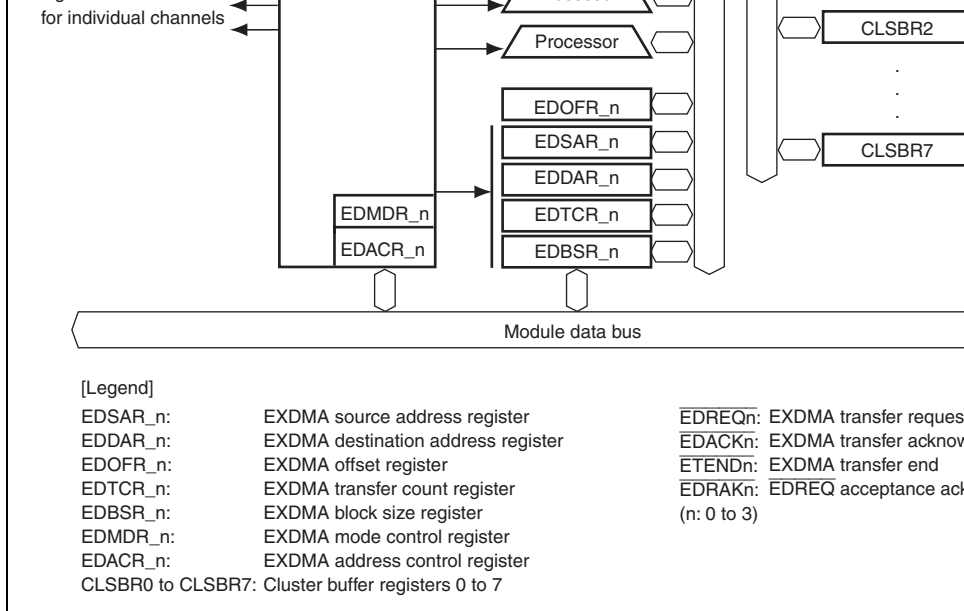


Figure 11.1 Block Diagram of EXDMAC

	$\overline{\text{EDREQ0}}$ acceptance acknowledge	$\overline{\text{EDRAK0}}$	Output	Notification to external de channel 0 external requ acceptance and start of e
1	EXDMA transfer request 1	$\overline{\text{EDREQ1}}$	Input	Channel 1 external requ
	EXDMA transfer acknowledge 1	$\overline{\text{EDACK1}}$	Output	Channel 1 single address acknowledge
	EXDMA transfer end 1	$\overline{\text{ETEND1}}$	Output	Channel 1 transfer end
	$\overline{\text{EDREQ1}}$ acceptance acknowledge	$\overline{\text{EDRAK1}}$	Output	Notification to external de channel 1 external requ acceptance and start of e
2	EXDMA transfer request 2	$\overline{\text{EDREQ2}}$	Input	Channel 2 external requ
	EXDMA transfer acknowledge 2	$\overline{\text{EDACK2}}$	Output	Channel 2 single address acknowledge
	EXDMA transfer end 2	$\overline{\text{ETEND2}}$	Output	Channel 2 transfer end
	$\overline{\text{EDREQ2}}$ acceptance acknowledge	$\overline{\text{EDRAK2}}$	Output	Notification to external de channel 2 external requ acceptance and start of e
3	EXDMA transfer request 3	$\overline{\text{EDREQ3}}$	Input	Channel 3 external requ
	EXDMA transfer acknowledge 3	$\overline{\text{EDACK3}}$	Output	Channel 3 single address acknowledge
	EXDMA transfer end 3	$\overline{\text{ETEND3}}$	Output	Channel 3 transfer end
	$\overline{\text{EDREQ3}}$ acceptance acknowledge	$\overline{\text{EDRAK3}}$	Output	Notification to external de channel 3 external requ acceptance and start of e

- EXDMA block size register_0 (EDBSR_0)
- EXDMA mode control register_0 (EDMDR_0)
- EXDMA address control register_0 (EDACR_0)

Channel 1

- EXDMA source address register_1 (EDSAR_1)
- EXDMA destination address register_1 (EDDAR_1)
- EXDMA offset register_1 (EDOFR_1)
- EXDMA transfer count register_1 (EDTCR_1)
- EXDMA block size register_1 (EDBSR_1)
- EXDMA mode control register_1 (EDMDR_1)
- EXDMA address control register_1 (EDACR_1)

Channel 2

- EXDMA source address register_2 (EDSAR_2)
- EXDMA destination address register_2 (EDDAR_2)
- EXDMA offset register_2 (EDOFR_2)
- EXDMA transfer count register_2 (EDTCR_2)
- EXDMA block size register_2 (EDBSR_2)
- EXDMA mode control register_2 (EDMDR_2)
- EXDMA address control register_2 (EDACR_2)

- Cluster buffer registers 0 to 7 (CLSBR0 to CLSBR7)

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

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	R
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	R
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	R
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	R

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

EDTCR can be read at all times by the CPU. When reading EDTCR for a channel on which EXDMA transfer processing is in progress, a longword-size read must be executed. Do not write to EDTCR for a channel on which EXDMA transfer is in progress.

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	

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	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17
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	All 0	R/W	Sets the repeat size, block size, or cluster size. When these bits are set to H'0001, one byte-, word-, or one longword-size is set. When these bits are set to H'0000, the maximum values are set (see Table 11.2). These bits are always fixed during an EXDMA operation.
15 to 0	BKSZ15 to BKSZ0	All 0	R/W	In an EXDMA operation, the remaining repeat size, block size, or cluster size is indicated. The value is decremented by one each time of a data transfer. When the remaining size becomes zero, the BKSZH16-17 are reloaded. Set the same initial value as for the BKSZH16-17 when writing.

11.3.6 EXDMA Mode Control Register (EDMDR)

EDMDR controls EXDMAC operations.

- EDMDR_0

Bit	31	30	29	28	27	26	25	
Bit Name	DTE	EDACKE	ETENDE	EDRAKE	EDREQS	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	D
Initial Value	0	0	0	0	0	0	0	
R/W	R	R	R	R	R/(W)*	R	R/(W)*	R
Bit	15	14	13	12	11	10	9	
Bit Name	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	D
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R
Bit	7	6	5	4	3	2	1	
Bit Name	DTF1	DTF0	—	—	—	EDMAP2	EDMAP1	ED
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R	R	R/W	R/W	R

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	—	—	—	EDMAP2	EDMAP1
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.

If this bit is cleared to 0 during an EXDMA operation, when the transfer is finished, this bit is cleared to 0 on completion of the currently executing one-block transfer. When this bit is cleared to 0 during an operation in cluster transfer mode, this bit is cleared to 0 on completion of the currently executing one-block transfer.

If an external source that ends (aborts) transfer occurs, this bit is automatically cleared to 0 and transfer is terminated.

Do not change the operating mode, transfer mode, or other parameters while this bit is set to 1.

0: Data transfer disabled

1: Data transfer enabled (during an EXDMA operation)

[Clearing conditions]

- When transfer of the total transfer size specified by the transfer size register ends
- When operation is halted by a repeat size error interrupt
- When operation is halted by an extended address area overflow interrupt
- When operation is halted by a transfer size error interrupt
- When 0 is written to terminate transfer

In block transfer mode, the value written is effective after one-block transfer ends.

In cluster transfer mode, the value written is effective after one-cluster transfer ends.

- When an address error or NMI interrupt occurs
- Reset, hardware standby mode

				0: $\overline{\text{ETEND}}$ pin output disabled 1: $\overline{\text{ETEND}}$ pin output enabled
28	EDRAKE	0	R/W	$\overline{\text{EDRAK}}$ Pin Output Enable Enables or disables output from the $\overline{\text{EDRAK}}$ pin. 0: $\overline{\text{EDRAK}}$ pin output disabled 1: $\overline{\text{EDRAK}}$ pin output enabled
27	EDREQS	0	R/W	$\overline{\text{EDREQ}}$ Select Selects whether a low level or the falling edge of the $\overline{\text{EDREQ}}$ signal used in external request mode is detected. 0: Low-level detection 1: Falling edge detection (the first transfer is completed on a low level after a transfer is enabled.)
26	NRD	0	R/W	Next Request Delay Selects the timing of the next transfer request to be accepted. 0: Next transfer request starts to be accepted when the transfer of the bus cycle in progress ends. 1: Next transfer request starts to be accepted one bus cycle of $B\phi$ from the completion of the bus cycle in progress.
25, 24	—	All 0	R	Reserved They are always read as 0 and cannot be modified.

19	ERRF	0	R(W)*	System Error Flag Flag that indicates the occurrence of an address error or NMI interrupt. This bit is only enabled in EDMA5 channels when the DTE bit for channels is disabled. This bit is reserved in EDMA5 channels 0 to EDMDR_3. They are always read as 0 and cannot be modified. 0: Address error or NMI interrupt is not generated 1: Address error or NMI interrupt is generated [Clearing condition] <ul style="list-style-type: none"> Writing 0 to ERRF after reading ERRF = 1 [Setting condition] <ul style="list-style-type: none"> When an address error or NMI interrupt occurs However, when an address error or an NMI interrupt has been generated in EXDMAC module stop mode, this bit is not set to 1.
18	—	0	R	Reserved They are always read as 0 and cannot be modified.

				<ul style="list-style-type: none"> • Writing 1 to the DTE bit <p>[Setting conditions]</p> <ul style="list-style-type: none"> • Transfer size error interrupt request is generated • Repeat size end interrupt request is generated • Extended repeat area overflow end interrupt request is generated
16	DTIF	0	R/(W)*	<p>Data Transfer Interrupt Flag</p> <p>Flag indicating that a transfer end interrupt request occurred by the transfer counter.</p> <p>0: Transfer end interrupt request is not generated by the transfer counter</p> <p>1: Transfer end interrupt request is generated by the transfer counter</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> • Writing 1 to the DTE bit • Writing 0 to DTIF while reading DTIF = 1 <p>[Setting condition]</p> <ul style="list-style-type: none"> • When EDTCR becomes 0 and transfer has completed
15	DTSZ1	0	R/W	Data Access Size 1 and 0
14	DTSZ0	0	R/W	<p>Selects the data access size.</p> <p>00: Byte-size (8 bits)</p> <p>01: Word-size (16 bits)</p> <p>10: Longword-size (32 bits)</p> <p>11: Setting prohibited</p>

request.

When this bit is set to 1 and the transfer count becomes smaller than the data access size for transfer request by EXDMAC transfer, the DTE is cleared to 0 by the next transfer request. At the same time, the ESIF bit is set to 1 to indicate that a transfer size error interrupt request is generated.

When cluster transfer read/write address mode is specified, this bit should be set to 1.

Transfer size error interrupt request occurs in the following conditions:

- In normal transfer and repeat transfer mode, the total transfer size set in EDTCR is smaller than the data access size
- In block transfer mode, the total transfer size set in EDTCR is smaller than the block size
- In cluster transfer mode, the total transfer size set in EDTCR is smaller than the cluster size

0: Transfer size error interrupt request disabled

1: Transfer size error interrupt request enabled

10	—	0	R	Reserved
----	---	---	---	----------

They are always read as 0 and cannot be modified.

8	DTIE	0	R/W	Data Transfer Interrupt Enable Enables or disables a transfer end interrupt request to the transfer counter. When this bit is set to 1 and the DTIF bit is set to 1, a transfer end interrupt is generated to the CPU or DTC. The transfer end interrupt is canceled by clearing this bit or the DTIF bit. 0: Transfer end interrupt request disabled 1: Transfer end interrupt request enabled
7	DTF1	0	R/W	Data Transfer Factor 1 and 0
6	DTF0	0	R/W	Selects a source to activate EXDMAC. For external requests, a sampling method is selected by the EDREQS bit. 00: Auto-request (cycle steal mode) 01: Auto-request (burst mode) 10: Setting prohibited 11: External request
5	—	0	R/W	Reserved The initial value should not be changed.

independently for each channel. This bit is enabled when the CPUPCE bit in CPUPCR is 1.

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)

Note: * Only 0 can be written to these bits after 1 is read to clear the flag.

R/W	R	R	R/W	R/W	R	R	R/W
Bit	15	14	13	12	11	10	9
Bit Name	SARIE	—	—	SARA4	SARA3	SARA2	SARA1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R	R	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1
Bit Name	DARIE	—	—	DARA4	DARA3	DARA2	DARA1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
31	AMS	0	R/W	<p>Address Mode Select</p> <p>Selects single address mode or dual address mode. When single address mode is selected, $\overline{\text{EDACK}}$ is not valid due to the EDACKE bit setting in EDMDR.</p> <p>0: Dual address mode 1: Single address mode</p>
30	DIRS	0	R/W	<p>Single Address Direction Select</p> <p>Specifies the data transfer direction in single address mode. In dual address mode, the specification of this bit is ignored.</p> <p>In cluster transfer mode, the internal cluster bus can be the source or destination in place of the external device with $\overline{\text{DACK}}$.</p> <p>0: EDSAR transferred as a source address 1: EDDAR transferred as a destination address</p>

Even if the repeat area is not specified (ARS1, B'10), the repeat size end interrupt can be requested at the end of a repeat-size transfer.

When this bit is set to 1 and the next transfer is generated at the end of a block- or cluster-size transfer in block transfer or cluster transfer mode, the DEDMDR is cleared to 0. At the same time, the REPEATEND bit in EDMDR is set to 1 to indicate that the repeat size end interrupt is requested.

0: Repeat size end interrupt request disabled

1: Repeat size end interrupt request enabled

25	ARS1	0	R/W	Area Select 1 and 0
24	ARS0	0	R/W	Select the block area or repeat area in block transfer, repeat transfer or cluster transfer mode. 00: Block area/repeat area on the source address side 01: Block area/repeat area on the destination address side 10: Block area/repeat area not specified 11: Setting prohibited
23, 22	—	All 0	R	Reserved They are always read as 0 and cannot be modified.

				11: Decrement (−1, −2, or −4 according to access size)
19, 18	—	All 0	R	Reserved They are always read as 0 and cannot be modified.
17	DAT1	0	R/W	Destination Address Update Mode 1 and 0
16	DAT0	0	R/W	These bits specify incrementing/decrementing transfer destination address (EDDAR). When transfer source is not specified in EDDAR in source address mode, the specification by these bits is ignored. 00: Fixed 01: Offset added 10: Incremented (+1, +2, or +4 according to transfer access size) 11: Decrement (−1, −2, or −4 according to transfer access size)

When used together with block transfer mode, interrupt is requested at the end of a block-size transfer. If the DTE bit is set to 1 in EDMDR for the channel which transfer is terminated by an interrupt, transfer is resumed from the state in which it ended.

If a source address extended repeat area is not designated, the specification by this bit is ignored.

0: Source address extended repeat area overflow interrupt request disabled

1: Source address extended repeat area overflow interrupt request enabled

14, 13	—	All 0	R	Reserved They are always read as 0 and cannot be modified.
12	SARA4	0	R/W	Source Address Extended Repeat Area
11	SARA3	0	R/W	These bits specify the source address (EDSAR) of the extended repeat area. The extended repeat area function updates the specified lower address bits, leaving the remaining upper address bits always the same. An extended repeat area size of 4 bytes to 16 Mbytes can be specified. The setting interval is a power-of-two number of bytes. When extended repeat area overflow results from incrementing or decrementing an address, the start address is the start address of the extended repeat area in the case of address incrementing, or the end address of the extended repeat area in the case of address decrementing. If SARIE bit is set to 1, an interrupt can be requested when an extended repeat area overflow occurs.
10	SARA2	0	R/W	
9	SARA1	0	R/W	
8	SARA0	0	R/W	

Table 11.3 shows the settings and ranges of the extended repeat area.

When used together with block transfer mode, an interrupt is requested at the end of a block-size transfer. If DTE bit is set to 1 in EDMDR for the channel, which transfer is terminated by an interrupt, transfer can be resumed from the state in which it ended. If the destination address extended repeat area is not designated, the specification by this bit is ignored.

0: Destination address extended repeat area interrupt request disabled

1: Destination address extended repeat area interrupt request enabled

6, 5	—	All 0	R	Reserved They are always read as 0 and cannot be modified.
4	DARA4	0	R/W	Destination Address Extended Repeat Area Enable
3	DARA3	0	R/W	These bits specify the destination address (E) of the extended repeat area. The extended repeat area function updates the specified lower address bits, leaving the remaining upper address bits always the same. An extended repeat area size of 4 bytes to 128 Mbytes can be specified. The setting interval is a power-of-two of bytes. When extended repeat area overflow results from incrementing or decrementing an address, the start address is the start address of the extended repeat area in the case of address incrementing, or the end address of the extended repeat area in the case of address decrementing. If the DARIE bit is set to 1, an interrupt can be requested when an extended repeat area overflow occurs. Table 11.3 shows the settings and ranges of the extended repeat area.
2	DARA2	0	R/W	
1	DARA1	0	R/W	
0	DARA0	0	R/W	

00110	Lower 6 bit (64-byte area) designated as extended repeat
00111	Lower 7 bit (128-byte area) designated as extended repeat
01000	Lower 8 bit (256-byte area) designated as extended repeat
01001	Lower 9 bit (512-byte area) designated as extended repeat
01010	Lower 10 bit (1-kbyte area) designated as extended repeat
01011	Lower 11 bit (2-kbyte area) designated as extended repeat
01100	Lower 12 bit (4-kbyte area) designated as extended repeat
01101	Lower 13 bit (8-kbyte area) designated as extended repeat
01110	Lower 14 bit (16-kbyte area) designated as extended repeat
01111	Lower 15 bit (32-kbyte area) designated as extended repeat
10000	Lower 16 bit (64-kbyte area) designated as extended repeat
10001	Lower 17 bit (128-kbyte area) designated as extended repeat
10010	Lower 18 bit (256-kbyte area) designated as extended repeat
10011	Lower 19 bit (512-kbyte area) designated as extended repeat
10100	Lower 20 bit (1-Mbyte area) designated as extended repeat
10101	Lower 21 bit (2-Mbyte area) designated as extended repeat
10110	Lower 22 bit (4-Mbyte area) designated as extended repeat
10111	Lower 23 bit (8-Mbyte area) designated as extended repeat
11000	Lower 24 bit (16-Mbyte area) designated as extended repeat
11001	Lower 25 bit (32-Mbyte area) designated as extended repeat
11010	Lower 26 bit (64-Mbyte area) designated as extended repeat
11011	Lower 27 bit (128-Mbyte area) designated as extended repeat
111XX	Setting prohibited

[Legend] X: Don't care

In cluster transfer mode, the same CLSBR is used for all channels. When the CPU writes to CLSBR conflicts with cluster transfer, the contents of transferred data are not guaranteed. In cluster transfer read/write address mode is specified and if another channel is set for cluster transfer, the transferred data may be overwritten.

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

Mode	Transfer mode	Activation source	Common function	Source	Destination
Dual address mode	<ul style="list-style-type: none"> Normal transfer mode Repeat transfer mode Block transfer mode (Repeat size/ block size = 1 to 65,536 bytes/ word/longword)	<ul style="list-style-type: none"> Auto-request (activated by the CPU) External request 	<ul style="list-style-type: none"> Total transfer size: 1 to 4 Gbytes, or no specification Offset addition Extended repeat area function 	EDSAR	ED
Single address mode	<ul style="list-style-type: none"> Direct data transfer to/from external devices using EDACK pin instead of source or destination address register Above transfer mode can be specified in addition to address register setting One transfer possible in one bus cycle (Transfer mode variations are the same as in dual address mode.)			EDSAR/ EDACK	ED ED

When the activation source is an auto-request, cycle steal mode or burst mode can be selected.

When the total transfer size is not specified (EDTCR = H'00000000), the transfer counter is not incremented and the transfer count is not restricted, allowing continuous transfer.

Dual address mode	(activated by the CPU)	One access size (byte/word/longword) to 32 bytes		the transfer source and written to the transfer destination
	• External request	• Total transfer size		
Cluster transfer		1 to 4 Gbytes, or no specification	EDSAR	Read from the transfer source
Read address mode (DIRS = 0)		• Offset addition		
Cluster transfer		• Extended repeat area function	—	Written to the transfer destination
Write address mode (DIRS = 1)				

In cluster transfer mode, the specified cluster size is transferred in response to a single transfer request. The cluster size can be from one access size (byte, word, or longword) to 32 bytes. With a cluster, a cluster-size transfer is performed in burst transfer mode. With a cluster-size transfer, a cluster transfer mode (dual address mode), block transfer mode (dual address mode) is used.

With auto-requests, cycle steal mode is set.

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

These consecutive read and write cycles are indivisible: another bus cycle (external access by another bus master, refresh cycle, or external bus release cycle) does not occur between the two cycles.

$\overline{\text{ETEND}}$ pin output can be enabled or disabled by means of the ETENDE bit in EDMDR. $\overline{\text{ETEND}}$ is output for two consecutive bus cycles. When an idle cycle is inserted before the bus cycle, $\overline{\text{ETEND}}$ signal is also output in the idle cycle. The $\overline{\text{EDACK}}$ signal is not output.

Figure 11.2 shows an example of the timing in dual address mode and figure 11.3 shows address mode operation.

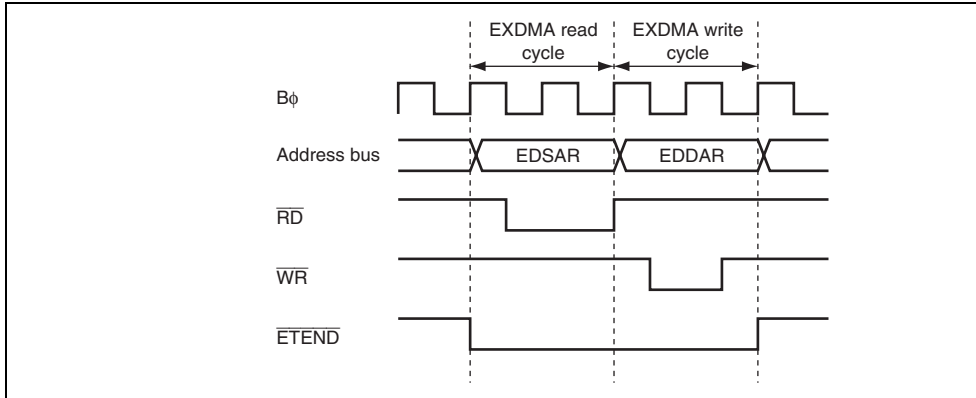


Figure 11.2 Example of Timing in Dual Address Mode

In single address mode, the $\overline{\text{EDACK}}$ pin is used instead of EDSAR or EDDAR to transfer data directly between an external device and external memory. One transfer operation is executed in one bus cycle.

In this mode, the data bus width must be the same as the data access size. For details on data bus width, see section 9, Bus Controller (BSC).

In this mode, the EXDMAC accesses the transfer source or transfer destination external device by outputting the strobe signal $\overline{\text{EDACK}}$ for the external device with $\overline{\text{DACK}}$, and at the same time, the EXDMAC accesses the other external device in the transfer by outputting an address. In this way, one transfer can be executed in one bus cycle. In the example of transfer between external memory and an external device with $\overline{\text{DACK}}$ shown in figure 11.4, data is output to the data bus by the external device and written to external memory in the same bus cycle.

The transfer direction, that is whether the external device with $\overline{\text{DACK}}$ is the transfer source or transfer destination, can be specified with the DIRS bit in EDACR. Transfer is performed from external memory (EDSAR) to the external device with $\overline{\text{DACK}}$ when DIRS = 0, and from the external device with $\overline{\text{DACK}}$ to the external memory (EDDAR) when DIRS = 1. The setting of the source or destination address register not used in the transfer is ignored.

The $\overline{\text{EDACK}}$ pin output is valid by the setting of EDACKE bit in EDMDR when single address mode is selected. The $\overline{\text{EDACK}}$ pin output is active-low.

$\overline{\text{ETEND}}$ pin output can be enabled or disabled by means of the ETENDE bit in EDMDR. $\overline{\text{ETEND}}$ is output for one bus cycle. When an idle cycle is inserted before the bus cycle, the $\overline{\text{ETEND}}$ pin is also output in the idle cycle.

Figure 11.5 shows an example of the timing in single address mode and figure 11.6 shows an example of single address mode operation.

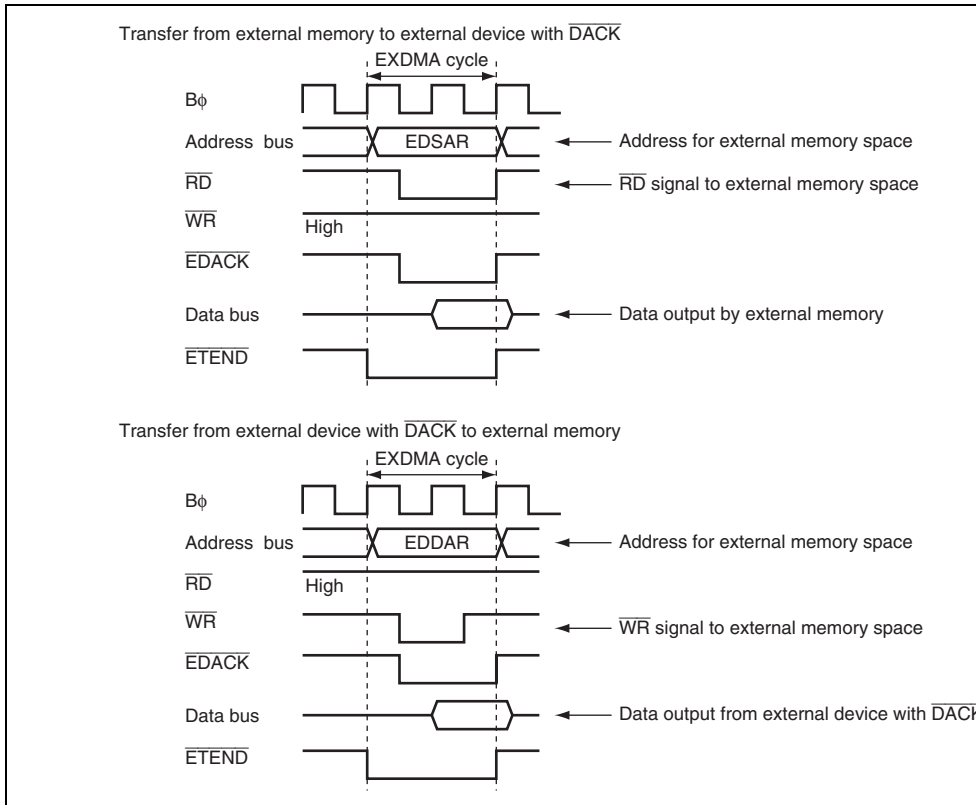


Figure 11.5 Example of Timing in Single Address Mode

11.5.2 Transfer Modes

(1) Normal Transfer Mode

In normal transfer mode, transfer of one data access size unit is processed in response to transfer request. The total transfer size of up to 4 Gbytes can be set by EDTCR. EDBSR in normal transfer mode.

The $\overline{\text{ETEND}}$ signal is output only for the last EXDMA transfer. The $\overline{\text{EDRAK}}$ signal is output each time a transfer request is accepted and transfer processing is started.

Figure 11.7 shows examples of transfer timing in normal transfer mode and figure 11.8 shows normal transfer mode operation in dual address mode.

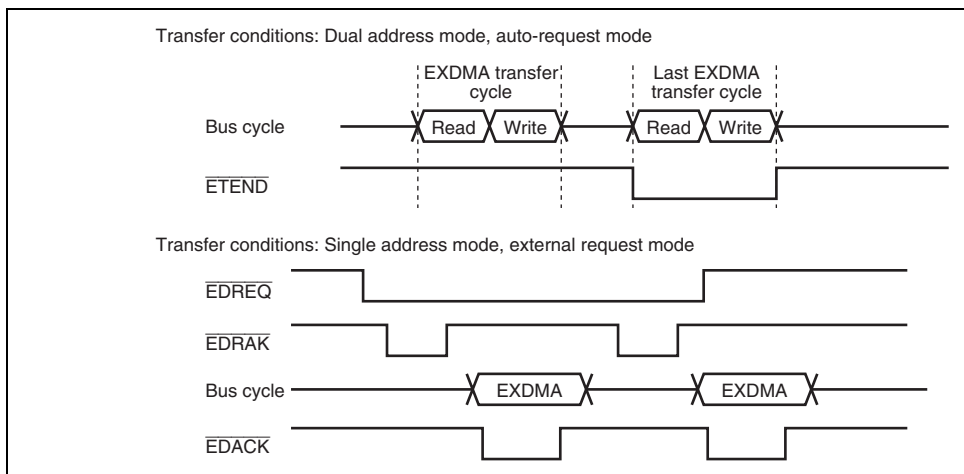


Figure 11.7 Examples of Timing in Normal Transfer Mode

In repeat transfer mode, transfer of one data access size unit is processed in response to one transfer request. The total transfer size of up to 4 Gbytes can be set by EDTCR. The repeat size of up to 64 kbytes \times data access size can be set by EDBSR.

The ARS1 and ARS0 bits in EDACR specify the repeat area on the source address or destination address side. The address specified for the repeat area is restored to the transfer start address at the end of a repeat-size transfer. This operation continues until transfer of total transfer size specified by EDTCR ends. EDTCR specified with H'00000000 is assumed as free-running mode and the transfer continues until the DTE bit in EDMDR is cleared to 0.

At the end of a repeat-size transfer, the EXDMA transfer is halted temporarily and a repeat end interrupt is requested to the CPU or DTC. When the RPTIE bit in EDACR is set to 1, the next transfer request is generated at the end of a repeat-size transfer, the ESIF bit in EDMDR is set to 1 and the DTE bit in EDMDR is cleared to 0 to terminate the transfer. At this time, a repeat end interrupt is requested to the CPU or DTC when the ESIE bit in EDMDR is set to 1.

The timing of EXDMA transfer including the $\overline{\text{ETEND}}$ or $\overline{\text{EDRAK}}$ output is the same as for normal transfer mode.

Figure 11.9 shows the repeat transfer mode operation in dual address mode.

The operation without specifying a repeat area on the source or destination address side is the same as for the normal transfer mode operation shown in figure 11.8. In this case, a repeat end interrupt can also be generated at the end of a repeat-size transfer.

(3) Block Transfer Mode

In block transfer mode, transfer of one block size unit is processed in response to one transfer request. The total transfer size of up to 4 Gbytes can be set by EDTCR. The block size of $n \times \text{kbytes} \times \text{data access size}$ can be set by EDBSR.

A transfer request from another channel is held pending during one block transfer. When the block transfer is completed, the bus mastership is released for another bus master.

A block area can be specified by the ARS1 or ARS0 bit in EDACR on the source or destination address side. The address specified for the block area is restored to the transfer start address when one-block transfer completes. When no repeat area is specified on the source and destination address sides, the address is not restored to the transfer start address and the operation proceeds to the next sequence. A repeat size end interrupt can be generated.

The $\overline{\text{ETEND}}$ signal is output for each block transfer in the EXDMA transfer cycle in which the block ends. The $\overline{\text{EDRAK}}$ signal is output once for one transfer request (for transfer of one block).

Caution is required when setting the extended repeat area overflow interrupt in block transfer mode. For details, see section 11.5.5, Extended Repeat Area Function.

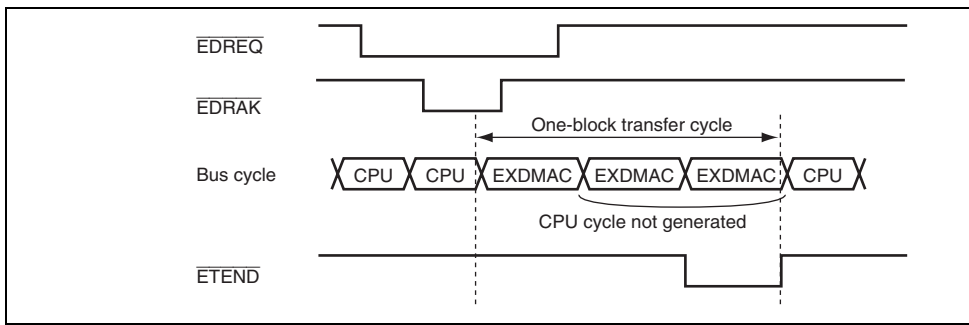


Figure 11.10 Example of Block Transfer Mode

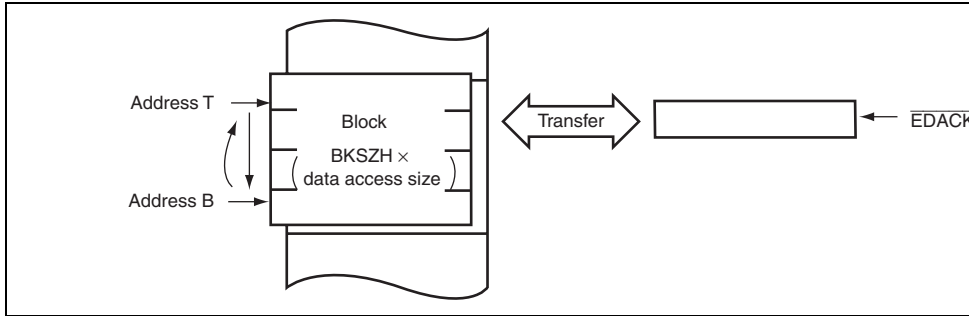


Figure 11.11 Block Transfer Mode Operation in Single Address Mode (with Block Area Specified)

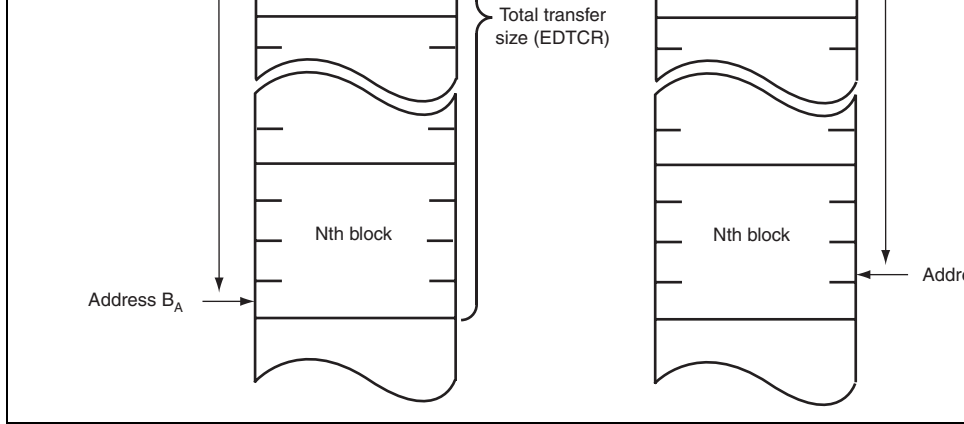


Figure 11.12 Block Transfer Mode Operation in Dual Address Mode (without BL Specified)

request activation. The bus mode can be selected from cycle steal mode and burst mode via request activation.

(2) Activation by External Request

Transfer is started by the transfer request signal ($\overline{\text{EDREQ}}$) from the external device for activation by an external request. When the EXDMA transfer is enabled ($\text{DTE} = 1$), the EXDMA transfer starts by $\overline{\text{EDREQ}}$ input.

The transfer request signal is accepted by the $\overline{\text{EDREQ}}$ pin. The EDREQS bit in EDMDR selects whether the $\overline{\text{EDREQ}}$ is detected by falling edge sensing or low level sensing.

When the EDRAKE bit in EDMDR is set to 1, the signal notifying transfer request acceptance is output from the $\overline{\text{EDRAK}}$ pin. The $\overline{\text{EDRAK}}$ signal is accepted for one external request and output when transfer processing starts.

When specifying an external request as an activation source, set the DDR bit to 0 and the DTR to 1 on the corresponding pin in advance. For details, see section 13, I/O Ports.

transfer request, the EXDMAC takes back the bus mastership, performs another transfer, and then releases the bus mastership again at the end of the transfer. This process is repeated until the transfer end condition is satisfied.

If a transfer request occurs in another channel during EXDMA transfer, the bus mastership is temporarily released for another bus master, then transfer is performed on the channel for which the transfer request was issued. For details on the operation when there are transfer requests in multiple channels, see section 11.5.8, Channel Priority Order.

Figure 11.13 shows an example of the timing in cycle steal mode. The transfer condition is as follows:

- Address mode: Single address mode
- Sampling method on the $\overline{\text{EDREQ}}$ pin: Low level sensing
- CPU internal bus master is operating in external space

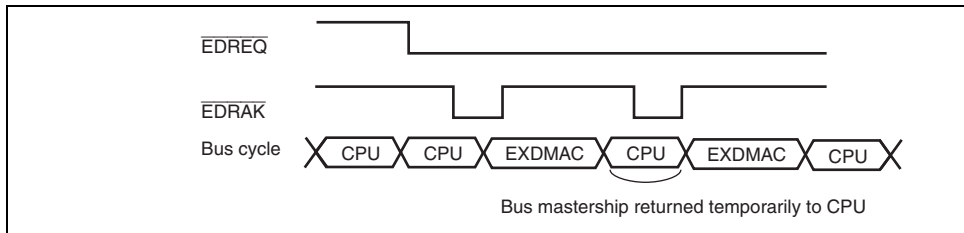


Figure 11.13 Example of Timing in Cycle Steal Mode

In block transfer mode and cluster transfer mode, the setting of burst mode is invalid (one-cluster transfer is processed in the same way as in burst mode). The EXDMAC always operates in cycle steal mode.

When the DTE bit is cleared to 0 in EDMDR, EXDMA transfer is halted. However, EXDMA transfer is executed for all transfer requests generated within the EXDMAC until the DTE bit is cleared to 0. If a transfer size error interrupt, a repeat size end interrupt, or extended repeat overflow interrupt is generated, the DTE bit is cleared to 0 and transfer is terminated.

Figure 11.14 shows an example of the timing in burst mode.

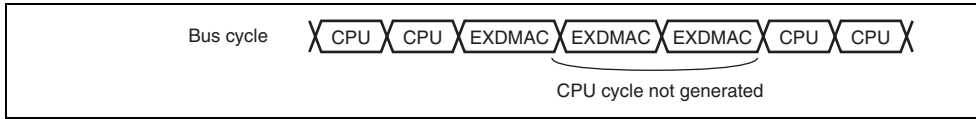


Figure 11.14 Example of Timing in Burst Mode

11.5.5 Extended Repeat Area Function

The EXDMAC has a function for designating an extended repeat area for source address and destination addresses. When an extended repeat area is designated, the address register value repeats within the range specified as the extended repeat area. Normally, when a ring buffer is involved in a transfer, an operation is required to restore the address register value to the start address each time the address register value becomes the last address in the buffer (i.e., ring buffer address overflow occurs). However, if the extended repeat area function is used, an operation that restores the address register value to the buffer start address is processed automatically within the EXDMAC.

The extended repeat area function can be set independently for the source address register (EDSAR) and the destination address register (EDDAR).

Figure 11.15 illustrates the operation of the extended repeat area function.

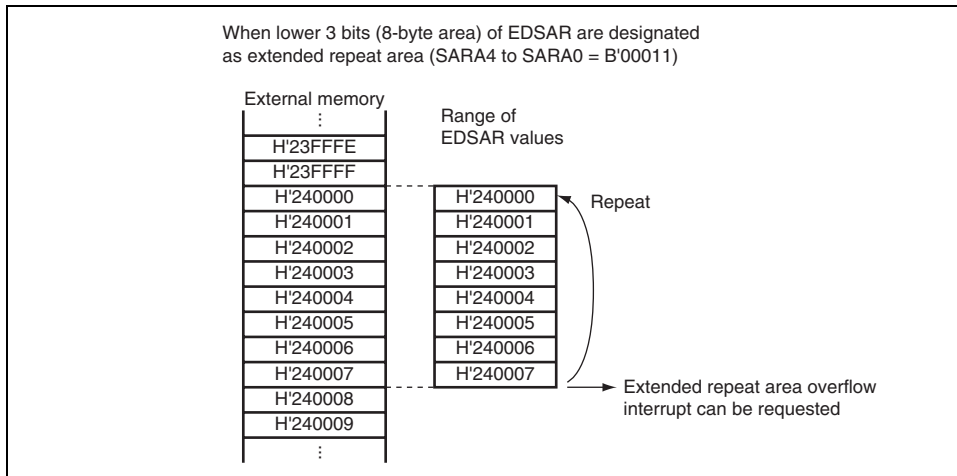


Figure 11.15 Example of Extended Repeat Area Function Operation

Figure 11.16 shows an example in which block transfer mode is used together with the extended repeat area function.

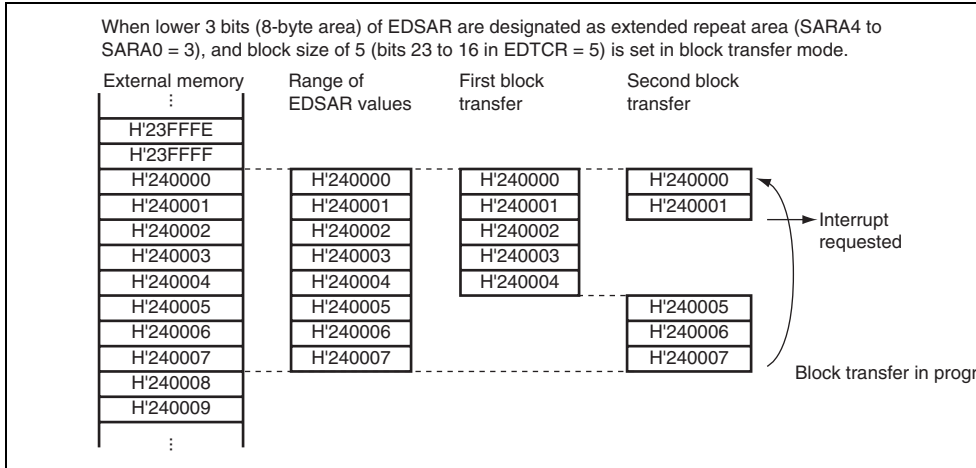


Figure 11.16 Example of Extended Repeat Area Function Operation in Block Transfer Mode

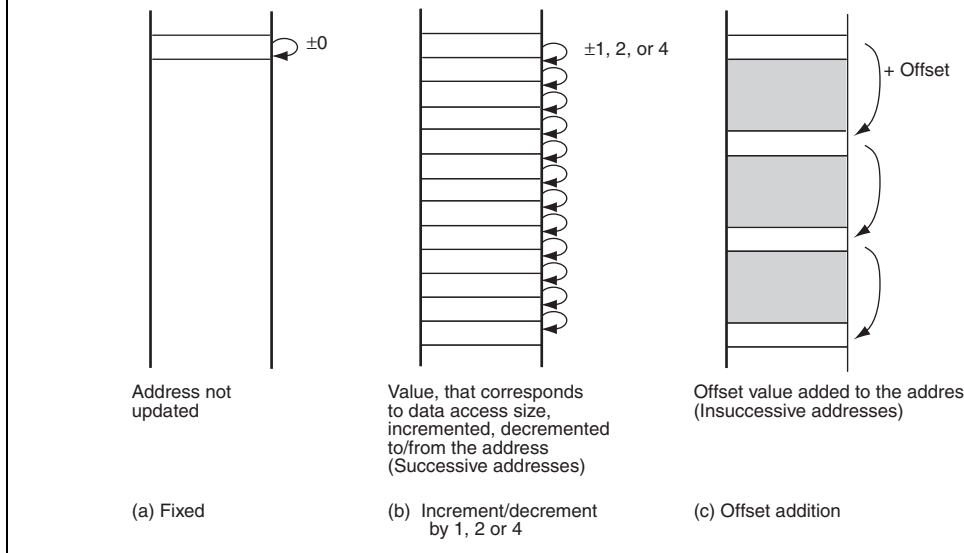


Figure 11.17 Address Update Method

For the fixed method (a), the same address is always indicated without the transfer destination or source address being updated.

For the method of increment/decrement by 1, 2 or 4 (b), the value corresponding to the data access size is incremented or decremented to or from the transfer destination or source address. A byte, word, or longword can be specified for the data access size. The value used for increment or decrement of an address is 1 for a byte-size, 2 for a word-size, and 4 for a longword-size transfer. This function allows continuous address transfer of EXDM.

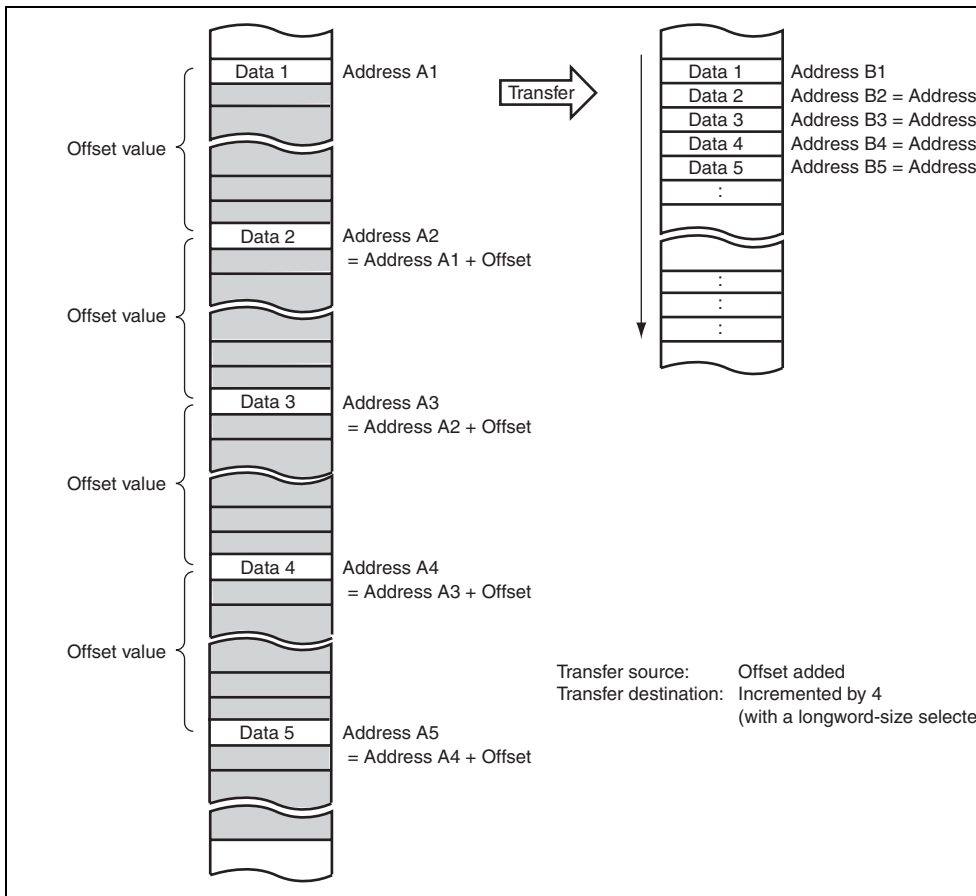


Figure 11.18 Address Update Function Using Offset

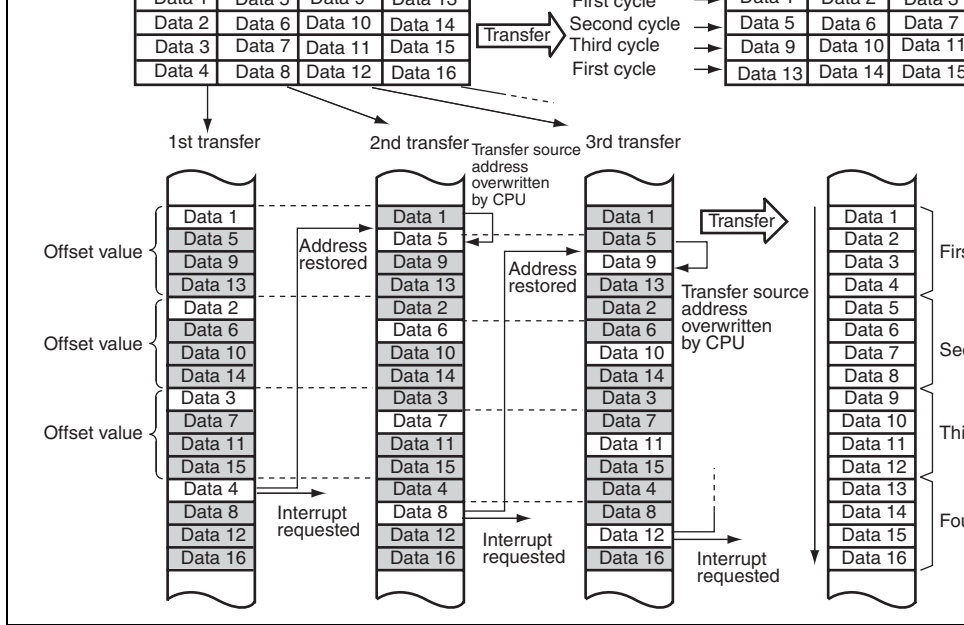


Figure 11.19 XY Conversion by Combining Repeat Transfer Mode and Offset Addressing

In figure 11.19, the source address side is set as a repeat area in EDACR and the offset address is set in EDACR. The offset value is the address that corresponds to $4 \times$ data access size (example: for a longword-size transfer, H'00000010 is specified in EDOFR). The repeat size is $4 \times$ data access size (example: for a longword-size transfer, $4 \times 4 = 16$ bytes are specified as a repeat size). The increment by 1, 2 or 4 is set for the transfer destination. The RPTIE bit in EDACR is used to generate a repeat size end interrupt request at the end of a repeat-size transfer.

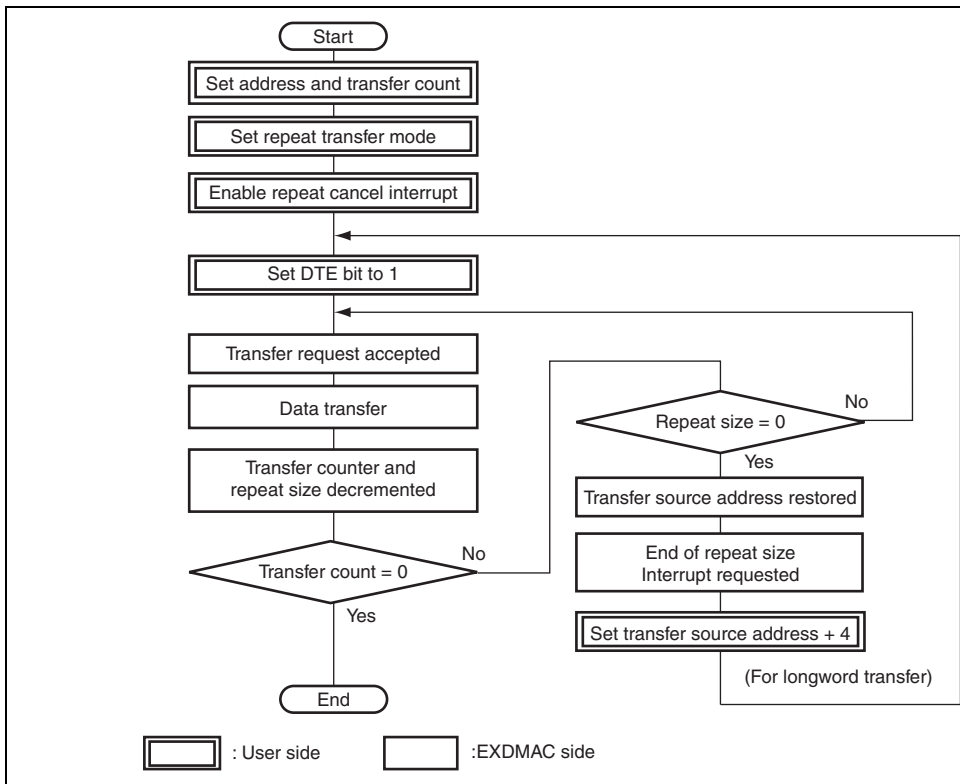


Figure 11.20 Flow of XY Conversion Combining Repeat Transfer Mode and Counter Addition

11.5.7 Registers during EXDMA Transfer Operation

EXDMAC register values are updated as EXDMA transfer processing is performed. The values depend on various settings and the transfer status. The following registers and bits are updated: EDSAR, EDDAR, EDTCR, bits BKSZH and BKSZ in EDBSR, and bits DTE, ERRF, ESIF and DTIF in EDMDR.

(1) EXDMA Source Address Register (EDSAR)

When the EDSAR address is accessed as the transfer source, the EDSAR value is output. EDSAR is updated with the address to be accessed next.

Bits SAT1 and SAT0 in EDACR specify incrementing or decrementing. The address is incremented when SAT1 and SAT0 = B'00, incremented by offset register value when SAT1 and SAT0 = B'01, incremented when SAT1 and SAT0 = B'10, and decremented when SAT1 and SAT0 = B'11. (The increment or decrement value is determined by the data access size.)

The DTSZ1 and DTSZ0 bits in EDMDR set the data access size. When DTSZ1 and DTSZ0 = B'00, the data is byte-size and the address is incremented or decremented by 1. When DTSZ1 and DTSZ0 = B'01, the data is word-size and the address is incremented or decremented by 2. When DTSZ1 and DTSZ0 = B'10, the data is longword-size and the address is incremented or decremented by 4. When a word-size or longword-size is specified but the source address is not aligned to the word or longword boundary, the data is divided into bytes or words for reading. When a byte or longword is divided for reading, the address is incremented or decremented by 1 or 2 to an actual byte-or word-size read. After a word-size or longword-size read, the address is incremented or decremented to or from the read start address according to the setting of SAT0.

(2) EXDMA Destination Address Register (EDDAR)

When the EDDAR address is accessed as the transfer destination, the EDDAR value is output, and then EDDAR is updated with the address to be accessed next.

Bits DAT1 and DAT0 in EDACR specify incrementing or decrementing. The address is incremented when DAT1 and DAT0 = B'00, incremented by offset register value when DAT1 and DAT0 = B'01, incremented when DAT1 and DAT0 = B'10, and decremented when DAT1 and DAT0 = B'11. (The increment or decrement value is determined by the data access size.)

The DTSZ1 and DTSZ0 bits in EDMDR set the data access size. When DTSZ1 and DTSZ0 = B'00, the data is byte-size and the address is incremented or decremented by 1. When DTSZ1 and DTSZ0 = B'01, the data is word-size and the address is incremented or decremented by 2. When DTSZ1 and DTSZ0 = B'10, the data is longword-size and the address is incremented or decremented by 4. When a word-size or longword-size is specified but the destination address is not at the word or longword boundary, the data is divided into bytes or words for writing. When a word or a longword is divided for writing, the address is incremented or decremented by the data access size according to an actual byte- or word-size written. After a word-size or longword-size write, the destination address is incremented or decremented to or from the write start address according to the DTSZ1 and SAT0.

When a block area (repeat area) is set for the destination address in block transfer mode (block transfer mode), the destination address is restored to the transfer start address at the end of the block transfer (repeat-size) transfer and is not affected by address updating.

When an extended repeat area is set for the destination address, the operation conforms to the setting. The upper addresses set for the extended repeat area is fixed, and is not affected by address updating.

EDTCR value does not change.

All of the bits of EDTCR may change, so when EDTCR is read by the CPU during EXDMA transfer, a longword access must be used. During a transfer operation, EDTCR may be updated without regard to accesses from the CPU, and the correct values may not be read if the upper and lower words are read separately. Do not write to EDTCR for a channel on which a transfer operation is in progress.

If there is conflict between an address update associated with EXDMA transfer and a write to the CPU, the CPU write has priority.

In the event of conflict between an EDTCR update from 1, 2, or 4 to 0 and a write (of a value) by the CPU, the CPU write value has priority as the EDTCR value, but transfer is terminated.

(4) EXDMA Block Size Register (EDBSR)

EDBSR is valid in block transfer or repeat transfer mode. EDBSR31 and EDBSR16 are BKSZH and EDBSR15 and EDBSR0 for BKSZ. The 16 bits of BKSZH holds a block size and repeat size and their values do not change. The 16 bits of BKSZ functions as a block size counter, the value of which is decremented by 1 when one data transfer is performed. When the BKSZ value is determined as 0 during EXDMA transfer, the EXDMAC does not store BKSZ and stores the BKSZH value.

The upper 16 bits of EDBSR is never updated, allowing a word-size access.

Do not write to EDBSR for a channel on which a transfer operation is in progress.

- When an extended repeat area overflow interrupt is requested, and transfer ends
- When an NMI interrupt is generated, and transfer halts
- When an address error is generated, and transfer halts
- A reset
- Hardware standby mode
- When 0 is written to the DTE bit, and transfer halts

Writes (except to the DTE bit) are prohibited to registers of a channel for which the DTE bit is 1. When changing register settings after a 0-write to the DTE bit, it is necessary to confirm that the DTE bit has been cleared to 0.

Figure 11.21 shows the procedure for changing register settings in an operating channel.

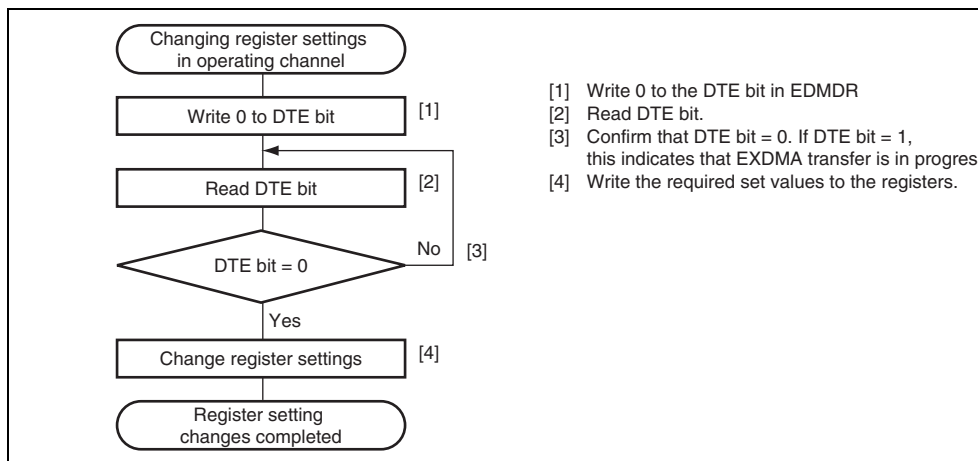


Figure 11.21 Procedure for Changing Register Settings in Operating Channel

the bus cycle in which 0 is written to the DTE bit has been processed. The ACT bit is he
between termination of the last EXDMA cycle and 0-write in the DTE bit.

(7) ERRF bit in EDMDR

This bit specifies termination of transfer by EXDMAC clearing the DTE bit to 0 for all o
an address error or NMI interrupt is generated. The EXDMAC also sets 1 to the ERRF b
EDMDR_0 regardless of the EXDMAC operation to indicate that an address error or NI
interrupt is generated. However, when an address error or an NMI interrupt has been gen
EXDMAC module stop mode, the ERRF bit is not set to 1.

(8) ESIF bit in EDMDR

The ESIF bit in EDMDR is set to 1 when a transfer size interrupt, repeat size end interru
extended repeat area overflow interrupt is requested. When the ESIF bit is set to 1 and th
bit in EDMDR is set to 1, a transfer escape interrupt is requested to the CPU or DTC.

The timing that the ESIF bit is set to 1 is when the EXDMA transfer bus cycle (the sour
interrupt request) terminates, the ACT bit in EDMDR is set to 0, and transfer is terminat

When the DTE bit is set to 1 to resume transfer during interrupt processing, the ESIF bit
automatically cleared to 0 to cancel the interrupt request.

For details on interrupts, see section 11.9, Interrupt Sources.

11.5.8 Channel Priority Order

The priority order of the EXDMAC channels is: channel 0 > channel 1 > channel 2 > channel 3.

Table 11.6 shows the EXDMAC channel priority order.

Table 11.6 EXDMAC Channel Priority Order

Channel	Channel P
Channel 0	High
Channel 1	▲
Channel 2	↑
Channel 3	Low

If transfer requests occur simultaneously for a number of channels, the highest-priority channel according to the priority order is selected for transfer. Transfer starts after the channel in priority releases the bus. If a bus request is issued from another bus master other than EXDMAC during transfer operation, another bus master cycle is initiated.

Channels are not switched during burst transfer, a block-size transfer in block transfer mode or cluster-size transfer in cluster transfer mode.

Figure 11.22 shows an example of the transfer timing when transfer requests occur simultaneously for channels 0, 1, and 2.



Figure 11.22 Example of Channel Priority Timing

11.5.9 Basic Bus Cycles

An example of the basic bus cycle timing is shown in figure 11.23. In this example, word transfer is performed from 16-bit, 2-state access space to 8-bit, 3-state access space. When mastership is transferred from the CPU to the EXDMAC, a source address read and destination address write are performed. The bus is not released in response to another bus request, between these read and write operations. As like CPU cycles, EXDMAC cycles conform to bus controller settings.

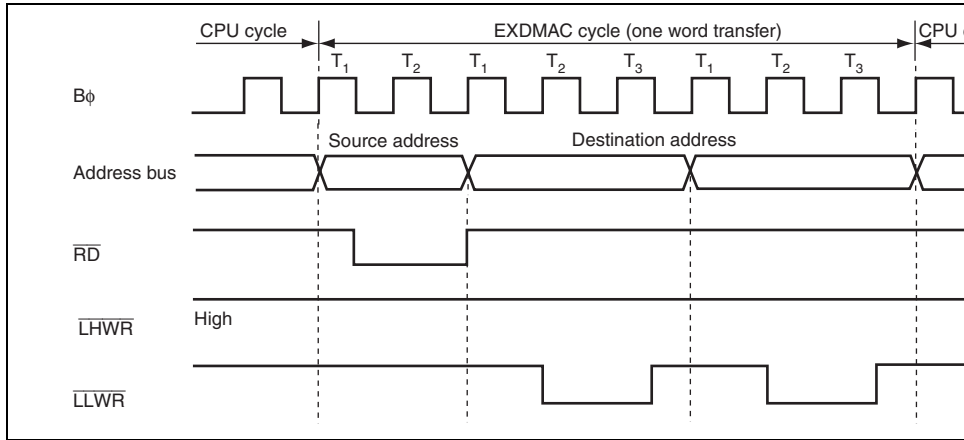


Figure 11.23 Example of EXDMA Transfer Bus Timing

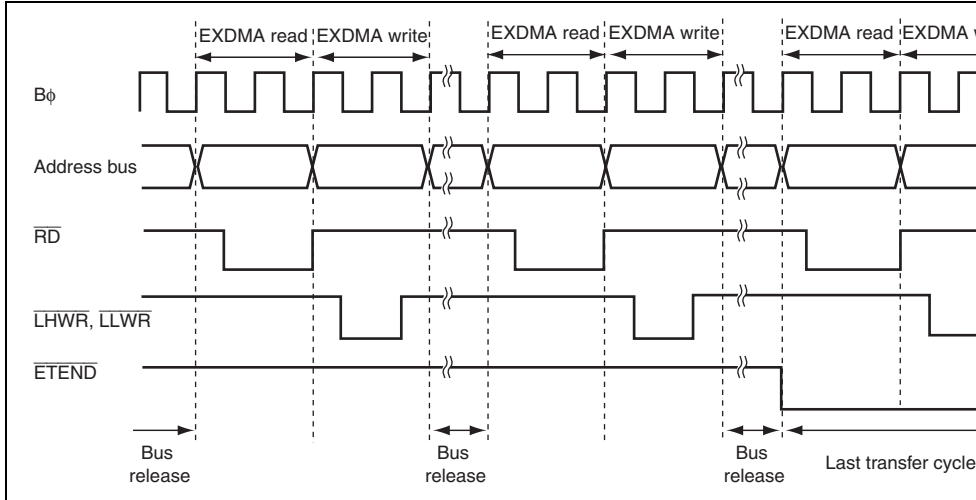


Figure 11.24 Example of Normal Transfer Mode (Cycle Steal Mode) Transfer

Figures 10.25 and 10.26 show examples of transfer when $\overline{\text{ETEND}}$ output is enabled, and longword-size, normal transfer mode (cycle steal mode) is performed from external 16-bit access space to external 16-bit, 2-state access space.

In figure 11.25, the transfer source (SAR) address is not at a longword boundary and the destination (DAR) address is at the longword boundary.

In figure 11.26, the transfer source (SAR) address is at the longword boundary and the destination (DAR) address is not at the longword boundary.

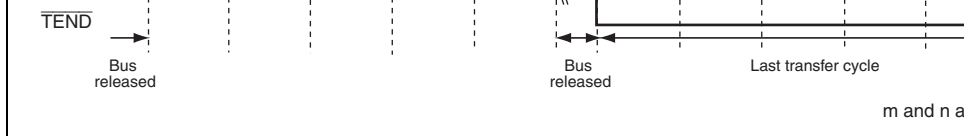


Figure 11.25 Example of Normal Transfer Mode (Cycle Steal Mode) Transfer (Transfer Source EDSAR = Odd Address, Source Address Incremented)

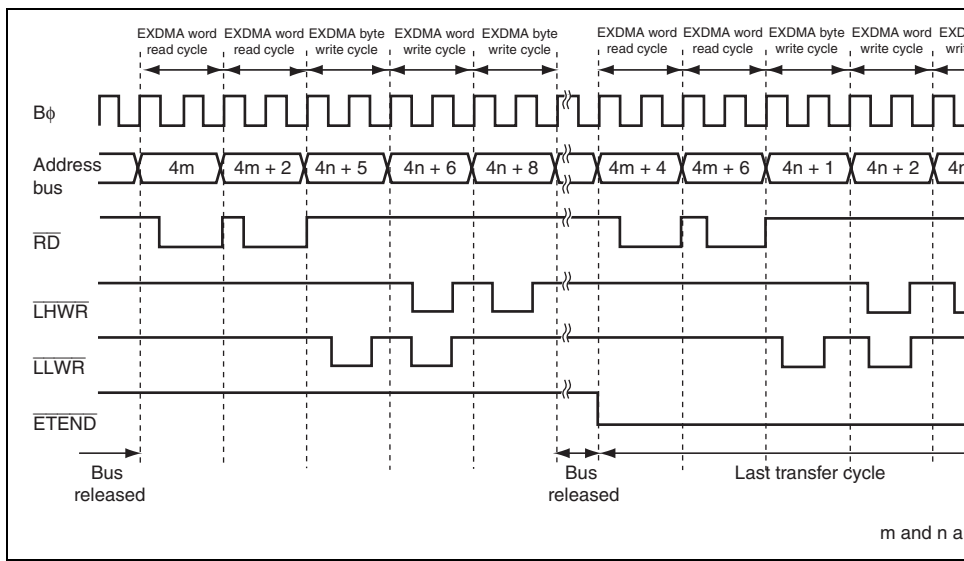


Figure 11.26 Example of Normal Transfer Mode (Cycle Steal Mode) Transfer (Transfer Destination EDDAR = Odd Address, Destination Address Decremented)

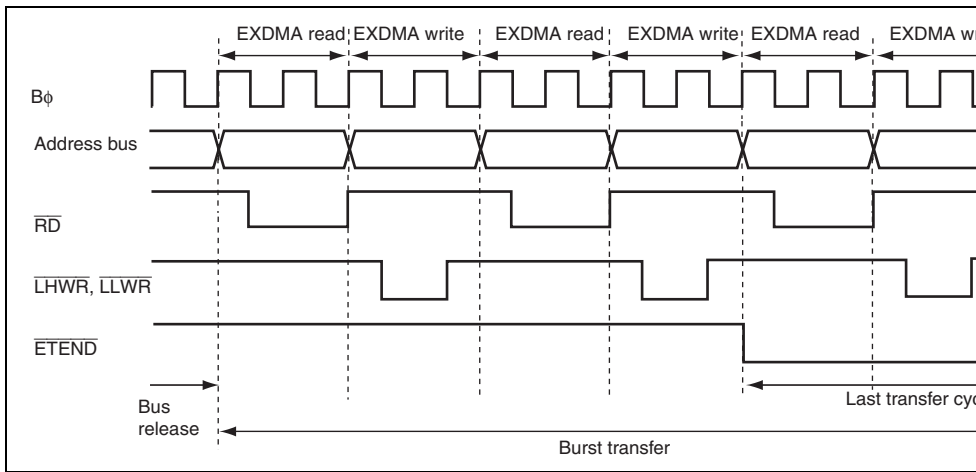


Figure 11.27 Example of Normal Transfer Mode (Burst Mode) Transfer

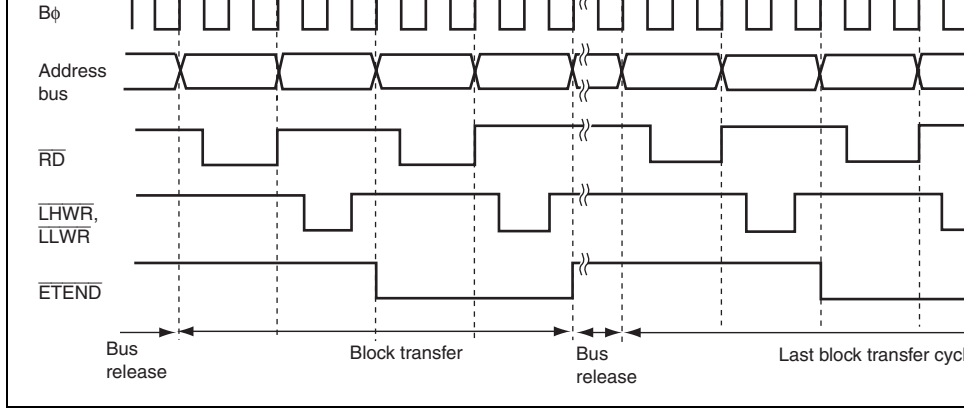


Figure 11.28 Example of Block Transfer Mode Transfer

is started. If $\overline{\text{EDREQ}}$ pin high level sampling is completed by the end of the EXDMA write cycle, acceptance resumes after the end of the write cycle, and $\overline{\text{EDREQ}}$ pin low level sampling is performed again. This sequence of operations is repeated until the end of the transfer.

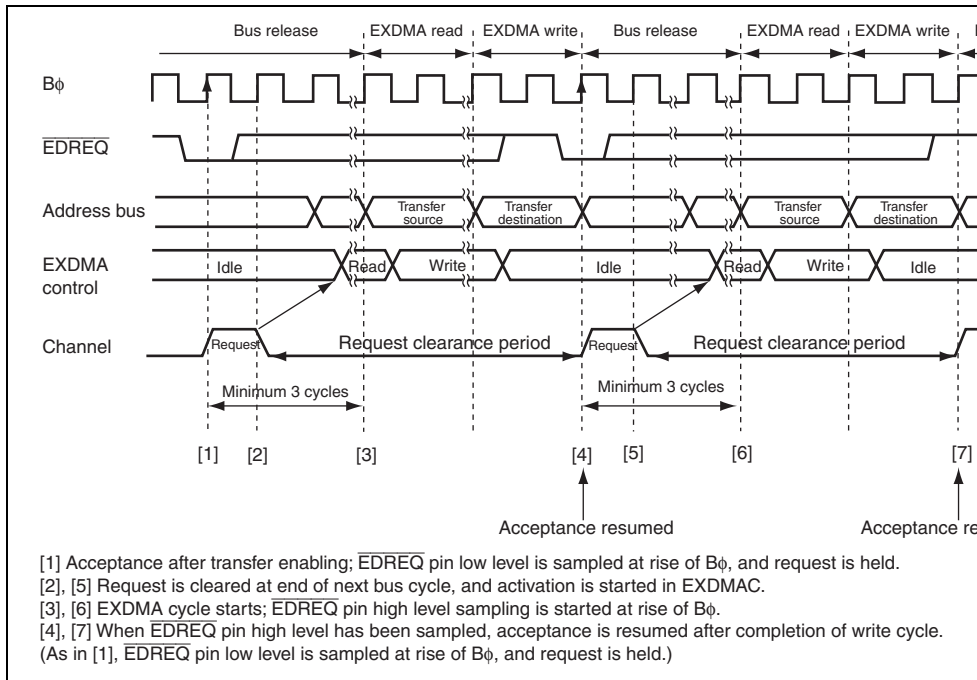


Figure 11.29 Example of Normal Transfer Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Falling Edge

performed again. This sequence of operations is repeated until the end of the transfer.

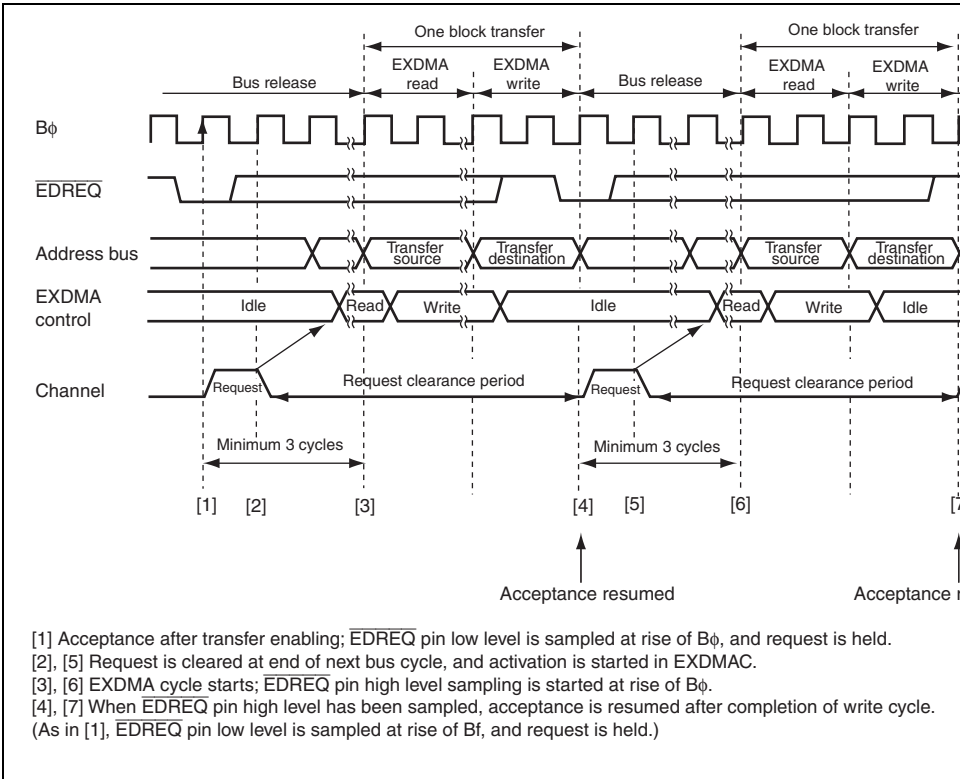


Figure 11.30 Example of Block Transfer Mode Transfer Activated by EDREQ Pin Falling Edge

until the end of the transfer.

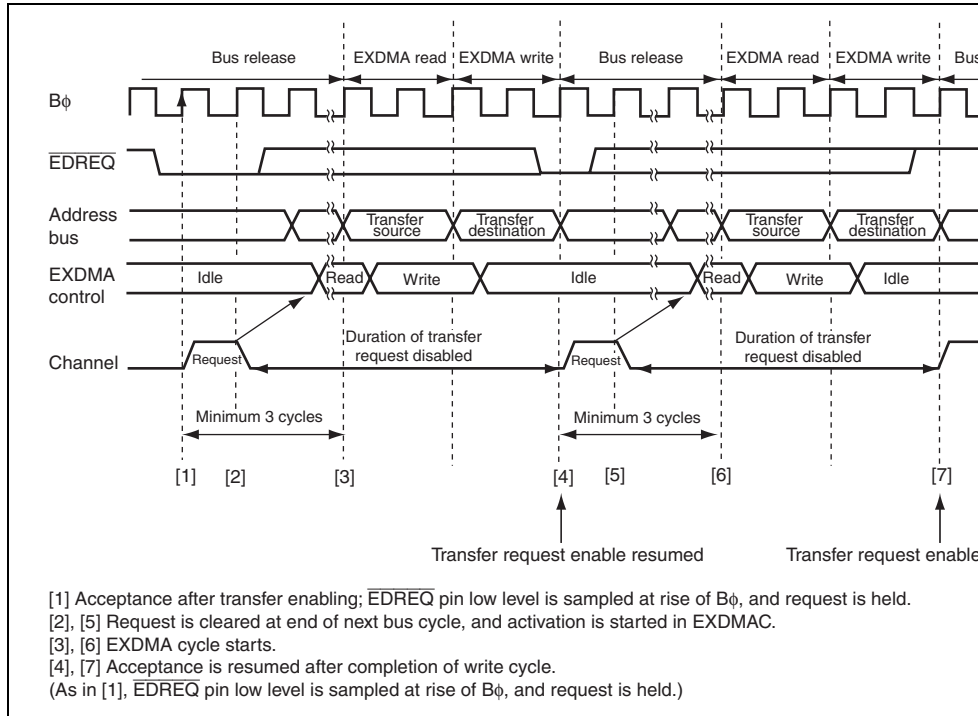


Figure 11.31 Example of Normal Transfer Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Low Level

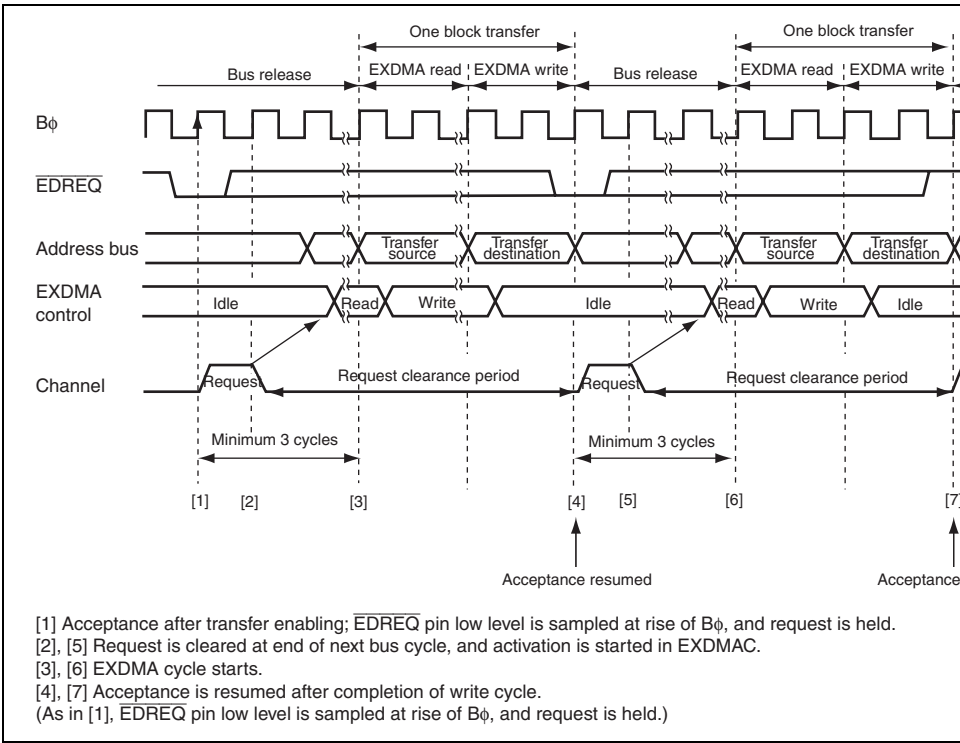


Figure 11.32 Example of Block Transfer Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Low Level

$\overline{\text{EDREQ}}$ pin is possible, the request is held within the EXDMAC. Then when activation is within the EXDMAC, the request is cleared. After the end of the write cycle, acceptance when one cycle of the request clearance period specified by $\text{NRD} = 1$ expires and $\overline{\text{EDREQ}}$ level sampling is performed again. This sequence of operations is repeated until the end of transfer.

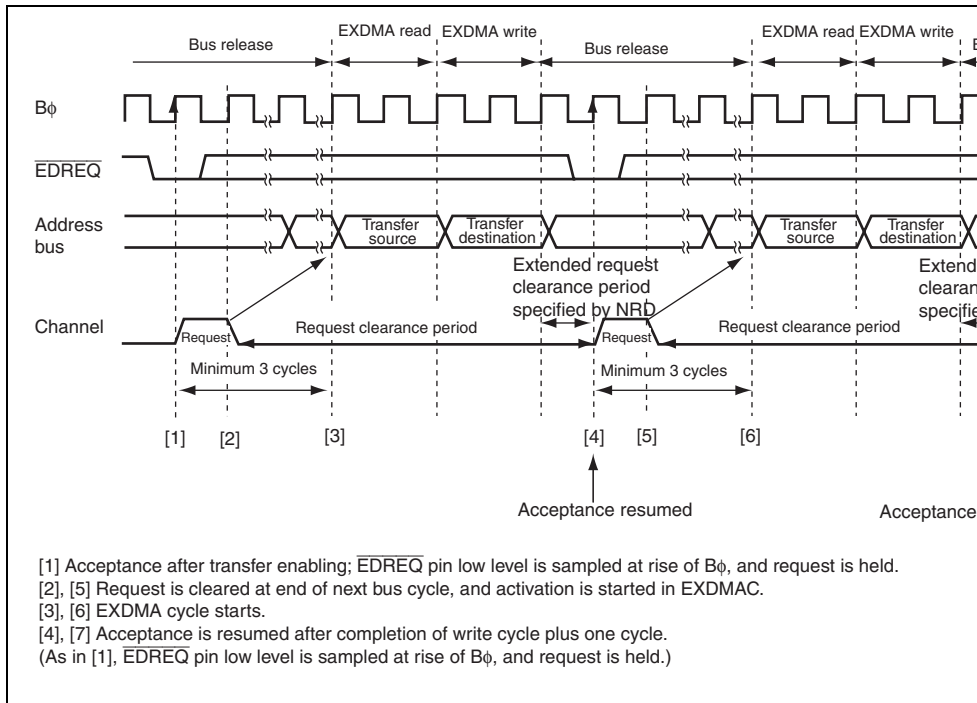


Figure 11.33 Example of Normal Transfer Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Low Level with $\text{NRD} = 1$ Specified

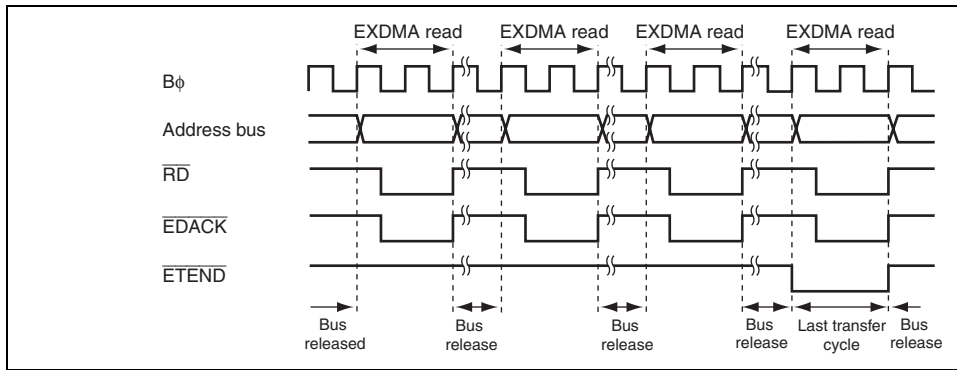


Figure 11.34 Example of Single Address Mode (Byte Read) Transfer

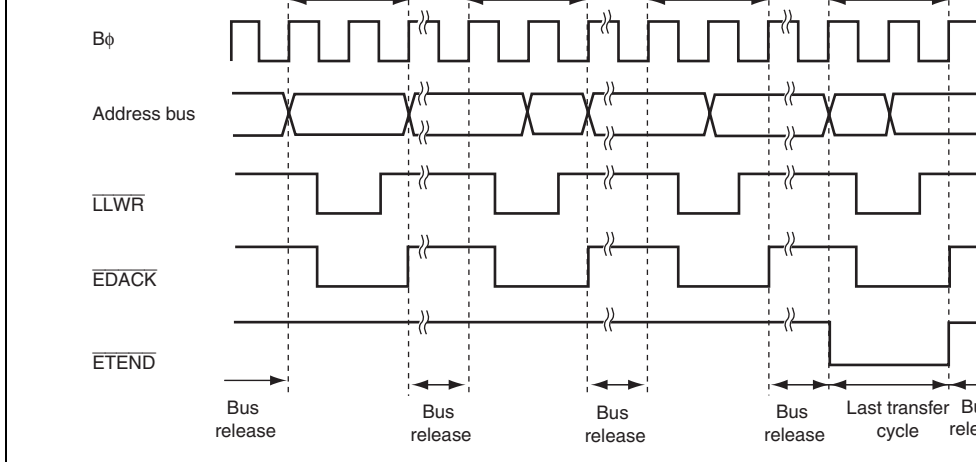


Figure 11.35 Example of Single Address Mode (Byte Write) Transfer

is started. If $\overline{\text{EDREQ}}$ pin high level sampling is completed by the end of the EXDMA cycle, acceptance resumes after the end of the single cycle, and $\overline{\text{EDREQ}}$ pin low level sampling is performed again. This sequence of operations is repeated until the end of the transfer.

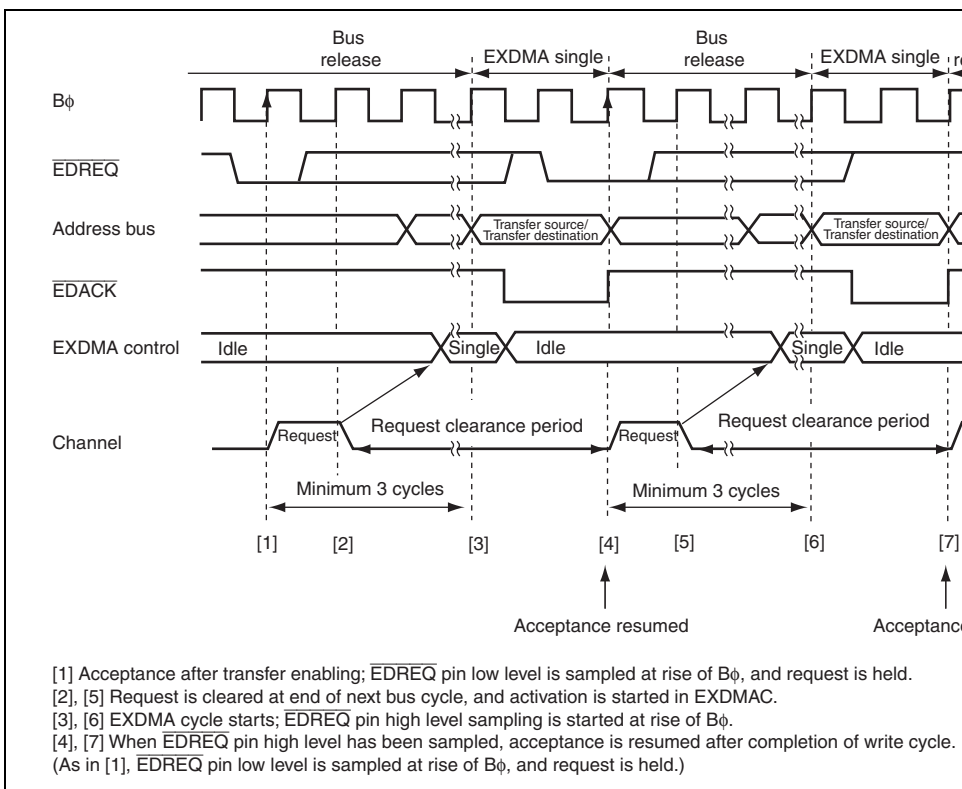


Figure 11.36 Example of Single Address Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Falling Edge

and $\overline{\text{EDREQ}}$ pin low level sampling is performed again. This sequence of operations is repeated until the end of the transfer.

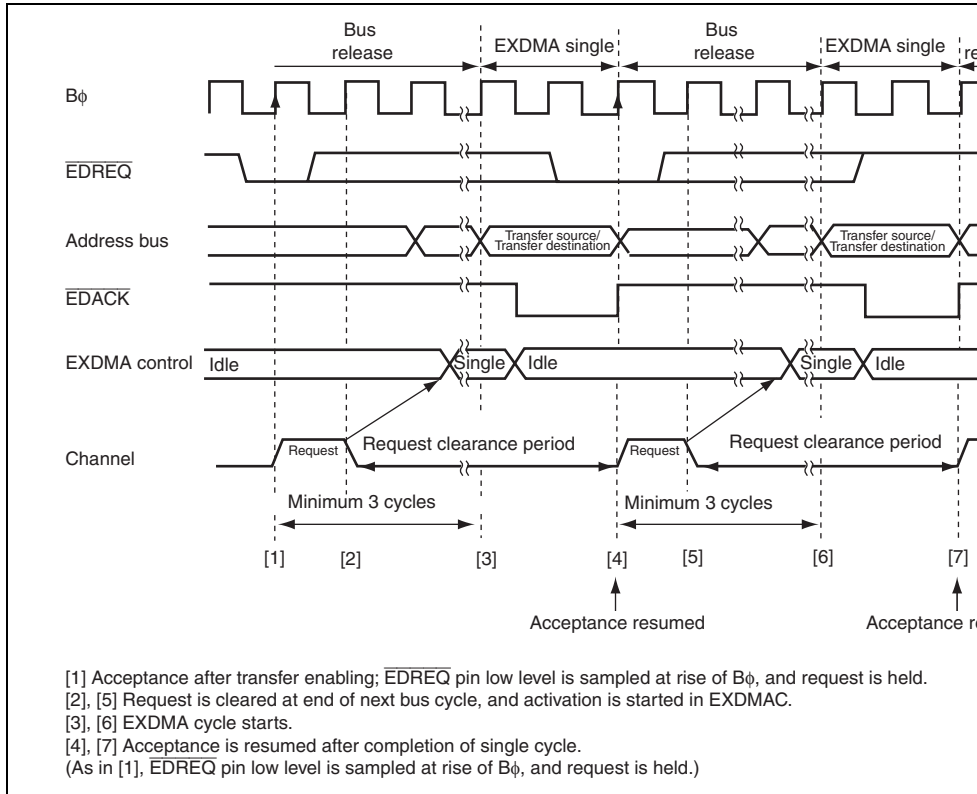


Figure 11.37 Example of Single Address Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Low Level

$\overline{\text{EDREQ}}$ pin is possible, the request is held within the EXDMAC. Then when activation within the EXDMAC, the request is cleared. After the end of the single cycle, acceptance when one cycle of the request clearance period specified by $\text{NRD} = 1$ expires and $\overline{\text{EDREQ}}$ level sampling is performed again. This sequence of operations is repeated until the end of transfer.

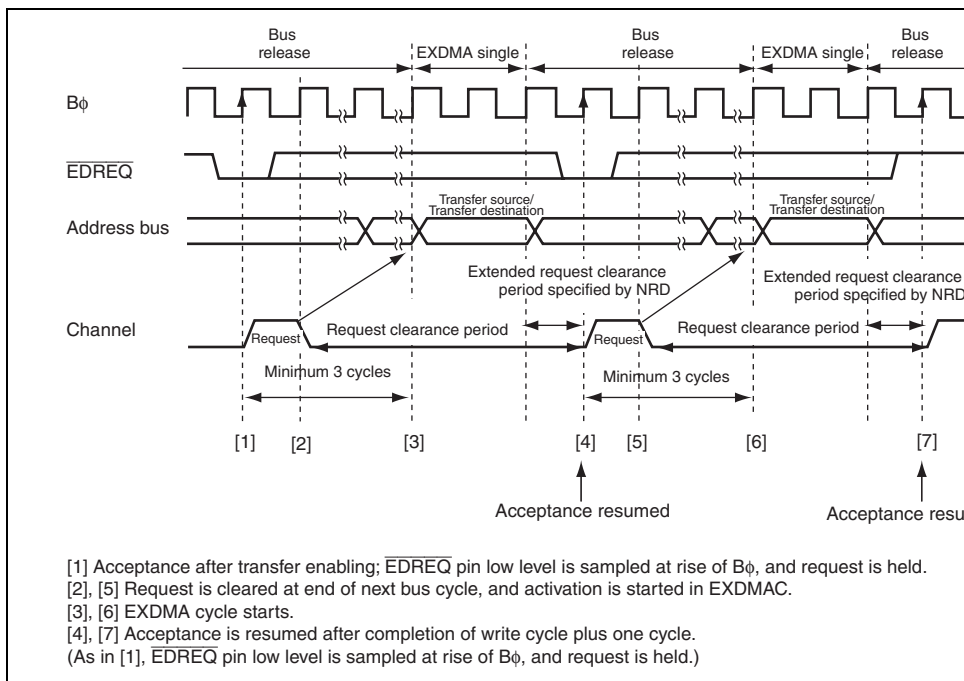


Figure 11.38 Example of Single Address Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Low Level with $\text{NRD} = 1$ Specified

is resumed on completion of the higher-priority channel transfer.

Figures 10.39 and 10.40 show operation timing examples for various conditions.

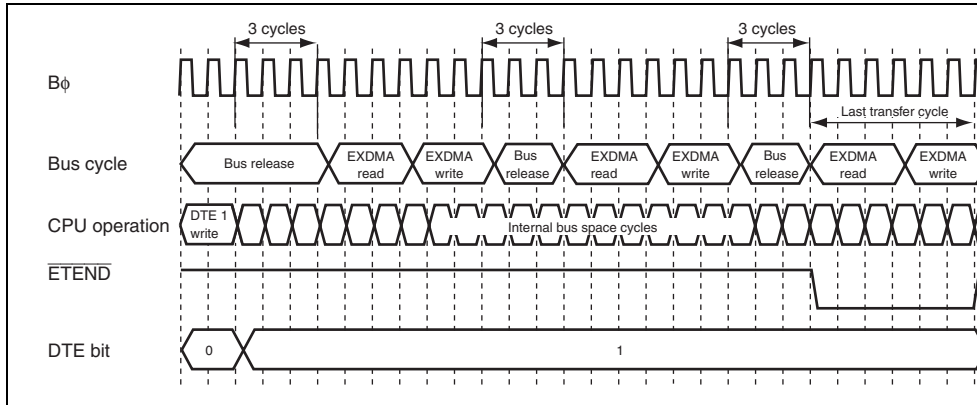


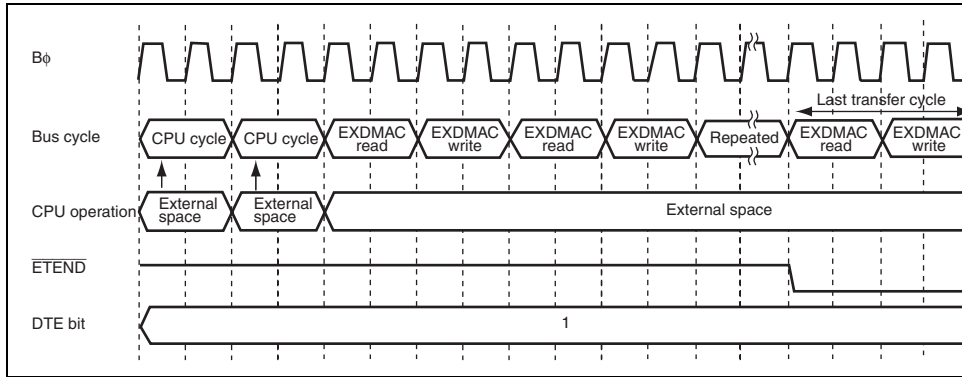
Figure 11.39 Auto-Request/Normal Transfer Mode/Cycle Steal Mode (No Conflict/Dual Address Mode)

**Figure 11.40 Auto-Request/Normal Transfer Mode/Cycle Steal Mode
(CPU Cycles/Single Address Mode)**

(2) Auto-Request/Normal Transfer Mode/Burst Mode

With auto-request (in burst mode), when the DTE bit is set to 1 in EDMDR, an EXDMA cycle is started a minimum of three cycles later. Once transfer is started, it continues (as until the transfer end condition is satisfied. Transfer requests for other channels are held until the end of transfer on the current channel.

Figures 10.41 to 10.43 show operation timing examples for various conditions.



**Figure 11.41 Auto-Request/Normal Transfer Mode/Burst Mode
(CPU Cycles/Dual Address Mode)**

**Figure 11.42 Auto-Request/Normal Transfer Mode/Burst Mode
(Conflict with Another Channel/Single Address Mode)**

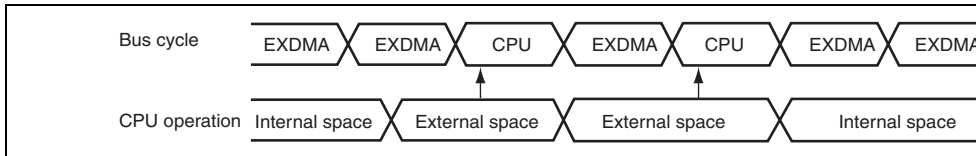


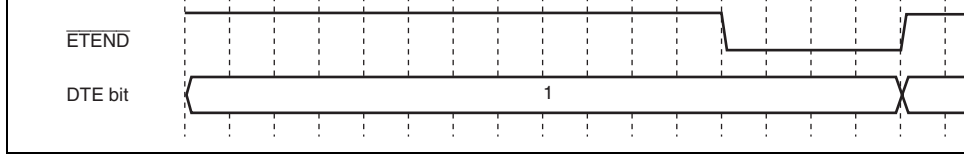
Figure 11.43 External Bus Master Cycle Steal Function (Auto-Request/Normal Transfer Mode/Burst Mode with CPU Cycles/Single Address Mode/EBCCS = 1)

(3) External Request/Normal Transfer Mode/Cycle Steal Mode

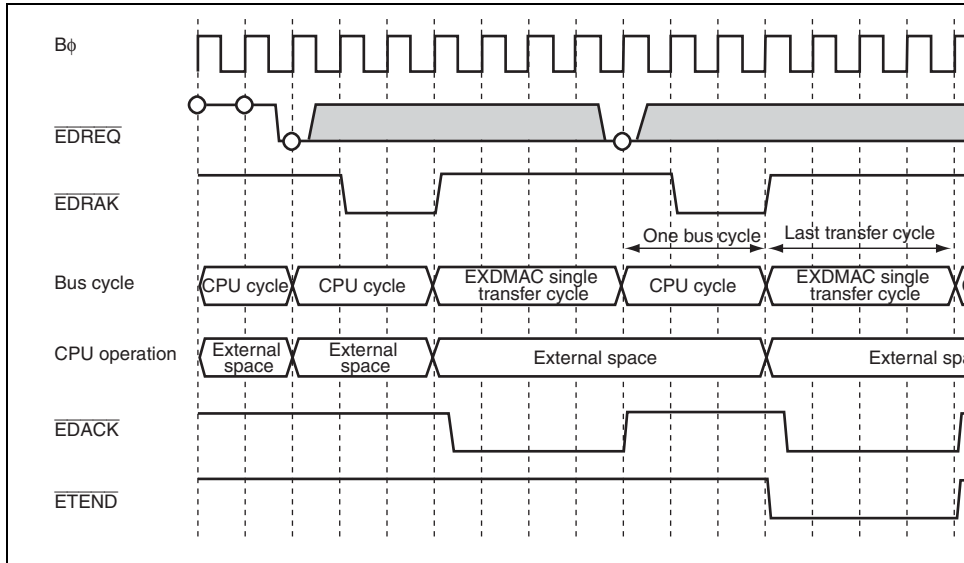
In external request mode, an EXDMA transfer cycle is started a minimum of three cycles after a transfer request is accepted. The next transfer request is accepted after the end of a one-transfer unit EXDMA cycle. For external bus space CPU cycles, at least one bus cycle is generated before the next EXDMA cycle.

If a transfer request is generated for another channel, an EXDMA cycle for the other channel is generated before the next EXDMA cycle.

The $\overline{\text{EDREQ}}$ pin sensing timing is different for low level sensing and falling edge sensing. The same applies to transfer request acceptance and transfer start timing.



**Figure 11.44 External Request/Normal Transfer Mode/Cycle Steal Mode
(No Conflict/Dual Address Mode/Low Level Sensing)**



**Figure 11.45 External Request/Normal Transfer Mode/Cycle Steal Mode
(CPU Cycles/Single Address Mode/Low Level Sensing)**



Figure 11.46 External Request/Normal Transfer Mode/Cycle Steal Mode (Conflict/Single Address Mode/Falling Edge Sensing)

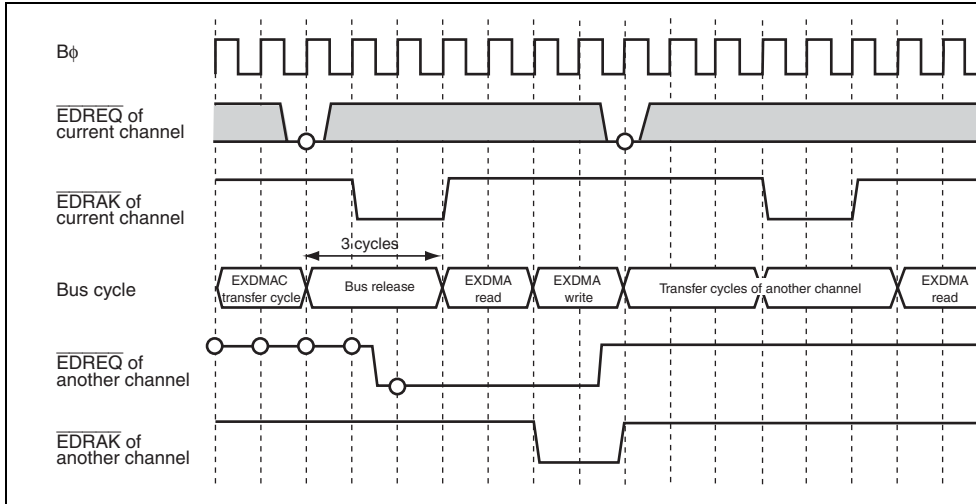
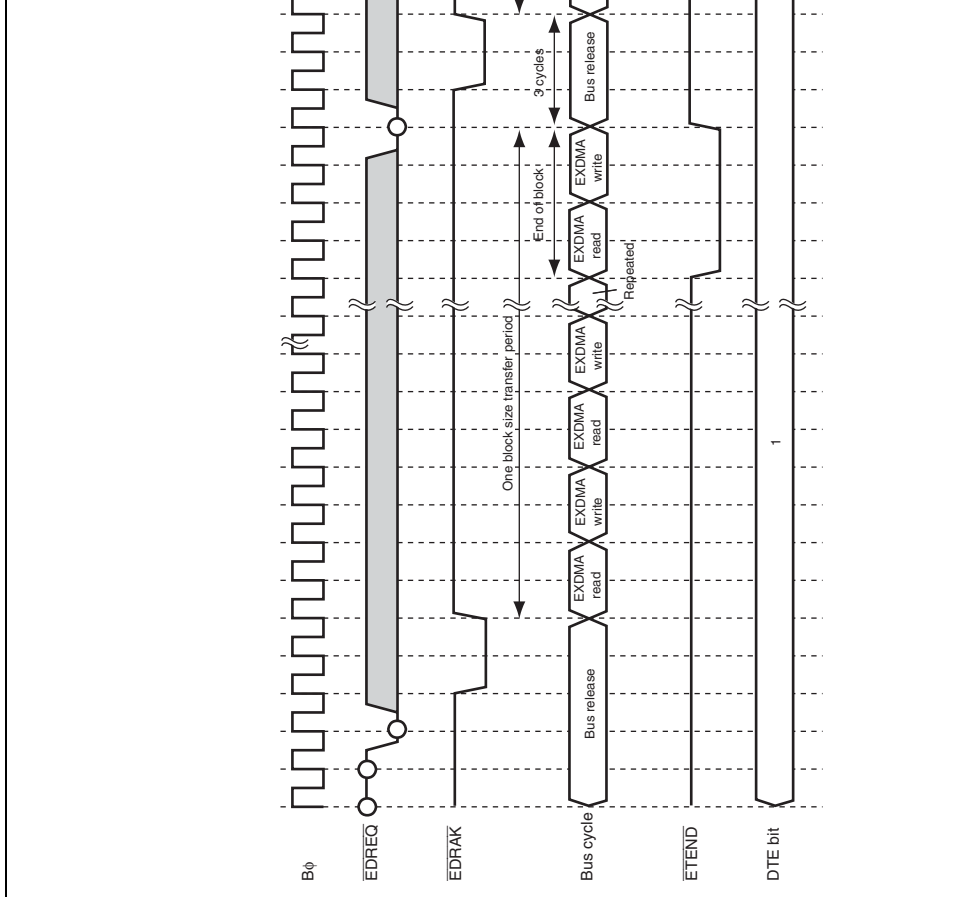
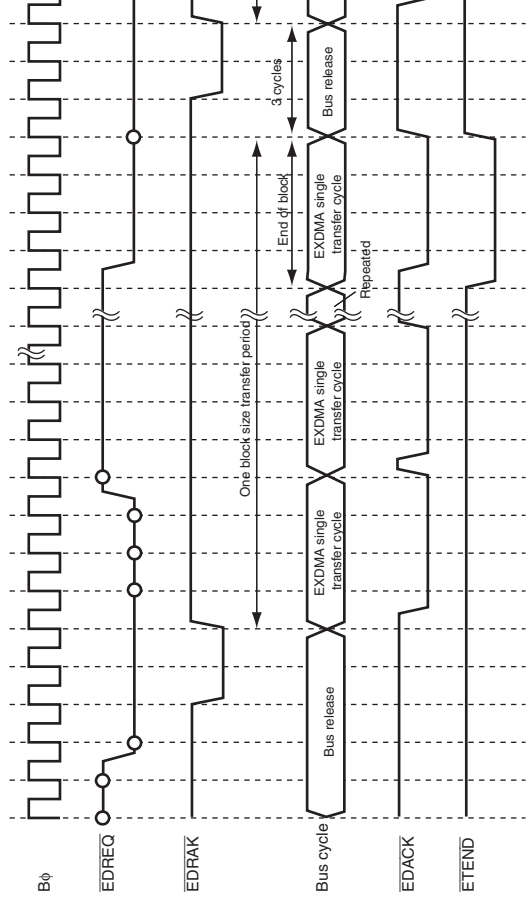


Figure 11.47 External Request/Normal Transfer Mode/Cycle Steal Mode (Conflict with Another Channel/Dual Address Mode/Low Level Sensing)

Figures 10.48 to 10.52 show operation timing examples for various conditions.



**Figure 11.48 External Request/Block Transfer Mode/Cycle Steal Mode
(No Conflict/Dual Address Mode/Low Level Sensing)**



**Figure 11.49 External Request/Block Transfer Mode/Cycle Steal Mode
(No Conflict/Single Address Mode/Falling Edge Sensing)**

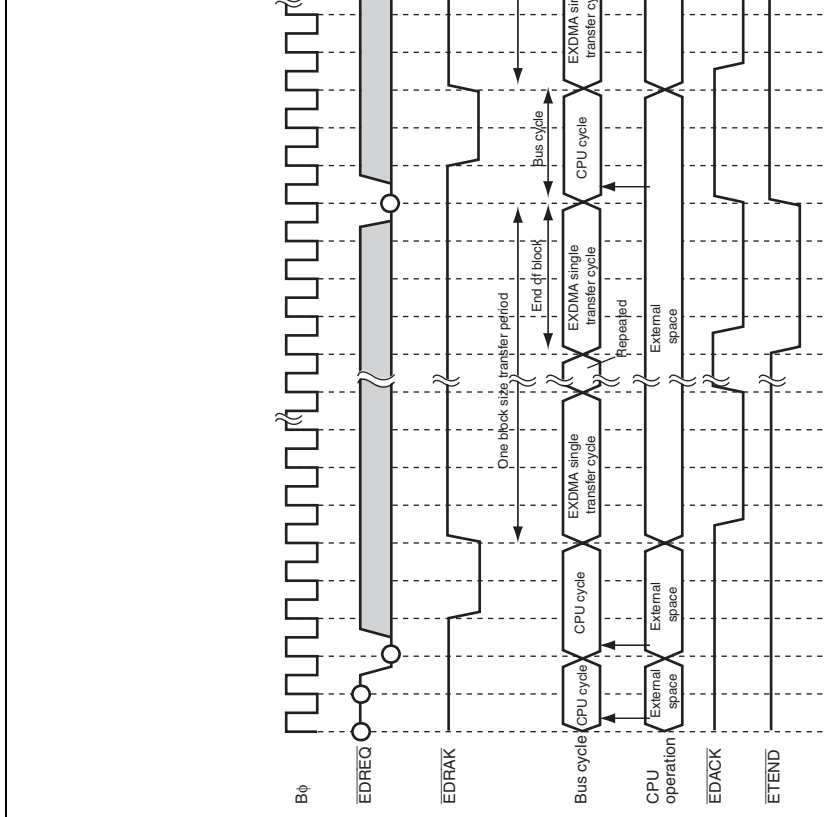


Figure 11.50 External Request/Block Transfer Mode/Cycle Steal Mode (CPU Cycles/Single Address Mode/Low Level Sensing)

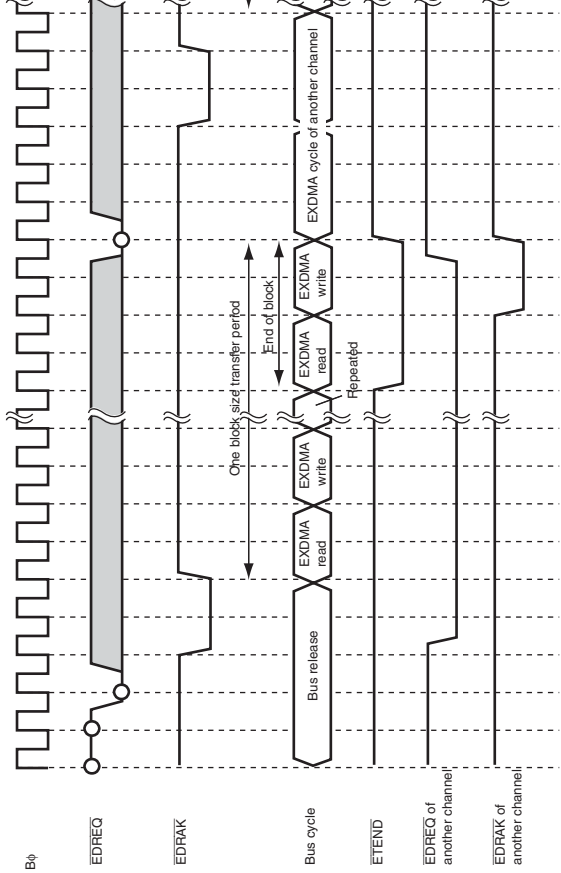
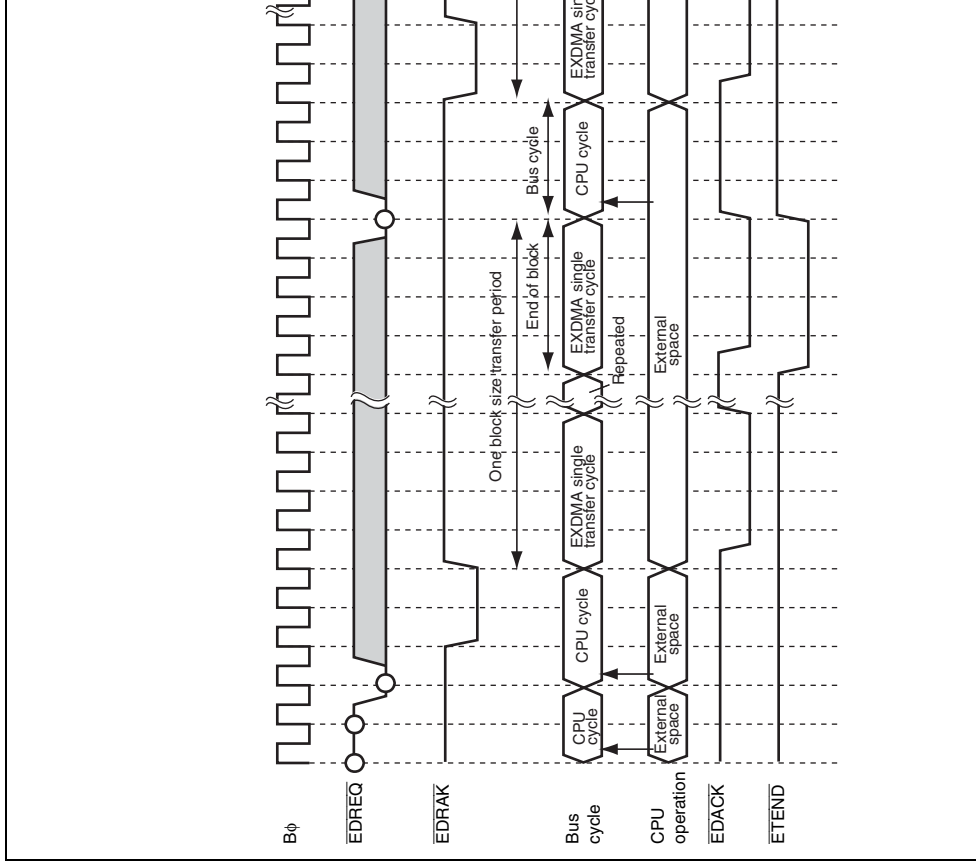


Figure 11.51 External Request/Block Transfer Mode/Cycle Steal Mode (Conflict with Another Channel/Dual Address Mode/Low Level Sensing)



**Figure 11.52 External Request/Block Transfer Mode/Cycle Steal Mode
(CPU Cycles/EBCCS = 1/Single Address Mode/Low Level Sensing)**

in this mode, both the transfer source and destination addresses are specified for transfer. EXDMAC internal registers. The transfer source address is set in the source address register (EDSAR), and the transfer destination address is set in the destination address register (EDDAR).

The transfer is processed by performing the consecutive read of a cluster-size from the transfer source address and then the consecutive write of that data to the transfer destination address. A data access size of 32 bytes can be specified as a cluster size. When one data access size is specified as a cluster size, block transfer mode (dual address mode) is used.

The cycles in a cluster-size transfer are indivisible: another bus cycle (external access by bus master, refresh cycle, or external bus release cycle) does not occur in a cluster-size transfer.

$\overline{\text{ETEND}}$ pin output can be enabled or disabled by means of the ETENDE bit in EDMDR. $\overline{\text{ETEND}}$ is output for the last write cycle. The $\overline{\text{EDACK}}$ signal is not output.

Figure 11.53 shows the data flow in the cluster transfer mode (dual address mode), figure 11.54 shows an example of the timing in cluster transfer dual address mode, and figure 11.55 shows the cluster transfer dual address mode operation.

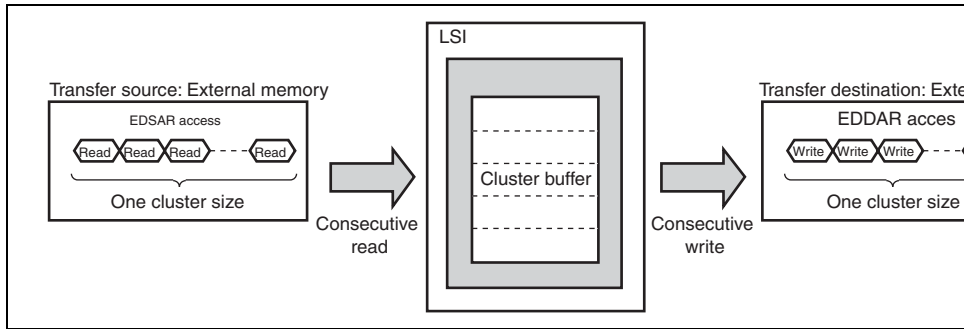


Figure 11.53 Data Flow in Cluster Transfer Dual Address Mode

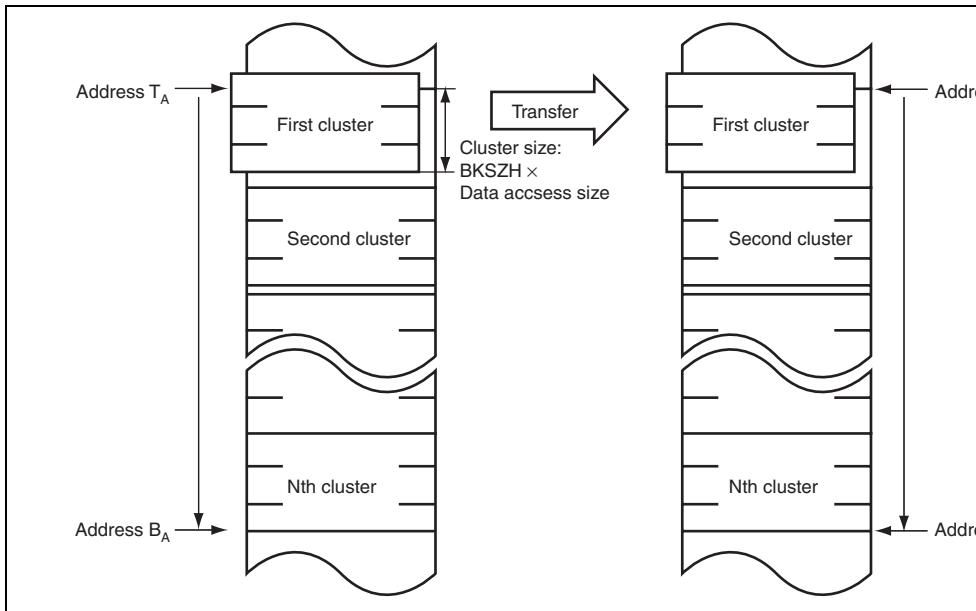


Figure 11.55 Cluster Transfer Dual Address Mode Operation

When a word or longword is specified as a data access size but the source or destination is not at the word or longword boundary, use the appropriate data access size for efficient data transfer.

In an example shown in figure 11.56, a longword-size transfer is performed with 4-longword specified as a cluster size in the cluster transfer dual address mode from the lower two bits to B'10.

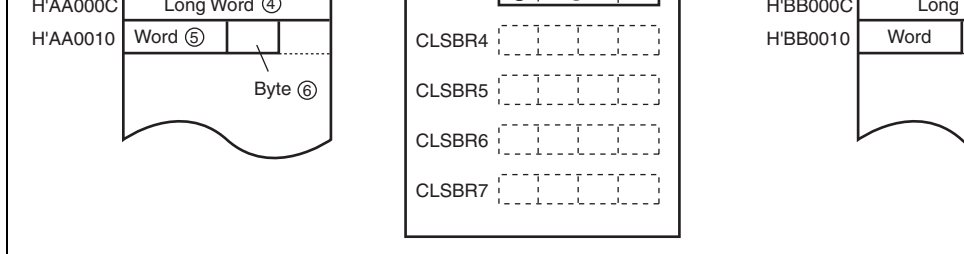


Figure 11.56 Odd Address Transfer

(2) Cluster Transfer Read Address Mode (AMS = 1, DIRS = 0)

In this mode, the transfer source address is specified in the source address register (EDSBR). Data is read from the transfer source and transferred to the cluster buffer. In this mode, the $\overline{\text{ETEND}}$ bit in the mode control register (EDMDR) must be set to 1.

Two data access size to 32 bytes can be specified as a cluster size for the consecutive read operation.

The cycles in a cluster-size transfer are indivisible: another bus cycle (external access by bus master, refresh cycle, or external bus release cycle) does not occur in a cluster-size transfer.

$\overline{\text{ETEND}}$ pin output can be enabled or disabled by means of the ETENDE bit in EDMDR. $\overline{\text{ETEND}}$ is output for the last read cycle. When an idle cycle is inserted before the last read cycle, $\overline{\text{ETEND}}$ signal is also output in the idle cycle.

In this mode, the EDACKE bit in EDMDR must be set to 0 to disable the $\overline{\text{EDACK}}$ pin output.

Figure 11.57 shows the data flow in the cluster transfer read address mode (from the external memory to the cluster buffer), and figure 11.58 shows an example of the timing in cluster transfer read address mode.

Figure 11.57 Data Flow in Cluster Transfer Read Address Mode (from External Memory to Cluster Buffer)

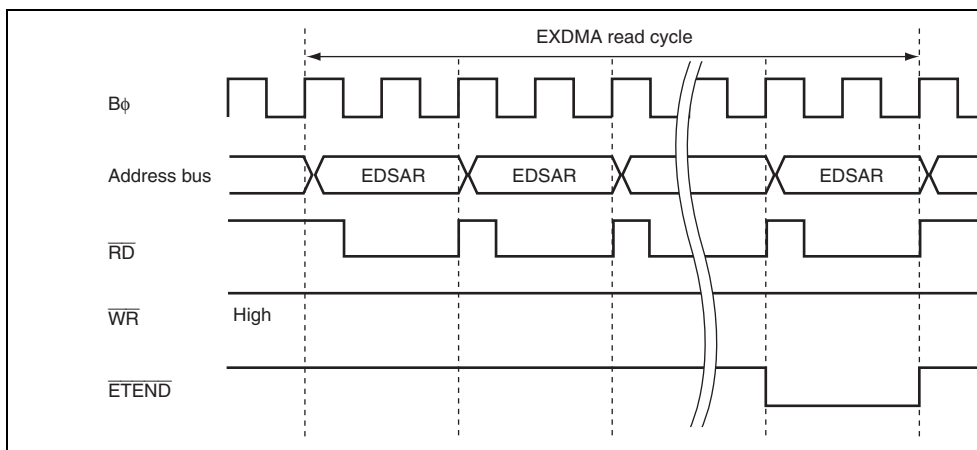


Figure 11.58 Timing in Cluster Transfer Read Address Mode (from External Memory to Cluster Buffer)

\overline{ETEND} pin output can be enabled or disabled by means of the ETENDE bit in EDMDR. \overline{ETEND} is output for the last write cycle. When an idle cycle is inserted before the last write cycle, \overline{ETEND} signal is also output in the idle cycle.

In this mode, the EDACKE bit in EDMCR must be set to 0 to disable the \overline{EDACK} pin output.

Figure 11.59 shows the data flow in the cluster transfer write address mode (from the cluster buffer to the external memory), and figure 11.60 shows an example of the timing in cluster transfer write address mode.

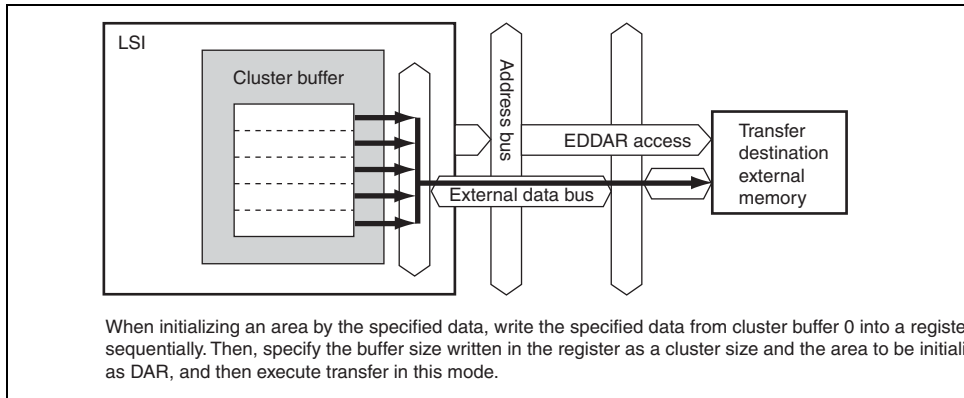


Figure 11.59 Data Flow in Cluster Transfer Write Address Mode (from Cluster Buffer to External Memory)

**Figure 11.60 Timing in Cluster Transfer Write Address Mode
(from Cluster Buffer to External Memory)**

11.6.2 Setting of Address Update Mode

The cluster transfer mode transfer is restricted by the address update mode function. There are the following four address update methods: increment, decrement, fixed, and offset addition.

When the address increment method is specified and if the specified address is not at the address boundary for the data access size (odd address for a word-size transfer, address beyond the address boundary for a longword-size transfer), the bus cycle is divided for transfer until the address becomes at the address boundary. When the address matches the boundary, transfer is processed in units of data access sizes. At the end of transfer, the bus cycle is divided again to transfer the remaining data in cluster transfer mode.

With address decrement, fixed, or offset addition method, specify the address, that matches the address boundary for the data access size, in EDSAR and EDDAR. When specifying the address that is not at the address boundary for the data access size, in EDSAR and EDDAR, fix the lower bit to 0 (lower one bit for a word-size transfer, and lower two bits for a longword-size transfer) in the address register so that the transfer is processed in units of data access sizes. The block transfer mode must be used for transfer of data by dividing the bus cycle according to the address boundary.

When the EDTCR value is smaller than the cluster size, a transfer size error occurs. In this case, when the TSEIE bit in EDMDR is cleared to 0, the cluster transfer mode is switched to the normal transfer mode to process the remaining data. With the decrement, fixed, or offset addition method, transfer is performed without fixing the lower bit to 0.

(1) Cluster transfer mode

In cluster transfer mode, a cluster-size transfer is processed in response to one transfer request.

In an example shown in figure 11.61, the $\overline{\text{ETEND}}$ pin output is enabled, and word-size transfer is performed with 4-byte cluster size in cluster transfer mode from the external 16-bit, 2-state access space to the external 16-bit, 2-state access space.

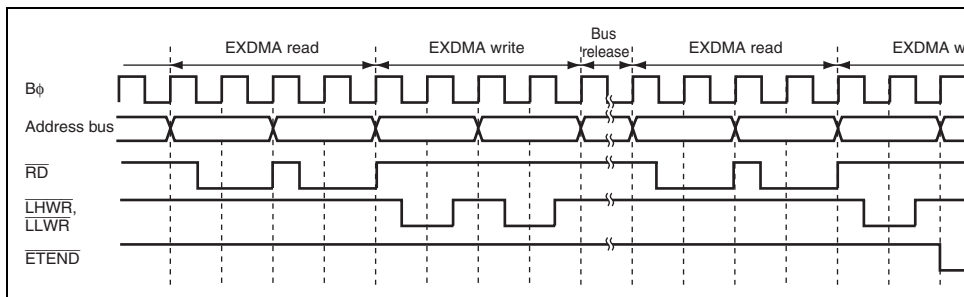


Figure 11.61 Example of Cluster Transfer Mode Transfer

is started if EDREQ pin high level sampling is completed by the end of the last cluster transfer cycle, acceptance resumes after the end of the write cycle, and EDREQ pin low level sampling is performed again. This sequence of operations is repeated until the end of the transfer.

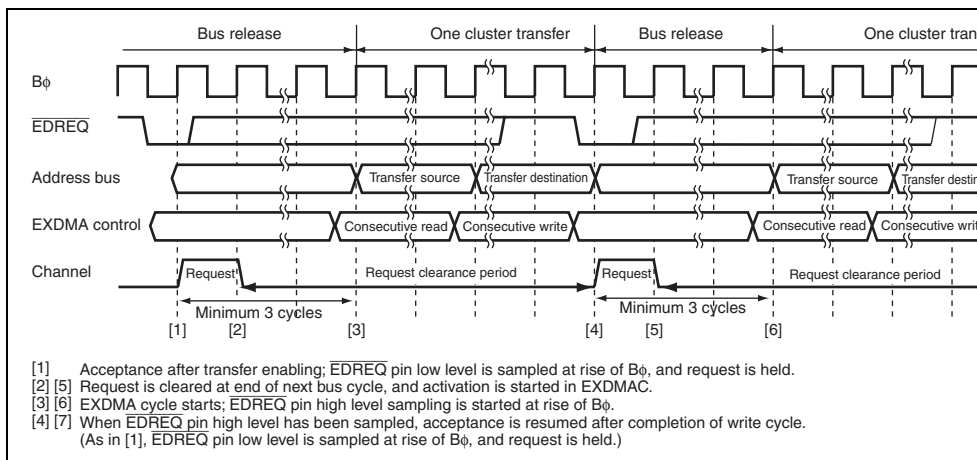


Figure 11.62 Example of Cluster Transfer Mode Transfer Activated by EDREQ Pin Falling Edge

resumes and $\overline{\text{EDREQ}}$ pin low level sampling is performed again. This sequence of operations is repeated until the end of the transfer.

When NRD bit = 0 in EDMDR, acceptance resumes at the end of the last cluster write cycle and $\overline{\text{EDREQ}}$ pin low level sampling is performed again. This sequence of operations is repeated until the end of the transfer.

When NRD bit = 1 in EDMDR, acceptance resumes after one cycle from the end of the last write cycle, and $\overline{\text{EDREQ}}$ pin low level sampling is performed again. This sequence of operations is repeated until the end of the transfer.

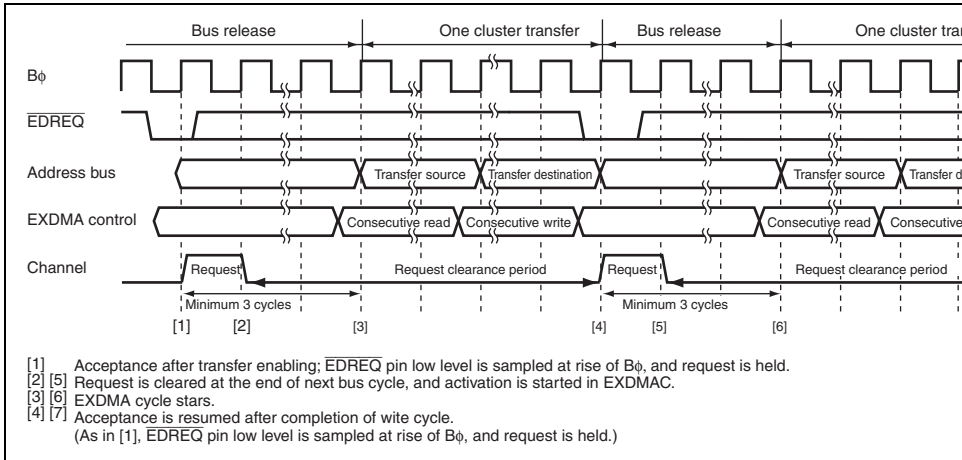


Figure 11.63 Example of Cluster Transfer Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Low Level

channel is resumed on completion of the higher-priority channel transfer.

The cluster transfer mode (read address mode and write address mode) can not be used with the cluster transfer mode (dual address mode) among more than one channel at the same time. When using the cluster transfer mode (read address mode and write address mode), do not set the transfer mode for another channel.

Figures 10.64 to 10.66 show operation timing examples for various conditions.

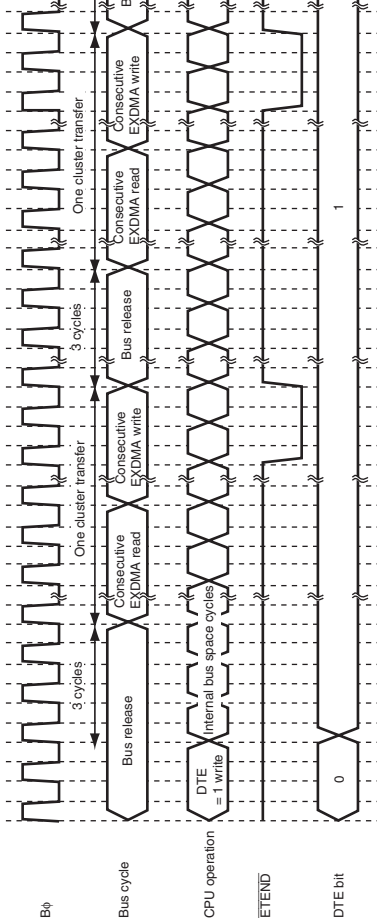


Figure 11.64 Auto-Request/Cluster Transfer Mode/Cycle Steal Mode (No Conflict/Dual Address Mode)

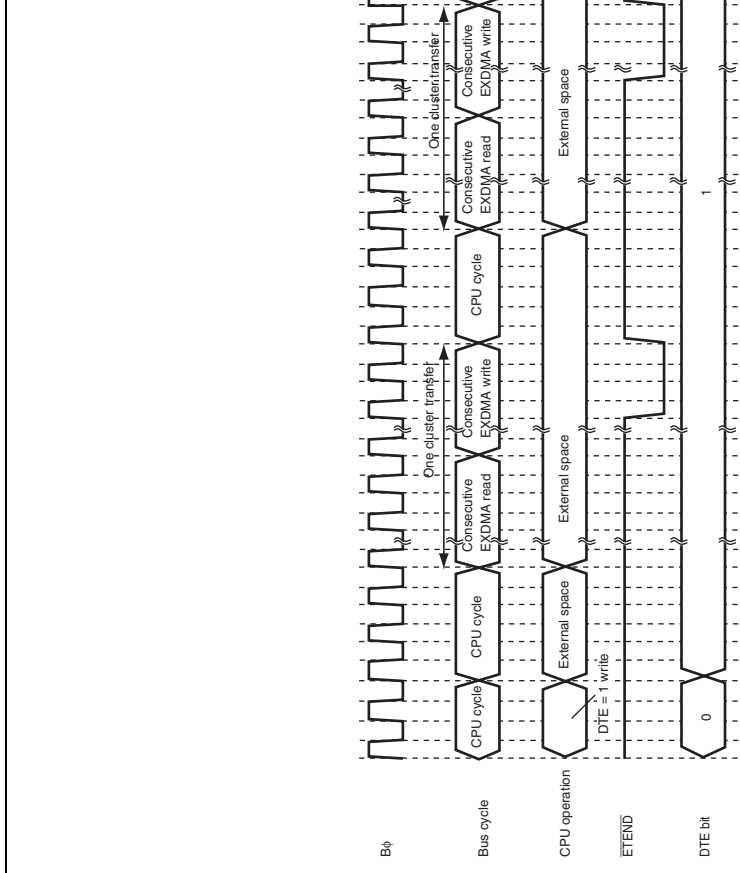
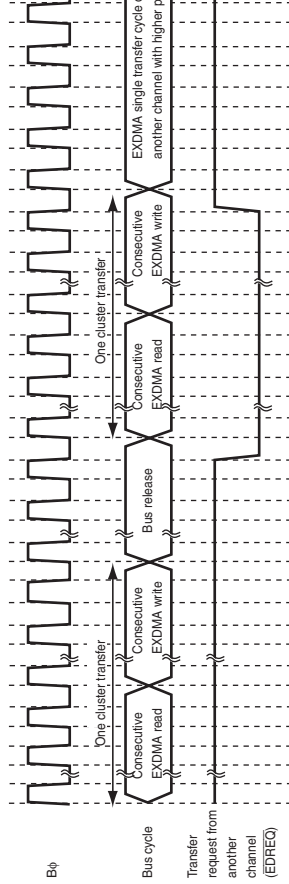


Figure 11.65 Auto-Request/Cluster Transfer Mode/Cycle Steal Mode (CPU Cycles/Dual Address Mode)

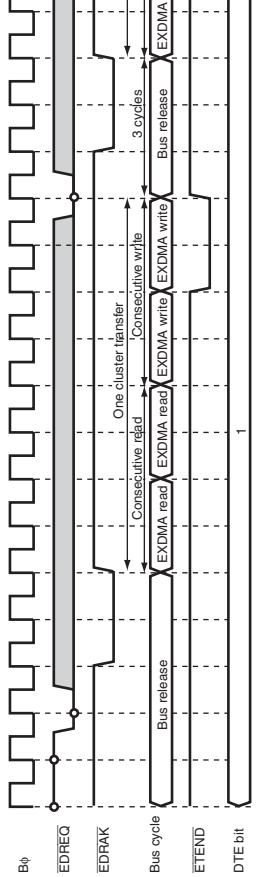


**Figure 11.66 Auto-Request/Cluster Transfer Mode/Cycle Steal Mode
(Conflict with Another Channel/Dual Address Mode)**

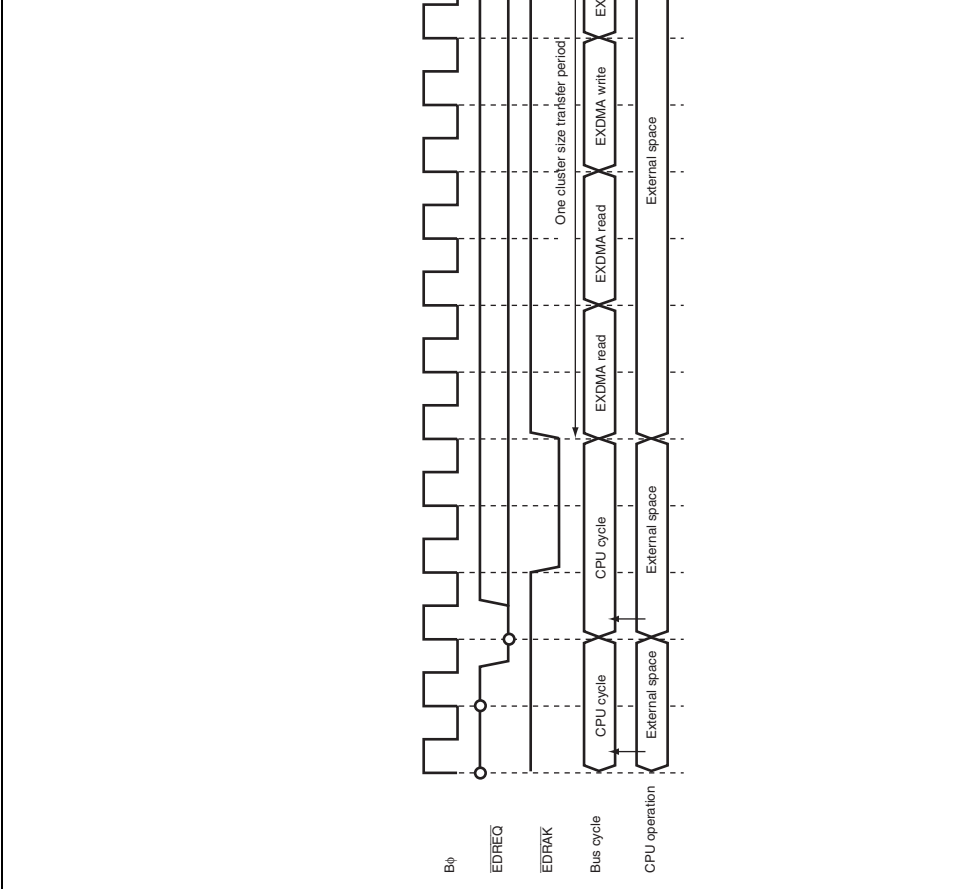
transfer mode for another channel.

The $\overline{\text{EDREQ}}$ pin sensing timing is different for low level sensing and falling edge sensing. The same applies to transfer request acceptance and transfer start timing.

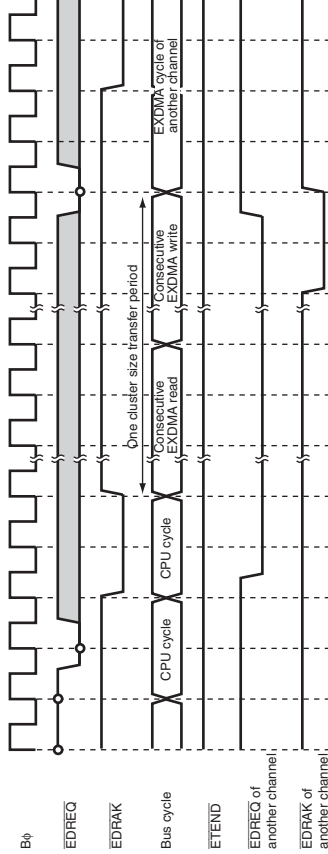
Figures 10.67 to 10.69 show operation timing examples for various conditions.



**Figure 11.67 External Request/Cluster Transfer Mode/Cycle Steal Mode
(No Conflict/Dual Address Mode/Low Level Sensing)**



**Figure 11.68 External Request/Cluster Transfer Mode/Cycle Steal Mode
(CPU Cycles/Dual Address Mode/Low Level Sensing)**



**Figure 11.69 External Request/Cluster Transfer Mode/Cycle Steal Mode
(Conflict with Another Channel/Dual Address Mode/Low Level Sensing)**

generated by the transfer counter. EXDMA transfer does not end if the EDTCR value has since before the start of transfer.

(2) Transfer End by Transfer Size Error Interrupt

When the following conditions are satisfied while the TSEIE bit in EDMDR is set to 1, a size error occurs and an EXDMA transfer is terminated. At this time, the DTE bit in EDMDR is cleared to 0 and the ESIF bit in EDMDR is set to 1.

- In normal transfer mode and repeat transfer mode, when the next transfer is requested while a transfer is disabled due to the EDTCR value less than the data access size.
- In block transfer mode, when the next transfer is requested while a transfer is disabled due to the EDTCR value less than the block size.
- In cluster transfer mode, when the next transfer is requested while a transfer is disabled due to the EDTCR value less than the cluster size.

When the TSEIE bit in EDMDR is cleared to 0, data is transferred until the EDTCR value is reached. A transfer size error is not generated. Operation in each transfer mode is described below.

- In normal transfer mode and repeat mode, when the EDTCR value is less than the data access size, data is transferred in bytes.
- In block transfer mode, when the EDTCR value is less than the block size, the specified amount of data in EDTCR is transferred instead of transferring the block size of data. When the EDTCR value is less than the data access size, data is transferred in bytes.
- In cluster transfer mode, when the EDTCR value is less than the cluster size, the specified amount of data in EDTCR is transferred instead of transferring the cluster size of data. When the EDTCR value is less than the data access size, data is transferred in bytes.

(4) Transfer End by Extended Repeat Area Overflow Interrupt

If an address overflows the extended repeat area when an extended repeat area specific interrupt has been made and the SARIE or DARIE bit in EDACR is set to 1, an extended repeat area overflow interrupt is requested. The interrupt request terminates EXDMA transfer, the DTE bit in EDMDR is cleared to 0, and the ESIF bit in EDMDR is set to 1 at the same time.

In dual address mode, if an extended repeat area overflow interrupt is requested during a write cycle, the following write cycle processing is still executed.

In block transfer mode, if an extended repeat area overflow interrupt is requested during a block, transfer continues to the end of the block. Transfer end by means of an extended repeat area overflow interrupt occurs between block-size transfers.

In cluster transfer mode, if an extended repeat area overflow interrupt is requested during a cluster, transfer continues to the end of the cluster. Transfer end by means of an extended repeat area overflow interrupt occurs between cluster-size transfers.

(5) Transfer End by 0-Write to DTE Bit in EDMDR

When 0 is written to the DTE bit in EDMDR by the CPU, etc., transfer ends after completion of the EXDMA cycle in which transfer is in progress or a transfer request was accepted.

In block transfer mode, EXDMA transfer ends after completion of one-block-size transfer in progress.

In cluster transfer mode, EXDMA transfer ends after completion of one-cluster-size transfer in progress.

units of transfers.

In single address mode, EXDMA transfer ends at the end of the EXDMA transfer bus cycle units of transfers.

(b) Block transfer mode

A block size EXDMA transfer is aborted. A block size transfer is not correctly executed, matching between the actual transfer and the transfer request is not guaranteed.

In dual address mode, a write cycle corresponding to a read cycle is executed as well as in normal transfer mode.

(c) Cluster transfer mode

A cluster size EXDMA transfer is aborted. If transfer is aborted in a read cycle, the read cycle is guaranteed. If transfer is aborted in a write cycle, the data not transferred is not guaranteed. Matching between the transfer counter and the address register is not guaranteed since the processing cannot be controlled.

(7) Transfer End by Address Error

If an address error occurs, the EXDMAC clears the DTE bit to 0 in all channels, and sets the DTE bit in EDMDR_0 to 1. An address error during EXDMA transfer forcibly terminates the transfer. To perform EXDMA transfer after an address error occurs, clear the ERRF bit to 0 and then set the DTE bit to 1 in each channel.

The transfer end timing after address error detection is the same as for the one when an NMI interrupt occurs.

channel register (CPUPCR). For details, see section 7.4, CPU Priority Control Function, DMAC, and EXDMAC.

The EXDMAC priority level can be set independently for each channel by the EDMA0 EDMA0 bits in EDMDR.

The CPU priority level, which corresponds to the priority level of exception handling, can be set by updating the values of the CPUP2 to CPUP0 bits in CPUPCR with the interrupt mask register (IMR) values.

When the CPUPCE bit in CPUPCR is set to 1 to enable the CPU priority level control function, if the EXDMAC priority level is lower than the CPU priority level, the transfer request of the corresponding channel is masked and the channel activation is disabled. When the priority level of another channel is the same or higher than the CPU priority level, the transfer request of that channel is accepted and transfer is enabled regardless of the priority levels of channels.

The CPU priority level control function holds pending the transfer source, which masks the transfer request. When the CPU priority level becomes lower than the channel priority level, updating one of them, the transfer request is accepted and transfer starts. The transfer request pending is cleared by writing 0 to the DTE bit.

When the CPUPCE bit is cleared to 0, the lowest CPU priority level is assumed.

These consecutive EXDMA read and write cycles are indivisible: refresh cycle, external release cycle, or external space access cycle by internal bus master (CPU, DTC, DMAC) occur between a read cycle and a write cycle.

In cluster transfer mode, the transfer cycle in one cluster is indivisible.

In block transfer mode and auto-request burst mode, the EXDMA transfer bus cycles continue during this period, the bus priority level of the internal bus master is lower than the EXDMAC so that external space access is held pending (when EBCCS = 0 in the bus control register 2 (BCR2)).

When switching to another channel, or in the auto-request cycle steal mode, the EXDMA transfer bus cycles and internal bus master cycles are alternatively executed. When the internal bus master is not issuing an external space access cycle, the EXDMA transfer bus cycles are continuously executed in the allowable range.

When the EBCCS bit in BCR2 is set to 1 to enable the arbitration function between the EXDMA and the internal bus master, the bus mastership is released, except for indivisible bus cycles transferred between the EXDMAC and the internal bus master alternatively. For details, see section 9, Bus Controller (BSC).

EXDMTEND1	Transfer end indicated by channel 1 transfer counter
EXDMTEND2	Transfer end indicated by channel 2 transfer counter
EXDMTEND3	Transfer end indicated by channel 3 transfer counter
EXDMEEND0	Channel 0 transfer size error Channel 0 repeat size end Channel 0 source address extended repeat area overflow Channel 0 destination address extended repeat area overflow
EXDMEEND1	Channel 1 transfer size error Channel 1 repeat size end Channel 1 source address extended repeat area overflow Channel 1 destination address extended repeat area overflow
EXDMEEND2	Channel 2 transfer size error Channel 2 repeat size end Channel 2 source address extended repeat area overflow Channel 2 destination address extended repeat area overflow
EXDMEEND3	Channel 3 transfer size error Channel 3 repeat size end Channel 3 source address extended repeat area overflow Channel 3 destination address extended repeat area overflow

Interrupt source can be enabled or disabled by setting the DTIE and ESIE bits in EDMC for relevant channels. The DTIE bit can be combined with the DTIF bit in EDMC to generate an EXDMTEND interrupt. The ESIE bit can be combined with the ESIF bit in EDMC to generate an EXDMEEND interrupt. Interrupt sources in EXDMEEND are not identified as common interrupt sources. The interrupt priority order among channels is determined by the interrupt controller. The interrupt priority order is shown in table 11.8. For details, see section 7, Interrupt Controller.

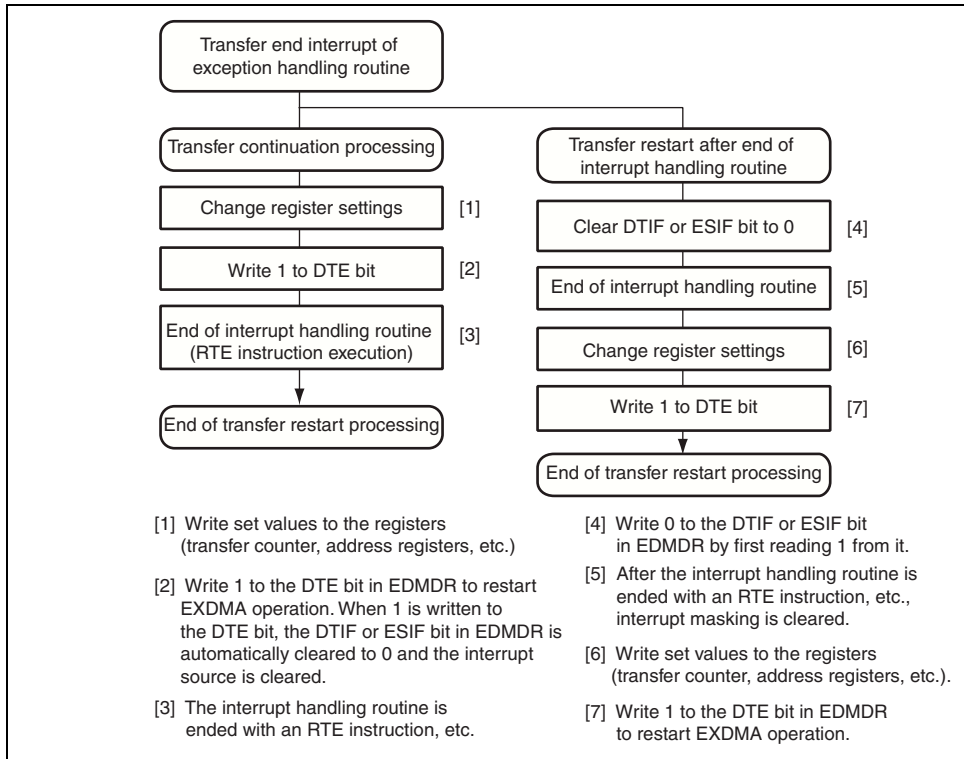
corresponding interrupt enable bit is set to 1, the condition for that interrupt is satisfied, and the ESIF bit in EDMDR is set to 1.

The transfer size error interrupt occurs when the EDTCR value is smaller than the data access size and a data-access-size transfer for one request cannot be performed for a transfer request. In block transfer mode, the block size is compared to the EDTCR value to determine a transfer size error. In cluster transfer mode, the cluster size is compared to the EDTCR value to determine a transfer size error.

The repeat size end interrupt occurs when the next transfer request is generated after the end of a repeat size transfer in repeat transfer mode. When the repeat area is not set in the address register, the transfer can be aborted periodically based on the set repeat size value. If the transfer end interrupt by the transfer counter occurs at the same time, the ESIF bit is set to 1.

The source/destination address extended repeat area overflow interrupt occurs when the address register overflows the specified extended repeat area. If the transfer end interrupt by the transfer counter occurs at the same time, the ESIF bit is set to 1.

Figure 11.70 shows the block diagram of various interrupts and their interrupt flags. The transfer end interrupt can be cleared either by clearing the DTIF or ESIF bit to 0 in EDMDR with an interrupt handling routine, or by re-setting the address registers and then setting the DTE bit in EDMDR to perform transfer continuation processing. An example of the procedure for clearing the transfer end interrupt and restarting transfer is shown in figure 11.71.

Figure 11.70 Interrupts and Interrupt Sources**Figure 11.71 Procedure for Clearing Transfer End Interrupt and Restarting T**

EXDMAC enters the module stop state. However, 1 cannot be written to the MSTPA14 bit when any of the EXDMAC's channels is enabled for transfer, or when an interrupt is requested. Before setting the MSTPA14 bit, first clear the DTE bit in EDMDR to 0, the DTIF or DTIE bit in EDMDR to 0.

When the EXDMAC clock stops, EXDMAC registers can no longer be accessed. The following EXDMAC register settings remain valid in the module stop state, and so should be disabled, if necessary, before making the module stop transition.

- ETENDE = 1 in EDMDR ($\overline{\text{ETEND}}$ pin enable)
- EDRAKE = 1 in EDMDR ($\overline{\text{EDRAK}}$ pin enable)
- EDACKE = 1 in EDMDR ($\overline{\text{EDACK}}$ pin enable)

3. $\overline{\text{EDREQ}}$ Pin Falling Edge Activation

Falling edge sensing on the $\overline{\text{EDREQ}}$ pin is performed in synchronization with EXDMAC internal operations, as indicated below.

1. Activation request standby state: Waits for low level sensing on $\overline{\text{EDREQ}}$ pin, then [2].
2. Transfer standby state: Waits for EXDMAC data transfer to become possible, then [3].
3. Activation request disabled state: Waits for high level sensing on $\overline{\text{EDREQ}}$ pin, then [1].

After EXDMAC transfer is enabled, the EXDMAC goes to state [1], so low level sensing is used for the initial activation after transfer is enabled.

When the transfer between the transfer source/destination and the cluster buffer completes, another cluster transfer, the transferred data in the cluster buffer may be overwritten by the channel cluster transfer. Therefore, in the cluster transfer mode (single address mode), do not set the cluster transfer mode for any other channels.

6. Cluster Transfer Mode and Endian

In cluster transfer mode, only a transfer to the areas in the big endian format is supported. When cluster transfer mode is specified, do not specify the areas in the little endian format (EDSAR and EDDAR). For details on the endian, see section 9, Bus Controller (BSC).

- 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 select 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 select 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 transfer.
 - 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

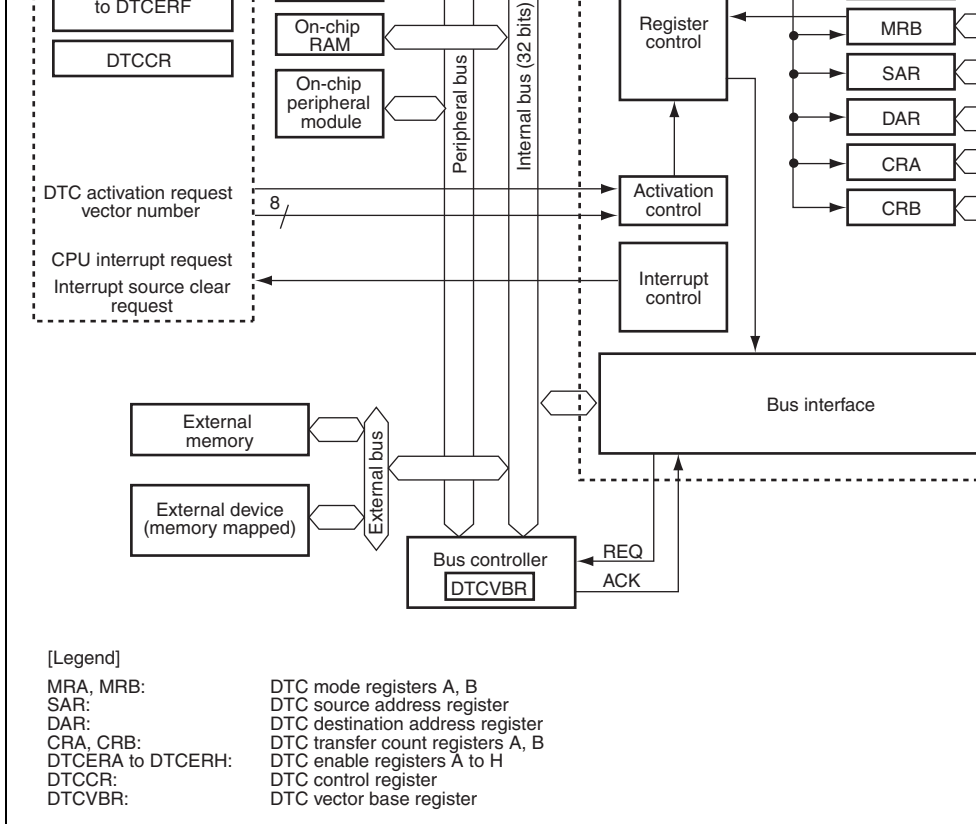


Figure 12.1 Block Diagram of DTC

These six registers MRA, MRB, SAR, DAR, CRA, and CRB cannot be directly accessed by the CPU. The contents of these registers are stored in the data area as transfer information. When a DTC activation request occurs, the DTC reads a start address of transfer information that is stored in the data area according to the vector address, reads the transfer information, and transfers it to the CPU. After the data transfer, it writes a set of updated transfer information back to the data area.

- DTC enable registers A to F (DTCERA to DTCERF)
- 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	—	<p>DTC Chain Transfer Enable</p> <p>Specifies the chain transfer. For details, see section 12.7.2, Chain Transfer. The chain transfer condition is selected by the CHNS bit.</p> <p>0: Disables the chain transfer 1: Enables the chain transfer</p>
6	CHNS	Undefined	—	<p>DTC Chain Transfer Select</p> <p>Specifies the chain transfer condition. If the condition for a chain transfer is met, the completion check of the specified transfer count is not performed and the source flag or DTCER is not cleared.</p> <p>0: Chain transfer every time 1: Chain transfer only when transfer counter = 0</p>
5	DISEL	Undefined	—	<p>DTC Interrupt Select</p> <p>When this bit is set to 1, a CPU interrupt request is generated every time after a data transfer ends. When this bit is set to 0, a CPU interrupt request is not generated when the specified number of data transfers ends.</p>
4	DTS	Undefined	—	<p>DTC Transfer Mode Select</p> <p>Specifies either the source or destination as repeat or block area during repeat or block transfer mode.</p> <p>0: Specifies the destination as repeat or block area 1: Specifies the source as repeat or block area</p>

1, 0 — Undefined — Reserved

The write value should always be 0.

[Legend]

x: Don't care

12.2.3 DTC Source Address Register (SAR)

SAR is a 32-bit register that designates the source address of data to be transferred by the DTC.

In full address mode, 32 bits of SAR are valid. In short address mode, the lower 24 bits of SAR are valid and bits 31 to 24 are ignored. At this time, the upper eight bits are filled with the value of bit 23.

If a word or longword access is performed while an odd address is specified in SAR or if a longword access is performed while address $4n + 2$ is specified in SAR, the bus cycle is divided into multiple cycles to transfer data. For details, see section 12.5.1, Bus Cycle Division.

SAR cannot be accessed directly from the CPU.

into multiple cycles to transfer data. For details, see section 12.5.1, Bus Cycle Division.

DAR cannot be accessed directly from the CPU.

12.2.5 DTC Transfer Count Register A (CRA)

CRA is a 16-bit register that designates the number of times data is to be transferred by

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 DTCE_n (n = 15 to 0) corresponding activation source is cleared and then an interrupt is requested to the CPU when the count reaches 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 lower eight bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and the contents of CRAH are sent to CRAL when the count reaches H'00. The transfer count is 1 when CRAH = CRAL = H'01, 255 when CRAH = CRAL = H'FF, and 256 when CRAH = CRAL = H'00.

In block transfer mode, CRA is divided into two parts: the upper eight bits (CRAH) and lower eight bits (CRAL). CRAH holds the block size while CRAL functions as an 8-bit block-transfer counter (1 to 256 for byte, word, or longword). CRAL is decremented by 1 every time a (byte, word or longword) data is transferred, and the contents of CRAH are sent to CRAL when the count reaches H'00. The block size is 1 byte (word or longword) when CRAH = CRAL = H'01, 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.

12.2.7 DTC enable registers A to H (DTCERA to DTCERH)

DTCER, which is comprised of eight registers, DTCERA to DTCERH, is a register that stores DTC activation interrupt sources. The correspondence between interrupt sources and DTCER is shown in table 12.1. Use bit manipulation instructions such as BSET and BCLR to read or write the DTCE bit. If all interrupts are masked, multiple activation sources can be set at one time (from the initial setting) by writing data after executing a dummy read on the relevant register.

Bit	15	14	13	12	11	10	9	8
Bit Name	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
Initial Value	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

7	DTCE7	0	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

These bits are not cleared when the DISEL bit is set. If the specified number of transfers have not ended

12.2.8 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 value always be 0.

0: Transfer read skip is not performed.

1: Transfer read skip is performed when the values of the transfer numbers match.

3	RCHNE	0	R/W	Chain Transfer Enable After DTC Repeat Transfer Enables/disables the chain transfer while transfer counter (CRAL) is 0 in repeat transfer mode. In repeat transfer mode, the CRAH value is written to CRAL when CRAL is 0. Accordingly, chain transfer does not occur when CRAL is 0. If this bit is set to 1, chain transfer is enabled when CRAH is written to CRAL. 0: Disables the chain transfer after repeat transfer 1: Enables the chain transfer after repeat transfer
2, 1	—	All 0	R	Reserved These are read-only bits and cannot be modified.
0	ERR	0	R/(W)*	Transfer Stop Flag Indicates that an address error or an NMI interrupt occurs. If an address error or an NMI interrupt occurs, the DTC stops. 0: No interrupt occurs 1: An interrupt occurs [Clearing condition] <ul style="list-style-type: none">• 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	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

12.3 Activation Sources

The DTC is activated by an interrupt request. The interrupt source is selected by DTCEB. The activation source can be selected by setting the corresponding bit in DTCER; the CPU interrupt source can be selected by clearing the corresponding bit in DTCER. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source interrupt corresponding DTCER bit is cleared.

12.4 Location of Transfer Information and DTC Vector Table

Locate the transfer information in the data area. The start address of transfer information is located at the address that is a multiple of four (4n). Otherwise, the lower two bits are ignored during access ([1:0] = B'00.) Transfer information can be located in either short address mode (three longwords) or full address mode (four longwords). The DTCMD bit in SYSCR selects either short address mode (DTCMD = 1) or full address mode (DTCMD = 0). For details, see section 3.2.2, System Control Register (SYSCR). Transfer information located in the data area is shown in figure 12.2

The DTC reads the start address of transfer information from the vector table according to the activation source, and then reads the transfer information from the start address. Figure 12.2 shows correspondences between the DTC vector address and transfer information.



Figure 12.2 Transfer Information on Data Area

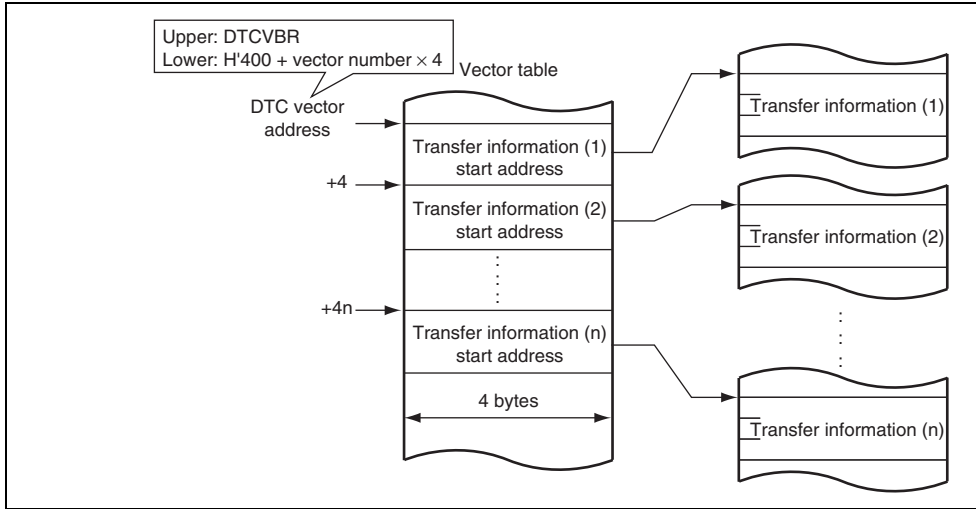


Figure 12.3 Correspondence between DTC Vector Address and Transfer Information

	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
	IRQ12	76	H'530	DTCEA3
	IRQ13	77	H'534	DTCEA2
	IRQ14	78	H'538	DTCEA1
	IRQ15	79	H'53C	DTCEA0
A/D_0	AD10 (A/D_0 conversion end)	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

TMR_2	CMI2A	122	H'5E8	DTCEC9
	CMI2B	123	H'5EC	DTCEC8
TMR_3	CMI3A	125	H'5F4	DTCEC7
	CMI3B	126	H'5F8	DTCEC6
DMAC	DMTEND0	128	H'600	DTCEC5
	DMTEND1	129	H'604	DTCEC4
	DMTEND2	130	H'608	DTCEC3
	DMTEND3	131	H'60C	DTCEC2
EXDMAC* ²	EXDMTEND0	132	H'610	DTCEC1
	EXDMTEND1	133	H'614	DTCEC0
	EXDMTEND2	134	H'618	DTCED15
	EXDMTEND3	135	H'61C	DTCED14
DMAC	DMEEND0	136	H'620	DTCED13
	DMEEND1	137	H'624	DTCED12
	DMEEND2	138	H'628	DTCED11
	DMEEND3	139	H'62C	DTCED10
EXDMAC* ²	EXDMEEND0	140	H'630	DTCED9
	EXDMEEND1	141	H'634	DTCED8
	EXDMEEND2	142	H'638	DTCED7
	EXDMEEND3	143	H'63C	DTCED6

SCI_4	RXI4	161	H'684	DTCEE13
	TXI4	162	H'688	DTCEE12
TPU_6	TGI6A	164	H'690	DTCEE11
	TGI6B	165	H'694	DTCEE10
	TGI6C	166	H'698	DTCEE9
	TGI6D	167	H'69C	DTCEE8
	TGI7A	169	H'6A4	DTCEE7
TPU_7	TGI7B	170	H'6A8	DTCEE6
	TGI8A	173	H'6B4	DTCEE5
TPU_8	TGI8B	174	H'6B8	DTCEE4
	TGI9A	177	H'6C4	DTCEE3
TPU_9	TGI9B	178	H'6C8	DTCEE2
	TGI9C	179	H'6CC	DTCEE1
	TGI9D	180	H'6D0	DTCEE0
	TGI10A	182	H'6D8	DTCEF15
TPU_10	TGI10B	183	H'6DC	DTCEF14
	TGI10V	186	H'6E8	DTCEF11
	TGI11A	188	H'6F0	DTCEF10
TPU_11	TGI11B	189	H'6F4	DTCEF9

- Notes:
1. The DTCE bits with no corresponding interrupt are reserved, and the write value always be 0. To leave software standby mode or all-module-clock-stop mode interrupt, write 0 to the corresponding DTCE bit.
 2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Table 12.2 shows the DTC transfer modes.

Table 12.2 DTC Transfer Modes

Transfer Mode	Size of Data Transferred at One Transfer Request	Memory Address Increment or Decrement	Tr C
Normal	1 byte/word/longword	Incremented/decremented by 1, 2, or 4, 1 or fixed	
Repeat* ¹	1 byte/word/longword	Incremented/decremented by 1, 2, or 4, 1 or fixed	
Block* ²	Block size specified by CRAH (1 to 256 bytes/words/longwords)	Incremented/decremented by 1, 2, or 4, 1 or fixed	

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 continue operation.

Setting the CHNE bit in MRB to 1 makes it possible to perform a number of transfers with 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 12.4 shows a flowchart of DTC operation, and table 12.3 summarizes the chain transfer conditions (combinations for performing the second and third transfers are omitted).

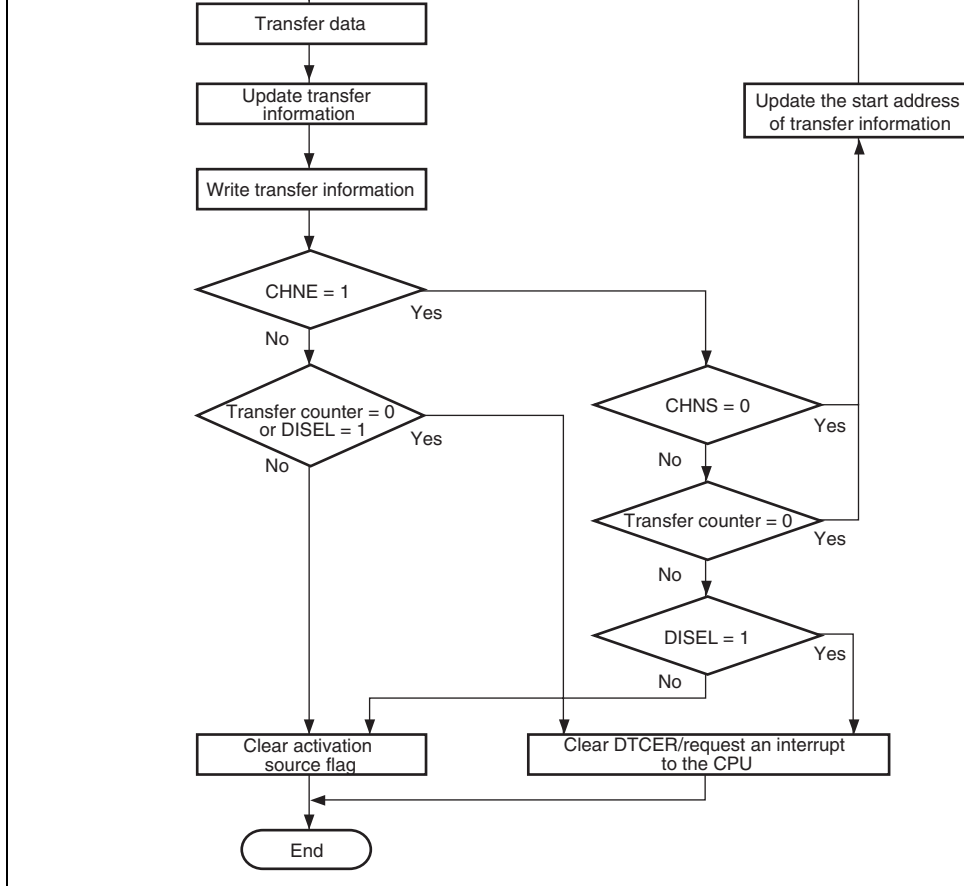


Figure 12.4 Flowchart of DTC Operation

1	1	0	Not 0	—	—	—	—	Ends at 1st transfer
1	1	—	0*2	0	—	0	Not 0	Ends at 2nd transfer
				0	—	0	0*2	Ends at 2nd transfer
				0	—	1		Interrupt request
1	1	1	Not 0	—	—	—	—	Ends at 1st transfer Interrupt request

- Notes: 1. CRA in normal mode transfer, CRAL in repeat transfer mode, or CRB in block transfer mode
2. When the contents of the CRAH is written to the CRAL in repeat transfer mode

12.5.1 Bus Cycle Division

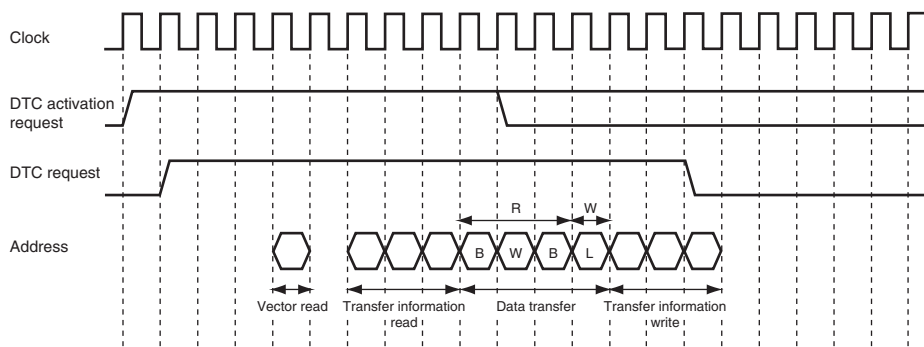
When the transfer data size is word and the SAR and DAR values are not a multiple of 2, the bus cycle is divided and the transfer data is read from or written to in bytes. Similarly, when the transfer data size is longword and the SAR and DAR values are not a multiple of 4, the bus cycle is divided and the transfer data is read from or written to in words.

Table 12.4 shows the relationship among, SAR, DAR, transfer data size, bus cycle division, and access data size. Figure 12.5 shows the bus cycle division example.

Table 12.4 Number of Bus Cycle Divisions and Access Size

SAR and DAR Values	Specified Data Size		
	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]

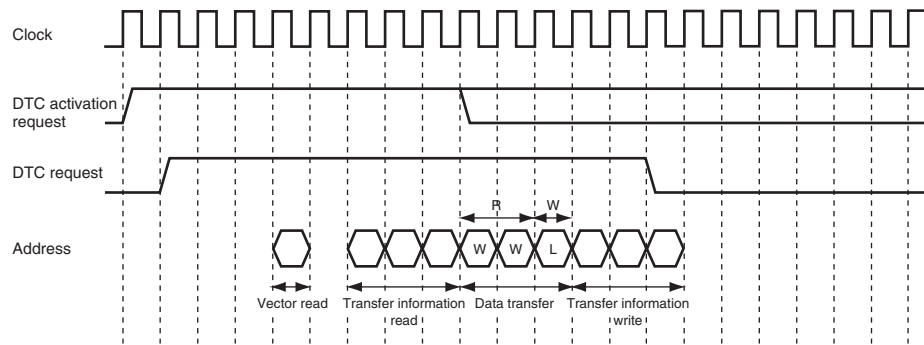


Figure 12.5 Bus Cycle Division Example

cleared to 0, the stored vector number is deleted, and the updated vector table and transfer information are read at the next activation.

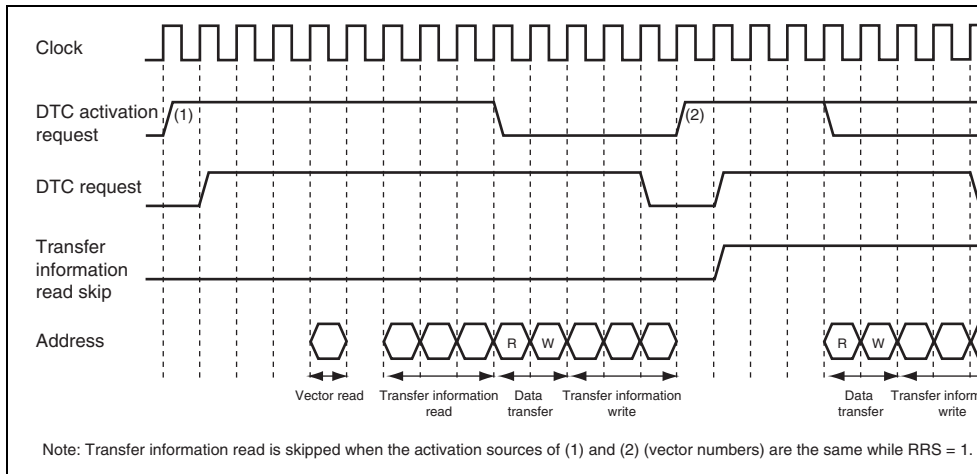


Figure 12.6 Transfer Information Read Skip Timing

SM1	DM1	SAR	DAR
0	0	Skipped	Skipped
0	1	Skipped	Written back
1	0	Written back	Skipped
1	1	Written back	Written back

12.5.4 Normal Transfer Mode

In normal transfer mode, one operation transfers one byte, one word, or one longword of data. From 1 to 65,536 transfers can be specified. The transfer source and destination addresses can be specified as incremented, decremented, or fixed. When the specified number of transfers is completed, an interrupt can be requested to the CPU.

Table 12.6 lists the register function in normal transfer mode. Figure 12.7 shows the memory access sequence in normal transfer mode.

Table 12.6 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 12.7 Memory Map in Normal Transfer Mode

12.5.5 Repeat Transfer Mode

In repeat transfer mode, one operation transfers one byte, one word, or one longword of data. When the DTS bit in MRB, either the source or destination can be specified as a repeat area. From 1 to 256 transfers can be specified. When the specified number of transfers ends, the transfer counter and address register specified as the repeat area is restored to the initial state, and transfer is repeated. The other address register is then incremented, decremented, or left fixed. In repeat transfer mode, the transfer counter (CRAL) is updated to the value specified in CRAH when CRAL becomes H'00. Thus the transfer counter value does not reach H'00, and therefore interrupt cannot be requested when DISEL = 0.

Table 12.7 lists the register function in repeat transfer mode. Figure 12.8 shows the memory map in 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.

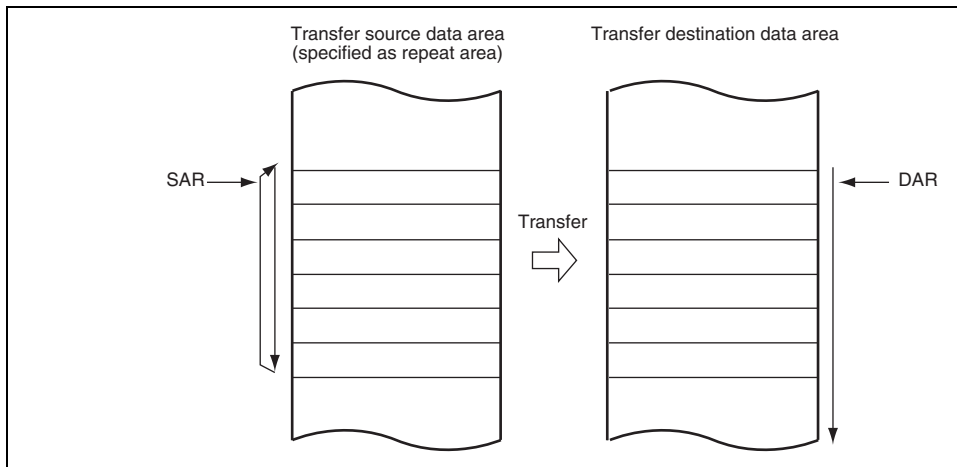


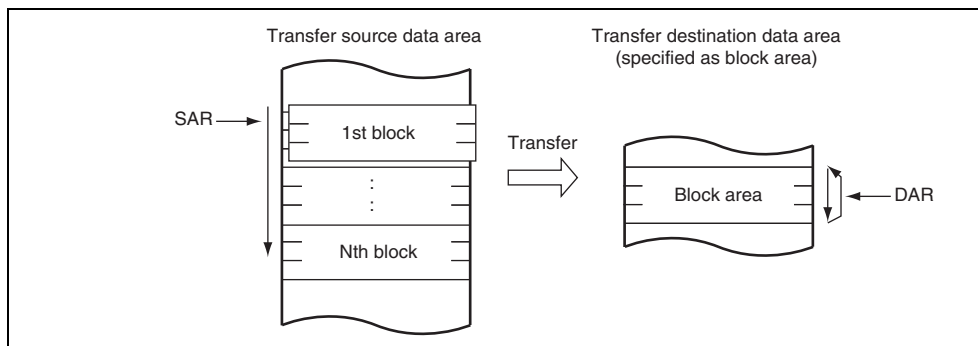
Figure 12.8 Memory Map in Repeat Transfer Mode (When Transfer Source is Specified as Repeat Area)

Table 12.8 lists the register function in block transfer mode. Figure 12.9 shows the memory map in block transfer mode.

Table 12.8 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 12.9 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 counter = 1 has been completed.

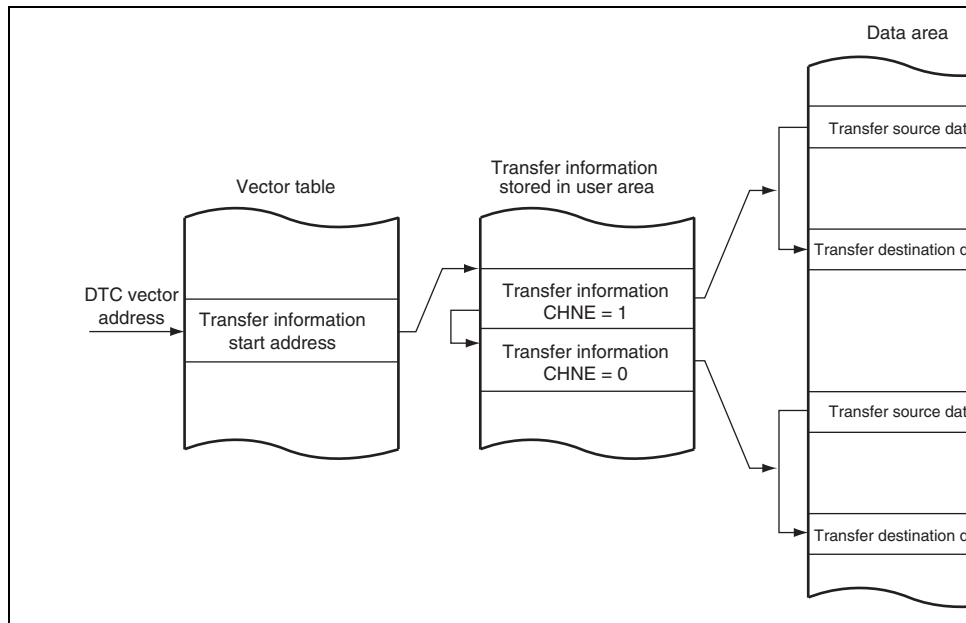


Figure 12.10 Operation of Chain Transfer

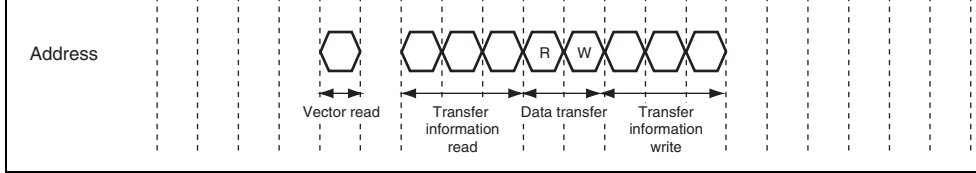


Figure 12.11 DTC Operation Timing
(Example of Short Address Mode in Normal Transfer Mode or Repeat Transfer Mode)

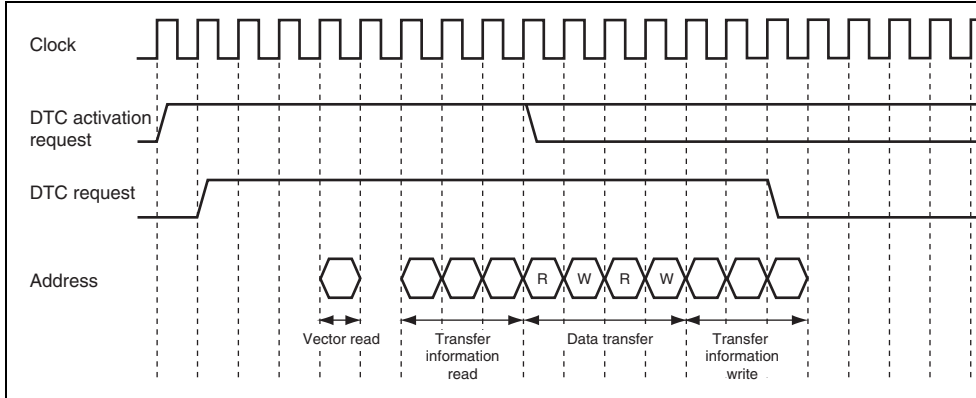


Figure 12.12 DTC Operation Timing
(Example of Short Address Mode in Block Transfer Mode with Block Size of 8)

Figure 12.13 DTC Operation Timing (Example of Short Address Mode in Chain

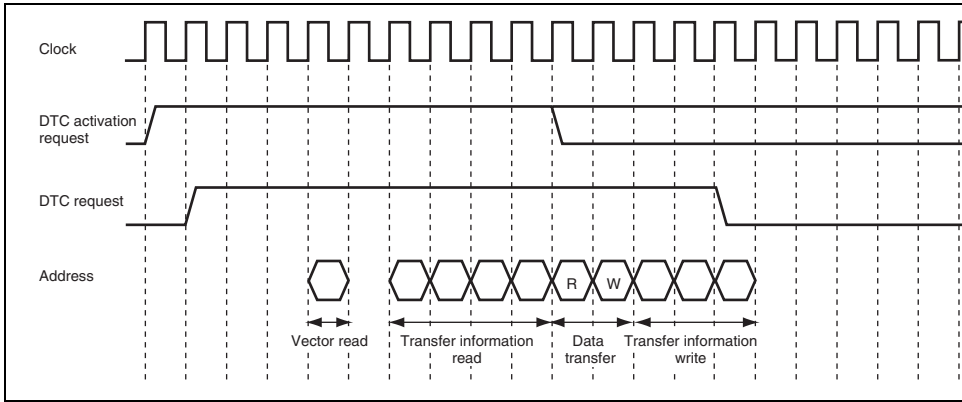


Figure 12.14 DTC Operation Timing (Example of Full Address Mode in Normal Transfer Mode or Repeat Transfer)

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	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	1

[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 address register
 7. When a word is transferred while an odd address is specified in the address register 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 9, Bus Controller (BSC).)

The number of execution cycles is calculated from the formula below. Note that Σ means the sum of all transfers activated by one activation event (the number in which the CHNE bit is set plus 1).

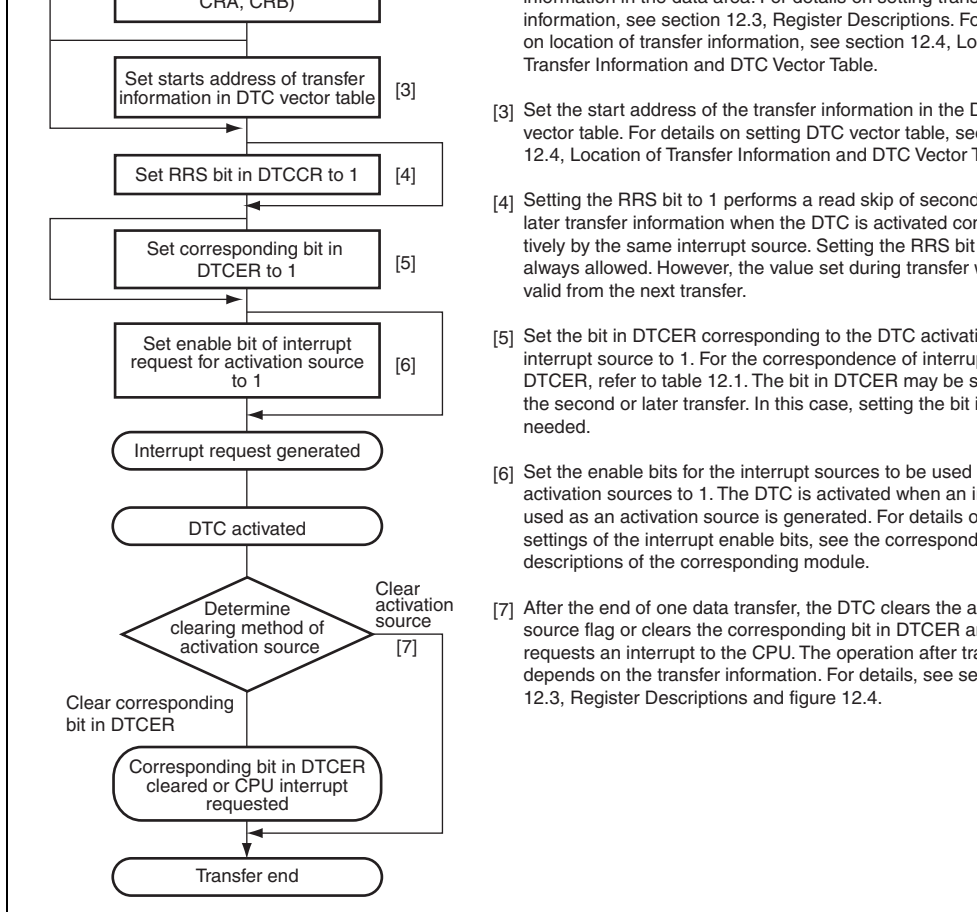
$$\text{Number of execution cycles} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L + M \cdot S_M) + N \cdot S_N$$

12.5.10 DTC Bus Release Timing

The DTC requests the bus mastership to the bus arbiter when an activation request occurs. The 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 information read, single data transfer, or transfer information writeback.

12.5.11 DTC Priority Level Control to the CPU

The priority of the DTC activation sources over the CPU can be controlled by the CPU priority level specified by bits CPUP2 to CPUP0 in CPUPCR and the DTC priority level specified by bits DTCP2 to DTCP0. For details, see section 7, Interrupt Controller.



- [3] Set the start address of the transfer information in the DTC vector table. For details on setting DTC vector table, see section 12.4, Location of Transfer Information and DTC Vector Table.
- [4] Setting the RRS bit to 1 performs a read skip of second or later transfer information when the DTC is activated consecutively by the same interrupt source. Setting the RRS bit is always allowed. However, the value set during transfer is invalid from the next transfer.
- [5] Set the bit in DTCER corresponding to the DTC activation interrupt source to 1. For the correspondence of interrupt sources and DTCER, refer to table 12.1. The bit in DTCER may be set for the second or later transfer. In this case, setting the bit is not needed.
- [6] Set the enable bits for the interrupt sources to be used as activation sources to 1. The DTC is activated when an interrupt request is generated. For details on interrupt enable settings of the interrupt enable bits, see the corresponding descriptions of the corresponding module.
- [7] After the end of one data transfer, the DTC clears the activation source flag or clears the corresponding bit in DTCER and requests an interrupt to the CPU. The operation after transfer depends on the transfer information. For details, see section 12.3, Register Descriptions and figure 12.4.

Figure 12.15 DTC with Interrupt Activation

- the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any
2. Set the start address of the transfer information for an RXI interrupt at the DTC vector.
 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 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. An RXI interrupt is generated, and the DTC is activated. The receive data is transferred to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.
 6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held. The DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. Termination processing should be performed in the interrupt handling routine.

12.7.2 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 updating. Repeat mode transfer to the PPG's NDR is performed in the first half of the chain transfer, and normal mode transfer to the TPU's TGR in the second half. This is because of the activation source and interrupt generation at the end of the specified number of transfers is restricted to the second half of the chain transfer (transfer when CHNE = 0).

3. Locate the TPU transfer information consecutively after the NDR transfer information.
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 the interrupt with TIER.
7. Set the initial output value in PODR, and the next output value in NDR. Set bits in DTCER for which output is to be performed to 1. Using PCR, select the TPU compare to 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 NDR. The set value of the next output trigger period is transferred to TGRA. The activation source flag is cleared.
10. When the specified number of transfers are completed (the TPU transfer CRA value is 0), the TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is sent to the CPU. Termination processing should be performed in the interrupt handling routine.

12.7.3 Chain Transfer when Counter = 0

By executing a second data transfer and performing re-setting of the first data transfer once the counter value is 0, it is possible to perform 256 or more repeat transfers.

An example is shown in which a 128-kbyte input buffer is configured. The input buffer is assumed to have been set to start at lower address H'0000. Figure 12.16 shows the chain transfer when the counter value is 0.

- for the first data transfer reaches 0, the second data transfer is started. Set the upper 8 bits of the transfer source address for the first data transfer to H'21. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
- Next, execute the first data transfer the 65536 times specified for the first data transfer by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper eight bits of the transfer source address for the second data transfer to H'20. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
 - Steps 4 and 5 are repeated endlessly. As repeat mode is specified for the second data transfer, no interrupt request is sent to the CPU.

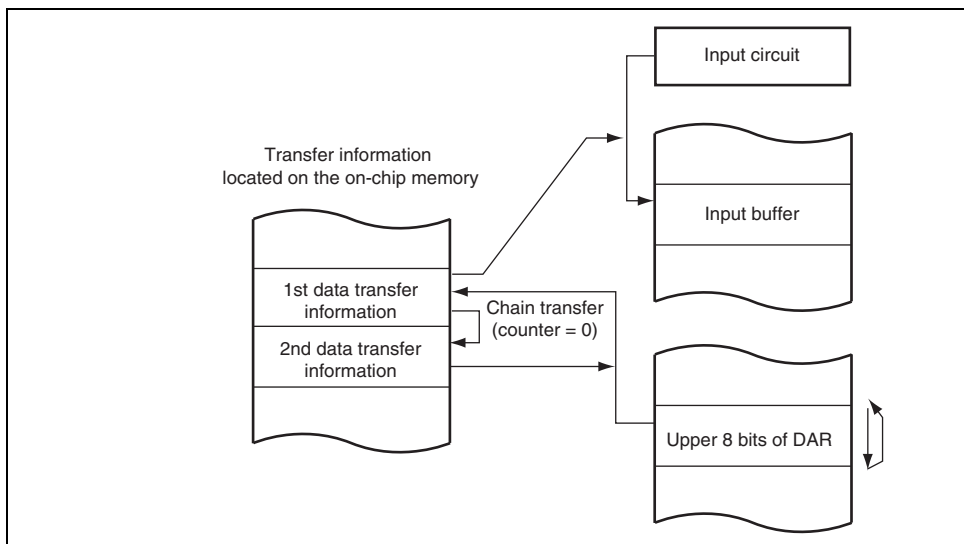


Figure 12.16 Chain Transfer when Counter = 0

Operation of the DTC can be disabled or enabled using the module stop control register. The initial setting is for operation of the DTC to be enabled. Register access is disabled by set module stop state. The module stop state cannot be set while the DTC is activated. For details refer to section 27, Power-Down Modes.

12.9.2 On-Chip RAM

Transfer information can be located in on-chip RAM. In this case, the RAME bit in SYSCON cannot be cleared to 0.

12.9.3 DMAC Transfer End Interrupt

When the DTC is activated by a DMAC transfer end interrupt, the DTE bit of DMDR is not controlled by the DTC but its value is modified with the write data regardless of the transfer counter value and DISEL bit setting. Accordingly, even if the DTC transfer counter value becomes 0, no interrupt request may be sent to the CPU in some cases.

When the DTC is activated by a DMAC transfer end interrupt, even if DISEL=0, an automatic clearing of the relevant activation source flag is not automatically cleared by the DTC. To clear the activation source flag, write 1 to the DTE bit by the DTC transfer and clear the activation source flag to 0.

12.9.4 DTCE Bit Setting

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all interrupts are disabled, multiple activation sources can be set at one time (only at the initial setting). After setting, write data after executing a dummy read on the relevant register.

The transfer information start address to be specified in the vector table should be address other than address 4n is specified, the lower 2 bits of the address are regarded as 0.

The source and destination addresses specified in SAR and DAR, respectively, will be transferred in the divided bus cycles depending on the address and data size.

12.9.7 Transfer Information Modification

When IBCCS = 1 and the DMAC is used, clear the IBCCS bit to 0 and then set to 1 again, modifying the DTC transfer information in the CPU exception handling routine initiated by the transfer end interrupt.

12.9.8 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 the same.

12.9.9 Points for Caution when Overwriting DTCER.

When overwriting of the DTC-transfer enable register (DTCER) and the generation of an interrupt that is a source for DTC activation are in competition, activation of the DTC and interrupt exception processing by the CPU will both proceed at the same time. Depending on the timing at this time, doubling of interrupts may occur. If there is a possibility of competition between overwriting of the DTCER and generation of an interrupt that is a source for DTC activation, proceed with overwriting of the DTCER according to the relevant procedure given below.

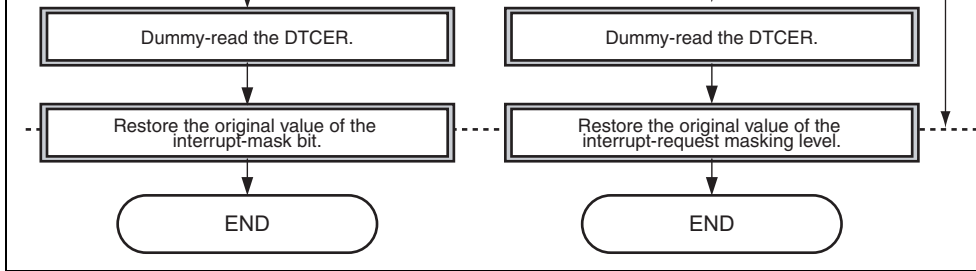


Figure 12.17 Example of Procedures for Overwriting the DTCER

Ports 2 and F include an open-drain control register (ODR) that controls on/off of the output buffer PMOSs.

Port N supports 5-V input and functions as an NMOS open-drain output pin.

All of the I/O ports can drive a single TTL load with a capacitive component of up to 30 pF. They can also drive Darlington transistors when functioning as output ports.

Pins on ports 2, 3, J, and K have Schmitt-trigger inputs. Schmitt-trigger input is enabled on other ports when they are used as IRQ, TPU, TMR, or IIC2 inputs.

Table 13.1 Port Functions

Port	Description	Bit	I/O	Function		Schmitt-Trigger Input* ¹	Input Pull-up MOS Function
				Input	Output		
Port 1	General I/O port function multiplexed with interrupt input, SCI I/O, DMAC I/O, EXDMAC I/O* ⁷ , A/D converter input, TPU input, and IIC2 I/O	7	P17/SCL0	$\overline{\text{IRQ7-A}}$ / TCLKD-B/ $\overline{\text{ADTRG1-A}}$	$\overline{\text{EDRAK1-A}}^{*7}$	$\overline{\text{IRQ7-A}}$, TCLKD-B, SCL0	—
		6	P16/SDA0/SCK3	$\overline{\text{IRQ6-A}}$ / TCLKC-B	$\overline{\text{DACK1-A}}$ / $\overline{\text{EDACK1-A}}^{*7}$	$\overline{\text{IRQ6-A}}$, TCLKC-B, SDA0	
		5	P15/SCL1	$\overline{\text{IRQ5-A}}$ / TCLKB-B/ RxD3	$\overline{\text{TEND1-A}}$ / $\overline{\text{ETEND1-A}}^{*7}$	$\overline{\text{IRQ5-A}}$, TCLKB-B, SCL1	
		4	P14/SDA1	$\overline{\text{DREQ1-A}}$ / $\overline{\text{IRQ4-A}}$ / TCLKA-B/ $\overline{\text{EDREQ1-A}}^{*7}$	TxD3	$\overline{\text{IRQ4-A}}$, TCLKA-B, SDA1	
		3	P13	$\overline{\text{ADTRG0-A}}$ / $\overline{\text{IRQ3-A}}$	$\overline{\text{EDRAK0-A}}^{*7}$	$\overline{\text{IRQ3-A}}$	

Port 2	General I/O port function multiplexed with interrupt input, PPG output, TPU I/O, TMR I/O, and SCI I/O	7	P27/ TIOCB5	TIOCA5/ $\overline{\text{IRQ15-A}}^{*6}$	PO7	All input functions	—
		6	P26/ TIOCA5	$\overline{\text{IRQ14-A}}$	PO6/TMO1/ TxD1	All input functions	
		5	P25/ TIOCA4	TMCI1/ RxD1/ $\overline{\text{IRQ13-A}}$	PO5	P25, TIOCA4, TMCI1, $\overline{\text{IRQ13-A}}$	
		4	P24/ TIOCB4/ SCK1	TIOCA4/ TMRI1/ $\overline{\text{IRQ12-A}}$	PO4	P24, TIOCB4, TIOCA4, TMRI1, $\overline{\text{IRQ12-A}}$	
		3	P23/ TIOCD3	$\overline{\text{IRQ11-A}}/$ TIOCC3	PO3	All input functions	
		2	P22/ TIOCC3	$\overline{\text{IRQ10-A}}$	PO2/TMO0/ TxD0	All input functions	
		1	P21/ TIOCA3	TMCI0/ RxD0/ $\overline{\text{IRQ9-A}}$	PO1	P21, TIOCA3, TMCI0, $\overline{\text{IRQ9-A}}$	
		0	P20/ TIOCB3/ SCK0	TIOCA3/ TMRI0/ $\overline{\text{IRQ8-A}}$	PO0	P20, TIOCB3, TIOCA3, TMRI0, $\overline{\text{IRQ8-A}}$	

3	—	P53/AN3/ $\overline{\text{IRQ3-B}}$	—	$\overline{\text{IRQ3-B}}$
2	—	P52/AN2/ $\overline{\text{IRQ2-B}}$	—	$\overline{\text{IRQ2-B}}$
1	—	P51/AN1/ $\overline{\text{IRQ1-B}}$	—	$\overline{\text{IRQ1-B}}$
0	—	P50/AN0/ $\overline{\text{IRQ0-B}}$	—	$\overline{\text{IRQ0-B}}$

		3	P63	TMRI3/ DREQ3/ IRQ11-B/ TMS/ EDREQ1-B*7	TxD6	TMRI3, IRQ11-B, TMS		
		2	P62/SCK4	IRQ10-B/ TRST	TMO2/ DACK2 EDACK0-B*7	IRQ10-B, TRST		
		1	P61	TMCI2/ RxD4/ IRQ9-B	TEND2/ ETEND0-B*7	TMCI2, IRQ9-B		
		0	P60	TMRI2/ DREQ2/ IRQ8-B/ EDREQ0-B*7	TxD4	TMRI2, IRQ8-B		
Port A	General I/O port function multiplexed with system clock output and bus control I/O	7	—	PA7	B ϕ	—	—	
		6	PA6	—	AS/AH/ BS-B			
		5	PA5	—	RD			
		4	PA4	—	LHWR/LUB			
		3	PA3	—	LLWR/LLB			
		2	PA2	—	BREQ/ WAIT-A			
		1	PA1	—	BACK/ (RD/WR-A)			
		0	PA0	—	BREQ0/ BS-A			

		3	PB3	—	—	CS3-A/ CS7-A CAS* ⁷	
		2	PB2	—	—	CS2-A/ CS6-A/ RAS* ⁷	
		1	PB1	—	—	CS1/ CS2-B/ CS5-A/ CS6-B/ CS7-B	
		0	PB0	—	—	CS0/ CS4-A/ CS5-B	
Port C	General I/O port function multiplexed with bus control I/O and A/D converter input	7	—	—	—	—	
		6	—	—	—	—	
		5	PC5	—	—	—	—
		4	PC4	ADTRG2	—	—	—
		3	PC3	—	—	LLCAS* ⁷ / DQMLL* ⁷	—
		2	PC2	—	—	LUCAS* ⁷ / DQMLU* ⁷	—
		1	PC1	—	—	CS4-C/ CS5-C/ CS6-C/ CS7-C	—
		0	PC0	—	WAIT-B/ ADTRG1-B	CS3-B	—

		1	PD1	—	A1		
		0	PD0	—	A0		
Port E ^{*3}	General I/O port function multiplexed with address output	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 function multiplexed with address output and SCI I/O	7	PF7/SCK5	—	A23	—	O
		6	PF6	RxD5/IrRxD	A22		
		5	PF5	—	A21/ TxD5/ IrTXD/		
		4	PF4	—	A20		
		3	PF3	—	A19		
		2	PF2	—	A18		
		1	PF1	—	A17		
		0	PF0	—	A16		

		1	PH1/D1*	—	—		
		0	PH0/D0* ²	—	—		
Port I	General I/O port function multiplexed with 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 J* ⁴	General I/O port function multiplexed with PPG and TPU I/O	7	PJ7/ TIOCB8	TIOCA8/ TCLKH	PO23	All input functions	O
		6	PJ6/ TIOCA8	—	PO22	All input functions	
		5	PJ5/ TIOCB7	TIOCA7/ TCLKG	PO21	All input functions	
		4	PJ4/ TIOCA7	—	PO20	All input functions	
		3	PJ3/ TIOCD6	TIOCC6/ TCLKF	PO19	All input functions	
		2	PJ2/ TIOCC6	TCLKE	PO18	All input functions	
		1	PJ1/ TIOCB6	TIOCA6	PO17	All input functions	
		0	PJ0/ TIOCA6	—	PO16	All input functions	

		3	PK3/ TIOCD9	TIOCC9	PO27	All input functions
		2	PK2/ TIOCC9	—	PO26	All input functions
		1	PK1/ TIOCB9	TIOCA9	PO25	All input functions
		0	PK0/ TIOCA9	—	PO24	All input functions
Port M*7	General I/O port	7	—	—	—	—
		6	—	—	—	—
		5	—	—	—	—
		4	PM4*7	—	—	—
		3	PM3*7	—	—	—
		2	PM2*7	—	—	—
		1	PM1*7	—	—	—
		0	PM0*7	—	—	—

1	PN1/SCL2	—	—	SCL2
0	PN0/SDA2	—	—	SDA2

- 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/O space.
 3. Ports D and E are disabled when PCJKE = 1.
 4. Ports J and K are disabled when PCJKE = 0.
 5. Output on the pins of port N is always NMOS open-drain output.
 6. Supported only by the H8SX/1648 Group, H8SX/1648A Group, and H8SX/1648L Group.
 7. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Port 3	8	0	0	0	0	—
Port 4	4	—	—	0	0	—
Port 5	8	—	—	0	0	—
Port 6	8	0	0	0	0	—
Port A	8	0	0	0	0	—
Port B	8	0	0	0	0	—
Port C* ¹	6	0	0	0	0	—
Port D* ²	8	0	0	0	0	0
Port E* ²	8	0	0	0	0	0
Port F	8	0	0	0	0	0
Port H	8	0	0	0	0	0
Port I	8	0	0	0	0	0
Port J* ³	8	0	0	0	0	0
Port K* ³	8	0	0	0	0	0
Port M* ⁴	5	0	0	0	0	—
Port N	4	0	0	0	0	—

[Legend]

O: Register exists

—: No register exists

- Notes:
1. Write the initial value to any of bits in port C registers.
 2. Do not access port D or E registers when PCJKE = 1.
 3. Do not access port J or K registers when PCJKE = 0.
 4. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

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

Notes: The lower six bits of port C registers are effective while the upper two bits are reserved.
The lower five bits of port M registers are effective while the upper three bits are reserved.
The lower four bits of port N registers are effective while the upper four bits are reserved.
Do not access port J or port K registers when PCJKE = 0.
Do not access port D or port E registers when PCJKE = 1.

Table 13.3 Startup Mode and Initial Value

Port	Startup Mode	
	External Extended Mode	Single-Chip Mode
Port A	H'80	H'00
Other ports	H'00	H'00

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

Notes: The lower six bits of port C registers are effective while the upper two bits are reserved.
 The lower five bits of port M registers are effective while the upper three bits are reserved.
 The lower four bits of port N registers are effective while the upper four bits are reserved.
 Do not access port J or port K registers when PCJKE = 0.
 Do not access port D or port E registers when PCJKE = 1.

13.1.3 Port Register (PORTn) (n = 1 to 6, A to F, H to K, M*, and N)

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

PORT is an 8-bit read-only register that reflects the port pin state. A write to PORT is ignored.
 When PORT is read, the DR bits that correspond to the respective DDR bits set to 1 are read.
 The status of each pin whose corresponding DDR bit is cleared to 0 is also read regardless of the 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
R/W	R	R	R	R	R	R	R

Notes: The upper four bits of port 4 registers are effective while the lower four bits are reserved.
 The lower six bits of port C registers are effective while the upper two bits are reserved.
 The lower five bits of port M registers are effective while the upper three bits are reserved.
 The lower four bits of port N registers are effective while the upper four bits are reserved.
 Do not access port J or port K registers when PCJKE = 0.
 Do not access port D or port E registers when PCJKE = 1.

or is used as an analog input/output pin.

When PORT is read, the pin state is always read regardless of the ICR value. When the ICR is cleared to 0 at this time, the read pin state is not reflected in a corresponding on-chip port module.

If ICR is modified, an internal edge may occur depending on the pin state. Accordingly, ICR should be modified when the corresponding input pins are not used. For example, an \overline{IRQ} should modify ICR while the corresponding interrupt is disabled, clear the IRQF flag in ISR of the interrupt controller to 0, and then enable the corresponding interrupt. If an edge occurs after the ICR setting, the edge should be cancelled.

The initial value of ICR is H'00.

Bit	7	6	5	4	3	2	1	0
Bit Name	Pn7ICR	Pn6ICR	Pn5ICR	Pn4ICR	Pn3ICR	Pn2ICR	Pn1ICR	Pn0ICR
Initial Value	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

Notes: The upper four bits of port 4 registers are effective while the lower four bits are reserved.
The lower six bits of port C registers are effective while the upper two bits are reserved.
The lower five bits of port M registers are effective while the upper three bits are reserved.
The lower four bits of port N registers are effective while the upper four bits are reserved.
Do not access port J or port K registers when PCJKE = 0.
Do not access port D or port E registers when PCJKE = 1.

Table 13.4 Input Pull-Up MOS State

Port	Pin State	Reset	Hardware Standby Mode	Software Standby Mode	O
Port D	Address output	OFF	OFF	OFF	O
	Port output	OFF	OFF	OFF	O
	Port input	OFF	OFF	ON/OFF	O
Port E	Address output	OFF	OFF	OFF	O
	Port output	OFF	OFF	OFF	O
	Port input	OFF	OFF	ON/OFF	O
Port F	Address output	OFF	OFF	OFF	O
	Peripheral module output	OFF	OFF	OFF	O
	Port output	OFF	OFF	OFF	O
	Port input	OFF	OFF	ON/OFF	O
Port H	Data input/output	OFF	OFF	OFF	O
	Port output	OFF	OFF	OFF	O
	Port input	OFF	OFF	ON/OFF	O
Port I	Data input/output	OFF	OFF	OFF	O
	Port output	OFF	OFF	OFF	O
	Port input	OFF	OFF	ON/OFF	O
Port J	Peripheral module output	OFF	OFF	OFF	O
	Port output	OFF	OFF	OFF	O
	Port input	OFF	OFF	ON/OFF	O

13.1.6 Open-Drain Control Register (PnODR) (n = 2 and F)

ODR is an 8-bit readable/writable register that selects the open-drain output function.

If a bit in ODR is set to 1, the pin corresponding to that bit in ODR functions as an NMO drain output. If a bit in ODR is cleared to 0, the pin corresponding to that bit in ODR functions as a CMOS output.

The initial value of ODR is H'00.

Bit	7	6	5	4	3	2	1	0
Bit Name	Pn7ODR	Pn6ODR	Pn5ODR	Pn4ODR	Pn3ODR	Pn2ODR	Pn1ODR	Pn0ODR
Initial Value	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

13.2 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: TIOCA_OE) indicates whether the output of the corresponding function is valid (1) or if another setting is specified (0). Table 13.5 lists each port output signal's valid setting. For details on the corresponding output signals, see the register description of each peripheral module. If the name of each peripheral module pin is followed by A or B, the pin function can be modified by the pin function control register (PFCR). For details, see section 13.3, Port Function Controller.

For a pin whose initial value changes according to the activation mode, "Initial value E" indicates the initial value when the LSI is started up in external extended mode and "Initial value S" indicates the initial value when the LSI is started in single-chip mode.

IIC2	SCL0 input/output	0	1	—
I/O port	P17 output	0	0	1
	P17 input (initial setting)	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

(2) P16/ $\overline{\text{DACK1-A}}$ / $\overline{\text{IRQ6-A}}$ / $\overline{\text{TCLKC-B}}$ / $\overline{\text{SDA0}}$ / $\overline{\text{SCK3}}$ / $\overline{\text{EDACK1-A}}$ *

The pin function is switched as shown below according to the combination of the EXDMAC, DMAC, SCI, and IIC2 register setting and P16DDR bit setting.

Module Name	Pin Function	Setting			
		EXDMAC*	DMAC	SCI	IIC2
		$\overline{\text{EDACK1A_OE}}$ *	$\overline{\text{DACK1A_OE}}$	$\overline{\text{SCK3_OE}}$	$\overline{\text{SDA0_OE}}$
EXDMAC*	$\overline{\text{EDACK1-A}}$ output*	1	—	—	—
DMAC	$\overline{\text{DACK1-A}}$ output	0	1	—	—
SCI	SCK3 output	0	0	1	—
IIC2	SDA0 input/output	0	0	0	1
I/O port	P16 output	0	0	0	0
	P16 input (initial setting)	0	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

	output				
IIC2	SCL1 input/output	0	0	1	—
I/O port	P15 output	0	0	0	1
	P15 input (initial setting)	0	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

(4) P14/TxD3/DREQ1-A/EDREQ1-A*/IRQ4-A/TCLKA-B/SDA1

The pin function is switched as shown below according to the combination of the SCI and IIC2 register setting and P14DDR bit setting.

Module Name	Pin Function	Setting		
		SCI	IIC2	I/O Port
		TxD3_OE	SDA1_OE	P14DDR
SCI	TxD3 output	1	—	—
IIC2	SDA1 input/output	0	1	—
I/O port	P14 output	0	0	1
	P14 input (initial setting)	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

(6) P12/SCK2/ $\overline{\text{DACK0-A}}$ / $\overline{\text{IRQ2-A}}$ / $\overline{\text{EDACK0-A}}$ *

The pin function is switched as shown below according to the combination of the EXDMAC DMAC and SCI register settings and P12DDR bit setting.

Module Name	Pin Function	Setting			
		EXDMAC*	DMAC	SCI	I/O
		$\overline{\text{EDACK0A_OE}}$ *	$\overline{\text{DACK0A_OE}}$	SCK2_OE	P12
EXDMAC*	$\overline{\text{EDACK0-A}}$ output*	1	—	—	—
DMAC	$\overline{\text{DACK0-A}}$ output	0	1	—	—
SCI	SCK2 output	0	0	1	—
I/O port	P12 output	0	0	0	1
	P12 input (initial setting)	0	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

I/O port	P11 output	0	0	1
	P11 input (initial setting)	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

(8) P10/TxD2/DREQ0-A/IRQ0-A/EDERQ0-A*

The pin function is switched as shown below according to the combination of the SCI register setting and P10DDR bit setting.

Module Name	Pin Function	Setting	
		SCI	I/O Port
		TxD2_OE	P10DDR
SCI	TxD2 output	1	—
I/O port	P10 output	0	1
	P10 input (initial setting)	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

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/ $\overline{\text{IRQ14}}$ -A

The pin function is switched as shown below according to the combination of the TPU, SCI, and PPG register settings and P26DDR bit setting.

Module Name	Pin Function	Setting			
		TPU	TMR	SCI	PPG
		TIOCA5_OE	TMO1_OE	TxD1_OE	PO6_OE
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
	P26 input (initial setting)	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/ $\overline{\text{IRQ12-A}}$

The pin function is switched as shown below according to the combination of the TPU, SCI, PPG register settings and P24DDR bit setting.

Module Name	Pin Function	Setting			
		TPU	SCI	PPG	I/O P
		TIOCB4_OE	SCK1_OE	PO4_OE	P24
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, SCI, and PPG register settings and P22DDR bit setting.

Module Name	Pin Function	Setting			
		TPU	TMR	SCI	PPG
		TIOCC3_OE	TMO0_OE	TxD0_OE	PO2_OE
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
	P22 input (initial setting)	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, SCI, PPG register settings and P20DDR bit setting.

Module Name	Pin Function	Setting			
		TPU	SCI	PPG	I/O P
		TIOCB3_OE	SCK0_OE	PO0_OE	P20
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

EXDMAC*	EDRAK3 output*	1	—	—	—
TPU	TIOCB2 output	0	1	—	—
PPG	PO15 output	0	0	1	—
I/O port	P37 output	0	0	0	1
	P37 input (initial setting)	0	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

(2) P36/PO14/TIOCA2/ $\overline{\text{EDRAK2}}$ *

The pin function is switched as shown below according to the combination of the EXDMAC* TPU and PPG register settings and P36DDR bit setting.

Module Name	Pin Function	Setting			
		EXDMAC*	TPU	PPG	I/O
		$\overline{\text{EDRAK2}}_{\text{OE}}$ *	TIOCA2_OE	PO14_OE	P36
EXDMAC*	$\overline{\text{EDRAK2}}$ output*	1	—	—	—
TPU	TIOCA2 output	0	1	—	—
PPG	PO14 output	0	0	1	—
I/O port	P36 output	0	0	0	1
	P36 input (initial setting)	0	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

		output					
TPU	TIOCB1	0	0	1	—	—	
	output						
PPG	PO13 output	0	0	0	1	—	
I/O port	P35 output	0	0	0	0	1	
	P35 input (initial setting)	0	0	0	0	0	

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

		output				
TPU	TIOCA1	0	0	1	—	—
	output					
PPG	PO12 output	0	0	0	1	—
I/O port	P34 output	0	0	0	0	1
	P34 input (initial setting)	0	0	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

I/O port	P33 output	0	0	1
	P33 input (initial setting)	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

(6) P32/PO10/TIOCC0/TCLKA-A/ $\overline{\text{DACK0-B}}$ / $\overline{\text{EDACK2}}$ *

The pin function is switched as shown below according to the combination of the EXDMAC, DMAC, TPU, and PPG register settings and P32DDR bit setting.

Module Name	Pin Function	Setting					I/O
		EXDMAC*	DMAC	TPU	PPG		
		$\overline{\text{EDACK2}}_{\text{OE}}$ *	$\overline{\text{DACK0B}}_{\text{OE}}$	TIOCC0_OE	PO10_OE	P32DDR	
EXDMAC*	$\overline{\text{EDACK2}}$ output*	1	—	—	—	—	—
DMAC	$\overline{\text{DACK0-B}}$ output	0	1	—	—	—	—
TPU	TIOCA1 output	0	0	1	—	—	—
PPG	PO12 output	0	0	0	1	—	—
I/O port	P34 output	0	0	0	0	0	1
	P34 input (initial setting)	0	0	0	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

		output				
TPU	TIOCB0	0	0	1	—	—
	output					
PPG	PO9 output	0	0	0	1	—
I/O port	P31 output	0	0	0	0	1
	P31 input (initial setting)	0	0	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

(8) P30/PO8/TIOCA0/DREQ0-B/EDREQ2*

The pin function is switched as shown below according to the combination of the TPU and PPG register settings and P33DDR bit setting.

Module Name	Pin Function	Setting		
		TPU	PPG	I/O Port
		TIOCA0_OE	PO8_OE	P33DDR
TPU	TIOCA0 output	1	—	—
PPG	PO8 output	0	1	—
I/O port	P30 output	0	0	1
	P30 input (initial setting)	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

13.2.5 Port 6

(1) P67/ $\overline{\text{IRQ15-B}}$ / $\overline{\text{EDRAK1-B}}$ *

- H8SX/1648 Group

The pin function is switched as shown below according to the combination of the SCI register setting and P67DDR bit setting.

Module Name	Pin Function	Setting	
		SCI	I/O Port
		$\overline{\text{SCK6_OE}}$	P67DDR
I/O port	P67 output	—	1
		1	0
	P67 input (initial setting)	0	0

- H8SX/1648A Group and H8SX/1648L Group

The pin function is switched as shown below according to the combination of the P67DDR register setting.

Module Name	Pin Function	Setting
		I/O Port
		P67DDR
I/O port	P67 output	1
	P67 input (initial setting)	0

(2) **P66/EDRAK-0-B***

- H8SX/1648 Group

The pin function is switched as shown below according to the combination of the SCI register setting and P66DDR bit setting.

Module Name	Pin Function	Setting	
		SCI	I/O Port
		TXD6_OE	P66DDR
I/O port	P66 output	—	1
		1	0
	P66 input (initial setting)	0	0

- H8SX/1648A Group and H8SX/1648L Group

The pin function is switched as shown below according to the combination of the P66DDR register setting.

Module Name	Pin Function	Setting
		I/O Port
		P66DDR
I/O port	P66 output	1
	P66 input (initial setting)	0

(3) P65/TMO3/DACK3/EDACK1-B*²/TCK/SCK6/IRQ13-B

The pin function is switched as shown below according to the combination of operation mode, EXDMAC*², DMAC, TMR, and SCI register settings and P65DDR bit setting.

Module Name	Pin Function	MCU Operating Mode	Setting				
			SCI	EXDMAC* ²	DMAC	TMR	I/O
			<u>SCK6_</u> OE	<u>EDACK1B_</u> OE* ²	<u>DACK3_OE</u>	<u>TMO3_</u> OE	<u>P65</u>
SCI	SCK6 output	Modes	1	—	—	—	—
EXDMAC* ²	<u>EDACK1-B</u> output* ²	other than the boundary scan enabled mode* ¹	0	1	0	—	—
DMAC	<u>DACK3</u> output		0	0	1	—	—
TMR	<u>TMO3</u> output		0	0	0	1	—
I/O port	<u>P65</u> output		0	0	0	0	1
	<u>P65</u> input (initial setting)		0	0	0	0	0

Notes: 1. These pins are boundary scan dedicated input pins during boundary scan enabled mode.

2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

I/O port	P64 output	mode ¹	0	0	1
	P64 input (initial setting)		0	0	0

- Notes: 1. These pins are boundary scan dedicated input pins during boundary scan enabled mode.
2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

(5) P63/TMRI3/DREQ3/ EDREQ1-B*²/IRQ11-B/TxD6/TMS

The pin function is switched as shown below according to the combination of operation mode, SCI register setting and P63DDR bit setting.

Module Name	Pin Function	MCU Operating Mode	Setting	
			SCI TxD6_OE	I/O Port P63DDR
SCI	TxD6 output	Modes other than the boundary scan enabled mode* ¹	1	—
I/O port	P63 output		0	1
	P63 input (initial setting)		0	0

- Notes: 1. These pins are boundary scan dedicated input pins during boundary scan enabled mode.
2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

DMAC	DACK2 output	the boundary scan enabled mode* ¹	0	1	—	—
TMR	TMO2 output		0	0	1	—
SCI	SCK4 output		0	0	0	1
I/O port	P62 output		0	0	0	0
	P62 input (initial setting)		0	0	0	0

- Notes: 1. These pins are boundary scan dedicated input pins during boundary scan enabled mode.
2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

(7) P61/TMCI2/RxD4/ $\overline{\text{TEND2}}$ / $\overline{\text{ETEND0-B}}$ / $\overline{\text{IRQ9-B}}$

The pin function is switched as shown below according to the combination of the EXDMAC register setting and P61DDR bit setting.

Module Name	Pin Function	Setting		
		EXDMAC*	DMAC	I/O Port
		$\overline{\text{ETEND0B_OE}}$ *	$\overline{\text{TEND2-OE}}$	P61DDR
EXDMAC*	$\overline{\text{ETEND0-B}}$ output*	1	—	—
DMAC	$\overline{\text{TEND2}}$ output	0	1	—
I/O port	P61 output	0	0	1
	P61 input (initial setting)	0	0	0

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

13.2.6 Port A

(1) PA7/B ϕ

The pin function is switched as shown below according to the PA7DDR bit setting.

Module Name	Pin Function	Setting
		I/O Port
		PA7DDR
I/O port	B ϕ 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

	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{RD}

The pin function is switched as shown below according to the combination of operating mode, EXPE bit, and the PA5DDR bit settings.

Module Name	Pin Function	Setting	
		MCU Operating Mode	I/O Port
		EXPE	PA5DDR
Bus controller	\overline{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 $\overline{\text{LHWR_OE}} = 1$, this pin functions as the $\overline{\text{LUB}}$ output; other $\overline{\text{LHWR}}$ 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 $\overline{\text{LLB}}$ output otherwise, the $\overline{\text{LLWR}}$.

(6) PA2/ $\overline{\text{BREQ}}$ / $\overline{\text{WAIT-A}}$

The pin function is switched as shown below according to the combination of the bus controller register setting and the PA2DDR bit setting.

Module Name	Pin Function	Setting		
		Bus Controller		I/O Port
		BCR_BRLE	BCR_WAITE	PA2DDR
Bus controller	$\overline{\text{BREQ}}$ input	1	—	—
	$\overline{\text{WAIT-A}}$ input	0	1	—
I/O port	PA2 output	0	0	1
	PA2 input (initial setting)	0	0	0

Bus controller	BACK output*	1	—	—	—
	$\overline{RD}/\overline{WR}$ -A 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/ \overline{BREQO} / $\overline{BS-A}$

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.

Module Name	Pin Function	Setting		
		I/O Port	Bus Controller	I/O Port
		$\overline{BS-A}_{OE}$	\overline{BREQO}_{OE}	PA0DD
Bus controller	$\overline{BS-A}$ output*	1	—	—
	\overline{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)

Module Name	Pin Function	SDRAM	$\overline{\text{CS7D}}_{\text{OE}}$	PB7DDR
Clock Pulse Generator	SD ϕ output* ^{1*3}	1	—	—
Bus controller	$\overline{\text{CS7-D}}$ output* ²	0	1	—
I/O port	PB7 output	0	0	1
	PB7 input (initial setting)	0	0	0

- Notes: 1. Valid in SDRAM mode
2. Valid in external extended mode (EXPE = 1)
3. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

(2) $\text{PB6}/\overline{\text{CS6-D}}/(\overline{\text{RD}}/\overline{\text{WR-B}})/\overline{\text{ADTRG0-B}}$

The pin function is switched as shown below according to the combination of operating mode, EXPE bit, bus controller register, port function control register (PFCR), and PB6DDR bit.

(initial setting)

Note: * Valid in external extended mode (EXPE = 1)

(3) $PB5/\overline{CS5-D}/\overline{OE}^{*2}/CKE^{*2}$

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 PB5DDR b

Module Name	Pin Function	Setting			
		Bus Controller		I/O Port	
		CKE_OE	\overline{OE} -OE	$\overline{CS5D}$ _OE	PB5
Bus controller	CKE output ^{*1*2}	1	—	—	—
	\overline{OE} output ^{*1*2}	0	1	—	—
	$\overline{CS5-D}$ output ^{*1}	0	0	—	—
I/O port	PB5 output	0	0	1	1
	PB5 input (initial setting)	0	0	0	0

Notes: 1. Valid in external extended mode (EXPE = 1)

2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

I/O port	PB4 output	0	0	1
	PB4 input (initial setting)	0	0	0

- Notes: 1. Valid in external extended mode (EXPE = 1)
2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

(5) $\overline{\text{PB3}}/\overline{\text{CS3-A}}/\overline{\text{CS7-A}}/\overline{\text{CAS}}^{*2}$

The pin function is switched as shown below according to the combination of operating mode, EXPE bit, port function control register (PFCR), and the PB3DDR bit settings.

Module Name	Pin Function	Setting			
		Bus Controller			I/O Port
		$\overline{\text{CAS}}_{\text{OE}}$	$\overline{\text{CS3A}}_{\text{OE}}$	$\overline{\text{CS7A}}_{\text{OE}}$	PB3
Bus controller	$\overline{\text{CAS}}$ output* ¹ * ²	1	—	—	—
	$\overline{\text{CS3-A}}$ output* ¹	—	1	—	—
	$\overline{\text{CS7-A}}$ output* ¹	—	—	1	—
I/O port	PB3 output	0	0	0	1
	PB3 input (initial setting)	0	0	0	0

- Notes: 1. Valid in external extended mode (EXPE = 1)
2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

	$\overline{CS2}$ -A output*	0	0	1	—
I/O port	PB2 output	0	0	0	1
	PB2 input (initial setting)	0	0	0	0

- Notes: 1. Valid in external extended mode (EXPE = 1)
2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

(7) $\overline{PB1}/\overline{CS1}/\overline{CS2-B}/\overline{CS5-A}/\overline{CS6-B}/\overline{CS7-B}$

The pin function is switched as shown below according to the combination of operating EXPE bit, port function control register (PFCR), and the PB1DDR bit settings.

Module Name	Pin Function	Setting				
		I/O Port				
		$\overline{CS1_OE}$	$\overline{CS2B_OE}$	$\overline{CS5A_OE}$	$\overline{CS6B_OE}$	$\overline{CS7B_OE}$
Bus controller	$\overline{CS1}$ output*	1	—	—	—	—
	$\overline{CS2-B}$ output*	—	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)

	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

PCn input 0
(initial setting)

[Legend]

n: 4 to 5

(2) $\overline{\text{PC3}}/\overline{\text{LLCAS}}^{*2}/\overline{\text{DQMLL}}^{*2}$

The pin function is switched as shown below according to the combination of operating EXPE bit, bus controller register, and PC3DDR bit settings.

Module Name	Pin Function	Setting		
		Bus Controller		I/O P
		$\overline{\text{LLCAS}}_{\text{OE}}$	$\overline{\text{DQMLL}}_{\text{OE}}$	PC3DDR
Bus controller	$\overline{\text{LLCAS}}$ output ^{*1*2}	1	—	—
	$\overline{\text{DQMLL}}$ output ^{*1*2}	—	1	—
I/O port	PC3 output*	0	0	1
	PC3 input (initial setting)	0	0	0

Notes: 1. Valid in external extended mode (EXPE = 1)

2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

I/O port	PC2 output	0	0	1
	PC2 input (initial setting)	0	0	0

- Notes: 1. Valid in external extended mode (EXPE = 1)
 2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

(4) $\overline{PC1}/\overline{CS4-C}/\overline{CS5-C}/\overline{CS6-C}/\overline{CS7-C}$

The pin function is switched as shown below according to the combination of operating mode, EXPE bit, port function control register (PFCR), and the PC1DDR bit settings.

Module Name	Pin Function	Setting				P
		I/O Port				
		$\overline{CS4C_OE}$	$\overline{CS5C_OE}$	$\overline{CS6C_OE}$	$\overline{CS7C_OE}$	
Bus controller	$\overline{CS4-C}$ output*	1	—	—	—	—
	$\overline{CS5-C}$ output*	—	1	—	—	—
	$\overline{CS6-C}$ output*	—	—	1	—	—
	$\overline{CS7-C}$ output*	—	—	—	1	—
I/O port	PC1 output	0	0	0	0	1
	PC1 input (initial setting)	0	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

I/O port	PC0 output	0	0	1
	PC0 input (initial setting)	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

EXPE bit, and the PDnDDR bit settings.

Module Name	Pin Function	Setting	
		MCU Operating Mode	I/O Port PDnDDR
Bus controller	Address output	On-chip ROM disabled extended mode	—
		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)

EXPE bit, and the PEnDDR bit settings.

Module Name	Pin Function	Setting	
		MCU Operating Mode	I/O Port PEnDDR
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)

SCI	SCK5 output	1	—	—
Bus controller	A23 output*	0	1	—
I/O port	PF7 output	0	0	1
	PF7 input (initial setting)	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

(2) PF6/A22/RxD5/IrRXD

The pin function is switched as shown below according to the combination of operating mode, EXPE bit, port function control register (PFCR), and PF6DDR bit settings.

Module Name	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)

Bus controller	A21 output*	0	0	1	—
I/O port	PF5 output	0	0	0	1
	PF5 input (initial setting)	0	0	0	0

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.

MCU Operating Mode	Module Name	Pin Function	Setting	
			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 mode, EXPE bit, port function control register (PFCR), and the PF2DDR bit settings.

MCU Operating Mode	Module Name	Pin Function	Setting	
			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.

MCU Operating Mode	Module Name	Pin Function	Setting	
			A16_OE	I/O Port 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)

TPU register, port function control register (PFCR), and PJ7DDR bit settings.

Module Name	Pin Function	Setting		
		PPG	TPU	I/O Port
		PO23_OE	TIOCB8_OE	PD7DDR
PPG	PO23 output*	1	—	—
TPU	TIOCB8 output*	0	1	—
I/O port	PJ7 output*	0	0	1
	PJ7 input*	0	0	0

Note: * Valid when PCJKE = 1

(2) PJ6/TIOCA8/PO22

The pin function is switched as shown below according to the combination of the PPG register, TPU register, port function control register (PFCR), and the PJ6DDR bit settings.

Module Name	Pin Function	Setting		
		PPG	TPU	I/O Port
		PO22_OE	TIOCA8_OE	PJ6DDR
PPG	PO22 output*	1	—	—
TPU	TIOCA8 output*	0	1	—
I/O port	PJ6 output*	0	0	1
	PJ6 input*	0	0	0

Note: * Valid when PCJKE = 1

I/O port	PJ5 output*	0	0	1
	PJ5 input*	0	0	0

Note: * Valid when PCJKE = 1

(4) PJ4/TIOCA7/PO20

The pin function is switched as shown below according to the combination of the PPG register, TPU register, port function control register (PFCR), and the PJ4DDR bit settings.

Module Name	Pin Function	Setting		
		PPG	TPU	I/O Port
		PO20_OE	TIOCA7_OE	PJ4DDR
PPG	PO20 output*	1	—	—
TPU	TIOCA7 output*	0	1	—
I/O port	PJ4 output*	0	0	1
	PJ4 input*	0	0	0

Note: * Valid when PCJKE = 1

I/O port	PJ3 output*	0	0	1
	PJ3 input*	0	0	0

Note: * Valid when PCJKE = 1

(6) PJ2/PO18/TIOCC6/TCLKE

The pin function is switched as shown below according to the combination of the PPG register, TPU register, port function control register (PFCR), and the PJ2DDR bit settings.

Module Name	Pin Function	Setting		
		PPG	TPU	I/O Port
		PO18_OE	TIOCC6_OE	PJ2DDR
PPG	PO18 output*	1	—	—
TPU	TIOCC6 output*	0	1	—
I/O port	PJ2 output*	0	0	1
	PJ2 input*	0	0	0

Note: * Valid when PCJKE = 1

I/O port	PJ1 output*	0	0	1
	PJ1 input*	0	0	0

Note: * Valid when PCJKE = 1

(8) PJ0/PO16/TIOCA6

The pin function is switched as shown below according to the combination of the PPG register, TPU register, port function control register (PFCR), and the PJ0DDR bit settings.

Module Name	Pin Function	Setting		
		PPG	TPU	I/O Port
		PO16_OE	TIOCA6_OE	PJ0DDR
PPG	PO16 output*	1	—	—
TPU	TIOCA6 output*	0	1	—
I/O port	PJ0 output*	0	0	1
	PJ0 input*	0	0	0

Note: * Valid when PCJKE = 1

TPU register, port function control register (PFCR), and the PK7DDR bit settings.

Module Name	Pin Function	Setting		
		PPG	TPU	I/O Port
		PO31_OE	TIOCB11_OE	PK7DDR
PPG	PO31 output*	1	—	—
TPU	TIOCB11 output*	0	1	—
I/O port	PK7 output*	0	0	1
	PK7 input*	0	0	0

Note: * Valid when PCJKE = 1

(2) PK6/PO30/TIOCA11

The pin function is switched as shown below according to the combination of the PPG register, TPU register, port function control register (PFCR), and the PK6DDR bit settings.

Module Name	Pin Function	Setting		
		PPG	TPU	I/O Port
		PO30_OE	TIOCA11_OE	PK6DDR
PPG	PO30 output*	1	—	—
TPU	TIOCA11 output*	0	1	—
I/O port	PK6 output*	0	0	1
	PK6 input*	0	0	0

Note: * Valid when PCJKE = 1

I/O port	PK5 output*	0	0	1
	PK5 input*	0	0	0

Note: * Valid when PCJKE = 1

(4) PK4/PO28/TIOCA10

The pin function is switched as shown below according to the combination of the PPG register, TPU register, port function control register (PFCR), and the PK4DDR bit settings.

Module Name	Pin Function	Setting		
		PPG	TPU	I/O Port
		PO28_OE	TIOCA10_OE	PK4DDR
PPG	PO28 output*	1	—	—
TPU	TIOCA10 output*	0	1	—
I/O port	PK4 output*	0	0	1
	PK4 input*	0	0	0

Note: * Valid when PCJKE = 1

I/O port	PK3 output*	0	0	1
	PK3 input*	0	0	0

Note: * Valid when PCJKE = 1

(6) PK2/PO26/TIOCC9

The pin function is switched as shown below according to the combination of the PPG register, TPU register, port function control register (PFCR), and the PK2DDR bit settings.

Module Name	Pin Function	Setting		
		PPG	TPU	I/O Port
		PO26_OE	TIOCC9_OE	PK2DDR
PPG	PO26 output*	1	—	—
TPU	TIOCC9 output*	0	1	—
I/O port	PK2 output*	0	0	1
	PK2 input*	0	0	0

Note: * Valid when PCJKE = 1

I/O port	PK1 output*	0	0	1
	PK1 input*	0	0	0

Note: * Valid when PCJKE = 1

(8) PK0/PO24/TIOCA9

The pin function is switched as shown below according to the combination of the PPG register, TPU register, port function control register (PFCR), and the PK0DDR bit settings.

Module Name	Pin Function	Setting		
		PPG	TPU	I/O Port
		PO24_OE	TIOCA9_OE	PK0DDR
PPG	PO24 output*	1	—	—
TPU	TIOCA9 output*	0	1	—
I/O port	PK0 output*	0	0	1
	PK0 input*	0	0	0

Note: * Valid when PCJKE = 1

I/O port	PM4 output	1
	PM4 input (initial setting)	0

(2) PM3

The pin function is switched as shown below according to the PM3DDR bit 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 PM2DDR bit setting.

		Setting
		I/O Port
Module Name	Pin Function	PM2DDR
I/O port	PM2 output	1
	PM2 input (initial setting)	0

(5) PM0

The pin function is switched as shown below according to the PM0DDR bit setting.

Module Name	Pin Function	Setting	
		I/O Port	PM0DDR
I/O port	PM0 output		1
	PM0 input (initial setting)		0

13.2.17 Port N

(1) PN3/SCL3

The pin function is switched as shown below according to the combination of the IIC2 register setting and the PN3DDR bit setting.

Module Name	Pin Function	Setting	
		IIC2 SCL3_OE	I/O Port PN3DDR
IIC2	SCL3 I/O	1	—
I/O port	PN3 output (open-drain output)	0	1
	PN3 input (initial setting)	0	0

PN2 input (initial setting)	0	0
--------------------------------	---	---

(3) PN1/SCL2

The pin function is switched as shown below according to the combination of the IIC2 register setting and the PN1DDR bit setting.

Module Name	Pin Function	Setting	
		IIC2	I/O Port
		SCL2_OE	PN1DDR
IIC2	SCL2 I/O	1	—
I/O port	PN1 output (open-drain output)	0	1
	PN1 input (initial setting)	0	0

PN0 input
(initial setting)

0

0

SCK3_OE SCK3

When SCMR.SMIF = 1:
 SCR.TE = 1 or SCR.RE = 1 wh
 SMR.GM = 0, SCR.CKE[1,0] =
 while SMR.GM = 1
 When SCMR.SMIF = 0:
 SCR.TE = 1 or SCR.RE = 1 wh
 SMR.C/A = 0, SCR.CKE[1,0] =
 while SMR.C/A = 1, SCR.CKE

	SDA0_OE	SDA0		ICCRA.ICE = 1
5	$\overline{\text{ETEND1A_OE}}^*$	$\overline{\text{ETEND1}}^*$	PFCR8.EDMAS1[A,B] = 00	SYSCR.EXPE=1, EDMDR_1.E 1
	$\overline{\text{TEND1A_OE}}$	$\overline{\text{TEND1}}$	PFCR7.DMAS1[A,B] = 00	DMDR_1.TENDE = 1
	SCL1_OE	SCL1		ICCRA.ICE = 1
4	TxD3_OE	TxD3		SCR.TE = 1
	SDA1_OE	SDA1		ICCRA.ICE = 1
3	$\overline{\text{EDRAK0A_OE}}^*$	$\overline{\text{EDRAK0}}^*$	PFCR8.EDMAS0[A,B] = 00	SYSCR.EXPE=1, EDMDR_0.E 1
2	$\overline{\text{EDACK0A_OE}}^*$	$\overline{\text{EDACK0}}^*$	PFCR8.EDMAS0[A,B] = 00	SYSCR.EXPE=1, EDACR_0.A EDMDR_0.EDACKE=1
	$\overline{\text{DACK0A_OE}}$	$\overline{\text{DACK0}}$	PFCR7.DMAS0[A,B] = 00	DMAC_0.DACR.AMS = 1, DMDR_0.DACKE = 1

	TEND0A_OE	TEND0	PFCR7.DMAS0[A,B] = 00	DMDR_0.TENDE = 1
	0	TxD2_OE	TxD2	SCR.TE = 1
P2	7	TIOCB5_OE	TIOCB5	TPU.TIOR_5.IOB3 = 0, TPU.TIOR_5.IOB[1,0] = 01/10
		PO7_OE	PO7	NDERL.NDER7 = 1
	6	TIOCA5_OE	TIOCA5	TPU.TIOR_5.IOA3 = 0, TPU.TIOR_5.IOA [1,0] = 01/10
		TMO1_OE	TMO1	TMR.TCSR_1,.OS3,2 = 01/10 TMR.TCSR_1, OS[1,0] = 01/10
		TxD1_OE	TxD1	SCR.TE = 1
		PO6_OE	PO6	NDERL.NDER6 = 1
	5	TIOCA4_OE	TIOCA4	TPU.TIOR_4.IOA3 = 0, TPU.TIOR_4.IOA[1,0] = 01/10
		PO5_OE	PO5	NDERL.NDER5 = 1
	4	TIOCB4_OE	TIOCB4	TPU.TIOR_4.IOB3 = 0, TPU.TIOR_4.IOB[1,0] = 01/10
		SCK1_OE	SCK1	When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 v SMR.GM = 0, SCR.CKE [1, 0] while SMR.GM = 1 When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 v SMR.C/A = 0, SCR.CKE [1, 0] while SMR.C/A = 1, SCR.CKE
		PO4_OE	PO4	NDERL.NDER4 = 1

		TxD0_OE	TxD0	SCR.TE = 1
		PO2_OE	PO2	NDERL.NDER2 = 1
1		TIOCA3_OE	TIOCA3	TPU.TIORH_3.IOA3 = 0, TPU.TIORH_3.IOA[1,0] = 01/10
		PO1_OE	PO1	NDERL.NDER1 = 1
0		TIOCB3_OE	TIOCB3	TPU.TIORH_3.IOB3 = 0, TPU.TIORH_3.IOB[1,0] = 01/10
		SCK0_OE	SCK0	When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 wh SMR.GM = 0, SCR.CKE [1, 0] while SMR.GM = 1 When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 wh SMR.C/A = 0, SCR.CKE [1, 0] while SMR.C/A = 1, SCR.CKE
		PO0_OE	PO0	NDERL.NDER0 = 1
P3	7	$\overline{\text{EDRAK3_OE}}^*$	$\overline{\text{EDRAK3}}^*$ PFCR8.EDMAS3[A,B]= 00	SYSCR.EXPE=1, EDMDR_3,
		TIOCB2_OE	TIOCB2	TPU.TIOR_2.IOB3 = 0, TPU.TIOR_2.IOB[1,0] = 01/10
		PO15_OE	PO15	NDERH.NDER15 = 1
6		$\overline{\text{EDRAK2_OE}}^*$	$\overline{\text{EDRAK2}}^*$ PFCR8.EDMAS2[A,B]= 00	SYSCR.EXPE=1, EDMDR_2,
		TIOCA2_OE	TIOCA2	TPU.TIOR_2.IOA3 = 0, TPU.TIOR_2.IOA[1,0] = 01/10
		PO14_OE	PO14	NDERH.NDER14 = 1

	$\overline{\text{TEND1B_OE}}$	$\overline{\text{TEND1}}$	PFCR7.DMAS1[A,B] = 01	DMDR_1.TENDE = 1
	TIOCA1_OE	TIOCA1		TPU.TIOR_1.IOA3 = 0, TPU.TIOR_1.IOA[1,0] = 01/10
	PO12_OE	PO12		NDERH.NDER12 = 1
3	TIOCD0_OE	TIOCD0		TPU.TMDR.BFB = 0, TPU.TIORL_0.IOD3 = 0, TPU.TIORL_0.IOD[1,0] = 01/10
	PO11_OE	PO11		NDERH.NDER11 = 1
2	$\overline{\text{EDACK2_OE}}$ *	$\overline{\text{EDACK2}}$ *	PFCR8.EDMAS2[A,B] = 00	SYSCR.EXPE=1, EDACR_2. EDMDR_2.EDACKE=1
	$\overline{\text{DACK0B_OE}}$	$\overline{\text{DACK0}}$	PFCR7.DMAS0[A,B] = 01	DMAC.DACR_0.AMS = 1, DMDR_0.DACKE = 1
	TIOCC0_OE	TIOCC0		TPU.TMDR.BFA = 0, TPU.TIORL_0.IOC3 = 0, TPU.TIORL_0.IOD[1,0] = 01/10
	PO10_OE	PO10		NDERH.NDER10 = 1
1	$\overline{\text{ETEND2_OE}}$ *	$\overline{\text{ETEND2}}$ *	PFCR8.EDMAS2[A,B] = 00	SYSCR.EXPE=1, EDMDR_2.
	$\overline{\text{TEND0B_OE}}$	$\overline{\text{TEND0}}$	PFCR7.DMAS0[A,B] = 01	DMDR_0.TENDE = 1
	TIOCB0_OE	TIOCB0		TPU.TIORH_0.IOB3 = 0, TPU.TIORH_0.IOB[1,0] = 01/10
	PO9_OE	PO9		NDERH.NDER9 = 1
0	TIOCA0_OE	TIOCA0		TPU.TIORH_0.IOA3 = 0, TPU.TIOH_0.IOA[1,0] = 01/10
	PO8_OE	PO8		NDERH.NDER8 = 1

	SCK6_OE	SCK6		When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 wh SMR.GM = 0, SCR.CKE[1,0] = while SMR.GM = 1 When SCMR.SMIF = 0: SCR.T SCR.RE = 1 while SMR.C/A = SCR.CKE[1,0] = 01 or while SF 1, SCR.CKE1 = 0
4	$\overline{\text{ETEND1B_OE}}^*$	$\overline{\text{ETEND1}}^*$	PFCR8.EDMAS1[A,B] = 01	SYSCR.EXPE=1, EDMDR_1.E
	$\overline{\text{TEND3_OE}}$	$\overline{\text{TEND3}}$	PFCR7.DMAS3[A,B] = 01	DMDR_3.TENDE = 1
3	TxD6_OE	TxD6		SCR.TE = 1
2	$\overline{\text{EDACK0B_OE}}^*$	$\overline{\text{EDACK0}}^*$	PFCR8.EDMAS0[A,B] = 01	SYSCR.EXPE=1, EDACR_0.A EDMDR_0.EDACKE=1
	$\overline{\text{DACK2_OE}}$	$\overline{\text{DACK2}}$	PFCR7.DMAS2[A,B] = 01	DMAC.DACR_2.AMS = 1, DMDR_2.DACKE = 1
	TMO2_OE	TMO2		TMR.TCSR_2.OS[3,2] = 01/10 TMR.TCSR_2.OS[1,0] = 01/10
	SCK4_OE	SCK4		When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 wh SMR.GM = 0, SCR.CKE [1,0] = while SMR.GM = 1 When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 wh SMR.C/A = 0, SCR.CKE [1,0] = while SMR.C/A = 1, SCR.CKE
1	$\overline{\text{ETEND0B_OE}}^*$	$\overline{\text{ETEND0}}^*$	PFCR8.EDMAS0[A,B] = 01	SYSCR.EXPE=1, EDMDR_0.E
	$\overline{\text{TEND2_OE}}$	$\overline{\text{TEND2}}$	PFCR7.DMAS2[A,B] = 01	DMDR_2.TENDE = 1
0	TxD4_OE	TxD4		SCR.TE = 1

4	$\overline{\text{LUB_OE}}$	$\overline{\text{LUB}}$		SYSCR.EXPE = 1, PFCR6.L or SRAMCR.BCSELn = 1
	$\overline{\text{LHWR_OE}}$	$\overline{\text{LHWR}}$		SYSCR.EXPE = 1, PFCR6.L
3	$\overline{\text{LLB_OE}}$	$\overline{\text{LLB}}$		SYSCR.EXPE = 1, SRAMCR.1
	$\overline{\text{LLWR_OE}}$	$\overline{\text{LLWR}}$		SYSCR.EXPE = 1
2	—	—	—	—
1	$\overline{\text{BACK_OE}}$	$\overline{\text{BACK}}$		SYSCR.EXPE = 1, BCR1.BR
	$(\text{RD}/\overline{\text{WR}})\text{-A_OE}$	$\text{RD}/\overline{\text{WR}}$	PFCR2.RDWRS = 0	SYSCR.EXPE = 1, PFCR2.R or SRAMCR.BCSELn = 1
0	$\overline{\text{BS-A_OE}}$	$\overline{\text{BS}}$	PFCR2.BSS = 0	SYSCR.EXPE = 1, PFCR2.B
	$\overline{\text{BREQO_OE}}$	$\overline{\text{BREQO}}$		SYSCR.EXPE = 1, BCR1.BR BCR1.BREQOE = 1
PB 7	$\overline{\text{SD}\phi\text{-OE}}^*$	$\overline{\text{SD}\phi}^*$		MD3 = 1
	$\overline{\text{CS7D_OE}}$	$\overline{\text{CS7}}$	PFCR1.CS7S[A,B] = 11	SYSCR.EXPE = 1, PFCR.CS
6	$(\text{RD}/\overline{\text{WR}})\text{-B_OE}$	$\text{RD}/\overline{\text{WR}}$	PFCR2.RDWRS = 1	SYSCR.EXPE = 1, PFCR2.R or SRAMCR.BCSELn = 1
	$\overline{\text{CS6D_OE}}$	$\overline{\text{CS6}}$	PFCR1.CS6S[A,B] = 11	SYSCR.EXPE = 1, PFCR0.C
5	$\overline{\text{CKE_OE}}^*$	$\overline{\text{CKE}}^*$		SYSCR.EXPE = 1, DRAMCR.DRAMCR.ETYPE=1, DRAMC
	$\overline{\text{OE_OE}}^*$	$\overline{\text{OE}}^*$		SYSCR.EXPE = 1, DRAMCR.DRAMCR.DTYPE=0, DRAMC
	$\overline{\text{CS5D_OE}}$	$\overline{\text{CS5}}$	PFCR1.CS5S[A,B] = 11	SYSCR.EXPE = 1, PFCR0.C
4	$\overline{\text{WE_OE}}^*$	$\overline{\text{WE}}^*$		SYSCR.EXPE=1, DRAMCR.D
	$\overline{\text{CS4B_OE}}$	$\overline{\text{CS4}}$	PFCR1.CS4S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.C

1	$\overline{\text{CS1_OE}}$	$\overline{\text{CS1}}$		SYSCR.EXPE = 1, PFCR0.CS
	$\overline{\text{CS2B_OE}}$	$\overline{\text{CS2}}$	PFCR2.CS2S = 1	SYSCR.EXPE = 1, PFCR0.CS
	$\overline{\text{CS5A_OE}}$	$\overline{\text{CS5}}$	PFCR1.CS5S[A,B] = 00	SYSCR.EXPE = 1, PFCR0.CS
	$\overline{\text{CS6B_OE}}$	$\overline{\text{CS6}}$	PFCR1.CS6S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS
	$\overline{\text{CS7B_OE}}$	$\overline{\text{CS7}}$	PFCR1.CS7S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS
0	$\overline{\text{CS0_OE}}$	$\overline{\text{CS0}}$		SYSCR.EXPE = 1, PFCR0.CS
	$\overline{\text{CS4A_OE}}$	$\overline{\text{CS4}}$	PFCR1.CS4S[A,B] = 00	SYSCR.EXPE = 1, PFCR0.CS
	$\overline{\text{CS5B_OE}}$	$\overline{\text{CS5}}$	PFCR1.CS5S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS
PC 3	$\overline{\text{LLCAS_OE}}^*$	$\overline{\text{LLCAS}}$		SYSCR.EXPE = 1, DRAMCR.D DRAMCR.DTYPE = 0
	$\overline{\text{DQMLL_OE}}^*$	$\overline{\text{DQMLL}}$		SYSCR.EXPE = 1, DRAMCR.D DRAMCR.DTYPE = 1
2	$\overline{\text{LUCAS_OE}}^*$	$\overline{\text{LUCAS}}$		SYSCR.EXPE=1,ABWCR.[AB ABWL2]=x0/01,DRAMCR.DRA DRAMCR.DTYPE=0
	$\overline{\text{DQMLU_OE}}^*$	$\overline{\text{DQMLU}}$		SYSCR.EXPE=1,ABWCR.[AB ABWL2]=x0/01,DRAMCR.DRA DRAMCR.DTYPE=1
1	$\overline{\text{CS4C_OE}}$	$\overline{\text{CS4}}$	PFCR1.CS4S[A,B] = 10	SYSCR.EXPE = 1, PFCR0.CS
	$\overline{\text{CS5C_OE}}$	$\overline{\text{CS5}}$	PFCR1.CS5S[A,B] = 10	SYSCR.EXPE = 1, PFCR0.CS
	$\overline{\text{CS6C_OE}}$	$\overline{\text{CS6}}$	PFCR1.CS6S[A,B] = 10	SYSCR.EXPE = 1, PFCR0.CS
	$\overline{\text{CS7C_OE}}$	$\overline{\text{CS7}}$	PFCR1.CS7S[A,B] = 10	SYSCR.EXPE = 1, PFCR0.CS
0	$\overline{\text{CS3B_OE}}$	$\overline{\text{CS3}}$	PFCR2.CS3S = 1	SYSCR.EXPE = 1, PFCR0.CS

	0	A0_OE	A0	SYSCR.EXPE = 1, PDDDR.P
PE	7	A15_OE	A15	SYSCR.EXPE = 1, PEDDR.P
	6	A14_OE	A14	SYSCR.EXPE = 1, PEDDR.P
	5	A13_OE	A13	SYSCR.EXPE = 1, PEDDR.P
	4	A12_OE	A12	SYSCR.EXPE = 1, PEDDR.P
	3	A11_OE	A11	SYSCR.EXPE = 1, PEDDR.P
	2	A10_OE	A10	SYSCR.EXPE = 1, PEDDR.P
	1	A9_OE	A9	SYSCR.EXPE = 1, PEDDR.P
	0	A8_OE	A8	SYSCR.EXPE = 1, PEDDR.P
PF	7	A23A_OE	A23	SYSCR.EXPE = 1, PFCR4.A
		SCK5_OE	SCK5	When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 v SMR.GM = 0, SCR.CKE[1,0] SMR.GM = 1 When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 v SMR.C/A = 0, SCR.CKE[1,0] while SMR.C/A = 1, SCR.CKE
	6	A22A_OE	A22	SYSCR.EXPE = 1, PFCR4.A
	5	A21A_OE	A21	SYSCR.EXPE = 1, PFCR4.A
		TxD5_OE	TxD5	SCR.TE = 1
		IrTxD_OE	IrTxD	SCR.TE = 1, IrCR.IrE = 1
	4	A20_OE	A20	SYSCR.EXPE = 1, PFCR4.A
	3	A19_OE	A19	SYSCR.EXPE = 1, PFCR4.A
	2	A18_OE	A18	SYSCR.EXPE = 1, PFCR4.A
	1	A17_OE	A17	SYSCR.EXPE = 1, PFCR4.A

	1	D1_E	D1	SYSCR.EXPE = 1
	0	D0_E	D0	SYSCR.EXPE = 1
PI	7	D15_E	D15	SYSCR.EXPE = 1, ABWCR.AB01
	6	D14_E	D14	SYSCR.EXPE = 1, ABWCR.AB01
	5	D13_E	D13	SYSCR.EXPE = 1, ABWCR.AB01
	4	D12_E	D12	SYSCR.EXPE = 1, ABWCR.AB01
	3	D11_E	D11	SYSCR.EXPE = 1, ABWCR.AB01
	2	D10_E	D10	SYSCR.EXPE = 1, ABWCR.AB01
	1	D9_E	D9	SYSCR.EXPE = 1, ABWCR.AB01
	0	D8_E	D8	SYSCR.EXPE = 1, ABWCR.AB01
PJ	7	TIOCB8_OE	TIOCB8	TPU.TIOR_8.IOB3 = 0, TPU.TIOR_8.IOB[1,0] = 01/10/
		PO23_OE	PO23	NDERL_1.NDER23 = 1
	6	TIOCA8_OE	TIOCA8	TPU.TIOR_8.IOA3 = 0, TPU.TIOR_8.IOA[1,0] = 01/10/
		PO22_OE	PO22	NDERL_1.NDER22 = 1
	5	TIOCB7_OE	TIOCB7	TPU.TIOR_7.IOB3 = 0, TPU.TIOR_7.IOB[1,0] = 01/10/
		PO21_OE	PO21	NDERL_1.NDER21 = 1

			TPU.TIORL_6.IOC3 = 0, TPU.TIORL_6.IOC[1,0] = 01/
		PO18_OE PO18	NDERL_1.NDER18 = 1
1		TIOCB6_OE TIOCB6	TPU.TIORH_6.IOB3 = 0, TPU.TIORH_6.IOB[1,0] = 01/
		PO17_OE PO17	NDERL_1.NDER17 = 1
0		TIOCA6_OE TIOCA6	TPU.TIORH_6.IOA3 = 0, TPU.TIORH_6.IOA[1,0] = 01/
		PO16_OE PO16	NDERL_1.NDER16 = 1
PK 7		TIOCB11_OE TIOCB11	TPU.TIOR_11.IOB3 = 0, TPU.TIOR_11.IOB[1,0] = 01/
		PO31_OE PO31	NDERH_1.NDER31 = 1
6		TIOCA11_OE TIOCA11	TPU.TIOR_11.IOA3 = 0, TPU.TIOR_11.IOA[1,0] = 01/
		PO30_OE PO30	NDERH_1.NDER30 = 1
5		TIOCB10_OE TIOCB10	TPU.TIOR_10.IOB3 = 0, TPU.TIOR_10.IOB[1,0] = 01/
		PO29_OE PO29	NDERH_1.NDER29 = 1
4		TIOCA10_OE TIOCA10	TPU.TIOR_10.IOA3 = 0, TPU.TIOR_10.IOA[1,0] = 01/
		PO28_OE PO28	NDERH_1.NDER28 = 1
3		TIOCD9_OE TIOCD9	TPU.TMDR_9.BFB = 0, TPU.TIORL_9.IOD3 = 0, TPU.TIORL_9.IOD[1,0] = 01/
		PO27_OE PO27	NDERH_1.NDER27 = 1

		PO24_OE	PO24	NDERH_1.NDER24 = 1
PN	3	SCL3_OE	SCL2	ICCRA.ICE = 1
	2	SDA3_OE	SDA3	ICCRA.ICE = 1
	1	SCL2_OE	SCL2	ICCRA.ICE = 1
	0	SDA2_OE	SDA2	ICCRA.ICE = 1

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

- Port function control register 6 (PFCR6)
- Port function control register 7 (PFCR7)
- Port function control register 8 (PFCR8)*
- Port function control register 9 (PFCR9)
- Port function control register A (PFCRA)
- Port function control register B (PFCRB)
- Port function control register C (PFCRC)
- Port function control register D (PFCRD)

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

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 \overline{CS} output.
5	CS5E	0	R/W	
4	CS4E	0	R/W	0: Pin functions as I/O port 1: Pin functions as \overline{CSn} output pin (n = 7 to 0)
3	CS3E	0	R/W	
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.

13.3.2 Port Function Control Register 1 (PFCR1)

PFCR1 selects the \overline{CS} output pins.

Bit	7	6	5	4	3	2	1
Bit Name	CS7SA	CS7SB	CS6SA	CS6SB	CS5SA	CS5SB	CS4SA
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

4	CS6E*	0	R/W	Selects the output pin for CS6 when CS6 output is enabled (CS6E = 1) 00: Specifies pin PB2 as CS6-A output 01: Specifies pin PB1 as CS6-B output 10: Specifies pin PC1 as CS6-C output 11: Specifies pin PB6 as CS6-D output
3	CS5SA*	0	R/W	CS5 Output Pin Select
2	CS5SB*	0	R/W	Selects the output pin for CS5 when CS5 output is enabled (CS5E = 1) 00: Specifies pin PB1 as CS5-A output 01: Specifies pin PB0 as CS5-B output 10: Specifies pin PC1 as CS5-C output 11: Specifies pin PB5 as CS5-D output
1	CS4SA*	0	R/W	CS4 Output Pin Select
0	CS4SB*	0	R/W	Selects the output pin for CS4 when CS4 output is enabled (CS4E = 1) 00: Specifies pin PB0 as CS4-A output 01: Specifies pin PB4 as CS4-B output 10: Specifies pin PC1 as CS4-C output 11: Setting prohibited

Note: * If multiple CS outputs are specified to a single pin according to the CSn output select bits (n = 4 to 7), multiple CS signals are output from the pin. For details see section 9.5.3, Chip Select Signals.

Bit	Bit Name	Initial Value	R/W	Description
7	CS3S* ¹	0	R/W	<p>$\overline{CS3}$ Output Pin Select</p> <p>Selects the output pin for $\overline{CS3}$ when $\overline{CS3}$ output enabled (CS3E = 1)</p> <p>0: Specifies pin PB3 as $\overline{CS3}$-A output pin</p> <p>1: Specifies pin PB0 as $\overline{CS3}$-B output pin</p>
6	CS2S* ¹	0	R/W	<p>$\overline{CS2}$ Output Pin Select</p> <p>Selects the output pin for $\overline{CS2}$ when $\overline{CS2}$ output enabled (CS2E = 1)</p> <p>0: Specifies pin PB2 as $\overline{CS2}$-A output pin</p> <p>1: Specifies pin PB1 as $\overline{CS2}$-B output pin</p>
5	BSS	0	R/W	<p>\overline{BS} Output Pin Select</p> <p>Selects the \overline{BS} output pin</p> <p>0: Specifies pin PA0 as \overline{BS}-A output pin</p> <p>1: Specifies pin PA6 as \overline{BS}-B output pin</p>
4	BSE	0	R/W	<p>\overline{BS} Output Enable</p> <p>Enables/disables the \overline{BS} output</p> <p>0: Disables the \overline{BS} output</p> <p>1: Enables the \overline{BS} output</p>
3	RDWRS* ²	0	R/W	<p>RD/\overline{WR} Output Pin Select</p> <p>Selects the output pin for RD/\overline{WR}</p> <p>0: Specifies pin PA1 as RD/\overline{WR}-A output pin</p> <p>1: Specifies pin PB6 as RD/\overline{WR}-B output pin</p>

0	WAITS	0	R/W	WAIT Input Pin Select
Selects the input pin for the wait request signal when accessing external spaces.				
0: Specifies pin PA2 as $\overline{\text{WAIT}}\text{-A}$ input pin				
1: Specifies pin PC0 as $\overline{\text{WAIT}}\text{-B}$ input pin				

- Notes:
1. If multiple $\overline{\text{CS}}$ outputs are specified to a single pin according to the $\overline{\text{CS}}_n$ output select bit (n = 2, 3), multiple $\overline{\text{CS}}$ signals are output from the pin. For details, see 9.5.3, Chip Select Signals.
 2. If an area is specified as a byte control SDRAM space, the pin functions as $\overline{\text{CS}}$ output regardless of the RDWRE bit value.

13.3.4 Port Function Control Register 4 (PFCR4)

PFCR4 enables or disables the address output.

Bit	7	6	5	4	3	2	1
Bit Name	A23E	A22E	A21E	A20E	A19E	A18E	A17E
Initial Value	0	0	0	0/1*	0/1*	0/1*	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
7	A23E	0	R/W	Address A23 Enable Enables/disables the address output (A23) 0: Disables the A23 output 1: Enables the A23 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
2	A18E	0/1*	R/W	Address A18 Enable Enables/disables the address output (A18) 0: Disables the A18 output 1: Enables the A18 output
1	A17E	0/1*	R/W	Address A17 Enable Enables/disables the address output (A17) 0: Disables the A17 output 1: Enables the A17 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

Notes: * The initial value changes depending on the operating mode.
The initial value is 1 when the on-chip ROM is disabled, and 0 when the on-chip ROM is enabled.

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	$\overline{\text{LHWR}}$ Output Enable Enables/disables $\overline{\text{LHWR}}$ output (valid in exte extended mode). 0: Specifies pin PA4 as I/O port 1: Specifies pin PA4 as $\overline{\text{LHWR}}$ 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 modified
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	—	All 0	R/W	Reserved These bits are always read as 0. The write va always be 0.

13.3.6 Port Function Control Register 7 (PFCR7)

PFCR7 selects the DMAC I/O pins ($\overline{\text{DREQ}}$, $\overline{\text{DACK}}$, and $\overline{\text{TEND}}$).

Bit	7	6	5	4	3	2	1
Bit Name	DMAS3A	DMAS3B	DMAS2A	DMAS2B	DMAS1A	DMAS1B	DMAS0A
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	DMAS3A	0	R/W	DMAC control pin select
6	DMAS3B	0	R/W	Selects the I/O port to control DMAC_3. 00: Setting invalid 01: Specifies pins P63 to P65 as DMAC control pins 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 invalid 01: Specifies pins P60 to P62 as DMAC control pins 10: Setting prohibited 11: Setting prohibited

00: Specifies pins P10 to P12 as DMAC control pins
 01: Specifies pins P30 to P32 as DMAC control pins
 10: Setting prohibited
 11: Setting prohibited

13.3.7 Port Function Control Register 8 (PFCR8)*

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.
 PFCR8 selects the EXDMAC I/O pins ($\overline{\text{EDREQ}}$, $\overline{\text{EDACK}}$, $\overline{\text{ETEND}}$, and $\overline{\text{EDRAK}}$).

Bit	7	6	5	4	3	2	1
Bit name	EDMAS3A	EDMAS3B	EDMAS2A	EDMAS2B	EDMAS1A	EDMAS1B	EDMAS0A
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	EDMAS3A	0	R/W	EXDMAC control pin select
6	EDMAS3B	0	R/W	Selects the I/O port to control EXDMAC_3. 00: Specifies pins P33 to P35, and P37 as EXDMAC control pins 01: Setting prohibited 10: Setting prohibited 11: Setting prohibited

2	EDMAS1D	0	R/W	Selects the I/O port to control EXDMAC_1. 00: Specifies pins P14 to P17 as EXDMAC control pins 01: Specifies pins P63 to P65, and P67 as EXDMAC control pins 10: Setting prohibited 11: Setting prohibited
1	EDMAS0A	0	R/W	EXDMAC control pin select
0	EDMAS0B	0	R/W	Selects the I/O port to control EXDMAC_0. 00: Specifies pins P10 to P13 as EXDMAC control pins 01: Specifies pins P60 to P62, and P66 as EXDMAC control pins 10: Setting prohibited 11: Setting prohibited

13.3.8 Port Function Control Register 9 (PFCR9)

PFCR9 selects the multiple functions for the TPU I/O pins.

Bit	7	6	5	4	3	2	1
Bit Name	TPUMS5	TPUMS4	TPUMS3A	TPUMS3B	TPUMS2	TPUMS1	TPUMS0A
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

				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
3	TPUMS2	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA2 function 0: Specifies P36 as output compare output and capture 1: Specifies P37 as input capture input and P3 output compare
2	TPUMS1	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA1 function 0: Specifies P34 as output compare output and capture 1: Specifies P35 as input capture input and P3 output compare

capture
 1: Specifies P33 as input capture input and P32
 output compare

13.3.9 Port Function Control Register A (PFCRA)

PFCRA selects the multiple functions for the TPU (unit 1) I/O pins. Do not access this register because writing to or reading from the bits in this register is not effective when the PCJKR.PFCRD is cleared to 0.

Bit	7	6	5	4	3	2	1	
Bit Name	TPUMS11	TPUMS10	TPUMS9A	TPUMS9B	TPUMS8	TPUMS7	TPUMS6A	TPUMS6B
Initial Value	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

Bit	Bit Name	Initial Value	R/W	Description
7	TPUMS11	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA11 function 0: Specifies PK6 as output compare output and capture 1: Specifies PK7 as input capture input and PK6 output compare

				capture 1: Specifies PK1 as input capture input and PK output compare
4	TPUMS9B	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCC9 function 0: Specifies PK2 as output compare output and capture 1: Specifies PK3 as input capture input and PK output compare
3	TPUMS8	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA8 function 0: Specifies PJ6 as output compare output and capture 1: Specifies PJ7 as input capture input and PJ compare
2	TPUMS7	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA7 function 0: Specifies PJ4 as output compare output and capture 1: Specifies PJ5 as input capture input and PJ compare
1	TPUMS6A	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA6 function 0: Specifies PJ0 as output compare output and capture 1: Specifies PJ1 as input capture input and PJ compare

- H8SX/1648 Group, H8SX/1648A Group
PFCRB selects the input pins for $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ8}}$.
- H8SX/1648L Group
PFCRB selects $\overline{\text{IRQ14}}$ interrupt / LVD interrupt*¹, and the input pins for the $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ13}}$ to $\overline{\text{IRQ8}}$.
- H8SX/1648G Group
PFCRB selects IRQ15, 32KOVI*², and the input pins for $\overline{\text{IRQ13}}$ to $\overline{\text{IRQ8}}$.
- H8SX/1648H Group
PFCRB selects IRQ15, 32KOVI*², $\overline{\text{IRQ14}}$ interrupt / LVD interrupt*¹, and the input pins for $\overline{\text{IRQ13}}$ to $\overline{\text{IRQ8}}$.

Bit	7	6	5	4	3	2	1
Bit Name	ITS15	ITS14	ITS13	ITS12	ITS11	ITS10	ITS9
Initial Value:	0	0	0	0	0	0	0
R/W:	R/W	R/W* ³	R/W	R/W	R/W	R/W	R/W

- Notes: 1. Supported only by the H8SX/1648L Group and the H8SX/1648H Group.
2. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.

- H8SX/1648L Group

Bit	Bit Name	Initial Value	R/W	Description
7	ITS15	0	R/W	<p>$\overline{\text{IRQ15}}$ Pin Select</p> <p>Selects an input pin for $\overline{\text{IRQ15}}$.</p> <p>0: Selects pin P27 as $\overline{\text{IRQ15}}$-A input</p> <p>1: Selects pin P67 as $\overline{\text{IRQ15}}$-B input</p>
6	ITS14	0	R/W	<p>LVD Interrupt / IRQ14 Interrupt Select</p> <p>Selects whether the LVD interrupt or IRQ14 interrupt is to be used.</p> <p>0: Selects pin P26 as $\overline{\text{IRQ14}}$-A</p> <p>1: Selects LVD interrupt</p>

- H8SX/1648G Group

Bit	Bit Name	Initial Value	R/W	Description
7	ITS15	0	R/W	<p>32KOVI / $\overline{\text{IRQ15}}$ Interrupt Select</p> <p>Selects whether 32KOVI interrupt or $\overline{\text{IRQ15}}$ interrupt is to be used.</p> <p>0: Selects the 32KOVI interrupt in the TM32K</p> <p>1: Selects pin P67 as $\overline{\text{IRQ15}}$-B input</p>
6	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value is always be 0.</p>

to be used.

0: Selects pin P26 as $\overline{\text{IRQ14}}$ -A

1: Selects LVD interrupt

- Each Group in Common

Bit	Bit Name	Initial Value	R/W	Description
5	ITS13	0	R/W	$\overline{\text{IRQ13}}$ Pin Select Selects an input pin for $\overline{\text{IRQ13}}$. 0: Selects pin P25 as $\overline{\text{IRQ13}}$ -A input 1: Selects pin P65 as $\overline{\text{IRQ13}}$ -B input
4	ITS12	0	R/W	$\overline{\text{IRQ12}}$ Pin Select Selects an input pin for $\overline{\text{IRQ12}}$. 0: Selects pin P24 as $\overline{\text{IRQ12}}$ -A input 1: Selects pin P64 as $\overline{\text{IRQ12}}$ -B input
3	ITS11	0	R/W	$\overline{\text{IRQ11}}$ Pin Select Selects an input pin for $\overline{\text{IRQ11}}$. 0: Selects pin P23 as $\overline{\text{IRQ11}}$ -A input 1: Selects pin P63 as $\overline{\text{IRQ11}}$ -B input
2	ITS10	0	R/W	$\overline{\text{IRQ10}}$ Pin Select Selects an input pin for $\overline{\text{IRQ10}}$. 0: Selects pin P22 as $\overline{\text{IRQ10}}$ -A input 1: Selects pin P62 as $\overline{\text{IRQ10}}$ -B input

13.3.11 Port Function Control Register C (PFCRC)

PFCRC selects input pins for $\overline{IRQ7}$ to $\overline{IRQ0}$.

Bit	7	6	5	4	3	2	1
Bit Name	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1
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	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

2	ITS2	0	R/W	<p>IRQ2 Pin Select</p> <p>Selects an input pin for $\overline{\text{IRQ2}}$.</p> <p>0: Selects pin P12 as $\overline{\text{IRQ2}}$-A input</p> <p>1: Selects pin P52 as $\overline{\text{IRQ2}}$-B input</p>
1	ITS1	0	R/W	<p>$\overline{\text{IRQ1}}$ Pin Select</p> <p>Selects an input pin for $\overline{\text{IRQ1}}$.</p> <p>0: Selects pin P11 as $\overline{\text{IRQ1}}$-A input</p> <p>1: Selects pin P51 as $\overline{\text{IRQ1}}$-B input</p>
0	ITS0	0	R/W	<p>$\overline{\text{IRQ0}}$ Pin Select</p> <p>Selects an input pin for $\overline{\text{IRQ0}}$.</p> <p>0: Selects pin P10 as $\overline{\text{IRQ0}}$-A input</p> <p>1: Selects pin P50 as $\overline{\text{IRQ0}}$-B input</p>

Bit	Bit Name	Value	R/W	Description
7	PCJKE*	0	R/W	Ports J and K Enable Enables or disables the port J and K pin function. 0: Ports J and K are disabled. 1: Ports J and K are enabled (ports D and E are disabled).
6 to 0	—	0	R/W	Reserved These bits are always read as 0 and cannot be modified. The initial value should not be changed.

Note: * This bit is only effective in single-chip mode. In other modes, do not change the value of this bit.

3. When a pin is used as an output, data to be output from the pin will be latched as the pin level if the input function corresponding to the pin is enabled. To use the pin as an output, disable the input function for the pin by setting ICR.

13.4.2 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 generation delay and 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 settings.
 - 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 enable bit that enables the pin function, first specify the input/output destination by the select bit and then enable the pin function by the enable bit.
4. The value of the PCJKE bit must be set during the initial setting immediately after a power-up. Set the PCJKE bit first and then set other bits in PFCR as required.
5. Do not change the value of the PCJKE bit once it has been set.

14.1 Features

- Maximum 16-pulse input/output
- Selection of eight counter input clocks for each channel
- The following operations can be set for each channel:
 - Waveform output at compare match
 - Input capture function
 - Counter clear operation
 - Synchronous operations:
 - Multiple timer counters (TCNT) can be written to simultaneously
 - Simultaneous clearing by compare match and input capture possible
 - Simultaneous input/output for registers possible by counter synchronous operation
 - Maximum of 15-phase PWM output possible by combination with synchronous operation
- Buffer operation settable for channels 0 and 3
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
- Cascaded operation
- Fast access via internal 16-bit bus
- 26 interrupt sources
- Automatic transfer of register data
- Programmable pulse generator (PPG) output trigger can be generated (unit 0 only)
- Conversion start trigger for the A/D converter can be generated (unit 0 only)
- Module stop state can be set

(TGR)	TGRB_0	TGRB_1	TGRB_2	TGRB_3	TGRB_4	TGRB_5
General registers/ buffer registers	TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	—	—
I/O pins	TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TIOCA5 TIOCB5
Counter clear function	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	O	O	O	O	O
	1 output	O	O	O	O	O
	Toggle output	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	TGR compare match or input capture

Interrupt trigger	TGRB_0 compare match or input capture	TGRB_1 compare match or input capture	TGRB_2 compare match or input capture	TGRB_3 compare match or input capture	TGRB_4 compare match or input capture	TGRB_5 compare match or input capture
Interrupt sources	5 sources Compare match or input capture 0A Compare match or input capture 0B Compare match or input capture 0C Compare match or input capture 0D Overflow	4 sources Compare match or input capture 1A Compare match or input capture 1B Overflow Underflow	4 sources Compare match or input capture 2A Compare match or input capture 2B Overflow Underflow	5 sources Compare match or input capture 3A Compare match or input capture 3B Compare match or input capture 3C Compare match or input capture 3D Overflow	4 sources Compare match or input capture 4A Compare match or input capture 4B Overflow Underflow	4 sources Compare match or input capture 4C Compare match or input capture 4D Overflow Underflow

[Legend]

- : Possible
- : Not possible

(TGR)	TGRB_6	TGRB_7	TGRB_8	TGRB_9	TGRB_10	TGRB_11
General registers/ buffer registers	TGRC_6 TGRD_6	—	—	TGRC_9 TGRD_9	—	—
I/O pins	TIOCA6 TIOCB6 TIOCC6 TIOCD6	TIOCA7 TIOCB7	TIOCA8 TIOCB8	TIOCA9 TIOCB9 TIOCC9 TIOCD9	TIOCA10 TIOCB10	TIOCA11 TIOCB11
Counter clear function	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	O	O	O	O	O
	1 output	O	O	O	O	O
	Toggle output	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	TGR compare match or input capture

Interrupt sources	5 sources	4 sources	4 sources	5 sources	4 sources	4
	Compare match or input capture 6A	Compare match or input capture 7A	Compare match or input capture 8A	Compare match or input capture 9A	Compare match or input capture 10A	C
	Compare match or input capture 6B	Compare match or input capture 7B	Compare match or input capture 8B	Compare match or input capture 9B	Compare match or input capture 10B	C
	Compare match or input capture 6C	Overflow Underflow	Overflow Underflow	Compare match or input capture 9C	Overflow Underflow	C
	Compare match or input capture 6D			Compare match or input capture 9D		U
	Overflow			Overflow		

[Legend]

- O: Possible
- : Not possible

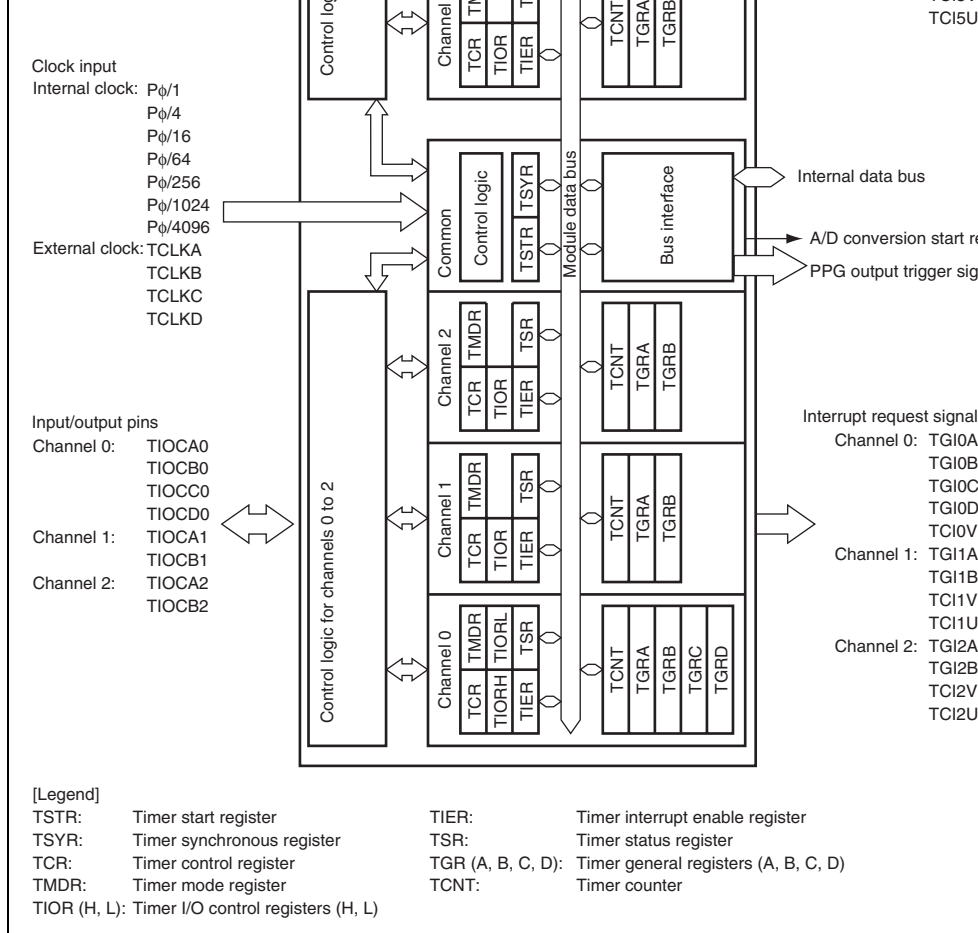


Figure 14.1 Block Diagram of TPU (Unit 0)

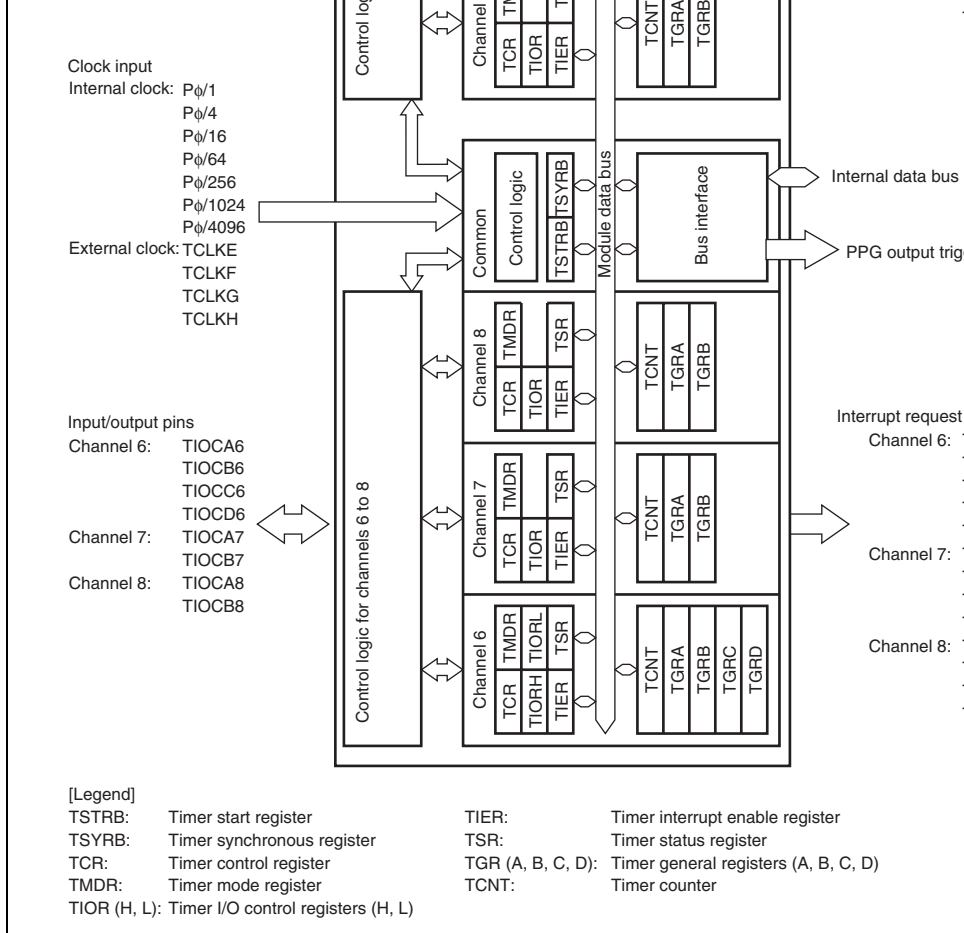


Figure 14.2 Block Diagram of TPU (Unit 1)

	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

6	TIOCA6	I/O	TGRA_6 input capture input/output compare output/PWM
	TIOCB6	I/O	TGRB_6 input capture input/output compare output/PWM
	TIOCC6	I/O	TGRC_6 input capture input/output compare output/PWM
	TIOCD6	I/O	TGRD_6 input capture input/output compare output/PWM
7	TIOCA7	I/O	TGRA_7 input capture input/output compare output/PWM
	TIOCB7	I/O	TGRB_7 input capture input/output compare output/PWM
8	TIOCA8	I/O	TGRA_8 input capture input/output compare output/PWM
	TIOCB8	I/O	TGRB_8 input capture input/output compare output/PWM
9	TIOCA9	I/O	TGRA_9 input capture input/output compare output/PWM
	TIOCB9	I/O	TGRB_9 input capture input/output compare output/PWM
	TIOCC9	I/O	TGRC_9 input capture input/output compare output/PWM
	TIOCD9	I/O	TGRD_9 input capture input/output compare output/PWM
10	TIOCA10	I/O	TGRA_10 input capture input/output compare output/PWM
	TIOCB10	I/O	TGRB_10 input capture input/output compare output/PWM
11	TIOCA11	I/O	TGRA_11 input capture input/output compare output/PWM
	TIOCB11	I/O	TGRB_11 input capture input/output compare output/PWM

- Timer mode register_0 (TMDR_0)
- Timer I/O control register H_0 (TIORH_0)
- Timer I/O control register L_0 (TIORL_0)
- Timer interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)

- 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 3:
 - Timer control register_3 (TCR_3)
 - Timer mode register_3 (TMDR_3)
 - Timer I/O control register H_3 (TIORH_3)
 - Timer I/O control register L_3 (TIORL_3)
 - Timer interrupt enable register_3 (TIER_3)
 - Timer status register_3 (TSR_3)
 - Timer counter_3 (TCNT_3)
 - Timer general register A_3 (TGRA_3)
 - Timer general register B_3 (TGRB_3)
 - Timer general register C_3 (TGRC_3)
 - Timer general register D_3 (TGRD_3)

- 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)

- Common Registers:
 - Timer start register (TSTR)
 - Timer synchronous register (TSYR)

Unit 1:

- Channel 6:
 - Timer control register_6 (TCR_6)
 - Timer mode register_6 (TMDR_6)
 - Timer I/O control register H_6 (TIORH_6)
 - Timer I/O control register L_6 (TIORL_6)
 - Timer interrupt enable register_6 (TIER_6)
 - Timer status register_6 (TSR_6)
 - Timer counter_6 (TCNT_6)
 - Timer general register A_6 (TGRA_6)
 - Timer general register B_6 (TGRB_6)
 - Timer general register C_6 (TGRC_6)
 - Timer general register D_6 (TGRD_6)

- Channel 8:
 - Timer control register_8 (TCR_8)
 - Timer mode register_8 (TMDR_8)
 - Timer I/O control register_8 (TIOR_8)
 - Timer interrupt enable register_8 (TIER_8)
 - Timer status register_8 (TSR_8)
 - Timer counter_8 (TCNT_8)
 - Timer general register A_8 (TGRA_8)
 - Timer general register B_8 (TGRB_8)

- Channel 9:
 - Timer control register_9 (TCR_9)
 - Timer mode register_9 (TMDR_9)
 - Timer I/O control register H_9 (TIORH_9)
 - Timer I/O control register L_9 (TIORL_9)
 - Timer interrupt enable register_9 (TIER_9)
 - Timer status register_9 (TSR_9)
 - Timer counter_9 (TCNT_9)
 - Timer general register A_9 (TGRA_9)
 - Timer general register B_9 (TGRB_9)
 - Timer general register C_9 (TGRC_9)
 - Timer general register D_9 (TGRD_9)

- Channel 11:
 - Timer control register_11 (TCR_11)
 - Timer mode register_11 (TMDR_11)
 - Timer I/O control register_11 (TIOR_11)
 - Timer interrupt enable register_11 (TIER_11)
 - Timer status register_11 (TSR_11)
 - Timer counter_11 (TCNT_11)
 - Timer general register A_11 (TGRA_11)
 - Timer general register B_11 (TGRB_11)

- Common Registers:
 - Timer start register (TSTRB)
 - Timer synchronous register (TSYRB)

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 sources. See tables 12.4 and 12.5 for details.
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the input clock edge. For details, see table 14.6. When the input clock is counted using both rising and falling edges, the input clock period is halved (e.g. P_{ϕ} rising edges = $P_{\phi}/2$ rising edge). If phase counting mode is used on channels 1, 2, 4, and 5, this setting is ignored and the phase counting mode setting has priority. Clock edge selection is valid when the input clock is faster than or slower than the input clock source or slower. This setting is ignored if the input clock source is selected or when overflow/underflow of another channel is selected.
2	TPSC2	0	R/W	
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 12.7 to 12.12 for details. To select the input clock as the clock source, the DDR bit and ICR1 bit of the corresponding pin should be set to 0 and 1, respectively. For details, see section 13, I/O Ports.
0	TPSC0	0	R/W	

			synchronous operation* ¹
1	0	0	TCNT clearing disabled
1	0	1	TCNT cleared by TGRC compare match capture* ²
1	1	0	TCNT cleared by TGRD compare match capture* ²
1	1	1	TCNT cleared by counter clearing for a channel performing synchronous clearing 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 because the buffer register setting has priority, and compare match/input capture does not occur.

Table 14.5 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 match capture
	0	1	0	TCNT cleared by TGRB compare match capture
	0	1	1	TCNT cleared by counter clearing for a channel performing synchronous clearing 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 be modified.

Table 14.7 TPSC2 to TPSC0 (Channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on P ϕ /1
	0	0	1	Internal clock: counts on P ϕ /4
	0	1	0	Internal clock: counts on P ϕ /16
	0	1	1	Internal clock: counts on P ϕ /64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKB pin input
	1	1	0	External clock: counts on TCLKC pin input
	1	1	1	External clock: counts on TCLKD pin input

Table 14.8 TPSC2 to TPSC0 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on P ϕ /1
	0	0	1	Internal clock: counts on P ϕ /4
	0	1	0	Internal clock: counts on P ϕ /16
	0	1	1	Internal clock: counts on P ϕ /64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKB pin input
	1	1	0	Internal clock: counts on P ϕ /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 in
1	1	1	Internal clock: counts on P ϕ /1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 14.10 TPSC2 to TPSC0 (Channel 3)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3	0	0	0	Internal clock: counts on P ϕ /1
	0	0	1	Internal clock: counts on P ϕ /4
	0	1	0	Internal clock: counts on P ϕ /16
	0	1	1	Internal clock: counts on P ϕ /64
	1	0	0	External clock: counts on TCLKA pin in
	1	0	1	Internal clock: counts on P ϕ /1024
	1	1	0	Internal clock: counts on P ϕ /256
	1	1	1	Internal clock: counts on P ϕ /4096

1	1	0	Internal clock: counts on P ϕ /1024
1	1	1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

Table 14.12 TPSC2 to TPSC0 (Channel 5)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	0	Internal clock: counts on P ϕ /1
	0	0	1	Internal clock: counts on P ϕ /4
	0	1	0	Internal clock: counts on P ϕ /16
	0	1	1	Internal clock: counts on P ϕ /64
	1	0	0	External clock: counts on TCLKA pin i
	1	0	1	External clock: counts on TCLKC pin i
	1	1	0	Internal clock: counts on P ϕ /256
	1	1	1	External clock: counts on TCLKD pin i

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	—	Reserved These bits are always read as 1 and cannot be modified.
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to normally operate, and TGRD are to be used together for buffer operation. 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 modified. 0: TGRB operates normally 1: TGRB and TGRD used together for buffer operation
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to normally operate, and TGRC are to be used together for buffer operation. 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 modified. 0: TGRA operates normally 1: TGRA and TGRC used together for buffer operation
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 always be 0.
0	MD0	0	R/W	0. See table 14.13 for details.

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 should be written to MD2.

To designate the input capture pin in TIOR, the DDR bit and ICR bit for the corresponding should be set to 0 and 1, respectively. For details, see section 13, I/O Ports.

- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIOR_4, TIOR_5

Bit	7	6	5	4	3	2	1	
Bit Name	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	
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	

- TIORL_0, TORL_3

Bit	7	6	5	4	3	2	1	
Bit Name	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	
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	

1	IOA1	0	R/W	For details, see tables 12.22, 12.24 to 12.26, and 12.29.
0	IOA0	0	R/W	

- TIORL_0, TIORL_3

Bit	Bit Name	Initial Value	R/W	Description
7	IOD3	0	R/W	I/O Control D3 to D0
6	IOD2	0	R/W	Specify the function of TGRD.
5	IOD1	0	R/W	For details, see tables 12.15, and 12.19.
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 12.23, and 12.27.
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 TCNT_1 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/count-up/clock Input capture at TCNT_1 count-up/clock

[Legend]

x: Don't care

- Notes:
1. When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and P ϕ /1 is used as the TCNT_1 count clock, this setting is invalid and input capture is not generated.
 2. When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

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 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 TCNT_4 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/count-up/down Input capture at TCNT_4 count-up/count-down

[Legend]

x: Don't care

- Notes:
1. When bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and P ϕ /1 is used as the 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 register, this 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 compare match/input capture
					Input capture at generation of TGRC_3 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 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	1	Input capture register	Capture input source is TIOCA0 pin
					Input capture at rising edge
1	0	0	0		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

Note: * When the bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and P ϕ /1 is used a count clock of TCNT_1, this setting is invalid and input capture is not generate

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 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/count-up Input capture at TCNT_1 count-up/count-down

[Legend]

x: Don't care

- Note:
1. When the bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and P ϕ /1 is used as a count clock of TCNT_1, this setting is invalid and input capture is not generated.
 2. When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

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 compare 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

Note: * When the bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and P ϕ /1 is used a count clock of TCNT_4, this setting is invalid and input capture is not generate

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 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/count Input capture at TCNT_4 count-up/count

[Legend]

x: Don't care

- Notes:
1. When the bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and P ϕ /1 is used as count clock of TCNT_4, this setting is invalid and input capture is not generated.
 2. When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer register 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 compare match/input capture
					Input capture at generation of TGRA_3 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	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	<p>A/D Conversion Start Request Enable</p> <p>Enables/disables generation of A/D conversion start requests by TGRA input capture/compare match.</p> <p>0: A/D conversion start request generation disabled</p> <p>1: A/D conversion start request generation enabled</p>
6	—	1	—	<p>Reserved</p> <p>This bit is always read as 1 and cannot be modified.</p>
5	TCIEU	0	R/W	<p>Underflow Interrupt Enable</p> <p>Enables/disables interrupt requests (TCIU) by TCFU flag when the TCFU flag in TSR is set to 1 in channels 2, 4, and 5.</p> <p>In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified.</p> <p>0: Interrupt requests (TCIU) by TCFU disabled</p> <p>1: Interrupt requests (TCIU) by TCFU enabled</p>
4	TCIEV	0	R/W	<p>Overflow Interrupt Enable</p> <p>Enables/disables interrupt requests (TCIV) by TCFV flag when the TCFV flag in TSR is set to 1.</p> <p>0: Interrupt requests (TCIV) by TCFV disabled</p> <p>1: Interrupt requests (TCIV) by TCFV enabled</p>

when the TGFC bit in TSR is set to 1 in channels 0 and 1.
 In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified.

0: Interrupt requests (TGIC) by TGFC bit disabled

1: Interrupt requests (TGIC) by TGFC bit enabled

1	TGIEB	0	R/W	TGR Interrupt Enable B Enables/disables interrupt requests (TGIB) by the TGFB bit in TSR when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables/disables interrupt requests (TGIA) by the TGFA bit in TSR when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

Note: * The bit 7 in TIER of unit 1 is a reserved bit. This bit is always read as 0 and its value should not be changed.

14.3.5 Timer Status Register (TSR)

TSR indicates the status of each channel. The TPU has six TSR registers, one for each channel.

Bit	7	6	5	4	3	2	1
Bit Name	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB
Initial Value	1	1	0	0	0	0	0
R/W	R	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to bits 5 to 0, to clear flags.

5	TCFU	0	R/(W)*	<p>Underflow Flag</p> <p>Status flag that indicates that a TCNT underflow has occurred when channels 1, 2, 4, and 5 are set to phase counting. In channels 0 and 3, bit 5 is reserved. It is always read as 1 and cannot be modified.</p> <p>[Setting condition]</p> <p>When the TCNT value underflows (changes from H'0000 to H'FFFF)</p> <p>[Clearing condition]</p> <p>When a 0 is written to TCFU after reading TCFU = 1</p> <p>(When the CPU is used to clear this flag by writing 0 to it, the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)</p>
4	TCFV	0	R/(W)*	<p>Overflow Flag</p> <p>Status flag that indicates that a TCNT overflow has occurred. It is always read as 1 and cannot be modified.</p> <p>[Setting condition]</p> <p>When the TCNT value overflows (changes from H'FFFF to H'0000)</p> <p>[Clearing condition]</p> <p>When a 0 is written to TCFV after reading TCFV = 1</p> <p>(When the CPU is used to clear this flag by writing 0 to it, the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)</p>

- When TCNT value is transferred to TGRD while TGRC 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 TGFD
(When the CPU is used to clear this flag by writing 0 to it while the corresponding interrupt is enabled to read the flag after writing 0 to it.)

2	TGFC	0	R/(W)*	<p>Input Capture/Output Compare Flag C</p> <p>Status flag that indicates the occurrence of TGRC capture or compare match in channels 0 and 3. In channels 1, 2, 4, and 5, bit 2 is reserved. It is read as 0 and cannot be modified.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by TGRC capture signal while TGRC is functioning as capture register <p>[Clearing conditions]</p> <ul style="list-style-type: none"> • When DTC is activated by a TGIC interrupt DISEL bit in MRB of DTC is 0 • When 0 is written to TGFC after reading TGFC (When the CPU is used to clear this flag by writing 0 to it 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 TGIB interrupt while the DISEL bit in MRB of DTC is 0
- When 0 is written to TGFB after reading TGFB (When the CPU is used to clear this flag by writing 0 to it, while the corresponding interrupt is enabled, to read the flag after writing 0 to it.)

0	TGFA	0	R/(W)*	Input Capture/Output Compare Flag A
---	------	---	--------	-------------------------------------

Status flag that indicates the occurrence of TGRA capture or compare match.

[Setting conditions]

- When TCNT = TGRA while TGRA is functioning as output compare register
- When TCNT value is transferred to TGRA by capture signal while TGRA is functioning as input capture register

[Clearing conditions]

- When DTC is activated by a TGIA interrupt while the DISEL bit in MRB of DTC is 0
- When DMAC is activated by a TGIA interrupt while the DTA bit in DMDR of DTC is 1
- When 0 is written to TGFA after reading TGFA (When the CPU is used to clear this flag by writing 0 to it, while the corresponding interrupt is enabled, 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

14.3.7 Timer General Register (TGR)

TGR is a 16-bit readable/writable register with a dual function as output compare and input capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two each for channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for input capture operation as buffer registers. The TGR registers cannot be accessed in 8-bit units; they must always be accessed in 16-bit units. TGR and buffer register combinations during buffer operation 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	—	Reserved The write value should 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 TCNT
3	CST3	0	R/W	If 0 is written to the CST bit during operation with
2	CST2	0	R/W	TIOC pin designated for output, the counter stop
1	CST1	0	R/W	TIOC pin output compare output level is retained
0	CST0	0	R/W	is written to when the CST bit is cleared to 0, the output level will be changed to the set initial outp 0: TCNT_5 to TCNT_0 count operation is stoppe 1: TCNT_5 to TCNT_0 performs count operation

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R/W	Reserved The write value should always be 0.
5	SYNC5	0	R/W	Timer Synchronization 5 to 0
4	SYNC4	0	R/W	These bits select whether operation is independent or synchronized with other channels.
3	SYNC3	0	R/W	
2	SYNC2	0	R/W	When synchronous operation is selected, synchronous presetting of multiple channels, and synchronous clearing through counter clearing on another channel are possible.
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	To set synchronous operation, the SYNC bits for both channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT source must also be set by means of bits CCLF0 and CCLR0 in TCR. 0: TCNT_5 to TCNT_0 operate independently (synchronous presetting/clearing is unrelated to other channels) 1: TCNT_5 to TCNT_0 perform synchronous operation (TCNT synchronous presetting/synchronous clearing is possible)

When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and

(a) Example of count operation setting procedure

Figure 14.3 shows an example of the count operation setting procedure.

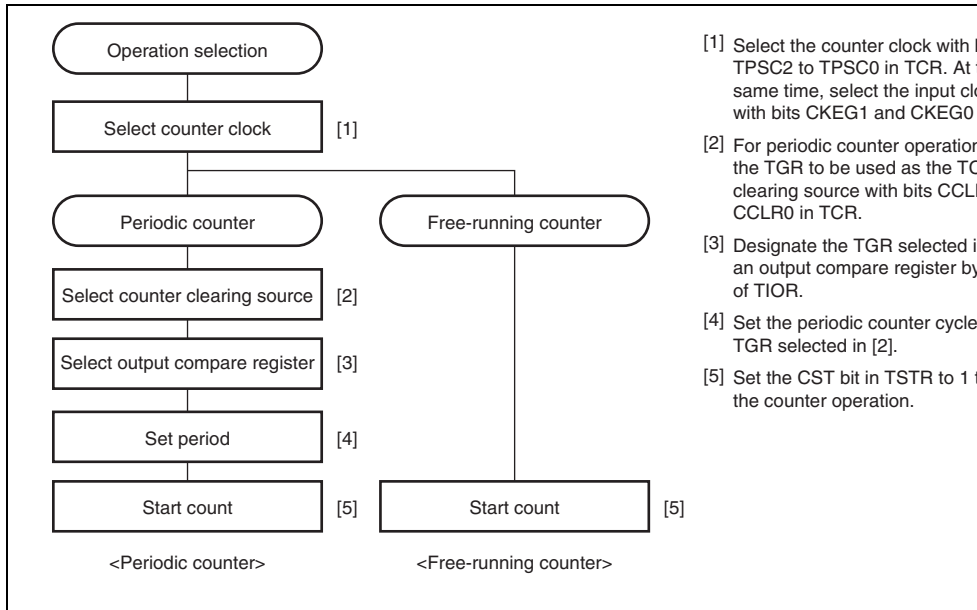


Figure 14.3 Example of Counter Operation Setting Procedure

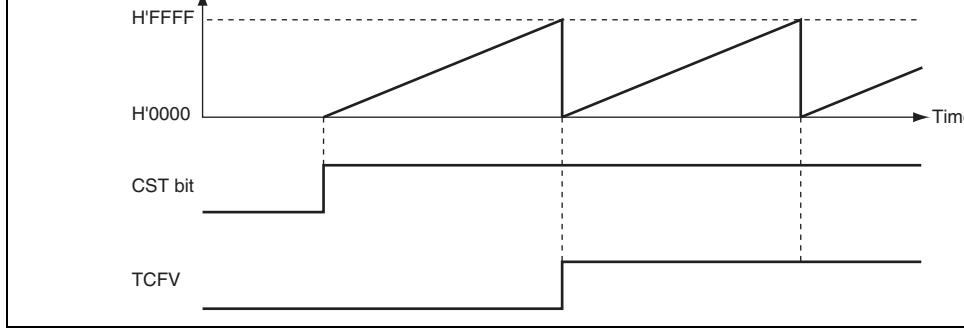


Figure 14.4 Free-Running Counter Operation

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 designated as an output compare register, and counter clearing by compare match is selected by means of the CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts count-up operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value reaches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

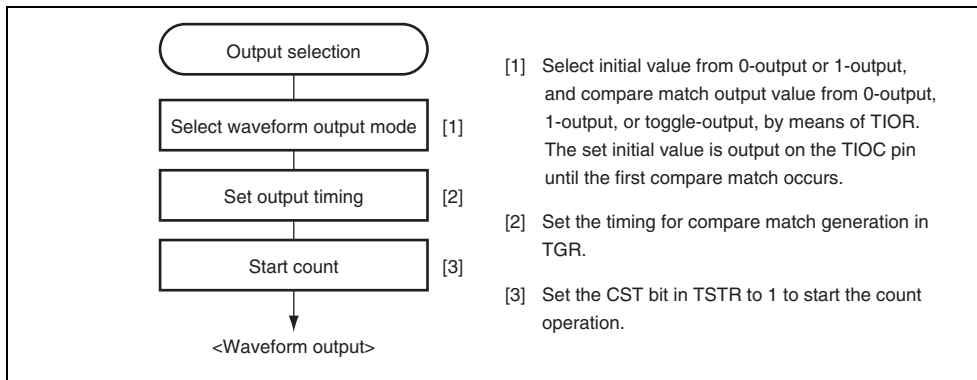
If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests a compare match interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 14.5 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 compare match.

(a) Example of setting procedure for waveform output by compare match

Figure 14.6 shows an example of the setting procedure for waveform output by a compare match.

**Figure 14.6 Example of Setting Procedure for Waveform Output by Compare Match**

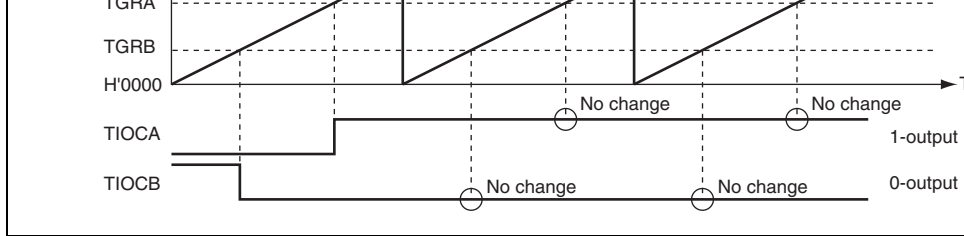


Figure 14.7 Example of 0-Output/1-Output Operation

Figure 14.8 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

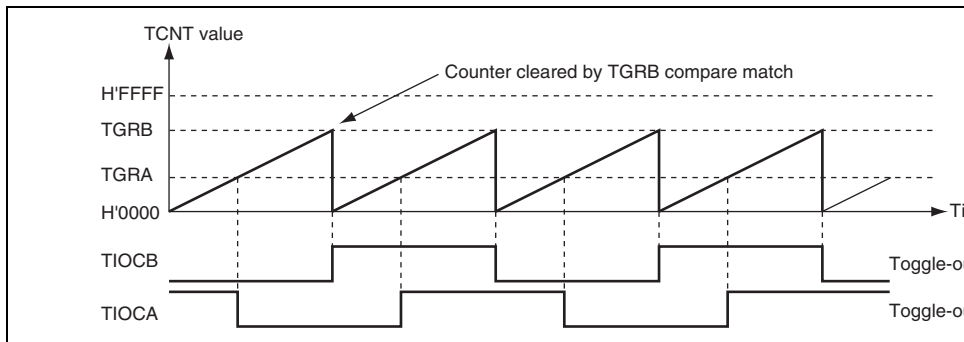


Figure 14.8 Example of Toggle Output Operation

(a) Example of setting procedure for input capture operation

Figure 14.9 shows an example of the setting procedure for input capture operation.

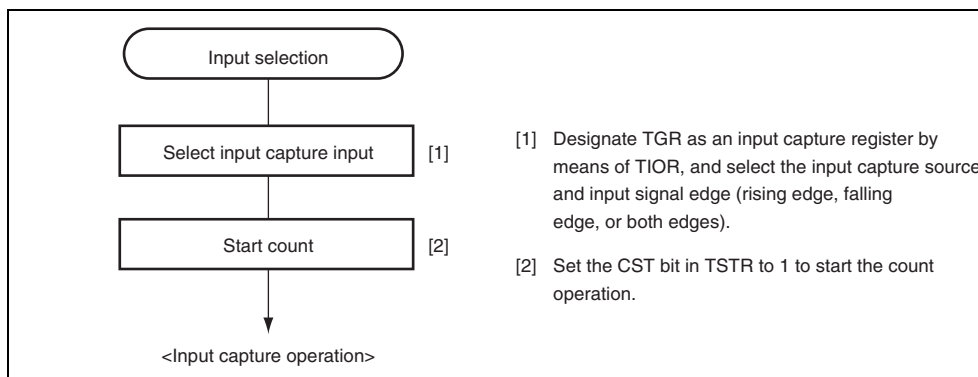


Figure 14.9 Example of Setting Procedure for Input Capture Operation

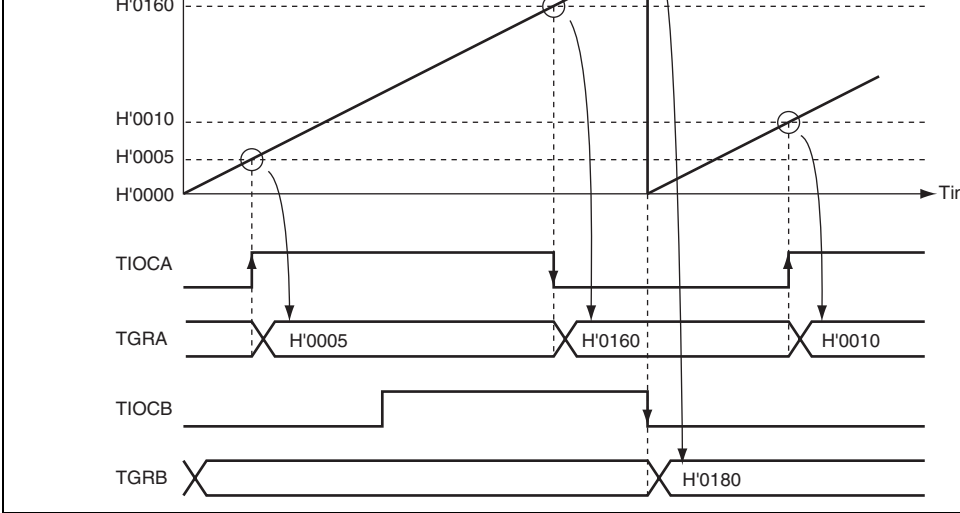


Figure 14.10 Example of Input Capture Operation

Figure 14.11 shows an example of the synchronous operation setting procedure.

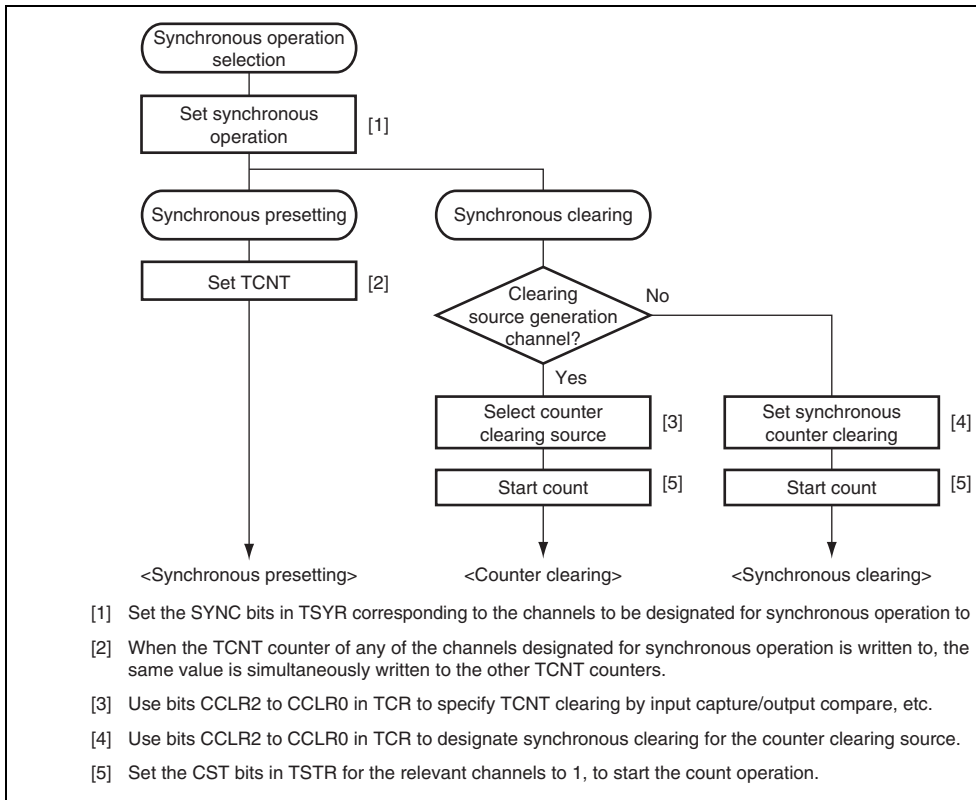


Figure 14.11 Example of Synchronous Operation Setting Procedure

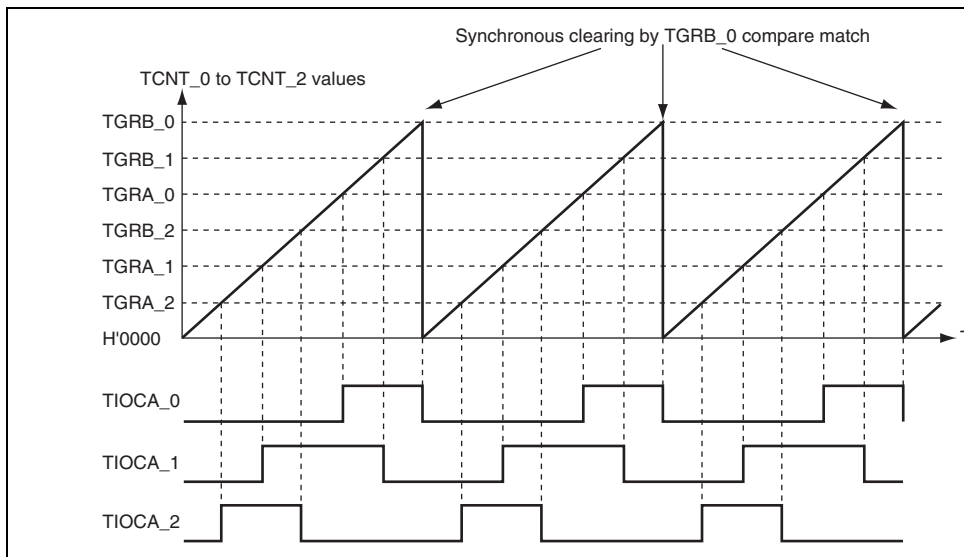


Figure 14.12 Example of Synchronous Operation

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 channel is transferred to the timer general register.
 This operation is illustrated in figure 14.13.

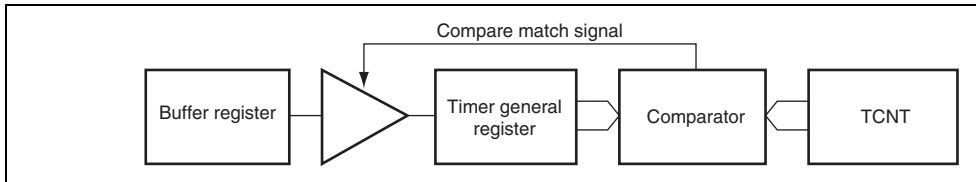


Figure 14.13 Compare Match Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 14.15 shows an example of the buffer operation setting procedure.

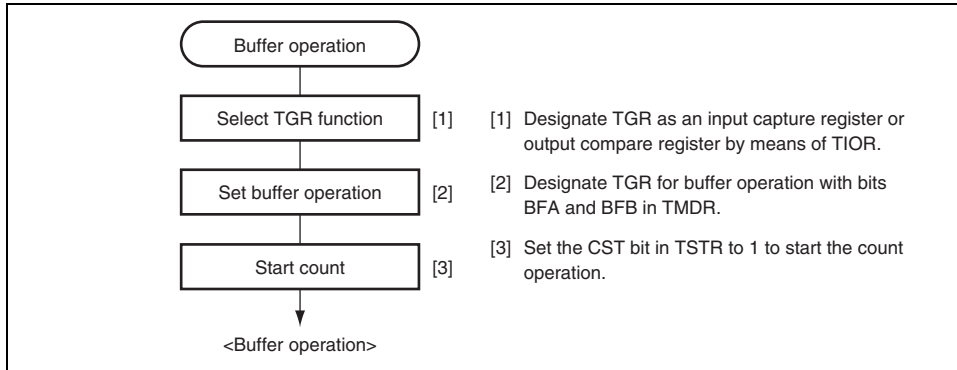


Figure 14.15 Example of Buffer Operation Setting Procedure

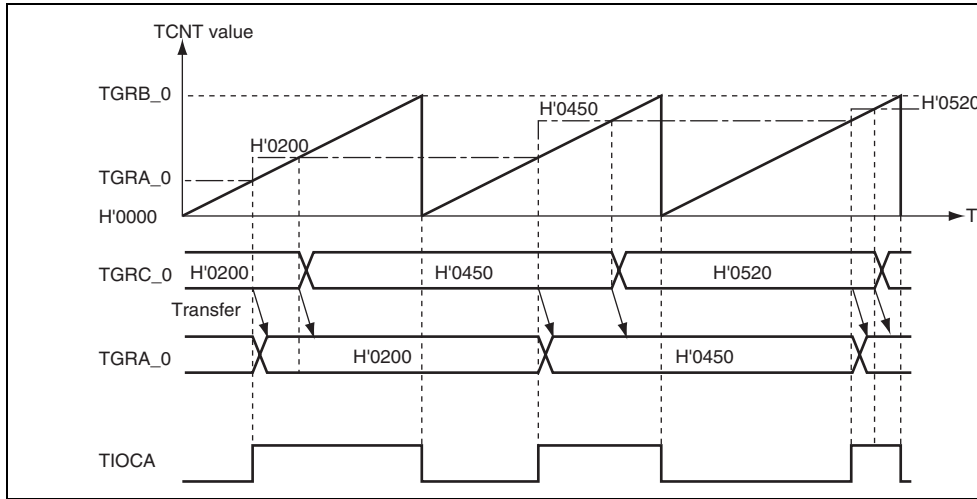


Figure 14.16 Example of Buffer Operation (1)

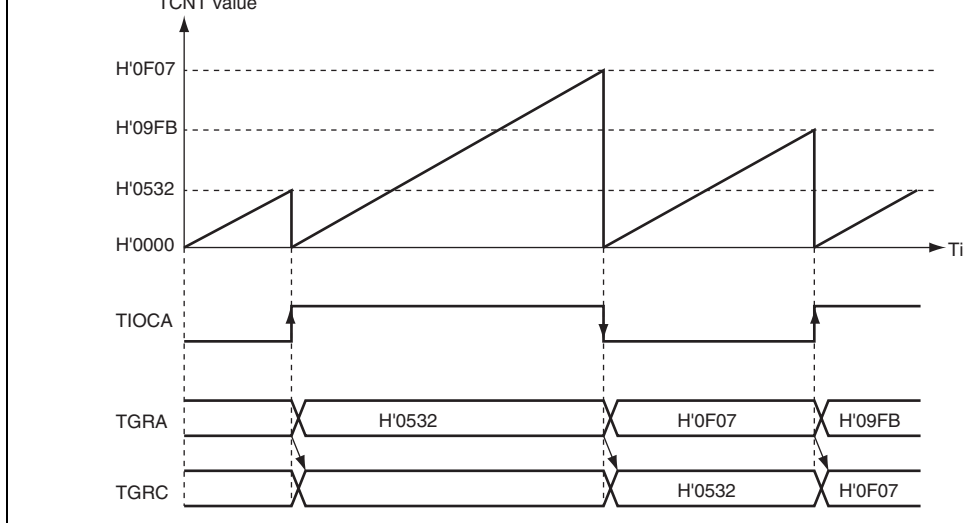


Figure 14.17 Example of Buffer Operation (2)

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is i and the counter operates independently in phase counting mode.

Table 14.31 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 14.18 shows an example of the setting procedure for cascaded operation.

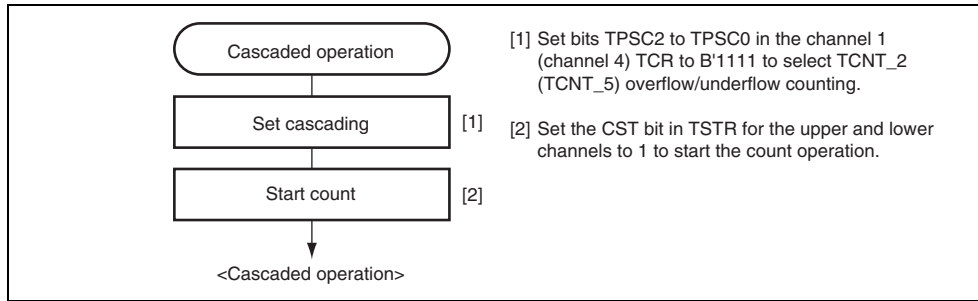


Figure 14.18 Cascaded Operation Setting Procedure

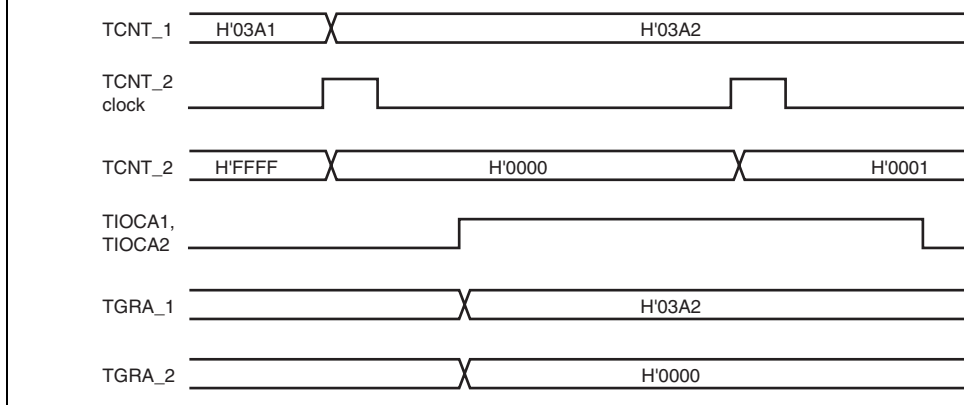


Figure 14.19 Example of Cascaded Operation (1)

Figure 14.20 illustrates the operation when counting upon TCNT_2 overflow/underflow 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.

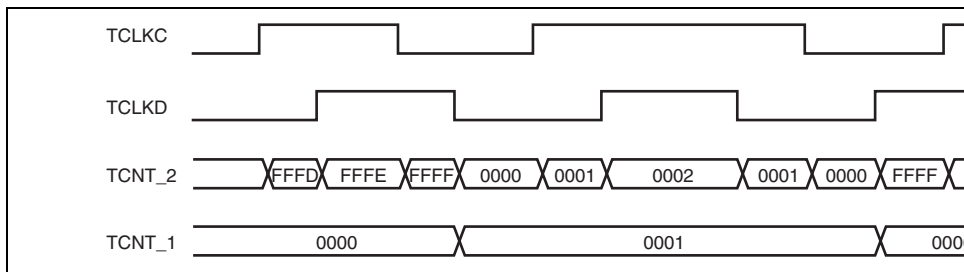


Figure 14.20 Example of Cascaded Operation (2)

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 TGRC and TGRB with TGRD. The outputs specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR are output from the TIOCA and TIOCC pins at compare matches A and C, respectively. The outputs specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR are output at compare matches B and D, respectively. The initial output value is the value set in TGRA or TGRB. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.

In PWM mode 1, a maximum 8-phase PWM output is possible.

2. PWM mode 2

PWM output is generated using one TGR as the cycle register and the others as duty cycle registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a synchronous register compare match, the output value of each pin is set to the initial value set in TIOR. If the set values of the cycle and duty cycle registers are identical, the output value does not change when a compare match occurs.

In PWM mode 2, a maximum 15-phase PWM output is possible by combined use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 14.32.

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 cy

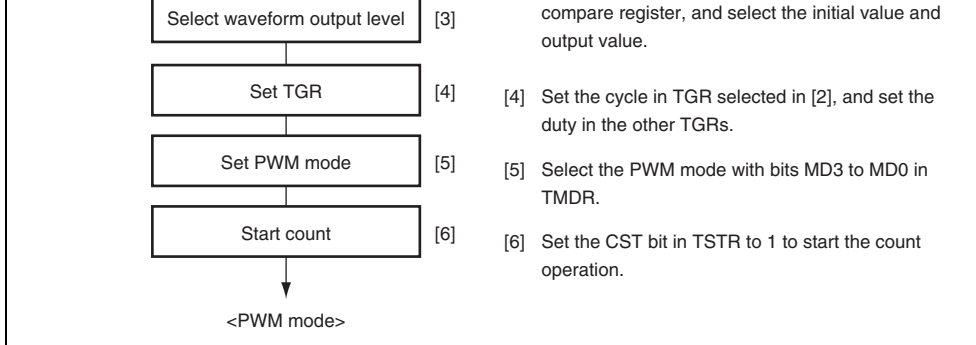


Figure 14.21 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 14.22 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the 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 registers the duty cycle.

In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), to output a 50% duty cycle PWM waveform.

In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other TGR registers are used to set the duty cycle.

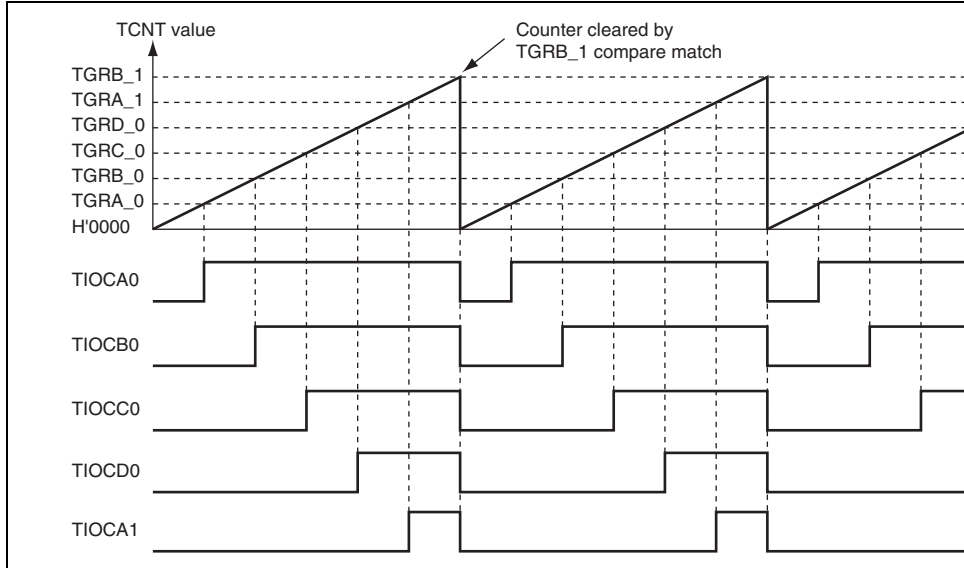


Figure 14.23 Example of PWM Mode Operation (2)

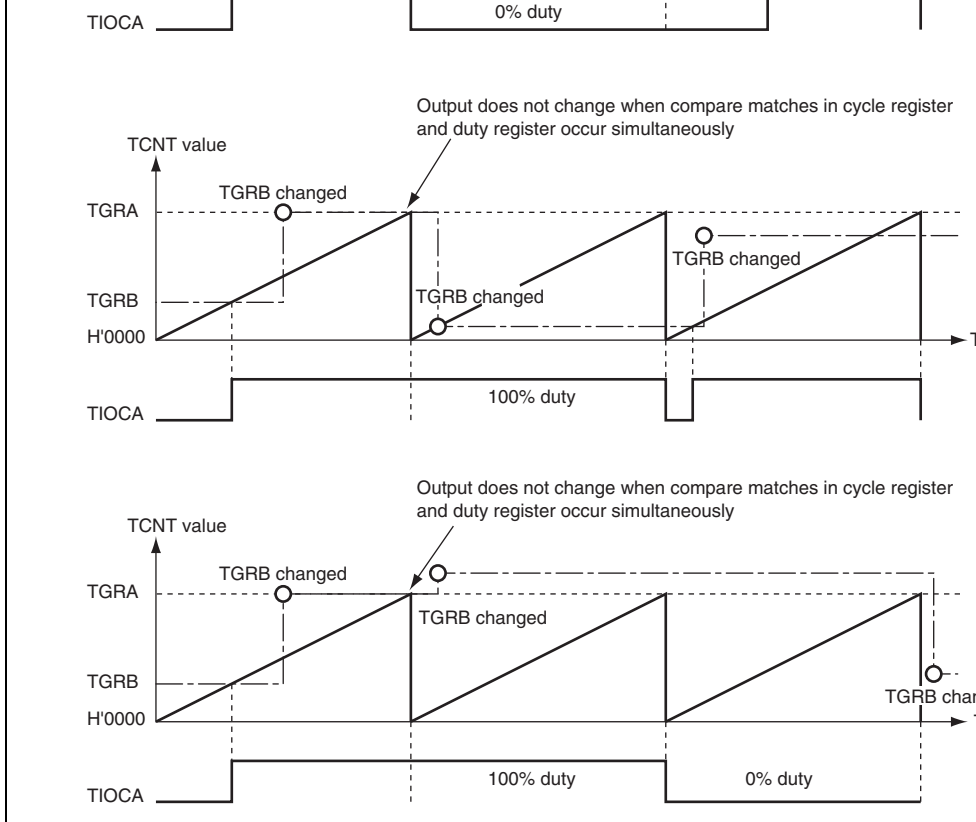


Figure 14.24 Example of PWM Mode Operation (3)

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 underflow occurs while TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.

Table 14.33 shows the correspondence between external clock pins and channels.

Table 14.33 Clock Input Pins in Phase Counting Mode

Channels	External Clock Pins	
	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 14.25 Example of Phase Counting Mode Setting Procedure

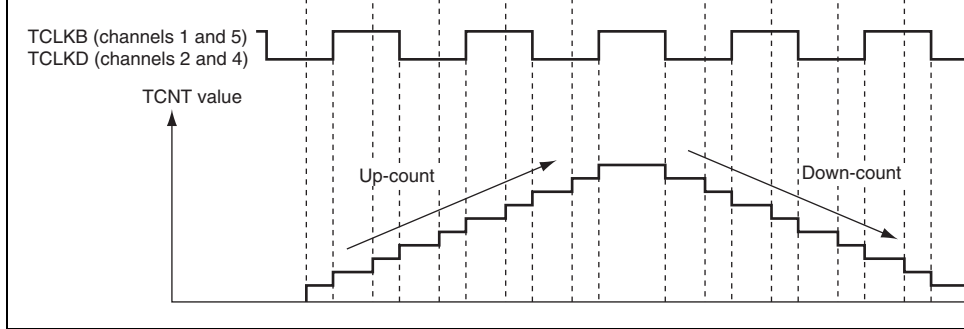


Figure 14.26 Example of Phase Counting Mode 1 Operation

Table 14.34 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	Down-count
	High level	
High level		
Low level		
	High level	
	Low level	

[Legend]

: Rising edge

: Falling edge

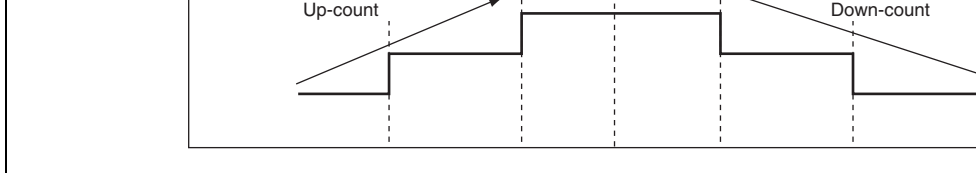


Figure 14.27 Example of Phase Counting Mode 2 Operation

Table 14.35 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	\uparrow	Don't care
Low level	\downarrow	Don't care
\uparrow	Low level	Don't care
\downarrow	High level	Up-count
High level	\downarrow	Don't care
Low level	\uparrow	Don't care
\uparrow	High level	Don't care
\downarrow	Low level	Down-count

[Legend]

\uparrow : Rising edge

\downarrow : Falling edge

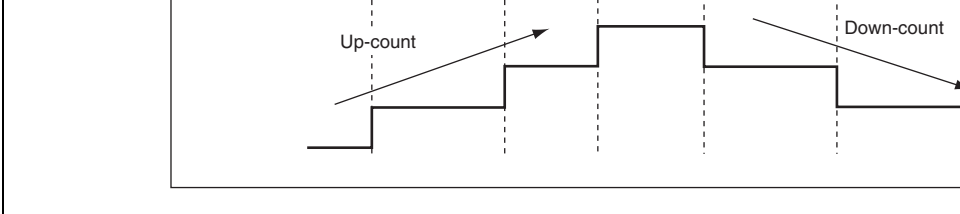


Figure 14.28 Example of Phase Counting Mode 3 Operation

Table 14.36 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

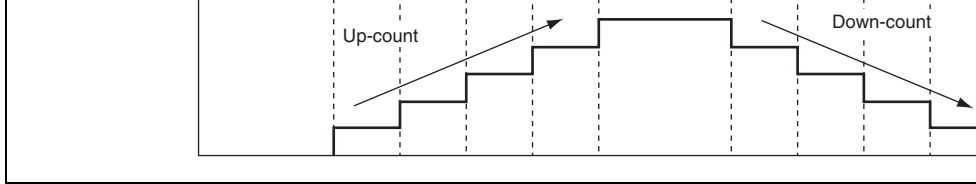


Figure 14.29 Example of Phase Counting Mode 4 Operation

Table 14.37 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		Up-count
	Low level	Don't care
	High level	Don't care
High level		Down-count
Low level		Down-count
	High level	Don't care
	Low level	Don't care

[Legend]

: Rising edge

: Falling edge

in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input capture 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-count values for the control cycles are stored.

This procedure enables accurate position/speed detection to be achieved.

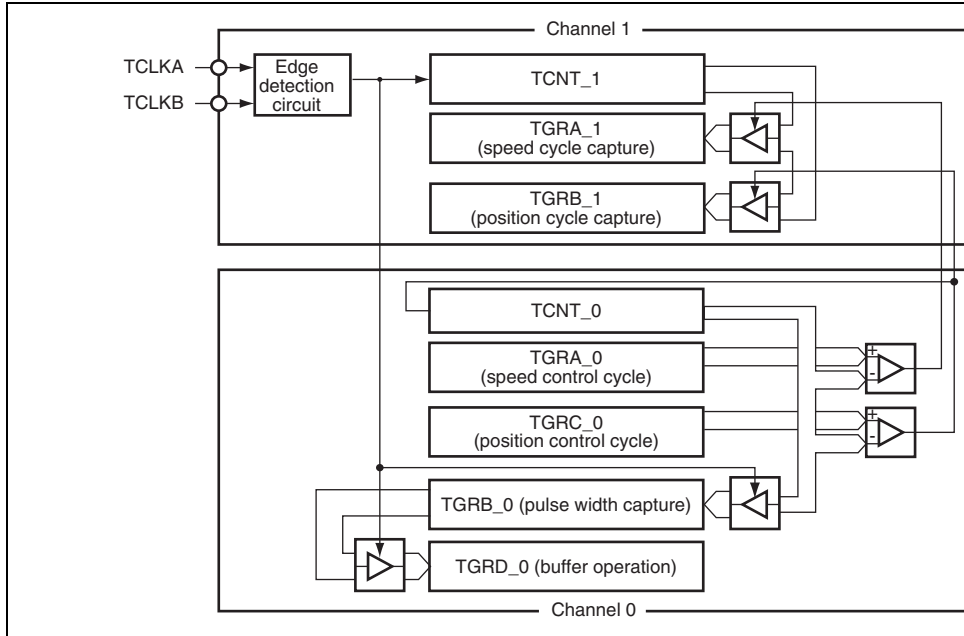


Figure 14.30 Phase Counting Mode Application Example

channel is fixed. For details, see section 7, Interrupt Controller.

Table 14.38 lists the TPU interrupt sources.

Table 14.38 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMA Activation
0	TGI0A	TGRA_0 input capture/compare match	TGFA_0	Possible	Possible
	TGI0B	TGRB_0 input capture/compare match	TGFB_0	Possible	Not possible
	TGI0C	TGRC_0 input capture/compare match	TGFC_0	Possible	Not possible
	TGI0D	TGRD_0 input capture/compare match	TGFD_0	Possible	Not possible
	TCI0V	TCNT_0 overflow	TCFV_0	Not possible	Not possible
1	TGI1A	TGRA_1 input capture/compare match	TGFA_1	Possible	Possible
	TGI1B	TGRB_1 input capture/compare match	TGFB_1	Possible	Not possible
	TCI1V	TCNT_1 overflow	TCFV_1	Not possible	Not possible
	TCI1U	TCNT_1 underflow	TCFU_1	Not possible	Not possible
2	TGI2A	TGRA_2 input capture/compare match	TGFA_2	Possible	Possible
	TGI2B	TGRB_2 input capture/compare match	TGFB_2	Possible	Not possible
	TCI2V	TCNT_2 overflow	TCFV_2	Not possible	Not possible
	TCI2U	TCNT_2 underflow	TCFU_2	Not possible	Not possible

	TCI4V	TCNT_4 overflow	TCFV_4	Not possible	Not possible
	TCI4U	TCNT_4 underflow	TCFU_4	Not possible	Not possible
5	TGI5A	TGRA_5 input capture/compare match	TGFA_5	Possible	Possible
	TGI5B	TGRB_5 input capture/compare match	TGFB_5	Possible	Not possible
	TCI5V	TCNT_5 overflow	TCFV_5	Not possible	Not possible
	TCI5U	TCNT_5 underflow	TCFU_5	Not possible	Not possible

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

(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 set to 1 by the occurrence of a TGR input capture/compare match on a channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has 16 input capture/compare match interrupts, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

(2) Overflow Interrupt

An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of a TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The TPU has six overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of a TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The TPU has four underflow interrupts, one each for channels 0, 3, 4, and 5.

The A/D converter can be activated by the TGRA input capture/compare match interrupt for a channel. For details, see section 10, DMA Controller (DMAC).

In TPU, one in each channel, totally six TGRA input capture/compare match interrupts can be used as DMAC activation sources.

14.8 A/D Converter Activation

Concerning the unit 0 in TPU, the TGRA input capture/compare match for each channel can be used to activate the A/D converter. (However, the A/D converter cannot be activated in unit 1.)

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

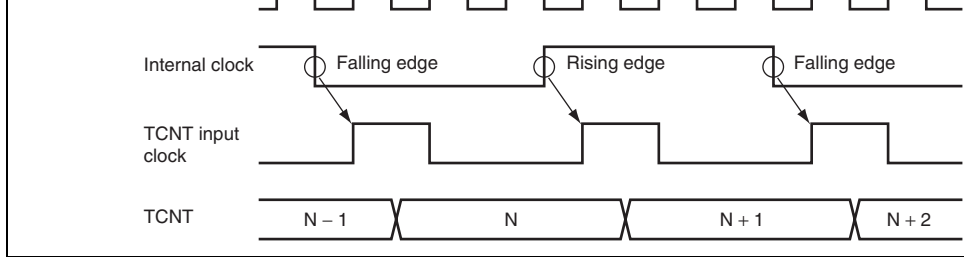


Figure 14.31 Count Timing in Internal Clock Operation

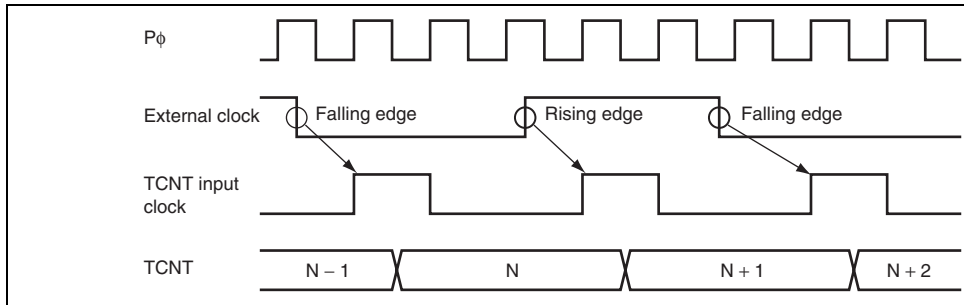


Figure 14.32 Count Timing in External Clock Operation

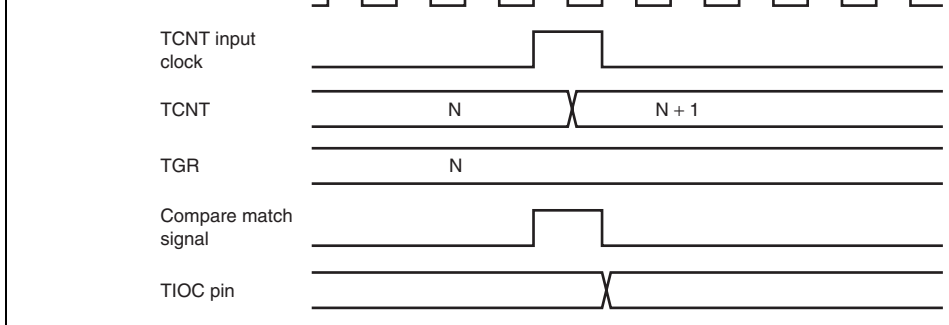


Figure 14.33 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 14.34 shows input capture signal timing.

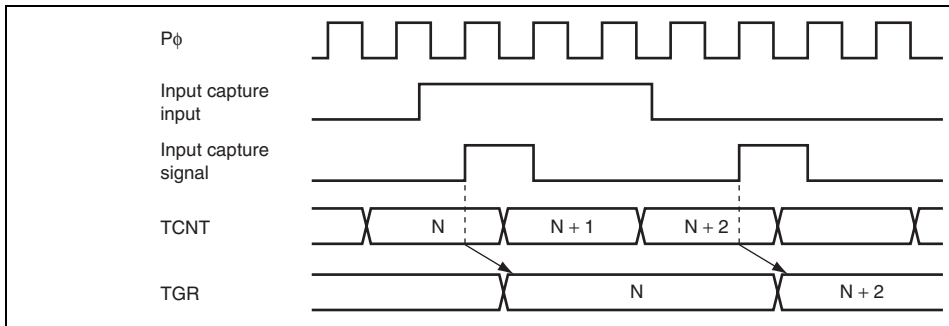


Figure 14.34 Input Capture Input Signal Timing

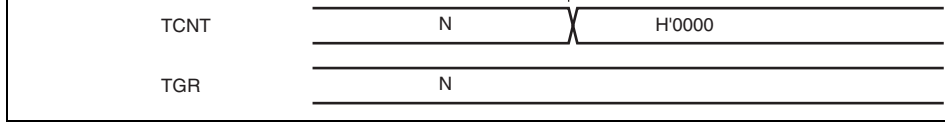


Figure 14.35 Counter Clear Timing (Compare Match)

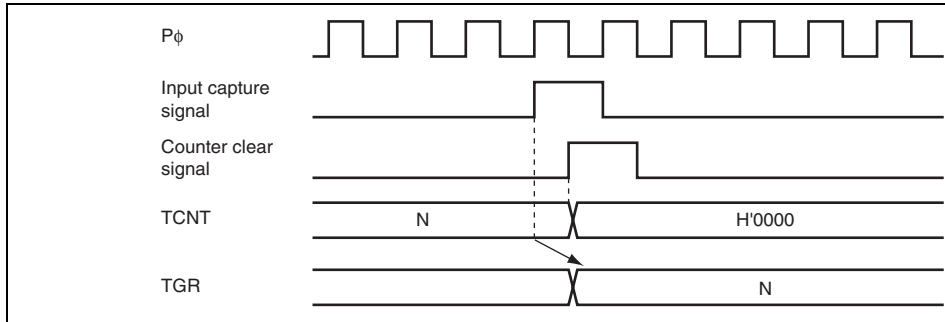


Figure 14.36 Counter Clear Timing (Input Capture)

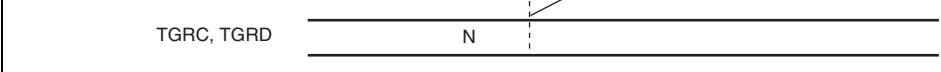


Figure 14.37 Buffer Operation Timing (Compare Match)

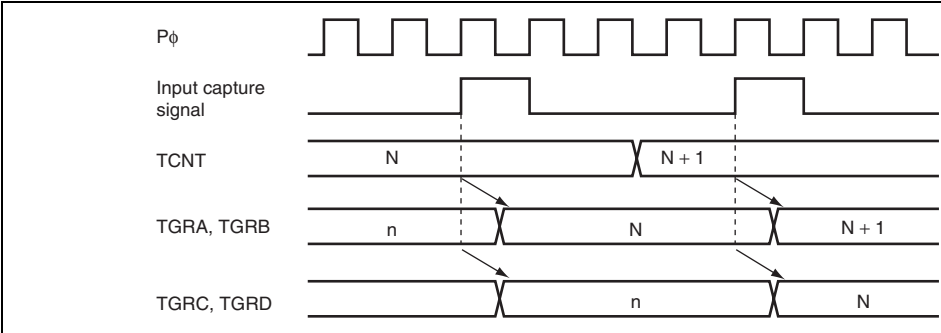


Figure 14.38 Buffer Operation Timing (Input Capture)

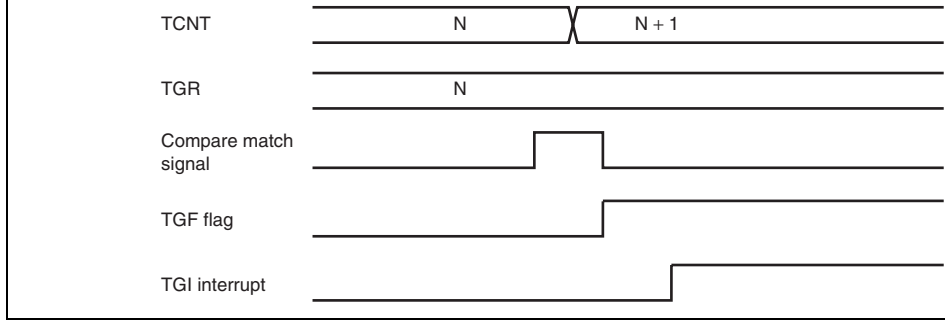


Figure 14.39 TGI Interrupt Timing (Compare Match)

(2) TGF Flag Setting Timing in Case of Input Capture

Figure 14.40 shows the timing for setting of the TGF flag in TSR by input capture occurring at the TGI interrupt request signal timing.

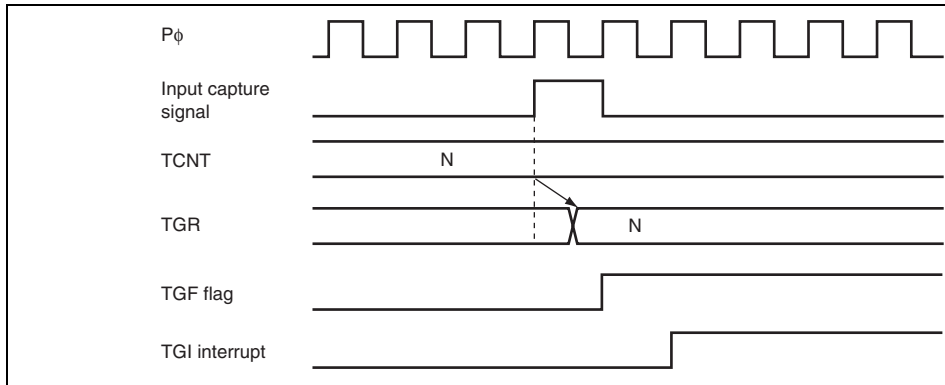


Figure 14.40 TGI Interrupt Timing (Input Capture)

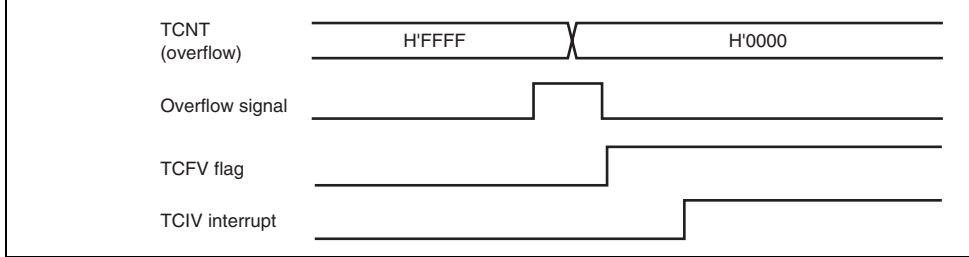


Figure 14.41 TCIV Interrupt Setting Timing

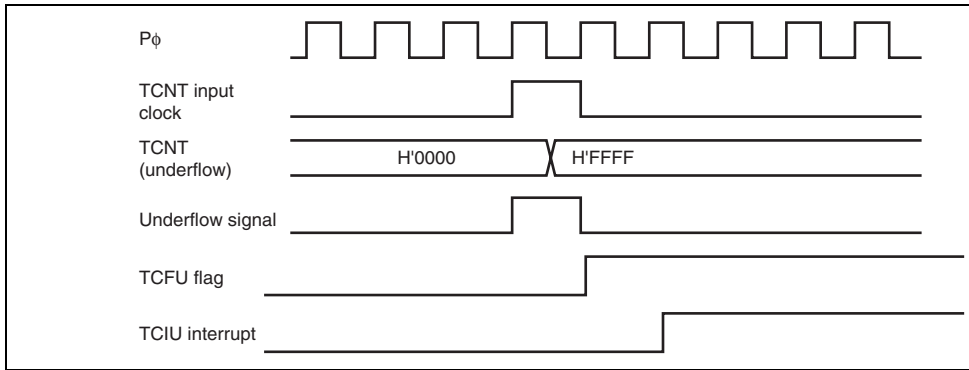


Figure 14.42 TCIU Interrupt Setting Timing

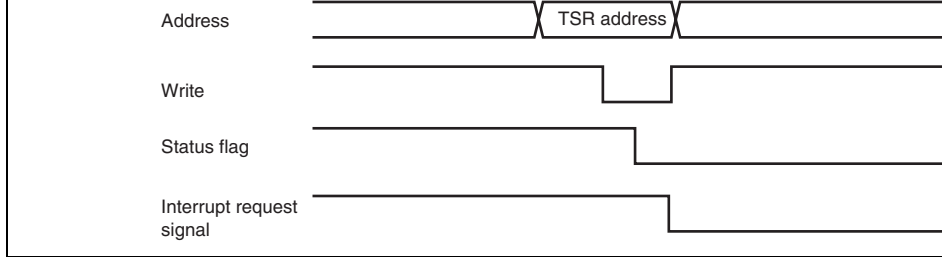


Figure 14.43 Timing for Status Flag Clearing by CPU

The status flag and interrupt request signal are cleared in synchronization with $P\phi$ after the DMAC transfer has started, as shown in figure 14.44. If conflict occurs for clearing the status flag and interrupt request signal due to activation of multiple DTC or DMAC transfers, it will be masked for five clock cycles ($P\phi$) for clearing them, as shown in figure 14.45. The next transfer request signal is masked for a longer period of either a period until the current transfer ends or a period for five clock cycles ($P\phi$) from the beginning of the transfer. Note that in the DTC transfer, the status flag may be cleared during outputting the destination address.

Figure 14.44 Timing for Status Flag Clearing by DTC/DMAC Activation (1)

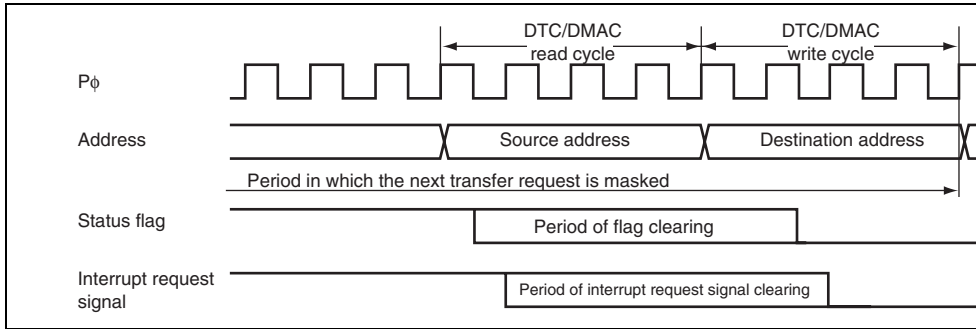


Figure 14.45 Timing for Status Flag Clearing by DTC/DMAC Activation (2)

The input clock pulse width must be at least 1.5 states in the case of single-edge detection and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 14.46 shows the input conditions in phase counting mode.

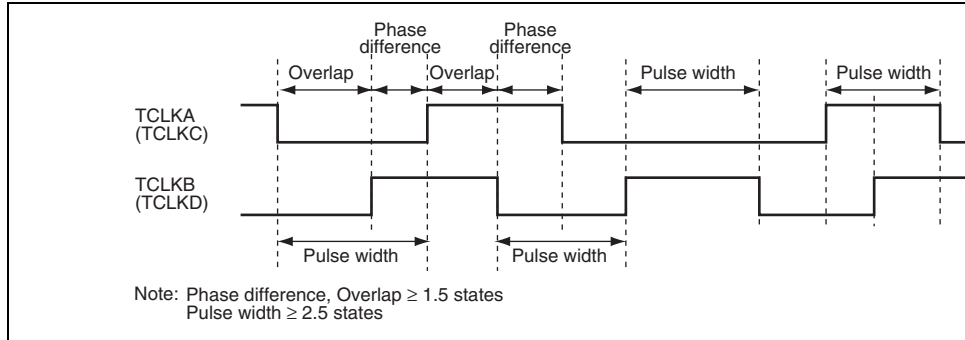


Figure 14.46 Phase Difference, Overlap, and Pulse Width in Phase Counting

14.10.4 Conflict between TCNT Write and Clear Operations

If the counter clearing signal is generated in the T2 state of a TCNT write cycle, TCNT c takes precedence and the TCNT write is not performed. Figure 14.47 shows the timing in case.

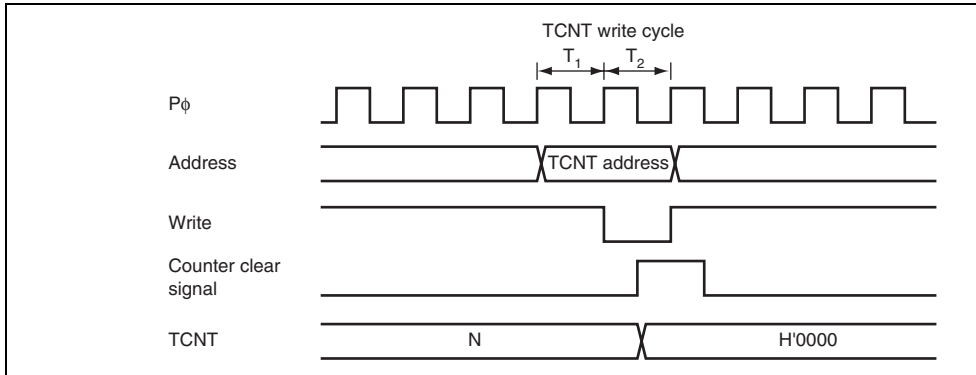


Figure 14.47 Conflict between TCNT Write and Clear Operations

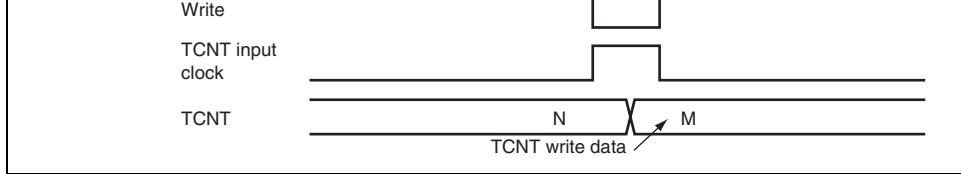


Figure 14.48 Conflict between TCNT Write and Increment Operations

14.10.6 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 precedence and the compare match signal is disabled. A compare match also does not occur when the value as before is written.

Figure 14.49 shows the timing in this case.

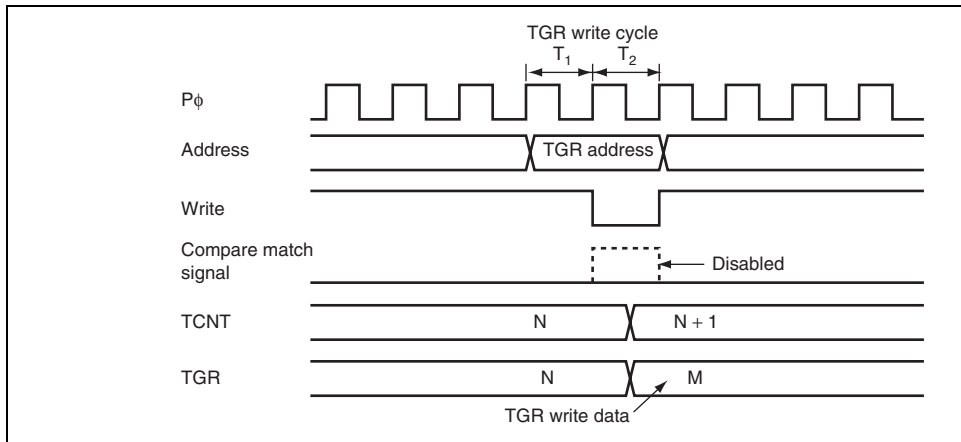


Figure 14.49 Conflict between TGR Write and Compare Match

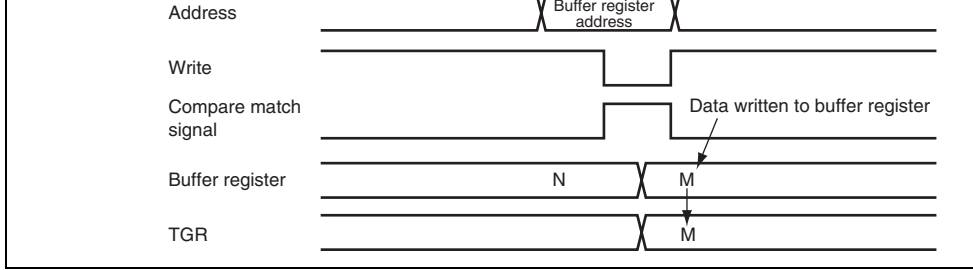


Figure 14.50 Conflict between Buffer Register Write and Compare Match

14.10.8 Conflict between TGR Read and Input Capture

If the input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data after input capture transfer.

Figure 14.51 shows the timing in this case.

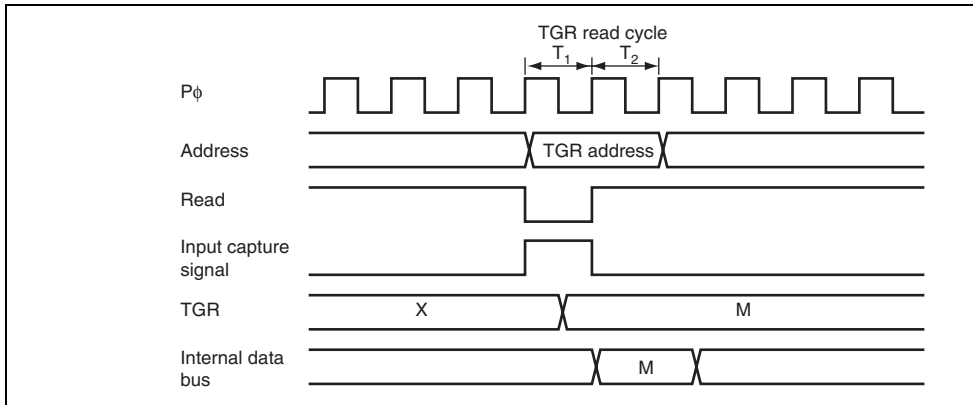


Figure 14.51 Conflict between TGR Read and Input Capture

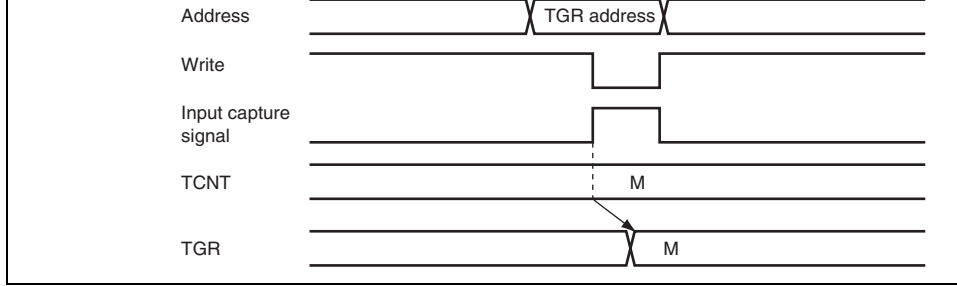


Figure 14.52 Conflict between TGR Write and Input Capture

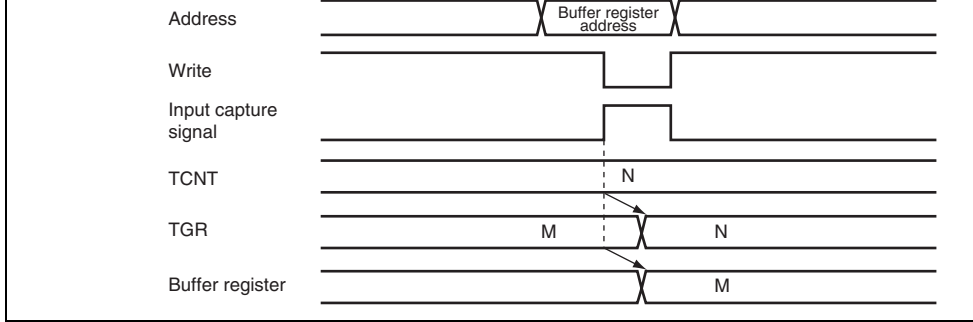


Figure 14.53 Conflict between Buffer Register Write and Input Capture

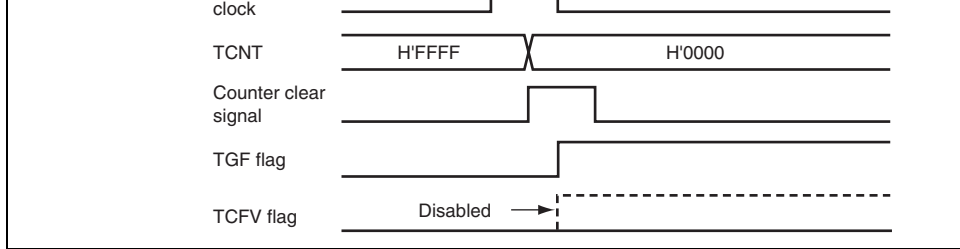


Figure 14.54 Conflict between Overflow and Counter Clearing

14.10.12 Conflict between TCNT Write and Overflow/Underflow

If an overflow/underflow occurs due to increment/decrement in the T2 state of a TCNT cycle, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 14.55 shows the operation timing when there is conflict between TCNT write and overflow.

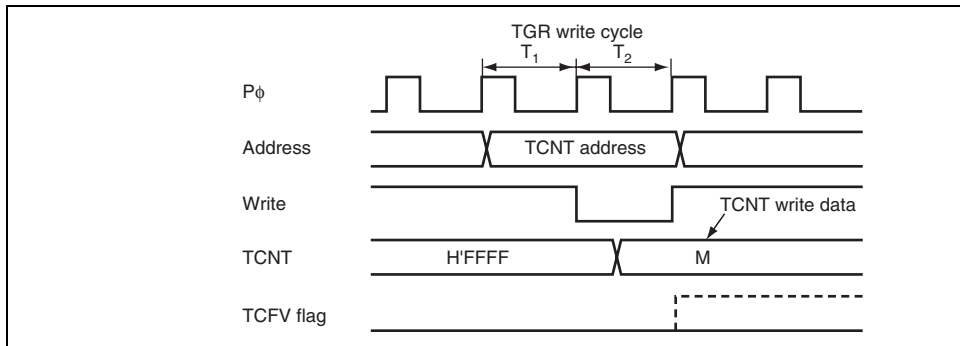


Figure 14.55 Conflict between TCNT Write and Overflow

be cleared to halt the output. For details, see section 13, I/O Ports.

14.10.15 Interrupts and Module Stop Mode

If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC and DMAC activation sources. Interrupts should therefore be disabled before entering module stop mode.

- 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

Table 15.1 List of PPG Functions

	Function	PPG0	PPG1	
PPG output trigger	TPU0	Compare match	Possible	Not possible
		Input capture	Possible	Not possible
	TPU1	Compare match	Not possible	Possible
		Input capture	Not possible	Not possible
Non-overlapping mode		Possible	Possible	
Output data transfer	DTC	Possible	Possible	
	DMAC	Possible	Possible	
Inverted output		Possible	Possible	

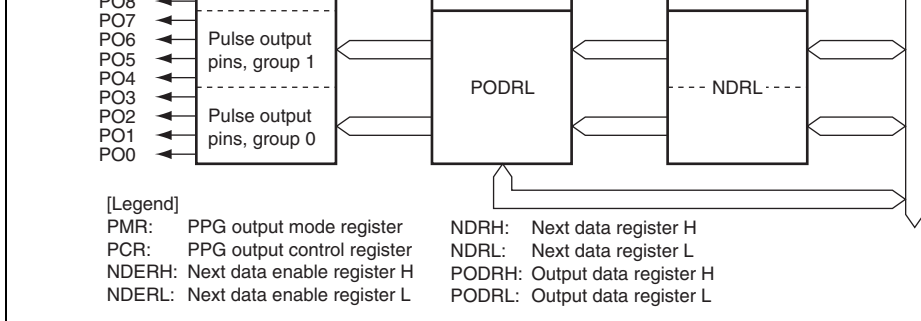


Figure 15.1 Block Diagram of PPG (Unit 0)

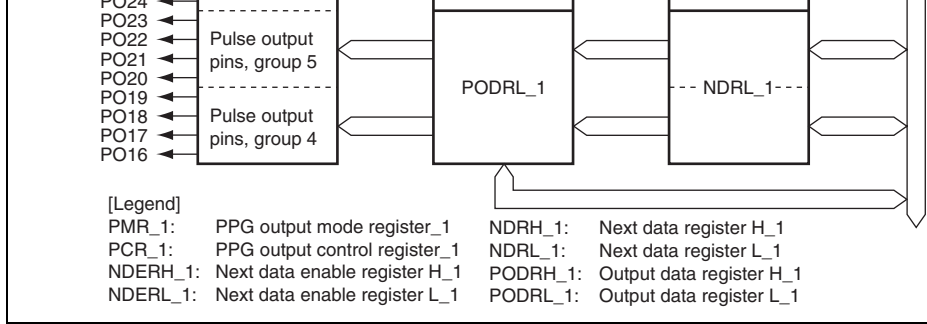


Figure 15.2 Block Diagram of PPG (Unit 1)

PO3	Output	
PO4	Output	Group 1 pulse output
PO5	Output	
PO6	Output	
PO7	Output	
PO8	Output	Group 2 pulse output
PO9	Output	
PO10	Output	
PO11	Output	
PO12	Output	Group 3 pulse output
PO13	Output	
PO14	Output	
PO15	Output	

PO24	Output	Group 6 pulse output
PO25	Output	
PO26	Output	
PO27	Output	
PO28	Output	Group 7 pulse output
PO29	Output	
PO30	Output	
PO31	Output	

- Next data register H (NDRH)
- Next data register L (NDRL)
- PPG output control register (PCR)
- PPG output mode register (PMR)

Unit 1:

- Next data enable register H_1 (NDERH_1)
- Next data enable register L_1 (NDERL_1)
- Output data register H_1 (PODRH_1)
- Output data register L_1 (PODRL_1)
- Next data register H_1 (NDRH_1)
- Next data register L_1 (NDRL_1)
- PPG output control register_1 (PCR_1)
- PPG output mode register_1 (PMR_1)

- 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

- NDERH

Bit	Bit Name	Initial Value	R/W	Description
7	NDER15	0	R/W	Next Data Enable 15 to 8
6	NDER14	0	R/W	When a bit is set to 1, the value in the corresponding NDRH bit is transferred to the PODRH bit by the output trigger. Values are not transferred from the PODRH for cleared bits.
5	NDER13	0	R/W	
4	NDER12	0	R/W	
3	NDER11	0	R/W	
2	NDER10	0	R/W	
1	NDER9	0	R/W	
0	NDER8	0	R/W	

1	NDER1	0	R/W
0	NDER0	0	R/W

- NDERH_1

Bit	Bit Name	Initial Value	R/W	Description
7	NDER31	0	R/W	Next Data Enable 31 to 24
6	NDER30	0	R/W	When a bit is set to 1, the value in the corresponding
5	NDER29	0	R/W	NDRH_1 bit is transferred to the PODRH_1 bit b
4	NDER28	0	R/W	selected output trigger. Values are not transferred
3	NDER27	0	R/W	NDRH_1 to PODRH_1 for cleared bits.
2	NDER26	0	R/W	
1	NDER25	0	R/W	
0	NDER24	0	R/W	

1	NDER17	0	R/W
0	NDER16	0	R/W

15.3.2 Output Data Registers H, L (PODRH, PODRL)

PODRH and PODRL store output data for use in pulse output. A bit that has been set for output by NDER is read-only and cannot be modified.

- PODRH

Bit	7	6	5	4	3	2	1
Bit Name	POD15	POD14	POD13	POD12	POD11	POD10	POD9
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

- 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

1	POD9	0	R/W
0	POD8	0	R/W

- **PODRL**

Bit	Bit Name	Initial Value	R/W	Description
7	POD7	0	R/W	Output Data Register 7 to 0
6	POD6	0	R/W	For bits which have been set to pulse output by the output trigger transfers NDRL values to this register during PPG operation. While NDERL is set to 1, cannot write to this register. While NDERL is cleared, initial output value of the pulse can be set.
5	POD5	0	R/W	
4	POD4	0	R/W	
3	POD3	0	R/W	
2	POD2	0	R/W	
1	POD1	0	R/W	
0	POD0	0	R/W	

1	POD25	0	R/W
0	POD24	0	R/W

- PODRL_1

Bit	Bit Name	Initial Value	R/W	Description
7	POD23	0	R/W	Output Data Register 23 to 16
6	POD22	0	R/W	For bits which have been set to pulse output by NDERL_1, the output trigger transfers NDRL_1 this register during PPG operation. While NDERL_1 is set to 1, the CPU cannot write to this register. While NDERL_1 is cleared, the initial output value of this register can be set.
5	POD21	0	R/W	
4	POD20	0	R/W	
3	POD19	0	R/W	
2	POD18	0	R/W	
1	POD17	0	R/W	
0	POD16	0	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	

- **NDRH**

If pulse output groups 2 and 3 have the same output trigger, all eight bits are mapped same address and can be accessed at one time, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 15 to 8
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3	NDR11	0	R/W	
2	NDR10	0	R/W	
1	NDR9	0	R/W	
0	NDR8	0	R/W	

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 trigger with PCR.
1	NDR9	0	R/W	
0	NDR8	0	R/W	

3	NDR3	0	R/W
2	NDR2	0	R/W
1	NDR1	0	R/W
0	NDR0	0	R/W

If pulse output groups 0 and 1 have different output triggers, the upper four bits and lower four 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 modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
3	NDR3	0	R/W	Next Data Register 3 to 0
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	

3	NDR27	0	R/W
2	NDR26	0	R/W
1	NDR25	0	R/W
0	NDR24	0	R/W

If pulse output groups 6 and 7 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	NDR31	0	R/W	Next Data Register 31 to 28
6	NDR30	0	R/W	The register contents are transferred to the corresponding PODRH_1 bits by the output trigger specified with PCR_1.
5	NDR29	0	R/W	
4	NDR28	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	NDR27	0	R/W	Next Data Register 27 to 24
2	NDR26	0	R/W	The register contents are transferred to the corresponding PODRH_1 bits by the output trigger specified with PCR_1.
1	NDR25	0	R/W	
0	NDR24	0	R/W	

3	NDR19	0	R/W
2	NDR18	0	R/W
1	NDR17	0	R/W
0	NDR16	0	R/W

If pulse output groups 4 and 5 have different output triggers, the upper four bits and lower four bits are mapped to different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR23	0	R/W	Next Data Register 23 to 20
6	NDR22	0	R/W	The register contents are transferred to the corresponding PODRL_1 bits by the output trigger specified with PCR_1.
5	NDR21	0	R/W	
4	NDR20	0	R/W	
3 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
3	NDR19	0	R/W	Next Data Register 19 to 16
2	NDR18	0	R/W	The register contents are transferred to the corresponding PODRL_1 bits by the output trigger specified with PCR_1.
1	NDR17	0	R/W	
0	NDR16	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

4	G2CMS0	1	R/W	These bits select output trigger of pulse output g 00: Compare match in TPU channel 6 01: Compare match in TPU channel 7 10: Compare match in TPU channel 8 11: Compare match in TPU channel 9
3	G1CMS1	1	R/W	Group 5 Compare Match Select 1 and 0
2	G1CMS0	1	R/W	These bits select output trigger of pulse output g 00: Compare match in TPU channel 6 01: Compare match in TPU channel 7 10: Compare match in TPU channel 8 11: Compare match in TPU channel 9
1	G0CMS1	1	R/W	Group 4 Compare Match Select 1 and 0
0	G0CMS0	1	R/W	These bits select output trigger of pulse output g 00: Compare match in TPU channel 6 01: Compare match in TPU channel 7 10: Compare match in TPU channel 8 11: Compare match in TPU channel 9

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 for output group 2.
				0: Normal operation (output values updated at compare match A in the selected TPU channel)
				1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)
1	G1NOV	0	R/W	Group 1 Non-Overlap Selects normal or non-overlapping operation for output group 1. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)
0	G0NOV	0	R/W	Group 0 Non-Overlap Selects normal or non-overlapping operation for output group 0. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)

				Selects direct output or inverted output for pulse group 6. 0: Inverted output 1: Direct output
5	G1INV	1	R/W	Group 5 Inversion Selects direct output or inverted output for pulse group 5. 0: Inverted output 1: Direct output
4	G0INV	1	R/W	Group 4 Inversion Selects direct output or inverted output for pulse group 4. 0: Inverted output 1: Direct output
3	G3NOV	0	R/W	Group 7 Non-Overlap Selects normal or non-overlapping operation for output group 7. 0: Normal operation (output values updated by match A on the selected TPU channel) 1: Non-overlapping operation (output values update compare match A or B on the selected TPU channel)

Selects normal or non-overlapping operation for output group 5.

0: Normal operation (output values updated by compare match A on the selected TPU channel)

1: Non-overlapping operation (output values updated by compare match A or B on the selected TPU channel)

0	G0NOV	0	R/W	Group 4 Non-Overlap
---	-------	---	-----	---------------------

Selects normal or non-overlapping operation for output group 4.

0: Normal operation (output values updated by compare match A on the selected TPU channel)

1: Non-overlapping operation (output values updated by compare match A or B on the selected TPU channel)

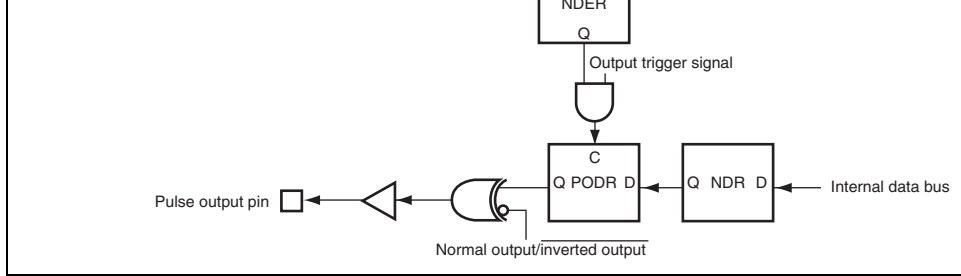


Figure 15.3 Schematic Diagram of PPG

15.4.1 Output Timing

If pulse output is enabled, the NDR contents are transferred to PODR and output when the specified compare match event occurs. Figure 15.4 shows the timing of these operations in the case of normal output in groups 2 and 3, triggered by compare match A.

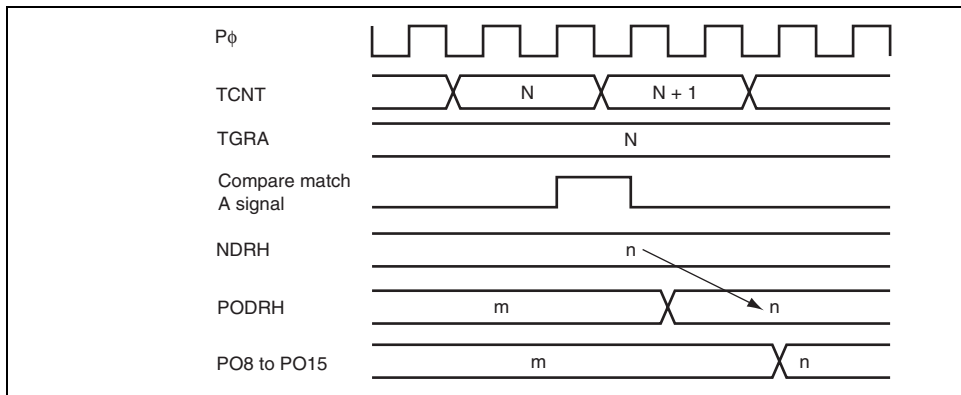


Figure 15.4 Timing of Transfer and Output of NDR Contents (Example)

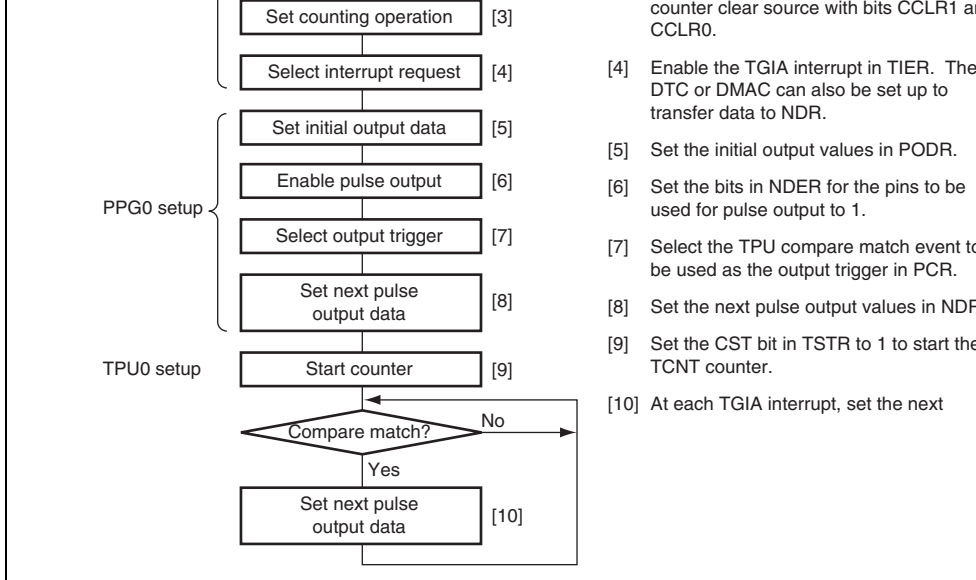


Figure 15.5 Setup Procedure for Normal Pulse Output (PPG0)

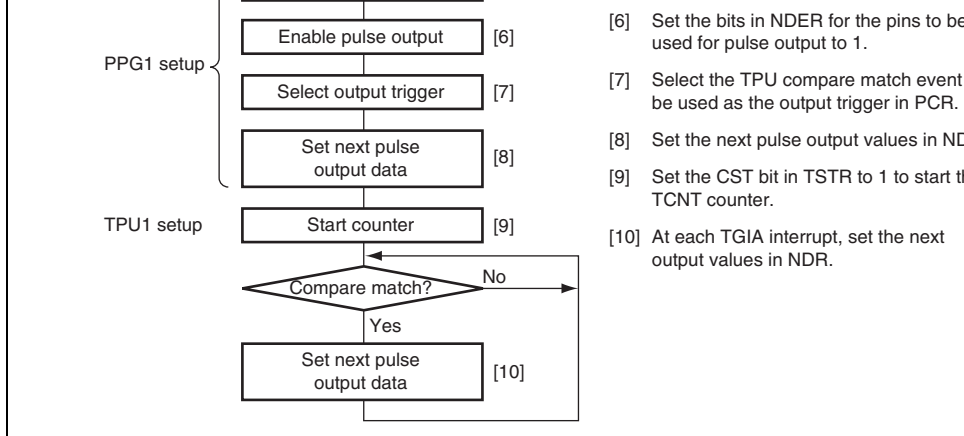


Figure 15.6 Setup Procedure for Normal Pulse Output (PPG1)

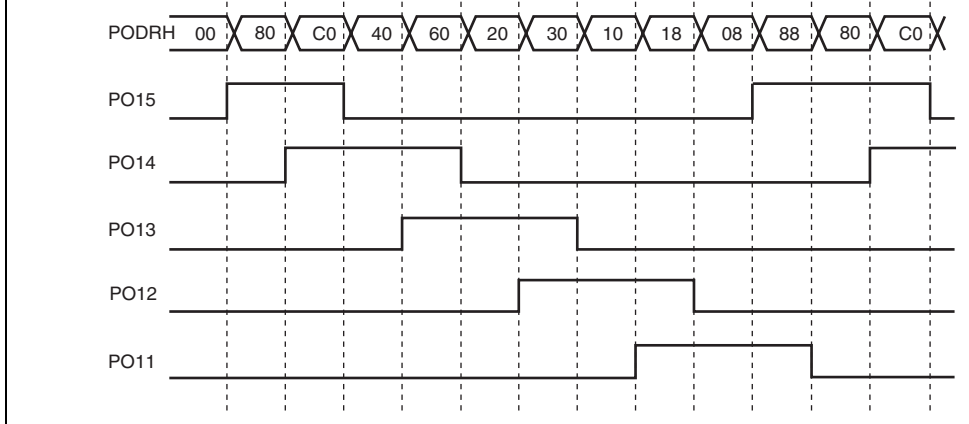


Figure 15.7 Normal Pulse Output Example (5-Phase Pulse Output)

1. Set up TGRA in TPU which is used as the output trigger to be an output compare register. Set the period of a cycle in TGRA so the counter will be cleared by compare match A. Set the TGIEA and TGIER 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 NDERH to select compare match in the TPU channel set up in the previous step to be the output trigger. Write output data H'80 in NDRH.
3. The timer counter in the TPU channel starts. When compare match A occurs, the NDRH contents are transferred to PODRH and output. The TGIA interrupt handling routine writes the next output data (H'C0) in NDRH.
4. 5-phase pulse output (one or two phases active at a time) can be obtained subsequently by writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88... at successive TGIA interrupts. If the DTC or DMAC is set for activation by the TGIA interrupt, pulse output can be obtained without imposing a load on the CPU.

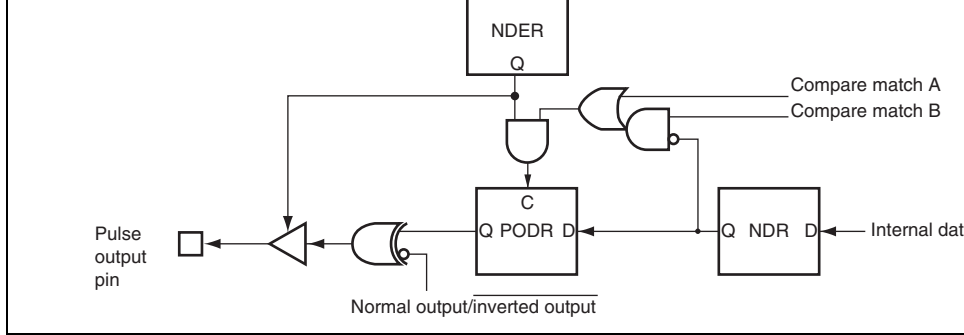


Figure 15.8 Non-Overlapping Pulse Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur before compare match A.

The NDR contents should not be altered during the interval from compare match B to compare match A (the non-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.

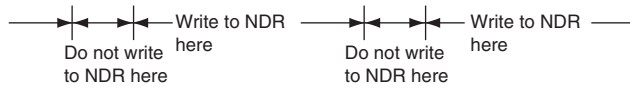


Figure 15.9 Non-Overlapping Operation and NDR Write Timing

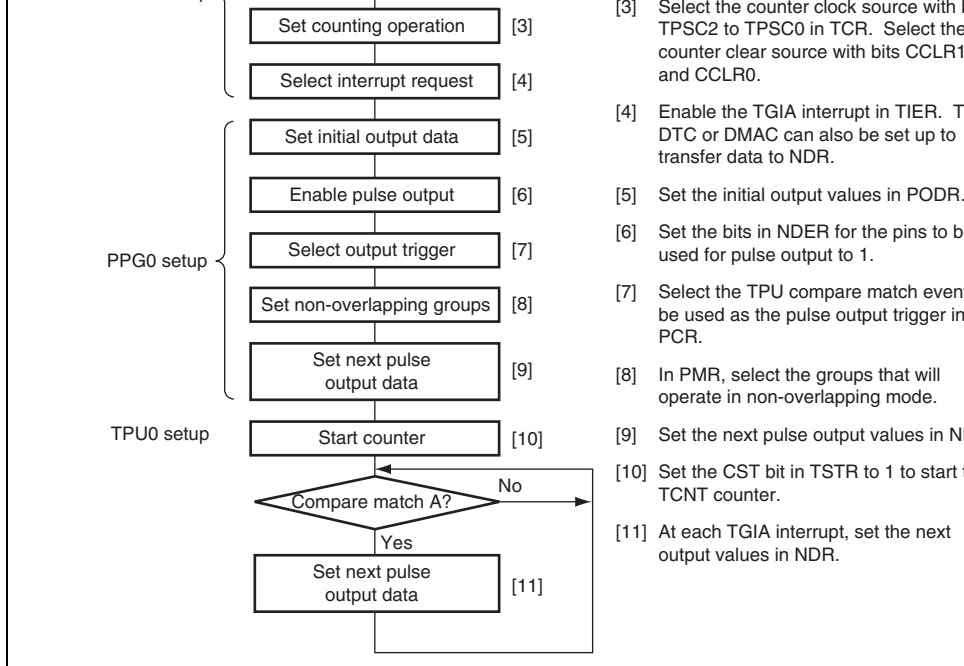


Figure 15.10 Setup Procedure for Non-Overlapping Pulse Output (PPG0)

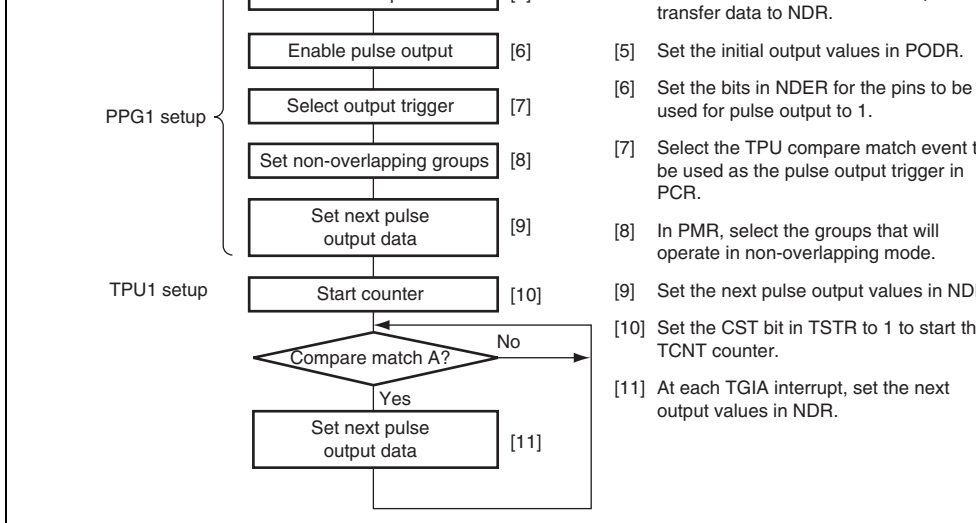


Figure 15.11 Setup Procedure for Non-Overlapping Pulse Output (PPG1)

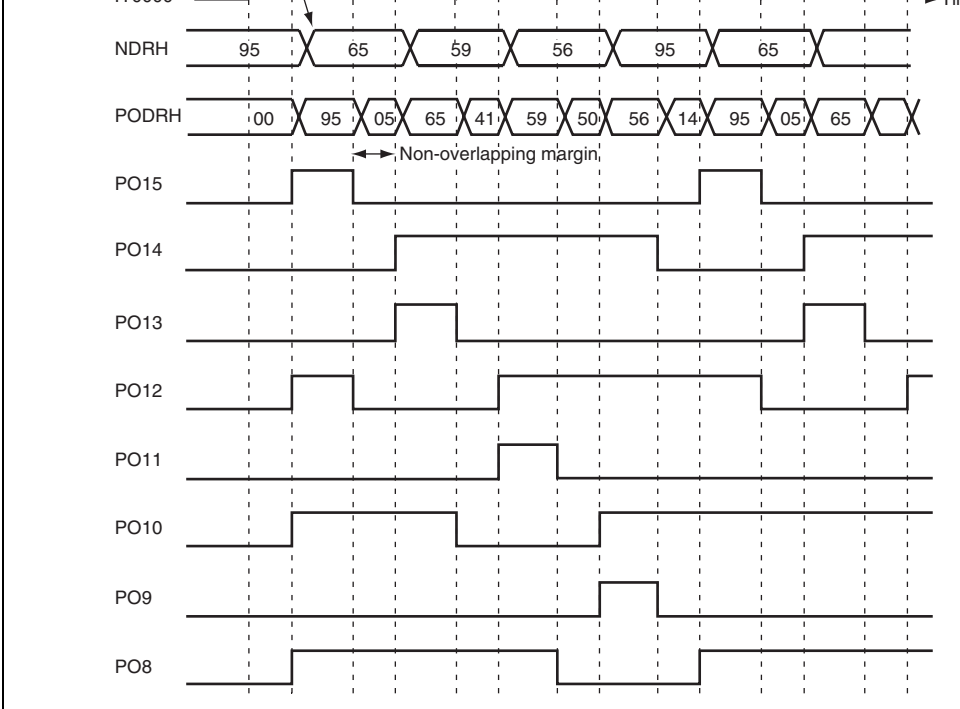


Figure 15.12 Non-Overlapping Pulse Output Example (4-Phase Complement)

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 by 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 without imposing a load on the CPU.

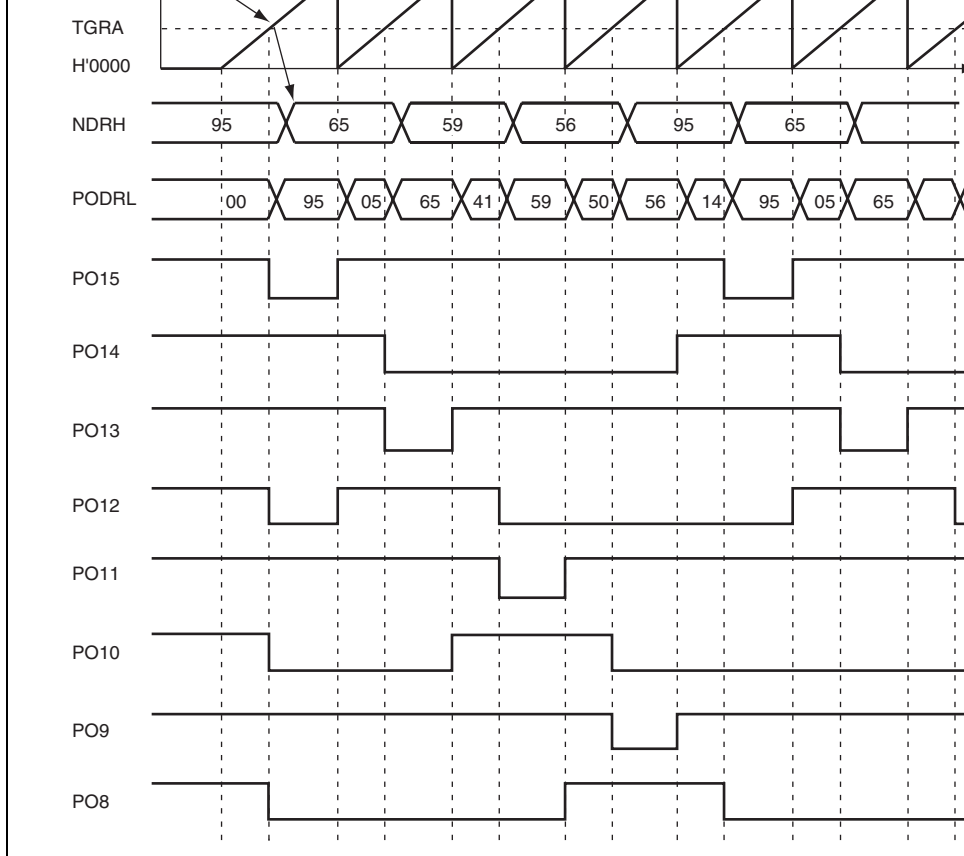


Figure 15.13 Inverted Pulse Output (Example)

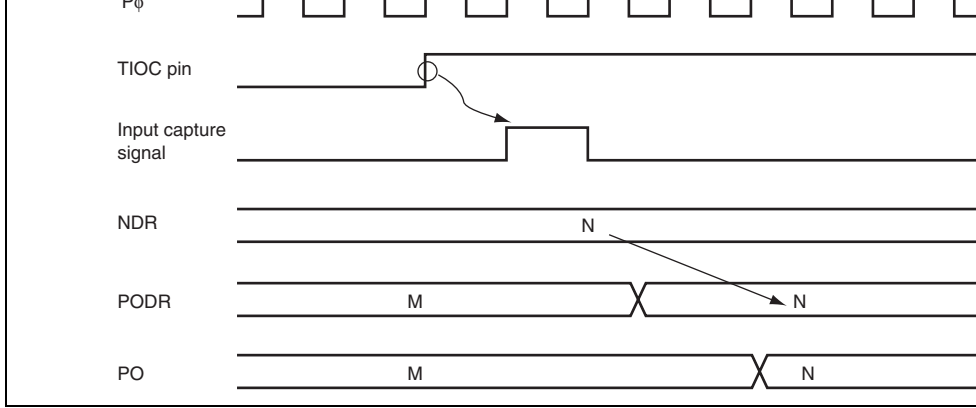


Figure 15.14 Pulse Output Triggered by Input Capture (Example)

Pins PO0 to PO15 are also used for other peripheral functions such as the TPU. When another peripheral function is enabled, the corresponding pins cannot be used for pulse output. Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of the pins.

Pin functions should be changed only under conditions in which the output trigger event occurs.

15.5.3 TPU Setting when PPG1 is in Use

When using PPG1, output toggling on compare-matches must be specified in the TIOR register. The TPU that acts as the activation source and output must be selected as the PPG1 function.

the same functions. Unit 2 and unit 3 can generate baud rate clock for SCI and have the functions.

16.1 Features

- Selection of seven clock sources
The counters can be driven by one of six internal clock signals (P ϕ /2, P ϕ /8, P ϕ /32, P ϕ /64, P ϕ /1024, or P ϕ /8192) or an external clock input (only internal clock available in unit 0 and unit 1).
- 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 independent 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_1 for the 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 independently (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 1.)
- Capable of generating baud rate clock for SCI_5 and SCI_6. (This is available only in unit 2 and unit 3). For details, see section 19, Serial Communication Interface (SCI, IrDA, IrDA).
- Module stop state specifiable

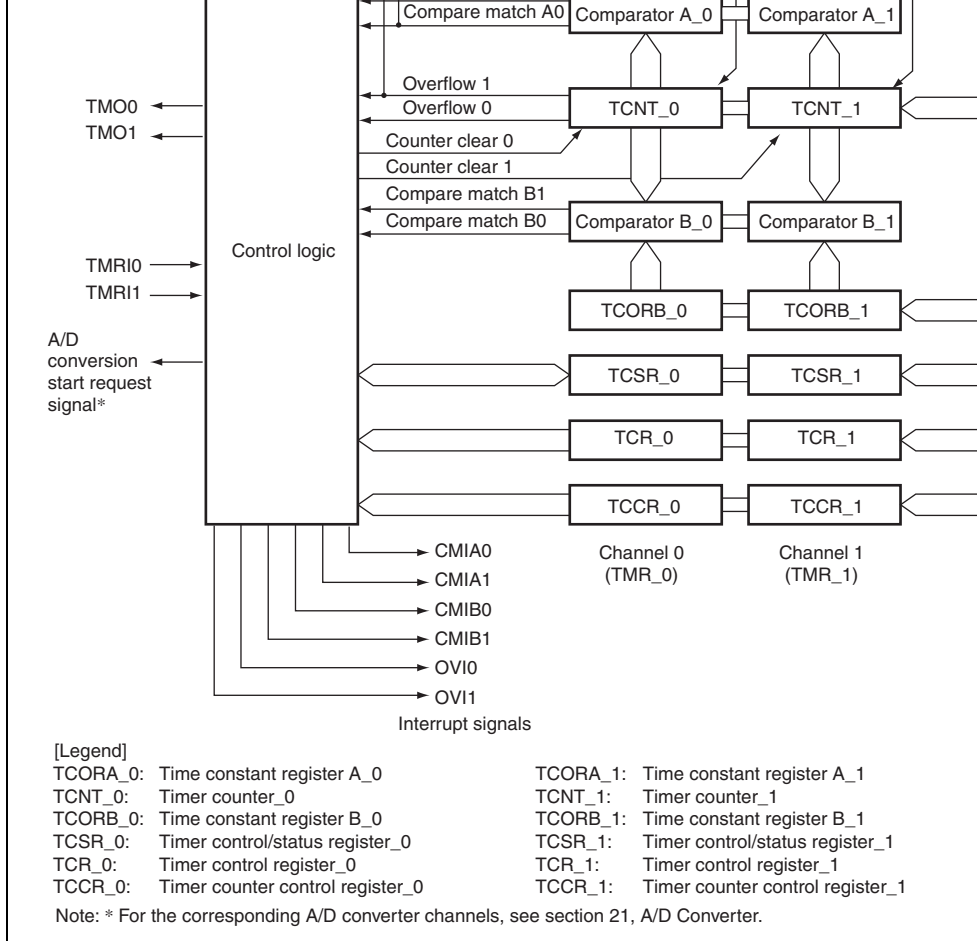


Figure 16.1 Block Diagram of 8-Bit Timer Module (Unit 0)

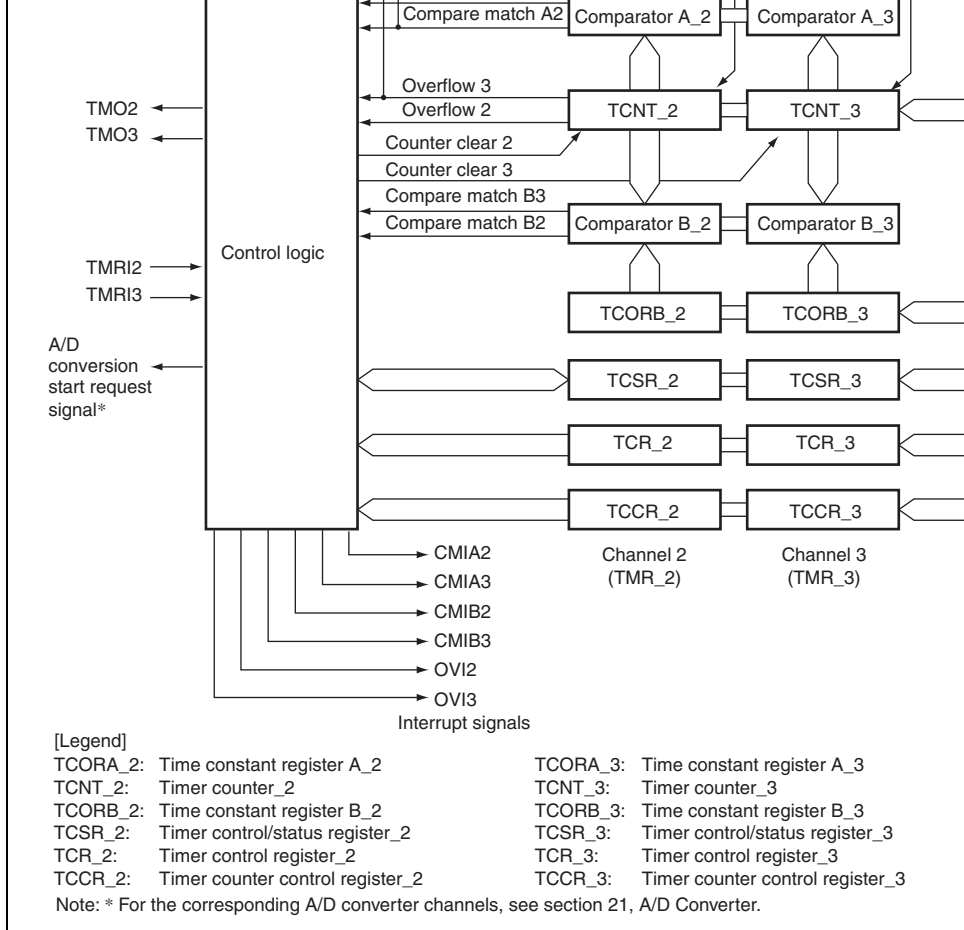


Figure 16.2 Block Diagram of 8-Bit Timer Module (Unit 1)

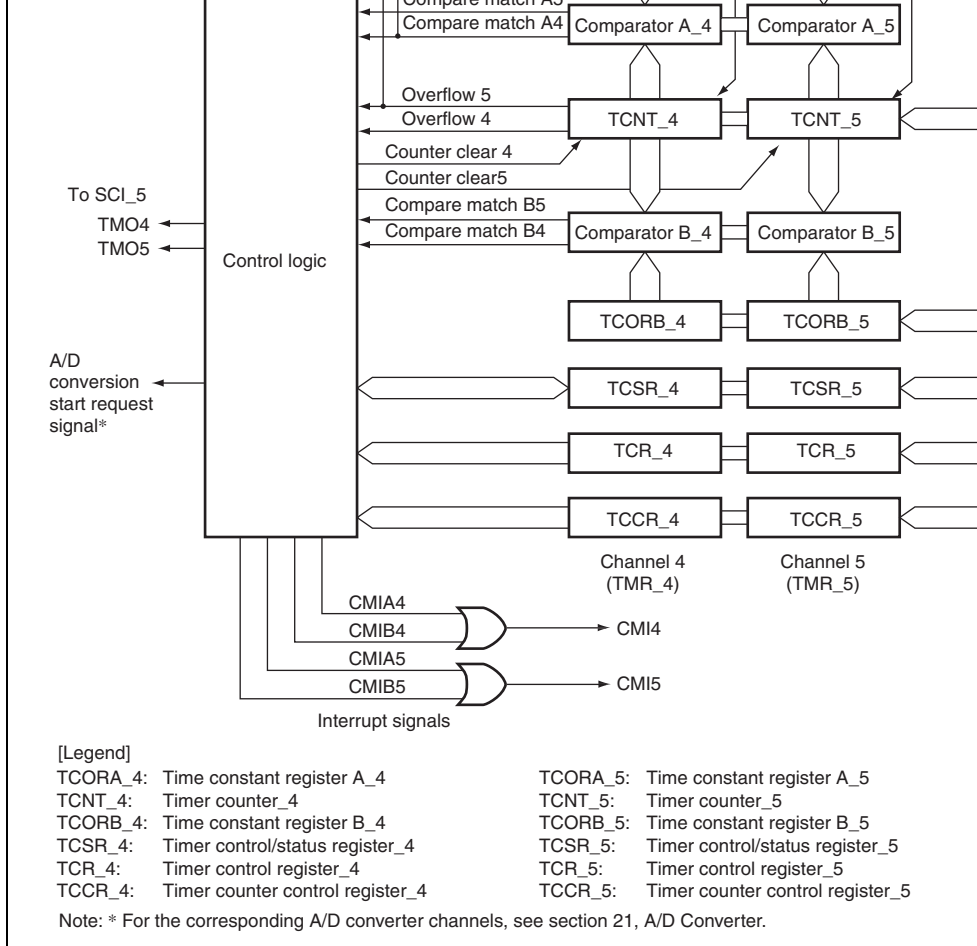


Figure 16.3 Block Diagram of 8-Bit Timer Module (Unit 2)

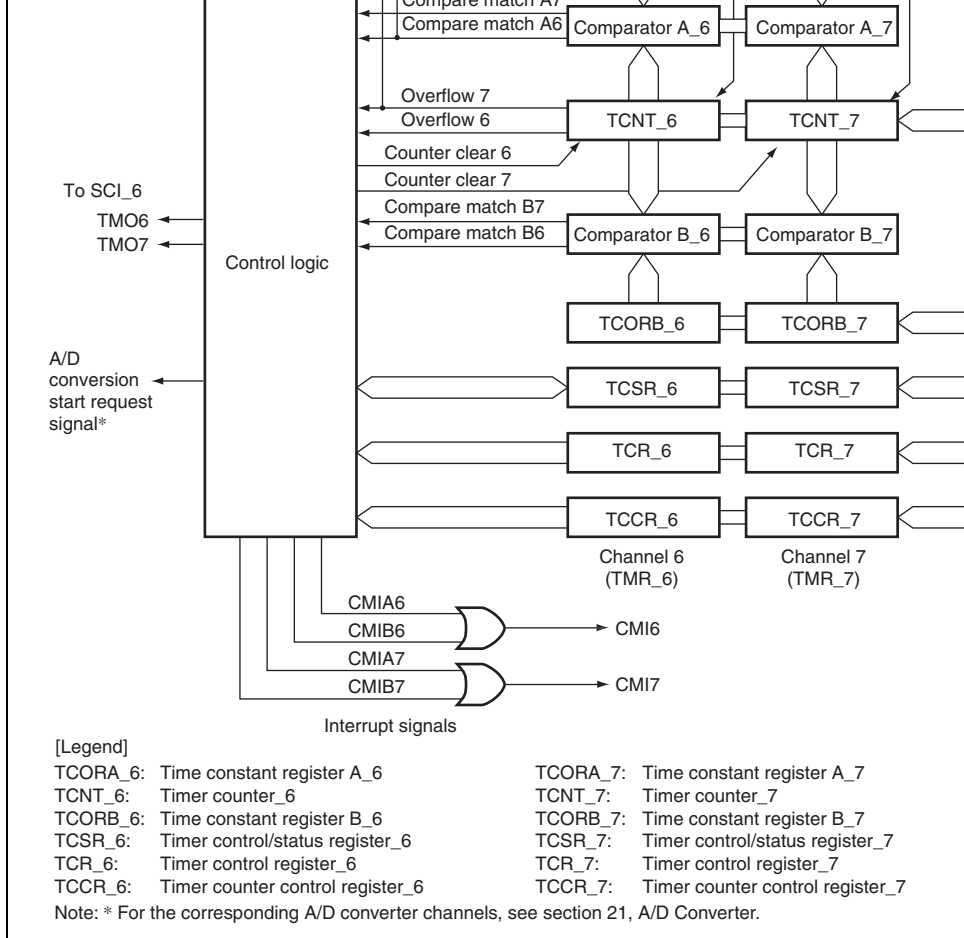


Figure 16.4 Block Diagram of 8-Bit Timer Module (Unit 3)

	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	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	R/W	R/W

16.3.2 Time Constant Register A (TCORA)

TCORA is an 8-bit readable/writable register. TCORA_0 and TCORA_1 comprise a single register so they can be accessed together by a word transfer instruction. The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding CMFA flag in TCSR is set to 1. Note however that comparison is disabled during the TCORA write cycle. The timer output from the TMO pin can be freely controlled by this match signal (compare match A) and the settings of bits OS1 and OS0 in TCSR. TCORA is 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	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/W	R/W	R/W

16.3.4 Timer Control Register (TCR)

TCR selects the TCNT clock source and the condition for clearing TCNT, and enables/disables 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) are enabled or disabled when the CMFB flag in TCS is set to 1. *2 0: CMFB interrupt requests (CMIB) are disabled 1: CMFB interrupt requests (CMIB) are enabled

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 cleared. 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 0) of the external reset input or when the external reset input is high (TMRIS in TCCR is set to 1)
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 the clock condition. See table 16.2.
0	CKS0	0	R/W	

- Notes:
1. To use an external reset or external clock, the DDR and ICR bits in the corresponding I/O Port Register should be set to 0 and 1, respectively. For details, see section 13, I/O Port Registers.
 2. In unit 2 and unit 3, one interrupt signal is used for CMIEB or CMIEA. For details, see section 16.7, 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 set
3	TMRIS	0	R/W	Timer Reset Input Select* Selects an external reset input when the CCLR1 and 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 CKS0 select the internal clock. See table 16.2.

Note: * Available only in unit 0 and unit 1. The write value should always be 0 in unit 2 and 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	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	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	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	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 and falling edges* ² .

- Notes: 1. If the clock input of channel 0 is the TCNT_1 overflow signal and that of channel 1 is the TCNT_0 compare match signal, no incrementing clock is generated. Do not use the ICR bit setting.
2. To use the external clock, the DDR and ICR bits in the corresponding pin should be set to 0 and 1, respectively. For details, see section 13, I/O Ports.

			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	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
	1	0	0	—	Counts at TCNT_3 overflow signal* ¹ .
TMR_3	0	0	0	—	Clock input prohibited
	0	0	1	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	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	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
	1	0	0	—	Counts at TCNT_2 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 and falling edges* ² .

Notes: 1. If the clock input of channel 2 is the TCNT_3 overflow signal and that of channel 1 is the TCNT_2 compare match signal, no incrementing clock is generated. Do not use the ICR setting.

2. To use the external clock, the DDR and ICR bits in the corresponding pin should be set to 0 and 1, respectively. For details, see section 13, I/O Ports.

			1	0	Uses internal clock. Counts at rising edge of
			1	1	Uses internal clock. Counts at falling edge of
	0	1	1	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 rising edge of
			1	1	Uses internal clock. Counts at falling edge of
	1	0	0	—	Counts at TCNT_5 overflow signal*.
TMR_5	0	0	0	—	Clock input prohibited
	0	0	1	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	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	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 rising edge of
			1	1	Uses internal clock. Counts at falling edge of
	1	0	0	—	Counts at TCNT_4 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_5 overflow signal and that of channel 5 is the TCNT_4 compare match signal, no incrementing clock is generated. Do not use the setting.

			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_7 overflow signal*.
TMR_7	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_6 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 6 is the TCNT_7 overflow signal and that of channel 7 is the TCNT_6 compare match signal, no incrementing clock is generated. Do not use this 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] <ul style="list-style-type: none"> • When TCNT matches TCORB [Clearing conditions] <ul style="list-style-type: none"> • When writing 0 after reading CMFB = 1 (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 the DTC is activated by a CMIB interrupt, the DISEL bit in MRB of the DTC is 0*³

- When the DTC is activated by a CMIA interrupt, the DISEL bit in MRB in the DTC is 0*³

5	OVF	0	R/(W)* ¹	<p>Timer Overflow Flag</p> <p>[Setting condition]</p> <p>When TCNT overflows from H'FF to H'00</p> <p>[Clearing condition]</p> <p>When writing 0 after reading OVF = 1</p> <p>(When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)</p>
4	ADTE	0	R/W	<p>A/D Trigger Enable*³</p> <p>Selects enabling or disabling of A/D converter start requests by compare match A.</p> <p>0: A/D converter start requests by compare match A disabled</p> <p>1: A/D converter start requests by compare match A enabled</p>
3	OS3	0	R/W	Output Select 3 and 2* ²
2	OS2	0	R/W	<p>These bits select a method of TMO pin output when compare match B of TCORB and TCNT occurs.</p> <p>00: No change when compare match B occurs</p> <p>01: 0 is output when compare match B occurs</p> <p>10: 1 is output when compare match B occurs</p> <p>11: Output is inverted when compare match B occurs (toggle output)</p>

- 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 until a compare match occurs after a reset.
 3. For the corresponding A/D converter channels, see section 21, A/D Converter.

- TCSR_1

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W) ^{*1}	Compare Match Flag B [Setting condition] <ul style="list-style-type: none"> • When TCNT matches TCORB [Clearing conditions] <ul style="list-style-type: none"> • When writing 0 after reading CMFB = 1 (When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled to read the flag after writing 0 to it.) • When the DTC is activated by a CMIB interrupt and the DISEL bit in MRB of the DTC is 0^{*3}

- When the DTC is activated by a CMIA interrupt, the DISEL bit in MRB of the DTC is 0*³

5	OVF	0	R/(W)* ¹	<p>Timer Overflow Flag</p> <p>[Setting condition]</p> <p>When TCNT overflows from H'FF to H'00</p> <p>[Clearing condition]</p> <p>Cleared by reading OVF when OVF = 1, then writing 0 to OVF</p> <p>(When the CPU is used to clear this flag by writing 0 to OVF while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)</p>
4	—	1	R	<p>Reserved</p> <p>This bit is always read as 1 and cannot be modified.</p>
3	OS3	0	R/W	Output Select 3 and 2* ²
2	OS2	0	R/W	<p>These bits select a method of TMO pin output when a compare match B of TCORB and TCNT occurs.</p> <p>00: No change when compare match B occurs</p> <p>01: 0 is output when compare match B occurs</p> <p>10: 1 is output when compare match B occurs</p> <p>11: Output is inverted when compare match B occurs (toggle output)</p>

- 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 until a compare match occurs after a reset.
 3. Available only in unit 0 and unit 1.

16.4 Operation

16.4.1 Pulse Output

Figure 16.5 shows an example of the 8-bit timer being used to generate a pulse output with a 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 is cleared 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 TCORA 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 TCORA and a pulse width determined by TCORB. No software intervention is required. The timer output is 0 until the first compare match occurs after a reset.

16.4.2 Reset Input

Figure 16.6 shows an example of the 8-bit timer being used to generate a pulse which is output 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 that 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 TCORA 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 TMRI input determined by TCORA and with a pulse width determined by TCORB and TCORA.

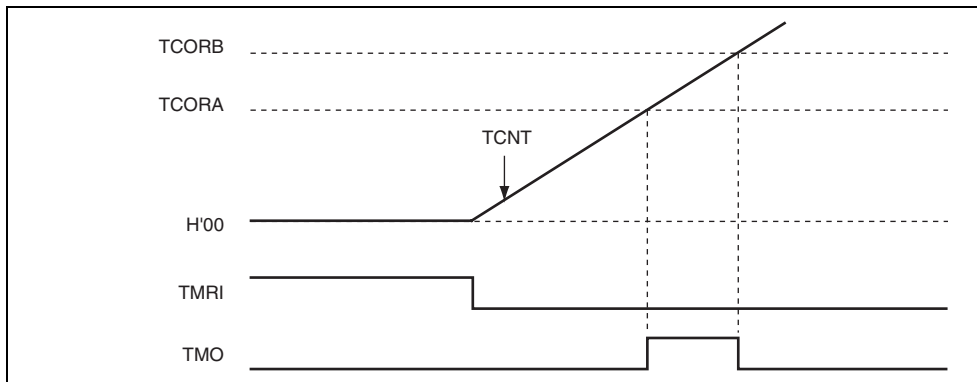


Figure 16.6 Example of Reset Input

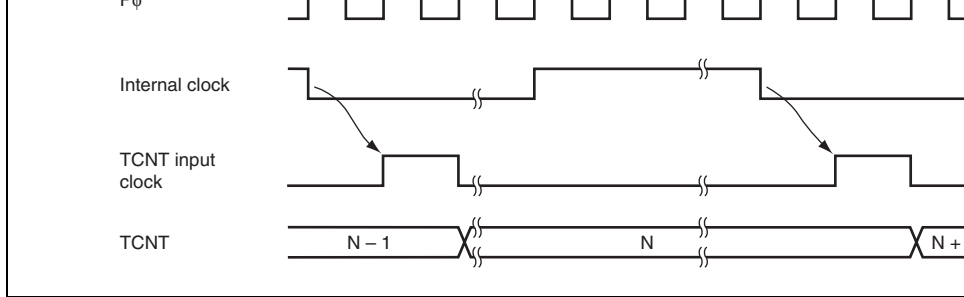


Figure 16.7 Count Timing for Internal Clock Input

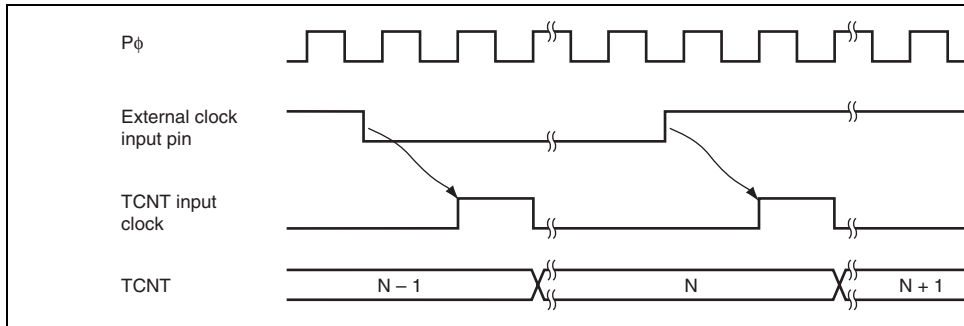


Figure 16.8 Count Timing for External Clock Input

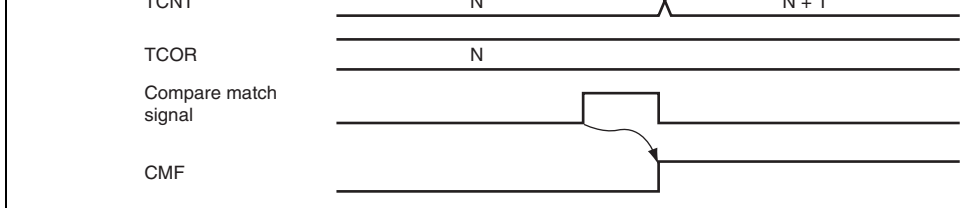


Figure 16.9 Timing of CMF Setting at Compare Match

16.5.3 Timing of Timer Output at Compare Match

When a compare match signal is generated, the timer output changes as specified by the bit OS0 in TCSR. Figure 16.10 shows the timing when the timer output is toggled by the compare match A signal.

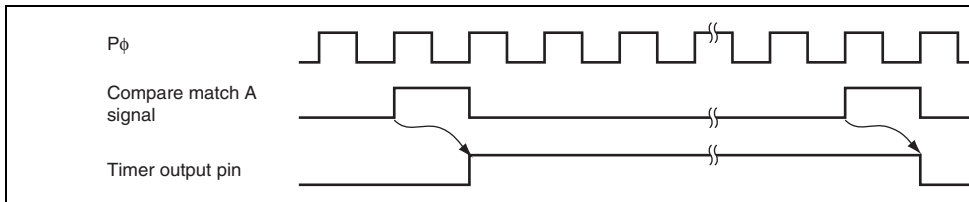


Figure 16.10 Timing of Toggled Timer Output at Compare Match A

16.5.5 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
16.12 and Figure 16.13 shows the timing of this operation.

Note: * Clearing by an external reset is available only in units 0 and 1.

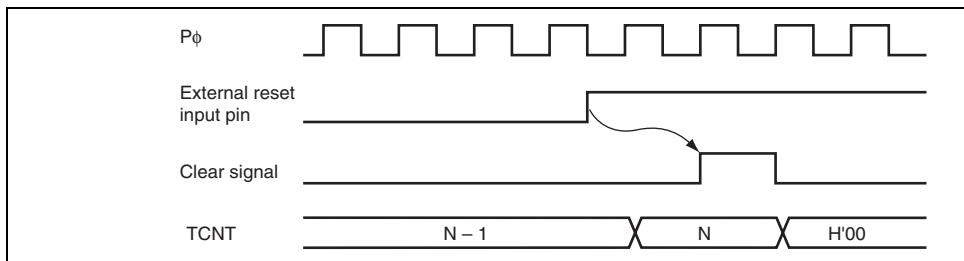


Figure 16.12 Timing of Clearance by External Reset (Rising Edge)

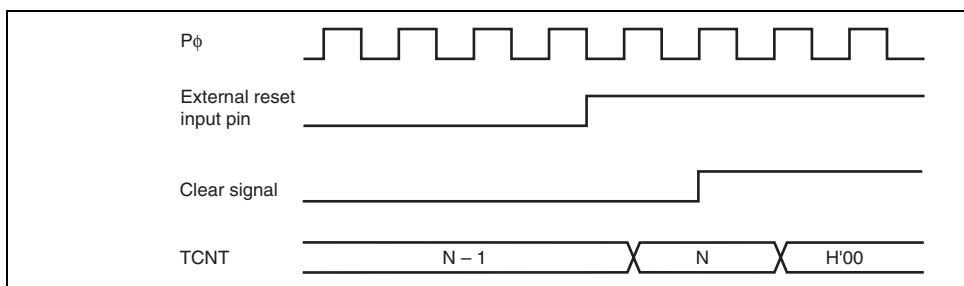


Figure 16.13 Timing of Clearance by External Reset (High Level)

Figure 16.14 Timing of OVF Setting

16.6 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-bit mode) or compare matches of the 8-bit channel 0 could be counted by the timer of channel 1 (compare match count mode).

16.6.1 16-Bit Counter Mode

When the bits CKS2 to CKS0 in TCR_0 are set to B'100, the timer functions as a single 16-bit timer with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bits.

(1) Setting of Compare Match Flags

- The CMF flag in TCSR_0 is set to 1 when a 16-bit compare match event occurs.
- The CMF flag in TCSR_1 is set to 1 when a lower 8-bit compare match event occurs.

(2) Counter Clear Specification

- If the CCLR1 and CCLR0 bits in TCR_0 have been set for counter clear at compare match, the 16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit compare match occurs. The 16-bit counter (TCNT0 and TCNT1 together) is cleared even if counter clear is set by the TMRI0 pin has been set.
- The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bits counter is cleared independently.

flag, generation of interrupts, output from the TMO pin, and counter clear are in accordance with the settings for each channel.

16.7 Interrupt Sources

16.7.1 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 and OVI. Their interrupt sources and priorities are shown in table 16.6. Each interrupt source is enabled or disabled by the corresponding interrupt enable bit in TCR or TCSR, and independent interrupt requests are sent for each to the interrupt controller. It is also possible to activate the DTC by the means of CMIA and CMIB interrupts (This is available in unit 0 and unit 1 only).

Table 16.6 8-Bit Timer (TMR_0 or TMR_1) Interrupt Sources (in Unit 0 and Unit 1)

Signal Name	Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority
CMIA0	CMIA0	TCORA_0 compare match	CMFA	Possible	High
CMIB0	CMIB0	TCORB_0 compare match	CMFB	Possible	Low
OVI0	OVI0	TCNT_0 overflow	OVF	Not possible	Low
CMIA1	CMIA1	TCORA_1 compare match	CMFA	Possible	High
CMIB1	CMIB1	TCORB_1 compare match	CMFB	Possible	Low
OVI1	OVI1	TCNT_1 overflow	OVF	Not possible	Low

Name	Name	Interrupt Source	Flag	Activation	Priority
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		

16.7.2 A/D Converter Activation

The A/D converter can be activated by a compare match A for the even channels of each unit. *

If the ADTE bit in TCSR is set to 1 when the CMFA flag in TCSR is set to 1 by the occurrence of a compare match A, a request to start A/D conversion is sent to the A/D converter. If the timer conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

Note: * For the corresponding A/D converter channels, see section 21, A/D Converter

f: Counter frequency
 ϕ : Operating frequency
N: TCOR value

16.8.2 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 has priority and the write is not performed as shown in figure 16.15.

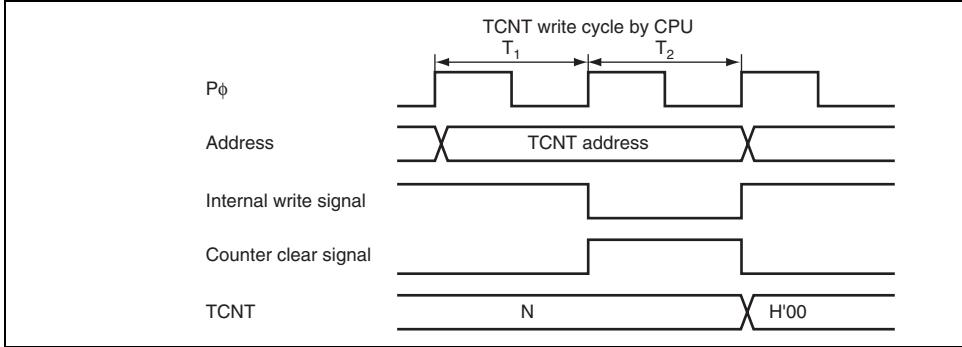


Figure 16.15 Conflict between TCNT Write and Clear

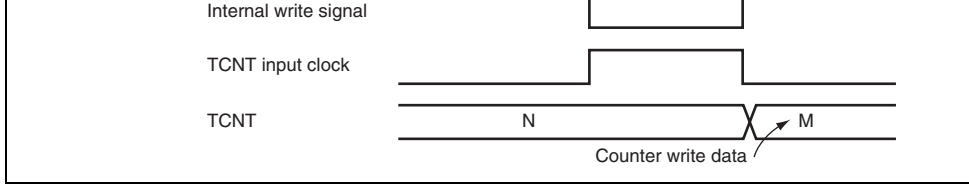


Figure 16.16 Conflict between TCNT Write and Increment

16.8.4 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 write has priority and the compare match signal is inhibited as shown in figure 16.17.

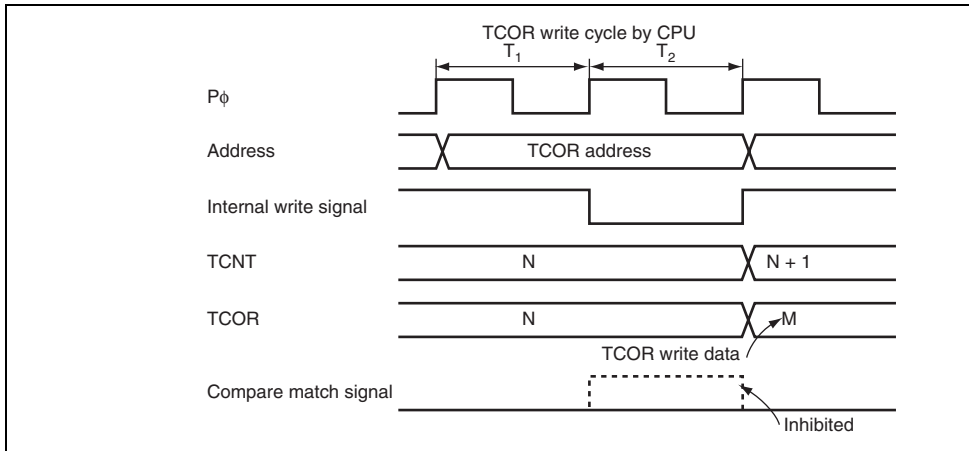


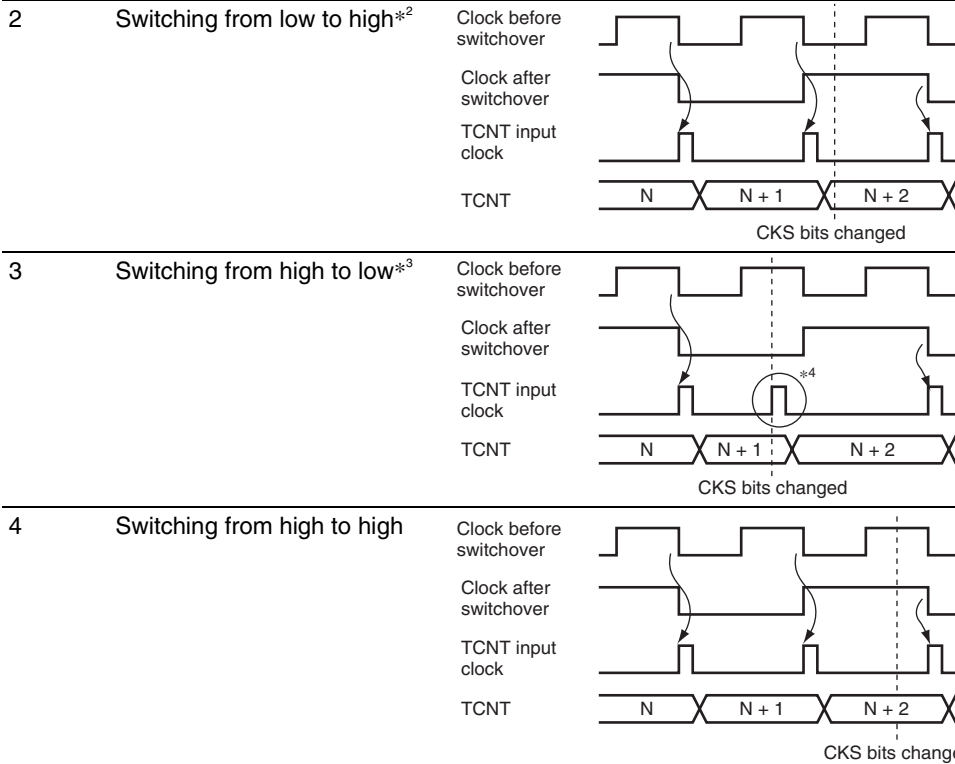
Figure 16.17 Conflict between TCOR Write and Compare Match

16.8.6 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the internal clock is switched. Table 16.9 shows the relationship between the timing at which the internal clock is switched (related 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 the clock pulse are always monitored. Table 16.9 assumes that the falling edge is selected. If the signal levels of the clocks before and after switching change from high to low as shown in the figure, the change is considered as the falling edge. Therefore, a TCNT clock pulse is generated and 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 falling edges of the internal clock, and when switching between internal and external clocks.



- 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 edge. TCNT is incremented.

module stop state. For details, see section 27, Power-Down Modes.

16.8.9 Interrupts in Module Stop State

If the module stop state is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC activation source. Interrupts should therefore be disabled before entering the module stop state.

approx. 45.5 days settable.

- Counter operational except in hardware standby mode or the reset state.
- Canceling software standby mode and deep software standby mode is possible.

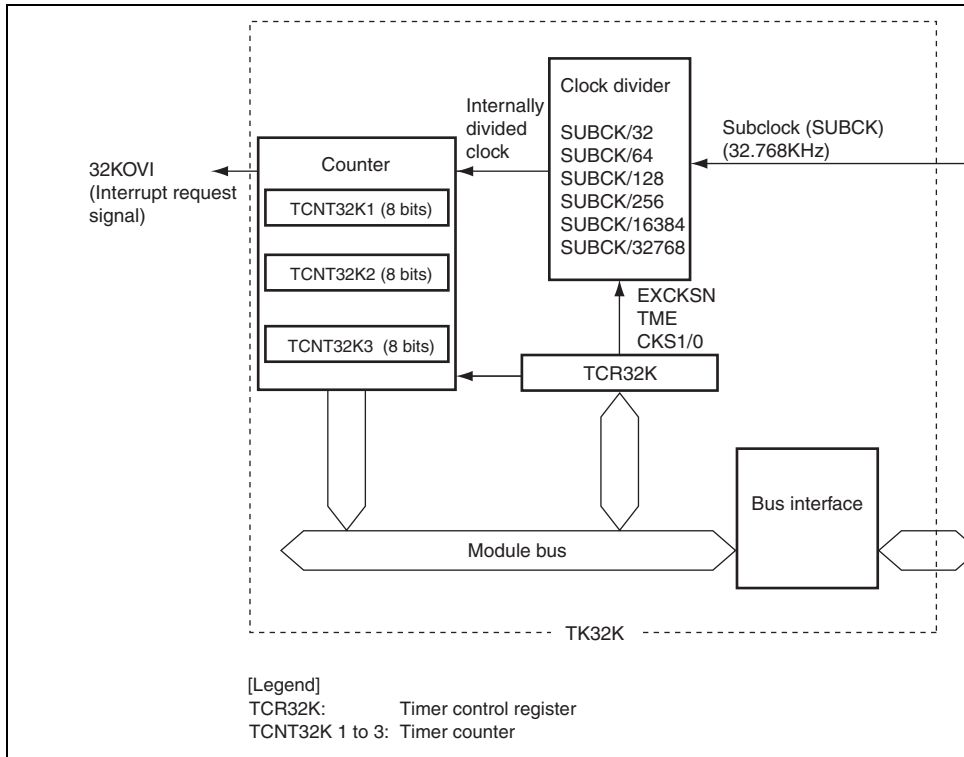


Figure 17.1 Block Diagram of TM32K

to TCNT32K. Change values of the bits when TME=0.

Bit	7	6	5	4	3	2	1
Bit Name	EXCKSN	—	TME	—	—	OSC32STP	CKS1
Initial Value	1	1	0	1	1	0	0
R/W	R/W	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	EXCKSN	1	R/W	Extended Clock Select and Clock Select 1, 0
1	CKS1	0	R/W	These bits select the internally divided clock source input to TCNT32K. The interrupt cycle for SUBCK 32.768 kHz is indicated in parentheses. When EXCKSN = 1: 00: Clock SUBCK/32 (cycle: 250 ms) 01: Clock SUBCK/64 (cycle: 500 ms) 10: Clock SUBCK/128 (cycle: 1 s) 11: Clock SUBCK/256 (cycle: 2 s) When EXCKSN = 0: 00: Clock SUBCK/16384 (cycle: 30 s) 01: Clock SUBCK/32768 (cycle: 60 s) 10: Clock SUBCK/16384 (cycle: approx. 22.7 da 11: Clock SUBCK/32768 (cycle: approx. 45.5 da
0	CKS0	0	R/W	
6	—	1	R	

17.2.2 Timer Counter (TCNT32K1, TCNT32K2, TCNT32K3)

The timer counter is a 24-bit counter for which either 8- or 24-bit operation is selectable. Allocation to the registers of the timer counter differs according to the setting of the EX in TCR32K.

(1) EXCKSN = 1 (Initial State)

The timer counter operates as an 8-bit counter. Only TCNT32K1 is used and TCNT32K2, TCNT32K3 become reserved registers. Clearing the TME bit in the timer control register (TCR32K) initializes TCNT32K1 to H'00.

- TCNT32K1

Bit	7	6	5	4	3	2	1
Bit Name	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1
Initial value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

- TCNT32K2

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

- TCNT32K1

Bit	7	6	5	4	3	2	1
Bit Name	TCNT23	TCNT22	TCNT21	TCNT20	TCNT19	TCNT18	TCNT17
Initial value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

- TCNT32K2

Bit	7	6	5	4	3	2	1
Bit Name	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9
Initial value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

- TCNT32K3

Bit	7	6	5	4	3	2	1
Bit Name	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1
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 subclock oscillator operation (OSC32STP = 1).

Table 17.1 Relationships between Counter Operations and 32KOV1 Cycles

Setting			Division Ratio	Selected Counter			Interrupt Cycles	
EXCKSN	CKS1	CKS2	Internally Divided CLK	TCNT 32K1	TCNT 32K2	TCNT 32K3	Counter value when interrupt generates	32KOV1
1	0	0	SUBCK/32	√	–	–	TCNT32K1=H'FF	250 m
1	0	1	SUBCK/64	√	–	–		500 m
1	1	0	SUBCK/128	√	–	–		1 s
1	1	1	SUBCK/256	√	–	–		2 s
0	0	0	SUBCK/16384	√	√	√	TCNT32K3=H'3B	30 s
0	0	1	SUBCK/32768	√	√	√		60 s
0	1	0	SUBCK/16384	√	√	√	TCNT32K1 to 3 =H'FFF3B	Approx. days
0	1	1	SUBCK/32768	√	√	√		Approx. days

√ : in use – : not in use

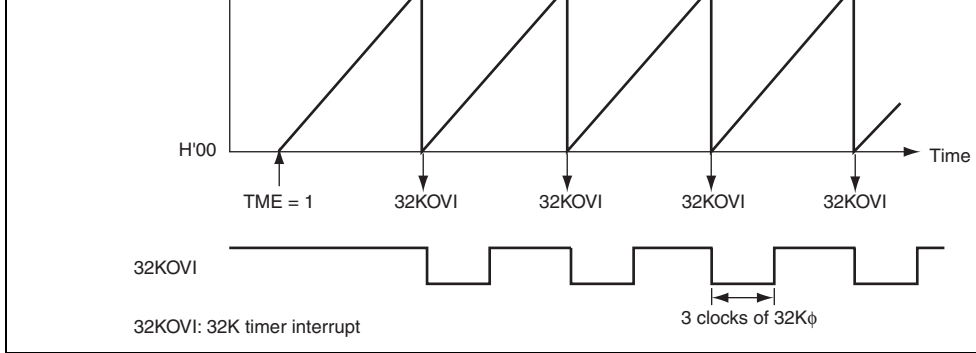


Figure 17.2 EXCKSN = 1 32K Timer Operation

17.3.3 EXCKSN=0 Operation

When the EXCKSN bit in TCR32K is 0, the timer counter operates as a 24-bit counter. The lower-order 8-bits are TCNT32K3, which operates as a free-running up-counter that counts from H'3B. The higher-order 16-bits in TCNT32K1 and TCNT32K2 act as an up-counter that counts from H'00_00 to H'FFFF3B the number of times TCNT32K3=H'3B. Generation of the 32KOV1 interrupt either for all times TCNT32K3=H'3B or when TCNT32K1 to TCNT32K3=H'FFFF3B can be selected by the setting of the CKS1 bit. This setting in combination with the setting of the CKS0 bit can be used to set the interrupt interval to 30s, 60s, approx. 22.7 days, or approx. 45.5 days.

When the EXCKSN bit is 0, the CKS1 bit can be changed while the timer counter is operating because the TME bit is 1. This allows variation of the 32KOV1 interrupt interval while the counter is in operation. When the value of the CKS1 bit is from 0 to 1, the higher-order 16-bit counter (TCNT32K1, TCNT32K2) resumes counting after being initialized to H'00_00.

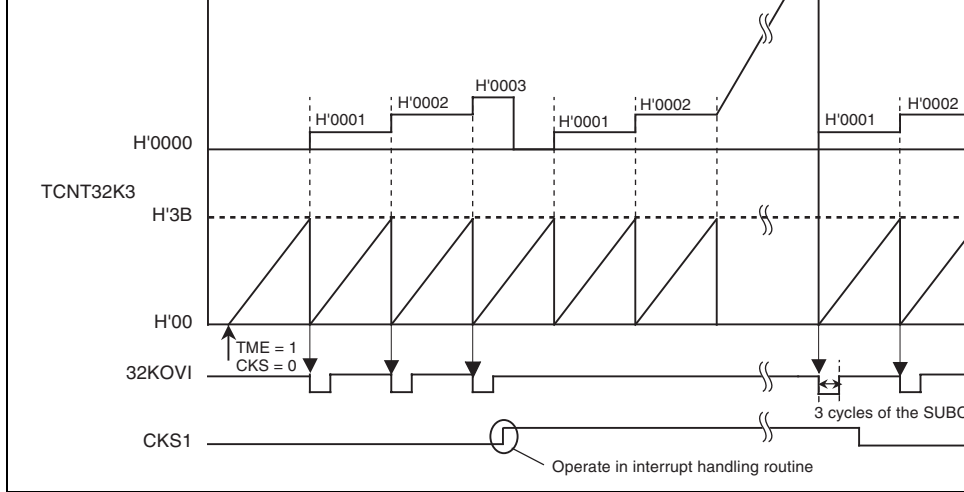


Figure 17.3 Operation of Changing the CKS1 bit when EXCKSN = 0

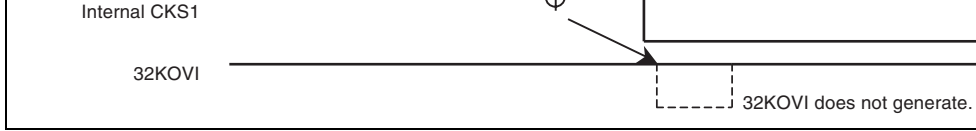


Figure 17.4 Conflict between the CKS1 bit being Changed from 1 to 0 and the 32

Table 17.2 TM32K Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation	Software Standby Mode Reset	Deep Standby
32KOV1	TCNT32K interrupt	IRQ15F	Impossible	Possible	Impossible
		DT32KIF	Impossible	Impossible	Possible

17.5.2 Note on Register Initialization

TCR32K, TCNT32K1, TCNT32K2, and TCNT32K3 of the 32K timer are initialized in h standby mode or in the pin reset state. A reset from the watchdog timer or deep-software-reset does not initialize these registers.

17.5.3 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 subclock oscillator.
- Before stopping the TM32K, clear the TME bit to 0 for one clock (30 μ s) or longer by subclock.
- When switching between subclock and main clock operation, wait for 500 μ s or more (the timer counter is updated next time) before reading the timer counter.

18.1 Features

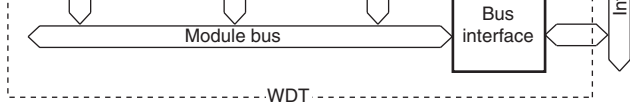
- Selectable from eight counter input clocks
- Switchable between watchdog timer mode and interval timer mode

- In watchdog timer mode

If the counter overflows, the WDT outputs $\overline{\text{WDTOVF}}$. It is possible to select whether or 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 18.1 Block Diagram of WDT

18.2 Input/Output Pin

Table 18.1 shows the WDT pin configuration.

Table 18.1 Pin Configuration

Name	Symbol	I/O	Function
Watchdog timer overflow*	$\overline{\text{WDTOVF}}$	Output	Outputs a counter overflow signal in watchdog timer mode

Note: * In boundary scan valid mode, counter overflow signal output cannot be used.

18.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT is initialized to H'00 when the TMSCSR 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

18.3.2 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/ \bar{T}	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 writing 0 to OVF

(When the CPU is used to clear this flag by writing 0 to OVF while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)

6	WT/IT	0	R/W	<p>Timer Mode Select</p> <p>Selects whether the WDT is used as a watchdog timer or interval timer.</p> <p>0: Interval timer mode When TCNT overflows, an interval timer interrupt (WOVI) is requested.</p> <p>1: Watchdog timer mode When TCNT overflows, the $\overline{\text{WDTOVF}}$ signal is asserted.</p>
5	TME	0	R/W	<p>Timer Enable</p> <p>When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.</p>
4, 3	—	All 1	R	<p>Reserved</p> <p>These are read-only bits and cannot be modified.</p>
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	Select the clock source to be input to TCNT. The clock cycle for $P\phi = 20$ MHz is indicated in parentheses.
0	CKS0	0	R/W	<p>000: Clock $P\phi/2$ (cycle: 25.6 μs)</p> <p>001: Clock $P\phi/64$ (cycle: 819.2 μs)</p> <p>010: Clock $P\phi/128$ (cycle: 1.6 ms)</p> <p>011: Clock $P\phi/512$ (cycle: 6.6 ms)</p> <p>100: Clock $P\phi/2048$ (cycle: 26.2 ms)</p> <p>101: Clock $P\phi/8192$ (cycle: 104.9 ms)</p> <p>110: Clock $P\phi/32768$ (cycle: 419.4 ms)</p> <p>111: Clock $P\phi/131072$ (cycle: 1.68 s)</p>

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)*	<p>Watchdog Timer Overflow Flag</p> <p>This bit is set when TCNT overflows in watchdog timer mode. This bit cannot be set in interval timer mode. Only 0 can be written.</p> <p>[Setting condition]</p> <p>When TCNT overflows (changed from H'FF to H'00) in watchdog timer mode</p> <p>[Clearing condition]</p> <p>Reading RSTCSR when WOVF = 1, and then writing 0 to WOVF</p>
6	RSTE	0	R/W	<p>Reset Enable</p> <p>Specifies whether or not this LSI is internally reset if TCNT overflows during watchdog timer operation.</p> <p>0: LSI is not reset even if TCNT overflows (Though LSI is not reset, TCNT and TCSR in WDT are reset.)</p> <p>1: LSI is reset if TCNT overflows</p>

18.4.1 Watchdog Timer Mode

To use the WDT in watchdog timer mode, set both the $\overline{WT/IT}$ and TME bits in TCSR to 1.

During watchdog timer operation, if TCNT overflows without being rewritten because of a crash or other error, the \overline{WDTOVF} signal is output. This ensures that TCNT does not overflow while the system is operating normally. Software must prevent TCNT overflows by rewriting the TCNT value (normally H'00 is written) before overflow occurs. This \overline{WDTOVF} signal can be used to reset the LSI internally in watchdog timer mode.

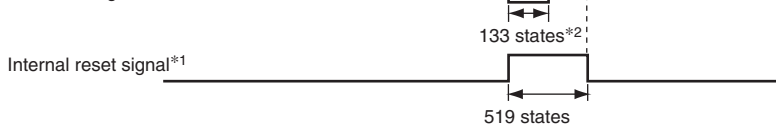
If TCNT overflows when the RSTE bit in RSTCSR is set to 1, a signal that resets this LSI internally is generated at the same time as the \overline{WDTOVF} signal. If a reset caused by a signal to the \overline{RES} pin occurs at the same time as a reset caused by a WDT overflow, the \overline{RES} pin has priority and the WOVF bit in RSTCSR is cleared to 0.

The \overline{WDTOVF} signal is output for 133 cycles of $P\phi$ when RSTE = 1 in RSTCSR, and for 519 cycles of $P\phi$ when RSTE = 0 in RSTCSR. The internal reset signal is output for 519 cycles of $P\phi$.

When RSTE = 1, an internal reset signal is generated. Since the system clock control register (SCKCR) is initialized, the multiplication ratio of $P\phi$ becomes the initial value.

When RSTE = 0, an internal reset signal is not generated. Neither SCKCR nor the multiplication ratio of $P\phi$ is changed.

When TCNT overflows in watchdog timer mode, the WOVF bit in RSTCSR is set to 1. If TCNT overflows when the RSTE bit in RSTCSR is set to 1, an internal reset signal is generated and the entire LSI is reset.



- Notes: 1. If TCNT overflows when the RSTE bit is set to 1, an internal reset is generated.
 2. 130 states when the RSTE bit is cleared to 0.

Figure 18.2 Operation in Watchdog Timer Mode

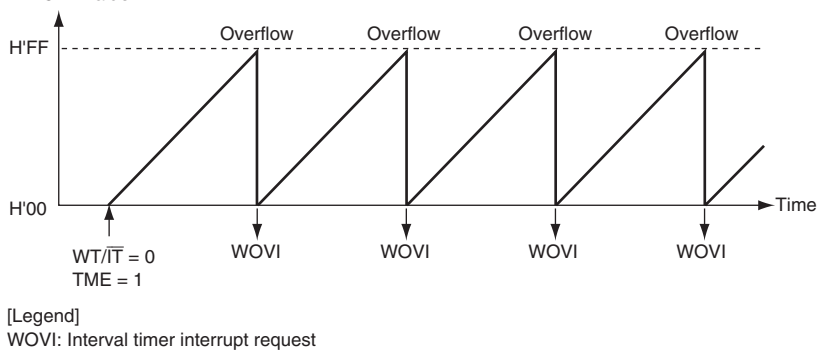


Figure 18.3 Operation in Interval Timer Mode

18.5 Interrupt Source

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. The OVF flag must be cleared to 0 in the interrupt handling routine.

Table 18.2 WDT Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation
WOVI	TCNT overflow	OVF	Impossible

TCNT and TCSR must be written to by a word transfer instruction. They cannot be written to by a byte transfer instruction.

For writing, TCNT and TCSR are assigned to the same address. Accordingly, perform data transfer as shown in figure 18.4. The transfer instruction writes the lower byte data to TCNT and TCSR.

To write to RSTCSR, execute a word transfer instruction for address H'FFA6. A byte transfer 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 WOVF bit in RSTCSR. Perform data transfer as shown in figure 18.4.

At data transfer, the transfer instruction clears the WOVF bit to 0, but has no effect on the RSTE bit. To write to the RSTE bit, perform data transfer as shown in figure 18.4. In this case, the transfer instruction writes the value in bit 6 of the lower byte to the RSTE bit, but has no effect on the WOVF bit.

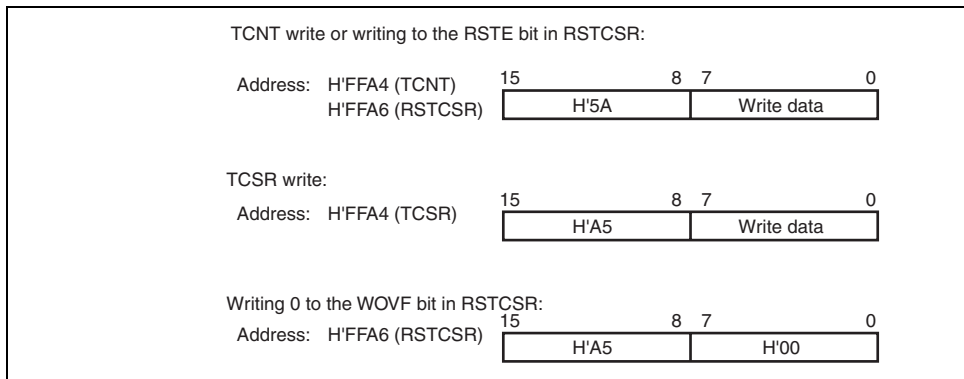


Figure 18.4 Writing to TCNT, TCSR, and RSTCSR

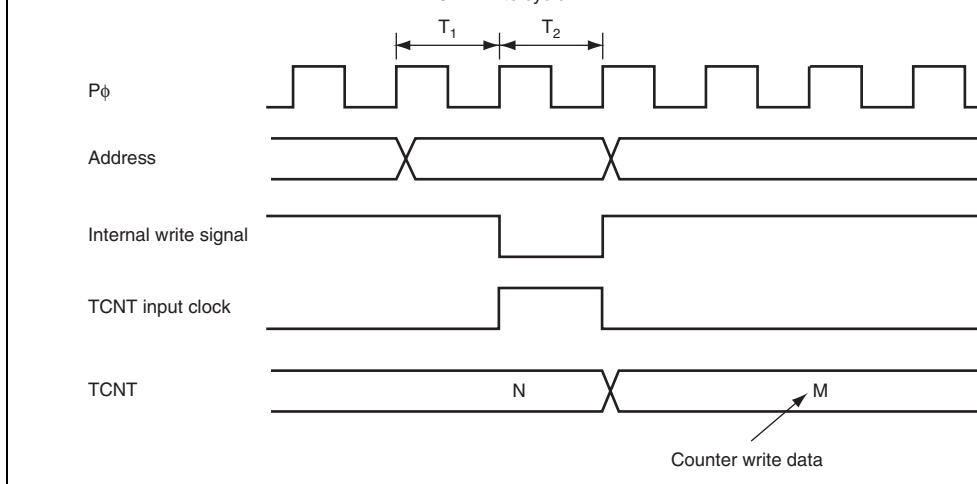


Figure 18.5 Conflict between TCNT Write and Increment

18.6.3 Changing Values of Bits CKS2 to CKS0

If bits CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors could occur in the incrementation. The watchdog timer must be stopped (by clearing the TME bit to 0) before the values of bits CKS2 to CKS0 are changed.

18.6.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the timer mode is switched from watchdog timer mode to interval timer mode while the timer is operating, errors could occur in the incrementation. The watchdog timer must be stopped (by 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. To ensure that the $\overline{\text{WDTOVF}}$ signal is not input logically to the $\overline{\text{RES}}$ pin. To reset the entire system by means of the $\overline{\text{WDTOVF}}$ signal, use a circuit like that shown in figure 18.6.

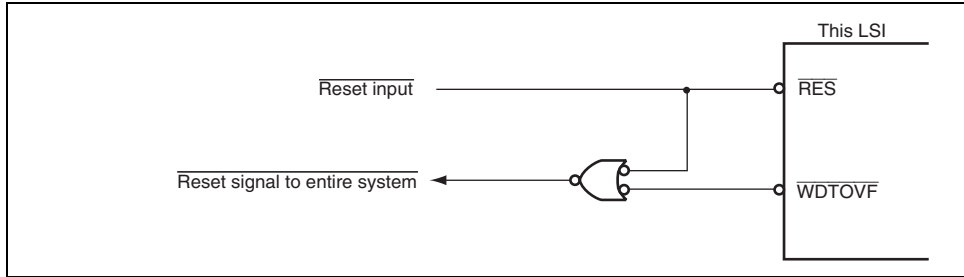


Figure 18.6 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example)

18.6.7 Transition to Watchdog Timer Mode or Software Standby Mode

When the WDT operates in watchdog timer mode, a transition to software standby mode is made even when the SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1. Instead, a transition to sleep mode is made.

To transit to software standby mode, the SLEEP instruction must be executed after halting the WDT (clearing the TME bit to 0).

When the WDT operates in interval timer mode, a transition to software standby mode is made 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 based on the IrDA Specifications version 1.0. This LSI incorporates the on-chip CRC (Cyclic Redundancy Check) computing unit that realizes high reliability of high-speed data transfer as an extended function. Since the CRC computing unit is not connected to SCI, operation is performed by writing data to registers.

Figure 19.1 shows a block diagram of the SCI_0 to SCI_4. Figure 19.2 shows a block diagram of the SCI_5 and SCI_6.

19.1 Features

- Choice of asynchronous or clock synchronous serial Communication mode
- Full-duplex Communication capability
The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously. Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected
The external clock can be selected as a transfer clock source (except for the smart card 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 receive-data-error. The transmit-data-empty and receive-data-full interrupt sources can activate the DMA controller (DMAC).
- Module stop state specifiable

16-MHz operation: 115.196 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 selected

24-MHz operation: 115.132 kbps, 460.526 kbps, 720 kbps, or 921.053 kbps can be selected

32-MHz operation: 720 kbps can be selected

Clock Synchronous Mode:

- 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 transmission
- Data can be automatically re-transmitted on receiving an error signal during transmission
- Both direct convention and inverse convention are supported

			115.152 kbps	115.15
$P\phi = 12$ MHz	—	—	460.52	230.20
$P\phi = 16$ MHz	—	720 kbps	921.50	
		460 784kbps	720 kb	
		115.196 kbps	460.78	115.19
$P\phi = 24$ MHz	—	—	921.05	
			720 kb	
			460.52	
			115.13	
$P\phi = 32$ MHz	—	720 kbps	720 kb	

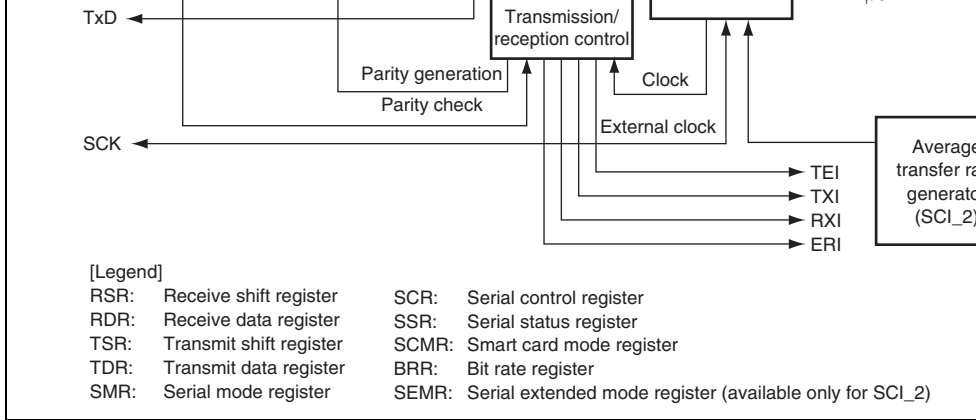


Figure 19.1 Block Diagram of SCI_0, 1, 2, 3, and 4

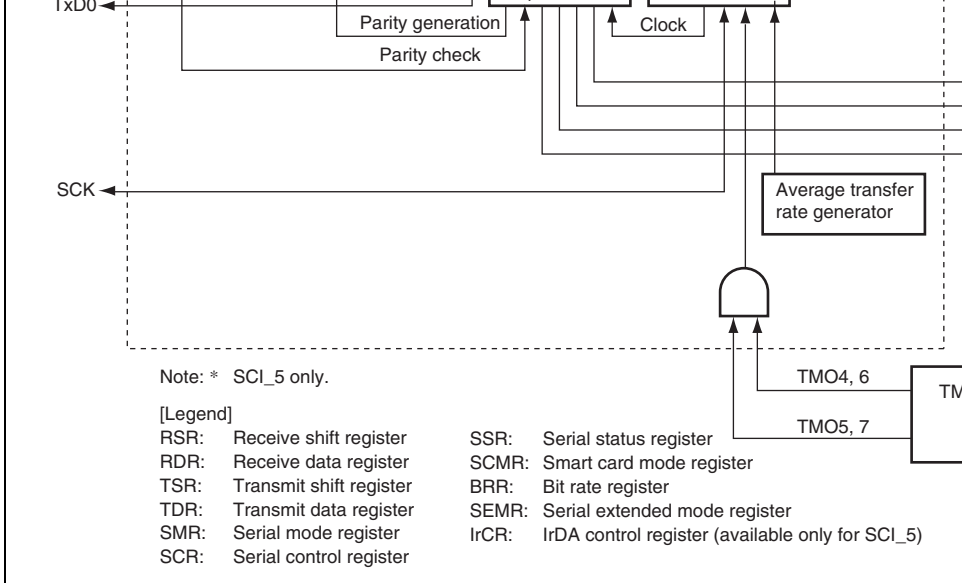


Figure 19.2 Block Diagram of SCL_5 and SCL_6

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	SCK5	I/O	Channel 5 clock input/output
	RxD5/IrRxD	Input	Channel 5 receive data input
	TxD5/IrTxD	Output	Channel 5 transmit data output
6	SCK6	I/O	Channel 6 clock input/output
	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 the 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)

Channel 3:

- Receive shift register_3 (RSR_3)
- Transmit shift register_3 (TSR_3)
- Receive data register_3 (RDR_3)
- Transmit data register_3 (TDR_3)
- Serial mode register_3 (SMR_3)
- Serial control register_3 (SCR_3)
- Serial status register_3 (SSR_3)
- Smart card mode register_3 (SCMR_3)
- Bit rate register_3 (BRR_3)

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)

• I2C control register_5 (ICR5)

Channel 6:

- Receive shift register_6 (RSR_6)
- Transmit shift register_6 (TSR_6)
- Receive data register_6 (RDR_6)
- Transmit data register_6 (TDR_6)
- Serial mode register_6 (SMR_6)
- Serial control register_6 (SCR_6)
- Serial status register_6 (SSR_6)
- Smart card mode register_6 (SCMR_6)
- Serial extended mode register_6 (SEMR_6)
- Bit rate register_6 (BRR_6)

receive the next data. Since RSR and RDR function as a double buffer in this way, continuous receive operations can be performed. After confirming that the RDRF bit in SSR is set to 1, RDR can be read from the CPU. 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

19.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffer structures of TDR and TSR enable continuous serial transmission. If the next transmit data has already been written to TDR when one frame of data is transmitted, the SCI transfers the data to TSR to continue transmission. Although TDR can be read from or written to by the CPU at all times, to achieve reliable serial transmission, write transmit data to TDR for only once after confirming that the TDRE bit in SSR is set to 1.

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

- When SMIF in SCMR = 0

Bit	7	6	5	4	3	2	1
Bit Name	C/ \bar{A}	CHR	PE	O/ \bar{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/ \bar{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

Bit Functions in Normal Serial Communication Interface Mode (When SMIF in SCMR = 0)

Bit	Bit Name	Initial Value	R/W	Description
7	C/ \bar{A}	0	R/W	Communication Mode 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length (valid only in asynchronous mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is used. In clock synchronous mode, a fixed data length is used. the MSB (bit 7) in TDR is not transmitted in transmission.

				1: Selects odd parity.
3	STOP	0	R/W	<p>Stop Bit Length (valid only in asynchronous mode). Selects the stop bit length in transmission.</p> <p>0: 1 stop bit 1: 2 stop bits</p> <p>In reception, only the first stop bit is checked. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.</p>
2	MP	0	R/W	<p>Multiprocessor Mode (valid only in asynchronous mode). When this bit is set to 1, the multiprocessor function is enabled. The PE bit and O/E bit settings are invalid in multiprocessor mode.</p>
1	CKS1	0	R/W	Clock Select 1, 0
0	CKS0	0	R/W	<p>These bits select the clock source for the baud rate generator.</p> <p>00: Pϕ clock (n = 0) 01: Pϕ/4 clock (n = 1) 10: Pϕ/16 clock (n = 2) 11: Pϕ/64 clock (n = 3)</p> <p>For the relation between the settings of these bits and the baud rate, see section 19.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 19.3.9, Bit Rate Register (BRR)).</p>

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 the data before transmission, and the parity bit is checked during reception. Set this bit to 1 in smart card interface mode.
4	O/ \bar{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 interface mode, see section 19.7.2, Data Format (Exception Transfer Mode).
3	BCP1	0	R/W	Base clock Pulse 1, 0
2	BCP0	0	R/W	These bits select the number of base clock cycles for 1-bit data transfer time in smart card interface mode. 00: 32 clock cycles (S = 32) 01: 64 clock cycles (S = 64) 10: 372 clock cycles (S = 372) 11: 256 clock cycles (S = 256) For details, see section 19.7.4, Receive Data Sampling Timing and Reception Margin. S is described in section 19.3.9, 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 SCMR = 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	<p>Transmit Interrupt Enable</p> <p>When this bit is set to 1, a TXI interrupt request is enabled.</p> <p>A TXI interrupt request can be cancelled by reading 1 from the TDRE flag and then clearing the flag to 0, or by clearing the TIE bit to 0.</p>
6	RIE	0	R/W	<p>Receive Interrupt Enable</p> <p>When this bit is set to 1, RXI and ERI interrupt requests are enabled.</p> <p>RXI and ERI interrupt requests can be cancelled by reading 1 from the RDRF, FER, PER, or OREF flag and then clearing the flag to 0, or by clearing the RIE bit to 0.</p>

When this bit is set to 1, reception is enabled. Under this condition, serial reception is started by detecting the start bit in asynchronous mode or the synchronous clock signal in clock synchronous mode. Note that SMR should be set to 0 prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by clearing this bit to 0, the RDRF, FER, PER, and ORER flags are not affected and the previous value is retained.

3	MPIE	0	R/W	<p>Multiprocessor Interrupt Enable (valid only when the multiprocessor bit in SMR is 1 in asynchronous mode)</p> <p>When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and ORER status flags in SSR is disabled. On receiving data in which the multiprocessor bit is 0, the MPIE bit is automatically cleared and normal reception is resumed. For details, see section 19.5, Multiprocessor Communication Function.</p> <p>When receive data including MPB = 0 in SSR is received, transfer of the received data from RSR to the CPU, detection of reception errors, and the settings of the RDRF, FER, and ORER flags in SSR are not performed. When receive data including MPB = 1 is received, the MPIE bit in SSR is set to 1, the MPIE bit is automatically cleared to 0, and RXI and ERI interrupt requests (in the case where the TIE and RIE bits in SCR are set to 1) and setting of the FER and ORER flags are enabled.</p>
---	------	---	-----	---

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 bit
output from the SCK pin.

1x: External clock

The clock with a frequency 16 times the bit
should be input from the SCK pin.

- Clock synchronous mode

0x: Internal clock

The SCK pin functions as the clock output

1x: External clock

The SCK pin functions as the clock input pin

- 1x: External clock or average transfer rate generator
 - When an external clock is used, the clock width frequency 16 times the bit rate should be input to the SCK pin.
 - When an average transfer rate generator is used, the clock width frequency 16 times the bit rate should be input to the SCK pin.
 - Clock synchronous mode
 - 0x: Internal clock
 - The SCK pin functions as the clock output pin.
 - 1x: External clock
 - The SCK pin functions as the clock input pin.
-

1x: External clock, TMR clock input or average transfer rate generator

When an external clock is used, the clock frequency 16 times the bit rate should be input to the SCK pin.

When an average transfer rate generator is used,

When TMR clock input is used.

- Clock synchronous mode

0x: Internal clock

The SCK pin functions as the clock output pin.

1x: External clock

The SCK pin functions as the clock input pin.

[Legend]

x: Don't care

When this bit is set to 1, RXI and ERI interrupt requests are enabled.

RXI and ERI interrupt requests can be cancelled by reading 1 from the RDRF, FER, PER, or ORER flag, then clearing the flag to 0, or by clearing the RIE bit.

5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled. Under this condition, serial transmission is started by writing transmit data to TDR, and clearing the TDRE flag to 0. Note that SMR should be set prior to setting this bit to 1 in order to designate the transmission format. If transmission is halted by clearing this bit to 0, the TDRE flag in SSR is fixed 1.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. Under this condition, serial reception is started by detecting a start bit in asynchronous mode or the synchronous clock in clock synchronous mode. Note that SMR should be set prior to setting the RE bit to 1 in order to designate the reception format. Even if reception is halted by clearing this bit to 0, the RDRF, FER, PER, and ORER flags are not affected and the previous value is retained.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (valid only when the 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

19.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. RDRF, ORER, PER, and FER can only be cleared. Some bits in SSR have different functions in 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	FER	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.

- When SMIF in SCMR = 1

Bit	7	6	5	4	3	2	1
Bit Name	TDRE	RDRF	ORER	ERS	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.

- When 0 is written to TDRE after reading TDR (When the CPU is used to clear this flag by v while the corresponding interrupt is enabled, to read the flag after writing 0 to it.)
- When a TXI interrupt request is issued allow DMAC or DTC to write data to TDR

6	RDRF	0	R/(W)*	<p>Receive Data Register Full</p> <p>Indicates whether receive data is stored in RDR.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> • When serial reception ends normally and rec is transferred from RSR to RDR <p>[Clearing conditions]</p> <ul style="list-style-type: none"> • When 0 is written to RDRF after reading RDR (When the CPU is used to clear this flag by v while the corresponding interrupt is enabled, to read the flag after writing 0 to it.) • When an RXI interrupt request is issued allow DMAC or DTC to read data from RDR <p>The RDRF flag is not affected and retains its pre value when the RE bit in SCR is cleared to 0.</p> <p>Note that when the next serial reception is comp while the RDRF flag is being set to 1, an overerr occurs and the received data is lost.</p>
---	------	---	--------	---

is set to 1, subsequent serial reception cannot be performed. Note that, in clock synchronous serial transmission also cannot continue.

[Clearing condition]

- When 0 is written to ORER after reading ORER (When the CPU is used to clear this flag by writing 0 to it while the corresponding interrupt is enabled, the CPU must read the flag after writing 0 to it.)

Even when the RE bit in SCR is cleared, the ORER flag is not affected and retains its previous value.

is transferred to RDR, however, the RDRF flag is set. In addition, when the FER flag is being set, the subsequent serial reception cannot be performed. In clock synchronous mode, serial transmission cannot continue.

[Clearing condition]

- When 0 is written to FER after reading FER = 1 (When the CPU is used to clear this flag by writing 0 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 FER flag is not affected and retains its previous value.

subsequent serial reception cannot be performed in clock synchronous mode, serial transmission 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 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.

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 TDRE • When a TXI interrupt request is issued allowing DMAC or DTC to write data to TDR
1	MPB	0	R	Multiprocessor Bit Stores the multiprocessor bit value in the receiver. When the RE bit in SCR is cleared to 0 its previous value is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Sets the multiprocessor bit value to be added to the transmit frame.

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

- When 0 is written to TDRE after reading TDR (When the CPU is used to clear this flag by v while the corresponding interrupt is enabled, to read the flag after writing 0 to it.)
- When a TXI interrupt request is issued allow DMAC or DTC to write data to TDR

6	RDRF	0	R/(W)*	<p>Receive Data Register Full</p> <p>Indicates whether receive data is stored in RDR.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> • When serial reception ends normally and rec is transferred from RSR to RDR <p>[Clearing conditions]</p> <ul style="list-style-type: none"> • When 0 is written to RDRF after reading RDR (When the CPU is used to clear this flag by v while the corresponding interrupt is enabled, to read the flag after writing 0 to it.) • When an RXI interrupt request is issued allow DMAC or DTC to read data from RDR <p>The RDRF flag is not affected and retains its pre value even when the RE bit in SCR is cleared to</p> <p>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.</p>
---	------	---	--------	--

is set to 1, subsequent serial reception cannot be performed. Note that, in clock synchronous serial transmission also cannot continue.

[Clearing condition]

- When 0 is written to OREER after reading OREER (When the CPU is used to clear this flag by software while the corresponding interrupt is enabled, it is recommended to read the flag after writing 0 to it.)

Even when the RE bit in SCR is cleared, the OREER flag is not affected and retains its previous value.

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 performed. In clock synchronous mode, serial transmission cannot continue.

[Clearing condition]

- When 0 is written to PER after reading PER (When the CPU is used to clear this flag by writing 0 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 value.

Set timing depends on the register setting.

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 TDRE after reading TDRE
- When a TXI interrupt request is issued allowing DMAC or DTC to write the next data to TDR

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 used for transmission/reception; when the 7-bit data format is 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 bit does not affect the logic level of the parity bit. To invert the parity bit, invert the O/\bar{E} bit in SMR. 0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR. 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR.
1	—	1	—	Reserved This bit is always read as 1.
0	SMIF	0	R/W	Smart Card Interface Mode Select When this bit is set to 1, smart card interface mode is selected. 0: Normal asynchronous or clock synchronous mode 1: Smart card interface mode

Asynchronous mode	0	$N = \frac{P\phi \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} \right\}$
	1	$N = \frac{P\phi \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} \right\}$
Clock synchronous mode		$N = \frac{P\phi \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface mode		$N = \frac{P\phi \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} \right\}$

[Legend]

B: Bit rate (bit/s)

N: BRR setting for baud rate generator ($0 \leq N \leq 255$)

$P\phi$: Operating frequency (MHz)

n and S: Determined by the SMR settings shown in the following table.

SMR Setting			SMR Setting		
CKS1	CKS0	n	BCP1	BCP0	S
0	0	0	0	0	3
0	1	1	0	1	6
1	0	2	1	0	3
1	1	3	1	1	2

Table 19.4 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

Bit Rate (bit/s)	Operating Frequency P ϕ (MHz)										
	8			9.8304			10			12	
	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 set to 1.
When the ABCS bit is set to 1, the bit rate is two times.

Table 19.4 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

Bit Rate (bit/s)	Operating Frequency P _φ (MHz)											
	17.2032			18			19.6608			20		
	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	100	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	27

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 19.5 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode)

$P\phi$ (MHz)	Maximum Bit Rate (bit/s)	n	N	$P\phi$ (MHz)	Maximum Bit Rate (bit/s)	n
8	250000	0	0	17.2032	537600	0
9.8304	307200	0	0	18	562500	0
10	312500	0	0	19.6608	614400	0
12	375000	0	0	20	625000	0
12.288	384000	0	0	25	781250	0
14	437500	0	0	30	937500	0
14.7456	460800	0	0	33	1031250	0
16	500000	0	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	1
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*			0	1	—	—	0	2	—	—	—
5M							0	0*	—	—	—	—	—	—	—

[Legend]

Space: Setting prohibited.

—: Can be set, but there will be error.

Note: * Continuous transmission or reception is not possible.

Table 19.9 BRR Settings for Various Bit Rates
(Smart Card Interface Mode, n = 0, S = 372)

Bit Rate (bit/sec)	Operating Frequency P ϕ (MHz)											
	7.1424			10.00			10.7136			13.5663		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	0	0.00	0	1	30	0	1	25	0	1	0

Bit Rate (bit/sec)	Operating Frequency P ϕ (MHz)											
	14.2848			16.00			18.00			20.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	15.99	0	2	0

Bit Rate (bit/sec)	Operating Frequency P ϕ (MHz)											
	25.00			30.00			33.00			35.76		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	3	12.49	0	3	5.01	0	4	7.59	0	4	0

19.3.10 Serial Extended Mode Register (SEMR_2)

SEMR_2 selects the clock source in asynchronous mode of SCI_2. The base clock is automatically specified when the average transfer rate operation is selected.

Bit	7	6	5	4	3	2	1
Bit Name	—	—	—	—	ABCS	ACS2	ACS1
Initial Value	Undefined	Undefined	Undefined	Undefined	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	R	Reserved These bits are always read as undefined and cannot be modified.
3	ABCS	0	R/W	Asynchronous Mode Base clock Select (valid only in asynchronous mode) Selects the base clock for a 1-bit period. 0: The base clock has a frequency 16 times the baud rate 1: The base clock has a frequency 8 times the baud rate

base clock with a frequency 16 times the transfer rate)

010: 460.606 kbps of average transfer rate specified
 $P\phi = 10.667$ MHz is selected (operated using the base clock with a frequency 8 times the transfer rate)

011: 720 kbps of average transfer rate specified
32 MHz is selected (operated using the base clock with a frequency 16 times the transfer rate)

100: Setting prohibited

101: 115.196 kbps of average transfer rate specified
 $P\phi = 16$ MHz is selected (operated using the base clock with a frequency 16 times the transfer rate)

110: 460.784 kbps of average transfer rate specified
 $P\phi = 16$ MHz is selected (operated using the base clock with a frequency 16 times the transfer rate)

111: 720 kbps of average transfer rate specified
16 MHz is selected (operated using the base clock with a frequency 8 times the transfer rate)

The average transfer rate only supports operation at clock frequencies of 10.667 MHz, 16 MHz, and 32 MHz.

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 cannot be modified.
4	ABCS	0	R/W	Asynchronous Mode Base Clock Select (valid only in asynchronous mode) Selects the base clock for a 1-bit period. 0: The base clock has a frequency 16 times the average transfer rate 1: The base clock has a frequency 8 times the average transfer rate
3	ACS3	0	R/W	Asynchronous Mode Clock Source Select
2	ACS2	0	R/W	These bits select the clock source for the average transfer rate function in the asynchronous mode. If the average transfer rate function is enabled, the clock is automatically specified regardless of the bit value. The average transfer rate only corresponds to 8MHz, 10.667MHz, 12MHz, 16MHz, 24MHz, and 32MHz. No other clock is available. Setting of ACS0 must be done in the asynchronous mode (the C/A bit in SMR = 0) and the external clock input must be selected (the CKE bit I SCR = 1). The setting examples are shown in figures 16.3 and 16.4. (Each number in the four-digit number below corresponds to the value in the bits ACS3 to ACS0 left to right respectively.)
1	ACS1	0	R/W	
0	ACS0	0	R/W	

average transfer rate specific to $P\phi = 8\text{MHz}$ is selected (operated using the base clock with a frequency 8 times the transfer rate)

0100: TMR clock input
This setting allows the TMR compare match output to be used as the base clock. The table below shows the correspondence between SCI channels and the compare match output.

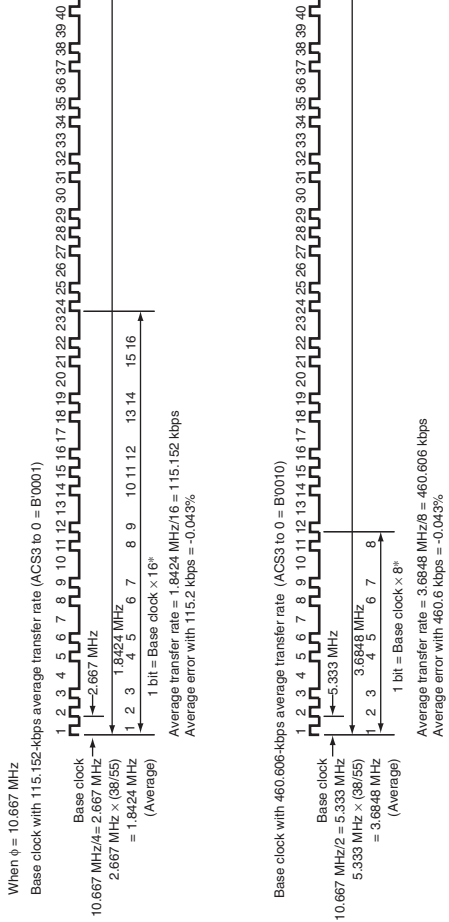
SCI Channel	TMR Unit	Compare Match Output
SCI_5	Unit 2	TMO4, TMO5
SCI_6	Unit 3	TMO6, TMO7

0101: 115.196 kbps of average transfer rate specific to $P\phi = 16\text{MHz}$ is selected (operated using the base clock with a frequency 16 times the transfer rate)

0110: 460.784 kbps of average transfer rate specific to $P\phi = 16\text{MHz}$ is selected (operated using the base clock with a frequency 16 times the transfer rate)

0111: 720 kbps of average transfer rate specific to $P\phi = 16\text{MHz}$ is selected (operated using the base clock with a frequency 8 times the transfer rate)

- 1011: 921.053 kbps of average transfer rate specific to $P\phi = 24$ or MHz or 460.526 kbps of average transfer rate specific to $P\phi = 12$ MHz is selected (operated using the base clock with a frequency 16 times the transfer rate)
 - 1100: 720 kbps of average transfer rate specific to 32 MHz is selected (operated using the base clock with a frequency 16 times the transfer rate)
 - 1101: Reserved (setting prohibited)
 - 111x: Reserved (setting prohibited)
-



Note: * The length of one bit varies according to the base clock synchronization.

Figure 19.3 Examples of Base Clock when Average Transfer Rate is Selected

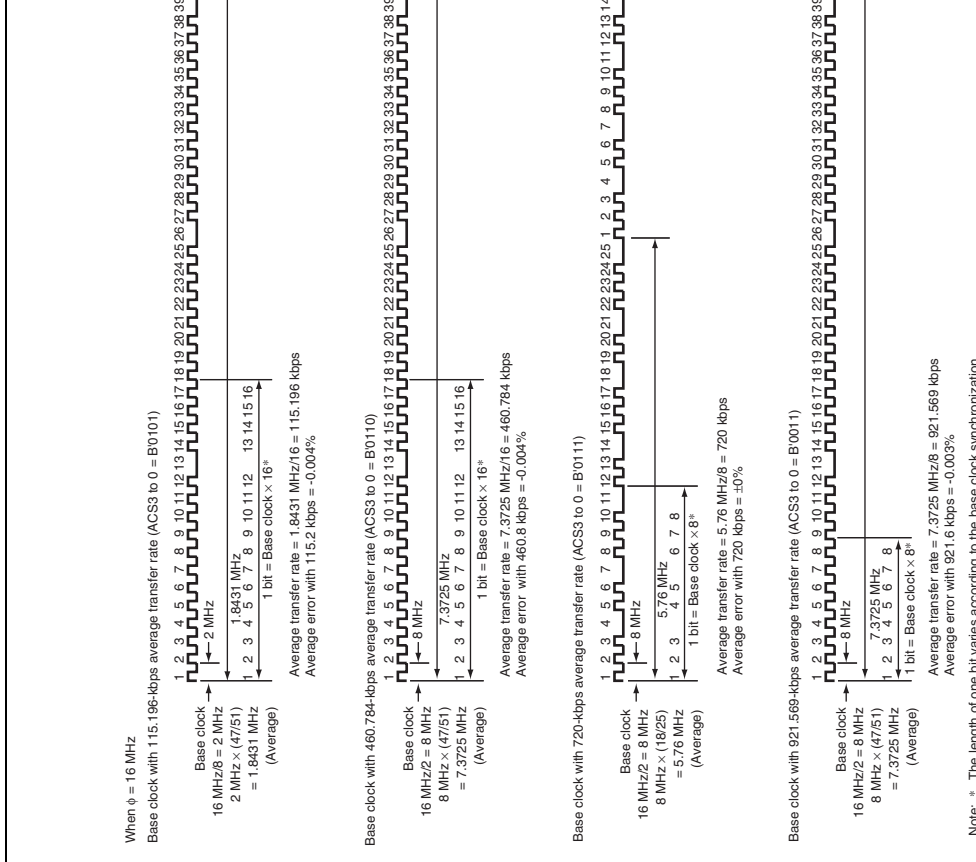


Figure 19.3 Examples of Base Clock when Average Transfer Rate is Selected

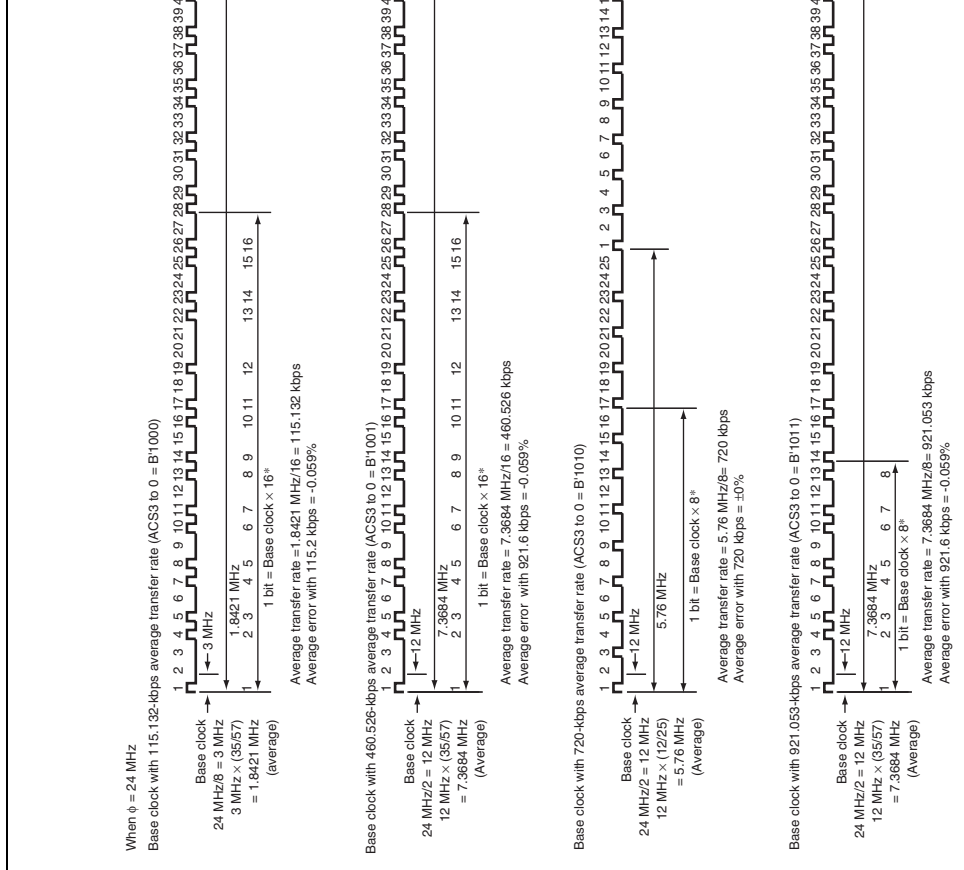


Figure 19.3 Examples of Base Clock when Average Transfer Rate is Selected

Example when TMR clock input is used in SCI_5
 187.5-kbps average transfer rate is generated by TMR when $\phi = 32$ MHz
 (1) TMO4 is set as a base clock and generates 4 MHz.
 (2) TMO5 is set as TCNT_4 compare match count and generates a clock enable multiplied by 3/4.
 The average transfer rate will be $3 \text{ MHz}/16 = 187.5 \text{ kbps}$.

TMR and SCI Settings:

- TCR_4 = H'09 (TCNT4 cleared by TCORA_4 compare match, TCNT4 incremented at rising edge of P0(2))
 - TCCR_4 = H'01
 - TCR_5 = H'0C (TCNT5 cleared by TCORA_5 compare match, TCNT5 incremented by TCNT_4 compare match A)
 - TCCR_5 = H'00
 - TCSR_4 = H'09 (0 output on TCORA_4 compare match, 1 output on TCORB_4 compare match)
 - TCSR_5 = H'09 (0 output on TCORA_5 compare match, 1 output on TCORB_5 compare match)
 - TCNT_4 = TCNT_5 = 0
 - TCORA_4 = H'03, TCORB_4 = H'01
 - TCORA_5 = H'03, TCORB_5 = H'00
 - SEMR_5 = H'04
- When SCI_6 is used, set TMO6 as a base clock and TMO7 as a clock enable.

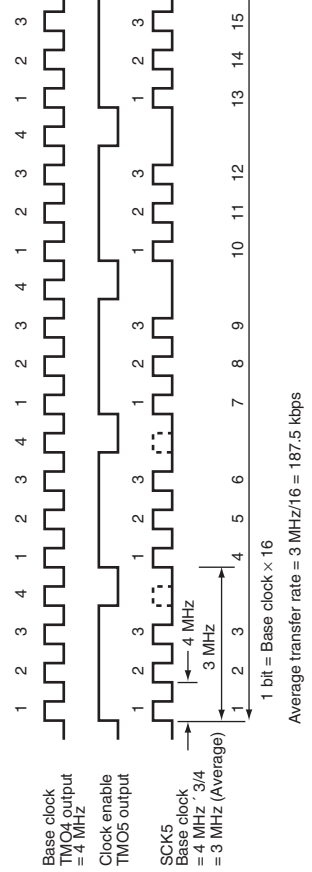
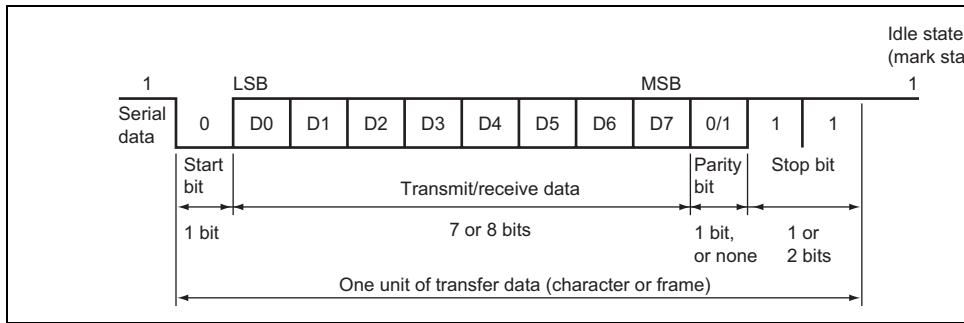


Figure 19.4 Example of Average Transfer Rate Setting when TMR Clock is In

Bit	Bit Name	Value	R/W	Description
7	IrE	0	R/W	<p>IrDA Enable*</p> <p>Sets the SCI_5 I/O to normal SCI or IrDA.</p> <p>0: TxD5/IrTxD and RxD5/IrRxD pins operate as TxD5 and RxD5.</p> <p>1: TxD5/IrTxD and RxD5/IrRxD pins are operated as IrTxD and IrRxD.</p>
6	IrCK2	0	R/W	IrDA Clock Select 2 to 0
5	IrCK1	0	R/W	Sets the pulse width of high state at encoding
4	IrCK0	0	R/W	<p>output pulse when the IrDA function is enabled</p> <p>000: Pulse-width = $B \times 3/16$ (Bit rate $\times 3/16$)</p> <p>001: Pulse-width = $P\phi/2$</p> <p>010: Pulse-width = $P\phi/4$</p> <p>011: Pulse-width = $P\phi/8$</p> <p>100: Pulse-width = $P\phi/16$</p> <p>101: Pulse-width = $P\phi/32$</p> <p>110: Pulse-width = $P\phi/64$</p> <p>111: Pulse-width = $P\phi/128$</p>
3	IrTxINV	0	R/W	<p>IrTx Data Invert</p> <p>This bit specifies the inversion of the logic level of the IrTx output. When inversion is done, the pulse width of the high state specified by the bits 6 to 4 becomes the pulse width in low state.</p> <p>0: Outputs the transmission data as it is as IrTx output</p> <p>1: Outputs the inverted transmission data as IrTx output</p>

Note: * The IrDA function should be used when the ABCS bit in SEMR_5 is set to 0 and ACS3 to ACS0 bits in SEMR_5 and SEMR_6 are set to B'0000.



**Figure 19.5 Data Format in Asynchronous Communication
(Example with 8-Bit Data, Parity, Two Stop Bits)**

19.4.1 Data Transfer Format

Table 19.11 shows the data transfer formats that can be used in asynchronous mode. Any transfer formats can be selected according to the SMR setting. For details on the multiprocessor bit, see section 19.5, Multiprocessor Communication Function.

0	1	0	1	S	8-bit data	P	STOP
1	0	0	0	S	7-bit data	STOP	
1	0	0	1	S	7-bit data	STOP	STOP
1	1	0	0	S	7-bit data	P	STOP
1	1	0	1	S	7-bit data	P	STOP
0	-	1	0	S	8-bit data	MPB	STOP
0	-	1	1	S	8-bit data	MPB	STOP
1	-	1	0	S	7-bit data	MPB	STOP
1	-	1	1	S	7-bit data	MPB	STOP

[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 by the 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 allowed in system design.

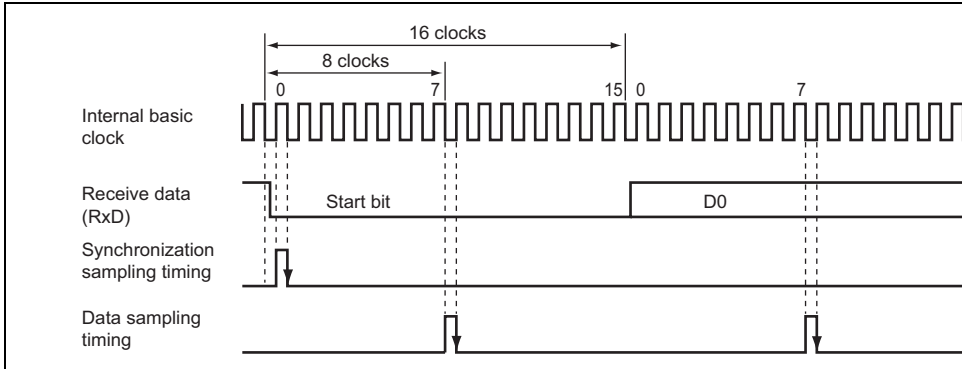


Figure 19.6 Receive Data Sampling Timing in Asynchronous Mode

Note: * This is an example when the ABCS bit in SEMR_2, 5, and 6 is 0. When the ABCS bit is 1, a frequency of 8 times the bit rate is used as a base clock and receive data is 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 pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 19.7.

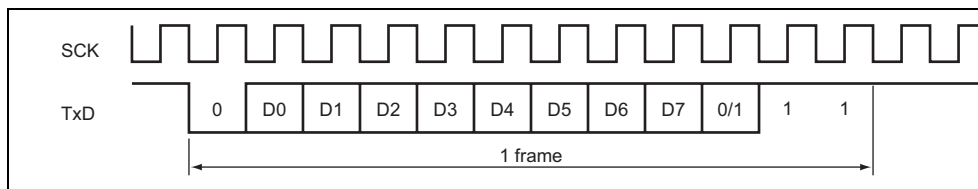


Figure 19.7 Phase Relation between Output Clock and Transmit Data (Asynchronous Mode)

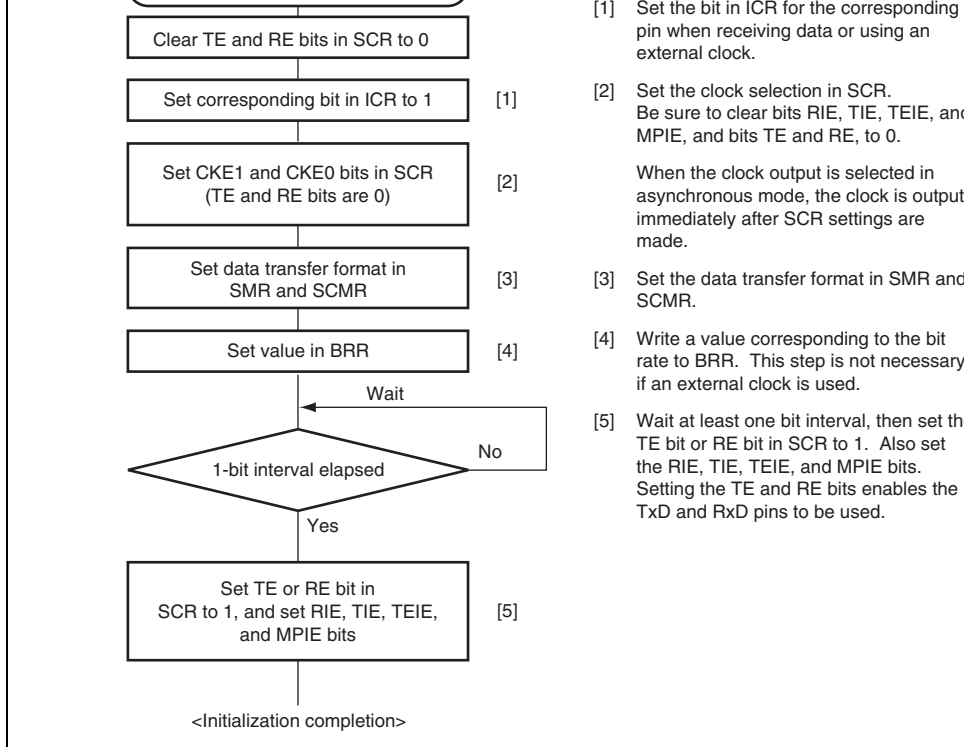
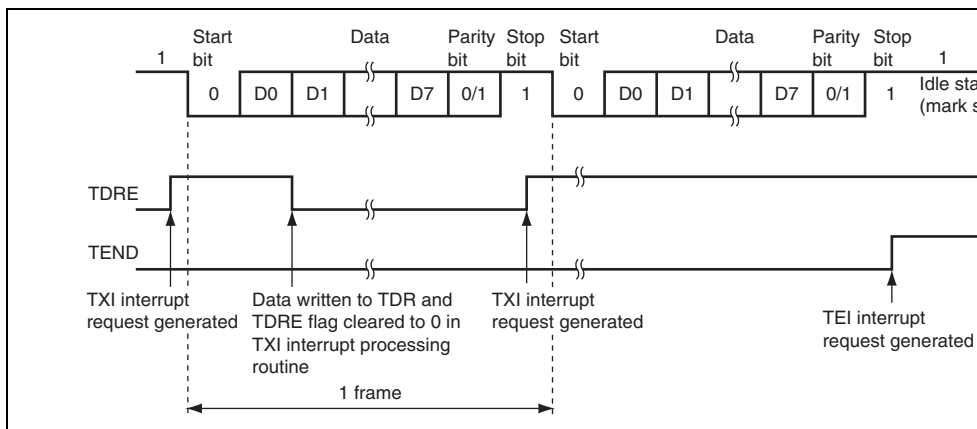


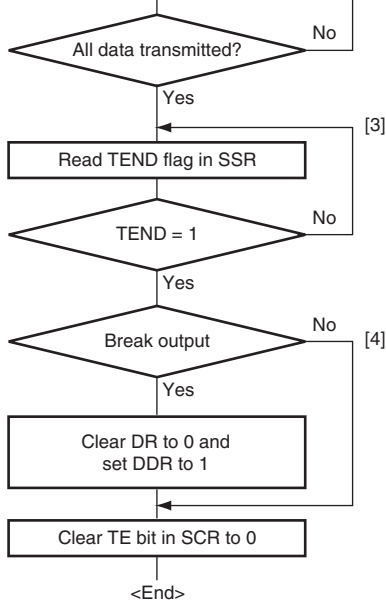
Figure 19.8 Sample SCI Initialization Flowchart

3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit, 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 stop bit is sent, and then serial transmission of the next frame is started.
6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the state is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TX interrupt request is generated.

Figure 19.10 shows a sample flowchart for transmission in asynchronous mode.



**Figure 19.9 Example of Operation for Transmission in Asynchronous Mode
(Example with 8-Bit Data, Parity, One Stop Bit)**

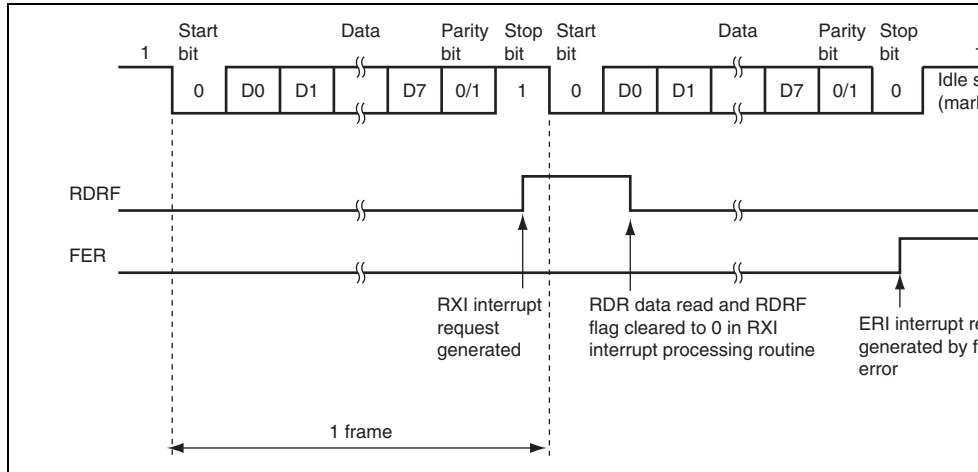


To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DMAC or DTC is initiated by a transmit data empty interrupt (TXI) request and writes 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 19.10 Example of Serial Transmission Flowchart

3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
5. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Because the RXI interrupt processing routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.



**Figure 19.11 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 error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

Note: * The RDRF flag retains the state it had before data reception.

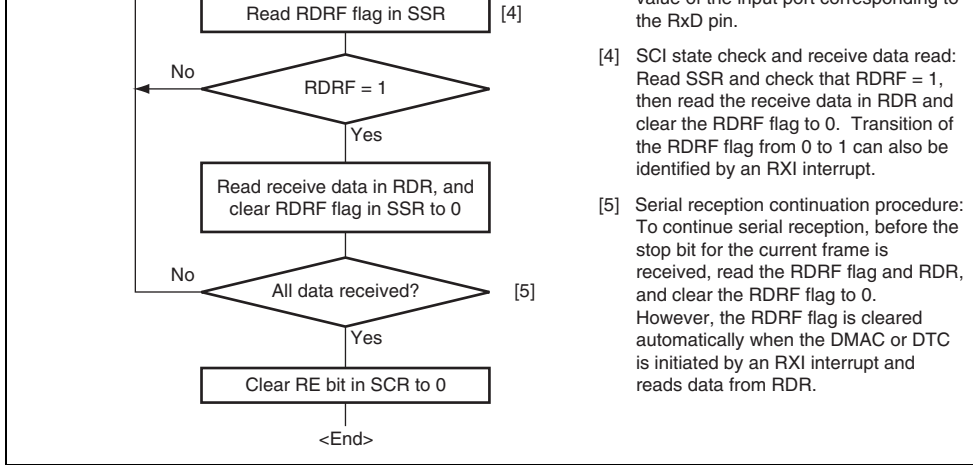


Figure 19.12 Sample Serial Reception Flowchart (1)

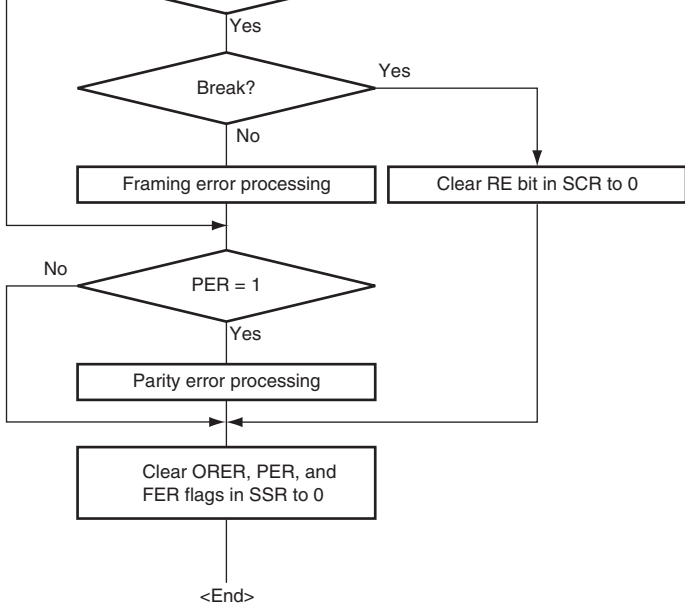


Figure 19.12 Sample Serial Reception Flowchart (2)

19.13 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends data which includes the ID code of the receiving station and a multiprocessor bit set to 1. It then transmits data added with a multiprocessor bit set 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 ID code. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set to 1, the transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status bits RDRF, FER, and ORER in SSR to 1 are prohibited until data with a 1 multiprocessor bit is received. On reception of a receive character with a 1 multiprocessor bit, the MPB bit in SCR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the MPIE bit in SCR is set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is invalid. All other bit settings are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

ID transmission cycle = Data transmission cycle =
receiving station Data transmission to
specification receiving station specified by ID

[Legend]
MPB: Multiprocessor bit

**Figure 19.13 Example of Communication Using Multiprocessor Format
(Transmission of Data H'AA to Receiving Station A)**

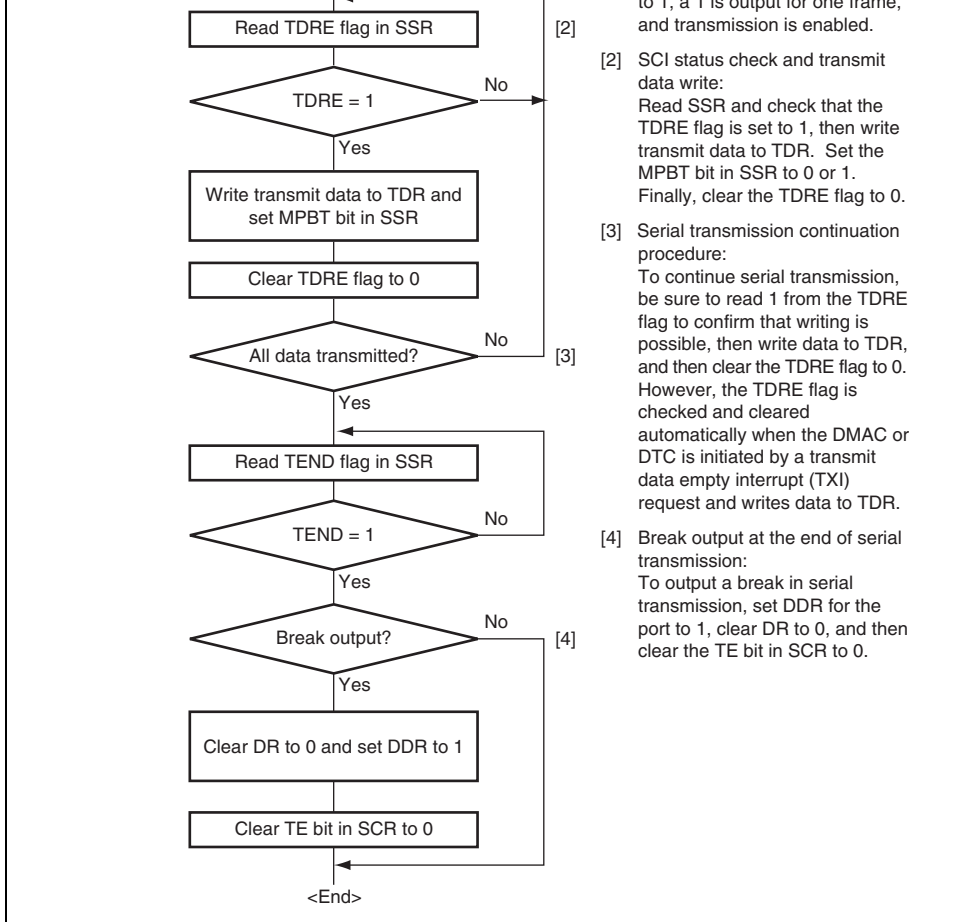
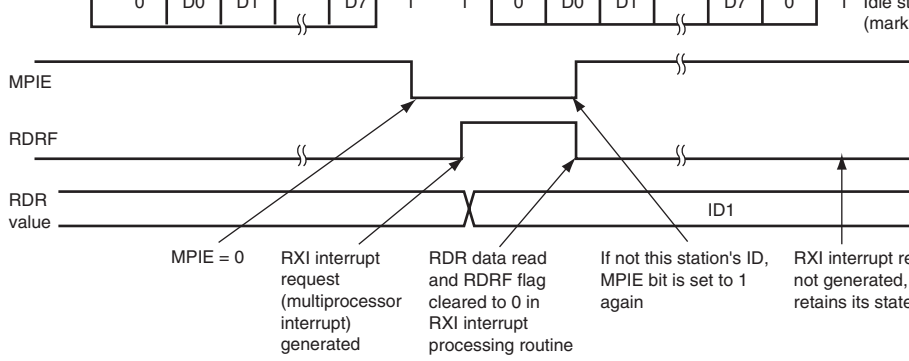
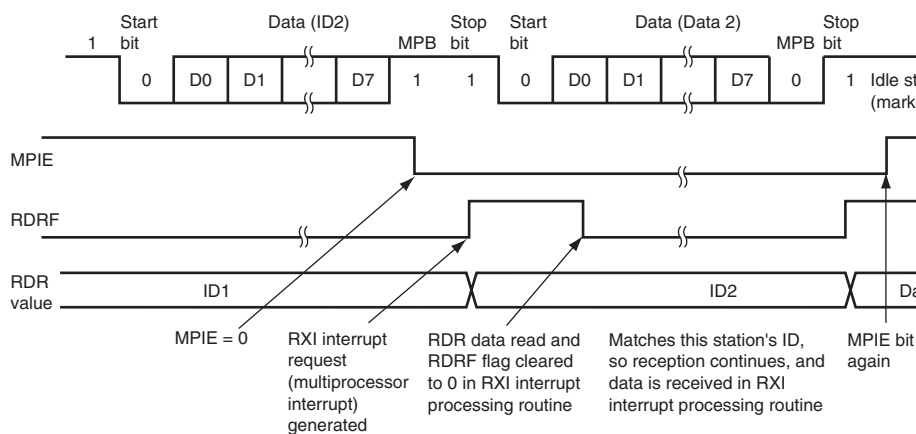


Figure 19.14 Sample Multiprocessor Serial Transmission Flowchart



(a) Data does not match station's ID



(b) Data matches station's ID

**Figure 19.15 Example of SCI Operation for Reception
(Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)**

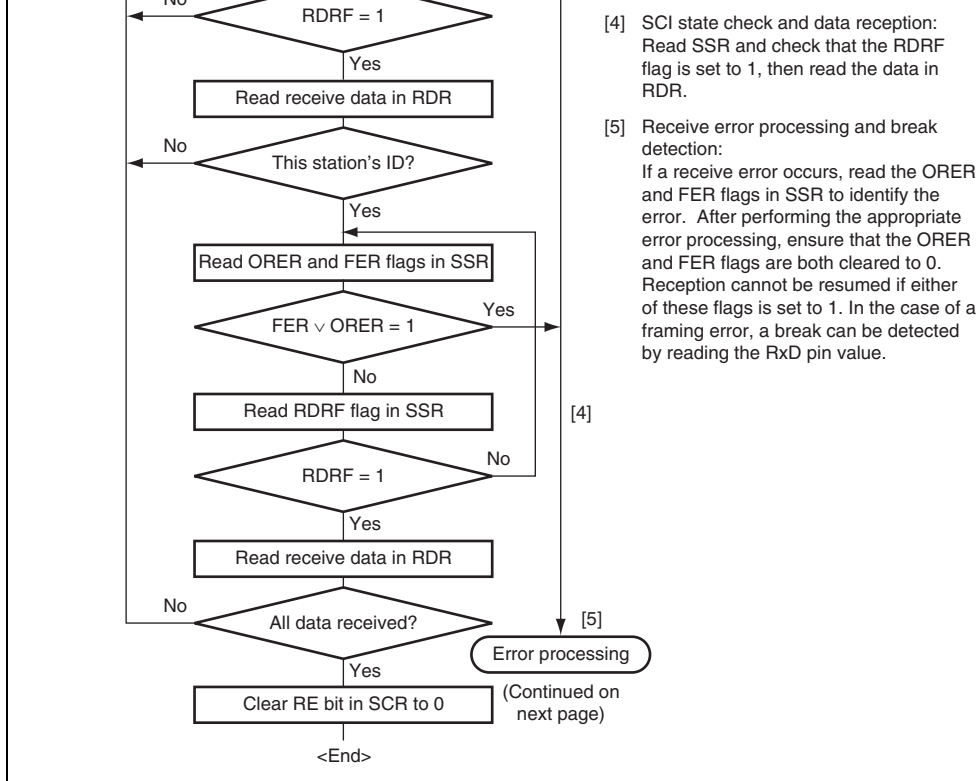


Figure 19.16 Sample Multiprocessor Serial Reception Flowchart (1)

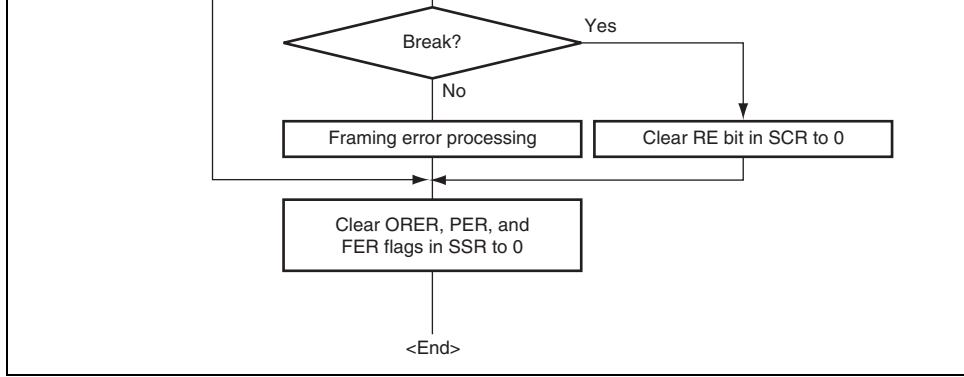


Figure 19.16 Sample Multiprocessor Serial Reception Flowchart (2)

receiver also have a double buffered structure, so that the next transmit data can be written during the previous transmission or the previous receive data can be read during reception, enabling continuous transfer.

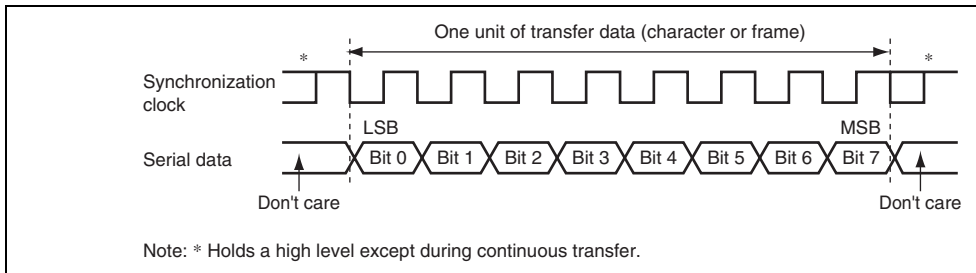
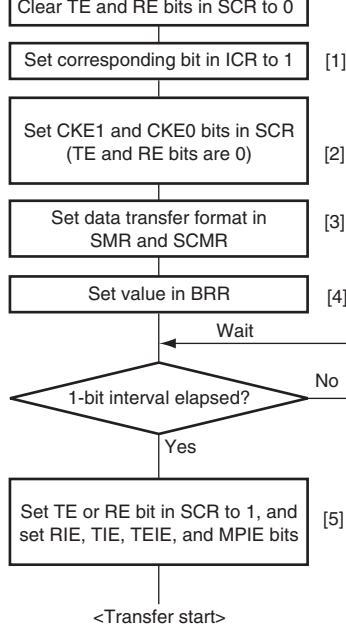


Figure 19.17 Data Format in Clock Synchronous Communication (LSB-First)

19.6.1 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 the SCKE and CKE0 bits in SCR. When the SCI is operated on an internal clock, the synchronization clock is output from the SCK pin. Eight synchronization clock pulses are output in the transfer period of one character, and when no transfer is performed the clock is fixed high. Note that in the case of reception only, the synchronization clock is output until an overrun error occurs or until the overrun error flag is cleared to 0.



- pin when receiving data or using an external clock.
- [1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.
 - [2] Set the data transfer format in SMR and SCMR.
 - [3] Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.
 - [4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits. Setting the TE and RE bits enables the TxD and RxD pins to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 19.18 Sample SCI Initialization Flowchart

3. 8-bit data is sent from the TxD pin synchronized with the output clock when clock output mode has been specified and synchronized with the input clock when use of an external clock has been specified.
4. The SCI checks the TDRE flag at the timing for sending the last bit.
5. If the TDRE flag is cleared to 0, the next transmit data is transferred from TDR to TSR and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TxD pin retains its output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt is generated. The SCK pin is fixed high.

Figure 19.20 shows a sample flowchart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) is set. Make sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

3. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Because the RXI interrupt processing routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

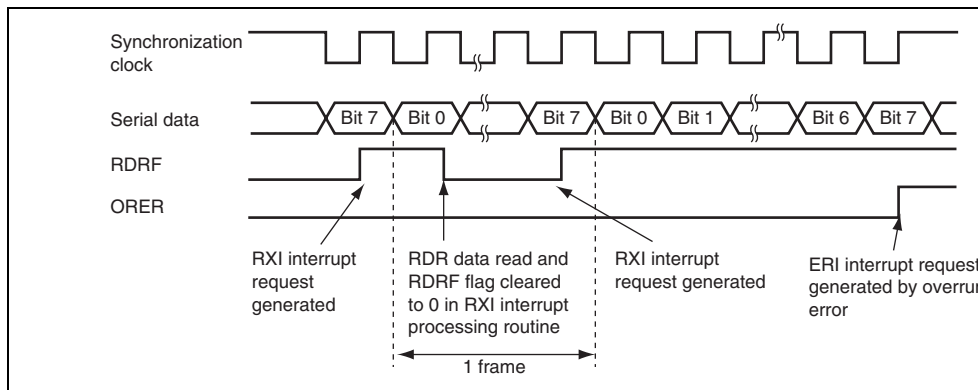
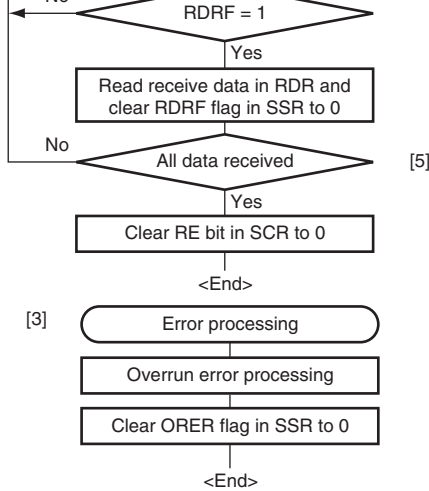


Figure 19.21 Example of Operation for Reception in Clock Synchronous Mode

Transfer cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 19.22 shows a sample timing diagram for serial data reception.



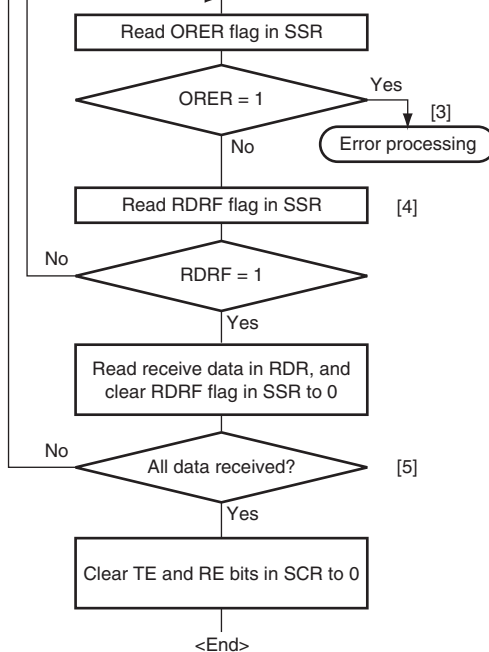
data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[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 19.22 Sample Serial Reception Flowchart

19.6.5 Simultaneous Serial Data Transmission and Reception (Clock Synchron

Figure 19.23 shows a sample flowchart for simultaneous serial transmit and receive operations. 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 and receive mode, after checking that the SCI has finished transmission and the TDRE and TE flags are set to 1, clear the TE bit to 0. Then simultaneously set both the TE and RE bits to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI has finished reception, clear the RE bit to 0. Then after checking that the RDRF bit and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneously set 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 if 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 the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0. 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. 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 19.23 Sample Flowchart of Simultaneous Serial Transmission and Reception

TxD and RxD pins and pull up the data transmission line to V_{CC} using a resistor. Setting 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 pin of this LSI.

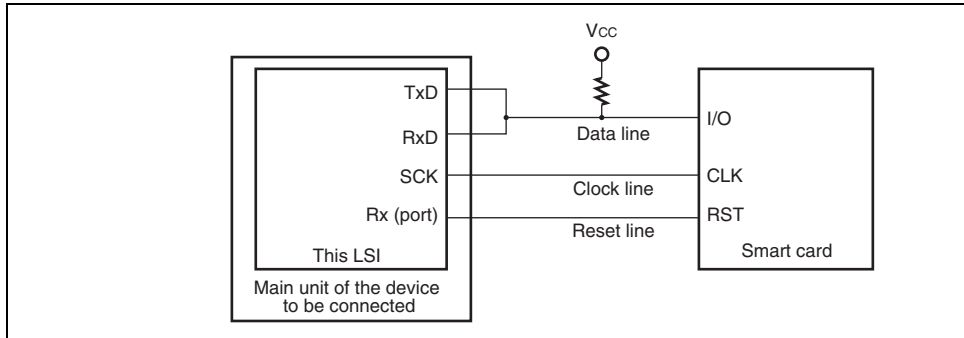


Figure 19.24 Pin Connection for Smart Card Interface

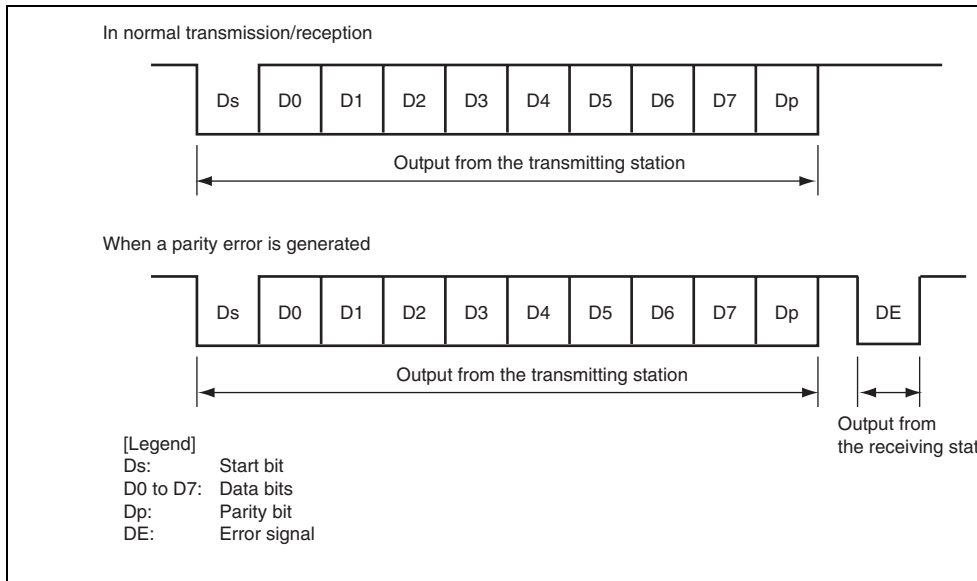


Figure 19.25 Data Formats in Normal Smart Card Interface Mode

For Communication with the smart cards of the direct convention and inverse convention follow the procedure below.

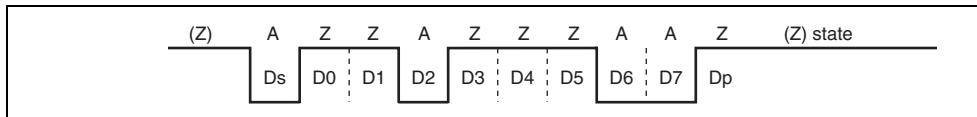


Figure 19.26 Direct Convention (SDIR = SINV = $O/\bar{E} = 0$)

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively, and data is transferred with MSB-first as the start character, as shown in figure 19.27. The data in the start character in the figure is H'3F. When using the inverse convention type, both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SN this LSI only inverts data bits D7 to D0, write 1 to the O/ \bar{E} bit in SMR to invert the parity both transmission and reception.

19.7.3 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 PER bit in SSR is set by error detection, clear the PER bit before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time after the end of the parity bit before the start of the next frame.
- Since the same data is not re-transmitted during transmission, the TEND flag is set 1 after transmission start.
- Although the ERS flag in block transfer mode displays the error signal status as in normal smart card interface mode, the flag is always read as 0 because no error signal is transmitted.

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

[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 margin determined by the formula below.

$$M = \left(0.5 - \frac{1}{2 \times 372} \right) \times 100\% = 49.866\%$$

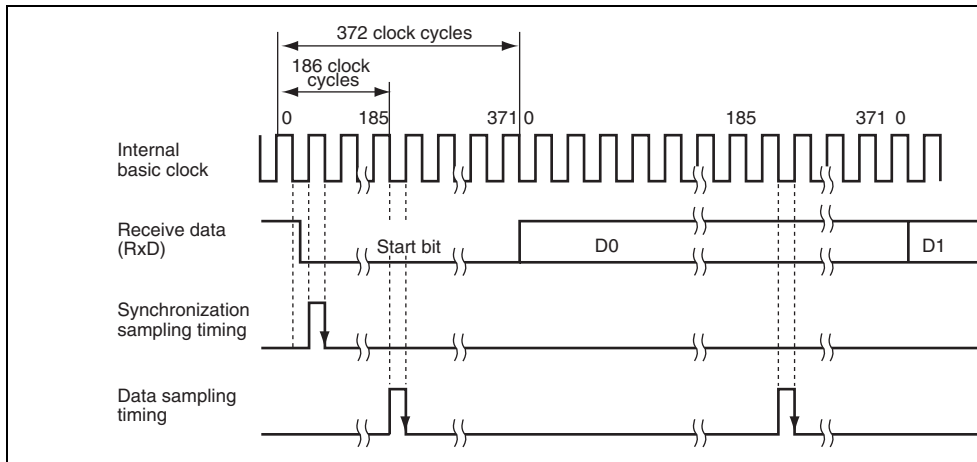


Figure 19.28 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

5. Set the SMIF, SDIR, and SINV bits in SCMR appropriately. When the DDR corresponding to the TxD pin is cleared to 0, the TxD and RxD pins are changed from port pins to SCK pins, 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, MIE, and 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-μs interval. Setting the TE and RE bits to 1 simultaneously is prohibited except for self-reception.

To switch from reception to transmission, first verify that reception has completed, then initialize the SCI. At the end of initialization, RE and TE should be set to 0 and 1, respectively. Reception 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 SCI. At 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.

- transferred from TDR to TDR allowing automatic data retransmission.
3. If no error signal is returned from the receiving end, the ERS bit in SSR is not set to 1.
4. In this case, one frame of data is determined to have been transmitted including re-transmission. The TEND bit in SSR is set to 1. Here, a TXI interrupt request is generated if the TIE bit in SCR is set to 1. Writing transmit data to TDR starts transmission of the next data.

Figure 19.31 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 generating a TXI interrupt request if the TIE bit in SCR has been set to 1. This activates the DTC or DMAC as a TXI request thus allowing transfer of transmit data if the TXI interrupt request is specifically enabled as a source of DTC or DMAC activation beforehand. The TDRE and TEND flags are automatically cleared to 0 at data transfer by the DTC or DMAC. If an error occurs, the SCI automatically retransmits the same data. During re-transmission, TEND remains as 0, thus not activating the DTC or DMAC. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including re-transmission in the case of error occurrence. However, the ERS flag is not automatically cleared; the ERS flag must be cleared by previously setting the RIE bit to 1 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 the DTC or DMAC prior to making SCI settings. For DTC or DMAC settings, see section 12, Data Transfer Controller (DTC) and section 10, DMA Controller (DMAC).

Note that the TEND flag is set in different timings depending on the GM bit setting in S
 Figure 19.30 shows the TEND flag set timing.

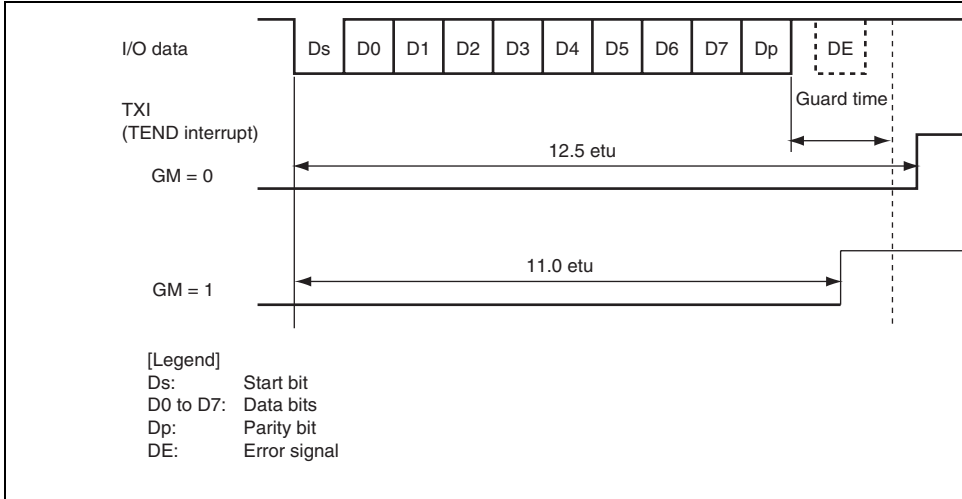


Figure 19.30 TEND Flag Set Timing during Transmission

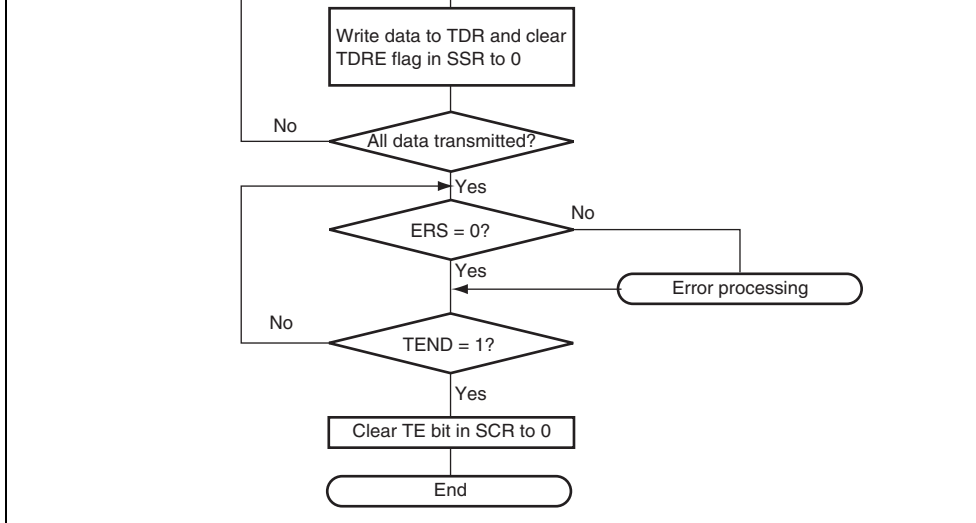


Figure 19.31 Sample Transmission Flowchart

4. In this case, data is determined to have been received successfully, and the RDRF bit is set to 1. Here, an RXI interrupt request is generated if the RIE bit in SCR is set to 1.

Figure 19.33 shows a sample flowchart for reception. All the processing steps are automatically performed using an RXI interrupt request to activate the DTC or DMAC. In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated when the RDRF flag is set to 1. This interrupt request then activates the DTC or DMAC by an RXI request thus allowing transfer of receive data if the DTC or DMAC interrupt request is specified as a source of DTC or DMAC activation beforehand. The RDRF flag is automatically cleared to 0 at data transfer by the DTC or DMAC. If an error occurs during 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 DMAC is not activated and receive data is skipped, therefore, the number of bytes of receive data in the DTC or DMAC is not transferred. Even if a parity error occurs and the PER bit is set to 1 during reception, receive data is transferred to RDR, thus allowing the data to be read.

Note: For operations in block transfer mode, see section 19.4, Operation in Asynchronous Mode.

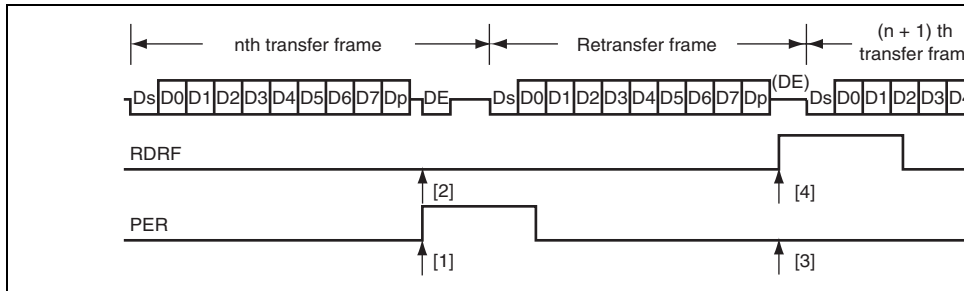


Figure 19.32 Data Re-Transfer Operation in SCI Reception Mode

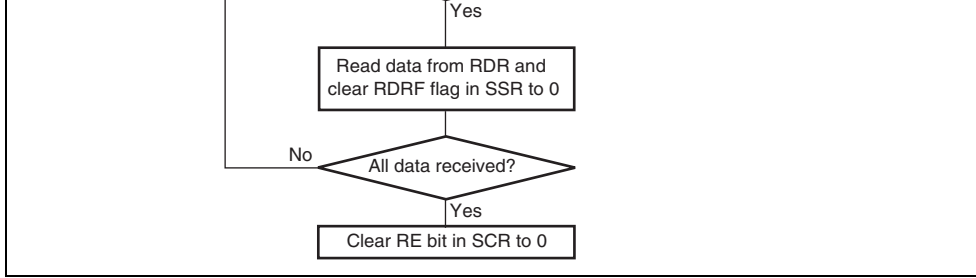


Figure 19.33 Sample Reception Flowchart

19.7.8 Clock Output Control

Clock output can be fixed using the CKE1 and CKE0 bits in SCR when the GM bit in SMCR is set to 1. Specifically, the minimum width of a clock pulse can be specified.

Figure 19.34 shows an example of clock output fixing timing when the CKE0 bit is controlled with GM = 1 and CKE1 = 0.

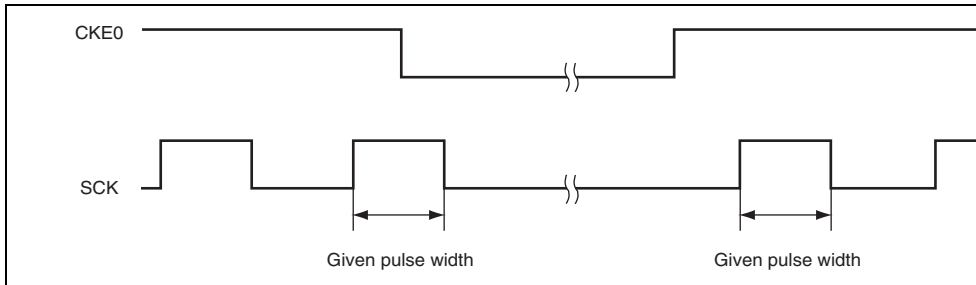


Figure 19.34 Clock Output Fixing Timing

- Set the CKE0 bit in SCR to 1 to start clock output.
- At mode switching
 - a) At transition from smart card interface mode to software standby mode
 1. Set the data register (DR) and data direction register (DDR) corresponding to pin to the values for the output fixed state in software standby mode.
 2. Write 0 to the TE and RE bits in SCR to stop transmission/reception. Simultaneously set the CKE1 bit to the value for the output fixed state in software standby mode.
 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 fixed at the specified level with the duty cycle retained.
 5. Make the transition to software standby mode.
 - b) At replacement from software standby mode to smart card interface
 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.

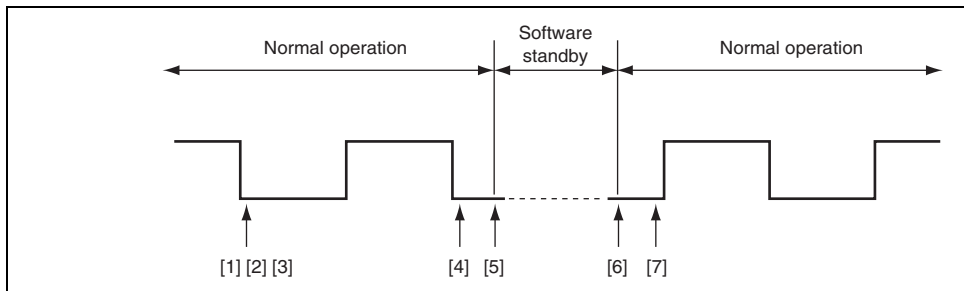


Figure 19.35 Clock Stop and Restart Procedure

rate, the transfer rate must be modified through programming.

Figure 19.36 is the IrDA block diagram.

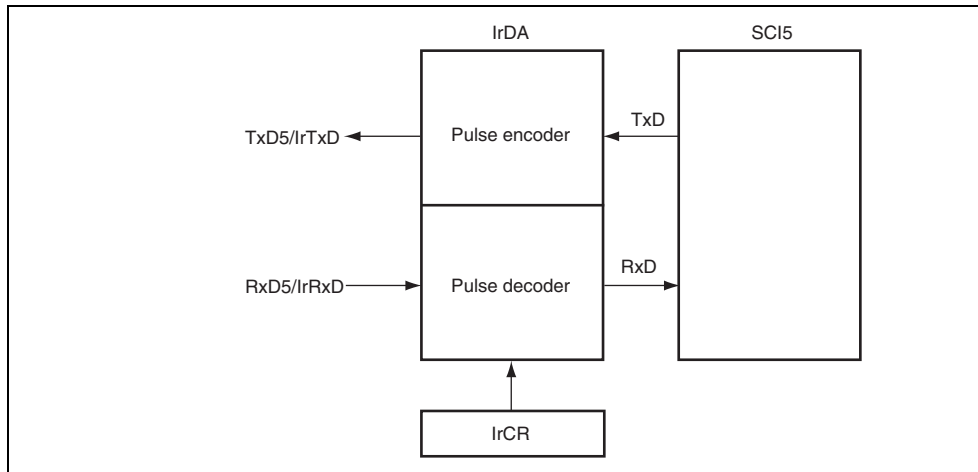


Figure 19.36 IrDA Block Diagram

Note: * The IrDA function should be used when the ABCS bit in SEMR_5 is set to 0. ACS3 to ACS0 bits in SEMR_5 and SEMR_6 are set to B'0000.

time, a high level pulse width of 1.6 μs can be specified because it is the smallest value in the range greater than 1.41 μs .

For serial data of level 1, no pulses are output.

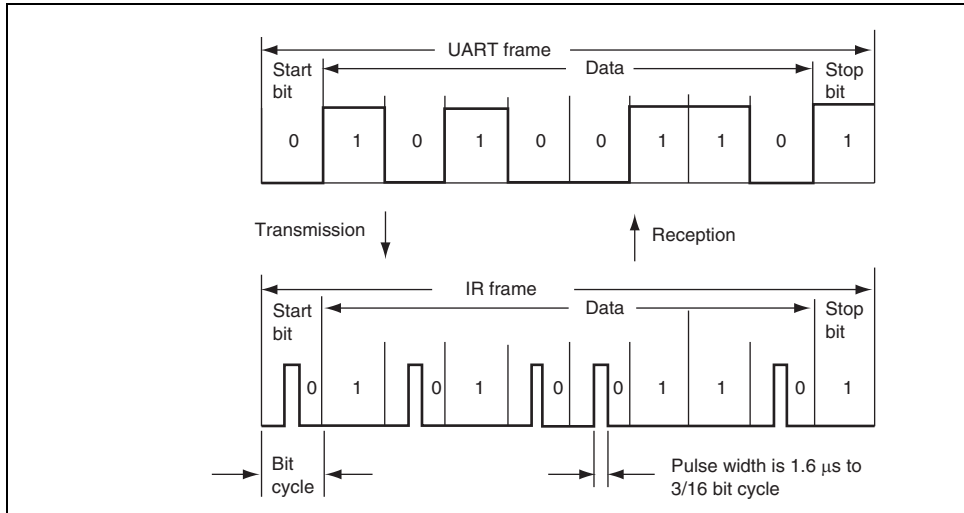


Figure 19.37 IrDA Transmission and Reception

(2) Reception

During reception, IR frames are converted to UART frames using the IrDA interface by inputting to SCI. 0 is output when the high level pulse is detected while 1 is output when a low level pulse is detected during one bit period. Note that a pulse shorter than the minimum pulse width of 1.6 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 data transfer 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 interrupt activates the DTC or DMAC to allow data transfer. The RDRF flag is automatically cleared to 0 at 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. If a TXI interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority over the TEI interrupt acceptance. However, note that if the TDRE and TEND flags are cleared to 0 simultaneously during the TXI interrupt processing routine, the SCI cannot branch to the TEI interrupt processing routine later.

Note that the priority order for interrupts is different between the group of SCI_0, 1, 2, 3, and 4 and the group of SCI_5 and SCI_6.

Table 19.14 SCI Interrupt Sources (SCI_0, 1, 2, 3, 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 19.16 shows the interrupt sources in smart card interface mode. A transmit end (TE) 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, 3, and the group of SCI_5 and SCI_6.

Table 19.16 SCI Interrupt Sources (SCI_0, 1, 2, 3, and 4)

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	P
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	Not possible	H
RXI	Receive data full	RDRF	Possible	Possible	L
TXI	Transmit data empty	TEND	Possible	Possible	L

Table 19.17 SCI Interrupt Sources (SCI_5 and SCI_6)

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	P
RXI	Receive data full	RDRF	Not possible	Possible	H
TXI	Transmit data empty	TDRE	Not possible	Possible	L
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	Not possible	L

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 12, Data Controller (DTC) and section 10, DMA Controller (DMAC).

In reception, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. This flag activates the DTC or DMAC by an RXI request thus allowing transfer of receive data if the RXI interrupt request is specified as a source of DTC or DMAC activation beforehand. The RDRF flag is automatically cleared to 0 at data transfer by the DTC or DMAC. If an error occurs, the RDRF flag is not set but the error flag is set. Therefore, the DTC or DMAC is not activated and an 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 pin directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag is set. The PER flag may also be set. Note that, since the SCI continues the receive operation even when receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

19.10.3 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 output) and level are determined by DR and DDR. This can be used to set the TxD pin to mark state (input level) or send a break during serial data transmission. To maintain the communication line in mark state (the state of 1) until TE is set to 1, set both DDR and DR to 1. Since the TE bit is cleared at this point, the TxD pin becomes an I/O port, and 1 is output from the TxD pin. To send a break during serial transmission, first set DDR to 1 and DR to 0, and then clear the TE bit to 0. When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state. The TxD pin becomes an I/O port, and 0 is output from the TxD pin.

19.10.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode)

Transmission cannot be started when a receive error flag (ORER, FER, or RER) is set to 1. 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 REB flag is cleared to 0.

- When the external clock source is used as a synchronization clock, update TDR by the TXR or DTC and wait for at least five Pφ clock cycles before allowing the transmit clock input. If the transmit clock is input within four clock cycles after TDR modification, the transmitter may malfunction (see figure 19.38).
- When using the DMAC or DTC to read RDR, be sure to set the receive end interrupt source to the DTC or DMAC activation source.

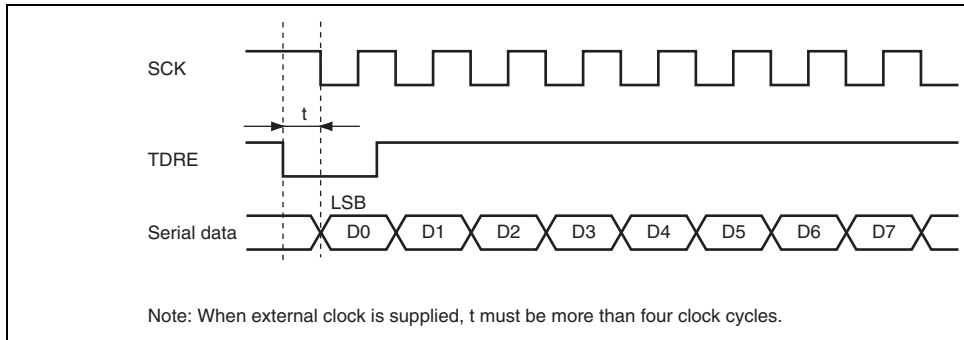


Figure 19.38 Sample Transmission using DTC in Clock Synchronous Mode

- The DTC is not activated by the RXI or TXI request by SCI_5 or SCI6.

Figure 19.39 shows a sample flowchart for transition to software standby mode during transmission. Figures 19.40 and 19.41 show the port pin states during transition to software standby mode.

Before specifying the module stop state or making a transition to software standby mode during transmission mode using DTC transfer, stop all transmit operations ($TE = TIE = TEIE = 0$). Setting the TE and TIE bits to 1 after cancellation sets the TXI flag to start transmission using DTC.

Reception: Before specifying the module stop state or making a transition to software standby mode, stop the receive operations ($RE = 0$). RSR, RDR, and SSR are reset. If transition is made 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, set the RE bit to 1, and then start reception. To receive data in a different reception mode, initialize the mode first.

For using the IrDA function, set the IrE bit in addition to setting the RE bit.

Figure 19.42 shows a sample flowchart for mode transition during reception.

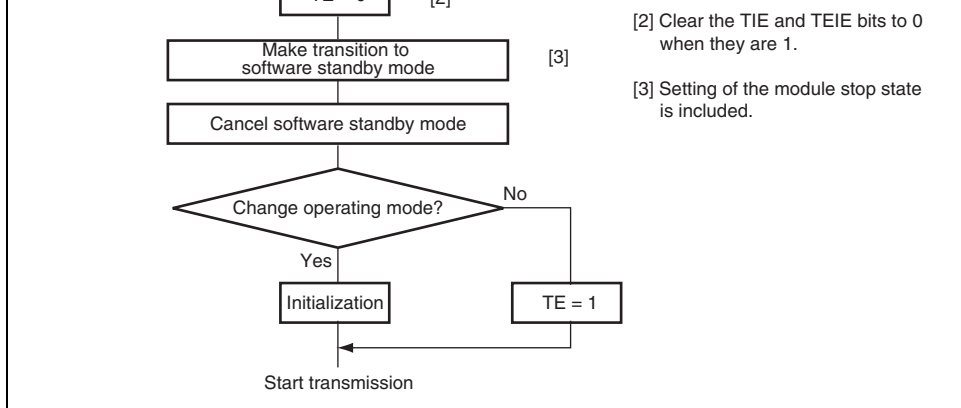


Figure 19.39 Sample Flowchart for Software Standby Mode Transition during Transmission

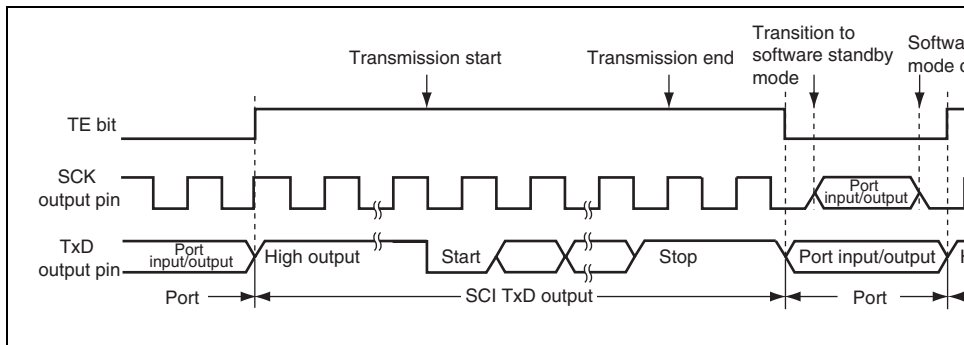


Figure 19.40 Port Pin States during Software Standby Mode Transition (Internal Clock, Asynchronous Transmission)

**Figure 19.41 Port Pin States during Software Standby Mode Transition
(Internal Clock, Clock Synchronous Transmission)**

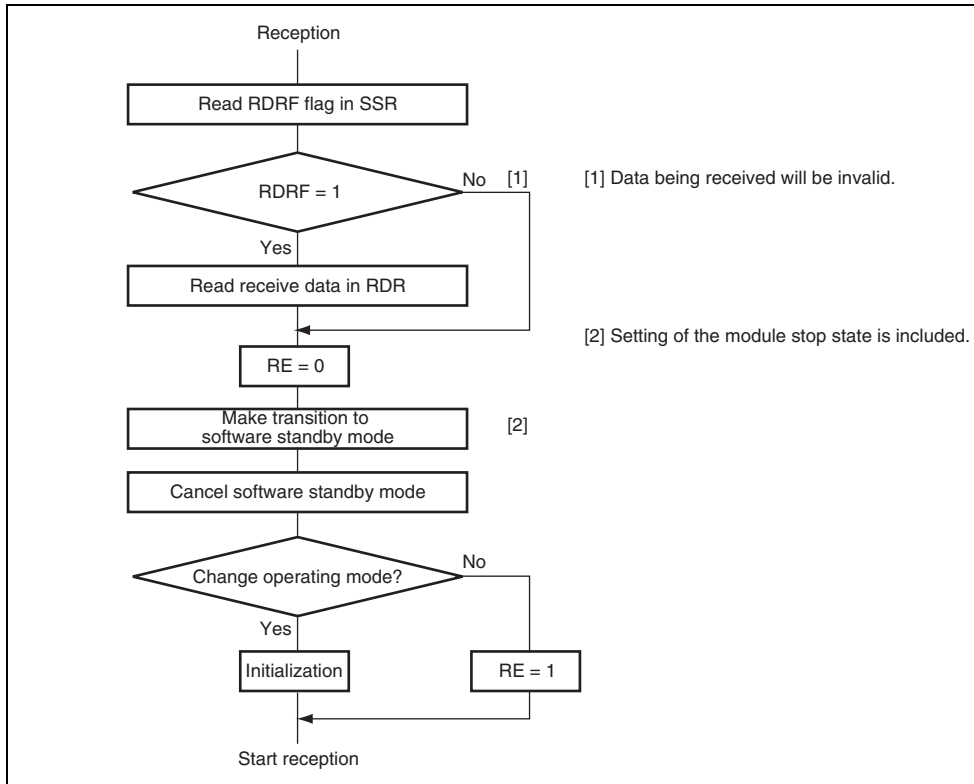


Figure 19.42 Sample Flowchart for Software Standby Mode Transition during Reception

- One of three generating polynomials selectable
- CRC code generation for LSB-first or MSB-first Communication selectable

Figure 19.43 shows a block diagram of the CRC operation circuit.

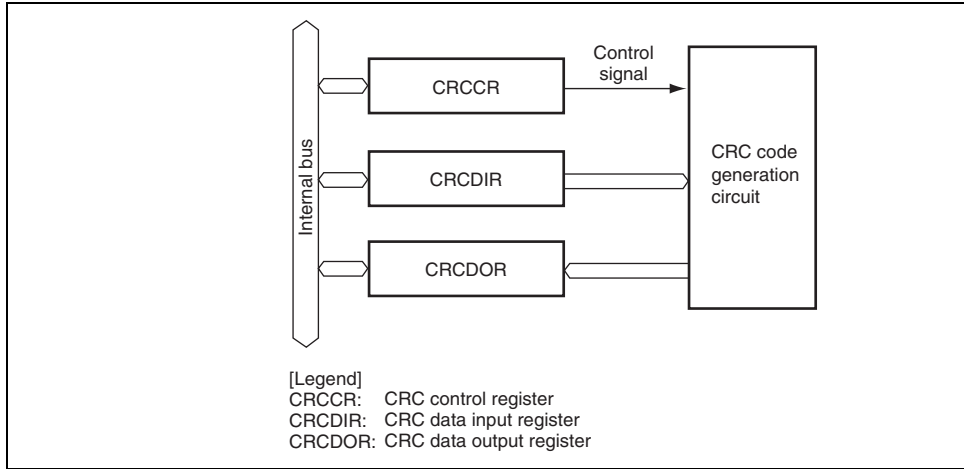


Figure 19.43 Block Diagram of CRC Operation Circuit

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'0000.
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 MSB-first 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 8) 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 written. 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. The bytes to be CRC-operated are written to CRCDIR after CRCDOR is cleared. When the CRC operation result is additionally written to the bytes to which CRC operation is to be performed, the CRC operation result will be H'0000 if the data contains no CRC error. When bits 1 and 0 of CRCCR (G1 and G0 bits) are set to 0 and 1, respectively, the lower byte of this register contains 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



3. Read from CRCDOR
CRC code = H'F78F

4. Serial transmission (LSB first)

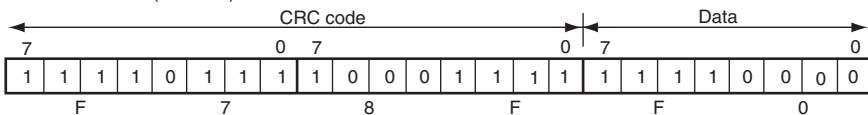
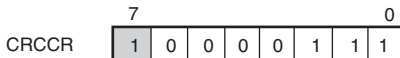
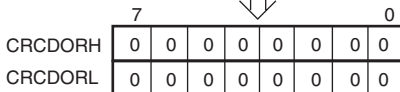


Figure 19.44 LSB-First Data Transmission

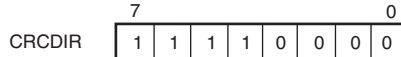
1. Write H'87 to CRCCR



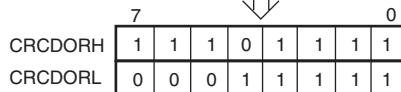
CRCDOR clearing



2. Write H'F0 to CRCDIR



CRC code generation



3. Read from CRCDOR
CRC code = H'EF1F

4. Serial transmission (MSB first)

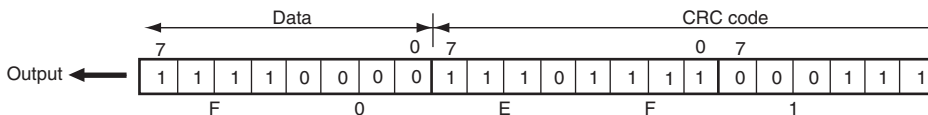



Figure 19.45 MSB-First Data Transmission

CRCDORH	7	0	0	0	0	0	0	0	0	0
CRCDORL	0	0	0	0	0	0	0	0	0	0

CRCDORH	7	1	1	1	1	0	1	1	1	1
CRCDORL	1	0	0	0	0	1	1	1	1	1

4. Write H'8F to CRCDIR


CRCDIR	7	1	0	0	0	1	1	1	1	0
--------	---	---	---	---	---	---	---	---	---	---

CRC code generation


CRCDORH	7	0	0	0	0	0	0	0	0	0
CRCDORL	1	1	1	1	0	1	1	1	1	0

5. Write H'F7 to CRCDIR

CRCDIR	7	1	1	1	1	0	1	1	1	0
--------	---	---	---	---	---	---	---	---	---	---

CRC code generation


CRCDORH	7	0	0	0	0	0	0	0	0	0
CRCDORL	0	0	0	0	0	0	0	0	0	0

6. Read from CRCDOR

CRC code = H'0000 → No error

Figure 19.46 LSB-First Data Reception

CRCDORH	7	0	0	0	0	0	0	0	0	0
CRCDORL	0	0	0	0	0	0	0	0	0	0

CRCDORH	7	1	1	1	0	1	1	1	1	1
CRCDORL	0	0	0	1	1	1	1	1	1	1

4. Write H'EF to CRCDIR

CRCDIR	7	1	1	1	0	1	1	1	1	0
--------	---	---	---	---	---	---	---	---	---	---

↓ CRC code generation

CRCDORH	7	0	0	0	1	1	1	1	1	0
CRCDORL	0	0	0	0	0	0	0	0	0	0

5. Write H'1F to CRCDIR

CRCDIR	7	0	0	0	1	1	1	1	1	0
--------	---	---	---	---	---	---	---	---	---	---

↓ CRC code generation

CRCDORH	7	0	0	0	0	0	0	0	0	0
CRCDORL	0	0	0	0	0	0	0	0	0	0

6. Read from CRCDOR

CRC code = H'0000 → No error

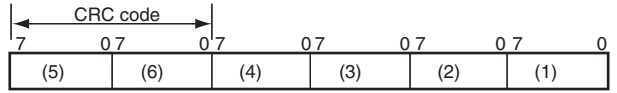
Figure 19.47 MSB-First Data Reception



CRCBORH	(5)
CRCBORL	(6)

2. Transmission data

(i) LSB-first transmission



(ii) MSB-first transmission

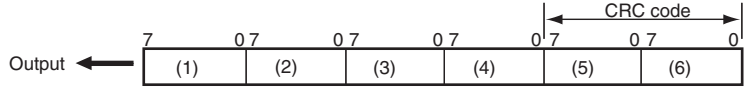


Figure 19.48 LSB-First and MSB-First Transmit Data

20.1 Features

- Continuous transmission/reception
Since the shift register, transmit data register, and receive data register are independent of 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 the serial clock (SCL) is monitored per bit, and the timing is synchronized automatically. If transmission or reception is not yet possible, drive the SCL signal low until preparations are completed
- Six interrupt sources
Transmit-data-empty (including slave-address match), transmit-end, receive-data-full (including slave-address match), arbitration lost, NACK detection, and stop condition detection
- Direct bus drive
Two pins, the SCL and serial data (SDA) pins function as NMOS open-drain outputs. A voltage of 5V can be input to unit 1.
- Module stop function setting

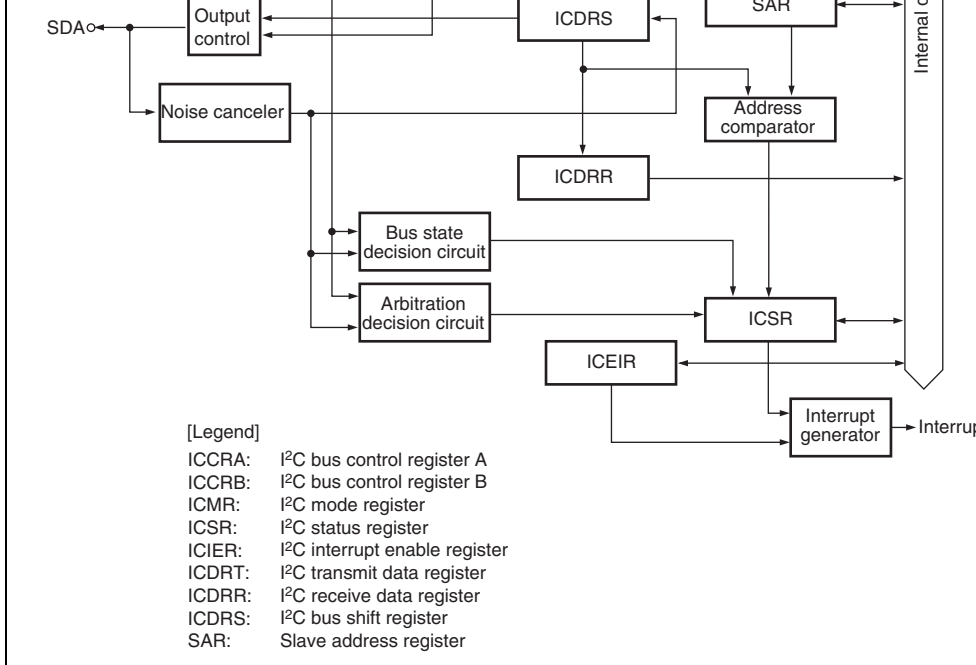


Figure 20.1 Block Diagram of I²C Bus Interface 2

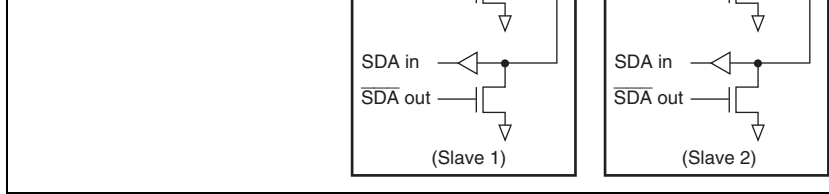


Figure 20.2 Connections to the External Circuit by the I/O Pins

20.2 Input/Output Pins

Table 20.1 shows the pin configuration of the I²C bus interface 2.

Table 20.1 Pin Configuration of the I²C 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
1	SCL2	I/O	Channel 2 serial clock I/O pin
	SDA2	I/O	Channel 2 serial data I/O pin
3	SCL3	I/O	Channel 3 serial clock I/O pin
	SDA3	I/O	Channel 3 serial data I/O pin

Note: The pin symbols are represented as SCL and SDA; channel numbers are omitted in the manual.

- I²C bus interrupt enable register_0 (ICIER_0)
 - I²C bus status register_0 (ICSR_0)
 - Slave address register_0 (SAR_0)
 - I²C bus transmit data register_0 (ICDRT_0)
 - I²C bus receive data register_0 (ICDRR_0)
 - I²C bus shift register_0 (ICDRS_0)
- Channel 1:
 - I²C bus control register A_1 (ICCRA_1)
 - I²C bus control register B_1 (ICCRB_1)
 - I²C bus mode register_1 (ICMR_1)
 - I²C bus interrupt enable register_1 (ICIER_1)
 - I²C bus status register_1 (ICSR_1)
 - Slave address register_1 (SAR_1)
 - I²C bus transmit data register_1 (ICDRT_1)
 - I²C bus receive data register_1 (ICDRR_1)
 - I²C bus shift register_1 (ICDRS_1)

Unit 1:

- Channel 2:
 - I²C bus control register A_2 (ICCRA_2)
 - I²C bus control register B_2 (ICCRB_2)
 - I²C bus mode register_2 (ICMR_2)
 - I²C bus interrupt enable register_2 (ICIER_2)
 - I²C bus status register_2 (ICSR_2)
 - Slave address register_2 (SAR_2)

- Slave address register_3 (SAR_3)
- I²C bus transmit data register_3 (ICDRT_3)
- I²C bus receive data register_3 (ICDRR_3)
- I²C bus shift register_3 (ICDRS_3)

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I ² C 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 operation. 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, MST and TRS are both reset by hardware, causing a transition to slave receive mode. Modification of the TRS should be made between transfer frames. Operating modes are described below according to 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. Modification setting according to the required transfer rate details on the transfer rate, see table 20.2.
1	CKS1	0	R/W	
0	CKS0	0	R/W	

			1	P ϕ /100	80.0 kHz	100 kHz	200 kHz	250 kHz	330 kHz	
			1	0	P ϕ /112	71.4 kHz	89.3 kHz	179 kHz	223 kHz	295 kHz
				1	P ϕ /128	62.5 kHz	78.1 kHz	156 kHz	195 kHz	258 kHz
1	0	0	0	P ϕ /56	143 kHz	179 kHz	357 kHz	446 kHz	589 kHz	
				1	P ϕ /80	100 kHz	125 kHz	250 kHz	313 kHz	413 kHz
			1	0	P ϕ /96	83.3 kHz	104 kHz	208 kHz	260 kHz	344 kHz
				1	P ϕ /128	62.5 kHz	78.1 kHz	156 kHz	195 kHz	258 kHz
	1	0	0	P ϕ /336	23.8 kHz	29.8 kHz	59.5 kHz	74.4 kHz	98.2 kHz	
				1	P ϕ /200	40.0 kHz	50.0 kHz	100 kHz	125 kHz	165 kHz
			1	0	P ϕ /224	35.7 kHz	44.6 kHz	89.3 kHz	112 kHz	147 kHz
				1	P ϕ /256	31.3 kHz	39.1 kHz	78.1 kHz	97.7 kHz	129 kHz

20.3.2 I²C Bus Control Register B (ICCRB)

ICCRB issues start/stop condition, manipulates the SDA pin, monitors the SCL pin, and 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 condition, write 1 to the MOV instruction.

6	SCP	1	R/W	<p>Start/Stop Condition Issue</p> <p>This bit controls the issuance of start or stop condition in master mode.</p> <p>To issue a start condition, write 1 to BBSY and SCP. A re-transmit start condition is issued in the same way. To issue a stop condition, write 0 to BBSY and SCP. This bit is always read as 1. If 1 is written, the data is not stored.</p>
5	SDAO	1	R	<p>This bit monitors the output level of SDA.</p> <p>0: When reading, the SDA pin outputs a low level. 1: When reading the SDA pin outputs a high level.</p>
4	—	1	R/W	<p>Reserved</p> <p>The write value should always be 1.</p>
3	SCLO	1	R	<p>This bit monitors the SCL output level.</p> <p>When reading and SCLO is 1, the SCL pin outputs a high level. When reading and SCLO is 0, the SCL pin outputs a low level.</p>
2	—	1	—	<p>Reserved</p> <p>This bit is always read as 0.</p>
1	IICRST	0	R/W	<p>IIC Control Module Reset</p> <p>This bit resets the IIC control module except the IIC registers. If hang-up occurs because of communication failure during I²C operation, by setting this bit to 1, the I²C control module can be reset without initializing the IIC registers.</p>
0	—	1	—	<p>Reserved</p> <p>This bit is always read as 1.</p>

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 a transfer except for the acknowledge bit. When set to 1, after the falling of the clock for the last transfer, the low period is extended for two transfer clocks. When this bit is cleared to 0, data and the acknowledge bit are transferred consecutively with no wait. 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 BC0 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 always read as 0. 1: When reading, 1 is always read. When writing, the settings of BC2 to BC0 are always 0.

001: 2
 010: 3
 011: 4
 100: 5
 101: 6
 110: 7
 111: 8

I²C control module can be reset without setting ports and initializing the registers.

20.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER enables or disables interrupt sources and the acknowledge bits, sets the acknowledge bits to 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 interrupt (TEI) request at the rising of the ninth clock with the TDRE bit in ICSR is set to 1. The TEI request is canceled by clearing the TEND bit or the TEIEN bit.
 0: Transmit end interrupt (TEI) request is disabled.
 1: Transmit end interrupt (TEI) request is enabled.

5	RIE	0	R/W	<p>Receive Interrupt Enable</p> <p>This bit enables or disables the receive full interrupt (RXI) request when receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. The RXI request can be canceled by clearing the RDRF or RIE bit to 0.</p> <p>0: Receive data full interrupt (RXI) request is disabled. 1: Receive data full interrupt (RXI) request is enabled.</p>
4	NAKIE	0	R/W	<p>NACK Receive Interrupt Enable</p> <p>This bit enables or disables the NACK receive interrupt (NAKI) request when the NACKF and AL bits are set to 1. The NAKI request can be canceled by clearing the NACKF or AL bit, or the NAKIE bit to 0.</p> <p>0: NACK receive interrupt (NAKI) request is disabled. 1: NACK receive interrupt (NAKI) request is enabled.</p>

1	ACKBR	0	R	<p>Receive Acknowledge</p> <p>In transmit mode, this bit stores the acknowledge that are returned by the receive device. This bit can be modified.</p> <p>0: Receive acknowledge = 0</p> <p>1: Receive acknowledge = 1</p>
0	ACKBT	0	R/W	<p>Transmit Acknowledge</p> <p>In receive mode, this bit specifies the bit to be used for the acknowledge timing.</p> <p>0: 0 is sent at the acknowledge timing</p> <p>1: 1 is sent at the acknowledge timing</p>

Bit	Bit Name	Value	R/W	Description
7	TDRE	0	R/W	<p>Transmit Data Register Empty</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> • When data is transferred from ICDRT to ICDRT and ICDRT becomes empty • When the TRS bits are set • When the start (re-transmit included) command has been issued • When switched from reception to transmission slave mode <p>[Clearing conditions]</p> <ul style="list-style-type: none"> • When 0 is written to this bit after reading TEND (When the CPU is used to clear this flag bit to 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.) • When data is written to ICDRT
6	TEND	0	R/W	<p>Transmit End</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> • When the ninth clock of SCL rises while the flag is 1 <p>[Clearing conditions]</p> <ul style="list-style-type: none"> • When 0 is written to this bit after reading TEND (When the CPU is used to clear this flag bit to 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.) • When data is written to ICDRT

4	NACKF	0	R/W	<ul style="list-style-type: none"> When data is read from ICDRR <p>No Acknowledge Detection Flag</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> When no acknowledge is detected from the device in transmission while the ACKE bit is set to 1 <p>[Clearing condition]</p> <ul style="list-style-type: none"> When 0 is written to this bit after reading NACKF = 1 <p>(When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)</p>
3	STOP	0	R/W	<p>Stop Condition Detection Flag</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> When a stop condition is detected after frame transfer <p>[Clearing condition]</p> <ul style="list-style-type: none"> When 0 is written to this bit after reading STOP = 1 <p>(When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)</p>

disagree at the rising of SCL in master transmit mode

- When the SDA pin outputs a high level in master transmit mode while a start condition is detected

[Clearing condition]

- When 0 is written to this bit after reading AAS (When the CPU is used to clear this flag bit, write 0 while the corresponding interrupt is enabled. Be 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 when a data frame following a start condition matches bits SVA0 in SAR. [Setting conditions] <ul style="list-style-type: none">• When the slave address is detected in slave receive mode• When the general call address is detected in slave receive mode [Clearing condition] <ul style="list-style-type: none">• When 0 is written to this bit after reading AAS
---	-----	---	-----	---

20.3.6 Slave Address Register (SAR)

SAR sets the slave address. In slave mode, if the upper 7 bits of SAR match the upper 7 bits of 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 the addresses of other slave devices connected to the bus.
0	—	0	R/W	Reserved Although this bit is readable/writable, only 0 should be written to.

20.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit read-only register that stores the receive data. When one byte of data is received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register; therefore, this register 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

20.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is an 8-bit write-only register that is used to transmit/receive data. In transmission, data is transferred from ICDSRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after one byte of data is received. This register cannot be read 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

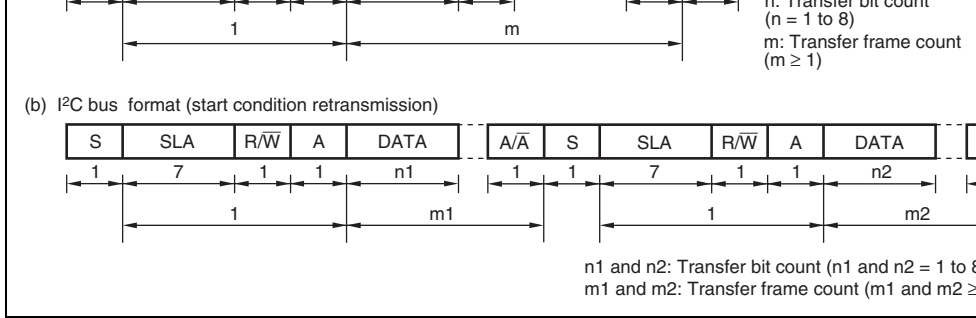


Figure 20.3 I²C Bus Formats

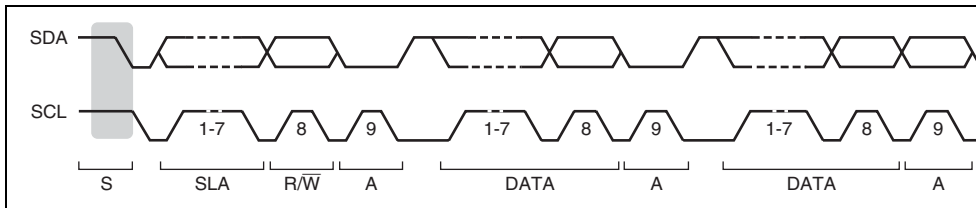


Figure 20.4 I²C Bus Timing

[Legend]

- S: Start condition. The master device drives SDA from high to low while SCL is high.
- SLA: Slave address
- R/W: Indicates the direction of data transfer; from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/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 high.

- 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 byte of the slave address and R/W) to ICDRT. After this, when TDRE is automatically cleared, the 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 set at the rising of the ninth transmit clock pulse. Read the ACKBR bit in ICIER to confirm that the slave device has been selected. Then, write the second byte data to ICDRT. When TDRE is 1, the slave device has not been acknowledged, so issue a stop condition. To issue a stop condition, write 0 to BBSY and SCP using the MOV instruction. SCL is fixed to a low level 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 end of the byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR is 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TDRE and NACKF.
 7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode.

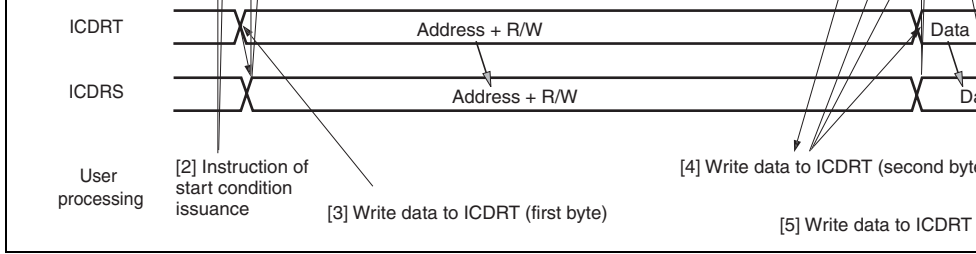


Figure 20.5 Master Transmit Mode Operation Timing 1

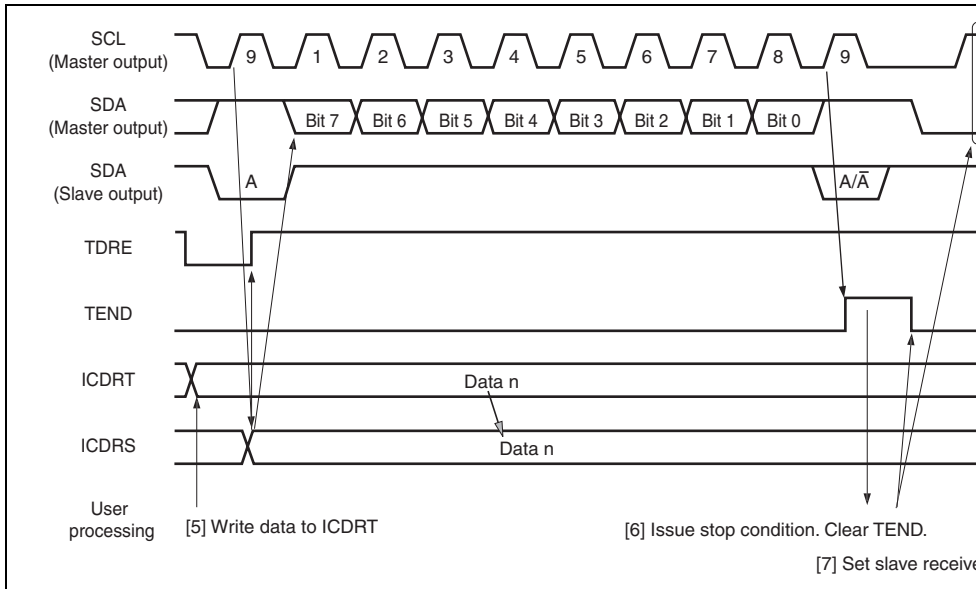


Figure 20.6 Master Transmit Mode Operation Timing 2

- data is received, in synchronization with the internal clock. The master mode output 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 at the rising of the ninth receive clock pulse. At this time, the received data is read by reading ICDRR. At the same time, RDRF is cleared.
 4. The continuous reception is performed by reading ICDRR and clearing RDRF to 0. When RDRF is set, if the eighth receive clock pulse falls after reading ICDRR by other process 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 ICCRA before reading ICDRR. 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 condition 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.

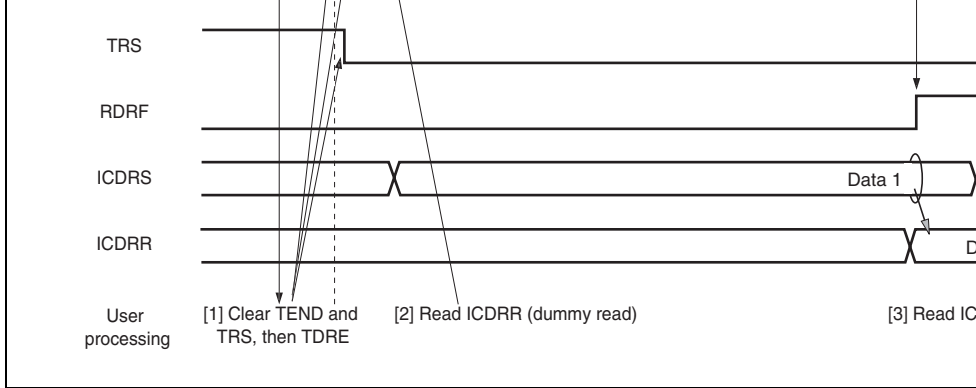


Figure 20.7 Master Receive Mode Operation Timing 1

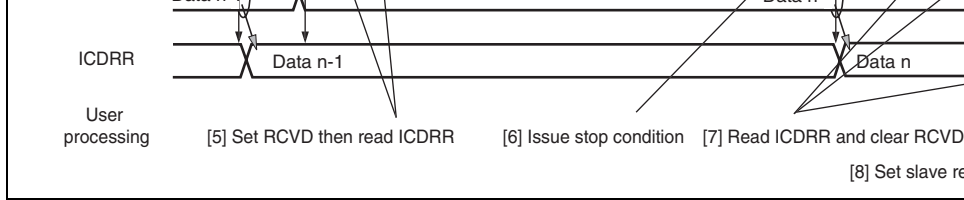


Figure 20.8 Master Receive Mode Operation Timing 2

20.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, and the master device outputs the receive clock pulse and returns an acknowledge signal. Figures 17.9 and 17.10 show the operation timings in slave transmit mode. The transmission procedure and operations in slave 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, and set the WAIT in ICMR, CKS3 to CKS0 in ICCRA, and others to perform initial settings. Set the MST and TRS bits in ICCRA to 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, on the rising of the ninth clock pulse. At this time, if the eighth bit data (R/W) is 1, TRS in ICSR and TDRE in ICSR are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by writing the transmit data to ICDRT every time TDRE is set.
3. If TDRE is set after writing the last transmit data to ICDRT, wait until TEND in ICSR is set to 1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for end processing, and read ICDRR (dummy read) to free SCL.

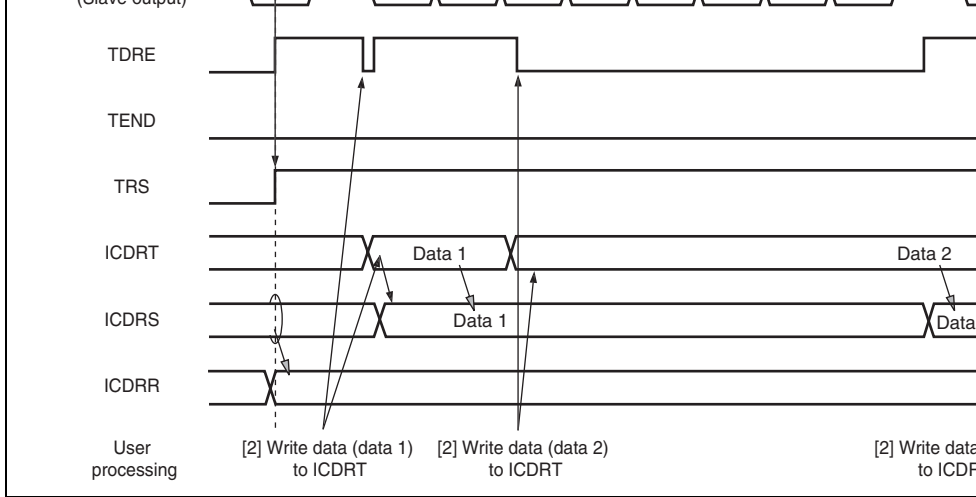


Figure 20.9 Slave Transmit Mode Operation Timing 1

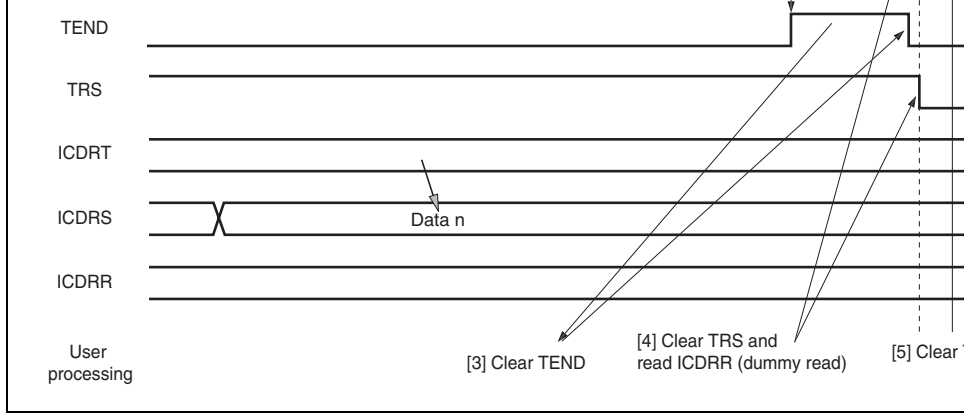


Figure 20.10 Slave Transmit Mode Operation Timing 2

2. When the slave address matches in the first frame following detection of the start condition, the slave address outputs the level specified by ACKBT in ICIER to SDA, at the rising edge of the ninth clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (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 1, the slave address is fixed to a low level until ICDRR is read. The change of the acknowledge (ACKBT) signal before reading ICDRR to be returned to the master device is reflected in the next transmission frame.
4. The last byte data is read by reading ICDRR.

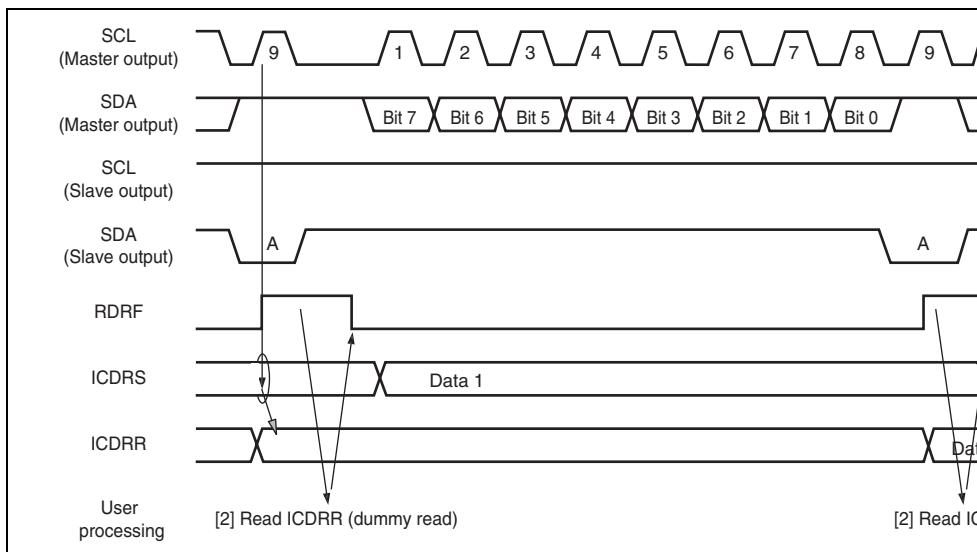


Figure 20.11 Slave Receive Mode Operation Timing 1

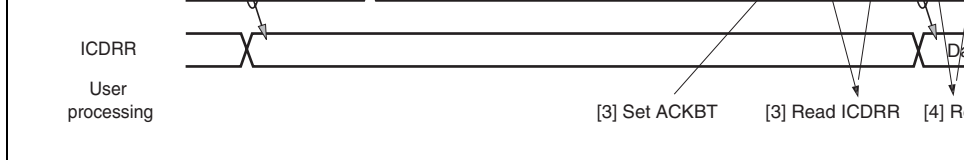


Figure 20.12 Slave Receive Mode Operation Timing 2

20.4.6 Noise Canceller

The logic levels at the SCL and SDA pins are routed through the noise cancellers before being latched internally. Figure 20.13 shows a block diagram of the noise canceller circuit.

The noise canceller consists of two cascaded latches and a match detector. The signal on SCL (or SDA) is sampled on the system clock, but is not passed forward to the next circuit until the outputs of both latches agree. If they do not agree, the previous value is held.

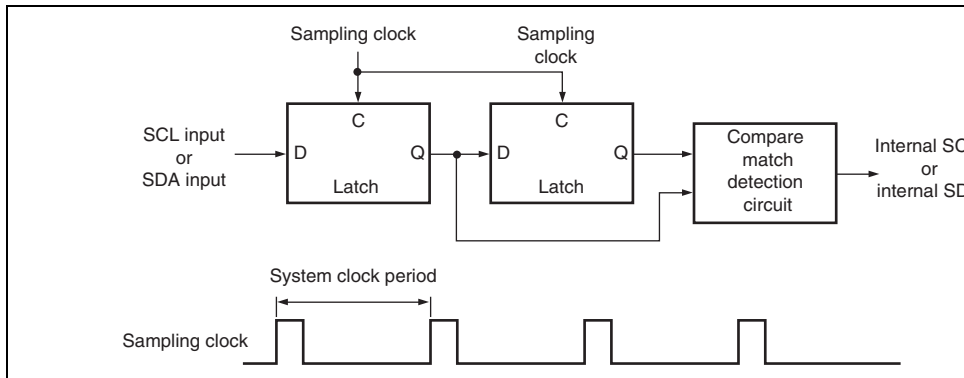


Figure 20.13 Block Diagram of Noise Canceller

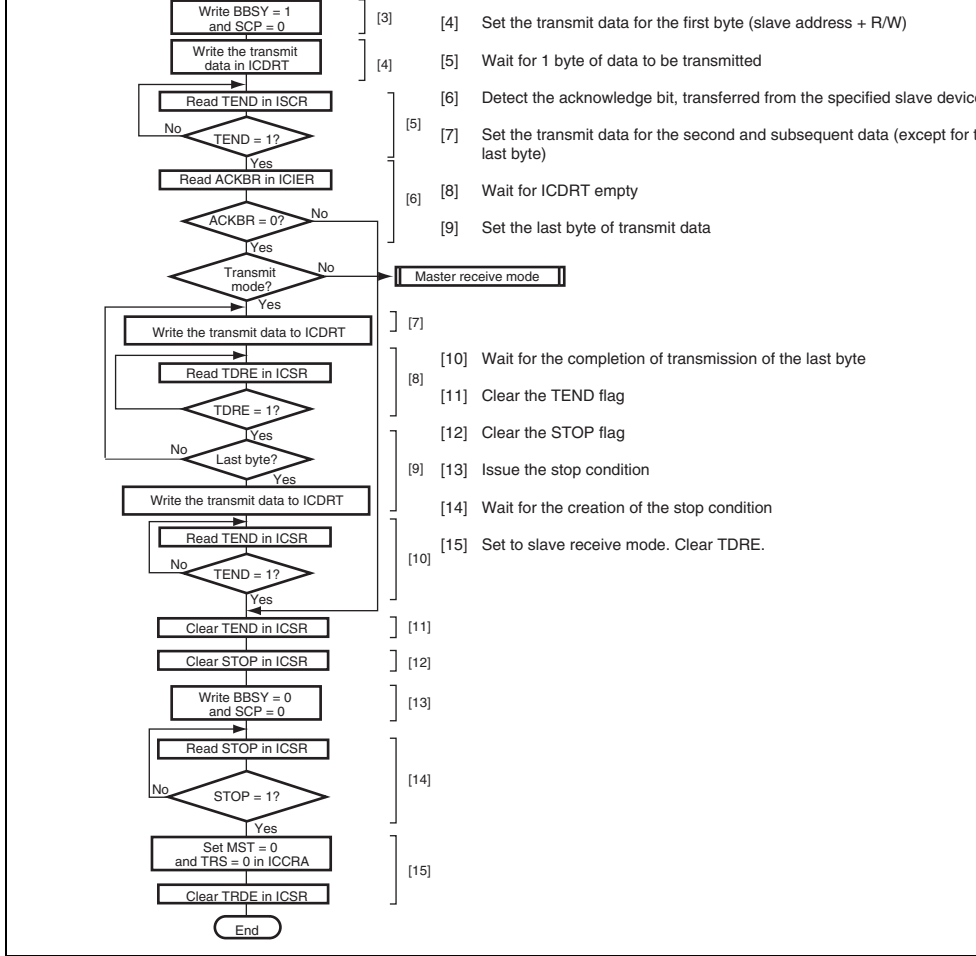
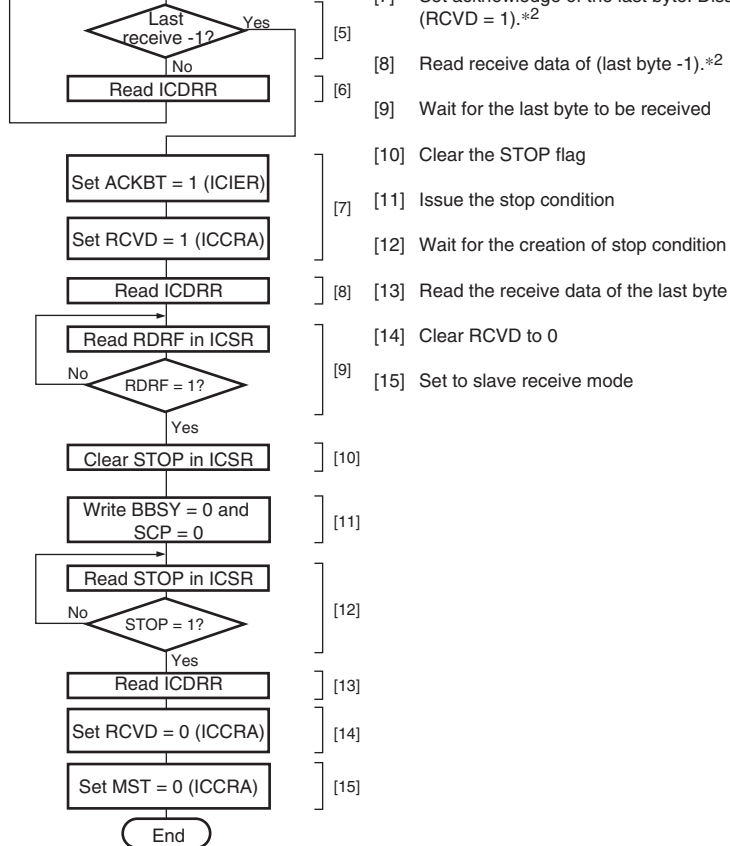


Figure 20.14 Sample Flowchart of Master Transmit Mode



- 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 20.15 Sample Flowchart for Master Receive Mode

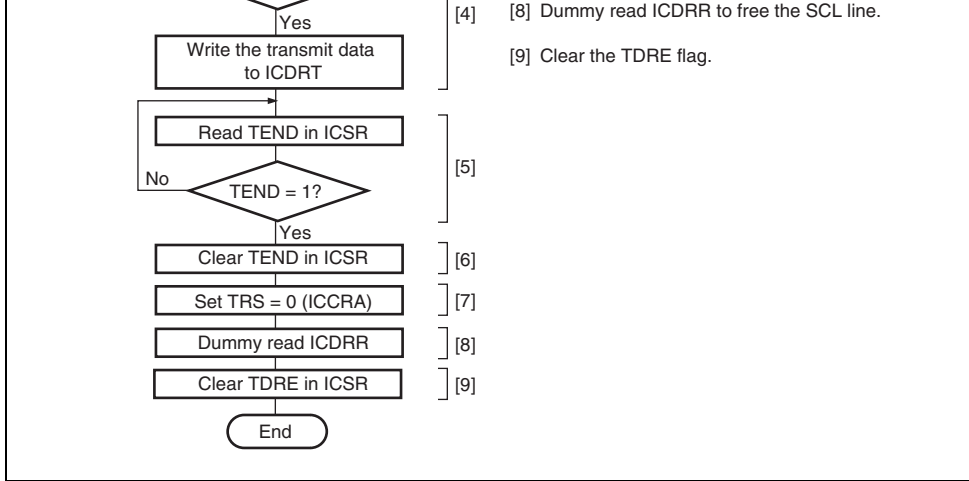
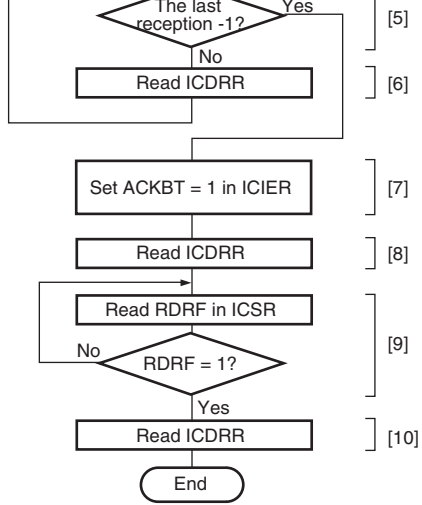


Figure 20.16 Sample Flowchart for Slave Transmit Mode



[5] [8] Read the receive data of (last byte -1).*

[6] [9] Wait for the reception of the last byte to be completed.

[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 20.17 Sample Flowchart for Slave Receive Mode

Receive data full	RXI	$(RDRF = 1) \cdot (RIE = 1)$
Stop recognition	STPI	$(STOP = 1) \cdot (STIE = 1)$
NACK detection	NAKI	$\{(NACKF = 1) + (AL = 1)\} \cdot (NAKIE = 1)$
Arbitration lost		

When one of the interrupt conditions in table 20.3 is 1 and the I bit in CCR is 0, the CPU interrupt exception handling. Clear the interrupt sources during interrupt exception handling that the TDRE and TEND bits are automatically cleared to 0 by writing data to ICDRT, and RDRF bit is cleared to 0 by reading ICDRR. In particular, the TDRE bit can be set again at the same time as data are for transmission written to ICDRT, and 1 extra byte can be transmitted. TDRE is again cleared to 0.

20.6 Bit Synchronous Circuit

This module has a possibility that the high-level period is shortened in the two states described 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 capacitance and pull-up resistance)

Therefore, this module monitors SCL and communicates bit by bit in synchronization.

Figure 20.18 shows the timing of the bit synchronous circuit, and table 20.4 shows the timing of monitoring SCL, i.e., a period from SCL's departure from low-level output and its entry to high-impedance state until monitoring SCL.

Table 20.4 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL
0	0	7.5 t _{cyc}
	1	19.5 t _{cyc}
1	0	17.5 t _{cyc}
	1	41.5 t _{cyc}

20.7 Usage Notes

1. Confirm the ninth falling edge of the clock cycle before issuing a stop or a repeated start condition.

The ninth falling edge can be confirmed by monitoring the SCLO bit in the I²C bus control register B (ICCRB).

If a stop or a repeated start condition is issued at certain timing in either of the following cases, the stop or repeated start condition may be issued incorrectly.

- The rising time of the SCL signal exceeds the time given in section 20.6, Bit Synchronous 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 low during the eighth clock.

2. The WAIT bit in the I²C bus mode register (ICMR) must be held at 0.

If the WAIT bit is set to 1, when a slave device holds the SCL signal low more than one transfer clock cycle during the eighth clock, the high level period of the ninth clock cycle is shorter than a given period.

3. Restriction in transfer rate setting value in multi-master mode

- In multi-master mode, set the MST and TRS bits by MOV instruction.
- When arbitration is lost, confirm that the MST and TRS bits are set to 0. If these bits are set to other than 0, set these bits to 0.

5. Notes on master receive mode

In master receive mode, when the RDRF bit is set to 1 at the eighth rising clock, the SDA signal is pulled to "Low" state. When ICDRR is read near at the eighth falling clock, the SDA signal level is released and the ninth clock is output by fixing the eighth clock of receive to "Low" state. Reading ICDRR is not required. As a result, the failure to receive data occurs. There are the following methods to avoid this phenomenon.

- In master receive mode, read ICDRR by the eighth rising clock.
- In master receive mode, set the RCVD bit to 1 and process the bit by the communication every one byte.

6. Module stop function setting

Operation of the IIC2 can be disabled or enabled using the module stop control register. The initial setting is for operation of the IIC2 to be halted. Register access is enabled by clearing the module stop state. For details, see section 27, Power-Down Modes.

21.1 Features

- 10-bit resolution
- Eight or four input channels (total eight input channels for the two units)
Four channels x three units (for units 0, 1, and 2)
Eight channels x one unit (for unit 0) + four channels x one unit (unit 2)
- Conversion time: Unit 0: (2.7 ms per channel)
 Units 1 and 2: (2.7 ms per channel)
- 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*¹
- Eight data registers for the A/D converter unit 0 and four data registers for units 1 and 2 (total twelve data registers for the three units)
Results of A/D conversion are held in a 16-bit data register for each channel.
- Sample and hold functionality
- Three types of conversion start
Conversion can be started by software, a conversion start trigger by the 16-bit timer (TPU)*¹ or 8-bit timer (TMR)*², or an external trigger signal.
- Function of starting units simultaneously
A/D conversion for multiple units can be started by external trigger (ADTRG0).
- Interrupt source
A/D conversion end interrupt (ADI) request can be generated.
- Module stop state specifiable

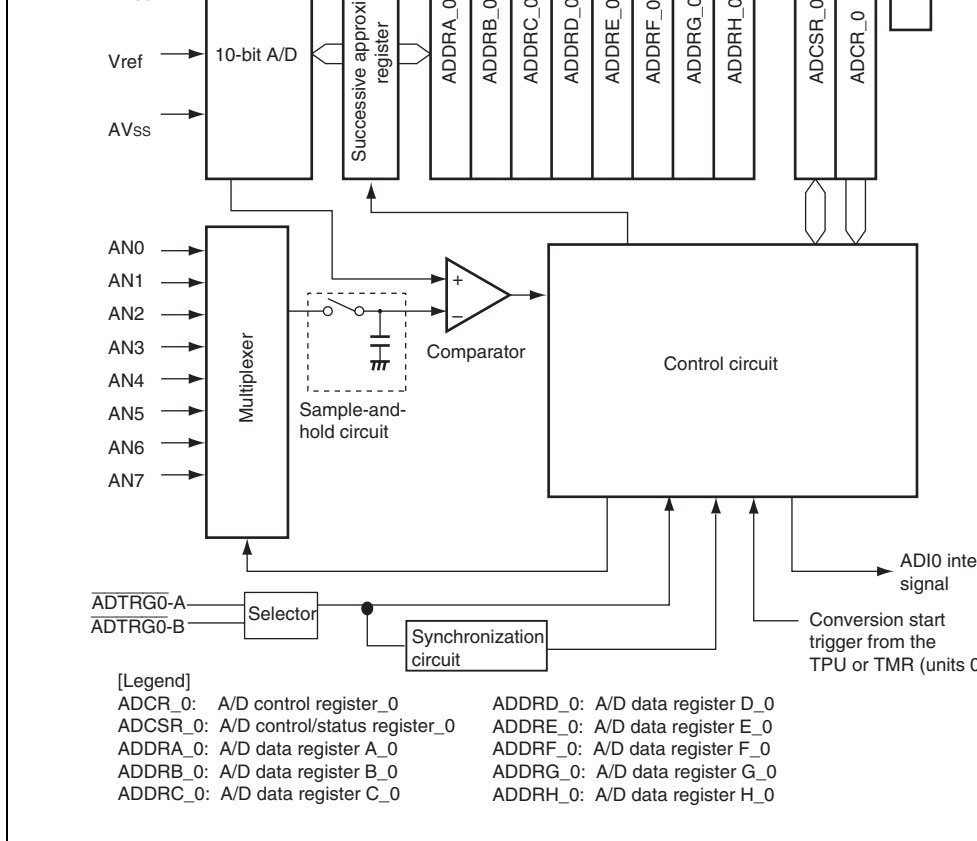


Figure 21.1 Block Diagram of A/D Converter Unit 0 (AD_0)

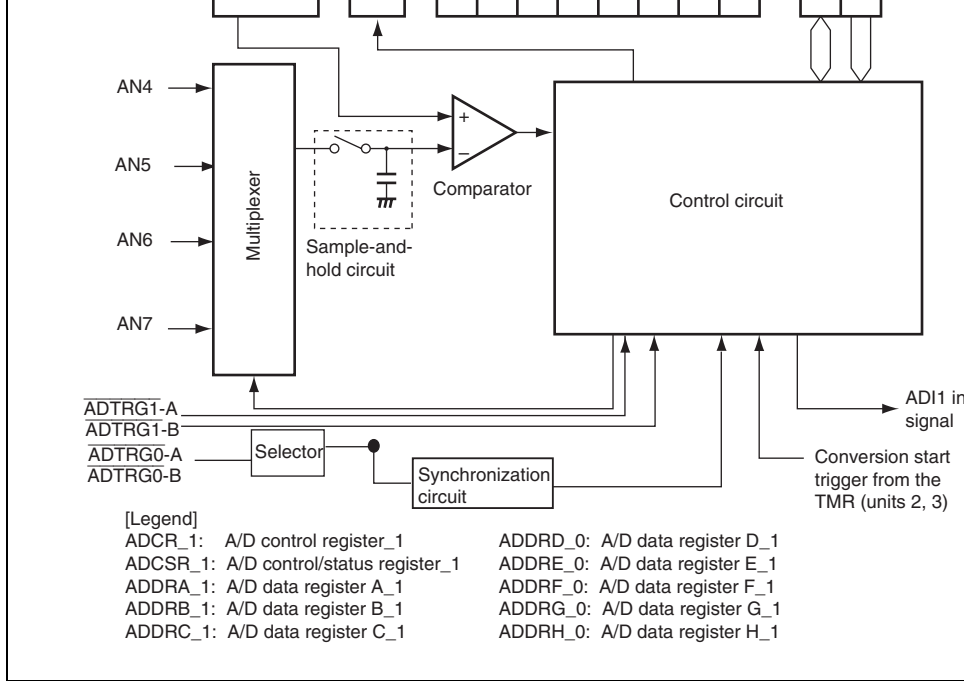


Figure 21.2 Block Diagram of A/D Converter Unit 1 (AD_1)

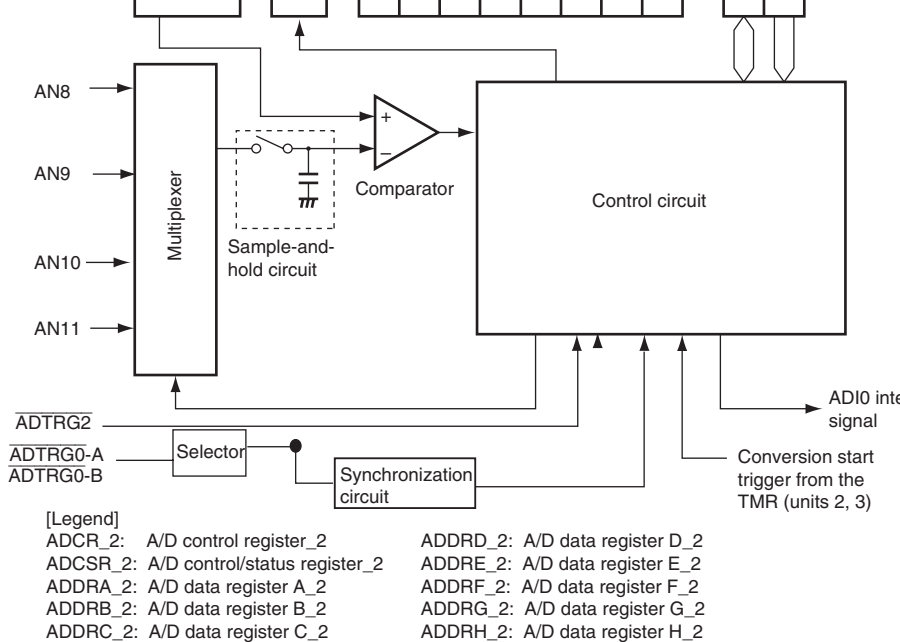


Figure 21.3 Block Diagram of A/D Converter Unit 2 (AD_2)

		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 0_A	$\overline{\text{ADTRG0-A}}$	Input	External trigger input starting A/D conversions
		A/D external trigger input pin 0_B	$\overline{\text{ADTRG0-B}}$	Input	External trigger input starting A/D conversions
1	AD_1	Analog input pin 4	AN4	Input	Analog inputs
		Analog input pin 5	AN5	Input	
		Analog input pin 6	AN6	Input	
		Analog input pin 7	AN7	Input	
		A/D external trigger input pin 1_A	$\overline{\text{ADTRG1_A}}$	Input	External trigger input starting A/D conversions
		A/D external trigger input pin 1_B	$\overline{\text{ADTRG1_B}}$	Input	External trigger input starting A/D conversions
		A/D external trigger input pin 0_A	$\overline{\text{ADTRG0-A}}$	Input	External trigger input starting A/D conversions
		A/D external trigger input pin 0_B	$\overline{\text{ADTRG0-B}}$	Input	External trigger input starting A/D conversions

Common	Analog power supply pin	AV_{CC}	Input	Analog block power s
	Analog ground pin	AV_{SS}	Input	Analog block ground
	Reference voltage pin	Vref	Input	A/D conversion refere voltage

- Note:
1. Selectable by setting of the TRGS1, TRGS0, and EXTRGS bits in ADCR.
 2. The $\overline{ADTRG0}$ input pin can be selected by the ADTRG0S bit in PFCR6. For details, see section 13, I/O Ports.

- A/D data register E_0 (ADDRE_0)
- A/D data register F_0 (ADDRF_0)
- A/D data register G_0 (ADDRG_0)
- A/D data register H_0 (ADDRH_0)
- A/D control/status register_0 (ADCSR_0)
- A/D control register_0 (ADCR_0)

Unit 1 (A/D_1) registers:

- A/D data register A_1 (ADDRA_1)
- A/D data register B_1 (ADDRB_1)
- A/D data register C_1 (ADDRC_1)
- A/D data register D_1 (ADDRD_1)
- A/D data register E_1 (ADDRE_1)
- A/D data register F_1 (ADDRF_1)
- A/D data register G_1 (ADDRG_1)
- A/D data register H_1 (ADDRH_1)
- A/D control/status register_1 (ADCSR_1)
- A/D control register_1 (ADCR_1)

- A/D control/status register_2 (ADCSR_2)
- A/D control register_2 (ADCR_2)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Bit Name										—	—	—	—	—
Initial Value	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

Table 21.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel	A/D Data Register Storing Conversion Result		
	Unit 1* ¹	Unit 1* ²	Unit 1* ³
AN0	ADDRA_0 (Unit 0)	—	—
AN1	ADDRB_0 (Unit 0)	—	—
AN2	ADDRC_0 (Unit 0)	—	—
AN3	ADDRD_0 (Unit 0)	—	—
AN4	ADDRE_0 (Unit 0)* ¹	ADDRE_1 (Unit 1)* ¹	—
AN5	ADDRF_0 (Unit 0)* ¹	ADDRF_1 (Unit 1)* ¹	—
AN6	ADDRG_0 (Unit 0)* ¹	ADDRG_1 (Unit 1)* ¹	—
AN7	ADDRH_0 (Unit 0)* ¹	ADDRH_1 (Unit 1)* ¹	—
AN8	—	—	ADDRA_2 (Unit 2)
AN9	—	—	ADDRB_2 (Unit 2)
AN10	—	—	ADDRC_2 (Unit 2)
AN11	—	—	ADDRD_2 (Unit 2)

Notes: *1 A/D conversion should be not performed on the same channel by multiple units.

*2 The ADDRA_1 to ADDRD_1 registers for unit 1 are not used.

*3 The ADDRE_2 to ADDRH_2 registers for unit 2 are not used.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	<p>A/D End Flag</p> <p>A status flag that indicates the end of A/D conversion.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> • Completion of A/D conversion in single mode • Completion of A/D conversion on all specified channels in scan mode <p>[Clearing conditions]</p> <ul style="list-style-type: none"> • Writing of 0 after reading ADF = 1 (When the CPU is used to clear this flag by software while the corresponding interrupt is enabled, to read the flag after writing 0 to it.) • Reading from ADDR after activation of the DTC by an ADI interrupt
6	ADIE	0	R/W	<p>A/D Interrupt Enable</p> <p>Setting this bit to 1 enables ADI interrupts by ADI.</p>
5	ADST	0	R/W	<p>A/D Start</p> <p>Clearing this bit to 0 stops A/D conversion, and the converter enters wait state.</p> <p>Setting this bit to 1 starts A/D conversion. In single mode, this bit is cleared to 0 automatically when A/D conversion on the specified channel ends. In scan mode, A/D conversion continues sequentially on the specified channels until this bit is cleared to 0 by software, or hardware standby mode.</p> <p>Note: Do not write to ADST when activation is by external trigger. For details, see section 21.7.3, "A/D activation by an External Trigger."</p>
4	—	0	R/W	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>

0101: AN3

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	Initial Value	R/W	Description
7	ADF	0	R/(W)*	<p>A/D End Flag</p> <p>A status flag that indicates the end of A/D conversion.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> Completion of A/D conversion in single mode Completion of A/D conversion on all specified channels in scan mode <p>[Clearing conditions]</p> <ul style="list-style-type: none"> Writing of 0 after reading ADF = 1 (When the CPU is used to clear this flag by software while the corresponding interrupt is enabled, to read the flag after writing 0 to it.) Reading from ADDR after activation of the DTC by an ADI interrupt
6	ADIE	0	R/W	<p>A/D Interrupt Enable</p> <p>Setting this bit to 1 enables ADI interrupts by ADI.</p>
5	ADST	0	R/W	<p>A/D Start</p> <p>Clearing this bit to 0 stops A/D conversion, and the converter enters wait state.</p> <p>Setting this bit to 1 starts A/D conversion. In single mode, this bit is cleared to 0 automatically when A/D conversion on the specified channel ends. In scan mode, A/D conversion continues sequentially on the specified channels until this bit is cleared to 0 by software, or hardware standby mode.</p> <p>Note: Do not write to ADST when activation is by external trigger. For details, see section 21.7.3, A/D activation by an External Trigger.</p>

0100: AN4

0101: AN5

0110: AN6

0111: AN7

1XXX: Setting prohibited

- When SCANE = 1 and SCANS = 0

00XX: Setting prohibited

0100: AN4

0101: AN4 and AN5

0110: AN4 to AN6

0111: AN4 to AN7

1XXX: Setting prohibited

- When SCANE = 1 and SCANS = 1

XXXX: Setting prohibited

[Legend]

x: Don't care

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

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	<p>A/D End Flag</p> <p>A status flag that indicates the end of A/D conversion.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> Completion of A/D conversion in single mode Completion of A/D conversion on all specified channels in scan mode <p>[Clearing conditions]</p> <ul style="list-style-type: none"> Writing of 0 after reading ADF = 1 (When the CPU is used to clear this flag by software while the corresponding interrupt is enabled, to read the flag after writing 0 to it.) Reading from ADDR after activation of the DTC by an ADI interrupt
6	ADIE	0	R/W	<p>A/D Interrupt Enable</p> <p>Setting this bit to 1 enables ADI interrupts by ADIFR.</p>
5	ADST	0	R/W	<p>A/D Start</p> <p>Clearing this bit to 0 stops A/D conversion, and the converter enters wait state.</p> <p>Setting this bit to 1 starts A/D conversion. In single mode, this bit is cleared to 0 automatically when A/D conversion on the specified channel ends. In scan mode, A/D conversion continues sequentially on the specified channels until this bit is cleared to 0 by software, or hardware standby mode.</p> <p>Note: Do not write to ADST when activation is by external trigger. For details, see section 21.7.3, "A/D activation by an External Trigger."</p>

1000: AN8

1001: AN9

1010: AN10

1011: AN11

1XXX: Setting prohibited

- When SCANE = 1 and SCANS = 0

0XXX: Setting prohibited

1000: AN8

1001: AN8 and AN9

1010: AN8 to AN10

1011: AN8 to AN11

1XXX: Setting prohibited

- When SCANE = 1 and SCANS = 1

XXXX: 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 start of conversion by a trigger signal.
0	EXTRGS	0	R/W	<p>000: Disables A/D conversion start by external trigger</p> <p>010: Enables A/D conversion start by external trigger TPU (unit 0)</p> <p>100: Enables A/D conversion start by external trigger TMR (units 0 and 1)</p> <p>110: Enables A/D conversion start by the $\overline{\text{ADTRG0}}$ pin</p> <p>001: External trigger disabled</p> <p>011: Setting prohibited</p> <p>101: Setting prohibited</p> <p>111: Enables A/D conversion start by the $\overline{\text{ADTRG0}}$ pin (starts units simultaneously)</p> <p>Note: Do not write to ADST when activation is by an external trigger. For details, see section 21.7.3, Notes on A/D activation by an External Trigger.</p>

2	CKS0	0	R/W	<p>These bits set the A/D conversion time. First select conversion time by setting bits CKS1 and CKS0 which are 0 and then set the mode of A/D conversion.</p> <p>CKS1, and CKS0</p> <p>00: A/D conversion time = 530 states*³ (max.)</p> <p>01: A/D conversion time = 266 states*³ (max.)</p> <p>10: A/D conversion time = 134 states*³ (max.)</p> <p>11: A/D conversion time = 68 states*³ (max.)</p>
1	—	0	R/W	<p>Reserved</p> <p>This bit is always read as 0. The write value should be 0.</p>

[Legend]

x: Don't care

- Notes:
1. To set A/D conversion to start by the $\overline{\text{ADTRG}}$ pin, the DDR bit and ICR bit for corresponding pin should be set to 0 and 1, respectively. For details, see sections 13.1 and 13.2, I/O Ports.
 2. Select the $\overline{\text{ADTRG0}}$ input pin by the ADTRG0S bit in PFCR6. For details, see section 13.1, I/O Ports.
 3. P ϕ criterion

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 start of conversion by a trigger signal.
0	EXTRGS	0	R/W	<p>000: Disables A/D conversion start by external trigger</p> <p>010: Setting prohibited</p> <p>100: Setting prohibited</p> <p>110: Enables A/D conversion start by the $\overline{\text{ADTRG1-A}}$</p> <p>001: Enables A/D conversion start by the $\overline{\text{ADTRG1-B}}$</p> <p>011: External trigger disabled</p> <p>101: Enables A/D conversion start by external trigger</p> <p>TMR (units 2 and 3)</p> <p>111: Enables A/D conversion start by the $\overline{\text{ADTRG0}}$ pin (starts units simultaneously)</p> <p>Note: Do not write to ADST when activation is by an external trigger. For details, see section 21.7.3, Notes on A/D activation by an External Trigger.</p>
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	<p>These bits select the A/D conversion operating mode</p> <p>0x: Single mode</p> <p>10: Scan mode. A/D conversion is performed continuously on channels 1 to 4.</p> <p>11: Setting prohibited</p>

- 010: A/D conversion time = 164 states*³ (max.)
- 011: A/D conversion time = 68 states*³ (max.)
- 100: A/D conversion time = 332 states*³ (max.)
- 101: A/D conversion time = 168 states*³ (max.)
- 110: A/D conversion time = 87 states*³ (max.)
- 111: A/D conversion time = 46 states*³ (max.)

1	ADSTCLR	0	R/W	A/D Start Clear
---	---------	---	-----	-----------------

This bit enables or disables automatic clearing of the ADST bit in scan mode.

0: The ADST bit is not automatically cleared to 0 in scan mode.

1: Clears the ADST bit to 0 upon completion of the A/D conversion for all of the selected channels in scan mode.

[Legend]

x: Don't care

- Notes:
1. To set A/D conversion to start by the $\overline{\text{ADTRG}}$ pin, the DDR bit and ICR bit for the corresponding pin should be set to 0 and 1, respectively. For details, see section 13, I/O Ports.
 2. Select the $\overline{\text{ADTRG0}}$ input pin by the ADTRG0S bit in PFCR6. For details, see section 13, I/O Ports.
 3. P ϕ criterion

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 start of conversion by a trigger signal.
0	EXTRGS	0	R/W	000: Disables A/D conversion start by external trigger 010: Setting prohibited 100: Setting prohibited 110: Enables A/D conversion start by the $\overline{\text{ADTRG2}}$ pin 001: Setting prohibited 011: External trigger disabled 101: Enables A/D conversion start by external trigger and TMR (units 2 and 3) 111: Enables A/D conversion start by the $\overline{\text{ADTRG0}}$ pin (starts units simultaneously) Note: Do not write to ADST when activation is by an external trigger. For details, see section 21.7.3, Notes on A/D activation by an External Trigger.
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	These bits select the A/D conversion-operating mode 0x: Single mode 10: Scan mode. A/D conversion is performed continuously on channels 1 to 4. 11: Setting prohibited

011: A/D conversion time = 68 states*³ (max.)
 100: A/D conversion time = 332 states*³ (max.)
 101: A/D conversion time = 168 states*³ (max.)
 110: A/D conversion time = 87 states*³ (max.)
 111: A/D conversion time = 46 states*³ (max.)

1	ADSTCLR	0	R/W	A/D Start Clear
---	---------	---	-----	-----------------

This bit enables or disables automatic clearing of the ADST bit in scan mode.

0: The ADST bit is not automatically cleared to 0 in scan mode.

1: Clears the ADST bit to 0 upon completion of the A/D conversion for all of the selected channels in scan mode.

[Legend]

x: Don't care

- Notes:
1. To set A/D conversion to start by the $\overline{\text{ADTRG}}$ pin, the DDR bit and ICR bit for the corresponding pin should be set to 0 and 1, respectively. For details, see section 13, I/O Ports.
 2. Select the $\overline{\text{ADTRG0}}$ input pin by the ADTRG0S bit in PFCR6. For details, see section 13, I/O Ports.
 3. $P\phi$ criterion.

single channel.

1. A/D conversion for the selected channel is started when the ADST bit in ADCSR is set to 1 by software, TPU*¹, TMR*², 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 bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains at 1 during A/D conversion, and is automatically cleared to 0 when 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 a wait state.

Notes: 1. Only possible in unit 0.

2. As conversion start trigger, units 0 and 1 of TMR, and units 2 and 3 of TMR are available in unit 0, and unit 1, respectively.

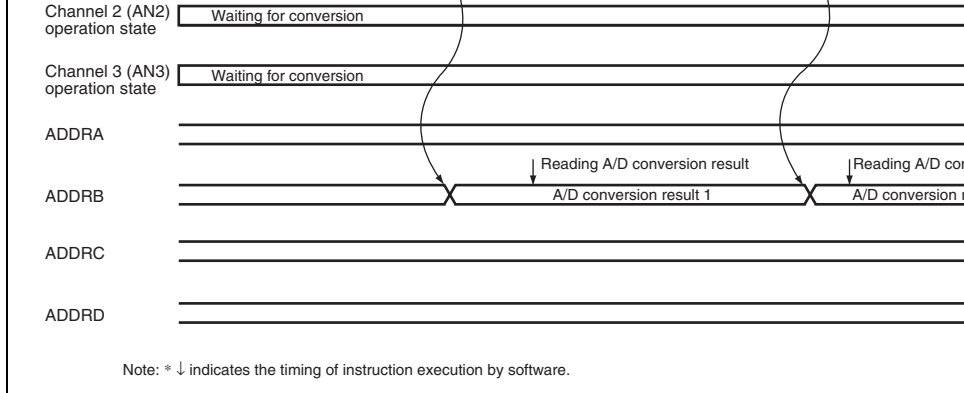


Figure 21.4 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

21.4.2 Scan Mode

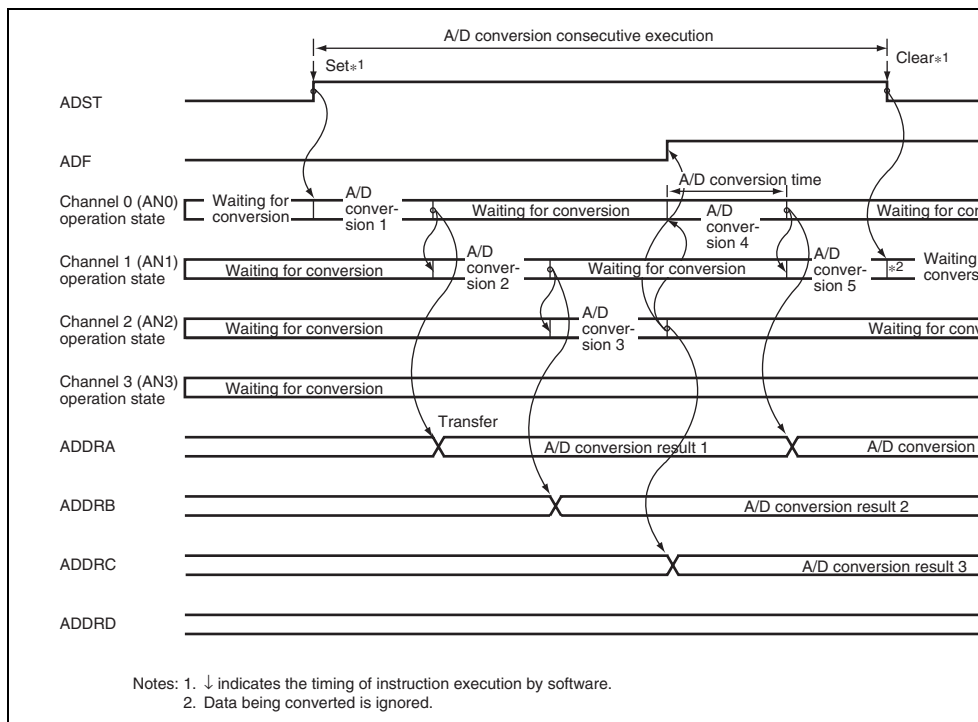
In scan mode, A/D conversion is to be performed sequentially on the analog inputs of the channels up to four or eight*¹ channels. Two types of scan mode are provided, that is, continuous scan mode where A/D conversion is repeatedly performed and one-cycle scan mode where A/D conversion is performed for the specified channels for one cycle.

(1) Continuous Scan Mode

1. When the ADST bit in ADCSR is set to 1 by software, TPU*¹, TMR*², or an external interrupt input, A/D conversion starts on the first channel in the specified channel group. Continuous A/D conversion on a maximum of four channels (SCANE and SCANS = B'10) or on a maximum of eight channels*¹ (SCANE and SCANS = B'11) can be selected. When continuous A/D conversion is performed on four channels, A/D conversion starts on AN0 when CH3 and CH2 of unit 0 = B'00, on AN4 when CH3 and CH2 of units 0 and 1 = B'01, on AN8 when CH3 and CH2 of units 0, 1 and 2 = B'10, and on AN12 when CH3 and CH2 of units 0, 1 and 2 = B'11.

the first channel in the group.

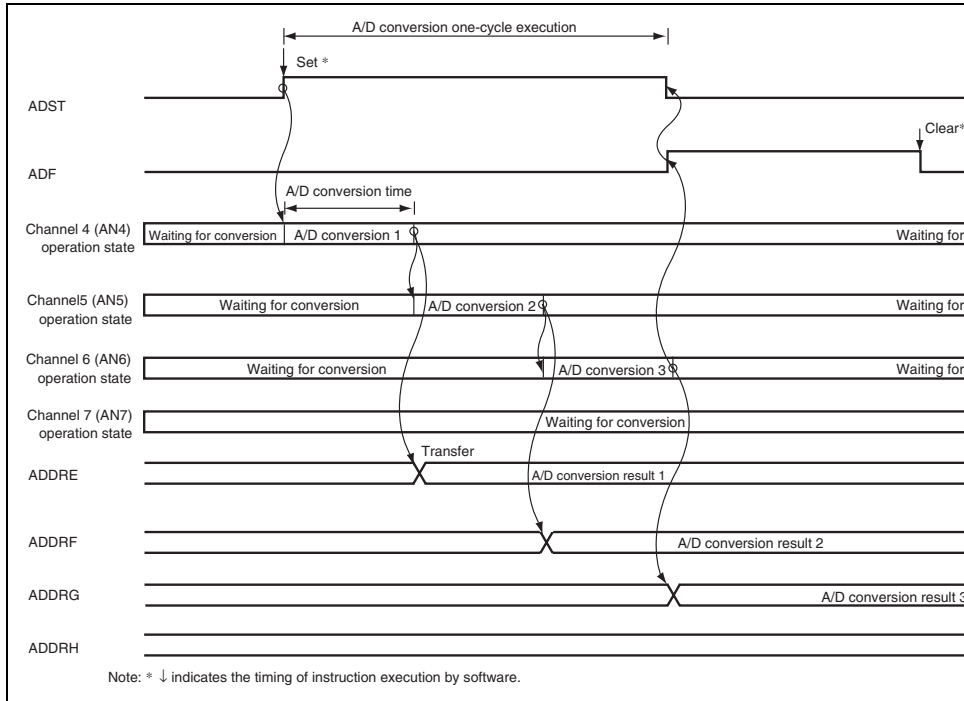
- Notes: 1. Only possible in unit 0.
 2. As conversion start trigger, units 0 and 1 of TMR, and units 2 and 3 of TMR are available in unit 0, and unit 1, respectively.



**Figure 21.5 Example of A/D Conversion
 (Continuous Scan Mode, Three Channels (AN0 to AN2) Selected)**

If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.

- The ADST bit is automatically cleared when A/D conversion is completed for all of channels that have been selected. A/D conversion stops and the A/D converter enters state.



**Figure 21.6 Example of A/D Conversion
(One-Cycle Scan Mode, Three Channels (AN4 to AN6) Selected)**

In scan mode, the values given in tables 21.3 and 21.4 apply to the first conversion time. The values given in table 21.5 apply to the second and subsequent conversions. In either case, CKS1 and CKS0 in ADCR should be set so that the conversion time is within the ranges by the A/D conversion characteristics.

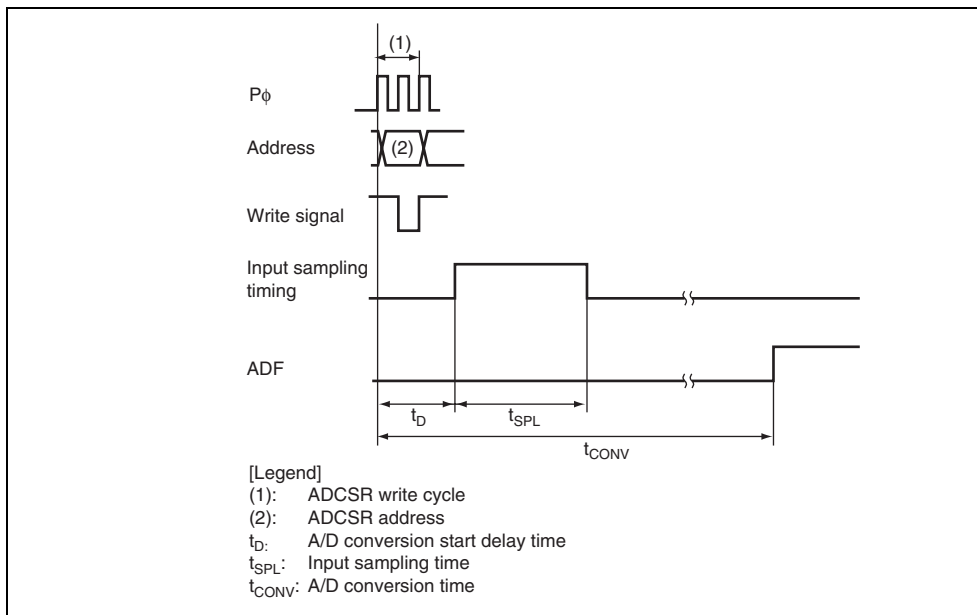


Figure 21.7 A/D Conversion Timing

Table 21.4 A/D Conversion Characteristics (EXCKS1 = 1) (Units 1 and 2)

Item	Symbol	CKS1 = 0						CKS1 = 1			
		CKS = 0			CKS = 1			CKS = 0		CKS = 1	
		Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.
A/D conversion start delay time	t_d	4	—	14	4	—	10	4	—	8	3
Input sampling time	t_{SPL}	—	120	—	—	60	—	—	30	—	—
A/D conversion time	t_{CONV}	326	—	336	166	—	172	86	—	90	45

Note: Values in the table are the number of states.

Table 21.5 A/D Conversion Time (Scan Mode) (Unit 0)

CKS1	CKS2	Conversion Time (Number of States)
0	0	512 (fixed)
	1	256 (fixed)
1	0	128 (fixed)
	1	64 (fixed)

21.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. For unit 0, an external trigger is input from the $\overline{\text{ADTRG0}}$ pin when the TRGS1, TRGS0, and EXTRGS bits are set to B'110 in ADCR_0. For unit 1, an external trigger is input from either the $\overline{\text{ADTRG1}}$ pin or the $\overline{\text{ADTRG2}}$ pin when the TRGS1, TRGS0, and EXTRGS bits are set to either B'110 or B'001 in ADCR_1. For unit 2, an external trigger is input from either the $\overline{\text{ADTRG2}}$ pin when the TRGS1, TRGS0, and EXTRGS bits are set to B'110 in ADCR_2. A/D conversion starts when the ADST bit in ADCSR is set to 1 on the falling edge of the ADTRG pin. Other operations, in both single and scan modes, are the same as when the ADST bit has been set to 1 by software. Figure 21.8 shows the timing.

Also, A/D conversion for multiple units can be externally triggered (multiple units can start simultaneously). For units 0, 1, and 2, an external trigger is input from the $\overline{\text{ADTRG0}}$ pin when the TRGS1, TRGS0, and EXTRGS bits are set to B'111 in ADCR_0, ADCR_1, and ADCR_2. A/D conversion starts when the ADST bit in ADCSR is set to 1 on the falling edge of the $\overline{\text{ADTRG0}}$ pin. The timing is different from the one when multiple units do not start simultaneously. Figure 21.9 shows the timing.

Figure 21.8 External Trigger Input Timing (TRGS1, TRGS0, and EXTRGS ≠ B'111)

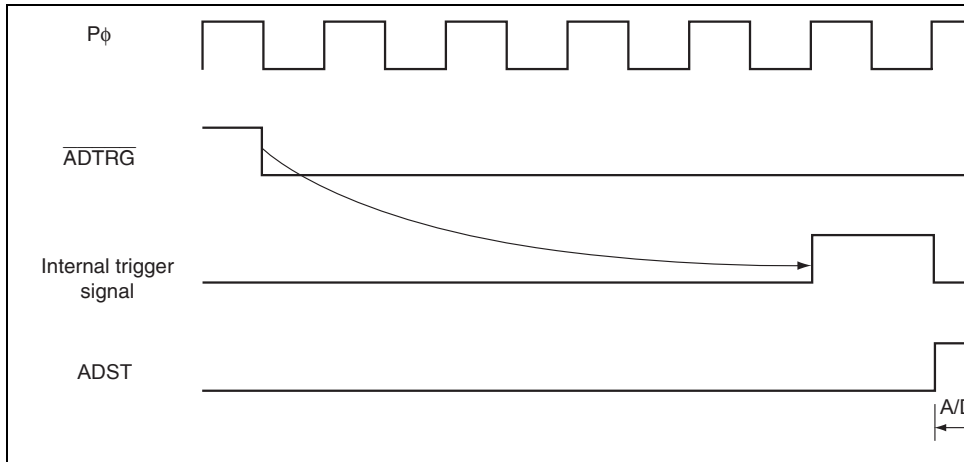


Figure 21.9 External Trigger Input Timing when Multiple Units Start Simultaneously (TRSG1, TRGS0, and EXTRGS = B'111)

Table 21.7 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
AD10	A/D conversion end	ADF	Possible*	Possible

Note: * Only possible in unit 0.

when the digital output changes from the minimum voltage value B'0000000000 (H'000) to the maximum voltage value B'0000000001 (H'001) (see figure 21.11).

- Full-scale error
The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H'3FF) (see figure 21.11).
- Nonlinearity error
The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error (see figure 21.11).
- Absolute accuracy
The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.

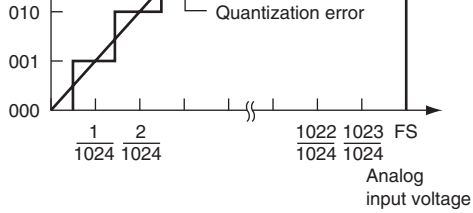


Figure 21.10 A/D Conversion Accuracy Definitions

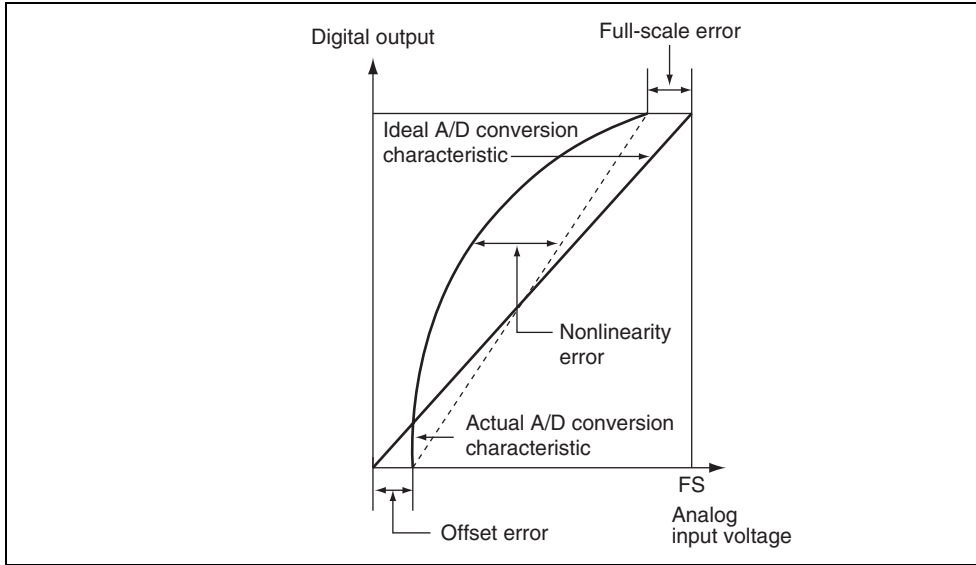


Figure 21.11 A/D Conversion Accuracy Definitions

21.7.2 A/D Input Hold Function in Software Standby Mode

When this LSI enters software standby mode with A/D conversion enabled, the analog input is retained, and the analog power supply current is equal to as during A/D conversion. If the power supply current needs to be reduced in software standby mode, set the CKS1 and CKS0 to 1 and clear the ADST, TRGS1, TRGS0, and EXTRGS bits all to 0 to disable A/D conversion. After that, enter software standby mode after executing a dummy read by one word.

21.7.3 Notes on A/D Activation by an External Trigger

If any of actions (1 to 3 below) is performed while activation by an external trigger* is in use, stopping A/D conversion may be impossible.

Note: * External trigger refers to input on the ADTRG pin or the conversion trigger from an external peripheral module (TMR or TPU).

1. When the setting for activation by an external trigger is in use, writing to change the setting to prohibition of external trigger by setting the ADST bit in ADCSR from 0 to 1.
2. Changing the setting from activation by an external trigger to prohibition of external trigger.
3. Changing the scan mode (SCANE and ADSTLCR bits; from continuous scan mode to single-cycle scan mode or single-cycle scan mode) while the setting is for activation by an external trigger.

If any of the above points apply, make the corresponding settings listed below.

1. If point 1 is applicable
Do not perform writing to change the value of the ADST bit in ADCSR from 0 to 1 while the setting for activation by an external trigger is in use.
2. If points 2 or 3 is applicable

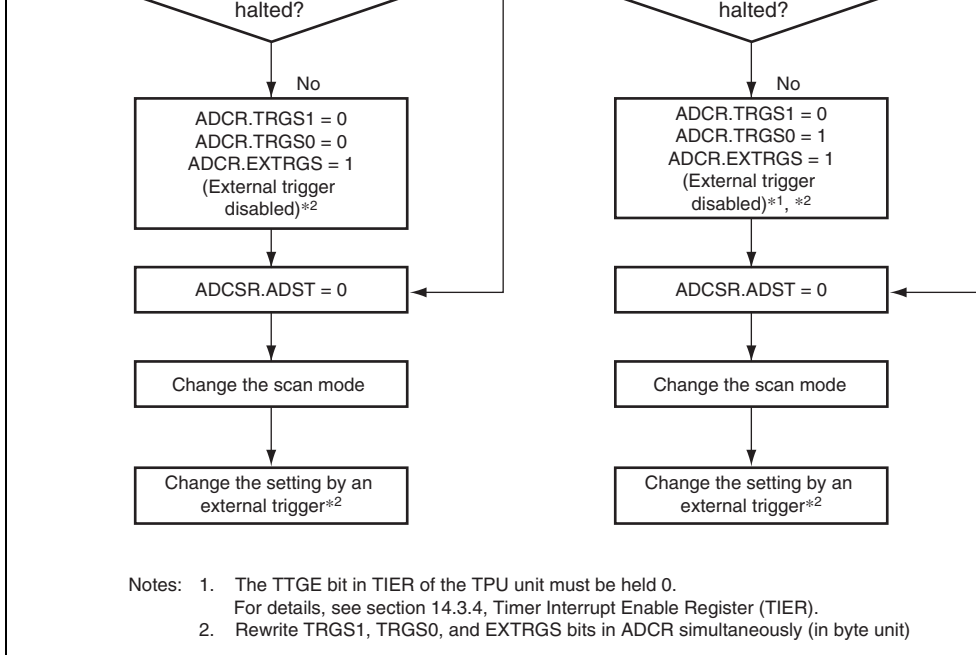


Figure 21.12 Procedure for Changing the Mode When the Setting for Activation External Trigger is in Use

a large differential coefficient (e.g., 5 mV/μs or greater) (see Figure 21.12). When converting a high-speed analog signal or conversion in scan mode, a low-impedance buffer should be

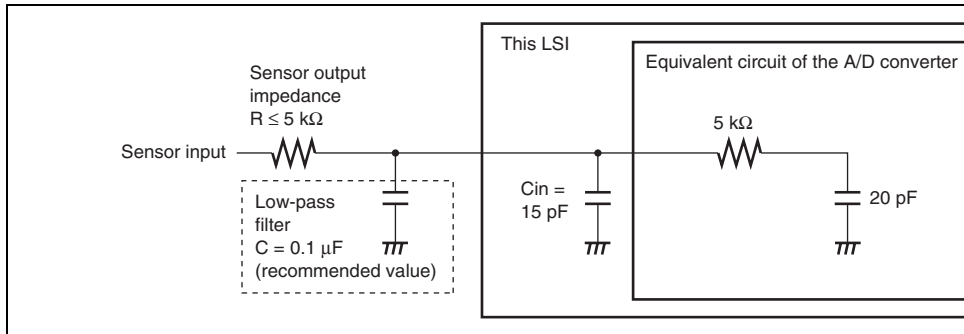


Figure 21.13 Example of Analog Input Circuit

21.7.5 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute accuracy. Be sure to make the connection to an electrically stable GND such as AVSS.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board, acting as antennas.

21.7.6 Setting Range of Analog Power Supply and Other Pins

If the conditions shown below are not met, the reliability of the LSI may be adversely affected.

- Analog input voltage range

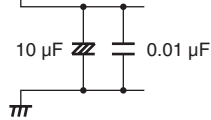
In board design, digital circuitry and analog circuitry should be as mutually isolated as possible. The design and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Digital circuitry must be isolated from the analog input pins (AN0 to AN11), analog reference voltage (Vref), and analog power supply (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable ground (Vss) on the board.

21.7.8 Notes on Noise Countermeasures

A protection circuit connected to prevent damage due to an abnormal voltage such as an electrostatic discharge surge at the analog input pins (AN0 to AN11) should be connected between AVcc and AVss as shown in figure 21.12. Also, the bypass capacitors connected to AVcc and the filter capacitors connected to the AN0 to AN11 pins must be connected to AVss.

If a filter capacitor is connected, the input currents at the AN0 to AN11 pins are averaged over time. As a result, an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (R_{in}), an error will arise in the input pin voltage. Careful consideration is therefore required when deciding the circuit configuration.

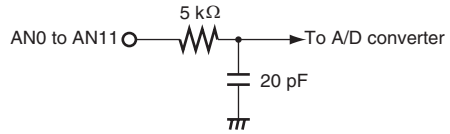


2. R_{in} : Input impedance

Figure 21.14 Example of Analog Input Protection Circuit

Table 21.8 Analog Pin Specifications

Item	Min.	Max.	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5	k Ω



Note: Values are reference values.

Figure 21.15 Analog Input Pin Equivalent Circuit

- Module stop state specifiable

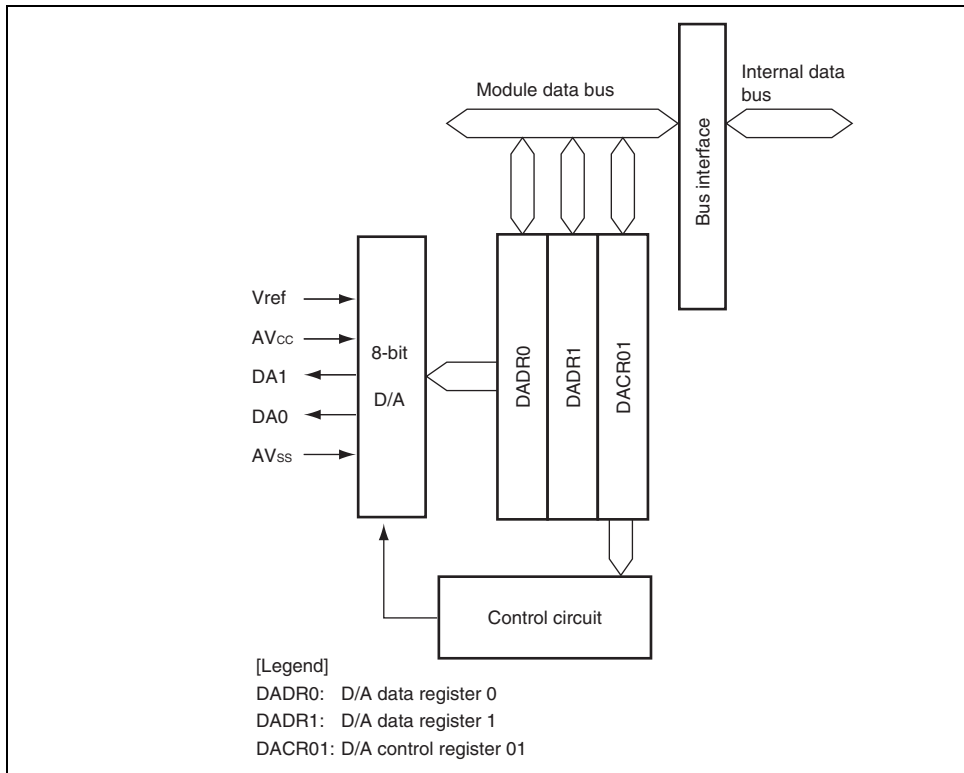


Figure 22.1 Block Diagram of D/A Converter

Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output

22.3 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)

22.3.1 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 conversion is to be performed. Whenever an analog output is enabled, the values in DADR are converted 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	<p>D/A Output Enable 1</p> <p>Controls D/A conversion and analog output.</p> <p>0: Analog output of channel 1 (DA1) is disabled.</p> <p>1: D/A conversion of channel 1 is enabled. Analog output of channel 1 (DA1) is enabled.</p>
6	DAOE0	0	R/W	<p>D/A Output Enable 0</p> <p>Controls D/A conversion and analog output.</p> <p>0: Analog output of channel 0 (DA0) is disabled.</p> <p>1: D/A conversion of channel 0 is enabled. Analog output of channel 0 (DA0) is enabled.</p>
5	DAE	0	R/W	<p>D/A Enable</p> <p>Used together with the DAOE0 and DAOE1 bits to control D/A conversion. When this bit is cleared to 0, D/A conversion is controlled independently for channels 0 and 1. When this bit is set to 1, D/A conversion for channels 0 and 1 is controlled together.</p> <p>Output of conversion results is always controlled by the DAOE0 and DAOE1 bits. For details, see Table 10. Control of D/A Conversion.</p>
4 to 0	—	All 1	R	<p>Reserved</p> <p>These are read-only bits and cannot be modified.</p>

			Analog output of channel 0 (DA0) is disabled and analog 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 analog 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 analog 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 $t_{\text{D CONV}}$ has elapsed. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is cleared. The output value is expressed by the following formula:

$$\text{Contents of DADR}/256 \times V_{\text{ref}}$$

3. If DADR0 is written to again, the conversion is immediately started. The conversion result continues to be output after the conversion time $t_{\text{D CONV}}$ has elapsed.
4. If the DAOE0 bit is cleared to 0, analog output is disabled.

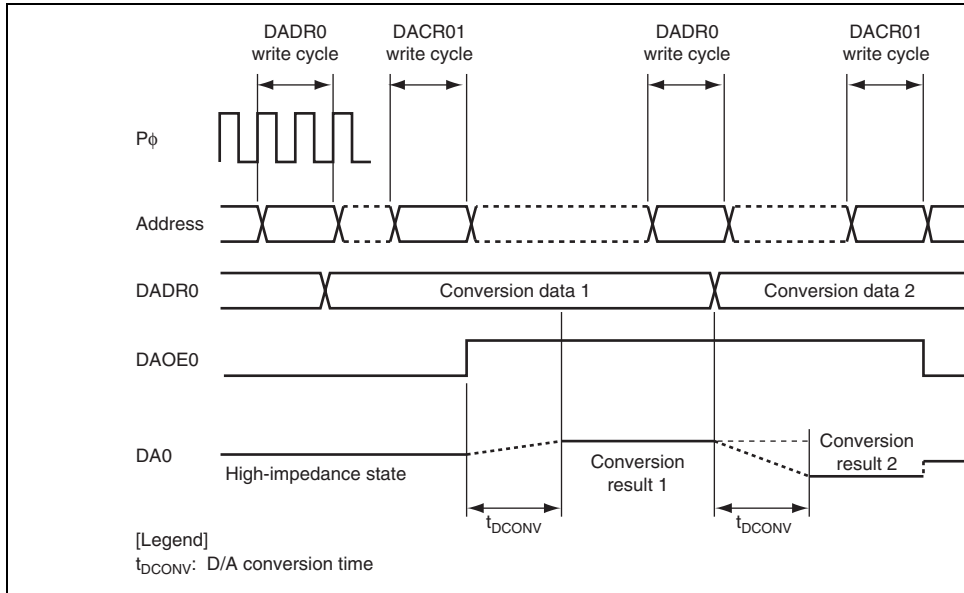


Figure 22.2 Example of D/A Converter Operation

When this LSI make a transition to software standby mode with D/A conversion enabled, outputs are retained, and the analog power supply current is equal to as during D/A conversion. When the analog power supply current needs to be reduced in software standby mode, clear the DAOE1, and DAE bits all to 0 to disable D/A conversion.

22.5.3 Notes on Deep Software Standby Mode

When this LSI make a transition to deep software standby mode, the D/A outputs enter high impedance state.

H8SX/1642H.

	Product Classification	RAM Size	RAM Address
Flash memory version	H8SX/1642	24 Kbytes	H'FF6000 to H'
	H8SX/1642A		
	H8SX/1642L		
	H8SX/1642G		
	H8SX/1642H		
	H8SX/1644	40 Kbytes	H'FF2000 to H'
	H8SX/1644A		
	H8SX/1644L		
	H8SX/1644G		
	H8SX/1644H		
	H8SX/1648	56 Kbytes	H'FEE000 to H'
	H8SX/1648A		
	H8SX/1648L		
	H8SX/1648G		
	H8SX/1648H		

	R5F61642L R5F61642G R5F61642H		
H8SX/1644	R5F61644 R5F61644A R5F61644L R5F61644G R5F61644H	512 Kbytes	H'000000 to H'07FFFF (modes 1, 2, 3, 6, and 7)
H8SX/1648	R5F61648 R5F61648A R5F61648L R5F61648G R5F61648H	1 Mbyte	H'000000 to H'0FFFFFFF (modes 1, 2, 3, 6, and 7)

- Two memory MATs

The start addresses of two memory spaces (memory MATs) are allocated to the same address. The mode setting in the initiation determines which memory MAT is initiated first. The memory MATs can be switched by using the bank-switching method after initiation.

- User MAT initiated at a reset in user mode: 256 Kbytes/512 Kbytes/1 Mbyte
- User boot MAT is initiated at reset in user boot mode: 16 Kbytes

- Programming/erasing interface by the download of on-chip program

This LSI has a programming/erasing program. After downloading this program to the RAM, programming/erasure can be performed by setting the parameters.

- Programming/erasing time

Programming time: 1 ms (typ.) for 128-byte simultaneous programming

Erasing time: 600 ms (typ.) per 1 block (64 Kbytes)

- Off-board programming mode
Programmer mode: Using a PROM programmer, the user MAT and user boot MAT can be programmed/erased.
- Programming/erasing protection
Protection against programming/erasure 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 overlaying the flash memory (user MAT) area and the on-chip RAM.

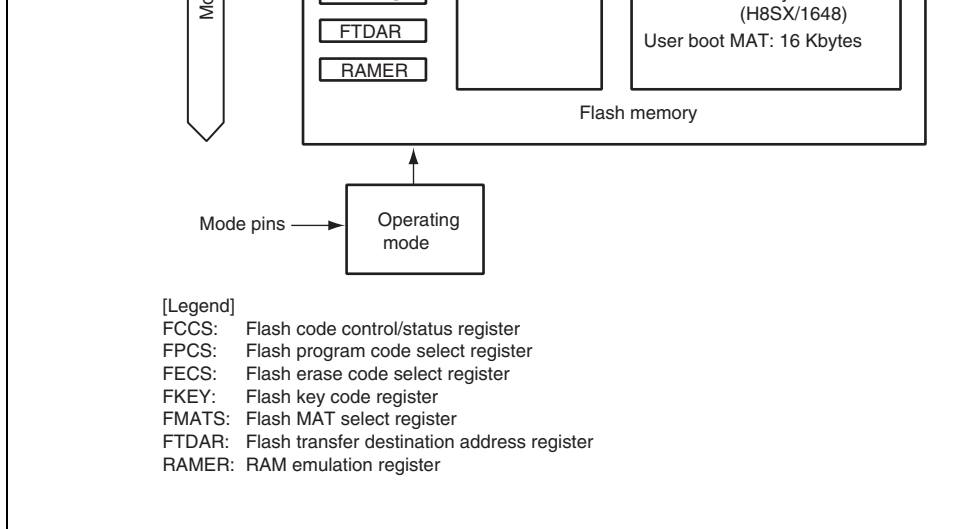


Figure 24.1 Block Diagram of Flash Memory

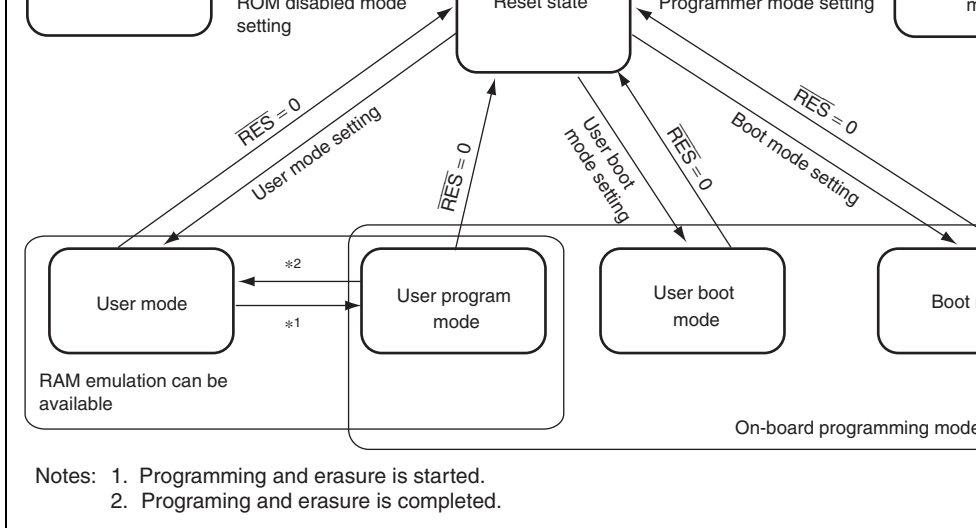


Figure 24.2 Mode Transition of Flash Memory

Programming/erasing control	Command	Programming/erasing interface	Programming/erasing interface	Command
All erasure	O (Automatic)	O	O	O (Automatic)
Block division erasure	O* ¹	O	O	×
Program data transfer	From host via SCI	From desired device via RAM	From desired device via RAM	Via program
RAM emulation	×	O	O	×
Reset initiation MAT	Embedded program storage area	User MAT	User boot MAT* ²	—
Transition to user mode	Changing mode and reset	Completing Programming/erasure* ³	Changing mode and reset	—

- Notes:
1. All-erasure is performed. After that, the specified block can be erased.
 2. First, the reset vector is fetched from the embedded program storage area. After the flash memory related registers are checked, the reset vector is fetched from the boot MAT.
 3. In this LSI, the user programming mode is defined as the period from the timing when the program concerning programming and erasure is started to the timing when the program is completed. For details on a program concerning programming and erasure, see section 24.8.2, User Program Mode.

The size of the user MAT is different from that of the user boot MAT. Addresses which e
the size of the 16-Kbyte user boot MAT should not be accessed. If an attempt is made, da
as an undefined value.

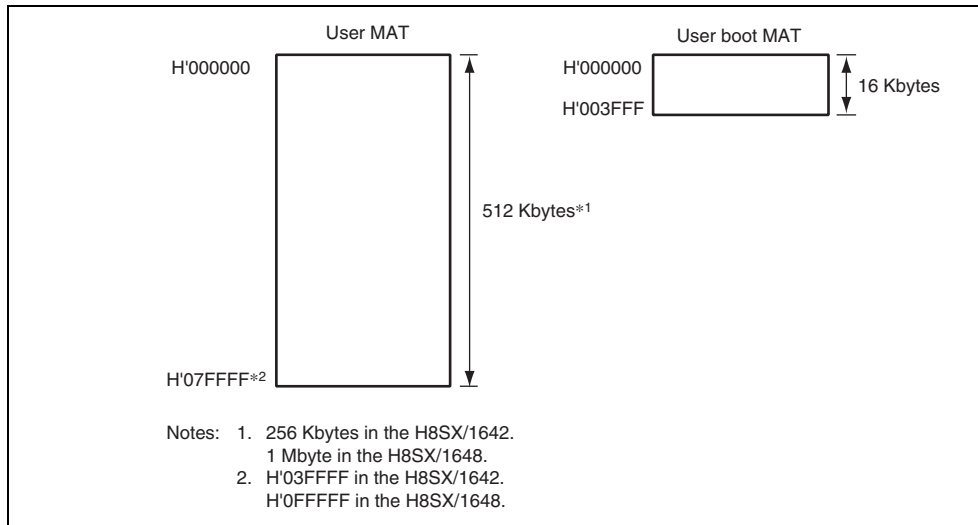


Figure 24.3 Memory MAT Configuration (H8SX/1644)

↑ EB0 Erase unit: 4 Kbytes ↕	H'000000	H'000001	H'000002	← Programming unit: 128 bytes →	H'000000
	H'000F80	H'000F81	H'000F82	-----	H'000F80
↑ EB1 Erase unit: 4 Kbytes ↕	H'001000	H'001001	H'001002	← Programming unit: 128 bytes →	H'001000
	H'001F80	H'001F81	H'001F82	-----	H'001F80
↑ EB2 Erase unit: 4 Kbytes ↕	H'002000	H'002001	H'002002	← Programming unit: 128 bytes →	H'002000
	H'002F80	H'002F81	H'002F82	-----	H'002F80
↑ EB3 Erase unit: 4 Kbytes ↕	H'003000	H'003001	H'003002	← Programming unit: 128 bytes →	H'003000
	H'003F80	H'003F81	H'003F82	-----	H'003F80
↑ EB4 Erase unit: 4 Kbytes ↕	H'004000	H'004001	H'004002	← Programming unit: 128 bytes →	H'004000
	H'004F80	H'004F81	H'004F82	-----	H'004F80
↑ EB5 Erase unit: 4 Kbytes ↕	H'005000	H'005001	H'005002	← Programming unit: 128 bytes →	H'005000
	H'005F80	H'005F81	H'005F82	-----	H'005F80
↑ EB6 Erase unit: 4 Kbytes ↕	H'006000	H'006001	H'006002	← Programming unit: 128 bytes →	H'006000
	H'006F80	H'006F81	H'006F82	-----	H'006F80
↑ EB7 Erase unit: 4 Kbytes ↕	H'007000	H'007001	H'007002	← Programming unit: 128 bytes →	H'007000
	H'007F80	H'007F81	H'007F82	-----	H'007F80
↑ EB8 Erase unit: 32 Kbytes ↕	H'008000	H'008001	H'008002	← Programming unit: 128 bytes →	H'008000
	H'00FF80	H'00FF81	H'00FF82	-----	H'00FF80
↑ EB9 Erase unit: 64 Kbytes ↕	H'010000	H'010001	H'010002	← Programming unit: 128 bytes →	H'010000
	H'01FF80	H'01FF81	H'01FF82	-----	H'01FF80
↑ EB10 ↕	H'020000	H'020001	H'020002	← Programming unit: 128 bytes →	H'020000
	H'02FF80	H'02FF81	H'02FF82	-----	H'02FF80
↑ EB11 Erase unit: 64 Kbytes ↕	H'030000	H'030001	H'030002	← Programming unit: 128 bytes →	H'030000
	H'03FF80	H'03FF81	H'03FF82	-----	H'03FF80

Figure 24.4 User MAT Block Structure of H8SX/1642

EB1 Erase unit: 4 Kbytes	H'000F80	H'000F81	H'000F82	←Programming unit: 128 bytes→	H'000F83
	H'001F80	H'001F81	H'001F82	-----	H'001F83
EB2 Erase unit: 4 Kbytes	H'002000	H'002001	H'002002	←Programming unit: 128 bytes→	H'002003
	H'002F80	H'002F81	H'002F82	-----	H'002F83
EB3 Erase unit: 4 Kbytes	H'003000	H'003001	H'003002	←Programming unit: 128 bytes→	H'003003
	H'003F80	H'003F81	H'003F82	-----	H'003F83
EB4 Erase unit: 4 Kbytes	H'004000	H'004001	H'004002	←Programming unit: 128 bytes→	H'004003
	H'004F80	H'004F81	H'004F82	-----	H'004F83
EB5 Erase unit: 4 Kbytes	H'005000	H'005001	H'005002	←Programming unit: 128 bytes→	H'005003
	H'005F80	H'005F81	H'005F82	-----	H'005F83
EB6 Erase unit: 4 Kbytes	H'006000	H'006001	H'006002	←Programming unit: 128 bytes→	H'006003
	H'006F80	H'006F81	H'006F82	-----	H'006F83
EB7 Erase unit: 4 Kbytes	H'007000	H'007001	H'007002	←Programming unit: 128 bytes→	H'007003
	H'007F80	H'007F81	H'007F82	-----	H'007F83
EB8 Erase unit: 32 Kbytes	H'008000	H'008001	H'008002	←Programming unit: 128 bytes→	H'008003
	H'00FF80	H'00FF81	H'00FF82	-----	H'00FF83
EB9 Erase unit: 64 Kbytes	H'010000	H'010001	H'010002	←Programming unit: 128 bytes→	H'010003
	H'01FF80	H'01FF81	H'01FF82	-----	H'01FF83
EB10	H'020000	H'020001	H'020002	←Programming unit: 128 bytes→	H'020003
	H'0AFF80	H'0AFF81	H'0AFF82	-----	H'0AFF83
EB15 Erase unit: 64 Kbytes	H'070000	H'070001	H'070002	←Programming unit: 128 bytes→	H'070003
	H'07FF80	H'07FF81	H'07FF82	-----	H'07FF83

Figure 24.4 User MAT Block Structure of H8SX/1644

EB1 Erase unit: 4 Kbytes	H'000F80	H'000F81	H'000F82	←Programming unit: 128 bytes→	H'000F83
	H'001F80	H'001F81	H'001F82	-----	H'001F83
EB2 Erase unit: 4 Kbytes	H'002000	H'002001	H'002002	←Programming unit: 128 bytes→	H'002003
	H'002F80	H'002F81	H'002F82	-----	H'002F83
EB3 Erase unit: 4 Kbytes	H'003000	H'003001	H'003002	←Programming unit: 128 bytes→	H'003003
	H'003F80	H'003F81	H'003F82	-----	H'003F83
EB4 Erase unit: 4 Kbytes	H'004000	H'004001	H'004002	←Programming unit: 128 bytes→	H'004003
	H'004F80	H'004F81	H'004F82	-----	H'004F83
EB5 Erase unit: 4 Kbytes	H'005000	H'005001	H'005002	←Programming unit: 128 bytes→	H'005003
	H'005F80	H'005F81	H'005F82	-----	H'005F83
EB6 Erase unit: 4 Kbytes	H'006000	H'006001	H'006002	←Programming unit: 128 bytes→	H'006003
	H'006F80	H'006F81	H'006F82	-----	H'006F83
EB7 Erase unit: 4 Kbytes	H'007000	H'007001	H'007002	←Programming unit: 128 bytes→	H'007003
	H'007F80	H'007F81	H'007F82	-----	H'007F83
EB8 Erase unit: 32 Kbytes	H'008000	H'008001	H'008002	←Programming unit: 128 bytes→	H'008003
	H'00FF80	H'00FF81	H'00FF82	-----	H'00FF83
EB9 Erase unit: 64 Kbytes	H'010000	H'010001	H'010002	←Programming unit: 128 bytes→	H'010003
	H'01FF80	H'01FF81	H'01FF82	-----	H'01FF83
EB10	H'020000	H'020001	H'020002	←Programming unit: 128 bytes→	H'020003
	H'0EFF80	H'0EFF81	H'0EFF82	-----	H'0EFF83
EB23 Erase unit: 64 Kbytes	H'0F0000	H'0F0001	H'0F0002	←Programming unit: 128 bytes→	H'0F0003
	H'0FFF80	H'0FFF81	H'0FFF82	-----	H'0FFF83

Figure 24.4 User MAT Block Structure of H8SX/1648

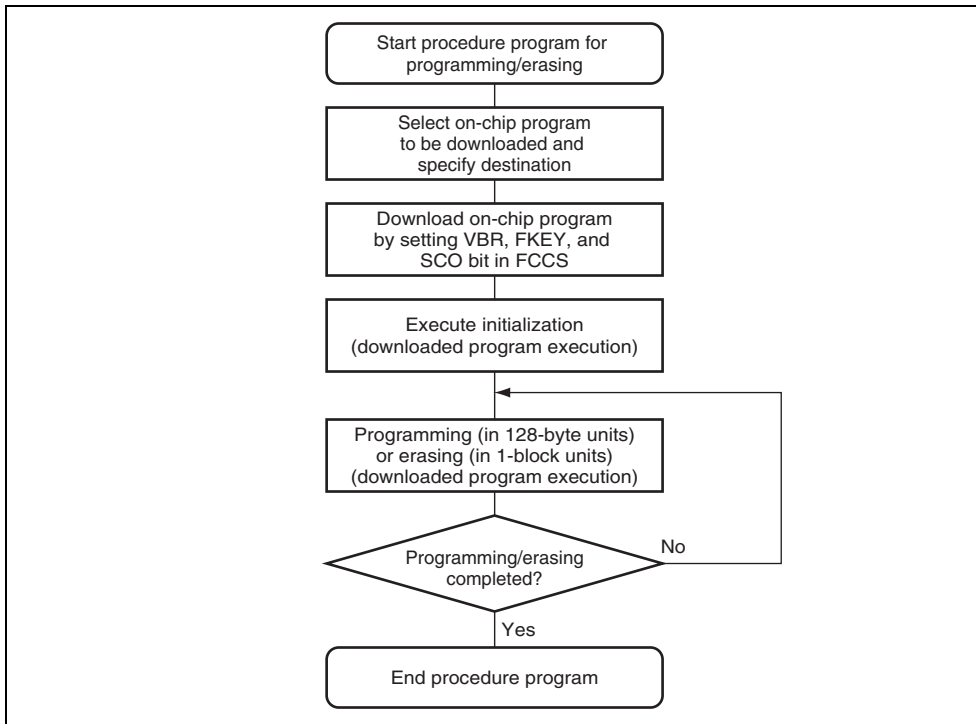


Figure 24.5 Procedure for Creating Procedure Program

(1) Selection of On-Chip Program to be Downloaded

This LSI has programming/erasing programs which can be downloaded to the on-chip RAM. The on-chip program to be downloaded is selected by the programming/erasing interface register. The start address of the on-chip RAM where an on-chip program is downloaded is specified by the flash transfer destination address register (FTDAR).

(3) Initialization of Programming/Erase

A pulse with the specified period must be applied when programming or erasing. The specified pulse width is made by the method in which wait loop is configured by the CPU instruction. Accordingly, the operating frequency of the CPU needs to be set before programming/erasing. The operating frequency of the CPU is set by the programming/erasing interface parameter.

(4) Execution of Programming/Erase

The start address of the programming destination and the program data are specified in 128-byte units when programming. The block to be erased is specified with the erase block number in 128-byte erase-block units when erasing. Specifications of the start address of the programming destination, program data, and erase block number are performed by the programming/erasing interface parameters, and the on-chip program is initiated. The on-chip program is executed by using the JSR or BSR instruction and executing the subroutine call of the specified address in the on-chip 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/erasure.

(5) When Programming/Erase is Executed Consecutively

When processing does not end by 128-byte programming or 1-block erasure, consecutive programming/erasure can be realized by updating the start address of the programming destination and program data, or the erase block number. Since the downloaded on-chip program is stored in on-chip RAM even after programming/erasure completes, download and initialization are not required when the same processing is executed consecutively.

TxD4	Output	Serial transmit data output (used in boot mode)
RxD4	Input	Serial receive data input (used in boot mode)

24.7 Register Descriptions

The flash memory has the following registers.

Programming/Erasing Interface Registers:

- Flash code control/status register (FCCS)
- Flash program code select register (FPCS)
- Flash erase code select register (FECS)
- Flash key code register (FKEY)
- Flash MAT select register (FMATS)
- 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)

	FKEY	0	—	0	0	—	—
	FMATS	—	—	0* ¹	0* ¹	0* ²	—
	FTDAR	0	—	—	—	—	—
Programming/ erasing interface parameters	DPFR	0	—	—	—	—	—
	FPFR	—	0	0	0	—	—
	FPEFEQ	—	0	—	—	—	—
	FMPAR	—	—	0	—	—	—
	FMPDR	—	—	0	—	—	—
	FEBS	—	—	—	0	—	—
RAM emulation	RAMER	—	—	—	—	—	0

Notes: 1. The setting is required when programming or erasing the user MAT in user boot mode.
 2. The setting may be required according to the combination of initiation mode and target memory MAT.

24.7.1 Programming/Erasing Interface Registers

The programming/erasing interface registers are 8-bit registers that can be accessed only by the CPU. These registers are initialized by a reset.

(1) Flash Code Control/Status Register (FCCS)

FCCS monitors errors during programming/erasing the flash memory and requests the CPU to download the program to be downloaded to the on-chip RAM.

5	—	0	R	
4	FLER	0	R	Flash Memory Error

Indicates that an error has occurred during programming or erasing the flash memory. When this bit is set to 1, the flash memory enters the error protection state. When this bit is set to 1, high voltage is applied to the internal flash memory. To reduce the damage to the flash memory, the reset must be released after the input period (period of $\overline{RES} = 0$) of at least 100 μ s.

0: Flash memory operates normally (Error protection is invalid)

[Clearing condition]

- At a reset

1: An error occurs during programming/erasing of the flash memory (Error protection is valid)

[Setting conditions]

- When an interrupt, such as NMI, occurs during programming/erasure.
- When the flash memory is read during programming/erasure (including a vector refresh during an instruction fetch).
- When the SLEEP instruction is executed during programming/erasure (including software step mode).
- When a bus master other than the CPU, such as the DMAC and DTC, obtains bus mastership during programming/erasure.

must be canceled, H'A5 must be written to FK. This operation must be executed in the on-chip. Dummy read of FCCS must be executed twice immediately after setting this bit to 1. All interrupts must be disabled during download. This bit is cleared when download is completed.

During program download initiated with this bit, particular processing which accompanies bank switching of the program storage area is executed. Before a download request, initialize the VBR to H'00000000. After download is completed, contents can be changed.

0: Download of the programming/erasing program requested.

[Clearing condition]

- When download is completed

1: Download of the programming/erasing program requested.

[Setting conditions] (When all of the following are satisfied)

- Not in RAM emulation mode (the RAMS bit RAMER is cleared to 0)
- H'A5 is written to FKEY
- Setting of this bit is executed in the on-chip

Note: * This is a write-only bit. This bit is always read as 0.

7 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modified.
0	PPVS	0	R/W	Program Pulse Verify Selects the programming program to be downloaded. 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	0
Bit Name	—	—	—	—	—	—	—	—
Initial Value	0	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 modified.
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
5	K5	0	R/W	FCCS is enabled. When a value other than H' written, the SCO bit cannot be set to 1. Theref
4	K4	0	R/W	on-chip program cannot be downloaded to the
3	K3	0	R/W	RAM.
2	K2	0	R/W	Only when H'5A is written can programming/e
1	K1	0	R/W	the flash memory be executed. When a value
0	K0	0	R/W	H'5A is written, even if the programming/erasi
				program is executed, programming/erasure ca
				performed.
				H'A5: Writing to the SCO bit is enabled. (The
				cannot be set to 1 when FKEY is a valu
				than H'A5.)
				H'5A: Programming/erasure of the flash memo
				enabled. (When FKEY is a value other t
				the software protection state is entered.
				H'00: Initial value

Bit	Bit Name	Initial Value	R/W	Description
7	MS7	0/1*	R/W	MAT Select
6	MS6	0	R/W	The memory MATs can be switched by writing to FMATS.
5	MS5	0/1*	R/W	
4	MS4	0	R/W	When H'AA is written to FMATS, the user boot selected. When a value other than H'AA is written to FMATS, the user MAT is selected. Switch the MATs following the memory MAT switching procedure in section 24.
3	MS3	0/1*	R/W	
2	MS2	0	R/W	Switching between User MAT and User Boot MAT.
1	MS1	0/1*	R/W	user boot MAT cannot be selected by FMATS in programming mode. The user boot MAT can be selected in boot mode or programmer mode.
0	MS0	0	R/W	

H'AA: The user boot MAT is selected. (The user boot MAT is selected when FMATS is a value other than H'AA.)
(Initial value when initiated in user boot mode.)

H'00: The user MAT is selected.
(Initial value when initiated in a mode except user boot mode.)

Note: * This bit is set to 1 in user boot mode, otherwise cleared to 0.

Bit	Bit Name	Value	R/W	Description
7	TDER	0	R/W	<p>Transfer Destination Address Setting Error</p> <p>This bit is set to 1 when an error has occurred in the start address specified by bits TDA6 to TDA0.</p> <p>A start address error is determined by whether the value set in bits TDA6 to TDA0 is within the range of H'00 to H'02 when download is executed by setting the bit in FCCS to 1. Make sure that this bit is cleared before setting the SCO bit to 1 and the value specified by bits TDA6 to TDA0 should be within the range of H'00 to H'02.</p> <p>0: The value specified by bits TDA6 to TDA0 is within the range.</p> <p>1: The value specified by bits TDA6 to TDA0 is not within the range of H'03 and H'FF and download has stopped.</p>
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 and H'02 and up to 4 Kbytes can be specified as the start address of the on-chip RAM.
4	TDA4	0	R/W	
3	TDA3	0	R/W	
2	TDA2	0	R/W	
1	TDA1	0	R/W	H'01: H'FFA000 is specified as the start address.
0	TDA0	0	R/W	<p>H'02: H'FFB000 is specified as the start address.</p> <p>H'03 to H'7F: Setting prohibited. (Specifying a value from H'03 to H'7F while the TDER bit to 1 and stops download of the on-chip program.)</p>

processing result is written in R0. The programming/erasing interface parameters are used for download control, initialization before programming or erasing, programming, and erasing. Table 24.4 shows the usable parameters and target modes. The meaning of the bits in the flash pass/fail result parameter (FPFR) varies in initialization, programming, and erasure.

Table 24.4 Parameters and Target Modes

Parameter	Download	Initialization	Programming	Erasure	R/W	Initial Value	Allocation
DPFR	0	—	—	—	R/W	Undefined	On-chip RAM
FPFR	0	0	0	0	R/W	Undefined	R0L
FPEFEQ	—	0	—	—	R/W	Undefined	ER0
FMPAR	—	—	0	—	R/W	Undefined	ER1
FMPDR	—	—	0	—	R/W	Undefined	ER0
FEBS	—	—	—	0	R/W	Undefined	ER0

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 SCFCCS 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/erasing registers, and the download pass and fail result parameter (DPFR) indicates the return value.

register ER1. This parameter is called the flash multipurpose address area parameter (FMAA).
The program data is always in 128-byte units. When the program data does not satisfy 128-byte program data is prepared by filling the dummy code (H'FF). The boundary of the address of the programming destination on the user MAT is aligned at an address where 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 that can be accessed using the MOV.B instruction of the 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 set in general register ER0. This parameter is called the flash multipurpose data destination address parameter (FMPDR).

For details on the programming procedure, see section 24.8.2, User Program Mode.

Erase: When the flash memory is erased, the erase block number on the user MAT must be passed to the erasing program which is downloaded.

The erase block number on the user MAT must be set in general register ER0. This parameter is called the flash erase block select parameter (FEBS).

One block is selected from the block numbers of 0 to 19 as the erase block number.

For details on the erasing procedure, see section 24.8.2, 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 program which can be downloaded. When the program type downloaded is not selected, more than two type programs are selected, or a program which is not 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 the value. 0: FKEY setting is normal (H'A5) 1: FKEY setting is abnormal (value other than H'A5)
0	SF	—	R/W	Success/Fail Returns the download result. Reads back the program downloaded to the on-chip RAM and determines whether it has been transferred to the on-chip Flash memory. 0: Download of the program has ended normally (no error) 1: Download of the program has ended abnormally (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 frequency with the operating frequencies supported by this L3 cache and 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 occurred)

6	MD	—	R/W	<p>Programming Mode Related Setting Error Detect</p> <p>Detects the error protection state and returns the error protection state. When the error protection state is entered, this bit is set to 1. Whether the error protection state is entered can be confirmed with the FLER bit in FCCS. For the conditions to enter the error protection state, see Section 24.9.3, Error Protection.</p> <p>0: Normal operation (FLER = 0)</p> <p>1: Error protection state, and programming cannot be performed (FLER = 1)</p>
5	EE	—	R/W	<p>Programming Execution Error Detect</p> <p>Writes 1 to this bit when the specified data could not be written because the user MAT was not erased. If this bit is set to 1, there is a high possibility that the user MAT has been written to partially. In this case, after the error factor, erase the user MAT. If FMATSH'AA and the user boot MAT is selected, an error occurs when programming is performed. In this case, both the user MAT and user boot MAT have not been written. Programming the user boot MAT should be performed in boot mode or programmer mode.</p> <p>0: Programming has ended normally</p> <p>1: Programming has ended abnormally (programming result is not guaranteed)</p>

When an address not in the flash memory are specified as the start address of the storage destination 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 abnormal

1	WA	—	R/W	Write Address Error Detect
---	----	---	-----	----------------------------

When the following items are specified as the start address of the programming destination, an error occurs.

- An area other than flash memory
- The specified address is not aligned with the byte boundary (lower eight bits of the address other than H'00 and H'80)

0: Setting of the start address of the programming destination is normal

1: Setting of the start address of the programming 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)

6	MD	—	R/W	<p>Erase Mode Related Setting Error Detect</p> <p>Detects the error protection state and returns the error protection state. When the error protection state is entered, this bit returns to 1. Whether the error protection state is entered can be confirmed with the FLER bit in FCCS. For conditions to enter the error protection state, see Section 24.9.3, Error Protection.</p> <p>0: Normal operation (FLER = 0)</p> <p>1: Error protection state, and programming cannot be performed (FLER = 1)</p>
5	EE	—	R/W	<p>Erase Execution Error Detect</p> <p>Returns 1 when the user MAT could not be erased normally when the flash memory related register settings are partially changed. If this bit is set to 1, there is a possibility that the user MAT has been erased partially. In this case, after removing the error factor, erase the user MAT. If FMATS is set to H'AA and the user MAT is selected, an error occurs when erasure is performed. In this case, both the user MAT and the boot MAT have not been erased. Erasing of the boot MAT should be performed in boot mode of the programmer mode.</p> <p>0: Erasure has ended normally</p> <p>1: Erasure has ended abnormally</p>

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 occurred)

(3) Flash Program/Erase Frequency Parameter (FPEFEQ: General Register ER)

FPEFEQ sets the operating frequency of the CPU. The operating frequency available in ranges from 8 MHz to 50 MHz.

Bit	31	30	29	28	27	26	25	
Bit Name	—	—	—	—	—	—	—	
Bit	23	22	21	20	19	18	17	
Bit Name	—	—	—	—	—	—	—	
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	

be shown in a number of two decimal places.

2. The value multiplied by 100 is converted to a binary digit and is written to FPEFEQ (general register ER0).

For example, when the operating frequency of the microcontroller is 35.000 MHz, the value is as follows:

1. The number of three decimal places of 35.000 is rounded.
 2. The formula of $35.00 \times 100 = 3500$ is converted to 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	
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 programming destination on the user MAT. Consecutive 128-byte programming is executed starting from the specified start address of the user MAT. Therefore, the start address of the programming destination must be a 128-byte boundary, and MOA6 to MOA0 are always cleared to 0.

Bit	23	22	21	20	19	18	17
Bit Name	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18	MOD17
Bit	15	14	13	12	11	10	9
Bit Name	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9
Bit	7	6	5	4	3	2	1
Bit Name	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1

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 which stores the program data for the user MAT. Containing 128-byte data is programmed to the user MAT from the specified start address.

- H8SX/1648

FEBS specifies the erase block number. Settable values range from 0 to 23 (H'0000). A value of 0 corresponds to block EB0 and a value of 23 corresponds to block EB23. An error occurs when a value over the range (from 0 to 23) is set.

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	

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	0	R	Reserved These are read-only bits and cannot be modified.
3	RAMS	0	R/W	RAM Select Selects the function which emulates the flash memory using the on-chip RAM. 0: Disables RAM emulation function 1: Enables RAM emulation function (all blocks of user MAT are protected against programming and erasing)
2	RAM2	0	R/W	Flash Memory Area Select
1	RAM1	0	R/W	These bits select the user MAT area overlaid with on-chip RAM when RAMS = 1. The following addresses correspond to the 4-Kbyte erase blocks.
0	RAM0	0	R/W	
				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	MD2	MD1	MD0
User boot mode	0	0	0	1
Boot mode	0	0	1	0
User program mode	0	1	1	0
	0	1	1	1

24.8.1 Boot Mode

Boot mode executes programming/erasure of the user MAT or user boot MAT by means of a control command and program data transmitted from the externally connected host via the SCI_4.

In boot mode, the tool for transmitting the control command and program data, and the program data must be prepared in the host. The serial communications mode is set to asynchronous. The system configuration in boot mode is shown in figure 24.6. Interrupts are ignored in boot mode. Configure the user system so that interrupts do not occur.

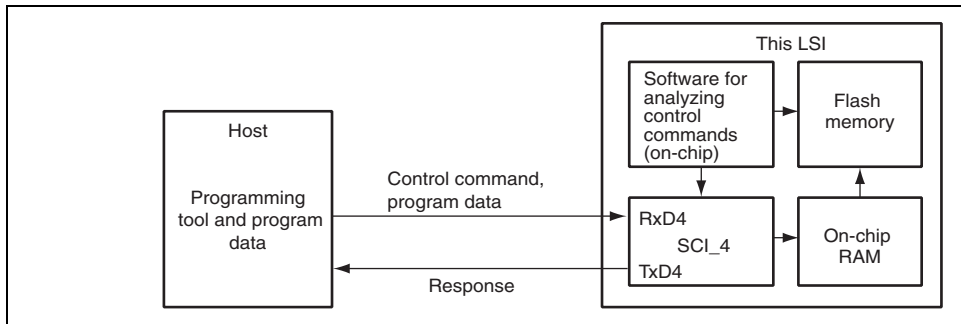


Figure 24.6 System Configuration in Boot Mode

adjustment end sign. When the host receives this bit adjustment end sign normally, it transmits the next byte of H'55 to this LSI. When reception is not executed normally, initiate boot mode again. The bit rate may not be adjusted within the allowable range depending on the combination of the bit rate of the host and the system clock frequency of this LSI. Therefore, the transfer bit rate of the host and the system clock frequency of this LSI must be as shown in table 24.6.

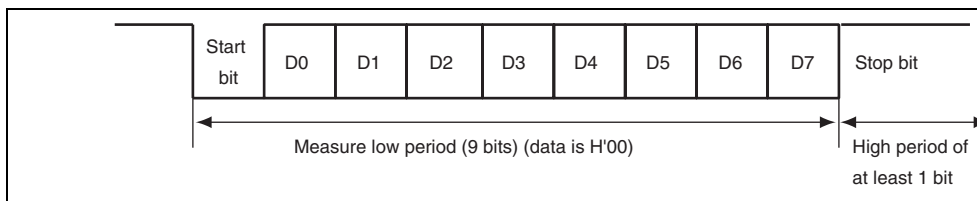


Figure 24.7 Automatic-Bit-Rate Adjustment Operation

Table 24.6 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

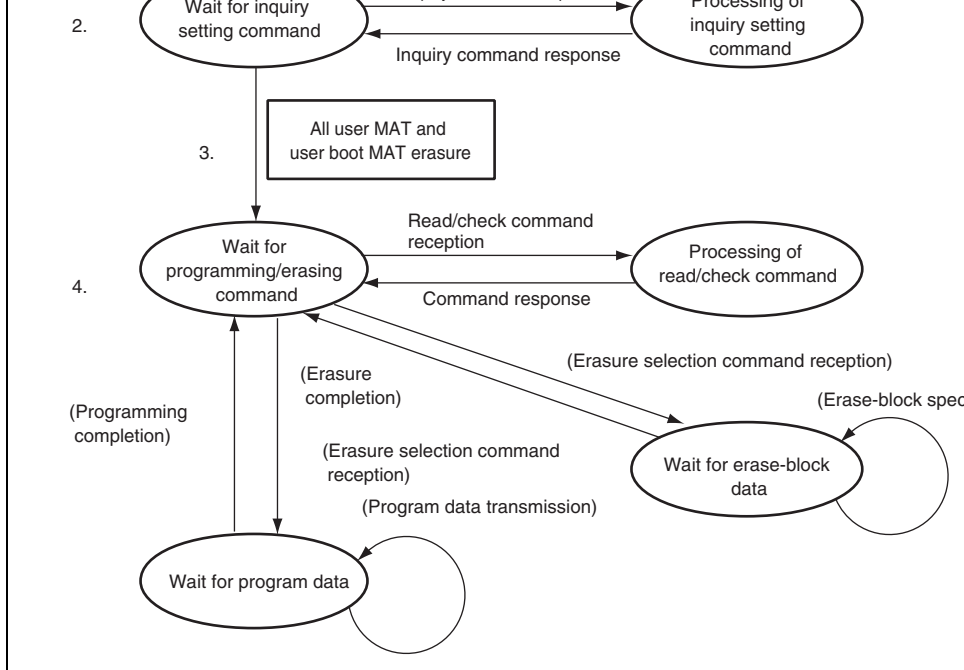


Figure 24.8 Boot Mode State Transition Diagram

waiting for erase block data is entered. The erase block number must be transmitted and an erasing command is transmitted. When the erasure is finished, the erase block number is set to H'FF and transmitted. Then the state of waiting for erase block data is returned to the state of waiting for programming/erasing command. Erasure must be executed when the specified block is programmed without a reset start after programming is executed in programming mode. When programming can be executed by only one operation, all blocks are erased when entering the state of waiting for programming/erasing command or another command. In 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/user boot MAT and acquisition of current status information.

Memory read of the user MAT/user boot MAT can only read the data programmed after the user MAT/user boot MAT has automatically been erased. No other data can be read.

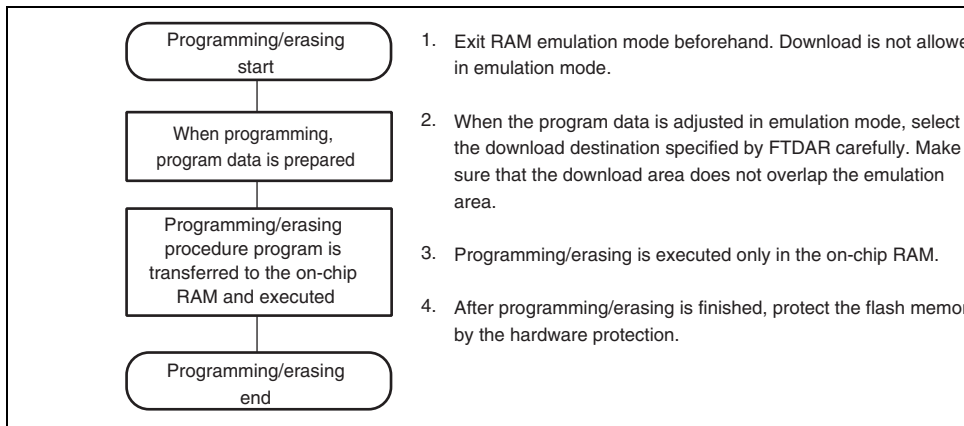


Figure 24.9 Programming/Erasing Flow

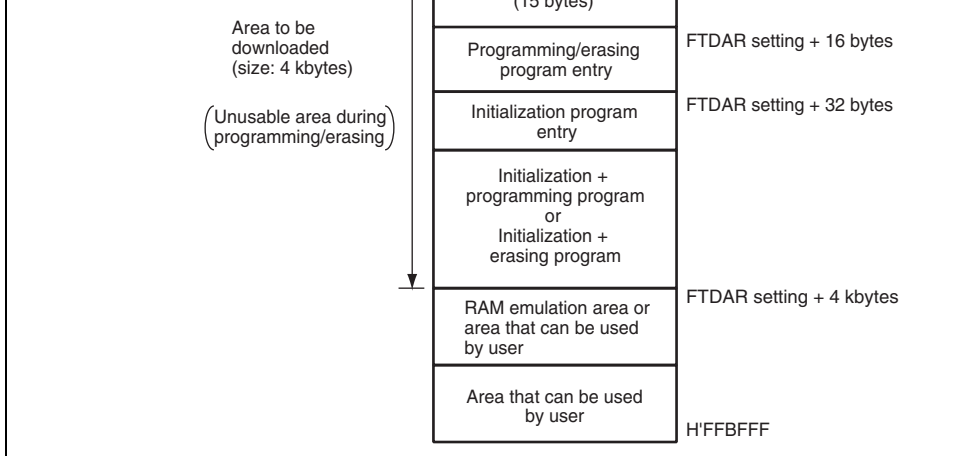


Figure 24.10 RAM Map when Programming/Erasure is Executed

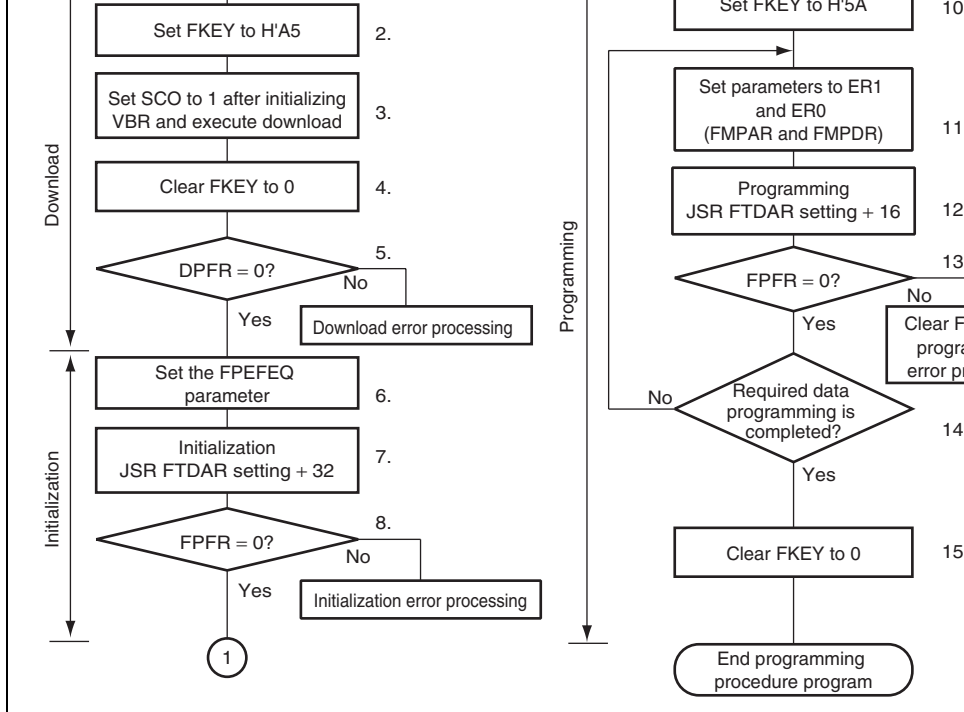


Figure 24.11 Programming Procedure in User Program Mode

H'FF, the program processing time can be shortened.

1. Select the on-chip program to be downloaded and the download destination. When the bit in FPCS is set to 1, the programming program is selected. Several programming/er programs cannot be selected at one time. If several programs are selected, a download returned to the SS bit in the DPFR parameter. The on-chip RAM start address of the 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 be 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 set 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 set to 0 when the procedure program is resumed, the SCO bit cannot be confirmed to be 1 in the procedure program. The download result can be confirmed by the return value of the DPFR parameter. To prevent incorrect decision, before setting the SCO bit to 1, set one byte in the on-chip RAM start address specified by FTDAR, which becomes the DPFR parameter return value other than the return value (e.g. H'FF). Since particular processing that is accompanied by bank switching as described below is performed when download is executed, initialize VBR contents to H'00000000. Dummy read of FCCS must be performed twice immediately 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 by 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 download 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 specified in 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 by the description below.
 - If the value of the DPFR parameter is the same as that before downloading, the start address of the download destination in FTDAR may be abnormal. In this case, confirm the setting of the TDER bit in FTDAR.
 - If the value of the DPFR parameter is different from that before downloading, check the TDER 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 initialization. The settable operating frequency of the FPEFEQ parameter ranges from 8 to 50 MHz. When the operating frequency is set otherwise, an error is returned to the FPFER parameter of the initialization program and initialization is not performed. For details on setting the frequency, see Section 24.7.2 (3), Flash Program/Erase Frequency Parameter (FPEFEQ: General Register E) of the 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 sure program storage area and stack area in the on-chip RAM and register values are not overwritten.
8. The return value in the initialization program, the FPCR parameter is determined.
 9. All interrupts and the use of a bus master other than the CPU are disabled during programming/erasure. The specified voltage is applied for the specified time when programming or erasing. If interrupts occur or the bus mastership is moved to other than CPU during programming/erasure, causing a voltage exceeding the specifications to be applied, the flash memory may be damaged. Therefore, interrupts are disabled by setting bit 1 (I bit) in the condition code register (CCR) to B'1 in interrupt control mode 0 and by setting bits 2 to 0 (I2 to I0 bits) in the extend register (EXR) to B'111 in interrupt control mode 0. Accordingly, interrupts other than NMI are held and not executed. Configure the user MAT so that NMI interrupts do not occur. The interrupts that are held must be executed after programming completes. When the bus mastership is moved to other than the CPU, such as the DMAC or DTC, the error protection state is entered. Therefore, make sure the DMAC does not acquire the bus.
 10. FKEY must be set to H'5A and the user MAT must be prepared for programming.
 11. The parameters required for programming are set. The start address of the programming destination on the user MAT (FMPAR parameter) is set in general register ER1. The start address of the program data storage area (FMPDR parameter) is set in general register ER2.
 - Example of FMPAR parameter setting: When an address other than one in the user MAT area is specified for the start address of the programming destination, even if the programming program is executed, programming is not executed and an error is returned by the FPCR parameter. Since the program data for one programming operation is 128 bytes, the lower eight bits of the address must be H'00 or H'80 to be aligned with the 128-byte boundary.

- The general registers other than LK0 and LK1 are held in the programming program.
- ROL is a return value of the FPFRR parameter.
 - Since the stack area is used in the programming program, a stack area of 128 bytes maximum must be allocated in RAM.
13. The return value in the programming program, the FPFRR parameter is determined.
 14. Determine whether programming of the necessary data has finished. If more than 128 bytes of data are to be programmed, update the FMPAR and FMPDR parameters in 128-byte increments and repeat steps 11 to 14. Increment the programming destination address by 128 bytes and update the programming data pointer correctly. If an address which has already been programmed is written to again, not only will a programming error occur, but also flash memory will be damaged.
 15. After programming finishes, clear FKEY and specify software protection. If this LSI is restarted by a reset immediately after programming has finished, secure the reset input pin (period of $\overline{\text{RES}} = 0$) of at least 100 μs .

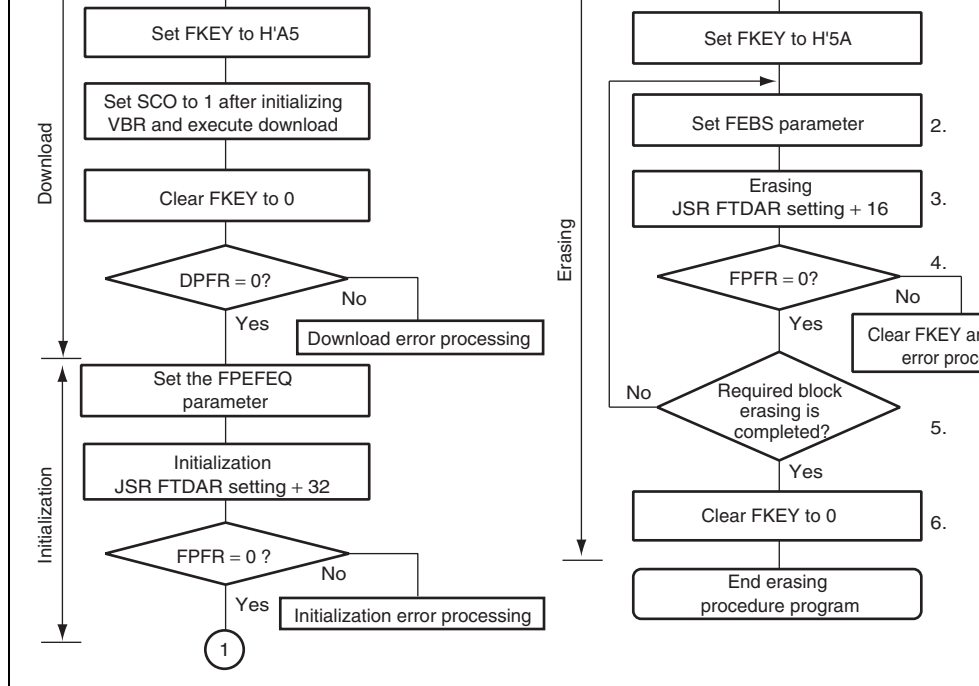


Figure 24.12 Erasing Procedure in User Program Mode

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 download is returned to the SS bit in the DPFPR 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 24.8.2 (2), Programming Procedure in User Program Mode.

2. Set the FEBS parameter necessary for erasure. Set the erase block number (FEBS parameter) of the user MAT in general register ER0. If a value other than an erase block number of user MAT is set, no block is erased even though the erasing program is executed, and the return value is returned to the FPFPR parameter.
3. Erasure is executed. Similar to as in programming, the entry point of the erasing program is the address which is 16 bytes after #DLTOP (start address of the download destination 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 and ER1 are held in the erasing program.
 - ROL is a return value of the FPFPR parameter.
 - Since the stack area is used in the erasing program, a stack area of 128 bytes at the maximum must be allocated in RAM.
4. The return value in the erasing program, the FPFPR parameter is determined.
 5. Determine whether erasure of the necessary blocks has finished. If more than one block is to 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 reset immediately after erasure has finished, secure the reset input period (period of at least 100 μ s).

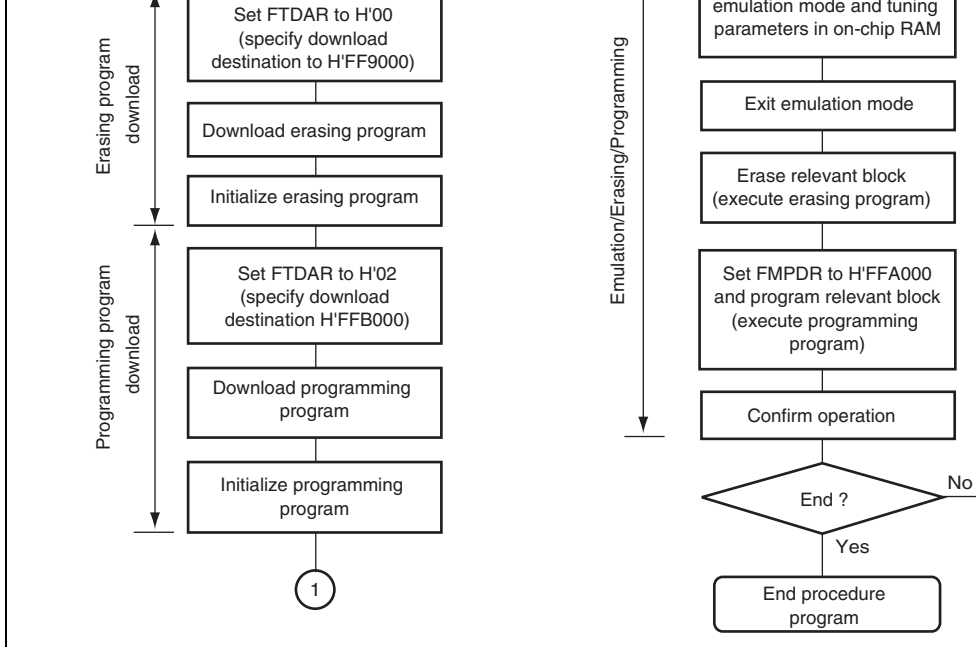


Figure 24.13 Repeating Procedure of Erasing, Programming, and RAM Emulation in User Program Mode

Initialization must be executed for both entry addresses: #DLTOP (start address of download destination for erasing program) + 32 bytes, and #DLTOP (start address of download destination for programming program) + 32 bytes.

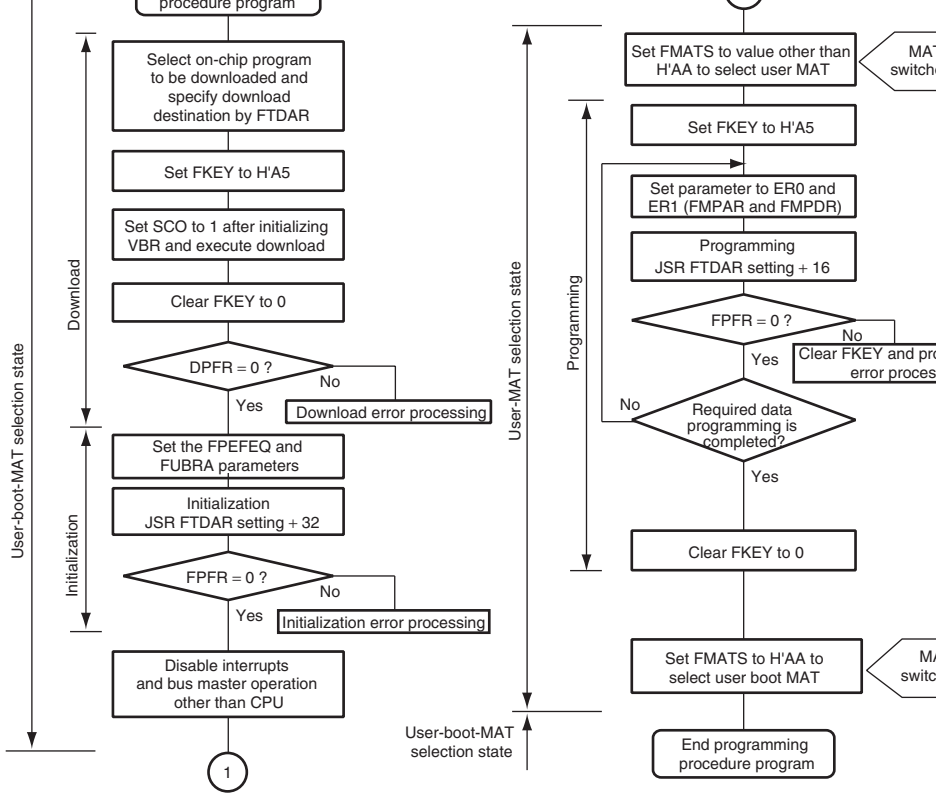
24.8.3 User Boot Mode

Branching to a programming/erasing program prepared by the user enables user boot mode. In this mode, a user-arbitrary boot mode to be used.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing of user boot MAT is only enabled in boot mode or programmer mode.

(1) Initiation in User Boot Mode

When the reset start is executed with the mode pins set to user boot mode, the built-in check routine runs and checks the user MAT and user boot MAT states. While the check routine is running, NMI and all other interrupts cannot be accepted. Next, processing starts from the execution start address of the reset vector in the user boot MAT. At this point, the user boot MAT is selected (FMATS = H'AA) as the execution memory MAT.



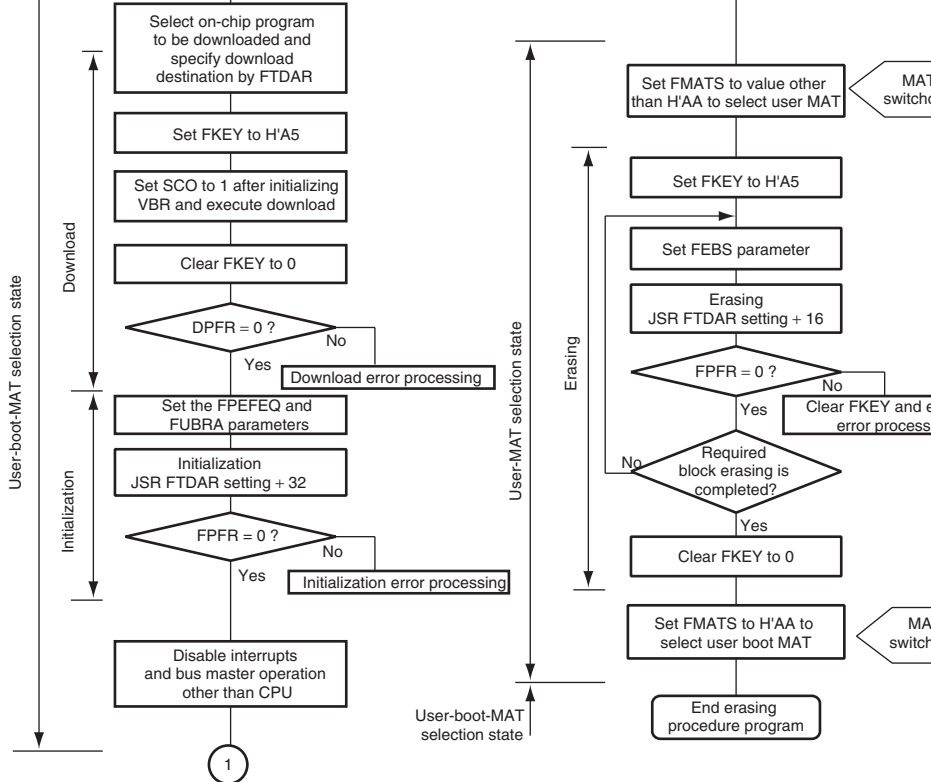
Note: The MAT must be switched by FMATS to perform the programming error processing in the user boot MAT

Figure 24.14 Procedure for Programming User MAT in User Boot Mode

description in section 24.11, Switching between User MAT and User Boot MAT.

Except for memory MAT switching, the programming procedure is the same as that in u program mode.

The area that can be executed in the steps of the procedure program (on-chip RAM, user and external space) is shown in section 24.8.4, On-Chip Program and Storable Area for Data.



Note: The MAT must be switched by FMATS to perform the erasing error processing in the user boot MAT.

Figure 24.15 Procedure for Erasing User MAT in User Boot Mode

24.8.4 On-Chip Program and Storable Area for Program Data

In the descriptions in this manual, the on-chip programs and program data storage areas are assumed to be in the on-chip RAM. However, they can be executed from part of the flash memory which is not to be programmed or erased as long as the following conditions are satisfied.

- The on-chip program is downloaded to and executed in the on-chip RAM specified by the FT-DAR. 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 stack area.
- Download requested by setting the SCO bit in FCCS to 1 should be executed from the on-chip 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 sleep mode, the required procedure programs, NMI handling vector table, and NMI handling routine should be transferred to the on-chip RAM before programming/erasure starts (download is determined).
- The flash memory is not accessible during programming/erasure. Programming/erasure is executed by the program downloaded to the on-chip RAM. Therefore, the procedure programs that initiate operation, the NMI handling vector table, and the NMI handling routines should be stored in the on-chip RAM other than the flash memory.
- After programming/erasure starts, access to the flash memory should be inhibited until the reset signal is cleared. The reset input state (period of $\overline{\text{RES}} = 0$) must be set to at least 100 μs when the operating mode is changed and the reset start is executed on completion of programming/erasure. Transitions to the reset state are inhibited during programming/erasure. When the reset signal is input, a reset input state (period of $\overline{\text{RES}} = 0$) of at least 100 μs is needed before the reset signal is released.

executed are determined by the combination of the processing contents, operating mode, structure of the memory MATs, as shown in tables 24.7 to 24.11.

Table 24.7 Executable Memory MAT

Processing Contents	Operating Mode	
	User Program Mode	User Boot Mode*
Programming	See table 24.8	See table 24.10
Erasing	See table 24.9	See table 24.11

Note: * Programming/Erasure is possible to the user MAT.

FCCS (download)

Operation for clearing FKEY	○	○	○
Decision of download result	○	○	○
Operation for download error	○	○	○
Operation for setting initialization parameter	○	○	○
Execution of initialization	○	×	○
Decision of initialization result	○	○	○
Operation for initialization error	○	○	○
NMI handling routine	○	×	○
Operation for disabling interrupts	○	○	○
Operation for writing H'5A to FKEY	○	○	○
Operation for setting programming parameter	○	×	○
Execution of programming	○	×	○
Decision of programming result	○	×	○
Operation for programming error	○	×	○
Operation for clearing FKEY	○	×	○

Note: * Transferring the program data to the on-chip RAM beforehand enables this a used.

Operation for clearing FKEY	○	○	○
Decision of download result	○	○	○
Operation for download error	○	○	○
Operation for setting initialization parameter	○	○	○
Execution of initialization	○	×	○
Decision of initialization result	○	○	○
Operation for initialization error	○	○	○
NMI handling routine	○	×	○
Operation for disabling interrupts	○	○	○
Operation for writing H'5A to FKEY	○	○	○
Operation for setting erasure parameter	○	×	○
Execution of erasure	○	×	○
Decision of erasure result	○	×	○
Operation for erasure error	○	×	○
Operation for clearing FKEY	○	×	○

FCCS (download)

Operation for clearing FKEY	○	○	○
Decision of download result	○	○	○
Operation for download error	○	○	○
Operation for setting initialization parameter	○	○	○
Execution of initialization	○	×	○
Decision of initialization result	○	○	○
Operation for initialization error	○	○	○
NMI handling routine	○	×	○
Operation for disabling interrupts	○	○	○
Switching memory MATs by FMATS	○	×	○
Operation for writing H'5A to FKEY	○	×	○
Operation for setting programming parameter	○	×	○
Execution of programming	○	×	○
Decision of programming result	○	×	○
Operation for programming error	○	× ^{*2}	○
Operation for clearing FKEY	○	×	○
Switching memory MATs by FMATS	○	×	○

- Notes:
1. Transferring the program data to the on-chip RAM beforehand enables this area to be used.
 2. Switching memory MATs by FMATS by a program in the on-chip RAM enables this area to be used.

Operation for clearing FKEY	○	○	○
Decision of download result	○	○	○
Operation for download error	○	○	○
Operation for setting initialization parameter	○	○	○
Execution of initialization	○	×	○
Decision of initialization result	○	○	○
Operation for initialization error	○	○	○
NMI handling routine	○	×	○
Operation for disabling interrupts	○	○	○
Switching memory MATs by FMATS	○	×	○
Operation for writing H'5A to FKEY	○	×	○
Operation for setting erasure parameter	○	×	○
Execution of erasure	○	×	○
Decision of erasure result	○	×	○
Operation for erasure error	○	×*	○
Operation for clearing FKEY	○	×	○
Switching memory MATs by FMATS	○	×	○

Note: * Switching memory MATs by FMATS by a program in the on-chip RAM enables area to be used.

Table 24.12 Hardware Protection

Item	Description	Function to be Pro	
		Download	Progr Erasing
Reset protection	<ul style="list-style-type: none"> 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 $\overline{\text{RES}}$ pin unless the $\overline{\text{RES}}$ pin is held low until oscillation has settled after a power is initially supplied. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ 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/erasure are disabled unless the required key code is written in FKEY.	○	○
Emulation protection	The programming/erasing protection state is entered when the RAMS bit in the RAM emulation register (RAMER) is set to 1.	○	○

24.9.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when a CPU runaway occurs or operations not according to the programming/erasing procedures are detected during programming/erasure of the flash memory. Aborting programming or erasure in such cases prevents damage to the flash memory due to excessive programming or erasing.

If an error occurs during programming/erasure of the flash memory, the FLER bit in FCCS is set to 1 and the error protection state is entered.

- When an interrupt request, such as NMI, occurs during programming/erasure.
- When the flash memory is read from during programming/erasure (including a vector search or an instruction fetch).
- When a SLEEP instruction is executed (including software-standby mode) during programming/erasure.
- When a bus master other than the CPU, such as the DMAC and DTC, obtains bus master during programming/erasure.

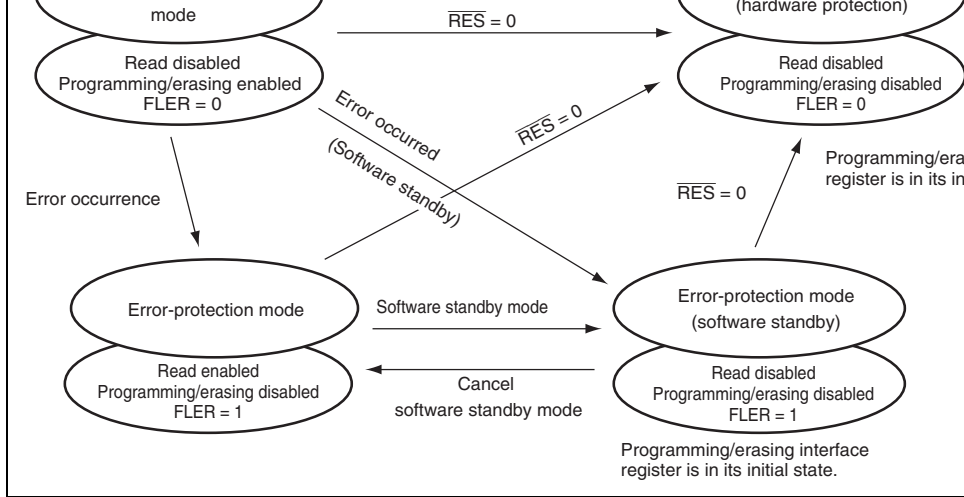


Figure 24.16 Transitions to Error Protection State

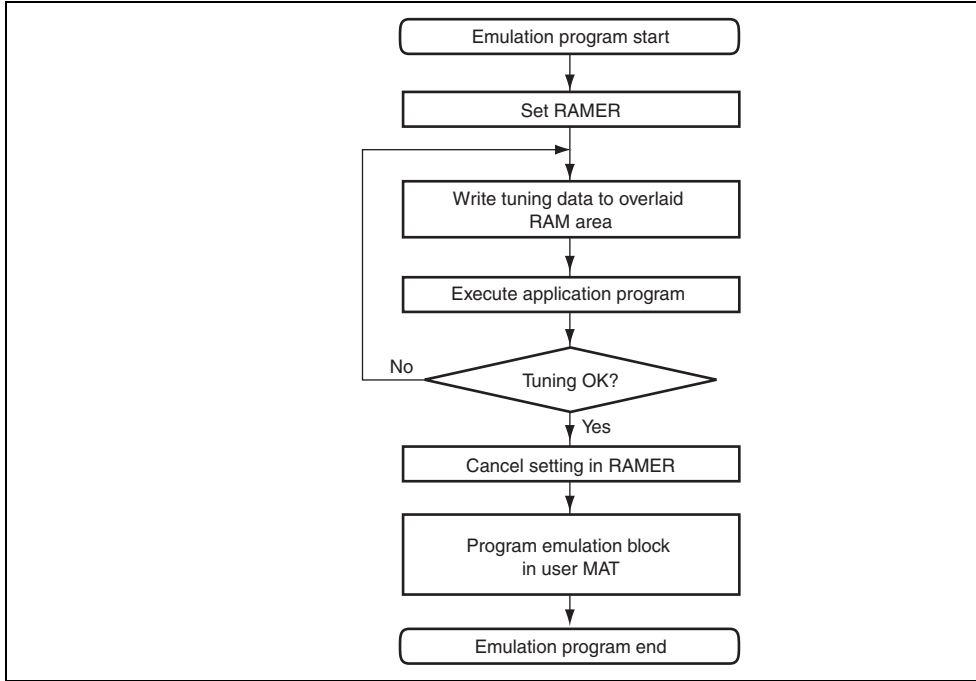


Figure 24.17 RAM Emulation Flow

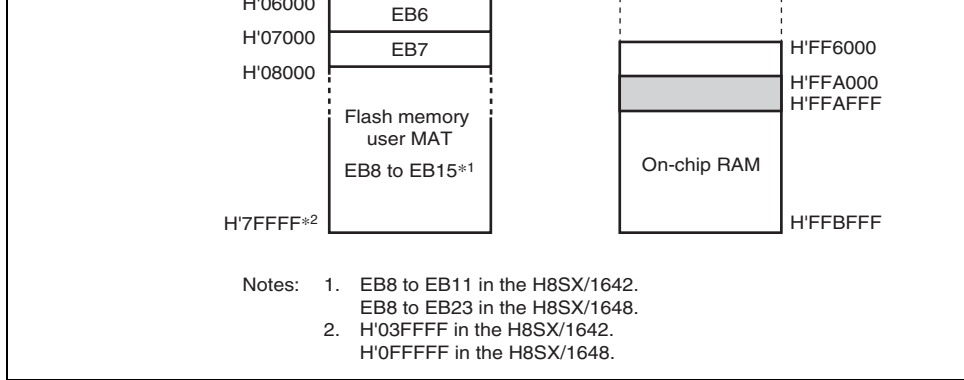


Figure 24.18 Address Map of Overlaid RAM Area (H8SX/1644)

The flash memory area that can be emulated is the one area selected by bits RAM2 to RAM0 and 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 RAMER to 1 and bits RAM2 to RAM0 to B'000.

For programming/erasing the user MAT, the procedure programs including a download of the on-chip program must be executed. At this time, the download area should be specified so that the overlaid RAM area is not overwritten by downloading the on-chip program. Since the overlaid RAM area in which the tuned data is stored is overlaid with the download area when FTDAR = H'0000, the tuned data must be saved in an unused area beforehand.

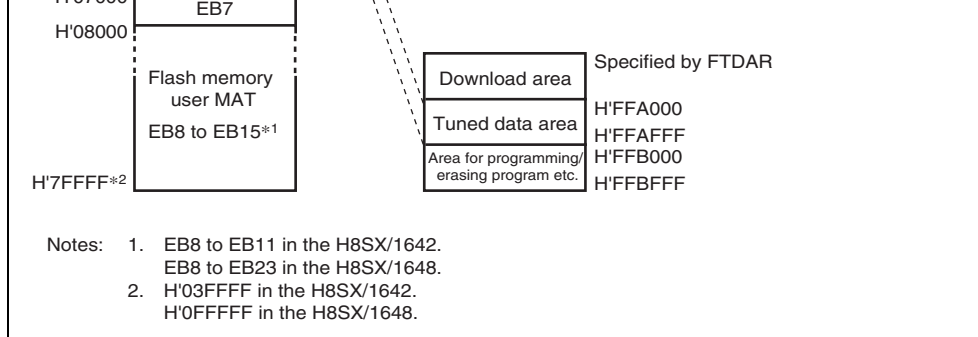


Figure 24.19 Programming Tuned Data (H8SX/1644)

1. After tuning program data is completed, clear the RAMS bit in RAMER to 0 to cancel 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. The address of the download destination should be specified by FTDAR so that the tuned data does not overlay the download area.
4. When block EB0 of the user MAT has not been erased, the programming program must be downloaded after block EB0 is erased. Specify the tuned data saved in the FMPAR and 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 the values of the RAM2 to RAM0 bits. Under this condition, the on-chip program cannot be downloaded. When data is to be actually programmed and erased, clear the RAMS bit to 0.

- for eight times (this prevents access to the flash memory during memory MAT switching).
3. If an interrupt request has occurred during memory MAT switching, there is no guarantee which memory MAT is accessed. Always mask the maskable interrupts before switching memory MATs. In addition, configure the system so that NMI interrupts do not occur during memory MAT switching.
 4. After the memory MATs have been switched, take care because the interrupt vector addresses also have been switched. If interrupt processing is to be the same before and after memory MAT switching, transfer the interrupt processing routines to the on-chip RAM and store the VBR to place the interrupt vector table in the on-chip RAM.
 5. The size of the user MAT is different from that of the user boot MAT. Addresses within the size of the 16-Kbyte user boot MAT should not be accessed. If an attempt is made to read as an undefined value.

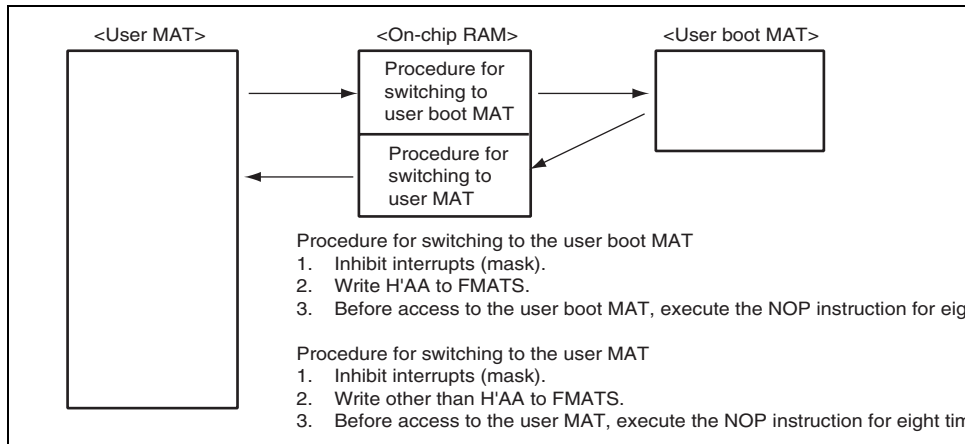


Figure 24.20 Switching between User MAT and User Boot MAT

	H8SX/1644	512 Kbytes	FZTAT512V3A
	H8SX/1648	1 Mbyte	FZTAT1024V3
User boot MAT	H8SX/1642	16 Kbytes	FZTATUSBT1
	H8SX/1644		
	H8SX/1648		

24.13 Standard Serial Communications Interface Specifications for Mode

The boot program initiated in boot mode performs serial communications using the host and the chip SCI_4. The serial communications 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 communications with the host. Initiating boot mode enables starting of the boot program and entry to the bit-rate-adjustment state. The program receives the command from the host to adjust the bit rate. After 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 device clock mode, and bit rate are selected. After selection of these settings, the program is instructed to 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 the chip RAM and erases the user MATs and user boot MATs before the transition.

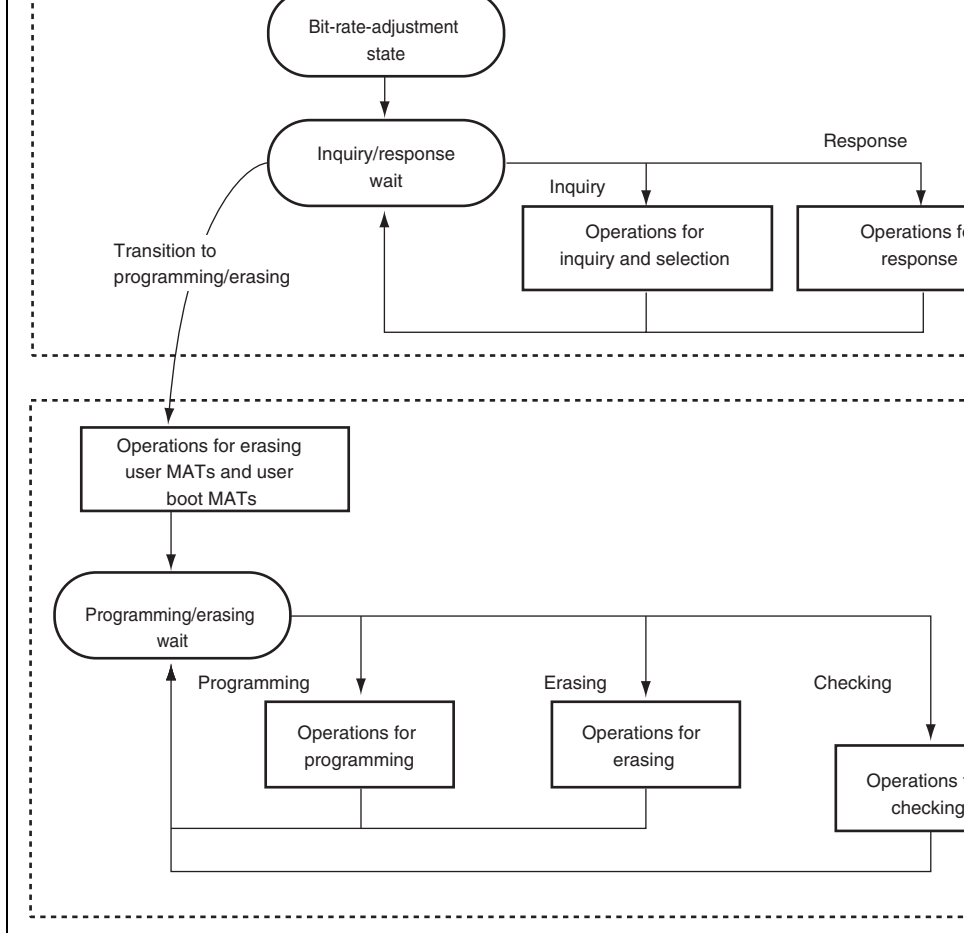


Figure 24.21 Boot Program States

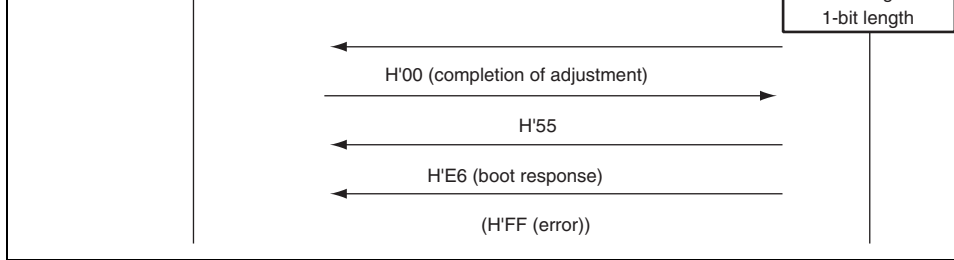


Figure 24.22 Bit-Rate-Adjustment Sequence

(2) Communications Protocol

After adjustment of the bit rate, the protocol for serial communications between the host and the 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 ACK responses after successful completion.

2. n-byte commands or n-byte responses

These commands and responses are comprised of n bytes of data. These are selections of data and responses to inquiries.

The program data size is not included under this heading because it is determined in a programming unit inquiry command.

3. Error response

The error response is a response to inquiries. It consists of an error response and an error acknowledgment 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.

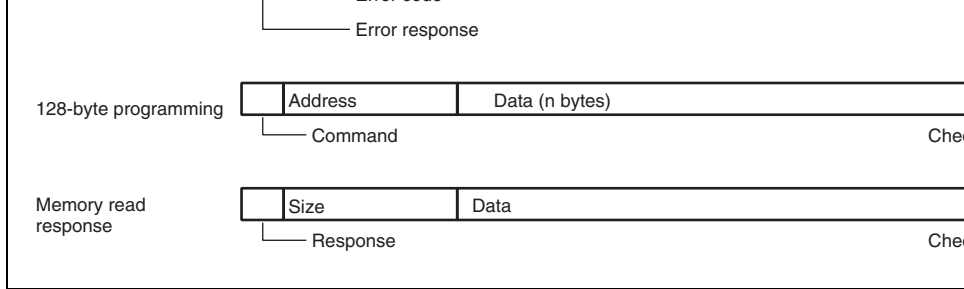


Figure 24.23 Communication Protocol Format

- **Command (one byte):** Commands including inquiries, selection, programming, erasing, and checksum checking
- **Response (one byte):** Response to an inquiry
- **Size (one byte):** The amount of data for transmission excluding the command, amount of data, and checksum
- **Checksum (one byte):** The checksum is calculated so that the total of all values from the 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 and 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 multiplication ratios, and the values of multiple
H'23	Operating clock frequency inquiry	Inquiry regarding the maximum and minimum values of the main clock and peripheral
H'24	User boot MAT information inquiry	Inquiry regarding the number of user boot MATs and the start and last addresses of each MAT
H'25	User MAT information inquiry	Inquiry regarding the a number of user MATs and the start and last addresses of each
H'26	Block for erasing information Inquiry	Inquiry regarding the number of blocks and the start and last addresses of each
H'27	Programming unit inquiry	Inquiry regarding the unit of programming
H'3F	New bit rate selection	Selection of new bit rate
H'40	Transition to programming/erasing state	Erasing of user MAT and user boot MAT and entry to programming/erasing state
H'4F	Boot program status inquiry	Inquiry into the operated status of the boot program

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, and checksum, that is, the amount of data contributed by the number of devices, character codes and product names
- Number of devices (one byte): The number of device types supported by the boot program
- Number of characters (one byte): The number of characters in the device codes and the boot 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 and response, including 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 inquiry.

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 command
- 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 mode can be selected using these respective values.

Number of multiplication ratios	Multiplication ratio	...				
...						
SUM						

- Response, H'32, (one byte): Response to the multiplication ratio inquiry
- Size (one byte): The amount of data that represents the number of types of multiplication ratios, and the multiplication ratios
- Number of types of multiplication (one byte): The number of types of multiplication that the device can be set
(e.g. when there are two multiplied clock types, which are the main and peripheral clock, the number of types will be H'02.)
- Number of multiplication ratios (one byte): The number of types of multiplication ratios for each type
(e.g. the number of multiplication ratios to which the main clock can be set and the peripheral clock can be set.)
- Multiplication ratio (one byte)
 Multiplication ratio: The value of the multiplication ratio (e.g. when the clock-frequency 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 division ratio is 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 multiplication ratios and as many groups of data are returned as there are types of multiplication.
- SUM (one byte): Checksum

...	
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 values, and the number of frequencies.
- Number of operating clock frequencies (one byte): The number of supported operating clock frequency types
(e.g. when there are two operating clock frequency types, which are the main and peripheral clocks, the number of types will be H'02.)
- Minimum value of operating clock frequency (two bytes): The minimum value of the multiplied or divided clock frequency.
The minimum and maximum values of the operating clock frequency represent the value in MHz, valid to the hundredths place of MHz, and multiplied by 100. (e.g. when the value is 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 clock frequencies.
- SUM (one byte): Checksum

- Response, H'34, (one byte): Response to user boot MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start address and area-last address
- Number of Areas (one byte): The number of consecutive user boot MAT areas
When user boot MAT areas are consecutive, the number of areas returned is H'01.
- Area-start address (four byte): Start address of the area
- Area-last address (four byte): Last address of the area
There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

(h) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses.

Command

H'25

- Command, H'25, (one byte): Inquiry regarding user MAT information

Response	H'35	Size	Number of areas	
	Start address area			Last address area
	...			
	SUM			

- 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 address 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

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-start 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
- Block last Address (four bytes): Last address of a block
There are as many groups of data representing the start and last addresses as there are blocks.
- SUM (one byte): Checksum

(j) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command

H'27

- Command, H'27, (one byte): Inquiry regarding programming unit

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 fixed at 1.
- Programming unit (two bytes): A unit for programming
This is the unit for reception of programming.
- SUM (one byte): Checksum

- Size (one byte): The number of bytes that represents the bit rate, input frequency, number of types of multiplication, 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 is 19200/100)
- 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 100 (e.g. when the value is 20.00 MHz, it will be 2000, which is 20.00*100.)
- Number of types of multiplication (one byte): The number of multiplication to which the device can be set.
- Multiplication ratio 1 (one byte): The value of multiplication or division ratios for the operating frequency
Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04.)
Division ratio: The inverse of the division ratio, as a negative number (e.g. when the clock frequency is divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- Multiplication ratio 2 (one byte): The value of multiplication or division ratios for the peripheral frequency
Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04.)
(Division ratio: The inverse of the division ratio, as a negative number (E.g. when the clock frequency is divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- SUM (one byte): Checksum

Response

- 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.

The frequency is not within the specified range.

(4) Receive Data Check

The methods for checking of receive data are listed below.

1. Input frequency

The received value of the input frequency is checked to ensure that it is within the minimum to maximum frequencies which matches the clock modes of the specified device. When the value is out of this range, an input-frequency error is generated.

2. Multiplication ratio

The received value of the multiplication ratio or division ratio is checked to ensure that it matches the clock modes of the specified device. When the value is out of this range, a multiplication ratio or division ratio error is generated.

3. Operating frequency error

Operating frequency is calculated from the received value of the input frequency and the multiplication or division ratio. The input frequency is input to the LSI and the LSI operates at the operating frequency. The expression is given below.

Operating frequency = Input frequency \times Multiplication ratio, or

Operating frequency = Input frequency \div 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 specified device. When it is out of this range, an operating frequency error is generated.

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 boot 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

The sequence of new bit-rate selection is shown in figure 24.24.

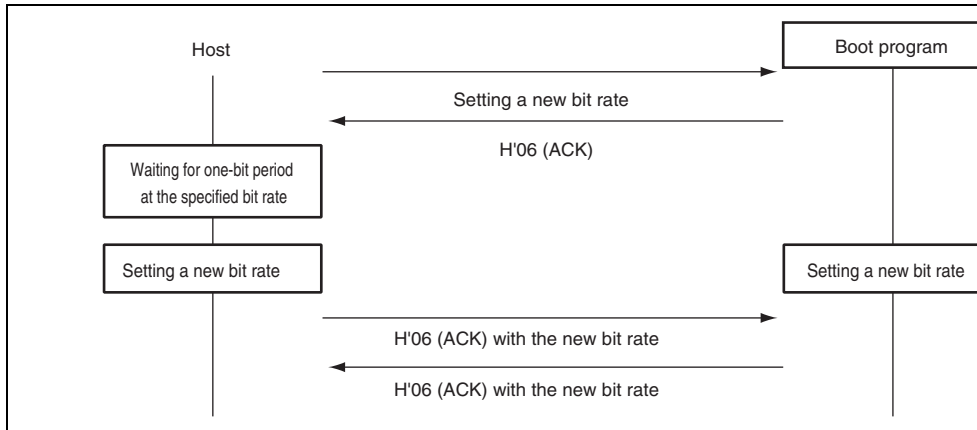


Figure 24.24 New Bit-Rate Selection Sequence

- 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 MAT and user boot MAT have been
by the transferred erasing program.

Error Response

H'C0

H'51

- Error response, H'C0, (one byte): Error response for user boot MAT blank check
- Error code, H'51, (one byte): Erasing error
An error occurred and erasure was not completed.

(6) Command Error

A command error will occur when a command is undefined, the order of commands is incorrect, or a command is unacceptable. Issuing a clock-mode selection command before a device ID command or an inquiry command after the transition to programming/erasing state command, are

Error Response

H'80

H'xx

- Error response, H'80, (one byte): Command error
- Command, H'xx, (one byte): Received command

be made, such as the multiplication-ratio inquiry (H'22) or operating frequency inquiry (H'23) 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, according to the returned information on multiplication ratios and operating frequencies.
7. After selection of the device and clock mode, the information of the user boot MAT and the user boot MAT should be made to inquire about the user boot MATs information inquiry (H'24), user boot MATs information inquiry (H'25), erased block information inquiry (H'26), and program 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'43	User MAT programming selection	Transfers the user MAT programming program
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'4A	User boot MAT sum check	Checks the checksum of the user boot MAT
H'4B	User MAT sum check	Checks the checksum of the user MAT
H'4C	User boot MAT blank check	Checks the blank data of the user boot MAT
H'4D	User MAT blank check	Checks the blank data of the user MAT
H'4C	User boot MAT blank check	Checks whether the contents of the user boot MAT are blank
H'4D	User MAT blank check	Checks whether the contents of the user MAT are blank
H'4F	Boot program status inquiry	Inquires into the boot program's status

command represents the data programmed according to the method specified by the selection 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 will wait for selection of programming or erasing.

Where the sequence of programming operations that is executed includes programming by another method or of another MAT, the procedure must be repeated from the programming selection command.

The sequence for the programming selection and 128-byte programming commands is shown in figure 24.25.

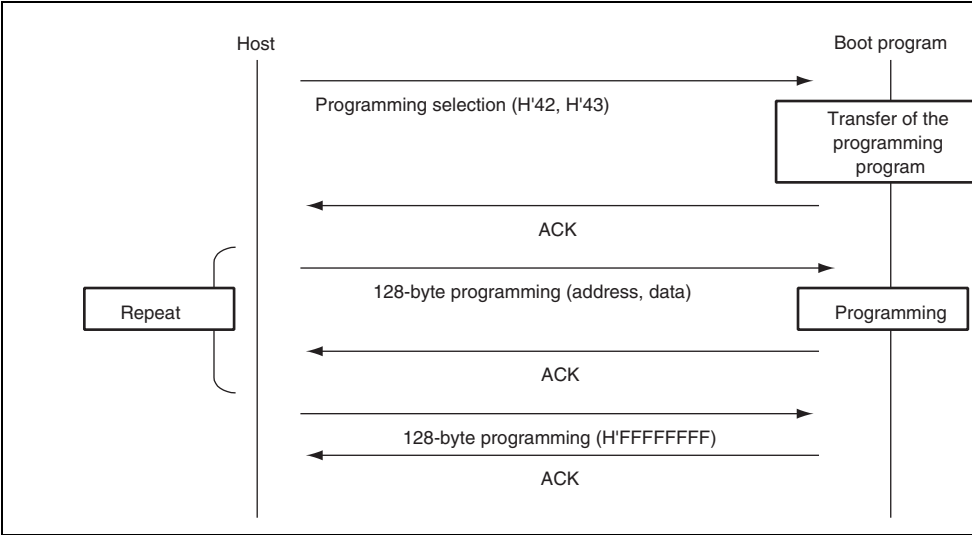


Figure 24.25 Programming Sequence

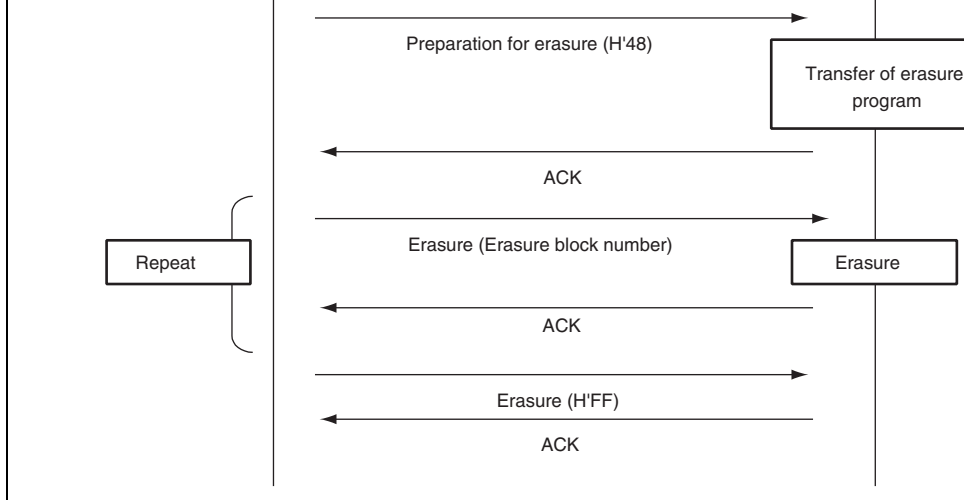


Figure 24.26 Erasure Sequence

Error Response

H'C2	ERROR
------	-------

- Error response : H'C2 (1 byte): Error response to user boot MAT programming selection
- ERROR : (1 byte): Error code
H'54: Selection processing error (transfer error occurs and processing is not completed)

(b) User MAT Programming Selection

The boot program will transfer a program for user MAT programming selection. The data programmed to the user MATs by the transferred program for programming.

Command

H'43

- Command, H'43, (one byte): User MAT programming selection

Response

H'06

- Response, H'06, (one byte): Response to user MAT programming selection
When the programming program has been transferred, the boot program will return A

Error Response

H'C3	ERROR
------	-------

- Error response : H'C3 (1 byte): Error response to user MAT programming selection
- ERROR : (1 byte): Error code
H'54: Selection processing error (transfer error occurs and processing is not completed)

- 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.

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'2A: Address error

The address is not in the specified MAT.

H'53: Programming error

A programming error has occurred and programming cannot be continued.

The specified address should match the unit for programming of data. For example, when programming is in 128-byte units, the lower eight bits of the address should be H'00 or H'80. When there are less than 128 bytes of data to be programmed, the host should fill the rest with H'FF.

Sending the 128-byte programming command with the address of H'FFFFFFFF will stop the programming operation. The boot program will interpret this as the end of the programming and wait for selection of programming or erasing.

- 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 completed.

(d) Erasure Selection

The boot program will transfer the erasure program. User MAT data is erased by the transfer of the erasure program.

Command

H'48

- Command, H'48, (one byte): Erasure selection

Response

H'06

- Response, H'06, (one byte): Response for erasure selection
After the erasure program has been transferred, the boot program will return ACK.

Error Response

H'C8	ERROR
------	-------

- Error Response, H'C8, (one byte): Error response to erasure selection
- ERROR: (one byte): Error code
 - H'54: Selection processing error (transfer error occurs and processing is not completed)

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.

On receiving block number H'FF, the boot program will stop erasure and wait for a selected command.

Command

H'58	Size	Block number	SUM
------	------	--------------	-----

- Command, H'58, (one byte): Erasure
- Size, (one byte): The number of bytes that represents the block number
This is fixed to 1.
- Block number (one byte): H'FF
Stop code for erasure
- SUM (one byte): Checksum

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 program should be executed from the erasure selection command.

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

Error Response	H'D2	ERROR
----------------	------	-------

- Error response: H'D2 (1 byte): Error response to memory read
- ERROR: (1 byte): Error code

H'11: Sum check error

H'2A: Address error

The read address is not in the MAT.

H'2B: Size error

The read size exceeds the MAT.

This is fixed to 4.

- Checksum of user boot program (four bytes): Checksum of user boot MATs
The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

(h) User MAT Sum Check

The boot program will return the byte-by-byte total of the contents of the bytes of the user 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

Error Response

H'CC	H'52
------	------

- Error Response, H'CC, (one byte): Response to blank check for user boot MAT
- Error Code, H'52, (one byte): Erasure has not been completed.

(j) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the result.

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 24.17 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 turn off the Vcc power supply nor remove the chip from the PROM programming/erasure in which a high voltage is applied to the flash memory. Doing damage the flash memory permanently. If a reset is input, the reset must be released reset input period of at least 100µs.
6. The flash memory is not accessible until FKEY is cleared after programming/erasure the operating mode is changed and this LSI is restarted by a reset immediately after programming/erasure has finished, secure the reset input period (period of $\overline{\text{RES}} = 0$) 100µs. Transition to the reset state during programming/erasure is inhibited. If a reset accidentally, the reset must be released after the reset input period of at least 100µs.
7. At powering on the Vcc power supply, fix the $\overline{\text{RES}}$ pin to low and set the flash memory hardware protection state. This power on timing must also be satisfied at a power-off 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 programming/erasure in on-board programming mode, it is recommended that automatic programming performed after execution of automatic erasure.
10. To program the flash memory, the program data and program must be allocated to addresses which are higher than those of the external interrupt vector table and H'FF must be written to all the system reserved areas in the exception handling vector table.
11. The programming program that includes the initialization routine and the erasing program that includes the initialization routine are each 4 Kbytes or less. Accordingly, when the clock frequency is 35 MHz, the download for each program takes approximately 60 µs at the maximum.

immediately after setting it to 1. Otherwise, downloads 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 general registers ER0 and ER1 are not saved during download of an c
program, initialization, programming, or erasure. When needed, save the general regis
before a download request or before execution of initialization, programming, or eras
the procedure program.

valid

- Six test modes:
BYPASS mode
EXTEST mode
SAMPLE/PRELOAD mode
CLAMP mode
HIGHZ mode
IDCODE mode

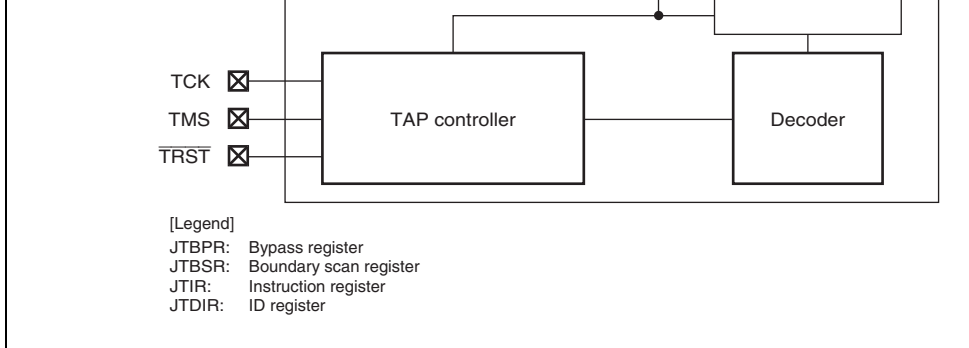


Figure 25.1 Block Diagram of Boundary Scan Function

25.3 Input/Output Pins

Table 25.1 shows the I/O pins used in the boundary scan function.

Table 25.1 Pin Configuration

Pin Name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. Input the clock the duty cycle of which is 50 percent when boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin
TRST	Input	Test reset input pin

TDI and TDO pins in BYPASS mode. The boundary scan register (JTBSR), which is a boundary scan register, is connected between the TDI and TDO pins when test data are being shifted in. The registers is accessible from the CPU.

Table 25.2 shows the availability of serial transfer for the registers.

Table 25.2 Serial Transfers for Registers

Register Abbreviation	Serial Input	Serial Output
JTIR	Available	Not available
JTBPR	Available	Available
JTBSR	Available	Available
JTID	Not available	Available

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

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 12	TS[3:0]	All 0	R/W	Test Bit Set Specify an instruction as shown in table 25.3.
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value is always 0.

0	1	1	1	Reserved
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	BYPASS

25.4.2 Bypass Register (JTBPR)

JTBPR is a 1-bit register and is connected between the TDI and TDO pins when JTIR is in BYPASS mode. JTBPR cannot be read from or written to by the CPU.

25.4.3 Boundary Scan Register (JTBSR)

JTBSR is a shift register to control the external input and output pins of this LSI and is connected across the pads. The initial values are undefined. JTBSR cannot be accessed by the CPU. EXTTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ instructions are issued to apply boundary-scan testing conformant to the JTAG standard.

Table 25.4 shows the correspondence between the JTBSR bits and the pins of this LSI.

			Output enable	292				Output enable
			Output	291				Output
7	D2	MD2	Input	290	15	G3	PF3	Input
—	E4	PM0	Input	289				Output enable
			Output enable	288				Output
			Output	287	17	H4	PF2	Input
8	D1	PN0	Input	286				Output enable
—	E3	PM1	Input	285				Output
			Output enable	284	18	H3	PF1	Input
			Output	283				Output enable
9	E2	PN1	Input	282				Output
—	E1	PM2	Input	281	19	H1	PF0	Input
			Output enable	280				Output enable
			Output	279				Output
10	F4	PC5	Input	278	20	H2	PE7	Input
			Output enable	277				Output enable
			Output	276				Output
11	F3	PF7	Input	275	21	J4	PE6	Input
			Output enable	274				Output enable
			Output	273				Output
12	F1	PF6	Input	272	22	J3	PE5	Input
			Output enable	271				Output enable
			Output	270				Output

			Input	235				Input
			Output enable	234				Output enable
28	L3	PE1	Input	233	38	P2	PD0	Input
			Output enable	232				Output enable
			Output	231				Output
29	L1	PE0	Input	230	40	N3	PN2	Input
			Output enable	229	—	R2	PM3	Input
			Output	228				Output enable
30	L2	PD7	Input	227				Output
			Output enable	226	41	P3	PN3	Input
			Output	225	—	N4	PM4	Input
31	L4	PD6	Input	224				Output enable
			Output enable	223				Output
			Output	222	45	P6	PC0	Input
33	M3	PD5	Input	221				Output enable
			Output enable	220				Output
			Output	219	46	N7	PC1	Input
34	N1	PD4	Input	218				Output enable
			Output enable	217				Output
			Output	216				

			Input	180	62	R11	P33	Input
			Output enable	179				Output enable
52	P8	P22	Input	178				Output
			Output enable	177	63	P11	P34	Input
			Output	176				Output enable
53	M9	P23	Input	175				Output
			Output enable	174	65	R12	PH0	Input
			Output	173				Output enable
54	N9	P24	Input	172				Output
			Output enable	171	66	P12	PH1	Input
			Output	170				Output enable
55	R9	P25	Input	169				Output
			Output enable	168	67	N12	PH2	Input
			Output	167				Output enable
56	P9	P30	Input	166				Output
			Output enable	165	68	R13	PH3	Input
			Output	164				Output enable
57	M10	P31	Input	163				Output
			Output enable	162	73	R15	PH7	Input
			Output	161				Output enable
58	N10	P32	Input	160				Output
			Output enable	159				
			Output	158				

			Input	120				Input
			Output enable	122				Output enable
			Output	121				Output
76	N15	PI1	Input	120	86	K14	P12	Input
			Output enable	119				Output enable
			Output	118				Output
77	M14	PI2	Input	117	87	J12	P13	Input
			Output enable	116				Output enable
			Output	115				Output
78	L12	PI3	Input	114	89	J14	P66	Input
			Output enable	113				Output enable
			Output	112				Output
75	M13	PI0	Input	111	90	H12	P67	Input
			Output enable	110				Output enable
			Output	109				Output
80	L13	PI4	Input	108	93	G15	P14	Input
			Output enable	107				Output enable
			Output	106				Output
81	L14	PI5	Input	105	94	G14	P15	Input
			Output enable	104				Output enable
			Output	103				Output
83	K12	PI7	Input	102	100	E14	P16	Input
			Output enable	101				Output enable
			Output	100				Output

			Input	60				Input
			Output enable	62	134	C5	PA0	Input
			Output	61				Output enable
107	C14	P60	Input	60				Output
			Output enable	59	135	A5	PA1	Input
			Output	58				Output enable
106	D12	P37	Input	57				Output
			Output enable	56	136	B5	PA2	Input
			Output	55				Output enable
108	B15	P61	Input	54				Output
			Output enable	53	137	D5	PA3	Input
			Output	52				Output enable
115	B12	MD0	Input	51				Output
116	D11	PC2	Input	50	138	A4	PA4	Input
			Output enable	49				Output enable
			Output	48				Output
117	A12	PC3	Input	47	139	B4	PA5	Input
			Output enable	46				Output enable
			Output	45				Output
133	B7	MD1	Input	44	140	C4	PA6	Input
42	M6	PB4	Input	43				Output enable
			Output enable	42				Output
			Output	41				

			Input	3
			Output enable	5
			Output	4
2	C3	PB2	Input	3
			Output enable	2
			Output	1
to TDO				

Bit	Bit Name	Initial Value	R/W	Descriptions
31 to 0	DID31 to DID0	H'0803A447	—	JTID is a register the value showing the decide IDCODE is fixed.



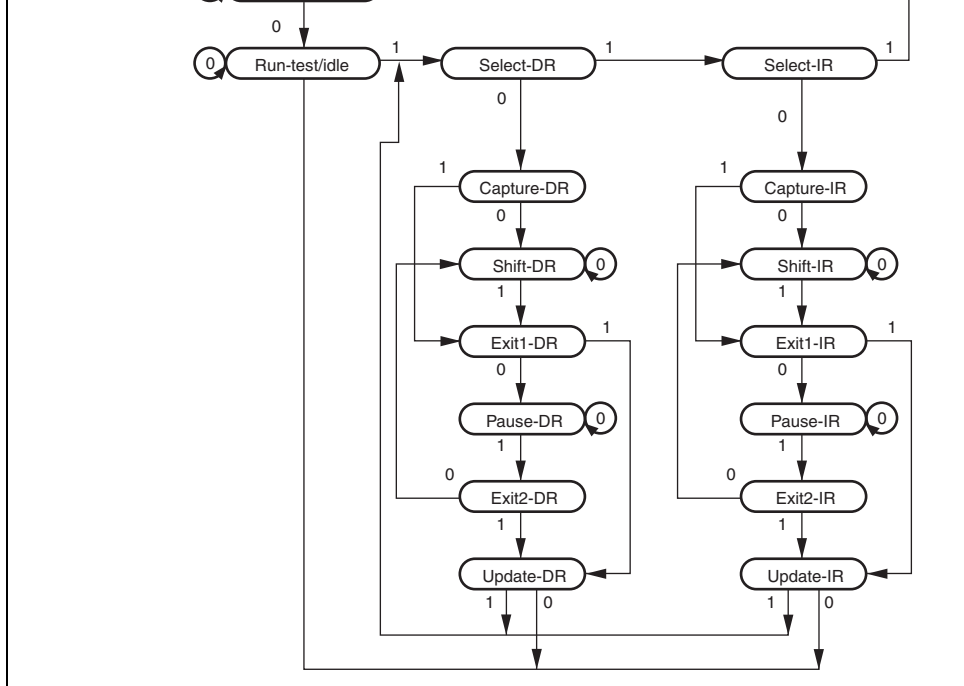


Figure 25.2 State Transitions of the TAP Controller

EXTEST (Instruction Code: B'0000): The EXTEST instruction is used to test external pins when this LSI is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified by the SAMPLE/PRELOAD instruction) from the boundary scan register to the print circuit board, and input pins are used to input test result.

SAMPLE/PRELOAD (Instruction Code: B'0100): The SAMPLE/PRELOAD instruction is used to input data from the LSI internal circuits to the boundary scan register, output data from the scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to the LSI and output signals are also directly output to the external circuits. The system circuit is not affected by this function.

In SAMPLE operation, the boundary scan register latches the snap shot of data transferred from input pins to internal circuit or data transferred from internal circuit to output pins. The latched data is read from the scan path. The scan register latches the snap data at the rising edge of TCK in Capture-DR state. The scan register latches snap shot without affecting the LSI normal operation.

In PRELOAD operation, initial value is written from the scan path to the parallel output latches of the boundary scan register prior to the EXTEST instruction execution. If the EXTEST instruction is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In EXTEST instruction, only the parallel latches are always output to the output pins.)

IDCODE (Instruction Code: B'0001): When the IDCODE instruction is selected, IDCODE register value is output to the TDO in Shift-DR state of the TAP controller. In this case, IDCODE register value is output from the LSB. During this instruction execution, test circuit does not affect the system circuit. INSTR is initialized by the IDCODE instruction in Test-Logic-Reset state of the TAP controller.

register is maintained regardless of the state of the TAP controller.
BYPASS is connected between TDI and TDO pins, leading to the same operation as when the BYPASS instruction has been selected.

25.6 Usage Notes

1. In serial transfer, data are input or output in LSB order (see figure 25.3).

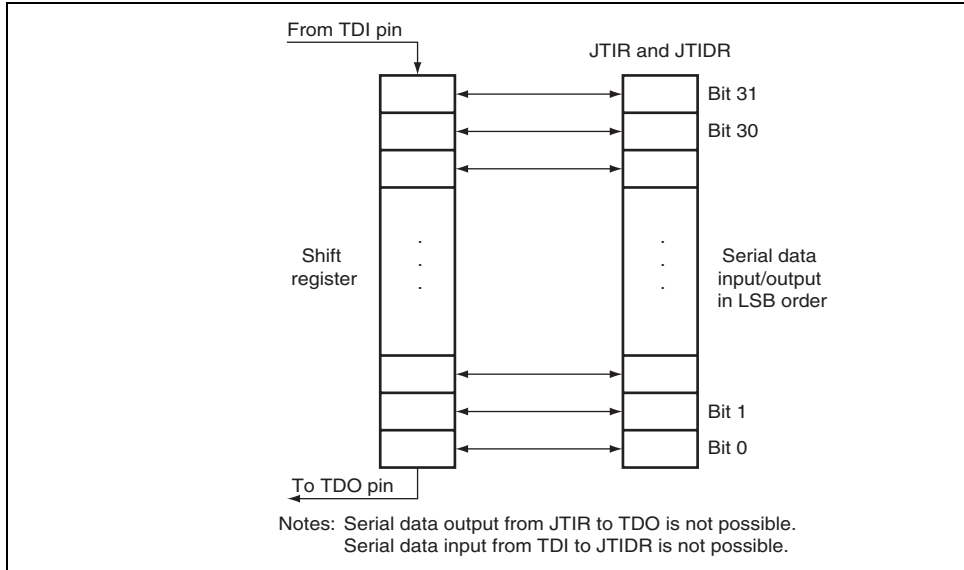


Figure 25.3 Serial Data Input/Output

7. Boundary scan pins (TCK, TMS, TRST, TDI, and TDO) cannot be boundary-scanned.
8. The boundary scan function is not available when this LSI are in the following states.
 - (1) Reset state
 - (2) Hardware standby mode, software standby mode, and deep software standby mode

changes the frequency through the setting of the system clock control register (SCKCR) and subclock control register (SUBCKCR)*.

This LSI supports four clocks: a system clock provided to the CPU and bus masters, a peripheral module clock provided to the peripheral modules, an external bus clock provided to the external bus, and a 32K timer clock*. Frequencies of the peripheral module clock, the external bus clock, and the system clock can be set independently, although the peripheral module clock and the external bus clock operate with the frequency lower than the system clock frequency.

The system clock, peripheral module clock, and external bus clock can be uniformly set to 32.768 kHz subclock.*

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

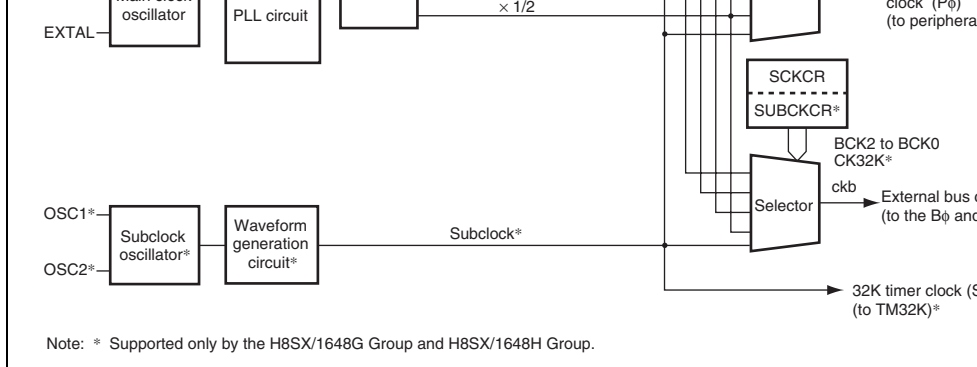


Figure 26.1 Block Diagram of Clock Pulse Generator

Table 26.1 Selection of Clock Pulse Generator

EXTAL Input Clock Frequency	I ϕ /P ϕ /B ϕ
8 MHz to 18 MHz	EXTAL $\times 4$, $\times 2$, $\times 1$, $\times 1/2$

SCKCR controls B ϕ output control and frequencies of the system, peripheral module, and 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	B ϕ Clock Output Enable Controls ϕ output on PA7. <ul style="list-style-type: none"> Normal operation 0: ϕ output 1: Fixed high
14	PSTOP0*	0	R/W	ϕ Clock Output Enable Controls ϕ output (SDRAM ϕ) on PB7. <ul style="list-style-type: none"> Normal operation 0: ϕ output 1: Fixed high

000: × 4

001: × 2

010: × 1

011: × 1/2

1xx: Setting prohibited

The frequencies of the peripheral module clock and external bus clock change to the same frequency as the system clock if the frequency of the system clock is lower than that of the two clocks.

7	—	0	R/W	Reserved
				Although this bit is readable/writable, only 0 should be written to.
6	PCK2	0	R/W	Peripheral Module Clock (P ϕ) Select
5	PCK1	1	R/W	These bits select the frequency of the peripheral module clock. The ratio to the input clock is as follows.
4	PCK0	0	R/W	
				PCK (2:0)
				000: × 4
				001: × 2
				010: × 1
				011: × 1/2
				1xx: Setting prohibited
				The frequency of the peripheral module clock should be lower than that of the system clock. Though the ratio can be set so as to make the frequency of the peripheral module clock higher than that of the system clock, the clocks will have the same frequency if the peripheral module clock frequency is lower than that of the system clock.

001: × 2

010: × 1

011: × 1/2

1xx: Setting prohibited

The frequency of the external bus clock should be set so as to make the frequency of the external bus clock higher than that of the system clock, the external bus clock will have the same frequency in reality.

[Legend]

x: Don't care

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

26.1.2 Subclock Control Register (SUBCKCR)*

SUBCKCR stops the main clock oscillator, selects the operating clock of the system clock, and 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 stopped during subclock operation.

1	WAKE32K	0	R/W	Wakeup Clock Select Selects the operating clock for use as the system clock after the transition from the subclock operation to software standby mode has been initiated by a hardware interrupt. 0: On leaving software standby mode, the main clock is the operating clock. 1: On leaving software standby mode, the subclock is the operating clock. This setting is valid when the WAKE32K bit (CK32K) is set to 1.
---	---------	---	-----	---

0	CK32K	0	R/W	Subclock Select 0: The system clock ($I\phi$), peripheral module clock ($P\phi$) and external bus clock ($B\phi$) operate on the subclock. 1: The system clock ($I\phi$), peripheral module clock ($P\phi$) and external bus clock ($B\phi$) operate on the system clock. When the OSC32STP bit in TCR32K is 1, 1 can be written to this bit. This bit is cleared to 0 when the device enters software standby mode while the value of WAKE32K is 0. Dummy read of this bit must be performed twice immediately after this bit is written to.
---	-------	---	-----	---

Note: Supported only by the H8SX/1648G Group and H8SX/1648H Group.

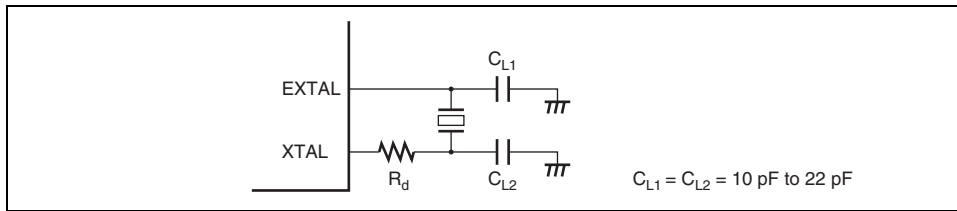


Figure 26.2 Connection of Crystal Resonator (Example)

Table 26.2 Damping Resistance Value

Frequency (MHz)	8	12	16	18
R_d (Ω)	200	0	0	0

Figure 26.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator with the characteristics shown in table 26.3.

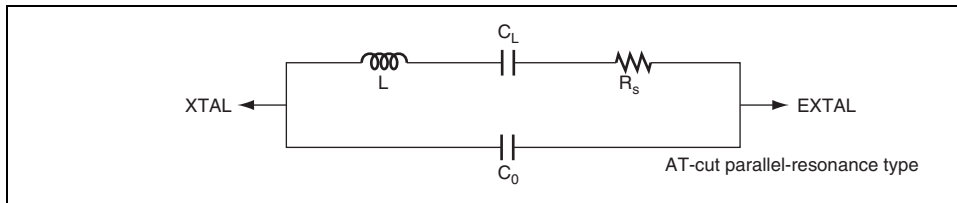


Figure 26.3 Crystal Resonator Equivalent Circuit

pin, put the external clock in high level during standby mode.

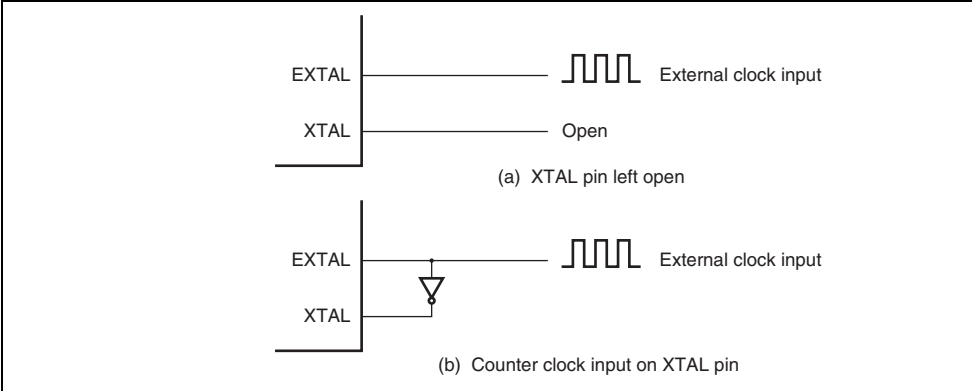


Figure 26.4 External Clock Input (Examples)

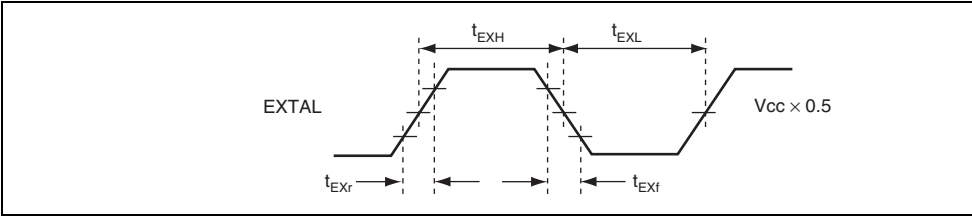


Figure 26.5 External Clock Input Timing

updated frequency.

26.5 Subclock Oscillator

Note: Supported only by the H8SX/1648G Group and H8SX/1648H Group.

26.5.1 Connecting 32.768 kHz Resonator

To supply a clock to the subclock oscillator, connect a 32.768-kHz crystal resonator, as shown in figure 26.6. The usage notes given in section 26.6.3, Notes on Board Design, apply to the connection of this crystal resonator.

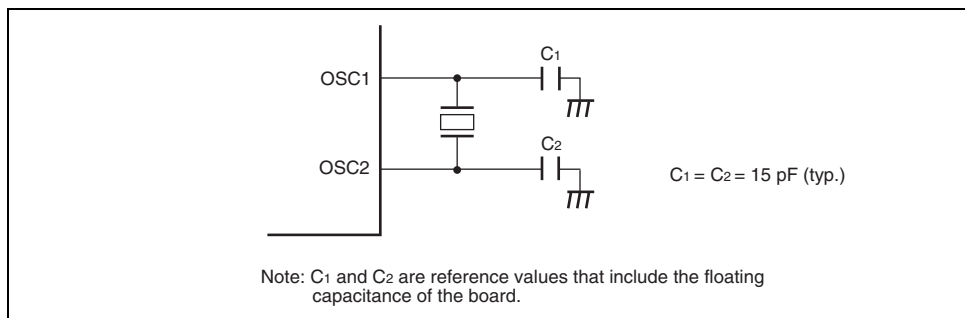


Figure 26.6 Connection Example of 32.768-kHz Crystal Resonator

Figure 26.7 shows an equivalent circuit for the 32.768-kHz crystal resonator.

26.5.2 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 26.8.

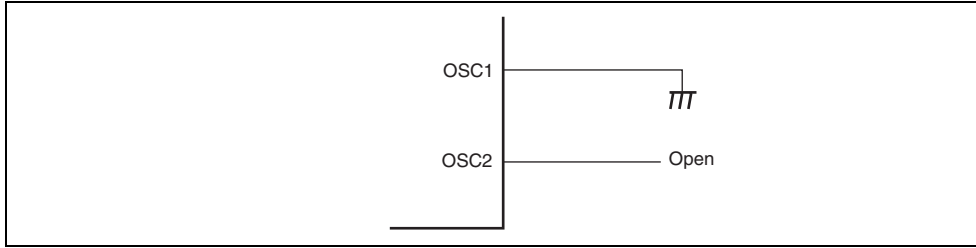


Figure 26.8 Pin Handling when Subclock is not Used

MHz, and $8 \text{ MHz} \leq B\phi \leq 50 \text{ MHz}$.

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 clock division ratio. For details, see section 27.7.3, Setting Oscillation Settling Time after Software Standby Mode.

3. The relationship among the system clock, peripheral module clock, and external bus clock is $S\phi \geq P\phi$ and $I\phi \geq B\phi$. In addition, the system clock setting has the highest priority. Accordingly, $P\phi$ or $B\phi$ may have the frequency set by bits ICK2 to ICK0 regardless of the settings of 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 setting SCKCR or SUBCKCR* while executing the external bus cycle with the write-data-buffer function.
5. Figure 26.9 shows the clock modification timing. After a value is written to SCKCR, the system clock waits for the current bus cycle to complete. After the current bus cycle completes, the system clock frequency will be modified within one cycle (worst case) of the external input clock.

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

26.6.2 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 example shown in this section as a reference. As the parameters for the resonator will depend on the 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.

26.6.3 Notes on Board Design

When using the crystal resonator, place the crystal resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 26.7 to prevent induction from interfering with correct oscillation.

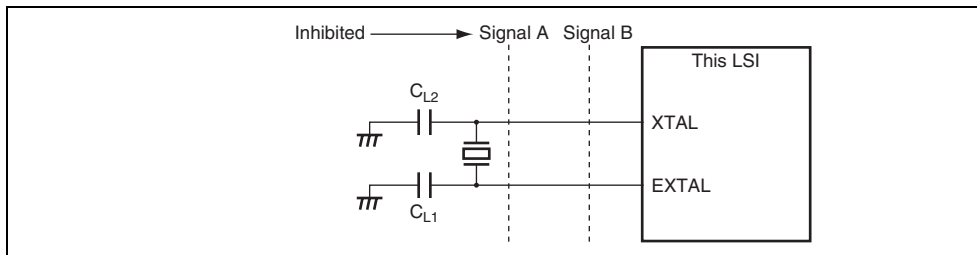


Figure 26.10 Note on Board Design for Oscillation Circuit

Note: * CB and CPB are laminated ceramic capacitors.

Figure 26.11 Recommended External Circuitry for PLL Circuit

All of the system clock, peripheral module clock, and external bus clock can be switched to 32.768kHz.*

- Module stop function
The functions for each peripheral module can be stopped to make a transition to a power-down mode.
- Transition function to power-down mode
Transition to a power-down mode is possible to stop the CPU, peripheral modules, and oscillator.
- Five power-down modes
 - Sleep mode
 - All-module-clock-stop mode
 - Software standby mode
 - Deep software standby mode
 - Hardware standby mode

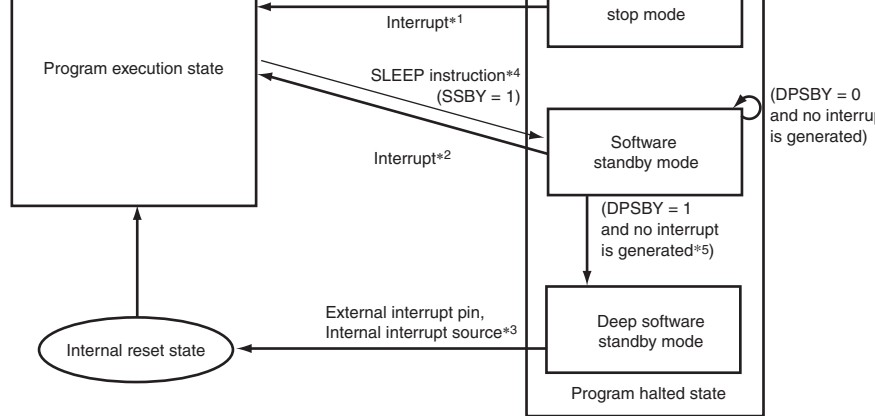
Table 27.1 shows conditions to shift to a power-down mode, states of the CPU and peripheral modules, and clearing method for each mode. After the reset state, since this LSI operates in a normal program execution state, the modules, other than the DMAC and DTC, are stopped.

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Subclock oscillator* ¹⁰	Operating* ⁹	Operating* ⁹	Operating* ⁹	Operating* ⁹	Halted
CPU	Halted (retained)	Halted (retained)	Halted (retained)	Halted (undefined)	Halted (undefined)
On-chip RAMs 6 to 4 (H'FEE000 to H'FF3FFF)	Operating (retained)	Halted (retained)	Halted (retained)	Halted (undefined)	Halted (undefined)
On-chip RAMs 3 to 0 (H'FF4000 to H'FFBFFF)	Operating (retained)	Halted (retained)	Halted (retained)	Halted (retained/undefined)* ⁵	Halted (undefined)
Watchdog timer	Operating	Operating	Halted (retained)	Halted (undefined)	Halted (undefined)
8-bit timer (unit 0/1)	Operating	Operating* ⁴	Halted (retained)	Halted (undefined)	Halted (undefined)
32K timer* ¹⁰	Operating	Operating	Operating	Operating	Halted (undefined)
Voltage detection circuit* ¹¹	Operating	Operating	Operating	Operating	Halted (undefined)
Power-on reset circuit* ¹¹	Operating	Operating	Operating	Operating	Halted (undefined)
Other peripheral modules	Operating	Halted* ¹	Halted* ¹	Halted* ⁷ (undefined)	Halted (undefined)
I/O ports	Operating	Retained	Retained* ⁶	Halted* ⁶ (undefined)	Hi-Z (undefined)

Notes: "Halted (retained)" in the table means that the internal values are retained and operations are suspended.

7. Some peripheral modules either do not have the register fields or do not support the following features:
 8. External interrupt, 32K timer interrupt*¹⁰, voltage monitoring interrupt*¹¹.
 9. Operating or halted can be selected by the OSC32STP bit in TCR32K.
 10. Supported only by the H8SX/1648G Group and H8SX/1648H Group.
 11. Supported only by the H8SX/1648L Group and H8SX/1648H Group.



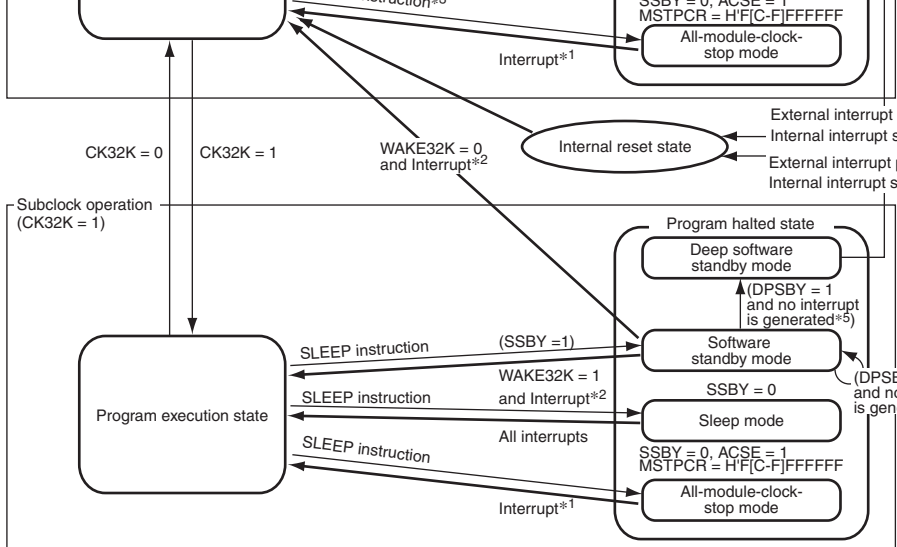
[Legend] \longrightarrow Transition after exception handling

- Notes:
1. NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$, 8-bit timer interrupts, watchdog timer interrupts, and Voltage monitoring interrupt*6. Note that the 8-bit timer interrupt is valid when the MSTPCRA9 or MSTPCRA8 bit is cleared to 0.
 2. NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$, and Voltage monitoring interrupt*6. Note that IRQ is valid only when the corresponding bit in SSIER is set to 1.
 3. Voltage monitoring interrupt*6 is generated only for NMI, and $\overline{\text{IRQ0-A}}$ to $\overline{\text{IRQ3-A}}$. Note that IRQ, and Voltage monitoring interrupt are valid only when the corresponding bit in DPSIER is set to 1.
 4. The SLPPIE bit in SBYCR is cleared to 0.
 5. If a conflict between a transition to deep software standby mode and generation of software standby mode clearing source occurs, a mode transition may be made from software standby mode to program execution state through execution of interrupt exception handling. In this case, a transition to deep software standby mode is not made. For details, refer to section 27.12, Usage Notes.
 6. Supported only by the H8SX/1648L Group.

From any state, a transition to hardware standby mode occurs when $\overline{\text{STBY}}$ is driven low.

From any state except hardware standby mode, a transition to the reset state occurs when $\overline{\text{RES}}$ is driven low.

Figure 27.1 (1) Mode Transitions (H8SX/1648 Group, H8SX/1648A Group, and H8SX/1648L Group)



[Legend] → Transition after exception handling

- Notes:
1. NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$, 8-bit timer interrupts, watchdog timer interrupts, 32K timer interrupts, and Voltage monitoring interrupt*6. Note that the 8-bit timer interrupt is valid when the MSTPCRA9 or MSTPCRA8 bit is cleared to 0.
 2. NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$, 32K timer interrupts, and Voltage monitoring interrupt*6. Note that IRQ is valid only when the corresponding bit in SSIER is set to 1.
 3. The SLPIE bit is cleared to 0.
 4. 32K timer interrupt, and Voltage monitoring interrupt*6 are generated only for NMI, and $\overline{\text{IRQ0-A}}$ to $\overline{\text{IRQ3-A}}$. Note that IRQ, 32K timer interrupt, and Voltage monitoring interrupt*6 are valid only when the corresponding bit in DPSIER is set to 1.
 5. If a conflict between a transition to deep software standby mode and generation of software standby mode clearing source occurs, a mode transition may be made from software standby mode to program execution state through execution of interrupt exception handling. In this case, a transition to deep software standby mode is not made. For details, refer to section 27.12, Usage Notes.
 6. Supported only by the H8SX/1648H Group.

From any state, a transition to hardware standby mode occurs when $\overline{\text{STBY}}$ is driven low.

From any state except hardware standby mode, a transition to the reset state occurs when $\overline{\text{RES}}$ is driven low.

Figure 27.1 (2) Mode Transitions (H8SX/1648G Group and H8SX/1648H Group)

- Deep standby wait control register (DPSWCR)
- Deep standby interrupt enable register (DPSIER)
- Deep standby interrupt flag register (DPSIFR)
- Deep standby interrupt edge register (DPSIEGR)
- Reset status register (RSTSR)
- Deep standby backup register (DPSBKRn) (n=15 to 0)

27.2.1 Standby Control Register (SBYCR)

SBYCR controls software standby mode.

Bit	15	14	13	12	11	10	9	
Bit name	SSBY	OPE	—	STS4	STS3	STS2	STS1	
Initial value:	0	1	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	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	

to used in watchdog timer mode, the output of the timer is disabled. In this case, a transition is always made to sleep mode or all-module-clock-stop mode after the SLEEP instruction is executed. When the SLPI bit is set 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 and bus control signals ($\overline{CS0}$ to $\overline{CS7}$, \overline{AS} , \overline{RD} , \overline{HWR} , and \overline{HWR}) are retained or these lines are set to the high-Z state in software standby mode or deep software standby mode. 0: In software standby mode or deep software standby mode, address bus and bus control signal lines are set to high-impedance. 1: In software standby mode or deep software standby mode, output states of address bus and bus control signals are retained.
13	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.

While oscillation is being settled, the timer is counted at the P ϕ clock frequency. Careful consideration is required 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

executed, this bit remains set to 1. For clearing this bit.

6 to 0 — All 0 R/W Reserved

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

- Notes: 1. With the F-ZTAT version, the flash memory settling time must be reserved.
 2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.

27.2.2 Module Stop Control Registers A and B (MSTPCRA and MSTPCRB)

MSTPCRA and MSTPCRB control module stop state. Setting a bit to 1 makes the core module enter module stop state, while clearing the bit to 0 clears module stop state.

- MSTPCRA

Bit	15	14	13	12	11	10	9
Bit name	ACSE	MSTPA14	MSTPA13	MSTPA12	MSTPA11	MSTPA10	MSTPA9
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
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 state for current consumption by stopping the bus control I/O ports operations when the CPU executes the instruction after module stop state has been set on-chip peripheral modules controlled by MSTPCRA. 0: All-module-clock-stop mode disabled 1: All-module-clock-stop mode enabled
14	MSTPA14	0	R/W	EXDMA controller (EXDMAC)*
13	MSTPA13	0	R/W	DMA controller (DMAC)
12	MSTPA12	0	R/W	Data transfer controller (DTC)
11	MSTPA11	1	R/W	Reserved
10	MSTPA10	1	R/W	These bits are always read as 1. The write value 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	MSTPA7	1	R/W	Reserved
6	MSTPA6	1	R/W	These bits are always read as 1. The write value always be 1.

1	MSTPA1	1	R/W	16-bit timer pulse unit (TPU channels 11 to 6)
0	MSTPA0	1	R/W	16-bit timer pulse unit (TPU channels 5 to 0)

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

- MSTPCRB

Bit	Bit Name	Initial Value	R/W	Module
15	MSTPB15	1	R/W	Programmable pulse generator (PPG_0: PO15)
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 communications interface_4 (SCI_4)
11	MSTPB11	1	R/W	Serial communications interface_3 (SCI_3)
10	MSTPB10	1	R/W	Serial communications interface_2 (SCI_2)
9	MSTPB9	1	R/W	Serial communications interface_1 (SCI_1)
8	MSTPB8	1	R/W	Serial communications interface_0 (SCI_0)
7	MSTPB7	1	R/W	I ² C bus interface 2_1 (IIC2_1)
6	MSTPB6	1	R/W	I ² C bus interface 2_0 (IIC2_0)
5	MSTPB5	1	R/W	User break controller (UBC)
4	MSTPB4	1	R/W	Reserved
3	MSTPB3	1	R/W	These bits are always read as 1. The write value always be 1.
2	MSTPB2	1	R/W	
1	MSTPB1	1	R/W	
0	MSTPB0	1	R/W	

Bit	15	14	13	12	11	10	9	M
Bit name	MSTPC15	MSTPC14	MSTPC13	MSTPC12	MSTPC11	MSTPC10	MSTPC9	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	
Bit	7	6	5	4	3	2	1	
Bit name	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	M
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 communications interface_5 (SCI_5), (IrDA)
14	MSTPC14	1	R/W	Serial communications 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	I ² C bus interfaces 2_3 and 2_2 (IIC2_3/ IIC2_2)
10	MSTPC10	1	R/W	Cyclic redundancy check calculator
9	MSTPC9	1	R/W	A/D converter (unit 2/1)
8	MSTPC8	1	R/W	Programmable pulse generator (PPG_1: PO31 to PO32)

On-chip RAM 4 (H'FF2000 to H'FF3FFF)

Always set the MSTPC5 and MSTPC4 bits to the sa

3	MSTPC3	0	R/W	On-chip RAM_3, 2 (H'FF4000 to H'FF7FFF)
2	MSTPC2	0	R/W	Always set the MSTPC3 and MSTPC2 bits to the sa
1	MSTPC1	0	R/W	On-chip RAM_1, 0 (H'FF8000 to H'FFBFFF)
0	MSTPC0	0	R/W	Always set the MSTPC1 and MSTPC0 bits to the s value.

27.2.4 Deep Standby Control Register (DPSBYCR)

DPSBYCR controls deep software standby mode.

DPSBYCR is initialized by input of the reset signal on the $\overline{\text{RES}}$ pin, but is not initialized by internal reset signal upon exit from deep software standby mode.

Bit	7	6	5	4	3	2	1	0
Bit name	DPSBY	IOKEEP	RAMCUT2	RAMCUT1	—	—	—	F
Initial value:	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

1	0	Enters software standby mode after execution of a SLEEP instruction.
1	1	Enters deep software standby mode after execution of a SLEEP instruction.

When deep software standby mode is canceled by an interrupt, this bit remains at 1. Write a 0 here to clear this bit. Setting of this bit has no effect when the WDT is in watchdog timer mode. In this case, executing the SLEEP instruction always initiates entry to sleep mode or module-clock-stop mode. Be sure to clear this bit when setting the SLPIE bit to 1.

simultaneously with exit from deep software standby mode.

1 The retained port states are released when a 0 is written to this bit following exit from deep software standby mode.

In operation in external extended mode, however, the address bus, bus control signals ($\overline{CS0}$, \overline{AS} , \overline{RD} , and \overline{LWR}), and data bus are set to the initial state upon exit from deep software standby mode.

5	RAMCUT2	0	R/W	On-chip RAM Power Off 2 RAMCUT 2, 1, and 0 control the internal power supply to the on-chip RAM in deep software standby mode. For details, see descriptions of the RAMCUT0 bit.
4	RAMCUT1	0	R/W	On-chip RAM Power Off 1 RAMCUT 2, 1, and 0 control the internal power supply to the on-chip RAM in deep software standby mode. For details, see descriptions of the RAMCUT0 bit.
3 to 1	—	All 0	R/W	Reserved These bits are always read as 0. The write value must always be 0.
0	RAMCUT0	1	R/W	On-chip RAM Power Off 0 RAMCUT 0 controls the internal power supply to the on-chip RAM in deep software standby mode, in combination with RAMCUT 2 and RAMCUT 1. RAMCUT 2 to 0 000: Power is supplied to the on-chip RAM. 111: Power is not supplied to the on-chip RAM. Settings other than those given above are prohibited.

Bit	Bit Name	Initial Value	R/W	Module
7, 6	—	All 0	R/W	Reserved

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

During the oscillation settling period, counting is performed with the clock frequency input to the

000000: Reserved

000001: Reserved

000010: Reserved

000011: Reserved

000100: Reserved

000101: Wait time = 64 states

000110: Wait time = 512 states

000111: Wait time = 1024 states

001000: Wait time = 2048 states

001001: Wait time = 4096 states

001010: Wait time = 16384 states

001011: Wait time = 32768 states

001100: Wait time = 65536 states

001101: Wait time = 131072 states

001110: Wait time = 262144 states

001111: Wait time = 524288 states

01xxxx: Reserved

Notes: 1. Supported only by the H8SX/1648G group, and H8SX/1648H group.
 2. Supported only by the H8SX/1648L Group, and H8SX/1648H group.

Bit	Bit Name	Initial Value	R/W	Module
7	—	0	R/W	Reserved This bit is always read as 0. The write value should be 0.
6	—	0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
5	DT32KIE* ¹	0	R/W	32K Timer Interrupt Enable Enables/disables exit from deep software standby mode by the 32K timer interrupt signal. 0: Disables exit from deep software standby mode by the 32K timer interrupt signal. 1: Enables exit from deep software standby mode by the 32K timer interrupt signal.
4	DLVDIE* ²	0	R/W	LVD Interrupt Enable Enables/disables exit from deep software standby mode by the voltage monitoring interrupt signal. 0: Disables exit from deep software standby mode by the voltage monitoring interrupt signal. 1: Enables exit from deep software standby mode by the voltage monitoring interrupt signal.

Enables or disables exit from deep software standby mode by IRQ2-A.

0: Disables exit from deep software standby mode A.

1: Enables exit from deep software standby mode A.

1	DIRQ1E	0	R/W	IRQ1 Interrupt Enable
				Enables or disables exit from deep software standby by IRQ1.
				0: Disables exit from deep software standby mode A.
				1: Enables exit from deep software standby mode A.

0	DIRQ0E	0	R/W	IRQ0 Interrupt Enable
				Enables or disables exit from deep software standby by IRQ0-A.
				0: Disables exit from deep software standby mode A.
				1: Enables exit from deep software standby mode A.

- Notes:
1. Supported only by the H8SX/1648G Group and H8SX/1648H Group.
 2. Supported only by the H8SX/1648L Group and H8SX/1648H Group.

Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)* ¹	R	R/(W)* ¹	R/(W)* ¹	R/(W)* ¹	R/(W)* ¹	R/(W)* ¹

- Notes:
1. Only 0 can be written to clear the flag.
 2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.
 3. Supported only by the H8SX/1648L Group and H8SX/1648H Group.

Bit	Bit Name	Initial Value	R/W	Module
7	DNMIF	0	R/(W)* ¹	NMI Flag [Setting condition] NMI input specified in DPSIEGR is generated. [Clearing condition] Writing a 0 to this bit after reading it as 1.
6	—	0	R/(W)* ¹	Reserved These bits are always read as 0. The write value always be 0.
5	DT32KIF* ²	0	R/(W)* ¹	32K Timer Interrupt Flag [Setting condition] 32K timer interrupt is generated. [Clearing condition] Writing a 0 to this bit after reading it as 1.
4	DLVDIF* ³	0	R/(W)* ¹	LVD Interrupt Flag [Setting condition] Voltage monitoring interrupt is generated. [Clearing condition] Writing a 0 to this bit after reading it as 1.

				[Clearing condition] Writing a 0 to this bit after reading it as 1.
1	DIRQ1F	0	R/(W)* ¹	IRQ1 Interrupt Flag [Setting condition] IRQ1-A input specified in DPSIEGR is generated. [Clearing condition] Writing a 0 to this bit after reading it as 1.
0	DIRQ0F	0	R/(W)* ¹	IRQ0 Interrupt Flag [Setting condition] IRQ0-A input specified in DPSIEGR is generated. [Clearing condition] Writing a 0 to this bit after reading it as 1.

- Notes:
1. Only 0 can be written to clear the flag.
 2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.
 3. Supported only by the H8SX/1648L Group and H8SX/1648H Group.

Bit	Bit Name	Initial Value	R/W	Module
7	DNMIEG	0	R/W	NMI Edge Select Selects the active edge for NMI pin input. 0: The interrupt request is generated by a falling 1: The interrupt request is generated by a rising
6 to 4	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.
3	DIRQ3EG	0	R/W	IRQ3 Interrupt Edge Select Selects the active edge for IRQ3-A pin input. 0: The interrupt request is generated by a falling 1: The interrupt request is generated by a rising
2	DIRQ2EG	0	R/W	IRQ2 Interrupt Edge Select Selects the active edge for IRQ2-A pin input. 0: The interrupt request is generated by a falling 1: The interrupt request is generated by a rising

27.2.9 Reset Status Register (RSTSR)

The DPSRSTF bit in RSTSR indicates that deep software standby mode has been canceled by an interrupt.

RSTSR is not initialized by the internal reset signal upon exit from deep software standby mode.

Bit	7	6	5	4	3	2	1
Bit name	DPSRSTF	—	—	—	—	LVDF*2	—
Initial value:	0	0	0	0	0	0*3	0*3
R/W:	R/(W)*1	R/W	R/W	R/W	R/W	R/(W)*4	R/W

- Notes:
1. Only 0 can be written to clear the flag.
 2. Supported only by the H8SX/1648L Group and H8SX/1648H Group.
 3. Initial value is undefined in the H8SX/1648L Group and H8SX/1648H Group.
 4. Only 0 can be written to clear the flag in the H8SX/1648L Group and H8SX/1648H Group.
 5. Readable only in the H8SX/1648L Group and H8SX/1648H Group.

6 to 3	—	All 0	R/W	Reserved	These bits are always read as 0. The write value always be 0.
--------	---	-------	-----	----------	---

- H8SX1648 Group, H8SX/1648A Group, and H8SX1648G Group

2 to 0	—	All 0	R/W	Reserved	These bits are always read as 0. The write value always be 0.
--------	---	-------	-----	----------	---

- H8SX1648L Group, H8SX1648H Group

2	LVDF	Undefined	R/(W)*	LVD Flag	This bit indicates that the voltage-detection circuit detected a low voltage (V_{cc} at or below V_{det}). For details, see section 5, Voltage Detection Circuit (LVD).
---	------	-----------	--------	----------	--

1	—	Undefined	R/W	Reserved	These bits are always read as 0. The write value always be 0.
---	---	-----------	-----	----------	---

0	PORF	Undefined	R	Power-on Reset Flag	This bit indicates that a power-on reset has been generated. For details, see section 4, Reset.
---	------	-----------	---	---------------------	--

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

	7	6	5	4	3	2	1
Bit name	BKUPn7	BKUPn6	BKUPn5	BKUPn4	BKUPn3	BKUPn2	BKUPn1
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

n: 15 to 0

Even if the frequencies specified by bits PCK2 to PCK0 and BCK2 to BCK0 are higher than the frequencies of the main clock, the specified values are not reflected in the peripheral module and external bus clocks. The peripheral module and external bus clocks are restricted to the operating clock specified by bits ICK2 to ICK0.

27.3.2 Switching to Subclock

When the CK32K bit in SUBCKCR is set to 1, a transition from the main clock operation to subclock operation is made at the end of the bus cycle regardless of the SCKCR setting. In subclock operation, the CPU, bus masters, peripheral modules, and all external buses operate at the 32.768-kHz subclock.

When the CK32K bit in SUBCKCR is set to 0 in the subclock operation, a transition to the main clock operation is made at the end of the bus cycle. Since a transition from the subclock operation to the main clock operation is made via software standby mode, the oscillation settling time of the main clock must elapse. Set the oscillation settling time of the main clock with bits STS4 to STS0 in SBYCR.

The main clock oscillator can be operated or stopped by the EXSTP bit in SUBCKCR in the subclock operation. When a transition is made from the subclock operation to the main clock operation with the main clock oscillator operating, the wait for the oscillation settling time of the main clock oscillator is not necessary. A transition to the main clock operation can be made at the 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 subclock operation while the SSBY bit in SBYCR is set to 1, this LSI enters software standby mode. When a transition is made to software standby mode in the subclock operation, the operating clock after clearing of software standby mode can be selected with the WAKE32K bit in SUBCKCR. This LSI is placed in the subclock operation if the WAKE32K bit is 1, or placed in the main clock operation if the WAKE32K bit is 0.

After the reset state is cleared, all modules other than the DMAC, DTC, EXDMAC*, and RAM are placed in a module stop state.

The registers of the module for which the module stop state is selected cannot be read from or written to.

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

27.5 Sleep Mode

27.5.1 Entry to Sleep Mode

When the SLEEP instruction is executed when the SSBY bit in SBYCR is 0, the CPU enters sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registers are retained. Other peripheral functions do not stop.

27.5.2 Exit from Sleep Mode

Sleep mode is exited by any interrupt, signals on the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin, and a reset caused by a watchdog timer overflow, a voltage monitoring reset*, or a power-on reset*.

- Exit from sleep mode by interrupt
When an interrupt occurs, sleep mode is exited and interrupt exception processing starts. Sleep mode is not exited if the interrupt is disabled, or interrupts other than NMI are masked. The CPU resumes operation.
- Exit from sleep mode by $\overline{\text{RES}}$ pin
Setting the $\overline{\text{RES}}$ pin level low selects the reset state. After the stipulated reset input duration, driving the $\overline{\text{RES}}$ pin high makes the CPU start the reset exception processing.

27.6 All-Module-Clock-Stop Mode

When the ACSE bit is set to 1 and all modules controlled by MSTPCRA and MSTPCRB stopped (MSTPCRA, MSTPCRB = H'FFFFFFF), or all modules except for the 8-bit timer 0 and 1) are stopped (MSTPCRA, MSTPCRB = H'F[C to F]FFFFFF), executing a SLEEP instruction with the SSBY bit in SBYCR cleared to 0 will cause all modules (except for the timer*¹, watchdog timer, 32K timer*², power-on reset circuit, voltage detection circuit*³), controller, and the I/O ports to stop operating, and to make a transition to all-module-clock mode at the end of the bus cycle.

When power consumption should be reduced ever more in all-module-clock-stop mode, modules controlled by MSTPCRC (MSTPCRC[15:8] = H'FFFF).

All-module-clock-stop mode is cleared by an external interrupt (NMI or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$), $\overline{\text{RES}}$ pin input, or an internal interrupt (8-bit timer*¹, watchdog timer, 32K timer*², voltage detection circuit*³), and the CPU returns to the normal program execution state via the exception handling state. All-module-clock-stop mode is not cleared if interrupts are disabled, if interrupts other than NMI are masked on the CPU side, or if the relevant interrupt is designated as a non-activation source.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

- Notes:
1. Operation or halting of the 8-bit timer can be selected by bits MSTPA9 and MSTPA10 in MSTPCRA.
 2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.
 3. Supported only by the H8SX/1648L Group and H8SX/1648H Group.

mode the oscillator stops, allowing power consumption to be significantly reduced.

If the WDT is used in watchdog timer mode, it is impossible to make a transition to software standby mode. The WDT should be stopped before the SLEEP instruction execution.

27.7.2 Exit from Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI, or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}^{*1}$), a timer interrupt (32Ktimer^{*2}), a voltage monitoring interrupt^{*3}, a power-on reset^{*3} or by means of the RES pin or STBY pin.

1. Exit from software standby mode by interrupt

When an NMI, IRQ0 to IRQ15^{*1}, 32K timer^{*2} or voltage-monitoring^{*3} interrupt request is input, clock oscillation starts, and after the elapse of the time set in bits STS4 to STS7 of the SBYCR, stable clocks are supplied to the entire LSI, software standby mode is cleared, and interrupt exception handling is started.

When clearing software standby mode with an IRQ0 to IRQ15^{*1} interrupt, set the corresponding enable bit to 1 and ensure that no interrupt with a higher priority than IRQ0 to IRQ15^{*1} is generated. Software standby mode cannot be cleared if the interrupt has been masked on the CPU side or has been designated as a DTC activation source.

2. Exit from voltage monitoring reset^{*3}

When a voltage monitoring reset is generated by the fall of power-voltage, software standby mode is cleared and a clock oscillation starts. At the same time, a clock signal is supplied throughout the LSI. After that, if power voltage rises, the voltage detection reset is released. Thereafter, CPU starts the reset exception handling.

5. Exit from software standby mode by $\overline{\text{STBY}}$ pin

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

- Notes:
1. By setting the SSIn bit in SSIER to 1, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$ can be used as a software standby mode clearing source.
 2. Supported only by the H8SX/1648G Group and H8SX/1648H Group.
 3. Supported only by the H8SX/1648L Group and H8SX/1648H Group.


27.7.3 Setting Oscillation Settling Time after Exit from Software Standby Mode


Bits STS4 to STS0 in SBYCR should be set as described below.

1. Using a crystal resonator
Set bits STS4 to STS0 so that the standby time is at least equal to the oscillation settling time.
Table 27.2 shows the standby times for operating frequencies and settings of bits STS4 to STS0.
2. Using an external clock
A PLL circuit settling time is necessary. Refer to table 27.2 to set the standby time.

			1	0	512	14.6	20.5	25.6	39.4	51.2	64.0
				1	1024	29.3	41.0	51.2	78.8	102.4	128.0
1	0	0	0	0	2048	58.5	81.9	102.4	157.5	204.8	256.0
				1	4096	0.12	0.16	0.20	0.32	0.41	0.51
			1	0	16384	0.47	0.66	0.82	1.26	1.64	2.05
				1	32768	0.94	1.31	1.64	2.52	3.28	4.10
	1	0	0	0	65536	1.87	2.62	3.28	5.04	6.55	8.19
				1	131072	3.74	5.24	6.55	10.08	13.11	16.38
			1	0	262144	7.49	10.49	13.11	20.16	26.21	32.77
				1	524288	14.98	20.97	26.21	40.33	52.43	65.54
1	0	0	0	0	Reserved	—	—	—	—	—	—

[Legend]

 : Recommended setting when external clock is in use

 : Recommended setting when crystal oscillator is in use

Note: * $P\phi$ is the output from the peripheral module frequency divider. The oscillation time, which includes a period where the oscillation by an oscillator is not stable, depends on the resonator characteristics. The above figures are for reference.

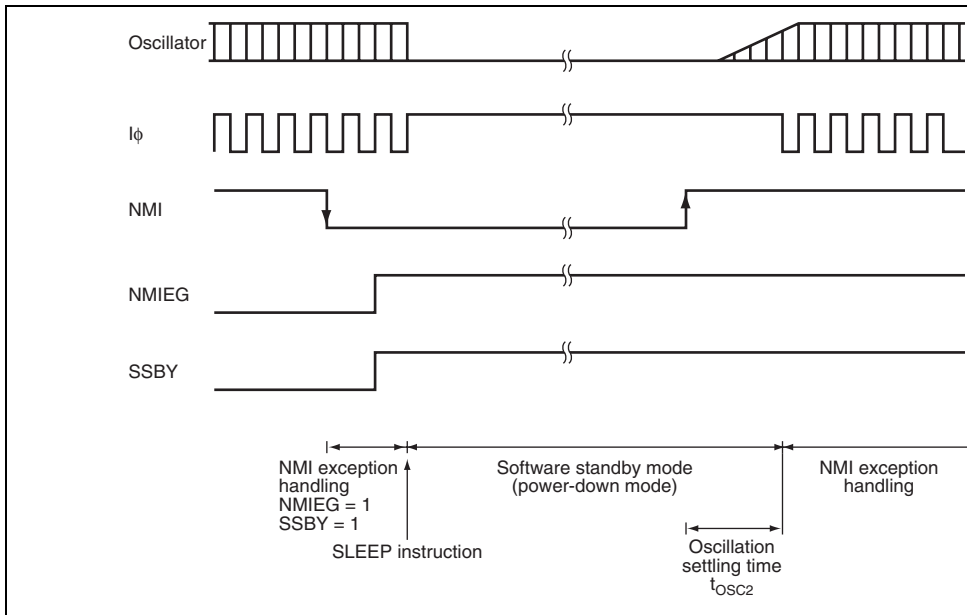


Figure 27.2 Software Standby Mode Application Example

software standby mode will be cleared regardless of the DPSBY bit setting, and the interrupt exception handling starts after the oscillation settling time for software standby mode expires. The bits STS4 to STS0 in SBYCR has elapsed.

When both of the SSBY bit in SBYCR and the DPSBY bit in DPSBYCR are set to 1 and a software standby mode-clearing source occurs, a transition to deep software standby mode is made immediately after software standby mode is entered.

In deep software standby mode, the CPU, on-chip peripheral functions (except the 32K timer), on-chip RAMs 6 to 4, and oscillator functionality are all halted. In addition, the internal power supply to these modules stops, resulting in a significant reduction in power consumption. At this time, the contents of all the registers of the CPU, on-chip peripheral functions (except the 32K timer^{*1}), and on-chip RAMs 6 to 4 become undefined.

The Contents of on-chip RAMs 3 to 0 can be retained when all the bits RAMCUT2 to RAMCUT0 in DPSBYCR have been cleared to 0. If these bits are set to all 1, the internal power supply to on-chip RAMs 3 to 0 stops and the power consumption is further reduced. At this time, the contents of the on-chip RAMs 3 to 0 become undefined.

The 32K timer^{*1}, voltage detection circuit^{*2}, and power-on reset circuit^{*2} can operate in software standby mode.

The states of I/O ports are retained in software standby mode.

- Notes: 1. Supported only by the H8SX/1648G Group and H8SX/1648H Group.
2. Supported only by the H8SX/1648L Group and H8SX/1648H Group.

can be specified with DPSIEGR. The DT32KIF bit is set to 1 when a 32KHz interrupt is generated. The DLVIF bit is set to 1 when a voltage-monitoring interrupt is generated.

When deep software standby mode clearing source is generated, internal power supply is generated simultaneously with the start of clock oscillation, and internal reset signal is generated throughout the entire LSI. Once the time specified by the WTSTS5 to WTSTS0 bits in DPSWCR has elapsed, a stable clock signal is being supplied throughout the LSI and the internal reset is cleared. Deep software standby mode is canceled on clearing of the internal reset, and then the reset-exception handling starts.

When deep software standby mode is canceled by an external interrupt pin, the DPSRSTSR bit in RSTSR is set to 1.

2. Exit from deep software standby mode by a voltage-monitoring reset*²

When a voltage monitoring reset is generated by the power-supply voltage falling, the LSI is released from deep software standby mode and clock oscillation starts. At the same time, a stable clock signal is supplied throughout the LSI. When the power-supply voltage has risen sufficiently, the LSI is released from the voltage-detection reset state. The CPU then starts reset-exception handling.

3. Exit from power-on reset*²

When a power-on reset is generated by the power-supply voltage falling, the LSI is released from deep software standby mode. If the power-supply voltage then rises sufficiently, clock oscillation starts and the LSI is released from the power-on reset state after the clock oscillation stabilization time has been secured. As soon as the clock oscillation starts, a stable clock signal is provided to the LSI. After that, the CPU starts reset-exception handling.

(2) Pins other than address bus, bus control and data bus pins

Whether the ports are initialized or retain the states that were held during software standby mode can be selected by the IOKEEP bit.

- When IOKEEP = 0
Ports are initialized by an internal reset caused by deep software standby mode.
- When IOKEEP = 1
The port states that were held in deep software standby mode are retained regardless of the internal state though the internal of the LSI is initialized by an internal reset caused by deep software standby mode. At this time, the port states that were held in software standby mode are retained even if settings of I/O ports or peripheral modules are set. Subsequently, the retained port states are released when the IOKEEP bit is cleared to 0 and operation is performed according to the internal settings.

IOKEEP bit is not initialized by an internal reset caused by clearing deep software standby mode.

1. Change the value of the PSTOP1 bit from 0 to 1 to fix the B ϕ output at the high level (that the B ϕ output was already fixed high).
2. Clear the IOKEEP bit to 0 to end retention of the B ϕ state.
3. Clear the PSTOP1 bit to 0 to enable B ϕ output.

For SDRAM ϕ , the procedure is the same as is given above, except that the PSTOP0 bit is the PSTOP1 bit.

For the port state when the IOKEEP bit is set to 1, see section 27.8.3, Pin State on Exit from Software Standby Mode.

Note: * SDRAM ϕ and control of this signal by the PSTOP0 bit are only supported by H8SX/1648G Group and H8SX/1648H Group.

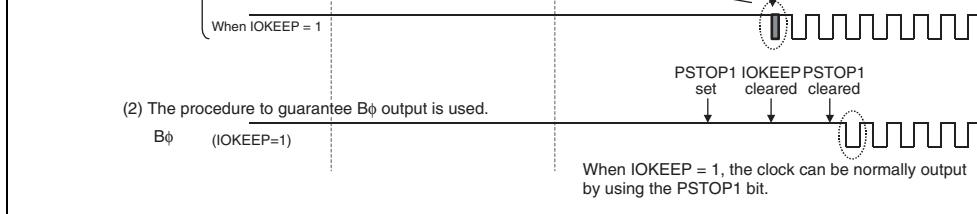


Figure 27.3 B ϕ Operation after Exit from Deep Software Standby Mode


27.8.5 Setting Oscillation Settling Time after Exit from Deep Software Standby Mode


The WTSTS5 to WTSTS0 bits in DPSWCR should be set as follows:

1. Using a crystal resonator
Specify the WTSTS5 to WTSTS0 bits so that the standby time is at least equal to the oscillation settling time. Table 27.3 shows EXTAL input clock frequencies and the settling time according to WTSTS5 to WTSTS0 settings.
2. Using an external clock
The PLL circuit settling time should be considered. See table 27.3 to set the standby time.

			1	64	3.6	4.0	4.6	5.3	6.4	8
			1	0	512	28.4	32.0	36.6	42.7	6
				1	1024	56.9	64.0	73.1	85.3	1
1	0	0	0	2048	113.8	128.0	146.3	170.7	204.8	2
			1	4096	0.23	0.26	0.29	0.34	0.41	0
			1	0	16384	0.91	1.02	1.17	1.37	2
				1	32768	1.82	2.05	2.34	2.73	4
1	0	0	0	65536	3.64	4.10	4.68	5.46	6.55	8
			1	131072	7.28	8.19	9.36	10.92	13.11	1
			1	0	262144	14.56	16.38	18.72	21.85	3
				1	524288	29.13	32.77	37.45	43.69	6
1	0	0	0	0	Reserved	—	—	—	—	—

[Legend]

 : Recommended setting when external clock is in use

 : Recommended setting when crystal oscillator is in use

Note: * The oscillation settling time, which includes a period where the oscillation by crystal oscillator is not stable, depends on the resonator characteristics.
The above figures are for reference.

transition to deep software standby mode is triggered by execution of a SLEEP instruction.

After that, deep software standby mode is canceled at the rising edge on the NMI pin.

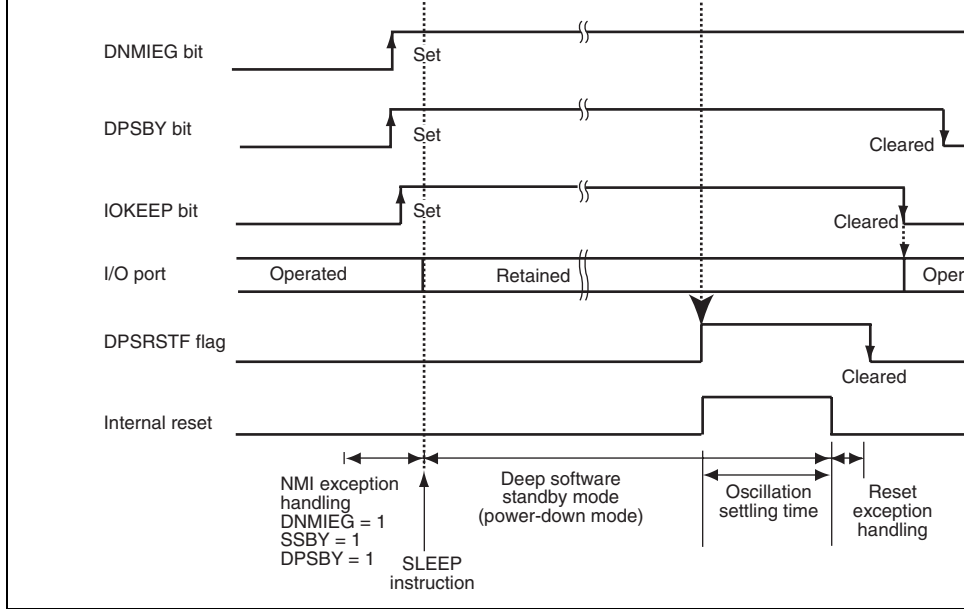


Figure 27.4 Deep Software Standby Mode Application Example (IOKEEP)

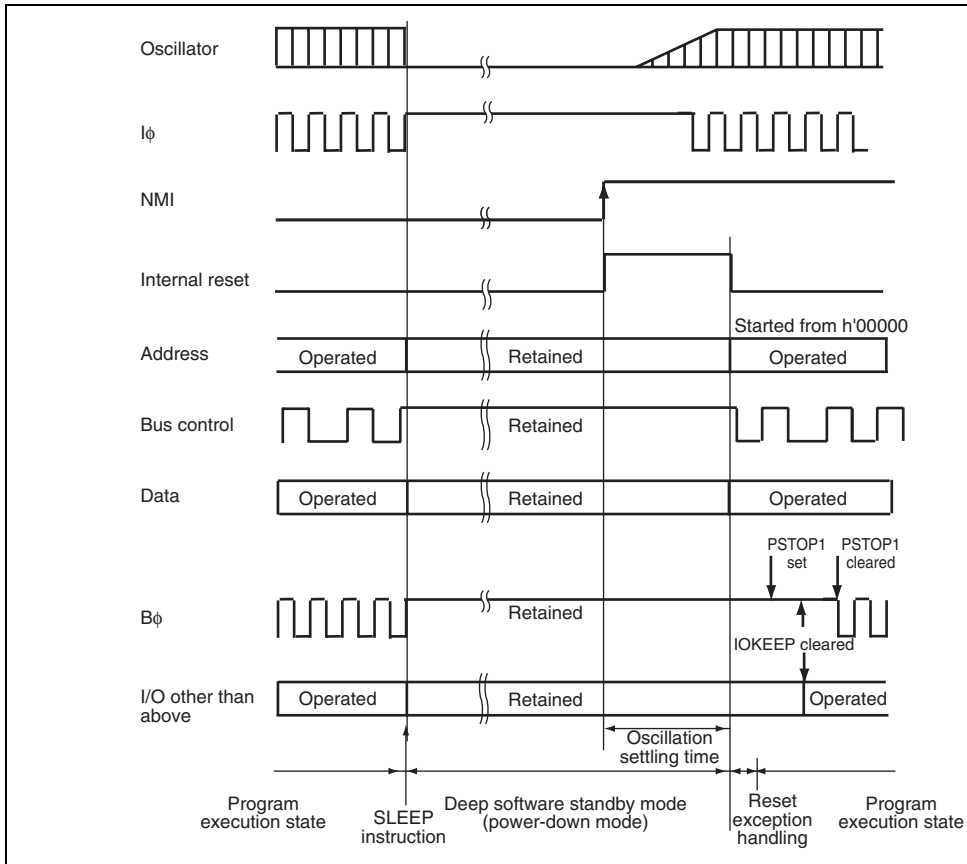


Figure 27.5 Example of Deep Software Standby Mode Operation in External Extended Mode (IOKEEP = OPE = 1)

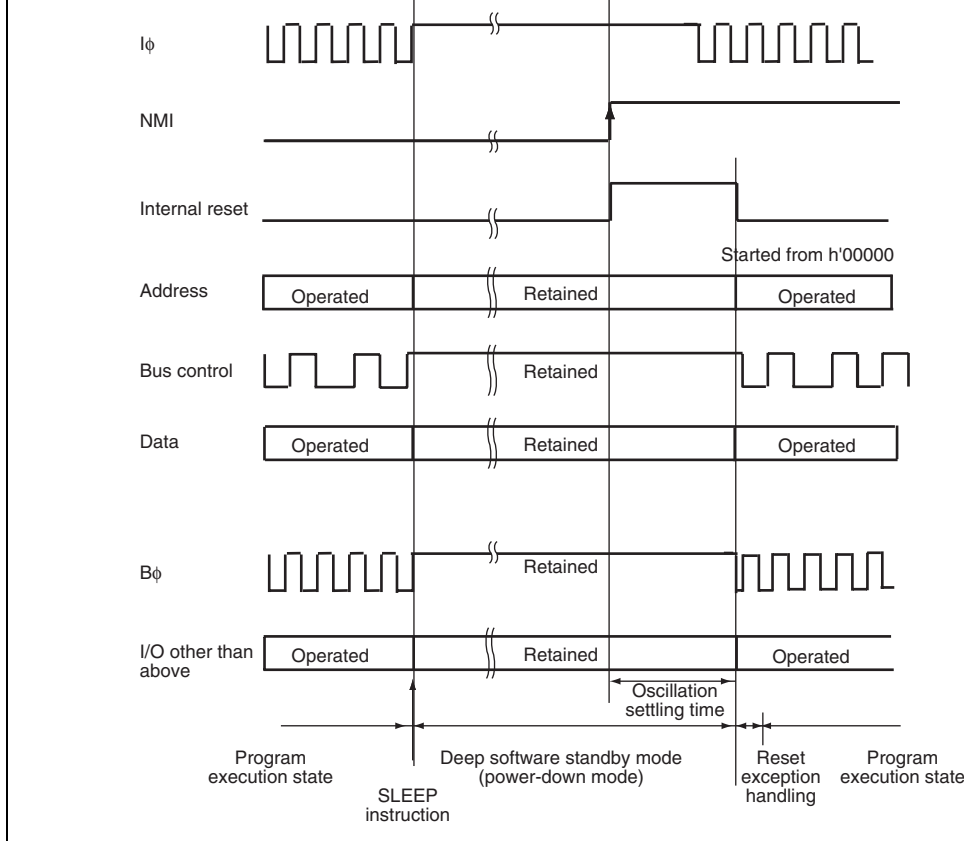


Figure 27.6 Example of Deep Software Standby Mode Operation in External Extended Mode (IOKEEP = 0, OPE = 1)

in Bφ output is also set.

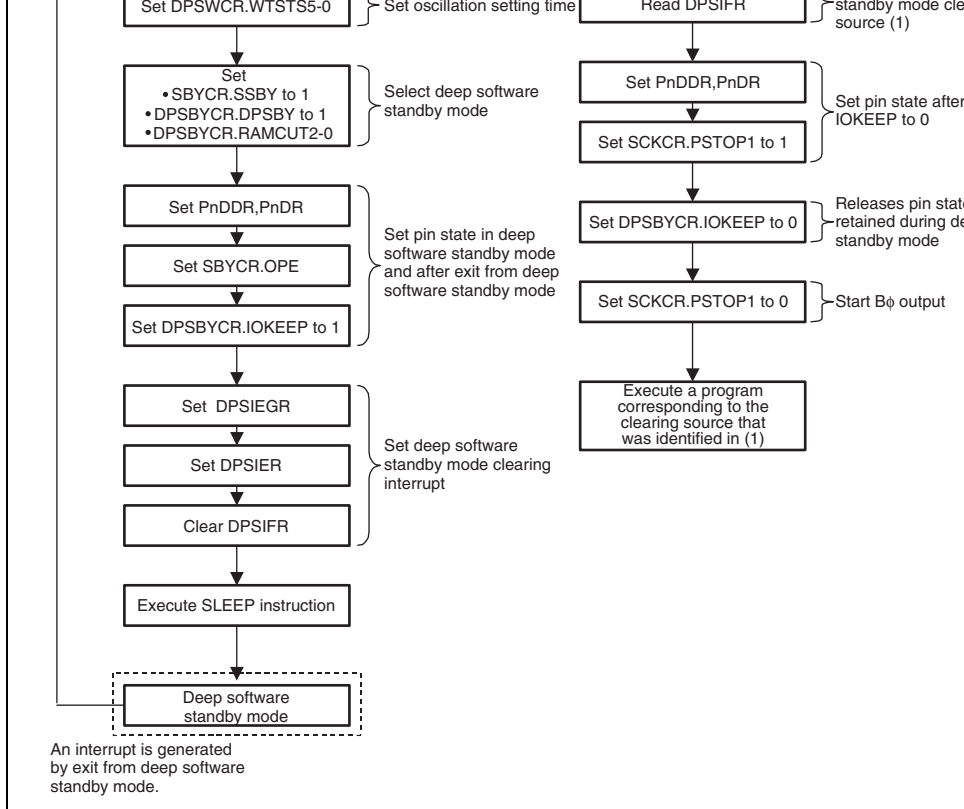


Figure 27.7 Flowchart of Deep Software Standby Mode Operation

27.9.2 Clearing Hardware Standby Mode

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is entered and clock oscillation started. Ensure that the $\overline{\text{RES}}$ pin is held low until clock oscillation settles (for details on the oscillation settling time, refer to table 27.2). When the $\overline{\text{RES}}$ pin is subsequently driven high, a transition is made to the program execution state via the reset exception handling state.

27.9.3 Hardware Standby Mode Timing

Figure 27.8 shows an example of hardware standby mode timing.

When the $\overline{\text{STBY}}$ pin is driven low after the $\overline{\text{RES}}$ pin has been driven low, a transition is made to hardware standby mode. Hardware standby mode is cleared by driving the $\overline{\text{STBY}}$ pin high, then waiting for the oscillation settling time, then changing the $\overline{\text{RES}}$ pin from low to high.

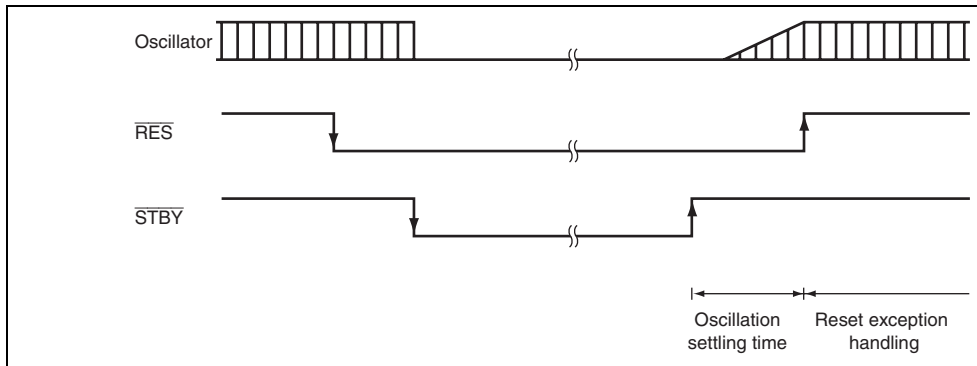


Figure 27.8 Hardware Standby Mode Timing

Timing.

In a power-on reset*, power on while driving the $\overline{\text{STBY}}$ or $\overline{\text{RES}}$ pin to a high-level.

Note: * Supported only by the H8SX/1648L Group and H8SX/1648H Group.

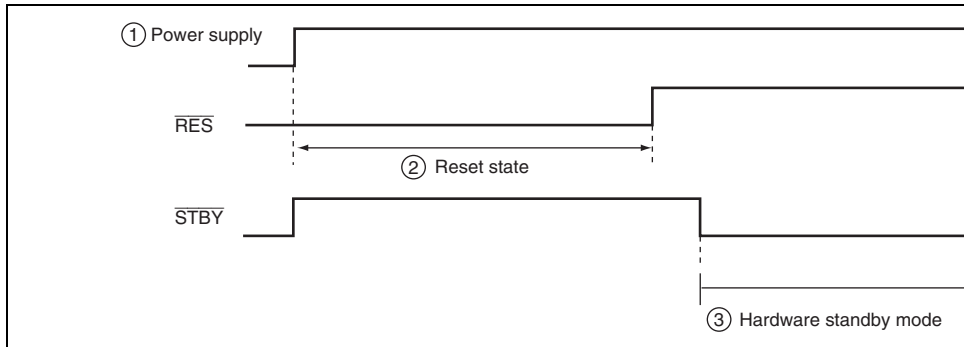


Figure 27.9 Timing Sequence at Power-On

Transitions to the power-down state are inhibited when sleep instruction exception handling is 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 to the power-down state is canceled. The power-down state is canceled by a canceling factor interrupt (see figure 27.10).

When a canceling factor interrupt is generated immediately before the execution of a SLEEP instruction, exception handling for the interrupt starts at this point. When execution returns from the exception service routine, the SLEEP instruction is executed to enter the power-down state. In this case, the power-down state is not canceled until the next canceling factor interrupt is generated (see figure 27.11).

When the SLPIE bit is set to 1 in the service routine for a canceling factor interrupt so that the execution of a SLEEP instruction will produce sleep instruction exception handling, the operation of the system is as shown in figure 27.12. Even if a canceling factor interrupt is generated immediately before the SLEEP instruction is executed, sleep instruction exception handling is initiated by execution of the SLEEP instruction. Therefore, the CPU executes the instruction following the SLEEP instruction after sleep instruction exception and exception service routine completion 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 SLEEP to 0.

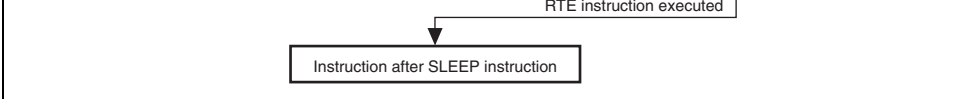


Figure 27.10 When Canceling Factor Interrupt is Generated after SLEEP Instruction Execution

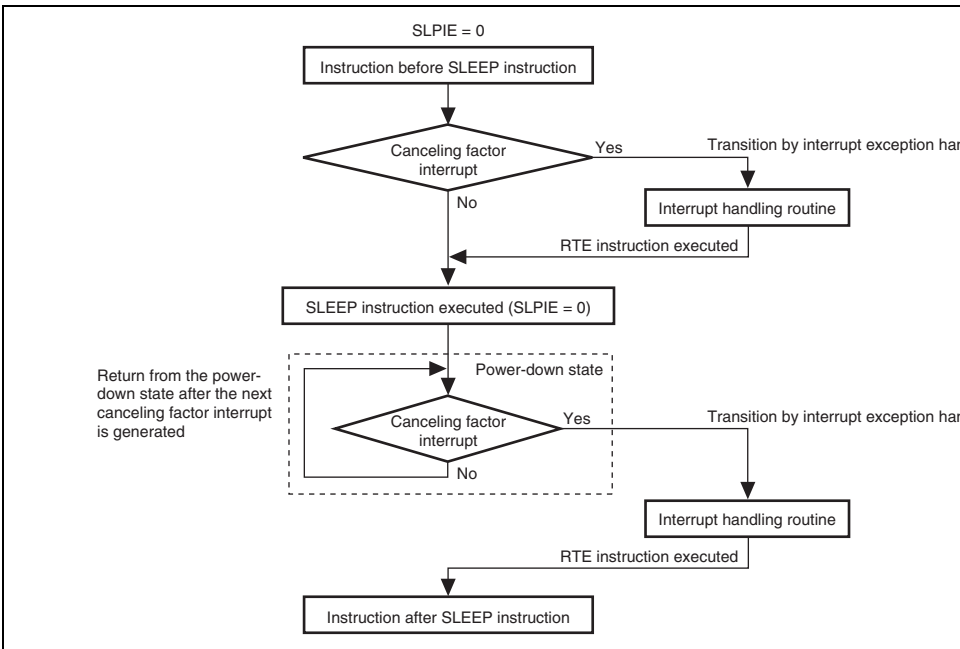


Figure 27.11 When Canceling Factor Interrupt is Generated before SLEEP Instruction Execution (Sleep Instruction Exception Handling Not Initiated)

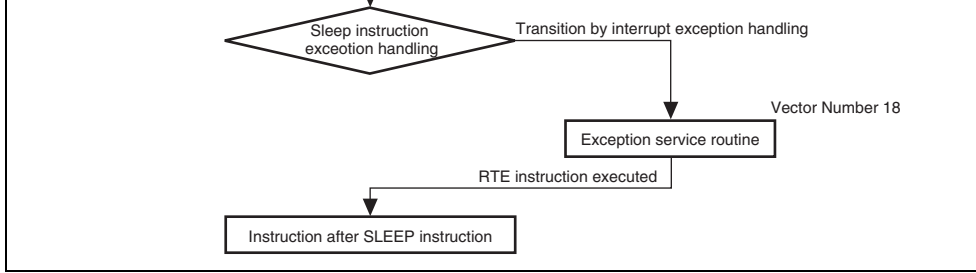


Figure 27.12 When Canceling Factor Interrupt is Generated before SLEEP Instruction Execution (Sleep Instruction Exception Handling Initiate

Tables 27.4 and 27.5 show the states of the B ϕ pin in each state of processing.

Table 27.4 ϕ Pin (PA7) State in Each State of Processing

Register Setting Value		Normal Operating Mode	Sleep Mode	All-Module-Clock-Stop Mode	Software Standby Mode		Deep Software Standby Mode	
DDR	PSTOP1				OPE = 0	OPE = 1	IOKEEP = 0	IOKEEP = 1
0	x	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
1	0	B ϕ output	B ϕ output	B ϕ output	High	High	High	High
1	1	High	High	High	High	High	High	High

Note: * SDRAM ϕ and control of this signal by the PSTOP0 bit are only supported by H8SX/1648G Group and H8SX/1648H Group.

Table 27.5 ϕ Pin (PB7) State in Each Processing State (When SDRAM Interface is Enabled)

Register Setting Value		Normal Operating Mode	Sleep Mode	All-Module-Clock-Stop Mode	Software Standby Mode		Deep Software Standby Mode	
PSTOP0					OPE = 0	OPE = 1	IOKEEP = 0	IOKEEP = 1
0		SDRAM ϕ output	SDRAM ϕ output	SDRAM ϕ output	High	High	High	High
1		High	High	High	High	High	High	High

Current consumption increases during the oscillation settling standby period.

27.12.3 Module Stop State of DMAC, EXDMAC*, or DTC

Depending on the operating state of the DMAC, EXDMAC, and DTC, bits MSTPA14 to MSTPA12 may not be set to 1, respectively. The module stop state setting for the DMAC, EXDMAC, or DTC should be carried out only when the DMAC or DTC is not activated.

For details, refer to section 10, DMA Controller (DMAC), section 11, EXDMA Controller (EXDMAC), and section 12, Data Transfer Controller (DTC).

Note: * EXDMAC is supported only by the H8SX/1648G Group and H8SX/1648H G

27.12.4 On-Chip Peripheral Module Interrupts

Relevant interrupt operations cannot be performed in a module stop state. Consequently, when a module stop state is entered when an interrupt has been requested, it will not be possible to clear the interrupt source or the DMAC or DTC activation source. Interrupts should therefore be disabled before entering a module stop state.

27.12.5 Writing to MSTPCRA, MSTPCRB, and MSTPCRC

MSTPCRA, MSTPCRB, and MSTPCRC should only be written to by the CPU.

If a conflict between a transition to deep software standby mode and generation of software standby mode clearing source occurs, a transition to deep software standby mode is not made until the software standby mode clearing sequence is executed. In this case, an interrupt exception handling for the input interrupt starts after the oscillation settling time for software standby mode (set by the STS4 to STS0 bits in SBYCR) has elapsed.

Note that if a conflict between a deep software standby mode transition and NMI interrupt occurs, the NMI interrupt exception handling routine is required.

If a conflict among a deep software standby mode transition, IRQ0 to IRQ15 interrupts, interrupt^{*1}, and voltage monitoring interrupt^{*2} occurs, a transition to deep software standby mode can be made without executing the interrupt execution handling by clearing the SSIn bit in SBYCR to 0 beforehand.

- Notes: 1. Supported only by the H8SX/1648G Group and H8SX/1648H Group.
2. Supported only by the H8SX/1648L Group and H8SX/1648H Group.

27.12.8 B ϕ or SDRAM ϕ * Output State

B ϕ and SDRAM ϕ outputs are undefined for a maximum of one cycle immediately after software standby mode is canceled with the IOKEEP bit cleared to 0 or immediately after the IOKEEP bit is cleared after cancellation of deep software standby mode with the IOKEEP bit set to 1.

However, B ϕ and SDRAM ϕ can be normally output by setting the IOKEEP, PSTOP1, and PSTOP0 bits to 1. For details, see section 27.8.4, B ϕ and SDRAM ϕ Operation after Exit from Deep Software Standby Mode.

Note: * SDRAM ϕ or PSTOP0 bit that controls it is supported only by the H8SX/1648L Group and H8SX/1648H Group.

clock. For details, refer to section 9.5.4, External Bus Interface.

- Among the internal I/O register area, addresses not listed in the list of registers are undefined or reserved addresses. Undefined and reserved addresses cannot be accessed. Do not access these addresses; otherwise, the operation when accessing these bits and subsequent operations cannot be guaranteed.
2. Register bits
 - Bit configurations of the registers are listed in the same order as the register addresses.
 - 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 control or 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 corresponding section.
 - The register states shown here are for the basic operating modes. If there is a specific mode for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

Time constant register B_4	TCORB_4	8	H'FEA46	TMR_4	16	3P
Time constant register B_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 register A_6	TCORA_6	8	H'FEA54	TMR_6	16	3P
Time constant register A_7	TCORA_7	8	H'FEA55	TMR_7	16	3P
Time constant register B_6	TCORB_6	8	H'FEA56	TMR_6	16	3P
Time constant register B_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
A/D data register A_1	ADDRA_1	16	H'FEA80	A/D_1	16	3P
A/D data register B_1	ADDRB_1	16	H'FEA82	A/D_1	16	3P
A/D data register C_1	ADDRC_1	16	H'FEA84	A/D_1	16	3P
A/D data register D_1	ADDRD_1	16	H'FEA86	A/D_1	16	3P
A/D data register E_2	ADDRE_2	16	H'FEA88	A/D_2	16	3P
A/D data register F_2	ADDRF_2	16	H'FEA8A	A/D_2	16	3P

A/D data register B_2	ADDRB_2	16	H'FEA9A	A/D_2	16	3
A/D data register C_2	ADDRC_2	16	H'FEA9C	A/D_2	16	3
A/D data register D_2	ADDRD_2	16	H'FEA9E	A/D_2	16	3
A/D control/status register_1	ADCSR_1	8	H'FEAA0	A/D_1	16	3
A/D control register_1	ADCR_1	8	H'FEAA1	A/D_1	16	3
A/D control/status register_2	ADCSR_2	8	H'FEAB0	A/D_2	16	3
A/D control register_2	ADCR_2	8	H'FEAB1	A/D_2	16	3
Port M data direction register* ²	PMDDR	8	H'FEE50	I/O port	8	3
Port M data register* ²	PMDR	8	H'FEE51	I/O port	8	3
Port M register* ²	PORTM	8	H'FEE52	I/O port	8	3
Port M input buffer control register* ²	PMICR	8	H'FEE53	I/O port	8	3
Port N data direction register	PNDDR	8	H'FEE54	I/O port	8	3
Port N data register	PNDR	8	H'FEE55	I/O port	8	3
Port N register	PORTN	8	H'FEE56	I/O port	8	3
Port N input buffer control register	PNICR	8	H'FEE57	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
Bit rate register_6	BRR_6	8	H'FF611	SCI_6	8	3

Next data enable register H_1	NDERH_1	8	H'FF638	PPG_1	8	3P
Next data enable register L_1	NDERL_1	8	H'FF639	PPG_1	8	3P
Output data register H_1	PODRH_1	8	H'FF63A	PPG_1	8	3P
Output data register L_1	PODRL_1	8	H'FF63B	PPG_1	8	3P
Next data register H_1* ¹	NDRH_1	8	H'FF63C	PPG_1	8	3P
Next data register L_1* ¹	NDRL_1	8	H'FF63D	PPG_1	8	3P
Next data register H_1* ¹	NDRH_1	8	H'FF63E	PPG_1	8	3P
Next data register L_1* ¹	NDRL_1	8	H'FF63F	PPG_1	8	3P
I ² C bus control register A_2	ICCRA_2	8	H'FF640	IIC2_2	8	3P
I ² C bus control register B_2	ICCRB_2	8	H'FF641	IIC2_2	8	3P
I ² C bus mode register_2	ICMR_2	8	H'FF642	IIC2_2	8	3P
I ² C bus interrupt enable register_2	ICIER_2	8	H'FF643	IIC2_2	8	3P
I ² C bus status register_2	ICSR_2	8	H'FF644	IIC2_2	8	3P
Slave address register_2	SAR_2	8	H'FF645	IIC2_2	8	3P
I ² C bus transmit data register_2	ICDRT_2	8	H'FF646	IIC2_2	8	3P
I ² C bus receive data register_2	ICDRR_2	8	H'FF647	IIC2_2	8	3P
I ² C bus control register A_3	ICCRA_3	8	H'FF648	IIC2_3	8	3P
I ² C bus control register B_3	ICCRB_3	8	H'FF649	IIC2_3	8	3P
I ² C bus mode register_3	ICMR_3	8	H'FF64A	IIC2_3	8	3P
I ² C bus interrupt enable register_3	ICIER_3	8	H'FF64B	IIC2_3	8	3P
I ² C bus status register_3	ICSR_3	8	H'FF64C	IIC2_3	8	3P
Slave address register_3	SAR_3	8	H'FF64D	IIC2_3	8	3P
I ² C bus transmit data register_3	ICDRT_3	8	H'FF64E	IIC2_3	8	3P
I ² C bus receive data register_3	ICDRR_3	8	H'FF64F	IIC2_3	8	3P

Break address mask register BL	BAMRBL	16	H'FFA0E	UBC	16	21
Break address register CH	BARCH	16	H'FFA10	UBC	16	21
Break address register CL	BARCL	16	H'FFA12	UBC	16	21
Break address mask register CH	BAMRCH	16	H'FFA14	UBC	16	21
Break address mask register CL	BAMRCL	16	H'FFA16	UBC	16	21
Break address register DH	BARDH	16	H'FFA18	UBC	16	21
Break address register DL	BARDL	16	H'FFA1A	UBC	16	21
Break address mask register DH	BAMRDH	16	H'FFA1C	UBC	16	21
Break address mask register DL	BAMRDL	16	H'FFA1E	UBC	16	21
Break control register A	BRCRA	16	H'FFA28	UBC	16	21
Break control register B	BRCRB	16	H'FFA2C	UBC	16	21
Break control register C	BRCRC	16	H'FFA30	UBC	16	21
Break control register D	BRCRD	16	H'FFA34	UBC	16	21
Timer control register* ²	TCR32K	8	H'FFABC	TM32K	8	21
Timer counter 1* ²	TCNT32K1	8	H'FFABD	TM32K	8	21
Timer counter 2* ²	TCNT32K2	8	H'FFABE	TM32K	8	21
Timer counter 3* ²	TCNT32K3	8	H'FFABF	TM32K	8	21
Timer start register	TSTRB	8	H'FFB00	TPU (unit 1)	16	21
Timer synchronous register	TSYRB	8	H'FFB01	TPU (unit 1)	16	21
Timer control register_6	TCR_6	8	H'FFB10	TPU_6	16	21
Timer mode register_6	TMDR_6	8	H'FFB11	TPU_6	16	21
Timer I/O control register H_6	TIORH_6	8	H'FFB12	TPU_6	16	21
Timer I/O control register L_6	TIORL_6	8	H'FFB13	TPU_6	16	21
Timer interrupt enable register_6	TIER_6	8	H'FFB14	TPU_6	16	21

Timer mode register_7	TMDR_7	8	H'FFB21	TPU_7	16	2P
Timer I/O control register_7	TIOR_7	8	H'FFB22	TPU_7	16	2P
Timer interrupt enable register_7	TIER_7	8	H'FFB24	TPU_7	16	2P
Timer status register_7	TSR_7	8	H'FFB25	TPU_7	16	2P
Timer counter_7	TCNT_7	16	H'FFB26	TPU_7	16	2P
Timer general register A_7	TGRA_7	16	H'FFB28	TPU_7	16	2P
Timer general register B_7	TGRB_7	16	H'FFB2A	TPU_7	16	2P
Timer control register_8	TCR_8	8	H'FFB30	TPU_8	16	2P
Timer mode register_8	TMDR_8	8	H'FFB31	TPU_8	16	2P
Timer I/O control register_8	TIOR_8	8	H'FFB32	TPU_8	16	2P
Timer interrupt enable register_8	TIER_8	8	H'FFB34	TPU_8	16	2P
Timer status register_8	TSR_8	8	H'FFB35	TPU_8	16	2P
Timer counter_8	TCNT_8	16	H'FFB36	TPU_8	16	2P
Timer general register A_8	TGRA_8	16	H'FFB38	TPU_8	16	2P
Timer general register B_8	TGRB_8	16	H'FFB3A	TPU_8	16	2P
Timer control register_9	TCR_9	8	H'FFB40	TPU_9	16	2P
Timer mode register_9	TMDR_9	8	H'FFB41	TPU_9	16	2P
Timer I/O control register H_9	TIORH_9	8	H'FFB42	TPU_9	16	2P
Timer I/O control register L_9	TIORL_9	8	H'FFB43	TPU_9	16	2P
Timer interrupt enable register_9	TIER_9	8	H'FFB44	TPU_9	16	2P
Timer status register_9	TSR_9	8	H'FFB45	TPU_9	16	2P
Timer counter_9	TCNT_9	16	H'FFB46	TPU_9	16	2P
Timer general register A_9	TGRA_9	16	H'FFB48	TPU_9	16	2P
Timer general register B_9	TGRB_9	16	H'FFB4A	TPU_9	16	2P

Timer counter_10	TCNT_10	16	H'FFB56	TPU_10	16	21
Timer general register A_10	TGRA_10	16	H'FFB58	TPU_10	16	21
Timer general register B_10	TGRB_10	16	H'FFB5A	TPU_10	16	21
Timer control register_11	TCR_11	8	H'FFB60	TPU_11	16	21
Timer mode register_11	TMDR_11	8	H'FFB61	TPU_11	16	21
Timer I/O control register_11	TIOR_11	8	H'FFB62	TPU_11	16	21
Timer interrupt enable register_11	TIER_11	8	H'FFB64	TPU_11	16	21
Timer status register_11	TSR_11	8	H'FFB65	TPU_11	16	21
Timer counter_11	TCNT_11	16	H'FFB66	TPU_11	16	21
Timer general register A_11	TGRA_11	16	H'FFB68	TPU_11	16	21
Timer general register B_11	TGRB_11	16	H'FFB6A	TPU_11	16	21
Port 1 data direction register	P1DDR	8	H'FFB80	I/O port	8	21
Port 2 data direction register	P2DDR	8	H'FFB81	I/O port	8	21
Port 3 data direction register	P3DDR	8	H'FFB82	I/O port	8	21
Port 6 data direction register	P6DDR	8	H'FFB85	I/O port	8	21
Port A data direction register	PADDR	8	H'FFB89	I/O port	8	21
Port B data direction register	PBDDR	8	H'FFB8A	I/O port	8	21
Port C data direction register	PCDDR	8	H'FFB8B	I/O port	8	21
Port D data direction register	PDDDR	8	H'FFB8C	I/O port	8	21
Port E data direction register	PEDDR	8	H'FFB8D	I/O port	8	21
Port F data direction register	PFDDR	8	H'FFB8E	I/O port	8	21
Port 1 input buffer control register	P1ICR	8	H'FFB90	I/O port	8	21
Port 2 input buffer control register	P2ICR	8	H'FFB91	I/O port	8	21
Port 3 input buffer control register	P3ICR	8	H'FFB92	I/O port	8	21
Port 4 input buffer control register	P4ICR	8	H'FFB93	I/O port	8	21
Port 5 input buffer control register	P5ICR	8	H'FFB94	I/O port	8	21

Port I register	PORTI	8	H'FFBA1	I/O port	8	2P
Port J register	PORTJ	8	H'FFBA2	I/O port	8	2P
Port K register	PORTK	8	H'FFBA3	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 J data register	PJDR	8	H'FFBA6	I/O port	8	2P
Port K data register	PKDR	8	H'FFBA7	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 J data direction register	PJDDR	8	H'FFBAA	I/O port	8	2P
Port K data direction register	PKDDR	8	H'FFBAB	I/O port	8	2P
Port H input buffer control register	PHICR	8	H'FFBAC	I/O port	8	2P
Port I input buffer control register	PIICR	8	H'FFBAD	I/O port	8	2P
Port J input buffer control register	PJICR	8	H'FFBAE	I/O port	8	2P
Port K input buffer control register	PKICR	8	H'FFBAF	I/O port	8	2P
Port D pull-up MOS control register	PDPCR	8	H'FFBB4	I/O port	8	2P
Port E pull-up MOS control register	PEPCR	8	H'FFBB5	I/O port	8	2P
Port F pull-up MOS control register	PFPCR	8	H'FFBB6	I/O port	8	2P
Port H pull-up MOS control register	PHPCR	8	H'FFBB8	I/O port	8	2P
Port I pull-up MOS control register	PIPCR	8	H'FFBB9	I/O port	8	2P
Port J pull-up MOS control register	PJPCR	8	H'FFBBA	I/O port	8	2P
Port K pull-up MOS control register	PKPCR	8	H'FFBBB	I/O port	8	2P
Port 2 open-drain control register	P2ODR	8	H'FFBBC	I/O port	8	2P
Port F open-drain control register	PFODR	8	H'FFBBD	I/O port	8	2P
Port function control register 0	PFCR0	8	H'FFBC0	I/O port	8	2P
Port function control register 1	PFCR1	8	H'FFBC1	I/O port	8	2P

Port function control register C	PFCRC	8	H'FFBCC	I/O port	8	21
Port function control register D	PFCRD	8	H'FFBCD	I/O port	8	21
Software standby release IRQ enable register	SSIER	16	H'FFBCE	INTC	8	21
Deep standby backup register 0	DPSBKR0	8	H'FFBF0	SYSTEM	8	21
Deep standby backup register 1	DPSBKR1	8	H'FFBF1	SYSTEM	8	21
Deep standby backup register 2	DPSBKR2	8	H'FFBF2	SYSTEM	8	21
Deep standby backup register 3	DPSBKR3	8	H'FFBF3	SYSTEM	8	21
Deep standby backup register 4	DPSBKR4	8	H'FFBF4	SYSTEM	8	21
Deep standby backup register 5	DPSBKR5	8	H'FFBF5	SYSTEM	8	21
Deep standby backup register 6	DPSBKR6	8	H'FFBF6	SYSTEM	8	21
Deep standby backup register 7	DPSBKR7	8	H'FFBF7	SYSTEM	8	21
Deep standby backup register 8	DPSBKR8	8	H'FFBF8	SYSTEM	8	21
Deep standby backup register 9	DPSBKR9	8	H'FFBF9	SYSTEM	8	21
Deep standby backup register 10	DPSBKR10	8	H'FFBFA	SYSTEM	8	21
Deep standby backup register 11	DPSBKR11	8	H'FFBFB	SYSTEM	8	21
Deep standby backup register 12	DPSBKR12	8	H'FFBFC	SYSTEM	8	21
Deep standby backup register 13	DPSBKR13	8	H'FFBFD	SYSTEM	8	21
Deep standby backup register 14	DPSBKR14	8	H'FFBFE	SYSTEM	8	21
Deep standby backup register 15	DPSBKR15	8	H'FFBFF	SYSTEM	8	21
DMA source address register_0	DSAR_0	32	H'FFC00	DMAC_0	16	21
DMA destination address register_0	DDAR_0	32	H'FFC04	DMAC_0	16	21
DMA offset register_0	DOFR_0	32	H'FFC08	DMAC_0	16	21
DMA transfer count register_0	DTCR_0	32	H'FFC0C	DMAC_0	16	21
DMA block size register_0	DBSR_0	32	H'FFC10	DMAC_0	16	21
DMA mode control register_0	DMDR_0	32	H'FFC14	DMAC_0	16	21
DMA address control register_0	DACR_0	32	H'FFC18	DMAC_0	16	21
DMA source address register_1	DSAR_1	32	H'FFC20	DMAC_1	16	21

DMA offset register_2	DOFR_2	32	H'FFC48	DMAC_2	16	21φ
DMA transfer count register_2	DTCR_2	32	H'FFC4C	DMAC_2	16	21φ
DMA block size register_2	DBSR_2	32	H'FFC50	DMAC_2	16	21φ
DMA mode control register_2	DMDR_2	32	H'FFC54	DMAC_2	16	21φ
DMA address control register_2	DACR_2	32	H'FFC58	DMAC_2	16	21φ
DMA source address register_3	DSAR_3	32	H'FFC60	DMAC_3	16	21φ
DMA destination address register_3	DDAR_3	32	H'FFC64	DMAC_3	16	21φ
DMA offset register_3	DOFR_3	32	H'FFC68	DMAC_3	16	21φ
DMA transfer count register_3	DTCR_3	32	H'FFC6C	DMAC_3	16	21φ
DMA block size register_3	DBSR_3	32	H'FFC70	DMAC_3	16	21φ
DMA mode control register_3	DMDR_3	32	H'FFC74	DMAC_3	16	21φ
DMA address control register_3	DACR_3	32	H'FFC78	DMAC_3	16	21φ
EXDMA source address register_0*2	EDSAR_0	32	H'FFC80	EXDMAC_0	16	21φ
EXDMA destination address register_0*2	EDDAR_0	32	H'FFC84	EXDMAC_0	16	21φ
EXDMA offset register_0*2	EDOFR_0	32	H'FFC88	EXDMAC_0	16	21φ
EXDMA transfer count register_0*2	EDTCR_0	32	H'FFC8C	EXDMAC_0	16	21φ
EXDMA block size register_0*2	EDBSR_0	32	H'FFC90	EXDMAC_0	16	21φ
EXDMA mode control register_0*2	EDMDR_0	32	H'FFC94	EXDMAC_0	16	21φ
EXDMA address control register_0*2	EDACR_0	32	H'FFC98	EXDMAC_0	16	21φ
EXDMA source address register_1*2	EDSAR_1	32	H'FFCA0	EXDMAC_1	16	21φ
EXDMA destination address register_1*2	EDDAR_1	32	H'FFCA4	EXDMAC_1	16	21φ
EXDMA offset register_1*2	EDOFR_1	32	H'FFCA8	EXDMAC_1	16	21φ
EXDMA transfer count register_1*2	EDTCR_1	32	H'FFCAC	EXDMAC_1	16	21φ
EXDMA block size register_1*2	EDBSR_1	32	H'FFCB0	EXDMAC_1	16	21φ

EXDMA mode control register_2* ²	EDMDR_2	32	H'FFCD4	EXDMAC_2	16	21
EXDMA address control register_2* ²	EDACR_2	32	H'FFCD8	EXDMAC_2	16	21
EXDMA source address register_2* ²	EDSAR_3	32	H'FFCE0	EXDMAC_3	16	21
EXDMA destination address register_3* ²	EDDAR_3	32	H'FFCE4	EXDMAC_3	16	21
EXDMA offset register_3* ²	EDOFR_3	32	H'FFCE8	EXDMAC_3	16	21
EXDMA transfer count register_3* ²	EDTCR_3	32	H'FFCEC	EXDMAC_3	16	21
EXDMA block size register_3* ²	EDBSR_3	32	H'FFCF0	EXDMAC_3	16	21
EXDMA mode control register_3* ²	EDMDR_3	32	H'FFCF4	EXDMAC_3	16	21
EXDMA address control register_3* ²	EDACR_3	32	H'FFCF8	EXDMAC_3	16	21
Cluster buffer register 0* ²	CLSBR0	32	H'FFD00	EXDMAC	16	21
Cluster buffer register 1* ²	CLSBR1	32	H'FFD04	EXDMAC	16	21
Cluster buffer register 2* ²	CLSBR2	32	H'FFD08	EXDMAC	16	21
Cluster buffer register 3* ²	CLSBR3	32	H'FFD0C	EXDMAC	16	21
Cluster buffer register 4* ²	CLSBR4	32	H'FFD10	EXDMAC	16	21
Cluster buffer register 5* ²	CLSBR5	32	H'FFD14	EXDMAC	16	21
Cluster buffer register 6* ²	CLSBR6	32	H'FFD18	EXDMAC	16	21
Cluster buffer register 7* ²	CLSBR7	32	H'FFD1C	EXDMAC	16	21
DMA module request select register_0	DMRSR_0	8	H'FFD20	DMAC_0	16	21
DMA module request select register_1	DMRSR_1	8	H'FFD21	DMAC_1	16	21
DMA module request select register_2	DMRSR_2	8	H'FFD22	DMAC_2	16	21
DMA module request select register_3	DMRSR_3	8	H'FFD23	DMAC_3	16	21
Interrupt priority register A	IPRA	16	H'FFD40	INTC	16	21
Interrupt priority register B	IPRB	16	H'FFD42	INTC	16	21
Interrupt priority register C	IPRC	16	H'FFD44	INTC	16	21
Interrupt priority register D	IPRD	16	H'FFD46	INTC	16	21

Interrupt priority register M	IPRM	16	H'FFD58	INTC	16	21φ
Interrupt priority register N	IPRN	16	H'FFD5A	INTC	16	21φ
Interrupt priority register O	IPRO	16	H'FFD5C	INTC	16	21φ
Interrupt priority register Q	IPRQ	16	H'FFD60	INTC	16	21φ
Interrupt priority register R	IPRR	16	H'FFD62	INTC	16	21φ
IRQ sense control register H	ISCRH	16	H'FFD68	INTC	16	21φ
IRQ sense control register L	ISCR L	16	H'FFD6A	INTC	16	21φ
DTC vector base register	DTCVBR	32	H'FFD80	BSC	16	21φ
Bus width control register	ABWCR	16	H'FFD84	BSC	16	21φ
Access state control register	ASTCR	16	H'FFD86	BSC	16	21φ
Wait control register A	WTCRA	16	H'FFD88	BSC	16	21φ
Wait control register B	WTCRB	16	H'FFD8A	BSC	16	21φ
Read strobe timing control register	RDNCR	16	H'FFD8C	BSC	16	21φ
\overline{CS} assertion period control register	CSACR	16	H'FFD8E	BSC	16	21φ
Idle control register	IDLCR	16	H'FFD90	BSC	16	21φ
Bus control register1	BCR1	16	H'FFD92	BSC	16	21φ
Bus control register2	BCR2	8	H'FFD94	BSC	16	21φ
Endian control register	ENDIANCR	8	H'FFD95	BSC	16	21φ
SRAM mode control register	SRAMCR	16	H'FFD98	BSC	16	21φ
Burst ROM interface control register	BROMCR	16	H'FFD9A	BSC	16	21φ
Address/data multiplexed I/O control register	MPXCR	16	H'FFD9C	BSC	16	21φ

Mode control register	MDCR	16	H'FFDC0	SYSTEM	16	21
System control register	SYSCR	16	H'FFDC2	SYSTEM	16	21
System clock control register	SCKCR	16	H'FFDC4	SYSTEM	16	21
Standby control register	SBYCR	16	H'FFDC6	SYSTEM	16	21
Module stop control register A	MSTPCRA	16	H'FFDC8	SYSTEM	16	21
Module stop control register B	MSTPCRB	16	H'FFDCA	SYSTEM	16	21
Module stop control register C	MSTPCRC	16	H'FFDCC	SYSTEM	16	21
Subclock control register	SUBCKCR ^{*2}	8	H'FFDCF	SYSTEM	16	21
Flash code control/status register	FCCS	8	H'FFDE8	FLASH	16	21
Flash program code select register	FPCS	8	H'FFDE9	FLASH	16	21
Flash erase code select register	FECS	8	H'FFDEA	FLASH	16	21
Flash key code register	FKEY	8	H'FFDEC	FLASH	16	21
Flash MAT select register	FMATS	8	H'FFDED	FLASH	16	21
Flash transfer destination address register	FTDAR	8	H'FFDEE	FLASH	16	21
Deep standby control register	DPSBYCR	8	H'FFE70	SYSTEM	8	21
Deep standby wait control register	DPSWCR	8	H'FFE71	SYSTEM	8	21
Deep standby interrupt enable register	DPSIER	8	H'FFE72	SYSTEM	8	21
Deep standby interrupt flag register	DPSIFR	8	H'FFE73	SYSTEM	8	21
Deep standby interrupt edge register	DPSIEGR	8	H'FFE74	SYSTEM	8	21
Reset status register	RSTSR	8	H'FFE75	SYSTEM	8	21
Low voltage detection control register ^{*3}	LVDCR	8	H'FFE78	SYSTEM	8	21
Serial extended mode register_2	SEMR_2	8	H'FFE84	SCI_2	8	21
Serial mode register_3	SMR_3	8	H'FFE88	SCI_3	8	21
Bit rate register_3	BRR_3	8	H'FFE89	SCI_3	8	21
Serial control register_3	SCR_3	8	H'FFE8A	SCI_3	8	21

Transmit data register_4	TDR_4	8	H'FFE93	SCI_4	8	2P
Serial status register_4	SSR_4	8	H'FFE94	SCI_4	8	2P
Receive data register_4	RDR_4	8	H'FFE95	SCI_4	8	2P
Smart card mode register_4	SCMR_4	8	H'FFE96	SCI_4	8	2P
I ² C bus control register A_0	ICCRA_0	8	H'FFEB0	IIC2_0	8	2P
I ² C bus control register B_0	ICCRB_0	8	H'FFEB1	IIC2_0	8	2P
I ² C bus mode register_0	ICMR_0	8	H'FFEB2	IIC2_0	8	2P
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

Timer interrupt enable register_4	TIER_4	8	H'FFEE4	TPU_4	16	2
Timer status register_4	TSR_4	8	H'FFEE5	TPU_4	16	2
Timer counter_4	TCNT_4	16	H'FFEE6	TPU_4	16	2
Timer general register A_4	TGRA_4	16	H'FFEE8	TPU_4	16	2
Timer general register B_4	TGRB_4	16	H'FFEEA	TPU_4	16	2
Timer control register_5	TCR_5	8	H'FFEF0	TPU_5	16	2
Timer mode register_5	TMDR_5	8	H'FFEF1	TPU_5	16	2
Timer I/O control register_5	TIOR_5	8	H'FFEF2	TPU_5	16	2
Timer interrupt enable register_5	TIER_5	8	H'FFEF4	TPU_5	16	2
Timer status register_5	TSR_5	8	H'FFEF5	TPU_5	16	2
Timer counter_5	TCNT_5	16	H'FFEF6	TPU_5	16	2
Timer general register A_5	TGRA_5	16	H'FFEF8	TPU_5	16	2
Timer general register B_5	TGRB_5	16	H'FFEFA	TPU_5	16	2
DTC enable register A	DTCERA	16	H'FFF20	INTC	16	2
DTC enable register B	DTCERB	16	H'FFF22	INTC	16	2
DTC enable register C	DTCERC	16	H'FFF24	INTC	16	2
DTC enable register D	DTCERD	16	H'FFF26	INTC	16	2
DTC enable register E	DTCERE	16	H'FFF28	INTC	16	2
DTC enable register F	DTCERF	16	H'FFF2A	INTC	16	2
DTC control register	DTCCR	8	H'FFF30	INTC	16	2
Interrupt control register	INTCR	8	H'FFF32	INTC	16	2
CPU priority control register	CPUPCR	8	H'FFF33	INTC	16	2
IRQ enable register	IER	16	H'FFF34	INTC	16	2
IRQ status register	ISR	16	H'FFF36	INTC	16	2
Port 1 register	PORT1	8	H'FFF40	I/O port	8	2
Port 2 register	PORT2	8	H'FFF41	I/O port	8	2
Port 3 register	PORT3	8	H'FFF42	I/O port	8	2

Port E register	PORTE	8	H'FFF4D	I/O port	8	2P
Port F register	PORTF	8	H'FFF4E	I/O port	8	2P
Port 1 data register	P1DR	8	H'FFF50	I/O port	8	2P
Port 2 data register	P2DR	8	H'FFF51	I/O port	8	2P
Port 3 data register	P3DR	8	H'FFF52	I/O port	8	2P
Port 6 data register	P6DR	8	H'FFF55	I/O port	8	2P
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_0	8	2P
PPG output mode register	PMR	8	H'FFF77	PPG_0	8	2P
Next data enable register H	NDERH	8	H'FFF78	PPG_0	8	2P
Next data enable register L	NDERL	8	H'FFF79	PPG_0	8	2P
Output data register H	PODRH	8	H'FFF7A	PPG_0	8	2P

Transmit data register_0	TDR_0	8	H'FFF83	SCI_0	8	21
Serial status register_0	SSR_0	8	H'FFF84	SCI_0	8	21
Receive data register_0	RDR_0	8	H'FFF85	SCI_0	8	21
Smart card mode register_0	SCMR_0	8	H'FFF86	SCI_0	8	21
Serial mode register_1	SMR_1	8	H'FFF88	SCI_1	8	21
Bit rate register_1	BRR_1	8	H'FFF89	SCI_1	8	21
Serial control register_1	SCR_1	8	H'FFF8A	SCI_1	8	21
Transmit data register_1	TDR_1	8	H'FFF8B	SCI_1	8	21
Serial status register_1	SSR_1	8	H'FFF8C	SCI_1	8	21
Receive data register_1	RDR_1	8	H'FFF8D	SCI_1	8	21
Smart card mode register_1	SCMR_1	8	H'FFF8E	SCI_1	8	21
A/D data register A_0	ADDRA_0	16	H'FFF90	A/D_0	16	21
A/D data register B_0	ADDRB_0	16	H'FFF92	A/D_0	16	21
A/D data register C_0	ADDRC_0	16	H'FFF94	A/D_0	16	21
A/D data register D_0	ADDRD_0	16	H'FFF96	A/D_0	16	21
A/D data register E_0	ADDRE_0	16	H'FFF98	A/D_0	16	21
A/D data register F_0	ADDRF_0	16	H'FFF9A	A/D_0	16	21
A/D data register G_0	ADDRG_0	16	H'FFF9C	A/D_0	16	21
A/D data register H_0	ADDRH_0	16	H'FFF9E	A/D_0	16	21
A/D control/status register_0	ADCSR_0	8	H'FFFA0	A/D_0	16	21
A/D control register_0	ADCR_0	8	H'FFFA1	A/D_0	16	21
Timer control/status register	TCSR	8	H'FFFA4	WDT	16	21
Timer counter	TCNT	8	H'FFFA5	WDT	16	21
Reset control/status register	RSTCSR	8	H'FFFA7	WDT	16	21
Timer control register_0	TCR_0	8	H'FFFB0	TMR_0	16	21
Timer control register_1	TCR_1	8	H'FFFB1	TMR_1	16	21
Timer control/status register_0	TCSR_0	8	H'FFFB2	TMR_0	16	21
Timer control/status register_1	TCSR_1	8	H'FFFB3	TMR_1	16	21

Timer start register	TSTR	8	H'FFFBC	TPU	16	2P
Timer synchronous register	TSYR	8	H'FFFBD	TPU	16	2P
Timer control register_0	TCR_0	8	H'FFFC0	TPU_0	16	2P
Timer mode register_0	TMDR_0	8	H'FFFC1	TPU_0	16	2P
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'FFCA	TPU_0	16	2P
Timer general register C_0	TGRC_0	16	H'FFCC	TPU_0	16	2P
Timer general register D_0	TGRD_0	16	H'FFCE	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'FFDA	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 A_3	TGRA_3	16	H'FFFF8	TPU_3	16	2
Timer general register B_3	TGRB_3	16	H'FFFFA	TPU_3	16	2
Timer general register C_3	TGRC_3	16	H'FFFFC	TPU_3	16	2
Timer general register D_3	TGRD_3	16	H'FFFFE	TPU_3	16	2

Notes: 1. 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 specified, the 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 output groups 0 and 1 by the PCR setting, the NDRL address is H'FFF7D. When different output triggers are specified, the NDRL addresses for pulse output groups 0 and 1 are H'FFF7F and H'FFF7D, respectively.

When the same output trigger is specified for pulse output groups 6 and 7 by setting, the NDRH address is H'FF63C. When different output triggers are specified, the NDRH addresses for pulse output groups 6 and 7 are H'FF63E and H'FF63C respectively.

When the same output trigger is specified for pulse output groups 4 and 5 by setting, the NDRL address is H'FF63D. When different output triggers are specified, the NDRL addresses for pulse output groups 4 and 5 are H'FF63F and H'FF63D, respectively.

2. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.
3. Supported only by the H8SX/1648L Group and the H8SX/1648H Group.

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
ADDRA_1								

ADDRG_2

ADDRH_2

ADDRE_1

ADDRF_1

ADDRG_1

ADDRH_1

ADDRA_2

ADDRB_2

ADDRC_2

ADDRD_2

ADCSR_1	ADF	ADIE	ADST	EXCKS	CH3	CH2	CH1	CH0
ADCR_1	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	EXTRGS
ADCSR_2	ADF	ADIE	ADST	EXCKS	CH3	CH2	CH1	CH0
ADCR_2	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	EXTRGS

SMR_5* ¹	C/Ā	CHR	PE	O/Ē	STOP	MP	CKS1	CKS0
	(GM)	(BLK)	(PE)	(O/Ē)	(BCP1)	(BCP0)		
BRR_5								
SCR_5* ¹	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_5								
SSR_5* ¹	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
	(ERS)							
RDR_5								
SCMR_5	—	—	—	—	SDIR	SINV	—	SMIF
SEMR_5	—	—	—	ABCS	ACS3	ACS2	ACS1	ACS0
IrCR	IrE	IrCKS2	IrCKS1	IrCKS0	IrTxINV	IrRxINV	—	—
SMR_6* ¹	C/Ā	CHR	PE	O/Ē	STOP	MP	CKS1	CKS0
	(GM)	(BLK)	(PE)	(O/Ē)	(BCP1)	(BCP0)		
BRR_6								
SCR_6* ¹	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_6								
SSR_6* ¹	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
	(ERS)							
RDR_6								
SCMR_6	—	—	—	—	SDIR	SINV	—	SMIF
SEMR_6	—	—	—	ABCS	ACS3	ACS2	ACS1	ACS0
PCR_1	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
PMR_1	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
NDERH_1	NDER31	NDER30	NDER29	NDER28	NDER27	NDER26	NDER25	NDER24
NDERL_1	NDER23	NDER22	NDER21	NDER20	NDER19	NDER18	NDER17	NDER16

ICMR_2	—	WAIT	—	—	BCWP	BC2	BC1	BC0
ICIER_2	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
ICSR_2	TDRE	TEND	RDRF	NACKF	STOP	AL	AAS	ADZ
SAR_2	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	—
ICDRT_2								
ICDRR_2								
ICCRA_3	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0
ICCRB_3	BBSY	SCP	SDAO	—	SCLO	—	IICRST	—
ICMR_3	—	WAIT	—	—	BCWP	BC2	BC1	BC0
ICIER_3	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
ICSR_3	TDRE	TEND	RDRF	NACKF	STOP	AL	AAS	ADZ
SAR_3	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	—
ICDRT_3								
ICDRR_3								
BARAH	BARA31	BARA30	BARA29	BARA28	BARA27	BARA26	BARA25	BARA24
	BARA23	BARA22	BARA21	BARA20	BARA19	BARA18	BARA17	BARA16
BARAL	BARA15	BARA14	BARA13	BARA12	BARA11	BARA10	BARA9	BARA8
	BARA7	BARA6	BARA5	BARA4	BARA3	BARA2	BARA1	BARA0
BAMRAH	BAMRA31	BAMRA30	BAMRA29	BAMRA28	BAMRA27	BAMRA26	BAMRA25	BAMRA24
	BAMRA23	BAMRA22	BAMRA21	BAMRA20	BAMRA19	BAMRA18	BAMRA17	BAMRA16
BAMRAL	BAMRA15	BAMRA14	BAMRA13	BAMRA12	BAMRA11	BAMRA10	BAMRA9	BAMRA8
	BAMRA7	BAMRA6	BAMRA5	BAMRA4	BAMRA3	BAMRA2	BAMRA1	BAMRA0
BARBH	BARB31	BARB30	BARB29	BARB28	BARB27	BARB26	BARB25	BARB24
	BARB23	BARB22	BARB21	BARB20	BARB19	BARB18	BARB17	BARB16

BARCL	BARC15	BARC14	BARC13	BARC12	BARC11	BARC10	BARC9	BARC8
	BARC7	BARC6	BARC5	BARC4	BARC3	BARC2	BARC1	BARC0
BAMRCH	BAMRC31	BAMRC30	BAMRC29	BAMRC28	BAMRC27	BAMRC26	BAMRC25	BAMRC24
	BAMRC23	BAMRC22	BAMRC21	BAMRC20	BAMRC19	BAMRC18	BAMRC17	BAMRC16
BAMRCL	BAMRC15	BAMRC14	BAMRC13	BAMRC12	BAMRC11	BAMRC10	BAMRC9	BAMRC8
	BAMRC7	BAMRC6	BAMRC5	BAMRC4	BAMRC3	BAMRC2	BAMRC1	BAMRC0
BARDH	BARD31	BARD30	BARD29	BARD28	BARD27	BARD26	BARD25	BARD24
	BARD23	BARD22	BARD21	BARD20	BARD19	BARD18	BARD17	BARD16
BARDL	BARD15	BARD14	BARD13	BARD12	BARD11	BARD10	BARD9	BARD8
	BARD7	BARD6	BARD5	BARD4	BARD3	BARD2	BARD1	BARD0
BAMRDH	BAMRD31	BAMRD30	BAMRD29	BAMRD28	BAMRD27	BAMRD26	BAMRD25	BAMRD24
	BAMRD23	BAMRD22	BAMRD21	BAMRD20	BAMRD19	BAMRD18	BAMRD17	BAMRD16
BAMRDL	BAMRD15	BAMRD14	BAMRD13	BAMRD12	BAMRD11	BAMRD10	BAMRD9	BAMRD8
	BAMRD7	BAMRD6	BAMRD5	BAMRD4	BAMRD3	BAMRD2	BAMRD1	BAMRD0
BRCRA	—	—	CMFCPA	—	CPA2	CPA1	CPA0	—
	—	—	IDA1	IDA0	RWA1	RWA0	—	—
BRCRB	—	—	CMFCPB	—	CPB2	CPB1	CPB0	—
	—	—	IDB1	IDB0	RWB1	RWB0	—	—
BRCRC	—	—	CMFCPC	—	CPC2	CPC1	CPC0	—
	—	—	IDC1	IDC0	RWC1	RWC0	—	—
BRCRD	—	—	DMFCPD	—	CPD2	CPD1	CPD0	—
	—	—	IDD1	IDD0	RWD1	RWD0	—	—

TIORH_6	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIORL_6	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
TIER_6	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
TSR_6	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
TCNT_6	_____							
TGRA_6	_____							
TGRB_6	_____							
TGRC_6	_____							
TGRD_6	_____							
TCR_7	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_7	—	—	—	—	MD3	MD2	MD1	MD0
TIOR_7	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIER_7	—	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_7	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_7	_____							
TGRA_7	_____							
TGRB_7	_____							

TGRB_8

TCR_9	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_9	—	—	BFB	BFA	MD3	MD2	MD1	MD0
TIORH_9	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIORL_9	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
TIER_9	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
TSR_9	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
TCNT_9								

TGRA_9

TGRB_9

TGRC_9

TGRD_9

TCR_10	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_10	—	—	—	—	MD3	MD2	MD1	MD0
TIOR_10	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIER_10	—	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_10	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_10								

TSR_11	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_11								
TGRA_11								
TGRB_11								
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	P30DDR
P6DDR	P67DDR	P66DDR	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	—	—	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR
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
P4ICR	P47ICR	P46ICR	P45ICR	P44ICR	—	—	—	—
P5ICR	P57ICR	P56ICR	P55ICR	P54ICR	P53ICR	P52ICR	P51ICR	P50ICR
P6ICR	P67ICR	P66ICR	P65ICR	P64ICR	P63ICR	P62ICR	P61ICR	P60ICR
PAICR	PA7ICR	PA6ICR	PA5ICR	PA4ICR	PA3ICR	PA2ICR	PA1ICR	PA0ICR
PBICR	PB7ICR	PB6ICR	PB5ICR	PB4ICR	PB3ICR	PB2ICR	PB1ICR	PB0ICR

PHDR	PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR
PIDR	PI7DR	PI6DR	PI5DR	PI4DR	PI3DR	PI2DR	PI1DR	PI0DR
PJDR	PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR
PKDR	PK7DR	PK6DR	PK5DR	PK4DR	PK3DR	PK2DR	PK1DR	PK0DR
PHDDR	PH7DDR	PH6DDR	PH5DDR	PH4DDR	PH3DDR	PH2DDR	PH1DDR	PH0DDR
PIDDR	PI7DDR	PI6DDR	PI5DDR	PI4DDR	PI3DDR	PI2DDR	PI1DDR	PI0DDR
PJDDR	PJ7DDR	PJ6DDR	PJ5DDR	PJ4DDR	PJ3DDR	PJ2DDR	PJ1DDR	PJ0DDR
PKDDR	PK7DDR	PK6DDR	PK5DDR	PK4DDR	PK3DDR	PK2DDR	PK1DDR	PK0DDR
PHICR	PH7ICR	PH6ICR	PH5ICR	PH4ICR	PH3ICR	PH2ICR	PH1ICR	PH0ICR
PIICR	PI7ICR	PI6ICR	PI5ICR	PI4ICR	PI3ICR	PI2ICR	PI1ICR	PI0ICR
PJICR	PJ7ICR	PJ6ICR	PJ5ICR	PJ4ICR	PJ3ICR	PJ2ICR	PJ1ICR	PJ0ICR
PKICR	PK7ICR	PK6ICR	PK5ICR	PK4ICR	PK3ICR	PK2ICR	PK1ICR	PK0ICR
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
PFPCR	PF7PCR	PF6PCR	PF5PCR	PF4PCR	PF3PCR	PF2PCR	PF1PCR	PF0PCR
PHPCR	PH7PCR	PH6PCR	PH5PCR	PH4PCR	PH3PCR	PH2PCR	PH1PCR	PH0PCR
PIPCR	PI7PCR	PI6PCR	PI5PCR	PI4PCR	PI3PCR	PI2PCR	PI1PCR	PI0PCR
PJPCR	PJ7PCR	PJ6PCR	PJ5PCR	PJ4PCR	PJ3PCR	PJ2PCR	PJ1PCR	PJ0PCR
PKPCR	PK7PCR	PK6PCR	PK5PCR	PK4PCR	PK3PCR	PK2PCR	PK1PCR	PK0PCR
P2ODR	P27ODR	P26ODR	P25ODR	P24ODR	P23ODR	P22ODR	P21ODR	P20ODR
PFODR	PF7ODR	PF6ODR	PF5ODR	PF4ODR	PF3ODR	PF2ODR	PF1ODR	PF0ODR
PFCR0	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E
PFCR1	CS7SA	CS7SB	CS6SA	CS6SB	CS5SA	CS5SB	CS4SA	CS4SB
PFCR2	CS3S	CS2S	BSS	BSE	RDWRS	RDWRE	ASOE	WAITS

SSIER	SSI15	SSI14	SSI13	SSI12	SSI11	SSI10	SSI9	SSI8
	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	SSIO
DPSBKR0	BKUP07	BKUP06	BKUP05	BKUP04	BKUP03	BKUP02	BKUP01	BKUP00
DPSBKR1	BKUP17	BKUP16	BKUP15	BKUP14	BKUP13	BKUP12	BKUP11	BKUP10
DPSBKR2	BKUP27	BKUP26	BKUP25	BKUP24	BKUP23	BKUP22	BKUP21	BKUP20
DPSBKR3	BKUP37	BKUP36	BKUP35	BKUP34	BKUP33	BKUP32	BKUP31	BKUP30
DPSBKR4	BKUP47	BKUP46	BKUP45	BKUP44	BKUP43	BKUP42	BKUP41	BKUP40
DPSBKR5	BKUP57	BKUP56	BKUP55	BKUP54	BKUP53	BKUP52	BKUP51	BKUP50
DPSBKR6	BKUP67	BKUP66	BKUP65	BKUP64	BKUP63	BKUP62	BKUP61	BKUP60
DPSBKR7	BKUP77	BKUP76	BKUP75	BKUP74	BKUP73	BKUP72	BKUP71	BKUP70
DPSBKR8	BKUP87	BKUP86	BKUP85	BKUP84	BKUP83	BKUP82	BKUP81	BKUP80
DPSBKR9	BKUP97	BKUP96	BKUP95	BKUP94	BKUP93	BKUP92	BKUP91	BKUP90
DPSBKR10	BKUP107	BKUP106	BKUP105	BKUP104	BKUP103	BKUP102	BKUP101	BKUP100
DPSBKR11	BKUP117	BKUP116	BKUP115	BKUP114	BKUP113	BKUP112	BKUP111	BKUP110
DPSBKR12	BKUP127	BKUP126	BKUP125	BKUP124	BKUP123	BKUP122	BKUP121	BKUP120
DPSBKR13	BKUP137	BKUP136	BKUP135	BKUP134	BKUP133	BKUP132	BKUP131	BKUP130
DPSBKR14	BKUP147	BKUP146	BKUP145	BKUP144	BKUP143	BKUP142	BKUP141	BKUP140
DPSBKR15	BKUP157	BKUP156	BKUP155	BKUP154	BKUP153	BKUP152	BKUP151	BKUP150

DSAR_0

DDAR_0



	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16
	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	DMAPO
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	BKSZH24
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16
	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	BKSZH24
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16
	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	DMAPO
DACR_3	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0
	—	—	SAT1	SAT0	—	—	DAT1	DAT0
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0

EDOFR_0*³

EDTCR_0*³

EDBSR_0*³

BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24
BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16
BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8
BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0

EDMDR_0*³

DTE	EDACHE	ETENDE	EDRAKE	EDREQS	NRD	—	—
ACT	—	—	—	ERRF	—	ESIF	DTIF
DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
DTF1	DTF0	—	—	—	EDMAP2	EDMAP1	EDMAP0

EDACR_0*³

AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0
—	—	SAT1	SAT0	—	—	DAT1	DAT0
SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0

EDOFR_1*³

EDTCR_1*³

EDBSR_1*³

BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24
BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16
BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8
BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0

EDMDR_1*³

DTE	EDACKE	ETENDE	EDRAKE	EDREQS	NRD	—	—
ACT	—	—	—	—	—	ESIF	DTIF
DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
DTF1	DTF0	—	—	—	EDMAP2	EDMAP1	EDMAP0

EDACR_1*³

AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0
—	—	SAT1	SAT0	—	—	DAT1	DAT0
SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0

EDSAR_2*³

EDTCR_2*³

EDBSR_2*³

BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24	BKSZH23
BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16	BKSZH15
BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8	BKSZ7
BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0	BKSZ0

EDMDR_2*³

DTE	EDACKE	ETENDE	EDRAKE	EDREQS	NRD	—	—	—
ACT	—	—	—	—	—	ESIF	DTIF	—
DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE	—
DTF1	DTF0	—	—	—	EDMAP2	EDMAP1	EDMAP0	—

EDACR_2*³

AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0	—
—	—	SAT1	SAT0	—	—	DAT1	DAT0	—
SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0	—
DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0	—

EDSAR_3*³

EDDAR_3*³

EDBSR_3* ³	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16
	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8
	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0

EDMDR_3* ³	DTE	EDACKE	ETENDE	EDRAKE	EDREQS	NRD	—	—
	ACT	—	—	—	—	—	ESIF	DTIF
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
	DTF1	DTF0	—	—	—	EDMAP2	EDMAP1	EDMAP0

EDACR_3* ³	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0
	—	—	SAT1	SAT0	—	—	DAT1	DAT0
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0

CLSBR0* ³								

CLSBR1* ³								

CLSBR2* ³								

CLSBR5*³

CLSBR6*³

CLSBR7*³

DMRSR_0

DMRSR_1

DMRSR_2

DMRSR_3

IPRA	—	IPRA14	IPRA13	IPRA12	—	IPRA10	IPRA9	IPRA8
	—	IPRA6	IPRA5	IPRA4	—	IPRA2	IPRA1	IPRA0
IPRB	—	IPRB14	IPRB13	IPRB12	—	IPRB10	IPRB9	IPRB8
	—	IPRB6	IPRB5	IPRB4	—	IPRB2	IPRB1	IPRB0
IPRC	—	IPRC14	IPRC13	IPRC12	—	IPRC10	IPRC9	IPRC8
	—	IPRC6	IPRC5	IPRC4	—	IPRC2	IPRC1	IPRC0
IPRD	—	IPRD14	IPRD13	IPRD12	—	IPRD10	IPRD9	IPRD8
	—	IPRD6	IPRD5	IPRD4	—	IPRD2	IPRD1	IPRD0
IPRE	—	—	—	—	—	IPRE10	IPRE9	IPRE8
	—	—	—	—	—	—	—	—

	—	IPRJ6	IPRJ5	IPRJ4	—	IPRJ2	IPRJ1	IPRJ0
IPRK	—	IPRK14	IPRK13	IPRK12	—	IPRK10* ³	IPRK9* ³	IPRK8* ³
	—	IPRK6	IPRK5	IPRK4	—	IPRK2	IPRK1	IPRK0
IPRL	—	IPRL14	IPRL13	IPRL12	—	IPRL10	IPRL9	IPRL8
	—	IPRL6	IPRL5	IPRL4	—	IPRL2	IPRL1	IPRL0
IPRM	—	IPRM14	IPRM13	IPRM12	—	IPRM10	IPRM9	IPRM8
	—	IPRM6	IPRM5	IPRM4	—	IPRM2	IPRM1	IPRM0
IPRN	—	IPRN14	IPRN13	IPRN12	—	IPRN10	IPRN9	IPRN8
	—	IPRN6	IPRN5	IPRN4	—	IPRN2	IPRN1	IPRN0
IPRO	—	IPRO14	IPRO13	IPRO12	—	IPRO10	IPRO9	IPRO8
	—	IPRO6	IPRO5	IPRO4	—	—	—	—
IPRQ	—	—	—	—	—	—	—	—
	—	IPRQ6	IPRQ5	IPRQ4	—	IPRQ2	IPRQ1	IPRQ0
IPRR	—	IPRR14	IPRR13	IPRR12	—	IPRR10	IPRR9	IPRR8
	—	IPRR6	IPRR5	IPRR4	—	IPRR2	IPRR1	IPRR0
ISCRH	IRQ15SR	IRQ15SF	IRQ14SR	IRQ14SF	IRQ13SR	IRQ13SF	IRQ12SR	IRQ12SF
	IRQ11SR	IRQ11SF	IRQ10SR	IRQ10SF	IRQ9SR	IRQ9SF	IRQ8SR	IRQ8SF
ISCR L	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	EDKC* ³	—	—	—	—	—	—
BCR2	—	—	EBCCS* ³	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
RAMER	—	—	—	—	RAMS	RAM2	RAM1	RAM0
DRAMCR* ³	DRAME	DTYPE	—	—	OEE	RAST	—	CAST
	BE	RCDM	DDS	EDDS	—	—	MXC1	MXC0
DRACCR* ³	—	—	TPC1	TPC0	—	—	RCD1	RCD0
	—	—	—	—	—	—	—	—
SDCR* ³	MRSE	—	—	—	—	—	—	—
	CKSPE	—	—	—	—	—	—	TRWL
REFCR* ³	CMF	CMIE	RCW1	RCW0	—	RTCK2	RTCK1	RTCK0
	RFSHE	RLW2	RLW1	RLW0	SLFRF	TPCS2	TPCS1	TPCS0
RTCNT* ³								
RTCOR* ³								

	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
MSTPCRB	MSTPB15	MSTPB14	MSTPB13	MSTPB12	MSTPB11	MSTPB10	MSTPB9	MSTPB8
	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
MSTPCRC	MSTPC15	MSTPC14	MSTPC13	MSTPC12	MSTPC11	MSTPC10	MSTPC9	MSTPC8
	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
SUBCKCR	—	—	—	—	—	EXSTP	WAKE32K	CK32K
FCCS	—	—	—	FLER	—	—	—	SCO
FPCS	—	—	—	—	—	—	—	PPVS
FECS	—	—	—	—	—	—	—	EPVB
FKEY	K7	K6	K5	K4	K3	K2	K1	K0
FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0
FTDAR	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0
DPSBYCR	DPSBY	IOKEEP	RAMCUT2	RAMCUT1	—	—	—	RAMCUT0
DPSWCR	—	—	WTSTS5	WTSTS4	WTSTS3	WTSTS2	WTSTS1	WTSTS0
DPSIER	—	—	—	—	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
DPSIFR	DNMIF	—	—	—	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
DPSIEGR	DNMIEG	—	—	—	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG
RSTSR	DPSRSTF	—	—	—	—	LVDF* ⁴	—	PORF* ⁴
LVDCR* ⁴	LVDE	LVDRI	—	LVDMON	—	—	—	—
SEMR_2	—	—	—	—	ABCS	ACS2	ACS1	ACS0
SMR_3* ¹	C/ \bar{A} (GM)	CHR (BLK)	PE (PE)	O/ \bar{E} (O/E)	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_3								
SCR_3* ¹	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_3								

SSR_4*	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_4								
SCMR_4	—	—	—	—	SDIR	SINV	—	SMIF
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	—
ICDRT_0								
ICDRR_0								
ICCRA_1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0
ICCRB_1	BBSY	SCP	SDAO	—	SCLO	—	IICRST	—
ICMR_1	—	WAIT	—	—	BCWP	BC2	BC1	BC0
ICIER_1	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
ICSR_1	TDRE	TEND	RDRF	NACKF	STOP	AL	AAS	ADZ
SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	—
ICDRT_1								
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								

TIER_4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_4	_____							
TGRA_4	_____							
TGRB_4	_____							
TCR_5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_5	—	—	—	—	MD3	MD2	MD1	MD0
TIOR_5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIER_5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_5	_____							
TGRA_5	_____							
TGRB_5	_____							
DTCERA	DTCEA15	DTCEA14	DTCEA13	DTCEA12	DTCEA11	DTCEA10	DTCEA9	DTCEA8
	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0
DTCERB	DTCEB15	—	DTCEB13	DTCEB12	DTCEB11	DTCEB10	DTCEB9	DTCEB8
	DTCEB7	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0
DTCERC	DTCEC15	DTCEC14	DTCEC13	DTCEC12	DTCEC11	DTCEC10	DTCEC9	DTCEC8
	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	—	—

IER	IRQ15E	IRQ14E	IRQ13E	IRQ12E	IRQ11E	IRQ10E	IRQ9E	IRQ8E
	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
ISR	IRQ15F	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	IRQ8F
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
PORT1	P17	P16	P15	P14	P13	P12	P11	P10
PORT2	P27	P26	P25	P24	P23	P22	P21	P20
PORT3	P37	P36	P35	P34	P33	P32	P31	P30
PORT4	P47	P46	P45	P44	—	—	—	—
PORT5	P57	P56	P55	P54	P53	P52	P51	P50
PORT6	P67	P66	P65	P64	P63	P62	P61	P60
PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
PORTC	—	—	PC5	PC4	PC3	PC2	PC1	PC0
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
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	P67DR	P66DR	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	—	—	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR

DADR0								
DADR1								
DACR01	DAOE1	DAOE0	DAE	—	—	—	—	—
PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
NDRH* ²	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
NDRL* ²	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
NDRH* ²	—	—	—	—	NDR11	NDR10	NDR9	NDR8
NDRL* ²	—	—	—	—	NDR3	NDR2	NDR1	NDR0
SMR_0* ¹	C/ \bar{A} (GM)	CHR (BLK)	PE (PE)	O/ \bar{E} (O/ \bar{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/ \bar{A} (GM)	CHR (BLK)	PE (PE)	O/ \bar{E} (O/ \bar{E})	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_1								
SCR_1* ¹	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_1								

ADDRD_0

ADDRE_0

ADDRF_0

ADDRG_0

ADDRH_0

ADCSR_0	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0
ADC_R_0	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	—	EXTRGS
TCSR	OVF	WT/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

TGRA_0

TGRB_0

TGRC_0

TGRD_0

TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
-------	---	-------	-------	-------	-------	-------	-------	-------

TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0
--------	---	---	---	---	-----	-----	-----	-----

TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
--------	------	------	------	------	------	------	------	------

TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
--------	------	---	-------	-------	---	---	-------	-------

TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
-------	------	---	------	------	---	---	------	------

TCNT_1

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

TSR_3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
TCNT_3	_____							
TGRA_3	_____							
TGRB_3	_____							
TGRC_3	_____							
TGRD_3	_____							

- Notes:
- Parts of the bit functions differ in normal mode and the smart card interface.
 - 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 specified, the 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 output groups 0 and 1 by the PCR setting, the NDRL address is H'FFF7D. When different output triggers are specified, the NDRL addresses for pulse output groups 0 and 1 are H'FFF7F and H'FFF7D, respectively.
 When the same output trigger is specified for pulse output groups 6 and 7 by setting, the NDRH address is H'FF63C. When different output triggers are specified, the NDRH addresses for pulse output groups 6 and 7 are H'FF63E and H'FF63C respectively.
 When the same output trigger is specified for pulse output groups 4 and 5 by setting, the NDRL address is H'FF63D. When different output triggers are specified, the NDRL addresses for pulse output groups 4 and 5 are H'FF63F and H'FF63D, respectively.
 - Supported only by the H8SX/1648G Group and the H8SX/1648H Group.
 - Supported only by the H8SX/1648L Group and the H8SX/1648H Group.

TCORA_5	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCORB_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCORB_5	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_5	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCCR_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCCR_5	Initialized	—	—	—	—	Initialized* ¹	Initialized
CRCCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
CRCDIR	Initialized	—	—	—	—	Initialized* ¹	Initialized
CRCDOR	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_7	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCSR_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCSR_7	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCORA_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCORA_7	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCORB_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCORB_7	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_7	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCCR_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCCR_7	Initialized	—	—	—	—	Initialized* ¹	Initialized

ADDRH_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRE_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRF_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRG_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRH_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRA_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRB_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRC_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRD_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADCSR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADCR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADCSR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADCR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
PMDDR* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
PMDR* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
PORTM* ²	—	—	—	—	—	—	—
PMICR* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
PNDDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PNDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PORTN	—	—	—	—	—	—	—
PNICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
SMR_5	Initialized	—	—	—	—	Initialized* ¹	Initialized
BRR_5	Initialized	—	—	—	—	Initialized* ¹	Initialized
SCR_5	Initialized	—	—	—	—	Initialized* ¹	Initialized

BRR_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
SCR_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TDR_6	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
SSR_6	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
RDR_6	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
SCMR_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
SEMR_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
PCR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
PMR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
NDERH_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
NDERL_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
PODRH_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
PODRL_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
NDRH_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
NDRL_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICCRA_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICCRB_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICMR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICIER_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICSR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
SAR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICDRT_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICDRR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized

ICDRR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
BARAH	Initialized	—	—	—	—	Initialized* ¹	Initialized
BARAL	Initialized	—	—	—	—	Initialized* ¹	Initialized
BAMRAH	Initialized	—	—	—	—	Initialized* ¹	Initialized
BAMRAL	Initialized	—	—	—	—	Initialized* ¹	Initialized
BARBH	Initialized	—	—	—	—	Initialized* ¹	Initialized
BARBL	Initialized	—	—	—	—	Initialized* ¹	Initialized
BAMRBH	Initialized	—	—	—	—	Initialized* ¹	Initialized
BAMRBL	Initialized	—	—	—	—	Initialized* ¹	Initialized
BARCH	Initialized	—	—	—	—	Initialized* ¹	Initialized
BARCL	Initialized	—	—	—	—	Initialized* ¹	Initialized
BAMRCH	Initialized	—	—	—	—	Initialized* ¹	Initialized
BAMRCL	Initialized	—	—	—	—	Initialized* ¹	Initialized
BARDH	Initialized	—	—	—	—	Initialized* ¹	Initialized
BARDL	Initialized	—	—	—	—	Initialized* ¹	Initialized
BAMRDH	Initialized	—	—	—	—	Initialized* ¹	Initialized
BAMRDL	Initialized	—	—	—	—	Initialized* ¹	Initialized
BRCRA	Initialized	—	—	—	—	Initialized* ¹	Initialized
BRCRB	Initialized	—	—	—	—	Initialized* ¹	Initialized
BRCRC	Initialized	—	—	—	—	Initialized* ¹	Initialized
BRCRD	Initialized	—	—	—	—	Initialized* ¹	Initialized

TMDR_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIORH_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIORL_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIER_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TSR_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRA_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRB_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRC_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRD_6	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_7	Initialized	—	—	—	—	Initialized* ¹	Initialized
TMDR_7	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIOR_7	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIER_7	Initialized	—	—	—	—	Initialized* ¹	Initialized
TSR_7	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_7	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRA_7	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRB_7	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_8	Initialized	—	—	—	—	Initialized* ¹	Initialized
TMDR_8	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIOR_8	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIER_8	Initialized	—	—	—	—	Initialized* ¹	Initialized
TSR_8	Initialized	—	—	—	—	Initialized* ¹	Initialized

TIER_9	Initialized	—	—	—	—	Initialized* ¹	Initialized
TSR_9	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_9	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRA_9	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRB_9	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRC_9	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRD_9	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_10	Initialized	—	—	—	—	Initialized* ¹	Initialized
TMDR_10	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIOR_10	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIER_10	Initialized	—	—	—	—	Initialized* ¹	Initialized
TSR_10	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_10	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRA_10	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRB_10	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_11	Initialized	—	—	—	—	Initialized* ¹	Initialized
TMDR_11	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIOR_11	Initialized	—	—	—	—	Initialized* ¹	Initialized

P3DDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
P6DDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PADDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PBDDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PCDDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PDDDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PEDDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PFDDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
P1ICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
P2ICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
P3ICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
P4ICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
P5ICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
P6ICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PAICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PBICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PCICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PDICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PEICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PFICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PORTH	—	—	—	—	—	—	—
PORTI	—	—	—	—	—	—	—

PIDDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PJDDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PKDDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PHICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PIICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PJICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PKICR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PDPCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PEPCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PFPCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PHPCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PIPCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PJPCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PKPCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
P2ODR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PFODR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PFCR0	Initialized	—	—	—	—	Initialized* ¹	Initialized
PFCR1	Initialized	—	—	—	—	Initialized* ¹	Initialized
PFCR2	Initialized	—	—	—	—	Initialized* ¹	Initialized
PFCR4	Initialized	—	—	—	—	Initialized* ¹	Initialized
PFCR6	Initialized	—	—	—	—	Initialized* ¹	Initialized
PFCR7	Initialized	—	—	—	—	Initialized* ¹	Initialized
PFCR8* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized

DPSBKR1	Initialized	—	—	—	—	—	Initialized
DPSBKR2	Initialized	—	—	—	—	—	Initialized
DPSBKR3	Initialized	—	—	—	—	—	Initialized
DPSBKR4	Initialized	—	—	—	—	—	Initialized
DPSBKR5	Initialized	—	—	—	—	—	Initialized
DPSBKR6	Initialized	—	—	—	—	—	Initialized
DPSBKR7	Initialized	—	—	—	—	—	Initialized
DPSBKR8	Initialized	—	—	—	—	—	Initialized
DPSBKR9	Initialized	—	—	—	—	—	Initialized
DPSBKR10	Initialized	—	—	—	—	—	Initialized
DPSBKR11	Initialized	—	—	—	—	—	Initialized
DPSBKR12	Initialized	—	—	—	—	—	Initialized
DPSBKR13	Initialized	—	—	—	—	—	Initialized
DPSBKR14	Initialized	—	—	—	—	—	Initialized
DPSBKR15	Initialized	—	—	—	—	—	Initialized
DSAR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
DDAR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
DOFR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
DTCR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
DBSR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
DMDR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
DACR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized

DSAR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
DDAR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
DOFR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
DTCR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
DBSR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
DMDR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
DACR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
DSAR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
DDAR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
DOFR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
DTCR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
DBSR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
DMDR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
DACR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDSAR_0* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDDAR_0* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDOFR_0* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDTCR_0* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDBSR_0* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDMDR_0* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDACR_0* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized

EDSAR_2* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDDAR_2* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDOFR_2* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDTCR_2* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDBSR_2* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDMDR_2* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDACR_2* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDSAR_3* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDDAR_3* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDOFR_3* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDTCR_3* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDBSR_3* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDMDR_3* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
EDACR_3* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
CLSBR0* ²	—	—	—	—	—	—	—
CLSBR1* ²	—	—	—	—	—	—	—
CLSBR2* ²	—	—	—	—	—	—	—
CLSBR3* ²	—	—	—	—	—	—	—
CLSBR4* ²	—	—	—	—	—	—	—
CLSBR5* ²	—	—	—	—	—	—	—
CLSBR6* ²	—	—	—	—	—	—	—
CLSBR7* ²	—	—	—	—	—	—	—

IPRD	Initialized	—	—	—	—	Initialized* ¹	Initialized
IPRE	Initialized	—	—	—	—	Initialized* ¹	Initialized
IPRF	Initialized	—	—	—	—	Initialized* ¹	Initialized
IPRG	Initialized	—	—	—	—	Initialized* ¹	Initialized
IPRH	Initialized	—	—	—	—	Initialized* ¹	Initialized
IPRI	Initialized	—	—	—	—	Initialized* ¹	Initialized
IPRJ* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
IPRK	Initialized	—	—	—	—	Initialized* ¹	Initialized
IPRL	Initialized	—	—	—	—	Initialized* ¹	Initialized
IPRM	Initialized	—	—	—	—	Initialized* ¹	Initialized
IPRN	Initialized	—	—	—	—	Initialized* ¹	Initialized
IPRO	Initialized	—	—	—	—	Initialized* ¹	Initialized
IPRQ	Initialized	—	—	—	—	Initialized* ¹	Initialized
IPRR	Initialized	—	—	—	—	Initialized* ¹	Initialized
ISCRH	Initialized	—	—	—	—	Initialized* ¹	Initialized
ISCLR	Initialized	—	—	—	—	Initialized* ¹	Initialized
DTCVBR	Initialized	—	—	—	—	Initialized* ¹	Initialized
ABWCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
ASTCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
WTCRA	Initialized	—	—	—	—	Initialized* ¹	Initialized
WTCRB	Initialized	—	—	—	—	Initialized* ¹	Initialized
RDNCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
CSACR	Initialized	—	—	—	—	Initialized* ¹	Initialized

RAMER	Initialized	—	—	—	—	Initialized* ¹	Initialized
DRAMCR* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
DRACCR* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
SDCR* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
REFCR* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
RTCNT* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
RTCOR* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
MDCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
SYSCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
SCKCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
SBYCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
MSTPCRA	Initialized	—	—	—	—	Initialized* ¹	Initialized
MSTPCRB	Initialized	—	—	—	—	Initialized* ¹	Initialized
MSTPCRC	Initialized	—	—	—	—	Initialized* ¹	Initialized
SUBCKCR* ²	Initialized	—	—	—	—	Initialized* ¹	Initialized
FCCS	Initialized	—	—	—	—	Initialized* ¹	Initialized
FPCS	Initialized	—	—	—	—	Initialized* ¹	Initialized
FECS	Initialized	—	—	—	—	Initialized* ¹	Initialized
FKEY	Initialized	—	—	—	—	Initialized* ¹	Initialized
FMATS	Initialized	—	—	—	—	Initialized* ¹	Initialized
FTDAR	Initialized	—	—	—	—	Initialized* ¹	Initialized

SEMR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
SMR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
BRR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
SCR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TDR_3	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
SSR_3	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
RDR_3	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
SCMR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
SMR_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
BRR_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
SCR_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
TDR_4	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
SSR_4	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
RDR_4	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
SCMR_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICCRA_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICCRB_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICMR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICIER_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICSR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
SAR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICDRT_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ICDRR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized

ICDRR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCSR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCSR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCORA_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCORA_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCORB_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCORB_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCCR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCCR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
TMDR_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIOR_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIER_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
TSR_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRA_4	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRB_4	Initialized	—	—	—	—	Initialized* ¹	Initialized

TGRB_5	Initialized	—	—	—	—	Initialized* ¹	Initialized
DTCERA	Initialized	—	—	—	—	Initialized* ¹	Initialized
DTCERB	Initialized	—	—	—	—	Initialized* ¹	Initialized
DTCERC	Initialized	—	—	—	—	Initialized* ¹	Initialized
DTCERD	Initialized	—	—	—	—	Initialized* ¹	Initialized
DTCERE	Initialized	—	—	—	—	Initialized* ¹	Initialized
DTCERF	Initialized	—	—	—	—	Initialized* ¹	Initialized
DTCCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
INTCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
CPUPCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
IER	Initialized	—	—	—	—	Initialized* ¹	Initialized
ISR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PORT1	—	—	—	—	—	—	—
PORT2	—	—	—	—	—	—	—
PORT3	—	—	—	—	—	—	—
PORT4	—	—	—	—	—	—	—
PORT5	—	—	—	—	—	—	—
PORT6	—	—	—	—	—	—	—
PORTA	—	—	—	—	—	—	—
PORTB	—	—	—	—	—	—	—
PORTC	—	—	—	—	—	—	—
PORTD	—	—	—	—	—	—	—

PBDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PCDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PDDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PEDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PFDR	Initialized	—	—	—	—	Initialized* ¹	Initialized
SMR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
BRR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
SCR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
TDR_2	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
SSR_2	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
RDR_2	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
SCMR_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
DADR0	Initialized	—	—	—	—	Initialized* ¹	Initialized
DADR1	Initialized	—	—	—	—	Initialized* ¹	Initialized
DACR01	Initialized	—	—	—	—	Initialized* ¹	Initialized
PCR	Initialized	—	—	—	—	Initialized* ¹	Initialized
PMR	Initialized	—	—	—	—	Initialized* ¹	Initialized
NDERH	Initialized	—	—	—	—	Initialized* ¹	Initialized
NDERL	Initialized	—	—	—	—	Initialized* ¹	Initialized
PODRH	Initialized	—	—	—	—	Initialized* ¹	Initialized
PODRL	Initialized	—	—	—	—	Initialized* ¹	Initialized
NDRH	Initialized	—	—	—	—	Initialized* ¹	Initialized
NDRL	Initialized	—	—	—	—	Initialized* ¹	Initialized

SMR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
BRR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
SCR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
TDR_1	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
SSR_1	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
RDR_1	Initialized	Initialized	—	Initialized	Initialized	Initialized* ¹	Initialized
SCMR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRA_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRB_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRC_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRD_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRE_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRF_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRG_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADDRH_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADCSR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
ADCR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCSR	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT	Initialized	—	—	—	—	Initialized* ¹	Initialized
RSTCSR	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCSR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCSR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized

TCCR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
TSTR	Initialized	—	—	—	—	Initialized* ¹	Initialized
TSYR	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TMDR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIORH_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIORL_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIER_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TSR_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRA_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRB_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRC_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRD_0	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
TMDR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIOR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIER_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
TSR_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRA_1	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRB_1	Initialized	—	—	—	—	Initialized* ¹	Initialized

TGRB_2	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TMDR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIORH_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIORL_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TIER_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TSR_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TCNT_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRA_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRB_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRC_3	Initialized	—	—	—	—	Initialized* ¹	Initialized
TGRD_3	Initialized	—	—	—	—	Initialized* ¹	Initialized

- Notes:
1. Not initialized in deep software standby mode but initialized by the internal reset when deep software standby mode is released.
 2. Supported only by the H8SX/1648G Group and the H8SX/1648H Group.
 3. Supported only by the H8SX/1648L Group and the H8SX/1648H Group.
 4. LVDCR is initialized by a pin reset or power-on reset not by a voltage-monitoring reset, deep software standby reset, or watchdog timer reset

Input voltage (Port N)	V_{in}	-0.3 to $V_{CC} + 6.0$
Input voltage (port 4, 5)	V_{in}	-0.3 to $+0.3$
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 exceeded.

Note: * The operating temperature range during programming/erasing of the flash memory is 0°C to $+75^{\circ}\text{C}$ for regular specifications and 0°C to $+85^{\circ}\text{C}$ for wide-range specifications.

Item		Symbol	Min.	Typ.	Max.	Unit	Conc
Schmitt trigger input voltage	$\overline{\text{IRQ}}$ input pin,	V_T^-	$V_{CC} \times 0.2$	—	—	V	
	TPU input pin,	V_T^+	—	—	$V_{CC} \times 0.7$	V	
	TMR input pin, IIC2 input pin, port 2, port 3, port J, port K	$V_T^+ - V_T^-$	$V_{CC} \times 0.06$	—	—	V	
	$\overline{\text{IRQ0-B}}$ to $\overline{\text{IRQ7-B}}$ input pins	V_T^-	$AV_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$AV_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	$AV_{CC} \times 0.06$	—	—	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		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$		
	Other input pins						
	Port N		$V_{CC} \times 0.7$	—	5.5		
	Port 4, 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	$\overline{\text{RES}}$	$ I_{in} $	—	—	10.0	μA	$V_{in} =$
	MD, $\overline{\text{STBY}}$, EMLE, NMI		—	—	1.0		$V_{CC} =$
	Port 4, 5		—	—	1.0		$V_{in} =$ AV_{CC}

leakage
current
(off state)

Input pull-up MOS current	Ports D to F, H, I		$-I_p$	10	—	300	μA	V 3 V	
Input capacitance	All input pins		C_{in}	—	—	15	pF	V f T	
Current consumption* ²	Normal operation	H8SX/1648 Group, H8SX/1648A Group	I_{cc} * ⁴	—	50	80	mA	f	
		H8SX/1648G Group		—	50	85			
	Sleep mode	H8SX/1648 Group, H8SX/1648A Group	I_{cc} * ⁴	—	45	52	mA	f	
		H8SX/1648G Group		—	48	60			
	Subclock operation	H8SX/1648G Group	I_{cc} * ⁴	—	5	10	mA	3 o r u	
	Standby mode	Software standby mode* ³		I_{cc} * ⁴	—	0.15	1.1	mA	T 5
		Deep software standby mode		I_{cc} * ⁴	—	20	60		
RAM power supply halted		RAM TM32K Halted* ⁷	I_{cc} * ⁴	—	3	8	μA	T 5	
RAM power supply halted		RAM TM32K Operation* ⁷	I_{cc} * ⁴	—	9	16			μA

Standby for A/D and D/A conversion — 0.5 1.0 μA

RAM standby voltage	V_{RAM}	2.5	—	—	V
Vcc start voltage* ⁶	V_{CCSTART}	—	—	0.8	V
Vcc rising gradient* ⁶	SV_{CC}	—	—	20	ms/V

- Notes:
- When the A/D and D/A converters are not used, the AV_{CC} , V_{ref} , and AV_{SS} pins should be open. Connect the AV_{CC} and V_{ref} pins to V_{CC} , and the AV_{SS} pin to V_{SS} .
 - Current consumption values are for $V_{\text{IHmin}} = V_{\text{CC}} - 0.5 \text{ V}$ and $V_{\text{ILmax}} = 0.5 \text{ V}$ with output pins unloaded and all input pull-up MOSs in the off state.
 - The values are for $V_{\text{RAM}} \leq V_{\text{CC}} < 3.0 \text{ V}$, $V_{\text{IHmin}} = V_{\text{CC}} \times 0.9$, and $V_{\text{ILmax}} = 0.3 \text{ V}$.
 - I_{CC} depends on f as follows:
 - H8SX/1648 Group and H8SX/1648A Group
 - $I_{\text{CCmax}} = 25 \text{ (mA)} + 1.1 \text{ (mA/MHz)} \times f \text{ (normal operation)}$
 - $I_{\text{CCmax}} = 27 \text{ (mA)} + 0.5 \text{ (mA/MHz)} \times f \text{ (sleep mode)}$
 - H8SX/1648G Group
 - $I_{\text{CCmax}} = 30 \text{ (mA)} + 1.1 \text{ (mA/MHz)} \times f \text{ (normal operation)}$
 - $I_{\text{CCmax}} = 35 \text{ (mA)} + 0.5 \text{ (mA/MHz)} \times f \text{ (sleep mode)}$
 - The values are for reference.
 - This can be applied when the $\overline{\text{RES}}$ pin is held low at power-on.
 - TM32K is supported only in the H8SX/1648G Group.
 - Supported only in the H8SX/1648G Group.

current (per pin)					
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 be open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

Schmitt trigger input voltage	$\overline{\text{IRQ}}$ input pin,	V_T^-	$V_{CC} \times 0.2$	—	—	V	
	TPU input pin,	V_T^+	—	—	$V_{CC} \times 0.7$	V	
	TMR input pin,	$V_T^+ - V_T^-$	$V_{CC} \times 0.06$	—	—	V	
	IIC2 input pin, port 2, port 3, port J, port K						
	$\overline{\text{IRQ0-B}}$ to $\overline{\text{IRQ7-B}}$ input pins	V_T^-	$AV_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$AV_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	$AV_{CC} \times 0.06$	—	—	V	
Input high voltage (except Schmitt trigger input pin)	$\overline{\text{MD}}$, $\overline{\text{RES}}$, $\overline{\text{STBY}}$, $\overline{\text{EMLE}}$, $\overline{\text{NMI}}$	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$		
	Other input pins						
	Port N		$V_{CC} \times 0.7$	—	5.5		
Input low voltage (except Schmitt trigger input pin)	$\overline{\text{MD}}$, $\overline{\text{RES}}$, $\overline{\text{STBY}}$, $\overline{\text{EMLE}}$	V_{IL}	-0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL, $\overline{\text{NMI}}$		-0.3	—	$V_{CC} \times 0.2$		
	Other input pins		-0.3	—	$V_{CC} \times 0.2$		
	Port 4, 5						
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	$\overline{\text{RES}}$	$ I_{in} $	—	—	10.0	μA	$V_{in} =$ V_{CC}
	$\overline{\text{MD}}$, $\overline{\text{STBY}}$, $\overline{\text{EMLE}}$, $\overline{\text{NMI}}$		—	—	1.0		
	Port 4, 5		—	—	1.0		$V_{in} =$ AV_{CC}

leakage current (off state)														
Input pull-up MOS current	Ports D to F, H, I		$-I_p$	10	—	300	μA	V						
Input capacitance	All input pins		C_{in}	—	—	15	pF	V						
Current consumption*2	Normal operation	H8SX/1648L Group	I_{cc}^{*4}	—	50	80	mA	f						
		H8SX/1648H Group			50	85								
	Sleep mode	H8SX/1648L Group		—	45	52								
		H8SX/1648H Group			48	60								
	Subclock operation	H8SX/1648H Group			—	5	10		3					
	Standby mode	Software standby mode*3			—	0.15	1.1	mA	T					
						—	—			3.5	5			
		Deep software standby mode	RAM retained*3				—	24	67	μA	T			
								—	—			200	5	
power supply halted			RAM TM32K						—	23	35	μA	T	
			Halted*7							—	—			60
		TM32K Operation*7			—	29	43	T						
						—	—		75	5				

Standby for A/D and D/A conversion	—	0.5	1.0	μA
power supply current				
RAM standby voltage	V_{RAM}	2.5	—	V
Vcc rising gradient*6	SV_{CC}	—	—	20 ms/V

- Notes:
- When the A/D and D/A converters are not used, the AV_{CC} , V_{ref} , and AV_{SS} pins should be open. Connect the AV_{CC} and V_{ref} pins to V_{CC} , and the AV_{SS} pin to V_{SS} .
 - Current consumption values are for $V_{\text{IH min}} = V_{\text{CC}} - 0.5 \text{ V}$ and $V_{\text{IL max}} = 0.5 \text{ V}$ with output pins unloaded and all input pull-up MOSs in the off state.
 - The values are for $V_{\text{RAM}} \leq V_{\text{CC}} < 3.0 \text{ V}$, $V_{\text{IH min}} = V_{\text{CC}} \times 0.9$, and $V_{\text{IL max}} = 0.3 \text{ V}$.
 - I_{CC} depends on f as follows:
 - H8SX/1648L Group
 - $I_{\text{CC max}} = 25 \text{ (mA)} + 1.1 \text{ (mA/MHz)} \times f \text{ (normal operation)}$
 - $I_{\text{CC max}} = 27 \text{ (mA)} + 0.5 \text{ (mA/MHz)} \times f \text{ (sleep mode)}$
 - H8SX/1648H Group
 - $I_{\text{CC max}} = 30 \text{ (mA)} + 1.1 \text{ (mA/MHz)} \times f \text{ (normal operation)}$
 - $I_{\text{CC max}} = 35 \text{ (mA)} + 0.5 \text{ (mA/MHz)} \times f \text{ (sleep mode)}$
 - The values are for reference.
 - This can be applied at power-on.
 - TM32K is supported only in the H8SX/1648H Group.
 - Supported only in the H8SX/1648H Group.

current (per pin)					
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 be open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

Note: * $V_{CC} = 2.95\text{ V to }3.60\text{ V}$ in the H8SX/1648L Group and H8SX/1648H Group.

Figure 29.1 Output Load Circuit

29.4.1 Clock Timing

Table 29.6 Clock Timing

Conditions: $V_{CC} = PLLV_{CC} = 3.0\text{ V to }3.6\text{ V}^*$, $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$, $V_{ref} = 3.0\text{ V to }AV_{CC}$,
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$, $I_{\phi} = 8\text{ MHz to }50\text{ MHz}$,
 $B\phi = 8\text{ MHz to }50\text{ MHz}$, $P\phi = 8\text{ MHz to }35\text{ MHz}$,
 $T_a = -20^{\circ}\text{C to }+75^{\circ}\text{C}$ (regular specifications),
 $T_a = -40^{\circ}\text{C to }+85^{\circ}\text{C}$ (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 oscillator oscillation frequency	f_{sub}	32.768	kHz
Subclock cycle time	t_{sub}	30.5	μ s

Note: * $V_{CC}=PLL=2.95V$ to $3.60V$ in the H8SX/1648L Group and H8SX/1648H Group

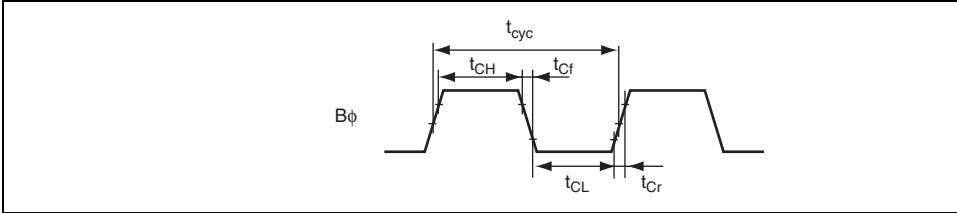


Figure 29.2 External Bus Clock Timing

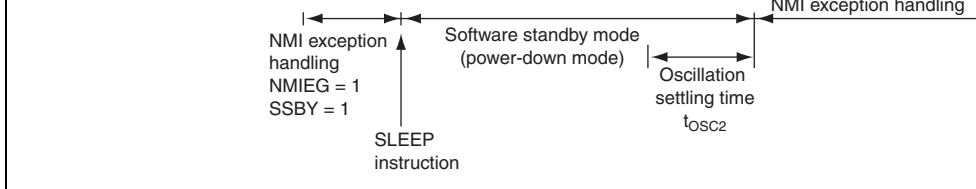


Figure 29.3 Oscillation Settling Timing after Software Standby Mode

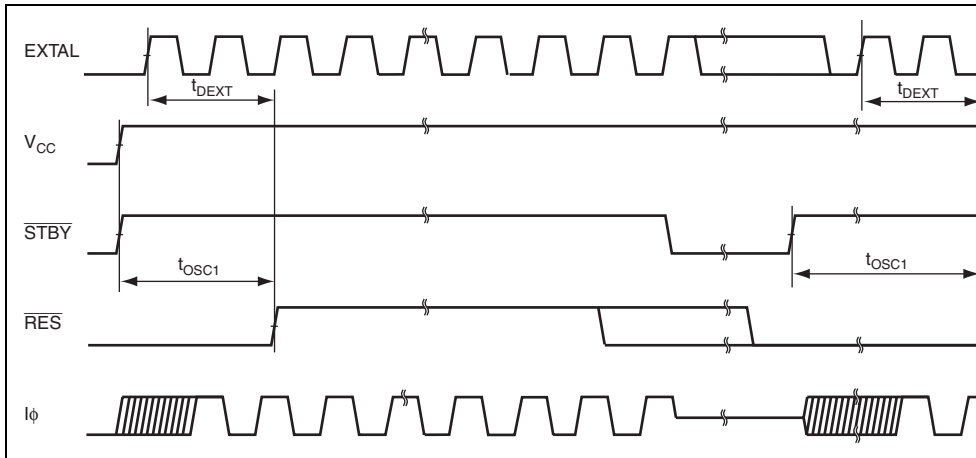


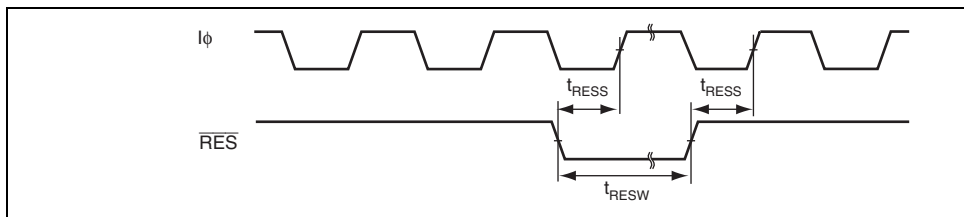
Figure 29.4 Oscillation Settling Timing

Table 29.7 Control Signal Timing

Conditions: $V_{CC} = PLLV_{CC} = 3.0\text{ V to }3.6\text{ V}^*$, $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$, $V_{ref} = 3.0\text{ V to }AV_{CC}$
 $PLLV_{SS} = AV_{SS} = 0\text{ V}$, $I\phi = 8\text{ MHz to }50\text{ MHz}$,
 $T_a = -20^\circ\text{C to }+75^\circ\text{C}$ (regular specifications),
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$ (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Co
$\overline{\text{RES}}$ setup time	t_{RESS}	200	—	ns	Figure 29.6
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 29.6
NMI hold time	t_{NMIH}	10	—	ns	
NMI pulse width (after leaving software standby mode)	t_{NMIW}	200	—	ns	
$\overline{\text{IRQ}}$ setup time	t_{IRQS}	150	—	ns	
$\overline{\text{IRQ}}$ hold time	t_{IROH}	10	—	ns	
$\overline{\text{IRQ}}$ pulse width (after leaving software standby mode)	t_{IRQW}	200	—	ns	

Note: * $V_{CC}=PLLV_{CC}=2.95\text{ V to }3.60\text{ V}$ in the H8SX/1648L Group and H8SX/1648H Group

**Figure 29.6 Reset Input Timing**

Note: * SSIER must be set to cancel software standby mode.

Figure 29.7 Interrupt Input Timing

29.4.3 Bus Timing

Table 29.8 Bus Timing (1)

Conditions: $V_{CC} = PLLV_{CC} = 3.0\text{ V to }3.6\text{ V}^*$, $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$, $V_{ref} = 3.0\text{ V to }AV_{CC}$,
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$, $B\phi = 8\text{ MHz to }50\text{ MHz}$,
 $T_a = -20^\circ\text{C to }+75^\circ\text{C}$ (regular specifications),
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$ (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Condition
Address delay time	t_{AD}	—	15	ns	Figure 29.36
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}	0.0	—	ns
Read data access time 2	t_{AC2}	—	$1.5 \times t_{cyc} - 20$	ns
Read data access time 4	t_{AC4}	—	$2.5 \times t_{cyc} - 20$	ns
Read data access time 5	t_{AC5}	—	$1.0 \times t_{cyc} - 20$	ns
Read data access time 6	t_{AC6}	—	$2.0 \times t_{cyc} - 20$	ns
Read data access time (from address) 1	t_{AA1}	—	$1.0 \times t_{cyc} - 20$	ns
Read data access time (from address) 2	t_{AA2}	—	$1.5 \times t_{cyc} - 20$	ns
Read data access time (from address) 3	t_{AA3}	—	$2.0 \times t_{cyc} - 20$	ns
Read data access time (from address) 4	t_{AA4}	—	$2.5 \times t_{cyc} - 20$	ns
Read data access time (from address) 5	t_{AA5}	—	$3.0 \times t_{cyc} - 20$	ns

WR pulse width 1	t_{WSW1}	$1.5 \times t_{cyc} - 13$	—	ns	
WR pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 13$	—	ns	
Write data delay time	t_{WDD}	—	20	ns	
Write data setup time 1	t_{WDS1}	$0.5 \times t_{cyc} - 13$	—	ns	
Write data setup time 2	t_{WDS2}	$1.0 \times t_{cyc} - 13$	—	ns	
Write data setup time 3	t_{WDS3}	$1.5 \times t_{cyc} - 13$	—	ns	
Write data hold time 1	t_{WDH1}	$0.5 \times t_{cyc} - 8$	—	ns	
Write data hold time 3	t_{WDH3}	$1.5 \times t_{cyc} - 8$	—	ns	
Byte control delay time	t_{UBD}	—	15	ns	Figure 29.14
Byte control pulse width 1	t_{UBW1}	—	$1.0 \times t_{cyc} - 15$	ns	Figure 29.14
Byte control pulse width 2	t_{UBW2}	—	$2.0 \times t_{cyc} - 15$	ns	Figure 29.14
Multiplexed address delay time 1	t_{MAD1}	—	15	ns	Figure 29.18
Multiplexed address hold time	t_{MAH}	$1.0 \times t_{cyc} - 15$	—	ns	Figure 29.18
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	
WAIT setup time	t_{WTS}	15	—	ns	Figure 29.18
WAIT hold time	t_{WTH}	5.0	—	ns	Figure 29.18
BREQ setup time	t_{BREOS}	20	—	ns	Figure 29.14
BACK delay time	t_{BACD}	—	15	ns	
Bus floating time	t_{BZD}	—	30	ns	
BREQO delay time	t_{BRQOD}	—	15	ns	Figure 29.14
BS delay time	t_{BSD}	1.0	15	ns	Figure 29.9, 29.14
RD/WR delay time	t_{RWD}	—	15	ns	Figure 29.14

Read data access time 1	t_{AC1}	—	1.0×tcyc-20	ns
Read data access time 3	t_{AC3}	—	2.0×tcyc-20	ns
Read data access time 7	t_{AC7}	—	4.0×tcyc-20	ns
Read data access time 8	t_{AC8}	—	3.0×tcyc-20	ns
Write data hold time 2	t_{WDH2}	1.0×tcyc-8	—	ns
Read command setup time 1	t_{RCS1}	1.5×tcyc-10	—	ns
Read command setup time 2	t_{RCS2}	2.0×tcyc-10	—	ns
Read command hold time	t_{RCH}	0.5×tcyc-10	—	ns
Write command setup time 1	t_{WCS1}	0.5×tcyc-10	—	ns
Write command setup time 2	t_{WCS2}	1.0×tcyc-10	—	ns
Write command hold time 1	t_{WCH1}	0.5×tcyc-10	—	ns
Write command hold time 2	t_{WCH2}	1.0×tcyc-10	—	ns
\overline{CAS} delay time 1	t_{CASD1}	—	15	ns
\overline{CAS} delay time 2	t_{CASD2}	—	15	ns
\overline{CAS} setup time 1	t_{CSR1}	0.5×tcyc-10	—	ns
\overline{CAS} setup time 2	t_{CSR2}	1.5×tcyc-10	—	ns
\overline{CAS} pulse width 1	t_{CASW1}	1.0×tcyc-15	—	ns
\overline{CAS} pulse width 2	t_{CASW2}	1.5×tcyc-15	—	ns
\overline{CAS} precharge time 1	t_{CPW1}	1.0×tcyc-15	—	ns
\overline{CAS} precharge time 2	t_{CPW2}	1.5×tcyc-15	—	ns
\overline{OE} delay time 1	t_{OED1}	—	15	ns
\overline{OE} delay time 2	t_{OED2}	—	15	ns
Precharge time 1	t_{PCH1}	1.0×tcyc-20	—	ns
Precharge time 2	t_{PCH2}	1.5×tcyc-20	—	ns

Table 29.8 Bus Timing (4)

Conditions: $V_{CC} = PLLV_{CC} = 3.0\text{ V to }3.6\text{ V}^*$, $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$, $V_{ref} = 3.0\text{ V to }AV_{CC}$
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$, $B\phi = 8\text{ MHz to }50\text{ MHz}$,
 $T_a = -20^\circ\text{C to }+75^\circ\text{C}$ (regular specifications),
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$ (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Condition
Read data setup time 3	t_{RDS3}	12	—	ns	Figure 29.34
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	

Note: * $V_{CC} = PLLV_{CC} = 2.95\text{ V to }3.6\text{ V}$ in the H8SX/1648L Group and H8SX/1648H Group

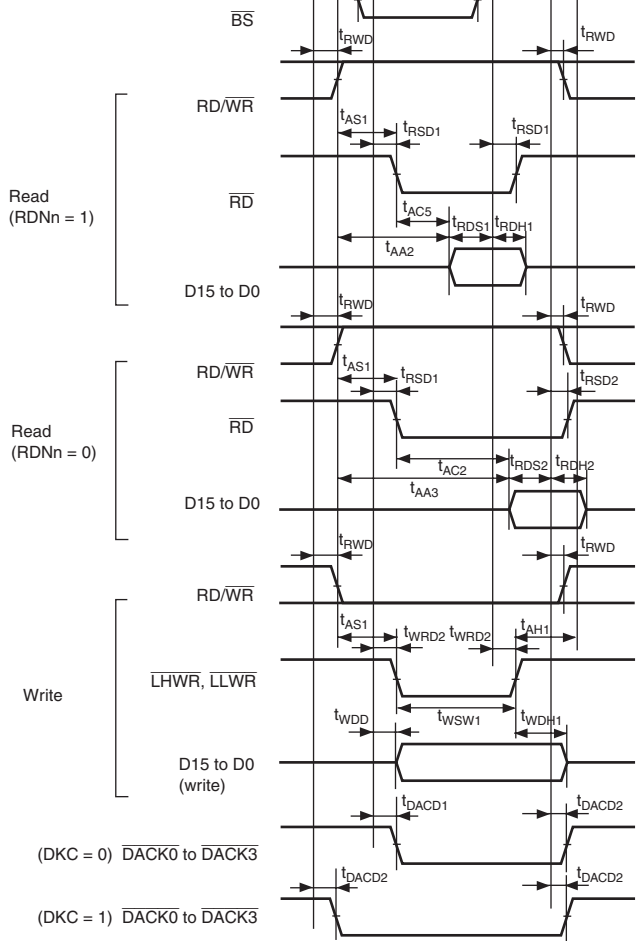


Figure 29.8 Basic Bus Timing: Two-State Access

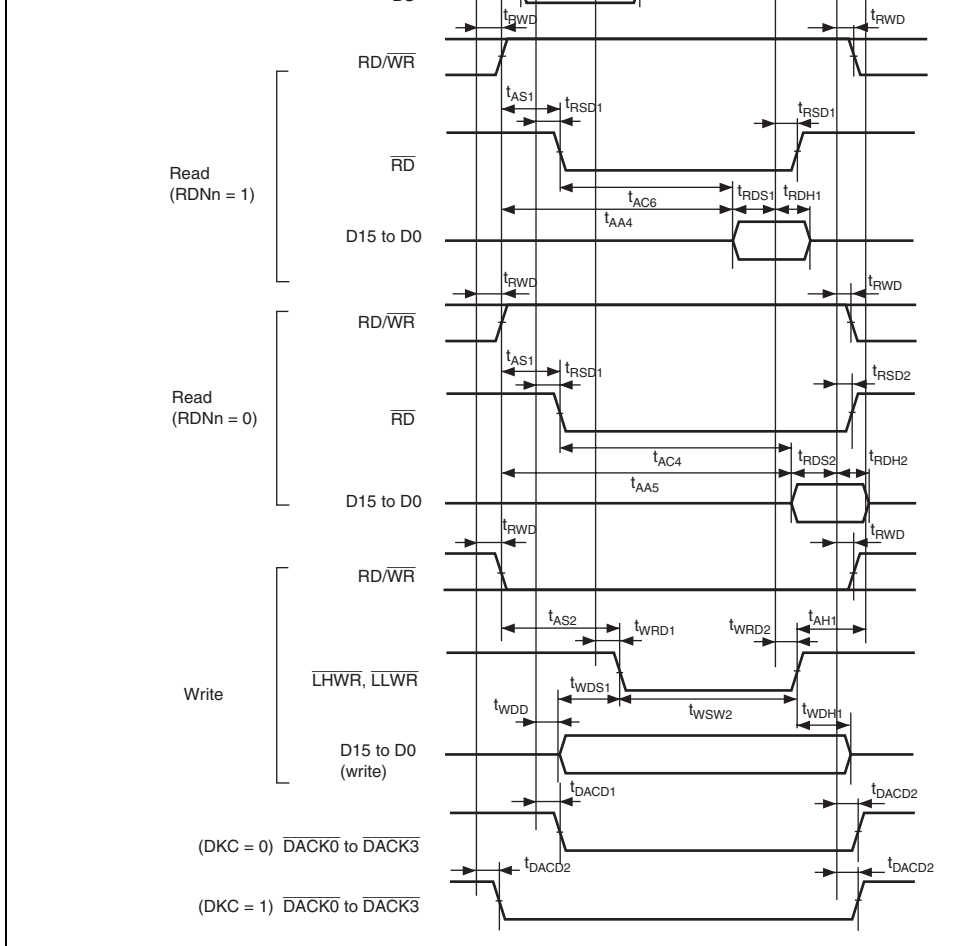


Figure 29.9 Basic Bus Timing: Three-State Access

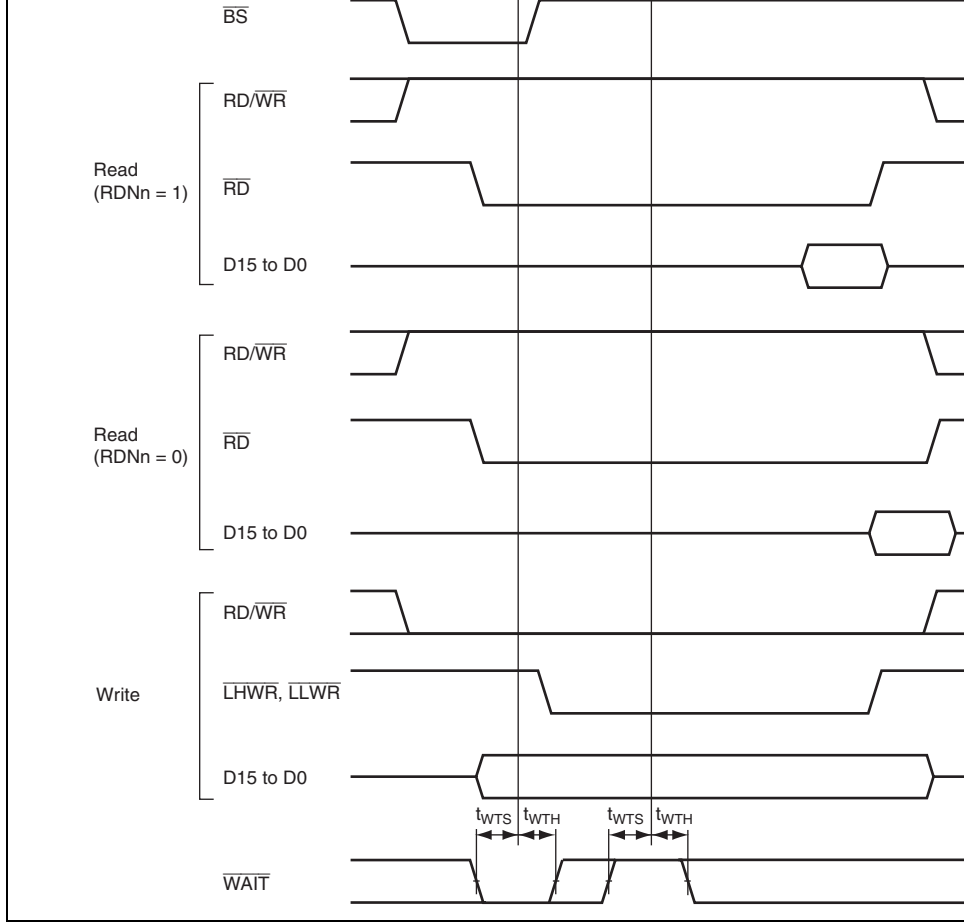


Figure 29.10 Basic Bus Timing: Three-State Access, One Wait

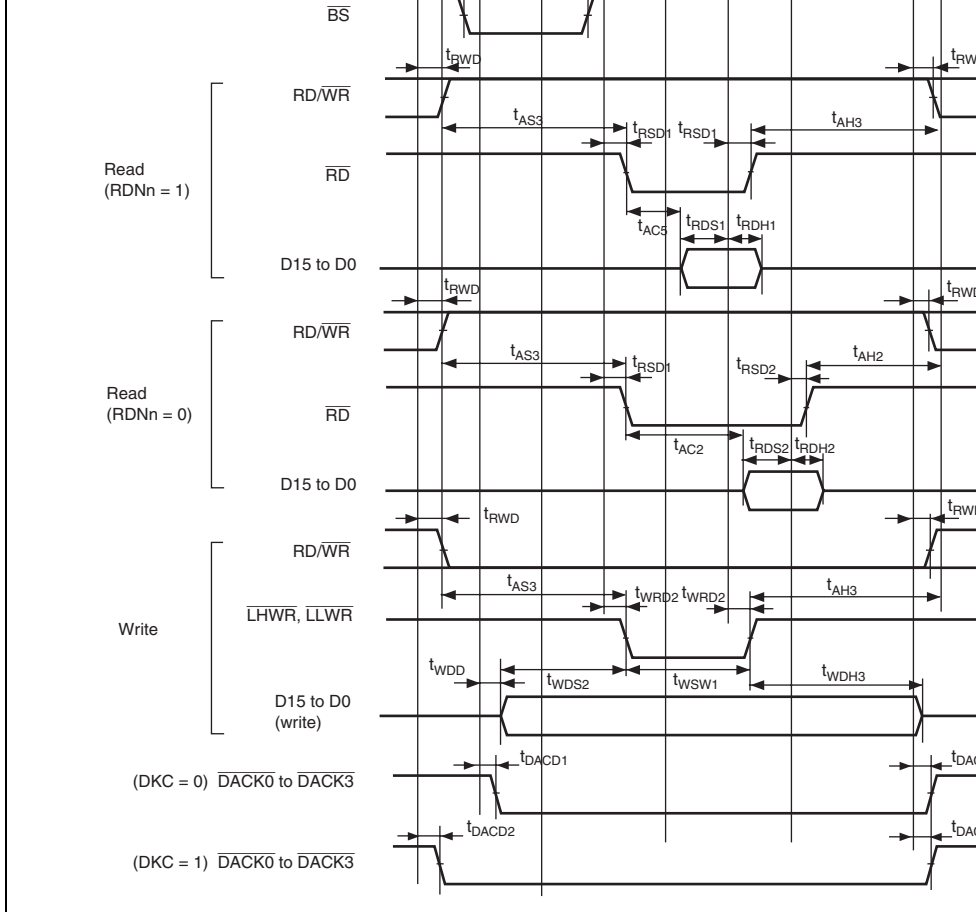


Figure 29.11 Basic Bus Timing: Two-State Access (\overline{CS} Assertion Period Extension)

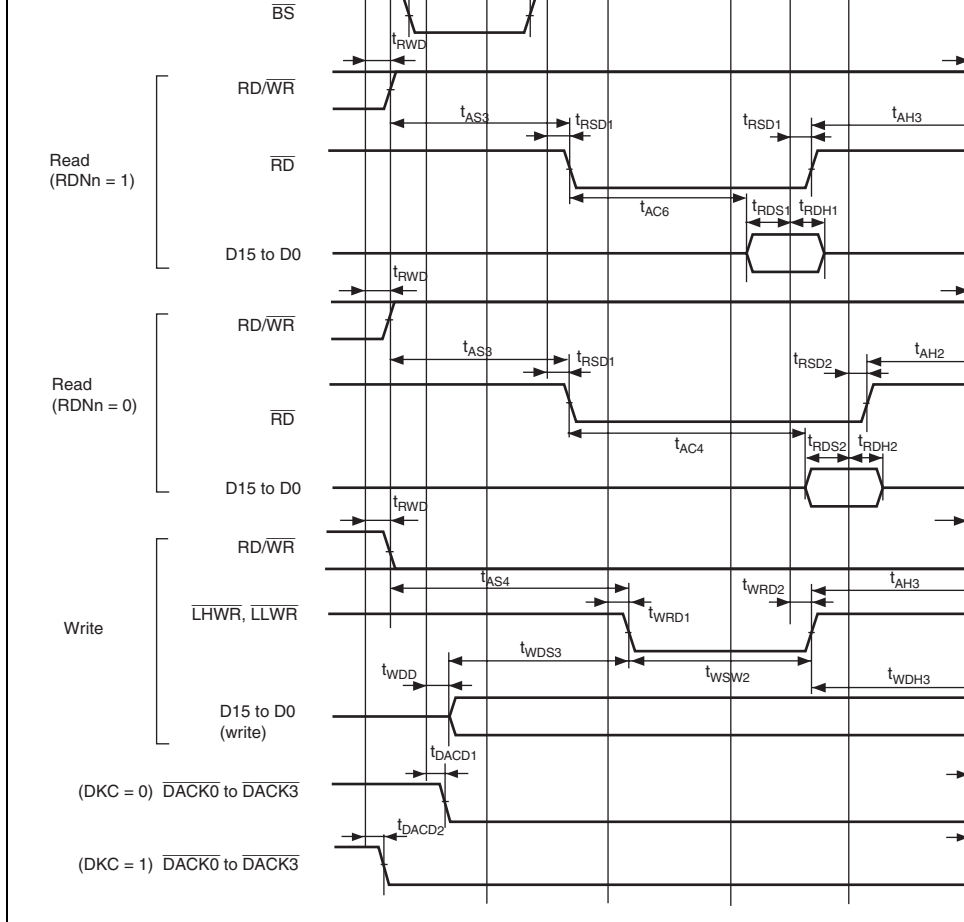


Figure 29.12 Basic Bus Timing: Three-State Access (\overline{CS} Assertion Period Extension)

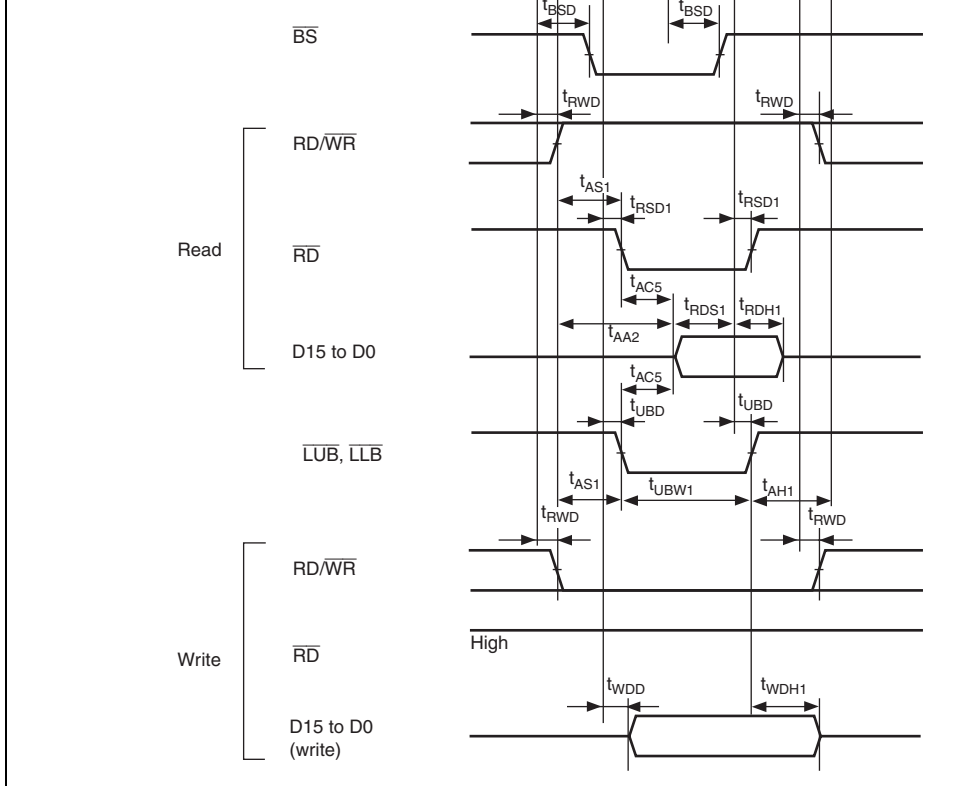


Figure 29.13 Byte Control SRAM: Two-State Read/Write Access

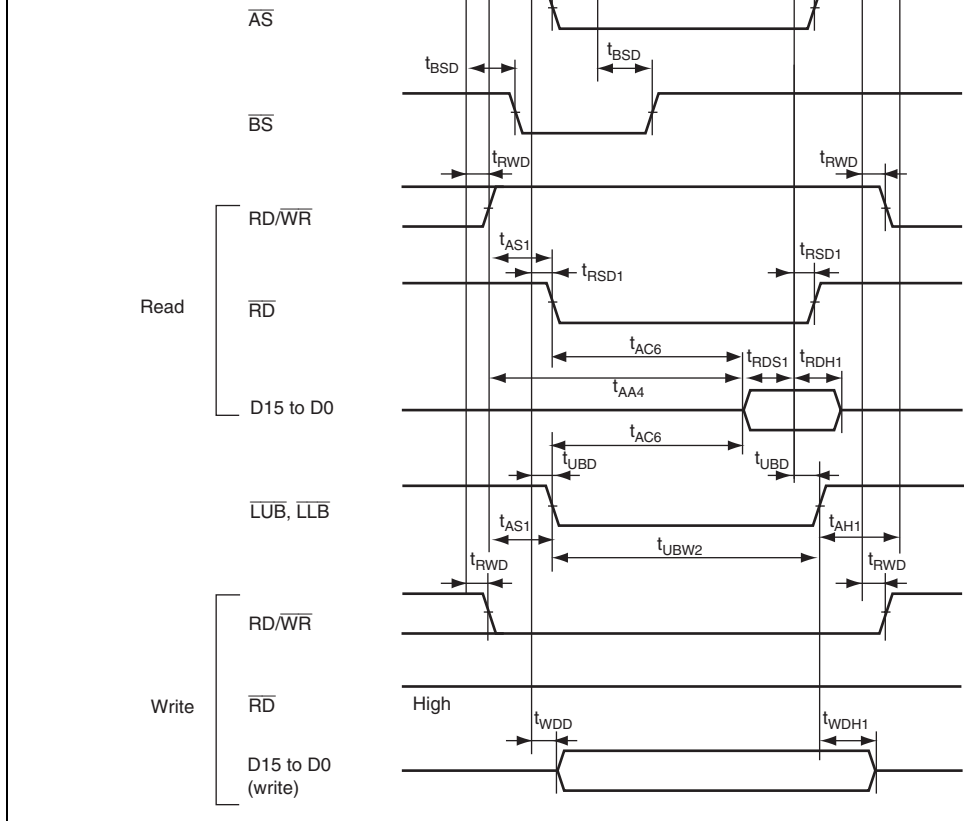


Figure 29.14 Byte Control SRAM: Three-State Read/Write Access

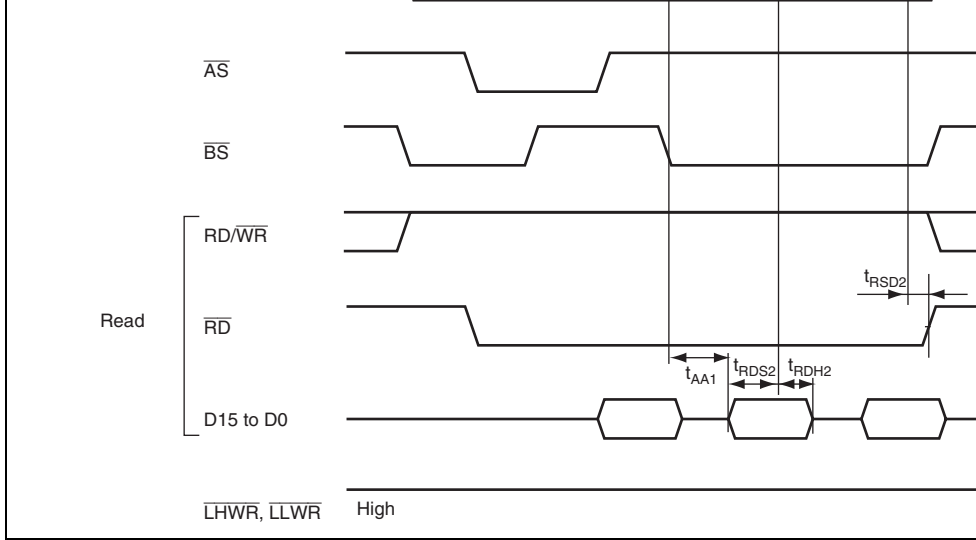


Figure 29.15 Burst ROM Access Timing: One-State Burst Access

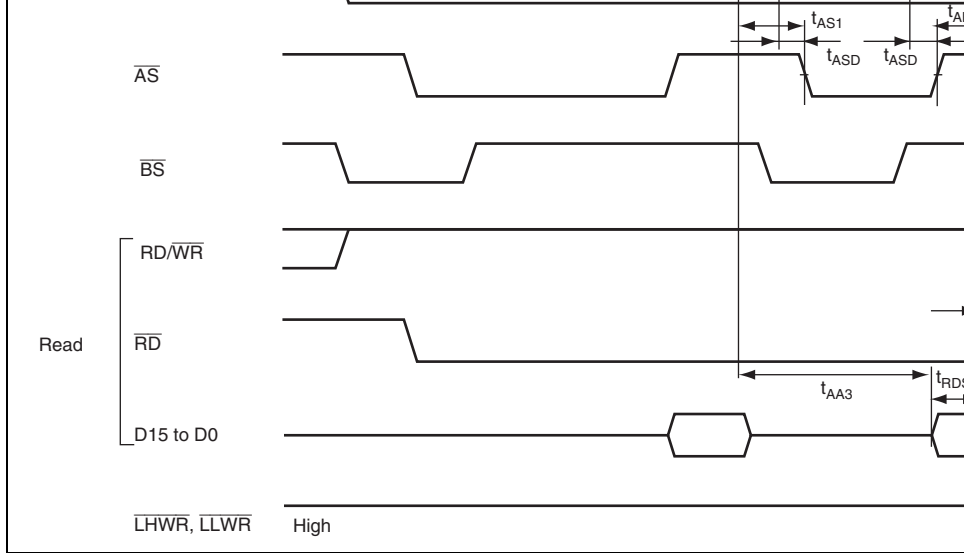
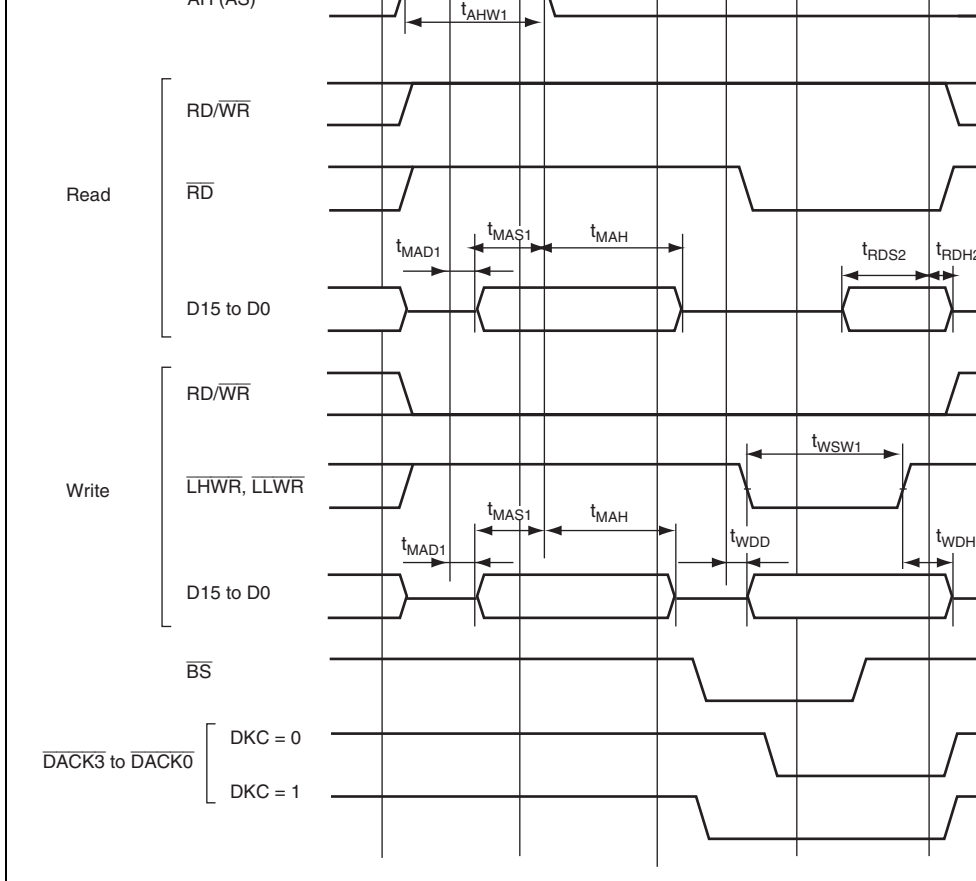


Figure 29.16 Burst ROM Access Timing: Two-State Burst Access



**Figure 29.17 Address/Data Multiplexed Access Timing (No Wait)
(Basic, Four-State Access)**

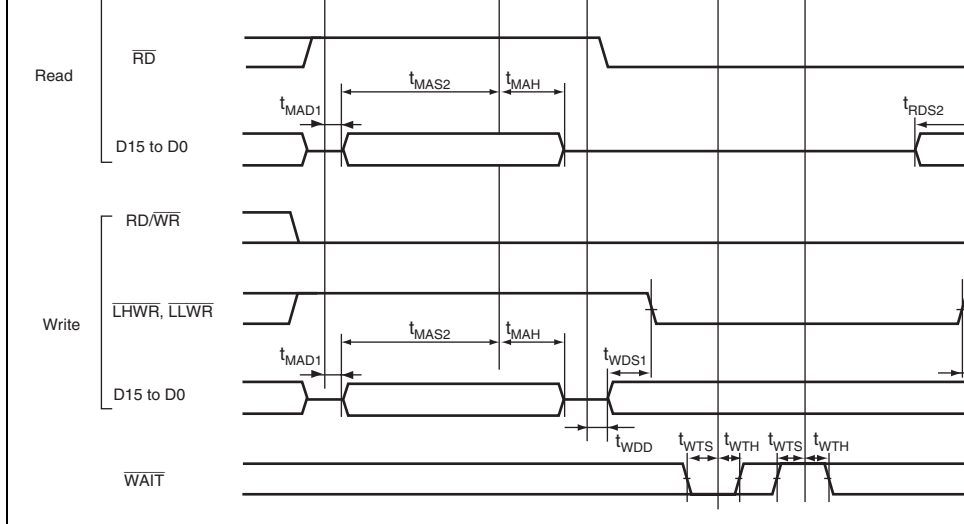


Figure 29.18 Address/Data Multiplexed Access Timing (Wait Control)
(Address Cycle Program Wait × 1 + Data Cycle Program Wait × 1 + Data Cycle Pin Wait × 1)

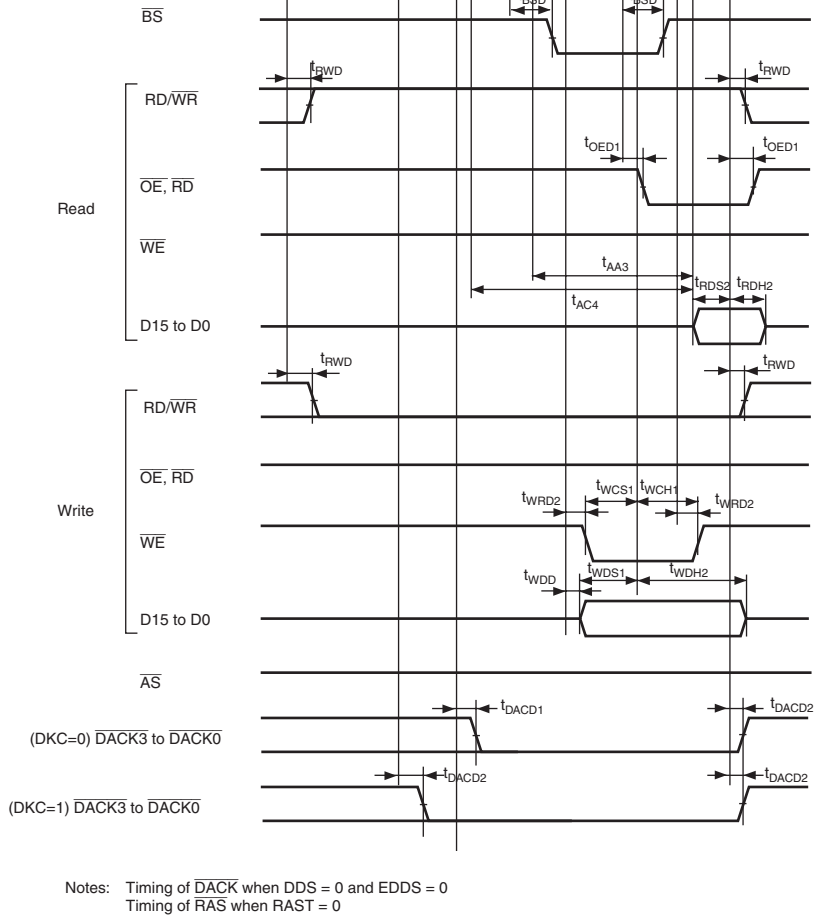


Figure 29.19 DRAM Access Timing / Two-State Access*

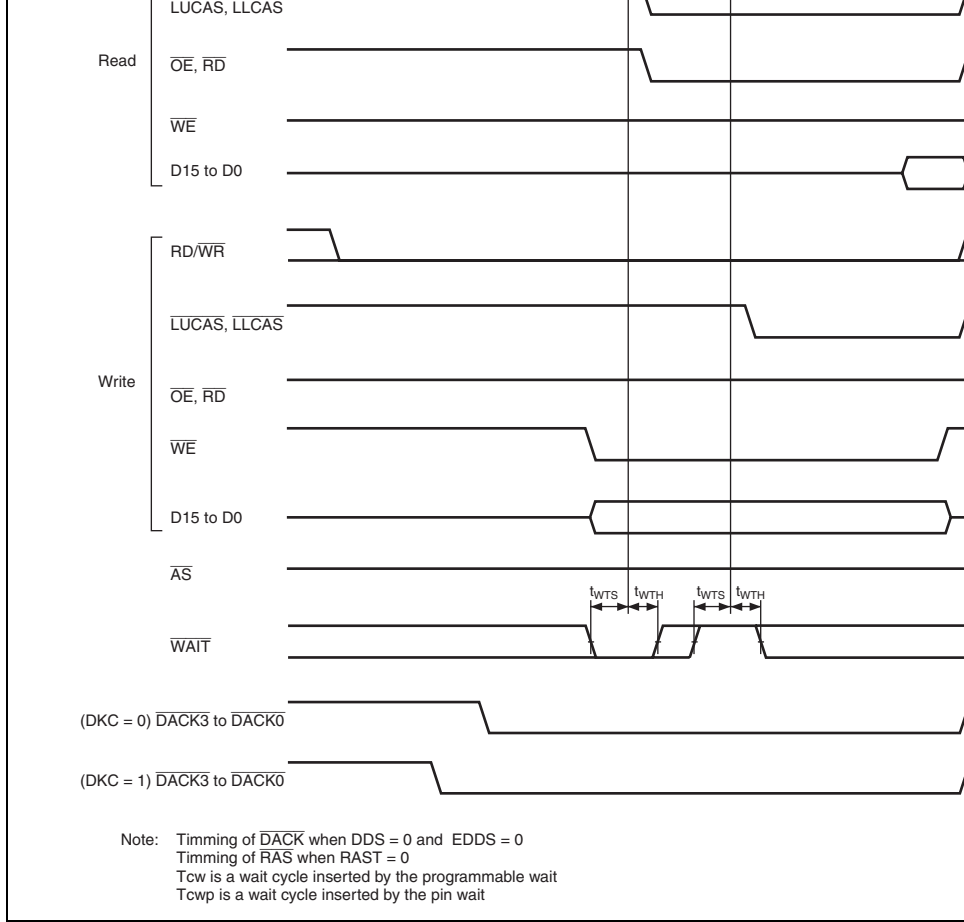


Figure 29.20 DRAM Access Timing / Two-State Access, One Wait*

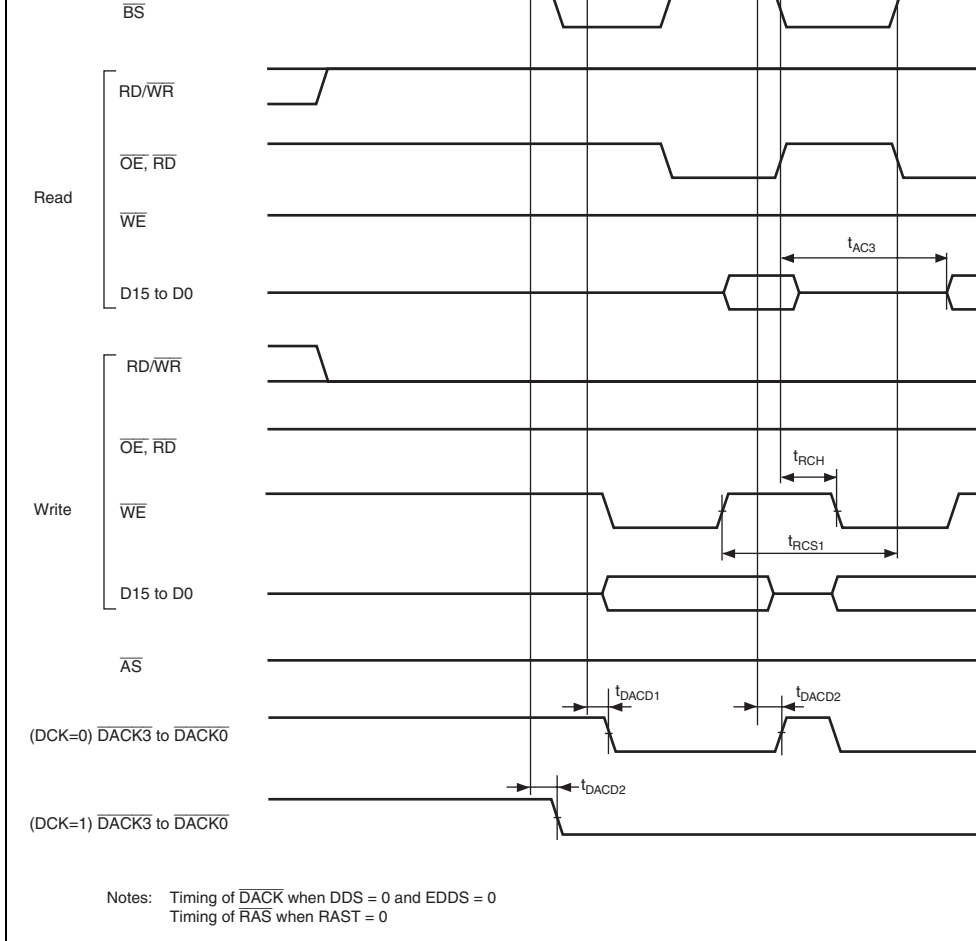


Figure 29.21 DRAM Access Timing / Two-State Burst Access*

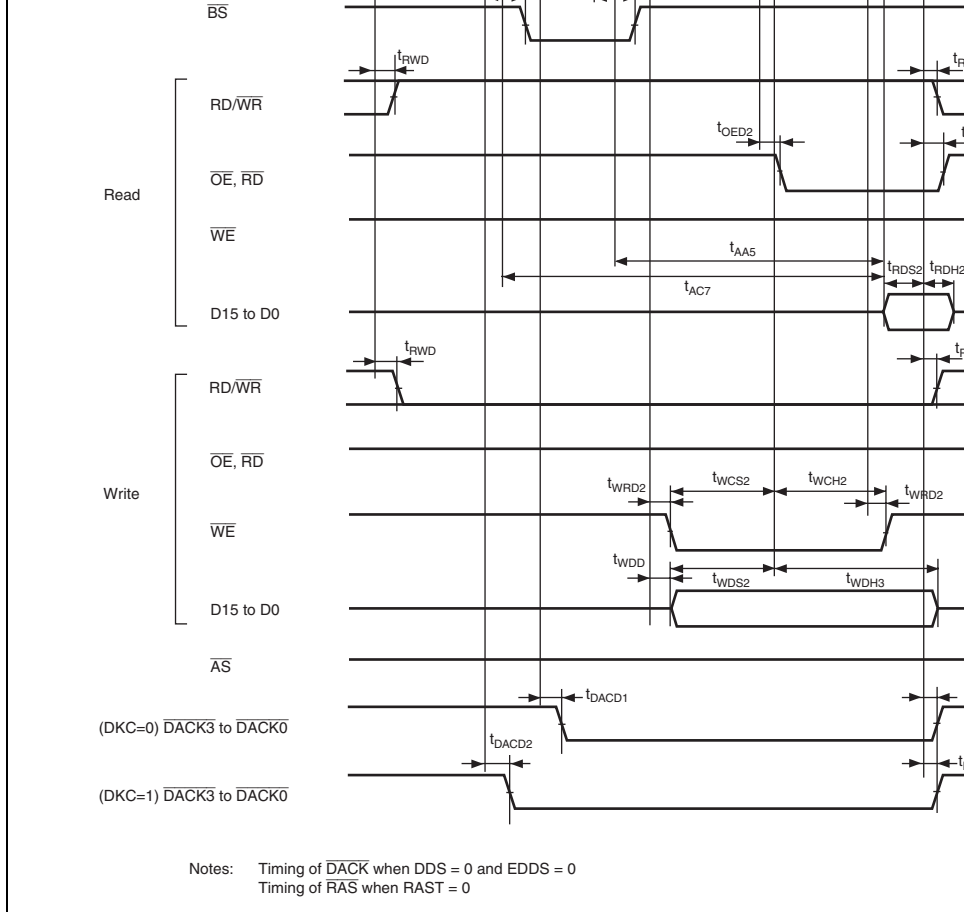


Figure 29.22 DRAM Access Timing / Three-State Access (When RAST=1)

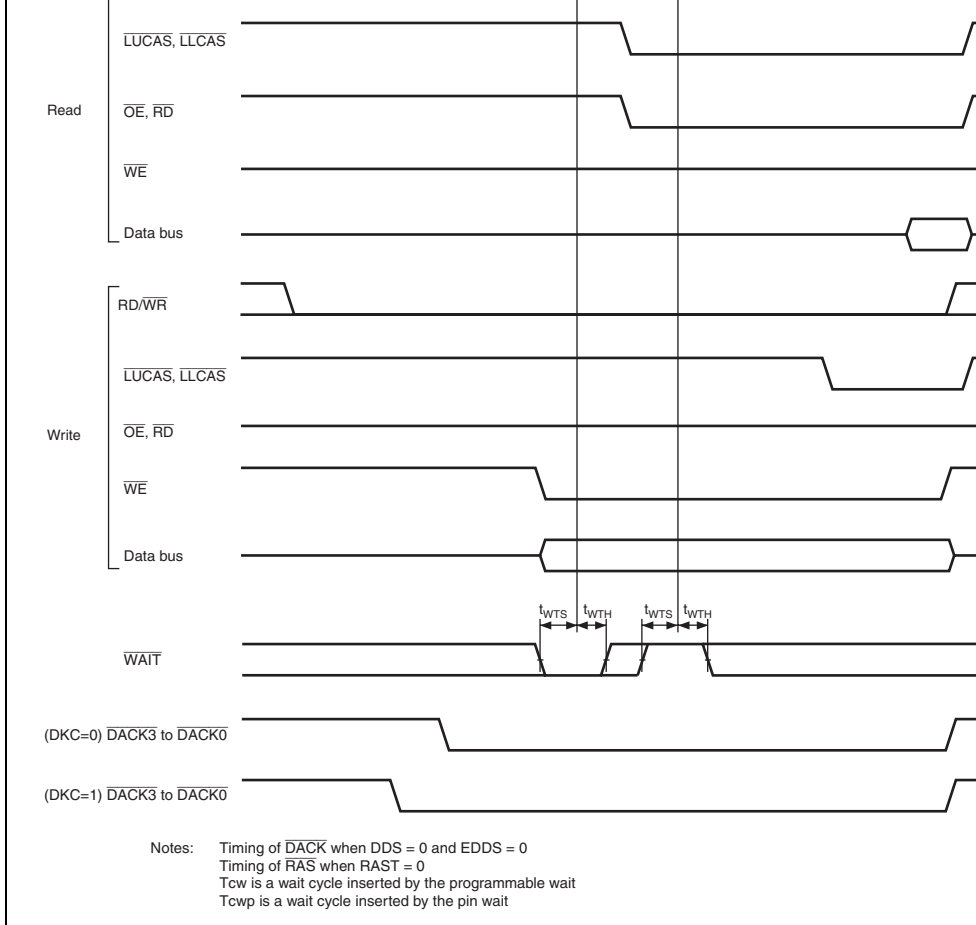


Figure 29.23 DRAM Access Timing / Three-State Access, One Wait*

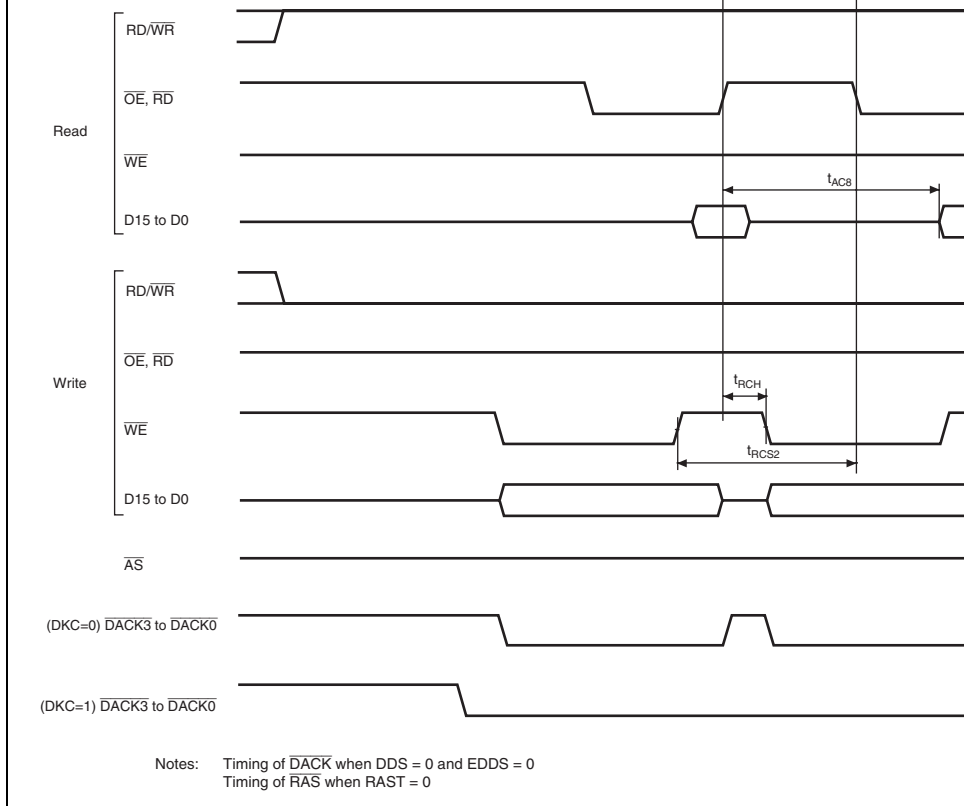


Figure 29.24 DRAM Access Timing / Three-State Burst Access*

Figure 29.25 CAS before RAS Refresh Timing*

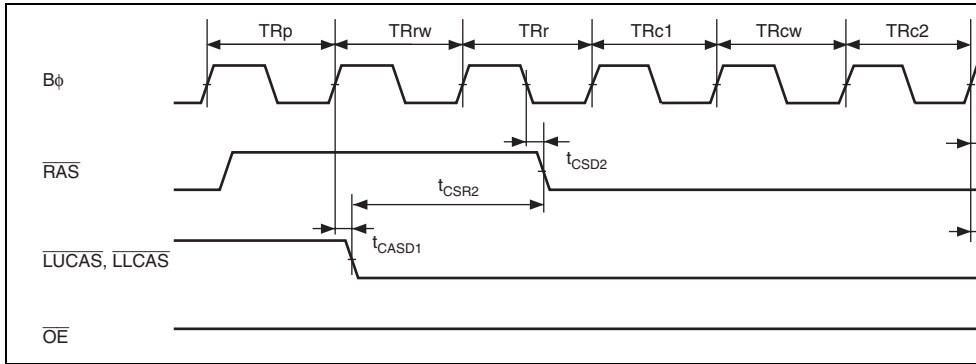


Figure 29.26 CAS before RAS Refresh Timing (Wait Cycle Inserted)*

Figure 29.27 Self-Refresh Timing (After Leaving Software Standby: RAST)

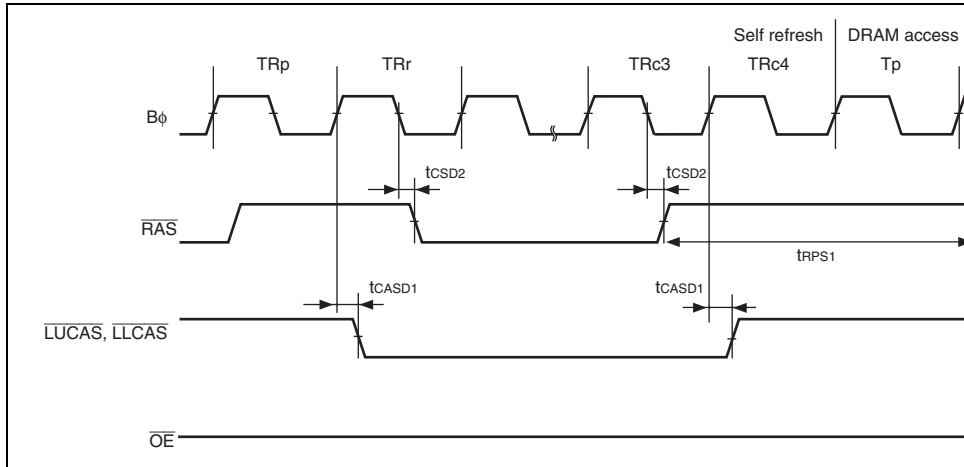


Figure 29.28 Self-Refresh Timing (After Leaving Software Standby: RAST)

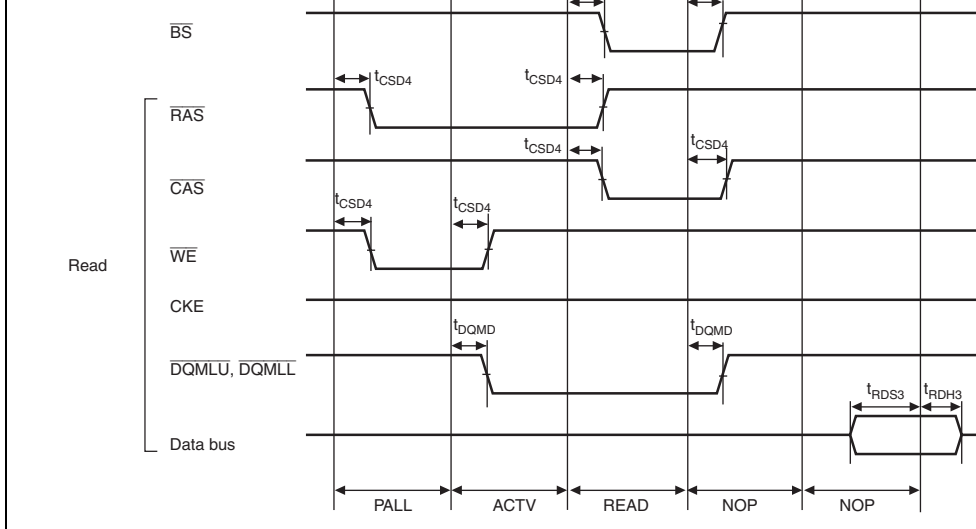


Figure 29.29 Synchronous DRAM Basic Read Access Timing (CAS Latency

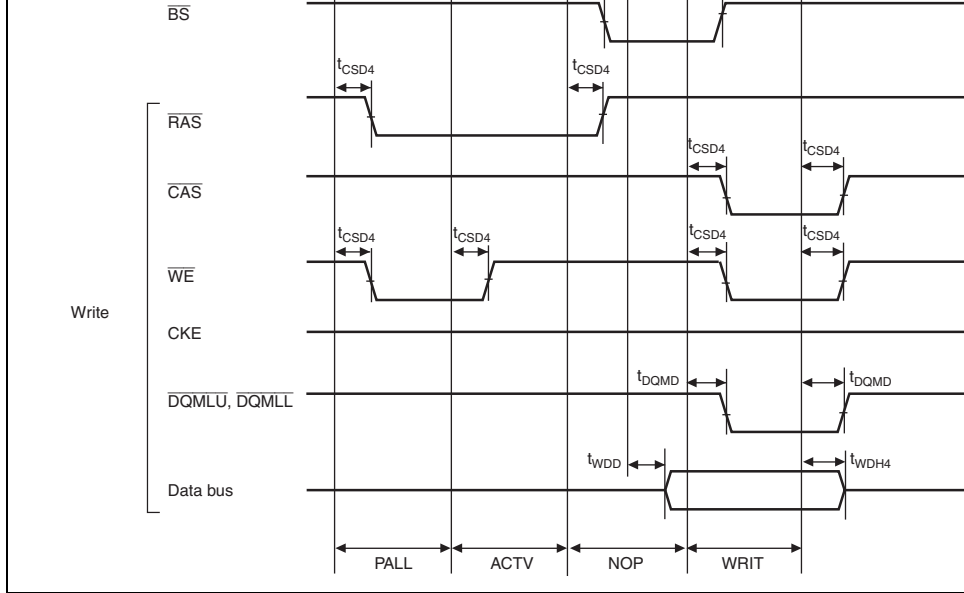


Figure 29.30 Synchronous DRAM Basic Write Access Timing (CAS Latency)

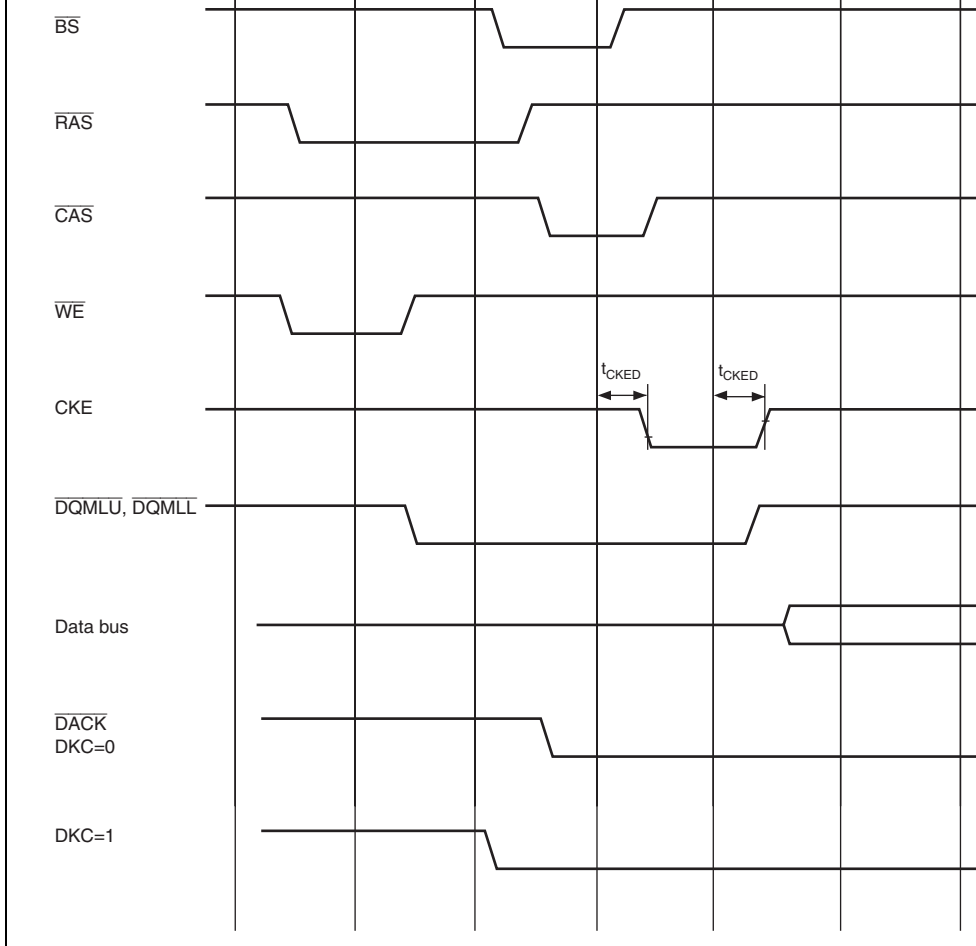


Figure 29.31 Extended Read Data Cycle (CAS Latency 2)*

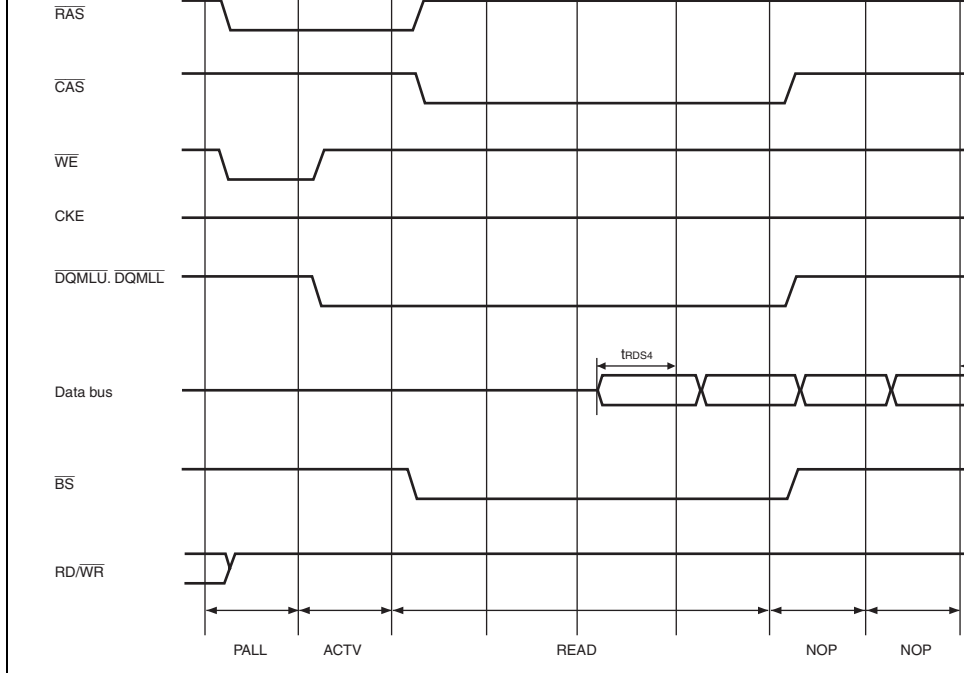


Figure 29.32 Synchronous DRAM Cluster Transfer Access Timing (Read)

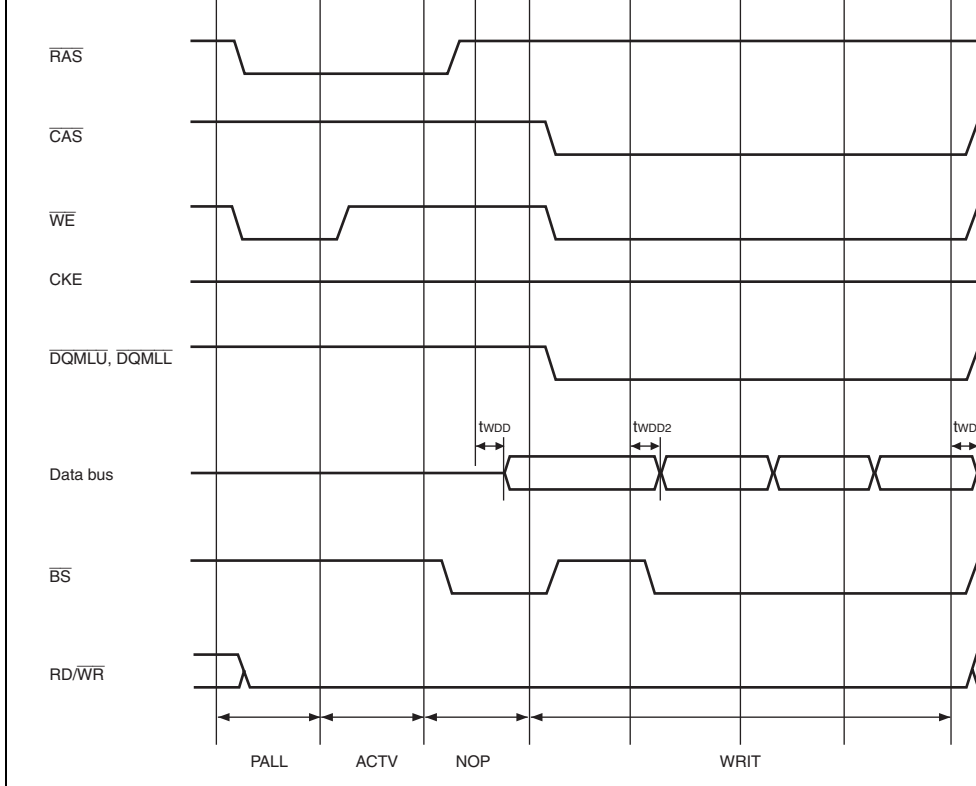


Figure 29.33 Synchronous DRAM Cluster Transfer Access Timing (Write)

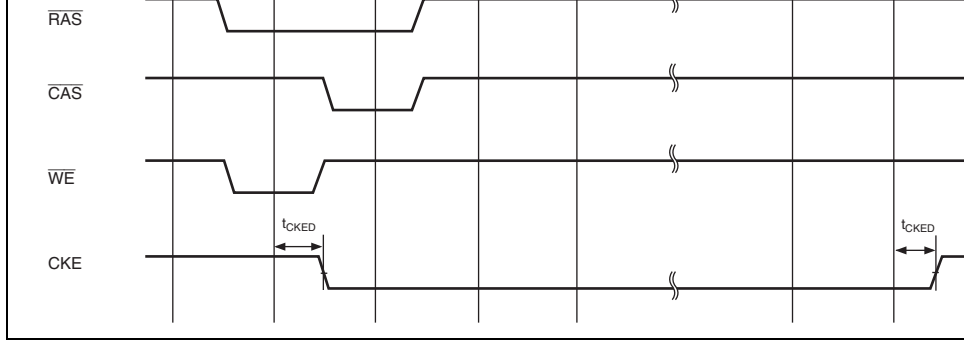


Figure 29.34 Synchronous DRAM Self-Refresh Timing*

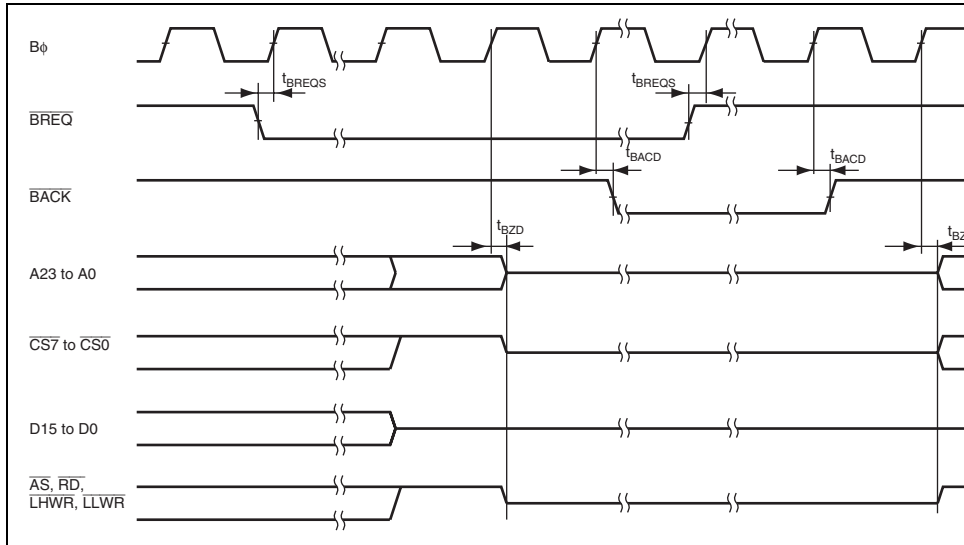


Figure 29.35 External Bus Release Timing

29.4.4 Timing of DMAC and EXDMAC

Table 29.9 Timing of DMAC and EXDMAC

Conditions: $V_{cc} = PLLV_{cc} = 3.0\text{ V to }3.6\text{ V}$, $AV_{cc} = 3.0\text{ V to }3.6\text{ V}$, $V_{ref} = 3.0\text{ V to }AV_{cc}$
 $V_{ss} = PLLV_{ss} = AV_{ss} = 0\text{ V}$, $B\phi = 8\text{ MHz to }50\text{ MHz}$,
 $T_a = -20^\circ\text{C to }+75^\circ\text{C}$ (regular specifications),
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$ (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Con
$\overline{\text{DREQ}}$ setup time	t_{DRQS}	20	—	ns	Figure 29
$\overline{\text{DREQ}}$ hold time	t_{DRQH}	5	—	ns	
$\overline{\text{TEND}}$ delay time	t_{TED}	—	15	ns	Figure 29
$\overline{\text{DACK}}$ delay time 1	t_{DADC1}	—	15	ns	Figures 29
$\overline{\text{DACK}}$ delay time 2	t_{DADC2}	—	15	ns	Figure 29
$\overline{\text{EDREQ}}$ setup time* ¹	t_{EDRQS}	20	—	ns	Figure 29
$\overline{\text{EDREQ}}$ hold time* ¹	t_{EDRQH}	5	—	ns	
$\overline{\text{ETEND}}$ delay time* ¹	t_{ETED}	—	15	ns	Figure 29
$\overline{\text{EDACK}}$ delay time 1* ¹	t_{EDADC1}	—	15	ns	Figures 29
$\overline{\text{EDACK}}$ delay time 2* ¹	t_{EDADC2}	—	15	ns	Figure 29
$\overline{\text{EDRAK}}$ delay time* ¹	t_{EDRKD}	—	15	ns	Figure 29

Notes: 1. Supported only by the H8SX/1648G Group and H8SX/1648H Group.
2. $V_{cc}=2.95\text{V to }3.60\text{V}$ in the H8SX/1648L Group and H8SX/1648H Group.

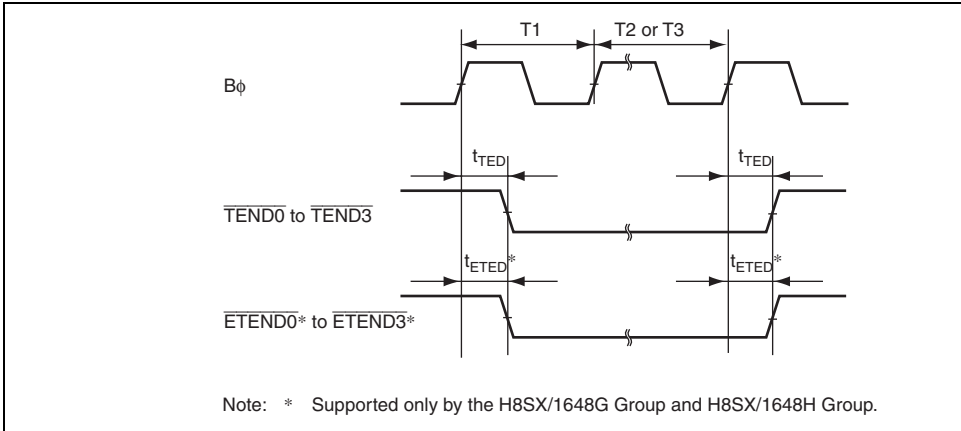
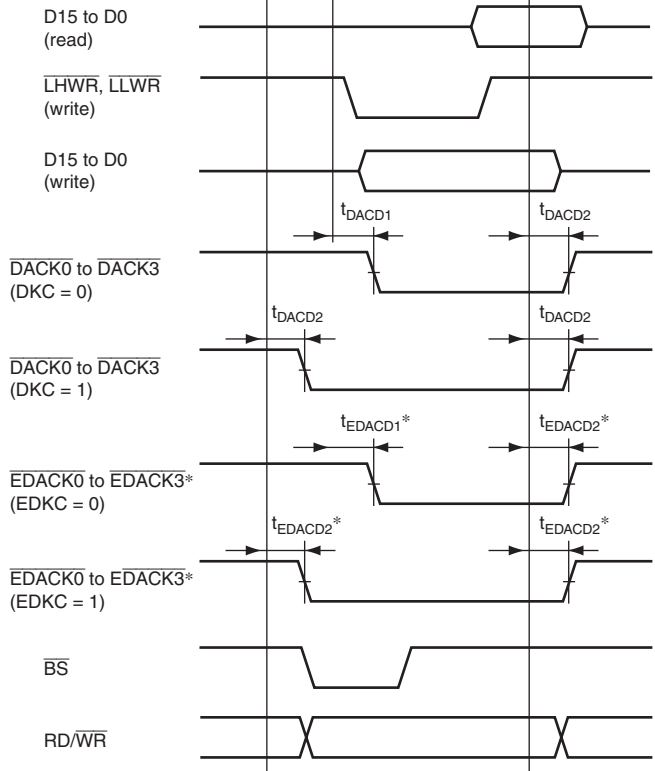
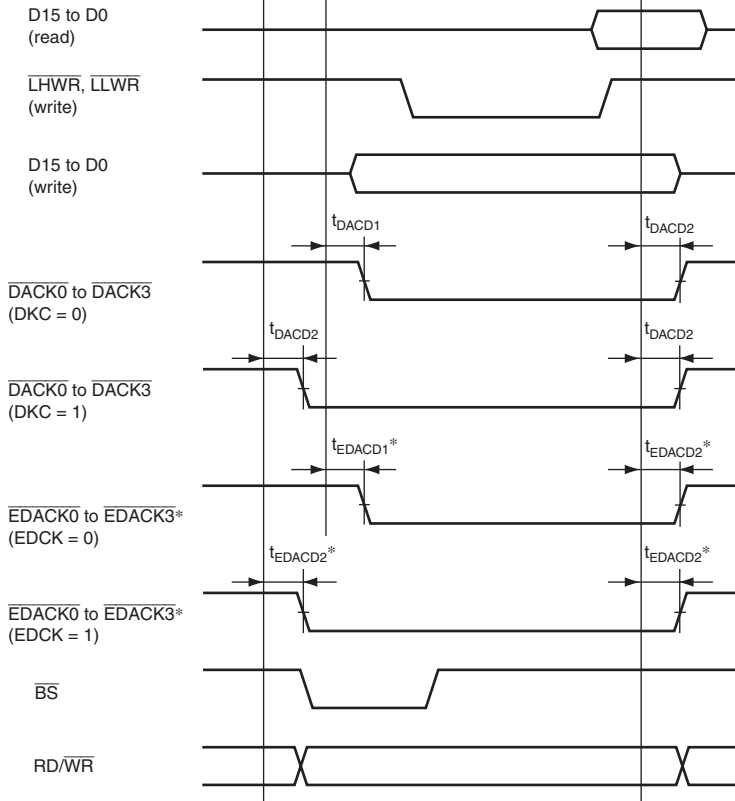


Figure 29.38 Timing of $\overline{\text{TEND}}$ and $\overline{\text{ETEND}}^*$ input (for the DMAC and EXDMAC)



Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Figure 29.39 Single Address Transfer Timing for the DMAC and EXDMAC (Two-State Access)



Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

Figure 29.40 Single Address Transfer Timing for the DMAC and EXDMA (Three-State Access)

Figure 29.41 Timing of EXDMAC and EDRAK Output Timing*

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

		Port N	—	200			
	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

TMS setup time	t_{TMSS}	20	—	ns	Figure
TMS hold time	t_{TMSH}	20	—	ns	
TDI setup time	t_{TDIS}	20	—	ns	
TDI hold time	t_{TDIH}	20	—	ns	
TDO data delay time	t_{TDOD}	—	23	ns	

- Notes: 1. $t_{TCKcyc} \geq t_{TCKcyc}$
2. $V_{CC}=PLL_{VCC}=2.95V$ to $3.60V$ in the H8SX/1648L Group and H8SX/1648H Group

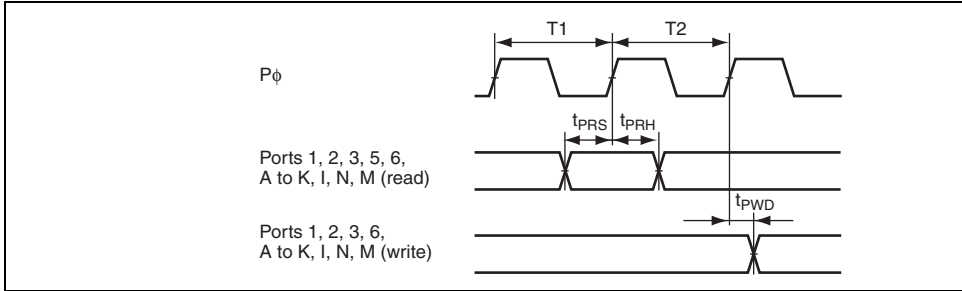


Figure 29.42 I/O Port Input/Output Timing

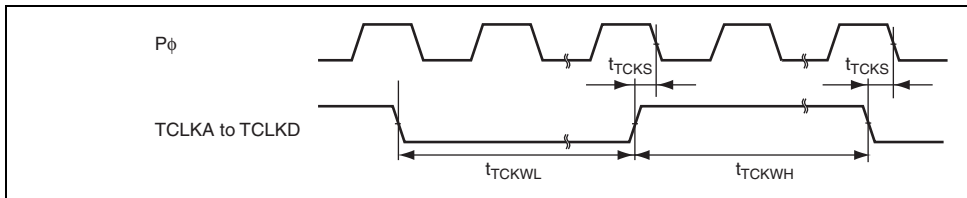


Figure 29.44 TPU Clock Input Timing

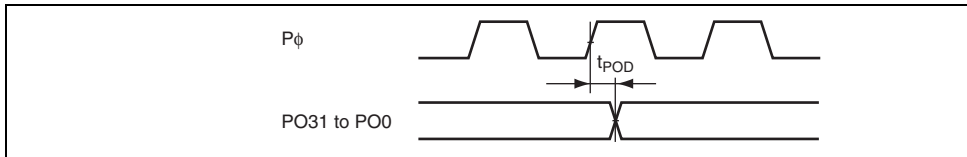


Figure 29.45 PPG Output Timing

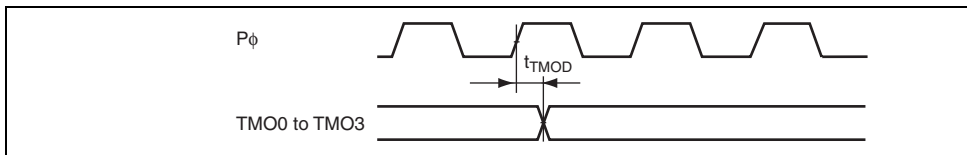


Figure 29.46 8-Bit Timer Output Timing

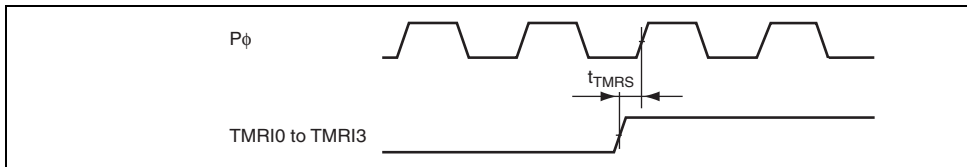


Figure 29.47 8-Bit Timer Reset Input Timing

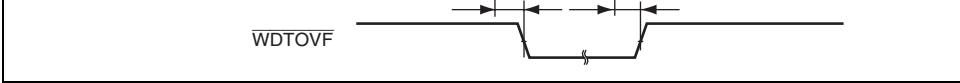


Figure 29.49 WDT Output Timing

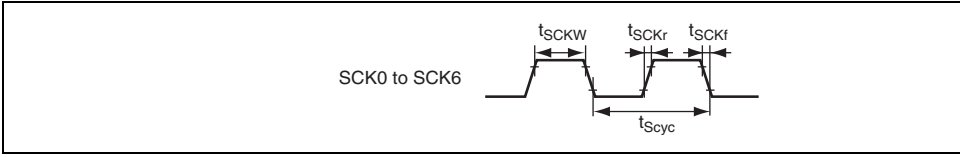


Figure 29.50 SCK Clock Input Timing

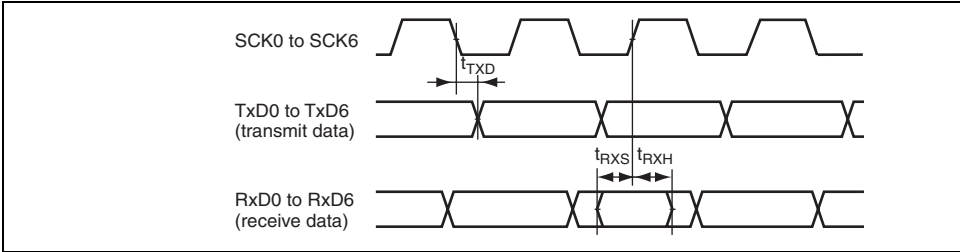


Figure 29.51 SCI Input/Output Timing: Clocked Synchronous Mode

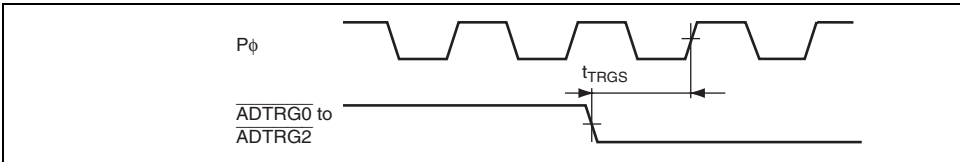


Figure 29.52 A/D Converter External Trigger Input Timing



Note: S, P, and Sr represent the following conditions:
 S: Start condition
 P: Stop condition
 Sr: Retransmit start condition

Figure 29.53 I²C Bus Interface 2 Input/Output Timing (Option)

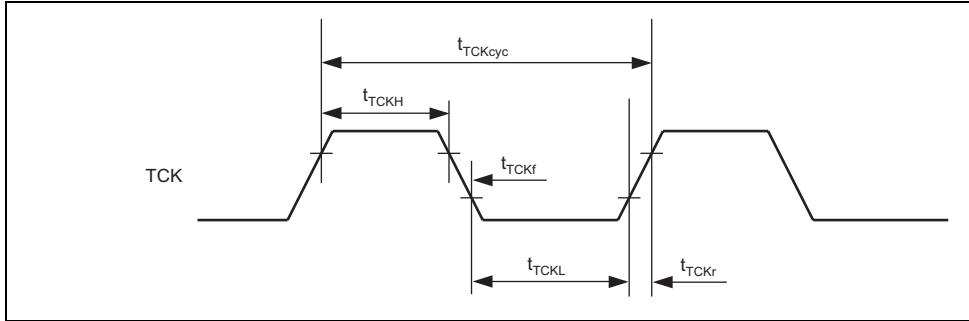


Figure 29.54 Boundary Scan TCK Timing

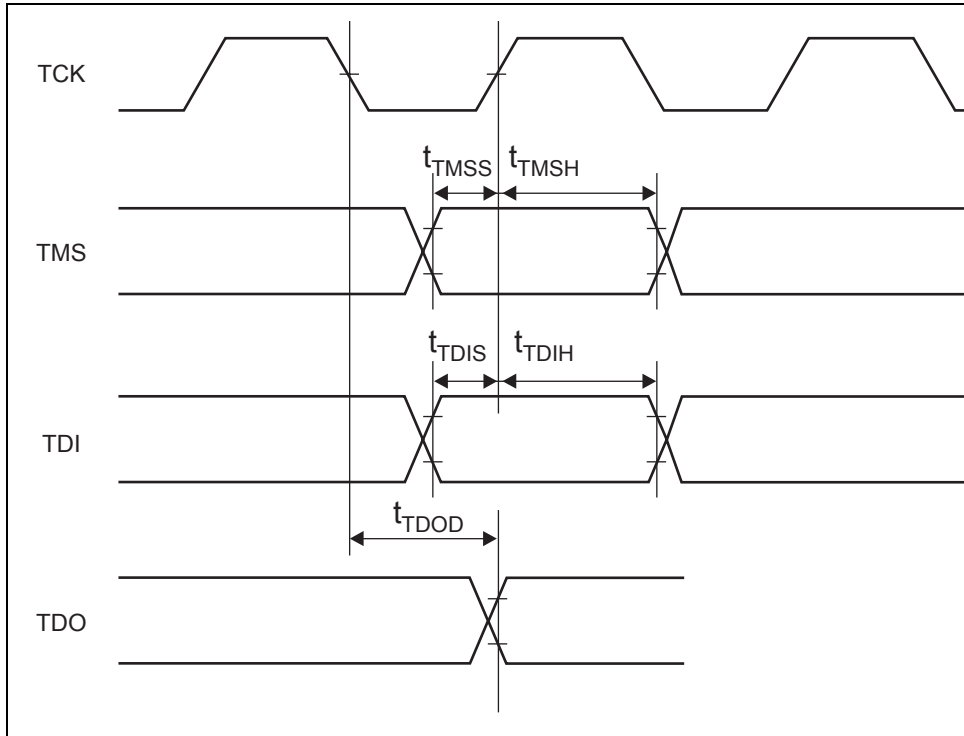


Figure 29.56 Boundary Scan Input/Output Timing

Conversion time		2.7	—	—	μs
Analog input capacitance		—	—	20	pF
Permissible signal source impedance		—	—	5	kΩ
Nonlinearity error	H8SX/1648 Group	—	—	±7.5	LSE
	H8SX/1648A Group				
	H8SX/1648L Group				
	H8SX/1648G Group	—	—	±3.5	LSE
	H8SX/1648H Group				
Offset error	H8SX/1648 Group	—	—	±7.5	LSE
	H8SX/1648A Group				
	H8SX/1648L Group				
	H8SX/1648G Group	—	—	±3.5	LSE
	H8SX/1648H Group				
Full-scale error	H8SX/1648 Group	—	—	±7.5	LSE
	H8SX/1648A Group				
	H8SX/1648L Group				
	H8SX/1648G Group	—	—	±3.5	LSE
	H8SX/1648H Group				
Quantization error		—	±0.5	—	LSE
Absolute accuracy	H8SX/1648 Group	—	—	±8.0	LSE
	H8SX/1648A Group				
	H8SX/1648L Group				
	H8SX/1648G Group	—	—	±4.0	LSE
	H8SX/1648H Group				

Note: * Vcc=PLL Vcc=2.95V to 3.60V in the H8SX/1648L Group and H8SX/1648H Group

Conversion time	—	—	10	μs	20-pF capac
Absolute accuracy	—	±2.0	±3.0	LSB	2-MΩ resist
	—	—	±2.0	LSB	4-MΩ resist

Note: * Vcc=PLLvcc=2.95V to 3.60V in the H8SX/1648L Group and H8SX/1648H Gr

Item	Symbol	Min.	Typ.	Max.	Unit	Tes Cor
Programming time* ¹ , * ² , * ⁴	t_p	—	1	10	ms/128 bytes	
Erasure time* ¹ , * ² , * ⁴	t_E	—	40	130	ms/4 kbyte- block	
		—	300	800	ms/32 kbyte- block	
		—	600	1500	ms/64 kbyte- block	
Programming time (total)* ¹ , * ² , * ⁴	Σ_{IP}	—	2.3	6	H8SX/1642, H8SX/1642A, H8SX/1642L, H8SX/1642G, H8SX/1642H, s/256 kbytes	$T_a =$ for a
		—	4.5	12	H8SX/1644, H8SX/1644A, H8SX/1644L, H8SX/1644G, H8SX/1644H, s/512 kbytes	
		—	9.0	24	H8SX/1648, H8SX/1648A, H8SX/1648L, H8SX/1648G, H8SX/1648H s/1 M byte	
Erasure time (total)* ¹ , * ² , * ⁴	Σ_{IE}	—	2.3	6	H8SX/1642, H8SX/1642A, H8SX/1642L, H8SX/1642G, H8SX/1642H, s/256 kbytes	$T_a =$
		—	4.5	12	H8SX/1644, H8SX/1644A, H8SX/1644L, H8SX/1644G, H8SX/1644H, s/512 kbytes	
		—	9.0	24	H8SX/1648, H8SX/1648A, H8SX/1648L, H8SX/1648G, H8SX/1648H s/1 M byte	

Overwrite count	N_{WEC}	100 ^{*v}	—	—	Times
Data save time* ⁴	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 (guaranteed value is from 1 to Min. value).
 4. Characteristics when programming is performed within the Min. value
 5. $V_{CC}=PLL_{VCC}=2.95V$ to $3.60V$ in the H8SX/1648L Group and H8SX/1648H Group

Voltage detection level	Voltage detection circuit (LVD)	V_{det}	3.00	3.10	3.20	V	Fig
	Power-on reset (POR)	V_{POR}	2.48	2.58	2.68		Fig
Internal reset time		t_{POR}	20	35	50	ms	Fig
Power-off time*		t_{VOFF}	200	—	—	us	Fig

Note: * Power-off time (t_{VOFF}) is the time over which V_{CC} is lower than minimum value of voltage-detection level of the POR and LVD.



Figure 29.57 Power-On Reset Timing

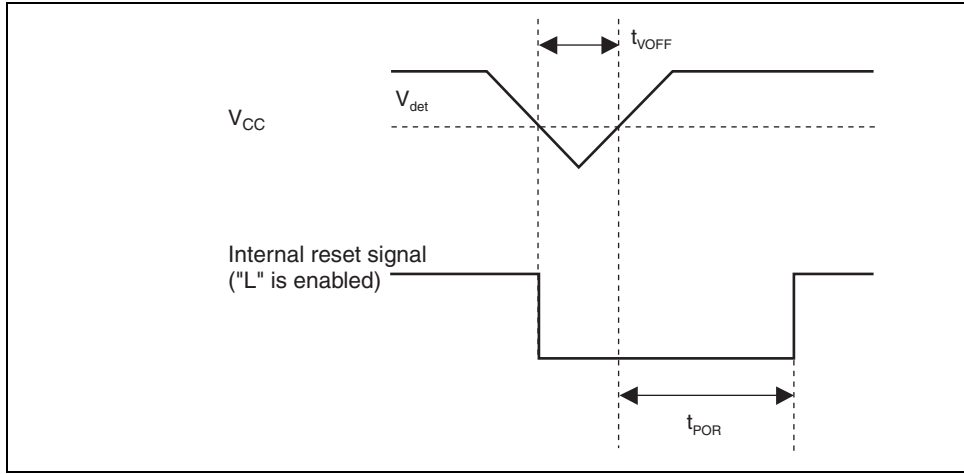


Figure 29.58 Voltage Detection Circuit Timing

Port 2	All	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep
Port 3	All	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep
Port 4	All	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep
P55 to P50	All	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
P56/ AN6/ DA0/ $\overline{\text{IRQ6-B}}$	All	Hi-Z	Hi-Z	Hi-Z	Hi-Z	[DAOE0 = 1]	[DAOE0 = 1]
						Keep	Keep
						[DAOE0 = 0]	[DAOE0 = 0]
P57/ AN7/ DA1/ $\overline{\text{IRQ7-B}}$	All	Hi-Z	Hi-Z	Hi-Z	Hi-Z	[DAOE1 = 1]	[DAOE1 = 1]
						Keep	Keep
						[DAOE1 = 0]	[DAOE1 = 0]
P67 to P60	All	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep
PA0/ $\overline{\text{BREQO}}$ / BS-A	All	Hi-Z	Hi-Z	[BREQO output]	[BREQO output]	[$\overline{\text{BREQO}}$ output]	[BREQO output]
				Hi-Z	Hi-Z	Hi-Z	Hi-Z
				[$\overline{\text{BS}}$ output]	[$\overline{\text{BS}}$ output]	[$\overline{\text{BS}}$ output]	[$\overline{\text{BS}}$ output]
				Keep	Hi-Z	Keep	Hi-Z
				[Other than above]	[Other than above]	[Other than above]	[Other than above]
				Keep	Keep	Keep	Keep
PA1/ BACK/ (RD/ $\overline{\text{WR-A}}$)	All	Hi-Z	Hi-Z	[$\overline{\text{BACK}}$ output]	[$\overline{\text{BACK}}$ output]	[$\overline{\text{BACK}}$ output]	[BACK output]
				Hi-Z	Hi-Z	Hi-Z	Hi-Z
				[RD/ $\overline{\text{WR-A}}$ output]	[RD/ $\overline{\text{WR-A}}$ output]	[RD/ $\overline{\text{WR-A}}$ output]	[RD/ $\overline{\text{WR-A}}$ output]
				Keep	Hi-Z	Keep	Hi-Z
				[Other than above]	[Other than above]	[Other than above]	[Other than above]
				Keep	Keep	Keep	Keep

LLWR/ LLB	mode (EXPE = 0)								
	External extended mode (EXPE = 1)	H	Hi-Z	H	Hi-Z	H	Hi-Z	H	Hi-Z
PA4/ LHWR/ LUB	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep	Keep	Keep
	External extended mode (EXPE = 1)	H	Hi-Z	[LHWR, LUB output] H [Other than above] Keep	[LHWR, LUB output] Hi-Z [Other than above] Keep	[LHWR, LUB output] H [Other than above] Keep	[LHWR, LUB output] Hi-Z [Other than above] Keep	[LHWR, LUB output] H [Other than above] Keep	[LHWR, LUB output] H [Other than above] Keep
PA5/RD	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep	Keep	Keep
	External extended mode (EXPE = 1)	H	Hi-Z	H	Hi-Z	H	Hi-Z	H	Hi-Z
PA6/ AS/ AH/ BS-B	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	[AS, BS output] H	[AS, AH, BS output] Hi-Z	[AS, BS output] H	[AS, AH, BS output] Hi-Z	[AS, AH, BS output] Hi-Z	[AS, AH, BS output] Hi-Z
	External extended mode (EXPE = 1)	H	Hi-Z	[AH output] L [Other than above] Keep	[Other than above] Keep	[AH output] L [Other than above] Keep	[Other than above] Keep	[Other than above] Keep	[Other than above] Keep

CS5-B	External extended mode (EXPE = 1)	H	Hi-Z	[Other than above] Keep	[Other than above] Keep	[Other than above] Keep	[Other than above] Keep
PB1/ $\overline{\text{CS1}}$ / $\overline{\text{CS2-B}}$ / $\overline{\text{CS5-A}}$ / $\overline{\text{CS6-B}}$ / $\overline{\text{CS7-B}}$	All	Hi-Z	Hi-Z	$\overline{\text{CS}}$ output H [Other than above] Keep	$\overline{\text{CS}}$ output Hi-Z [Other than above] Keep	$\overline{\text{CS}}$ output H [Other than above] Keep	$\overline{\text{CS}}$ output Hi-Z [Other than above] Keep
PB2/ $\overline{\text{CS2-A}}$ / $\overline{\text{CS6-A/RAS}}^*$	All	Hi-Z	Hi-Z	$\overline{\text{CS}}$ output H [Other than above] Keep	$\overline{\text{CS}}$ output Hi-Z [Other than above] Keep	$\overline{\text{CS}}$ output H [Other than above] Keep	$\overline{\text{CS}}$ output Hi-Z [Other than above] Keep
PB3/ $\overline{\text{CS3}}$ / $\overline{\text{CS7-A}}$ / CAS*	All	Hi-Z	Hi-Z	$\overline{\text{CS}}$ output H [Other than above] Keep	$\overline{\text{CS}}$ output Hi-Z [Other than above] Keep	$\overline{\text{CS}}$ output H [Other than above] Keep	$\overline{\text{CS}}$ output Hi-Z [Other than above] Keep
PB4/ $\overline{\text{CS4-B}}$ / $\overline{\text{WE}}^*$	All	Hi-Z	Hi-Z	$\overline{\text{CS}}$ output H [Other than above] Keep	$\overline{\text{CS}}$ output Hi-Z [Other than above] Keep	$\overline{\text{CS}}$ output H [Other than above] Keep	$\overline{\text{CS}}$ output Hi-Z [Other than above] Keep
PB5/ $\overline{\text{CS5-B}}$ / $\overline{\text{OE}}^*$	All	Hi-Z	Hi-Z	$\overline{\text{CS}}$ output H [Other than above] Keep	$\overline{\text{CS}}$ output Hi-Z [Other than above] Keep	$\overline{\text{CS}}$ output H [Other than above] Keep	$\overline{\text{CS}}$ output Hi-Z [Other than above] Keep

SDRAM mode				H	Hi-Z	H	Hi-Z	H	Hi-Z
				[Other than above]	[Other than above]	[Other than above]	[Other than above]	[Other than above]	[Other than above]
				Keep	Keep	Keep	Keep	Keep	Keep
PC0/ CS3-B/ WAIT-B/ ADTRGH1-B	All	Hi-Z	Hi-Z	[WAIT input] Hi-Z	[WAIT input] Hi-Z	[WAIT input] Hi-Z	[WAIT input] Hi-Z	[WAIT input] Hi-Z	[WAIT input] Hi-Z
				[CS output] H	[CS output] Hi-Z	[CS output] H	[CS output] Hi-Z	[CS output] H	[CS output] Hi-Z
				[Other than above]	[Other than above]	[Other than above]	[Other than above]	[Other than above]	[Other than above]
				Keep	Keep	Keep	Keep	Keep	Keep
PC1/ CS4-C/ CS5-C	All	Hi-Z	Hi-Z	[CS output] H	[CS output] Hi-Z	[CS output] H	[CS output] Hi-Z	[CS output] H	[CS output] Hi-Z
				[Other than above]	[Other than above]	[Other than above]	[Other than above]	[Other than above]	[Other than above]
				Keep	Keep	Keep	Keep	Keep	Keep
PC2 to PC5	All	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep	Keep	Keep

	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep
Port E	External extended mode (EXPE = 1)	L	Hi-Z	Keep	Hi-Z	Keep	Hi-Z
	ROM enabled extended mode	Hi-Z	Hi-Z	Keep	[Address output] Hi-Z [Other than above] Keep	Keep	[Address output] Hi-Z [Other than above] Keep
	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep
PF3 to PF0	External extended mode (EXPE = 1)	L	Hi-Z	Keep	Hi-Z	Keep	Hi-Z
	ROM enabled extended mode	Hi-Z	Hi-Z	Keep	[Address output] Hi-Z [Other than above] Keep	Keep	[Address output] Hi-Z [Other than above] Keep
	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep
PF7 to PF4	External extended mode (EXPE = 1)	L/ Hi-Z*	Hi-Z	Keep	[Address output] Hi-Z [Other than above] Keep	Keep	[Address output] Hi-Z [Other than above] Keep
	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep	Keep

	mode (EXPE = 1)	mode 16-bit bus mode	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	H
Port J	Hi-Z		Hi-Z	Hi-Z	Keep	Keep	Keep	Keep	K
Port K	Hi-Z		Hi-Z	Hi-Z	Keep	Keep	Keep	Keep	K
Port M*	Hi-Z		Hi-Z	Hi-Z	Keep	Keep	Keep	Keep	K
Port N	Hi-Z		Hi-Z	Hi-Z	Keep	Keep	Keep	Keep	K

[Legend]

H: High-level output

L: Low-level output

Keep: Input pins become high-impedance, output pins retain their state.

Hi-Z: High impedance

Note: * Supported only by the H8SX/1648G Group and H8SX/1648H Group.

H8SX/1642L	R5F61642L	R5F61642LFPV	PLQP0144KA-A (FF
H8SX/1644L	R5F61644L	R5F61644LFPV	PLQP0144KA-A (FF
H8SX/1648L	R5F61648L	R5F61648LFPV	PLQP0144KA-A (FF
H8SX/1642G	R5F61642G	R5F61642GBGV	PLBG0176GA-A (BF
H8SX/1644G	R5F61644G	R5F61644GBGV	PLBG0176GA-A (BF
H8SX/1648G	R5F61648G	R5F61648GBGV	PLBG0176GA-A (BF
H8SX/1642H	R5F61642H	R5F61642HBGV	PLBG0176GA-A (BF
H8SX/1644H	R5F61644H	R5F61644HBGV	PLBG0176GA-A (BF
H8SX/1648H	R5F61648H	R5F61648HBGV	PLBG0176GA-A (BF

Note: * Pb-free version

JEITA Package Code P-LQFP144-20x20-0.50	RENESAS Code PLOP0144KA-A	Previous Code 144P6Q-A / FP-144L / FP-144LV	MASS[Typ.] 1.2g
--	------------------------------	--	--------------------

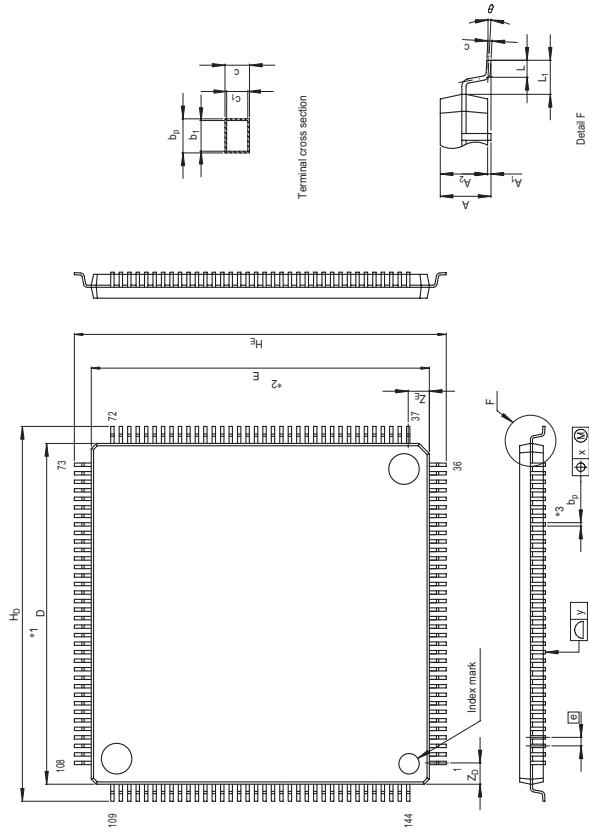


Figure C.1 Package Dimensions (FP-144LV)

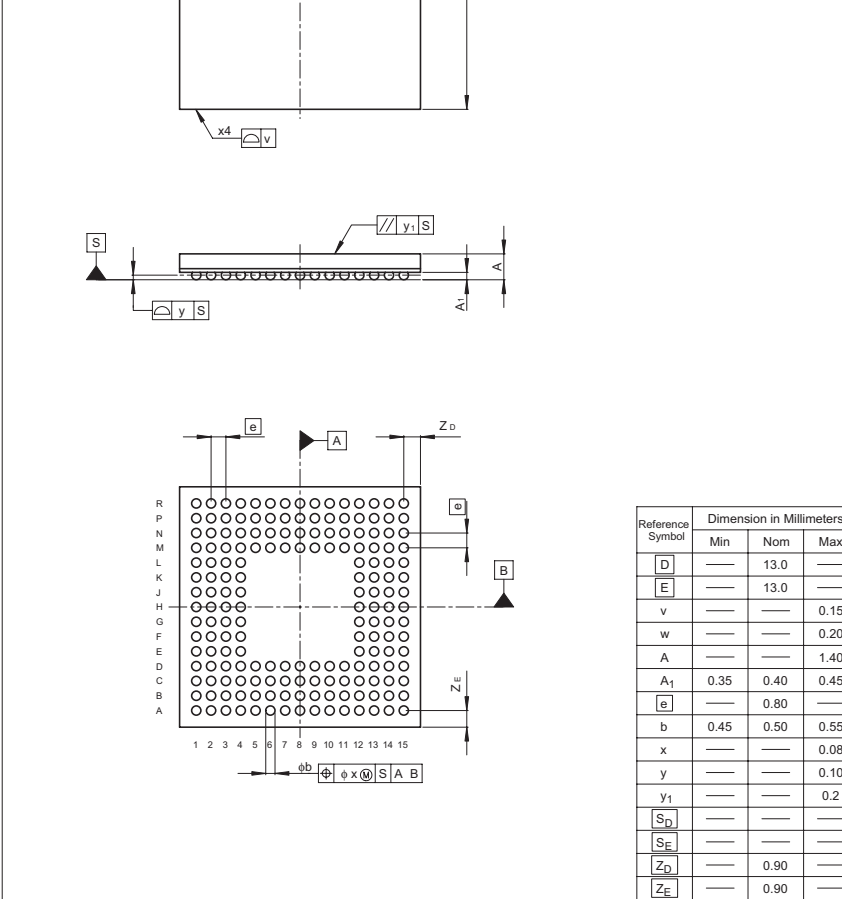


Figure C.2 Package Dimensions (BP-176V)

MD3*	(Always used as mode pins)
MD2 to MD0	(Always used as mode pins)
NMI	<ul style="list-style-type: none"> • Connect this pin to VCC via a pull-up resistor
EXTAL	(Always used as a clock pin)
XTAL	<ul style="list-style-type: none"> • Leave this pin open
WDTOVF	<ul style="list-style-type: none"> • Leave this pin open
Port 1	<ul style="list-style-type: none"> • Connect these pins to VCC via a pull-up resistor or to VSS via a pull-down resistor, respectively
Port 2	
Port 3	
Port 6	
Port 7	
PA2 to PA0	
PB7 to PB1	
Port C	
PF7 to PF5	
Port J	
Port K	
Port N	
Ports 4, 5	<ul style="list-style-type: none"> • Connect these pins to AVcc via a pull-up resistor or to AVss via a pull-down resistor, respectively

Port D	<ul style="list-style-type: none"> 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)	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Connect this pin to AVcc 	

- Notes:
- Do not change the initial value (input-buffer disabled) of PnICR, where n corresponds to an unused pin.
 - When the pin function is changed from its initial state, use a pull-up or pull-down resistor as needed.
- *1 Supported only by the H8SX/1648G Group and H8SX/1648H Group.
- *2 Always used as a reset signal input pin in case of the H8SX/1648 Group, H8SX/1648G Group, and H8SX/1648H Group.

2. CPU
 3. MCU Operating Modes
 4. Resets
 5. Voltage Detection Circuit (LVD)
 6. Exception Handling
 7. Interrupt Controller
 8. User Break Controller (UBC)
 9. Bus Controller (BSC)
 10. DMA Controller (DMAC)
 11. EXDMA Controller (EXDMAC)
 12. Data Transfer Controller (DTC)
 13. I/O Ports
 14. 16-Bit Timer Pulse Unit (TPU)
 15. Programmable Pulse Generator (PPG)
 16. 8-bit Timer
 17. 32K Timer (TM32K)
 18. Watch Dog Timer (WDT)
 19. Serial Communication Interface (SCI, IrDA, CF)
 20. I2C Bus Interface 2 (IIC2)
 21. A/D Converter
 22. D/A Converter
 23. RAM
 24. Flash Memory
 25. Boundary Scan
 26. Clock Pulse Generator
-

- I/O ports

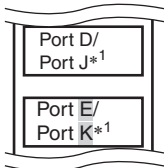
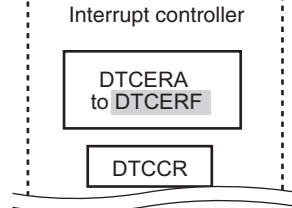
1.2 List of Products	12	Deleted
Figure 1.1 How to Read the Product Name Code		H8SX/1648 Group
1.3 Block Diagram	13	Added
Figure 1.2 Block Diagram		<ul style="list-style-type: none"> Main clock oscillator Amended
		
Section 3 MCU Operating Modes	86	Amended
3.4.1 Address Map		H'FEC000 to H'FEE000 of each mode
Figure 3.1 Address Map in Each Operating Mode of H8SX/1648, 1648L, 1648G, and 1648H (1)		[Before amendment] Access prohibited area → [After amendment] Reserved area* ³
Figure 3.1 Address Map in Each Operating Mode of H8SX/1648, 1648L, 1648G, and 1648H (2)	87	Amended and added
		<ul style="list-style-type: none"> H'FEC000 to H'FEE000 of each mode [Before amendment] Access prohibited area → [After amendment] Reserved area* ¹
		Note 1 is added.

Figure 3.3 Address Map in Each Operating Mode of H8SX/1642, 1642L, 1642G, and , 1642H (1)	90	Amended H'FEC000 to H'FE6000 of each mode [Before amendment] Access prohibited area → [After amendment] Reserved area* ³
Figure 3.3 Address Map in Each Operating Mode of H8SX/1642, 1642L, 1642G, and 1642H (2)	91	Amended H'FEC000 to H'F6000 of each mode [Before amendment] Access prohibited area → [After amendment] Reserved area* ¹ Note 1 is added
Section 4 Reset 4.3.1 Reset Status Register (RSTSR)	96	Amended Description of Bit 7 [Before amendment] External interrupt source → [After amendment] Interrupt source
Section 6 Exception Handling 6.6.1 Interrupt sources	121	Added Table 6.7 Interrupt sources Type of interrupt IRQ0 to IRQ15 is added.
Section 7 Interrupt Controller 7.3.4 IRQ Enable Register	136	Amended IER enables interrupt requests IRQ15 to IRQ0.
Section 9 Bus Controller (BSC) 9.11.12 Fast-Page Access Operation Figure 9.73 Word Read Timing in 8-Bit Access Space (BE = 1, RCDM = 0, CAS Latency = 2)	329	Amended Positions of dotted lines have been corrected.

Section 10 DMA Controller (DMAC)	384	Added
10.1 Features		Module stop state can be set.
10.3.5 DMA Block Size Register (DBSR)	392	Amended
		Bit table
		Initial Value of the Bit 31-16 and 15-0
		[Before amendment] Undefined →
		[After amendment] All 0
10.5.8 Priority of Channels	436	Amended
Figure 10.22 Example of Timing for Channel Priority		Positions of dotted lines have been corrected.

Figure 12.1 Block Diagram of DTC



12.4 Location of Transfer Information and DTC Vector Table	579	Amended
Table 12.1 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs		DTC Vector Address Offset IRQ12 76 H'530 IRQ13 77 H'534 IRQ14 78 H'538
12.9.3 DMAC Transfer End Interrupt	600	Added
		When the DTC is activated by a DMAC transfer end interrupt, even if DISEL=0, an automatic clearing of relevant activation source flag is not automatically performed by the DTC. Therefore, write 1 to the DTE bit by the transfer and clear the activation source flag to 0.
12.9.9 Points for Caution when Overwriting DTCER.	601	Added

13.3.5 Port Function Control Register 6 (PFCR6) 688

Amended

Bit Name **ADTRG1S** to the bit 1 in the bit table is

Amended

- Descriptions of the Bit 0 "ADTRG0S"

Selects the external trigger input pins of the A/D (unit0).

13.3.6 Port Function Control Register 7 (PFCR7) 688

Amended

Descriptions for bits 7,6,5, and 4

[Before amendment] 00: Setting prohibited →

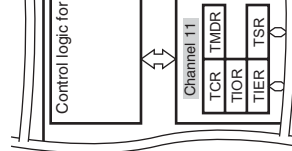
[After amendment] 00: Setting **invalid**

13.3.10 Port Function Control Register B (PFCRB) 696

Amended

Bit name in the bit table

ITS13



14.3.4 Timer Interrupt Enable Register (TIER)	741	Amended Bit name in the bit table Bit 2 TGIEC
Section 18. Watchdog Timer (WDT) 18.3.3 Reset Control/Status Register (RSTCSR)	877	Deleted Description below for bit 7 is deleted (When the CPU is used to clear this flag by writing the corresponding interrupt is enabled, be sure to flag after writing 0 to it.)
Section 19 Serial Communications Interface (SCI, IrDA, CRC) 19.3.7 Serial Status Register (SSR)	905	Amended The bit name FER
19.3.12 IrDA Control Register (IrCR)	931	Amended Description for the Bit 7 in the register table 1: TxD5/IrTxD and RxD5/IrRxD pins are operate a and IrRxD
19.7.8 Clock Output Control • At mode switching a) At transition from smart card interface mode to software standby mode	967	Deleted 1. Set the data register (DR) and data direction register (DDR) corresponding to the SCK pin to the values output fixed state in software standby mode. (SCI and 4 only)

corresponding interrupt is enabled, be sure to read after writing 0 to it)

1001 Deleted

The description below for the bit 1 is deleted:

[Clearing condition]

When 0 is written to this bit after reading STOP = ~~the CPU is used to clear this flag by writing 0 when the corresponding interrupt is enabled, be sure to read after writing 0 to it.)~~

Deleted

The description below for the bit 0 is deleted:

[Clearing condition]

When 0 is written to this bit after reading STOP = ~~the CPU is used to clear this flag by writing 0 when the corresponding interrupt is enabled, be sure to read after writing 0 to it.)~~

Section 21 A/D Converter	1030	Added
--------------------------	------	-------

21.3.2 A/D Control/Status Register for Unit 0 (ADCSR_0)

Descriptions for bit 5 in the register table:

Note: Do not write to ADST when activation is by external trigger. For details, see section 21 on A/D activation by an External Trigger.

21.3.3 A/D Control/Status Register for Unit 1 (ADCSR_1)	1032	Added
---	------	-------

Descriptions for bit 5 in the register table:

Note: Do not write to ADST when activation is by external trigger. For details, see section 21 on A/D activation by an External Trigger.

			external trigger. For details, see section 21.1 on A/D activation by an External Trigger.
21.3.6 A/D Control Register (ADCR_1) Unit 1	1038	Amended and added	Descriptions for bits 7,6, and 0 in the register table 001: External trigger disabled Note: Do not write to ADST when activation is by an external trigger. For details, see section 21.1 on A/D activation by an External Trigger.
21.3.7 A/D Control Register (ADCR_2) Unit 2	1040	Amended and added	Descriptions for bits 7,6, and 0 in the register table 001: External trigger disabled Note: Do not write to ADST when activation is by an external trigger. For details, see section 21.1 on A/D activation by an External Trigger.
21.7 Usage Notes	1053	Added	21.7.3 Notes on A/D Activation by an External Trigger
Section 22 D/A Converter	1064	Amended	
22.5.2 D/A Output Hold Function in Software Standby Mode			When this LSI make a transition to software standby with D/A conversion enabled, the D/A outputs are
22.5.3 Notes on Deep Software Standby Mode		Added	
Section 24 Flash Memory	1081	Added	
24.7.1 Programming/Erasing Interface Registers		Bit 0 R/W*	Note: This is a write-only bit. This bit is always read

5. Do not turn off the Vcc power supply nor remove the PROM programmer during programming in which a high voltage is applied to the flash memory. Doing so may damage the flash memory permanently. When a reset is input, the reset must be released after an input period of at least 100ms.
7. At powering on the Vcc power supply, fix the FLOW pin low and set the flash memory to hardware protection state. This power on timing must also be satisfied for power-off and power-on caused by a power failure or other factors.

Section 27 Power-Down Modes

1193

Deleted

Table 27.1, Notes:4

~~However, outputs from pins are disabled even if operation is selected.~~

27.2.4 Deep Standby Control Register (DPSBYCR)

1203
to
1205

Amended

DPSBYCR controls deep software standby mode. DPSBYCR is initialized by input of the reset signal at the RES pin, but is not initialized by the internal reset signal upon exit from deep software standby mode.

Amended

Descriptions for bit 7 in the register table:

When deep software standby mode is canceled by an interrupt, this bit remains at 1.

Amended

Descriptions for bit 4 in the register table:

RAMCUT 2, 1, and 0 control the internal power supply to the on-chip RAM in deep software standby mode.

DPSWCR selects the time for which the MCU waits for the clock to settle when deep software standby mode is exited by an external interrupt.

DPSWCR is initialized by input of the reset signal on the RES pin, but is not initialized by the internal reset signal upon exit from deep software standby mode.

Amended

Descriptions for bits 5 to 0 in the register table:

These bits select the time for which the MCU waits for the clock to settle when deep software standby mode is exited by an external interrupt.

27.2.6 Deep Standby
Interrupt Enable Register
(DPSIER) 1208,
1209

Amended

DPSIER enables or disables interrupts to clear deep software standby mode.

DPSIER is initialized by input of the reset signal on the RES pin, but is not initialized by the internal reset signal upon exit from deep software standby mode.

Amended

Descriptions for bit 3 in the register table:

Enables or disables exit from deep software standby mode by IRQ3-A.

0: Disables exit from deep software standby mode by IRQ3-A.

1: Enables exit from deep software standby mode by IRQ3-A.

Descriptions for bit 1 in the register table:

Enables or disables exit from deep software standby mode by **IRQ1-A**.

0: Disables exit from deep software standby mode by **IRQ1-A**.

1: Enables exit from deep software standby mode by **IRQ1-A**.

Amended

Descriptions for bit 0 in the register table:

Enables or disables exit from deep software standby mode by **IRQ0-A**.

0: Disables exit from deep software standby mode by **IRQ0-A**.

1: Enables exit from deep software standby mode by **IRQ0-A**.

Descriptions for bit 3 R/W in the register table
R/(W)^{*1}
IRQ3-A input specified in DPSIEGR is generated.

Descriptions for bit 2 R/W in the register table
R/(W)^{*1}
IRQ2-A input specified in DPSIEGR is generated.

Descriptions for bit 1 R/W in the register table
R/(W)^{*1}
IRQ1-A input specified in DPSIEGR is generated.

Descriptions for bit 0 R/W in the register table
R/(W)^{*1}
IRQ0-A input specified in DPSIEGR is generated.

IRQ2-A input specified in DPSIEGR is generated

Descriptions for bit 1 in the register table

IRQ1-A input specified in DPSIEGR is generated

Descriptions for bit 0 in the register table

IRQ0-A input specified in DPSIEGR is generated

27.2.9 Reset Status Register (RSTSR)	1213, 1214	<p>Amended</p> <p>The DPSRSTF bit in RSTSR indicates that deep standby mode has been canceled by an interrupt source specified in DPSIEGR and an internal reset is generated.</p> <p>RSTSR is not initialized by the internal reset signal upon exit from deep software standby mode.</p> <p>Descriptions for bit 7 in the register table</p> <p>Indicates that deep software standby mode has been canceled by an interrupt source specified in DPSIEGR and an internal reset is generated.</p> <p>[Setting condition]Deep software standby mode is canceled by an interrupt source.</p>
27.2.10 Deep Standby Backup Register (DPSBKRn)	1215	<p>Amended</p> <p>DPSBKRn (n = 15 to 0) is a 16-bit readable/writable register to store data during deep software standby mode.</p> <p>Although data in on-chip RAM is not retained in deep software standby mode, data in this register is retained.</p> <p>DPSBKRn (n=15 to 0) is not initialized by the internal reset signal upon exit from deep software standby mode.</p>

At power-on, the PLEB pin must be driven low with the pin driven high for a given time in order to clear the state.

27.11 B ϕ Clock Output Control*	1241	Added
--------------------------------------	------	-------

Table 27.4 ϕ Pin (PA7) State in Each Processing State

Break address mask register BH	BAMRBH	16	H'FFA0C	UBC
Break address mask register BL	BAMRBL	16	H'FFA0E	UBC
Break address register CH	BARCH	16	H'FFA10	UBC
Break address register CL	BARCL	16	H'FFA12	UBC
Break address mask register CH	BAMRCH	16	H'FFA14	UBC
Break address mask register CL	BAMRCL	16	H'FFA16	UBC
Break address register DH	BARDH	16	H'FFA18	UBC
Break address register DL	BARDL	16	H'FFA1A	UBC
Break address mask register DH	BAMRDH	16	H'FFA1C	UBC
Break address mask register DL	BAMRDL	16	H'FFA1E	UBC
Break control register A	BRCRA	16	H'FFA28	UBC
Break control register B	BRCRB	16	H'FFA2C	UBC
Break control register C	BRCRC	16	H'FFA30	UBC
Break control register D	BRCRD	16	H'FFA34	UBC

register 5					
Deep standby backup register 6	DPSBKR6	8	H'FFBF6	SYSTEM	8
Deep standby backup register 7	DPSBKR7	8	H'FFBF7	SYSTEM	8
Deep standby backup register 8	DPSBKR8	8	H'FFBF8	SYSTEM	8
Deep standby backup register 9	DPSBKR9	8	H'FFBF9	SYSTEM	8
Deep standby backup register 10	DPSBKR10	8	H'FFBFA	SYSTEM	8
Deep standby backup register 11	DPSBKR11	8	H'FFBFB	SYSTEM	8
Deep standby backup register 12	DPSBKR12	8	H'FFBFC	SYSTEM	8
Deep standby backup register 13	DPSBKR13	8	H'FFBFD	SYSTEM	8
Deep standby backup register 14	DPSBKR14	8	H'FFBFE	SYSTEM	8
Deep standby backup register 15	DPSBKR15	8	H'FFBFF	SYSTEM	8

	BAMRA	BAMRA	BAMRA	BAMRA	BAMRA	BAMRA	BAMRA	23
	15	14	13	12	11	10	9	8
BAMRAL	BAMRA	BAMRA	BAMRA	BAMRA	BAMRA	BAMRA	BAMRA	E
	7	6	5	4	3	2	1	0
	BARB3	BARB3	BARB2	BARB2	BARB2	BARB2	BARB2	E
	1	0	9	8	7	6	5	4
	BARB2	BARB2	BARB2	BARB2	BARB1	BARB1	BARB1	E
	3	2	1	0	9	8	7	6
	BARB1	BARB1	BARB1	BARB1	BARB1	BARB1	BARB1	E
	5	4	3	2	1	0		9
	BARB7	BARB6	BARB5	BARB4	BARB3	BARB2	BARB1	E
	BAMRB	BAMRB	BAMRB	BAMRB	BAMRB	BAMRB	BAMRB	E
	31	30	29	28	27	26	25	24
BAMRBH	BAMRB	BAMRB	BAMRB	BAMRB	BAMRB	BAMRB	BAMRB	E
	23	22	21	20	19	18	17	16
	BAMRB	BAMRB	BAMRB	BAMRB	BAMRB	BAMRB	BAMRB	E
	15	14	13	12	11	10	9	8
BAMRBL	BAMRB	BAMRB	BAMRB	BAMRB	BAMRB	BAMRB	BAMRB	E
	7	6	5	4	3	2	1	0
	BARC3	BARC3	BARC2	BARC2	BARC2	BARC2	BARC2	E
	1	0	9	8	7	6	5	4
	BARC2	BARC2	BARC2	BARC2	BARC1	BARC1	BARC1	E
	3	2	1	0	9	8	7	6

	C7	C6	C5	C4	C3	C2	C1	C0
BARDH	BARD 31	BARD 30	BARD 29	BARD 28	BARD 27	BARD 26	BARD 25	BARD 24
	BARD 23	BARD 22	BARD 21	BARD 20	BARD 19	BARD 18	BARD 17	BARD 16
BARDL	BARD 15	BARD 14	BARD 13	BARD 12	BARD 11	BARD 10	BARD 9	BARD 8
	BARD 7	BARD 6	BARD 5	BARD 4	BARD 3	BARD 2	BARD 1	BARD 0
BAMRH	BAMR D31	BAMR D30	BAMR D29	BAMR D28	BAMR D27	BAMR D26	BAMR D25	BAMR D24
	BAMR D23	BAMR D22	BAMR D21	BAMR D20	BAMR D19	BAMR D18	BAMR D17	BAMR D16
BAMRL	BAMR D15	BAMR D14	BAMR D13	BAMR D12	BAMR D11	BAMR D10	BAMR D9	BAMR D8
	BAMR D7	BAMR D6	BAMR D5	BAMR D4	BAMR D3	BAMR D2	BAMR D1	BAMR D0
BRCRA	—	—	CMFC PA	—	CPA2	CPA1	CPA0	—
	—	—	IDA1	IDA0	RWA1	RWA0	—	—
BRCRB	—	—	CMFC PB	—	CPB2	CPB1	CPB0	—
	—	—	IDB1	IDB0	RWB1	RWB0	—	—
BRCRC	—	—	CMFC PC	—	CPC2	CPC1	CPC0	—
	—	—	IDC1	IDC0	RWC1	RWC0	—	—
BRCRD	—	—	DMFC PD	—	CPD2	CPD1	CPD0	—
	—	—	IDD1	IDD0	RWD1	RWD0	—	—

DPSBK	BKUP6	BKUP6	BKUP6	BKUP6	BKUP6	BKUP6	BKUP6	BKUP6	BKUP6
R6	7	6	5	4	3	2	1	0	0
DPSBK	BKUP7	BKUP7	BKUP7	BKUP7	BKUP7	BKUP7	BKUP7	BKUP7	BKUP7
R7	7	6	5	4	3	2	1	0	0
DPSBK	BKUP8	BKUP8	BKUP8	BKUP8	BKUP8	BKUP8	BKUP8	BKUP8	BKUP8
R8	7	6	5	4	3	2	1	0	0
DPSBK	BKUP9	BKUP9	BKUP9	BKUP9	BKUP9	BKUP9	BKUP9	BKUP9	BKUP9
R9	7	6	5	4	3	2	1	0	0
DPSBK	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1
R10	07	06	05	04	03	02	01	00	00
DPSBK	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1
R11	17	16	15	14	13	12	11	10	10
DPSBK	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1
R12	27	26	25	24	23	22	21	20	20
DPSBK	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1
R13	37	36	35	34	33	32	31	30	30
DPSBK	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1
R14	47	46	45	44	43	42	41	40	40
DPSBK	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1	BKUP1
R15	57	56	55	54	53	52	51	50	50

1284 Amended

SCKCR	PSTO	PSTO		—	—	—	ICK2	ICK
	P1	P0* ³						
	—	PCK2	PCK1	PCK0	—	—	BCK2	BCK

28.3 Register States in Each Operating Mode 1293 Added

PORTM* ²		—	—	—	—	—	—
PORTN	—	—	—	—	—	—	

BARCL	Initialized	—	—	—	—	Initialized*1	Initializ
BAMRCH	Initialized	—	—	—	—	Initialized*1	Initializ
BAMRCL	Initialized	—	—	—	—	Initialized*1	Initializ
BARDH	Initialized	—	—	—	—	Initialized*1	Initializ
BARDL	Initialized	—	—	—	—	Initialized*1	Initializ
BAMRDH	Initialized	—	—	—	—	Initialized*1	Initializ
BAMRDL	Initialized	—	—	—	—	Initialized*1	Initializ
BRCRA	Initialized	—	—	—	—	Initialized*1	Initializ
BRCRB	Initialized	—	—	—	—	Initialized*1	Initializ
BRCRC	Initialized	—	—	—	—	Initialized*1	Initializ
BRCRD	Initialized	—	—	—	—	Initialized*1	Initializ

Note: * : Supported only by the H8SX/1648L Group.
H8SX/1648H Group.

All	—	Added
		Below are the sections added and those functions added to the corresponding sections due to the ac modules. (1) Section 5 Voltage Detection Circuit (LVD) (2) Section 11 EXDMA Controller (EXDMAC) (3) Section 17 32K Timer (TM32K)
Cover	i	Replaced
		Product names due to the addition of the H8SX/1648L, H8SX/1648G, and H8SX/1648H Groups
How to Use This Manual	v	Replaced
		Name of groups: H8SX/1648A, H8SX/1648L, H8SX/1648G, and H8SX/1648H Groups
Section 1 Overview	1	Replaced
1.1 Features		
1.1.2 Overview of Functions	2	Replaced
		Table 1.1 Overview of Functions
	10	Added
		Table 1.2 Comparison of Support Functions in the H8SX/1648, 1648A, 1648L, 1648G, and 1648H Groups
1.2 List of Products	12	Replaced
		Table 1.3 List of Products
		Figure 1.1 How to Read the Product Name Code
		<ul style="list-style-type: none"> • Small Package

1.4.3 Pin Functions	24	Replaced Table 1.5 Pin Functions
Section 3 MCU Operating Modes	77	Replaced Replaced due to the addition of the SDRAM inter
3.1 Operating Mode Selection		
3.2.1 Mode Control Register (MDCR)	79	Replaced
3.3.8 Pin Functions	85	Replaced Table 3.4 Pin Functions in Each Operating Mode (Advanced Mode)
3.4.1 Address Map	86 to 91	Replaced Figure 3.1 Address Map in Each Operating Mode H8SX/1648, 1648A, 1648L, 1648G, and 1648H (Figure 3.1 Address Map in Each Operating Mode H8SX/1648, 1648A, 1648L, 1648G, and 1648H (Figure 3.2 Address Map in Each Operating Mode H8SX/1644, 1644A, 1644L, 1644G, and 1644H (Figure 3.2 Address Map in Each Operating Mode H8SX/1644, 1644A, 1644L, 1644G, and 1644H (Figure 3.3 Address Map in Each Operating Mode H8SX/1642, 1642A, 1642L, 1642G, and 1642H (Figure 3.3 Address Map in Each Operating Mode H8SX/1642, 1642A, 1642L, 1642G, and 1642H (

4.3 Power-On Reset (POR) (H8SX/1648L Group and H8SX/1648H Group)	99	Added Added due to the addition of the power-on reset.
4.6 Power Supply Monitoring Reset (H8SX/1648L Group, H8SX/1648H Group)	100	Added Added due to the addition of the power supply mo reset.
4.9 Determination of Reset Generation Source	101	Replaced Replaced due to the addition of the power supply monitoring reset.
Section 6 Exception Handling	115	Replaced
6.3.3 On-Chip Peripheral Functions after Reset Release		Replaced due to the addition of the EXDMAC.
6.5.1 Address Error Source	118	Replaced Table 6.5 Bus Cycle and Address Error Replaced due to the addition of the EXDMAC.
6.5.2 Address Error Exception Handling	119	Replaced Replaced due to the addition of the EXDMAC.
6.6.1 Interrupt Sources	121	Replaced Table 6.7 Interrupt Sources Replaced due to the addition of the LVD, EXDMA TM32K.
Section 7 Interrupt Controller	127	Replaced
7.1 Features		CPU priority control function Replaced due to the addition of the EXDMAC.

H8SX/1648 Group,
H8SX/1648A Group and
H8SX/1648L Group:

Interrupt Priority Registers A
to I, K to O, Q, and R

(IPRA to IPRI, IPRK to IPRO,
IPRQ, and IPRR)

H8SX/1648G Group, and
H8SX/1648H Group:

Interrupt Priority Registers A
to O, Q, and R

(IPRA to IPRO, IPRQ, and
IPRR)

7.3.4 IRQ Enable Register (IER)	136, 137	Replaced Replaced due to the addition of the LVD and TM32K.
7.3.5 IRQ Sense Control Registers H and L (ISCRH, ISCLR)	138 to 143	Replaced Replaced due to the addition of the LVD and TM32K.
7.3.6 IRQ Status Register (ISR)	143 to 146	Replaced Replaced due to the addition of the LVD and TM32K.
7.4.1 External Interrupts	148	Replaced (1) NMI Interrupts Replaced due to the addition of the EXDMAC.
7.5 Interrupt Exception Handling Vector Table	150 to 155	Replaced Table 7.2 Interrupt Sources, Vector Address Offset, and Interrupt Priority Replaced due to the addition of the LVD, EXDMAC, and TM32K.

		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)
		Replaced due to the addition of the DRAM interface SDRAM interface.
9.2.3 Wait Control Registers A and B (WTCRA, WTCRB)	193	Replaced WTCRB
9.2.4 Read Strobe Timing Control Register (RDNCR)	195	Replaced Notes: 2. Replaced due to the addition of the EXDMAC.
9.2.5 \overline{CS} Assertion Period Control Registers (CSACR)	196	Replaced Notes: 2. Replaced due to the addition of the EXDMAC.
9.2.6 Idle Control Register (IDLCR)	200	Replaced Description of bit 15 Replaced due to the addition of the EXDMAC.
9.2.7 Bus Control Register 1 (BCR1)	202	Added Bit 6 added Added due to the addition of the EXDMAC.

			SDRAM interface.
9.2.14	DRAM Access Control Register (DRACCR)	214	Added Added due to the addition of the DRAM interface SDRAM interface.
9.2.15	Synchronous DRAM Control Register (SDCR)	215	Added Added due to the addition of the SDRAM interface
9.2.16	Refresh Control Register (REFCR)	216	Added Added due to the addition of the DRAM interface SDRAM interface.
9.2.17	Refresh Timer Counter (RTCNT)	220	Added Added due to the addition of the DRAM interface SDRAM interface.
9.2.18	Refresh Time Constant Register (RTCOR)		Added Added due to the addition of the DRAM interface SDRAM interface.
9.3	Bus Configuration	221	Replaced Replaced due to the addition of the EXDMAC.
9.4	Multi-Clock Function and Number of Access Cycles	222	Replaced Synchronization Clocks and Their Corresponding Replaced due to the addition of the EXDMAC.
9.5.1	Input/Output Pins	226 to 229	Replaced Table 9.2 Pin Configuration Table 9.3 Pin Functions in Each Interface
9.5.4	External Bus Interface	233	Replaced Table 9.4 Interface Names and Area Names Table 9.5 Areas Specifiable for Each Interface

9.6.3 Basic Timing	248 to 253	<p>Replaced</p> <p>Figure 9.15 16-Bit 2-State Access Space Bus Tim Access for Even Address)</p> <p>Figure 9.16 16-Bit 2-State Access Space Bus Tim Access for Odd Address)</p> <p>Figure 9.17 16-Bit 2-State Access Space Bus Tim (Word Access for Even Address)</p> <p>Figure 9.18 16-Bit 3-State Access Space Bus Tim Access for Even Address)</p> <p>Figure 9.19 6-Bit 3-State Access Space Bus Tim Access for Odd Address)\</p> <p>Figure 9.20 16-Bit 3-State Access Space Bus Tim (Word Access for Even Address)</p> <p>Replaced due to the addition of the EXDMAC.</p>
9.6.5 Read Strobe (RD) Timing	256	<p>Replaced</p> <p>Replaced due to the addition of the EXDMAC.</p>
9.6.6 Extension of Chip Select (CS) Assertion Period	258	<p>Replaced</p> <p>Figure 9.23 Example of Timing when Chip Select Period is Extended</p> <p>Replaced due to the addition of the EXDMAC.</p>
9.6.7 $\overline{\text{DACK}}$ and EDACK* Signal Output Timing	260	<p>Replaced</p> <p>Replaced due to the addition of the EXDMAC.</p>

Signal Output Timing		Replaced due to the addition of the EXDMAC
9.8.4 Basic Timing	272	Replaced Replaced due to the addition of the EXDMAC
9.8.6 Read Strobe (RD) Timing	274	Replaced Replaced due to the addition of the EXDMAC
9.8.7 Extension of Chip Select (CS) Assertion Period		Replaced Replaced due to the addition of the EXDMAC
9.9.5 Basic Timing	277, 278	Replaced Figure 9.31 8-Bit Access Space Access Timing (ABWHn = 1, ABWLn = 1) Figure 9.32 16-Bit Access Space Access Timing (ABWHn = 0, ABWLn = 1) Replaced due to the addition of the EXDMAC
9.9.6 Address Cycle Control	279	Replaced Figure 9.33 Access Timing of 3 Address Cycles (ADDEX = 1)
9.9.8 Read Strobe (RD) Timing	281	Replaced Figure 9.34 Read Strobe Timing Replaced due to the addition of the EXDMAC
9.9.9 Extension of Chip Select (\overline{CS}) Assertion Period	282	Replaced Figure 9.35 Chip Select (CS) Assertion Period Extension Timing in Data Cycle
9.9.10 \overline{DACK} and \overline{EDACK}^* Signal Output Timing	284	Replaced Replaced due to the addition of the EXDMAC
9.10 DRAM Interface	285	Added Added due to the addition of the SDRAM interface

9.12.2 Pin States in Idle Cycle	365	Replaced Table 9.31 Pin States in Idle Cycle
9.13 Bus Release	366	Replaced
9.13.1 Operation		Replaced due to the addition of the DRAM interface, SDRAM interface, and EXDMAC.
9.13.2 Pin States in External Bus Released State	368	Replaced Table 9.32 Pin States in Bus Released State
9.13.3 Transition Timing	370	Added Figure 9.103 Bus Released State Transition Timing (Interface is Used)
9.14 Internal Bus	372	Replaced
9.14.1 Access to Internal Address Space		Table 9.34 Number of Access Cycles for Registers and Chip Peripheral Modules Replaced due to the Port M, EXDMAC, and TM32
9.16 Bus Arbitration	375	Replaced Replaced due to the addition of the DRAM interface, SDRAM interface, and EXDMAC.
9.16.1 Operation		Replaced Replaced due to the addition of the DRAM interface, SDRAM interface, and EXDMAC.
9.16.2 Bus Transfer Timing	376	Replaced (1) CPU Replaced due to the addition of the EXDMAC.
	377	Added (4) EXDMAC (6) Refresh

Section 10. DMA Controller (DMAC)	456	Replaced
10.7.2 Bus Arbitration among DMAC and Other Bus Masters		Replaced due to the addition of the EXDMAC.
Section 12. Data Transfer Controller (DTC)	579	Replaced
12.4 Location of Transfer Information and DTC Vector Table		Table 12.1 Interrupt Sources, DTC Vector Address Corresponding DTCEs
Section 13. I/O Ports	603	Replaced
		Table 13.1 Port Functions
		Replaced due to the addition of the EXDMAC and
13.1.1 Data Direction Register (PnDDR) (n = 1, 2, 3, 6, A, to F, H to K, M*, and N)	614	Replaced
		Replaced due to the addition of the Port M.
13.1.2 Data Register (PnDR) (n = 1, 2, 3, 6, A, to F, H to K, M*, and N)	615	Replaced
		Replaced due to the addition of the Port M.
13.1.3 Port Register (PORTn) (n = 1 to 6, A to F, H to K, M*, and N)		Replaced
		Replaced due to the addition of the Port M.
13.1.4 Input Buffer Control Register (PnICR) (n = 1 to 6, A to F, H to K, M*, and N)	616	Replaced
		Replaced due to the addition of the Port M.

		SDRAM interface.
13.2.8 Port C	647	Replaced Replaced due to the addition of the DRAM interface and SDRAM interface.
13.2.16 Port M*	666	Added Added due to the addition of the Port M.
13.2 Output Buffer Control	670	Replaced Table 13.5 Available Output Signals and Settings Port Replaced due to the addition of the EXDMAC, DRAM interface and SDRAM interface.
13.3 Port Function Controller	681	Added Added due to the addition of the EXDMAC.
13.3.7 Port Function Control Register 8 (PFCR8)*	689	Added Added due to the addition of the EXDMAC.
13.3.10 Port Function Control Register B (PFCRB)	694	Replaced Replaced due to the addition of the TM32K and LV
23. RAM	1065	Replaced Replaced due to the addition of the H8SX/1648A, H8SX/1648L, H8SX/1648G, and H8SX/1648H Gr
24. Flash Memory	1067	Replaced
24.1 Features		<ul style="list-style-type: none"> • ROM size Replaced due to the addition of the H8SX/1648A, H8SX/1648L, H8SX/1648G, and H8SX/1648H Gr

Register (SCRCR)		Replaced due to the addition of the subclock oscillator TM32K.
26.1.2 Subclock Control Register (SUBCKCR)*	1181	Added Added due to the addition of the subclock oscillator
26.5 Subclock Oscillator	1185	Added Added due to the addition of the subclock oscillator
Section 27. Power-Down Modes 27.1 Features	1192 to 1194	Replaced <ul style="list-style-type: none"> Multi-clock function Table 27.1 States of Operation Figure 27.1 (1) Mode Transitions (H8SX/1648 Group, H8SX/1648A Group, and H8SX/1648L Group) Replaced due to the addition of the subclock oscillator TM32K, and LVD.
	1195	Added Figure 27.1 (2) Mode Transitions (H8SX/1648G Group, H8SX/1648H Group)
27.2.1 Standby Control Register (SBYCR)	1196	Replaced Replaced due to the addition of the subclock oscillator
27.2.2 Module Stop Control Registers A and B (MSTPCRA and MSTPCRB)	1199	Replaced <ul style="list-style-type: none"> MSTPCRA Replaced due to the addition of the EXDMAC.
27.2.6 Deep Standby Interrupt Enable Register (DPSIER)	1208	Added Bit 5 and Bit 4 Added due to the addition of the TM32K and LVD.
27.2.7 Deep Standby Interrupt Flag Register (DPSIFR)	1210	Replaced Replaced due to the addition of the TM32K and LVD.

27.5.2 Exit from Sleep Mode		Replaced due to the addition of the EXDMAC. Replaced
27.6 All-Module-Clock-Stop Mode	1218	Replaced due to the addition of the LVD and power-on reset. Replaced
27.7.2 Exit from Software Standby Mode	1219	Replaced due to the addition of the LVD, TM32K, power-on reset. Replaced
		1. Exit from software standby mode by interrupt
		Replaced due to the addition of the 32K timer, voltage monitoring reset, and power-on reset.
		Added
		2. Exit from voltage monitoring reset* ³
		3. Exit from power-on reset* ³
		Added due to the addition of the voltage monitoring and power on reset.
27.8.1 Entry to Deep Software Standby Mode	1223	Replaced
		Replaced due to the addition of the LVD and TM32K.

27.8.4 B ϕ or SDRAM ϕ * Operation after Exit from Deep Software Standby Mode	1227	Replaced Replaced due to the addition of the SDRAM ϕ
27.9.4 Timing Sequence at Power-On	1237	Replaced Replaced due to the addition of the power-on res
27.11 ϕ Clock Output Control*	1241	Replaced Replaced due to the addition of the SDRAM ϕ .
27.12.3 Module Stop State of DMAC, EXDMAC*, or DTC	1242	Replaced Replaced due to the addition of the EXDMAC.
27.12.7 Conflict between a transition to deep software standby mode and interrupts	1243	Replaced Replaced due to the addition of the 32K timer and monitoring interrupt
27.12.8 B ϕ or SDRAM ϕ * Output State		Replaced Replaced due to the SDRAM ϕ

Port N data register

Added due to the erroneous omission of information

Added

Timer control register

Timer counter 1

Timer counter 2

Timer counter3

Added due to the addition of the TM32K.

Added

Port function control register 8*2

Added due to the addition of the EXDMAC.

EXDMA offset register_1*²
EXDMA transfer count register_1*²
EXDMA block size register_1*²
EXDMA mode control register_1*²
EXDMA address control register_1*²
EXDMA source address register_2*²
EXDMA destination address register_2*²
EXDMA offset register_2*²
EXDMA transfer count register_2*²
EXDMA block size register_2*²
EXDMA mode control register_2*²
EXDMA address control register_2*²
EXDMA source address register_3*²
EXDMA destination address register_3*²
EXDMA offset register_3*²
EXDMA transfer count register_3*²
EXDMA block size register_3*²
EXDMA mode control register_3*²
EXDMA address control register_3*²
Added due to the addition of the EXDMAC.

		SDRAM interface.
		Added
		Low voltage detection control register* ³
		Added due to the addition of the LVD.
28.2 Register Bits	1266	Added
		PMDDR* ³
		PMDR* ³
		PORTM* ³
		PMICR* ³
		Added due to the addition of the Port M.
		Added
		PNDDR
		PNDR
		Added due to the erroneous omission of information.
	1269	Added
		TCR32K* ³ , TCNT32K1* ³ , TCNT32K2* ³ , TCNT32K3* ³
		Added due to the TM32K.
	1273	Added
		PFCR8* ³
		Added due to the addition of the EXDMAC.

CLSBR0*³, CLSBR1*³, CLSBR2*³, CLSBR3*³
CLSBR4*³, CLSBR5*³, CLSBR6*³, CLSBR7*³
Added due to the addition of the EXDMAC.

1281, Replaced
1282 IPRE
Added
IPRJ
Replaced
IPRK

1283, Replaced
1284 BCR1
BCR2
Added
DRAMCR*³, DRACCR*³, SDCR*³, REFCR*³, RT
RTCOR*³
Added due to the addition of the DRAM interface
SDRAM interface.
Replaced
MDCR
SCKCR

1284 Replaced
RSTSR
Added
LVDCR*⁴
Added due to the addition of the LVD.

1296	TCR32K* ² , TCNT32K1* ² , TCNT32K2* ² , TCNT32K3* ² Due to the addition of the TM32K.
1299	Added PFCR8* ² Due to the addition of the EXDMAC.
1301 to 1302	Added EDSAR_0* ² , EDDAR_0* ² , EDOFR_0* ² , EDTCR_0* ² , EDBSR_0* ² , EDMDR_0* ² , EDACR_0* ² , EDSAR_1* ² , EDDAR_1* ² , EDOFR_1* ² , EDTCR_1* ² , EDBSR_1* ² , EDMDR_1* ² , EDACR_1* ² , EDSAR_2* ² , EDDAR_2* ² , EDOFR_2* ² , EDTCR_2* ² , EDBSR_2* ² , EDMDR_2* ² , EDACR_2* ² , EDSAR_3* ² , EDDAR_3* ² , EDOFR_3* ² , EDTCR_3* ² , EDBSR_3* ² , EDMDR_3* ² , EDACR_3* ² , CLSBR0* ² , CLSBR1* ² , CLSBR2* ² , CLSBR3* ² , CLSBR4* ² , CLSBR5* ² , CLSBR6* ² , CLSBR7* ² Added due to the addition of the EXDMAC.
1303	Added IPRJ

29.1 Absolute Maximum Ratings

Table 29.1 Absolute Maximum Ratings

29.2 DC Characteristics, H8SX/1648 Group, H8SX/1648A Group, and H8SX/1648G Group	1314	Replaced Replaced due to the addition of the H8SX/1648A and H8SX/1648G Group.
29.3 DC Characteristics H8SX/1648L Group and H8SX/1648H Group	1318	Added Added due to the addition of the H8SX/1648L Group and H8SX/1648H Group.
29.4.4 Timing of DMAC and EXDMAC	1356	Replaced Table 29.9 Timing of DMAC and EXDMAC Figure 29.37 Timing of \overline{DREQ} and \overline{EDREQ}^* input (DMAC and EXDMAC*) Figure 29.38 Timing of \overline{TEND} and \overline{ETEND}^* input (DMAC and EXDMAC*) Figure 29.39 Single Address Transfer Timing for DMAC and EXDMAC* (Two-State Access) Figure 29.40 Single Address Transfer Timing for DMAC and EXDMAC* (Three-State Access) Figure 29.41 Timing of EXDMAC and EDRAK Output (DMAC and EXDMAC*) Timing* Replaced due to the addition of the EXDMAC.
29.4.5 Timing of On-Chip Peripheral Modules	1361	Replaced Figure 29.42 Timing of I/O Port Input/Output3 Replaced due to the addition of the port M.

Circuit and Voltage-Detection
Circuit Characteristics
(H8SX/1648L Group,
H8SX/1648H Group)

Added due to the addition of the power-on reset a

Appendix	1381	Replaced
B. Product Lineup		Replaced due to the addition of the H8SX/1648A (H8SX/1648G Group, H8SX/1648L Group, and H8SX/1648H Group.
C. Package Dimensions	1382	Added Figure C.1 Package Dimensions (BP-176V)
D. Treatment of Unused Pins	1384	Replaced

8-bit timers (TMK) 829

A

A/D conversion accuracy.....	1051
Absolute accuracy.....	1051
Acknowledge.....	1004
Address error.....	118
Address map.....	85
Address mode.....	539
Address modes.....	410, 490
Address/data multiplexed I/O interface.....	236, 275
All-module-clock-stop mode.....	1192, 1218
Area 0.....	238
Area 1.....	239
Area 2.....	239
Area 3.....	240
Area 4.....	241
Area 5.....	242
Area 6.....	242
Area 7.....	243
Area division.....	230
Asynchronous mode.....	933
AT-cut parallel-resonance type.....	1183
Average transfer rate generator.....	886

Boot mode.....	
Boundary scan commands.....	
Buffer operation.....	
Burst access mode.....	
Burst mode.....	
Burst ROM interface.....	
Bus access modes.....	
Bus arbitration.....	
Bus configuration.....	
Bus controller (BSC).....	
Bus cycle division.....	
Bus mode.....	
Bus release.....	
Bus width.....	
Bus-released state.....	
Byte control SRAM interface.....	

C

Cascaded connection.....	
Cascaded operation.....	
Chain transfer.....	
Chip select signals.....	
Clock pulse generator.....	
Clock synchronization cycle (Tsy).....	
Clocked synchronous mode.....	
Cluster transfer dual address mode.....	
Cluster transfer modes.....	
Cluster transfer read address mode.....	

Cycle steal mode.....	499
Cycle stealing mode.....	421

D

D/A converter	1059
Data direction register	614
Data register.....	615
Data transfer controller (DTC)	567
Direct convention	958
DMA controller (DMAC).....	383
Double-buffered structure.....	933
Download pass/fail result parameter.....	1088
DRAM interface	285
DRAM Interface.....	236
DTC vector address	579
DTC vector address offset.....	579
Dual address mode.....	410, 490

E

Endian and data alignment	244
Endian format	234
Error protection	1124
Error signal	958
Exception handling.....	111
Exception-handling state	76
EXDMA controller (EXDMAC)	461
Extended repeat area.....	407

F

Flash erase block select parameter.....	
Flash memory	
Flash multipurpose address area parameter	
Flash multipurpose data destination parameter	
Flash pass and fail parameter	
Flash program/erase frequency parameter	10
Free-running count operation.....	
Frequency divider	1
Full address mode	
Full-scale error	

G

General illegal instructions	
------------------------------------	--

H

Hardware protection	
Hardware standby mode	1

I

I/O ports.....	
I ² C bus format	

Interrupt control mode 2	158
Interrupt controller.....	127
Interrupt exception handling sequence	160
Interrupt exception handling vector table	150
Interrupt response times.....	161
Interrupt sources	148
Interrupt sources and vector address offsets	150
Interval timer	880
Interval timer mode.....	880
Inverse convention.....	959
IRQn interrupts	148
J	
JTAG interface	1021
L	
Little endian.....	234
M	
Mark state	933, 974
Master receive mode.....	1007
Master transmit mode	1005
MCU operating modes.....	77

N

NMI interrupt.....	
Noise canceler.....	
Nonlinearity error.....	
Non-overlapping pulse output.....	
Normal transfer mode	
Normal transfer mode	
Number of access cycles	

O

Offset addition	
Offset addition method	
Offset error.....	
On-board programming	
On-board programming mode.....	
On-chip baud rate generator.....	
On-chip ROM disabled extended r.....	
On-chip ROM enabled extended r.....	
Open-drain control register	
Oscillator.....	
Output buffer control	
Output trigger.....	
Overflow	

Power-down modes	1191
Procedure program	1117
Processing states	76
Product lineup.....	1381
Program execution state.....	76
Program stop state	76
Programmable pulse generator (PPG)	793
Programmer mode	1070, 1130
Programming/erasing interface.....	1076
Programming/erasing interface parameters.....	1086
Programming/erasing interface register	1079
Protection.....	1123
Pull-up MOS control register.....	617
PWM modes	762

Q

Quantization error.....	1051
-------------------------	------

R

RAM	1065
Read strobe (\overline{RD}) timing.....	256
Register addresses	1246
Register Bits	1264
Register configuration in each port.....	613

CLSBR.....	131, 1259, 1260
CPUPCR.....	131, 1259, 1260
CRA	131, 1259, 1260
CRB	131, 1259, 1260
CRCCR.....	980, 1246, 1247
CRCDIR	981, 1246, 1247
CRCDOR.....	981, 1246, 1247
CSACR	196, 1256, 1257
DACR	402, 1253, 1254
DACR01	1061, 1260, 1261
DADR0	1060, 1260, 1261
DADR1	1060, 1260, 1261
DAR.....	1061, 1260, 1261
DBSR.....	392, 1253, 1254
DDAR.....	389, 1253, 1254
DDR.....	614, 1253, 1254
DMDR	393, 1253, 1254
DMRSR	408, 1255, 1256
DOFR.....	390, 1253, 1254
DPFR	390, 1253, 1254
DPSBYCR.....	390, 1253, 1254
DPSIEGR.....	390, 1253, 1254
DPSIER.....	390, 1253, 1254
DPSIFR.....	390, 1253, 1254
DPSWCR.....	390, 1253, 1254
DR.....	615, 1260, 1261
DRACCR.....	615, 1260, 1261
DRAMCR	615, 1260, 1261
DSAR.....	388, 1253, 1254

ENDIANCR.....	204, 1256, 1283, 1304	NDERL	799, 1260,
EXR.....	46	NDRH	804, 1261,
FCCS	1079, 1257, 1284, 1304	NDRL.....	804, 1261,
FEBS.....	1097	ODR.....	618, 1252,
FECS.....	1082, 1257, 1284, 1304	PC	
FKEY.....	1083, 1257, 1284, 1304	PCR.....	
FMATS.....	1084	PCR (I/O port).....	617, 1252,
FMPAR.....	1095	PCR (PPG).....	1260,
FMPDR.....	1096	PFCR0.....	682, 1252,
FPCS.....	1082, 1257, 1284, 1304	PFCR1.....	682, 1252,
FPEFEQ.....	1093, 1107	PFCR2.....	684, 1253,
FPFR.....	1089	PFCR4.....	685, 1253,
FTDAR.....	1085, 1257, 1284, 1304	PFCR6.....	687, 1253,
General registers	43	PFCR7.....	688, 1253,
ICCRA.....	992, 1258, 1285, 1305	PFCR8.....	
ICCRB.....	993, 1258, 1285, 1305	PFCR9.....	689, 690, 1253,
ICDRR.....	1003, 1258, 1285, 1305	PFCRB.....	694, 1253,
ICDRS	1003	PFCRC	697, 1253,
ICDRT	1003, 1258, 1285, 1305	PMR	811, 1260,
ICIER.....	996, 1258, 1285, 1305	PODRH.....	801, 1260,
ICMR.....	995, 1258, 1285, 1305	PODRL	801, 1261,
ICR	616, 1251, 1271, 1298	PORT	615, 1259,
ICSR	999, 1258, 1285, 1305	RAMER	1098, 1257,
IDLCR.....	199, 1256, 1283, 1304	RDNCR.....	195, 1256,
IER.....	136, 1259, 1287, 1307	RDR	894, 1261,
INTCR.....	130, 1259, 1287, 1307	REFCR.....	
IPR.....	134, 1255, 1281, 1303	RSR.....	
IrCR.....	931, 1247, 1266, 1294	RSTCSR.....	877, 1261,
ISCRH.....	138, 1256, 1282, 1303	RSTSR	

SDID.....	1172
SEMR.....	922, 1257, 1284, 1305
SMR.....	895, 1261, 1288, 1309
SRAMCR.....	205, 1256, 1283, 1304
SSIER.....	147, 1253, 1273, 1300
SSR.....	905, 1261, 1288, 1309
SYSCR.....	81, 1257, 1284, 1304
TCCR.....	840, 1262, 1289, 1310
TCNT.....	745
TCNT (TMR).....	837, 1262, 1289, 1310
TCNT (TPU).....	1262, 1290, 1310
TCNT (WDT).....	875, 1261, 1289, 1309
TCORA.....	837, 1262, 1289, 1310
TCORB.....	838, 1262, 1289, 1310
TCR.....	715
TCR (TMR).....	838, 1261, 1289, 1309
TCR (TPU).....	1262, 1290, 1310
TCR32K.....	864
TCSR (TMR).....	845, 1261, 1289, 1309
TCSR (WDT).....	875, 1261, 1289, 1309
TDR.....	894, 1261, 1288, 1309
TGR.....	745, 1262, 1290, 1310
TIER.....	740, 1262, 1290, 1310
TIOR.....	722, 1262, 1290, 1310
TMDR.....	720, 1262, 1290, 1310
TSR.....	741, 895
TSR (TPU).....	1262, 1290, 1310
TSTR.....	746, 1262, 1290, 1310
TSYR.....	747, 1262, 1290, 1310

Scan mode.....	
SDRAM interface.....	
Serial communication interface (SC)	
Short address mode.....	
Single address mode.....	
Single mode.....	
Slave receive mode.....	
Slave transmit mode.....	
Sleep instruction exception handling	
Sleep mode.....	1
Slot illegal instructions.....	
Smart card interface.....	
Software protection.....	
Software standby mode.....	1
Space state.....	
Stack status after exception handling	
Standard serial communication inter	
specifications for boot mode.....	
Start bit.....	
State transition of TAP controller..	
State transitions.....	
Stop bit.....	
Strobe assert/negate timing.....	
Synchronous clearing.....	
Synchronous DRAM interface.....	
Synchronous operation.....	
Synchronous presetting.....	
System clock (I ϕ).....	2

Trap instruction exception handling 122

Write data buffer function for external
data bus
Write data buffer function for
peripheral modules.....

U

User boot MAT..... 1072
User boot mode..... 1070, 1113
User break controller (UBC) 171
User MAT..... 1072
User program mode 1070, 1103

**Renesas 32-Bit CISC Microcomputer
Hardware Manual
H8SX/1648, H8SX/1648A, H8SX/1648L, H8SX/1648G,
H8SX/1648H Group**

Publication Date: Rev.1.00, Sep. 13, 2007
Rev.2.00, Jul. 31, 2008

Published by: Sales Strategic Planning Div.
Renesas Technology Corp.

Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

© 2008. Renesas Technology Corp., All rights reserved. Printed in Japan.



RENESAS SALES OFFICES

<http://www.ren>

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, AZIA Center, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.

7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.

10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

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, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, M
Tel: <603> 7955-9390, Fax: <603> 7955-9510



H8SX/1648, H8SX/1648A, H8SX/1648L,
H8SX/1648G, H8SX/1648H Group
Hardware Manual



Renesas Electronics Corporation

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0365-0200

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [renesas](#) manufacturer:

Other Similar products are found below :

[EL4511CUZ-T7](#) [PYB15-Q24-S5-H-U](#) [PQA30-D24-S24-DH](#) [PQA30-D48-S12-TH](#) [PYB30-Q24-T312-H-U](#) [PYB15-Q24-S5-H-T](#) [PYB15-Q24-S12-H-T](#) [V7815-500-SMT](#) [PYB20-Q48-S12-H-T](#) [PQZ6-Q24-S15-D](#) [PYB20-Q48-S5-H-T](#) [PYB20-Q24-S12-H-T](#) [VLED15-120-350](#) [VGS-75-12](#) [PYB15-Q24-S12-H-U](#) [VGS-50-15](#) [VGS-50-24](#) [R5F100GFAFB#V0](#) [VGS-25-24](#) [VGS-50-5](#) [VGS-100-12](#) [M30620FCAFP#U3](#) [ETSA120500UD-P5P-SZ](#) [PDQ2-D24-S12-S](#) [PDS1-S12-D12-M](#) [PDS1-S12-D15-M](#) [PYB15-Q24-S12-T](#) [PYB15-Q24-S24](#) [PYB20-Q24-S24-DIN](#) [PYB20-Q48-S12](#) [R0K33062PS000BE](#) [R0K505220S000BE](#) [R0K561664S000BE](#) [R0K570865S000BE](#) [HC55185AIMZ](#) [R5F72145BDFA#V1](#) [R5S72631P200FP](#) [R7S721001VCBGAC0](#) [EPSA050250UB-P5P-EJ](#) [HS0005PUU01H](#) [IS82C55A-5](#) [ISL29035EVAL1Z](#) [ISL55110IVZ](#) [ISL6730AEVAL1Z](#) [ISL68200DEMO1Z](#) [ISL78235EVAL2Z](#) [ISL78268EVAL1Z](#) [ISL91107IRA-EVZ](#) [ISL9220IRTZEVAL1Z](#) [ISL95870BIRZ](#)