REJ09B0467-0100

Everywhere you imagine. RENESAS

16

H8S/2456, H8S/2456R, H8S/2454 Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer
H8S Family / H8S/2400 Series

H8S/2456	R4F2456
	R4S2456
H8S/2456R	R4F2456R
	R4S2456R
H8S/2454	R4F2454
	R4S2454

Rev.1.00
Revision Date: Sep. 19, 2008

RenesasTechnology
www.renesas.com

Hardware Manual

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed 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 and 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 above.
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 life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect 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 specific 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 injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or 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 very high. You should implement safety measures so that Renesas products may not be easily detached from your 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.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions may occur due to the false recognition of the pin state as an input signal. 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-on reset 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 stabilization 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 part number, confirm that the change will not lead to problems.

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

How to Use This Manual

1. Objective and Target Users

This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users, i.e. those who will be using this LSI in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.

This manual is organized in the following items: an overview of the product, descriptions of the CPU, system control functions, and peripheral functions, electrical characteristics of the device, and usage notes.

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, and 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 locations in the manual.

The following documents have been prepared for the H8S/2456, H8S/2456R, H8S/2454 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 No.
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	H8S/2456, H8S/2456R H8S/2454 Group Hardware Manual	This manual
Software Manual	Detailed descriptions of the CPU and instruction set	H8S/2600 Series H8S/2000 Series Software Manual	REJ09B0139
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.		

2. Description of Numbers and Symbols

Aspects of the notations for register names, bit names, numbers, and symbolic names in this manual are explained below.

(1) Overall notation

In descriptions involving the names of bits and bit fields within this manual, the modules and registers to which the bits belong may be clarified by giving the names in the forms "module name"."register name"."bit name" or "register name"."bit name".

(2) Register notation

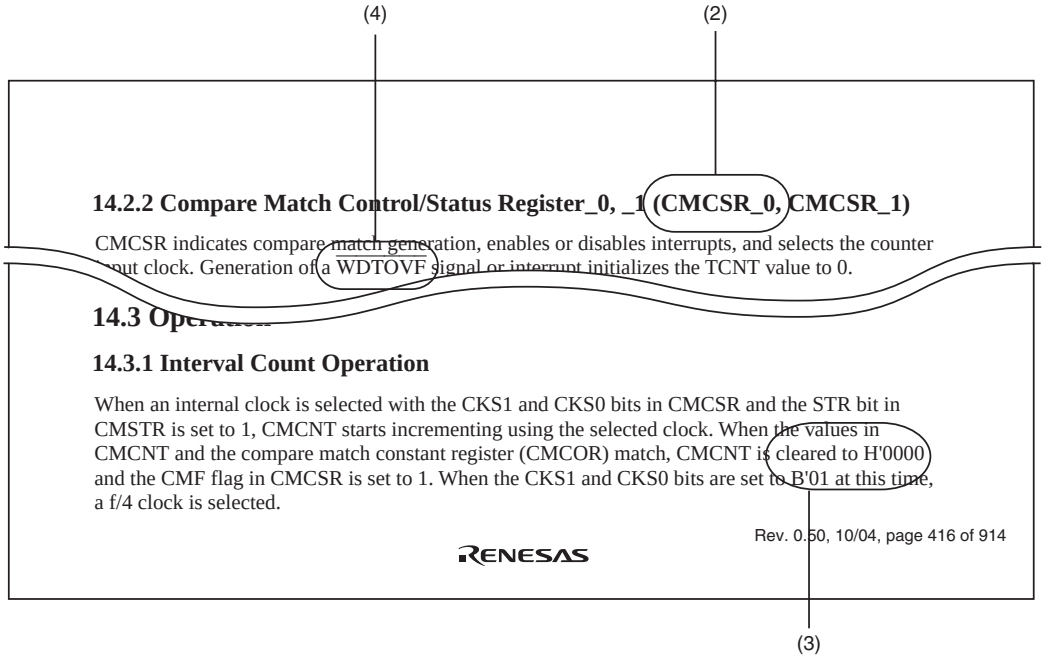
The style "register name"_"instance number" is used in cases where there is more than one instance of the same function or similar functions.
[Example] CMCSR_0: Indicates the CMCSR register for the compare-match timer of channel 0.

(3) Number notation

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 0xnnnn, 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] $\overline{\text{WDTOVF}}$



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

3. Description of Registers

Each register description includes a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meanings of the bit settings. The standard format and notation for bit charts and tables are described below.

[Table of Bits]

Bit	Bit Name	Initial Value	R/W	Description
15	–	0	R	Reserved
14	–	0	R	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 contents of this 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.

4. Description of Abbreviations

The abbreviations used in this manual are listed below.

- Abbreviations specific to this product

Abbreviation	Description
BSC	Bus controller
CPG	Clock pulse generator
INT	Interrupt controller
SCI	Serial communication interface
TMR	8-bit timer
TPU	16-bit timer pulse unit
WDT	Watchdog timer

- Abbreviations other than those listed above

Abbreviation	Description
ACIA	Asynchronous communication 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

Contents

Section 1 Overview

1.1 Features

The H8S/2456 Group, H8S/2454 Group, and H8S/2456R Group are CISC (Complex Instruction Set Computer) microprocessors that integrate an H8S/2600 CPU core which has an internal 16-bit architecture and is upward-compatible with Renesas Technology original H8/300, H8/300H, and H8S CPUs.

The on-chip peripheral functions provided for enabling system configuration at a low cost are the DMA controller, EXDMA controller*, data transfer controller, serial communication interface, I^2C bus interface 2, synchronous serial communication unit, USB function module, A/D converter, D/A converter, and various timers. On-chip ROM is flash memory whose size is 256 Kbytes and 128 Kbytes.

Note: * Not supported by the H8S/2454 Group.

1.1.1 Applications

Application field examples: PC peripheral equipment, office automation equipment, consumer equipment, etc.

1.1.2 Overview of Specifications

The specifications of this LSI are summarized in table 1.1.

Table 1.1 Overview of Specifications

Type	Module/ Function	Description
Memory	ROM	Expanded ROM: Flash memory version, 256 Kbytes and 128 Kbytes ROM-less version
	RAM	RAM size: 64 Kbytes (in planning) and 48 Kbytes
CPU	CPU	• 16-bit high-speed H8S/2600 CPU (CISC type) Upward-compatible with H8/300, H8/300H, and H8S CPUs on an object level • General register mode (Sixteen 16-bit general registers) • Eight addressing modes • Address space: 4 Gbytes (program: 4 Gbytes, data: 4 Gbytes) • Number of basic instructions 69 types (arithmetic and logic, multiply and divide, bit-manipulation, and multiply-and-accumulate instructions) • Minimum instruction execution time (ns) 30.3 ns when system clock ϕ = 33 MHz and Vcc = 3.0 to 3.6 V (ADD instruction) • Multiplier is included (16 $\times$ 16 $\rightarrow$ 32 bits) • Multiply-and-accumulate instructions are supported (16 $\times$ 16 + 32 $\rightarrow$ 32 bits)
	Operating mode	Advanced mode

Type	Module/ Function	Description
CPU	MCU operating mode	• Mode 1: Expanded mode with on-chip ROM disabled, 16-bit bus (MD2 and MD1 pins are low and MD0 pin is high) • Mode 2: Expanded mode with on-chip ROM disabled, 8-bit bus (MD2 pin is low, MD1 pin is high, and MD0 pin is low) • Mode 3: Boot mode (MD2 pin is low and MD1 and MD0 pins are high) • Mode 4: Expanded mode with on-chip ROM enabled, 8-bit bus (MD2 pin is high and MD1 and MD0 pins are low) • Mode 7: Single-chip mode (MD2, MD1, and MD0 pins are high) • Power-down modes (a power-down mode is entered when the SLEEP instruction is executed)
Interrupts (sources)	Interrupt controller	• External interrupt pins H8S/2456 Group, H8S/2456R Group: 29 pins (NMI, $\overline{\text{IRQ15-A}}$ to $\overline{\text{IRQ0-A}}$, $\overline{\text{IRQ15-B}}$ to $\overline{\text{IRQ0-13B}}$) H8S/2454 Group: 17 pins (NMI, $\overline{\text{IRQ7-A}}$ to $\overline{\text{IRQ0-A}}$, $\overline{\text{IRQ7-B}}$ to $\overline{\text{IRQ0-B}}$) • Internal interrupt sources H8S/2456 Group, H8S/2456R Group: 107 sources H8S/2454 Group: 105 sources • Two interrupt control modes (specified by the interrupt control register) • Eight priority levels can be set (specified by the interrupt priority registers) • Independent vector addresses

Type	Module/ Function	Description
DMA	DMA controller (DMAC)	• DMA transfer is possible on four channels • Three activation sources (auto-request, on-chip module interrupt, and external request) • Byte or word can be set as the transfer unit • Short address mode or full address mode can be selected • 16-Mbyte address space can be specified directly
	EXDMA controller (EXDMAC)	• DMA transfer is possible on two channels • Two activation sources (auto-request and external request) • Two transfer modes (normal mode and block transfer mode) • Dual address mode or single address mode can be selected • 16-Mbyte address space can be specified directly • Repeat area can be set Note: EXDMAC is supported only by the H8S/2456 Group and H8S/2456R Group.
	Data transfer controller (DTC)	• Transfer is possible on any number of channels • An interrupt source can trigger data transfer (chain transfer is possible) • Three transfer modes (normal mode, repeat mode, and block transfer mode) • Byte or word can be set as the transfer unit • Activation by software is possible
External bus extension	Bus controller (BSC)	• External address space: 16 Mbytes • Manages the external address space divided into eight areas Chip select signals ($\overline{CS0}$ to $\overline{CS7}$) can be output 8-bit access or 16-bit access can be selected 2-state access or 3-state access can be selected Program wait states can be inserted • External memory interfaces (burst ROM, DRAM, synchronous DRAM*[1], address/data multiplexed I/O) • Bus arbitration function (bus arbitration of the bus masters CPU, DTC, DMAC, and EXDMAC)

Type	Module/ Function	Description
Clock	Clock pulse generator (CPG)	• This LSI has a single on-chip clock pulse generator circuit • Consists of an oscillator, a system-clock PLL circuit, a divider, and a PLL circuit for the USB, and the system clock frequency can be changed System clock (ϕ) cycle: 8 to 33 MHz • Six power-down modes Divided clock mode, sleep mode, module stop function, all module clock stop mode, software standby mode, and hardware standby mode
A/D converter	A/D converter (ADC)	• Two units • 10-bit resolution • Number of input channels H8S/2456 Group and H8S/2456R Group: 16 channels — Unit 0: 8 channels — Unit 1: 8 channels H8S/2454 Group: 10 channels — Unit 0: 8 channels — Unit 1: 2 channels • Sample and hold functionality • Conversion time: 4.0 μs per channel (when A/D conversion clock is set to 10 MHz) • Two kinds of operating modes (single mode and scan mode) • Three types of A/D conversion start (software, trigger by timer (TPU or TMR), or external trigger)
D/A converter	D/A converter (DAC)	• Resolution (8 bits) × Number of output channels (2 channels) • Conversion time: Maximum 10 μs (with 20-pF load) • Output voltage: 0 V to Vref

Type	Module/ Function	Description
Timer	16-bit timer pulse unit (TPU)	• 16-bit timer × 12 channels (general pulse timer unit) • Eight counter input clocks can be selected for each channel • Maximum 16-pulse input/output (when external expanded mode is set) • Maximum 32-pulse input/output (when single-chip mode is set) • Counter clear operation, simultaneous write to multiple timer counters (TCNT), simultaneous clearing by compare match and input capture, register simultaneous input/output possible by counter synchronous operation, and maximum of 15-phase PWM output by combination with synchronous operation • Buffer operation, phase counting mode (two-phase encoder input), and cascaded operation settable for channels • Input capture function • Output compare function (waveform output at compare match)
	8-bit timer (TMR)	• 8-bit timer × 2 channels (operation as a 16-bit timer is also possible) • Selection of seven clock sources: Six internal clock signals or an external clock input • Pulse output with an arbitrary duty cycle or PWM output
	Programmable pulse generator (PPG)	• 16-bit pulse output • Pulse outputs are divided into four groups Non-overlap mode is available Inverted output can be specified • Can operate together with the data transfer controller (DTC) and DMA controller (DMAC)
Watchdog timer	Watchdog timer (WDT)	• 8-bit timer × 1 channel (eight counter input clocks can be selected) • Switchable between watchdog timer mode and interval timer mode
Serial interface	Serial communication interface (SCI)	• Five channels (asynchronous or clocked synchronous serial communication mode) • Full-duplex communication capability • Choice of any bit rate and choice of LSB-first or MSB-first

Type	Module/ Function	Description
Smart Card/SIM		SCI supports Smart Card (SIM) interface
High-function communi-cations	I^2C bus interface 2 (IIC2)	• Four channels • Continuous transmission/reception • 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
	Synchronous serial communication unit (SSU)	• One channel • Master mode or slave mode can be selected • Standard mode or bidirectional mode can be selected • Full-duplex communication capability • Consecutive serial communication capability
	USB function module	• The protocol block conforming to USB2.0 and transceiver process USB protocol automatically. • Transfer speed: Supports full-speed (12 Mbps) • Power mode: Self power mode or bus power mode can be selected
I/O ports		• Input-only pins: 18 (144-pin version), 17 (145-pin version)*[2], 11 (120-pin version) • Input/output pins: 96 (144-pin version or 145-pin version)*[2], 83 (120-pin version) • Pull-up resistor pins: 40 • Open-drain pins: 91
Package		• 144-pin QFP package (PLQP0144KA-A) (code: FP-144LV, body size: 20 × 20 mm, pin pitch: 0.50 mm) • 145-pin TLP package (PTLG0145JB-A) (body size: 9 × 9 mm, pin pitch: 0.65 mm) • 120-pin QFP package (PLQP0120LA-A) (code: FP-120BV, body size: 14 × 14 mm, pin pitch: 0.40 mm) • Pb-free package

Type	Module/ Function	Description
Operating frequency/ power supply voltage		• Operating frequency: 8 to 33 MHz • Power supply voltage: V_{CC} = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V • Current consumption: 55 mA typ. (V_{CC} = 3.3 V, AV_{CC} = 3.3 V, ϕ = 33 MHz)
Operating environment temperature (°C)		–20°C to +75°C (regular specifications)

Notes: 1. Supported only by the H8S/2456R Group.
2. Note that the function of 145-pin version partly differs from that of 144-pin version.

1.2 List of Products

Table 1.2 lists the products and figure 1.1 shows how to read the product type name.

Table 1.2 Products

Product Type Name	ROM Size	RAM Size	Package	Remarks
R4F2456 R4F2456R	256 Kbytes/ 128 Kbytes	64 Kbytes/ 48 Kbytes	PLQP0144KA-A PTLG0145JB-A*	Flash memory version
R4F2454	256 Kbytes/ 128 Kbytes	64 Kbytes/ 48 Kbytes	PLQP0120LA-A	Flash memory version

Note: * In planning

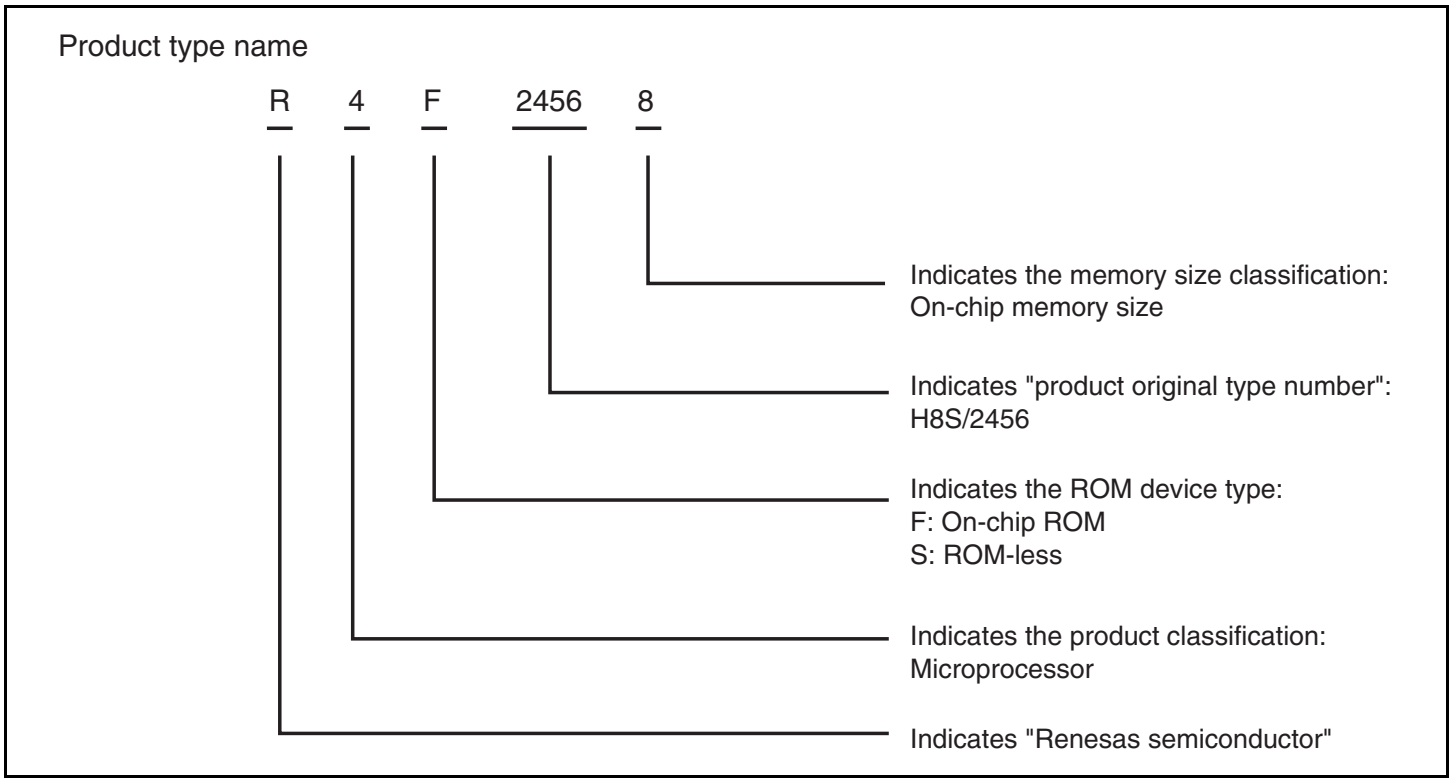
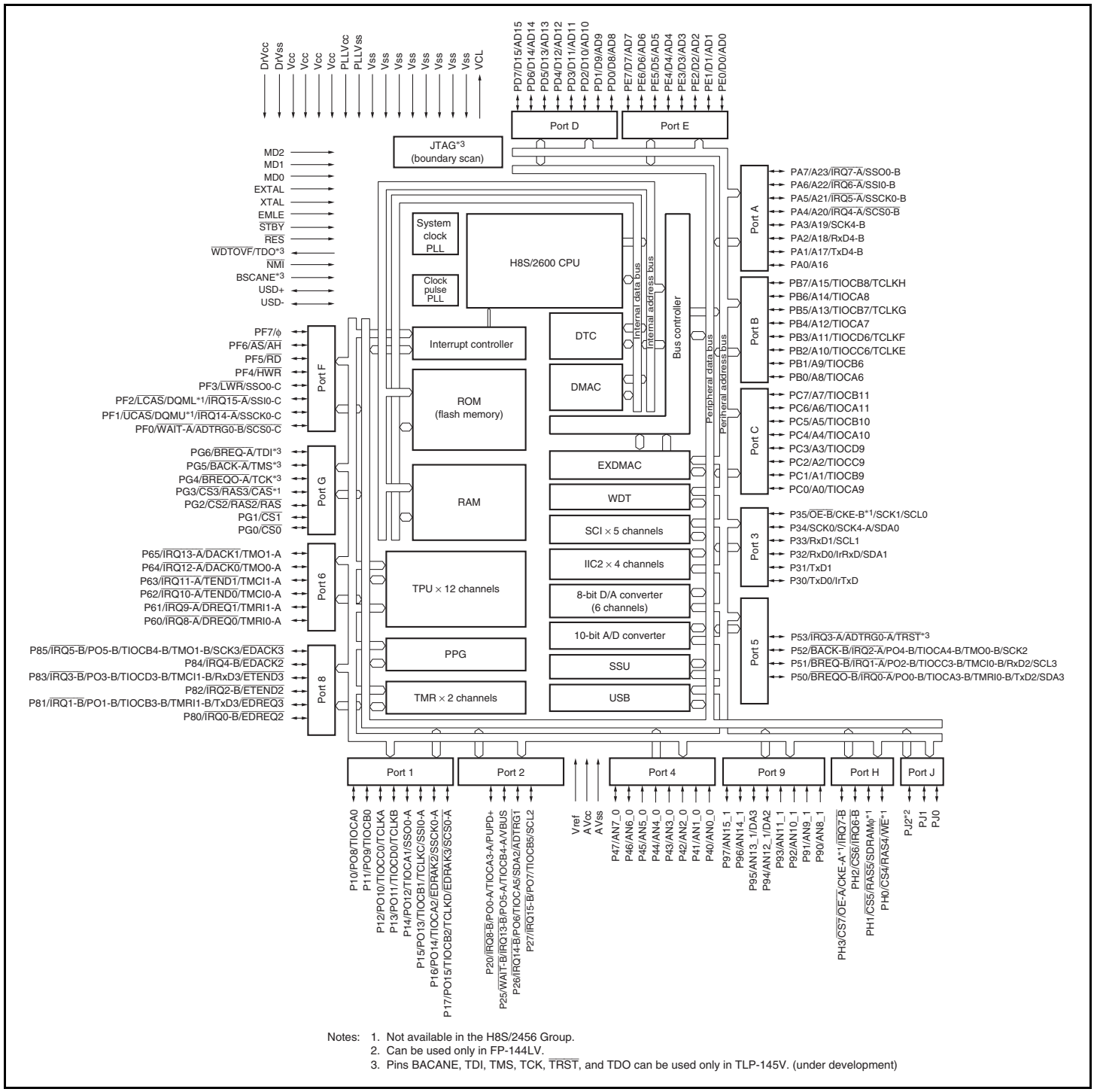


Figure 1.1 Meaning of Product Type Name

1.3 Block Diagrams



Notes: 1. Not available in the H8S/2456 Group.
2. Can be used only in FP-144LV.
3. Pins BACANE, TDI, TMS, TCK, $\overline{\text{TRST}}$, and TDO can be used only in TLP-145V. (under development)

Figure 1.2 Block Diagram of H8S/2456 Group and H8S/2456R Group

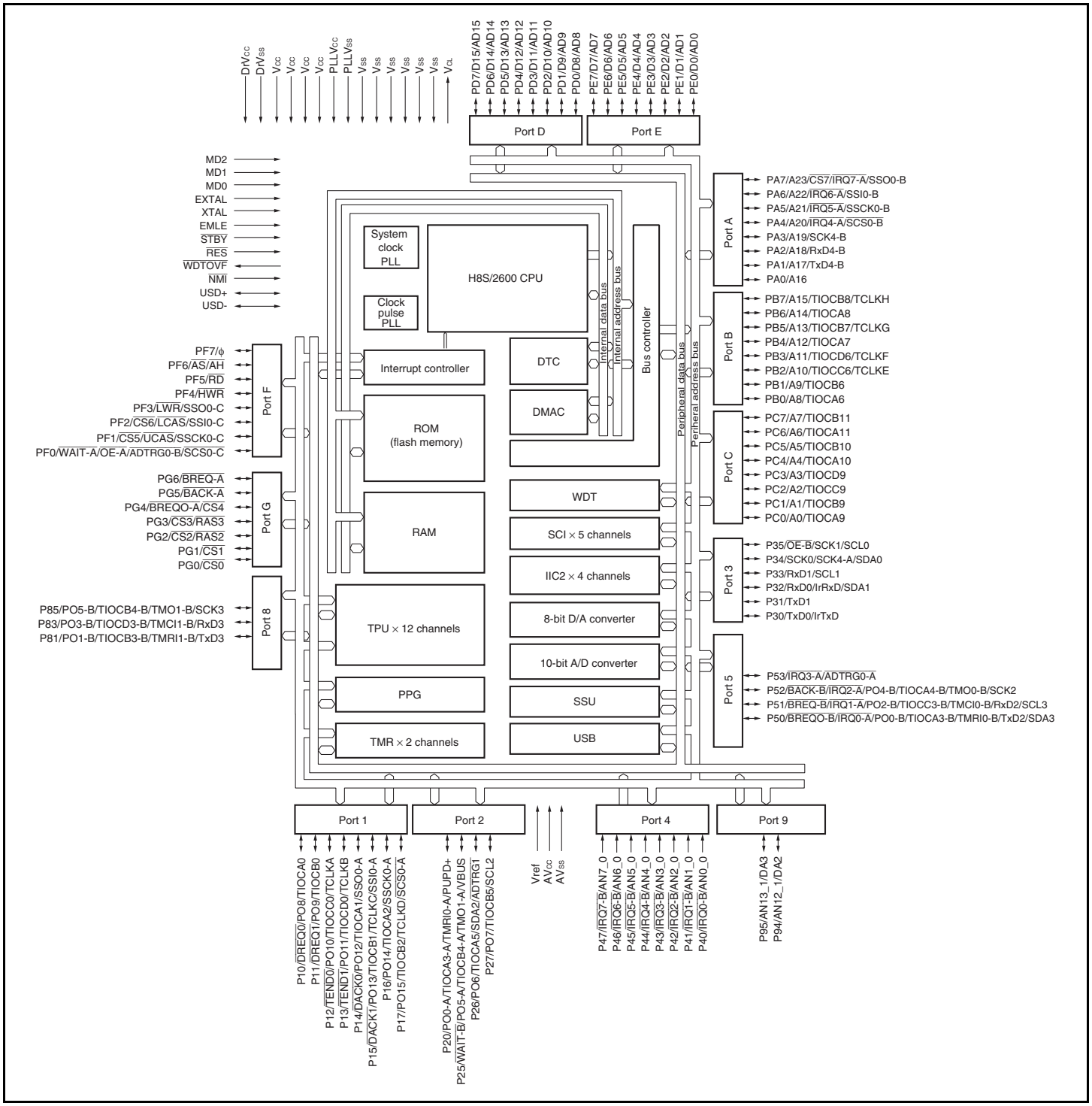
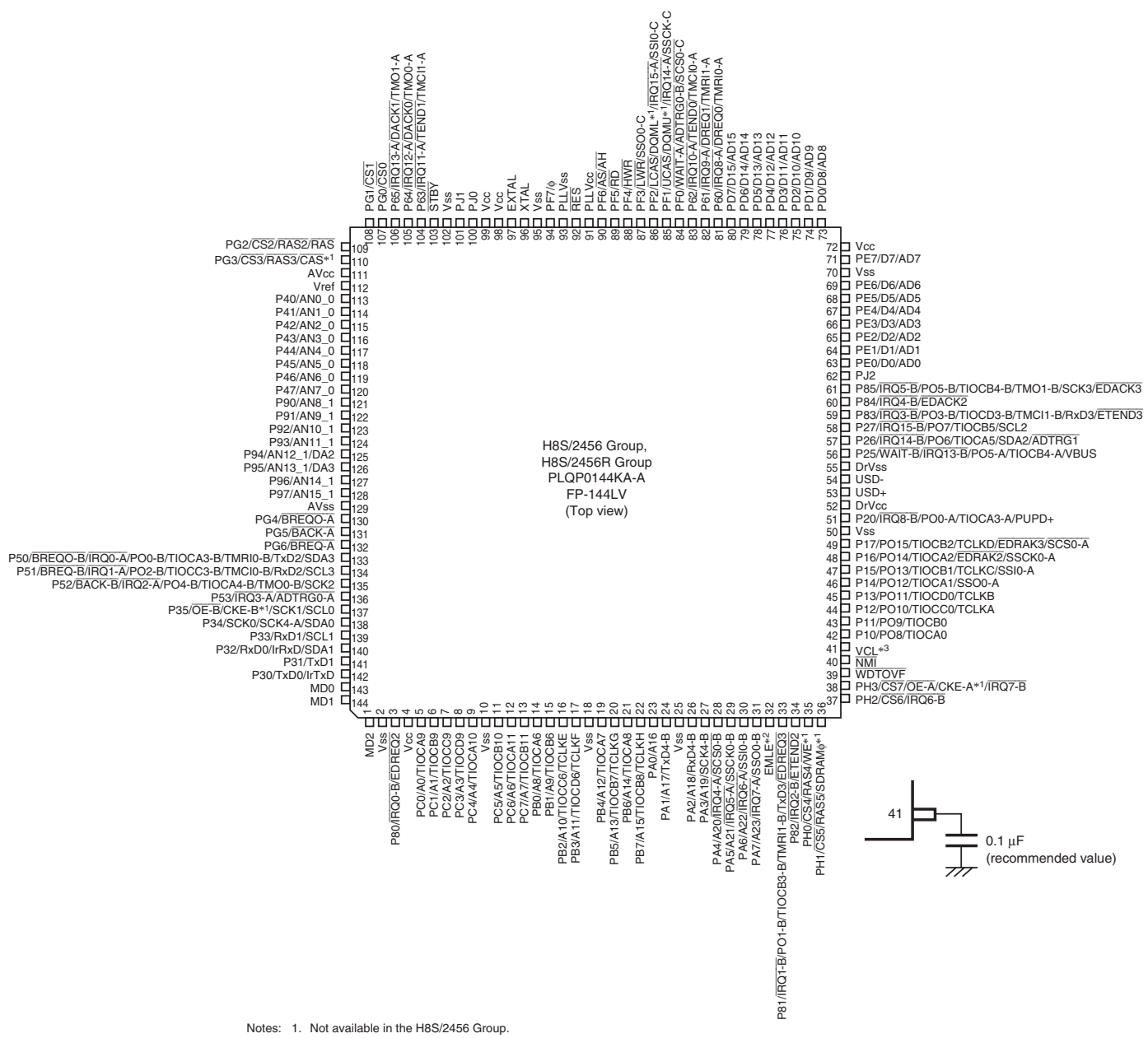


Figure 1.3 Block Diagram of H8S/2454 Group

1.4 Pin Description

1.4.1 Pin Assignments



Notes: 1. Not available in the H8S/2456 Group.
2. Emulator enable pin. In normal operating mode, this pin should be fixed low.
Driving this pin high in the flash-memory version enables the on-chip emulation function.
When the on-chip emulation function is in use, pins P53, PG4, PG5, PG6, and WDTOVF are used exclusively as the on-chip emulator pins.
3. The VCL pin should be connected to an external capacitor.

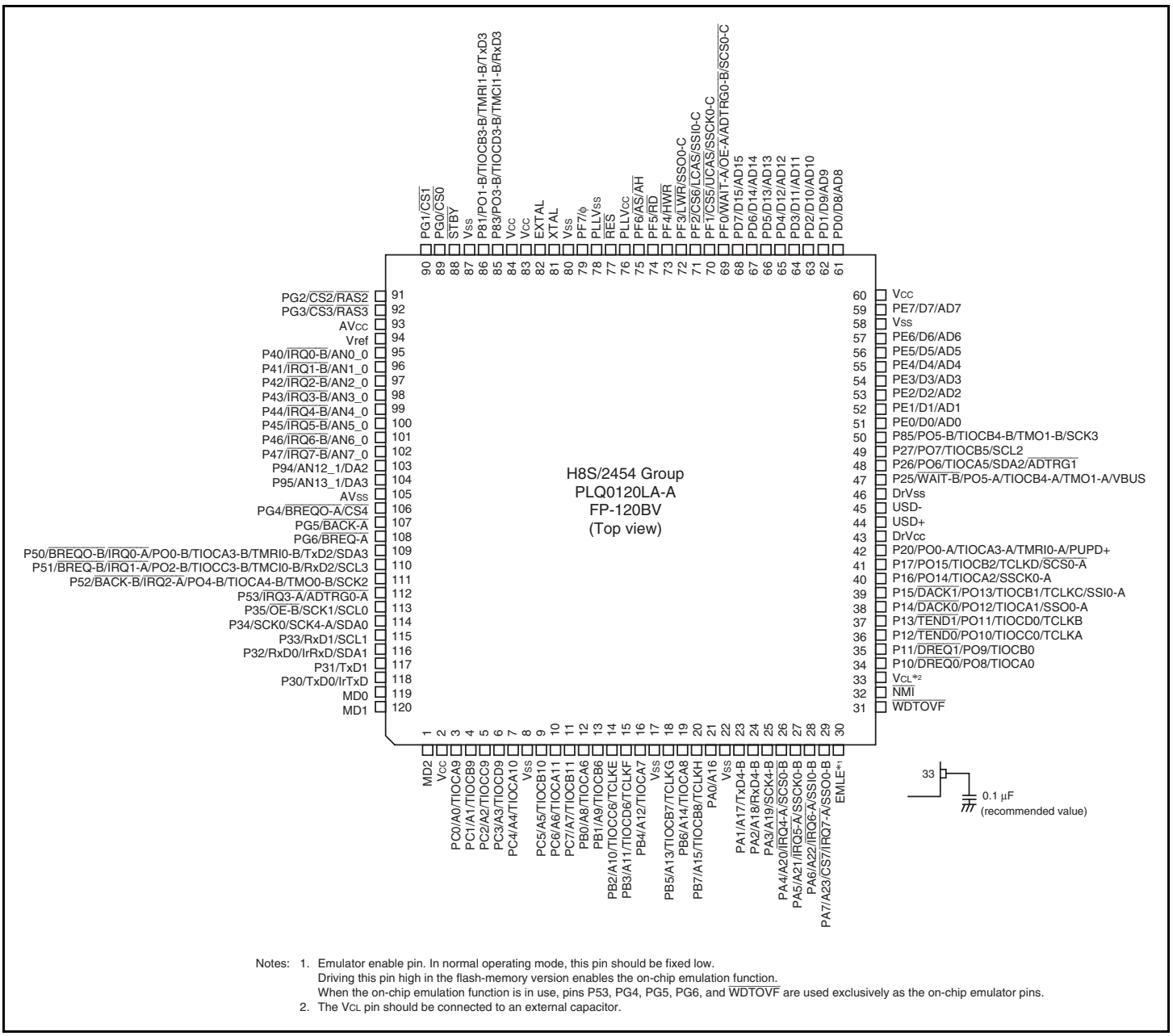
Figure 1.4 Pin Assignments for H8S/2456 Group and H8S/2456R Group (1)

	1	2	3	4	5	6	7	8	9	10	11	12	13
A	Vss	MD1	MD0	P32	P35	P50	AVss	P94	P90	P44	P40	PG2	PG3
B	MD2	Vcc	P31	P34	P51	PG4	P93	P47	P45	P42	AVcc	Vref	PG1
C	PC0	P80	PC1	P30	P33	P52	PG5	P92	P46	P43	P41	PG0	P65
D	PC4	PC2	PC3	P53	PG6	P97	P96	P95	P91	P63	PJ0	P64	$\overline{\text{STBY}}$
E	PC7	Vss	PC5	PB0	NC					Vss	Vcc	PJ1	Vcc
F	PB3	PC6	PB1	Vss						PF7	Vss	XTAL	EXTAL
G	PB6	PB2	PA0	PB4		H8S/2456 Group, H8S/2456R Group PTLG0145JB-A (Perspective top view)				PF6	$\overline{\text{RES}}$	PF5	PLLVss
H	Vss	PB7	PA3	PB5						PF2	PF4	PF1	PLLVcc
J	PA5	PA2	PA7	PA1						P62	PF0	P60	PF3
K	EMLE	PA6	P82	PA4	P15	P16	P27	P83	PE0	PE4	PD7	PD6	P61
L	PH0	P81	VCL	P12	P17	P20	DrVcc	P26	BSCANE *	PE3	PD4	PD2	PD5
M	PH1	PH3	$\overline{\text{WDTOVF}}$	P11	P13	USDt	DrVss	P85	PE2	PE6	Vss	PD3	PD0
N	$\overline{\text{NMI}}$	PH2	P10	P14	Vss	USD–	P25	P84	PE1	PE5	PE7	Vcc	PD1

Note: Connect NC to VSS or leave it open.
The VCL pin must be connected to an external capacitor (recommended value: 0.1 µF).

* Boundary scan enable pin. When the boundary scan function is used, this pin should be fixed high. At this time, pins P53, PG4 to PG6, and $\overline{\text{WDTOVF}}$ are used exclusively for boundary scan. Therefore, the corresponding pin functions of those pins are not available. When the boundary scan function is not used, this pin should be fixed low.

Figure 1.5 Pin Assignments for H8S/2456 Group and H8S/2456R Group (2) (LGA is in Planning)



Notes: 1. Emulator enable pin. In normal operating mode, this pin should be fixed low.
Driving this pin high in the flash-memory version enables the on-chip emulation function.
When the on-chip emulation function is in use, pins P53, PG4, PG5, PG6, and WDTOVF are used exclusively as the on-chip emulator pins.
2. The VCL pin should be connected to an external capacitor.

Figure 1.6 Pin Assignments for H8S/2454 Group

1.4.2 Pin Assignments in Each Operating Mode

Table 1.3 Pin Assignments in Each Operating Mode of H8S/2456 Group and H8S/2456R Group

Pin No.		Pin Name					
					Mode 7		Flash Memory Programmer Mode
144-Pin*4	145-Pin*5 (in planning)	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	
1	B1	MD2	MD2	MD2	MD2	MD2	Vss
2	A1	Vss	Vss	Vss	Vss	Vss	Vss
3	C2	P80/$\overline{\text{IRQ0-B}}$/ $\overline{\text{EDREQ2}}$	P80/$\overline{\text{IRQ0-B}}$/ $\overline{\text{EDREQ2}}$	P80/$\overline{\text{IRQ0-B}}$/ $\overline{\text{EDREQ2}}$	P80/$\overline{\text{IRQ0-B}}$/ $\overline{\text{EDREQ2}}$	P80/$\overline{\text{IRQ0-B}}$/ $\overline{\text{EDREQ2}}$	NC
4	B2	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
5	C1	A0	A0	PC0/A0	PC0/A0	PC0/TIOCA9	A0
6	C3	A1	A1	PC1/A1	PC1/A1	PC1/TIOCB9	A1
7	D2	A2	A2	PC2/A2	PC2/A2	PC2/TIOCC9	A2
8	D3	A3	A3	PC3/A3	PC3/A3	PC3/TIOCD9	A3
9	D1	A4	A4	PC4/A4	PC4/A4	PC4/TIOCA10	A4
10	E2	Vss	Vss	Vss	Vss	Vss	Vss
11	E3	A5	A5	PC5/A5	PC5/A5	PC5/TIOCB10	A5
12	F2	A6	A6	PC6/A6	PC6/A6	PC6/TIOCA11	A6
13	E1	A7	A7	PC7/A7	PC7/A7	PC7/TIOCB11	A7
14	E4	A8	A8	PB0/A8	PB0/A8	PB0/TIOCA6	A8
15	F3	A9	A9	PB1/A9	PB1/A9	PB1/TIOCB6	A9
16	G2	A10	A10	PB2/A10	PB2/A10	PB2/TIOCC6/ TCLKE	A10
17	F1	A11	A11	PB3/A11	PB3/A11	PB3/TIOCD6/ TCLKF	A11
18	F4	Vss	Vss	Vss	Vss	Vss	Vss
19	G4	A12	A12	PB4/A12	PB4/A12	PB4/TIOCA7	A12
20	H4	A13	A13	PB5/A13	PB5/A13	PB5/TIOCB7/ TCLKG	A13
21	G1	A14	A14	PB6/A14	PB6/A14	PB6/TIOCA8	A14
22	H2	A15	A15	PB7/A15	PB7/A15	PB7/TIOCB8/ TCLKH	A15
23	G3	A16	A16	PA0/A16	PA0/A16	PA0	A16
24	J4	A17	A17	PA1/A17/TxD4_B	PA1/A17/TxD4_B	PA1/TxD4_B	A17

Pin No.		Pin Name					
					Mode 7		Flash Memory Programmer Mode
144-Pin*4	145-Pin*5 (in planning)	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	
25	H1	Vss	Vss	Vss	Vss	Vss	Vss
26	J2	A18	A18	PA2/A18/RxD4-B	PA2/A18/RxD4-B	PA2/RxD4-B	A18
27	H3	A19	A19	PA3/A19/SCK4-B	PA3/A19/SCK4-B	PA3/SCK4-B	NC
28	K4	A20/IRQ4-A	A20/IRQ4-A	PA4/A20/IRQ4-A/ SCS0-B	PA4/A20/IRQ4-A/ SCS0-B	PA4/IRQ4-A/ SCS0-B	NC
29	J1	PA5/A21/ IRQ5-A/SSCK0-B	PA5/A21/ IRQ5-A/SSCK0-B	PA5/A21/ IRQ5-A/SSCK0-B	PA5/A21/ IRQ5-A/SSCK0-B	PA5/ IRQ5-A/SSCK0-B	NC
30	K2	PA6/A22/ IRQ6-A/SSI0-B	PA6/A22/ IRQ6-A/SSI0-B	PA6/A22/ IRQ6-A/SSI0-B	PA6/A22/ IRQ6-A/SSI0-B	PA6/ IRQ6-A/SSI0-B	NC
31	J3	PA7/A23/ IRQ7-A/SSO0-B	PA7/A23/ IRQ7-A/SSO0-B	PA7/A23/ IRQ7-A/SSO0-B	PA7/A23/ IRQ7-A/SSO0-B	PA7/ IRQ7-A/SSO0-B	NC
32	K1	EMLE	EMLE	EMLE	EMLE	EMLE	Vss
33	L2	P81/IRQ1-B/ PO1-B/ TIOCB3-B/ TMRI1-B/TxD3/ EDREQ3	P81/IRQ1-B/ PO1-B/ TIOCB3-B/ TMRI1-B/TxD3/ EDREQ3	P81/IRQ1-B/ PO1-B/ TIOCB3-B/ TMRI1-B/TxD3/ EDREQ3	P81/IRQ1-B/ PO1-B/ TIOCB3-B/ TMRI1-B/TxD3/ EDREQ3	P81/IRQ1-B/ PO1-B/ TIOCB3-B/ TMRI1-B/TxD3/ EDREQ3	NC
34	K3	P82/IRQ2-B/ ETEND2	P82/IRQ2-B/ ETEND2	P82/IRQ2-B/ ETEND2	P82/IRQ2-B/ ETEND2*3	P82/IRQ2-B	NC
35	L1	PH0/CS4/ RAS4/WE*1	PH0/CS4/ RAS4/WE*1	PH0/CS4/ RAS4/WE*1	PH0/CS4/ RAS4/WE*1	PH0	NC
36	M1	PH1/CS5/ RAS5/SDRAM *1	PH1/CS5/ RAS5/SDRAM *1	PH1/CS5/ RAS5/SDRAM *1	PH1/CS5/ RAS5/SDRAM *1	PH1/ SDRAM *1	NC
37	N2	PH2/CS6/IRQ6-B	PH2/CS6/IRQ6-B	PH2/CS6/IRQ6-B	PH2/CS6/IRQ6-B	PH2/IRQ6-B	NC
38	M2	PH3/CS7/ OE-A/CKE-A*1/ IRQ7-B	PH3/CS7/ OE-A/CKE-A*1/ IRQ7-B	PH3/CS7/ OE-A/CKE-A*1/ IRQ7-B	PH3/CS7/ OE-A/CKE-A*1/ IRQ7-B	PH3/IRQ7-B	NC
39	M3	WDTOVF/TDO*3	WDTOVF/TDO*3	WDTOVF/TDO*3	WDTOVF/TDO*3	WDTOVF/TDO*3	NC
40	N1	NMI	NMI	NMI	NMI	NMI	Vcc
41	L3	VCL	VCL	VCL	VCL	VCL	VCL
42	N3	P10/PO8/TIOCA0	P10/PO8/TIOCA0	P10/PO8/TIOCA0	P10/PO8/TIOCA0	P10/PO8/TIOCA0	NC
43	M4	P11/PO9/TIOCB0	P11/PO9/TIOCB0	P11/PO9/TIOCB0	P11/PO9/TIOCB0	P11/PO9/TIOCB0	NC
44	L4	P12/PO10/ TIOCC0/TCLKA	P12/PO10/ TIOCC0/TCLKA	P12/PO10/ TIOCC0/TCLKA	P12/PO10/ TIOCC0/TCLKA	P12/PO10/ TIOCC0/TCLKA	OE
45	M5	P13/PO11/ TIOCD0/TCLKB	P13/PO11/ TIOCD0/TCLKB	P13/PO11/ TIOCD0/TCLKB	P13/PO11/ TIOCD0/TCLKB	P13/PO11/ TIOCD0/TCLKB	CE

Pin No.		Pin Name					
					Mode 7		Flash Memory Programmer Mode
144-Pin*4	145-Pin*5 (in planning)	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	
46	N4	P14/PO12/ TIOCA1/SSO0-A	P14/PO12/ TIOCA1/SSO0-A	P14/PO12/ TIOCA1/SSO0-A	P14/PO12/ TIOCA1/SSO0-A	P14/PO12/ TIOCA1/SSO0-A	$\overline{WE}$
47	K5	P15/PO13/ TIOCB1/TCLKC/ SSI0-A	P15/PO13/ TIOCB1/TCLKC/ SSI0-A	P15/PO13/ TIOCB1/TCLKC/ SSI0-A	P15/PO13/ TIOCB1/TCLKC/ SSI0-A	P15/PO13/ TIOCB1/TCLKC/ SSI0-A	NC
48	K6	P16/PO14/ TIOCA2/ $\overline{EDRAK2}$/ SSCK0-A	P16/PO14/ TIOCA2/ $\overline{EDRAK2}$/ SSCK0-A	P16/PO14/ TIOCA2/ $\overline{EDRAK2}$/ SSCK0-A	P16/PO14/ TIOCA2/ $\overline{EDRAK2}$/ SSCK0-A	P16/PO14/ TIOCA2/ SSCK0-A	NC
49	L5	P17/PO15/ TIOCB2/TCLKD/ $\overline{EDRAK3}$/$\overline{SCS0-A}$	P17/PO15/ TIOCB2/TCLKD/ $\overline{EDRAK3}$/$\overline{SCS0-A}$	P17/PO15/ TIOCB2/TCLKD/ $\overline{EDRAK3}$/$\overline{SCS0-A}$	P17/PO15/ TIOCB2/TCLKD/ $\overline{EDRAK3}$/$\overline{SCS0-A}$	P17/PO15/ TIOCB2/TCLKD/ $\overline{SCS0-A}$	NC
50	N5	Vss	Vss	Vss	Vss	Vss	Vss
51	L6	P20/$\overline{IRQ8-B}$/ PO0-A/TIOCA3 -A/PUPD+	P20/$\overline{IRQ8-B}$/ PO0-A/TIOCA3 -A/PUPD+	P20/$\overline{IRQ8-B}$/ PO0-A/TIOCA3 -A/PUPD+	P20/$\overline{IRQ8-B}$/ PO0-A/TIOCA3 -A/PUPD+	P20/$\overline{IRQ8-B}$/ PO0-A/TIOCA3 -A/PUPD+	NC
52	L7	DrVcc	DrVcc	DrVcc	DrVcc	DrVcc	Vcc
53	M6	USD+	USD+	USD+	USD+	USD+	NC
54	N6	USD-	USD-	USD-	USD-	USD-	NC
55	M7	DrVss	DrVss	DrVss	DrVss	DrVss	Vss
56	N7	P25/$\overline{WAIT-B}$/ $\overline{IRQ13-B}$/PO5-A/ TIOCB4 -A/VBUS	P25/$\overline{WAIT-B}$/ $\overline{IRQ13-B}$/PO5-A/ TIOCB4-A/VBUS	P25/$\overline{WAIT-B}$/ $\overline{IRQ13-B}$/PO5-A/ TIOCB4-A/VBUS	P25/$\overline{WAIT-B}$/ $\overline{IRQ13-B}$/PO5-A/TIOCB4 -A/VBUS	P25/$\overline{IRQ13-B}$/ PO5-A/ TIOCB4-A/VBUS	NC
57	L8	P26/$\overline{IRQ14-B}$/ PO6/TIOCA5/ SDA2/$\overline{ADTRG1}$	P26/$\overline{IRQ14-B}$/ PO6/TIOCA5/ SDA2/$\overline{ADTRG1}$	P26/$\overline{IRQ14-B}$/ PO6/TIOCA5/ SDA2/$\overline{ADTRG1}$	P26/$\overline{IRQ14-B}$/ PO6/TIOCA5/ SDA2/$\overline{ADTRG1}$	P26/$\overline{IRQ14-B}$/ PO6/TIOCA5/ SDA2/$\overline{ADTRG1}$	NC
58	K7	P27/$\overline{IRQ15-B}$/ PO7/TIOCB5/ SCL2	P27/$\overline{IRQ15-B}$/ PO7/TIOCB5/ SCL2	P27/$\overline{IRQ15-B}$/ PO7/TIOCB5/ SCL2	P27/$\overline{IRQ15-B}$/ PO7/TIOCB5/ SCL2	P27/$\overline{IRQ15-B}$/ PO7/TIOCB5/ SCL2	NC
59	K8	P83/$\overline{IRQ3-B}$/ PO3-B/ TIOCD3-B/ TMCI1-B/RxD3/ $\overline{ETEND3}$	P83/$\overline{IRQ3-B}$/ PO3-B/ TIOCD3-B/ TMCI1-B/RxD3/ $\overline{ETEND3}$	P83/$\overline{IRQ3-B}$/ PO3-B/ TIOCD3-B/ TMCI1-B/RxD3/ $\overline{ETEND3}$	P83/$\overline{IRQ3-B}$/ PO3-B/ TIOCD3-B/ TMCI1-B/RxD3/ $\overline{ETEND3}$	P83/$\overline{IRQ3-B}$/ PO3-B/ TIOCD3-B/ TMCI1-B/RxD3	NC
60	N8	P84/$\overline{IRQ4-B}$/ $\overline{EDACK2}$	P84/$\overline{IRQ4-B}$/ $\overline{EDACK2}$	P84/$\overline{IRQ4-B}$/ $\overline{EDACK2}$	P84/$\overline{IRQ4-B}$/ $\overline{EDACK2}$	P84/$\overline{IRQ4-B}$/	NC

Pin No.		Pin Name					
					Mode 7		Flash Memory Programmer Mode
144-Pin*4	145-Pin*5 (in planning)	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	
61	M8	P85/IRQ5-B/ PO5-B/ TIOCB4-B/ TMO1-B/SCK3/ EDACK3	P85/IRQ5-B/ PO5-B/ TIOCB4-B/ TMO1-B/SCK3/ EDACK3	P85/IRQ5-B/ PO5-B/ TIOCB4-B/ TMO1-B/SCK3/ EDACK3	P85/IRQ5-B/ PO5-B/ TIOCB4-B/ TMO1-B/SCK3/ EDACK3	P85/IRQ5-B/ PO5-B/ TIOCB4-B/ TMO1-B/SCK3	NC
62	L9	PJ2*2/BSCANE*3	PJ2*2/BSCANE*3	PJ2*2/BSCANE*3	PJ2*2/BSCANE*3	PJ2*2/BSCANE*3	NC
63	K9	PE0/D0/AD0	PE0/D0/AD0	PE0/D0/AD0	PE0/D0/AD0	PE0	NC
64	N9	PE1/D1/AD1	PE1/D1/AD1	PE1/D1/AD1	PE1/D1/AD1	PE1	NC
65	M9	PE2/D2/AD2	PE2/D2/AD2	PE2/D2/AD2	PE2/D2/AD2	PE2	NC
66	L10	PE3/D3/AD3	PE3/D3/AD3	PE3/D3/AD3	PE3/D3/AD3	PE3	NC
67	K10	PE4/D4/AD4	PE4/D4/AD4	PE4/D4/AD4	PE4/D4/AD4	PE4	NC
68	N10	PE5/D5/AD5	PE5/D5/AD5	PE5/D5/AD5	PE5/D5/AD5	PE5	NC
69	M10	PE6/D6/AD6	PE6/D6/AD6	PE6/D6/AD6	PE6/D6/AD6	PE6	NC
70	M11	Vss	Vss	Vss	Vss	Vss	Vss
71	N11	D7/AD7	PE7/D7/AD7	PE7/D7/AD7	PE7/D7/AD7	PE7	NC
72	N12	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
73	M13	D8/AD8	D8/AD8	D8/AD8	D8/AD8	PD0	I/O0
74	N13	D9/AD9	D9/AD9	D9/AD9	D9/AD9	PD1	I/O1
75	L12	D10/AD10	D10/AD10	D10/AD10	D10/AD10	PD2	I/O2
76	M12	D11/AD11	D11/AD11	D11/AD11	D11/AD11	PD3	I/O3
77	L11	D12/AD12	D12/AD12	D12/AD12	D12/AD12	PD4	I/O4
78	L13	D13/AD13	D13/AD13	D13/AD13	D13/AD13	PD5	I/O5
79	K12	D14/AD14	D14/AD14	D14/AD14	D14/AD14	PD6	I/O6
80	K11	D15/AD15	D15/AD15	D15/AD15	D15/AD15	PD7	I/O7
81	J12	P60/IRQ8-A/ DREQ0/TMRI0-A	P60/IRQ8-A/ DREQ0/TMRI0-A	P60/IRQ8-A/ DREQ0/TMRI0-A	P60/IRQ8-A/ DREQ0/TMRI0-A	P60/IRQ8-A/ DREQ0/TMRI0-A	NC
82	K13	P61/IRQ9-A/ DREQ1/TMRI1-A	P61/IRQ9-A/ DREQ1/TMRI1-A	P61/IRQ9-A/ DREQ1/TMRI1-A	P61/IRQ9-A/ DREQ1/TMRI1-A	P61/IRQ9-A/ DREQ1/TMRI1-A	NC
83	J10	P62/IRQ10-A/ TEND0/TMCI0-A	P62/IRQ10-A/ TEND0/TMCI0-A	P62/IRQ10-A/ TEND0/TMCI0-A	P62/IRQ10-A/ TEND0/TMCI0-A	P62/IRQ10-A/ TEND0/TMCI0-A	NC
84	J11	PF0/WAIT-A/ ADTRG0-B/ SCS0-C	PF0/WAIT-A/ ADTRG0-B/ SCS0-C	PF0/WAIT-A/ ADTRG0-B/ SCS0-C	PF0/WAIT-A/ ADTRG0-B/ SCS0-C	PF0/ ADTRG0-B/ SCS0-C	NC

Pin No.		Pin Name					
					Mode 7		Flash Memory Programmer Mode
144-Pin*4	145-Pin*5 (in planning)	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	
85	H12	PF1/$\overline{\text{UCAS}}$/ DQMU*1 /$\overline{\text{IRQ14-A}}$/ SSCK0-C	PF1/$\overline{\text{UCAS}}$/ DQMU*1/ $\overline{\text{IRQ14-A}}$/ SSCK0-C	PF1/$\overline{\text{UCAS}}$/ DQMU*1/ $\overline{\text{IRQ14-A}}$/ SSCK0-C	PF1/$\overline{\text{UCAS}}$/ DQMU*1/ $\overline{\text{IRQ14-A}}$/ SSCK0-C	PF1/$\overline{\text{IRQ14-A}}$/ SSCK0-C	NC
86	H10	PF2/$\overline{\text{LCAS}}$/ DQML*1/ $\overline{\text{IRQ15-A}}$/SSI0-C	PF2/$\overline{\text{LCAS}}$/ DQML*1/ $\overline{\text{IRQ15-A}}$/SSI0-C	PF2/$\overline{\text{LCAS}}$/ DQML*1/ $\overline{\text{IRQ15-A}}$/SSI0-C	PF2/$\overline{\text{LCAS}}$/ DQML*1/ $\overline{\text{IRQ15-A}}$/SSI0-C	PF2/$\overline{\text{IRQ15-A}}$/ SSI0-C	NC
87	J13	PF3/$\overline{\text{LWR}}$/ SSO0-C	PF3/$\overline{\text{LWR}}$/ SSO0-C	PF3/$\overline{\text{LWR}}$/ SSO0-C	PF3/$\overline{\text{LWR}}$/ SSO0-C	PF3/ SSO0-C	NC
88	H11	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	PF4	NC
89	G12	$\overline{\text{RD}}$	$\overline{\text{RD}}$	$\overline{\text{RD}}$	$\overline{\text{RD}}$	PF5	NC
90	G10	PF6/$\overline{\text{AS}}$/$\overline{\text{AH}}$	PF6/$\overline{\text{AS}}$/$\overline{\text{AH}}$	PF6/$\overline{\text{AS}}$/$\overline{\text{AH}}$	PF6/$\overline{\text{AS}}$/$\overline{\text{AH}}$	PF6	NC
91	H13	PLLVcc	PLLVcc	PLLVcc	PLLVcc	PLLVcc	Vcc
92	G11	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
93	G13	PLLVss	PLLVss	PLLVss	PLLVss	PLLVss	Vss
94	F10	PF7/ϕ	PF7/ϕ	PF7/ϕ	PF7/ϕ	PF7/ϕ	NC
95	F11	Vss	Vss	Vss	Vss	Vss	Vss
96	F12	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL
97	F13	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL
98	E11	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
99	E13	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
100	D11	PJ0	PJ0	PJ0	PJ0	PJ0	NC
101	E12	PJ1	PJ1	PJ1	PJ1	PJ1	NC
102	E10	Vss	Vss	Vss	Vss	Vss	Vss
103	D13	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	Vcc
104	D10	P63/$\overline{\text{IRQ11-A}}$/ $\overline{\text{TEND1}}$/TMCI1-A	P63/$\overline{\text{IRQ11-A}}$/ $\overline{\text{TEND1}}$/TMCI1-A	P63/$\overline{\text{IRQ11-A}}$/ $\overline{\text{TEND1}}$/TMCI1-A	P63/$\overline{\text{IRQ11-A}}$/ $\overline{\text{TEND1}}$/TMCI1-A	P63/$\overline{\text{IRQ11-A}}$/ $\overline{\text{TEND1}}$/TMCI1-A	NC
105	D12	P64/$\overline{\text{IRQ12-A}}$/ $\overline{\text{DACK0}}$/TMO0-A	P64/$\overline{\text{IRQ12-A}}$/ $\overline{\text{DACK0}}$/TMO0-A	P64/$\overline{\text{IRQ12-A}}$/ $\overline{\text{DACK0}}$/TMO0-A	P64/$\overline{\text{IRQ12-A}}$/ $\overline{\text{DACK0}}$/TMO0-A	P64/$\overline{\text{IRQ12-A}}$/ $\overline{\text{DACK0}}$/TMO0-A	NC
106	C13	P65/$\overline{\text{IRQ13-A}}$/ $\overline{\text{DACK1}}$/TMO1-A	P65/$\overline{\text{IRQ13-A}}$/ $\overline{\text{DACK1}}$/TMO1-A	P65/$\overline{\text{IRQ13-A}}$/ $\overline{\text{DACK1}}$/TMO1-A	P65/$\overline{\text{IRQ13-A}}$/ $\overline{\text{DACK1}}$/TMO1-A	P65/$\overline{\text{IRQ13-A}}$/ $\overline{\text{DACK1}}$/TMO1-A	NC
107	C12	PG0/$\overline{\text{CS0}}$	PG0/$\overline{\text{CS0}}$	PG0/$\overline{\text{CS0}}$	PG0/$\overline{\text{CS0}}$	PG0	NC
108	B13	PG1/$\overline{\text{CS1}}$	PG1/$\overline{\text{CS1}}$	PG1/$\overline{\text{CS1}}$	PG1/$\overline{\text{CS1}}$	PG1	NC
109	A12	PG2/$\overline{\text{CS2}}$/ $\overline{\text{RAS2}}$/$\overline{\text{RAS}}$	PG2/$\overline{\text{CS2}}$/ $\overline{\text{RAS2}}$/$\overline{\text{RAS}}$	PG2/$\overline{\text{CS2}}$/ $\overline{\text{RAS2}}$/$\overline{\text{RAS}}$	PG2/$\overline{\text{CS2}}$/ $\overline{\text{RAS2}}$/$\overline{\text{RAS}}$	PG2	NC

Pin No.		Pin Name					
					Mode 7		Flash Memory Programmer Mode
144-Pin*4	145-Pin*5 (in planning)	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	
110	A13	PG3/$\overline{CS3}$/ $\overline{RAS3}$/$\overline{CAS}$*1	PG3/$\overline{CS3}$/ $\overline{RAS3}$/$\overline{CAS}$*1	PG3/$\overline{CS3}$/ $\overline{RAS3}$/$\overline{CAS}$*1	PG3/$\overline{CS3}$/ $\overline{RAS3}$/$\overline{CAS}$*1	PG3	NC
111	B11	AVcc	AVcc	AVcc	AVcc	AVcc	Vcc
112	B12	Vref	Vref	Vref	Vref	Vref	Vcc
113	A11	P40/AN0_0	P40/AN0_0	P40/AN0_0	P40/AN0_0	P40/AN0_0	NC
114	C11	P41/AN1_0	P41/AN1_0	P41/AN1_0	P41/AN1_0	P41/AN1_0	NC
115	B10	P42/AN2_0	P42/AN2_0	P42/AN2_0	P42/AN2_0	P42/AN2_0	NC
116	C10	P43/AN3_0	P43/AN3_0	P43/AN3_0	P43/AN3_0	P43/AN3_0	Vss
117	A10	P44/AN4_0	P44/AN4_0	P44/AN4_0	P44/AN4_0	P44/AN4_0	Vcc
118	B9	P45/AN5_0	P45/AN5_0	P45/AN5_0	P45/AN5_0	P45/AN5_0	Vss
119	C9	P46/AN6_0	P46/AN6_0	P46/AN6_0	P46/AN6_0	P46/AN6_0	NC
120	B8	P47/AN7_0	P47/AN7_0	P47/AN7_0	P47/AN7_0	P47/AN7_0	NC
121	A9	P90/AN8_1	P90/AN8_1	P90/AN8_1	P90/AN8_1	P90/AN8_1	NC
122	D9	P91/AN9_1	P91/AN9_1	P91/AN9_1	P91/AN9_1	P91/AN9_1	NC
123	C8	P92/AN10_1	P92/AN10_1	P92/AN10_1	P92/AN10_1	P92/AN10_1	NC
124	B7	P93/AN11_1	P93/AN11_1	P93/AN11_1	P93/AN11_1	P93/AN11_1	NC
125	A8	P94/AN12_1/DA2	P94/AN12_1/DA2	P94/AN12_1/DA2	P94/AN12_1/DA2	P94/AN12_1/DA2	NC
126	D8	P95/AN13_1/DA3	P95/AN13_1/DA3	P95/AN13_1/DA3	P95/AN13_1/DA3	P95/AN13_1/DA3	NC
127	D7	P96/AN14_1	P96/AN14_1	P96/AN14_1	P96/AN14_1	P96/AN14_1	NC
128	D6	P97/AN15_1	P97/AN15_1	P97/AN15_1	P97/AN15_1	P97/AN15_1	NC
129	A7	AVss	AVss	AVss	AVss	AVss	Vss
130	B6	PG4/$\overline{BREQO-A}$/ TCK*3	PG4/$\overline{BREQO-A}$/ TCK*3	PG4/$\overline{BREQO-A}$/ TCK*3	PG4/$\overline{BREQO-A}$/ TCK*3	PG4/TCK*3	NC
131	C7	PG5/ $\overline{BACK-A}$/TMS*3	PG5/$\overline{BACK-A}$/ TMS*3	PG5/$\overline{BACK-A}$/ TMS*3	PG5/$\overline{BACK-A}$/ TMS*3	PG5/TMS*3	NC
132	D5	PG6/ $\overline{BREQ-A}$/TDI*3	PG6/ $\overline{BREQ-A}$/TDI*3	PG6/ $\overline{BREQ-A}$/TDI*3	PG6/ $\overline{BREQ-A}$/TDI*3	PG6/TDI*3	NC
133	A6	P50/$\overline{BREQO-B}$/ $\overline{IRQ0-A}$/PO0-B/ TIOCA3-B/ TMRI0-B/TxD2/ SDA3	P50/$\overline{BREQO-B}$/ $\overline{IRQ0-A}$/PO0-B/ TIOCA3-B/ TMRI0-B/TxD2/ SDA3	P50/$\overline{BREQO-B}$/ $\overline{IRQ0-A}$/PO0-B/ TIOCA3-B/ TMRI0-B/TxD2/ SDA3	P50/$\overline{BREQO-B}$/ $\overline{IRQ0-A}$/PO0-B/ TIOCA3-B/ TMRI0-B/TxD2/ SDA3	P50/$\overline{IRQ0-A}$/ PO0-B/ TIOCA3-B/ TMRI0-B/TxD2/ SDA3	Vss

Pin No.		Pin Name					
					Mode 7		Flash Memory Programmer Mode
144-Pin*[4]	145-Pin*[5] (in planning)	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	
134	B5	P51/BREQ-B/ IRQ1-A/PO2-B/ TIOCC3-B/ TMCI0-B/RxD2/ SCL3	P51/BREQ-B/ IRQ1-A/PO2-B/ TIOCC3-B/ TMCI0-B/RxD2/ SCL3	P51/BREQ-B/ IRQ1-A/PO2-B/ TIOCC3-B/ TMCI0-B/RxD2/ SCL3	P51/BREQ-B/ IRQ1-A/PO2-B/ TIOCC3-B/ TMCI0-B/RxD2/ SCL3	P51 PO2-B/ TIOCC3-B/ TMCI0-B/RxD2/ SCL3	Vss
135	C6	P52/BACK-B/ IRQ2-A/PO4-B/ TIOCA4-B/ TMO0-B/SCK2	P52/BACK-B/ IRQ2-A/PO4-B/ TIOCA4-B/ TMO0-B/SCK2	P52/BACK-B/ IRQ2-A/PO4-B/ TIOCA4-B/ TMO0-B/SCK2	P52/BACK-B/ IRQ2-A/PO4-B/ TIOCA4-B/ TMO0-B/SCK2	P52 PO4-B/ TIOCA4-B/ TMO0-B/SCK2	Vcc
136	D4	P53/IRQ3-A/ ADTRG0-A/ TRST*[3]	P53/IRQ3-A/ ADTRG0-A/ TRST*[3]	P53/IRQ3-A/ ADTRG0-A/ TRST*[3]	P53/IRQ3-A/ ADTRG0-A/ TRST*[3]	P53/IRQ3-A/ ADTRG0-A/ TRST*[3]	NC
137	A5	P35/OE-B/ CKE-B*[1]/ SCK1/SCL0	P35/OE-B/ CKE-B*[1]/ SCK1/SCL0	P35/OE-B/ CKE-B*[1]/ SCK1/SCL0	P35/OE-B/ CKE-B*[1]/ SCK1/SCL0	P35/SCK1/ SCL0	NC
138	B4	P34/SCK0/ SCK4-A/SDA0	P34/SCK0/ SCK4-A/SDA0	P34/SCK0/ SCK4-A/SDA0	P34/SCK0/ SCK4-A/SDA0	P34/SCK0/ SCK4-A/SDA0	NC
139	C5	P33/RxD1/SCL1	P33/RxD1/SCL1	P33/RxD1/SCL1	P33/RxD1/SCL1	P33/RxD1/SCL1	NC
140	A4	P32/RxD0/ IrRxD/SDA1	P32/RxD0/ IrRxD/SDA1	P32/RxD0/ IrRxD/SDA1	P32/RxD0/ IrRxD/SDA1	P32/RxD0/ IrRxD/SDA1	Vcc
141	B3	P31/TxD1	P31/TxD1	P31/TxD1	P31/TxD1	P31/TxD1	NC
142	C4	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD	NC
143	A3	MD0	MD0	MD0	MD0	MD0	Vss
144	A2	MD1	MD1	MD1	MD1	MD1	Vss
—	E5	NC	NC	NC	NC	NC	NC

Notes: 1. Not available in the H8S/2456 Group.
2. Can be used only in FP-144LV version.
3. Can be used only in TLP-145V version.
4. The 144-pin code is FP-144LV.
5. The 145-pin code is TLP-145V.

Table 1.4 Pin Assignments in Each Operating Mode of H8S/2454 Group

Pin No.	Pin Name					
				Mode 7		Flash Memory Programmer Mode
120-Pin	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	
1	MD2	MD2	MD2	MD2	MD2	Vss
2	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
3	A0	A0	PC0/A0	PC0/A0	PC0/TIOCA9	A0
4	A1	A1	PC1/A1	PC1/A1	PC1/TIOCB9	A1
5	A2	A2	PC2/A2	PC2/A2	PC2/TIOCC9	A2
6	A3	A3	PC3/A3	PC3/A3	PC3/TIOCD9	A3
7	A4	A4	PC4/A4	PC4/A4	PC4/TIOCA10	A4
8	Vss	Vss	Vss	Vss	Vss	Vss
9	A5	A5	PC5/A5	PC5/A5	PC5/TIOCB10	A5
10	A6	A6	PC6/A6	PC6/A6	PC6/TIOCA11	A6
11	A7	A7	PC7/A7	PC7/A7	PC7/TIOCB11	A7
12	A8	A8	PB0/A8	PB0/A8	PB0/TIOCA6	A8
13	A9	A9	PB1/A9	PB1/A9	PB1/TIOCB6	A9
14	A10	A10	PB2/A10	PB2/A10	PB2/TIOCC6/ TCLKE	A10
15	A11	A11	PB3/A11	PB3/A11	PB3/TIOCD6/ TCLKF	A11
16	A12	A12	PB4/A12	PB4/A12	PB4/TIOCA7	A12
17	Vss	Vss	Vss	Vss	Vss	Vss
18	A13	A13	PB5/A13	PB5/A13	PB5/TIOCB7/ TCLKG	A13
19	A14	A14	PB6/A14	PB6/A14	PB6/TIOCA8	A14
20	A15	A15	PB7/A15	PB7/A15	PB7/TIOCB8/ TCLKH	A15
21	A16	A16	PA0/A16	PA0/A16	PA0	A16
22	Vss	Vss	Vss	Vss	Vss	Vss
23	A17	A17	PA1/A17/TxD4-B	PA1/A17/TxD4-B	PA1/TxD4-B	A17
24	A18	A18	PA2/A18/RxD4-B	PA2/A18/RxD4-B	PA2/RxD4-B	A18
25	A19	A19	PA3/A19/SCK4-B	PA3/A19/SCK4-B	PA3/SCK4-B	NC
26	A20/$\overline{\text{IRQ4-A}}$	A20/$\overline{\text{IRQ4-A}}$	PA4/A20/$\overline{\text{IRQ4-A}}$/ $\overline{\text{SCS0-B}}$	PA4/A20/$\overline{\text{IRQ4-A}}$/ $\overline{\text{SCS0-B}}$	PA4/$\overline{\text{IRQ4-A}}$/ $\overline{\text{SCS0-B}}$	NC
27	PA5/A21/$\overline{\text{IRQ5-A}}$/ SSCK0-B	PA5/A21/$\overline{\text{IRQ5-A}}$/ SSCK0-B	PA5/A21/$\overline{\text{IRQ5-A}}$/ SSCK0-B	PA5/A21/$\overline{\text{IRQ5-A}}$/ SSCK0-B	PA5/$\overline{\text{IRQ5-A}}$/ SSCK0-B	NC

Pin No.	Pin Name					
				Mode 7		Flash Memory Programmer Mode
120-Pin	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	
28	PA6/A22/ IRQ6-A/SSI0-B	PA6/A22/ IRQ6-A/SSI0-B	PA6/A22/ IRQ6-A/SSI0-B	PA6/A22/ IRQ6-A/SSI0-B	PA6/IRQ6-A/ SSI0-B	NC
29	PA7/A23/CS7/ IRQ7-A/SSO0-B	PA7/A23/CS7/ IRQ7-A/SSO0-B	PA7/A23/CS7/ IRQ7-A/SSO0-B	PA7/A23/CS7/ IRQ7-A/SSO0-B	PA7/IRQ7-A/ SSO0-B	NC
30	EMLE	EMLE	EMLE	EMLE	EMLE	Vss
31	WDTOVF	WDTOVF	WDTOVF	WDTOVF	WDTOVF	NC
32	NMI	NMI	NMI	NMI	NMI	Vcc
33	VCL	VCL	VCL	VCL	VCL	VCL
34	P10/DREQ0/ PO8/TIOCA0	P10/DREQ0/ PO8/TIOCA0	P10/DREQ0/ PO8/TIOCA0	P10/DREQ0/ PO8/TIOCA0	P10/DREQ0/ PO8/TIOCA0	NC
35	P11/DREQ1*[3]/ PO9/TIOCB0	P11/DREQ1*[3]/ PO9/TIOCB0	P11/DREQ1*[3]/ PO9/TIOCB0	P11/DREQ1*[3]/ PO9/TIOCB0	P11/DREQ1*[3]/ PO9/TIOCB0	NC
36	P12/TEND0/ PO10/TIOCC0/ TCLKA	P12/TEND0/ PO10/TIOCC0/ TCLKA	P12/TEND0/ PO10/TIOCC0/ TCLKA	P12/TEND0/ PO10/TIOCC0/ TCLKA	P12/TEND0/ PO10/TIOCC0/ TCLKA	OE
37	P13/TEND1/ PO11/TIOCD0/ TCLKB	P13/TEND1/ PO11/TIOCD0/ TCLKB	P13/TEND1/ PO11/TIOCD0/ TCLKB	P13/TEND1/ PO11/TIOCD0/ TCLKB	P13/TEND1/ PO11/TIOCD0/ TCLKB	CE
38	P14/DACK0/ PO12/TIOCA1/ SSO0-A	P14/DACK0/ PO12/TIOCA1/ SSO0-A	P14/DACK0/ PO12/TIOCA1/ SSO0-A	P14/DACK0/ PO12/TIOCA1/ SSO0-A	P14/DACK0/ PO12/TIOCA1/ SSO0-A	WE
39	P15/DACK1/ PO13/TIOCB1/ TCLKC/SSI0-A	P15/DACK1/ PO13/TIOCB1/ TCLKC/SSI0-A	P15/DACK1/ PO13/TIOCB1/ TCLKC/SSI0-A	P15/DACK1/ PO13/TIOCB1/ TCLKC/SSI0-A	P15/DACK1/ PO13/TIOCB1/ TCLKC/SSI0-A	NC
40	P16/PO14/ TIOCA2/ SSCK0-A	P16/PO14/ TIOCA2/ SSCK0-A	P16/PO14/ TIOCA2/ SSCK0-A	P16/PO14/ TIOCA2/ SSCK0-A	P16/PO14/ TIOCA2/ SSCK0-A	NC
41	P17/PO15/ TIOCB2/ TCLKD/SCS0-A	P17/PO15/ TIOCB2/ TCLKD/SCS0-A	P17/PO15/ TIOCB2/ TCLKD/SCS0-A	P17/PO15/ TIOCB2/ TCLKD/SCS0-A	P17/PO15/ TIOCB2/ TCLKD/SCS0-A	NC
42	P20/PO0-A/ TIOCA3-A/ TMRI0-A/PUPD+	P20/PO0-A/ TIOCA3-A/ TMRI0-A/PUPD+	P20/PO0-A/ TIOCA3-A/ TMRI0-A/PUPD+	P20/PO0-A/ TIOCA3-A/ TMRI0-A/PUPD+	P20/PO0-A/ TIOCA3-A/ TMRI0-A/PUPD+	NC
43	DrVcc	DrVcc	DrVcc	DrVcc	DrVcc	Vcc
44	USD+	USD+	USD+	USD+	USD+	NC
45	USD-	USD-	USD-	USD-	USD-	NC
46	DrVss	DrVss	DrVss	DrVss	DrVss	DrVss

Pin No.	Pin Name					
				Mode 7		Flash Memory Programmer Mode
120-Pin	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	
47	P25/$\overline{\text{WAIT-B}}$/ PO5-A/ TIOCB4-A/ TMO1-A/VBUS	P25/$\overline{\text{WAIT-B}}$/ PO5-A/ TIOCB4-A/ TMO1-A/VBUS	P25/$\overline{\text{WAIT-B}}$/ PO5-A/ TIOCB4-A/ TMO1-A/VBUS	P25/$\overline{\text{WAIT-B}}$/ PO5-A/ TIOCB4-A/ TMO1-A/VBUS	P25/ PO5-A/ TIOCB4-A/ TMO1-A/VBUS	NC
48	P26/PO6/ TIOCA5/SDA2/ $\overline{\text{ADTRG1}}$	P26/PO6/ TIOCA5/SDA2/ $\overline{\text{ADTRG1}}$	P26/PO6/ TIOCA5/SDA2/ $\overline{\text{ADTRG1}}$	P26/PO6/ TIOCA5/SDA2/ $\overline{\text{ADTRG1}}$	P26/PO6/ TIOCA5/SDA2/ $\overline{\text{ADTRG1}}$	NC
49	P27/PO7/ TIOCB5/SCL2	P27/PO7/ TIOCB5/SCL2	P27/PO7/ TIOCB5/SCL2	P27/PO7/ TIOCB5/SCL2	P27/PO7/ TIOCB5/SCL2	NC
50	P85/PO5-B/ TIOCB4-B/ TMO1-B/SCK3	P85/PO5-B/ TIOCB4-B/ TMO1-B/SCK3	P85/PO5-B/ TIOCB4-B/ TMO1-B/SCK3	P85/PO5-B/ TIOCB4-B/ TMO1-B/SCK3	P85/PO5-B/ TIOCB4-B/ TMO1-B/SCK3	NC
51	PE0/D0/AD0	PE0/D0/AD0	PE0/D0/AD0	PE0/D0/AD0	PE0	NC
52	PE1/D1/AD1	PE1/D1/AD1	PE1/D1/AD1	PE1/D1/AD1	PE1	NC
53	PE2/D2/AD2	PE2/D2/AD2	PE2/D2/AD2	PE2/D2/AD2	PE2	NC
54	PE3/D3/AD3	PE3/D3/AD3	PE3/D3/AD3	PE3/D3/AD3	PE3	NC
55	PE4/D4/AD4	PE4/D4/AD4	PE4/D4/AD4	PE4/D4/AD4	PE4	NC
56	PE5/D5/AD5	PE5/D5/AD5	PE5/D5/AD5	PE5/D5/AD5	PE5	NC
57	PE6/D6/AD6	PE6/D6/AD6	PE6/D6/AD6	PE6/D6/AD6	PE6	NC
58	Vss	Vss	Vss	Vss	Vss	Vss
59	D7/AD7	PE7/D7/AD7	PE7/D7/AD7	PE7/D7/AD7	PE7	NC
60	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
61	D8/AD8	D8/AD8	D8/AD8	D8/AD8	PD0	I/O0
62	D9/AD9	D9/AD9	D9/AD9	D9/AD9	PD1	I/O1
63	D10/AD10	D10/AD10	D10/AD10	D10/AD10	PD2	I/O2
64	D11/AD11	D11/AD11	D11/AD11	D11/AD11	PD3	I/O3
65	D12/AD12	D12/AD12	D12/AD12	D12/AD12	PD4	I/O4
66	D13/AD13	D13/AD13	D13/AD13	D13/AD13	PD5	I/O5
67	D14/AD14	D14/A14D	D14/AD14	D14/AD14	PD6	I/O6
68	D15/AD15	D15/AD15	D15/AD15	D15/AD15	PD7	I/O7
69	PF0/$\overline{\text{WAIT-A}}$/ $\overline{\text{OE-A}}$/ADTRG0-B/ $\overline{\text{SCS0-C}}$	PF0/$\overline{\text{WAIT-A}}$/ $\overline{\text{OE-A}}$/ADTRG0-B/ $\overline{\text{SCS0-C}}$	PF0/$\overline{\text{WAIT-A}}$/ $\overline{\text{OE-A}}$/ADTRG0-B/ $\overline{\text{SCS0-C}}$	PF0/$\overline{\text{WAIT-A}}$/ $\overline{\text{OE-A}}$/ADTRG0-B/ $\overline{\text{SCS0-C}}$	PF0/ ADTRG0-B/ $\overline{\text{SCS0-C}}$	NC
70	PF1/$\overline{\text{CS5}}$/$\overline{\text{UCAS}}$/ SSCK0-C	PF1/$\overline{\text{CS5}}$/$\overline{\text{UCAS}}$/ SSCK0-C	PF1/$\overline{\text{CS5}}$/$\overline{\text{UCAS}}$/ SSCK0-C	PF1/$\overline{\text{CS5}}$/$\overline{\text{UCAS}}$/ SSCK0-C	PF1/SSCK0-C	NC

Pin No.	Pin Name					
				Mode 7		Flash Memory Programmer Mode
120-Pin	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	
71	PF2/$\overline{CS6}$/ $\overline{LCAS}$/SSI0-C	PF2/$\overline{CS6}$/ $\overline{LCAS}$/SSI0-C	PF2/$\overline{CS6}$/ $\overline{LCAS}$/SSI0-C	PF2/$\overline{CS6}$/ $\overline{LCAS}$/SSI0-C	PF2/ SSI0-C	NC
72	PF3/$\overline{LWR}$/ SSO0-C	PF3/$\overline{LWR}$/ SSO0-C	PF3/$\overline{LWR}$/ SSO0-C	PF3/$\overline{LWR}$/ SSO0-C	PF3/ SSO0-C	NC
73	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	PF4	NC
74	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	PF5	NC
75	PF6/$\overline{AS}$/$\overline{AH}$	PF6/$\overline{AS}$/$\overline{AH}$	PF6/$\overline{AS}$/$\overline{AH}$	PF6/$\overline{AS}$/$\overline{AH}$	PF6	NC
76	PLLVcc	PLLVcc	PLLVcc	PLLVcc	PLLVcc	Vcc
77	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$
78	PLLVss	PLLVss	PLLVss	PLLVss	PLLVss	Vss
79	PF7/ϕ	PF7/ϕ	PF7/ϕ	PF7/ϕ	PF7/ϕ	NC
80	Vss	Vss	Vss	Vss	Vss	Vss
81	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL
82	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL
83	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
84	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
85	P83/PO3-B/ TIOCD3-B/ TMCI1-B/RxD3	P83/PO3-B/ TIOCD3-B/ TMCI1-B/RxD3	P83/PO3-B/ TIOCD3-B/ TMCI1-B/RxD3	P83/PO3-B/ TIOCD3-B/ TMCI1-B/RxD3	P83/PO3-B/ TIOCD3-B/ TMCI1-B/RxD3	NC
86	P81/PO1-B/ TIOCB3-B/ TMRI1-B/TxD3	P81/PO1-B/ TIOCB3-B/ TMRI1-B/TxD3	P81/PO1-B/ TIOCB3-B/ TMRI1-B/TxD3	P81/PO1-B/ TIOCB3-B/ TMRI1-B/TxD3	P81/PO1-B/ TIOCB3-B/ TMRI1-B/TxD3	NC
87	Vss	Vss	Vss	Vss	Vss	Vss
88	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	Vcc
89	PG0/$\overline{CS0}$	PG0/$\overline{CS0}$	PG0/$\overline{CS0}$	PG0/$\overline{CS0}$	PG0	NC
90	PG1/$\overline{CS1}$	PG1/$\overline{CS1}$	PG1/$\overline{CS1}$	PG1/$\overline{CS1}$	PG1	NC
91	PG2/$\overline{CS2}$/$\overline{RAS2}$*[3]	PG2/$\overline{CS2}$/$\overline{RAS2}$*[3]	PG2/$\overline{CS2}$/$\overline{RAS2}$*[3]	PG2/$\overline{CS2}$/$\overline{RAS2}$*[3]	PG2	NC
92	PG3/$\overline{CS3}$/$\overline{RAS3}$*[3]	PG3/$\overline{CS3}$/$\overline{RAS3}$*[3]	PG3/$\overline{CS3}$/$\overline{RAS3}$*[3]	PG3/$\overline{CS3}$/$\overline{RAS3}$*[3]	PG3	NC
93	AVcc	AVcc	AVcc	AVcc	AVcc	Vcc
94	Vref	Vref	Vref	Vref	Vref	Vcc
95	P40/$\overline{IRQ0-B}$/ AN0_0	P40/$\overline{IRQ0-B}$/ AN0_0	P40/$\overline{IRQ0-B}$/ AN0_0	P40/$\overline{IRQ0-B}$/ AN0_0	P40/$\overline{IRQ0-B}$/ AN0_0	NC
96	P41/$\overline{IRQ1-B}$/ AN1_0	P41/$\overline{IRQ1-B}$/ AN1_0	P41/$\overline{IRQ1-B}$/ AN1_0	P41/$\overline{IRQ1-B}$/ AN1_0	P41/$\overline{IRQ1-B}$/ AN1_0	NC

Pin No.	Pin Name					
				Mode 7		Flash Memory Programmer Mode
120-Pin	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	
97	P42/IRQ2-B/ AN2_0	P42/IRQ2-B/ AN2_0	P42/IRQ2-B/ AN2_0	P42/IRQ2-B/ AN2_0	P42/IRQ2-B/ AN2_0	NC
98	P43/IRQ3-B/ AN3_0	P43/IRQ3-B/ AN3_0	P43/IRQ3-B/ AN3_0	P43/IRQ3-B/ AN3_0	P43/IRQ3-B/ AN3_0	Vss
99	P44/IRQ4-B/ AN4_0	P44/IRQ4-B/ AN4_0	P44/IRQ4-B/ AN4_0	P44/IRQ4-B/ AN4_0	P44/IRQ4-B/ AN4_0	Vcc
100	P45/IRQ5-B/ AN5_0	P45/IRQ5-B/ AN5_0	P45/IRQ5-B/ AN5_0	P45/IRQ5-B/ AN5_0	P45/IRQ5-B/ AN5_0	Vss
101	P46/IRQ6-B/ AN6_0	P46/IRQ6-B/ AN6_0	P46/IRQ6-B/ AN6_0	P46/IRQ6-B/ AN6_0	P46/IRQ6-B/ AN6_0	NC
102	P47/IRQ7-B/ AN7_0	P47/IRQ7-B/ AN7_0	P47/IRQ7-B/ AN7_0	P47/IRQ7-B/ AN7_0	P47/IRQ7-B/ AN7_0	NC
103	P94/AN12_1/DA2	P94/AN12_1/DA2	P94/AN12_1/DA2	P94/AN12_1/DA2	P94/AN12_1/DA2	NC
104	P95/AN13_1/DA3	P95/AN13_1/DA3	P95/AN13_1/DA3	P95/AN13_1/DA3	P95/AN13_1/DA3	NC
105	AVss	AVss	AVss	AVss	AVss	Vss
106	PG4/BREQO-A/ CS4	PG4/BREQO-A/ CS4	PG4/BREQO-A/ CS4	PG4/BREQO-A/ CS4	PG4	NC
107	PG5/BACK-A	PG5/BACK-A	PG5/BACK-A	PG5/BACK-A	PG5	NC
108	PG6/BREQ-A	PG6/BREQ-A	PG6/BREQ-A	PG6/BREQ-A	PG6	NC
109	P50/BREQO-B/ IRQ0-A/PO0-B/ TIOCA3-B/ TMRI0-B/ TxD2/SDA3	P50/BREQO-B/ IRQ0-A/PO0-B/ TIOCA3-B/ TMRI0-B/ TxD2/SDA3	P50/BREQO-B/ IRQ0-A/PO0-B/ TIOCA3-B/ TMRI0-B/ TxD2/SDA3	P50/BREQO-B/ IRQ0-A/PO0-B/ TIOCA3-B/ TMRI0-B/ TxD2/SDA3	P50/IRQ0-A/ PO0-B/ TIOCA3-B/ TMRI0-B/ TxD2/SDA3	Vss
110	P51/BREQ-B/ IRQ1-A/PO2-B/ TIOCC3-B/ TMCI0-B/ RxD2/SCL3	P51/BREQ-B/ IRQ1-A/PO2-B/ TIOCC3-B/ TMCI0-B/ RxD2/SCL3	P51/BREQ-B/ IRQ1-A/PO2-B/ TIOCC3-B/ TMCI0-B/ RxD2/SCL3	P51/BREQ-B/ IRQ1-A/PO2-B/ TIOCC3-B/ TMCI0-B/ RxD2/SCL3	P51/IRQ1-A/ PO2-B/ TIOCC3-B/ TMCI0-B/ RxD2/SCL3	Vss
111	P52/BACK-B/ IRQ2-A/PO4-B/ TIOCA4-B/ TMO0-B/SCK2	P52/BACK-B/ IRQ2-A/PO4-B/ TIOCA4-B/ TMO0-B/SCK2	P52/BACK-B/ IRQ2-A/PO4-B/ TIOCA4-B/ TMO0-B/SCK2	P52/BACK-B/ IRQ2-A/PO4-B/ TIOCA4-B/ TMO0-B/SCK2	P52/ IRQ2-A/ PO4-B/ TIOCA4-B/ TMO0-B/SCK2	Vcc
112	P53/IRQ3-A/ ADTRG0-A	P53/IRQ3-A/ ADTRG0-A	P53/IRQ3-A/ ADTRG0-A	P53/IRQ3-A/ ADTRG0-A	P53/IRQ3-A/ ADTRG0-A	NC
113	P35/OE-B/ SCK1/SCL0	P35/OE-B/ SCK1/SCL0	P35/OE-B/ SCK1/SCL0	P35/OE-B/ SCK1/SCL0	P35/SCK1/ SCL0	NC
114	P34/SCK0/ SCK4-A/SDA0	P34/SCK0/ SCK4-A/SDA0	P34/SCK0/ SCK4-A/SDA0	P34/SCK0/ SCK4-A/SDA0	P34/SCK0/ SCK4-A/SDA0	NC

Pin No.	Pin Name					
				Mode 7		Flash Memory Programmer Mode
120-Pin	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	
115	P33/RxD1/SCL1	P33/RxD1/SCL1	P33/RxD1/SCL1	P33/RxD1/SCL1	P33/RxD1/SCL1	NC
116	P32/RxD0/ IrRxD/SDA1	P32/RxD0/ IrRxD/SDA1	P32/RxD0/ IrRxD/SDA1	P32/RxD0/ IrRxD/SDA1	P32/RxD0/ IrRxD/SDA1	Vcc
117	P31/TxD1	P31/TxD1	P31/TxD1	P31/TxD1	P31/TxD1	NC
118	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD	NC
119	MD0	MD0	MD0	MD0	MD0	Vss
120	MD1	MD1	MD1	MD1	MD1	Vss

1.4.3 Pin Functions

Table 1.5 Pin Functions

		Pin No.				
		H8S/2456, H8S/2456R		H8S/2454		
Type	Symbol	144-Pin	145-Pin	120-Pin	I/O	Function
Power supply	V_{CC}	4, 72, 98, 99	B2, N12, E11,E13	2, 60, 83, 84	Input	For connection to the power supply. V_{CC} pins should be connected to the system power supply.
	V_{SS}	2, 10, 18, 25, 50, 70, 95, 102	A1, E2, F4, H1, N5, M11 E10, F11	8, 17, 22, 58, 80, 87	Input	For connection to ground. V_{SS} pins should be connected to the system power supply (0 V).
	$PLLV_{CC}$	91	H13	76	Input	Power supply pin for the on-chip PLL oscillator.
	$PLLV_{SS}$	93	G13	78	Input	Ground pin for the on-chip PLL oscillator.
	VCL	41	L3	33	Output	This pin must not be connected to the power supply and should be connected to the V_{SS} pin via a 0.1-µF (recommended value) capacitor (place it close to pin).
	DrV_{CC}	52	L7	43	Input	Power supply pin for the USB on-chip transceiver. Pins should be connected to the system power supply.
	DrV_{SS}	55	M7	46	Input	Ground pin for the USB on-chip transceiver.

		Pin No.				
		H8S/2456, H8S/2456R		H8S/2454		
Type	Symbol	144-Pin	145-Pin	120-Pin	I/O	Function
Clock	XTAL	96	F12	81	Input	For connection to a crystal oscillator. See section 24, Clock Pulse Generator, for typical connection diagrams for a crystal resonator and external clock input.
	EXTAL	97	F13	82	Input	For connection to a crystal oscillator. The EXTAL pin can also input an external clock. See section 24, Clock Pulse Generator, for typical connection diagrams for a crystal resonator and external clock input.
	ϕ	94	F10	79	Output	Supplies the system clock to external devices.
	SDRAMϕ *1	36	M1	—	Output	When a synchronous DRAM is connected, this pin is connected to the CLK pin of the synchronous DRAM. For details, see section 6, Bus Controller (BSC).
Operating mode control	MD2	1	B1	1	Input	These pins set the operating mode. These pins should not be changed during operation.
	MD1	144	A2	120		
	MD0	143	A3	119		

		Pin No.				
		H8S/2456, H8S/2456R		H8S/2454		
Type	Symbol	144-Pin	145-Pin	120-Pin	I/O	Function
System control	$\overline{\text{RES}}$	92	G11	77	Input	Reset pin. When this pin is driven low, the chip is reset.
	$\overline{\text{STBY}}$	103	D13	88	Input	When this pin is driven low, a transition is made to hardware standby mode.
	EMLE	32	K1	30	Input	On-chip emulator enable pin. When the on-chip emulator is used, this pin should be fixed high. At this time, pins P53, PG4 to PG6, and $\overline{\text{WDTOVF}}$ are used exclusively by the on-chip emulator. Therefore, the corresponding pin functions of those pins are not available. When the on-chip emulator is not used, this pin should be fixed low.
	BSCANE*2	—	L9	—	Input	Boundary scan enable pin. When the boundary scan function is used, this pin should be fixed high. At this time, pins P53, PG4 to PG6, and $\overline{\text{WDTOVF}}$ are used exclusively for boundary scan. Therefore, the corresponding pin functions of those pins are not available. When the boundary scan function is not used, this pin should be fixed low.
Address bus	A23 to A0	31 to 26, 24 to 19, 17 to 11, 9 to 5	J3, K2, J1, K4, H3, J2, J4, G3, H2, G1, H4, G4, F1, G2, F3, E4, E1, F2, E3, D1, D3, D2, C3, C1	29 to 23, 21 to 18, 16 to 9, 7 to 3	Output	These pins output an address.

		Pin No.				
		H8S/2456, H8S/2456R		H8S/2454		
Type	Symbol	144-Pin	145-Pin	120-Pin	I/O	Function
Data bus	D15 to D0	80 to 73, 71, 69 to 63	K11, K12, L13, L11, M12, L12, N13, M13, N11, M11, N10, L9, M10, N9, K10, L8	68 to 61, 59, 57 to 51	Input/ output	These pins constitute a bidirectional data bus. When an address/data multiplexed I/O space is accessed, an address is also output.
Address/ data multiplexed bus	AD15 to AD0	80 to 73, 71, 69 to 63	K11, K12, L13, L11, M12, L12, N13, M13, N11, M11, N10, L9, M10, N9, K10, L8	68 to 61, 59, 57 to 51	Input/ output	These pins output an address, and input or output data.
Bus control	$\overline{CS7}$ to $\overline{CS0}$	38 to 35, 110 to 107	M2, N2, M1, L1, A13, A12, B13, C12	29, 71, 70, 106, 92 to 89	Output	Signals that select division areas 7 to 0 in the external address space
	$\overline{AS}$	90	G10	75	Output	When this pin is low, it indicates that address output on the address bus is valid.
	$\overline{AH}$	90	G10	75	Output	Signal for holding the address when an address/data multiplexed I/O space is being accessed.
	$\overline{RD}$	89	G12	74	Output	When this pin is low, it indicates that the external address space is being read.

		Pin No.				
		H8S/2456, H8S/2456R		H8S/2454		
Type	Symbol	144-Pin	145-Pin	120-Pin	I/O	Function
Bus control	$\overline{\text{HWR}}$	88	H11	73	Output	Strobe signal indicating that an external address space is to be written to, and the upper half (D15 to D8) of the data bus is enabled. Also functions as the write enable signal for accessing the DRAM space.
	$\overline{\text{LWR}}$	87	J13	72	Output	Strobe signal indicating that an external address space is to be written to, and the lower half (D7 to D0) of the data bus is enabled.
	$\overline{\text{BREQ-A}}$	132	D5	108	Input	The external bus master requests the bus to this LSI.
	$\overline{\text{BREQ-B}}$	134	B5	110		
	$\overline{\text{BREQO-A}}$	130	B6	106	Output	External bus request signal when the internal bus master accesses an external space in the external bus release state.
	$\overline{\text{BREQO-B}}$	133	A6	109		
	$\overline{\text{BACK-A}}$	131	C7	107	Output	Indicates the bus is released to the external bus master.
	$\overline{\text{BACK-B}}$	135	C6	111		
	$\overline{\text{UCAS}}$	85	H12	70	Output	Upper column address strobe signal for accessing the 16-bit DRAM space. Also functions as the column address strobe signal for accessing the 8-bit DRAM space.
	$\overline{\text{LCAS}}$	86	H10	71	Output	Lower column address strobe signal for accessing the 16-bit DRAM space.
	DQMU*1	85	H12	—	Output	Upper data mask enable signal for accessing the 16-bit continuous synchronous DRAM space. Also functions as the data mask enable signal for accessing the 8-bit continuous synchronous DRAM space.
	DQML*1	86	H10	—	Output	Lower-data mask enable signal for accessing the 16-bit continuous synchronous DRAM interface space.

Type	Symbol	Pin No. H8S/2456, H8S/2456R 144-Pin	Pin No. H8S/2456, H8S/2456R 145-Pin	Pin No. H8S/2454 120-Pin	I/O	Function
Bus control	$\overline{RAS}$/$\overline{RAS2}$	109	A12	91	Output	Row address strobe signal for the DRAM when the DRAM interface is set. $\overline{RAS}$ signal is a row address strobe signal when areas 2 to 5 are set as the continuous DRAM space.
	$\overline{RAS3}$	110	A13	92		
	$\overline{RAS4}$*3	35	L1	—		
	$\overline{RAS5}$*3	36	M1	—		
	$\overline{RAS}$*1	109	A12	—	Output	Row address strobe signal for the synchronous DRAM when the synchronous DRAM interface is set.
	$\overline{CAS}$*1	110	A13	—	Output	Column address strobe signal for the synchronous DRAM when the synchronous DRAM interface is set.
	$\overline{WE}$*1	35	L1	—	Output	Write enable signal for the synchronous DRAM when the synchronous DRAM interface is set.
	$\overline{WAIT\text{-}A}$	84	J11	69	Input	Requests insertion of a wait state in the bus cycles when accessing an external 3-state address space.
	$\overline{WAIT\text{-}B}$	56	N7	47		
	$\overline{OE\text{-}A}$	38	M2	69	Output	Output enable signal when accessing the DRAM space. The output pins of $\overline{OE}$ and ($\overline{OE}$) are selected by the port function control register 2 (PFCR2) of port 3.
	$\overline{OE\text{-}B}$	137	A5	113		
	CKE-A*1	38	M2	—	Output	Clock enable signal when the synchronous DRAM interface is set. The output pins of CKE and ($\overline{CKE}$) are selected by the port function control register 2 (PFCR2) of port 3.
	CKE-B*1	137	A5	—		

Type	Symbol	Pin No. H8S/2456, H8S/2456R 144-Pin	Pin No. H8S/2456, H8S/2456R 145-Pin	Pin No. H8S/2454 120-Pin	I/O	Function
Interrupts	$\overline{\text{NMI}}$	40	N1	32	Input	Nonmaskable interrupt request pin. This pin should be fixed high when not used.
	$\overline{\text{IRQ15-A}}$ to $\overline{\text{IRQ8-A}}$*3	86, 85, 106 to 104, 83 to 81	H10, H12, C13, D12, D10, J10, K13, J12	—		These pins request a maskable interrupt. The input pins of $\overline{\text{IRQn-A}}$ and $\overline{\text{IRQn-B}}$ are selected by the IRQ pin select register (ITSR) of the interrupt controller. (n = 0 to 15, m=0 to 8, 13 to 15 for the H8S/2456R Group and H8S/2456) (n = 0 to 7 for the H8S/2454 Group)
	$\overline{\text{IRQ7-A}}$ to $\overline{\text{IRQ0-A}}$	31 to 28, 136 to 133	J3, K2, J1, K4, D4, C6, B5, A6	29 to 26, 112 to 109		
	$\overline{\text{IRQ15-B}}$ to $\overline{\text{IRQ13-B}}$*3 $\overline{\text{IRQ8-B}}$*3	58 to 56 51	K7, L8, N7, L6,	—		
	$\overline{\text{IRQ7-B}}$ to $\overline{\text{IRQ0-B}}$	38, 37, 61 to 59, 34, 33, 3	M2, N2, M8, N8, K8, K3, L2, C2	102 to 95		
DMA controller (DMAC)	$\overline{\text{DREQ1}}$	82	K13	35	Input	These signals request DMAC activation.
	$\overline{\text{DREQ0}}$	81	J12	34		
	$\overline{\text{TEND1}}$	104	D10	37	Output	These signals indicate the end of DMAC data transfer.
	$\overline{\text{TEND0}}$	83	J10	36		
	$\overline{\text{DACK1}}$	106	C13	39	Output	DMAC single address transfer acknowledge signals.
	$\overline{\text{DACK0}}$	105	D12	38		
EXDMA controller (EXDMAC) *3	$\overline{\text{EDREQ3}}$	33	L2	—	Input	These signals request EXDMAC activation.
	$\overline{\text{EDREQ2}}$	3	C2			
	$\overline{\text{ETEND3}}$	59	K8	—	Output	These signals indicate the end of EXDMAC data transfer.
	$\overline{\text{ETEND2}}$	34	K3			
	$\overline{\text{EDACK3}}$	61	M8	—	Output	EXDMAC single address transfer acknowledge signals.
	$\overline{\text{EDACK2}}$	60	N8			
	$\overline{\text{EDRAK3}}$	49	L5	—	Output	These signals notify an external device of acceptance and start of execution of a DMA transfer request.
	$\overline{\text{EDRAK2}}$	48	K6			

Type	Symbol	Pin No. H8S/2456, H8S/2456R 144-Pin	Pin No. H8S/2456, H8S/2456R 145-Pin	Pin No. H8S/2454 120-Pin	I/O	Function
16-bit timer pulse unit (TPU)	TCLKH	22	H2	20	Input	External clock input pins of the timer.
	TCLKG	20	H4	18		
	TCLKF	17	F1	15		
	TCLKE	16	G2	14		
	TCLKD	49	L5	41		
	TCLKC	47	K5	39		
	TCLKB	45	M5	37		
	TCLKA	44	L4	36		
	TIOCA0	42	N3	34	Input/ output	TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
	TIOCB0	43	M4	35		
	TIOCC0	44	L4	36		
	TIOCD0	45	M5	37		
	TIOCA1	46	N4	38	Input/ output	TGRA_1 and TGRB_1 input capture input/output compare output/PWM output pins.
	TIOCB1	47	K5	39		
	TIOCA2	48	K6	40	Input/ output	TGRA_2 and TGRB_2 input capture input/output compare output/PWM output pins.
	TIOCB2	49	L5	41		
	TIOCA3-A	51	L6	42	Input/ output	TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.
	TIOCA3-B	133	A6	109		
	TIOCB3-B	33	L2	86		
	TIOCC3-B	134	B5	110		
	TIOCD3-B	59	K8	85		
	TIOCB4-A	56	N7	47	Input/ output	TGRA_4 and TGRB_4 input capture input/output compare output/PWM output pins.
	TIOCA4-B	135	C6	111		
	TIOCB4-B	61	M8	50		
	TIOCA5	57	L8	48	Input/ output	TGRA_5 and TGRB_5 input capture input/output compare output/PWM output pins.
	TIOCB5	58	K7	49		
	TIOCA6	14	E4	12	Input/ output	TGRA_6 to TGRD_6 input capture input/output compare output/PWM output pins.
	TIOCB6	15	F3	13		
	TIOCC6	16	G2	14		
	TIOCD6	17	F1	15		
	TIOCA7	19	G4	16	Input/ output	TGRA_7 and TGRB_7 input capture input/output compare output/PWM output pins.
	TIOCB7	20	H4	18		

		Pin No.				
		H8S/2456, H8S/2456R		H8S/2454		
Type	Symbol	144-Pin	145-Pin	120-Pin	I/O	Function
16-bit timer pulse unit (TPU)	TIOCA8	21	G1	19	Input/ output	TGRA_8 and TGRB_8 input capture input/output compare output/PWM output pins.
	TIOCB8	22	H2	20		
	TIOCA9	5	C1	3	Input/ output	TGRA_9 to TGRD_9 input capture input/output compare output/PWM output pins.
	TIOCB9	6	C3	4		
	TIOCC9	7	D2	5		
	TIOCD9	8	D3	6		
	TIOCA10	9	D1	7	Input/ output	TGRA_10 and TGRB_10 input capture input/output compare output/PWM output pins.
	TIOCB10	11	E3	9		
	TIOCA11	12	F2	10	Input/ output	TGRA_11 and TGRB_11 input capture input/output compare output/PWM output pins.
	TIOCB11	13	E1	11		
Program-mable pulse generator (PPG)	PO15 to PO8	49 to 42	L5, K6, K5,N4, M5, L4, M4, N3	41 to 34	Output	Pulse output pins.
	PO7	58	K7	49		
	PO6	57	L8	48		
	PO5-A	56	N7	47		
	PO0-A	51	L6	42		
	PO5-B	61	M8	50		
	PO4-B	135	C6	111		
	PO3-B	59	K8	85		
	PO2-B	134	B5	110		
	PO1-B	33	L2	86		
	PO0-B	133	A6	109		

		Pin No.				
		H8S/2456, H8S/2456R		H8S/2454		
Type	Symbol	144-Pin	145-Pin	120-Pin	I/O	Function
8-bit timer (TMR)	TMO0-A	105	D12	46	Output	Waveform output pins with output compare function.
	TMO1-A	106	C13	47		
	TMO0-B	135	C6	111		
	TMO1-B	61	M8	50		
	TMCI0-A	83	J10	44	Input	External event input pins.
	TMCI1-A	104	D10	45		
	TMCI0-B	134	B5	110		
	TMCI1-B	59	K8	85		
	TMRI0-A	81	J12	42	Input	Counter reset input pins.
	TMRI1-A	82	K13	43		
	TMRI0-B	133	A6	109		
	TMRI1-B	33	L2	86		
Watchdog timer (WDT)	$\overline{\text{WDTOVF}}$	39	M3	31	Output	Counter overflow signal output pin in watchdog timer mode.
Serial communi-cation interface (SCI)/ Smart Card interface (SCI_0 with IrDA function)	TxD4-B	24	J4	23	Output	Data output pins.
	TxD3	33	L2	86		
	TxD2	133	A6	109		
	TxD1	141	B3	117		
	TxD0/ IrTxD	142	C4	118		
	RxD4-B	26	J2	24	Input	Data input pins.
	RxD3	59	K8	85		
	RxD2	134	C5	110		
	RxD1	139	A4	115		
	RxD0/ IrRxD	140		116		
	SCK4-A	138	B4	114	Input/ output	Clock input/output pins.
	SCK4-B	27	H3	25		
	SCK3	61	M8	50		
	SCK2	135	C6	111		
	SCK1	137	A5	113		
	SCK0	138	B4	114		

		Pin No.				
		H8S/2456, H8S/2456R		H8S/2454		
Type	Symbol	144-Pin	145-Pin	120-Pin	I/O	Function
I²C bus interface 2 (IIC2)	SCL3	134	B5	110	Input/ output	I²C clock input/output pins.
	SCL2	58	K7	49		
	SCL1	139	C5	115		
	SCL0	137	A5	113		
	SDA3	133	A6	109	Input/ output	I²C data input/output pins.
	SDA2	57	L8	48		
	SDA1	140	A4	116		
	SDA0	138	B4	114		
Synchro-nous serial communi-cation unit (SSU)	SSO0-A	46	N4	38	Input/ output	Data input/output pins.
	SSO0-B	31	J3	29		
	SSI0-A	47	K5	39	Input/ output	Data input/output pins.
	SSI0-B	30	K2	28		
	SSCK0-A	48	K6	40	Input/ output	Clock input/output pins.
	SSCK0-B	29	J1	27		
	$\overline{\text{SCS0-A}}$	49	L5	41	Input/ output	Chip select input/output pins.
	$\overline{\text{SCS0-B}}$	28	K4	26		
Universal serial interface (USB)	USD+	53	M6	44	Input/ output	USB data input/output pin.
	USD-	54	N6	45		
	VBUS	56	N7	47	Input	USB cable connection/disconnection detection pin.
	PUPD+	51	L6	42	output	PULLUP control pin for D+signal.
Boundary scan*2 (JTAG)	$\overline{\text{TRST}}$	—	D4	—	Input	TAP controller reset pin.
	TMS	—	C7	—	Input	Control signal input pin for boundary scan.
	TDO	—	M3	—	Output	Data output pin for boundary scan.
	TDI	—	D5	—	Input	Data input pin for boundary scan.
	TCK	—	B6	—	Input	Clock input pin for boundary scan.

Type	Symbol	Pin No. H8S/2456, H8S/2456R 144-Pin	Pin No. H8S/2456, H8S/2456R 145-Pin	Pin No. H8S/2454 120-Pin	I/O	Function
A/D converter	AN15_1*[3]	128	D6	—	Input	Analog input pins.
	AN14_1*[3]	127	D7			
	AN13_1	126	D8	104	Input	Analog input pins.
	AN12_1	125	A8	103		
	AN11_1 to AN8_1*[3]	124 to 121	B7, C8, D9, A9	—	Input	Analog input pins.
	AN7_0 to AN0_0	120 to 113	B8, C9, B9, A10, C10, B10, C11, A11	102 to 95	Input	Analog input pins.
	$\overline{\text{ADTRG0-A}}$	136	D4	112	Input	Pin for input of an external trigger to start A/D conversion.
	$\overline{\text{ADTRG0-B}}$	84	J11	69		
	$\overline{\text{ADTRG1}}$	57	L8	48		
D/A converter	DA3	126	D8	104	Output	Analog output pins.
	DA2	125	A8	103		
A/D converter, D/A converter	AV_{CC}	111	B11	93	Input	Analog power-supply pin for the A/D converter and D/A converter. When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+3 V).
	AV_{SS}	129	A7	105	Input	Ground pin for the A/D converter and D/A converter. This pin should be connected to the system power supply (0 V).
	Vref	112	B12	94	Input	Reference voltage input pin for the A/D converter and D/A converter. When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+3 V).

		Pin No.				
		H8S/2456, H8S/2456R		H8S/2454		
Type	Symbol	144-Pin	145-Pin	120-Pin	I/O	Function
I/O ports	P17 to P10	49 to 42	L5, K6, K5, N4, M5, L4, M4, N3	41 to 34	Input/ output	8-bit input/output pins.
	P27 to P25	58 to 56	K7, L8, N7,	49 to 47	Input/ output	4-bit input/output pins.
	P20	51	L6	42		
	P35 to P30	137 to 142	A5, B4, C5, A4, B3, C4	113 to 118	Input/ output	6-bit input/output pins.
	P47 to P40	120 to 113	B8, C9, B9, A10, C10, B10, C11, A11	102 to 95	Input	8-bit input pins.
	P53 to P50	136 to 133	D4, C6, B5, A6	112 to 109	Input/ output	4-bit input/output pins.
	P65 to P60	106 to 104, 83 to 81	C13, D12, D10, J10, K13, J12	—	Input/ output	6-bit input/output pins.
	P85	61	M8	50	Input/ output	6-bit input/output pins in the H8S/2456 Group and H8S/2456R Group. 3-bit input/output pins in the H8S/2454 Group.
	P84*[3]	60	N8	—		
	P83	59	K8	85		
	P82*[3]	34	K3	—		
	P81	33	L2	86		
	P80*[3]	3	C2	—		
	P97*[3], P96*[3], P95, P94, P93 to P90*[3]	128 to 121	D6, D7, D8, A8, B7, C8, D9, A9	104, 103	Input	8-bit input/output pins in the H8S/2456 Group and H8S/2456R Group. 2-bit input/output pins in the H8S/2454 Group.
	PA7 to PA0	31 to 26, 24, 23	J3, K2, J1, K4, H3, J2, J4, G3	29 to 23, 21	Input/ output	8-bit input/output pins.
	PB7 to PB0	22 to 19, 17 to 14	H2, G1, H4, G4, F1, G2, F3, E4	20 to 18, 16 to 12	Input/ output	8-bit input/output pins.

Type	Symbol	Pin No. H8S/2456, H8S/2456R 144-Pin	Pin No. H8S/2456, H8S/2456R 145-Pin	Pin No. H8S/2454 120-Pin	I/O	Function
I/O ports	PC7 to PC0	13 to 11, 9 to 5	E1, F2, E3, D1, D3, D2, C3, C1	11 to 9, 7 to 3	Input/ output	8-bit input/output pins.
	PD7 to PD0	80 to 73	K11, K12, L13, L11, M12, L12, N13, M13	68 to 61	Input/ output	8-bit input/output pins.
	PE7 to PE0	71, 69 to 63	N11, M10, N10, K10, L10, M9, N9, K9	59, 57 to 51	Input/ output	8-bit input/output pins.
	PF7 to PF0	94, 90 to 84	F10, G10, G12, H11, J13, H10, H12, J11	79, 75 to 69	Input/ output	8-bit input/output pins.
	PG6 to PG0	132 to 130, 110 to 107	D5, C7, B6, A13, A12, B13, C12	108 to 106, 92 to 89	Input/ output	7-bit input/output pins.
	PH3 to PH0*[3]	38 to 35	M2, N2, M1, L1	—	Input/ output	4-bit input/output pins.
	PJ2*[3]*[4]	62	—	—	Input	3-bit input pins.
	PJ1*[3]	101	E12			
	PJ0*[3]	100	D11			

Notes: 1. Not supported by the H8S/2456 Group or H8S/2454 Group.
2. Can be used only in the 145-pin version.
3. Not supported by the H8S/2454 Group.
4. Can be used only in the 144-pin version.

Section 2 CPU

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

2.1 Features

- Upward-compatible with H8/300 and H8/300H CPUs
  Can execute H8/300 and H8/300H CPUs object programs
- General-register architecture
  Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-bit registers
- Sixty-nine basic instructions
  8/16/32-bit arithmetic and logic instructions
  Multiply and divide instructions
  Powerful bit-manipulation instructions
  Multiply-and-accumulate instruction
- Eight addressing modes
  Register direct [Rn]
  Register indirect [@ERn]
  Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  Immediate [#xx:8, #xx:16, or #xx:32]
  Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  Memory indirect [@@aa:8]
- 16-Mbyte address space
  Program: 16 Mbytes
  Data: 16 Mbytes
- High-speed operation
  All frequently-used instructions execute in one or two states
  8/16/32-bit register-register add/subtract: 1 state
  8 × 8-bit register-register multiply: 2 states

16 ÷ 8-bit register-register divide: 12 states
16 × 16-bit register-register multiply: 4 states
32 ÷ 16-bit register-register divide: 20 states

- Two CPU operating modes
  Normal mode*
  Advanced mode

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

- Power-down state
  Transition to power-down state by SLEEP instruction
  CPU clock speed selection

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

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

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

Instruction	Mnemonic	Execution States H8S/2600	Execution States H8S/2000
MULXU	MULXU.B Rs, Rd	2*	12
	MULXU.W Rs, ERd	2*	20
MULXS	MULXS.B Rs, Rd	3*	13
	MULXS.W Rs, ERd	3*	21
CLRMAC	CLRMAC	1*	Not supported
LDMAC	LDMAC ERs, MACH	1*	
	LDMAC ERs, MACL	1*	
STMAC	STMAC MACH, ERd	1*	
	STMAC MACL, ERd	1*	

Note: * The number of execution states is incremented following a MAC instruction.
In addition, there are differences in address space, CCR and EXR register functions, power-down modes, etc., depending on the model.

2.1.2 Differences from H8/300 CPU

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

- More general registers and control registers
  Eight 16-bit expanded registers, and one 8-bit and two 32-bit control registers, have been added.
- Expanded address space
  Normal mode supports the same 64-Kbyte address space as the H8/300 CPU.
  Advanced mode supports a maximum 16-Mbyte address space.
- Enhanced addressing
  The addressing modes have been enhanced to make effective use of the 16-Mbyte address space.
- Enhanced instructions
  Addressing modes of bit-manipulation instructions have been enhanced.
  Signed multiply and divide instructions have been added.
  A multiply-and-accumulate instruction has been added.
  Two-bit shift and rotate instructions have been added.
  Instructions for saving and restoring multiple registers have been added.
  A test and set instruction has been added.

- Higher speed
  Basic instructions execute twice as fast.

Note: Normal mode is not available in this LSI.

2.1.3 Differences from H8/300H CPU

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

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

2.2 CPU Operating Modes

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

2.2.1 Normal Mode

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

- Address Space

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

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

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

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

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

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

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

Note: Normal mode is not available in this LSI.

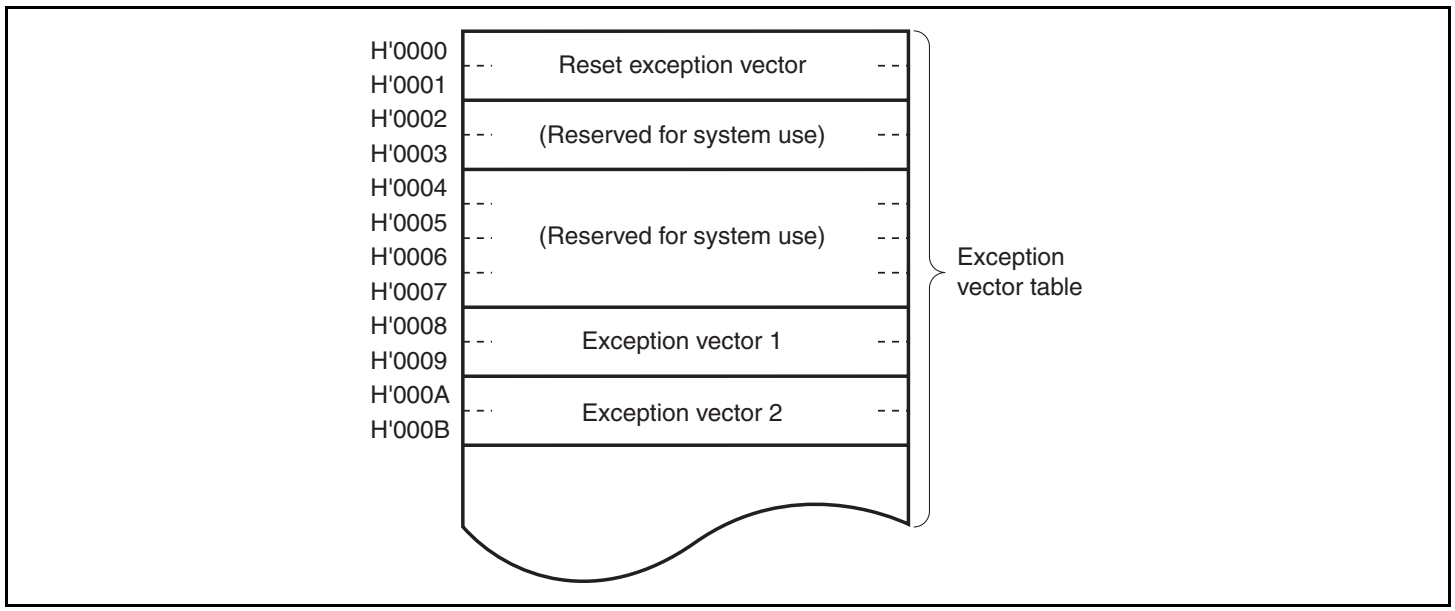


Figure 2.1 Exception Vector Table (Normal Mode)

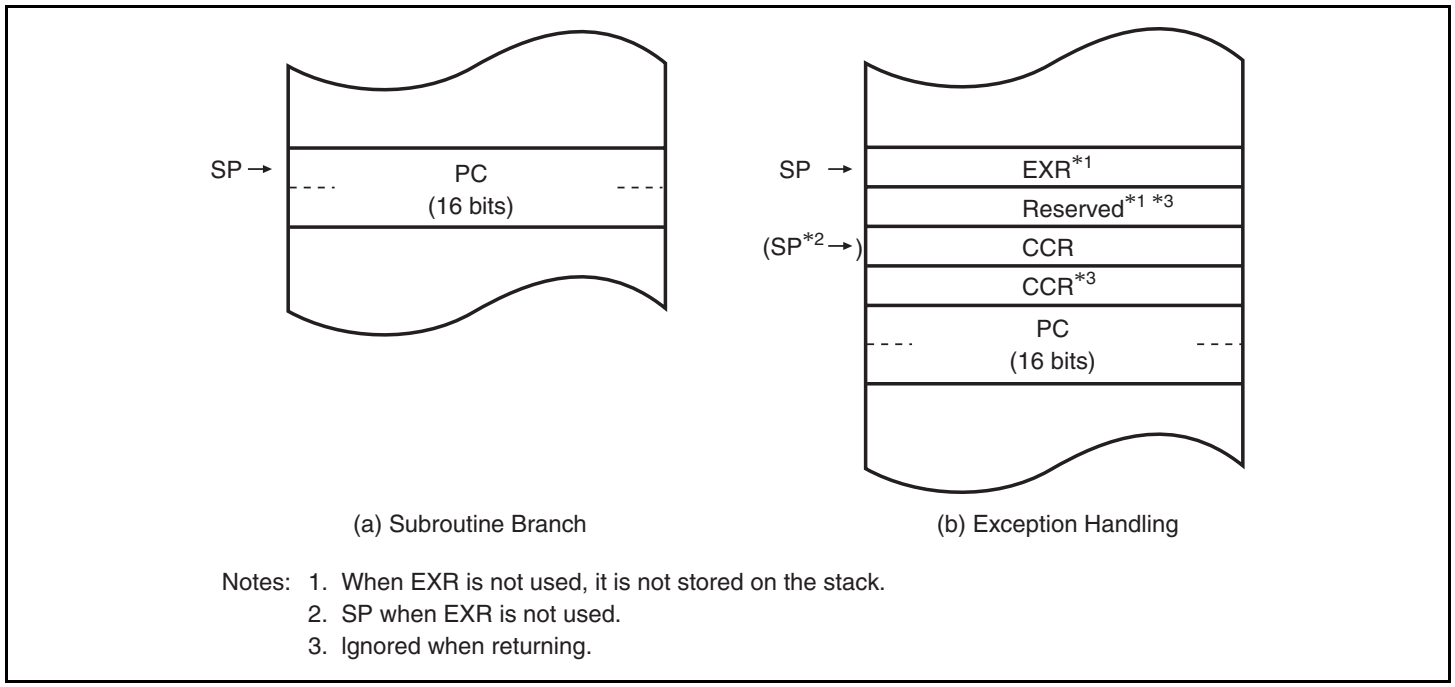


Figure 2.2 Stack Structure in Normal Mode

2.2.2 Advanced Mode

- Address Space

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

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

  All instructions and addressing modes can be used.
- Exception Vector Table and Memory Indirect Branch Addresses

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

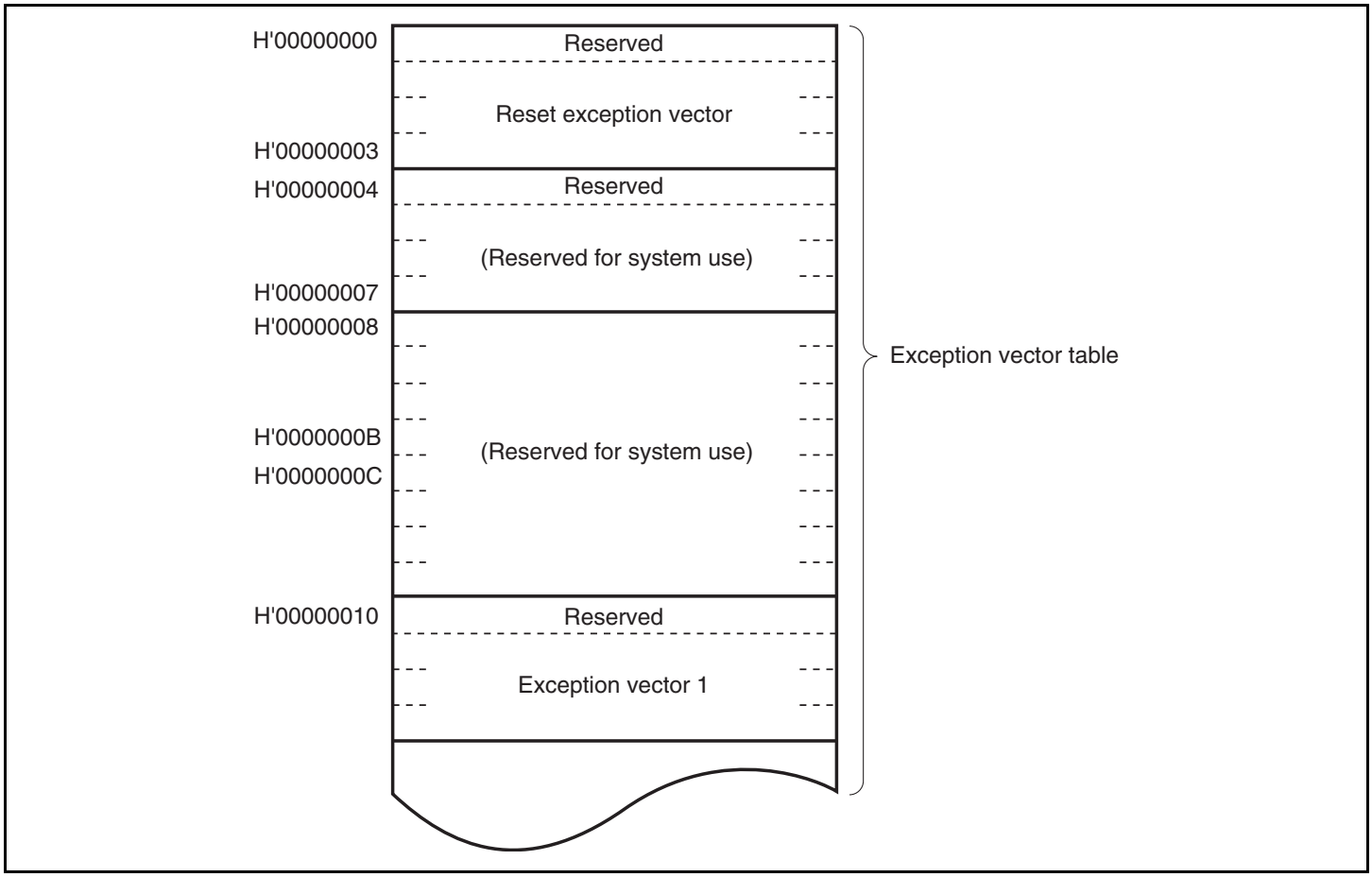


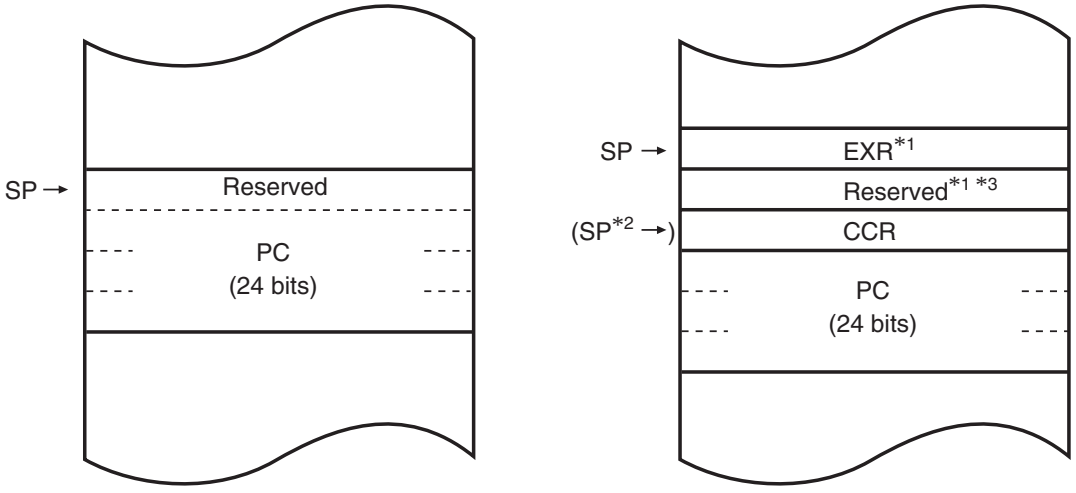
Figure 2.3 Exception Vector Table (Advanced Mode)

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

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

- Stack Structure

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



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

Figure 2.4 Stack Structure in Advanced Mode

2.3 Address Space

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

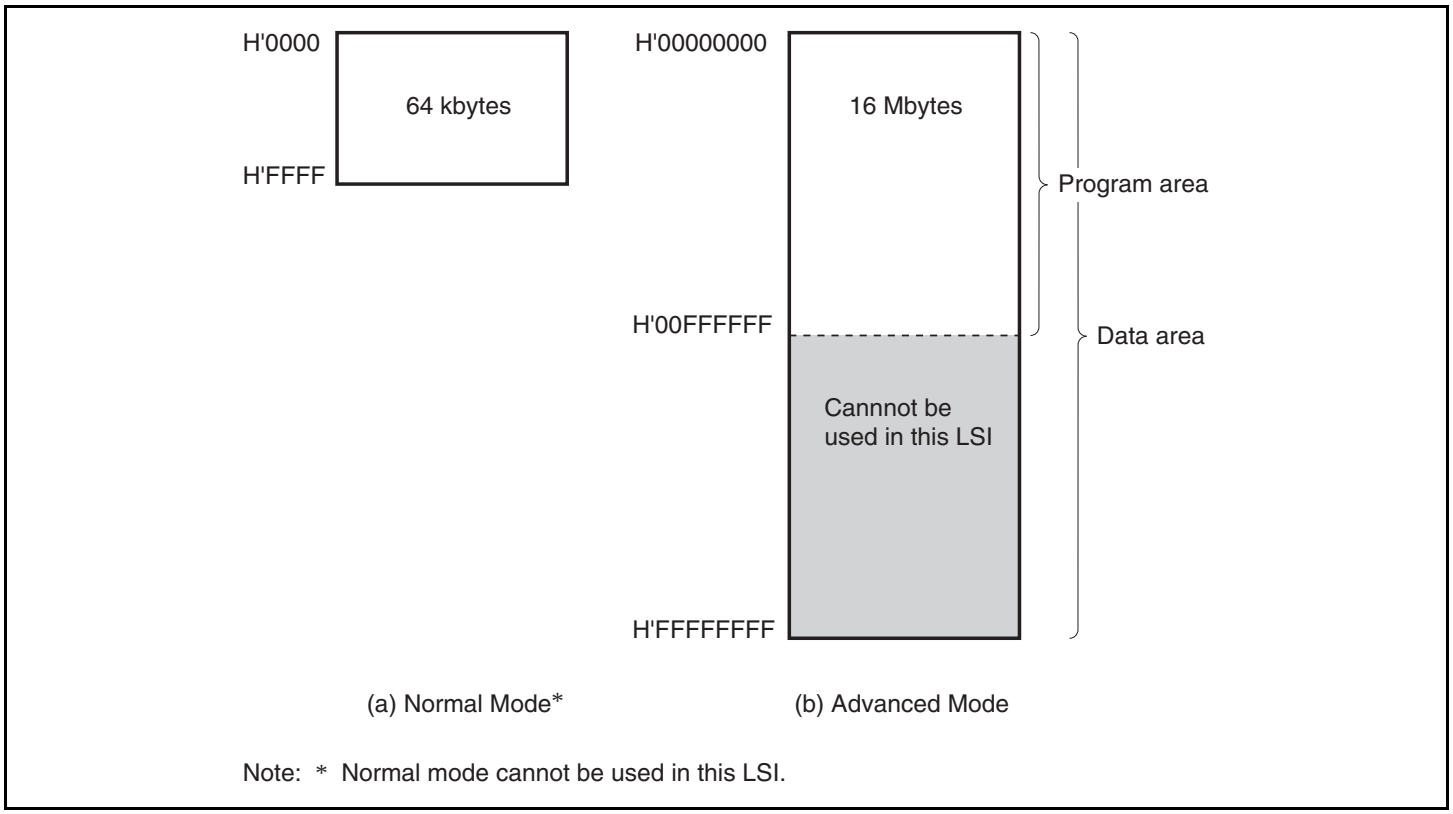


Figure 2.5 Memory Map

Note: Normal mode is not available in this LSI.

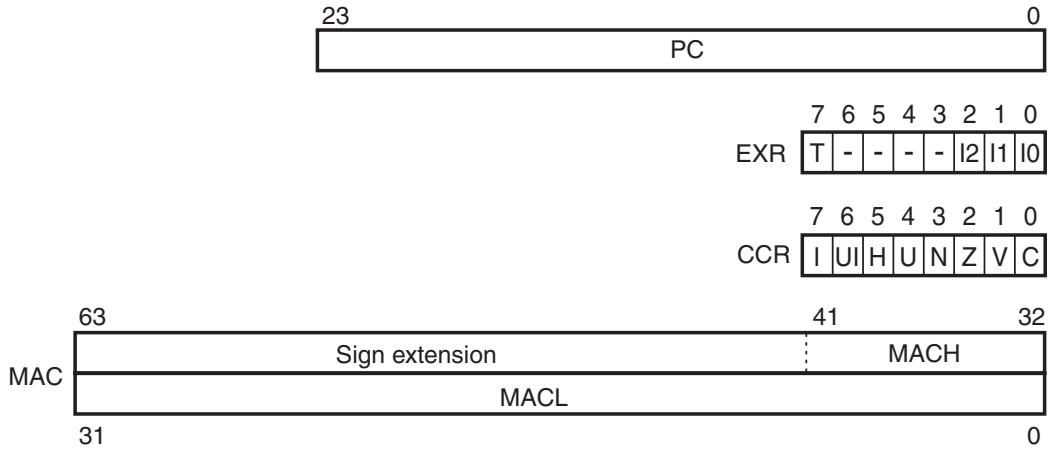
2.4 Registers

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

General Registers (Rn) and Extended Registers (En)

	15 — 0	7 — 0	7 — 0
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 (CR)



Legend:

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

Note: * UI cannot be used as an interrupt mask bit in this LSI.

Figure 2.6 CPU Registers

2.4.1 General Registers

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

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

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

The usage of each register can be selected independently.

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

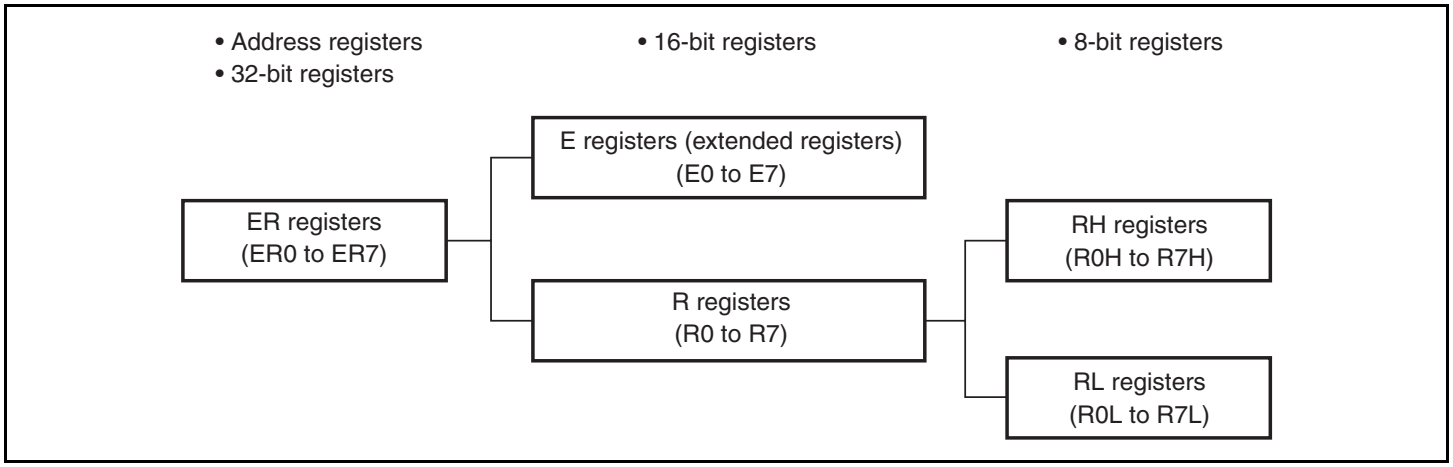


Figure 2.7 Usage of General Registers

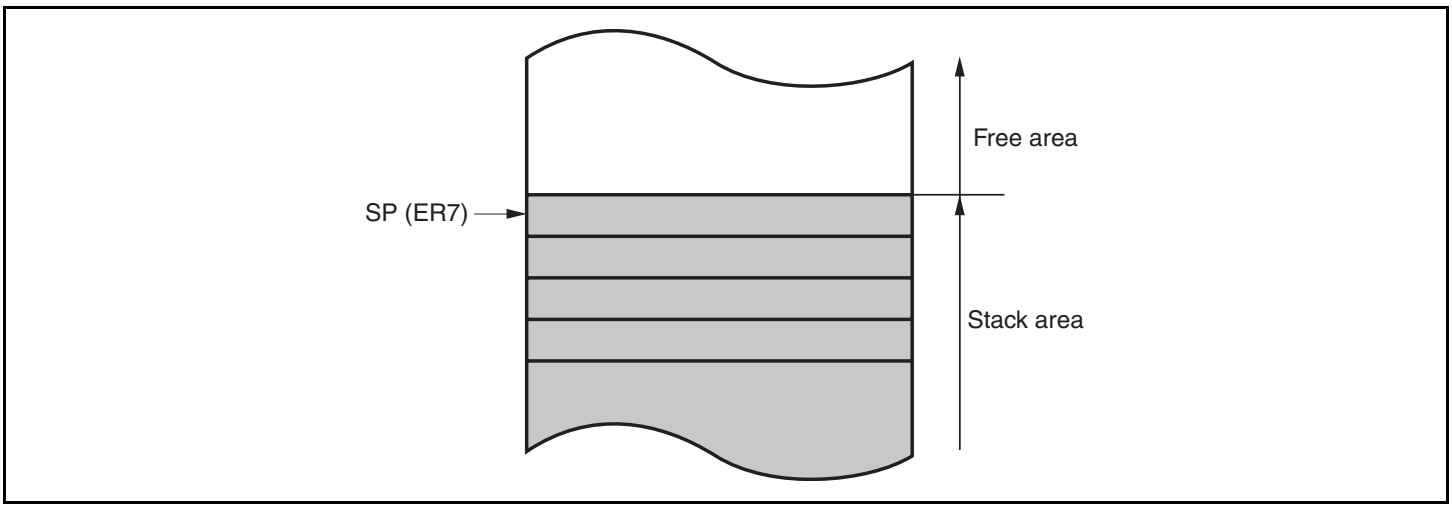


Figure 2.8 Stack

2.4.2 Program Counter (PC)

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

2.4.3 Extended Register (EXR)

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

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

2.4.4 Condition-Code Register (CCR)

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

Operations can be performed on the CCR bits by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as branching conditions for conditional branch (Bcc) instructions.

Bit	Bit Name	Initial Value	R/W	Description
7	I	1	R/W	Interrupt Mask Bit Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 by hardware at the start of an exception-handling sequence. For details, see section 5, Interrupt Controller.
6	UI	Undefined	R/W	User Bit or Interrupt Mask Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit cannot be used as an interrupt mask bit in this LSI.
5	H	Undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.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, the H 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, the H 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 and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of data as a sign bit.

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

2.4.5 Multiply-Accumulate Register (MAC)

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

2.4.6 Initial Values of CPU Internal Registers

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

2.5 Data Formats

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

2.5.1 General Register Data Formats

Figure 2.9 shows the data formats in general registers.

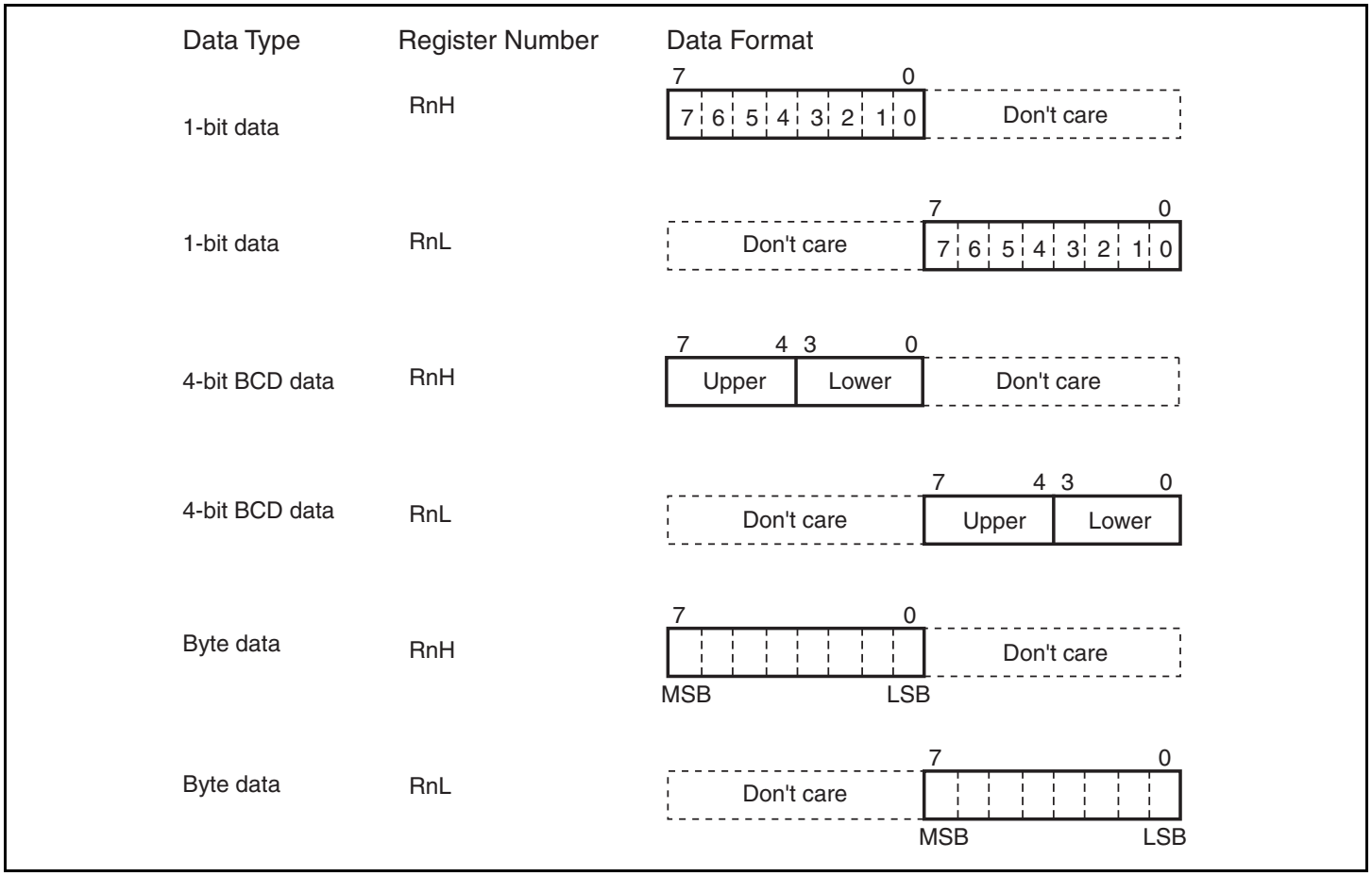


Figure 2.9 General Register Data Formats (1)

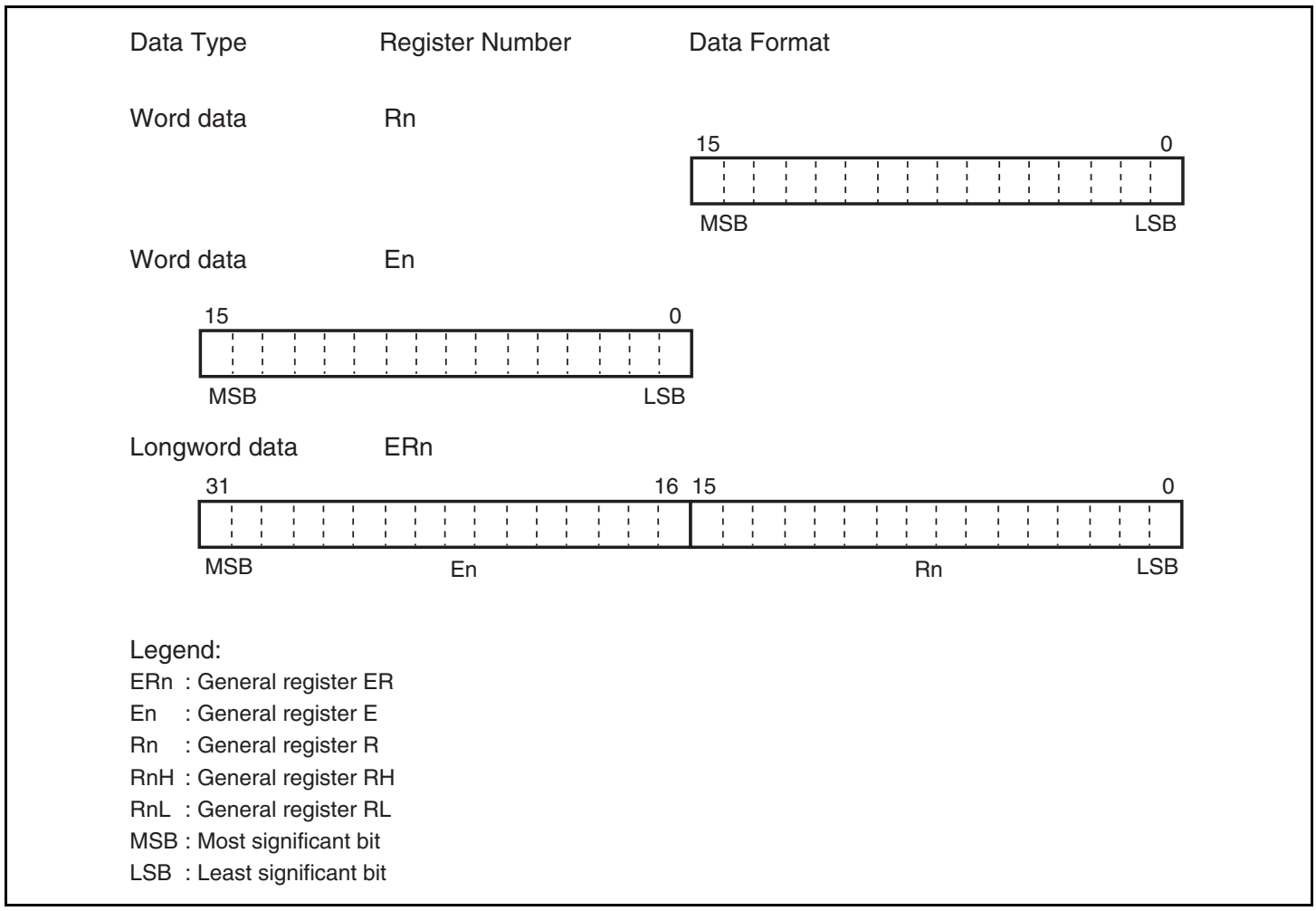


Figure 2.9 General Register Data Formats (2)

2.5.2 Memory Data Formats

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

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

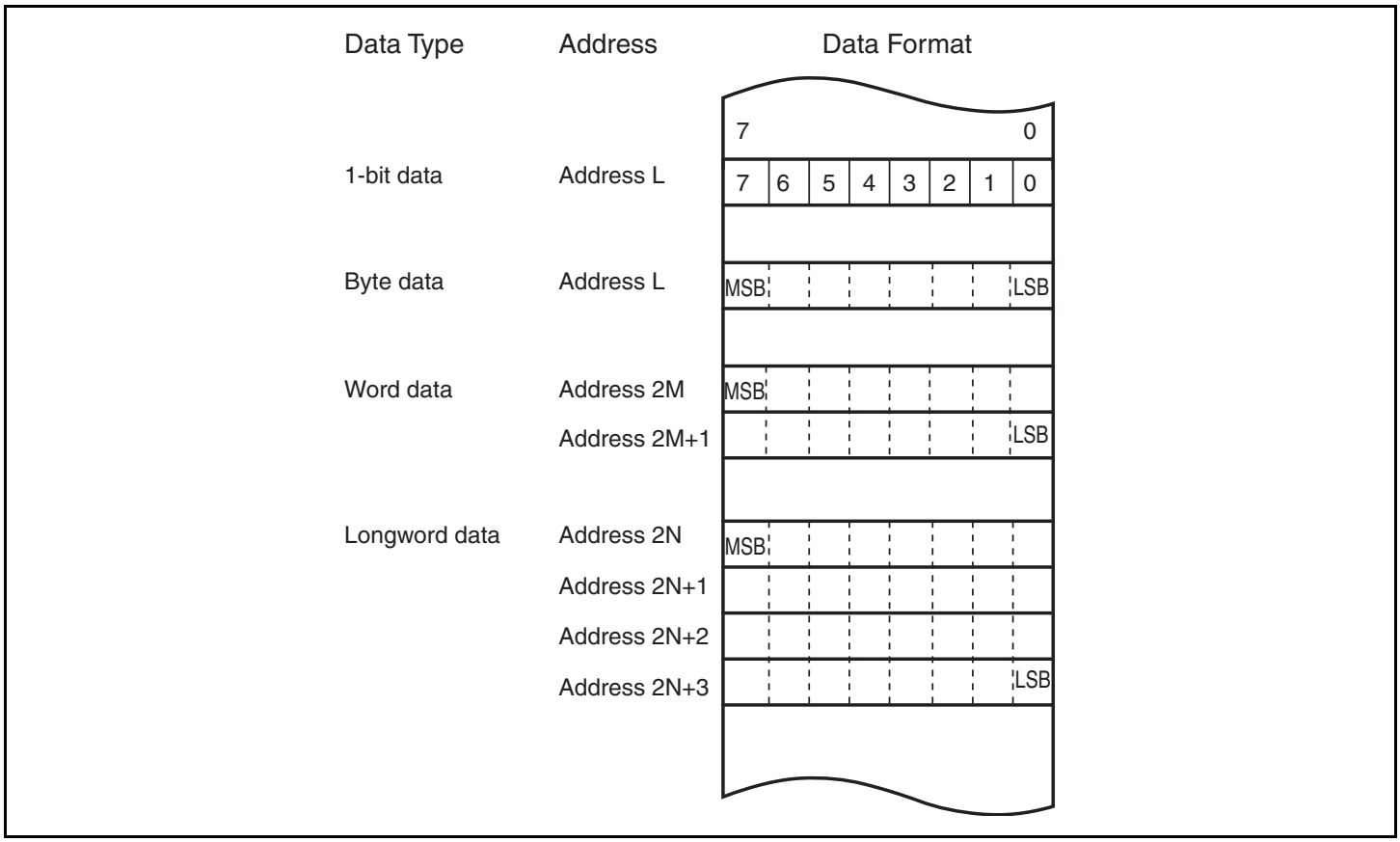


Figure 2.10 Memory Data Formats

2.6 Instruction Set

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

Table 2.1 Instruction Classification

Function	Instructions	Size	Types
Data transfer	MOV	B/W/L	5
	POP*1, PUSH*1	W/L	
	LDM, STM	L	
	MOVFPE*3, MOVTPE*3	B	
Arithmetic operations	ADD, SUB, CMP, NEG	B/W/L	23
	ADDX, SUBX, DAA, DAS	B	
	INC, DEC	B/W/L	
	ADDS, SUBS	L	
	MULXU, DIVXU, MULXS, DIVXS	B/W	
	EXTU, EXTS	W/L	
	TAS*4	B	
	MAC, LDMAC, STMAC, CLRMAC	—	
Logic operations	AND, OR, XOR, NOT	B/W/L	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	B/W/L	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	Bcc*2, JMP, BSR, JSR, RTS	—	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—	9
Block data transfer	EEPMOV	—	1
		Total:	69

Legend:
B: Byte
W: Word
L: Longword

Notes: 1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W Rn, @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.
2. Bcc is the general name for conditional branch instructions.

3. Cannot be used in this LSI.
4. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

2.6.1 Table of Instructions Classified by Function

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

Table 2.2 Operation Notation

Symbol	Description
Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
MAC	Multiply-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR

Symbol	Description
→	Move
~	NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

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

Table 2.3 Data Transfer Instructions

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

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

Table 2.4 Arithmetic Operations Instructions (1)

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

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

Table 2.4 Arithmetic Operations Instructions (2)

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

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

Table 2.5 Logic Operations Instructions

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

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

Table 2.6 Shift Instructions

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

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

Table 2.7 Bit Manipulation Instructions (1)

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

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

Table 2.7 Bit Manipulation Instructions (2)

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

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

Table 2.8 Branch Instructions

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

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

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

Table 2.9 System Control Instructions

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

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

Table 2.10 Block Data Transfer Instructions

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

2.6.2 Basic Instruction Formats

The H8S/2600 Series instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op), a register field (r), an effective address extension (EA), and a condition field (cc).

Figure 2.11 shows examples of instruction formats.

- Operation Field

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

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

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

  Specifies the branching condition of Bcc instructions.

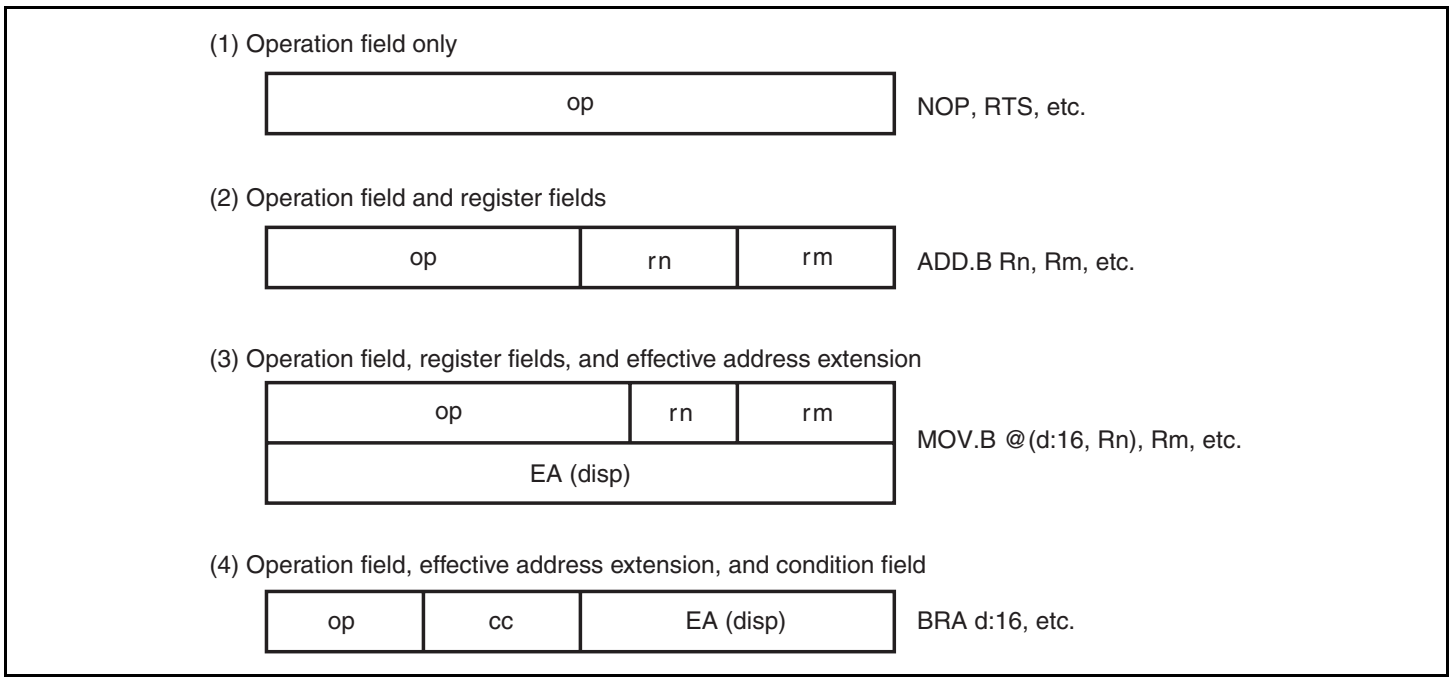


Figure 2.11 Instruction Formats (Examples)

2.7 Addressing Modes and Effective Address Calculation

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

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

Table 2.11 Addressing Modes

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

2.7.1 Register Direct—Rn

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

2.7.2 Register Indirect—@ERn

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

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

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

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

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

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

2.7.5 Absolute Address—@aa:8 /@aa:16 / @aa:24 /@aa:32

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

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

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

Table 2.12 Absolute Address Access Ranges

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

Note: * Not available in this LSI.

2.7.6 Immediate—#xx:8 / #xx:16/ #xx:32

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

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

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

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

2.7.8 Memory Indirect—@@aa:8

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

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

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

Note: Normal mode is not available in this LSI.

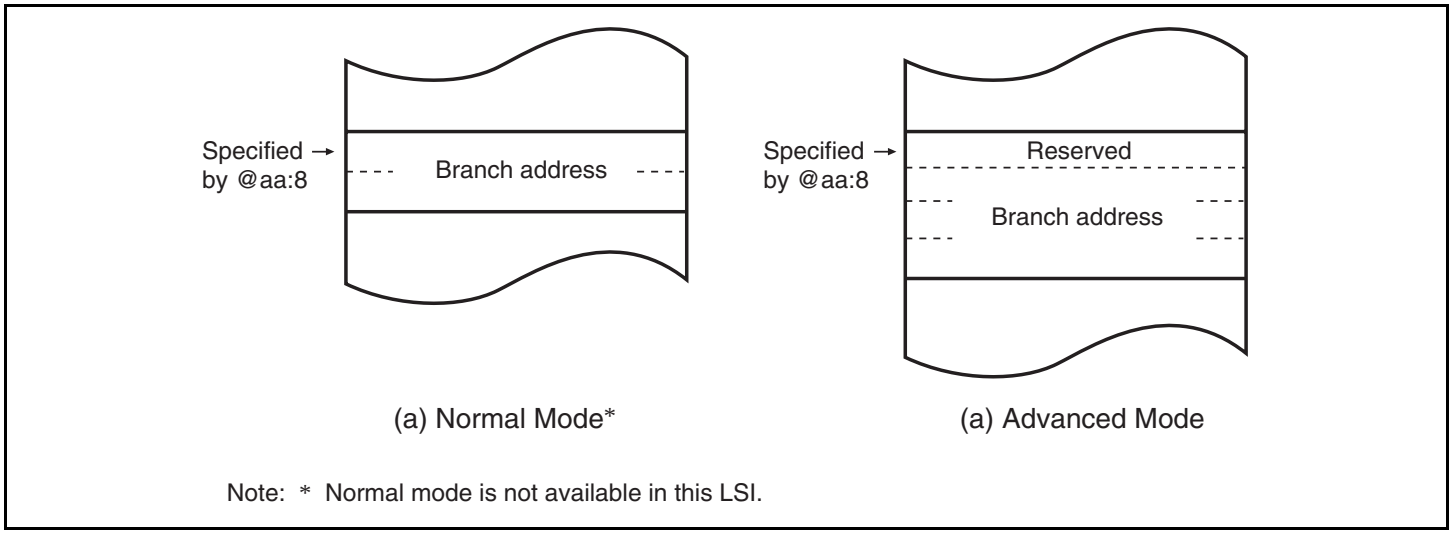


Figure 2.12 Branch Address Specification in Memory Indirect Mode

2.7.9 Effective Address Calculation

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

Note: Normal mode is not available in this LSI.

Table 2.13 Effective Address Calculation (1)

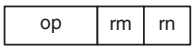
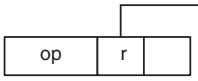
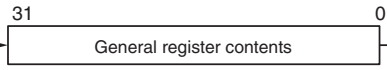
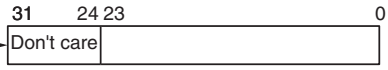
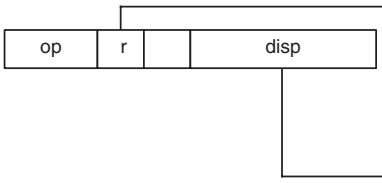
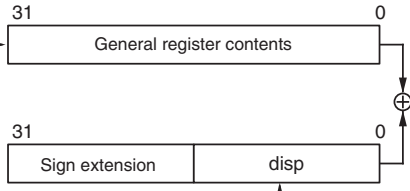
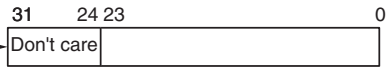
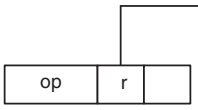
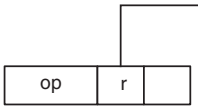
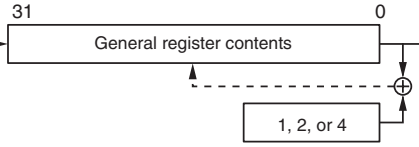
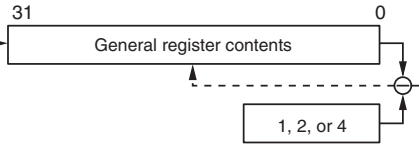
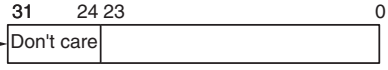
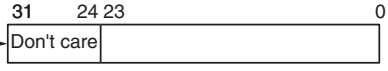
No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct (Rn) 		Operand is general register contents.
2	Register indirect (@ERn) 		
3	Register indirect with displacement @(d:16,ERn) or @(d:32,ERn) 		
4	Register indirect with post-increment or pre-decrement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

Table 2.13 Effective Address Calculation (2)

No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
5	Absolute address @aa:8 op abs		31 24 23 8 7 0 Don't care H'FFFF
	@aa:16 op abs		31 24 23 16 15 0 Don't care Sign extension
	@aa:24 op abs		31 24 23 0 Don't care
	@aa:32 op abs		31 24 23 0 Don't care
6	Immediate #xx:8/#xx:16/#xx:32 op IMM		Operand is immediate data.
7	Program-counter relative @(d:8,PC)/@(d:16,PC) op disp	23 0 PC contents 23 0 Sign extension disp	31 24 23 0 Don't care
8	Memory indirect @@aa:8 • Normal mode* op abs	31 8 7 0 H'000000 abs 15 0 Memory contents	31 24 23 16 15 0 Don't care H'00
	• Advanced mode op abs	31 8 7 0 H'000000 abs 31 0 Memory contents	31 24 23 0 Don't care

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

2.8 Processing States

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

- Reset State

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

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

- Exception-Handling State

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

- Program Execution State

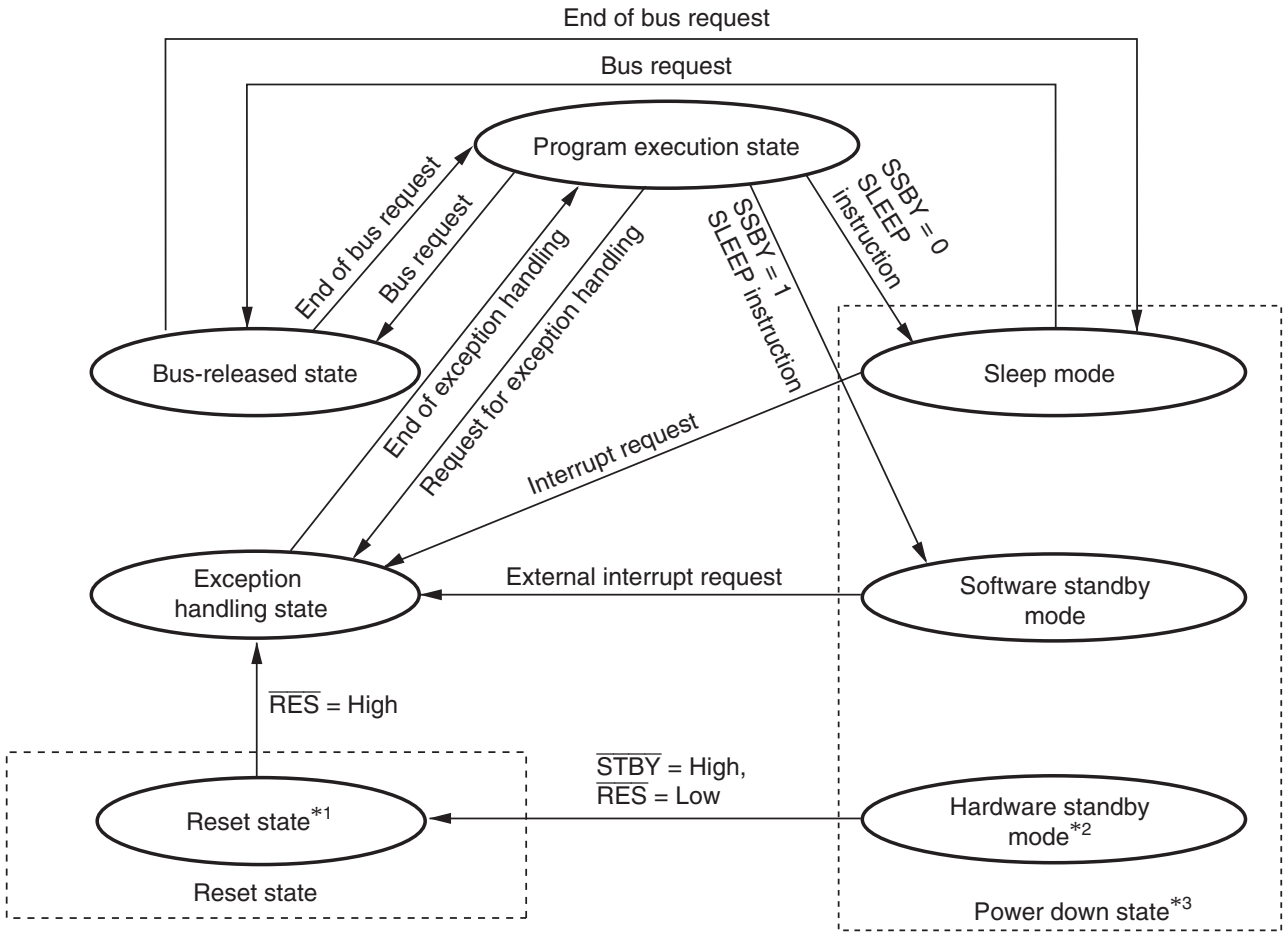
  In this state the CPU executes program instructions in sequence.

- Bus-Released State

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

- Program stop state

  This is a power-down state in which the CPU stops operating. The program stop state occurs when a SLEEP instruction is executed or the CPU enters hardware standby mode. For further details, see section 25, Power-Down Modes.



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever the $\overline{RES}$ pin goes low. A transition can also be made to the reset state when the watchdog timer overflows.
2. In every state, when the $\overline{STBY}$ pin becomes low, the hardware standby mode is entered.
3. For details, refer to section 25, Power-Down Modes.

Figure 2.13 State Transitions

2.9 Usage Note

2.9.1 Usage Notes on Bit-wise Operation Instructions

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

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

Section 3 MCU Operating Modes

3.1 Operating Mode Selection

The H8S/2456 Group, H8S/2454 Group, and H8S/2456R Group have five operating modes (modes 1 to 4 and 7). The operating mode is selected by the setting of mode pins (MD2 to MD0).

Modes 1, 2, and 4 are externally expanded modes in which the CPU can access an external memory and peripheral devices. In an externally expanded mode, the external address space can be designated as an 8-bit or 16-bit address space for each area by the bus controller at the beginning of program execution. If a 16-bit address space is designated for any one area, the 16-bit bus mode is selected. If an 8-bit address space is designated for all areas, the 8-bit bus mode is selected.

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

Mode 3 is a boot mode in which the flash memory can be programmed or erased. For details on the boot mode, refer to section 22, Flash Memory.

The settings for pins MD2 to MD0 should not be changed during LSI operation.

Table 3.1 MCU Operating Modes

MCU Operating Mode	MD2	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM	External Data Bus	
							Initial Value	Max. Value
1*	0	0	1	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
2*	0	1	0	Advanced	Expanded mode with on-chip ROM disabled	Disabled	8 bits	16 bits
3	0	1	1	Advanced	Boot mode	Enabled	—	16 bits
4	1	0	0	Advanced	Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7	1	1	1	Advanced	Single-chip mode	Enabled	—	16 bits

Note: * Only modes 1 and 2 may be used in ROM-less versions.

3.2 Register Descriptions

The following registers are related to operating mode setting.

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

3.2.1 Mode Control Register (MDCR)

MDCR monitors the current operating mode of this LSI.

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

Note: * Determined by the settings of pins MD2 to MD0.

3.2.2 System Control Register (SYSCR)

SYSCR selects saturation operation for the MAC instruction, controls CPU access to the flash memory control registers, sets the external bus mode, and enables or disables on-chip RAM.

Bit	Bit Name	Initial Value	R/W	Descriptions
7, 6	—	All 1	R/W	Reserved The initial value should not be modified.
5	MACS	0	R/W	MAC Saturation Operation Control Selects either saturation operation or non-saturation operation for the MAC instruction. 0: MAC instruction performs non-saturation operation 1: MAC instruction performs saturation operation
4	—	0	R/W	Reserved The initial value should not be modified.
3	FLSHE	0	R/W	Flash Memory Control Register Enable Controls CPU access to the flash memory control registers (FLMCR1, DFPR, and FLMSTR). If this bit is set to 1, the flash memory control registers can be read from and written to. If this bit is cleared to 0, the flash memory control registers are not selected. At this time, the contents of the flash memory control registers are retained. 0 should be written to this bit in LSIs other than the flash memory version. 0: Flash memory control registers are not selected for addresses H'FFFEB0 to H'FFFEB3 1: Flash memory control registers are selected for addresses H'FFFEB0 to H'FFFEB3
2	—	0	—	Reserved This bit is always read as 0 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Descriptions
1	EXPE	—	R/W	External Bus Mode Enable
0	RAME	1	R/W	RAM Enable Enables or disables the on-chip RAM. This bit is initialized when the reset state is canceled. 0: On-chip RAM is disabled 1: On-chip RAM is enabled

3.3 Operating Mode Descriptions

3.3.1 Mode 1

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled.

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

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

3.3.2 Mode 2

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled.

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

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

3.3.3 Mode 3

This mode is a boot mode of the flash memory. This mode is the same as mode 7, except for the programming and erasure of the flash memory. Mode 3 is only available in the flash memory version.

3.3.4 Mode 4

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

Ports A to C function as input ports immediately after a reset, but can be set to function as an address bus depending on each port register setting. Port D functions as a data bus and parts of ports F to H function as bus control signals. For details on function switching of ports A to C, see section 10, I/O Ports.

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

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

3.3.5 Mode 7

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

The initial mode immediately after a reset is single-chip mode, with all I/O ports available for use as input/output ports. However, setting the EXPE bit in SYSCR to 1 switches the mode to an externally expanded mode in which the external address spaces are enabled. When an externally expanded mode is selected, all areas are initially designated as a 16-bit access space. The functions of pins in ports A to H are the same as those in an externally expanded mode with on-chip ROM enabled.

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

3.3.6 Pin Functions

Table 3.2 shows the pin functions in each operating mode.

Table 3.2 Pin Functions in Each Operating Mode

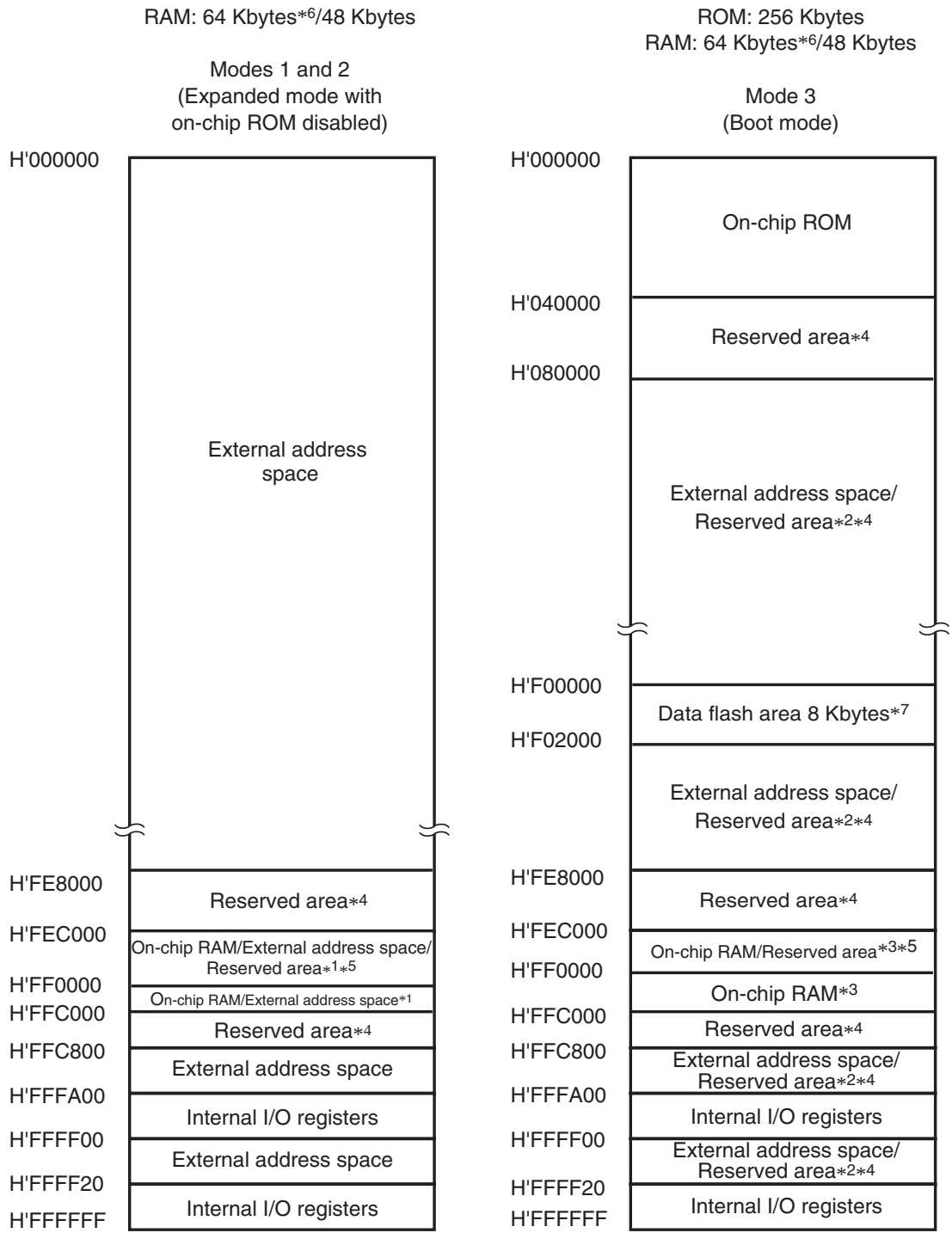
Port		Mode 1	Mode 2	Mode 3	Mode 4	Mode 7
Port A	PA7 to PA5	P*/A	P*/A	P*/A	P*/A	P*/A
	PA4 to PA0	A	A			
Port B		A	A	P*/A	P*/A	P*/A
Port C		A	A	P*/A	P*/A	P*/A
Port D		D	D	P*/D	D	P*/D
Port E		P/D*	P*/D	P*/D	P*/D	P*/D
Port F	PF7, PF6	P/C*	P/C*	P*/C	P/C*	P*/C
	PF5, PF4	C	C		C	
	PF3	P/C*	P/C*		P/C*	
	PF2 to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

Legend: P: I/O port
A: Address bus output
D: Data bus input/output
C: Control signals, clock input/output
*: Immediately after a reset

Note: Port H is not supported in the H8S/2454 Group.

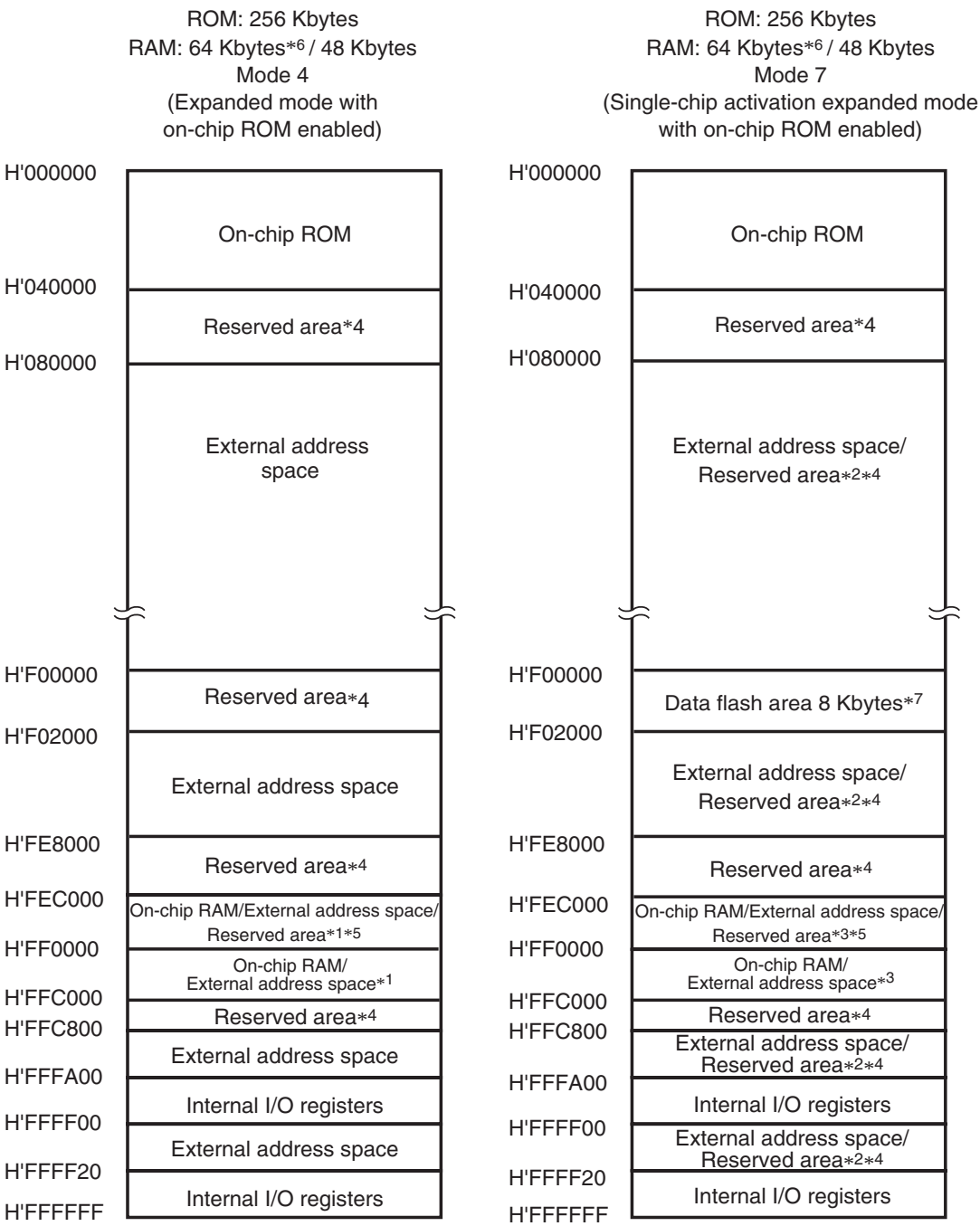
3.4 Memory Map in Each Operating Mode

Figures 3.1 to 3.5 show memory maps in each operating mode.



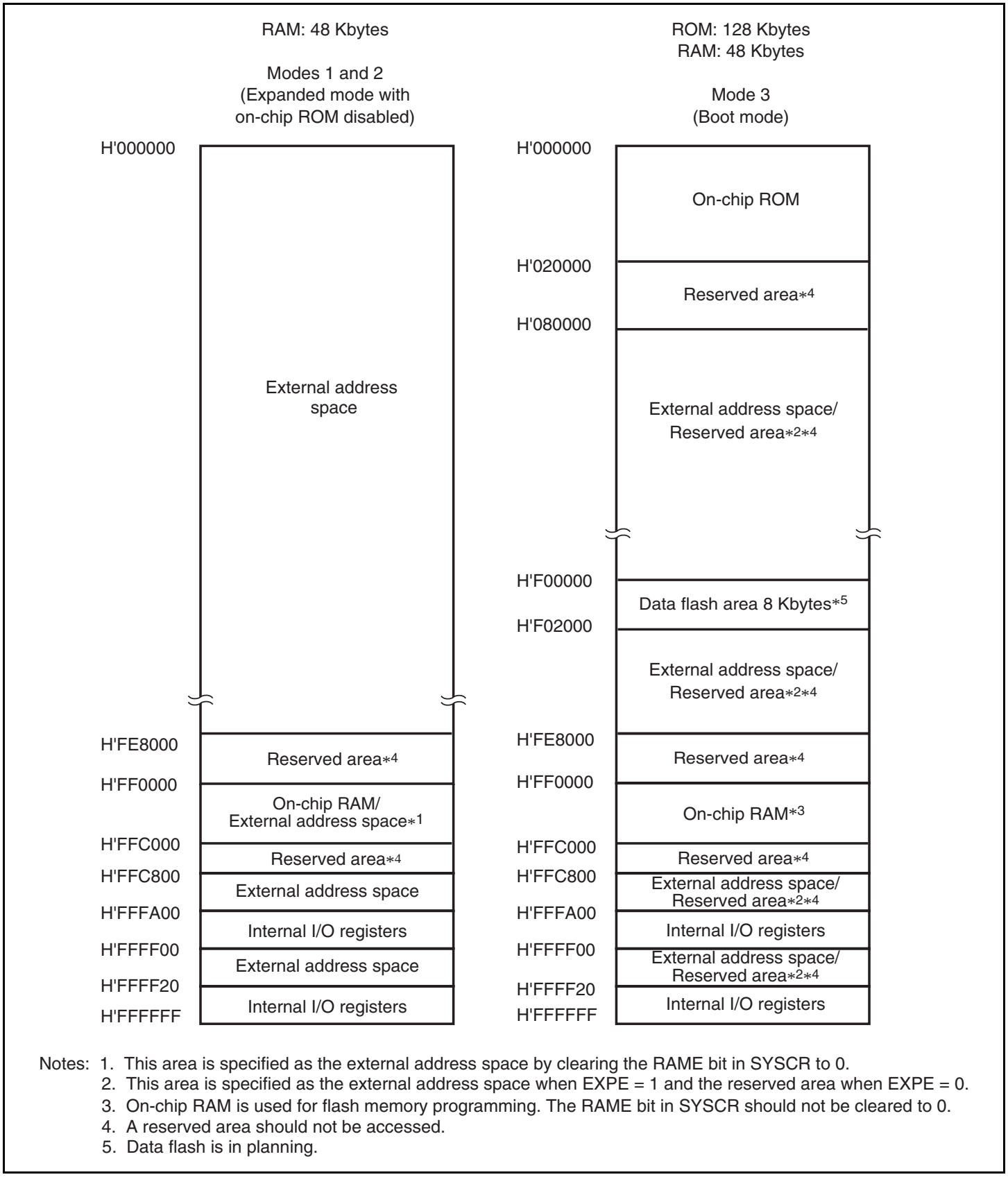
Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.
2. This area is specified as the external address space when EXPE = 1 and the reserved area when EXPE = 0.
3. On-chip RAM is used for flash memory programming. The RAME bit in SYSCR should not be cleared to 0.
4. A reserved area should not be accessed.
5. Area from H'FEC000 to H'FEFFFF in the H8S/24568, H8S/24568R, and H8S/24548 Groups is reserved and should not be accessed.
6. 64-Kbyte version (H8S/24569, H8S/24569R, and H8S/24549) is under development.
7. Data flash is in planning.

Figure 3.1 Memory Map in Each Operating Mode (ROM: 256-Kbyte Version): H8S/24569, H8S/24569R, H8S/24568, H8S/24568R, H8S/24549, and H8S/24548



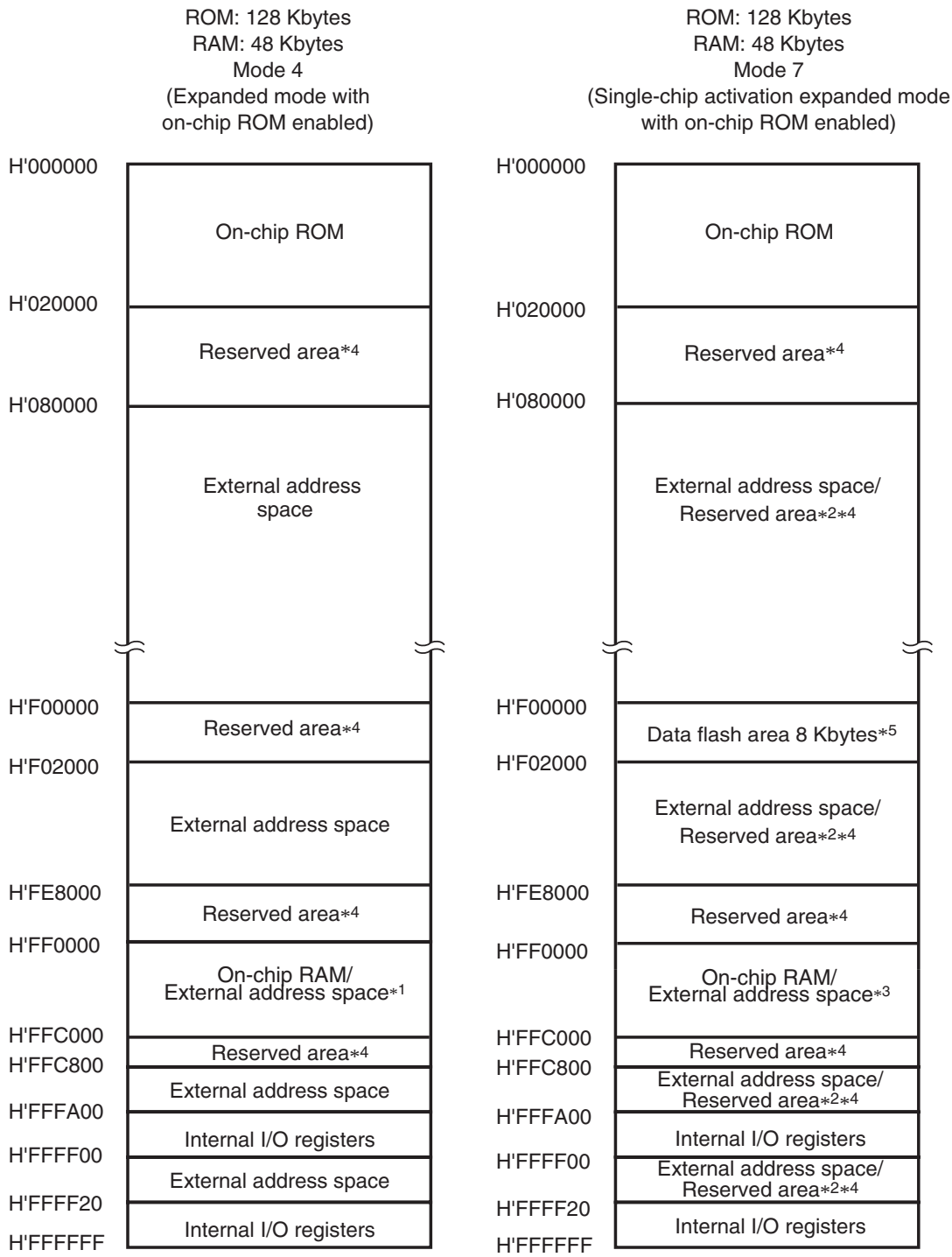
Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.
2. This area is specified as the external address space when EXPE = 1 and the reserved area when EXPE = 0.
3. While EXPE = 1, this area is specified as the external address space when RAME = 0 and the on-chip RAM area when RAME = 1. While EXPE = 0, this area is specified as the on-chip RAM area.
4. A reserved area should not be accessed.
5. Area from H'FEC000 to H'FEFFFF in the H8S/24568, H8S/24568R, and H8S/24548 Groups is reserved and should not be accessed.
6. 64-Kbyte version (H8S/24569, H8S/24569R, and H8S/24549) is under development.
7. Data flash is in planning.

Figure 3.2 Memory Map in Each Operating Mode (ROM: 256-Kbyte Version): H8S/24569, H8S/24569R, H8S/24549, and H8S/24548



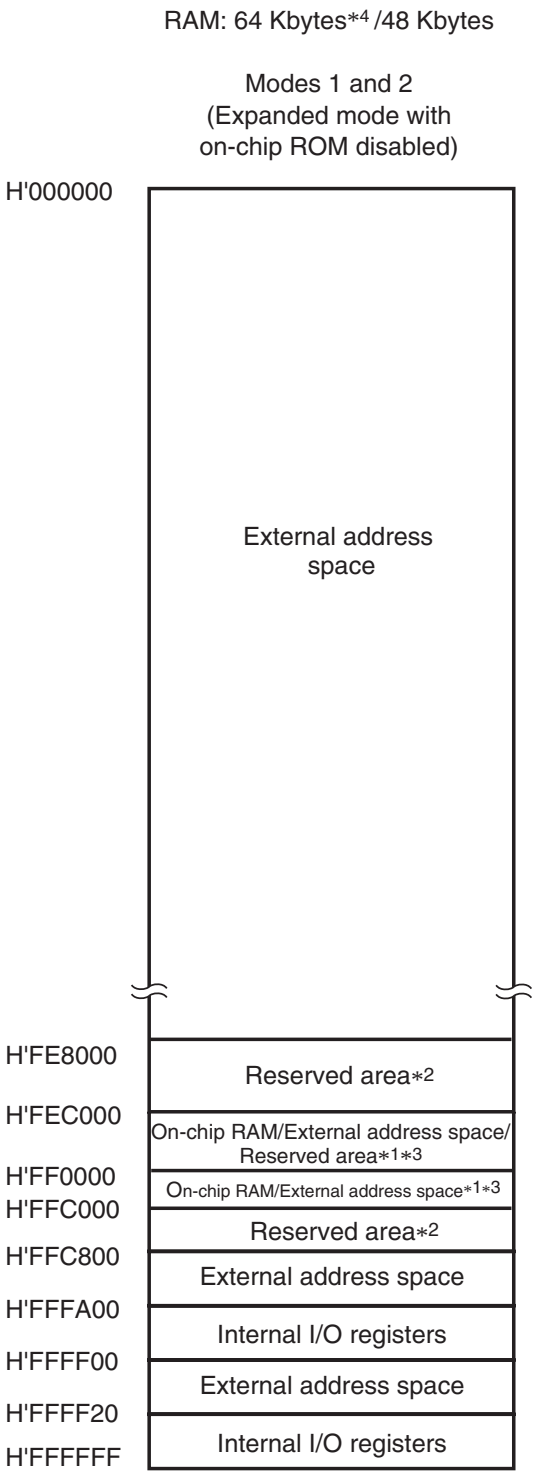
Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.
2. This area is specified as the external address space when EXPE = 1 and the reserved area when EXPE = 0.
3. On-chip RAM is used for flash memory programming. The RAME bit in SYSCR should not be cleared to 0.
4. A reserved area should not be accessed.
5. Data flash is in planning.

Figure 3.3 Memory Map in Each Operating Mode (ROM: 128-Kbyte Version): H8S/24565, H8S/24565R, and H8S/24545



Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.
2. This area is specified as the external address space when EXPE = 1 and the reserved area when EXPE = 0.
3. While EXPE = 1, this area is specified as the external address space when RAME = 0 and the on-chip RAM area when RAME = 1. While EXPE = 0, this area is specified as the on-chip RAM area.
4. A reserved area should not be accessed.
5. Data flash is in planning.

Figure 3.4 Memory Map in Each Operating Mode (ROM: 128-Kbyte Version): H8S/24565, H8S/24565R, and H8S/24545



Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.
2. A reserved area should not be accessed.
3. Area from H'FEC000 to H'FEFFFF in the H8S/24561, H8S/24561R, and H8S/24541 Groups is reserved and should not be accessed.
4. 64-Kbyte version (H8S/24562, H8S/24562R, and H8S/24542) is in planning.

Figure 3.5 Memory Map in Each Operating Mode (ROM-Less Version): H8S/24562, H8S/24562R, H8S/24561, H8S/24561R, H8S/24542, and H8S/24541

Section 4 Exception Handling

4.1 Exception Handling Types and Priority

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

Table 4.1 Exception Types and Priority

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

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

4.2 Exception Sources and Exception Vector Table

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

Table 4.2 Exception Handling Vector Table

			Vector Address*1	
Exception Source		**Vector Number**	**Normal Mode*2**	**Advanced Mode**
Power-on reset		0	H'0000 to H'0001	H'0000 to H'0003
Manual reset*3		1	H'0002 to H'0003	H'0004 to H'0007
Reserved for system use		2	H'0004 to H'0005	H'0008 to H'000B
		3	H'0006 to H'0007	H'000C to H'000F
Illegal instruction		4	H'0008 to H'0019	H'0010 to H'0013
Trace		5	H'000A to H'000B	H'0014 to H'0017
Interrupt (direct transition)*3		6	H'000C to H'000D	H'0018 to H'001B
Interrupt (NMI)		7	H'000E to H'000F	H'001C to H'001F
Trap instruction (#0)		8	H'0010 to H'0011	H'0020 to H'0023
(#1)		9	H'0012 to H'0013	H'0024 to H'0027
(#2)		10	H'0014 to H'0015	H'0028 to H'002B
(#3)		11	H'0016 to H'0017	H'002C to H'002F
Reserved for system use		12	H'0018 to H'0019	H'0030 to H'0033
		13	H'001A to H'001B	H'0034 to H'0037
		14	H'001C to H'001D	H'0038 to H'003B
		15	H'001E to H'001F	H'003C to H'003F
External interrupt	IRQ0	16	H'0020 to H'0021	H'0040 to H'0043
	IRQ1	17	H'0022 to H'0023	H'0044 to H'0047
	IRQ2	18	H'0024 to H'0025	H'0048 to H'004B
	IRQ3	19	H'0026 to H'0027	H'004C to H'004F
	IRQ4	20	H'0028 to H'0029	H'0050 to H'0053
	IRQ5	21	H'002A to H'002B	H'0054 to H'0057
	IRQ6	22	H'002C to H'002D	H'0058 to H'005B

Exception Source		Vector Number	Vector Address[*1] Normal Mode[*2]	Vector Address[*1] Advanced Mode
External interrupt	IRQ7	23	H'002E to H'002F	H'005C to H'005F
	IRQ8[*5]	24	H'0030 to H'0031	H'0060 to H'0063
	IRQ9[*5]	25	H'0032 to H'0033	H'0064 to H'0067
	IRQ10[*5]	26	H'0034 to H'0035	H'0068 to H'006B
	IRQ11[*5]	27	H'0036 to H'0037	H'006C to H'006F
	IRQ12[*5]	28	H'0038 to H'0039	H'0070 to H'0073
External interrupt	IRQ13[*5]	29	H'003A to H'003B	H'0074 to H'0077
	IRQ14[*5]	30	H'003C to H'003D	H'0078 to H'007B
	IRQ15[*5]	31	H'003E to H'003F	H'007C to H'007F
Internal interrupt[*4]		32 ⋮ 157	H'0040 to H'0041 ⋮ H'013A to H'013B	H'0080 to H'0083 ⋮ H'0274 to H'0277

Notes: 1. Lower 16 bits of the address.
2. Not available in this LSI.
3. Not available in this LSI. It is reserved for system use.
4. For details of internal interrupt vectors, see section 5.5, Interrupt Exception Handling Vector Table.
5. Reserved for system use in the H8S/2454 Group.

4.3 Reset

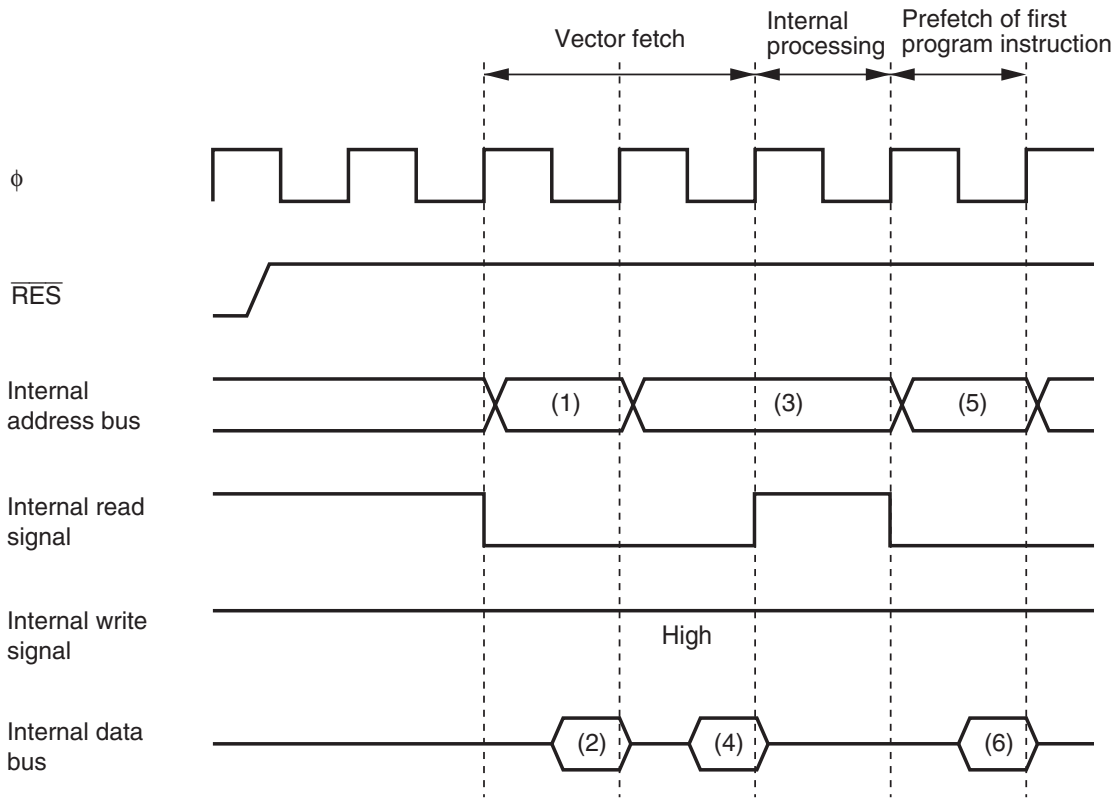
A reset has the highest exception priority. When the $\overline{\text{RES}}$ pin goes low, all processing halts and this LSI enters the reset. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-up. To reset this LSI during operation, hold the $\overline{\text{RES}}$ pin low for at least 20 states. A reset initializes the internal state of the CPU and the registers of on-chip peripheral modules. This LSI can also be reset by overflow of the watchdog timer. For details see section 14, Watchdog Timer (WDT). The interrupt control mode is 0 immediately after reset.

4.3.1 Reset Exception Handling

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

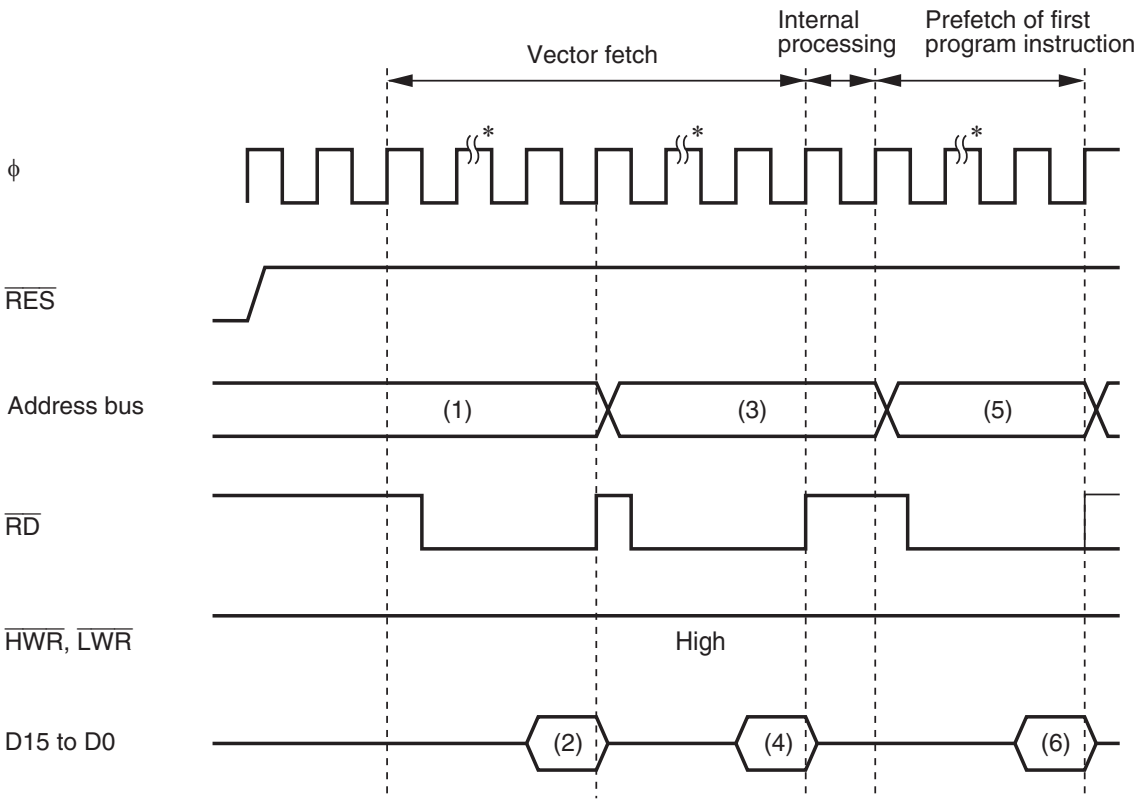
1. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized, the T bit is cleared to 0 in EXR, and the I bit is set to 1 in EXR and CCR.
2. The reset exception handling vector address is read and transferred to the PC, and program execution starts from the address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

Figures 4.1 and 4.2 show examples of the reset sequence.



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

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



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

Note: * Seven program wait states are inserted.

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

4.3.2 Interrupts after Reset

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

4.3.3 On-Chip Peripheral Functions after Reset Release

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

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

4.4 Trace Exception Handling

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

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

Interrupts are accepted even within the trace exception handling routine.

Table 4.3 Status of CCR and EXR after Trace Exception Handling

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

Legend:
1: Set to 1
0: Cleared to 0
—: Retains value prior to execution

4.5 Interrupt Exception Handling

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

The interrupt exception handling is as follows:

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

4.6 Trap Instruction Exception Handling

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

The trap instruction exception handling is as follows:

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

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

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

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

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

Legend:
1: Set to 1
0: Cleared to 0
—: Retains value prior to execution

4.7 Illegal Instruction Exception Handling

Illegal instruction exception handling starts when the CPU executing an illegal instruction code is detected. Illegal instruction exception handling can be executed at all times in the program execution state.

The illegal instruction exception handling is as follows:

1. The values in the 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 exception is generated, the start address of the exception service routine is loaded from the vector table to the PC, and program execution starts from that address.

Table 4.5 shows the status of CCR and EXR after execution of illegal instruction exception handling.

Table 4.5 Status of CCR and EXR after Illegal Instruction Exception Handling

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

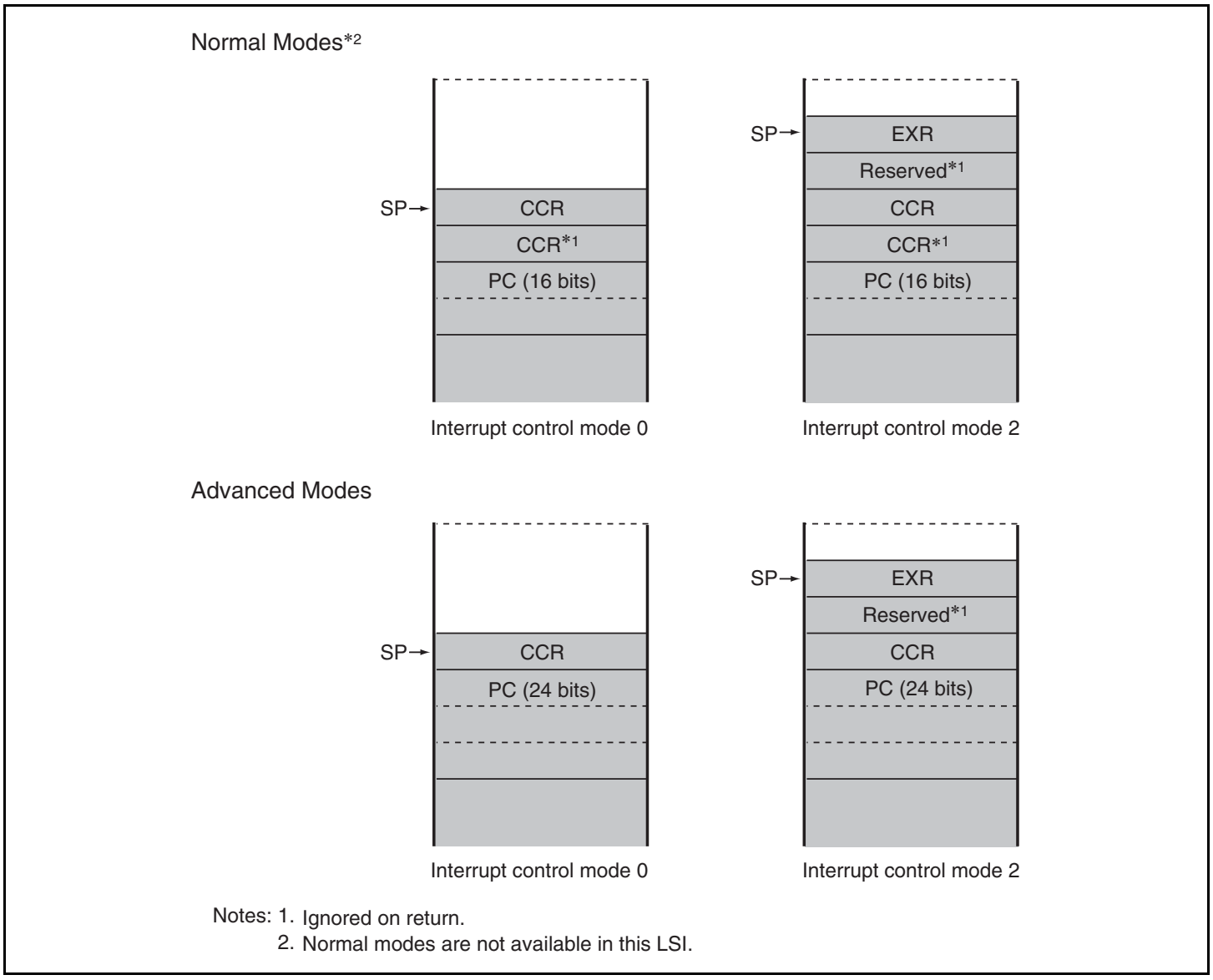
Legend:
1: Set to 1
0: Cleared to 0
—: Retains value prior to execution

Illegal instruction codes will not be searched for in the fields that do not affect instruction definitions, such as the EA extension or register fields. Instruction codes for an instruction formed with several words are detected independently, and combined instruction codes are not detected.

Undefined instruction codes must not be executed. The general register contents after execution of an undefined instruction code or illegal instruction exception handling cannot be guaranteed. The stack pointer during illegal instruction exception handling and the PC value that will be saved are also not guaranteed.

4.8 Stack Status after Exception Handling

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



Notes: 1. Ignored on return.
2. Normal modes are not available in this LSI.

Figure 4.3 Stack Status after Exception Handling

4.9 Usage Note

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

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

Use the following instructions to restore registers:

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

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

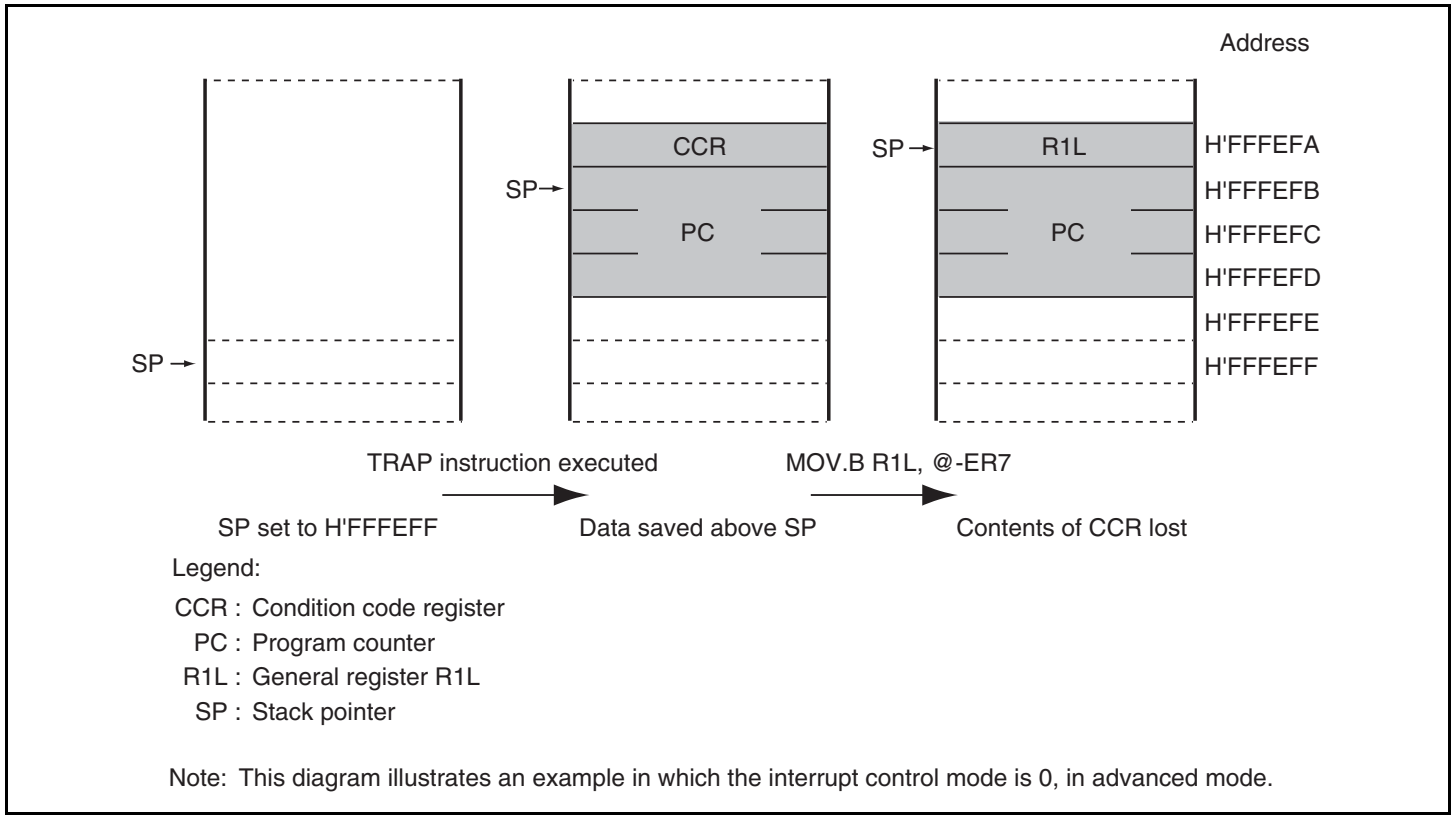


Figure 4.4 Operation when SP Value Is Odd

Section 5 Interrupt Controller

5.1 Features

- Two interrupt control modes

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

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

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

  NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge can be selected for NMI. Falling edge, rising edge, or both edge detection, or level sensing, can be selected for $\overline{\text{IRQn-A}}$ and $\overline{\text{IRQn-B}}$.

  Note: n = 15 to 0 for H8S/2456 Group and H8S/2456R Group, n = 7 to 0 for H8S/2454 Group
- DTC and DMAC control

  DTC and DMAC activations are performed by means of interrupts.

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

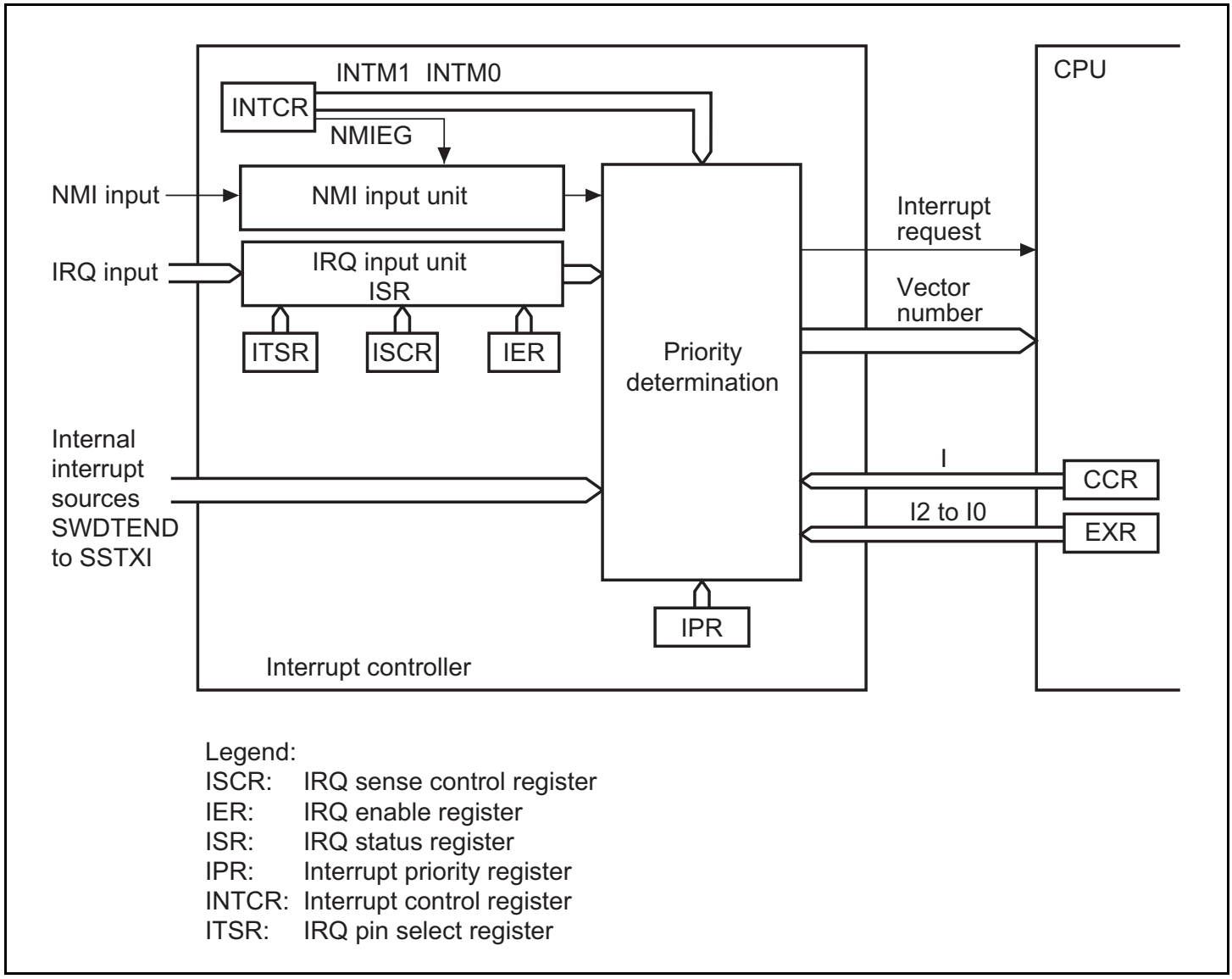


Figure 5.1 Block Diagram of Interrupt Controller

5.2 Input/Output Pins

Table 5.1 shows the pin configuration of the interrupt controller.

Table 5.1 Pin Configuration

Name	I/O	Function
NMI	Input	Nonmaskable external interrupt Rising or falling edge can be selected.
$\overline{\text{IRQ15-A}}$ to $\overline{\text{IRQ0-A}}$* $\overline{\text{IRQ15-B}}$ to $\overline{\text{IRQ0-B}}$*	Input	Maskable external interrupts Rising, falling, or both edges, or level sensing, can be selected.

Note: * $\overline{\text{IRQ7-A}}$ to $\overline{\text{IRQ0-A}}$ and $\overline{\text{IRQ7-B}}$ to $\overline{\text{IRQ0-B}}$ in the H8S/2454 Group.
$\overline{\text{IRQ12-B}}$ to $\overline{\text{IRQ9-B}}$ are not supported in the H8S/2456 Group.

5.3 Register Descriptions

The interrupt controller has the following registers.

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

5.3.1 Interrupt Control Register (INTCR)

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

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0 and the initial value should not be changed.
5	INTM1	0	R/W	Interrupt Control Select Mode 1 and 0
4	INTM0	0	R/W	These bits select either of two interrupt control modes for the interrupt controller. 00: Interrupt control mode 0 Interrupts are controlled by I bit. 01: Setting prohibited. 10: Interrupt control mode 2 Interrupts are controlled by bits I2 to I0, and IPR. 11: Setting prohibited.
3	NMIEG	0	R/W	NMI Edge Select Selects the input edge for the NMI pin. 0: Interrupt request generated at falling edge of NMI input 1: Interrupt request generated at rising edge of NMI input
2 to 0	—	All 0	—	Reserved These bits are always read as 0 and the initial value should not be changed.

5.3.2 Interrupt Priority Registers A to N (IPRA to IPRN)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0 and the initial value should not be changed.
6	IPR6	1	R/W	Sets the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)
5	IPR5	1	R/W	
4	IPR4	1	R/W	
3	—	0	—	Reserved This bit is always read as 0 and the initial value should not be changed.
2	IPR2	1	R/W	Sets the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)
1	IPR1	1	R/W	
0	IPR0	1	R/W	

5.3.3 IRQ Enable Register (IER)

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

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ15E	0	R/W	IRQ15 Enable* The IRQ15 interrupt request is enabled when this bit is 1.
14	IRQ14E	0	R/W	IRQ14 Enable* The IRQ14 interrupt request is enabled when this bit is 1.
13	IRQ13E	0	R/W	IRQ13 Enable* The IRQ13 interrupt request is enabled when this bit is 1.
12	IRQ12E	0	R/W	IRQ12 Enable* The IRQ12 interrupt request is enabled when this bit is 1.
11	IRQ11E	0	R/W	IRQ11 Enable* The IRQ11 interrupt request is enabled when this bit is 1.
10	IRQ10E	0	R/W	IRQ10 Enable* The IRQ10 interrupt request is enabled when this bit is 1.
9	IRQ9E	0	R/W	IRQ9 Enable* The IRQ9 interrupt request is enabled when this bit is 1.
8	IRQ8E	0	R/W	IRQ8 Enable* The IRQ8 interrupt request is enabled when this bit is 1.
7	IRQ7E	0	R/W	IRQ7 Enable The IRQ7 interrupt request is enabled when this bit is 1.
6	IRQ6E	0	R/W	IRQ6 Enable The IRQ6 interrupt request is enabled when this bit is 1.

Bit	Bit Name	Initial Value	R/W	Description
5	IRQ5E	0	R/W	IRQ5 Enable The IRQ5 interrupt request is enabled when this bit is 1.
4	IRQ4E	0	R/W	IRQ4 Enable The IRQ4 interrupt request is enabled when this bit is 1.
3	IRQ3E	0	R/W	IRQ3 Enable The IRQ3 interrupt request is enabled when this bit is 1.
2	IRQ2E	0	R/W	IRQ2 Enable The IRQ2 interrupt request is enabled when this bit is 1.
1	IRQ1E	0	R/W	IRQ1 Enable The IRQ1 interrupt request is enabled when this bit is 1.
0	IRQ0E	0	R/W	IRQ0 Enable The IRQ0 interrupt request is enabled when this bit is 1.

Note: * These bits are reserved in the H8S/2454 Group.

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

ISCR select the source that generates an interrupt request at pins $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$.

- ISCRH (H8S/2456 Group only)

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

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

Bit	Bit Name	Initial Value	R/W	Description
3	IRQ9SCB	0	R/W	IRQ9 Sense Control B
2	IRQ9SCA	0	R/W	IRQ9 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ9}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ9}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ9}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ9}}$ input
1	IRQ8SCB	0	R/W	IRQ8 Sense Control B
0	IRQ8SCA	0	R/W	IRQ8 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ8}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ8}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ8}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ8}}$ input

- ISCRL

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ7SCB	0	R/W	IRQ7 Sense Control B
14	IRQ7SCA	0	R/W	IRQ7 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ7}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ7}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ7}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ7}}$ input
13	IRQ6SCB	0	R/W	IRQ6 Sense Control B
12	IRQ6SCA	0	R/W	IRQ6 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ6}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ6}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ6}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ6}}$ input
11	IRQ5SCB	0	R/W	IRQ5 Sense Control B
10	IRQ5SCA	0	R/W	IRQ5 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ5}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ5}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ5}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ5}}$ input

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

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

5.3.5 IRQ Status Register (ISR)

ISR is an IRQ15 to IRQ0 interrupt request flag register.

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

Notes: 1. Only 0 can be written, to clear the flag.
2. These bits are reserved in the H8S/2454 Group.

5.3.6 IRQ Pin Select Register (ITSR)

ITSR selects input pins $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$.

- H8S/2456 Group

Bit	Bit Name	Initial Value	R/W	Description
15	ITS15	0	R/W	Selects the $\overline{\text{IRQ15}}$ input pin. 0: PF2/$\overline{\text{IRQ15-A}}$ selected 1: P27/$\overline{\text{IRQ15-B}}$ selected
14	ITS14	0	R/W	Selects the $\overline{\text{IRQ14}}$ input pin. 0: PF1/$\overline{\text{IRQ14-A}}$ selected 1: P26/$\overline{\text{IRQ14-B}}$ selected
13	ITS13	0	R/W	Selects the $\overline{\text{IRQ13}}$ input pin. 0: P65/$\overline{\text{IRQ13-A}}$ selected 1: P25/$\overline{\text{IRQ13-B}}$ selected
12	—	0	R/W	Reserved The initial value should not be changed.
11	—	0	R/W	Reserved The initial value should not be changed.
10	—	0	R/W	Reserved The initial value should not be changed.
9	—	0	R/W	Reserved The initial value should not be changed.
8	ITS8	0	R/W	Selects the $\overline{\text{IRQ8}}$ input pin. 0: P60/$\overline{\text{IRQ8-A}}$ selected 1: P20/$\overline{\text{IRQ8-B}}$ selected
7	ITS7	0	R/W	Selects the $\overline{\text{IRQ7}}$ input pin. 0: PA7/$\overline{\text{IRQ7-A}}$ selected 1: PH3/$\overline{\text{IRQ7-B}}$ selected
6	ITS6	0	R/W	Selects the $\overline{\text{IRQ6}}$ input pin. 0: PA6/$\overline{\text{IRQ6-A}}$ selected 1: PH2/$\overline{\text{IRQ6-B}}$ selected

Bit	Bit Name	Initial Value	R/W	Description
5	ITS5	0	R/W	Selects the $\overline{\text{IRQ5}}$ input pin. 0: PA5/$\overline{\text{IRQ5-A}}$ selected 1: P85/$\overline{\text{IRQ5-B}}$ selected
4	ITS4	0	R/W	Selects the $\overline{\text{IRQ4}}$ input pin. 0: PA4/$\overline{\text{IRQ4-A}}$ selected 1: P84/$\overline{\text{IRQ4-B}}$ selected
3	ITS3	0	R/W	Selects the $\overline{\text{IRQ3}}$ input pin. 0: P53/$\overline{\text{IRQ3-A}}$ selected 1: P83/$\overline{\text{IRQ3-B}}$ selected
2	ITS2	0	R/W	Selects the $\overline{\text{IRQ2}}$ input pin. 0: P52/$\overline{\text{IRQ2-A}}$ selected 1: P82/$\overline{\text{IRQ2-B}}$ selected
1	ITS1	0	R/W	Selects the $\overline{\text{IRQ1}}$ input pin. 0: P51/$\overline{\text{IRQ1-A}}$ selected 1: P81/$\overline{\text{IRQ1-B}}$ selected
0	ITS0	0	R/W	Selects the $\overline{\text{IRQ0}}$ input pin. 0: P50/$\overline{\text{IRQ0-A}}$ selected 1: P80/$\overline{\text{IRQ0-B}}$ selected

- H8S/2454 Group

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R/W	Reserved The initial value should not be changed.
7	ITS7	0	R/W	Selects the $\overline{\text{IRQ7}}$ input pin. 0: PA7/$\overline{\text{IRQ7-A}}$ selected 1: P47/$\overline{\text{IRQ7-B}}$ selected
6	ITS6	0	R/W	Selects the $\overline{\text{IRQ6}}$ input pin. 0: PA6/$\overline{\text{IRQ6-A}}$ selected 1: P46/$\overline{\text{IRQ6-B}}$ selected
5	ITS5	0	R/W	Selects the $\overline{\text{IRQ5}}$ input pin. 0: PA5/$\overline{\text{IRQ5-A}}$ selected 1: P45/$\overline{\text{IRQ5-B}}$ selected
4	ITS4	0	R/W	Selects the $\overline{\text{IRQ4}}$ input pin. 0: PA4/$\overline{\text{IRQ4-A}}$ selected 1: P44/$\overline{\text{IRQ4-B}}$ selected
3	ITS3	0	R/W	Selects the $\overline{\text{IRQ3}}$ input pin. 0: P53/$\overline{\text{IRQ3-A}}$ selected 1: P43/$\overline{\text{IRQ3-B}}$ selected
2	ITS2	0	R/W	Selects the $\overline{\text{IRQ2}}$ input pin. 0: P52/$\overline{\text{IRQ2-A}}$ selected 1: P42/$\overline{\text{IRQ2-B}}$ selected
1	ITS1	0	R/W	Selects the $\overline{\text{IRQ1}}$ input pin. 0: P51/$\overline{\text{IRQ1-A}}$ selected 1: P41/$\overline{\text{IRQ1-B}}$ selected
0	ITS0	0	R/W	Selects the $\overline{\text{IRQ0}}$ input pin. 0: P50/$\overline{\text{IRQ0-A}}$ selected 1: P40/$\overline{\text{IRQ0-B}}$ selected

5.3.7 Software Standby Release IRQ Enable Register (SSIER)

SSIER selects the $\overline{\text{IRQ}}$ pins used to recover from the software standby state.

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

Note: * These bits are reserved in the H8S/2454 Group.

5.4 Interrupt Sources

5.4.1 External Interrupts

The H8S/2456 Group and H8S/2456R Group each have seventeen external interrupts: NMI and IRQ15 to IRQ0. The H8S/2454 Group has nine external interrupts: NMI and IRQ7 to IRQ0. These interrupts can be used to restore the chip from software standby mode.

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

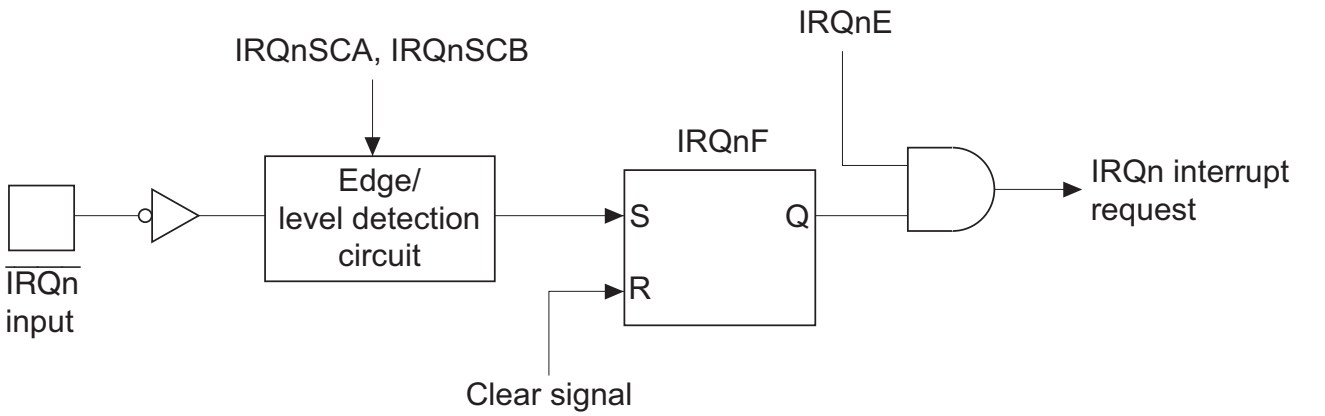
IRQn Interrupts (n = 0 to 15 for H8S/2456 Group and H8S/2456R Group, n = 0 to 7 for H8S/2454 Group): An IRQn interrupt is requested by an input signal at the $\overline{\text{IRQn}}$ pin. The IRQn interrupts have the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at the $\overline{\text{IRQn}}$ pin.
- Enabling or disabling of IRQn interrupt requests can be selected with IER.
- The interrupt priority level can be set with IPR.
- The status of IRQn interrupt requests is indicated in ISR. ISR flags can be cleared to 0 by software.

When IRQn interrupt requests occur at low level of the $\overline{\text{IRQn}}$ pin, the corresponding $\overline{\text{IRQ}}$ pin should be held low until an interrupt handling starts. Then the corresponding $\overline{\text{IRQ}}$ pin should be set to high in the interrupt handling routine and clear the IRQnF bit in ISR to 0. Interrupts may not be executed when the corresponding $\overline{\text{IRQ}}$ pin is set to high before the interrupt handling starts.

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

A block diagram of IRQn interrupts is shown in figure 5.2.



Note: n = 0 to 15 for H8S/2456 Group and H8S/2456R Group, n = 0 to 7 for H8S/2454 Group

Figure 5.2 Block Diagram of IRQ Interrupts

5.4.2 Internal Interrupts

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

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

5.5 Interrupt Exception Handling Vector Table

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

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

Table 5.2 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*1 Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
External pin	NMI	7	H'001C	—	High	—	—
	IRQ0	16	H'0040	IPRA14 to IPRA12		○	—
	IRQ1	17	H'0044	IPRA10 to IPRA8		○	—
	IRQ2	18	H'0048	IPRA6 to IPRA4		○	—
	IRQ3	19	H'004C	IPRA2 to IPRA0		○	—
	IRQ4	20	H'0050	IPRB14 to IPRB12		○	—
	IRQ5	21	H'0054	IPRB10 to IPRB8		○	—
	IRQ6	22	H'0058	IPRB6 to IPRB4		○	—
	IRQ7	23	H'005C	IPRB2 to IPRB0		○	—
	IRQ8*2	24	H'0060	IPRC14 to IPRC12		○	—
	IRQ9*2	25	H'0064	IPRC10 to IPRC8		○	—
	IRQ10*2	26	H'0068	IPRC6 to IPRC4		○	—
	IRQ11*2	27	H'006C	IPRC2 to IPRC0		○	—
	IRQ12*2	28	H'0070	IPRD14 to IPRD12		○	—
	IRQ13*2	29	H'0074	IPRD10 to IPRD8		○	—
	IRQ14*2	30	H'0078	IPRD6 to IPRD4		○	—
	IRQ15*2	31	H'007C	IPRD2 to IPRD0		○	—
DTC	SWDTEND	32	H'0080	IPRE14 to IPRE12		○	—
WDT	WOVI0	33	H'0084	IPRE10 to IPRE8		—	—
—	Reserved for system use	34	H'0088	IPRE6 to IPRE4	Low	—	—

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*1 Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
Refresh controller	CMI	35	H'008C	IPRE2 to IPRE0	High	—	—
—	Reserved for system use	36	H'0090	IPRF14 to IPRF12		—	—
		37	H'0094			—	—
A/D_0	ADI0	38	H'0098	IPRF10 to IPRF8		○	○
	Reserved for system use	39	H'009C			—	—
TPU_0	TGI0A	40	H'00A0	IPRF6 to IPRF4		○	○
	TGI0B	41	H'00A4			○	—
	TGI0C	42	H'00A8			○	—
	TGI0D	43	H'00AC	IPRF6 to IPRF4		○	—
	TCI0V	44	H'00B0			—	—
	Reserved for system use	45	H'00B4			—	—
		46	H'00B8			—	—
		47	H'00BC			—	—
TPU_1	TGI1A	48	H'00C0	IPRF2 to IPRF0		○	○
	TGI1B	49	H'00C4			○	—
	TCI1V	50	H'00C8			—	—
	TCI1U	51	H'00CC			—	—
TPU_2	TGI2A	52	H'00D0	IPRG14 to IPRG12		○	○
	TGI2B	53	H'00D4			○	—
	TCI2V	54	H'00D8			—	—
	TCI2U	55	H'00DC			—	—
TPU_3	TGI3A	56	H'00E0	IPRG10 to IPRG8		○	○
	TGI3B	57	H'00E4			○	—
	TGI3C	58	H'00E8			○	—
	TGI3D	59	H'00EC			○	—
	TCI3V	60	H'00F0		Low	—	—

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*1 Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
TPU_3	Reserved for system use	61	H'00F4	IPRG10 to IPRG8	High	—	—
		62	H'00F8			—	—
		63	H'00FC			—	—
TPU_4	TGI4A	64	H'0100	IPRG6 to IPRG4		○	○
	TGI4B	65	H'0104			○	—
	TCI4V	66	H'0108			—	—
	TCI4U	67	H'010C			—	—
TPU_5	TGI5A	68	H'0110	IPRG2 to IPRG0		○	○
	TGI5B	69	H'0114			○	—
	TCI5V	70	H'0118			—	—
	TCI5U	71	H'011C			—	—
TMR_0	CMIA0	72	H'0120	IPRH14 to IPRH12		○	—
	CMIB0	73	H'0124			○	—
	OVI0	74	H'0128			—	—
	Reserved for system use	75	H'012C	IPRH14 to IPRH12		—	—
TMR_1	CMIA1	76	H'0130	IPRH10 to IPRH8		○	—
	CMIB1	77	H'0134			○	—
	OVI1	78	H'0138			—	—
	Reserved for system use	79	H'013C			—	—
DMAC	DMTEND0A	80	H'0140	IPRH6 to IPRH4		○	—
	DMTEND0B	81	H'0144			○	—
	DMTEND1A	82	H'0148			○	—
	DMTEND1B	83	H'014C		Low	○	—

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*1 Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
EXDMAC*2	Reserved for system use	84	H'0150	IPRH0 to IPRH0	High	—	—
		85	H'0154	IPRI14 to IPRI12		—	—
	EXDMTEND2	86	H'0158	IPRI10 to IPRI8		—	—
	EXDMTEND3	87	H'015C	IPRI6 to IPRI4		—	—
SCI_0	ERI0	88	H'0160	IPRI2 to IPRI0		—	—
	RXI0	89	H'0164			○	○
	TXI0	90	H'0168			○	○
	TEI0	91	H'016C			—	—
SCI_1	ERI1	92	H'0170	IPRJ14 to IPRJ12		—	—
	RXI1	93	H'0174			○	○
	TXI1	94	H'0178			○	○
	TEI1	95	H'017C			—	—
SCI_2	ERI2	96	H'0180	IPRJ10 to IPRJ8		—	—
	RXI2	97	H'0184			○	—
	TXI2	98	H'0188			○	—
	TEI2	99	H'018C			—	—
SCI_3	ERI3	100	H'0190	IPRJ6 to IPRJ4		—	—
	RXI3	101	H'0194			○	—
	TXI3	102	H'0198			○	—
	TEI3	103	H'019C			—	—
SCI_4	ERI4	104	H'01A0	IPRJ2 to IPRJ0		—	—
	RXI4	105	H'01A4			○	—
	TXI4	106	H'01A8			○	—
	TEI4	107	H'01AC			—	—
	Reserved for system use	108	H'01B0	IPRK14 to IPRK12		—	—
		109	H'01B4			—	—
		110	H'01B8			—	—
		111	H'01BC		Low	—	—

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*1 Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
A/D_1	ADI1	112	H'01C0	IPRK10 to IPRK8	High	○	—
	Reserved for system use	113	H'01C4			—	—
		114	H'01C8			—	—
		115	H'01CC			—	—
IIC2_0	IICI0	116	H'01D0	IPRK6 to IPRK4		—	—
	Reserved for system use	117	H'01D4			—	—
IIC2_1	IICI1	118	H'01D8			—	—
	Reserved for system use	119	H'01DC			—	—
TPU_6	TGI6A	120	H'01E0	IPRK2 to IPRK0		○	—
	TGI6B	121	H'01E4			○	—
	TGI6C	122	H'01E8			○	—
	TGI6D	123	H'01EC			○	—
	TCI6V	124	H'01F0			—	—
TPU_7	TGI7A	125	H'01F4	IPRL14 to IPRL12		○	—
	TGI7B	126	H'01F8			○	—
	TCI7V	127	H'01FC			—	—
	TCI7U	128	H'0200			—	—
TPU_8	TGI8A	129	H'0204	IPRL10 to IPRL8		○	—
	TGI8B	130	H'0208			○	—
	TCI8V	131	H'020C			—	—
	TCI8U	132	H'0210			—	—
TPU_9	TGI9A	133	H'0214	IPRL6 to IPRL4		○	—
	TGI9B	134	H'0218			○	—
	TGI9C	135	H'021C			○	—
	TGI9D	136	H'0220			○	—
	TCI9V	137	H'0224		Low	—	—

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*1 Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
TPU_10	TGI10A	138	H'0228	IPRL2 to IPRL0	High	○	—
	TGI10B	139	H'022C			○	—
	TCI10V	140	H'0230			—	—
	TCI10U	141	H'0234			—	—
TPU_11	TGI11A	142	H'0238	IPRM14 to IPRM12		○	—
	TGI11B	143	H'023C			○	—
	TCI11V	144	H'0240			—	—
	TCI11U	145	H'0244			—	—
USB	USBINTN0	146	H'0248	IPRM10 to IPRM8		—	—
	USBINTN1	147	H'024C			—	—
	USBINTN2	148	H'0250			—	—
	USBINTN3	149	H'0254			—	—
	USBINTN0	150	H'0258	IPRM6 to IPRM4		—	—
	Reserved for system use	151	H'025C			—	—
	RESUME	152	H'0260			—	—
IIC2_2	IICI2	153	H'0264	IPRM2 to IPRM0		—	—
IIC2_3	IICI3	154	H'0268			—	—
SSU	SSERI	155	H'026C	IPRN14 to IPRN12		—	—
	SSRXI	156	H'0270			—	—
	SSTXI	157	H'0274		Low	—	—

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*1 Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
—	Reserved for system use	158	H'0278	IPRN10 to IPRN8	High	—	—
		159	H'027C			—	—
		160	H'0280			—	—
		161	H'0284			—	—
		162	H'0288	IPRN6 to IPRN4		—	—
		163	H'028C			—	—
		164	H'0290			—	—
		165	H'0294			—	—
		166	H'0298	IPRN2 to IPRN0		—	—
		167	H'029C			—	—
		168	H'02A0			—	—
		169	H'02A4			—	—
	Reserved for system use	170	H'02A8	—		—	—
		|	|			|	|
		255	H'03FC		Low	—	—

Notes: 1. Lower 16 bits of the start address.
2. Not supported in the H8S/2454 Group.

5.6 Interrupt Control Modes and Interrupt Operation

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

Table 5.3 Interrupt Control Modes

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

5.6.1 Interrupt Control Mode 0

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

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

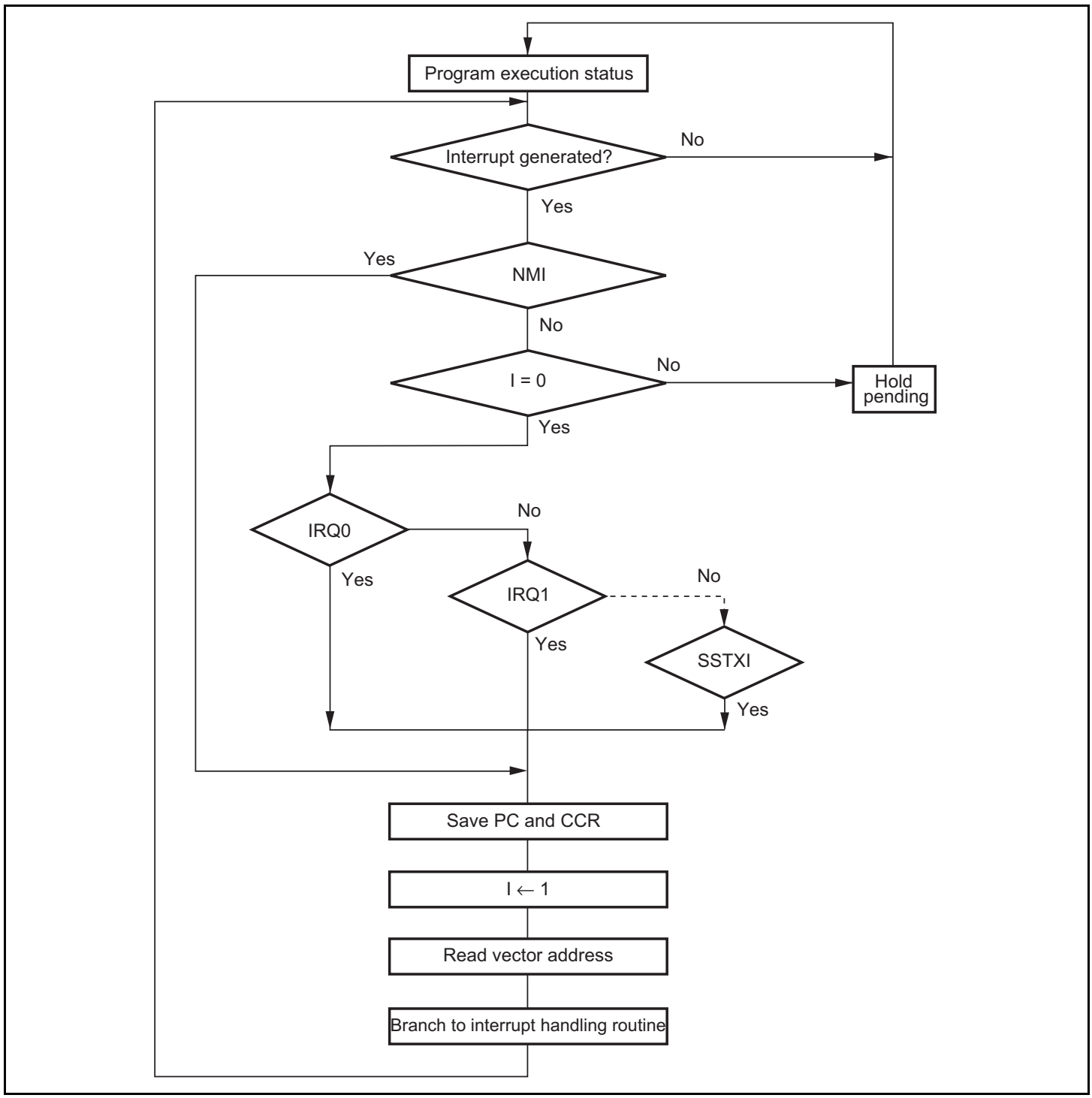


Figure 5.3 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

5.6.2 Interrupt Control Mode 2

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

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

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

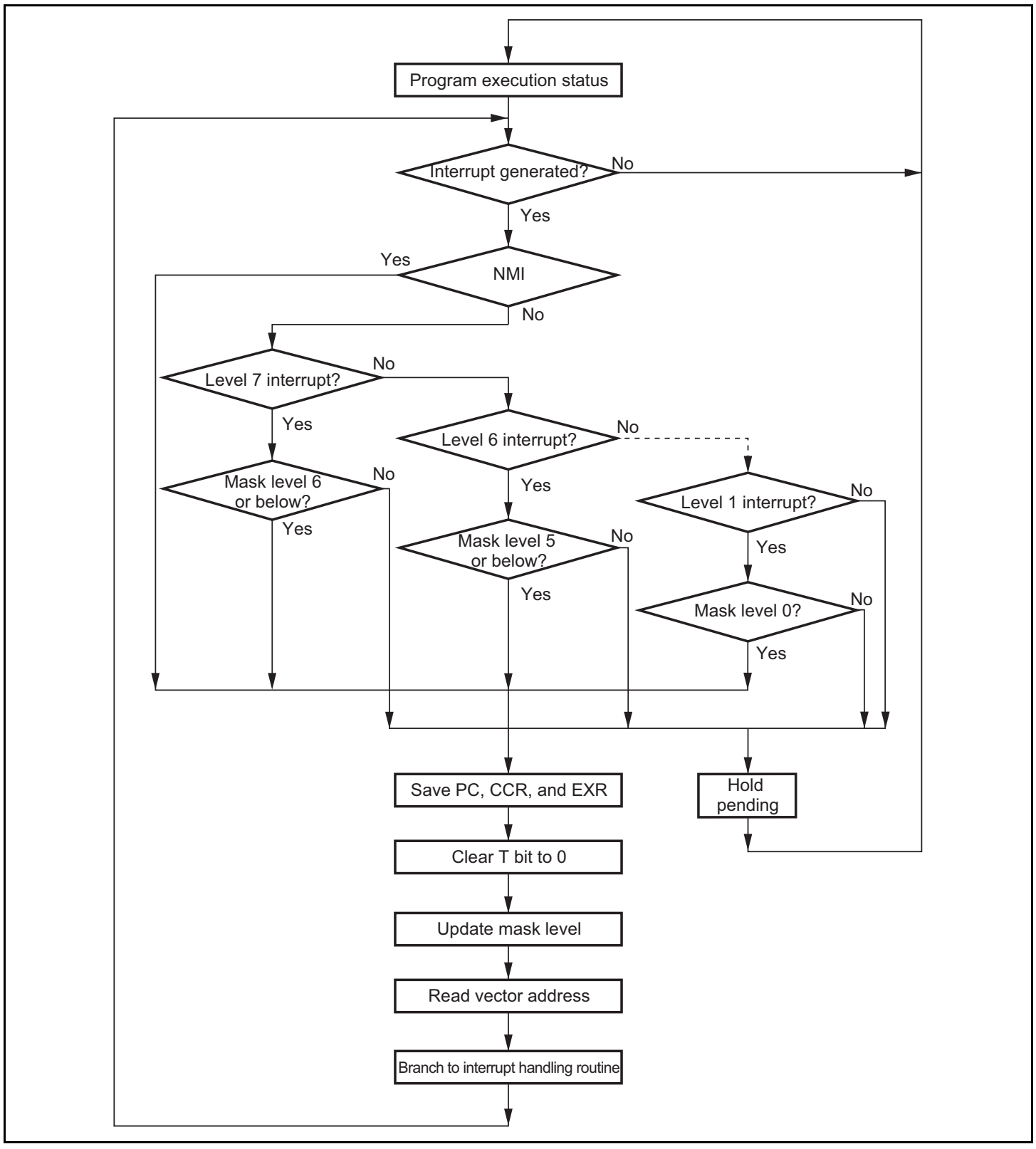


Figure 5.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

5.6.3 Interrupt Exception Handling Sequence

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

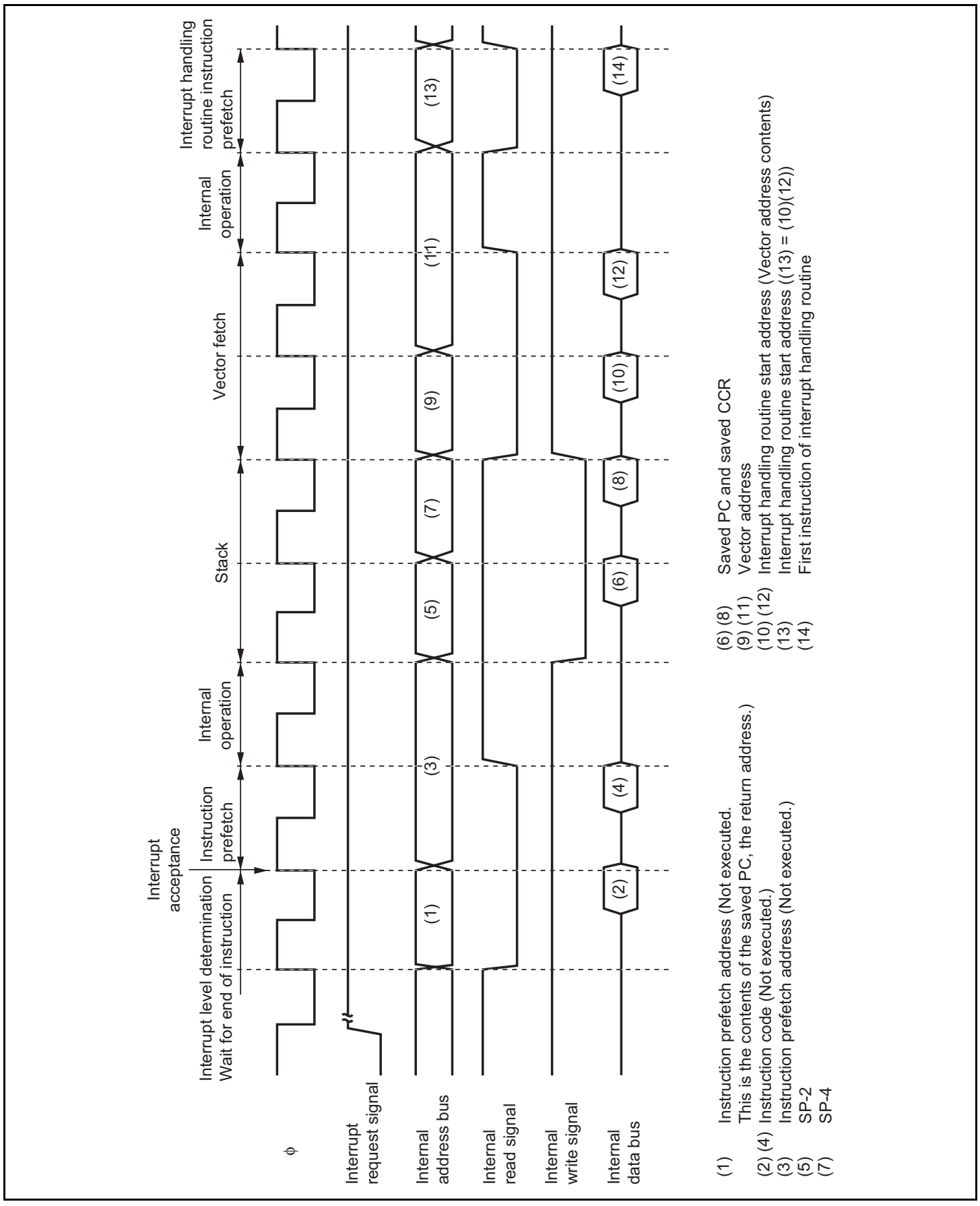


Figure 5.5 Interrupt Exception Handling

5.6.4 Interrupt Response Times

Table 5.4 shows interrupt response times - the interval between generation of an interrupt request and execution of the first instruction in the interrupt handling routine. The execution status symbols used in table 5.4 are explained in table 5.5. This LSI is capable of fast word transfer to on-chip memory, and have the program area in on-chip ROM and the stack area in on-chip RAM, enabling high-speed processing.

Table 5.4 Interrupt Response Times

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

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

Table 5.5 Number of States in Interrupt Handling Routine Execution Statuses

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

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

5.6.5 DTC and DMAC Activation by Interrupt

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

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

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

5.7 Usage Notes

5.7.1 Conflict between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to mask interrupts, the masking becomes effective after execution of the instruction. When an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request of higher priority than that interrupt, interrupt exception handling will be executed for the higher-priority interrupt, and the lower-priority interrupt will be ignored. The same also applies when an interrupt source flag is cleared to 0. Figure 5.6 shows an example in which the TCIEV bit in the TPU's TIER_0 register is cleared to 0. The above conflict will not occur if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.

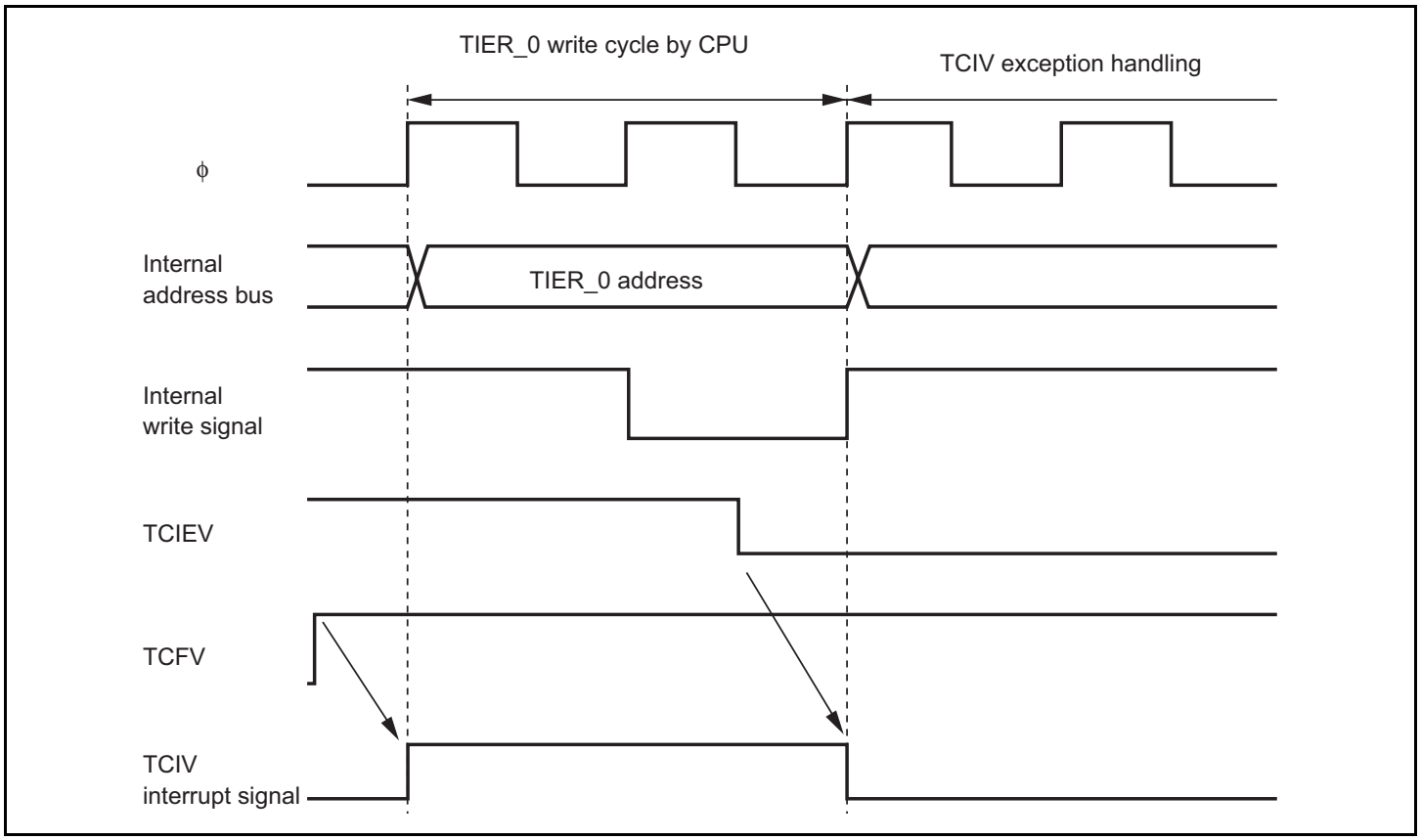


Figure 5.6 Conflict between Interrupt Generation and Disabling

5.7.2 Instructions that Disable Interrupts

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

5.7.3 Times when Interrupts are Disabled

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

5.7.4 Interrupts during Execution of EEPMOV Instruction

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

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

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

```
L1:   EEPMOV.W
      MOV.W     R4,R4
      BNEL1
```

5.7.5 Change of IRQ Pin Select Register (ITSR) Setting

When the ITSR setting is changed, an edge occurs internally and the IRQnF bit (n = 0 to 15 for H8S/2456 Group, n = 0 to 7 for H8S/2454 Group) of ISR may be set to 1 at the unintended timing if the selected pin level before the change is different from the selected pin level after the change. If the IRQn interrupt request (n = 0 to 15 for H8S/2456 Group, n = 0 to 7 for H8S/2454 Group) is enabled, the interrupt exception handling is executed. To prevent the unintended interrupt, ITSR setting should be changed while the IRQn interrupt request is disabled, then the IRQnF bit should be cleared to 0.

5.7.6 IRQ Status Register (ISR)

Depending on the pin status following a reset, IRQnF may be set to 1. Therefore, always read ISR and clear it to 0 after resets.

Section 6 Bus Controller (BSC)

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

The bus controller also has a bus arbitration function, and controls the operation of the bus mastership—the CPU, DMA controller (DMAC), EXDMA controller (EXDMAC)*, and data transfer controller (DTC). A block diagram of the bus controller is shown in figure 6.1.

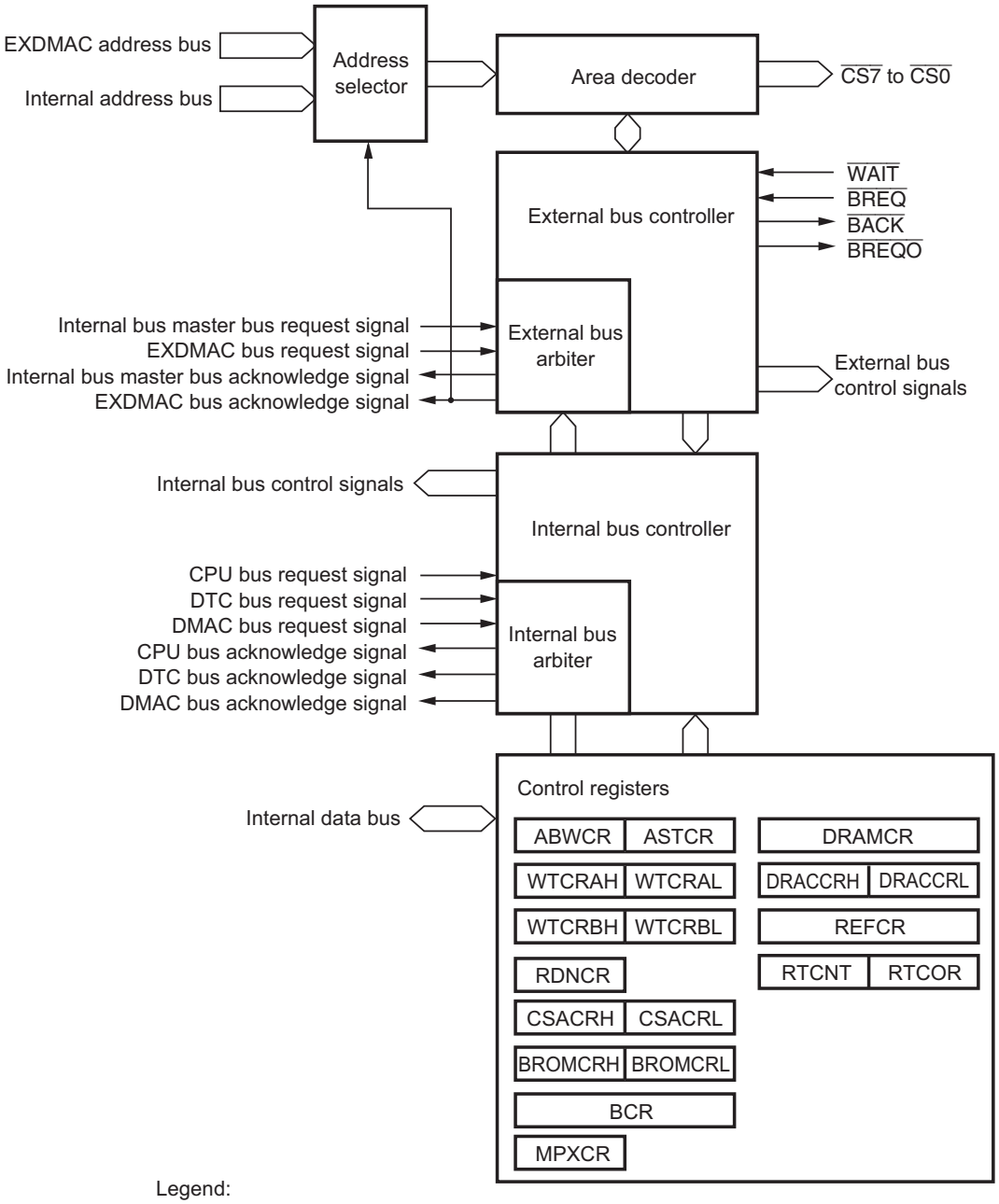
Note: * Not supported by the H8S/2454 Group.

6.1 Features

- Manages external address space in area units
  Manages the external address space divided into eight areas of 2 Mbytes
  Bus specifications can be set independently for each area
  Burst ROM, DRAM, synchronous DRAM*[1], and address/data multiplexed I/O interfaces can be set
- Basic bus interface
  Chip select signals ($\overline{CS0}$ to $\overline{CS7}$) can be output for areas 0 to 7
  8-bit access or 16-bit access can be selected for each area
  2-state access or 3-state access can be selected for each area
  Program wait cycles can be inserted for each area
  Extension cycles can be inserted while $\overline{CS}$ is asserted for each area
  Wait cycles can be inserted by the $\overline{WAIT}$ pin
  The negation timing of the read strobe signal ($\overline{RD}$) can be modified
- Burst ROM interface
  Burst ROM interface 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 6 and 7
- DRAM interface
  DRAM interface can be set for areas 2 to 5
- Synchronous DRAM interface*[1]
  Continuous synchronous DRAM space can be set for areas 2 to 5

- Idle cycle insertion
  Idle cycles can be inserted between external read cycles to different areas
  Idle cycles can be inserted before the write cycle after a read cycle
  Idle cycles can be inserted before the read cycle after a write cycle
- Write buffer function
  External write cycles and internal accesses can be executed in parallel
  DMAC single address transfers and internal accesses can be executed in parallel
- Bus arbitration function
  Includes a bus arbiter that arbitrates bus mastership between the CPU, DMAC, DTC, and EXDMAC*2

Notes: 1. Not supported by the H8S/2456 Group and H8S/2454 Group.
2. Not supported by the H8S/2454 Group.



Legend:

ABWCR	: Bus width control register
ASTCR	: Access state control register
WTCRAH, WTCRAL, WTCRBH, and WTCRBL	: Wait control registers AH, AL, BH, and BL
RDNCR	: Read strobe timing control register
CSACRH and CSACRL	: $\overline{CS}$ assertion period control registers H and L
BROMCRH	: Area 0 burst ROM interface control register
BROMCRL	: Area 1 burst ROM interface control register
BCR	: Bus control register
MPXCR	: Address/data multiplexed I/O control register
DRAMCR	: DRAM control register
DRACCRH and DRACCRL	: DRAM access control registers H and L
REFCR	: Refresh control register
RTCNT	: Refresh timer counter
RTCOR	: Refresh time constant register

Figure 6.1 Block Diagram of Bus Controller

6.2 Input/Output Pins

Table 6.1 shows the pin configuration of the bus controller.

Table 6.1 Pin Configuration

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

Name	Symbol	I/O	Function
Chip select 4/ row address strobe 4/ write enable*1	$\overline{CS4}$/ $\overline{RAS4}$/ $\overline{WE}$*1	Output	Strobe signal indicating that area 4 is selected, DRAM row address strobe signal when area 4 is DRAM space, or write enable signal of the synchronous DRAM when the synchronous DRAM interface is selected.
Chip select 5/ row address strobe 5/ SDRAMϕ*1	$\overline{CS5}$/ $\overline{RAS5}$/ SDRAMϕ*1	Output	Strobe signal indicating that area 5 is selected, DRAM row address strobe signal when area 5 is DRAM space, or dedicated clock signal for the synchronous DRAM when the synchronous DRAM interface is selected.
Chip select 6	$\overline{CS6}$	Output	Strobe signal indicating that area 6 is selected.
Chip select 7	$\overline{CS7}$	Output	Strobe signal indicating that area 7 is selected.
Upper column address strobe/ upper data mask enable*1	$\overline{UCAS}$/ DQMU*1	Output	16-bit DRAM space upper column address strobe signal, 8-bit DRAM space column address strobe signal, upper data mask signal of 16-bit synchronous DRAM space, or data mask signal of 8-bit synchronous DRAM space.
Lower column address strobe/ lower data mask enable*1	$\overline{LCAS}$/ DQML*1	Output	16-bit DRAM space lower column address strobe signal or lower data mask signal for the 16-bit synchronous DRAM space.
Output enable/clock enable	$\overline{OE}$/ CKE*1	Output	Output enable signal for the DRAM space or clock enable signal for the synchronous DRAM space.
Wait	$\overline{WAIT}$	Input	Wait request signal when accessing external address space.
Bus request	$\overline{BREQ}$	Input	Request signal for release of bus to external bus master.
Bus request acknowledge	$\overline{BACK}$	Output	Acknowledge signal indicating that bus has been released to external bus master.
Bus request output	$\overline{BREQO}$	Output	External bus request signal used when internal bus master accesses external address space when external bus is released.

Name	Symbol	I/O	Function
Data transfer acknowledge 1 (DMAC)	$\overline{\text{DACK1}}$	Output	Data transfer acknowledge signal for single address transfer by DMAC channel 1.
Data transfer acknowledge 0 (DMAC)	$\overline{\text{DACK0}}$	$\overline{\text{DACK0}}$	Data transfer acknowledge signal for single address transfer by DMAC channel 0.
Data transfer acknowledge 3*2 (EXDMAC)	$\overline{\text{EDACK3}}$*2	Output	Data transfer acknowledge signal for single address transfer by EXDMAC channel 3.
Data transfer acknowledge 2*2 (EXDMAC)	$\overline{\text{EDACK2}}$*2	Output	Data transfer acknowledge signal for single address transfer by EXDMAC channel 2.

Notes: 1. Not supported by the H8S/2456 Group and H8S/2454 Group
2. Not supported by the H8S/2454 Group.

6.3 Register Descriptions

The bus controller has the following registers.

- Bus width control register (ABWCR)
- Access state control register (ASTCR)
- Wait control register AH (WTCRAH)
- Wait control register AL (WTCRAL)
- Wait control register BH (WTCRBH)
- Wait control register BL (WTCRBL)
- Read strobe timing control register (RDNCR)
- $\overline{\text{CS}}$ assertion period control register H (CSACRH)
- $\overline{\text{CS}}$ assertion period control register L (CSACRL)
- Area 0 burst ROM interface control register (BROMCRH)
- Area 1 burst ROM interface control register (BROMCRL)
- Bus control register (BCR)
- Address/data multiplexed I/O control register (MPXCR)
- DRAM control register (DRAMCR)
- DRAM access control register (DRACCR)
- Refresh control register (REFCR)
- Refresh timer counter (RTCNT)
- Refresh time constant register (RTCOR)

6.3.1 Bus Width Control Register (ABWCR)

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

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

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

6.3.2 Access State Control Register (ASTCR)

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

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

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

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

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

- WTCRAH

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

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

- WTCRAL

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
6	W52	1	R/W	Area 5 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 5 while AST5 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
5	W51	1	R/W	
4	W50	1	R/W	
3	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
2	W42	1	R/W	Area 4 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 4 while AST4 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
1	W41	1	R/W	
0	W40	1	R/W	

- WTCRBH

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

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

Legend: X: Don't care.

Note: * The synchronous DRAM interface is not supported by the H8S/2456 Group and H8S/2454 Group.

- WTCRBL

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

6.3.4 Read Strobe Timing Control Register (RDNCR)

RDNCR selects the read strobe signal ($\overline{RD}$) negation timing in a basic bus interface read access.

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

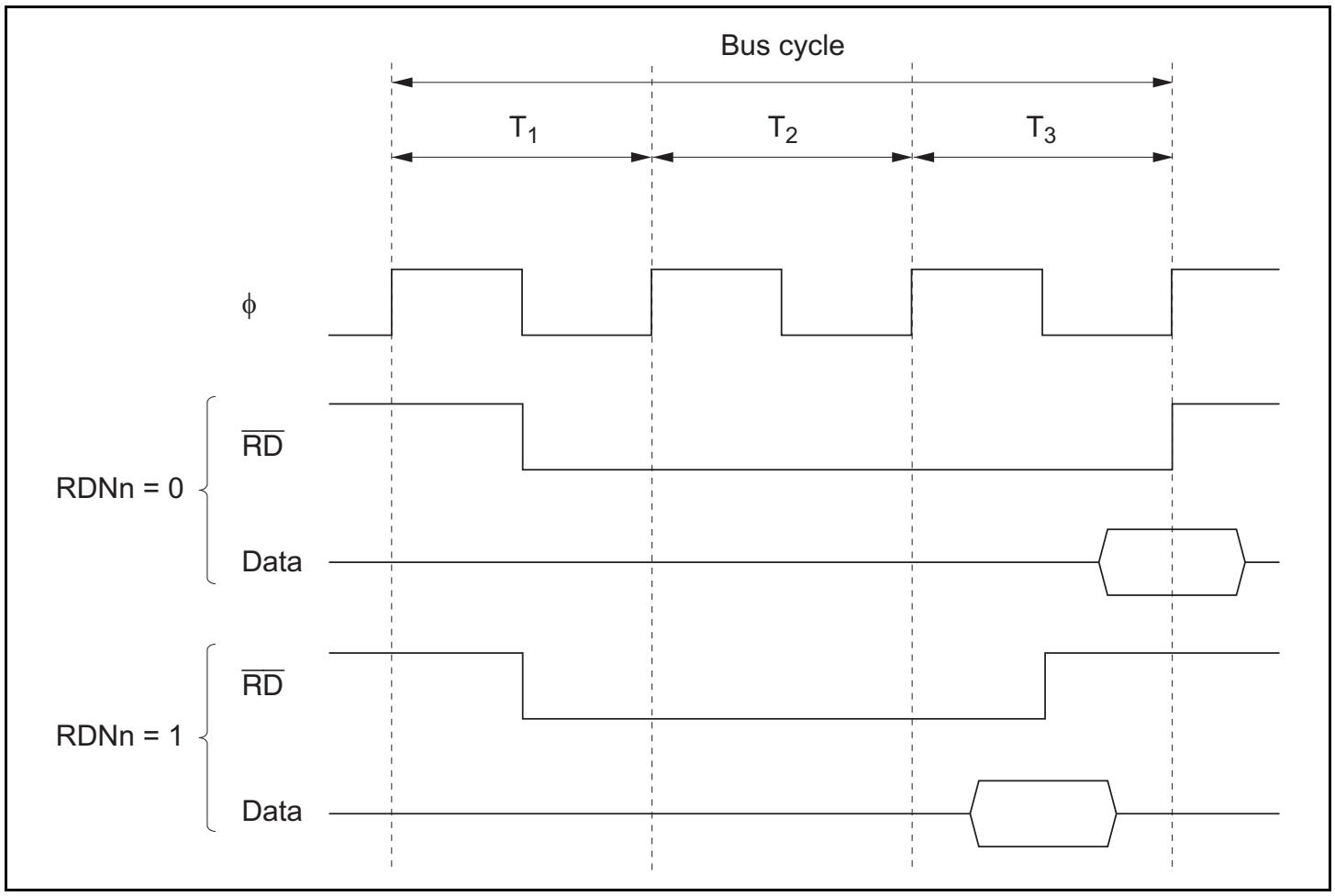


Figure 6.2 Read Strobe Negation Timing (Example of 3-State Access Space)

6.3.5 $\overline{CS}$ Assertion Period Control Registers H, L (CSACRH, CSACRL)

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

- CSACRH

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

- CSACRL

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

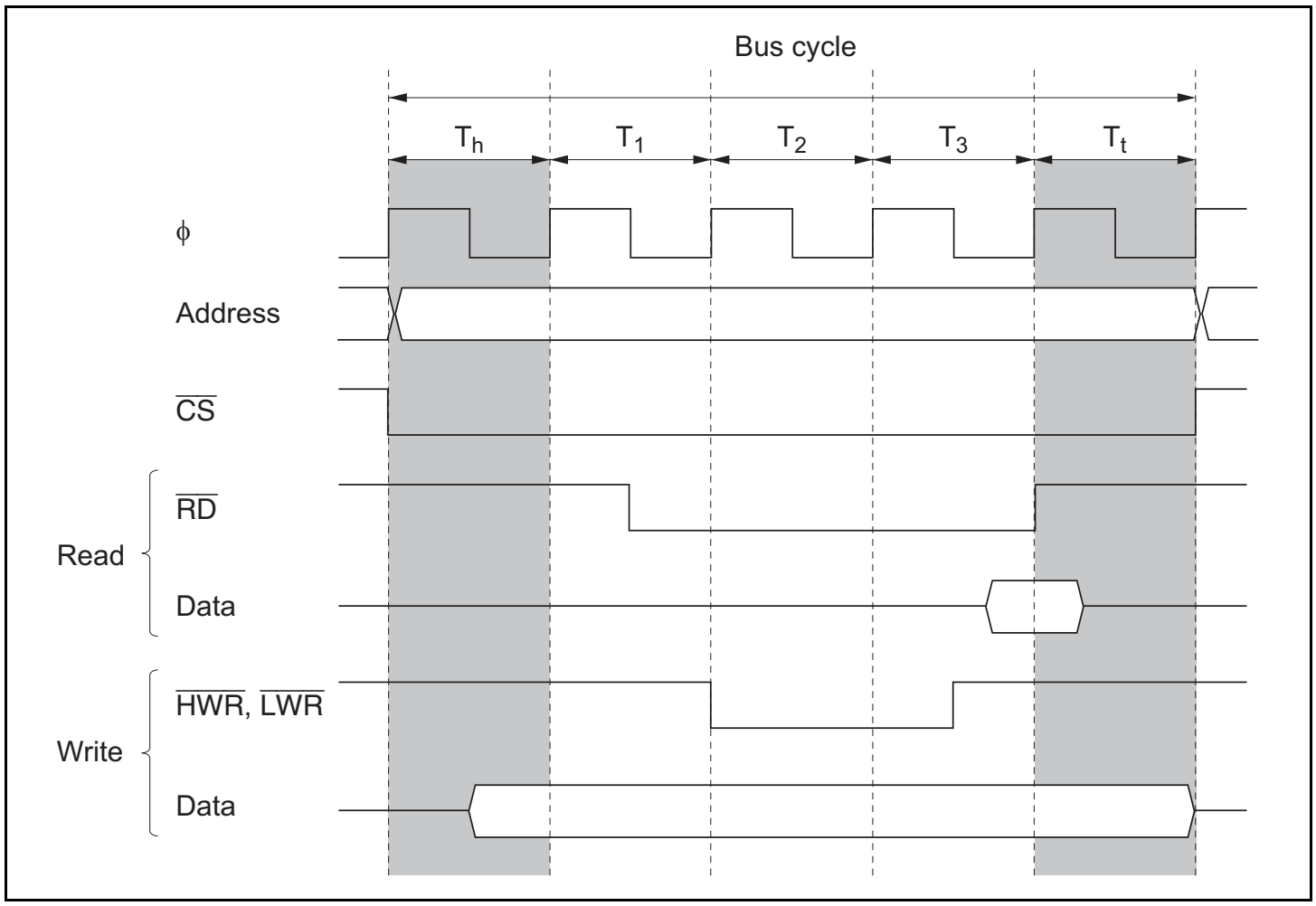


Figure 6.3 $\overline{CS}$ and Address Assertion Period Extension (Example of 3-State Access Space and RDNn = 0)

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

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

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

(n = 1 or 0)

6.3.7 Bus Control Register (BCR)

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

Bit	Bit Name	Initial Value	R/W	Description
15	BRLE	0	R/W	External Bus Release Enable Enables or disables external bus release. 0: External bus release disabled $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$, and $\overline{\text{BREQO}}$ pins can be used as I/O ports 1: External bus release enabled
14	BREQOE	0	R/W	$\overline{\text{BREQO}}$ Pin Enable Controls outputting the bus request signal (BREQO) to the external bus master in the external bus released state, when an internal bus master performs an external address space access, or when a refresh request is generated. 0: $\overline{\text{BREQO}}$ output disabled $\overline{\text{BREQO}}$ pin can be used as I/O port 1: $\overline{\text{BREQO}}$ output enabled
13	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.
12	IDLC	1	R/W	Idle Cycle State Number Select Specifies the number of states in the idle cycle set by ICIS2, ICIS1, and ICIS0. 0: Idle cycle comprises 1 state 1: Idle cycle comprises 2 states
11	ICIS1	1	R/W	Idle Cycle Insert 1 When consecutive external read cycles are executed in different areas, an idle cycle can be inserted between the bus cycles. 0: Idle cycle not inserted 1: Idle cycle inserted

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

6.3.8 Address/Data Multiplexed I/O Control Register (MPXCR)

MPXCR is used to make address/data multiplexed I/O interface settings.

Bit	Bit Name	Initial Value	R/W	Description
7	MPXE	0	R/W	Address/Data Multiplexed I/O Interface Enable These bits select the bus interface for areas 6 and 7. 0: Basic bus interface 1: Address/data multiplexed I/O interface
6 to 1	—	All 0	R/W	Reserved These bits can be read from or written to. However, the write value should always be 0.
0	ADDEX	0	R/W	Address Output Cycle Extension Specifies whether a wait cycle is inserted for the address output cycle of the address/data multiplexed I/O interface. 0: No wait cycle inserted 1: One wait cycle inserted

6.3.9 DRAM Control Register (DRAMCR)

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

Note: The synchronous DRAM interface is not supported by the H8S/2456 Group and H8S/2454 Group.

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

Bit	Bit Name	Initial Value	R/W	Description
12	CAST	0	R/W	Column Address Output Cycle Number Select Selects whether the column address output cycle in DRAM access comprises 3 states or 2 states. The setting of this bit applies to all areas designated as DRAM space. 0: Column address output cycle comprises 2 states 1: Column address output cycle comprises 3 states
11	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10	RMTS2	0	R/W	DRAM/Continuous Synchronous DRAM Space Select These bits designate DRAM/continuous synchronous DRAM space for areas 2 to 5. When continuous DRAM space is set, it is possible to connect large-capacity DRAM exceeding 2 Mbytes per area. In this case, the $\overline{RAS}$ signal is output from the $\overline{CS2}$ pin. When continuous synchronous DRAM space is set, it is possible to connect large-capacity synchronous DRAM exceeding 2 Mbytes per area. In this case, the $\overline{RAS}$, $\overline{CAS}$, and $\overline{WE}$ signals are output from $\overline{CS2}$, $\overline{CS3}$, and $\overline{CS4}$ pins, respectively. When synchronous DRAM mode is set, the mode registers of the synchronous DRAM can be set. 000: Normal space 001: Normal space in areas 3 to 5 DRAM space in area 2 010: Normal space in areas 4 and 5 DRAM space in areas 2 and 3 011: DRAM space in areas 2 to 5 100: Continuous synchronous DRAM space (setting possible only in H8S/2456R Group) 101: Synchronous DRAM mode setting (setting possible only in H8S/2456R Group) 110: Setting prohibited 111: Continuous DRAM space in areas 2 to 5
9	RMTS1	0	R/W	
8	RMTS0	0	R/W	
7	BE	0	R/W	Burst Access Enable Selects enabling or disabling of burst access to areas designated as DRAM/continuous synchronous DRAM space. DRAM/continuous synchronous DRAM space burst access is performed in fast page mode. When using EDO page mode DRAM, the $\overline{OE}$ signal must be connected. 0: Full access 1: Access in fast page mode

Bit	Bit Name	Initial Value	R/W	Description
6	RCDM	0	R/W	$\overline{\text{RAS}}$ Down Mode When access to DRAM space is interrupted by an access to normal space, an access to an internal I/O register, etc., this bit selects whether the $\overline{\text{RAS}}$ signal is held low while waiting for the next DRAM access ($\overline{\text{RAS}}$ down mode), or is driven high again ($\overline{\text{RAS}}$ up mode). The setting of this bit is valid only when the BE bit is set to 1. If this bit is cleared to 0 when set to 1 in the $\overline{\text{RAS}}$ down state, the $\overline{\text{RAS}}$ down state is cleared at that point, and $\overline{\text{RAS}}$ goes high. When continuous synchronous DRAM space is set, reading from and writing to this bit is enabled. However, the setting does not affect the operation. 0: $\overline{\text{RAS}}$ up mode selected for DRAM space access 1: $\overline{\text{RAS}}$ down mode selected for DRAM space access
5	DDS	0	R/W	DMAC Single Address Transfer Option Selects whether full access is always performed or burst access is enabled when DMAC single address transfer is performed on the DRAM/synchronous DRAM. When the BE bit is cleared to 0 in DRAMCR, disabling DRAM/synchronous DRAM burst access, DMAC single address transfer is performed in full access mode regardless of the setting of this bit. This bit has no effect on other bus master external accesses or DMAC dual address transfers. 0: Full access is always executed 1: Burst access is enabled

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

Bit	Bit Name	Initial Value	R/W	Description
2	MXC2	0	R/W	Address Multiplex Select
1	MXC1	0	R/W	These bits select the size of the shift toward the lower half of the row address in row address/column address multiplexing. In burst operation on the DRAM/synchronous DRAM interface, these bits also select the row address bits to be used for comparison. When the MXC2 bit is set to 1 while continuous synchronous DRAM space is set, the address precharge setting command (Precharge-sel) is output to the upper column address. For details, refer to sections 6.7.2 and 6.8.2, Address Multiplexing. DRAM interface 000: 8-bit shift • When 8-bit access space is designated: Row address bits A23 to A8 used for comparison • When 16-bit access space is designated: Row address bits A23 to A9 used for comparison 001: 9-bit shift • When 8-bit access space is designated: Row address bits A23 to A9 used for comparison • When 16-bit access space is designated: Row address bits A23 to A10 used for comparison 010: 10-bit shift • When 8-bit access space is designated: Row address bits A23 to A10 used for comparison • When 16-bit access space is designated: Row address bits A23 to A11 used for comparison
0	MXC0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
2	MXC2	0	R/W	011: 11-bit shift
1	MXC1	0	R/W	• When 8-bit access space is designated: Row address bits A23 to A11 used for comparison When 16-bit access space is designated: Row address bits A23 to A12 used for comparison
0	MXC0	0	R/W	Synchronous DRAM interface 100: 8-bit shift • When 8-bit access space is designated: Row address bits A23 to A8 used for comparison • When 16-bit access space is designated: Row address bits A23 to A9 used for comparison The precharge-sel is A15 to A9 of the column address. 101: 9-bit shift • When 8-bit access space is designated: Row address bits A23 to A9 used for comparison • When 16-bit access space is designated: Row address bits A23 to A10 used for comparison The precharge-sel is A15 to A10 of the column address. 110: 10-bit shift • When 8-bit access space is designated: Row address bits A23 to A10 used for comparison • When 16-bit access space is designated: Row address bits A23 to A11 used for comparison The precharge-sel is A15 to A11 of the column address.

Bit	Bit Name	Initial Value	R/W	Description
2	MXC2	0	R/W	111: 11-bit shift
1	MXC1	0	R/W	• When 8-bit access space is designated: Row address bits A23 to A11 used for comparison
0	MXC0	0	R/W	• When 16-bit access space is designated: Row address bits A23 to A12 used for comparison The precharge-sel is A15 to A12 of the column address.

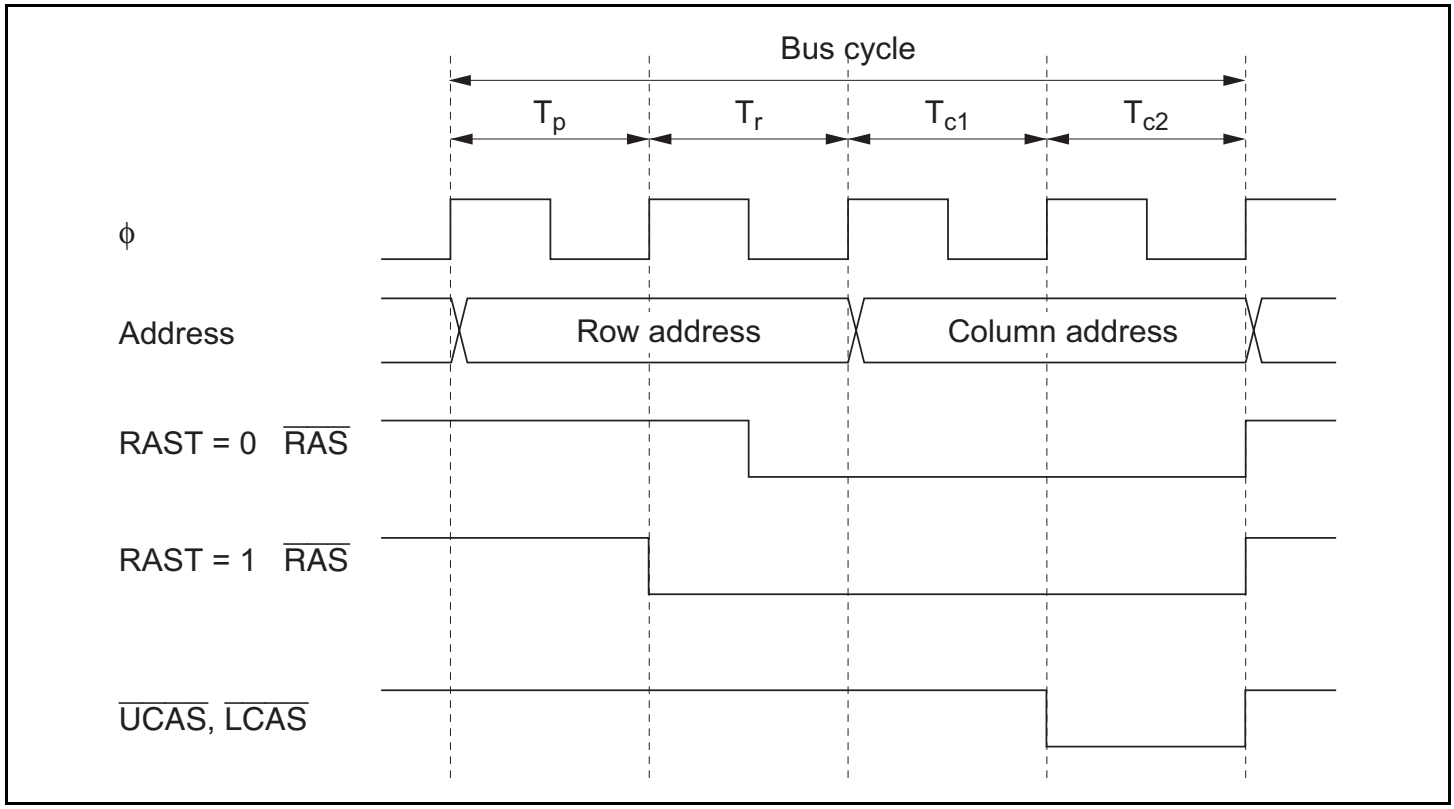


Figure 6.4 $\overline{\text{RAS}}$ Signal Assertion Timing (2-State Column Address Output Cycle, Full Access)

6.3.10 DRAM Access Control Register (DRACCR)

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

Note: The synchronous DRAM interface is not supported by the H8S/2456 Group and H8S/2454 Group.

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

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

Note: * Not supported by the H8S/2456 Group and H8S/2454 Group.

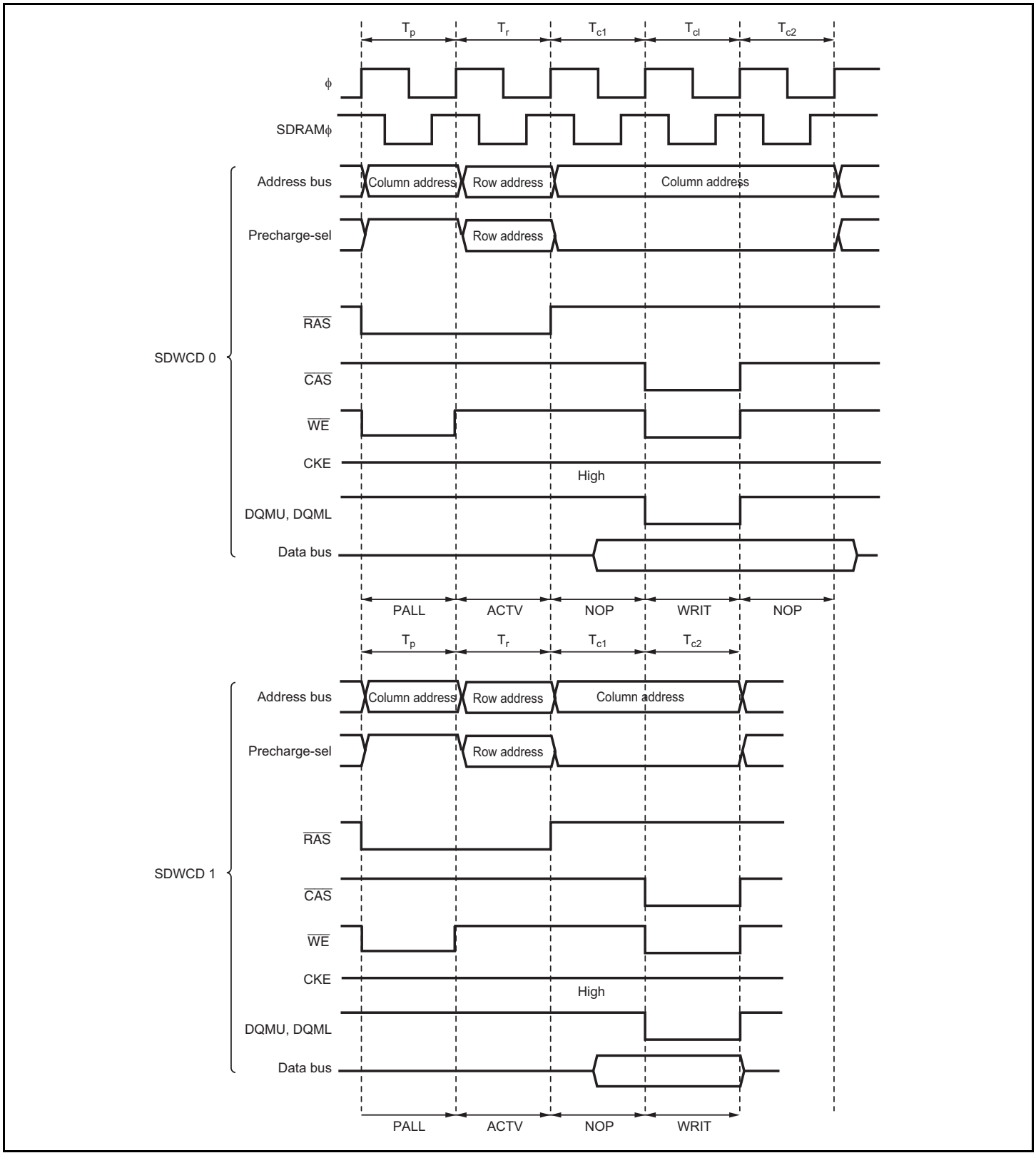


Figure 6.5 CAS Latency Control Cycle Disable Timing during Continuous Synchronous DRAM Space Write Access (for CAS Latency 2)

6.3.11 Refresh Control Register (REFCR)

REFCR specifies DRAM/synchronous DRAM interface refresh control.

Note: The synchronous DRAM interface is not supported by the H8S/2456 Group and H8S/2454 Group.

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

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

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

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

6.3.12 Refresh Timer Counter (RTCNT)

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

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

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

6.3.13 Refresh Time Constant Register (RTCOR)

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

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

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

6.4 Bus Control

6.4.1 Area Division

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

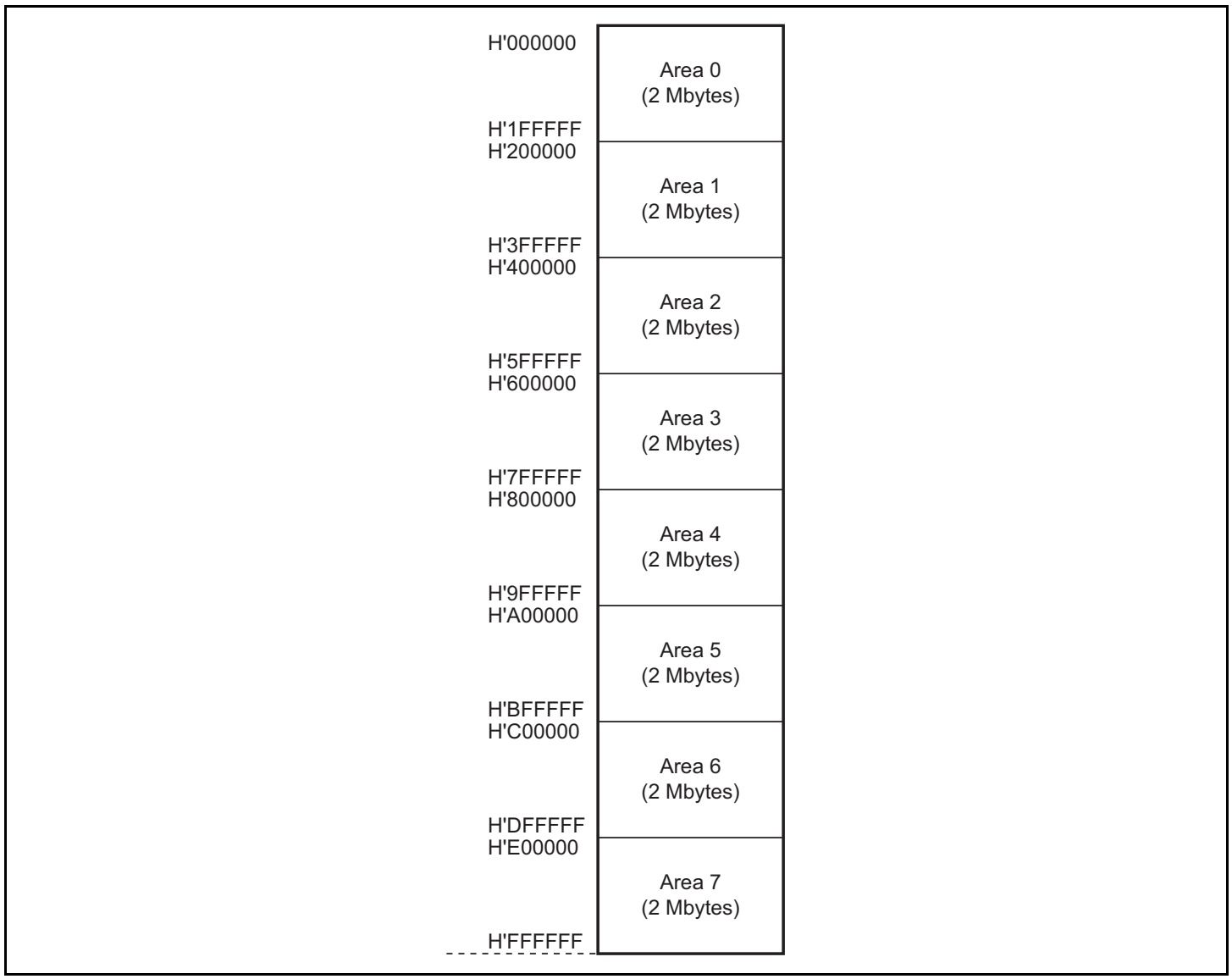


Figure 6.6 Area Divisions

6.4.2 Bus Specifications

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

(1) Bus Width

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

(2) Number of Access States

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

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

Note: The synchronous DRAM interface is not supported by the H8S/2456 Group and H8S/2454 Group.

(3) Number of Program Wait States

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

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

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

(n = 0 to 7)

(4) Read Strobe Timing

RDNCR can be used to select either of two negation timings (at the end of the read cycle or one half-state before the end of the read cycle) for the read strobe ($\overline{RD}$) used in the basic bus interface space.

(5) Chip Select ($\overline{CS}$) Assertion Period Extension States

Some external I/O devices require a setup time and hold time between address and $\overline{CS}$ signals and strobe signals such as $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$. CSACR can be used to insert states in which only the $\overline{CS}$, $\overline{AS}$, and address signals are asserted before and after a basic bus space access cycle.

6.4.3 Memory Interfaces

The memory interfaces in this LSI comprise a basic bus interface that allows direct connection of ROM, SRAM, and so on; an address/data multiplexed I/O interface that allows direct connection of peripheral LSIs that require address/data multiplexing, a DRAM interface that allows direct connection of DRAM; a synchronous DRAM interface that allows direct connection of synchronous DRAM; and a burst ROM interface that allows direct connection of burst ROM. The interface can be selected independently for each area.

An area for which the basic bus interface is designated functions as normal space. An area for which the address/data multiplexed I/O interface is designated functions as address/data multiplexed I/O space, an area for which the DRAM interface is designated functions as DRAM space, an area for which the synchronous DRAM interface is designated functions as continuous synchronous DRAM space, and an area for which the burst ROM interface is designated functions as burst ROM space.

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

Note: The synchronous DRAM interface is not supported by the H8S/2456 Group and H8S/2454 Group.

(1) Area 0

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

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

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

(2) Area 1

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

When area 1 external address space is accessed, the $\overline{\text{CS1}}$ signal can be output.

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

(3) Areas 2 to 5

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

When area 2 to 5 external space is accessed, signals $\overline{CS2}$ to $\overline{CS5}$ can be output.

The basic bus interface, DRAM interface, or synchronous DRAM interface can be selected for the memory interface of areas 2 to 5. With the DRAM interface, signals $\overline{CS2}$ and $\overline{CS5}$ are used as $\overline{RAS}$ signals.

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

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

(4) Area 6

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

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

Either the basic bus interface or address/data multiplexed I/O interface can be used for the memory interface of area 6.

(5) Area 7

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

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

Either the basic bus interface or address/data multiplexed I/O interface can be used for the memory interface of area 7.

6.4.4 Chip Select Signals

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

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

In expanded mode with on-chip ROM disabled, the $\overline{CS0}$ pin is placed in the output state after a reset. Pins $\overline{CS1}$ to $\overline{CS7}$ are placed in the input state after a reset and so the corresponding DDR bits and PFCR0 bits should be set to 1 when outputting signals $\overline{CS1}$ to $\overline{CS7}$.

In expanded mode with on-chip ROM enabled, pins $\overline{CS0}$ to $\overline{CS7}$ are all placed in the input state after a reset and so the corresponding DDR bits and PFCR0 bits should be set to 1 when outputting signals $\overline{CS0}$ to $\overline{CS7}$.

When areas 2 to 5 are designated as DRAM space, outputs $\overline{CS2}$ to $\overline{CS5}$ are used as $\overline{RAS}$ signals.

When areas 2 to 5 are designated as continuous synchronous DRAM space in the H8S/2456R Group, outputs $\overline{CS2}$, $\overline{CS3}$, $\overline{CS4}$, and $\overline{CS5}$ are used as $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, and CLK signals.

Note: The A23E bit in PFCR1 should be cleared to 0 when $\overline{CS7}$ signal is output in the H8S/2454 Group.

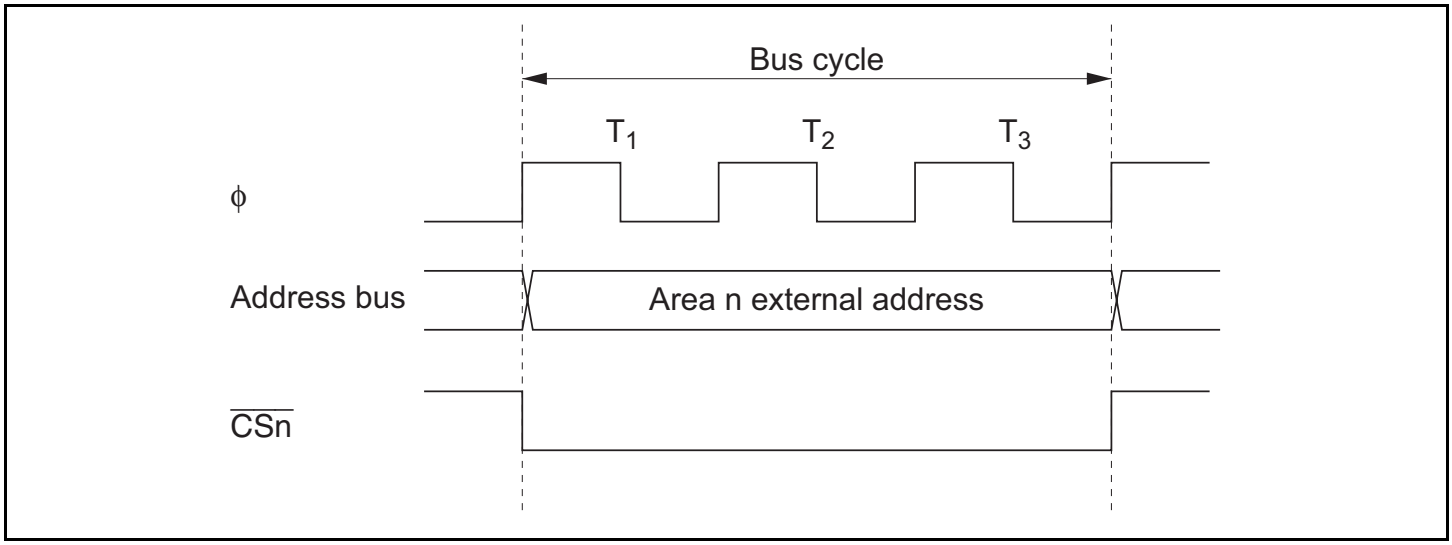


Figure 6.7 $\overline{CSn}$ Signal Output Timing (n = 0 to 7)

6.5 Basic Bus Interface

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

6.5.1 Data Size and Data Alignment

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

(1) 8-Bit Access Space

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

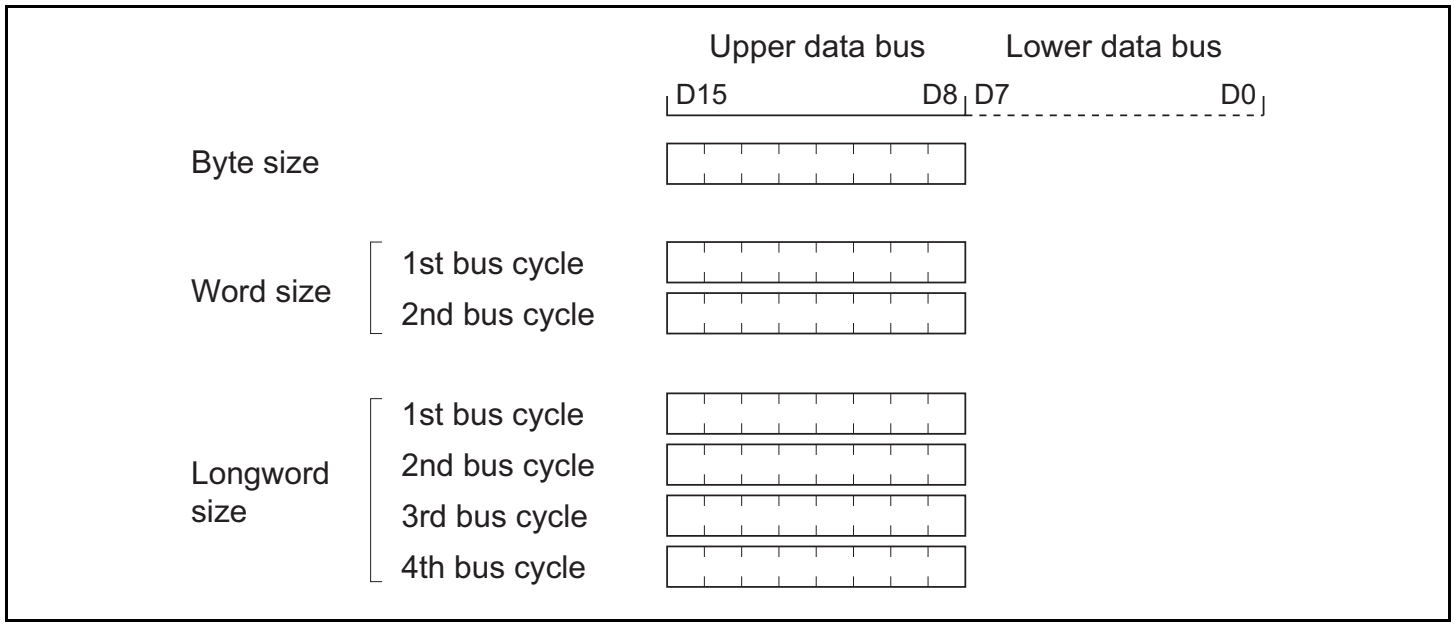


Figure 6.8 Access Sizes and Data Alignment Control (8-Bit Access Space)

(2) 16-Bit Access Space

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

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

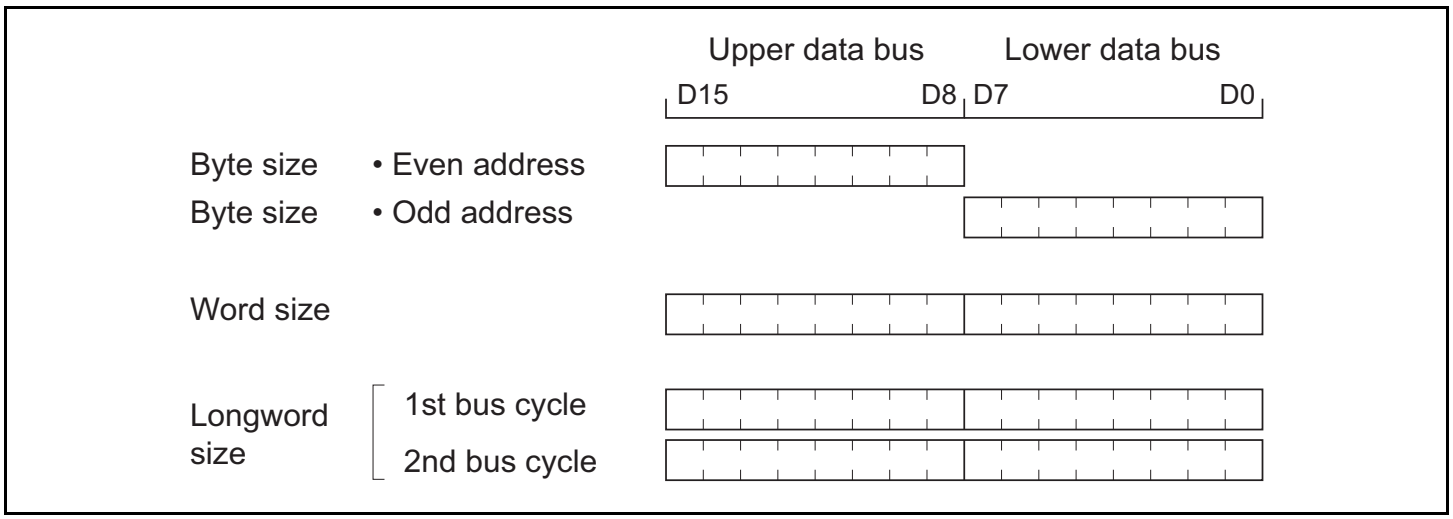


Figure 6.9 Access Sizes and Data Alignment Control (16-Bit Access Space)

6.5.2 Valid Strobes

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

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

Table 6.3 Data Buses Used and Valid Strobes

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

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

6.5.3 Basic Timing

(1) 8-Bit, 2-State Access Space

Figure 6.10 shows the bus timing for an 8-bit, 2-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used. The $\overline{\text{LWR}}$ pin is always fixed high. Wait states can be inserted.

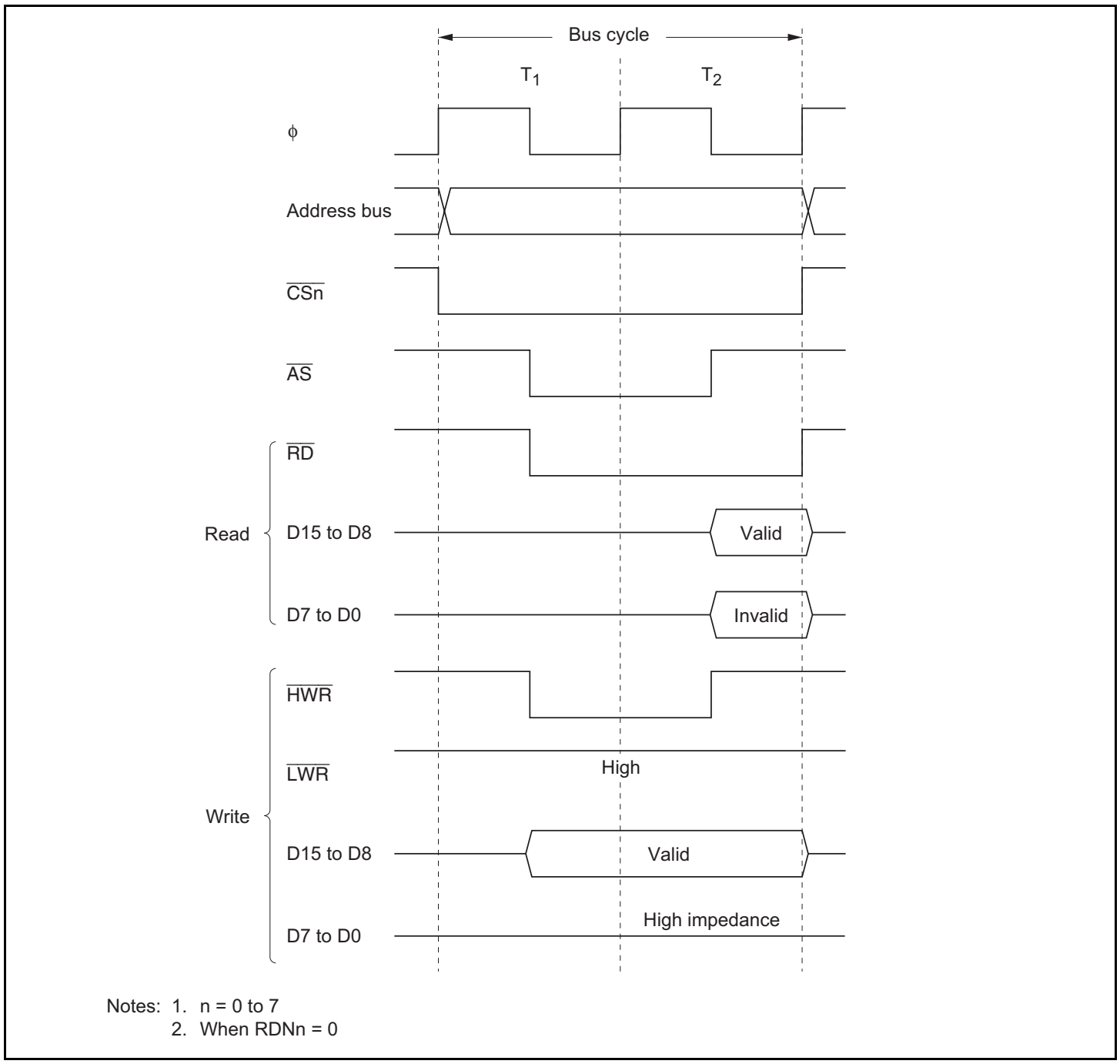


Figure 6.10 Bus Timing for 8-Bit, 2-State Access Space

(2) 8-Bit, 3-State Access Space

Figure 6.11 shows the bus timing for an 8-bit, 3-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used. The $\overline{\text{LWR}}$ pin is always fixed high. Wait states can be inserted.

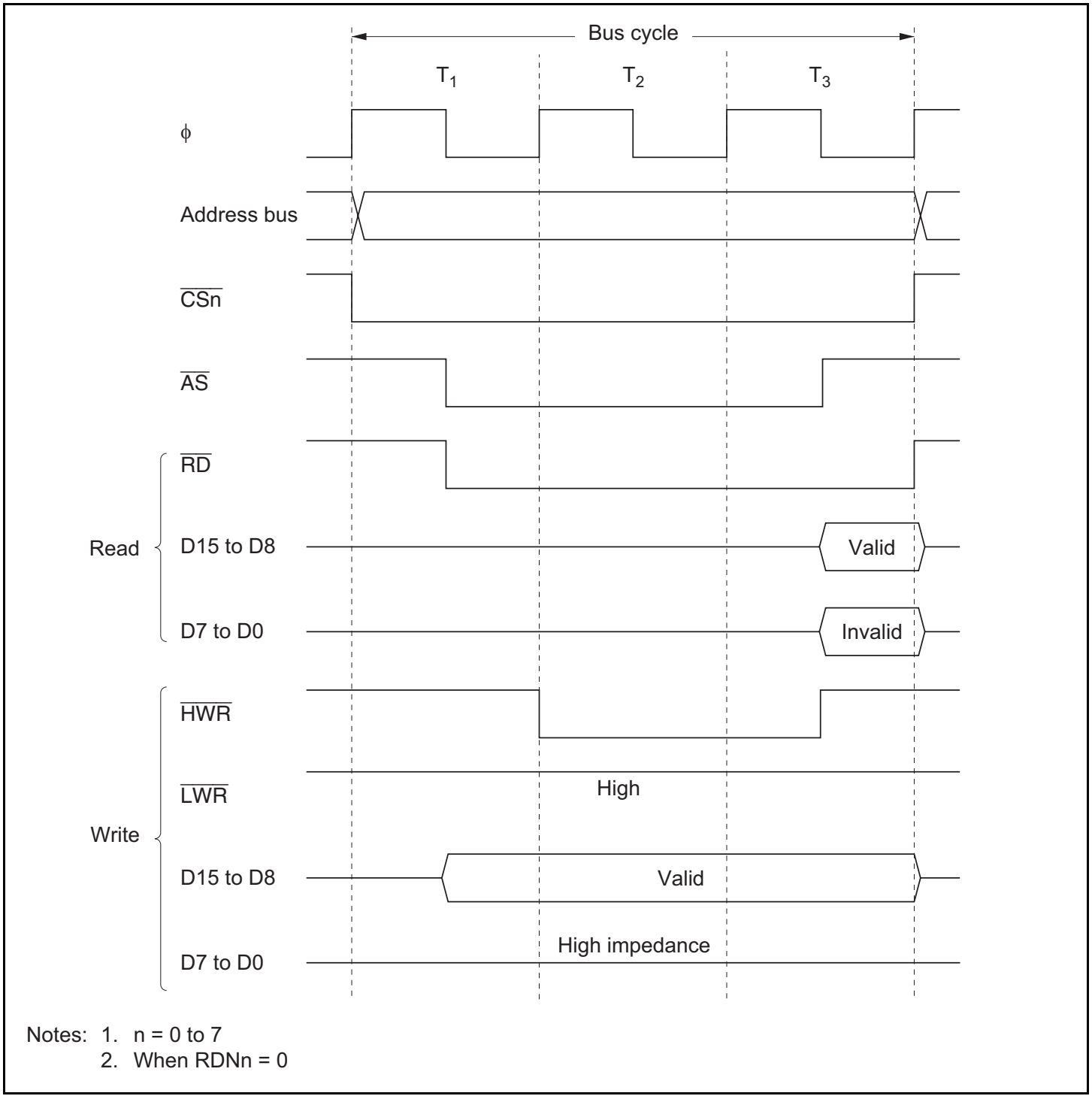


Figure 6.11 Bus Timing for 8-Bit, 3-State Access Space

(3) 16-Bit, 2-State Access Space

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

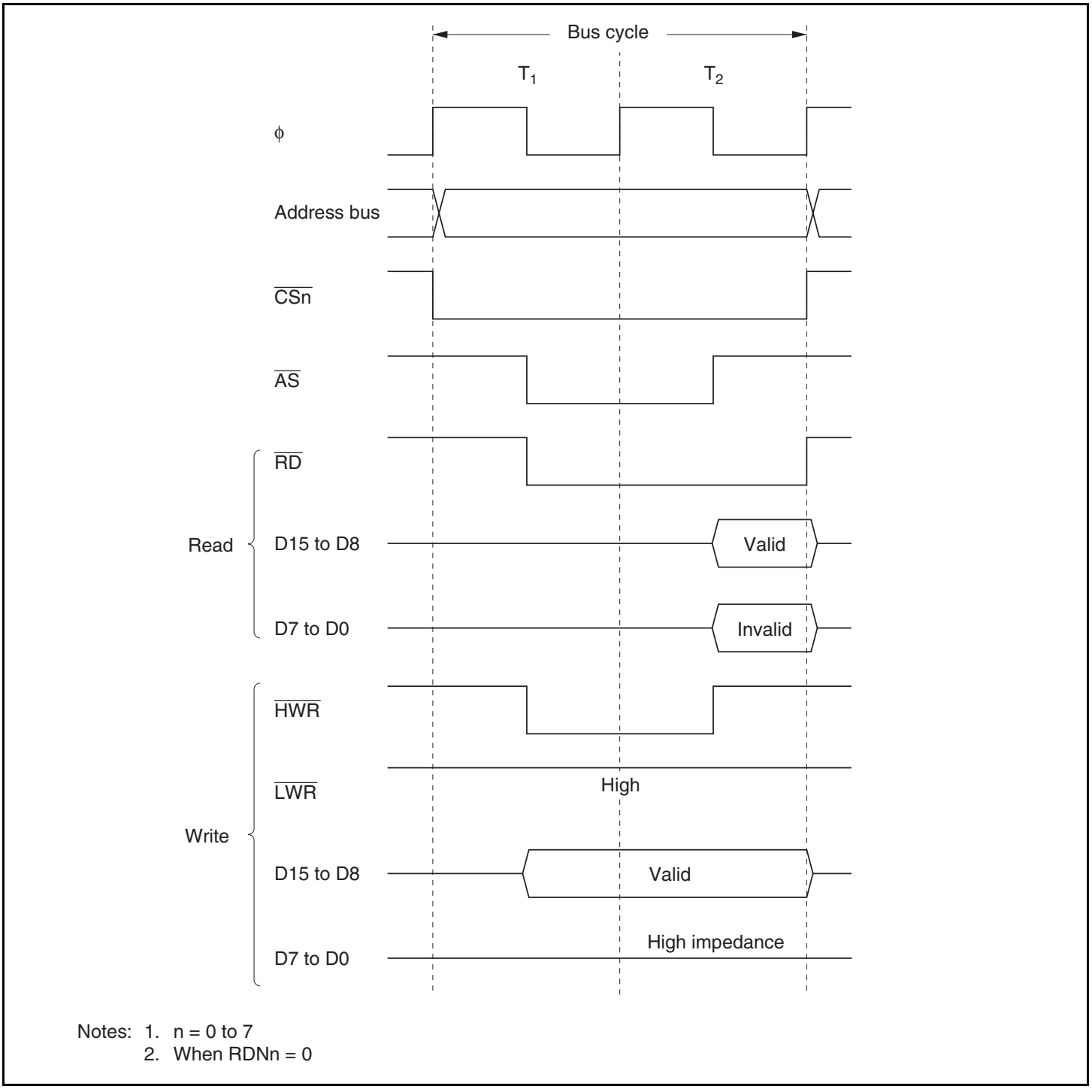
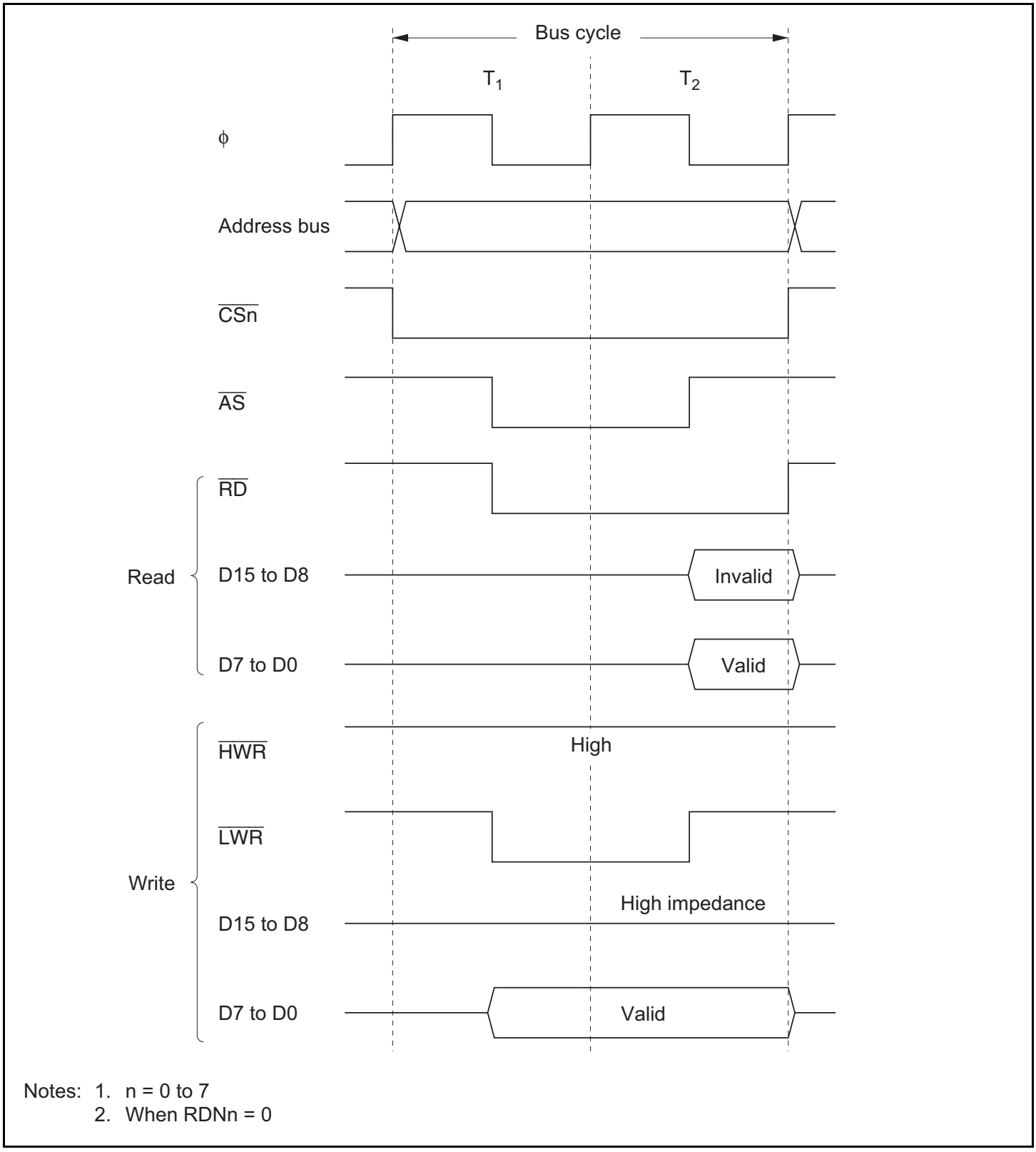
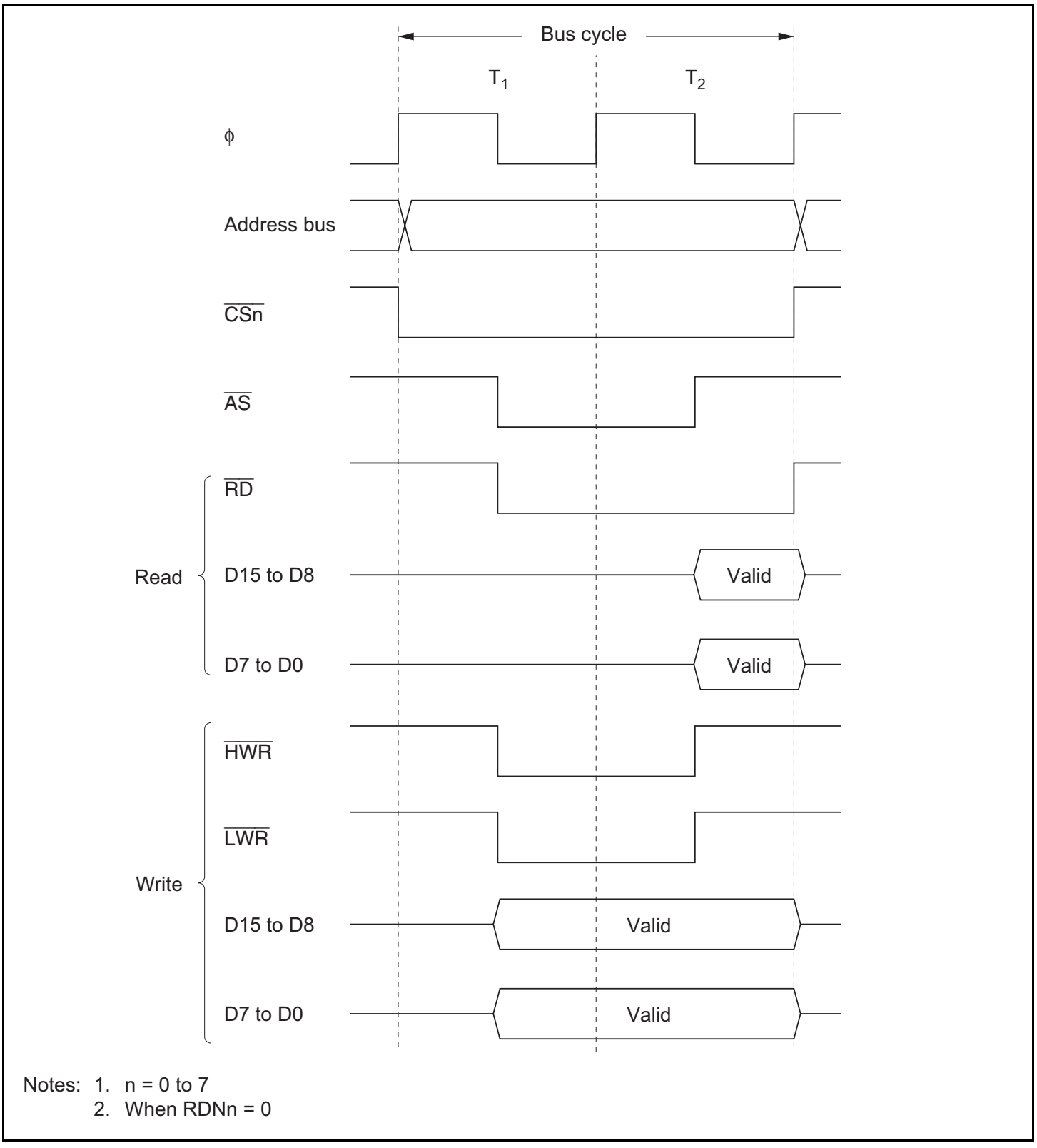


Figure 6.12 Bus Timing for 16-Bit, 2-State Access Space (Even Address Byte Access)



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

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



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

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

(4) 16-Bit, 3-State Access Space

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

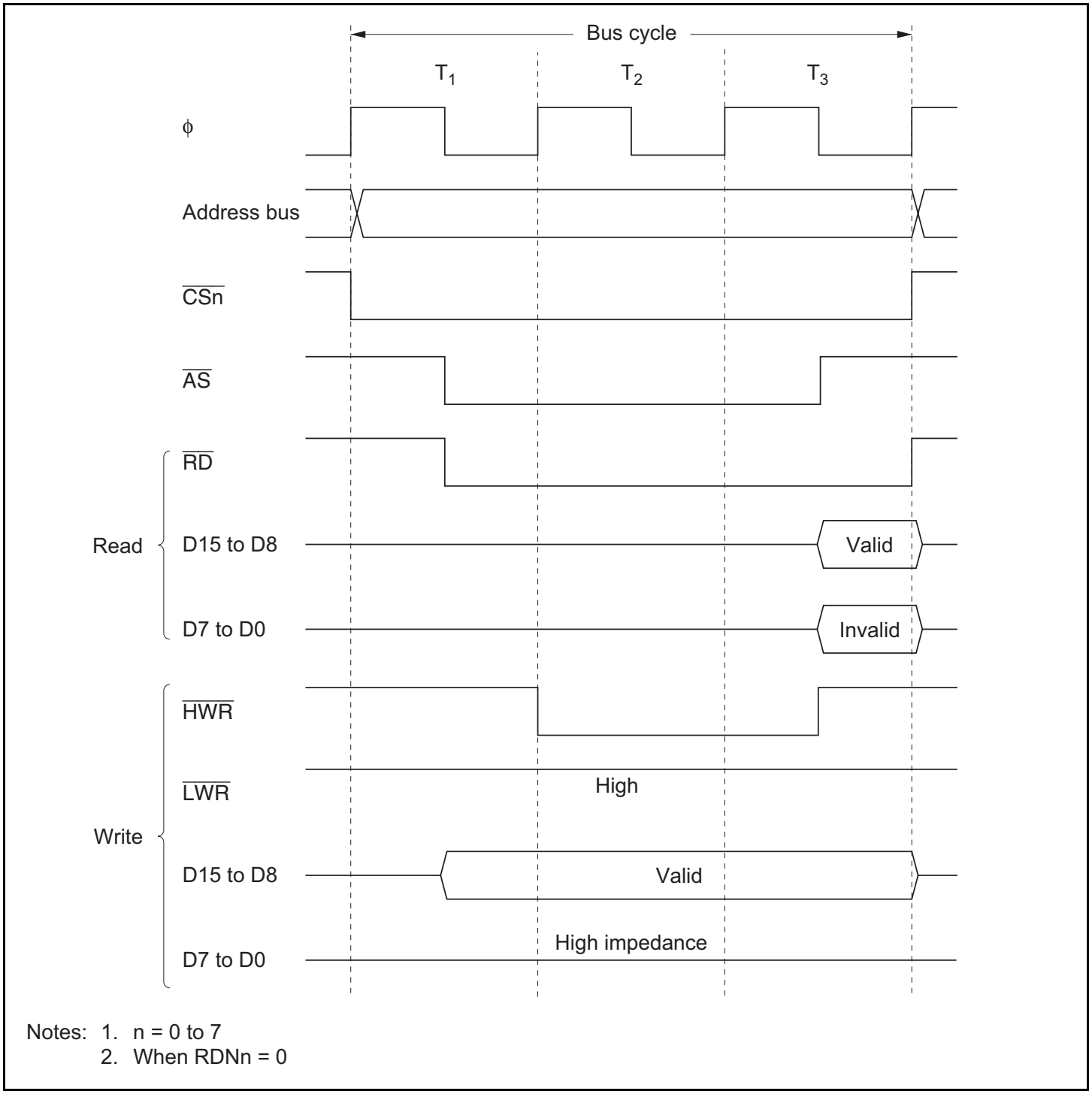
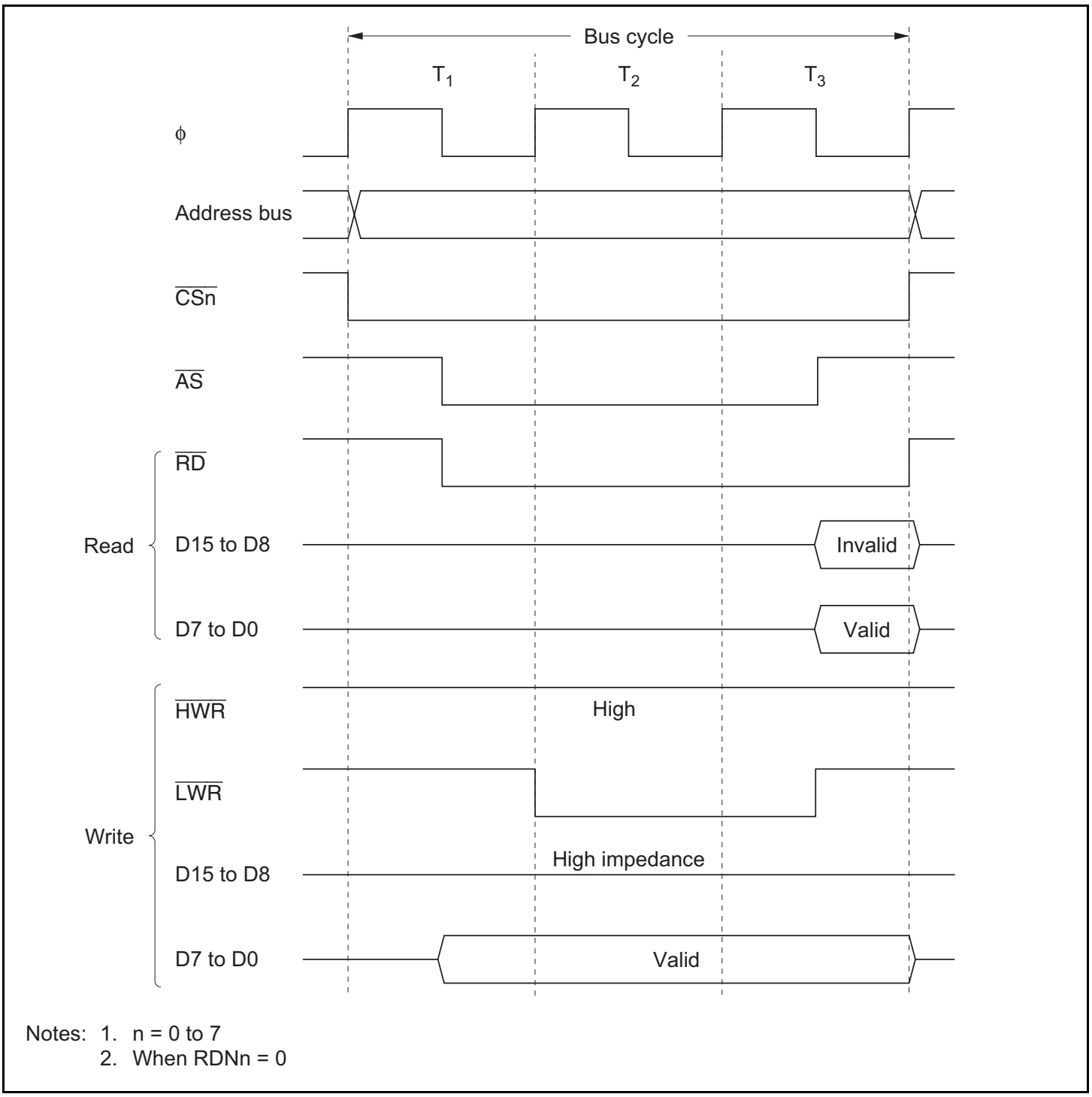
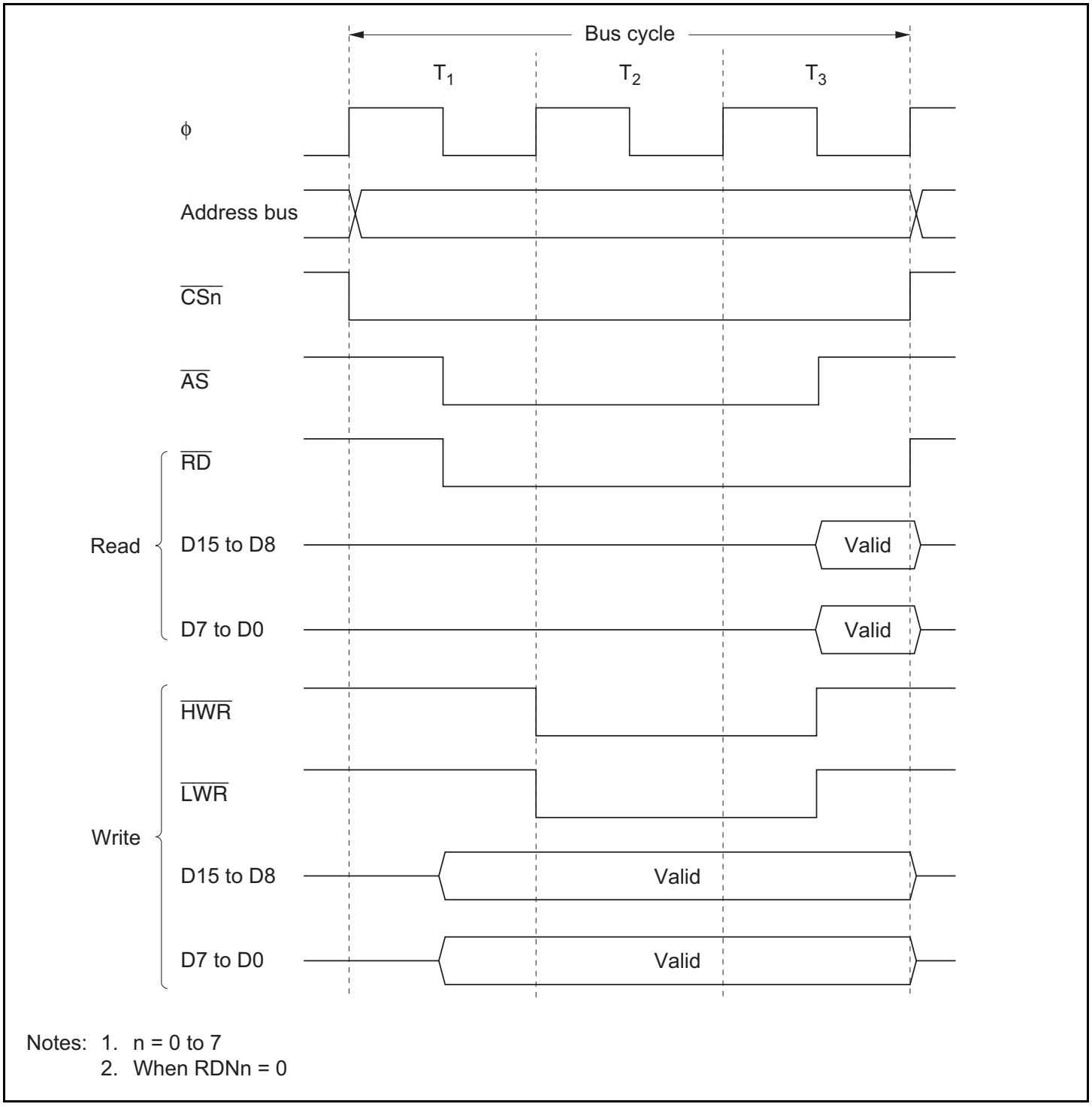


Figure 6.15 Bus Timing for 16-Bit, 3-State Access Space (Even Address Byte Access)



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

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



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

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

6.5.4 Wait Control

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

(1) Program Wait Insertion

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

(2) Pin Wait Insertion

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

The settings after a reset are: 3-state access, insertion of 7 program wait states, and $\overline{\text{WAIT}}$ input disabled.

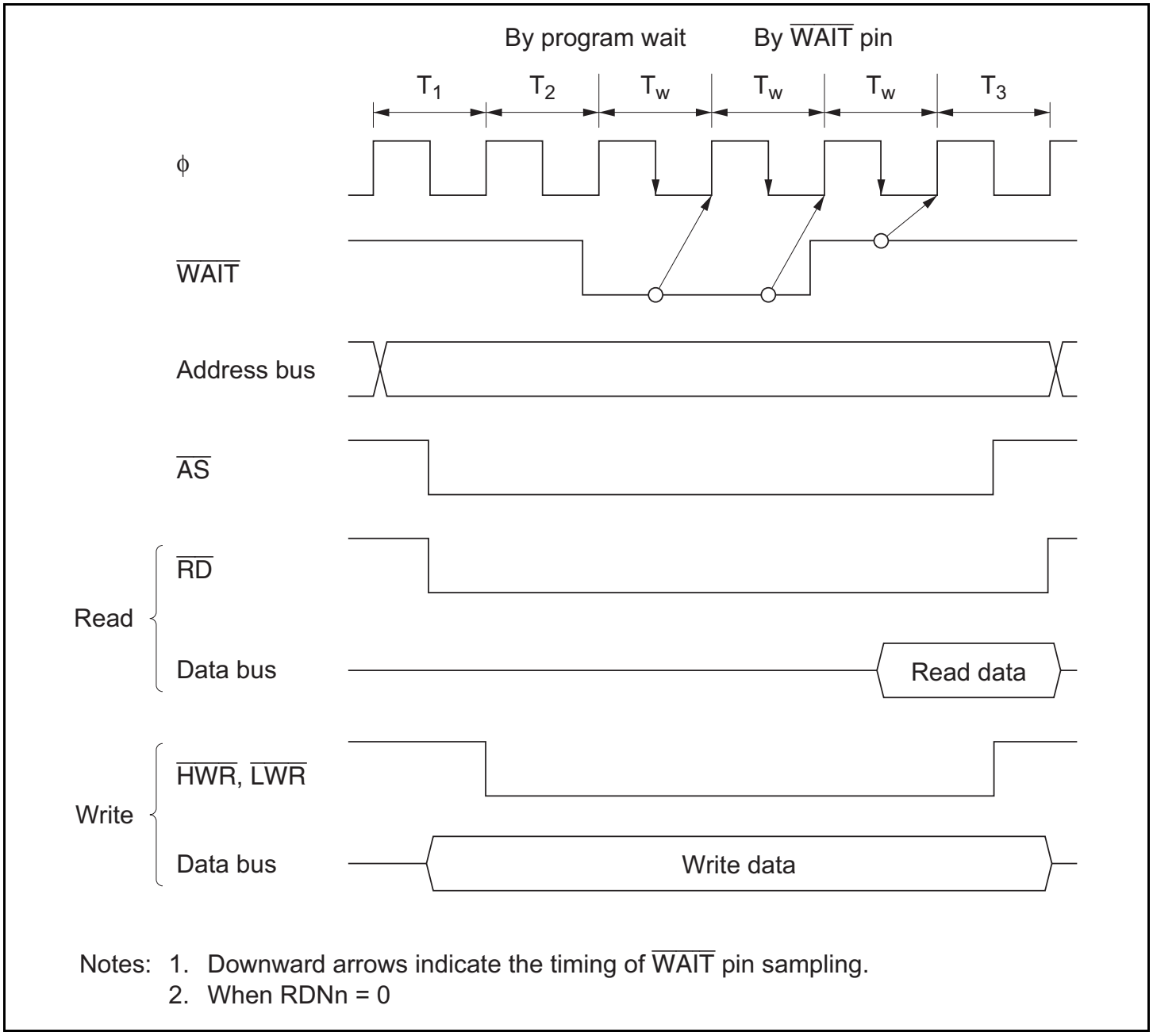


Figure 6.18 Example of Wait State Insertion Timing

6.5.5 Read Strobe ($\overline{RD}$) Timing

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

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

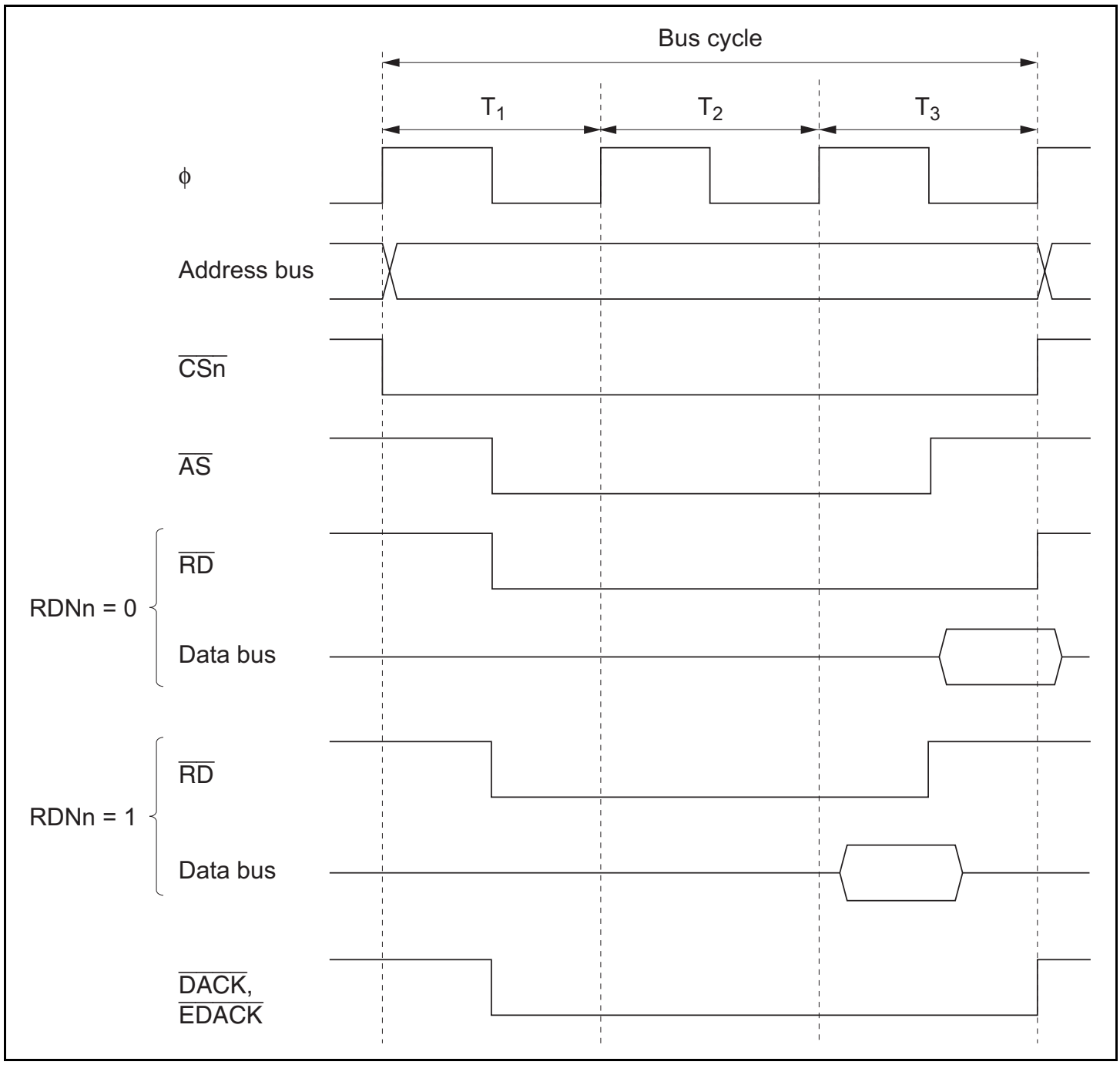


Figure 6.19 Example of Read Strobe Timing

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

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

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

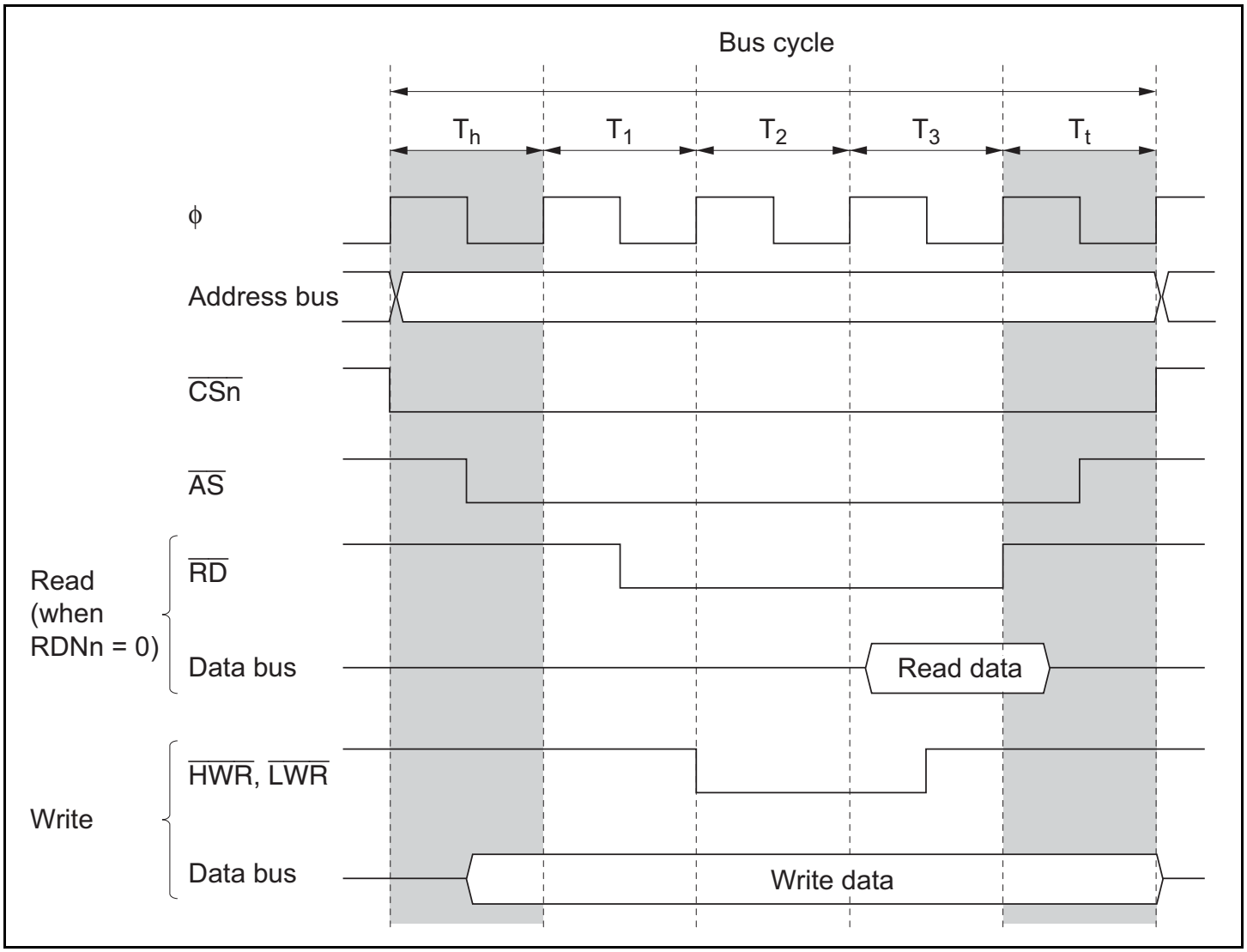


Figure 6.20 Example of Timing when Chip Select Assertion Period Is Extended

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

6.6 Address/Data Multiplexed I/O Interface

If areas 6 and 7 of the external address space are specified as address/data multiplexed I/O space in this LSI, the address/data multiplexed I/O interfacing can be performed. In the address/data multiplexed I/O interface, peripheral LSIs that require address/data multiplexing can be connected directly to this LSI.

6.6.1 Setting Address/Data Multiplexed I/O Space

In the address/data multiplexed I/O interface, areas 6 and 7 are designated as the address/data multiplexed I/O space by setting the MPXE bit in MPXCR to 1.

6.6.2 Address/Data Multiplexing

With the address/data multiplexed I/O space, the data bus and address bus are multiplexed. Table 6.4 shows the relation between the bus width and corresponding address output.

Table 6.4 Multiplexed Address/Data

Bus Width	Cycle	Data Pins															
		PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
8 bits	Address	A7	A6	A5	A4	A3	A2	A1	A0	—	—	—	—	—	—	—	—
	Data	D15	D14	D13	D12	D11	D10	D9	D8	—	—	—	—	—	—	—	—
16 bits	Address	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
	Data	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

6.6.3 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 ABW7 and ABW6 bits in ABWCRA. For the 8-bit access space, D15 to D8 are valid for both address and data. For the 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 6.5.1, Data Size and Data Alignment.

6.6.4 Address Hold Signal

In the address/data multiplexed I/O space, a hold signal ($\overline{AH}$) that indicates the timing for latching the address is output. The $\overline{AH}$ output pin is multiplexed with the $\overline{AS}$ output pin. When the external address space is specified as the address/data multiplexed I/O space, the multiplexed pin functions as the $\overline{AH}$ output pin. Note however that the multiplexed pin will function as the $\overline{AS}$ output pin until the address/data multiplexed I/O space is specified.

6.6.5 Basic Timing

The bus cycle in the address/data multiplexed I/O interface consists of an address cycle and a data cycle. The data cycle is based on the basic bus interface timing specified by ABWCR, ASTCR, WTCRAH, RDNCR, and CSACR.

(1) 8-Bit, 2-State Data Access Space

Figure 6.21 shows the bus timing for an 8-bit, 2-state data access space. When an 8-bit access space is accessed, the upper halves (D15 to D8) of both the address bus and data bus are used. Wait states cannot be inserted in the data cycle.

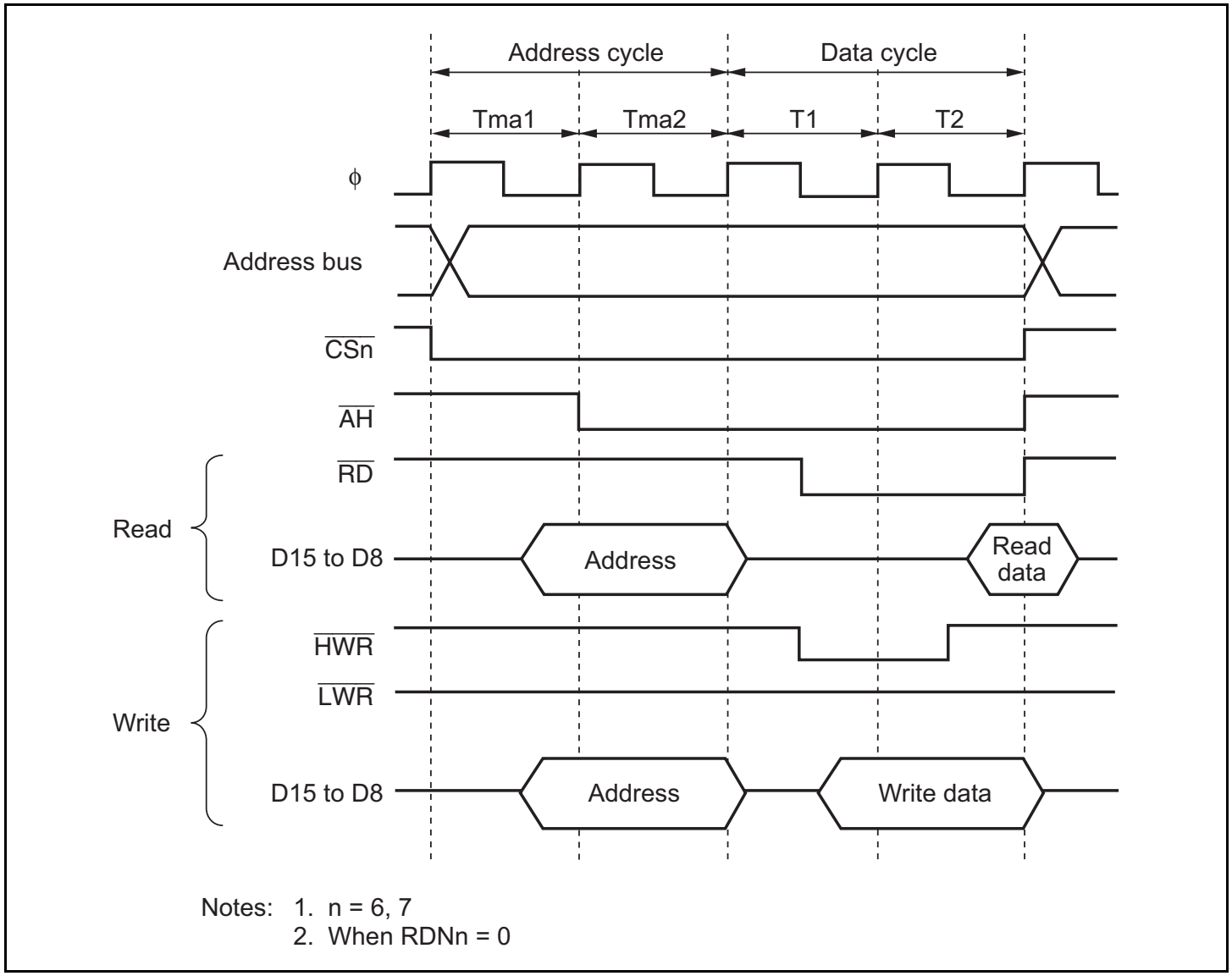


Figure 6.21 Bus Timing for 8-Bit, 2-State Data Access Space

(2) 8-Bit, 3-State Data Access Space

Figure 6.22 shows the bus timing for an 8-bit, 3-state data access space. When an 8-bit access space is accessed, the upper halves (D15 to D8) of both the address bus and data bus are used. Wait states can be inserted in the data cycle.

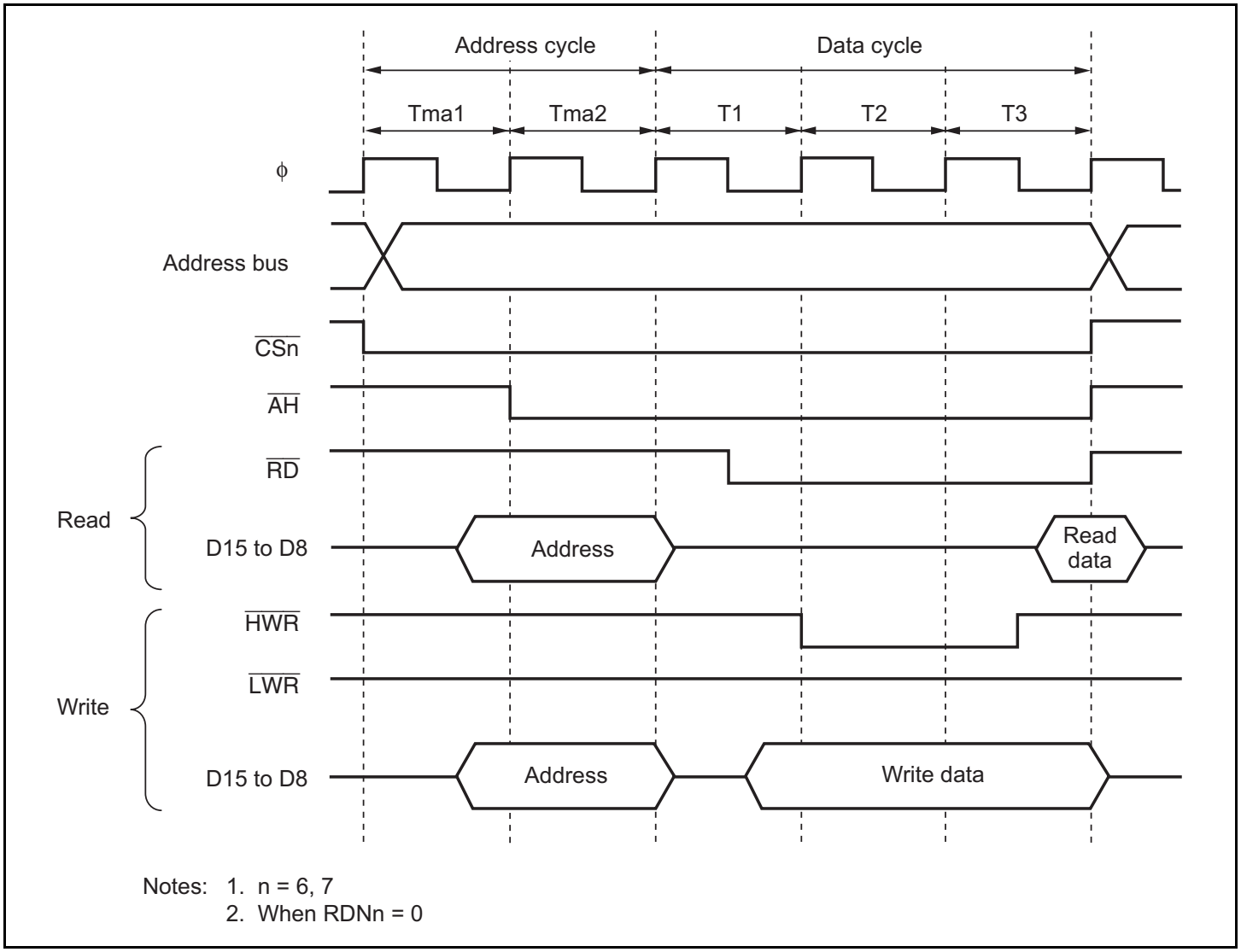


Figure 6.22 Bus Timing for 8-Bit, 3-State Data Access Space

(3) 16-Bit, 2-State Data Access Space

Figures 6.23 to 6.25 show bus timings for a 16-bit, 2-state data access space. When a 16-bit access space is accessed, the entire address bus (D15 to D0) is used for all addresses, and the upper half (D15 to D8) of the data bus is used for even addresses and the lower half (D7 to D0) of the data bus is used for odd addresses. Wait states cannot be inserted in the data cycle.

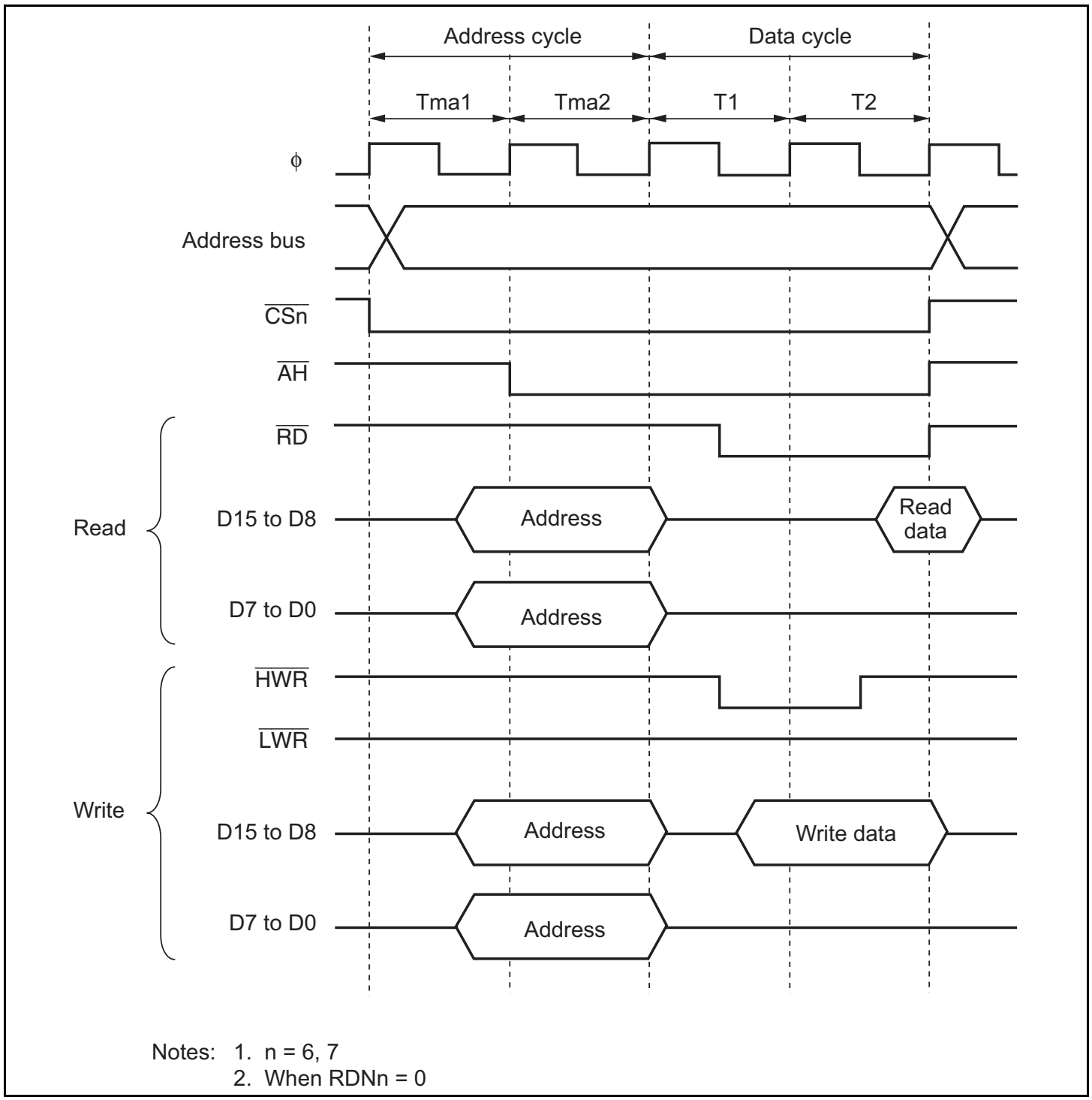


Figure 6.23 Bus Timing for 16-Bit, 2-State Data Access Space (Even Address Byte Access)

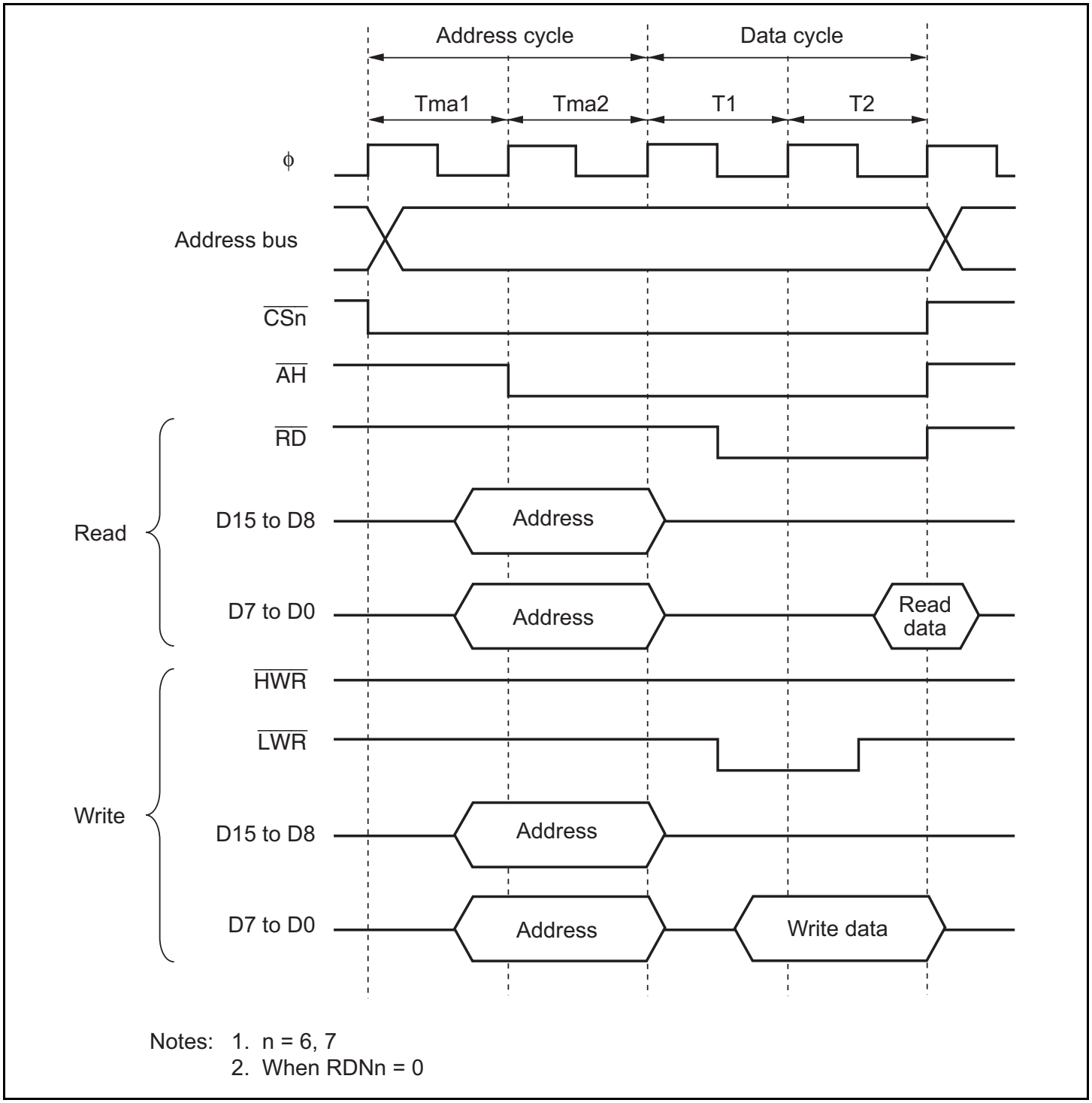


Figure 6.24 Bus Timing for 16-Bit, 2-State Data Access Space (Odd Address Byte Access)

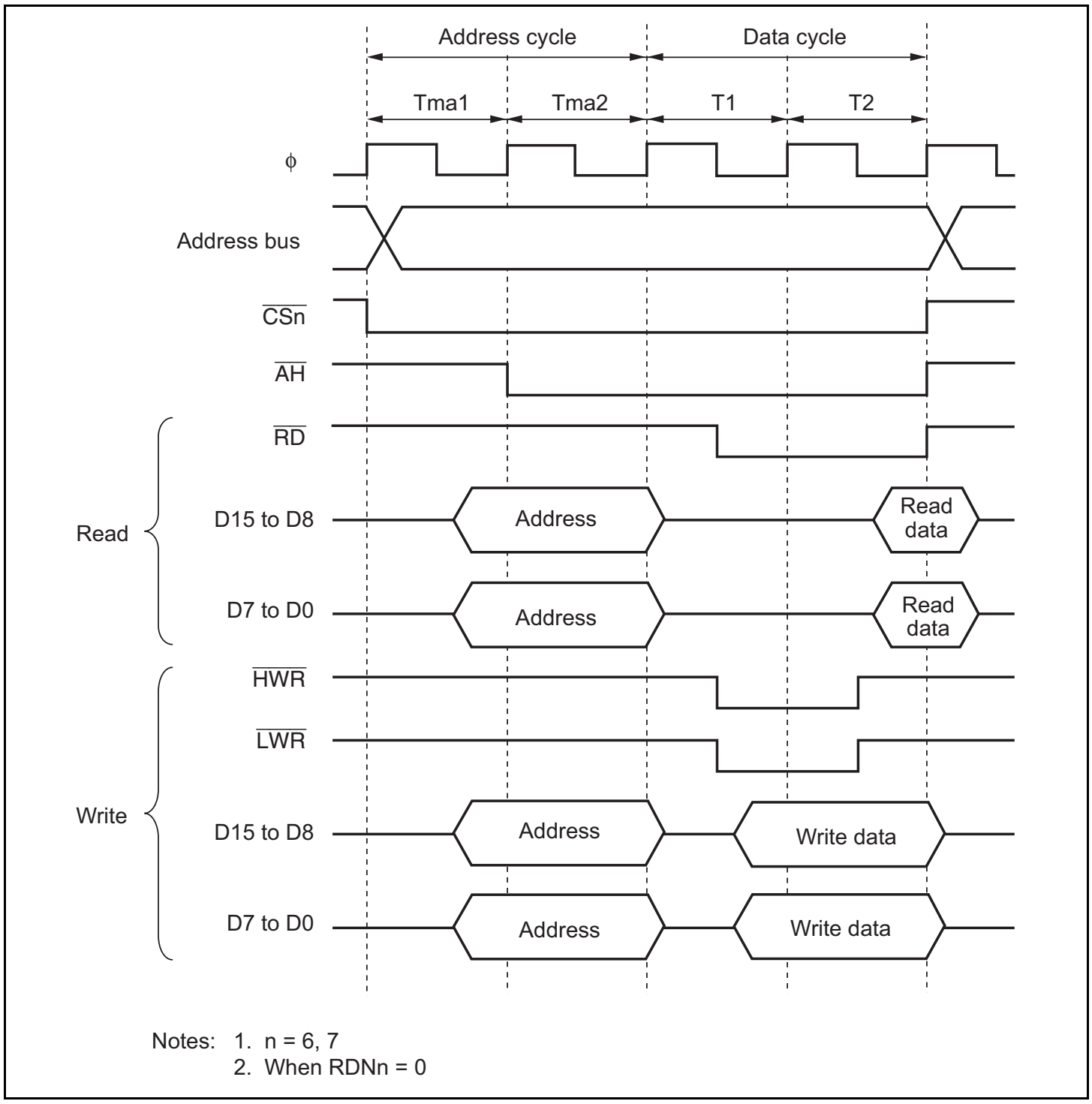


Figure 6.25 Bus Timing for 16-Bit, 2-State Data Access Space (Word Access)

(4) 16-Bit, 3-State Data Access Space

Figures 6.26 to 6.28 show bus timings for a 16-bit, 3-state data access space. When a 16-bit access space is accessed, the entire address bus (D15 to D0) is used for all addresses, and the upper half (D15 to D8) of the data bus is used for even addresses and the lower half (D7 to D0) of the data bus is used for odd addresses. Wait states can be inserted in the data cycle.

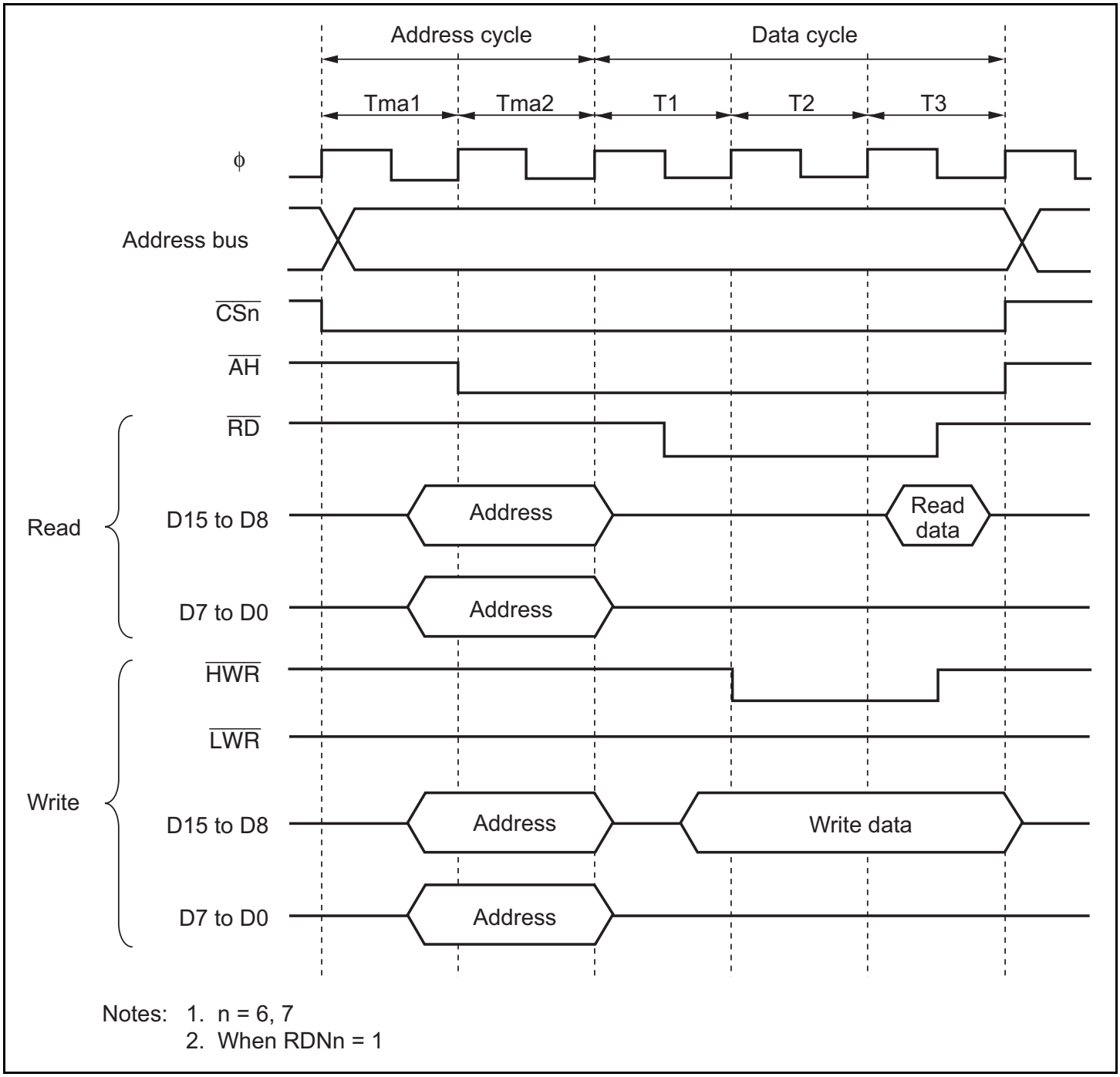


Figure 6.26 Bus Timing for 16-Bit, 3-State Data Access Space (Even Address Byte Access)

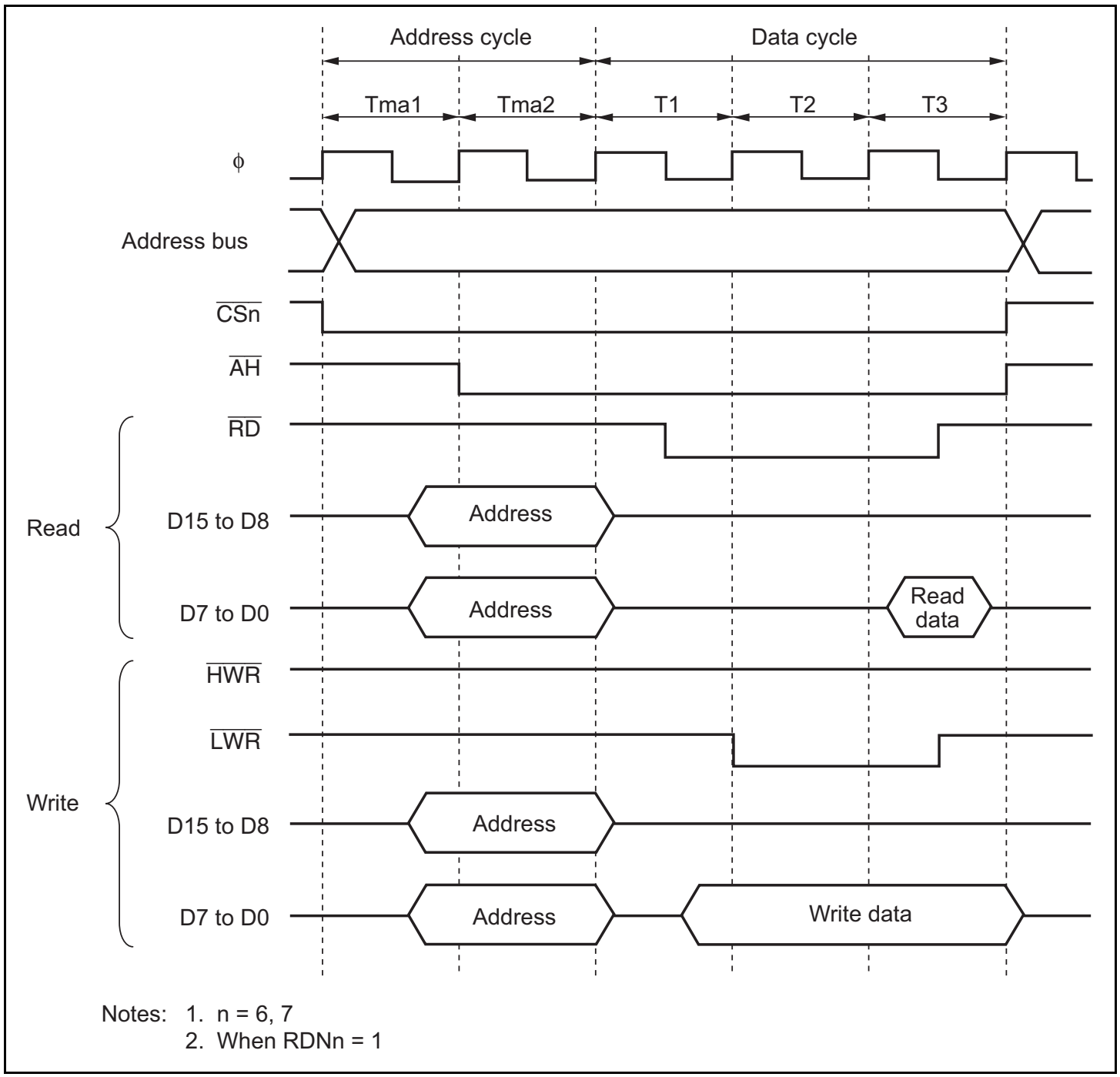


Figure 6.27 Bus Timing for 16-Bit, 3-State Data Access Space (Odd Address Byte Access)

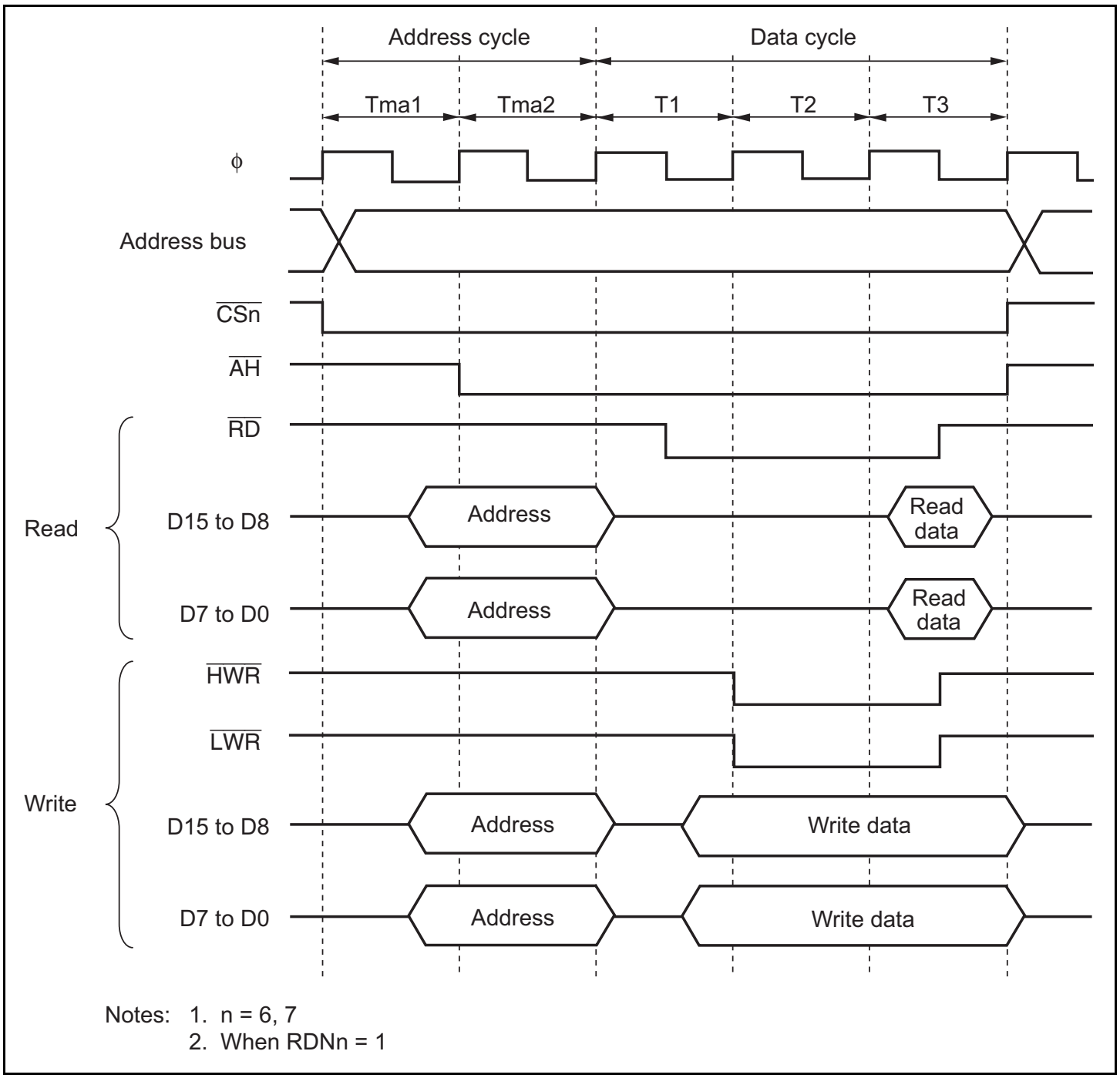


Figure 6.28 Bus Timing for 16-Bit, 3-State Data Access Space (Word Access)

6.6.6 Wait Control

(1) Address Cycle

A single address wait cycle Tmaw can be inserted between Tma1 and Tma2 cycles by setting the ADDEX bit in MPXCR to 1. Figure 6.29 shows the access timing when the address cycle is three cycles.

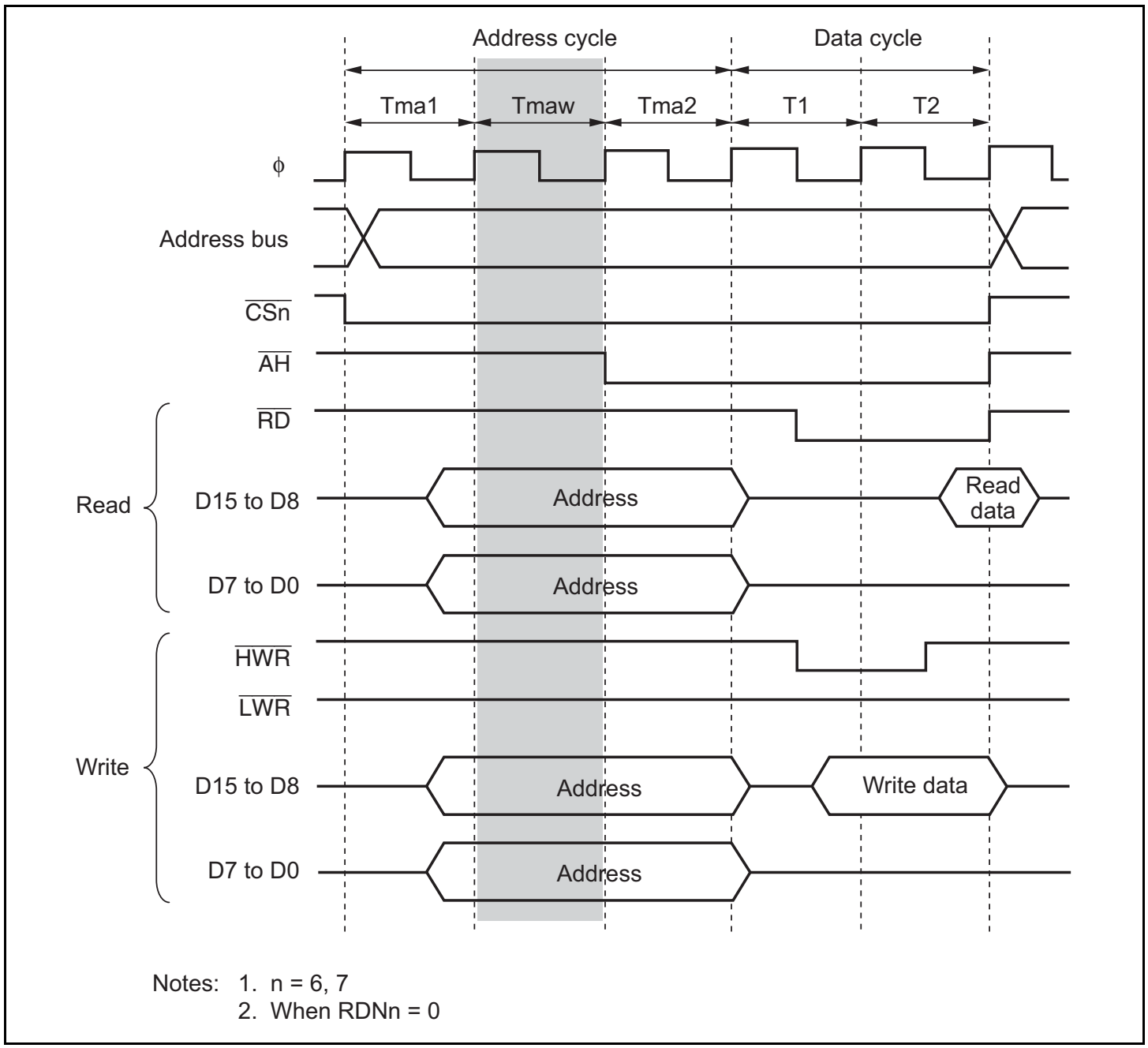


Figure 6.29 Example of Access Timing with Address Wait

(2) Data Cycle

In the data cycle, program wait insertion and pin wait insertion by the $\overline{\text{WAIT}}$ pin are enabled in the same way as in the basic bus interface. For details, refer to section 6.5.4, Wait Control. Wait control settings do not affect the address cycles.

6.6.7 Read Strobe ($\overline{\text{RD}}$) Timing

In the address/data multiplexed I/O interface, the read strobe timing of data cycles can be modified in the same way as in the basic bus interface. For details, refer to section 6.5.5, Read Strobe ($\overline{\text{RD}}$) Timing. Figure 6.30 shows an example when the read strobe timing is modified.

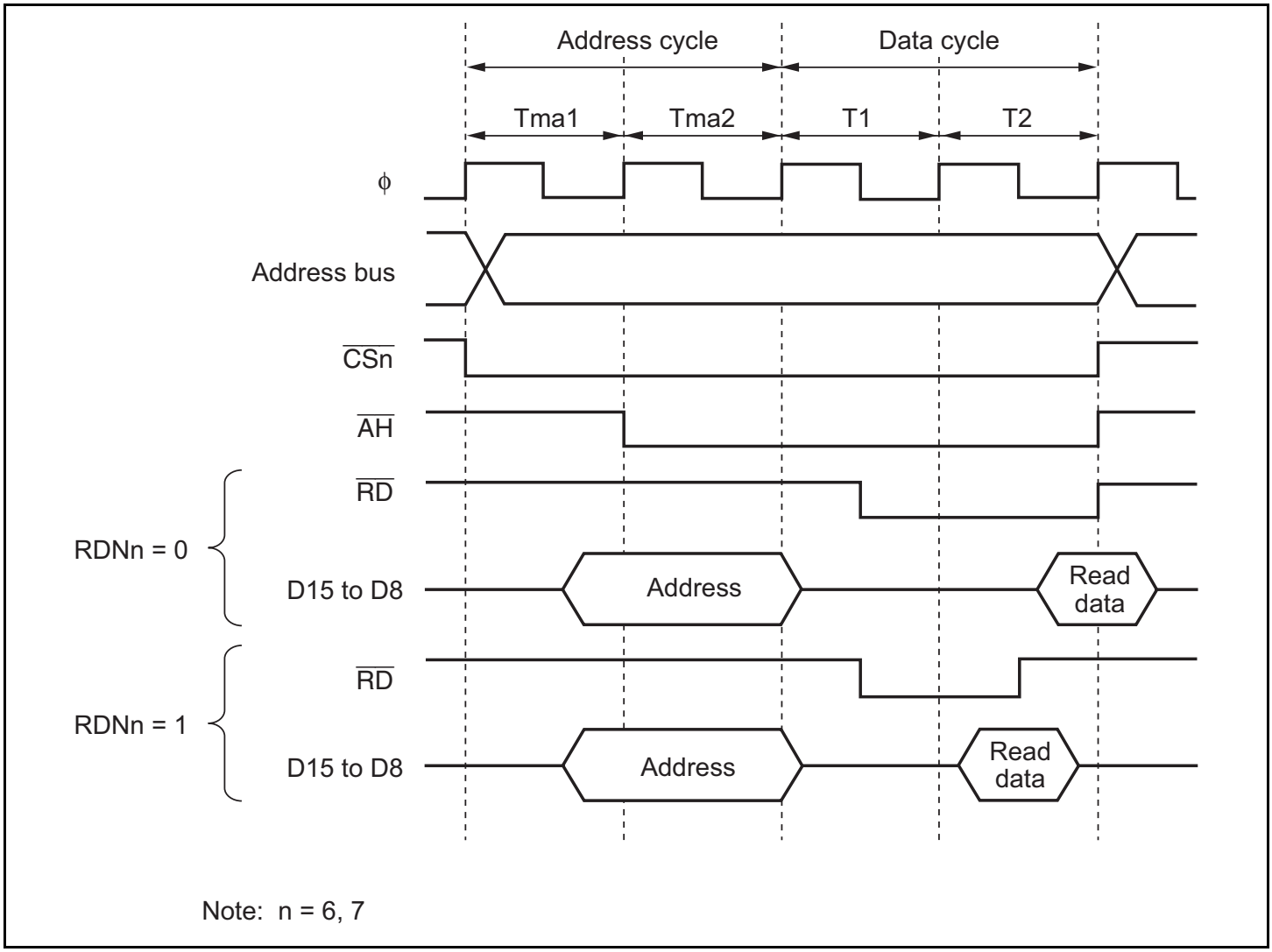


Figure 6.30 Example of Read Strobe Timing

6.6.8 Extension of Chip Select ($\overline{CS}$) Assertion Period in Data Cycle

In the address/data multiplexed I/O interface, extension cycles can be inserted before and after the data cycle. For details, see section 6.5.6, Extension of Chip Select ($\overline{CS}$) Assertion Period. Figure 6.31 shows an example of the timing when the chip select assertion period is extended in the data cycle.

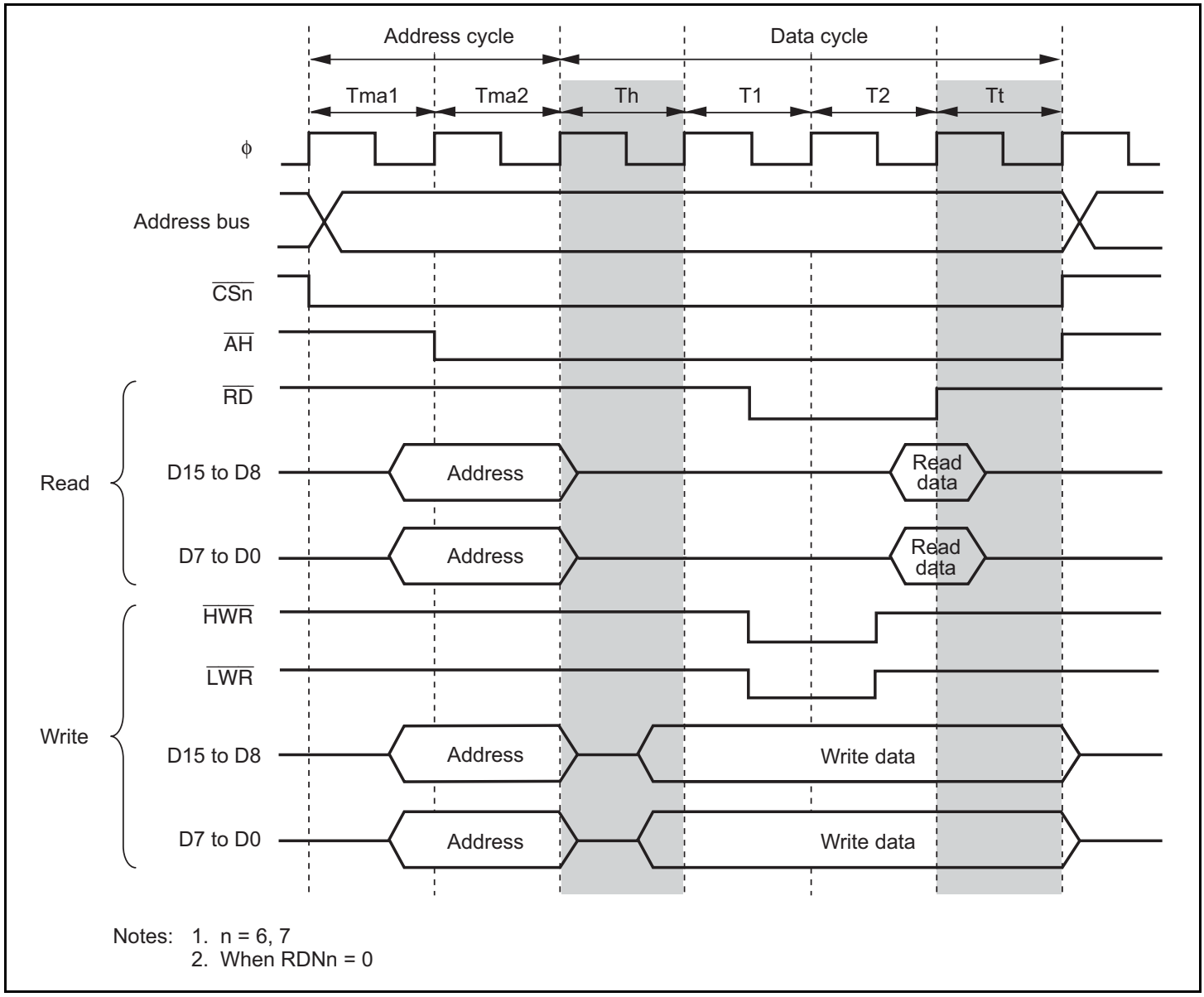


Figure 6.31 Example of Timing when Chip Select Assertion Period Is Extended in Data Cycle

When consecutively reading from the same area connected to a peripheral LSI whose output floating time is long, data outputs from the peripheral LSI may conflict with address outputs from this LSI. The data conflict can be avoided by inserting the $\overline{CS}$ assertion period extension cycle after the access cycle. Figure 6.32 shows an example of the operation. In the figure, both bus cycles A and B are read access cycles to the same area which is address/data multiplexed I/O space. (a) shows an example of conflict occurring between data outputs from the peripheral LSI whose output floating time is long and address outputs from this LSI because the $\overline{CS}$ assertion period extension cycle is not inserted. (b) shows an example of the data conflict being avoided by inserting the $\overline{CS}$ assertion period extension cycle.

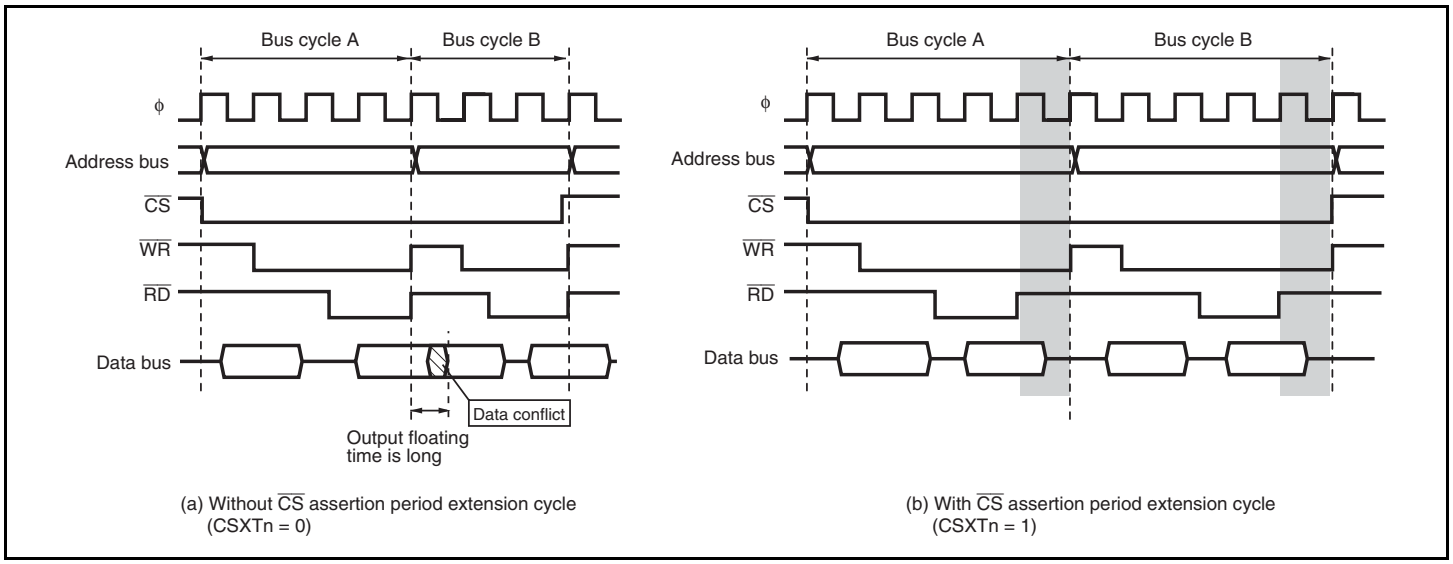


Figure 6.32 Consecutive Read Accesses to Same Area (Address/Data Multiplexed I/O Space)

6.7 DRAM Interface

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

6.7.1 Setting DRAM Space

Areas 2 to 5 are designated as DRAM space by setting bits RMTS2 to RMTS0 in DRAMCR. The relation between the settings of bits RMTS2 to RMTS0 and DRAM space is shown in table 6.5. Possible DRAM space settings are: one area (area 2), two areas (areas 2 and 3), four areas (areas 2 to 5), and continuous area (areas 2 to 5).

Table 6.5 Relation between Settings of Bits RMTS2 to RMTS0 and DRAM Space

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

Note: * Reserved (setting prohibited) in the H8S/2456 Group and H8S/2454 Group.

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

6.7.2 Address Multiplexing

With DRAM space, the row address and column address are multiplexed. In address multiplexing, the size of the shift of the row address is selected with bits MXC2 to MXC0 in DRAMCR. Table 6.6 shows the relation between the settings of MXC2 to MXC0 and the shift size.

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

Table 6.6 Relation between Settings of Bits MXC2 to MXC0 and Address Multiplexing

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

Legend:
×: Don't care.

6.7.3 Data Bus

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

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

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

6.7.4 Pins Used for DRAM Interface

Table 6.7 shows the pins used for DRAM interfacing and their functions. Since the $\overline{CS2}$ to $\overline{CS5}$ pins are in the input state after a reset, set the corresponding DDR to 1 when $\overline{RAS2}$ to $\overline{RAS5}$ signals are output.

Table 6.7 DRAM Interface Pins

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

6.7.5 Basic Timing

Figure 6.33 shows the basic access timing for DRAM space.

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

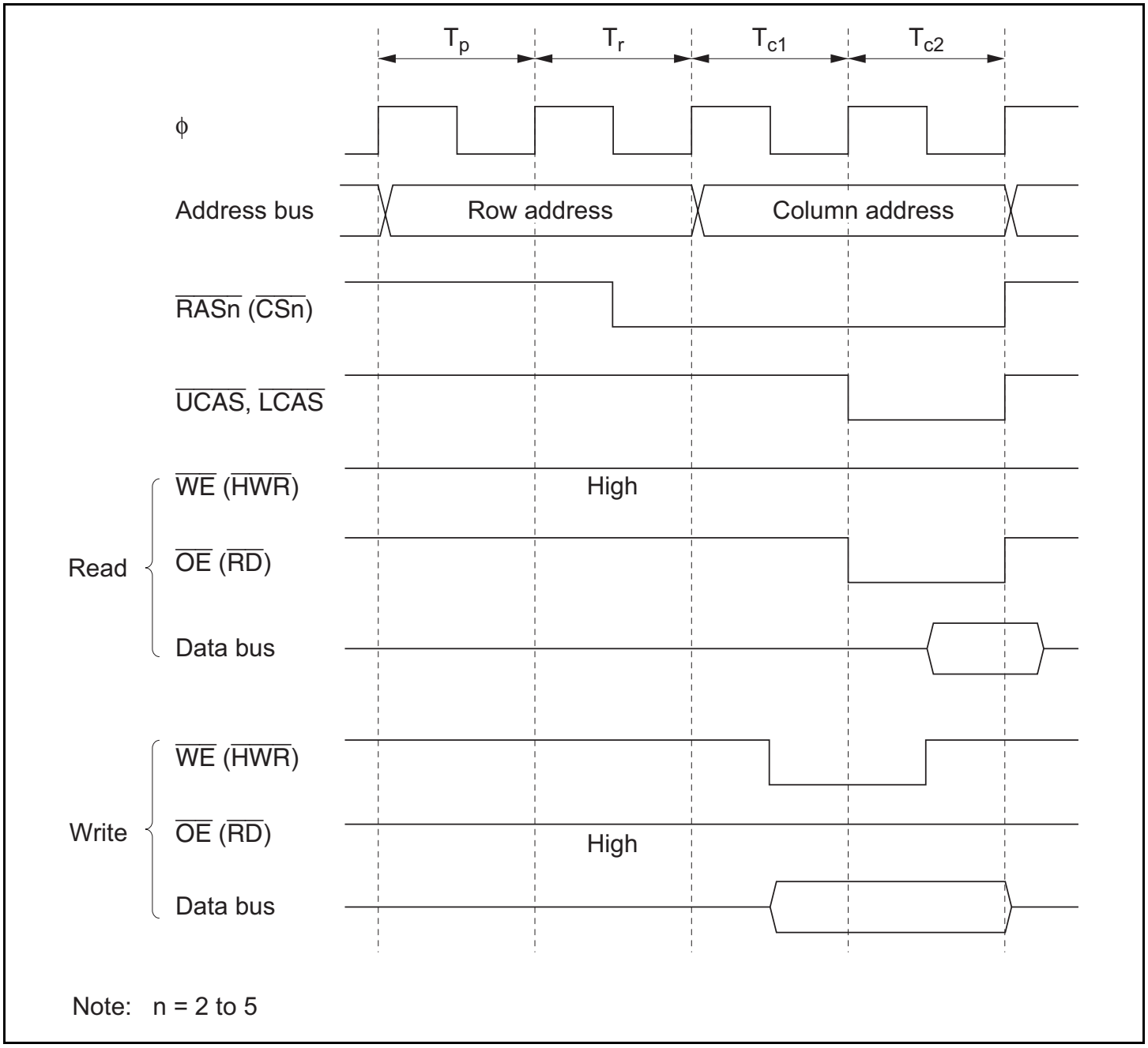


Figure 6.33 DRAM Basic Access Timing (RAST = 0, CAST = 0)

When DRAM space is accessed, the $\overline{RD}$ signal is output as the $\overline{OE}$ signal for DRAM. When connecting DRAM provided with an EDO page mode, the $\overline{OE}$ signal should be connected to the ($\overline{OE}$) pin of the DRAM. Setting the OEE bit to 1 in DRAMCR enables the $\overline{OE}$ signal for DRAM space to be output from a dedicated $\overline{OE}$ pin. In this case, the $\overline{OE}$ signal for DRAM space is output from both the $\overline{RD}$ pin and the ($\overline{OE}$) pin, but in external read cycles for other than DRAM space, the signal is output only from the $\overline{RD}$ pin.

6.7.6 Column Address Output Cycle Control

The column address output cycle can be changed from 2 states to 3 states by setting the CAST bit to 1 in DRAMCR. Use the setting that gives the optimum specification values ($\overline{CAS}$ pulse width, etc.) according to the DRAM connected and the operating frequency of this LSI. Figure 6.34 shows an example of the timing when a 3-state column address output cycle is selected.

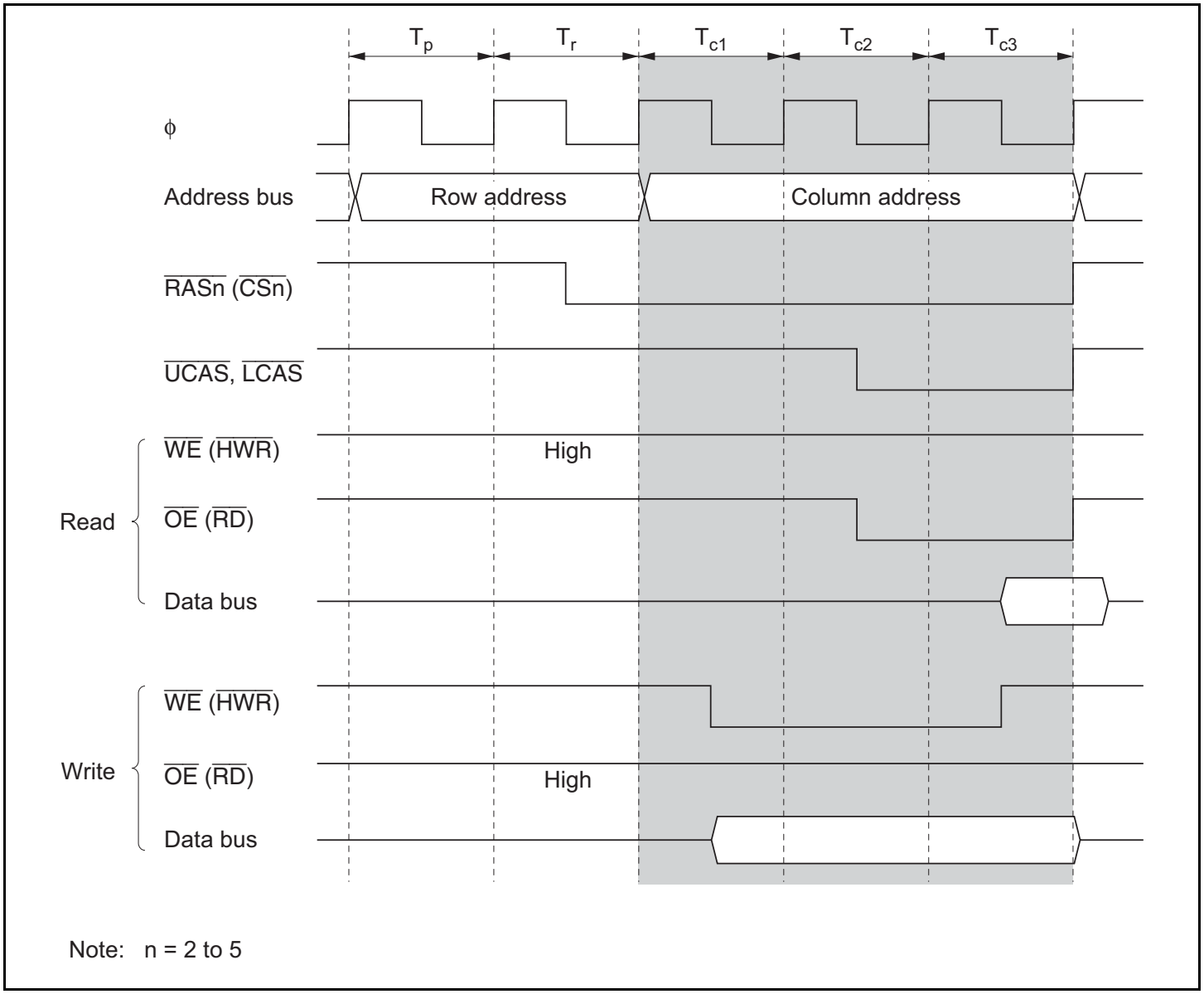


Figure 6.34 Example of Access Timing with 3-State Column Address Output Cycle (RAST = 0)

6.7.7 Row Address Output State Control

If the RAST bit is set to 1 in DRAMCR, the $\overline{\text{RAS}}$ signal goes low from the beginning of the T_r state, and the row address hold time and DRAM read access time are changed relative to the fall of the $\overline{\text{RAS}}$ signal. Use the optimum setting according to the DRAM connected and the operating frequency of this LSI. Figure 6.35 shows an example of the timing when the $\overline{\text{RAS}}$ signal goes low from the beginning of the T_r state.

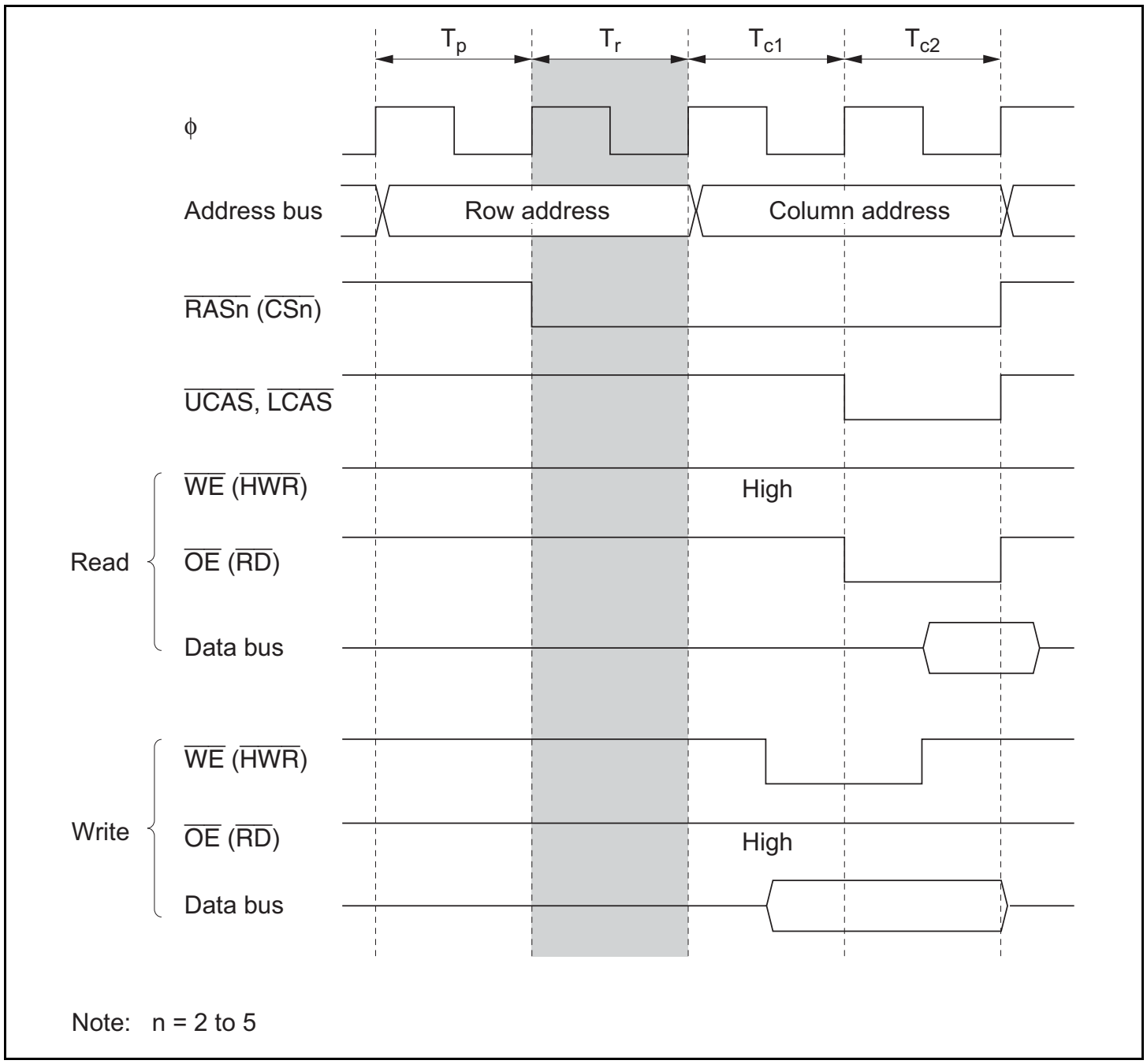


Figure 6.35 Example of Access Timing when RAS Signal Goes Low from Beginning of T_r State (CAST = 0)

If a row address hold time or read access time is necessary, making a setting in bits RCD1 and RCD0 in DRACCR allows from one to three T_{rw} states, in which row address output is maintained, to be inserted between the T_r cycle, in which the $\overline{RAS}$ signal goes low, and the T_{c1} cycle, in which the column address is output. Use the setting that gives the optimum row address signal hold time relative to the falling edge of the $\overline{RAS}$ signal according to the DRAM connected and the operating frequency of this LSI. Figure 6.36 shows an example of the timing when one T_{rw} state is set.

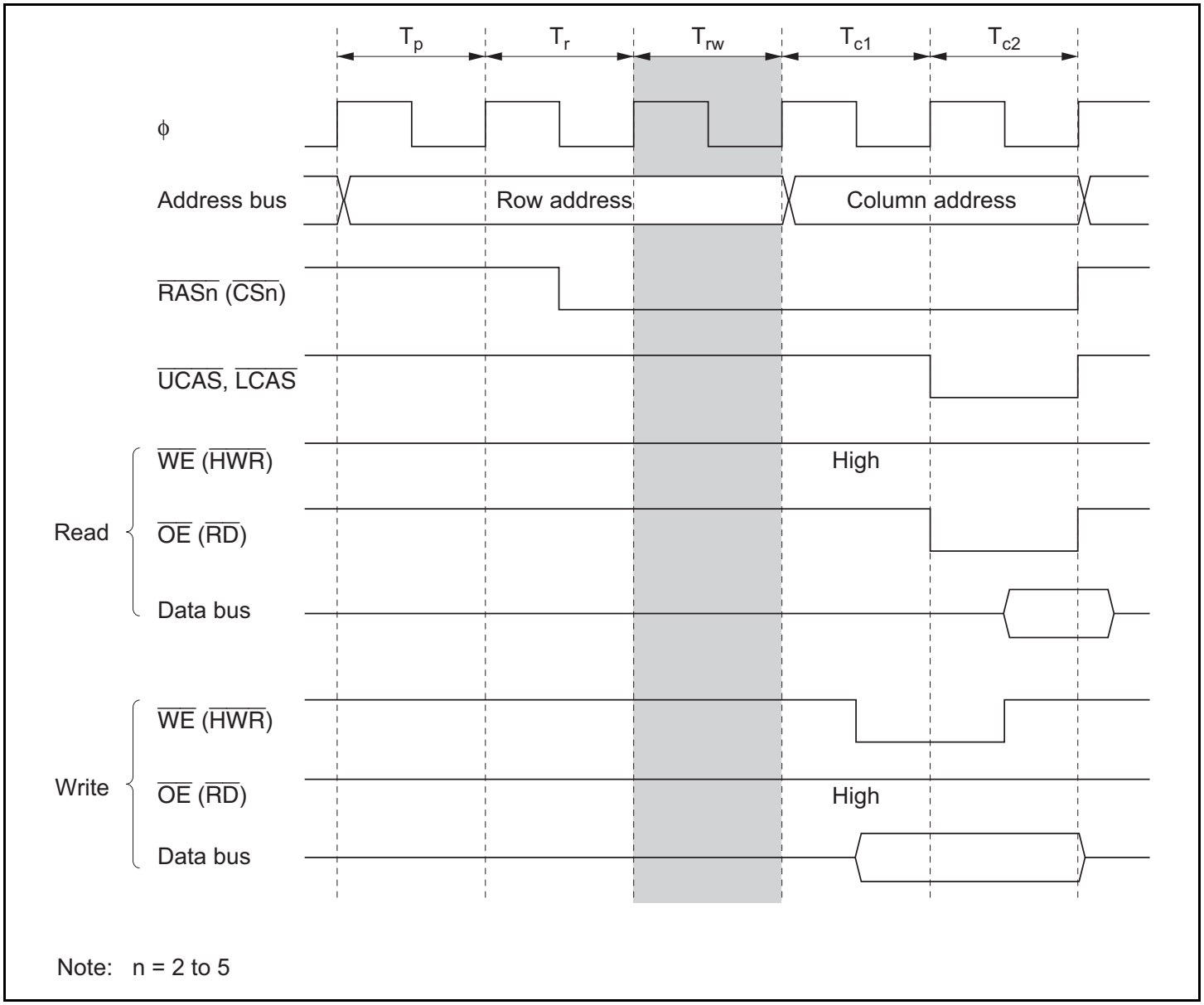


Figure 6.36 Example of Timing with One Row Address Output Maintenance State (RAST = 0, CAST = 0)

6.7.8 Precharge State Control

When DRAM is accessed, a $\overline{RAS}$ precharge time must be secured. With this LSI, one T_p state is always inserted when DRAM space is accessed. From one to four T_p states can be selected by setting bits TPC1 and TPC0 in DRACCR. Set the optimum number of T_p cycles according to the DRAM connected and the operating frequency of this LSI. Figure 6.37 shows the timing when two T_p states are inserted. The setting of bits TPC1 and TPC0 is also valid for T_p states in refresh cycles.

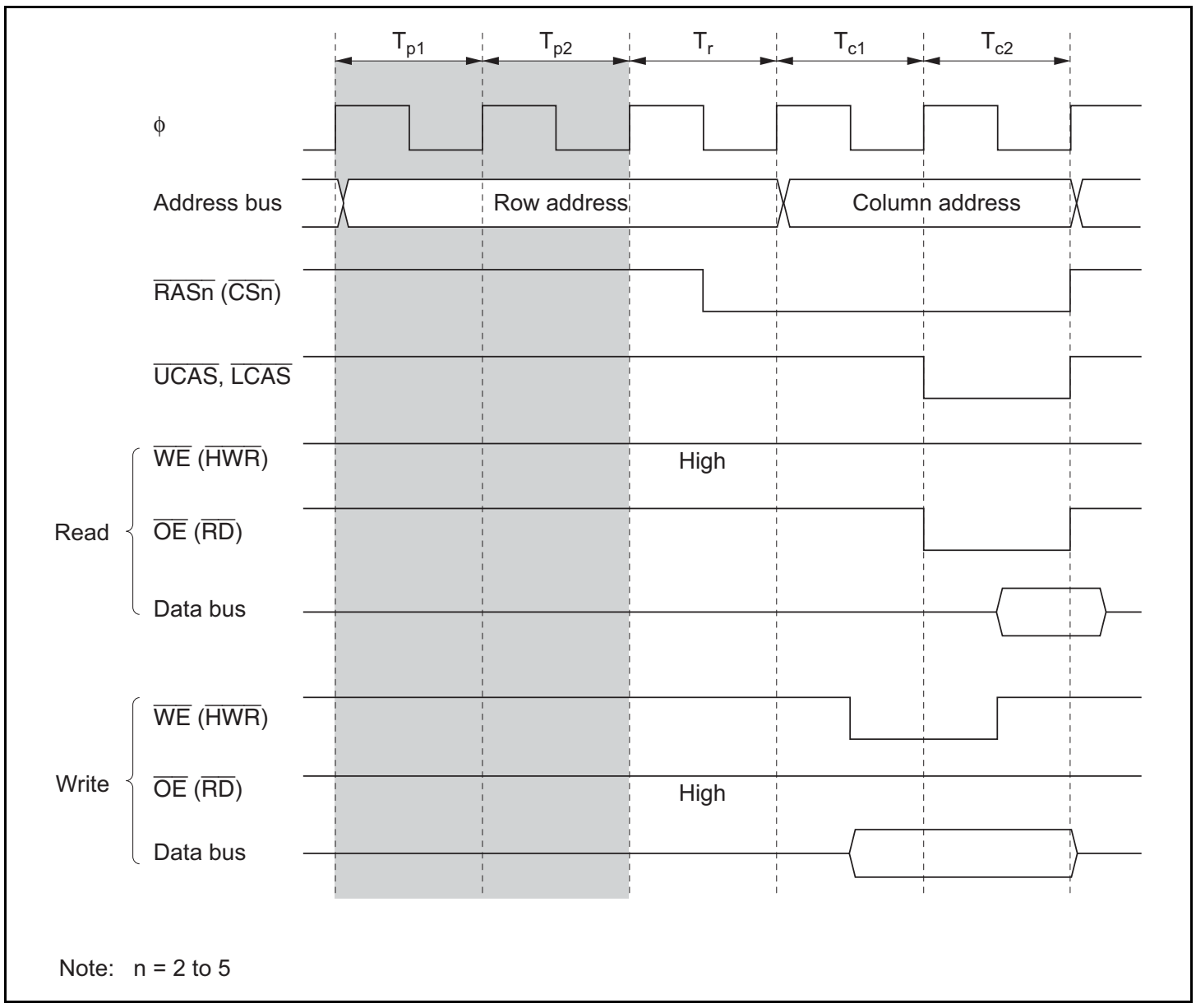


Figure 6.37 Example of Timing with Two-State Precharge Cycle (RAST = 0, CAST = 0)

6.7.9 Wait Control

There are two ways of inserting wait states in a DRAM access cycle: program wait insertion and pin wait insertion using the $\overline{\text{WAIT}}$ pin.

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

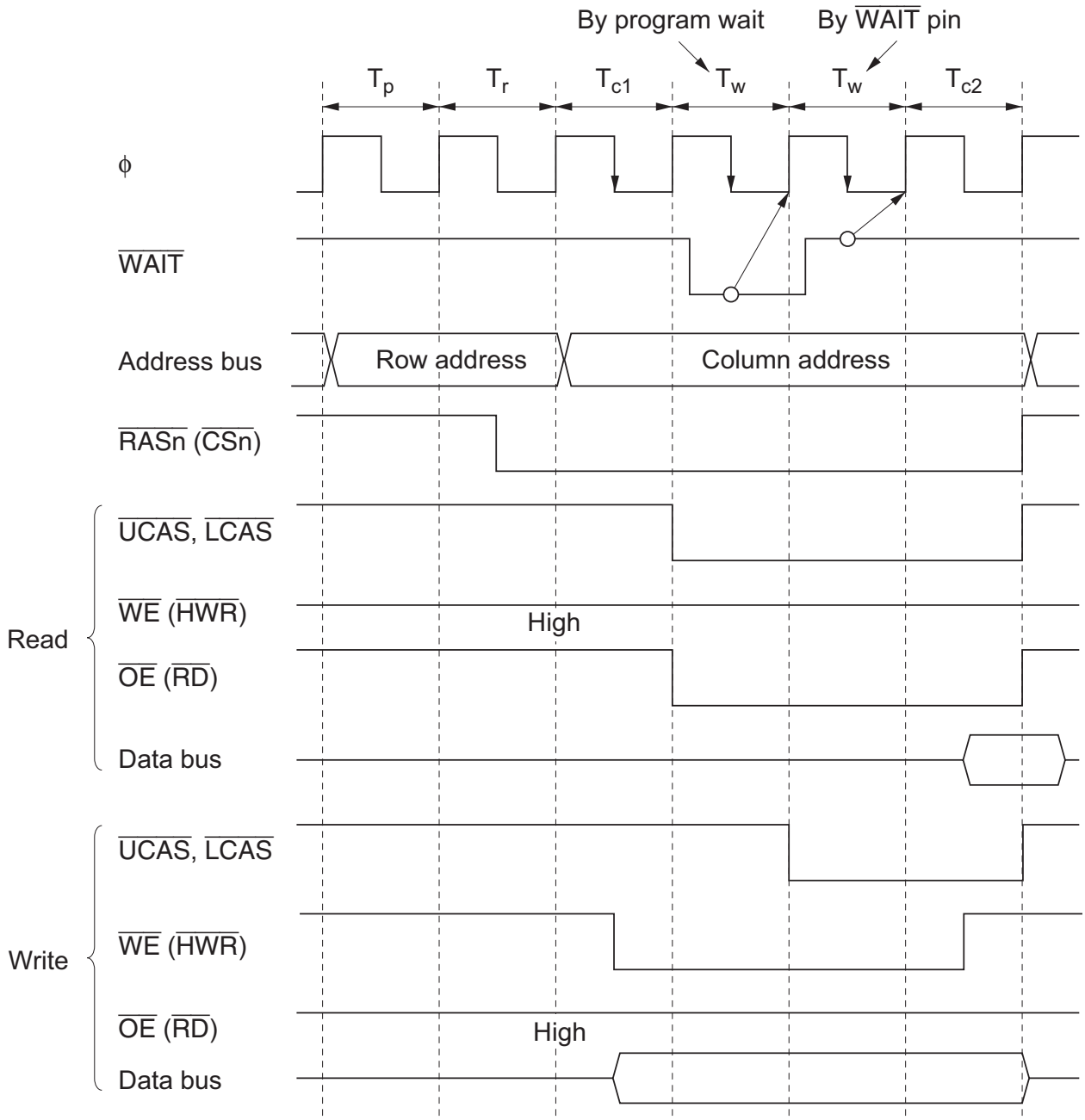
(1) Program Wait Insertion

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

(2) Pin Wait Insertion

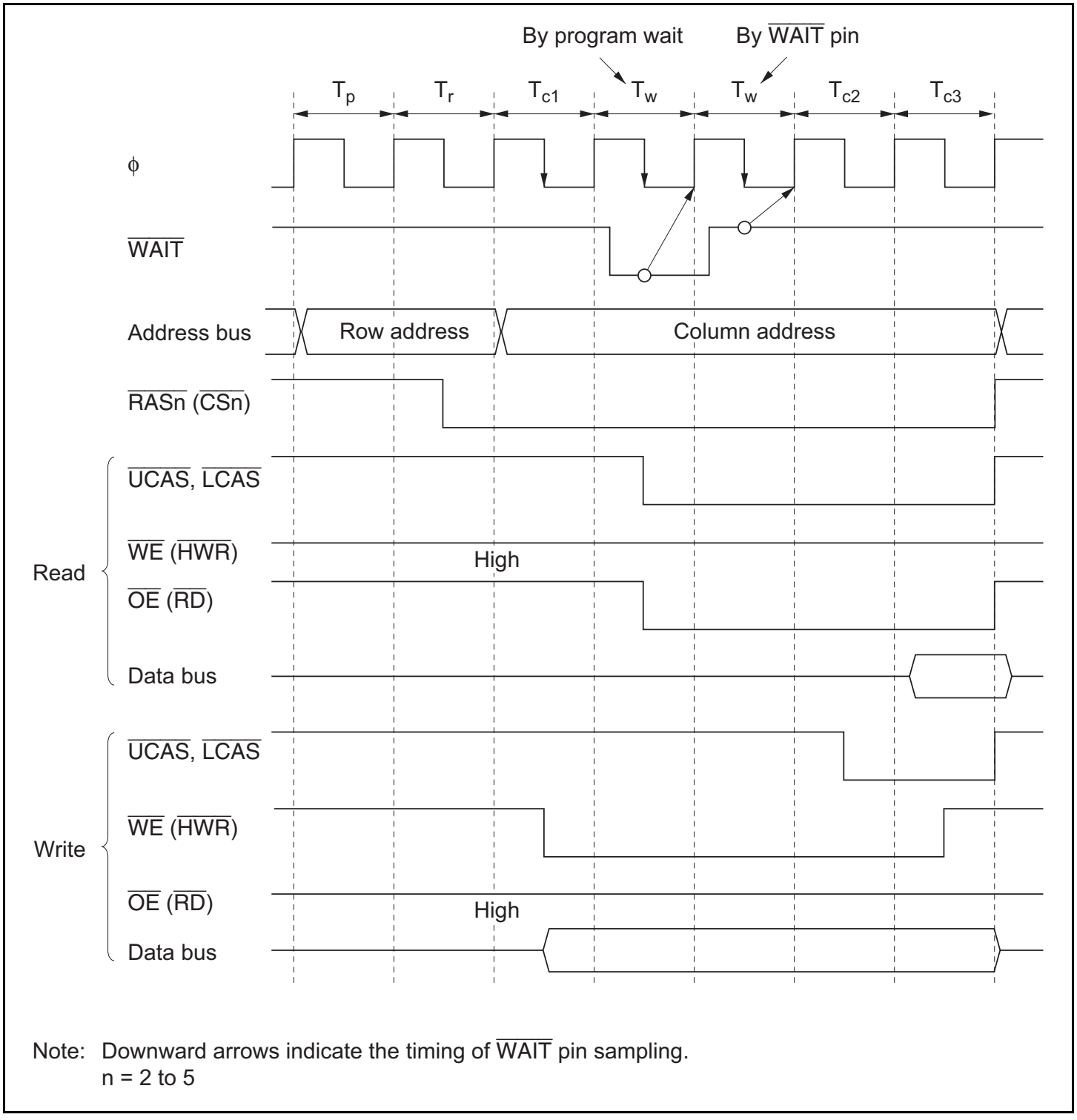
When the WAITE bit in BCR is set to 1 and the ASTCR bit is set to 1, wait input by means of the $\overline{\text{WAIT}}$ pin is enabled. When DRAM space is accessed in this state, a program wait (T_w) is first inserted. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_{c1}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

Figures 6.38 and 6.39 show examples of wait cycle insertion timing in the case of 2-state and 3-state column address output cycles.



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

Figure 6.38 Example of Wait State Insertion Timing (2-State Column Address Output)



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

Figure 6.39 Example of Wait State Insertion Timing (3-State Column Address Output)

6.7.10 Byte Access Control

When DRAM with a ×16-bit configuration is connected, the 2-CAS access method is used for the control signals needed for byte access. Figure 6.40 shows the control timing for 2-CAS access, and figure 6.41 shows an example of 2-CAS DRAM connection.

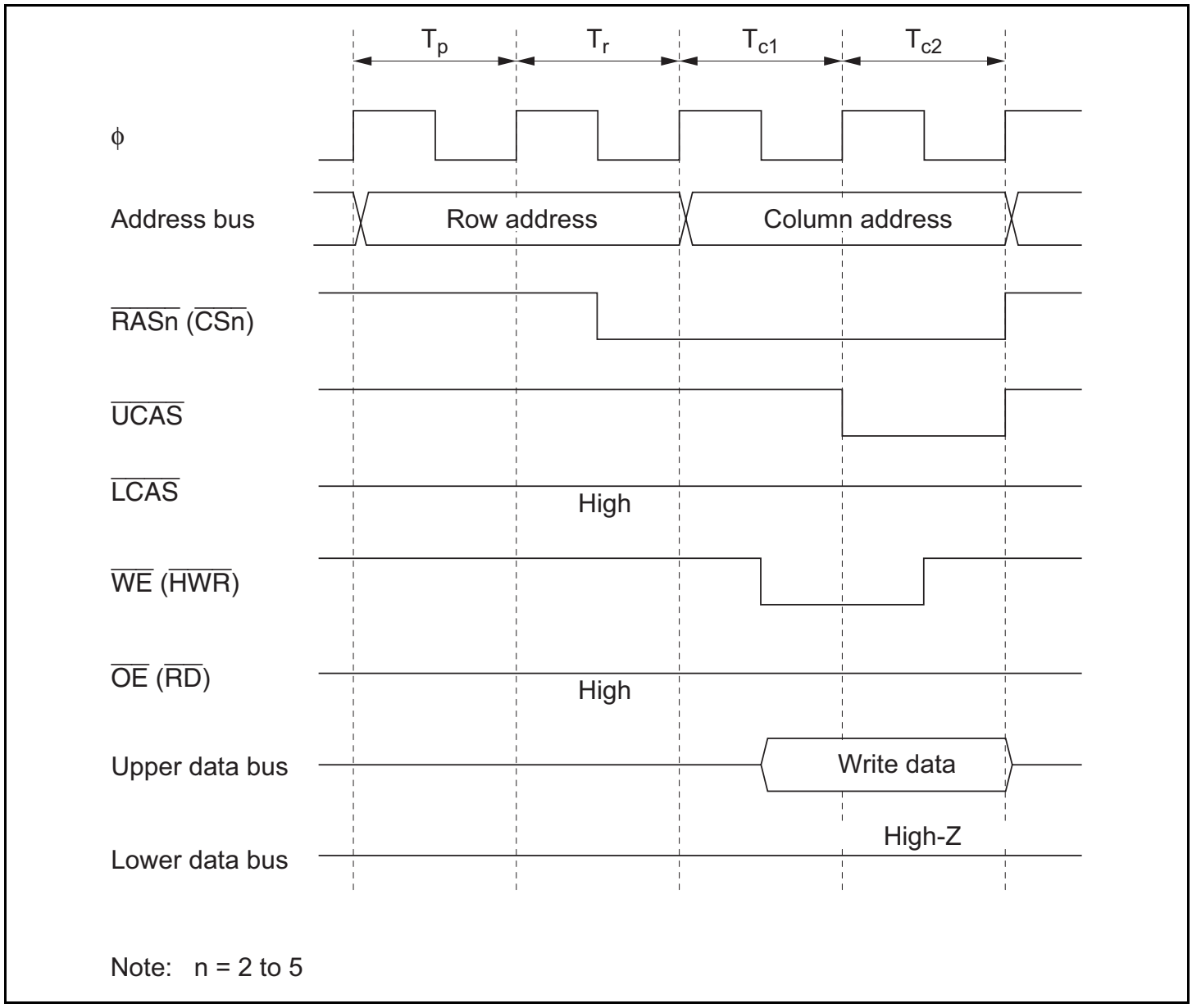


Figure 6.40 2-CAS Control Timing
(Upper Byte Write Access: RAST = 0, CAST = 0)

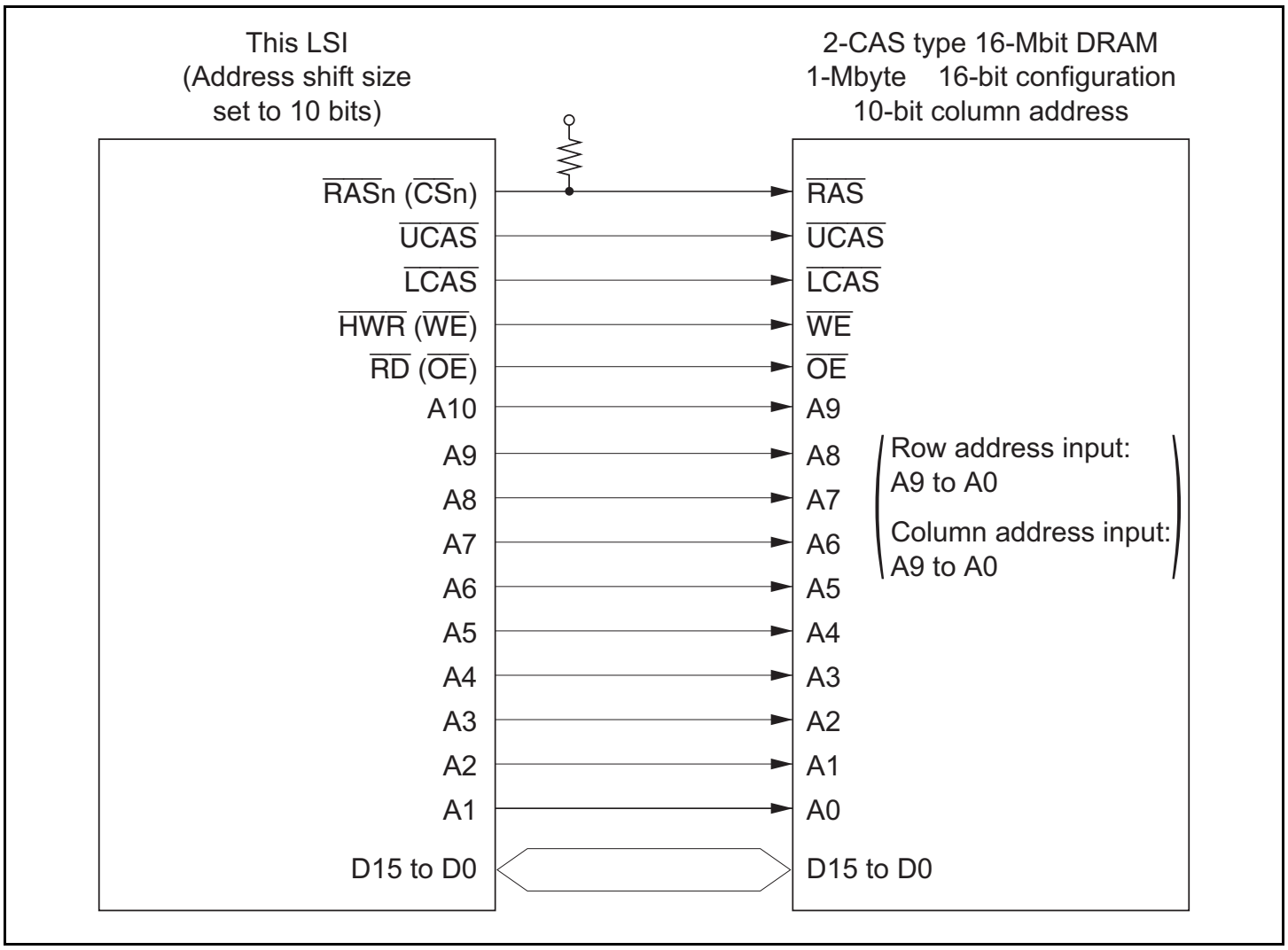


Figure 6.41 Example of 2-CAS DRAM Connection

6.7.11 Burst Operation

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

(1) Burst Access (Fast Page Mode)

Figures 6.42 and 6.43 show the operation timing for burst access. When there are consecutive access cycles for DRAM space, the $\overline{\text{CAS}}$ signal and column address output cycles (two states) continue as long as the row address is the same for consecutive access cycles. The row address used for the comparison is set with bits MXC2 to MXC0 in DRAMCR.

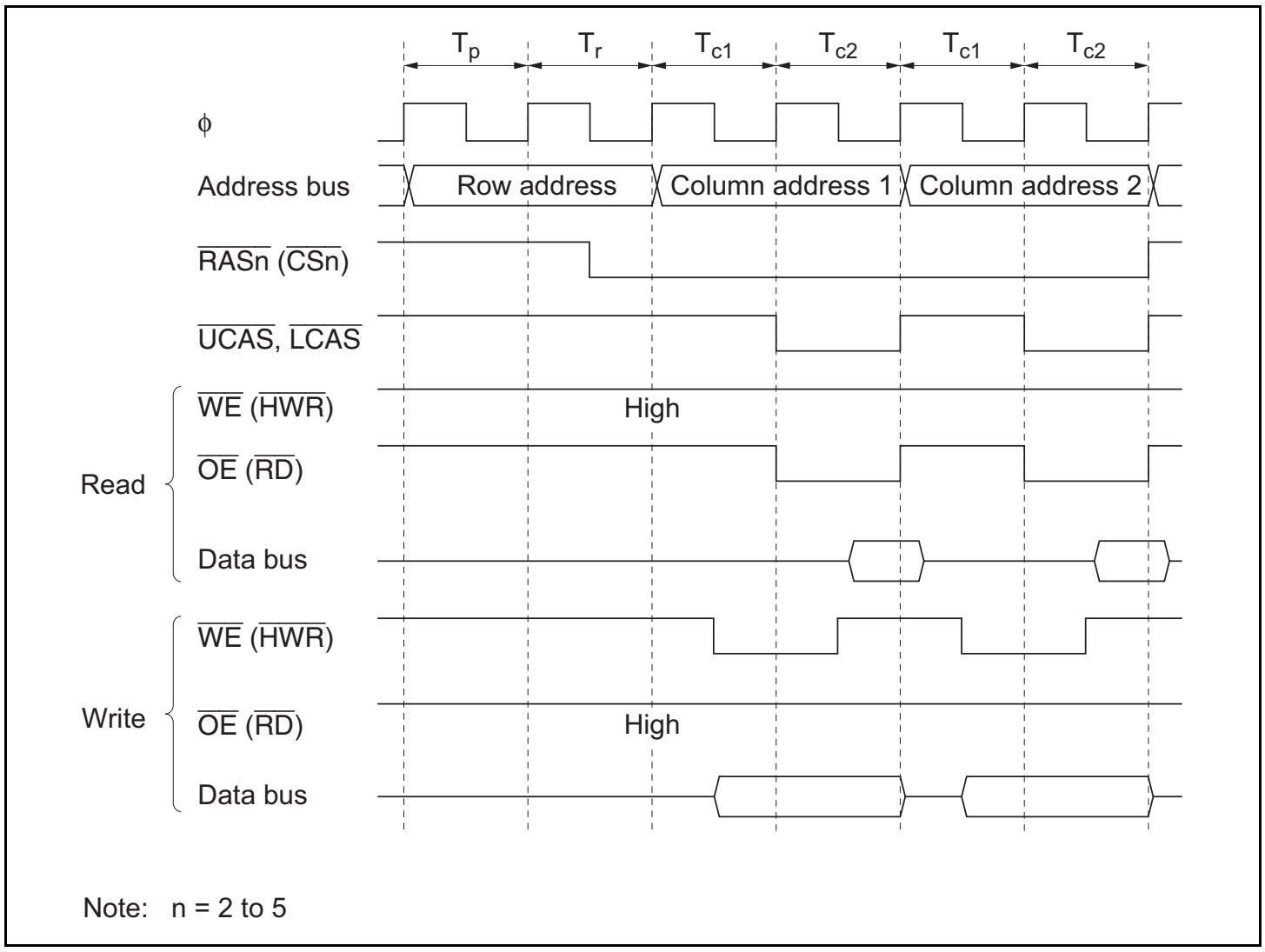


Figure 6.42 Operation Timing in Fast Page Mode (RAST = 0, CAST = 0)

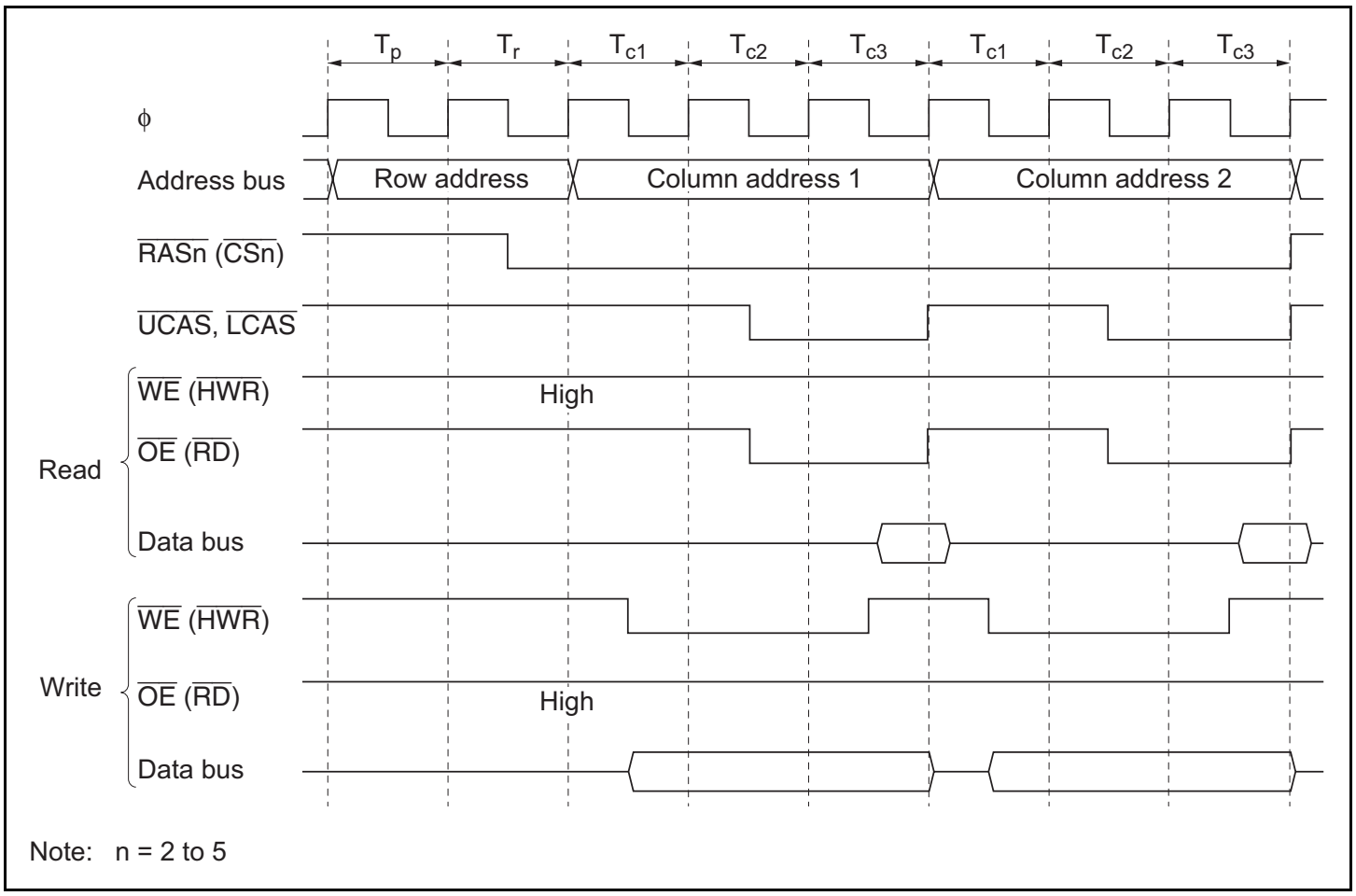


Figure 6.43 Operation Timing in Fast Page Mode (RAST = 0, CAST = 1)

The bus cycle can also be extended in burst access by inserting wait states. The wait state insertion method and timing are the same as for full access. For details see section 6.7.9, Wait Control.

(2) RAS Down Mode and RAS Up Mode

Even when burst operation is selected, it may happen that access to DRAM space is not continuous, but is interrupted by access to another space. In this case, if the $\overline{\text{RAS}}$ signal is held low during the access to the other space, burst operation can be resumed when the same row address in DRAM space is accessed again.

- RAS Down Mode

  To select RAS down mode, set both the RCDM bit and the BE bit to 1 in DRAMCR. If access to DRAM space is interrupted and another space is accessed, the $\overline{\text{RAS}}$ signal is held low during the access to the other space, and burst access is performed when the row address of the next DRAM space access is the same as the row address of the previous DRAM space access. Figure 6.44 shows an example of the timing in RAS down mode.

Note, however, that the $\overline{RAS}$ signal will go high if:

— a refresh operation is initiated in the RAS down state

— self-refreshing is performed

— the chip enters software standby mode

— the external bus is released

— the RCDM bit or BE bit is cleared to 0

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

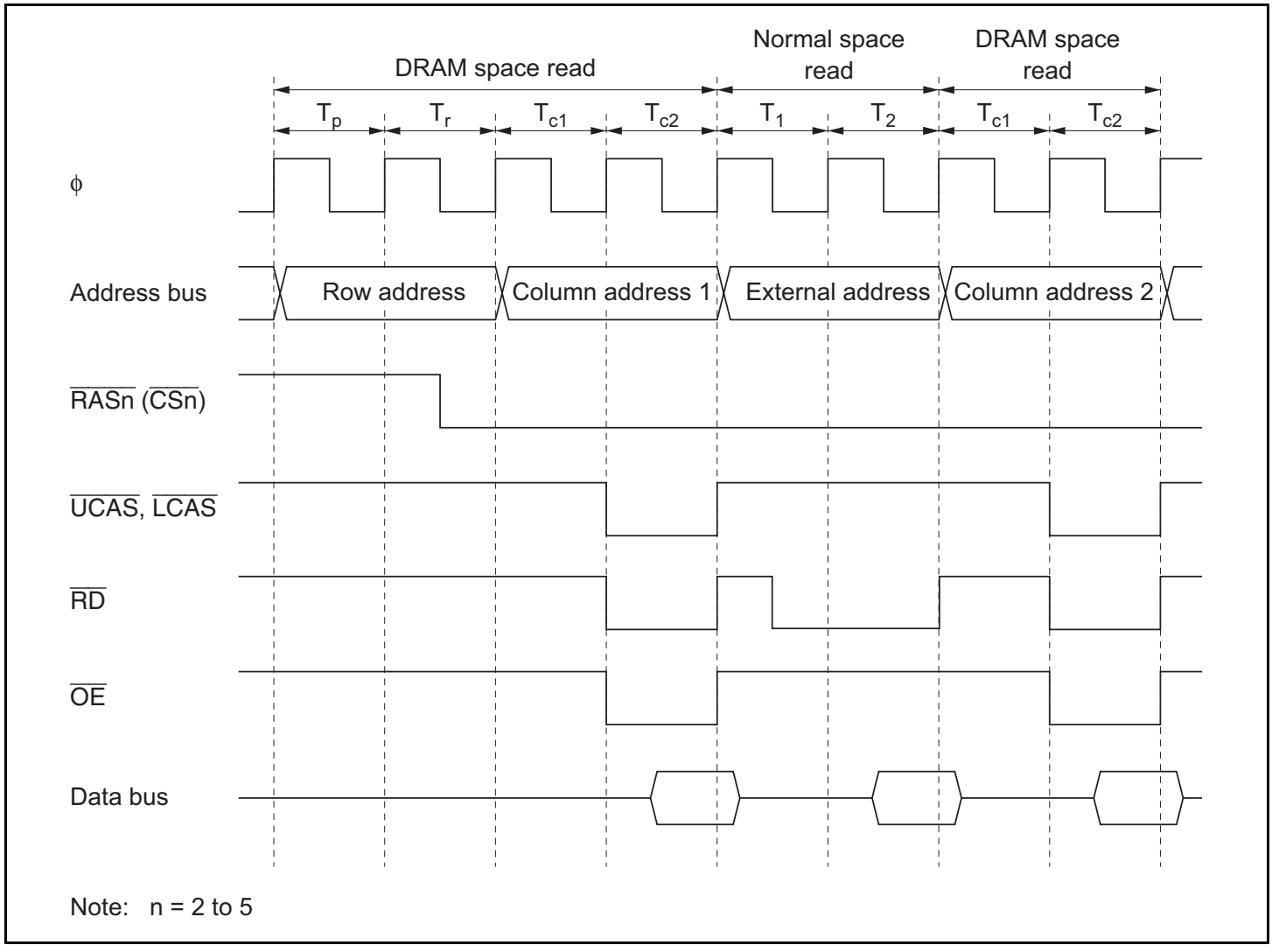


Figure 6.44 Example of Operation Timing in RAS Down Mode (RAST = 0, CAST = 0)

- RAS Up Mode

  To select RAS up mode, clear the RCDM bit to 0 in DRAMCR. Each time access to DRAM space is interrupted and another space is accessed, the $\overline{RAS}$ signal goes high again. Burst operation is only performed if DRAM space is continuous. Figure 6.45 shows an example of the timing in RAS up mode.

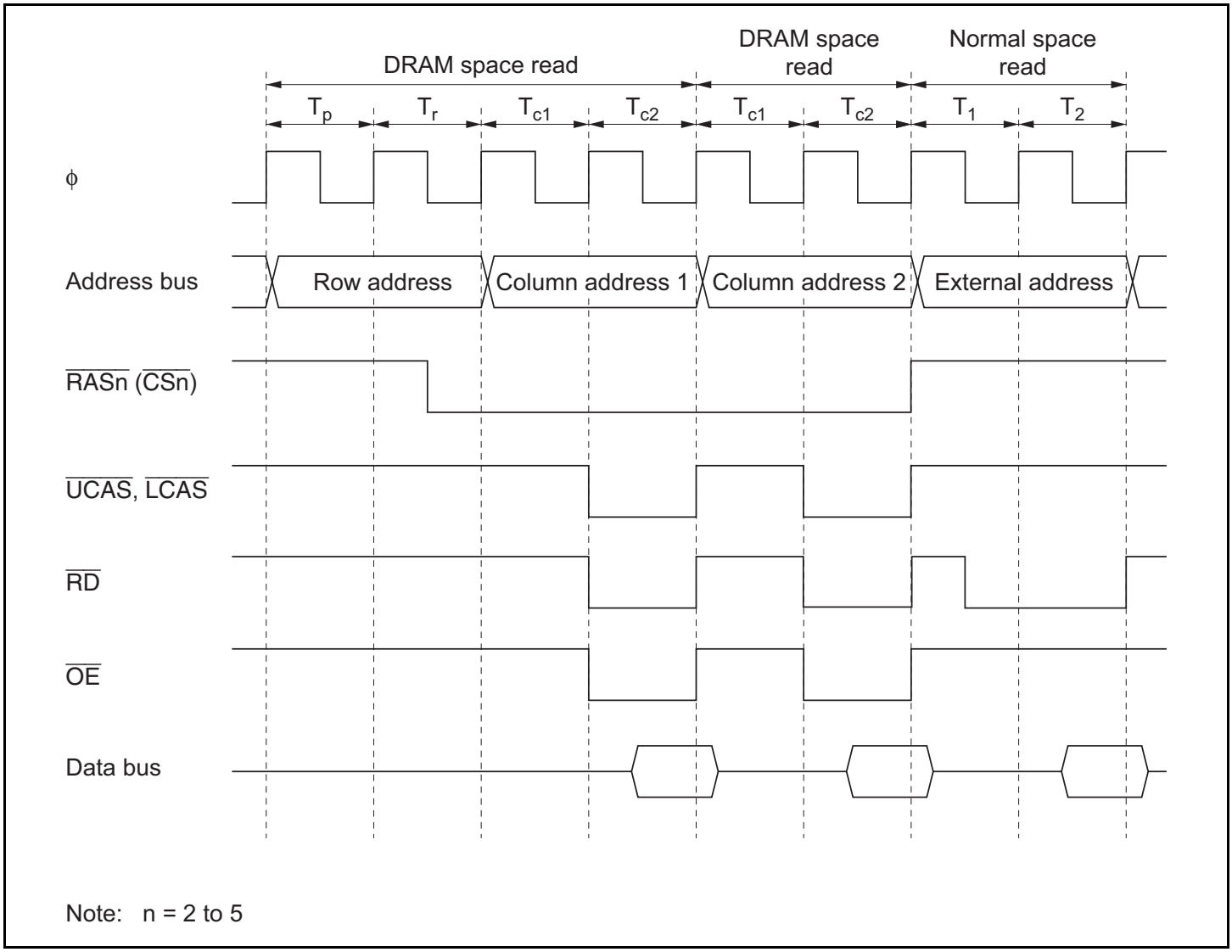


Figure 6.45 Example of Operation Timing in RAS Up Mode (RAST = 0, CAST = 0)

6.7.12 Refresh Control

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

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

(1) CAS-before-RAS (CBR) Refreshing

To select CBR refreshing, set the RFSHE bit to 1 in REFCR.

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

When bits RTCK2 to RTCK0 in REFCR are set, RTCNT starts counting up. RTCNT and RTCOR settings should therefore be completed before setting bits RTCK2 to RTCK0. RTCNT operation is shown in figure 6.46, compare match timing in figure 6.47, and CBR refresh timing in figure 6.48.

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

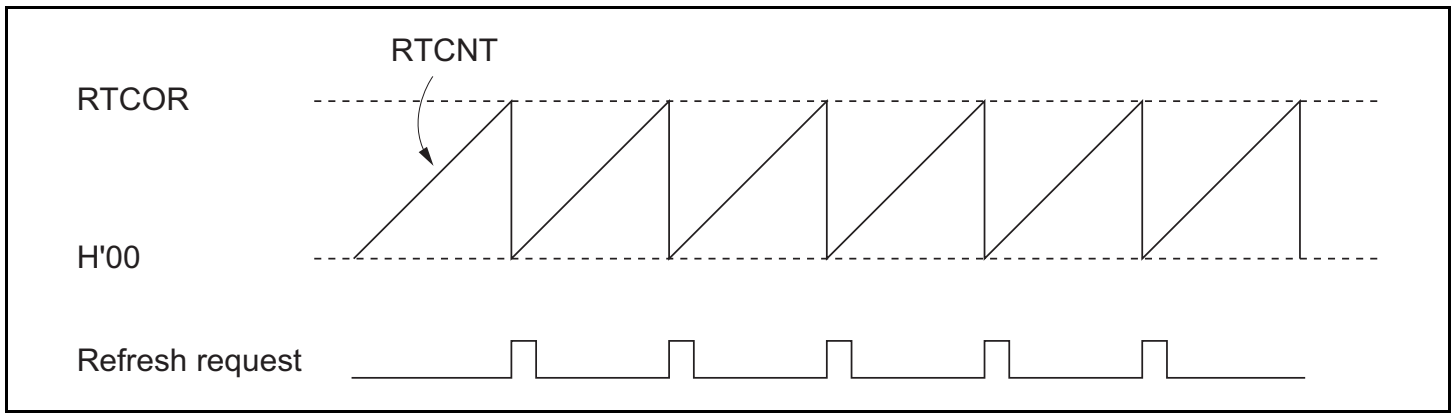


Figure 6.46 RTCNT Operation

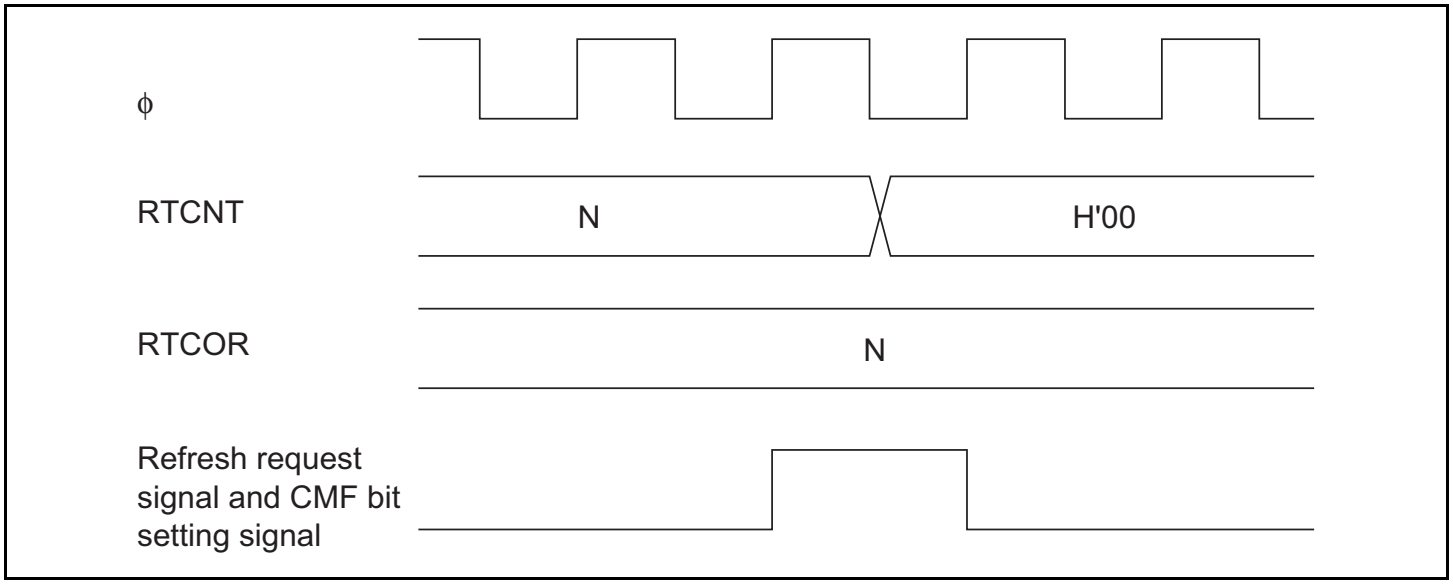


Figure 6.47 Compare Match Timing

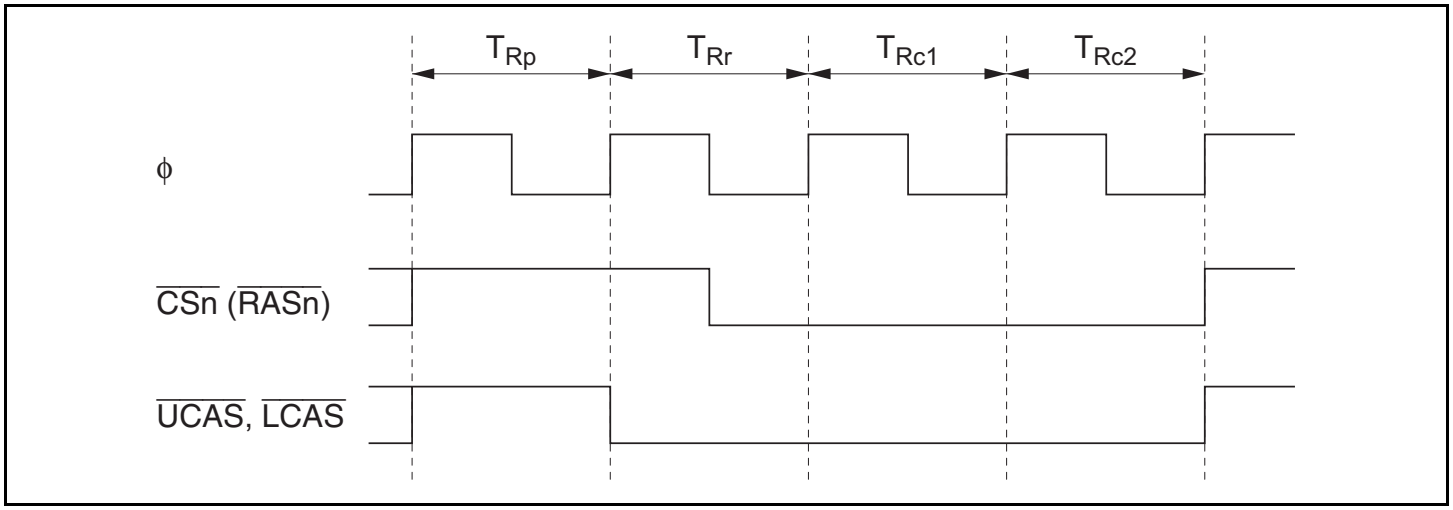


Figure 6.48 CBR Refresh Timing

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

Figure 6.49 shows the timing when bits RCW1 and RCW0 are set.

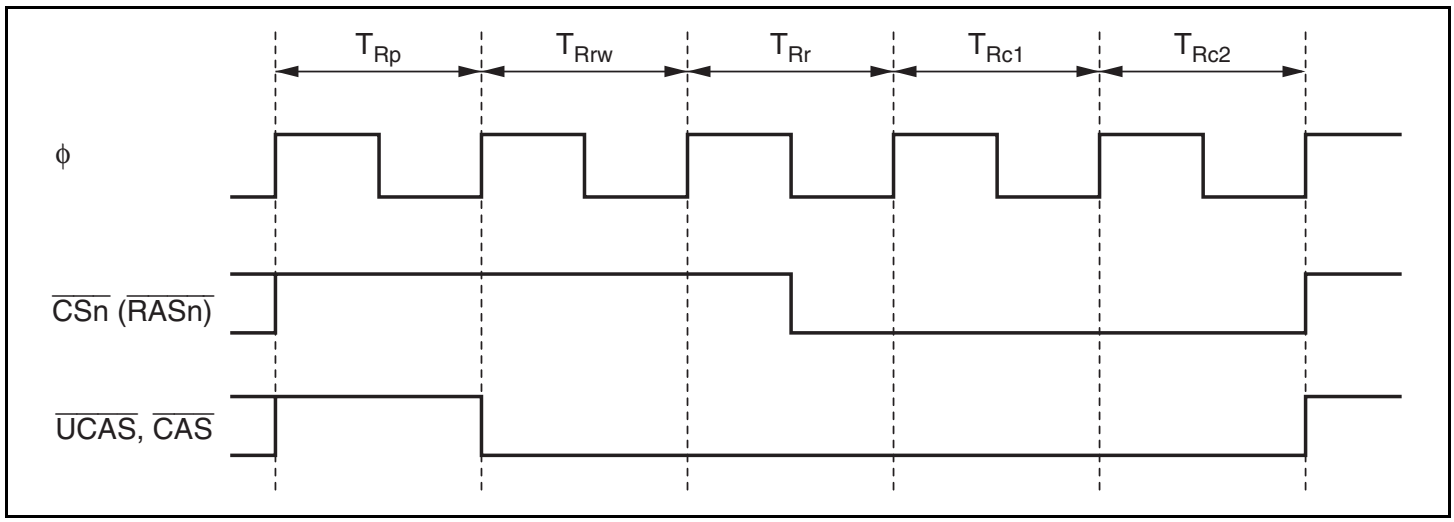


Figure 6.49 CBR Refresh Timing
(RCW1 = 0, RCW0 = 1, RLW1 = 0, RLW0 = 0)

Depending on the DRAM used, modification of the $\overline{WE}$ signal may not be permitted during the refresh period. In this case, the CBRM bit in REFCR should be set to 1. The bus controller will then insert refresh cycles in appropriate breaks between bus cycles. Figure 6.50 shows an example of the timing when the CBRM bit is set to 1. In this case the $\overline{CS}$ signal is not controlled, and retains its value prior to the start of the refresh period.

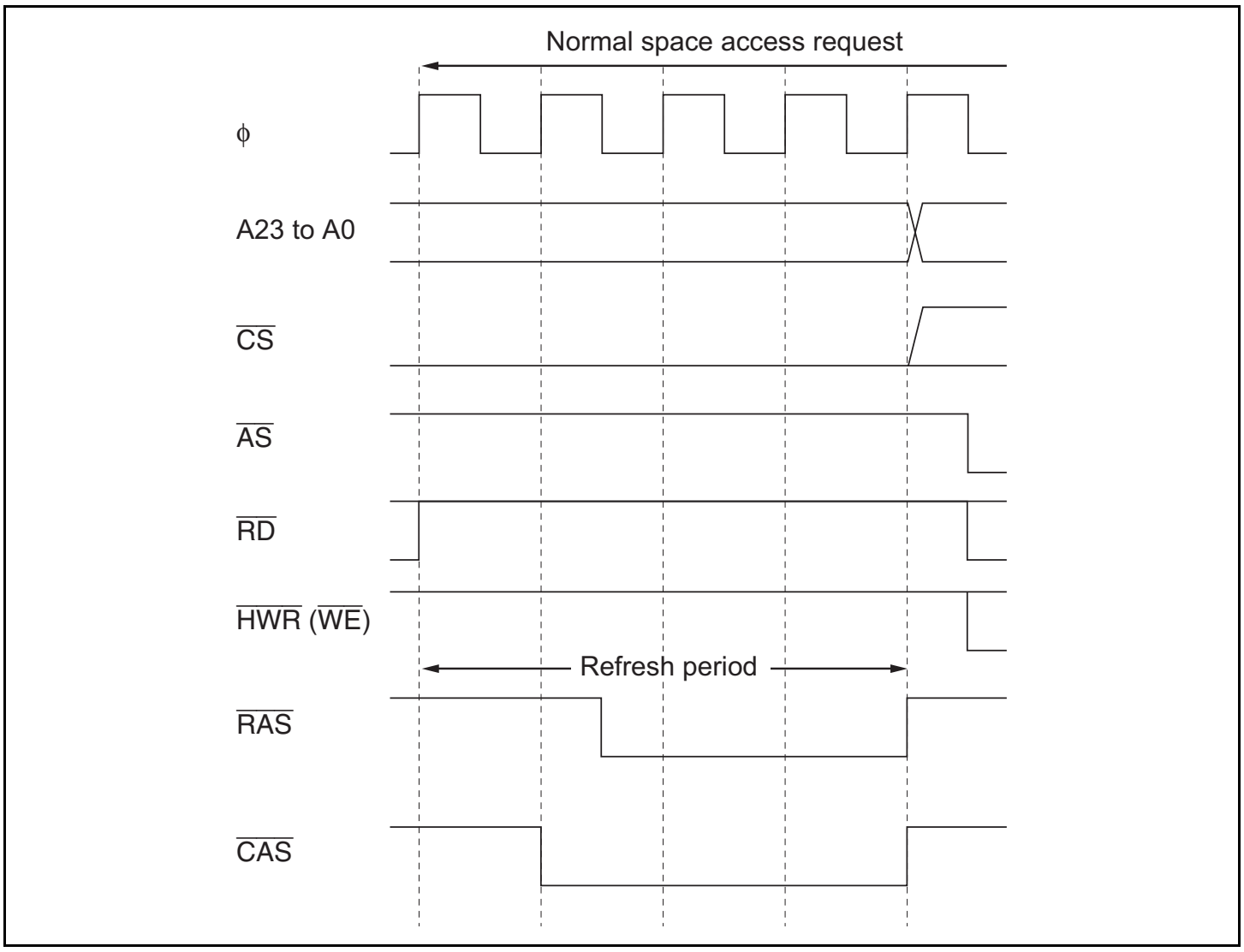


Figure 6.50 Example of CBR Refresh Timing (CBRM = 1)

(2) Self-Refreshing

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

To select self-refreshing, set the RFSHE bit and SLFRF bit to 1 in REFCR. When a SLEEP instruction is executed to enter software standby mode, the $\overline{CAS}$ and $\overline{RAS}$ signals are output and DRAM enters self-refresh mode, as shown in figure 6.51.

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

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

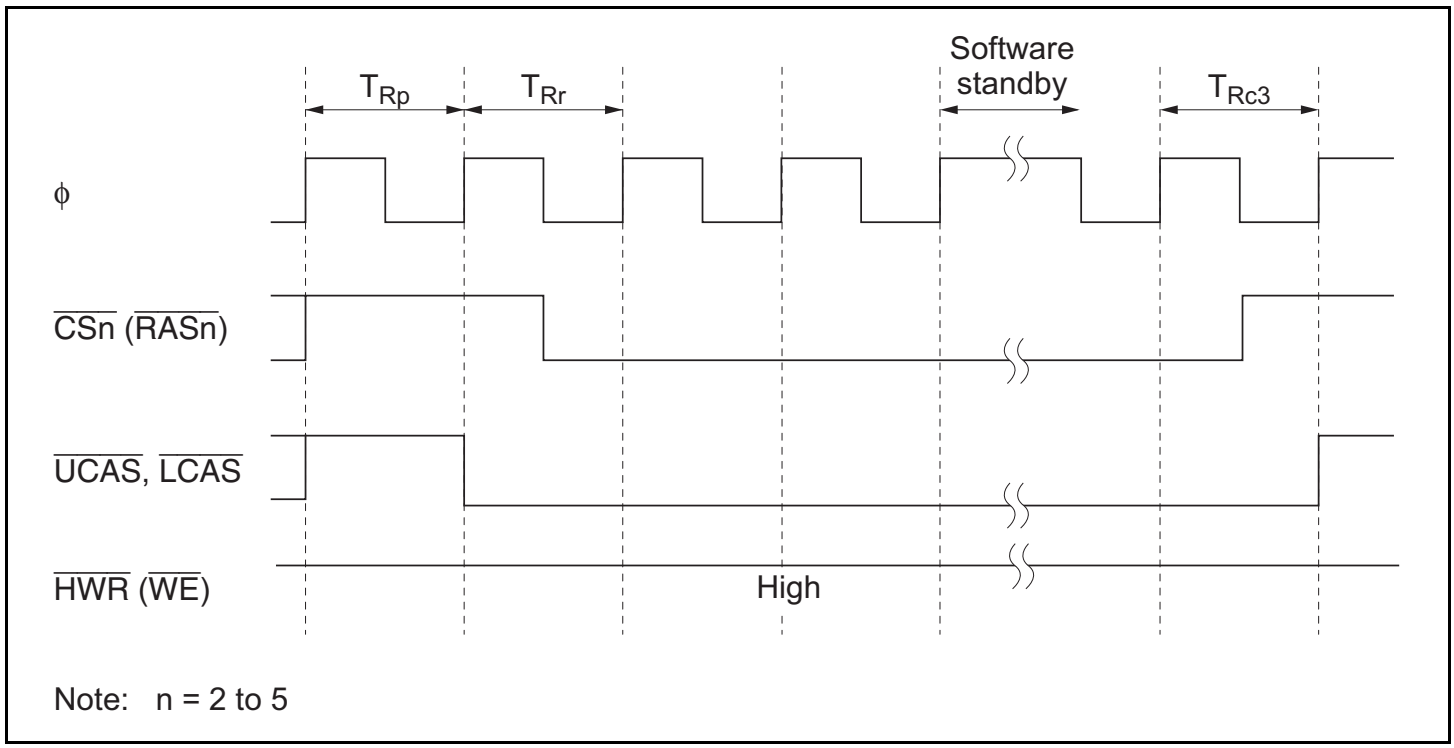


Figure 6.51 Self-Refresh Timing

In some DRAMs provided with a self-refresh mode, the $\overline{RAS}$ signal precharge time immediately after self-refreshing is longer than the normal precharge time. A setting can be made in bits TPCS2 to TPCS0 in REFCR to make the precharge time immediately after self-refreshing from 1 to 7 states longer than the normal precharge time. In this case, too, normal precharging is performed according to the setting of bits TPC1 and TPC0 in DRACCR, and therefore a setting should be made to give the optimum post-self-refresh precharge time, including this time. Figure 6.52 shows an example of the timing when the precharge time immediately after self-refreshing is extended by 2 states.

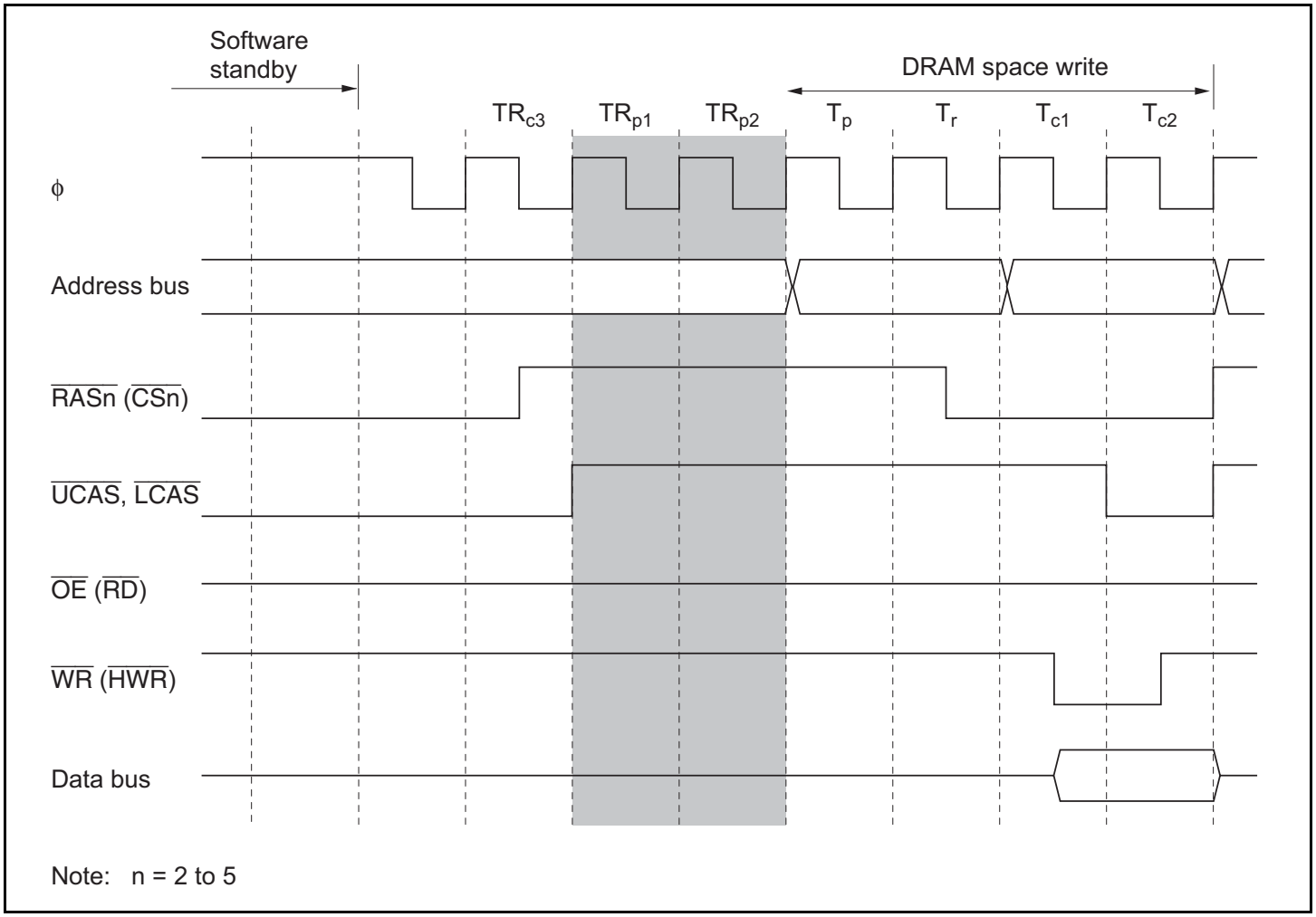


Figure 6.52 Example of Timing when Precharge Time after Self-Refreshing Is Extended by 2 States

(3) Refreshing and All-Module-Clocks-Stopped Mode

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

6.7.13 DMAC and EXDMAC Single Address Transfer Mode and DRAM Interface

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

(1) When DDS = 1 or EDDS = 1

Burst access is performed by determining the address only, irrespective of the bus master. With the DRAM interface, the $\overline{\text{DACK}}$ or $\overline{\text{EDACK}}$ output goes low from the T_{c1} state.
Figure 6.53 shows the $\overline{\text{DACK}}$ or $\overline{\text{EDACK}}$ output timing for the DRAM interface when DDS = 1 or EDDS = 1.

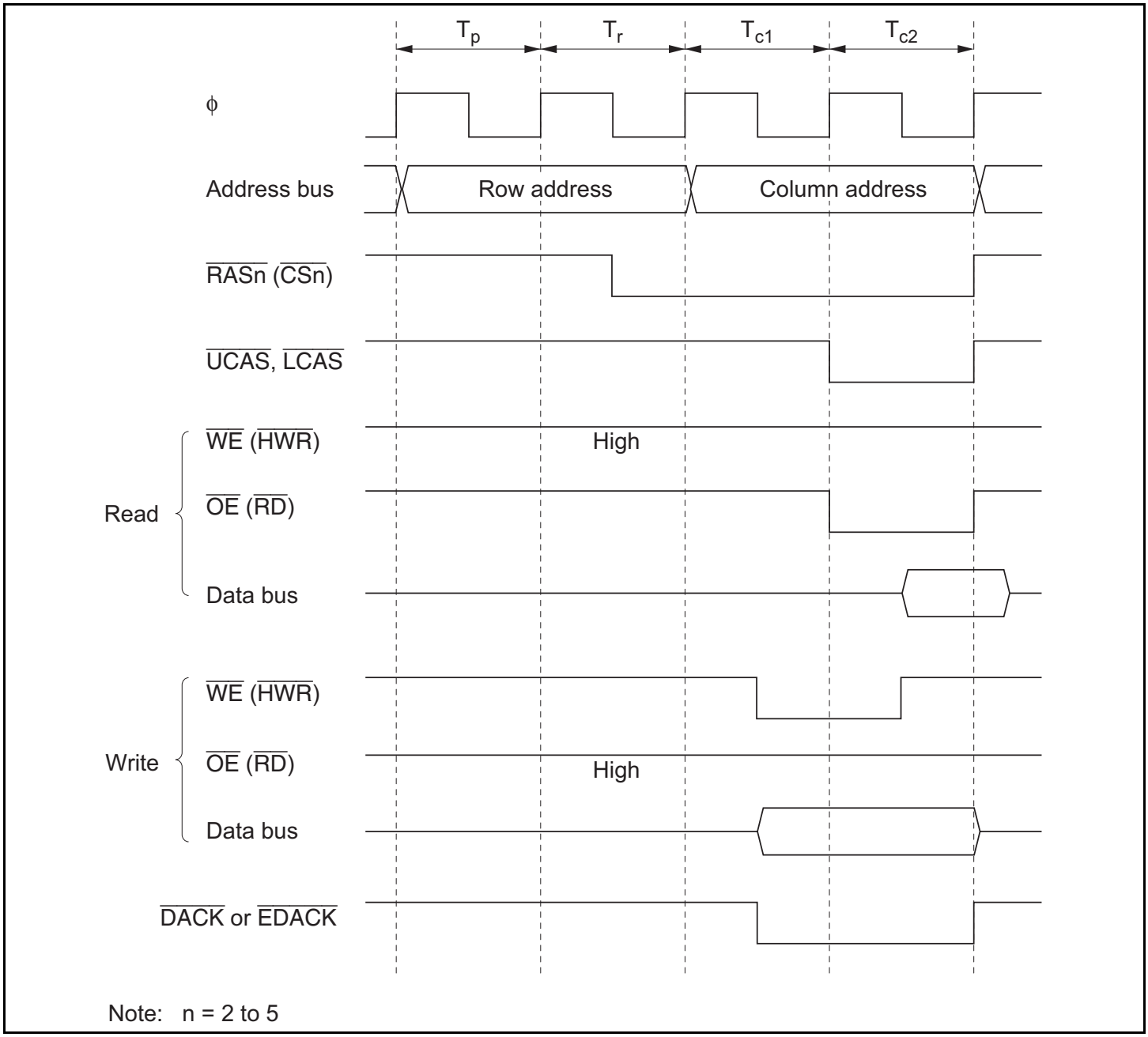


Figure 6.53 Example of $\overline{\text{DACK}}$/$\overline{\text{EDACK}}$ Output Timing when DDS = 1 or EDDS = 1 (RAST = 0, CAST = 0)

(2) When DDS = 0 or EDDS = 0

When DRAM space is accessed in DMAC or EXDMAC single address transfer mode, full access (normal access) is always performed. With the DRAM interface, the $\overline{\text{DACK}}$ or $\overline{\text{EDACK}}$ output goes low from the T_r state.
In modes other than DMAC or EXDMAC single address transfer mode, burst access can be used when accessing DRAM space.
Figure 6.54 shows the $\overline{\text{DACK}}$ or $\overline{\text{EDACK}}$ output timing for the DRAM interface when DDS = 0 or EDDS = 0.

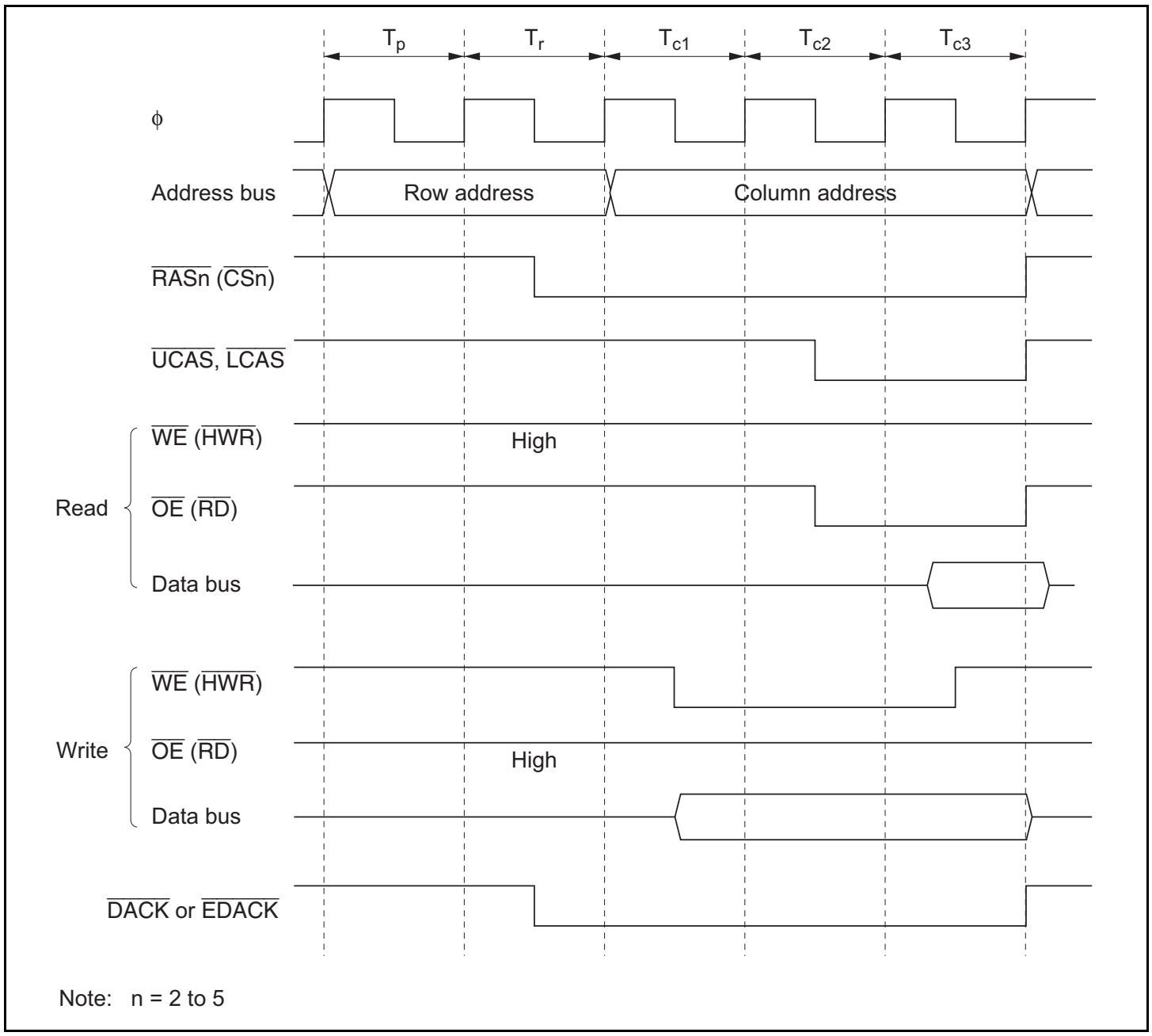


Figure 6.54 Example of $\overline{\text{DACK}}$/$\overline{\text{EDACK}}$ Output Timing when DDS = 0 or EDDS = 0 (RAST = 0, CAST = 1)

6.8 Synchronous DRAM Interface

In the H8S/2456R Group, external address space areas 2 to 5 can be designated as continuous synchronous DRAM space, and synchronous DRAM interfacing performed. The synchronous DRAM interface allows synchronous DRAM to be directly connected to this LSI. A synchronous DRAM space of up to 8 Mbytes can be set by means of bits RMTS2 to RMTS0 in DRAMCR. Synchronous DRAM of CAS latency 1 to 4 can be connected.

Note: The synchronous DRAM interface is not supported by the H8S/2456 Group and H8S/2454 Group.

6.8.1 Setting Continuous Synchronous DRAM Space

Areas 2 to 5 are designated as continuous synchronous DRAM space by setting bits RMTS2 to RMTS0 in DRAMCR. The relation between the settings of bits RMTS2 to RMTS0 and synchronous DRAM space is shown in table 6.8. Possible synchronous DRAM interface settings are and continuous area (areas 2 to 5).

Table 6.8 Relation between Settings of Bits RMTS2 to RMTS0 and Synchronous DRAM Space

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

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

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

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

6.8.2 Address Multiplexing

With continuous synchronous DRAM space, the row address and column address are multiplexed. In address multiplexing, the size of the shift of the row address is selected with bits MXC2 to MXC0 in DRAMCR. The address-precharge-setting command (Precharge-sel) can be output on the upper column address. Table 6.9 shows the relation between the settings of MXC2 to MXC0 and the shift size. The MXC2 bit should be set to 1 when the synchronous DRAM interface is used.

Table 6.9 Relation between Settings of Bits MXC2 to MXC0 and Address Multiplexing

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

Legend:

×: Don't care.

P: Precharge-sel

6.8.3 Data Bus

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

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

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

6.8.4 Pins Used for Synchronous DRAM Interface

Table 6.10 shows pins used for the synchronous DRAM interface and their functions.

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

Table 6.10 Synchronous DRAM Interface Pins

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

6.8.5 Synchronous DRAM Clock

The synchronous clock (SDRAMϕ) is output from the $\overline{CS5}$ pin. SDRAMϕ is shifted by 90° phase from ϕ. Therefore, a stable margin is ensured for the synchronous DRAM that operates at the rising edge of clocks. Figure 6.55 shows the relationship between ϕ and SDRAMϕ.

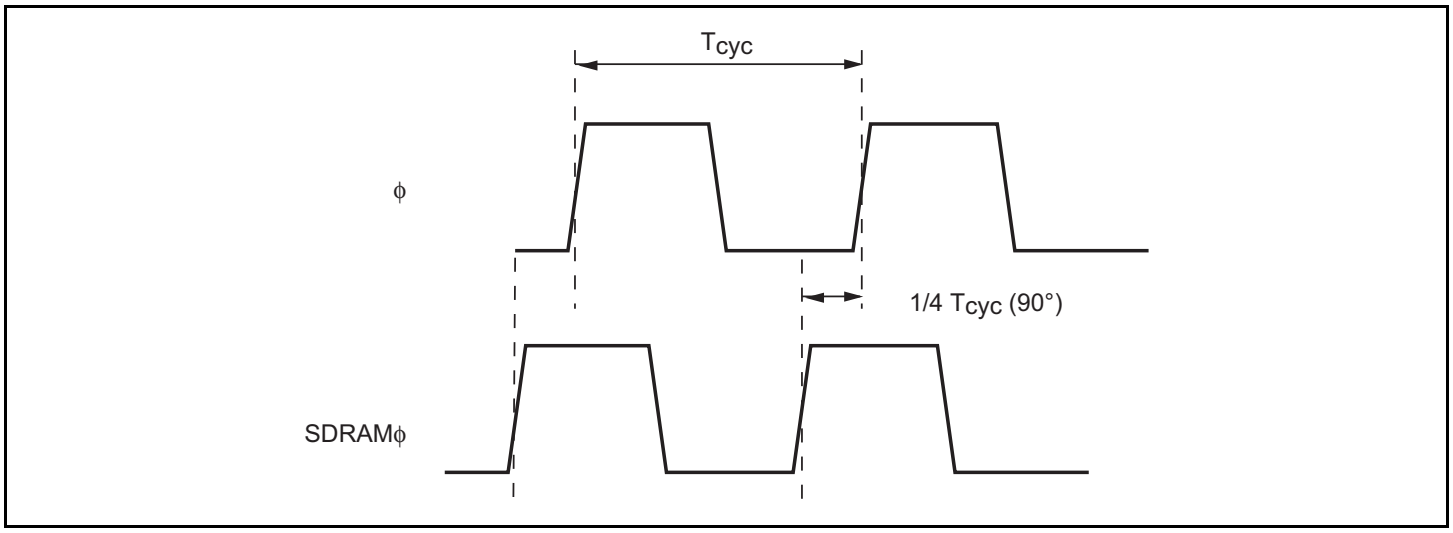


Figure 6.55 Relationship between ϕ and SDRAMϕ

6.8.6 Basic Timing

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

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

Figure 6.56 shows the basic timing for synchronous DRAM.

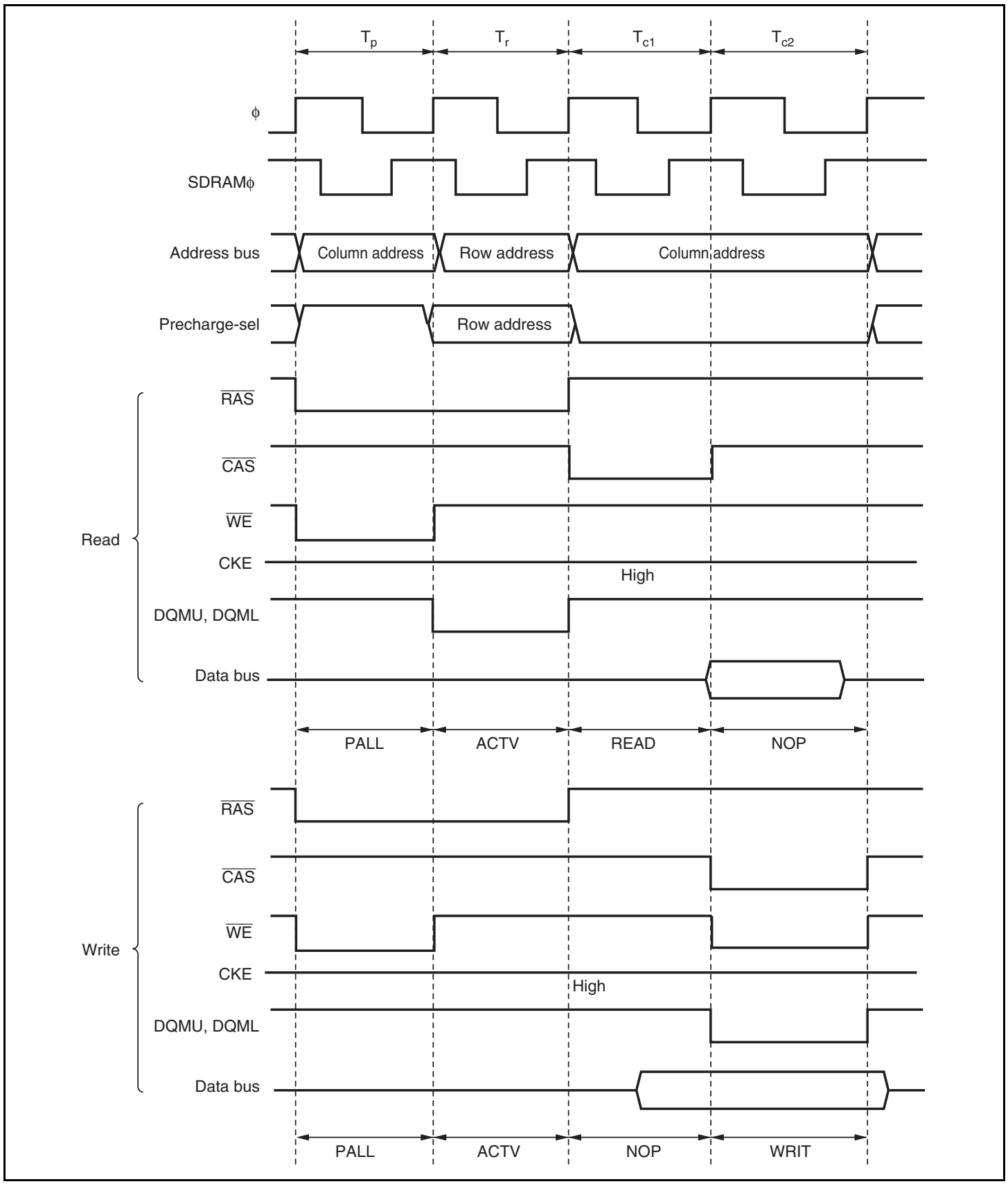


Figure 6.56 Basic Access Timing of Synchronous DRAM (CAS Latency 1)

6.8.7 CAS Latency Control

CAS latency is controlled by settings of the W22 to W20 bits of WTCRB. Set the CAS latency count, as shown in table 6.11, by the setting of synchronous DRAM. Depending on the setting, the CAS latency control cycle (T_{c1}) is inserted. WTCRB can be set regardless of the setting of the AST2 bit of ASTCR. Figure 6.57 shows the CAS latency control timing when synchronous DRAM of CAS latency 3 is connected.

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

Table 6.11 Setting CAS Latency

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

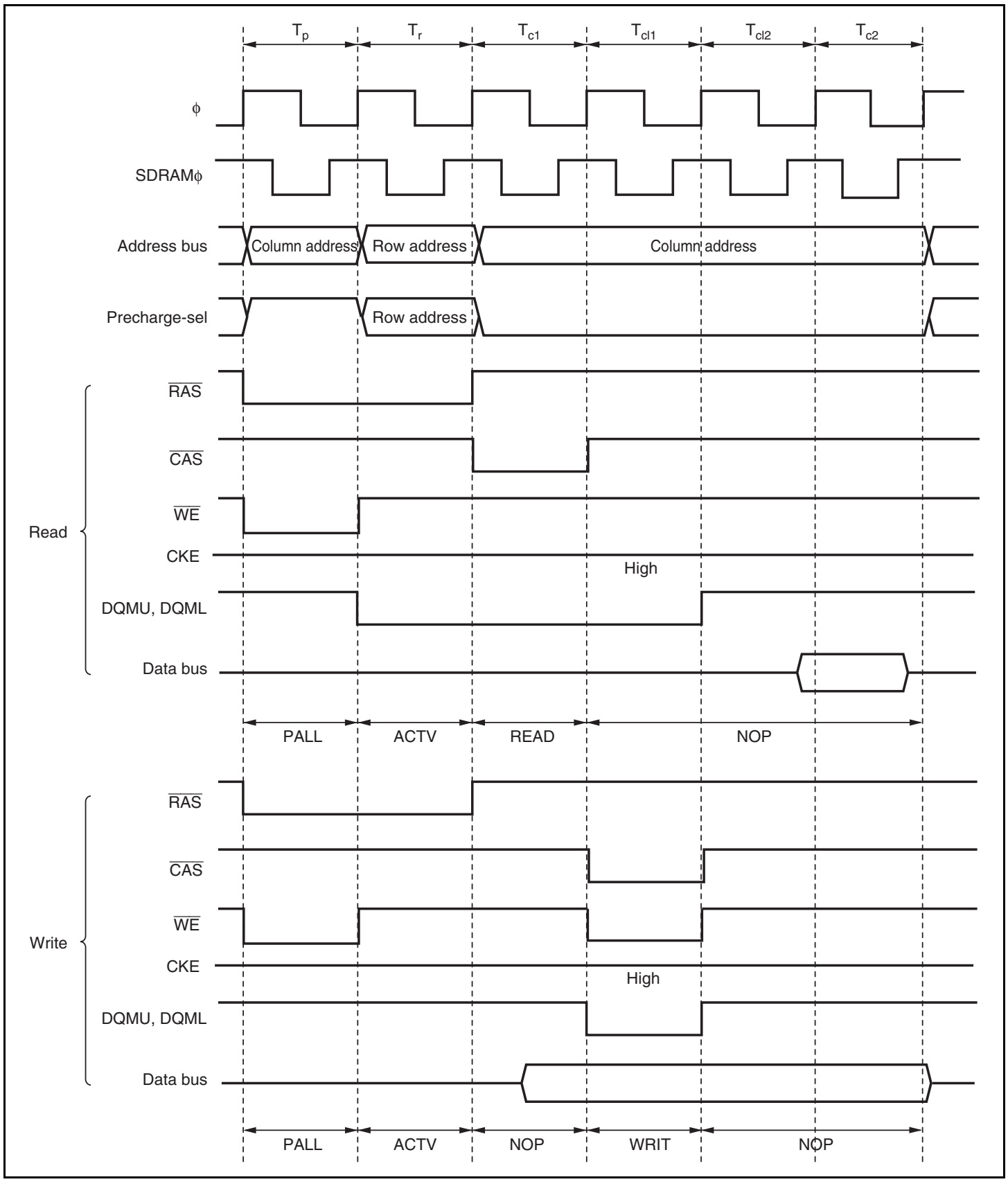


Figure 6.57 CAS Latency Control Timing (SDWCD = 0, CAS Latency 3)

6.8.8 Row Address Output State Control

When the command interval specification from the ACTV command to the next READ/WRIT command cannot be satisfied, 1 to 3 states (Trw) that output the NOP command can be inserted between the Tr cycle that outputs the ACTV command and the Tc1 cycle that outputs the column address by setting the RCD1 and RCD0 bits of DRACCR. Use the optimum setting for the wait time according to the synchronous DRAM connected and the operating frequency of this LSI. Figure 6.58 shows an example of the timing when the one Trw state is set.

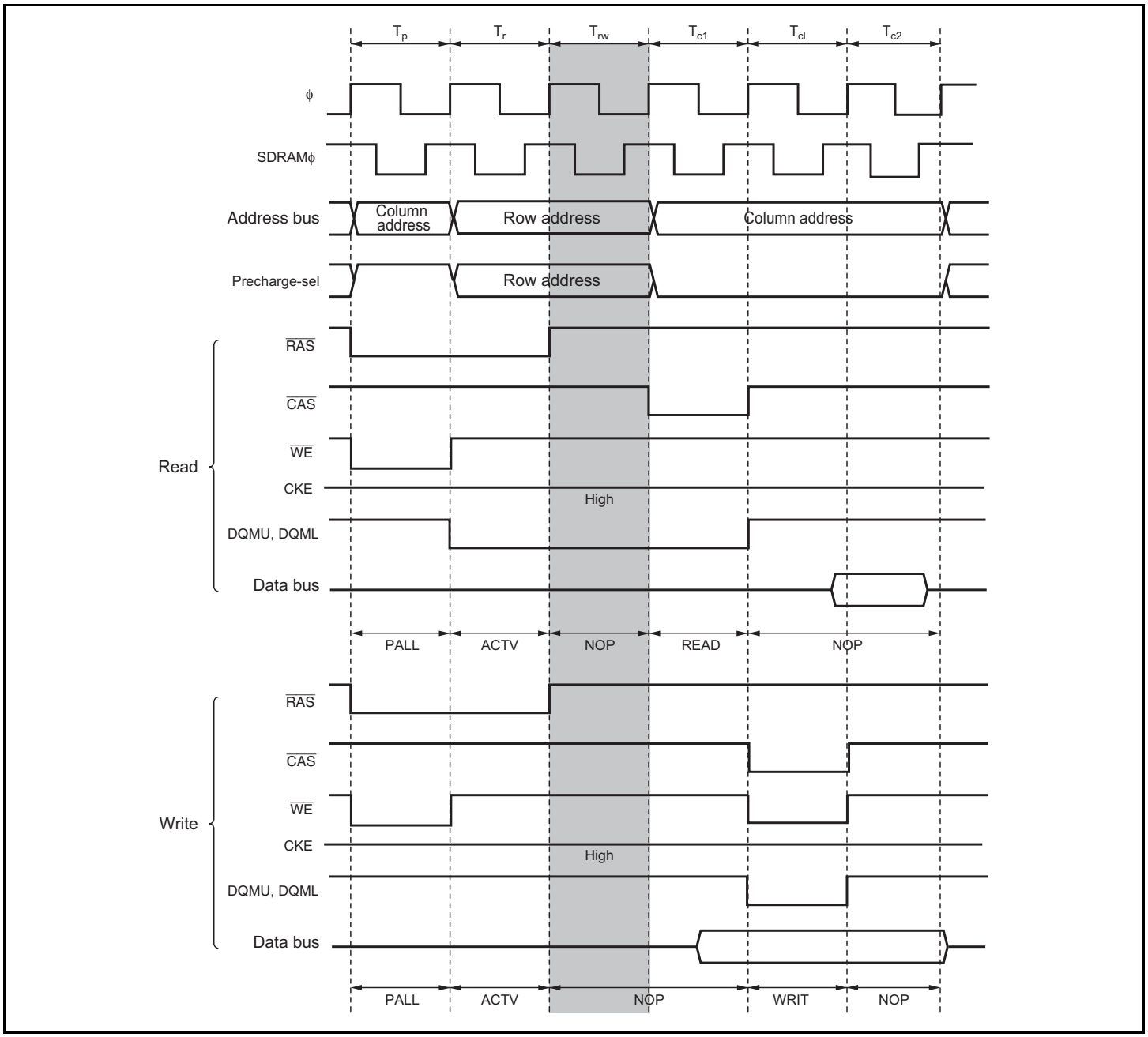


Figure 6.58 Example of Access Timing when Row Address Output Hold State Is 1 State (RCD1 = 0, RCD0 = 1, SDWCD = 0, CAS Latency 2)

6.8.9 Precharge State Count

When the interval specification from the PALL command to the next ACTV/REF command cannot be satisfied, from one to four T_p states can be selected by setting bits TPC1 and TPC0 in DRACCR. Set the optimum number of T_p cycles according to the synchronous DRAM connected and the operating frequency of this LSI. Figure 6.59 shows the timing when two Tp states are inserted.

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

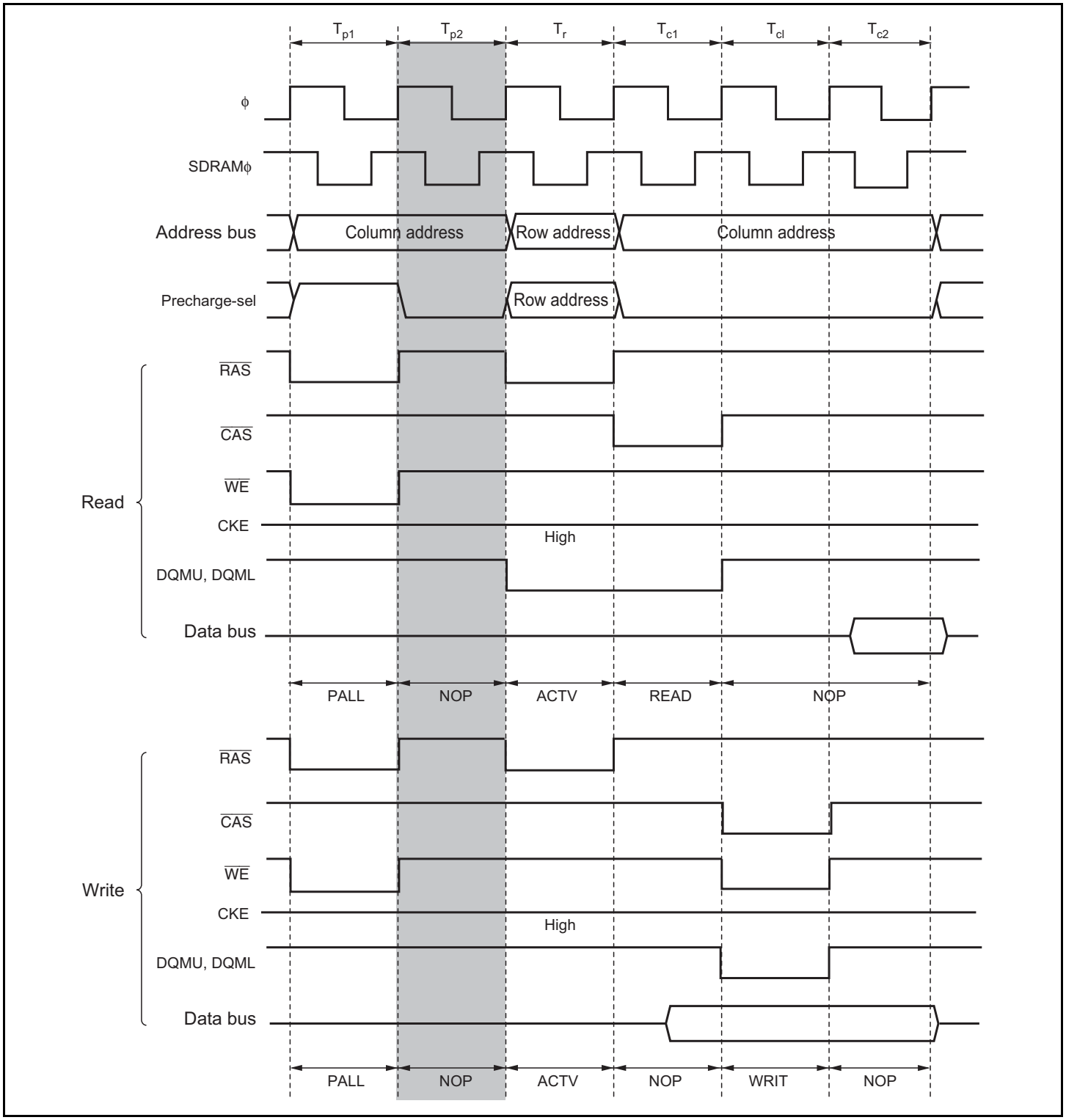


Figure 6.59 Example of Timing with Two-State Precharge Cycle (TPC1 = 0, TPC0 = 1, SDWCD = 0, CAS Latency 2)

6.8.10 Bus Cycle Control in Write Cycle

By setting the SDWCD bit of the DRACCR to 1, the CAS latency control cycle (Tc1) that is inserted by the WTCRB register in the write access of the synchronous DRAM can be disabled. Disabling the CAS latency control cycle can reduce the write-access cycle count as compared to synchronous DRAM read access. Figure 6.60 shows the write access timing when the CAS latency control cycle is disabled.

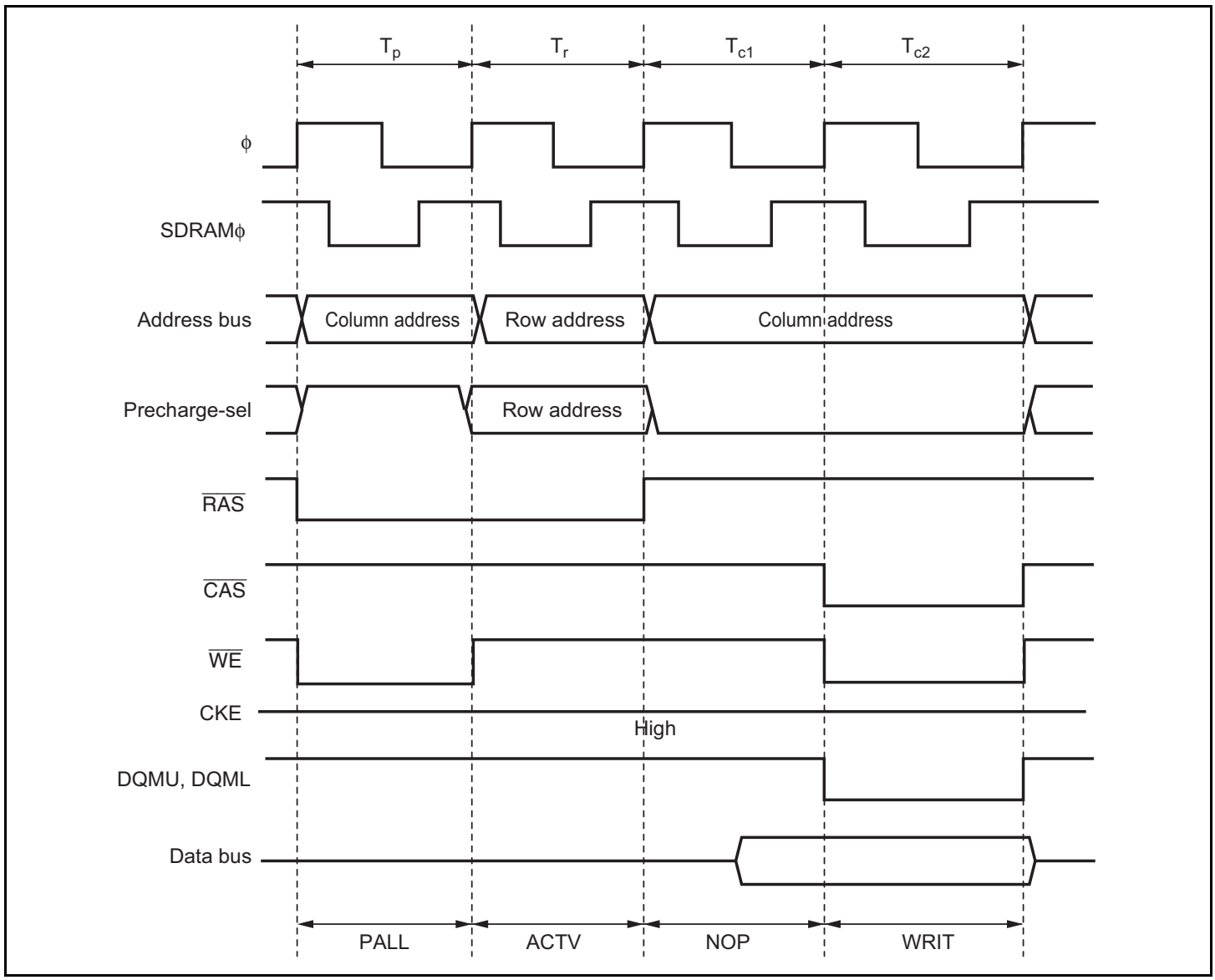


Figure 6.60 Example of Write Access Timing when CAS Latency Control Cycle Is Disabled (SDWCD = 1)

6.8.11 Byte Access Control

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

Figures 6.61 and 6.62 show the control timing for DQM, and figure 6.63 shows an example of connection of byte control by DQMU and DQML.

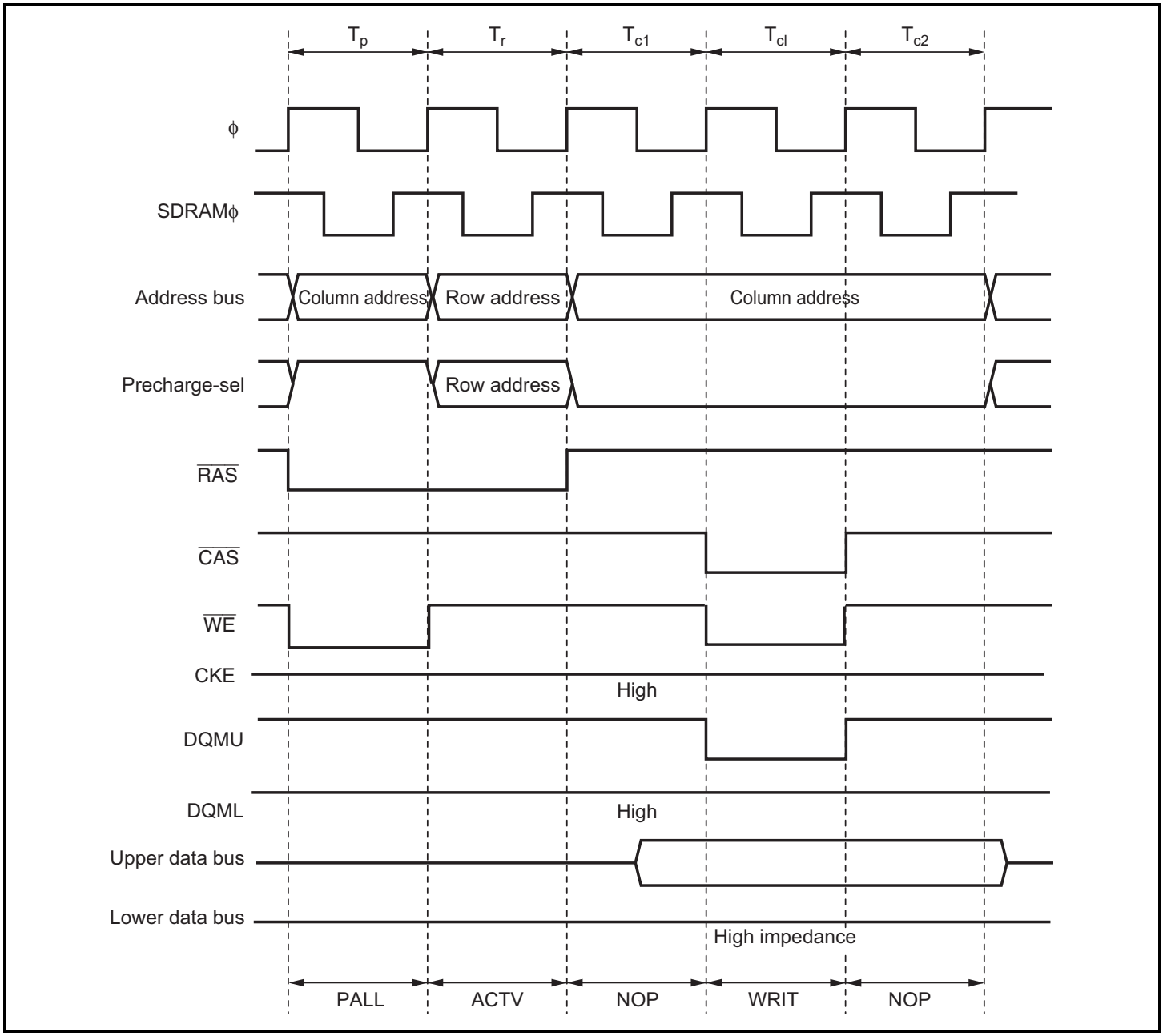


Figure 6.61 DQMU and DQML Control Timing (Upper Byte Write Access: SDWCD = 0, CAS Latency 2)

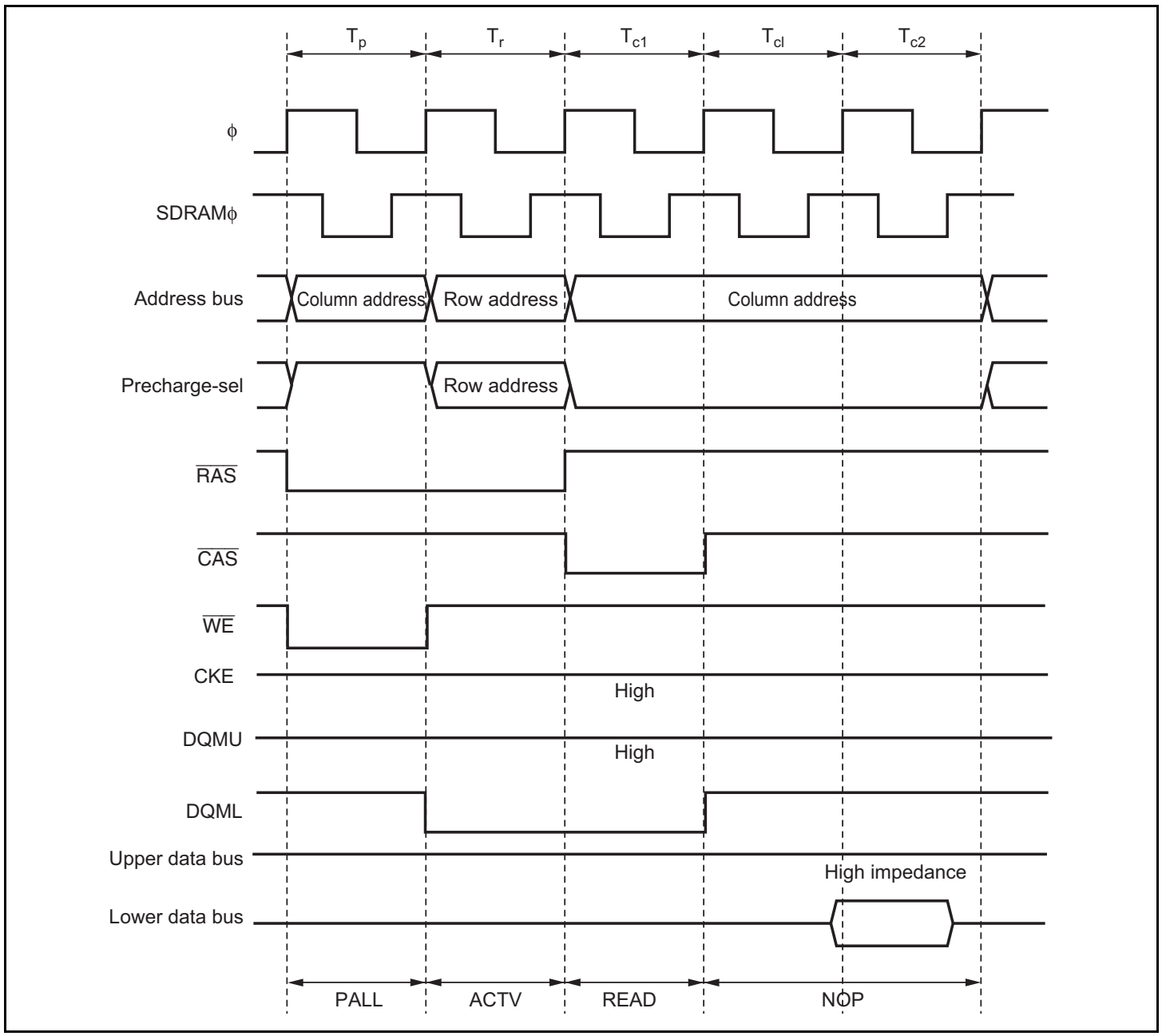
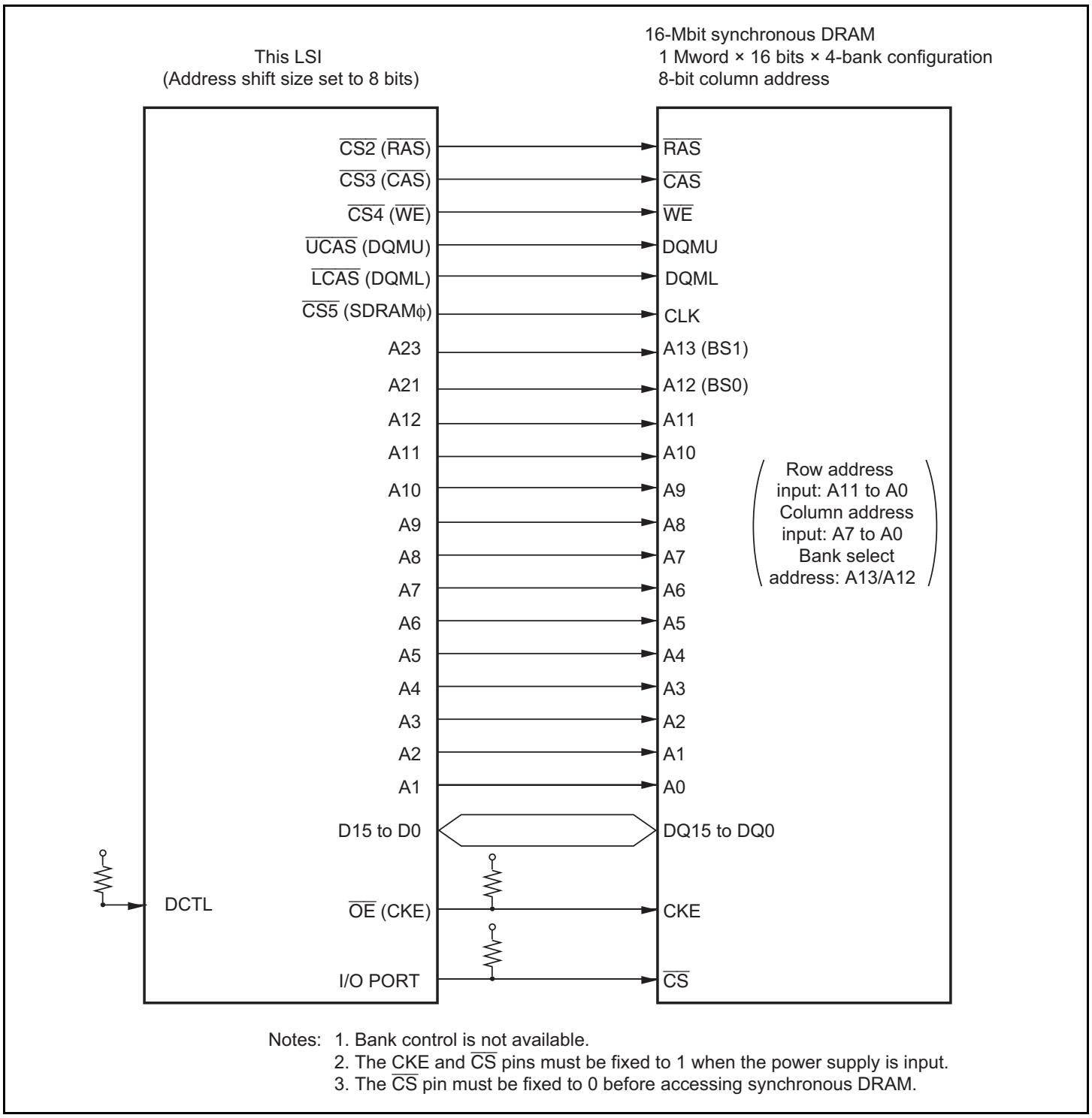


Figure 6.62 DQMU and DQML Control Timing (Lower Byte Read Access: CAS Latency 2)



Notes: 1. Bank control is not available.
2. The CKE and CS pins must be fixed to 1 when the power supply is input.
3. The CS pin must be fixed to 0 before accessing synchronous DRAM.

Figure 6.63 Example of DQMU and DQML Byte Control

6.8.12 Burst Operation

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

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

(1) Burst Access Operation Timing

Figure 6.64 shows the operation timing for burst access. When there are consecutive access cycles for continuous synchronous DRAM space, the column address output cycles continue as long as the row address is the same for consecutive access cycles. The row address used for the comparison is set with bits MXC2 to MXC0 in DRAMCR.

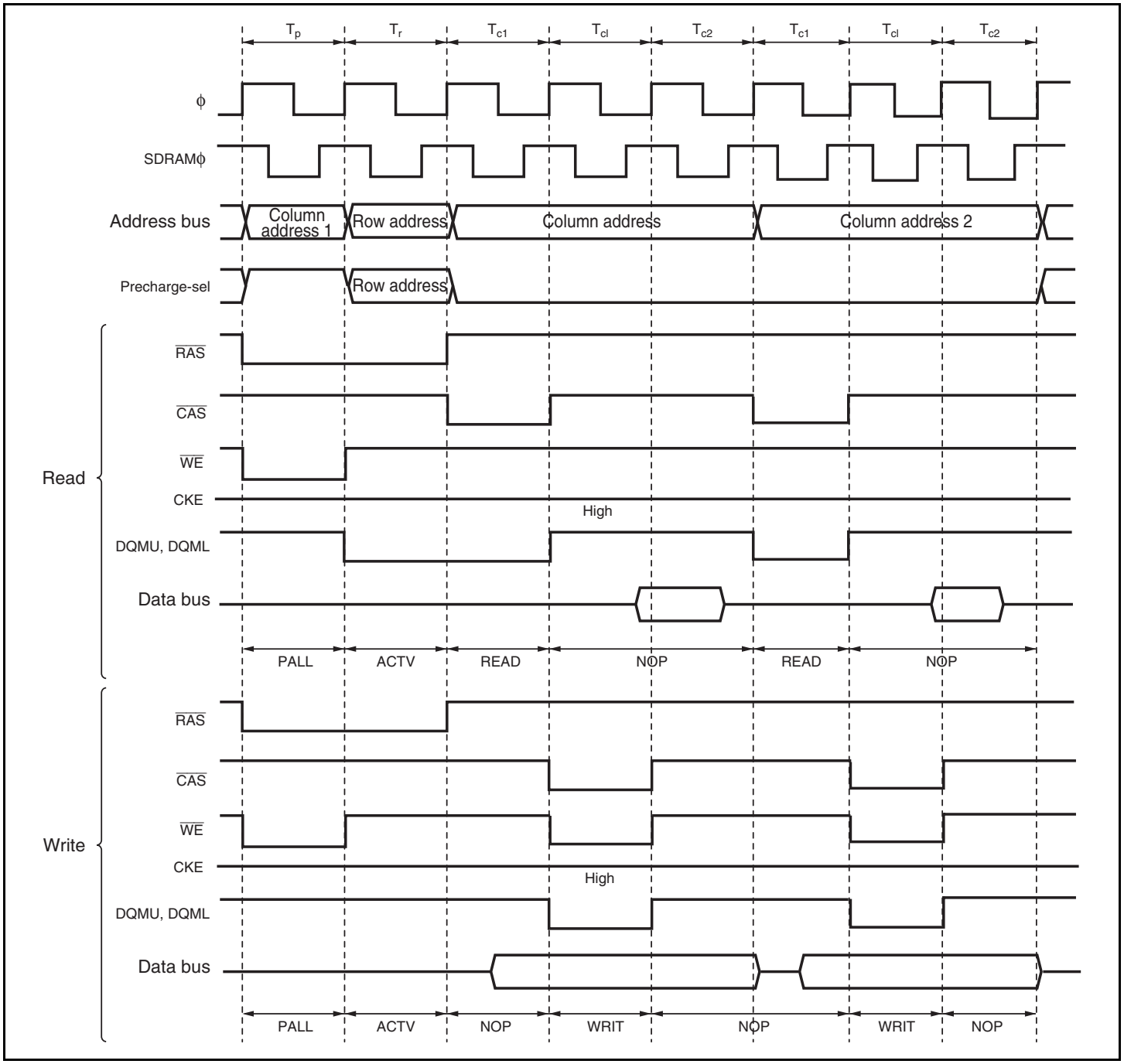


Figure 6.64 Operation Timing of Burst Access
(BE = 1, SDWCD = 0, CAS Latency 2)

(2) RAS Down Mode

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

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

Figure 6.65 shows an example of the timing in RAS down mode.

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

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

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

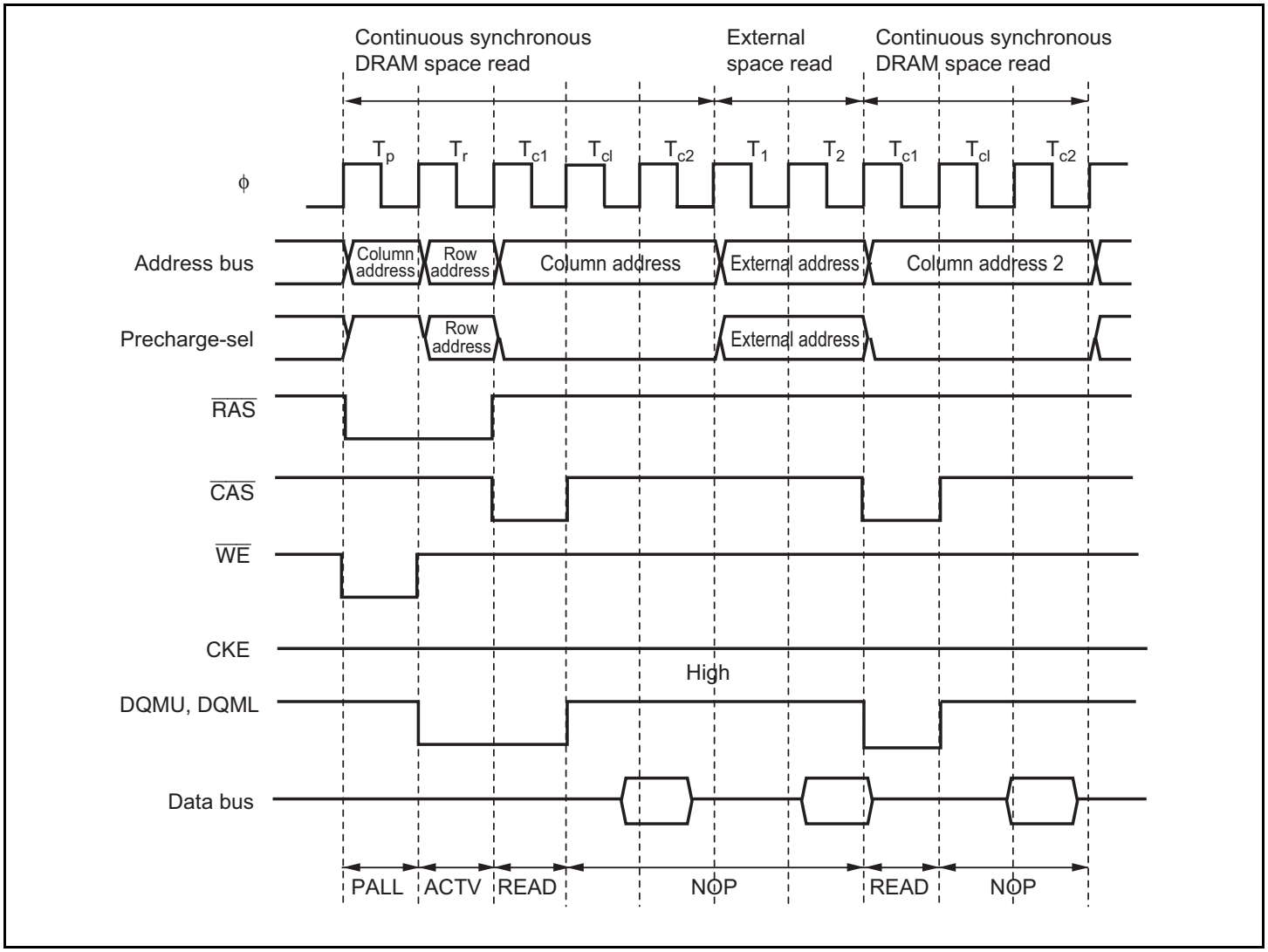


Figure 6.65 Example of Operation Timing in RAS Down Mode (BE = 1, CAS Latency 2)

6.8.13 Refresh Control

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

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

(1) Auto Refreshing

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

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

When bits RTCK2 to RTCK0 are set, RTCNT starts counting up. RTCNT and RTCOR settings should therefore be completed before setting bits RTCK2 to RTCK0. Auto refresh timing is shown in figure 6.66.

Since the refresh counter operation is the same as the operation in the DRAM interface, see section 6.7.12, Refresh Control.

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

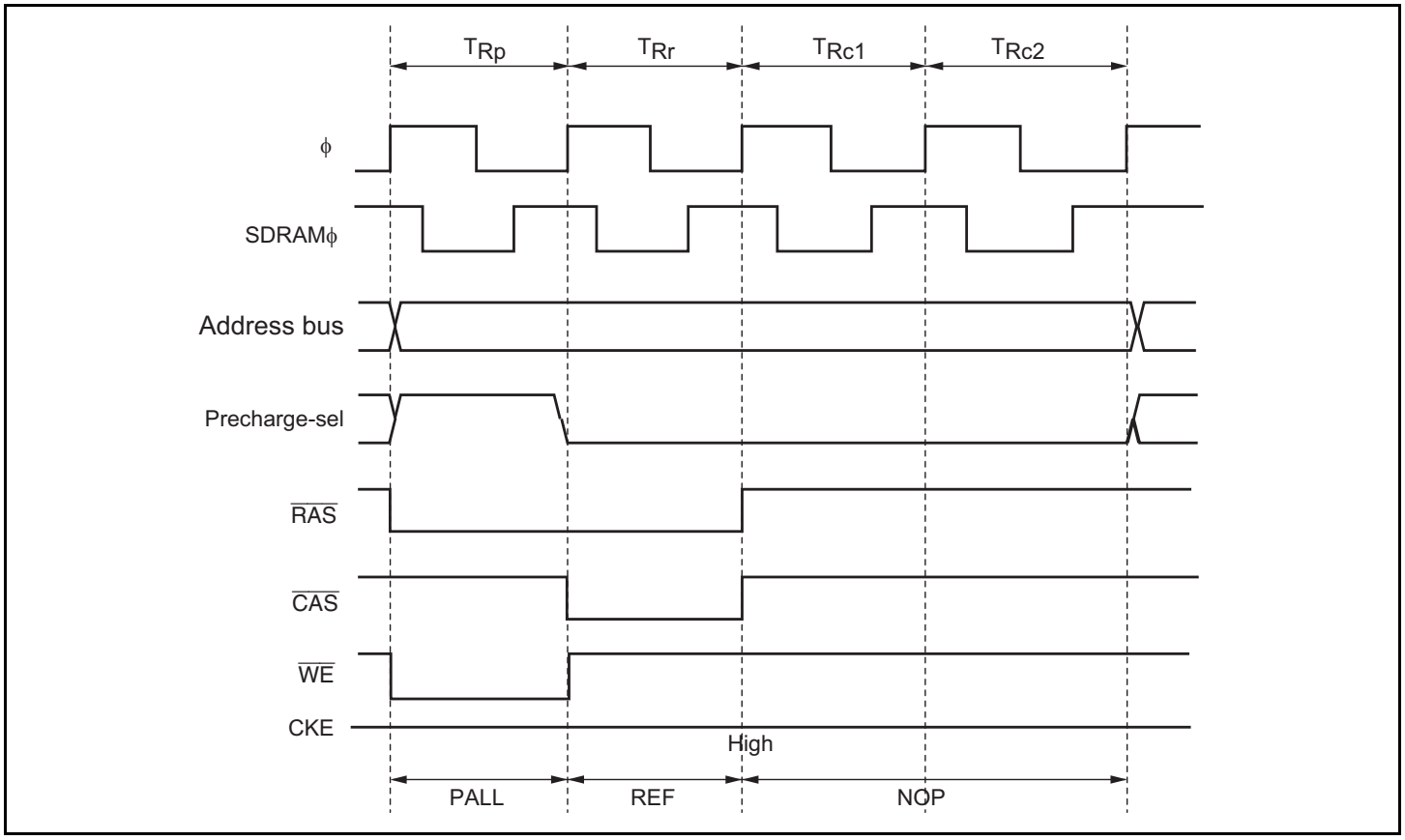


Figure 6.66 Auto Refresh Timing

When the interval specification from the PALL command to the REF command cannot be satisfied, setting the RCW1 and RCW0 bits of REFCR enables one to three wait states to be inserted after the T_{Rp} cycle that is set by the TPC1 and TPC0 bits of DRACCR. Set the optimum number of waits according to the synchronous DRAM connected and the operating frequency of this LSI. Figure 6.67 shows the timing when one wait state is inserted. Since the setting of bits TPC1 and TPC0 of DRACCR is also valid in refresh cycles, the command interval can be extended by the RCW1 and RCW0 bits after the precharge cycles.

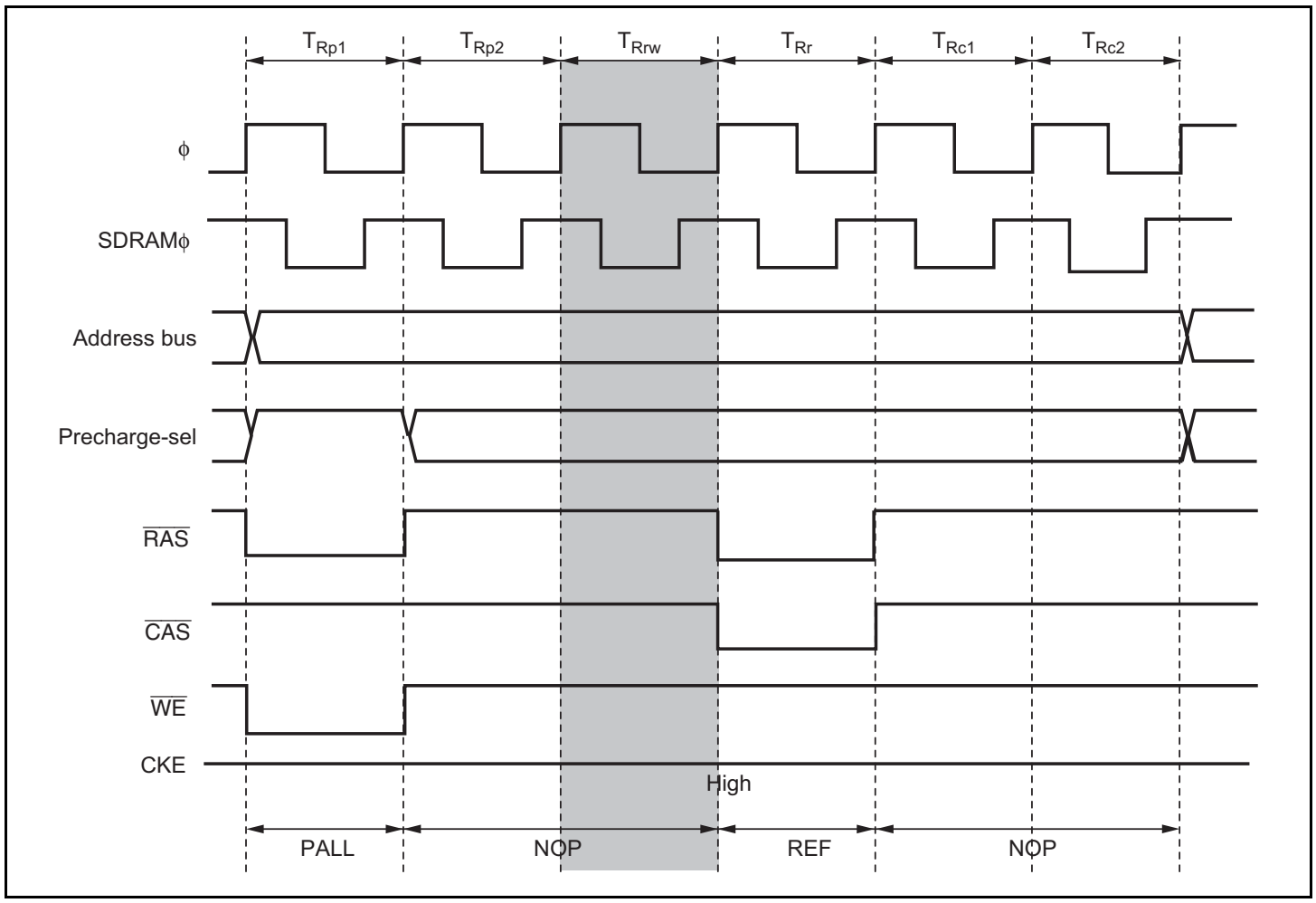


Figure 6.67 Auto Refresh Timing
(TPC = 1, TPC0 = 1, RCW1 = 0, RCW0 = 1)

When the interval specification from the REF command to the ACTV cannot be satisfied, setting the RLW1 and RLW0 bits of REFCR enables one to three wait states to be inserted in the refresh cycle. Set the optimum number of waits according to the synchronous DRAM connected and the operating frequency of this LSI. Figure 6.68 shows the timing when one wait state is inserted.

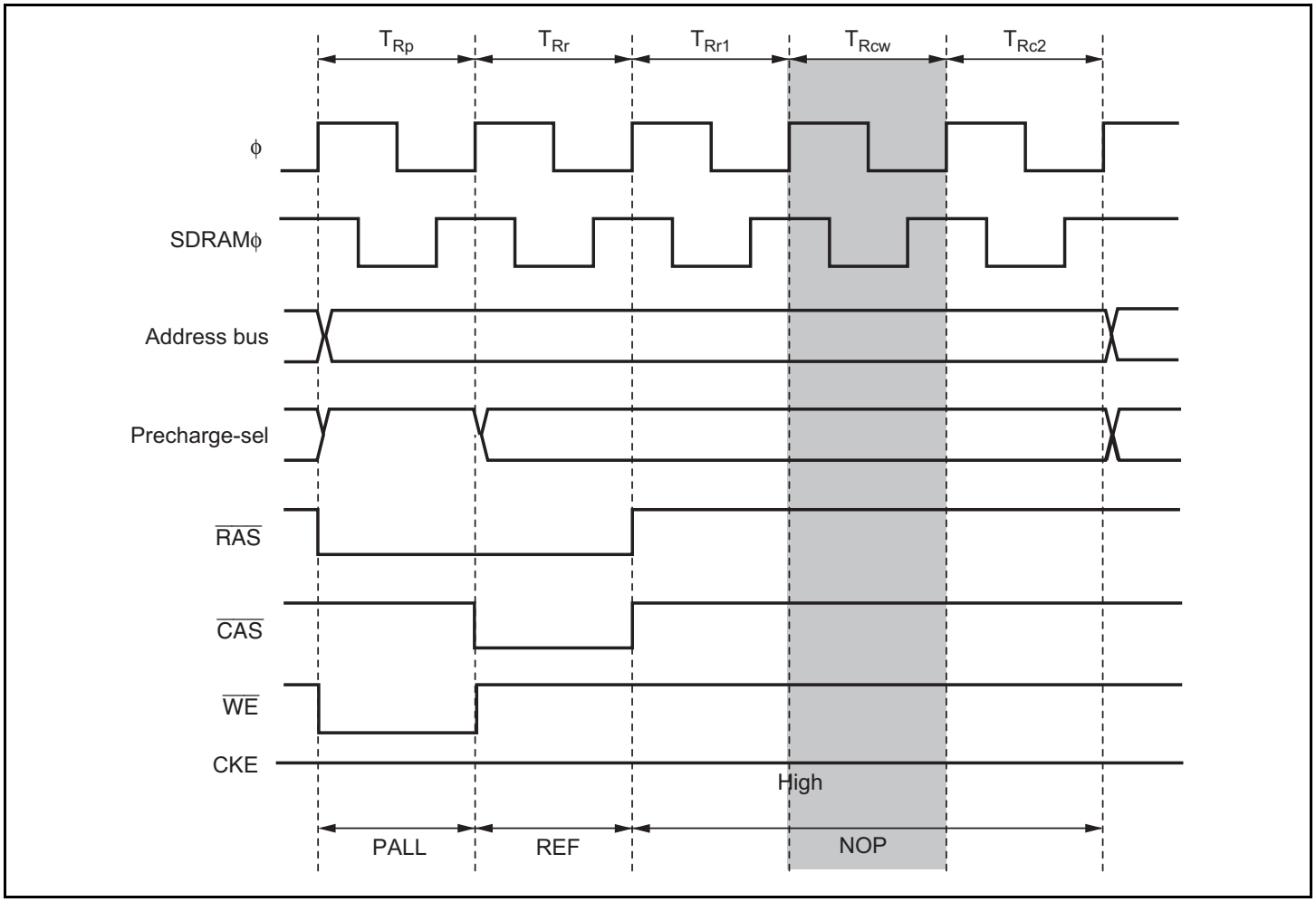


Figure 6.68 Auto Refresh Timing
(TPC = 0, TPC0 = 0, RLW1 = 0, RLW0 = 1)

(2) Self-Refreshing

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

To select self-refreshing, set the RFSHE bit to 1 in REFCR. When a SLEEP instruction is executed to enter software standby mode, the SELF command is issued, as shown in figure 6.69.

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

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

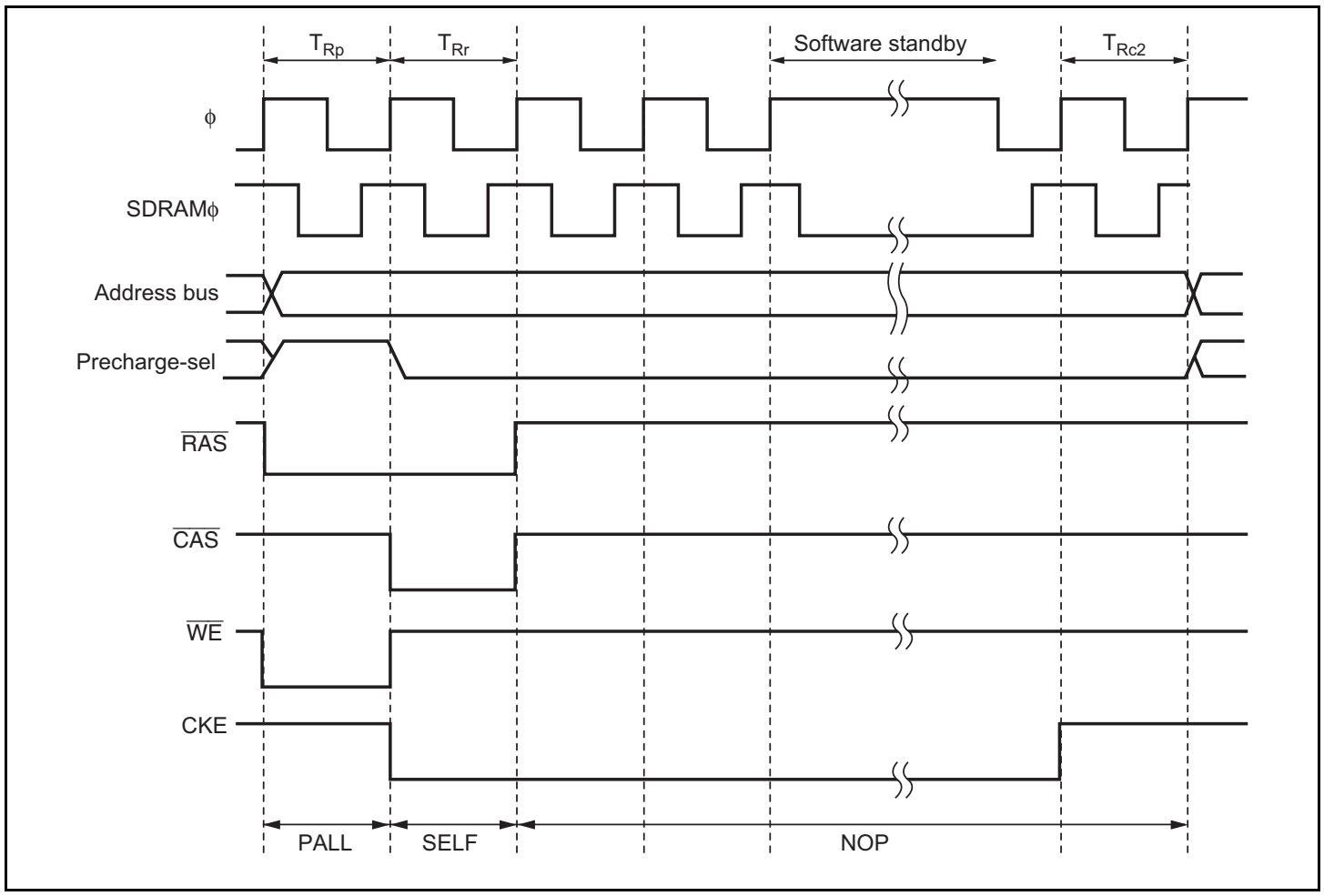


Figure 6.69 Self-Refresh Timing
(TPC1 = 1, TPC0 = 0, RCW1 = 0, RCW0 = 0, RLW1 = 0, RLW0 = 0)

In some synchronous DRAMs provided with a self-refresh mode, the interval between clearing self-refreshing and the next command is specified. A setting can be made in bits TPCS2 to TPCS0 in REFCR to make the precharge time after self-refreshing from 1 to 7 states longer than the normal precharge time. In this case, too, normal precharging is performed according to the setting of bits TPC1 and TPC0 in DRACCR, and therefore a setting should be made to give the optimum post-self-refresh precharge time, including this time. Figure 6.70 shows an example of the timing when the precharge time after self-refreshing is extended by 2 states.

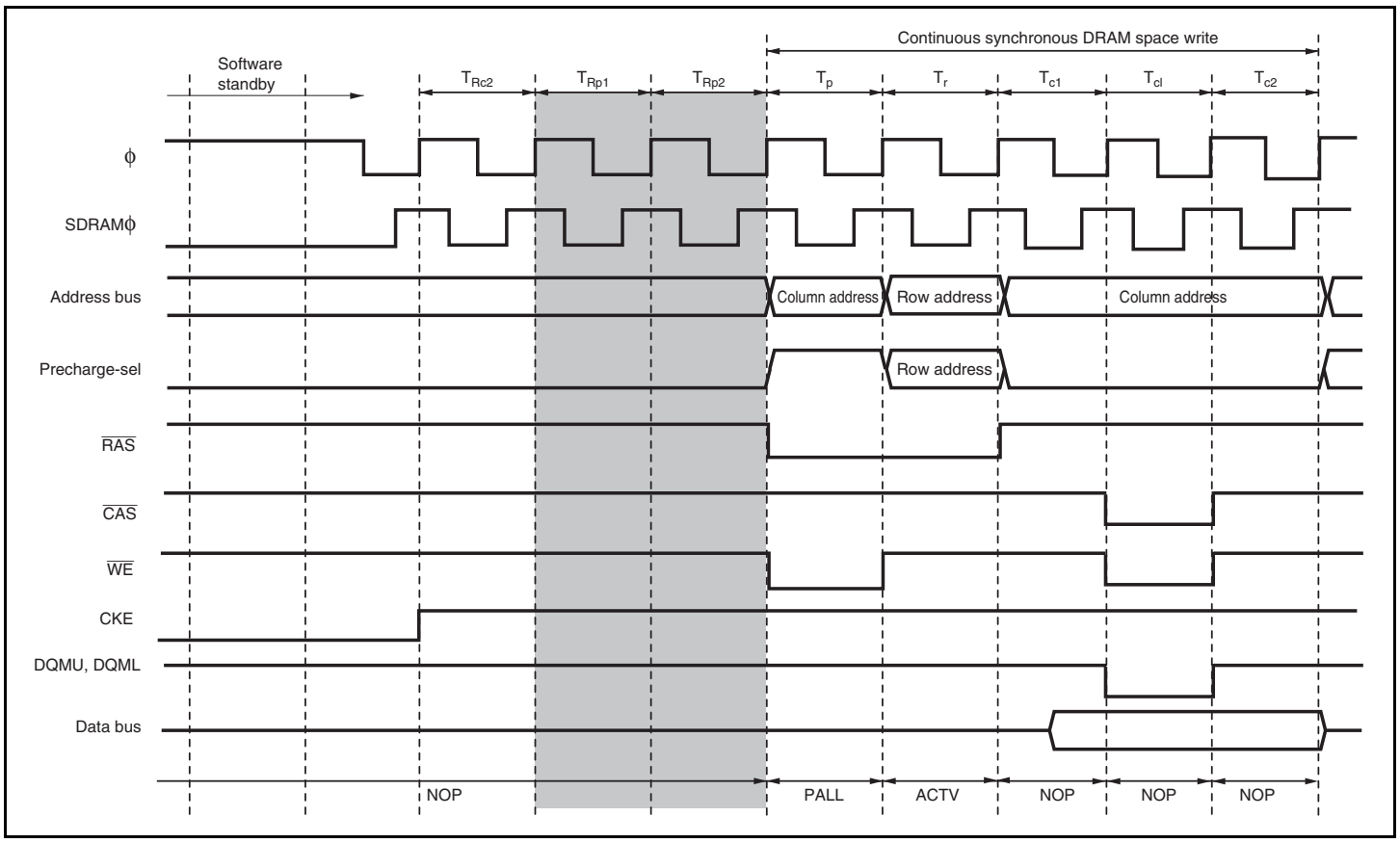


Figure 6.70 Example of Timing when Precharge Time after Self-Refreshing Is Extended by 2 States (TPCS2 to TPCS0 = H'2, TPC1 = 0, TPC0 = 0, CAS Latency 2)

(3) Refreshing and All-Module-Clocks-Stopped Mode

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

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

(4) Software Standby

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

6.8.14 Mode Register Setting of Synchronous DRAM

To use synchronous DRAM, mode must be set after power-on. To set mode, set the RMTS2 to RMTS0 bits in DRAMCR to H'5 and enable the synchronous DRAM mode register setting. After that, access the continuous synchronous DRAM space in bytes. When the value to be set in the synchronous DRAM mode register is X, value X is set in the synchronous DRAM mode register by writing to the continuous synchronous DRAM space of address H'400000 + X for 8-bit bus configuration synchronous DRAM and by writing to the continuous synchronous DRAM space of address H'400000 + 2X for 16-bit bus configuration synchronous DRAM.

The value of the address signal is fetched at the issuance time of the MRS command as the setting value of the mode register in the synchronous DRAM. Mode of burst read/burst write in the synchronous DRAM is not supported by this LSI. For setting the mode register of the synchronous DRAM, set the burst read/single write with the burst length of 1. Figure 6.71 shows the setting timing of the mode in the synchronous DRAM.

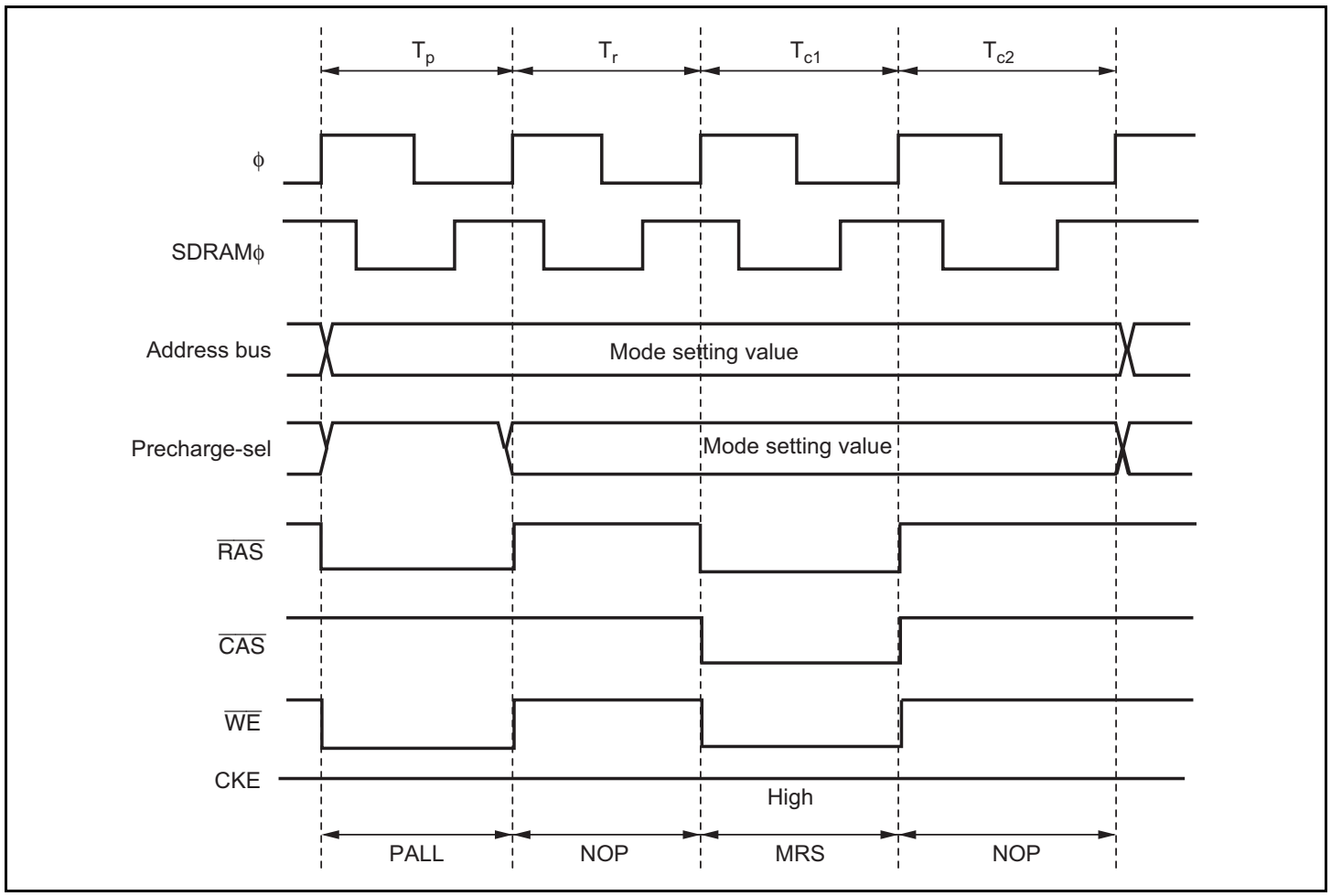


Figure 6.71 Synchronous DRAM Mode Setting Timing

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

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

(1) Output Timing of $\overline{\text{DACK}}$ or $\overline{\text{EDACK}}$

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

Figure 6.72 shows the $\overline{\text{DACK}}$ or $\overline{\text{EDACK}}$ output timing for the synchronous DRAM interface when DDS = 1 or EDDS = 1.

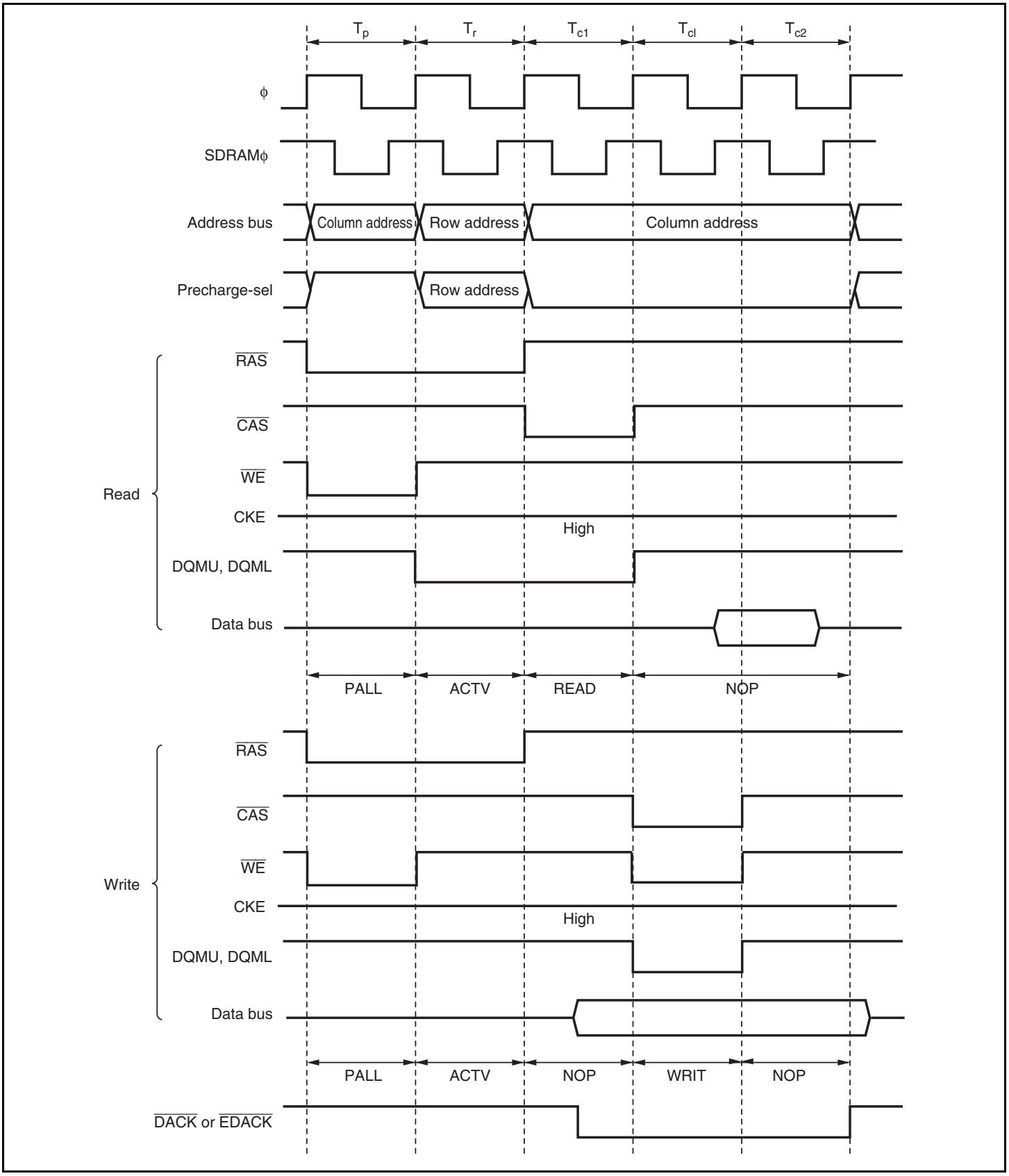


Figure 6.72 Example of $\overline{\text{DACK}}$/$\overline{\text{EDACK}}$ Output Timing when DDS = 1 or EDDS = 1

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

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

Figure 6.73 shows the $\overline{\text{DACK}}$ or $\overline{\text{EDACK}}$ output timing for connecting the synchronous DRAM interface when DDS = 0 or EDDS = 0.

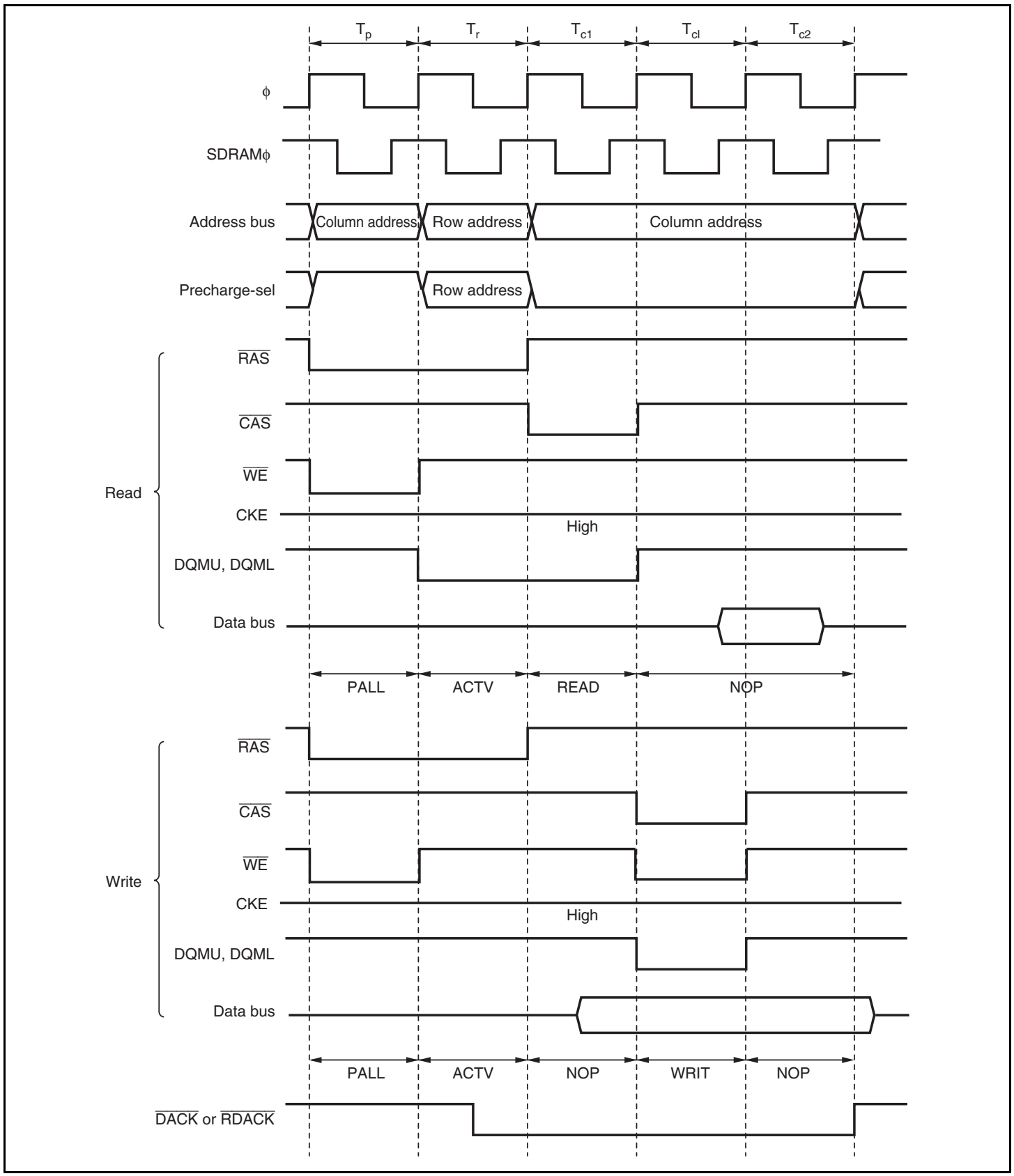


Figure 6.73 Example of $\overline{\text{DACK}}$/$\overline{\text{EDACK}}$ Output Timing when DDS = 0 or EDDS = 0

(2) Read Data Extension

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

Figure 6.74 shows the timing chart when the read data is extended by two cycles.

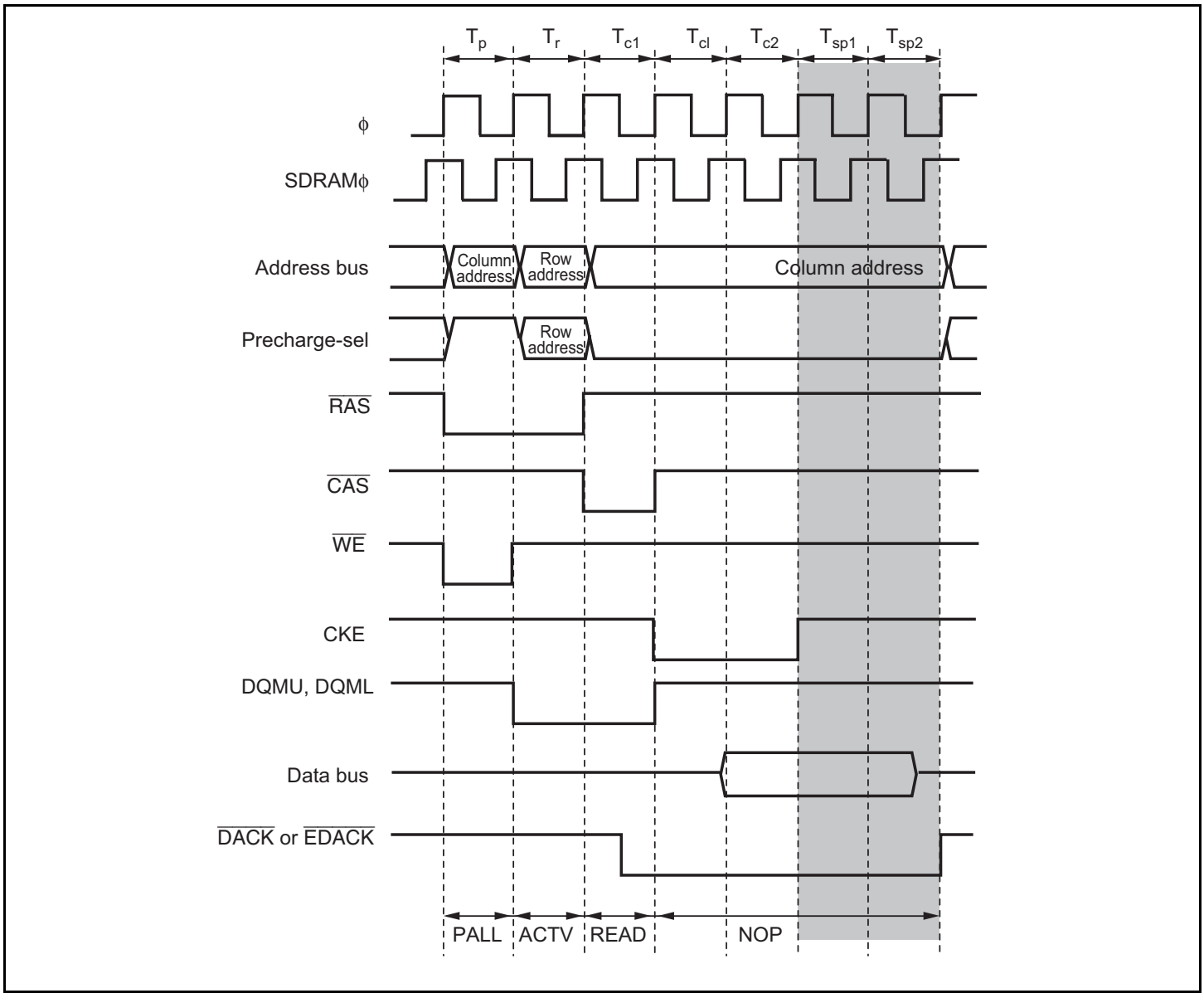


Figure 6.74 Example of Timing when the Read Data Is Extended by Two States (DDS = 1, or EDDS = 1, RDXC1 = 0, RDXC0 = 1, CAS Latency 2)

6.9 Burst ROM Interface

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

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

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

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

6.9.1 Basic Timing

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

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

The basic access timing for burst ROM space is shown in figures 6.75 and 6.76.

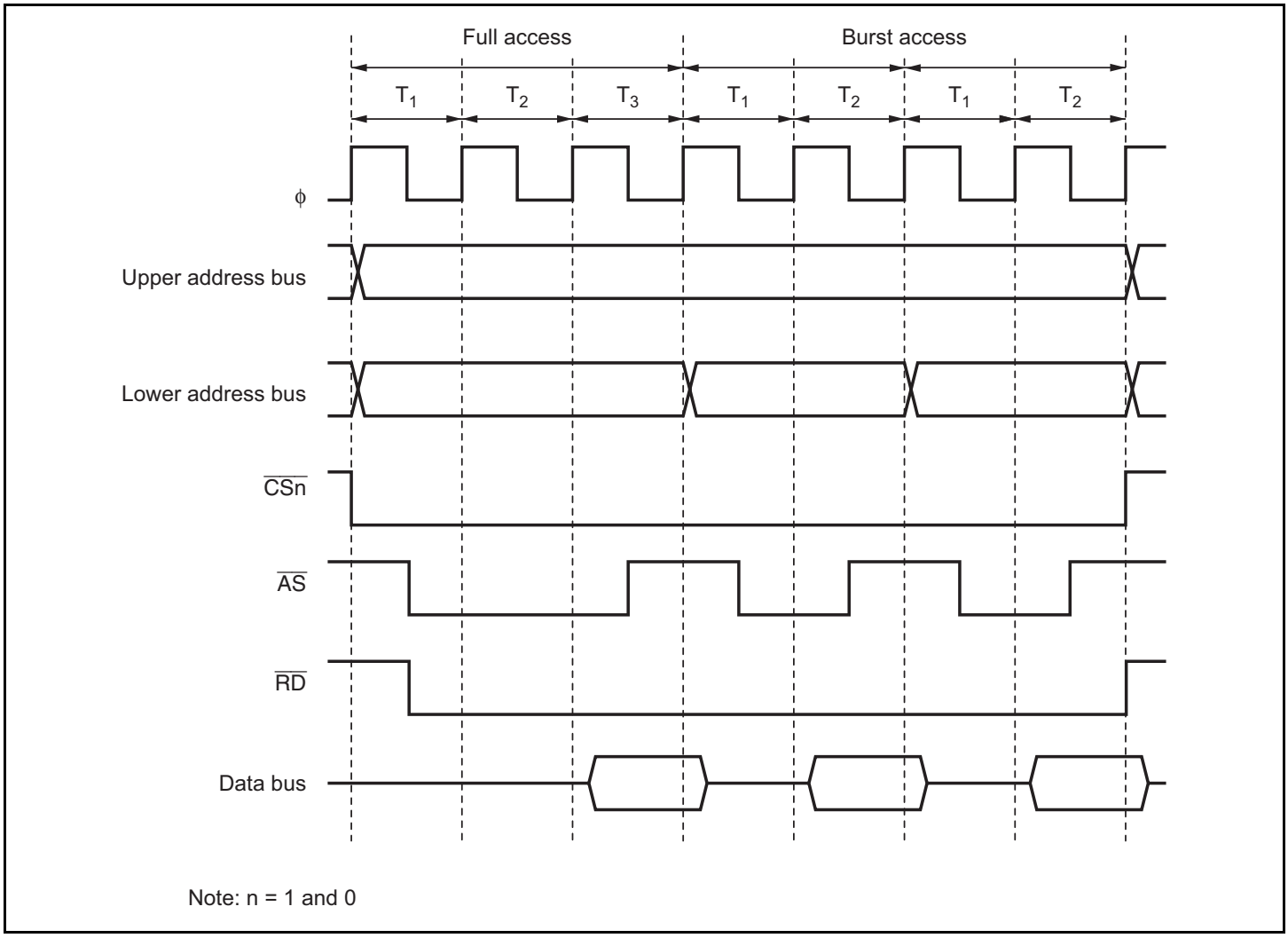


Figure 6.75 Example of Burst ROM Access Timing (ASTn = 1, 2-State Burst Cycle)

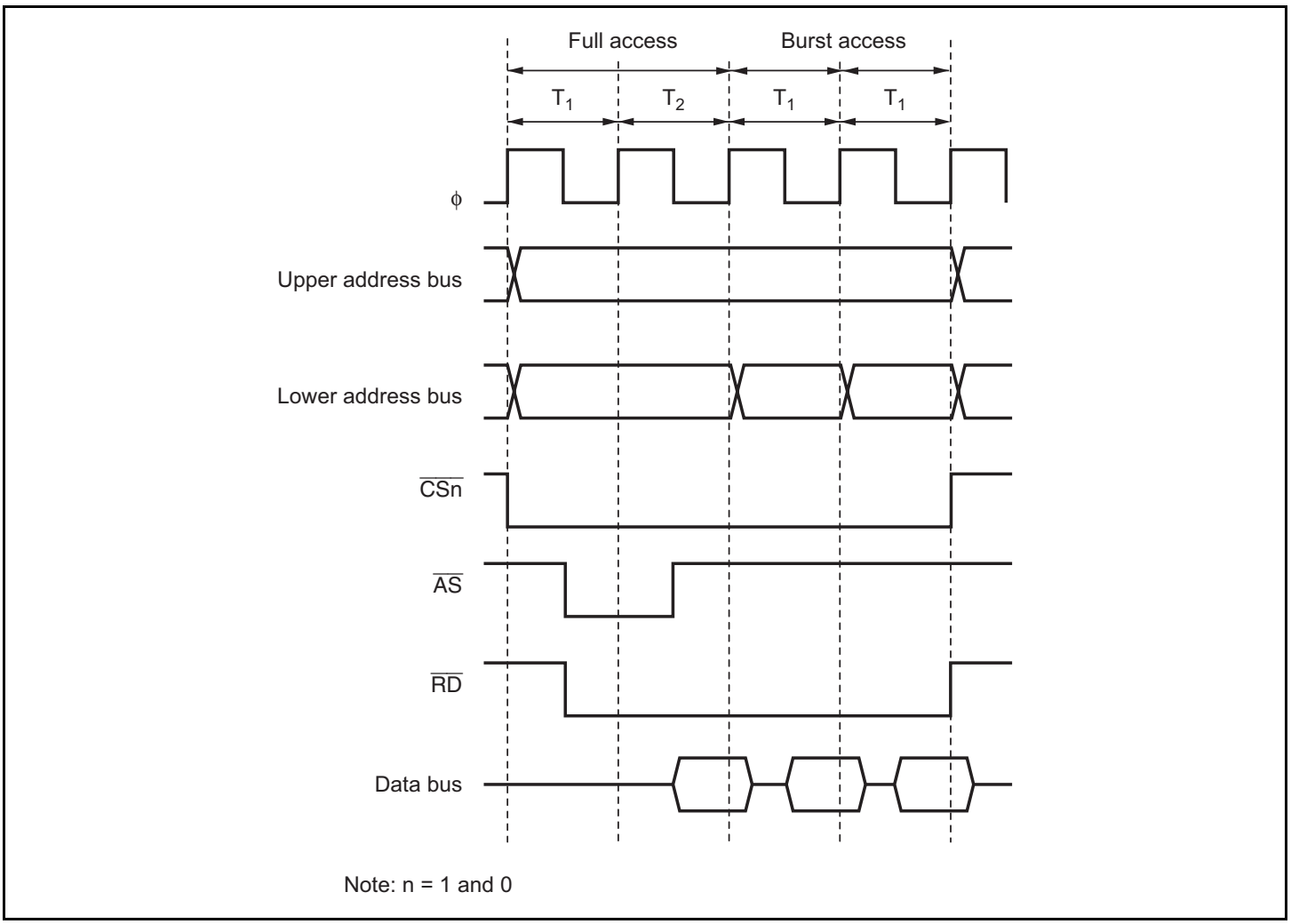


Figure 6.76 Example of Burst ROM Access Timing (ASTn = 0, 1-State Burst Cycle)

6.9.2 Wait Control

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

6.9.3 Write Access

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

6.10 Idle Cycle

6.10.1 Operation

When this LSI accesses external address space, it can insert an idle cycle (T_i) between bus cycles in the following three cases: (1) when read accesses in different areas occur consecutively, (2) when a write cycle occurs immediately after a read cycle, and (3) when a read cycle occurs immediately after a write cycle. Insertion of a 1-state or 2-state idle cycle can be selected with the IDLC bit in BCR. By inserting an idle cycle it is possible, for example, to avoid data collisions between ROM, etc., with a long output floating time, and high-speed memory, I/O interfaces, and so on.

(1) Consecutive Reads in Different Areas

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

Figure 6.77 shows an example of the operation in this case. In this example, bus cycle A is a read cycle for ROM with a long output floating time, and bus cycle B is a read cycle for SRAM, each being located in a different area. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is inserted, and a data collision is prevented.

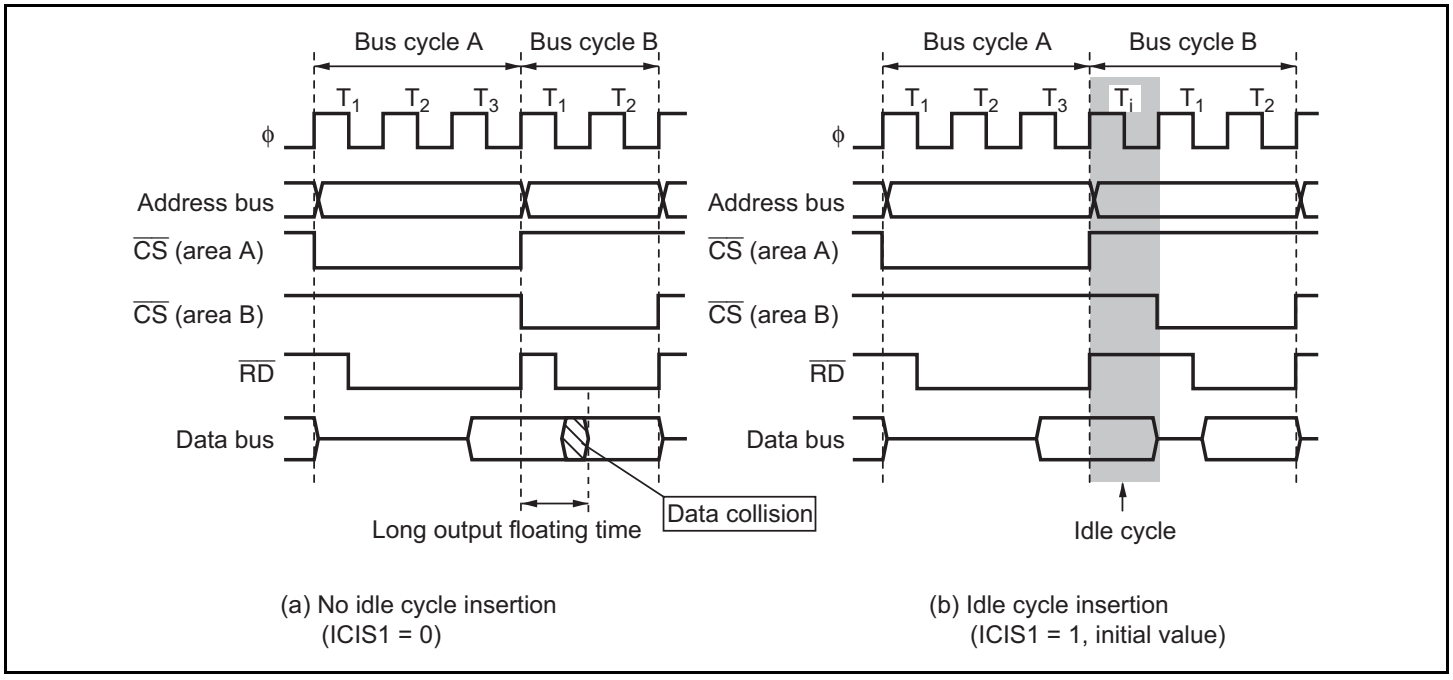


Figure 6.77 Example of Idle Cycle Operation (Consecutive Reads in Different Areas)

(2) Write after Read

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

Figure 6.78 shows an example of the operation in this case. In this example, bus cycle A is a read cycle for ROM with a long output floating time, and bus cycle B is a CPU write cycle. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and the CPU write data. In (b), an idle cycle is inserted, and a data collision is prevented.

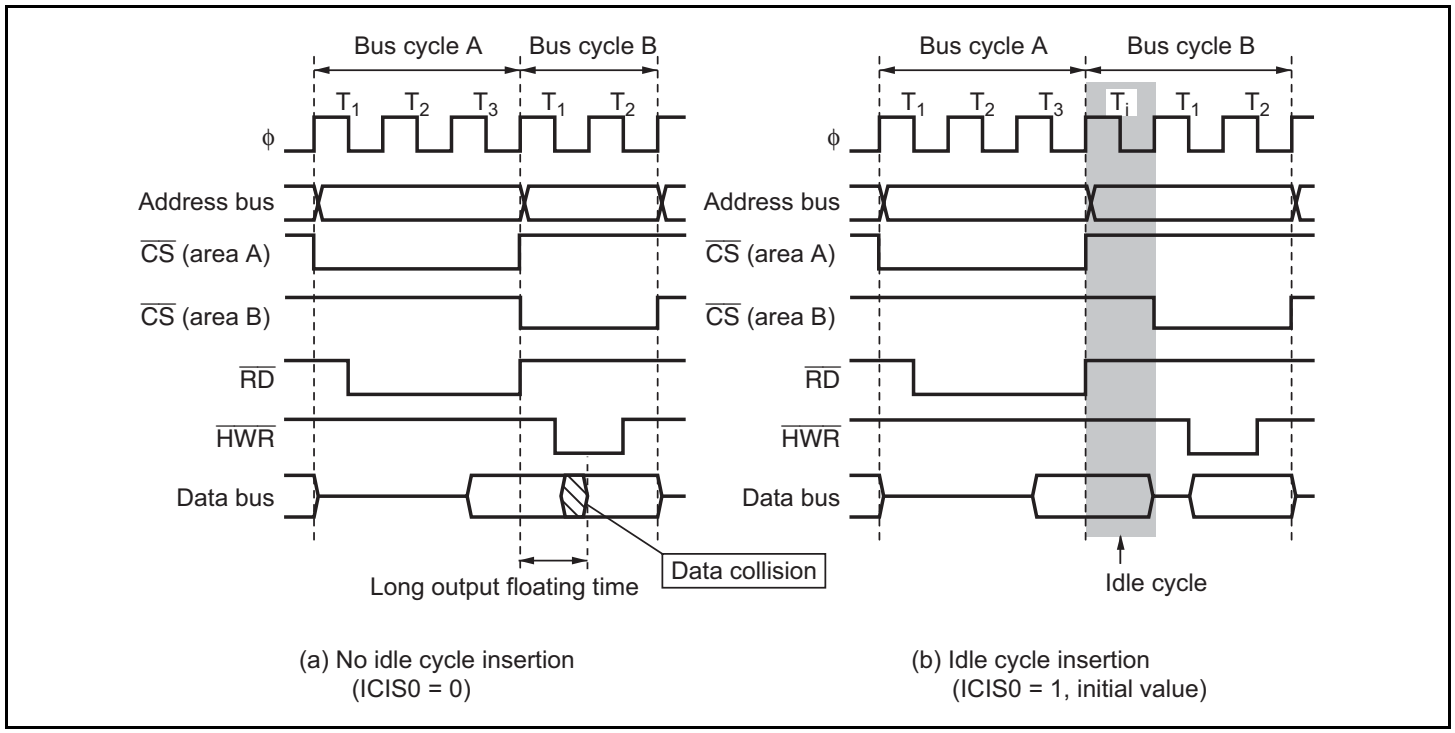


Figure 6.78 Example of Idle Cycle Operation (Write after Read)

(3) Read after Write

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

Figure 6.79 shows an example of the operation in this case. In this example, bus cycle A is a CPU write cycle and bus cycle B is a read cycle from an external device. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the CPU write data and read data from an external device. In (b), an idle cycle is inserted, and a data collision is prevented.

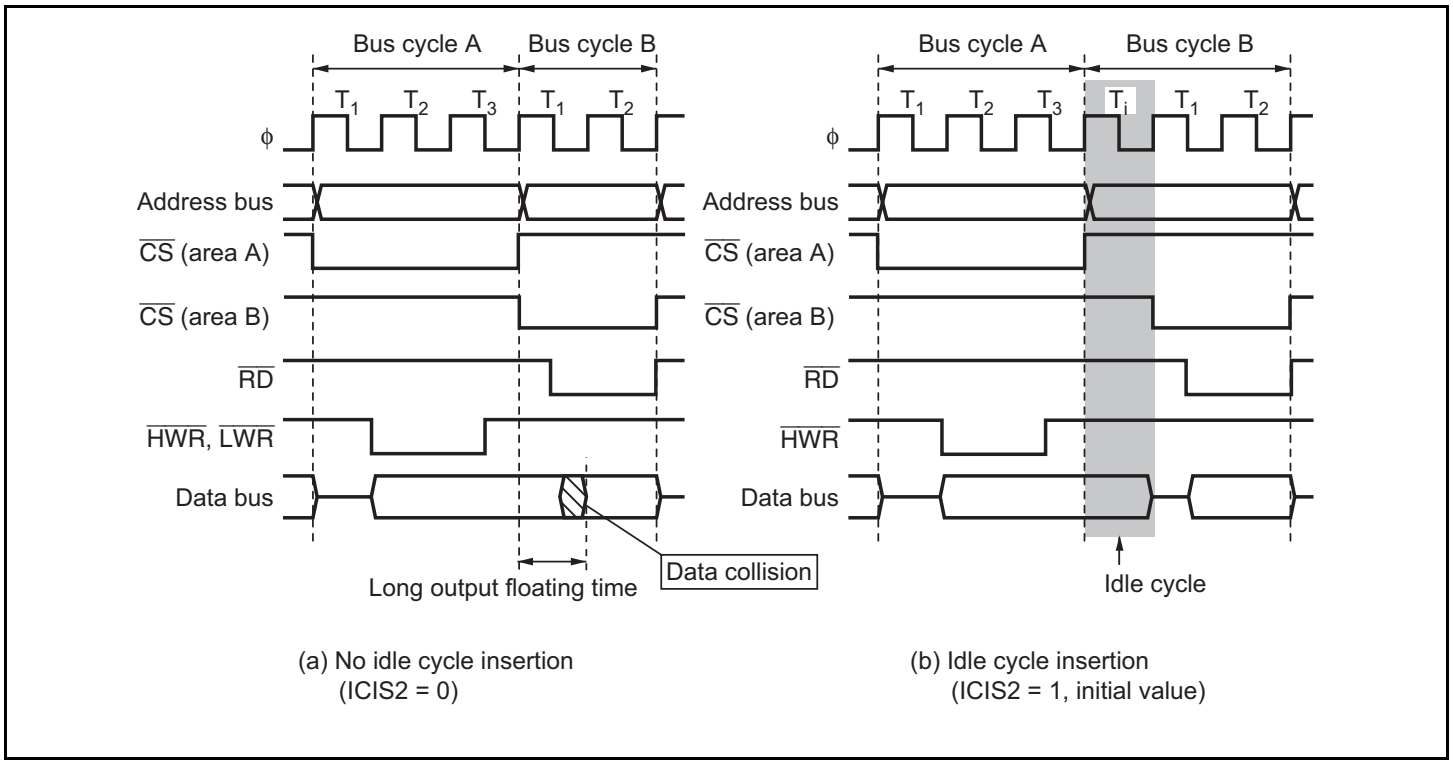


Figure 6.79 Example of Idle Cycle Operation (Read after Write)

(4) Relationship between Chip Select ($\overline{CS}$) Signal and Read ($\overline{RD}$) Signal

Depending on the system's load conditions, the $\overline{RD}$ signal may lag behind the $\overline{CS}$ signal. An example is shown in figure 6.80. In this case, with the setting for no idle cycle insertion (a), there may be a period of overlap between the bus cycle A $\overline{RD}$ signal and the bus cycle B $\overline{CS}$ signal. Setting idle cycle insertion, as in (b), however, will prevent any overlap between the $\overline{RD}$ and $\overline{CS}$ signals. In the initial state after reset release, idle cycle insertion (b) is set.

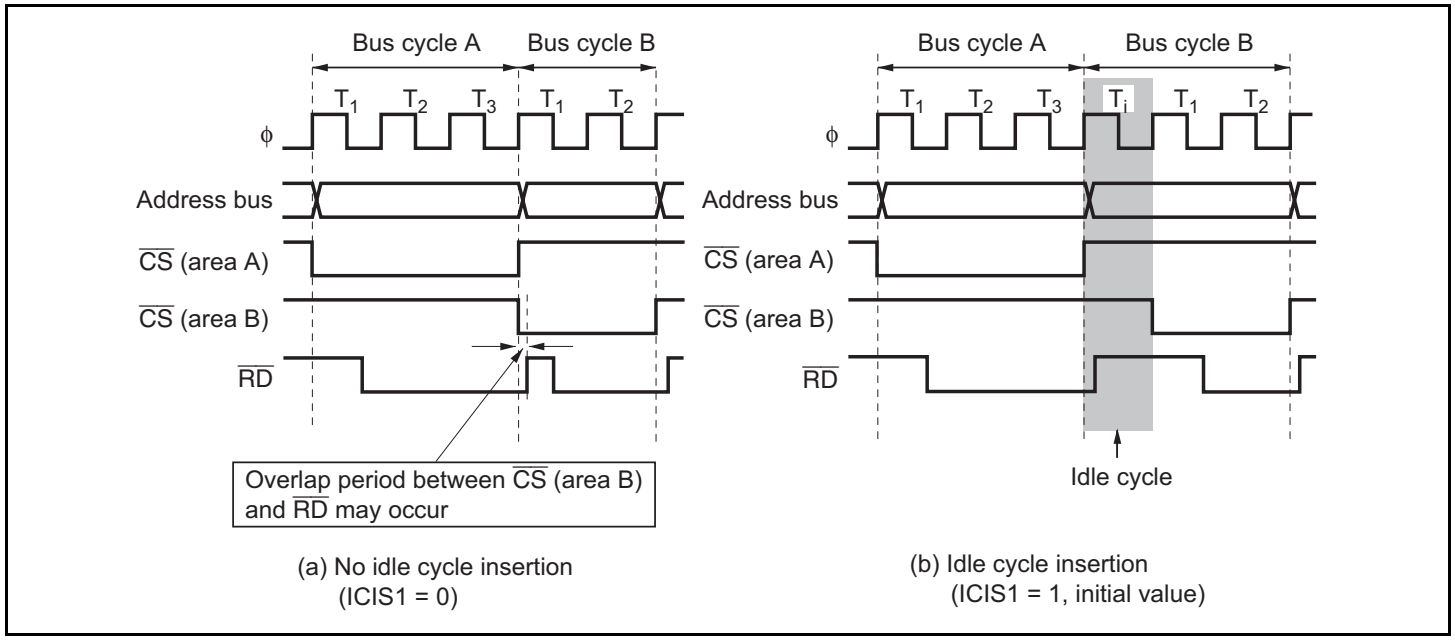


Figure 6.80 Relationship between Chip Select ($\overline{CS}$) and Read ($\overline{RD}$)

(5) Idle Cycle in Case of DRAM Space Access after Normal Space Access

In a DRAM space access following a normal space access, the settings of bits ICIS2, ICIS1, ICIS0, and IDLC in BCR are valid. However, in the case of consecutive reads in different areas, for example, if the second read is a full access to DRAM space, only a T_p cycle is inserted, and a T_i cycle is not. The timing in this case is shown in figure 6.81.

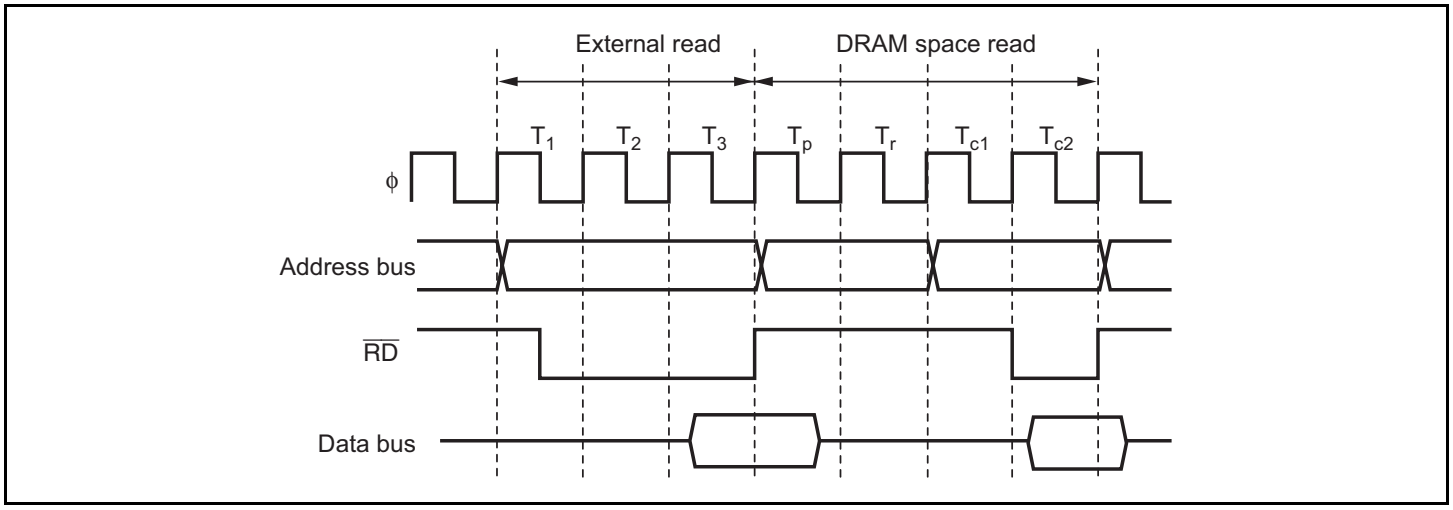


Figure 6.81 Example of DRAM Full Access after External Read (CAST = 0)

In burst access in RAS down mode, the settings of bits ICIS2, ICIS1, ICIS0, and IDLC are valid and an idle cycle is inserted. The timing in this case is illustrated in figures 6.82 and 6.83.

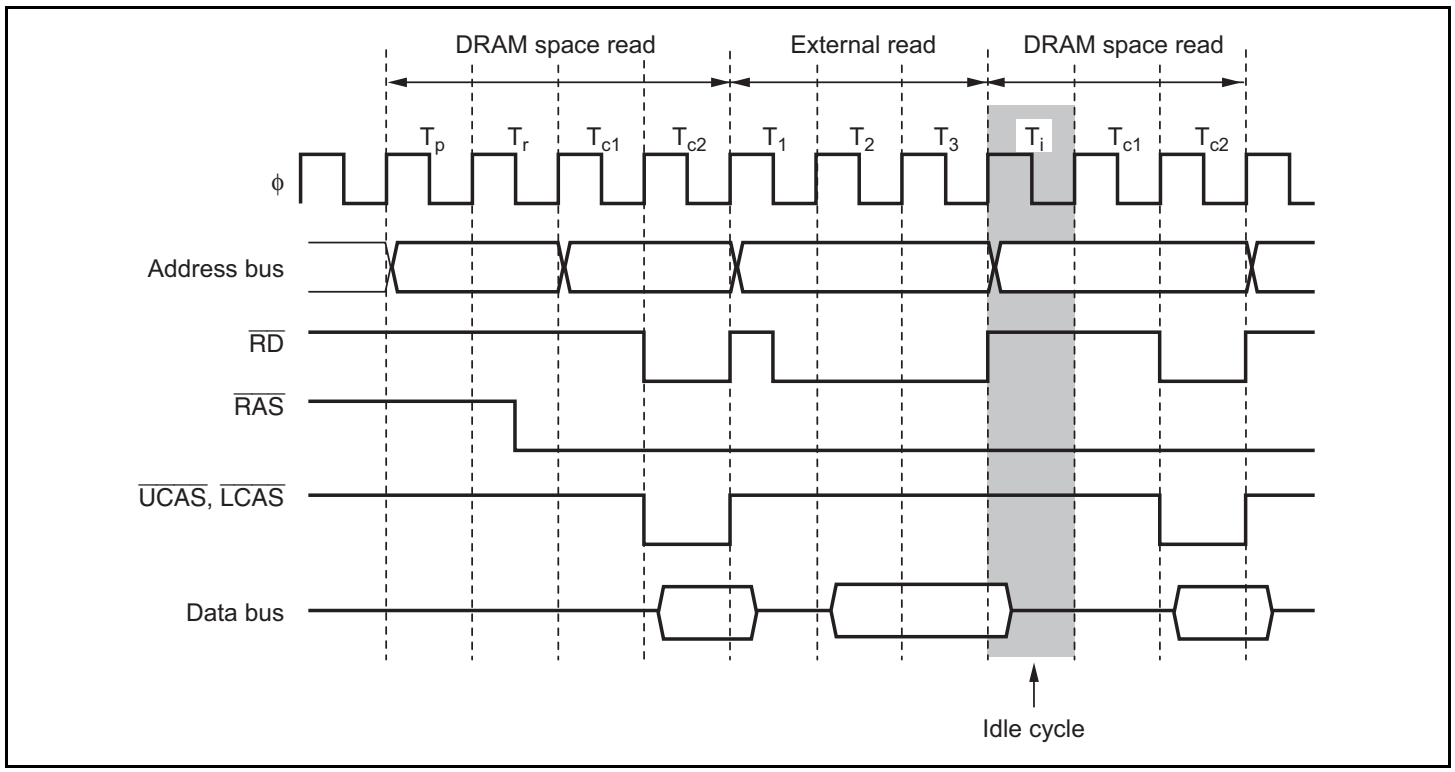


Figure 6.82 Example of Idle Cycle Operation in RAS Down Mode (Consecutive Reads in Different Areas) (IDLC = 0, RAST = 0, CAST = 0)

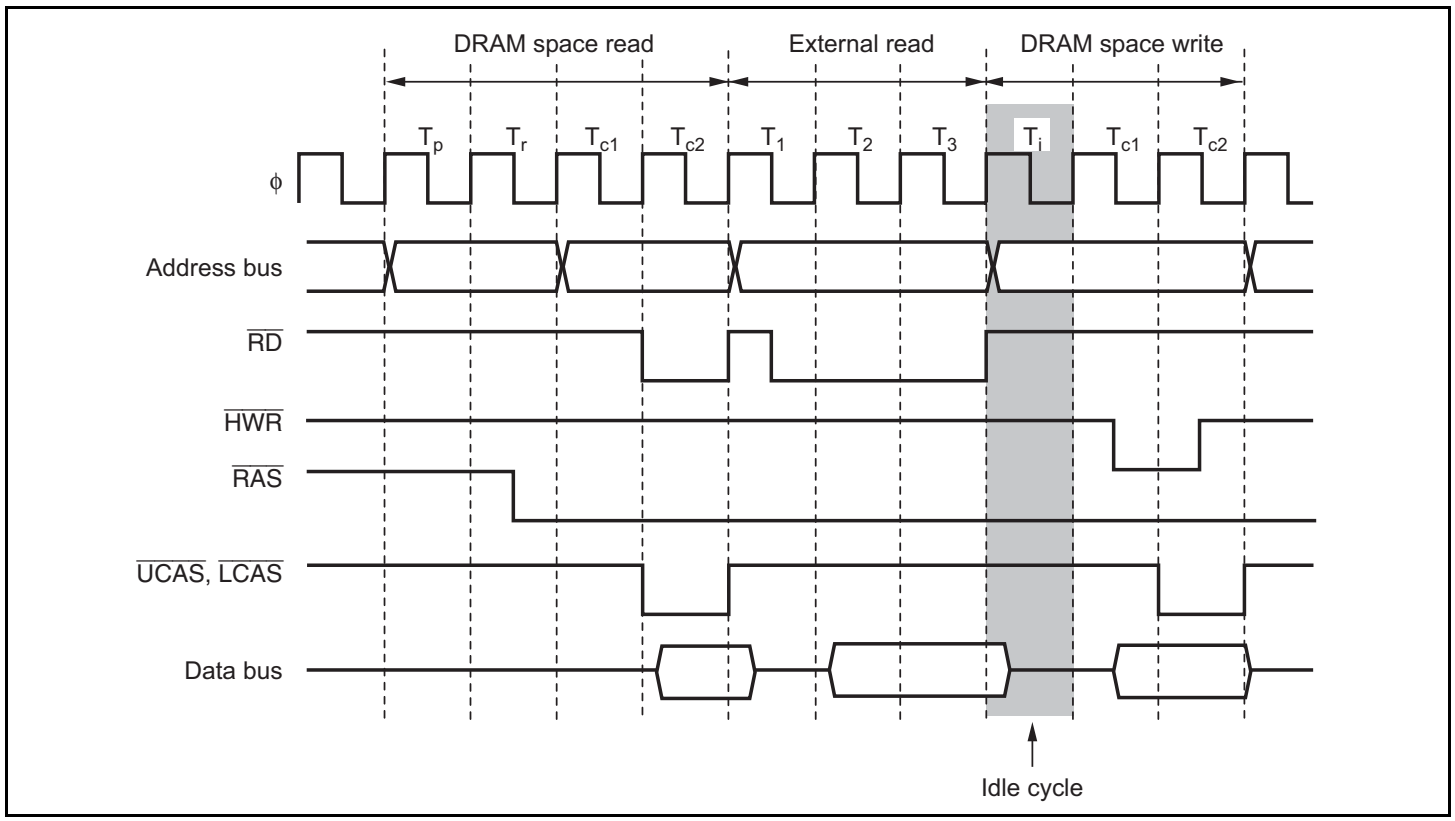


Figure 6.83 Example of Idle Cycle Operation in RAS Down Mode (Write after Read) (IDLC = 0, RAST = 0, CAST = 0)

(6) Idle Cycle in Case of Continuous Synchronous DRAM Space Access after Normal Space Access

In a continuous synchronous DRAM space access following a normal space access, the settings of bits ICIS2, ICIS1, ICIS0, and IDLC in BCR are valid. However, in the case of consecutive reads in different areas, for example, if the second read is a full access to continuous synchronous DRAM space, only Tp cycle is inserted, and Ti cycle is not. The timing in this case is shown in figure 6.84.

Note: The synchronous DRAM interface is not supported by the H8S/2456 Group and H8S/2454 Group.

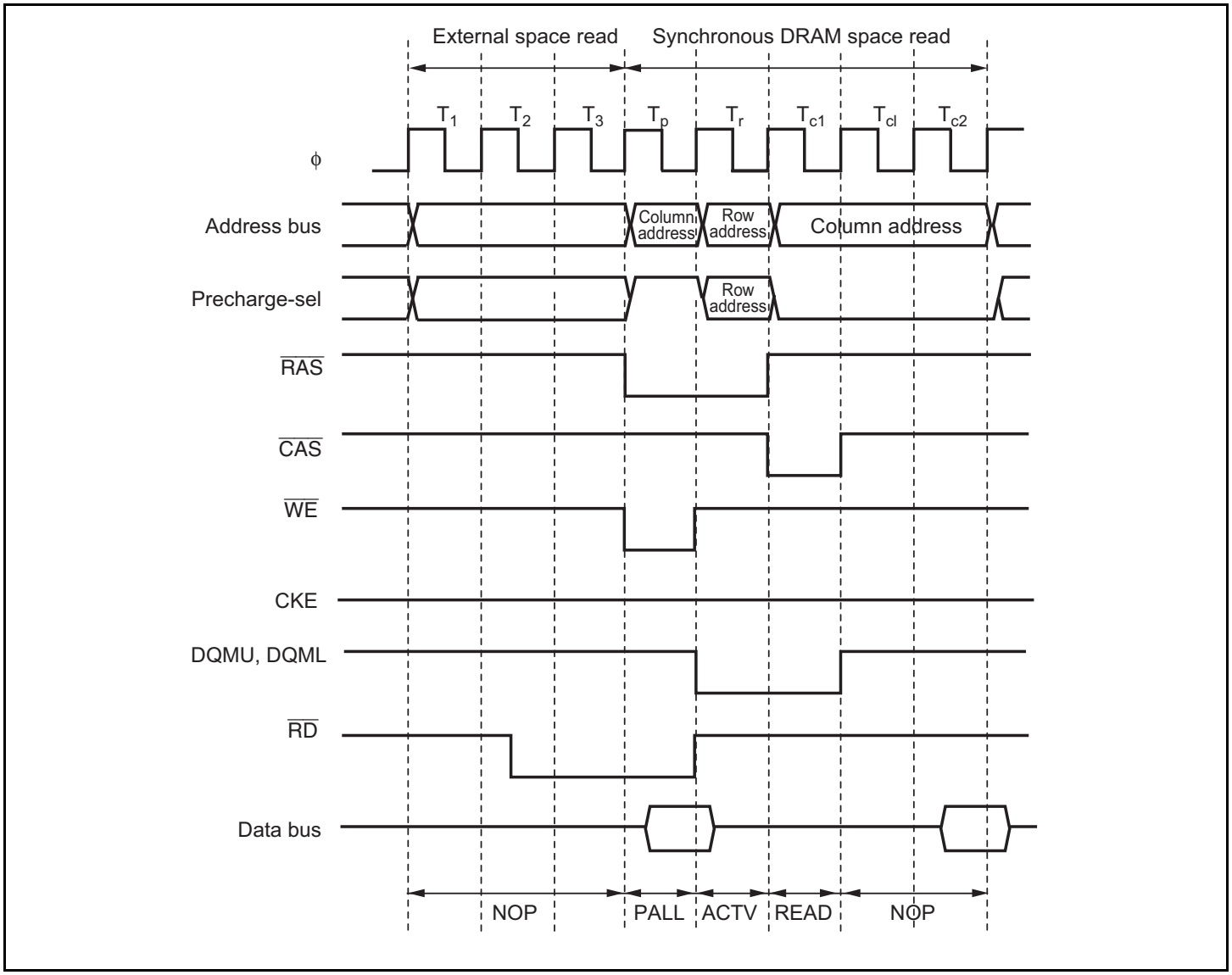


Figure 6.84 Example of Synchronous DRAM Full Access after External Read (CAS Latency 2)

In burst access in RAS down mode, the settings of bits ICIS2, ICIS1, ICIS0, and IDLC are valid and an idle cycle is inserted. However, in read access, note that the timings of DQMU and DQML differ according to the settings of the IDLC bit. The timing in this case is illustrated in figures 6.85 and 6.86. In write access, DQMU and DQML are not in accordance with the settings of the IDLC bit. The timing in this case is illustrated in figure 6.87.

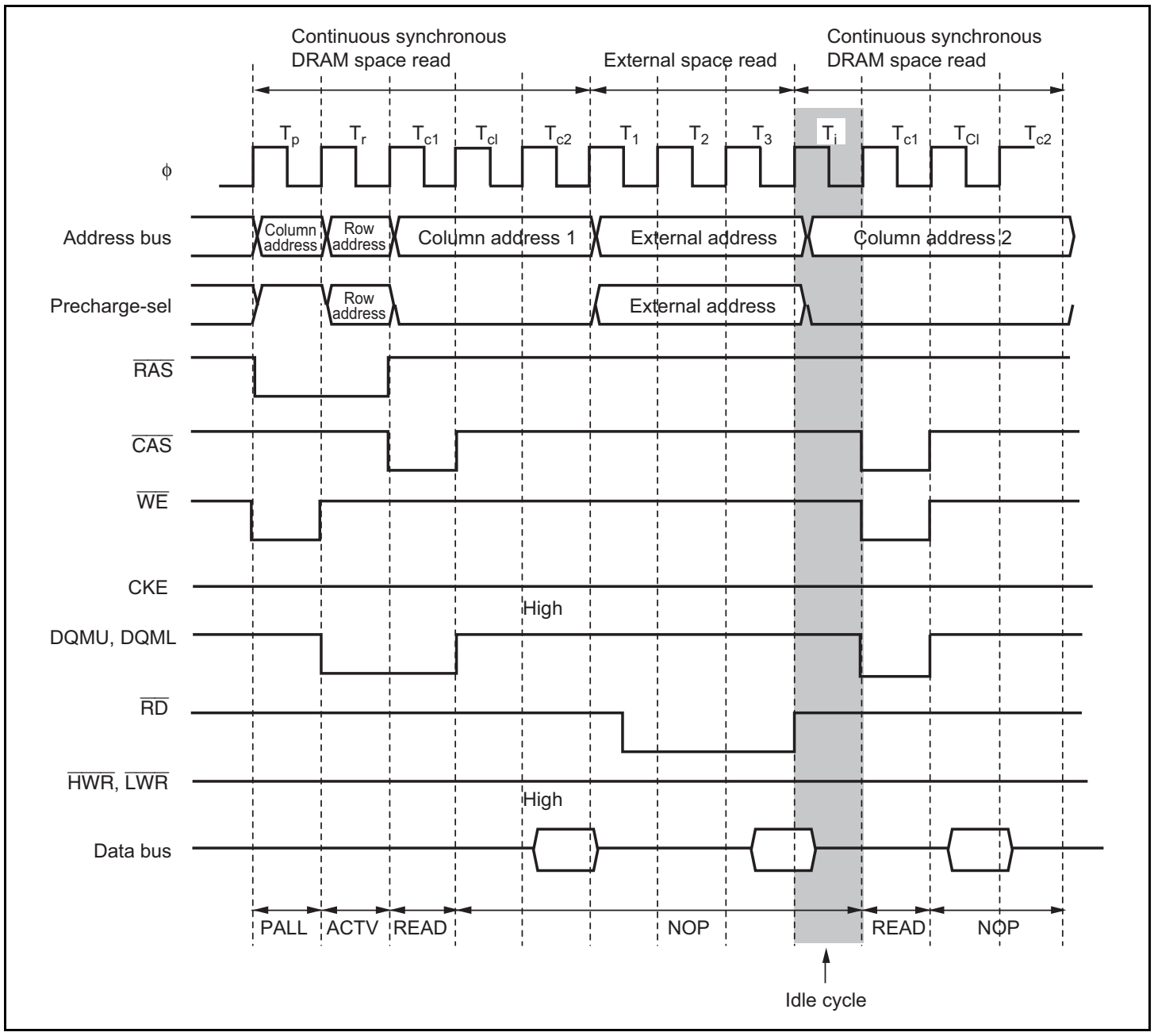


Figure 6.85 Example of Idle Cycle Operation in RAS Down Mode (Read in Different Area) (IDLC = 0, CAS Latency 2)

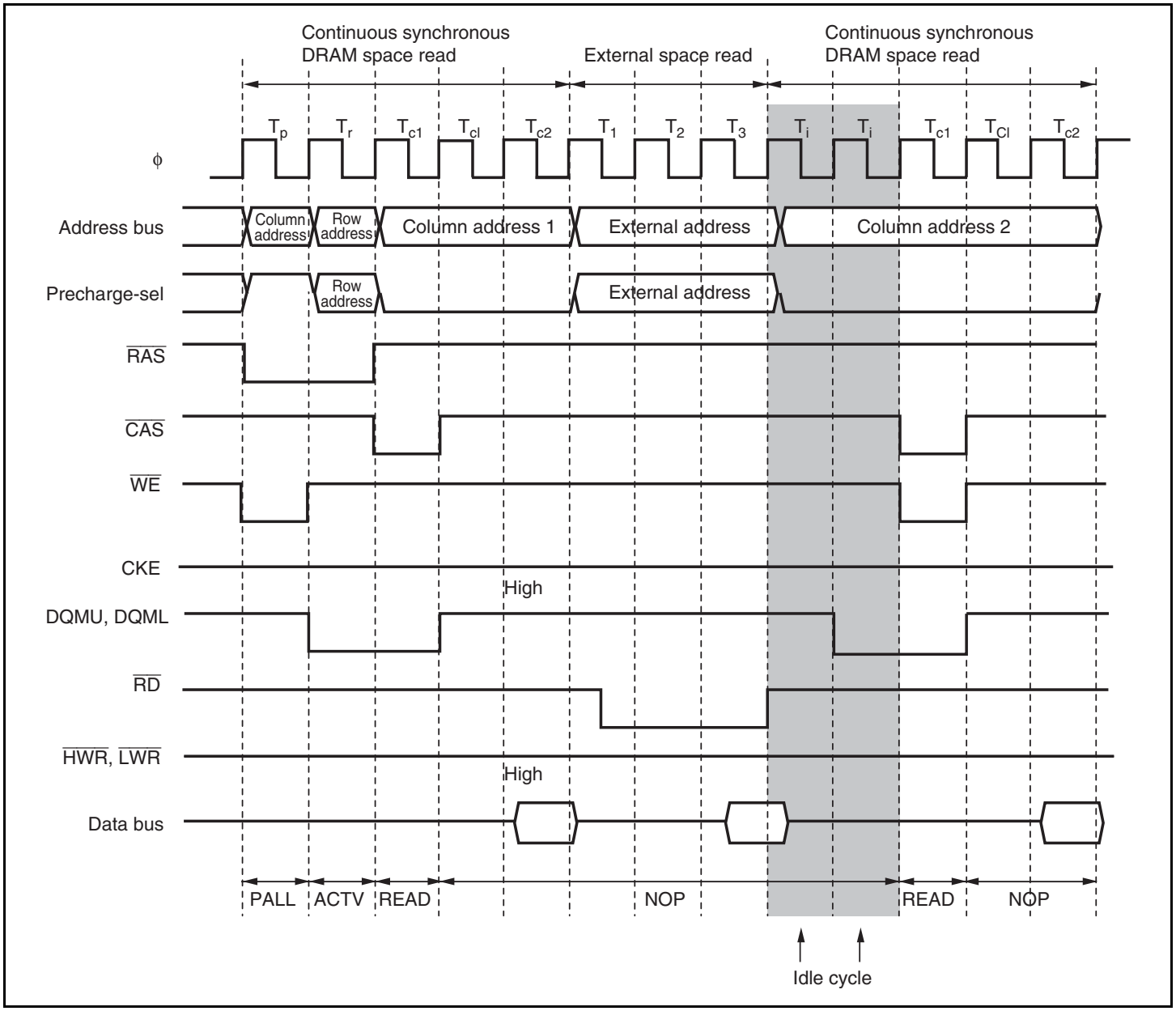


Figure 6.86 Example of Idle Cycle Operation in RAS Down Mode (Read in Different Area) (IDLC = 1, CAS Latency 2)

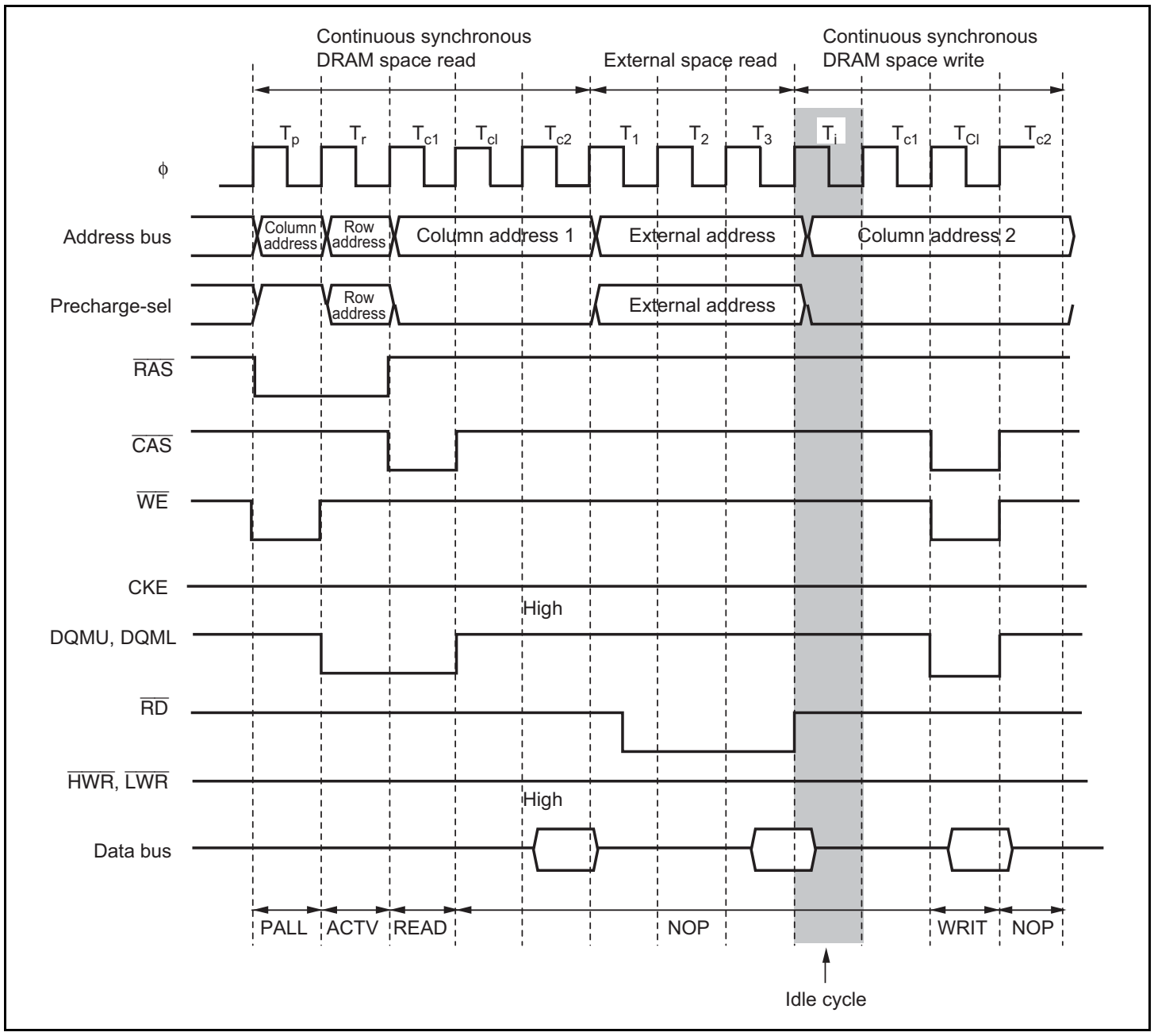


Figure 6.87 Example of Idle Cycle Operation in RAS Down Mode (Write after Read) (IDLC = 0, CAS Latency 2)

(7) Idle Cycle in Case of Normal Space Access after DRAM Space Access

(a) Normal space access after DRAM space read access

While the DRMI bit is cleared to 0 in DRACCR, idle cycle insertion after DRAM space access is disabled. Idle cycle insertion after DRAM space access can be enabled by setting the DRMI bit to 1. The conditions and number of states of the idle cycle to be inserted are in accordance with the settings of bits ICIS1, ICIS0, and IDLC in BCR are valid. Figures 6.88 and 6.89 show examples of idle cycle operation when the DRMI bit is set to 1.

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

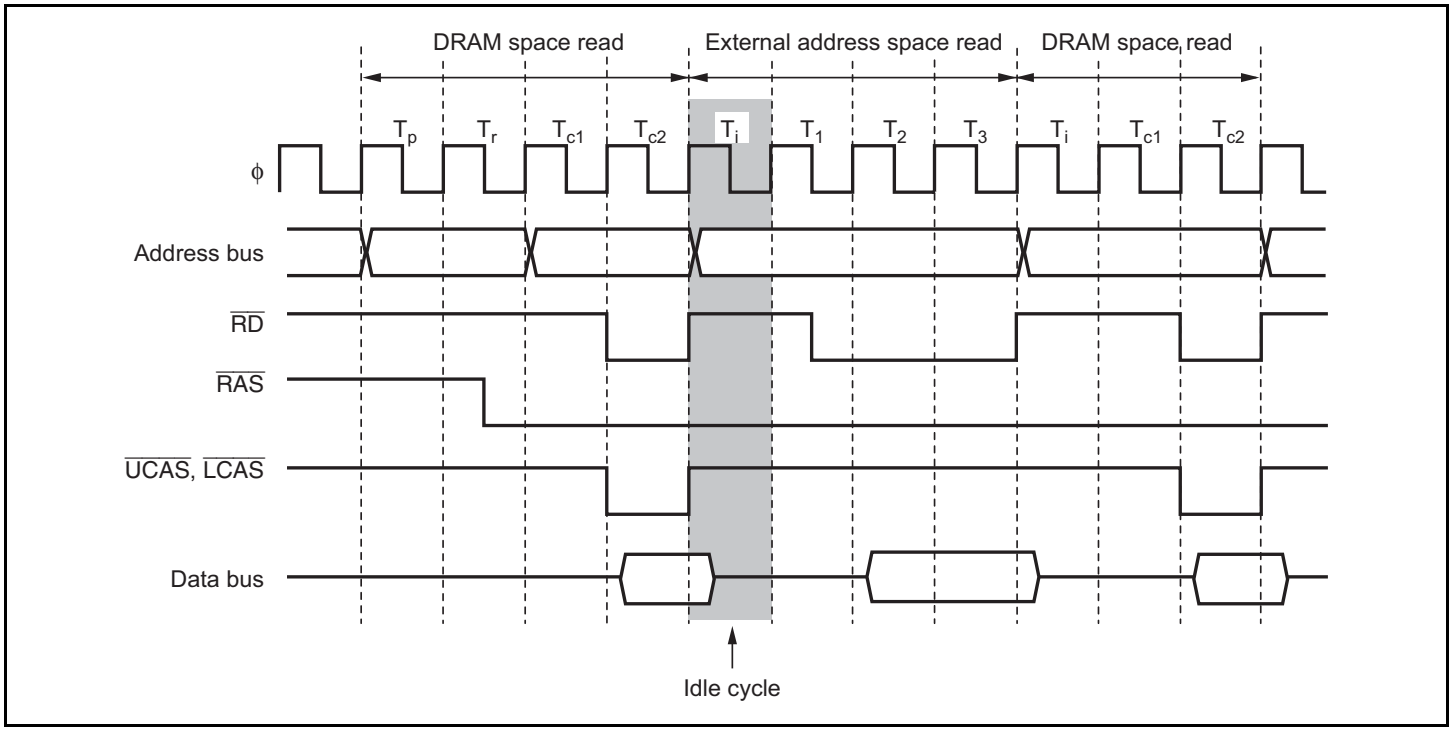


Figure 6.88 Example of Idle Cycle Operation after DRAM Access (Consecutive Reads in Different Areas) (IDLC = 0, RAST = 0, CAST = 0)

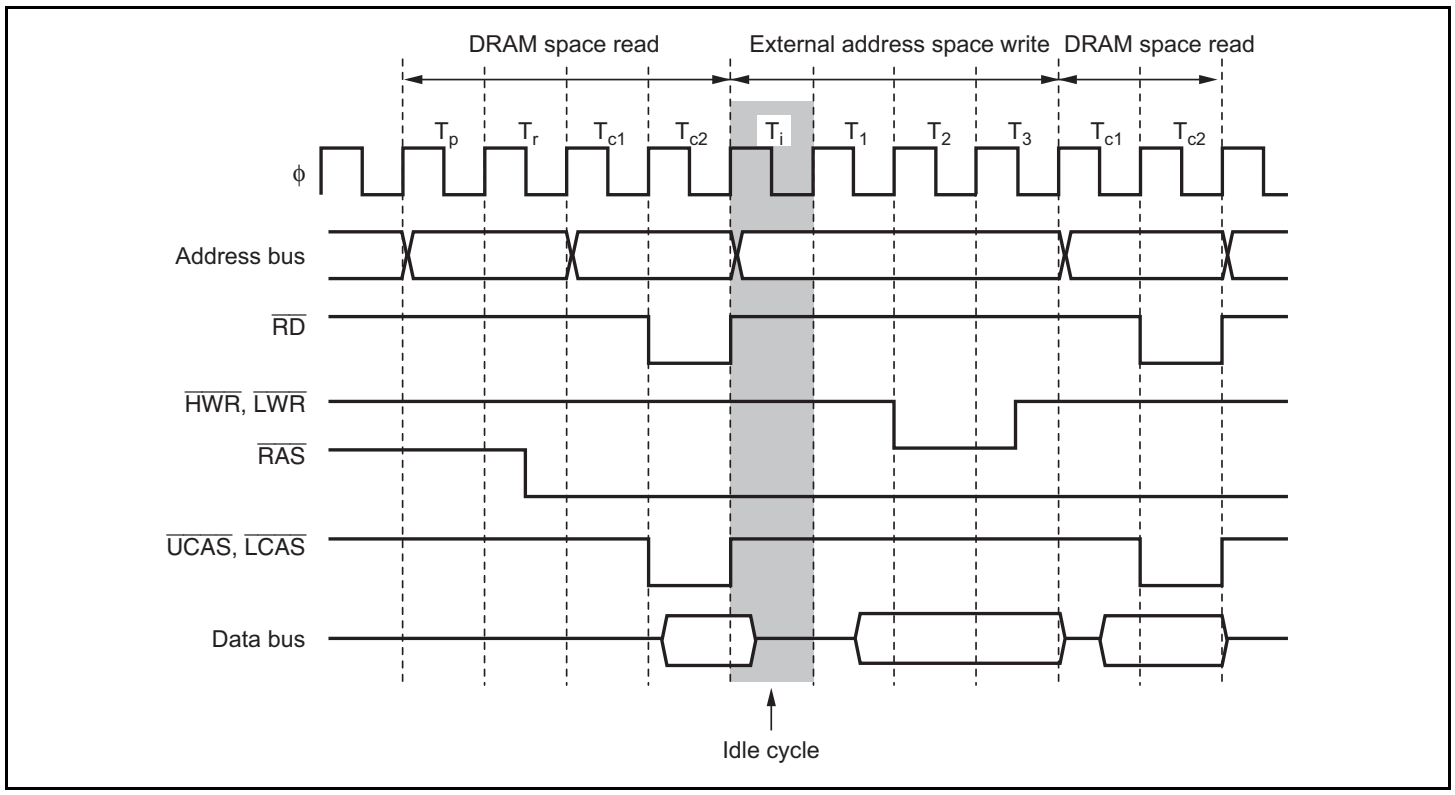


Figure 6.89 Example of Idle Cycle Operation after DRAM Access (Write after Read) (IDLC = 0, RAST = 0, CAST = 0)

(b) Normal space access after DRAM space write access

While the ICIS2 bit is set to 1 in BCR and a normal space read access occurs after DRAM space write access, idle cycle is inserted in the first read cycle. The number of states of the idle cycle to be inserted is in accordance with the setting of the IDLC bit. It does not depend on the DRMI bit in DRACCR. Figure 6.90 shows an example of idle cycle operation when the ICIS2 bit is set to 1.

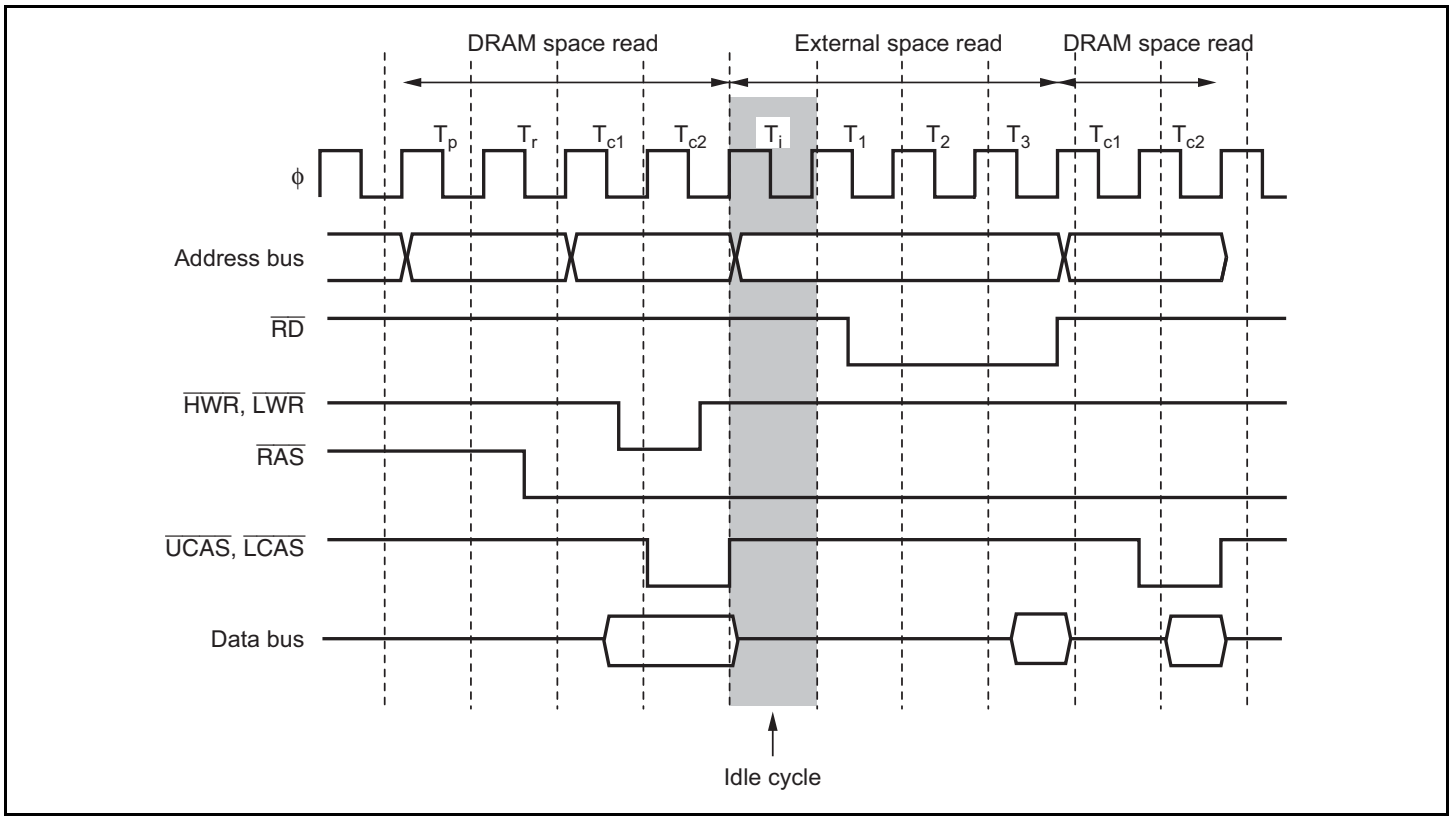


Figure 6.90 Example of Idle Cycle Operation after DRAM Write Access (IDLC = 0, ICIS1 = 0, RAST = 0, CAST = 0)

(8) Idle Cycle in Case of Normal Space Access after Continuous Synchronous DRAM Space Access:

Note: The synchronous DRAM interface is not supported by the H8S/2456 Group and H8S/2454 Group.

(a) Normal space access after a continuous synchronous DRAM space read access

While the DRMI bit is cleared to 0 in DRACCR, idle cycle insertion after continuous synchronous DRAM space read access is disabled. Idle cycle insertion after continuous synchronous DRAM space read access can be enabled by setting the DRMI bit to 1. The conditions and number of states of the idle cycle to be inserted are in accordance with the settings of bits ICIS1, ICIS0, and IDLC in RCR. Figure 6.91 shows an example of idle cycle operation when the DRMI bit is set to 1. When the DRMI bit is cleared to 0, an idle cycle is not inserted after continuous synchronous DRAM space read access even if bits ICIS1 and ICIS0 are set to 1.

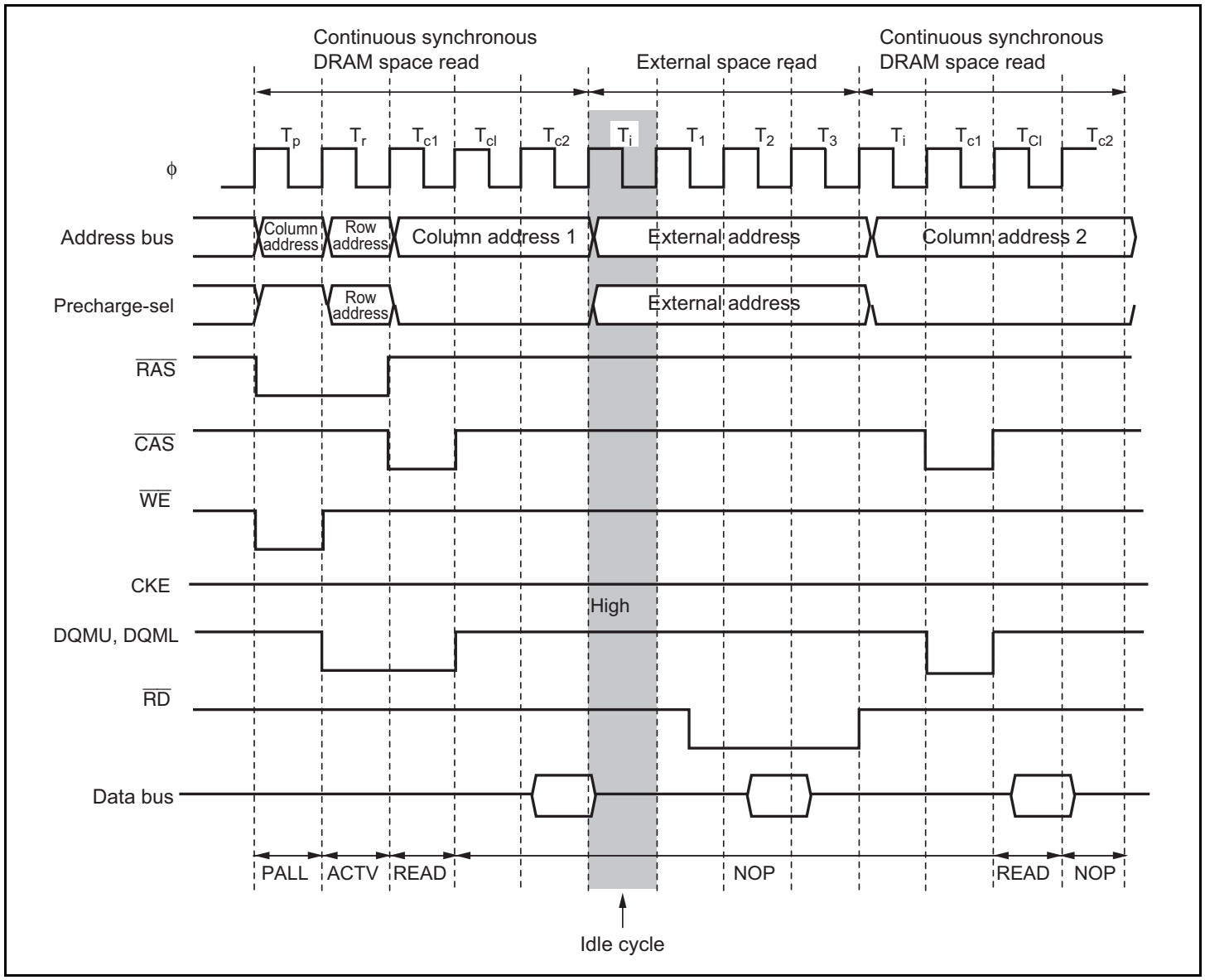


Figure 6.91 Example of Idle Cycle Operation after Continuous Synchronous DRAM Space Read Access (Read between Different Area) (IDLC = 0, CAS Latency 2)

(b) Normal space access after a continuous synchronous DRAM space write access

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

Figure 6.92 shows an example of idle cycle operation when the ICIS2 bit is set to 1.

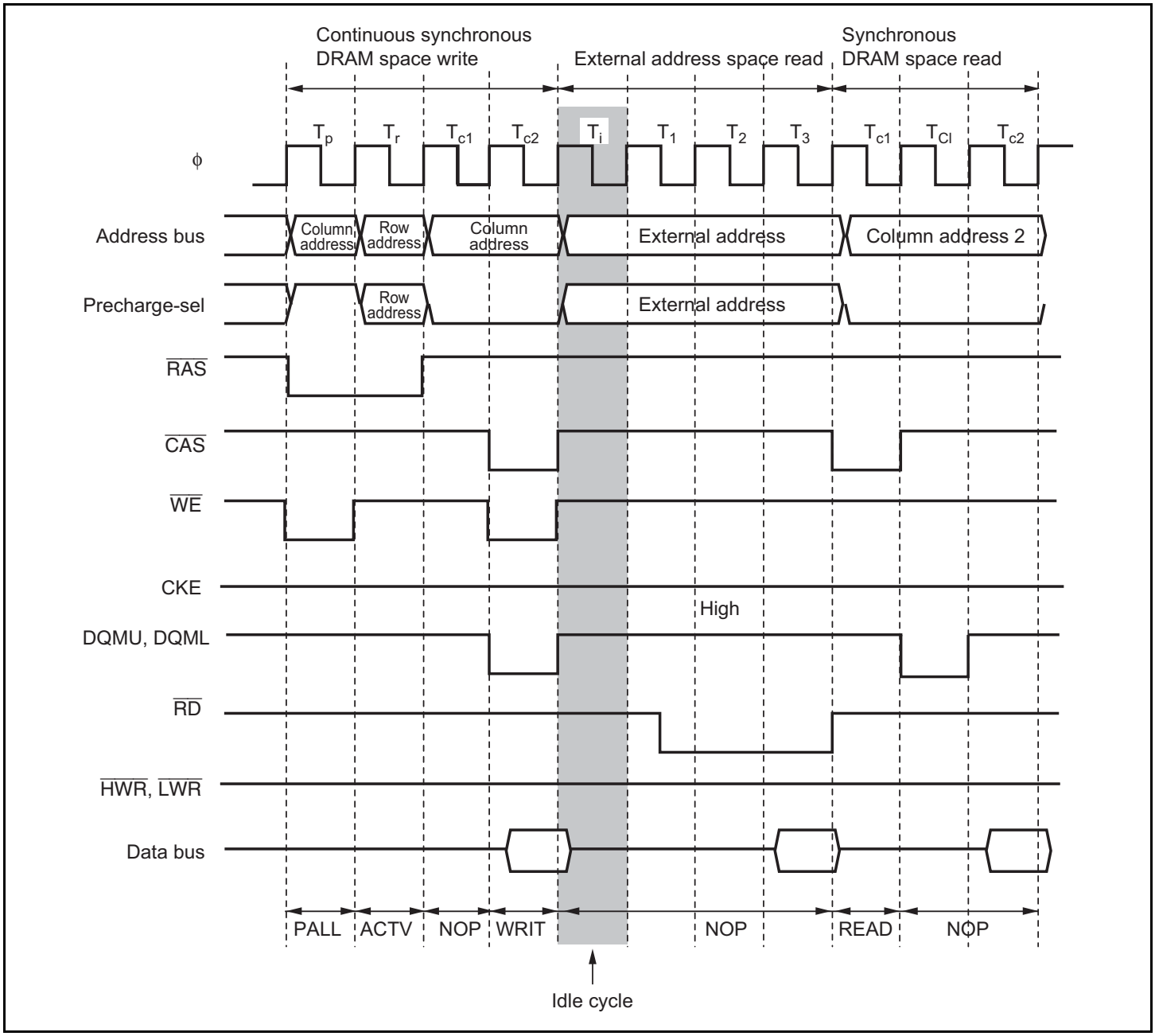


Figure 6.92 Example of Idle Cycle Operation after Continuous Synchronous DRAM Space Write Access (IDLC = 0, ICIS1 = 0, SDWCD = 1, CAS Latency 2)

Table 6.12 shows whether there is an idle cycle insertion or not in the case of mixed accesses to normal space and DRAM space/continuous synchronous DRAM space.

Table 6.12 Idle Cycles in Mixed Accesses to Normal Space and DRAM Continuous Synchronous DRAM Space

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

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

Note: * Not supported by the H8S/2456 Group and H8S/2454 Group.

Setting the DRMI bit in DRACCR to 1 enables an idle cycle to be inserted in the case of consecutive read and write operations in DRAM/continuous synchronous DRAM space burst access. Figures 6.93 and 6.94 show an example of the timing for idle cycle insertion in the case of consecutive read and write accesses to DRAM/continuous synchronous DRAM space.

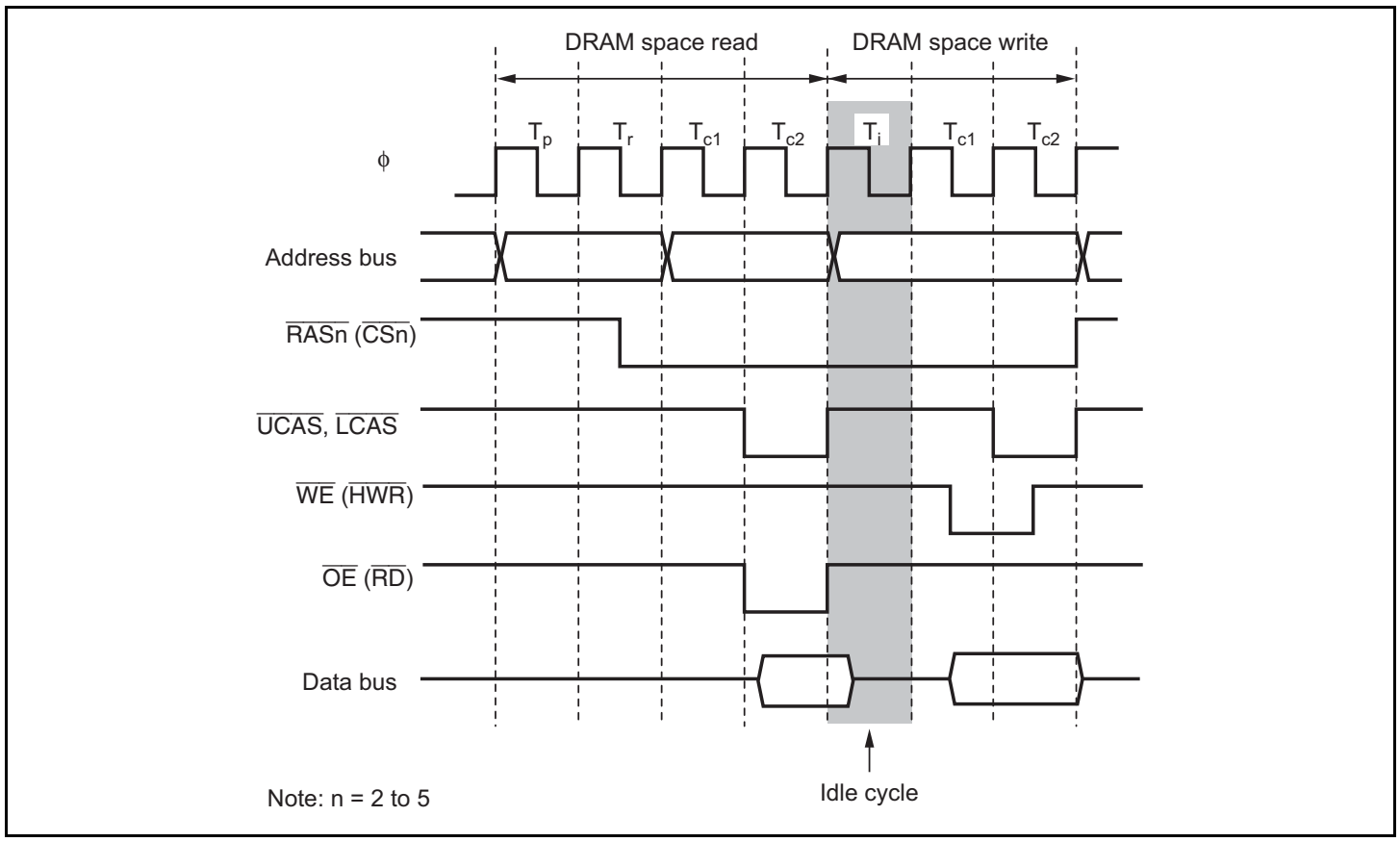


Figure 6.93 Example of Timing for Idle Cycle Insertion in Case of Consecutive Read and Write Accesses to DRAM Space in RAS Down Mode

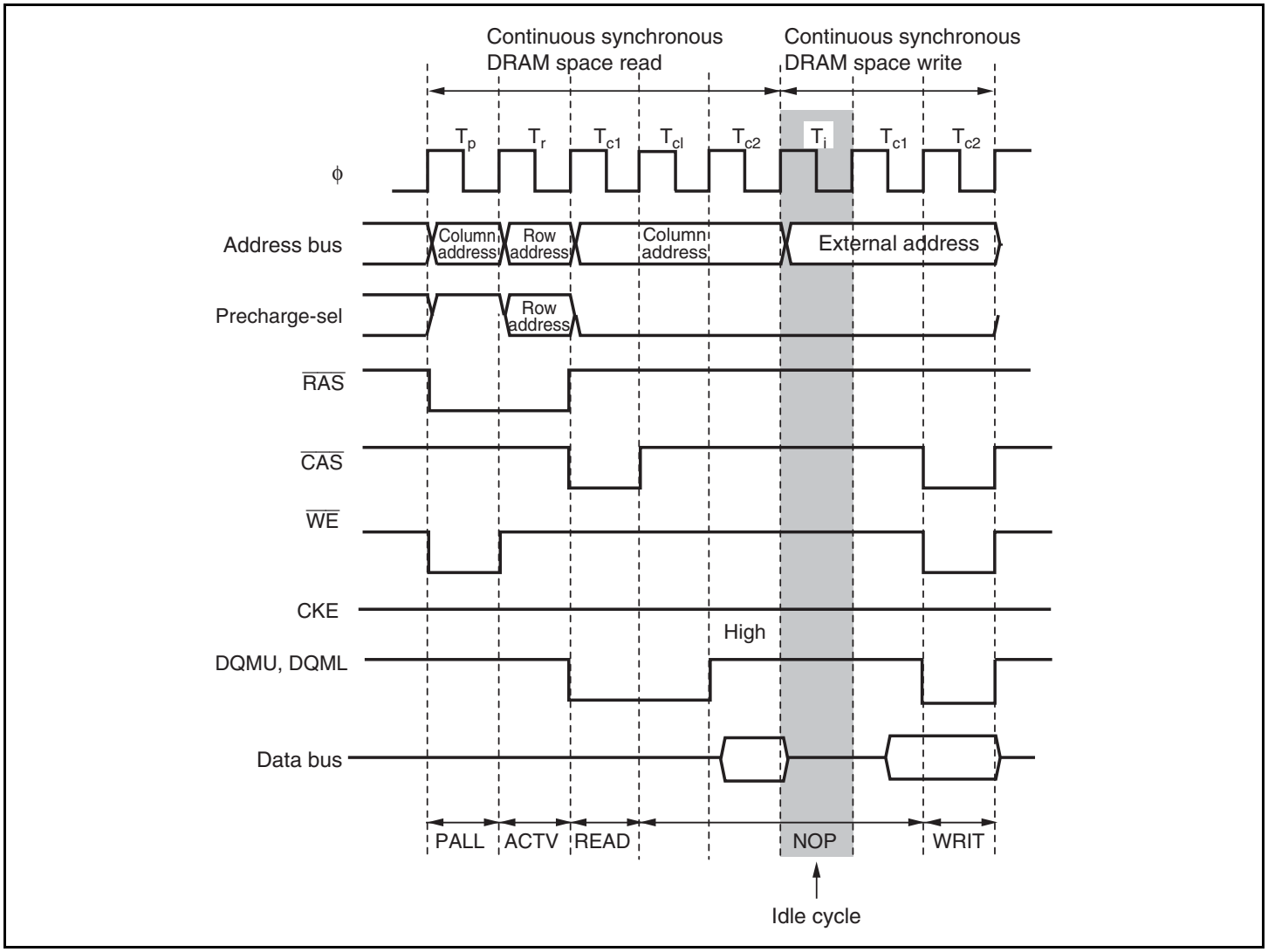


Figure 6.94 Example of Timing for Idle Cycle Insertion in Case of Consecutive Read and Write Accesses to Continuous Synchronous DRAM Space in RAS Down Mode (SDWCD = 1, CAS Latency 2)

6.10.2 Pin States in Idle Cycle

Table 6.13 shows the pin states in an idle cycle.

Table 6.13 Pin States in Idle Cycle

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

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

6.11 Write Data Buffer Function

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

Figure 6.95 shows an example of the timing when the write data buffer function is used. When this function is used, if an external address space write or DMA single address mode transfer continues for two states or longer, and there is an internal access next, an external write only is executed in the first state, but from the next state onward an internal access (on-chip memory or internal I/O register read/write) is executed in parallel with the external address space write rather than waiting until it ends.

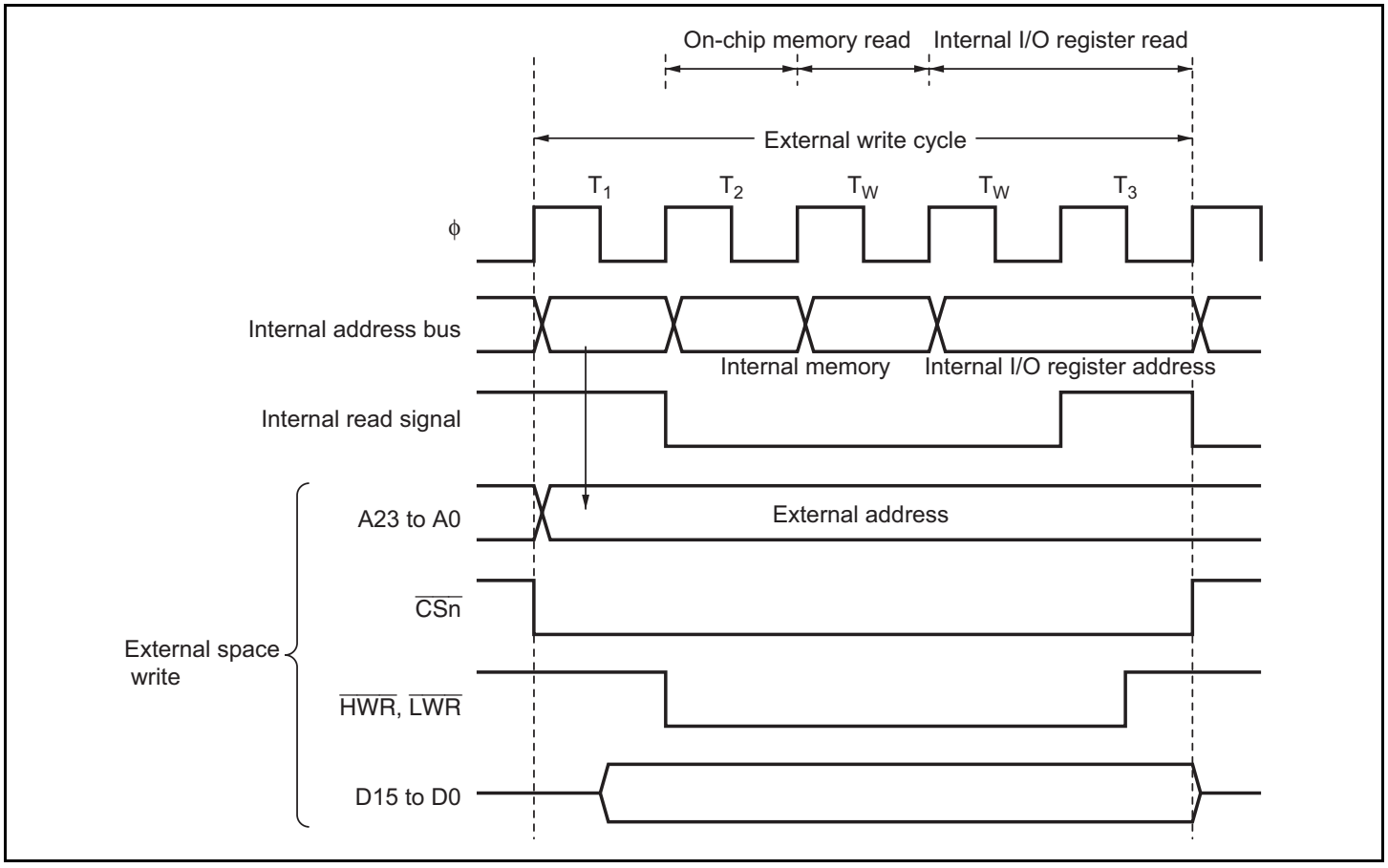


Figure 6.95 Example of Timing when Write Data Buffer Function Is Used

6.12 Bus Release

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

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

Note: * Not supported by the H8S/2454 Group.

6.12.1 Operation

In externally expanded mode, the bus can be released to an external device by setting the BRLE bit to 1 in BCR. Driving the $\overline{\text{BREQ}}$ pin low 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.

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

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

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

When the $\overline{\text{BREQ}}$ pin is driven high, the $\overline{\text{BACK}}$ pin is driven high at the prescribed timing and the external bus released state is terminated.

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

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

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

(High) Refresh > External bus release (Low)

6.12.2 Pin States in External Bus Released State

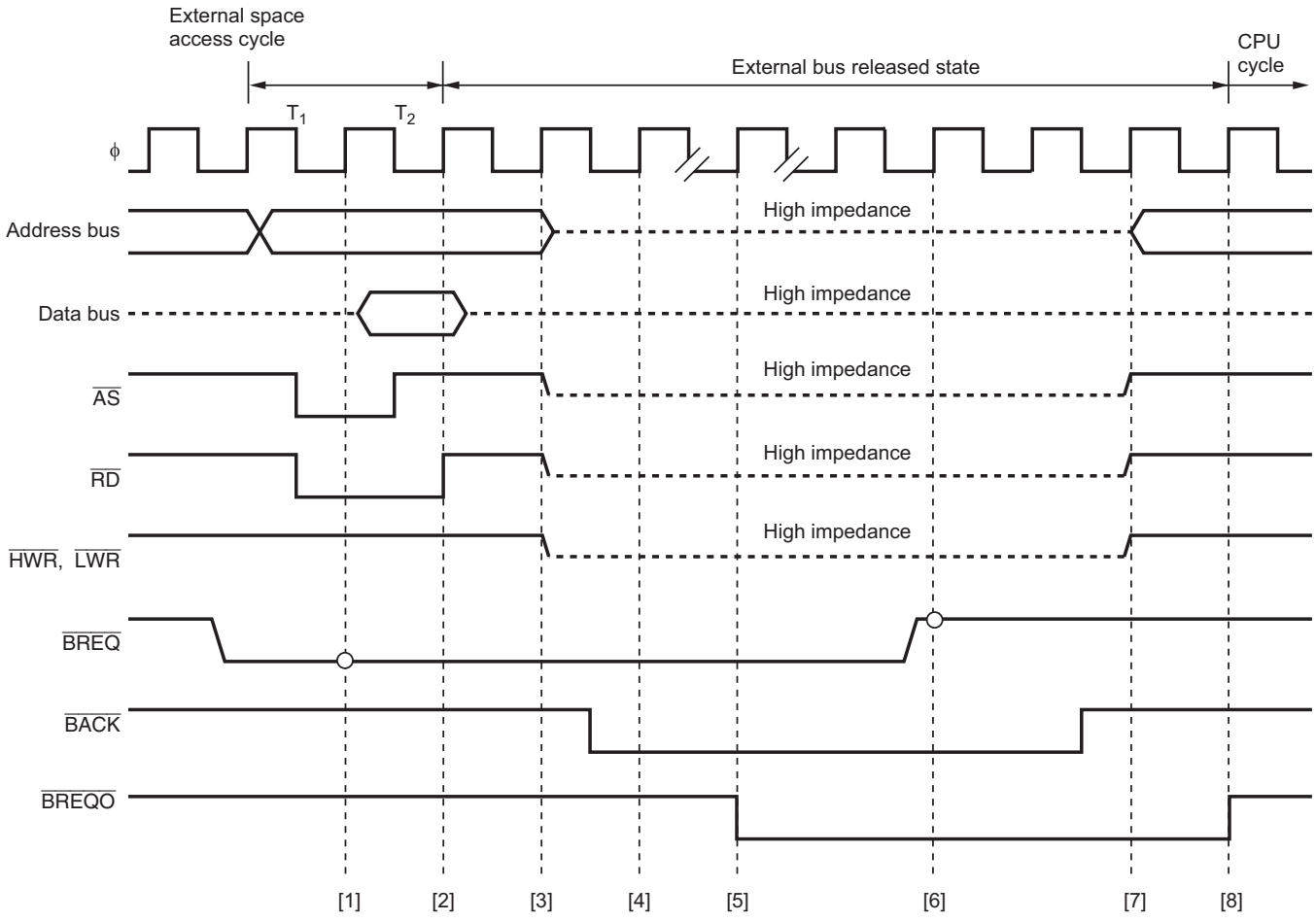
Table 6.14 shows pin states in the external bus released state.

Table 6.14 Pin States in Bus Released State

Pins	Pin State
A23 to A0	High impedance
D15 to D0	High impedance
$\overline{\text{CSn}}$ (n = 7 to 0)	High impedance
$\overline{\text{UCAS}}$, $\overline{\text{LCAS}}$	High impedance
$\overline{\text{AS}}$/$\overline{\text{AH}}$	High impedance
$\overline{\text{RD}}$	High impedance
$\overline{\text{OE}}$	High impedance
$\overline{\text{HWR}}$, $\overline{\text{LWR}}$	High impedance
$\overline{\text{DACKn}}$ (n = 1, 0)	High
$\overline{\text{EDACKn}}$ (n = 3, 2)	High

6.12.3 Transition Timing

Figure 6.96 shows the timing for transition to the bus released state.



[1] Low level of $\overline{\text{BREQ}}$ signal is sampled at rise of φ.

[2] Bus control signal returns to be high at end of external space access cycle.
At least one state from sampling of $\overline{\text{BREQ}}$ signal.

[3] $\overline{\text{BACK}}$ signal is driven low, releasing bus to external bus master.

[4] $\overline{\text{BREQ}}$ signal state is also sampled in external bus released state.

[5] High level of $\overline{\text{BREQ}}$ signal is sampled.

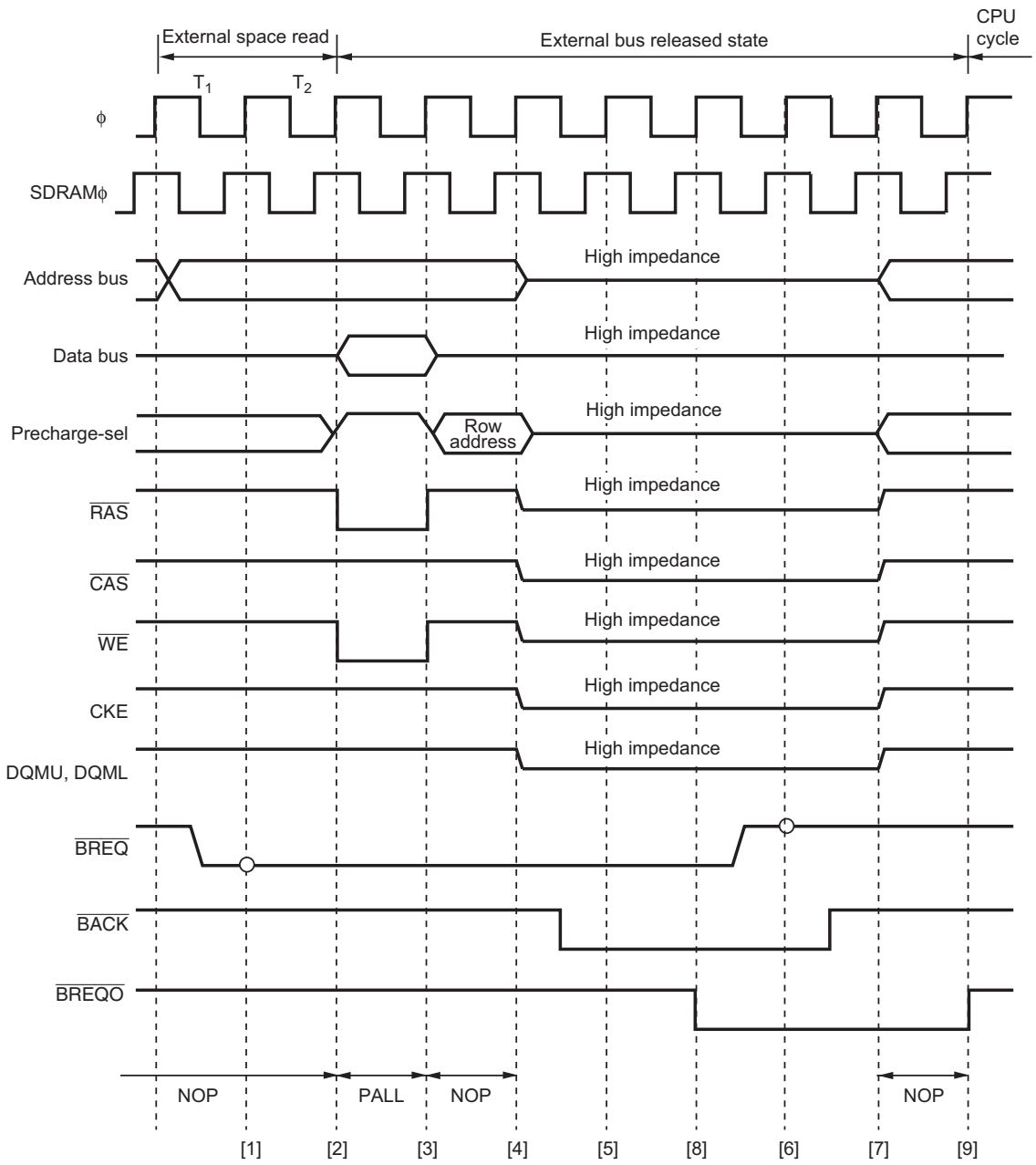
[6] $\overline{\text{BACK}}$ signal is driven high, ending external bus release cycle.

[7] When there is external access or refresh request of internal bus master during external bus release while BREQOE bit is set to 1, $\overline{\text{BREQO}}$ signal goes low.

[8] Normally $\overline{\text{BREQO}}$ signal goes high 1.5 states after rising edge of $\overline{\text{BACK}}$ signal.

Figure 6.96 Bus Released State Transition Timing

Figure 6.97 shows the timing for transition to the bus released state with the synchronous DRAM interface.



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

Figure 6.97 Bus Release State Transition Timing when Synchronous DRAM Interface

Note: The synchronous DRAM interface is not supported by the H8S/2456 Group and H8S/2454 Group.

6.13 Bus Arbitration

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

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

Note: * The EXDMAC is not supported by the H8S/2454 Group.

6.13.1 Operation

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

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

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

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

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

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

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

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

Note: * The EXDMAC is not supported by the H8S/2454 Group.

6.13.2 Bus Transfer Timing

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

(1) CPU

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

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

Note: * The EXDMAC is not supported by the H8S/2454 Group.

(2) DTC

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

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

(3) DMAC

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

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

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

(4) EXDMAC

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

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

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

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

Note: The EXDMAC is not supported by the H8S/2454 Group.

(5) External Bus Release

When the $\overline{\text{BREQ}}$ pin goes low and an external bus release request is issued while the BRLE bit is set to 1 in BCR, a bus request is sent to the bus arbiter.

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

6.14 Bus Controller Operation in Reset

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

6.15 Usage Notes

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

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

6.15.2 External Bus Release Function and Software Standby

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

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

6.15.3 External Bus Release Function and CBR Refreshing/Auto Refreshing

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

Note: The auto refresh control function is not supported by the H8S/2456 Group and H8S/2454 Group.

6.15.4 $\overline{\text{BREQO}}$ Output Timing

When the BREQOE bit is set to 1 and the $\overline{\text{BREQO}}$ signal is output, $\overline{\text{BREQO}}$ may go low before the $\overline{\text{BACK}}$ signal.

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

6.15.5 Notes on Usage of the Synchronous DRAM

(1) Connection Clock

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

(2) $\overline{\text{WAIT}}$ Pin

In the continuous synchronous DRAM space, insertion of the wait state by the $\overline{\text{WAIT}}$ pin is disabled regardless of the setting of the WAITE bit in BCR.

(3) Bank Control

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

(4) Burst Access

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

(5) CAS Latency

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

Note: The synchronous DRAM interface is not supported by the H8S/2456 Group and H8S/2454 Group.

Section 7 DMA Controller (DMAC)

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

7.1 Features

- Selectable as short address mode or full address mode

  Short address mode
  - Maximum of 4 channels can be used
  - Dual address mode or single address mode can be selected
  - In dual address mode, one of the two addresses, transfer source and transfer destination, is specified as 24 bits and the other as 16 bits
  - In single address mode, transfer source or transfer destination address only is specified as 24 bits
  - In single address mode, transfer can be performed in one bus cycle
  - Choice of sequential mode, idle mode, or repeat mode for dual address mode and single address mode

  Full address mode
  - Maximum of 2 channels can be used
  - Transfer source and transfer destination addresses as specified as 24 bits
  - Choice of normal mode or block transfer mode
- 16-Mbyte address space can be specified directly
- Byte or word can be set as the transfer unit
- Activation sources: internal interrupt, external request, auto-request (depending on transfer mode)
  - Six compare match/input capture interrupts of 16-bit timer-pulse unit (TPU0 to TPU5)
  - Transmission complete interrupt and reception complete interrupt of serial communication interface (SCI_0, SCI_1)
  - Conversion end interrupt of A/D converter (A/D_0)
  - EP1FIFO full interrupt and EP2FIFO empty interrupt of USB
  - External request
  - Auto-request
- Module stop mode can be set

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

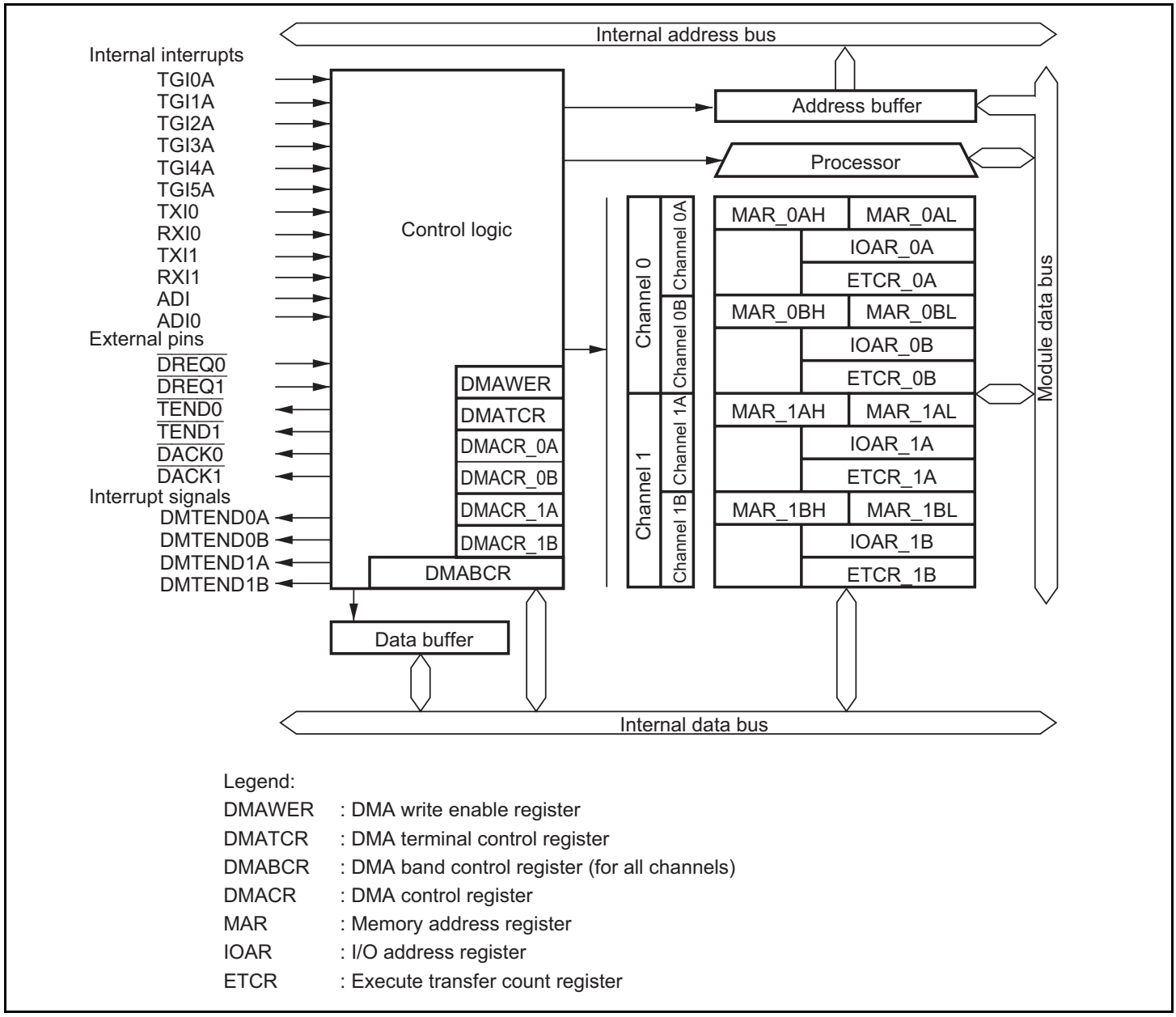


Figure 7.1 Block Diagram of DMAC

7.2 Input/Output Pins

Table 7.1 shows the pin configuration of the interrupt controller.

Table 7.1 Pin Configuration

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

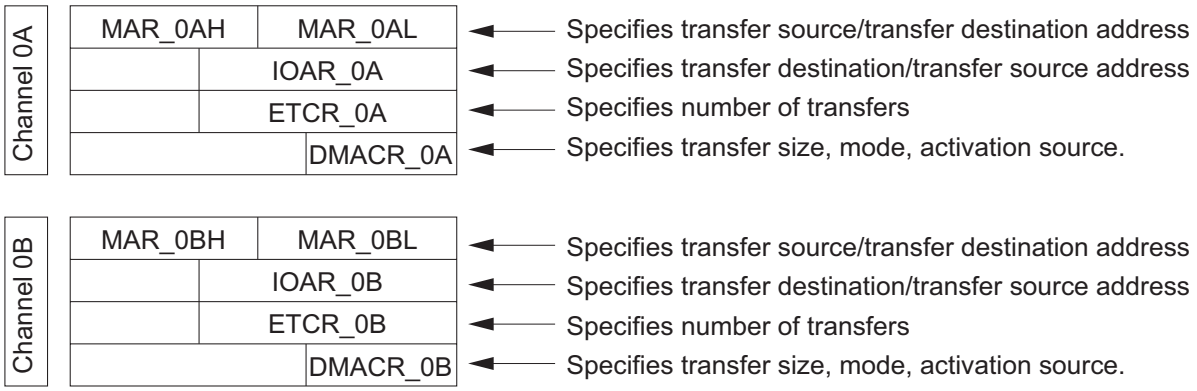
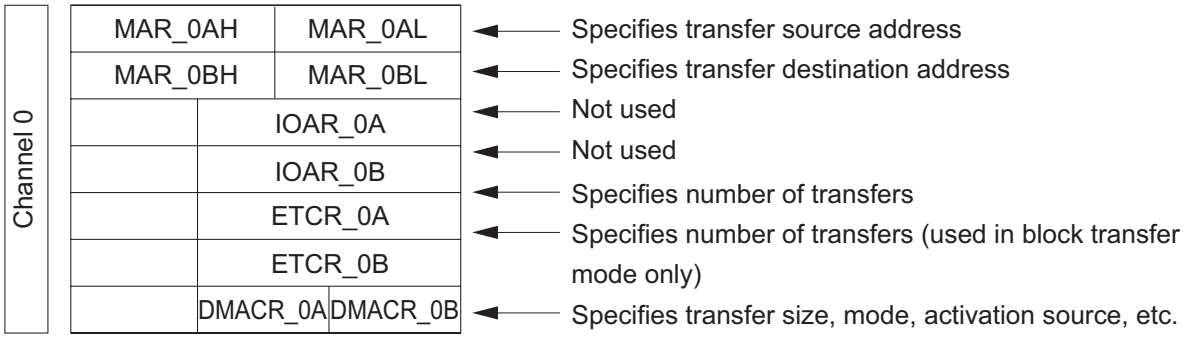
7.3 Register Descriptions

- Memory address register_0AH (MAR_0AH)
- Memory address register_0AL (MAR_0AL)
- I/O address register_0A (IOAR_0A)
- Transfer count register_0A (ECTR_0A)
- Memory address register_0BH (MAR_0BH)
- Memory address register_0BL (MAR_0BL)
- I/O address register_0B (IOAR_0B)
- Transfer count register_0B (ECTR_0B)
- Memory address register_1AH (MAR_1AH)
- Memory address register_1AL (MAR_1AL)
- I/O address register_1A (IOAR_1A)
- Transfer count register_1A (ETCR_1A)
- Memory address register_1BH (MAR_1BH)
- Memory address register_1BL (MAR_1BL)
- I/O address register_1B (IOAR_1B)
- Transfer count register_1B (ETCR_1B)
- DMA control register_0A (DMACR_0A)
- DMA control register_0B (DMACR_0B)

- DMA control register_1A (DMACR_1A)
- DMA control register_1B (DMACR_1B)
- DMA band control register H (DMABCRH)
- DMA band control register L (DMABCRL)
- DMA write enable register (DMAWER)
- DMA terminal control register (DMATCR)

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

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

FAE0	Description
0	Short address mode specified (channels 0A and 0B operate independently) 
1	Full address mode specified (channels 0A and 0B operate in combination as channel 0) 

7.3.1 Memory Address Registers (MARA and MARB)

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

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

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

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

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

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

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

7.3.2 I/O Address Registers (IOARA and IOARB)

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

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

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

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

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

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

7.3.3 Execute Transfer Count Registers (ETCRA and ETCRB)

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

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

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

(1) Short Address Mode

The function of ETCR in sequential mode and idle mode differs from that in repeat mode.

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

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

(2) Full Address Mode

The function of ETCR in normal mode differs from that in block transfer mode.

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

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

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

7.3.4 DMA Control Registers (DMACRA and DMACRB)

DMACR controls the operation of each DMAC channel.

The DMA has four DMACR registers: DMACR_0A in channel 0 (channel 0A), DMACR_0B in channel 0 (channel 0B), DMACR_1A in channel 1 (channel 1A), and DMACR_1B in channel 1 (channel 1B). In short address mode, channels A and B operate independently, and in full address mode, channels A and B operate together. The bit functions in the DMACR registers differ according to the transfer mode.

(1) Short Address Mode:

- DMACR_0A, DMACR_0B, DMACR_1A, and DMARC_1B

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

Bit	Bit Name	Initial Value	R/W	Description
5	RPE	0	R/W	Repeat Enable Used in combination with the DTIE bit in DMABCR to select the mode (sequential, idle, or repeat) in which transfer is to be performed. • When DTIE = 0 (no transfer end interrupt) 0: Transfer in sequential mode 1: Transfer in repeat mode • When DTIE = 1 (with transfer end interrupt) 0: Transfer in sequential mode 1: Transfer in idle mode
4	DTDIR	0	R/W	Data Transfer Direction Used in combination with the SAE bit in DMABCR to specify the data transfer direction (source or destination). The function of this bit is therefore different in dual address mode and single address mode. • When SAE = 0 0: Transfer with MAR as source address and IOAR as destination address 1: Transfer with IOAR as source address and MAR as destination address • When SAE = 1 0: Transfer with MAR as source address and $\overline{\text{DACK}}$ pin as write strobe 1: Transfer with $\overline{\text{DACK}}$ pin as read strobe and MAR as destination address

Bit	Bit Name	Initial Value	R/W	Description
3	DTF3	0	R/W	Data Transfer Factor 3 to 0
2	DTF2	0	R/W	These bits select the data transfer factor (activation source). There are some differences in activation sources for channel A and channel B.
1	DTF1	0	R/W	
0	DTF0	0	R/W	• Channel A 0000: Setting prohibited 0001: Activated by A/D converter conversion end interrupt 0010: Setting prohibited 0011: Setting prohibited 0100: Activated by SCI channel 0 transmission complete interrupt 0101: Activated by SCI channel 0 reception complete interrupt 0110: Activated by SCI channel 1 transmission complete interrupt 0111: Activated by SCI channel 1 reception complete interrupt 1000: Activated by TPU channel 0 compare match/input capture A interrupt 1001: Activated by TPU channel 1 compare match/input capture A interrupt 1010: Activated by TPU channel 2 compare match/input capture A interrupt 1011: Activated by TPU channel 3 compare match/input capture A interrupt 1100: Activated by TPU channel 4 compare match/input capture A interrupt 1101: Activated by TPU channel 5 compare match/input capture A interrupt 1110: Setting prohibited 1111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
3	DTF3	0	R/W	• Channel B
2	DTF2	0	R/W	0000: Setting prohibited
1	DTF1	0	R/W	0001: Activated by A/D converter conversion end interrupt
0	DTF0	0	R/W	0010: Activated by $\overline{\text{DREQ}}$ pin falling edge input (detected as a low level in the first transfer after transfer is enabled)*
				0011: Activated by $\overline{\text{DREQ}}$ pin low-level input*
				0100: Activated by SCI channel 0 transmission complete interrupt
				0101: Activated by SCI channel 0 reception complete interrupt
				0110: Activated by SCI channel 1 transmission complete interrupt
				0111: Activated by SCI channel 1 reception complete interrupt
				1000: Activated by TPU channel 0 compare match/input capture A interrupt
				1001: Activated by TPU channel 1 compare match/input capture A interrupt
				1010: Activated by TPU channel 2 compare match/input capture A interrupt
				1011: Activated by TPU channel 3 compare match/input capture A interrupt
				1100: Activated by TPU channel 4 compare match/input capture A interrupt
				1101: Activated by TPU channel 5 compare match/input capture A interrupt
				1110: Setting prohibited
				1111: Setting prohibited
				The same factor can be selected for more than one channel. In this case, activation starts with the highest-priority channel according to the relative channel priorities. For relative channel priorities, see section 7.5.12, Multi-Channel Operation.

(2) Full Address Mode

- DMACR_0A and DMACR_1A

Bit	Bit Name	Initial Value	R/W	Description
15	DTSZ	0	R/W	Data Transfer Size Selects the size of data to be transferred at one time. 0: Byte-size transfer 1: Word-size transfer
14	SAID	0	R/W	Source Address Increment/Decrement
13	SAIDE	0	R/W	Source Address Increment/Decrement Enable These bits specify whether source address register MARA is to be incremented, decremented, or left unchanged, when data transfer is performed. 00: MARA is fixed 01: MARA is incremented after a data transfer • When DTSZ = 0, MARA is incremented by 1 • When DTSZ = 1, MARA is incremented by 2 10: MARA is fixed 11: MARA is decremented after a data transfer • When DTSZ = 0, MARA is decremented by 1 • When DTSZ = 1, MARA is decremented by 2
12	BLKDIR	0	R/W	Block Direction
11	BLKE	0	R/W	Block Enable These bits specify whether normal mode or block transfer mode is to be used for data transfer. If block transfer mode is specified, the BLKDIR bit specifies whether the source side or the destination side is to be the block area. x0: Transfer in normal mode 01: Transfer in block transfer mode (destination side is block area) 11: Transfer in block transfer mode (source side is block area)

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

Legend:
x: Don't care

- DMACR_0B and DMACR_1B

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.
6	DAID	0	R/W	Destination Address Increment/Decrement
5	DAIDE	0	R/W	Destination Address Increment/Decrement Enable These bits specify whether destination address register MARB is to be incremented, decremented, or left unchanged, when data transfer is performed. 00: MARB is fixed 01: MARB is incremented after a data transfer • When DTSZ = 0, MARB is incremented by 1 • When DTSZ = 1, MARB is incremented by 2 10: MARB is fixed 11: MARB is decremented after a data transfer • When DTSZ = 0, MARB is decremented by 1 • When DTSZ = 1, MARB is decremented by 2
4	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	DTF3	0	R/W	Data Transfer Factor 3 to 0
2	DTF2	0	R/W	These bits select the data transfer factor (activation source). The factors that can be specified differ between normal mode and block transfer mode.
1	DTF1	0	R/W	
0	DTF0	0	R/W	
				• Normal Mode
				0000: Setting prohibited
				0001: Setting prohibited
				0010: Activated by $\overline{\text{DREQ}}$ pin falling edge input (detected as a low level in the first transfer after transfer is enabled)*
				0011: When USBDRQE bit in PFCR3 is 0: Activated by $\overline{\text{DREQ}}$ pin low-level input When USBDRQE bit in PFCR3 is 1: Activated by USB interrupt signal low-level input
				010x: Setting prohibited
				0110: Auto-request (cycle steal)
				0111: Auto-request (burst)
				1xxx: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
3	DTF3	0	R/W	• Block Transfer Mode
2	DTF2	0	R/W	0000: Setting prohibited
1	DTF1	0	R/W	0001: Activated by A/D converter conversion end interrupt
0	DTF0	0	R/W	0010: Activated by $\overline{\text{DREQ}}$ pin falling edge input (detected as a low level in the first transfer after transfer is enabled)*
				0011: When USBDRQE bit in PFCR3 is 0: Activated by $\overline{\text{DREQ}}$ pin low-level input When USBDRQE bit in PFCR3 is 1: Activated by USB interrupt signal low-level input
				0100: Activated by SCI channel 0 transmission complete interrupt
				0101: Activated by SCI channel 0 reception complete interrupt
				0110: Activated by SCI channel 1 transmission complete interrupt
				0111: Activated by SCI channel 1 reception complete interrupt
				1000: Activated by TPU channel 0 compare match/input capture A interrupt
				1001: Activated by TPU channel 1 compare match/input capture A interrupt
				1010: Activated by TPU channel 2 compare match/input capture A interrupt
				1011: Activated by TPU channel 3 compare match/input capture A interrupt
				1100: Activated by TPU channel 4 compare match/input capture A interrupt
				1101: Activated by TPU channel 5 compare match/input capture A interrupt
				1110: Setting prohibited
				1111: Setting prohibited
				The same factor can be selected for more than one channel. In this case, activation starts with the highest-priority channel according to the relative channel priorities. For relative channel priorities, see section 7.5.12, Multi-Channel Operation.

Legend:
×: Don't care
Note: * This setting is prohibited when the USBDRQE bit in PFCR3 is 1.

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

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

(1) Short Address Mode:

- DMABCRH

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

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

- DMABCRL

Bit	Bit Name	Initial Value	R/W	Description
7	DTE1B	0	R/W	Data Transfer Enable 1B
6	DTE1A	0	R/W	Data Transfer Enable 1A
5	DTE0B	0	R/W	Data Transfer Enable 0B
4	DTE0A	0	R/W	Data Transfer Enable 0A If the DTE bit is cleared to 0 when DTIE = 1, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC. When DTE = 0, data transfer is disabled and the DMAC ignores the activation source selected by the DTF3 to DTF0 bits in DMACR. When DTE = 1, data transfer is enabled and the DMAC waits for a request by the activation source selected by the DTF3 to DTF0 bits in DMACR. When a request is issued by the activation source, DMA transfer is executed. [Clearing conditions] • When initialization is performed • When the specified number of transfers have been completed in a transfer mode other than repeat mode • When 0 is written to the DTE bit to forcibly suspend the transfer, or for a similar reason [Setting condition] When 1 is written to the DTE bit after reading DTE = 0

Bit	Bit Name	Initial Value	R/W	Description
3	DTIE1B	0	R/W	Data Transfer End Interrupt Enable 1B
2	DTIE1A	0	R/W	Data Transfer End Interrupt Enable 1A
1	DTIE0B	0	R/W	Data Transfer End Interrupt Enable 0B
0	DTIE0A	0	R/W	Data Transfer End Interrupt Enable 0A These bits enable or disable an interrupt to the CPU or DTC when transfer ends. If the DTIE bit is set to 1 when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC. A transfer end interrupt can be canceled either by clearing the DTIE bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE bit to 1.

(2) Full Address Mode:

- DMABCRH

Bit	Bit Name	Initial Value	R/W	Description
15	FAE1	0	R/W	Full Address Enable 1 Specifies whether channel 1 is to be used in short address mode or full address mode. In full address mode, channels 1A and 1B are used together as channel 1. 0: Short address mode 1: Full address mode
14	FAE0	0	R/W	Full Address Enable 0 Specifies whether channel 0 is to be used in short address mode or full address mode. In full address mode, channels 0A and 0B are used together as channel 0. 0: Short address mode 1: Full address mode

Bit	Bit Name	Initial Value	R/W	Description
13, 12	—	All 0	R/W	Reserved These bits can be read from or written to. However, the write value should always be 0.
11	DTA1	0	R/W	Data Transfer Acknowledge 1 These bits enable or disable clearing when DMA transfer is performed for the internal interrupt source selected by the DTF3 to DTF0 bits in DMACR of channel 1. It the DTA1 bit is set to 1 when DTE1 = 1, the internal interrupt source is cleared automatically by DMA transfer. When DTE1 = 1 and DTA1 = 1, the internal interrupt source does not issue an interrupt request to the CPU or DTC. It the DTA1 bit is cleared to 0 when DTE1 = 1, the internal interrupt source is not cleared when a transfer is performed, and can issue an interrupt request to the CPU or DTC in parallel. In this case, the interrupt source should be cleared by the CPU or DTC transfer. When DTE1 = 0, the internal interrupt source issues an interrupt request to the CPU or DTC regardless of the DTA1 bit setting. The state of the DTME1 bit does not affect the above operations.
10	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.

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

- DMABCRL

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
4	DTE0	0	R/W	Data Transfer Enable 0 Enables or disables DMA transfer for the activation source selected by the DTF3 to DTF0 bits in DMACR of channel 0. When DTE0 = 0, data transfer is disabled and the activation source is ignored. If the activation source is an internal interrupt, an interrupt request is issued to the CPU or DTC. If the DTE0 bit is cleared to 0 when DTIE0 = 1, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU. When DTE0 = 1 and DTME0 = 1, data transfer is enabled and the DMAC waits for a request by the activation source. When a request is issued by the activation source, DMA transfer is executed. [Clearing conditions] • When initialization is performed • When the specified number of transfers have been completed • When 0 is written to the DTE0 bit to forcibly suspend the transfer, or for a similar reason [Setting condition] When 1 is written to the DTE0 bit after reading DTE0 = 0
3	DTIE1B	0	R/W	Data Transfer Interrupt Enable 1B Enables or disables an interrupt to the CPU or DTC when transfer on channel 1 is interrupted. If the DTME1 bit is cleared to 0 when DTIE1B = 1, the DMAC regards this as indicating a break in the transfer, and issues a transfer break interrupt request to the CPU or DTC. A transfer break interrupt can be canceled either by clearing the DTIE1B bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the DTME1 bit to 1.

Bit	Bit Name	Initial Value	R/W	Description
2	DTIE1A	0	R/W	Data Transfer End Interrupt Enable 1A Enables or disables an interrupt to the CPU or DTC when transfer ends. If the DTE1 bit is cleared to 1 when DTIE1A = 1, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC. A transfer end interrupt can be canceled either by clearing the DTIE1A bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE1 bit to 1.
1	DTIE0B	0	R/W	Data Transfer Interrupt Enable 0B Enables or disables an interrupt to the CPU or DTC when transfer on channel 1 is interrupted. If the DTME0 bit is cleared to 0 when DTIE0B = 1, the DMAC regards this as indicating a break in the transfer, and issues a transfer break interrupt request to the CPU or DTC. A transfer break interrupt can be canceled either by clearing the DTIE0B bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the DTME0 bit to 1.
0	DTIE0A	0	R/W	Data Transfer End Interrupt Enable 0A Enables or disables an interrupt to the CPU or DTC when transfer ends. If the DTE0 bit is cleared to 0 when DTIE0A = 1, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC. A transfer end interrupt can be canceled either by clearing the DTIE0A bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE0 bit to 1.

7.3.6 DMA Write Enable Register (DMAWER)

The DMAC can activate the DTC with a transfer end interrupt, rewrite the channel on which the transfer ended using a DTC chain transfer, and then reactivate the DTC. DMAWER applies restrictions for changing all bits of DMACR, and specific bits for DMATCR and DMABCR for the specific channel, to prevent inadvertent rewriting of registers other than those for the channel concerned. The restrictions applied by DMAWER are valid for the DTC.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	WE1B	0	R/W	Write Enable 1B Enables or disables writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR. 0: Writes are disabled 1: Writes are enabled
2	WE1A	0	R/W	Write Enable 1A Enables or disables writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR. 0: Writes are disabled 1: Writes are enabled
1	WE0B	0	R/W	Write Enable 0B Enables or disables writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, and bit 4 in DMATCR. 0: Writes are disabled 1: Writes are enabled
0	WE0A	0	R/W	Write Enable 0A Enables or disables writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR. 0: Writes are disabled 1: Writes are enabled

Figure 7.2 shows the transfer areas for activating the DTC with a channel 0A transfer end interrupt request, and reactivating channel 0A. The address register and count register areas are set again during the first DTC transfer, then the control register area is set again during the second DTC chain transfer. When re-setting the control register area, perform masking by setting bits in DMAWER to prevent modification of the contents of other channels.

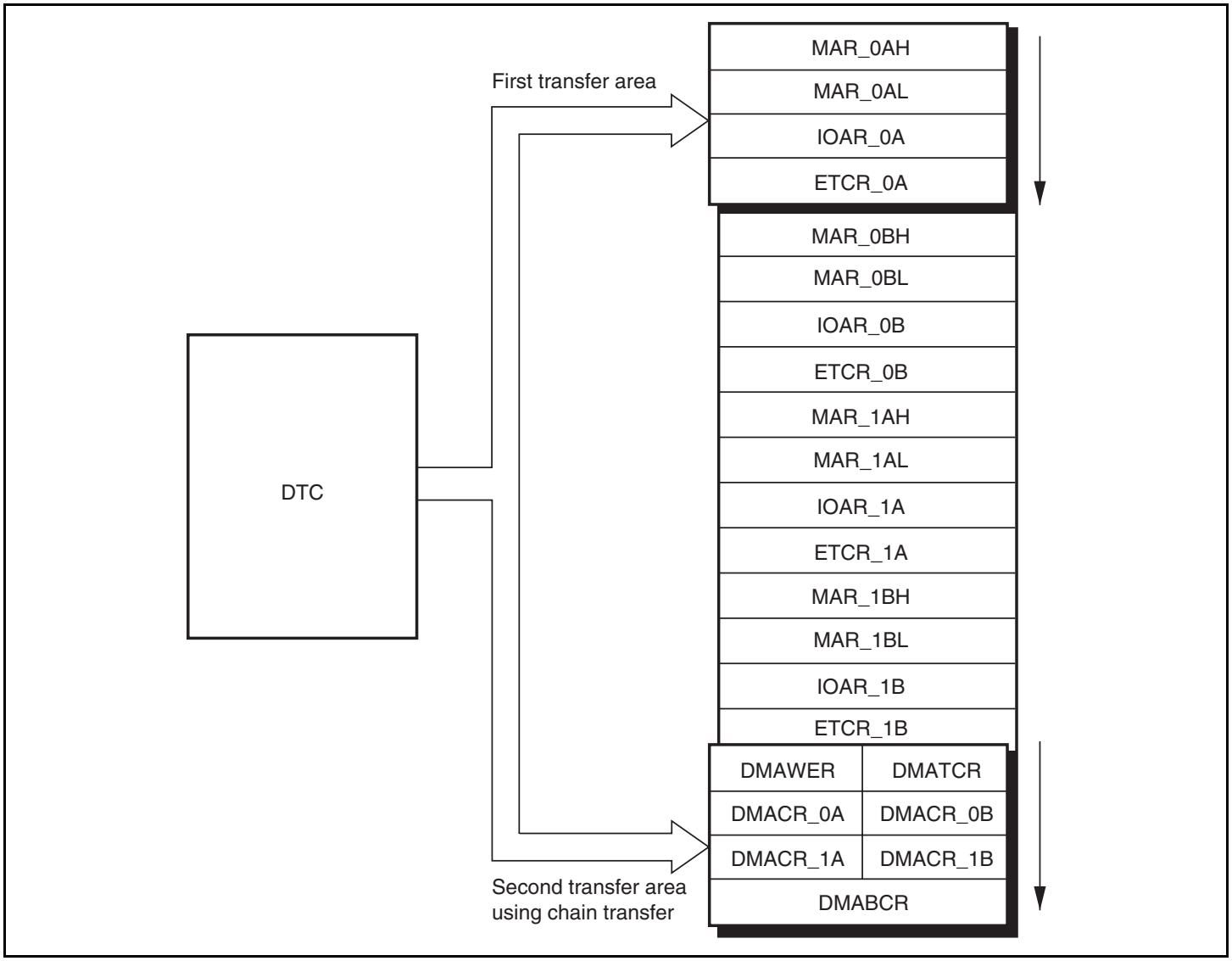


Figure 7.2 Areas for Register Re-Setting by DTC (Channel 0A)

Writes by the DTC to bits 15 to 12 (FAE and SAE) in DMABCR are invalid regardless of the DMAWER settings. These bits should be changed, if necessary, by CPU processing.

In writes by the DTC to bits 7 to 4 (DTE) in DMABCR, 1 can be written without first reading 0. To reactivate a channel set to full address mode, write 1 to both Write Enable A and Write Enable B for the channel to be reactivated.

MAR, IOAR, and ETCR can always be written to regardless of the DMAWER settings. When modifying these registers, the channel to be modified should be halted.

7.3.7 DMA Terminal Control Register (DMATCR)

DMATCR controls enabling or disabling of output from the DMAC transfer end pin. A port can be set for output automatically, and a transfer end signal output, by setting the appropriate bit. The TEND pin is available only for channel B in short address mode. Except for the block transfer mode, a transfer end signal asserts in the transfer cycle in which the transfer counter contents reaches 0 regardless of the activation source. In the block transfer mode, a transfer end signal asserts in the transfer cycle in which the block counter contents reaches 0.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
5	TEE1	0	R/W	Transfer End Enable 1 Enables or disables transfer end pin 1 ($\overline{\text{TEND1}}$) output. 0: $\overline{\text{TEND1}}$ pin output disabled 1: $\overline{\text{TEND1}}$ pin output enabled
4	TEE0	0	R/W	Transfer End Enable 0 Enables or disables transfer end pin 0 ($\overline{\text{TEND0}}$) output. 0: $\overline{\text{TEND0}}$ pin output disabled 1: $\overline{\text{TEND0}}$ pin output enabled
3 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

7.4 Activation Sources

DMAC activation sources consist of internal interrupt requests, external requests, and auto-requests. The DMAC activation sources that can be specified depend on the transfer mode and channel, as shown in table 7.3.

Table 7.3 DMAC Activation Sources

		Short Address Mode		Full Address Mode	
Activation Source		**Channels 0A and 1A**	**Channels 0B and 1B**	**Normal Mode**	**Block Transfer Mode**
Internal interrupts	ADI0	○	○	×	○
	TXI0	○	○	×	○
	RXI0	○	○	×	○
	TXI1	○	○	×	○
	RXI1	○	○	×	○
	TGI0A	○	○	×	○
	TGI1A	○	○	×	○
	TGI2A	○	○	×	○
	TGI3A	○	○	×	○
	TGI4A	○	○	×	○
	TGI5A	○	○	×	○
	USBINTN0	×	×	○	○
	USBINTN1	×	×	○	○
External requests	$\overline{\text{DREQ}}$ pin falling edge input	×	○	○	○
	$\overline{\text{DREQ}}$ pin low-level input	×	○	○	○
Auto-request		×	×	○	×

Legend:

○: Can be specified

×: Cannot be specified

7.4.1 Activation by Internal Interrupt Request

An interrupt request selected as a DMAC activation source can also simultaneously generate an interrupt request for the CPU or DTC. For details, see section 5, Interrupt Controller.

With activation by an internal interrupt request, the DMAC accepts the interrupt request independently of the interrupt controller. Consequently, interrupt controller priority settings are irrelevant.

If the DMAC is activated by a CPU interrupt source or an interrupt request that is not used as a DTC activation source (DTA = 1), the interrupt request flag is cleared automatically by the DMA transfer. With ADI, TXI, and RXI interrupts, however, the interrupt source flag is not cleared unless the relevant register is accessed in a DMA transfer. If the same interrupt is used as an activation source for more than one channel, the interrupt request flag is cleared when the highest-priority channel is activated. Transfer requests for other channels are held pending in the DMAC, and activation is carried out in order of priority.

When DTE = 0 after completion of a transfer, an interrupt request from the selected activation source is not sent to the DMAC, regardless of the DTA bit setting. In this case, the relevant interrupt request is sent to the CPU or DTC.

When an interrupt request signal for DMAC activation is also used for an interrupt request to the CPU or DTC activation (DTA = 0), the interrupt request flag is not cleared by the DMAC.

If the DMAC is activated by a USB interrupt source, setting bits DTF[3:0] in DMACR to 4'b0011 and the USBDRQE bit in PFCR3 to 1 activates the DMAC at a low-level input of the USB interrupt signal.

The DMAC stands by for a transfer request while the USB interrupt source is held high. While the USB interrupt source is held low, transfers continue in succession, with the bus being released each time a byte or word is transferred. If the USB interrupt source goes high in the middle of a transfer, the transfer is interrupted and the DMAC stands by for a transfer request.

7.4.2 Activation by External Request

If an external request ($\overline{\text{DREQ}}$ pin) is specified as a DMAC activation source, the relevant port should be set to input mode in advance*. Level sensing or edge sensing can be used for external requests.

External request operation in normal mode of short address mode or full address mode is described below.

When edge sensing is selected, a byte or word is transferred each time a high-to-low transition is detected on the $\overline{\text{DREQ}}$ pin. The next data transfer may not be performed if the next edge is input before data transfer is completed.

When level sensing is selected, the DMAC stands by for a transfer request while the $\overline{\text{DREQ}}$ pin is held high. While the $\overline{\text{DREQ}}$ pin is held low, transfers continue in succession, with the bus being released each time a byte or word is transferred. If the $\overline{\text{DREQ}}$ pin goes high in the middle of a transfer, the transfer is interrupted and the DMAC stands by for a transfer request.

Note: * If the relevant port is set as an output pin for another function, DMA transfers using the channel in question cannot be guaranteed.

7.4.3 Activation by Auto-Request

Auto-request is activated by register setting only, and transfer continues to the end. With auto-request activation, cycle steal mode or burst mode can be selected.

In cycle steal mode, the DMAC releases the bus to another bus master each time a byte or word is transferred. DMA and CPU cycles are usually repeated alternately. In burst mode, the DMAC keeps possession of the bus until the end of the transfer so that transfer is performed continuously.

7.5 Operation

7.5.1 Transfer Modes

Table 7.4 lists the DMAC transfer modes.

Table 7.4 DMAC Transfer Modes

Transfer Mode		Transfer Source	Remarks
Short address mode	Dual address mode • 1-byte or 1-word transfer for a single transfer request • Specify source and destination addresses to transfer data in two bus cycles. (1) Sequential mode • Memory address incremented or decremented by 1 or 2 • Number of transfers: 1 to 65,536 (2) Idle mode • Memory address fixed • Number of transfers: 1 to 65,536 (3) Repeat mode • Memory address incremented or decremented by 1 or 2 • Continues transfer after sending number of transfers (1 to 256) and restoring the initial value	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmission complete interrupt • SCI reception complete interrupt • A/D converter conversion end interrupt • External request	• Up to 4 channels can operate independently • External request applies to channel B only • Single address mode applies to channel B only
	Single address mode • 1-byte or 1-word transfer for a single transfer request • 1-bus cycle transfer by means of $\overline{\text{DACK}}$ pin instead of using address for specifying I/O • Sequential mode, idle mode, or repeat mode can be specified	• External request	

Transfer Mode		Transfer Source	Remarks
Full address mode	Normal mode (1) Auto-request • Transfer request is internally held • Number of transfers (1 to 65,536) is continuously sent • Burst/cycle steal transfer can be selected	• Auto-request	• Max. 2-channel operation, combining channels A and B
	(2) External request • 1-byte or 1-word transfer for a single transfer request • Number of transfers: 1 to 65,536	• External request	
	(3) Internal request • 1-byte or 1-word transfer for a single transfer request • Number of transfers: 1 to 65,536	• USB interrupt	
	Block transfer mode • Transfer of 1-block, size selected for a single transfer request • Number of transfers: 1 to 65,536 • Source or destination can be selected as block area • Block size: 1 to 256 bytes or word	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmission complete interrupt • SCI reception complete interrupt • A/D converter conversion end interrupt • USB interrupt • External request	

7.5.2 Sequential Mode

Sequential mode can be specified by clearing the RPE bit in DMACR to 0. In sequential mode, MAR is updated after each byte or word transfer in response to a single transfer request, and this is executed the number of times specified in ETCR. One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR.

Table 7.5 summarizes register functions in sequential mode.

Table 7.5 Register Functions in Sequential Mode

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	Incremented/ decremented every transfer
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
	Transfer counter		Number of transfers	Decremented every transfer; transfer ends when count reaches H'0000

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is incremented or decremented by 1 or 2 each time a byte or word is transferred. IOAR specifies the lower 16 bits of the other address. The 8 bits above IOAR have a value of H'FF.

Figure 7.3 illustrates operation in sequential mode.

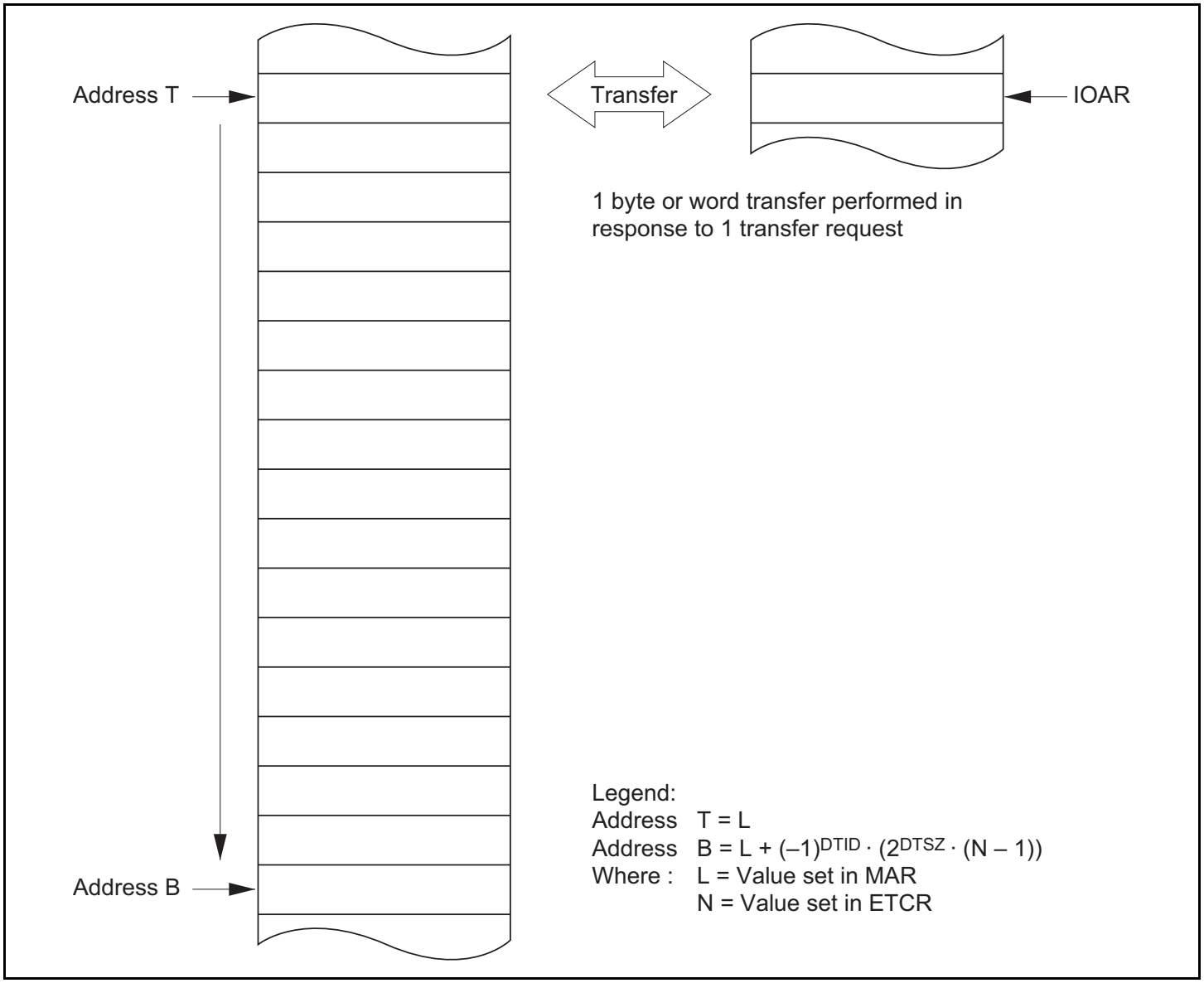
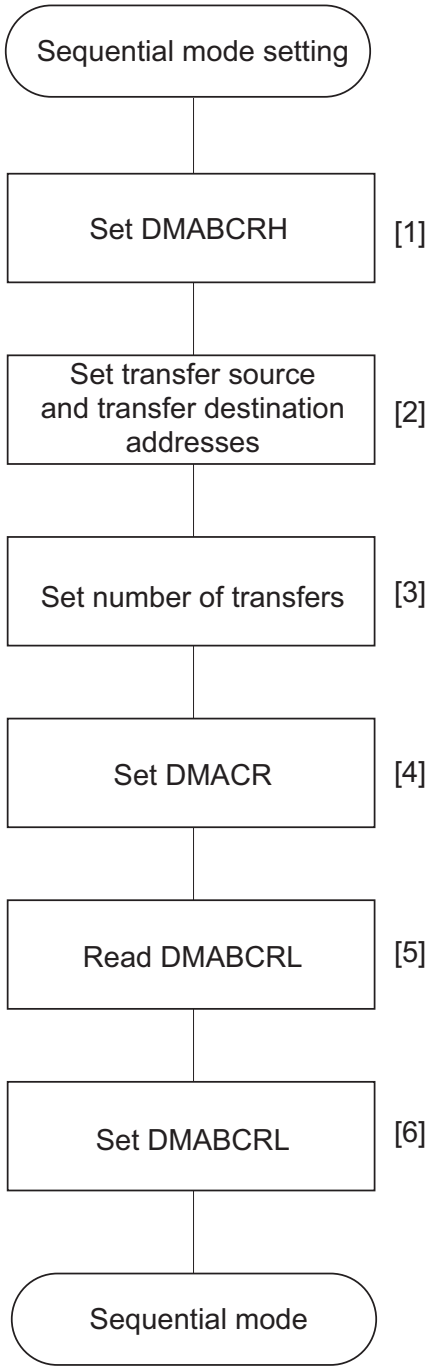


Figure 7.3 Operation in Sequential Mode

The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 each time a data transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and data transfer ends. If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC. The maximum number of transfers, when H'0000 is set in ETCR, is 65,536.

Transfer requests (activation sources) consist of A/D converter conversion end interrupts, external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts. External requests can only be specified for channel B.

Figure 7.4 shows an example of the setting procedure for sequential mode.



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Clear the RPE bit to 0 to select sequential mode.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts with the DTIE bit.
- Set the DTE bit to 1 to enable transfer.

Figure 7.4 Example of Sequential Mode Setting Procedure

7.5.3 Idle Mode

Idle mode can be specified by setting the RPE bit in DMACR and DTIE bit in DMABCRL to 1. In idle mode, one byte or word is transferred in response to a single transfer request, and this is executed the number of times specified in ETCR. One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR. Table 7.6 summarizes register functions in idle mode.

Table 7.6 Register Functions in Idle Mode

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
23 0 MAR	Source address register	Destination address register	Start address of transfer destination or transfer source	Fixed
23 15 0 H'FF IOAR	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
15 0 ETCR	Transfer counter		Number of transfers	Decremented every transfer; transfer ends when count reaches H'0000

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is neither incremented nor decremented by a data transfer. IOAR specifies the lower 16 bits of the other address. The upper 8 bits of IOAR have a value of H'FF.

Figure 7.5 illustrates operation in idle mode.

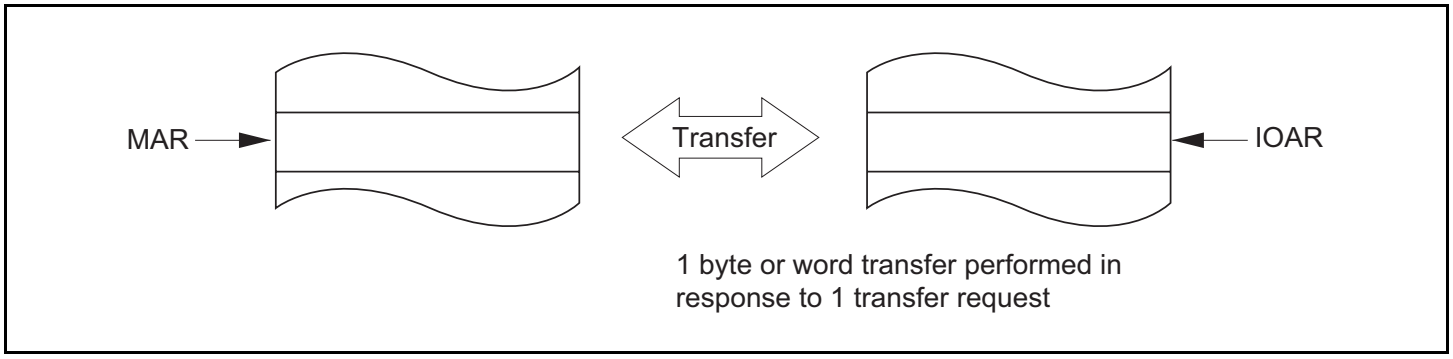
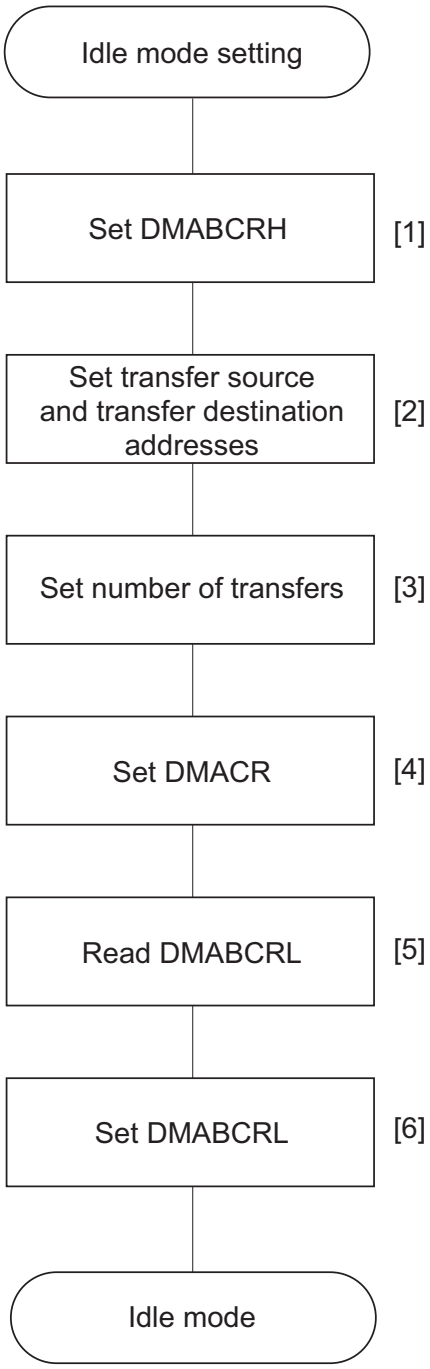


Figure 7.5 Operation in Idle Mode

The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 each time a transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and data transfer ends. If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC. The maximum number of transfers, when H'0000 is set in ETCR, is 65,536.

Transfer requests (activation sources) consist of A/D converter conversion end interrupts, external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts. External requests can only be specified for channel B.

Figure 7.6 shows an example of the setting procedure for idle mode.



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Set the RPE bit to 1.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Set the DTIE bit to 1.
- Set the DTE bit to 1 to enable transfer.

Figure 7.6 Example of Idle Mode Setting Procedure

7.5.4 Repeat Mode

Repeat mode can be specified by setting the RPE bit in DMACR to 1, and clearing the DTIE bit in DMABCRL to 0. In repeat mode, MAR is updated after each byte or word transfer in response to a single transfer request, and this is executed the number of times specified in ETCRL. On completion of the specified number of transfers, MAR and ETCRL are automatically restored to their original settings and operation continues. One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR. Table 7.7 summarizes register functions in repeat mode.

Table 7.7 Register Functions in Repeat Mode

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	Incremented/ decremented every transfer. Initial setting is restored when the value reaches H'0000
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
	Holds number of transfers		Number of transfers	Fixed
	Transfer counter		Number of transfers	Decremented every transfer. Loaded with ETCRH value when the value reaches H'00

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is incremented or decremented by 1 or 2 each time a byte or word is transferred. IOAR specifies the lower 16 bits of the other address. The upper 8 bits of IOAR have a value of H'FF. The number of transfers is specified as 8 bits by ETCRH and ETCRL. The maximum number of transfers, when H'00 is set in both ETCRH and ETCRL, is 256.

In repeat mode, ETCRL functions as the transfer counter, and ETCRH is used to hold the number of transfers. ETCRL is decremented by 1 each time a data transfer is executed, and when its value reaches H'00, it is loaded with the value in ETCRH. At the same time, the value set in MAR is restored in accordance with the values of the DTSZ and DTID bits in DMACR. The MAR restoration operation is as shown below.

$$MAR = MAR - (-1)^{DTID} \cdot 2^{DTSZ} \cdot ETCRH$$

The same value should be set in ETCRH and ETCRL.

In repeat mode, operation continues until the DTE bit in DMABCRL is cleared. To end the transfer operation, therefore, the DTE bit should be cleared to 0. A transfer end interrupt request is not sent to the CPU or DTC. By setting the DTE bit to 1 again after it has been cleared, the operation can be restarted from the transfer after that terminated when the DTE bit was cleared.

Figure 7.7 illustrates operation in repeat mode.

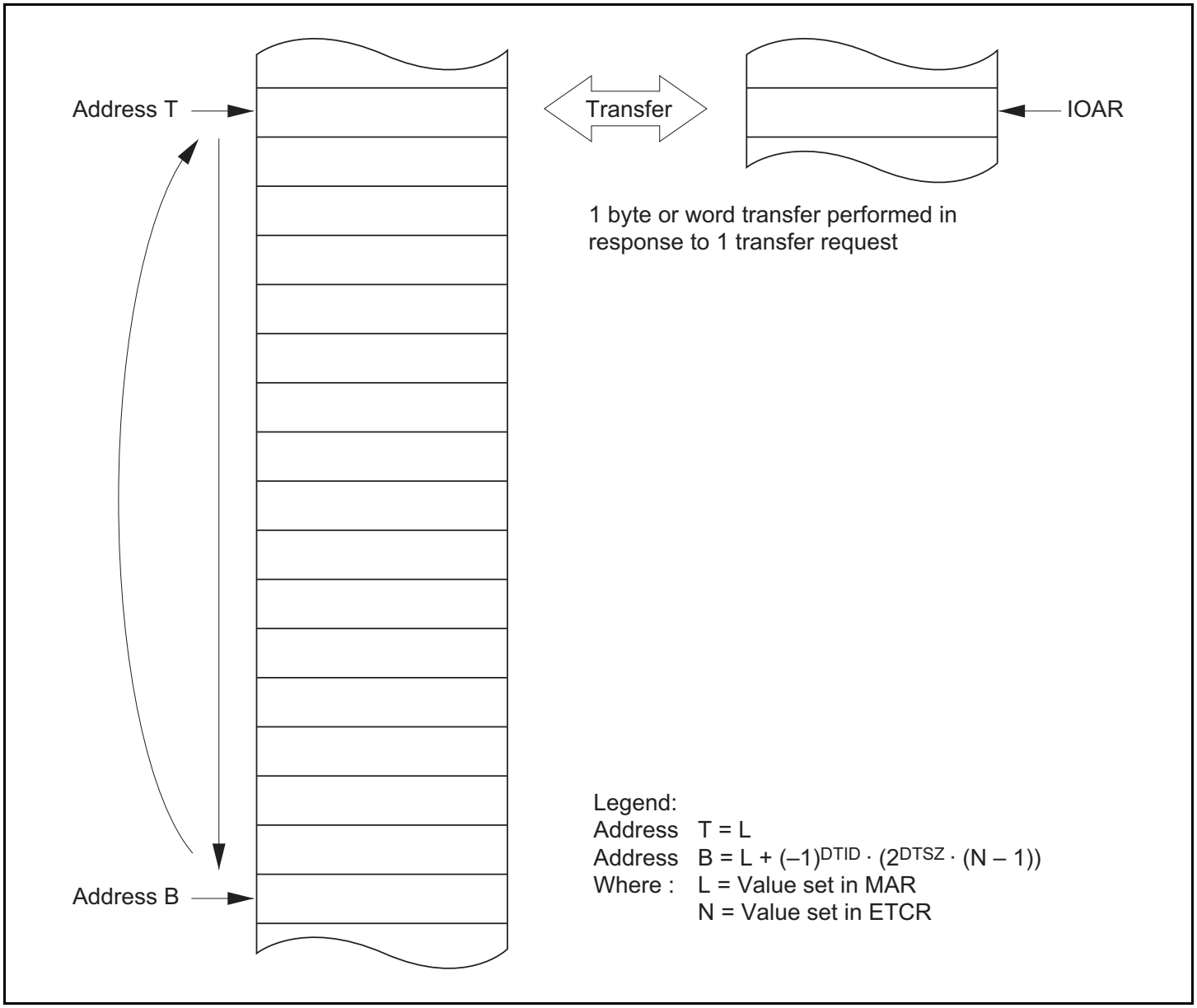
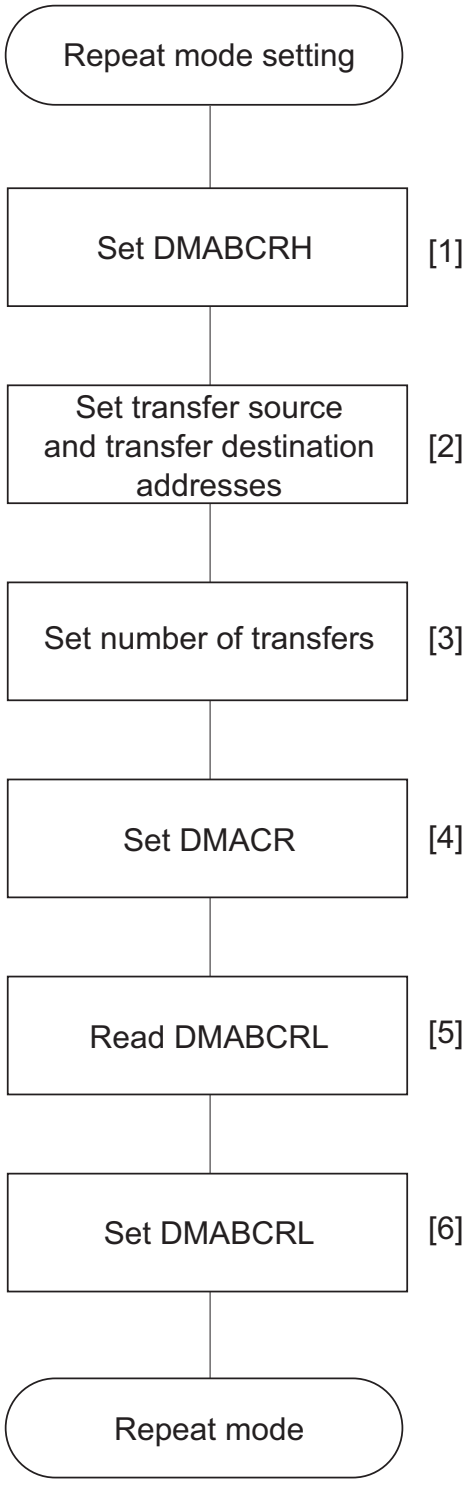


Figure 7.7 Operation in Repeat mode

Transfer requests (activation sources) consist of A/D converter conversion end interrupts, external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts. External requests can only be specified for channel B.

Figure 7.8 shows an example of the setting procedure for repeat mode.



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in both ETCRH and ETCRL.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Set the RPE bit to 1.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Clear the DTIE bit to 0.
- Set the DTE bit to 1 to enable transfer.

Figure 7.8 Example of Repeat Mode Setting Procedure

7.5.5 Single Address Mode

Single address mode can only be specified for channel B. This mode can be specified by setting the SAE bit in DMABCRH to 1 in short address mode.

One address is specified by MAR, and the other is set automatically to the data transfer acknowledge pin ($\overline{\text{DACK}}$). The transfer direction can be specified by the DTDIR bit in DMACR. Table 7.8 summarizes register functions in single address mode.

Table 7.8 Register Functions in Single Address Mode

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
23 MAR 0	Source address register	Destination address register	Start address of transfer destination or transfer source	See sections 7.5.2, Sequential Mode, 7.5.3, Idle Mode, and 7.5.4, Repeat Mode.
$\overline{\text{DACK}}$ pin	Write strobe	Read strobe	(Set automatically by SAE bit in DMABCRH; IOAR is invalid)	Strobe for external device
15 ETCR 0	Transfer counter		Number of transfers	See sections 7.5.2, Sequential Mode, 7.5.3, Idle Mode, and 7.5.4, Repeat Mode.

MAR specifies the start address of the transfer source or transfer destination as 24 bits. IOAR is invalid; in its place the strobe for external devices ($\overline{\text{DACK}}$) is output.

Figure 7.9 illustrates operation in single address mode (when sequential mode is specified).

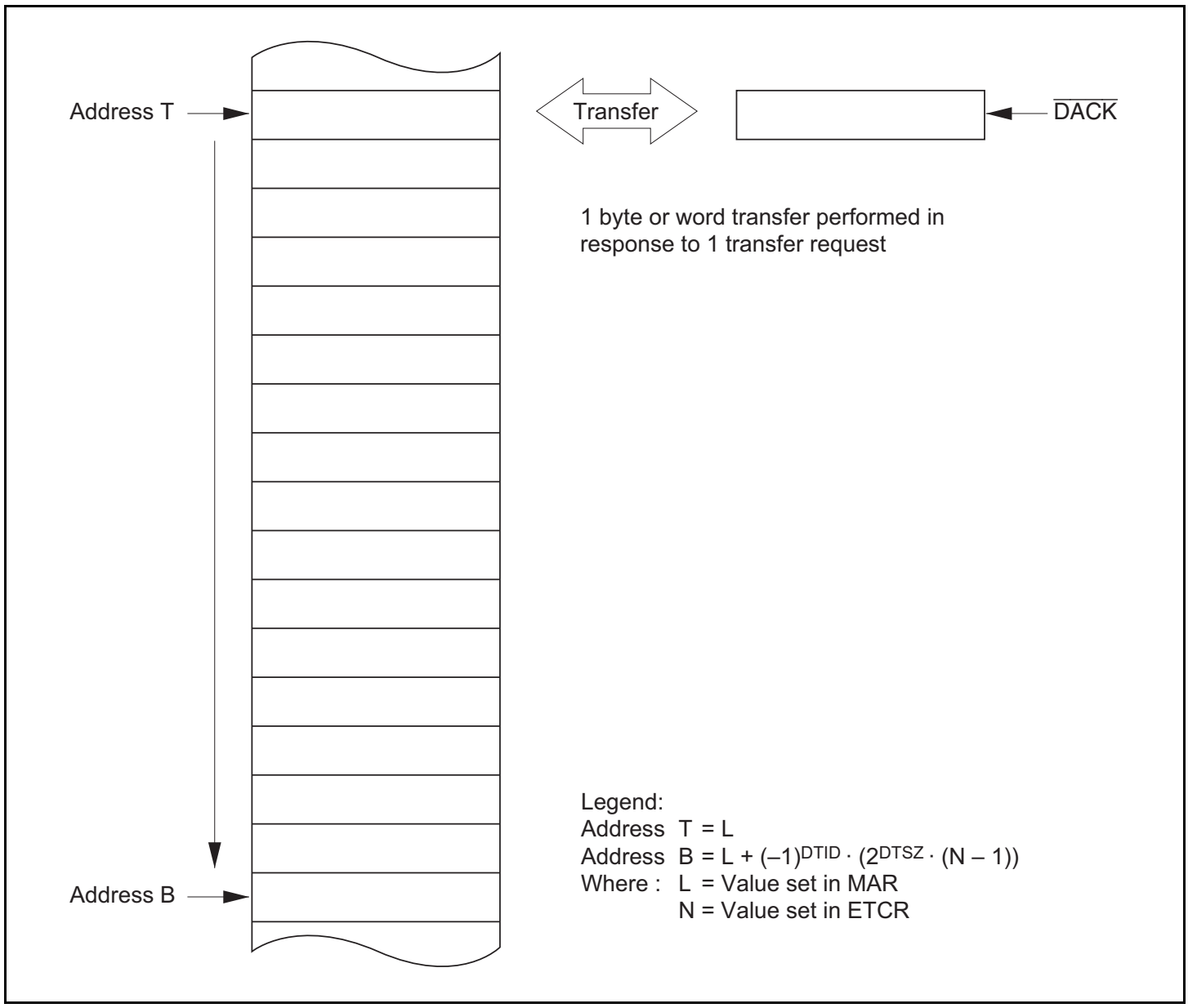
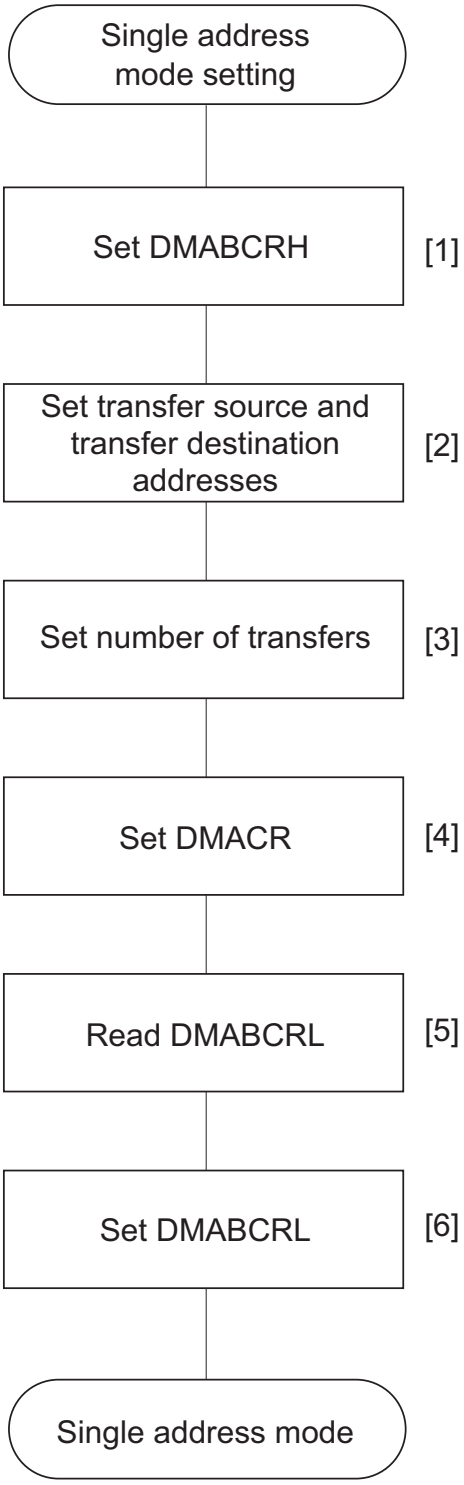


Figure 7.9 Operation in Single Address Mode (When Sequential Mode Is Specified)

Figure 7.10 shows an example of the setting procedure for single address mode (when sequential mode is specified).



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Set the SAE bit to 1 to select single address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address/transfer destination address in MAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Clear the RPE bit to 0 to select sequential mode.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts with the DTIE bit.
- Set the DTE bit to 1 to enable transfer.

Figure 7.10 Example of Single Address Mode Setting Procedure (When Sequential Mode Is Specified)

7.5.6 Normal Mode

In normal mode, transfer is performed with channels A and B used in combination. Normal mode can be specified by setting the FAE bit in DMABCRH to 1 and clearing the BLKE bit in DMACRA to 0. In normal mode, MAR is updated after data transfer of a byte or word in response to a single transfer request, and this is executed the number of times specified in ETCRA. The transfer source is specified by MARA, and the transfer destination by MARB. Table 7.9 summarizes register functions in normal mode.

Table 7.9 Register Functions in Normal Mode

Register	Function	Initial Setting	Operation
	Source address register	Start address of transfer source	Incremented/decremented every transfer, or fixed
	Destination address register	Start address of transfer destination	Incremented/decremented every transfer, or fixed
	Transfer counter	Number of transfers	Decremented every transfer; transfer ends when count reaches H'0000

MARA and MARB specify the start addresses of the transfer source and transfer destination, respectively, as 24 bits. MAR can be incremented or decremented by 1 or 2 each time a byte or word is transferred, or can be fixed. Incrementing, decrementing, or holding a fixed value can be set separately for MARA and MARB.

The number of transfers is specified by ETCRA as 16 bits. ETCRA is decremented by 1 each time a transfer is performed, and when its value reaches H'0000 the DTE bit in DMABCRL is cleared and transfer ends. If the DTIE bit in DMABCRL is set to 1 at this time, an interrupt request is sent to the CPU or DTC. The maximum number of transfers, when H'0000 is set in ETCRA, is 65,536.

Figure 7.11 illustrates operation in normal mode.

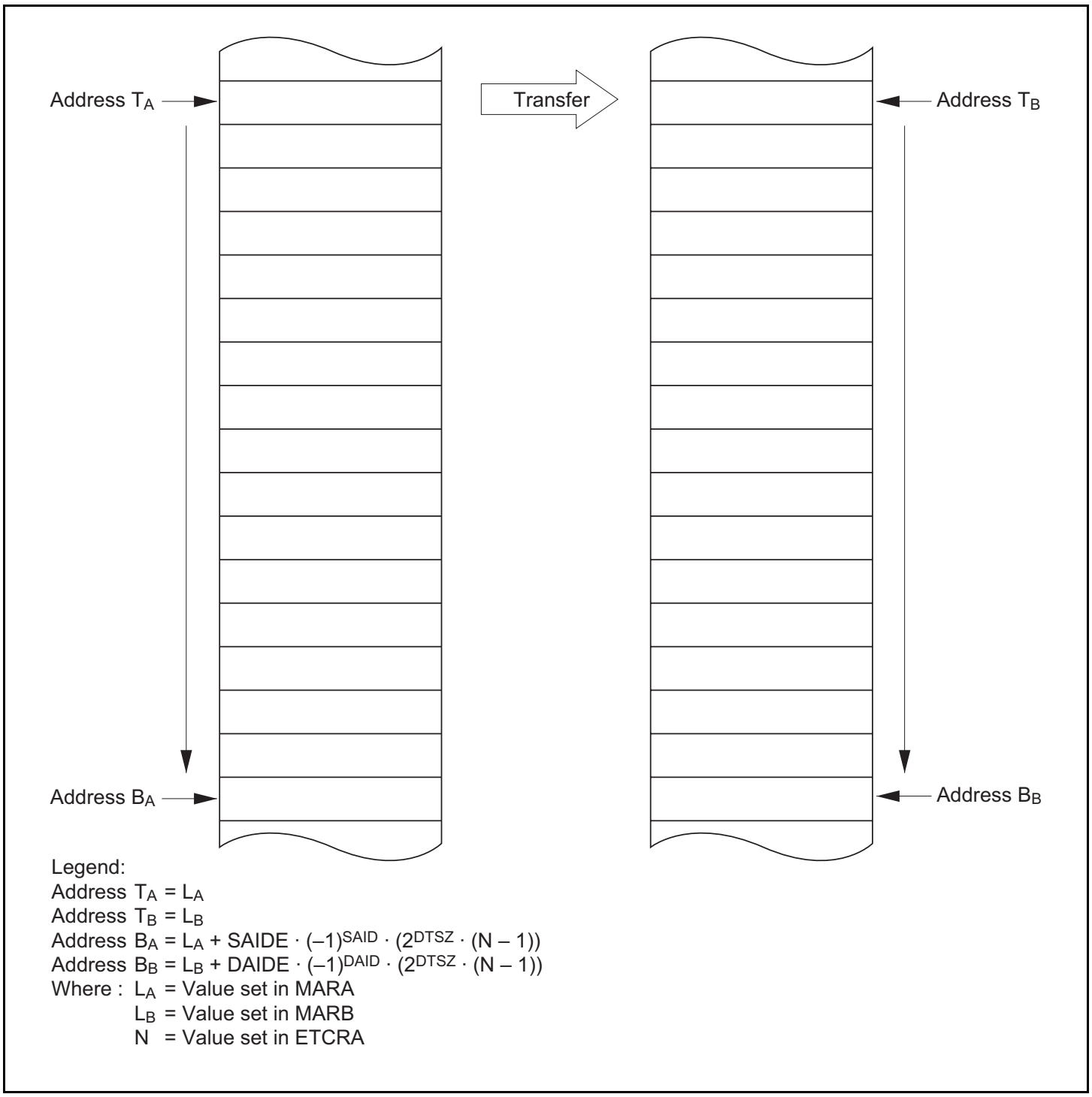
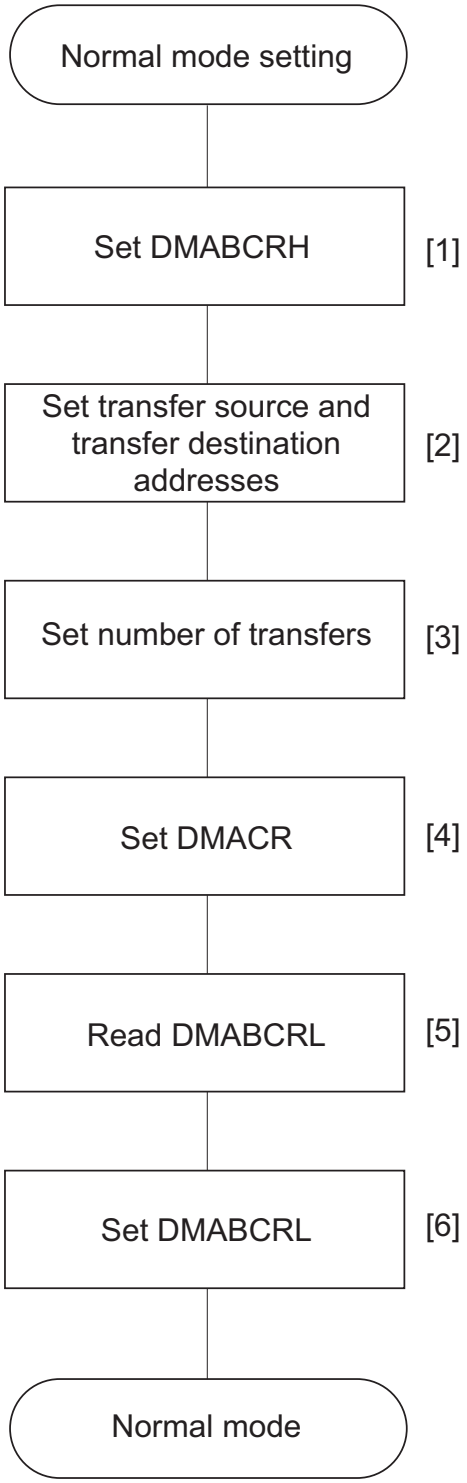


Figure 7.11 Operation in Normal Mode

Transfer requests (activation sources) are external requests, EP1FIFO full interrupt and EP2FIFO empty interrupt of the USB, and auto-requests. With auto-requests, the DMAC is only activated by register setting, and the specified number of transfers are performed automatically. With auto-requests, cycle steal mode or burst mode can be selected. In cycle steal mode, the bus is released to another bus master each time a transfer is performed. In burst mode, the bus is held continuously until transfer ends.

Figure 7.12 shows an example of the setting procedure for normal mode.



[1] Set each bit in DMABCRH.
- Set the FAE bit to 1 to select full address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address in MARA, and the transfer destination address in MARB.

[3] Set the number of transfers in ETCRA.

[4] Set each bit in DMACRA and DMACRB.
- Set the transfer data size with the DTSZ bit.
- Specify whether MARA is to be incremented, decremented, or fixed, with the SAID and SAIDE bits.
- Clear the BLKE bit to 0 to select normal mode.
- Specify whether MARB is to be incremented, decremented, or fixed, with the DAID and DAIDE bits.
- Select the activation source with bits DTF3 to DTF0.

[5] Read DTE = 0 and DTME = 0 in DMABCRL.

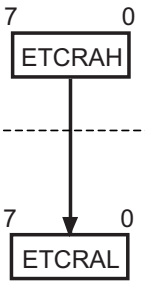
[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts with the DTIE bit.
- Set both the DTME bit and the DTE bit to 1 to enable transfer.

Figure 7.12 Example of Normal Mode Setting Procedure

7.5.7 Block Transfer Mode

In block transfer mode, data transfer is performed with channels A and B used in combination. Block transfer mode can be specified by setting the FAE bit in DMABCRH and the BLKE bit in DMACRA to 1. In block transfer mode, a data transfer of the specified block size is carried out in response to a single transfer request, and this is executed for the number of times specified in ETCRB. The transfer source is specified by MARA, and the transfer destination by MARB. Either the transfer source or the transfer destination can be selected as a block area (an area composed of a number of bytes or words). Table 7.10 summarizes register functions in block transfer mode.

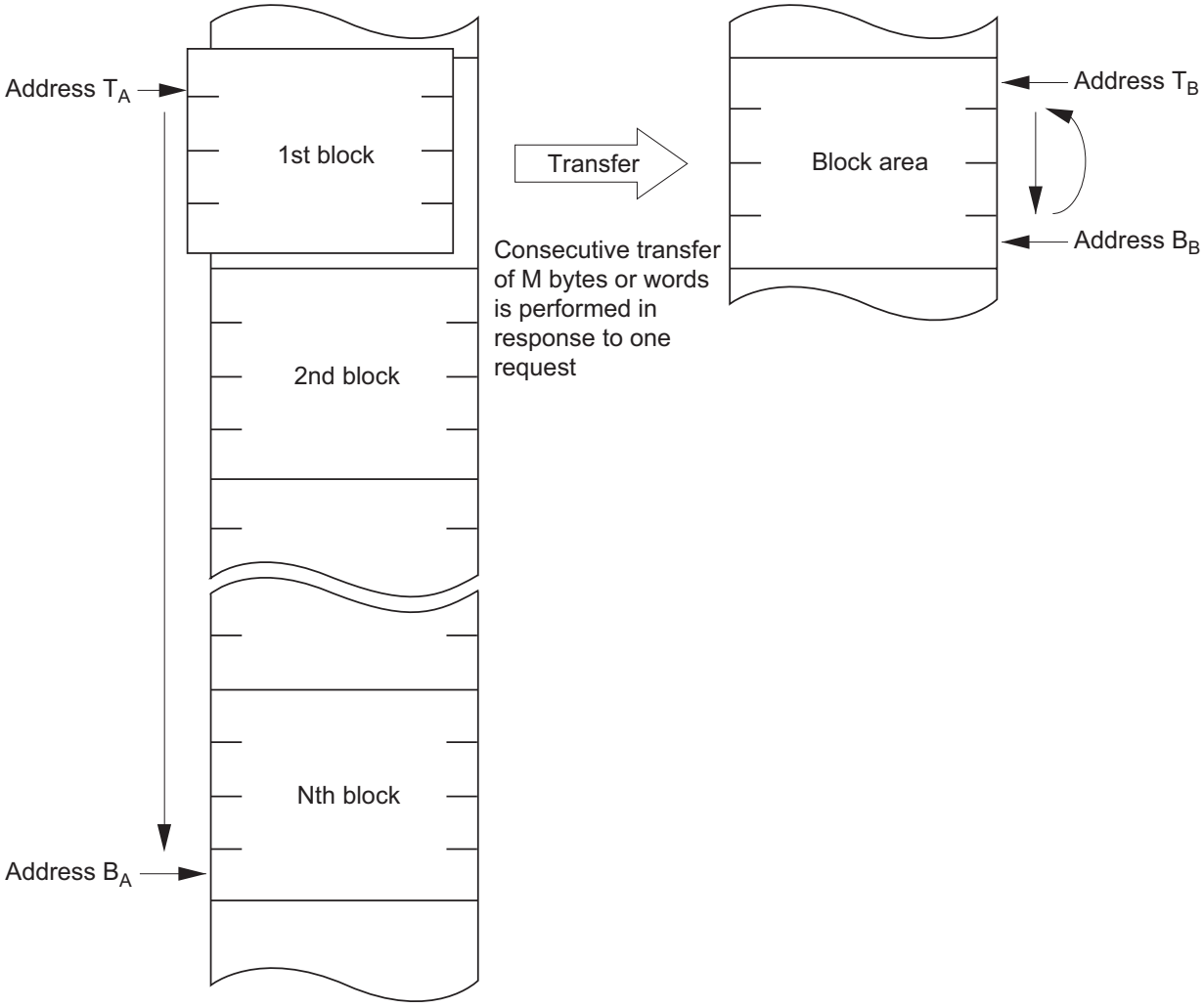
Table 7.10 Register Functions in Block Transfer Mode

Register	Function	Initial Setting	Operation
	Source address register	Start address of transfer source	Incremented/decremented every transfer, or fixed
	Destination address register	Start address of transfer destination	Incremented/decremented every transfer, or fixed
	Holds block size Block size counter	Block size Block size	Fixed Decremented every transfer; ETCRAH value copied when count reaches H'00
	Block transfer counter	Number of block transfers	Decremented every block transfer; transfer ends when count reaches H'0000

MARA and MARB specify the start addresses of the transfer source and transfer destination, respectively, as 24 bits. MAR can be incremented or decremented by 1 or 2 each time a byte or word is transferred, or can be fixed. Incrementing, decrementing, or holding a fixed value can be set separately for MARA and MARB. Whether a block is to be designated for MARA or for MARB is specified by the BLKDIR bit in DMACRA.

To specify the number of transfers, if M is the size of one block (where M = 1 to 256) and N transfers are to be performed (where N = 1 to 65,536), M is set in both ETCRAH and ETCRAL, and N in ETCRB.

Figure 7.13 illustrates operation in block transfer mode when MARB is designated as a block area.



Legend:
Address T_A = L_A
Address T_B = L_B
Address B_A = $L_A + SAIDE \cdot (-1)^{SAID} \cdot (2^{DTSZ} \cdot (M \cdot N - 1))$
Address B_B = $L_B + DAIDE \cdot (-1)^{DAID} \cdot (2^{DTSZ} \cdot (N - 1))$
Where : L_A = Value set in MARA
L_B = Value set in MARB
N = Value set in ETCRB
M = Value set in ETCRAH and ETCRAL

Figure 7.13 Operation in Block Transfer Mode (BLKDIR = 0)

Figure 7.14 illustrates operation in block transfer mode when MARA is designated as a block area.

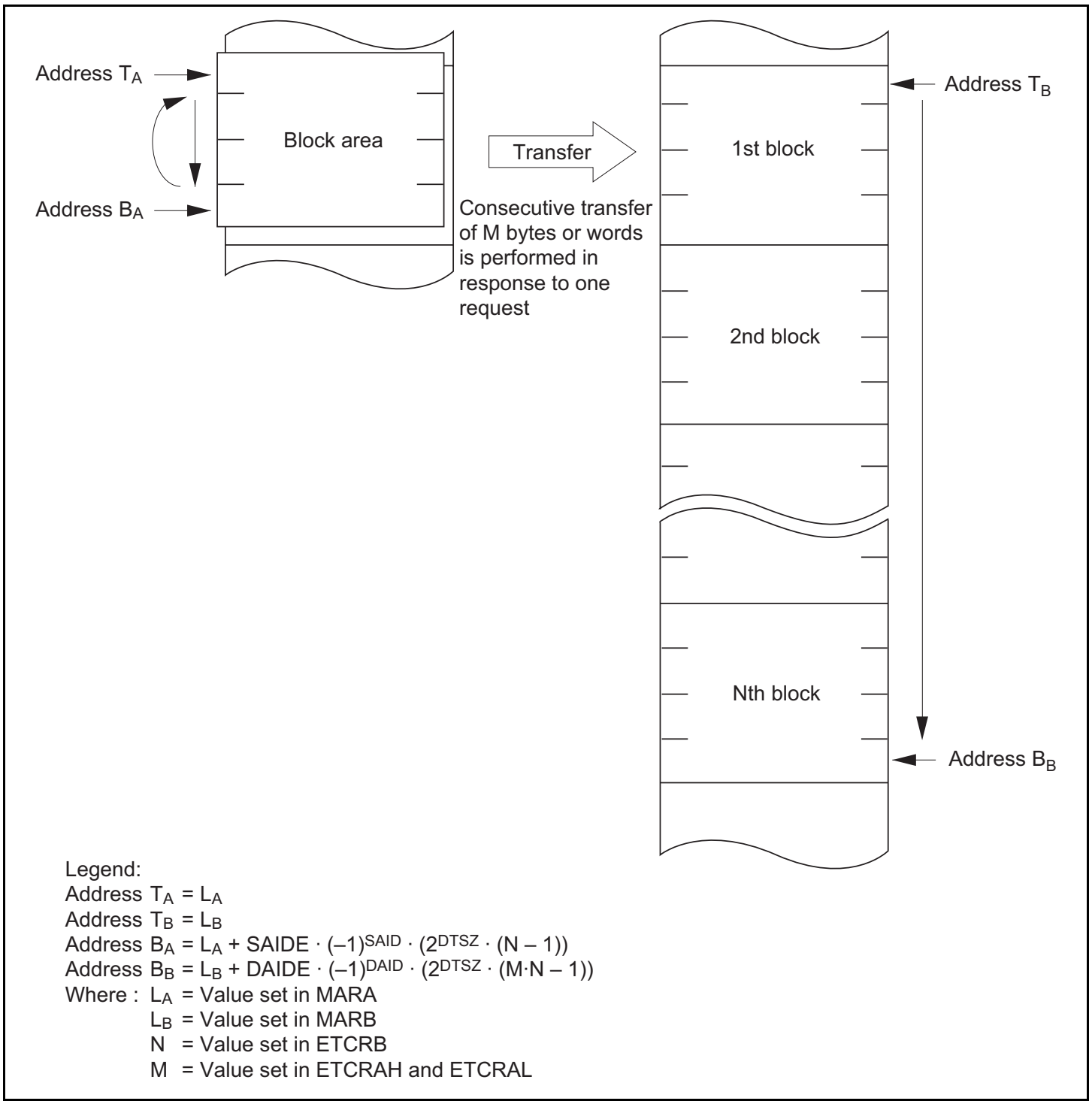


Figure 7.14 Operation in Block Transfer Mode (BLKDIR = 1)

ETCRAL is decremented by 1 each time a byte or word transfer is performed. In response to a single transfer request, burst transfer is performed until the value in ETCRAL reaches H'00. ETCRAL is then loaded with the value in ETCRAH. At this time, the value in the MAR register for which a block designation has been given by the BLKDIR bit in DMACRA is restored in accordance with the DTSZ, SAID/DAID, and SAIDE/DAIDE bits in DMACR.

ETCRB is decremented by 1 after every block transfer, and when the count reaches H'0000 the DTE bit in DMABCRL is cleared and transfer ends. If the DTIE bit in DMABCRL is set to 1 at this point, an interrupt request is sent to the CPU or DTC.

Figure 7.15 shows the operation flow in block transfer mode.

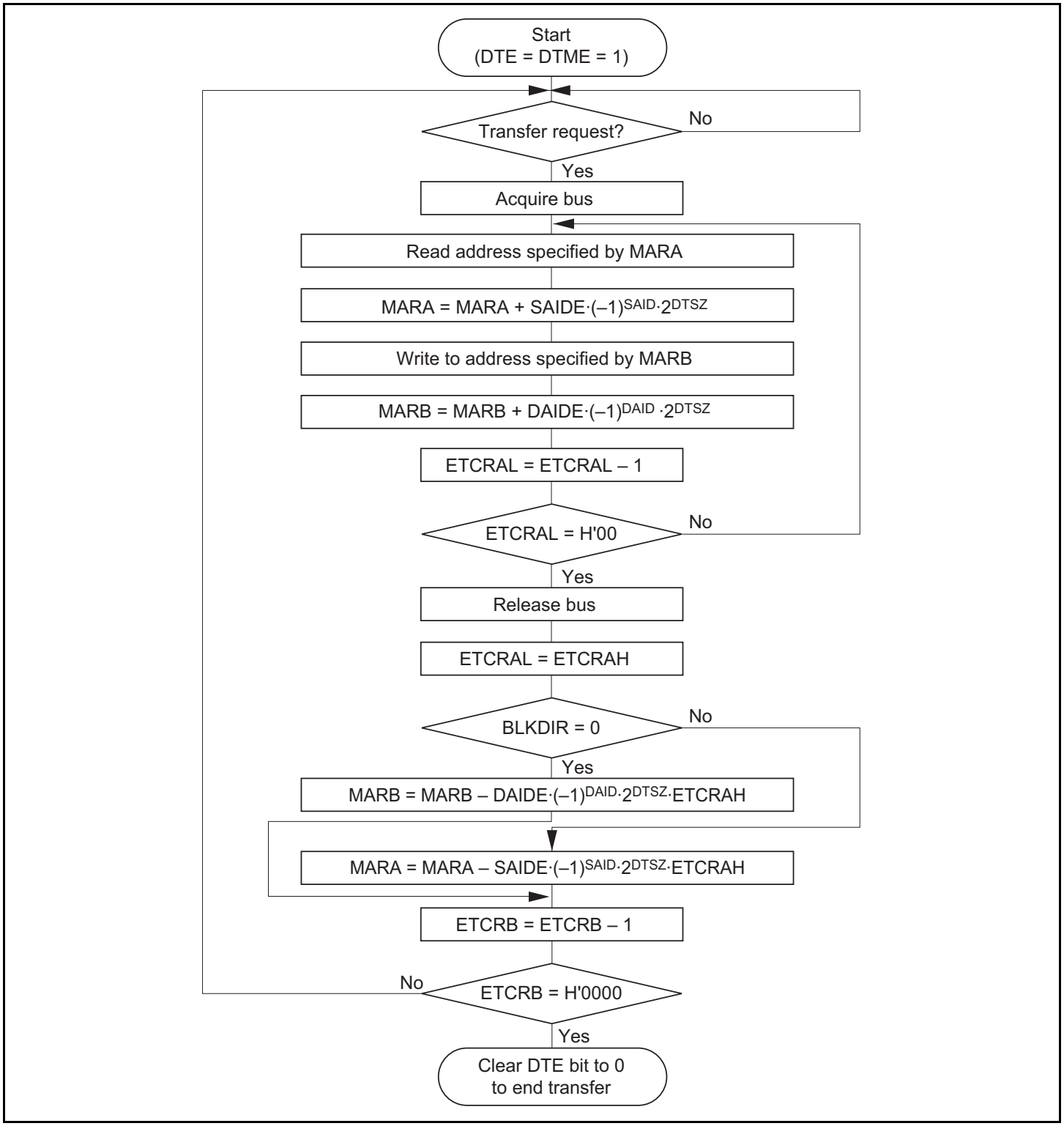


Figure 7.15 Operation Flow in Block Transfer Mode

Transfer requests (activation sources) consist of A/D converter conversion end interrupts, external requests, SCI transmission complete and reception complete interrupts, EP1FIFO full interrupt and EP2FIFO empty interrupt of the USB, and TPU channel 0 to 5 compare match/input capture A interrupts.

Figure 7.16 shows an example of the setting procedure for block transfer mode.

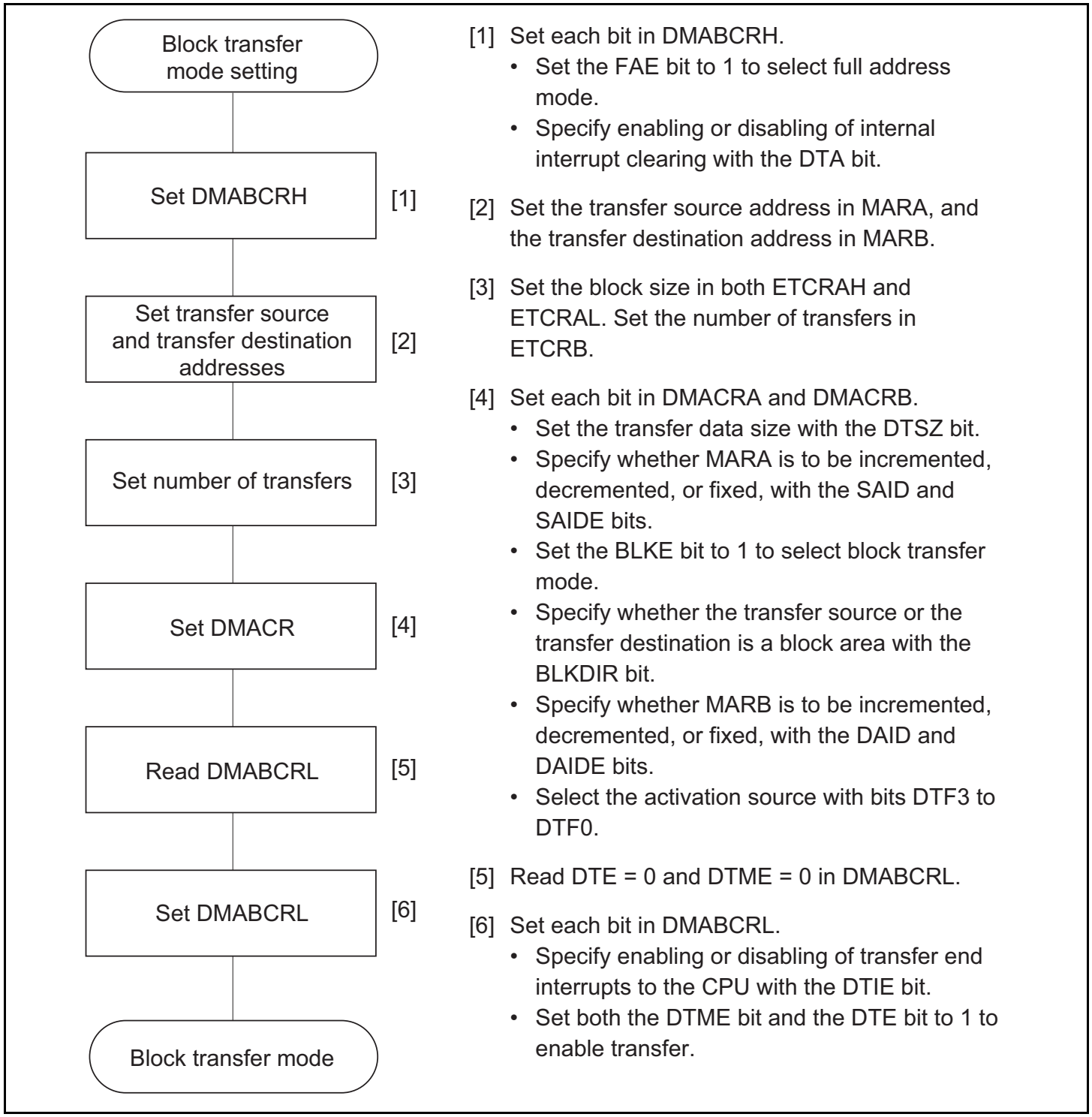


Figure 7.16 Example of Block Transfer Mode Setting Procedure

7.5.8 Basic Bus Cycles

An example of the basic DMAC bus cycle timing is shown in figure 7.17. In this example, word-size transfer is performed from 16-bit, 2-state access space to 8-bit, 3-state access space. When the bus is transferred from the CPU to the DMAC, a source address read and destination address write are performed. The bus is not released in response to another bus request, etc., between these read and write operations. As like CPU cycles, DMA cycles conform to the bus controller settings.

The address is not output to the external address bus in an access to on-chip memory or an internal I/O register.

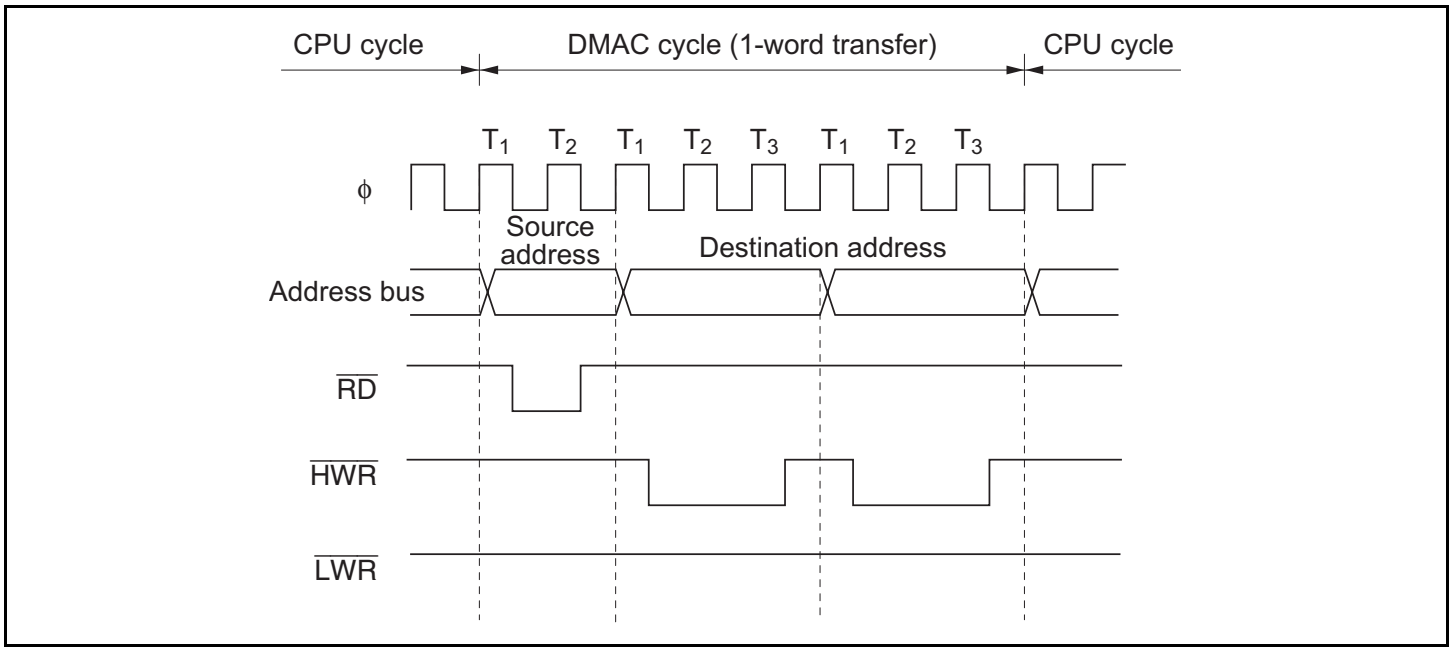


Figure 7.17 Example of DMA Transfer Bus Timing

7.5.9 DMA Transfer (Dual Address Mode) Bus Cycles

(1) Short Address Mode

Figure 7.18 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and byte-size short address mode transfer (sequential/idle/repeat mode) is performed from external 8-bit, 2-state access space to internal I/O space.

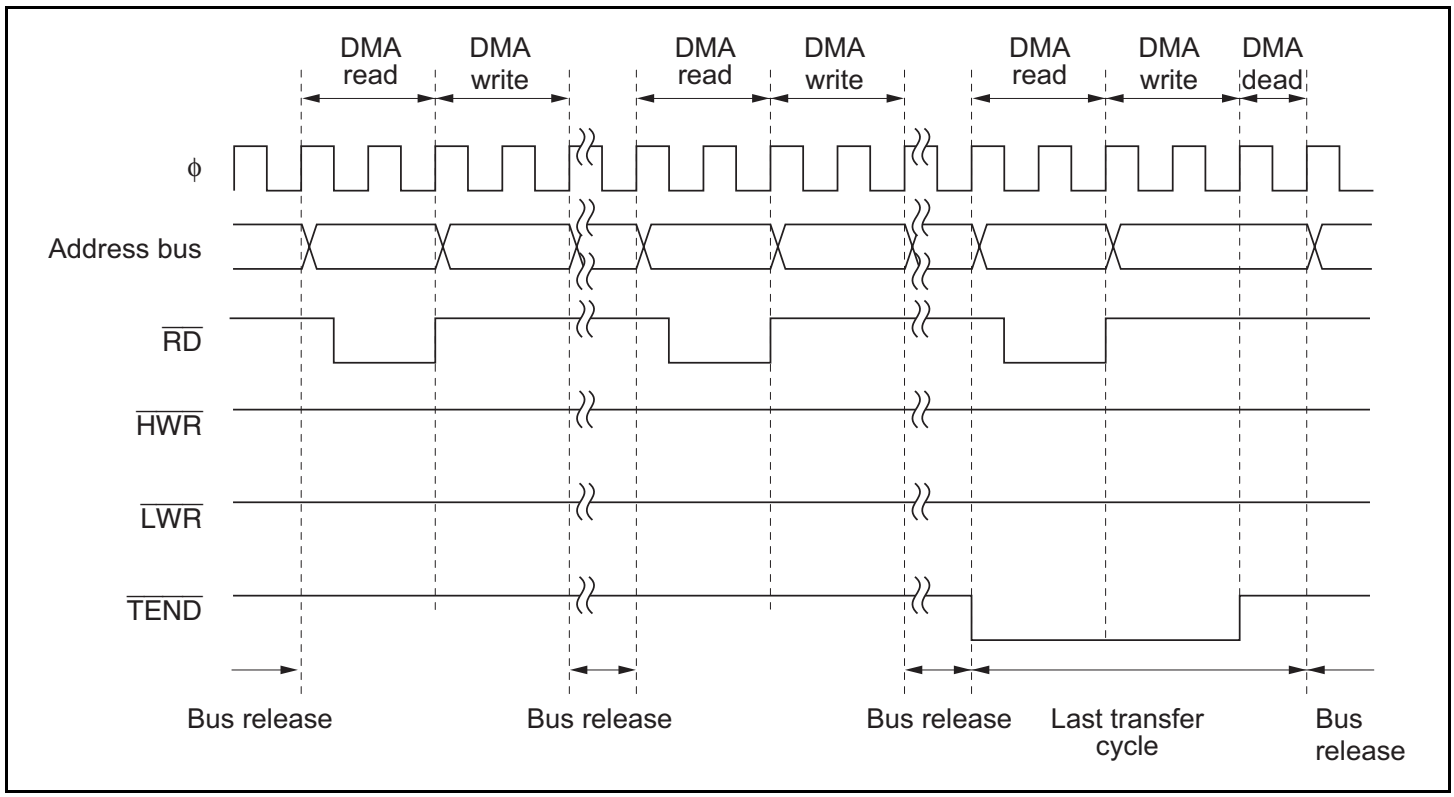


Figure 7.18 Example of Short Address Mode Transfer

A byte or word transfer is performed for a single transfer request, and after the transfer, the bus is released. While the bus is released, one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

In repeat mode, when $\overline{\text{TEND}}$ output is enabled, $\overline{\text{TEND}}$ output goes low in the transfer end cycle.

(2) Full Address Mode (Cycle Steal Mode)

Figure 7.19 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and word-size full address mode transfer (cycle steal mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

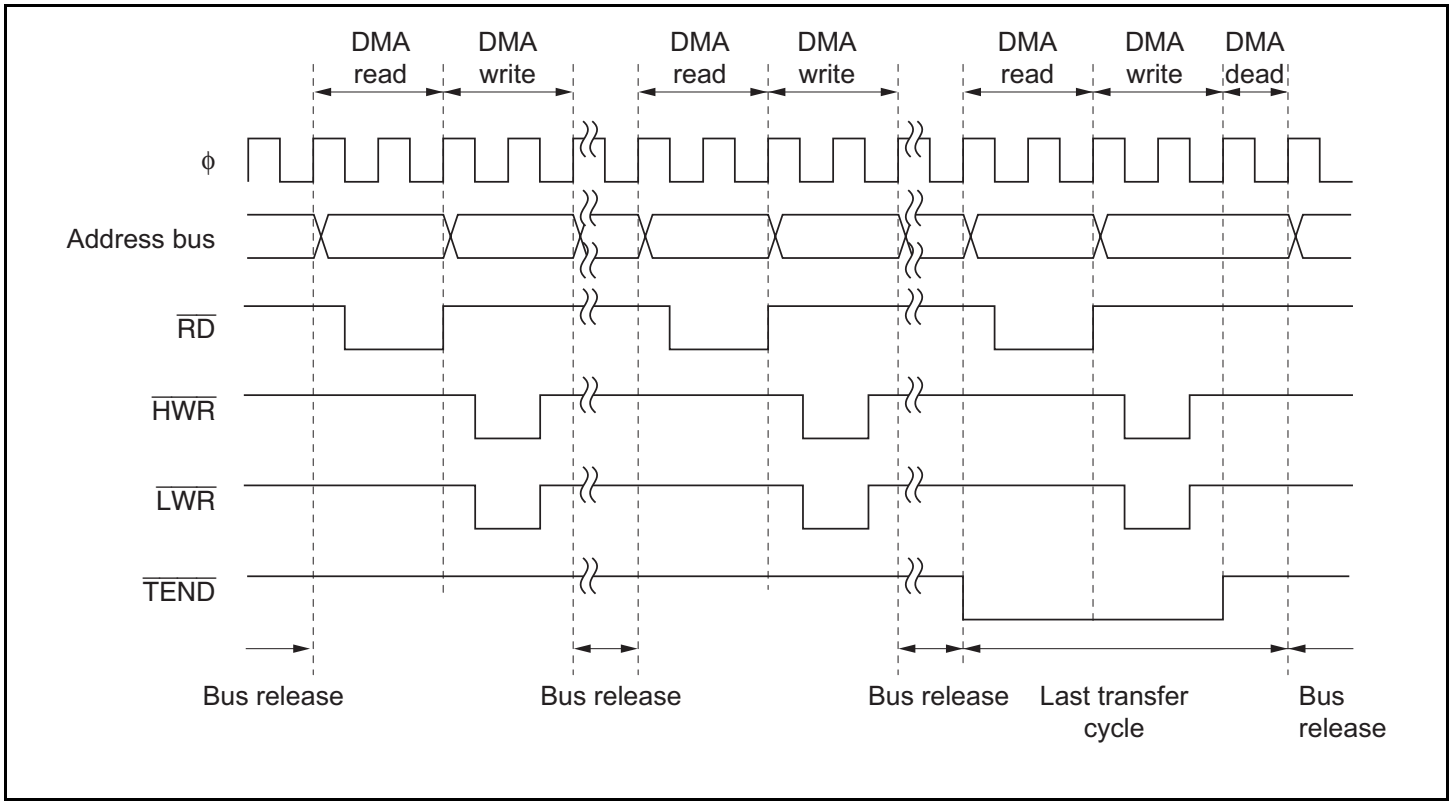


Figure 7.19 Example of Full Address Mode Transfer (Cycle Steal)

A byte or word transfer is performed for a single transfer request, and after the transfer, the bus is released. While the bus is released, one bus cycle is executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

(3) Full Address Mode (Burst Mode)

Figure 7.20 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and word-size full address mode transfer (burst mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

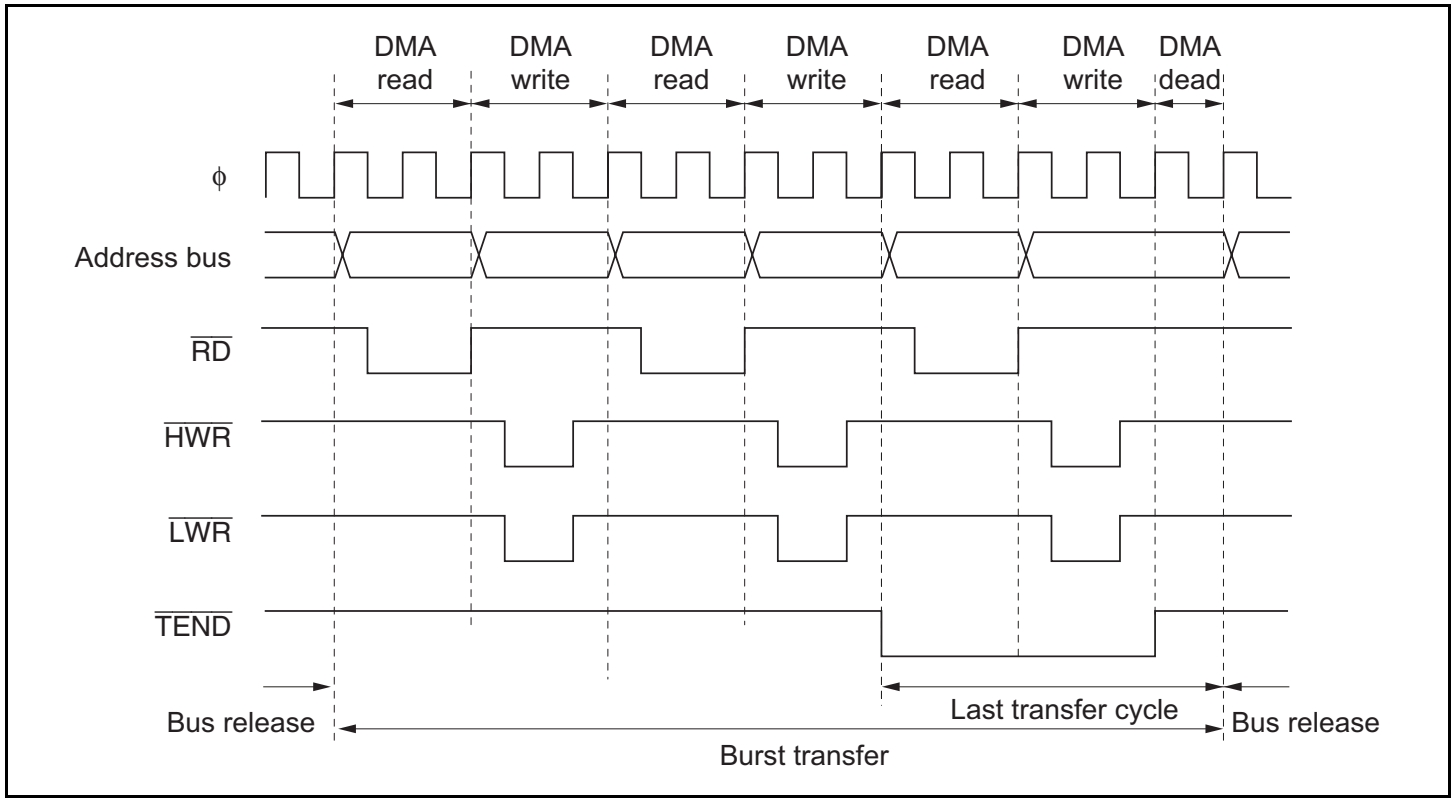


Figure 7.20 Example of Full Address Mode Transfer (Burst Mode)

In burst mode, one-byte or one-word transfers are executed consecutively until transfer ends.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

If a request from another higher-priority channel is generated after burst transfer starts, that channel has to wait until the burst transfer ends.

If an NMI interrupt is generated while a channel designated for burst transfer is in the transfer enabled state, the DTME bit in DMABCRL is cleared and the channel is placed in the transfer disabled state. If burst transfer has already been activated inside the DMAC, the bus is released on completion of a one-byte or one-word transfer within the burst transfer, and burst transfer is suspended. If the last transfer cycle of the burst transfer has already been activated inside the DMAC, execution continues to the end of the transfer even if the DTME bit is cleared.

(4) Full Address Mode (Block Transfer Mode)

Figure 7.21 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and word-size full address mode transfer (block transfer mode) is performed from internal 16-bit, 1-state access space to external 16-bit, 2-state access space.

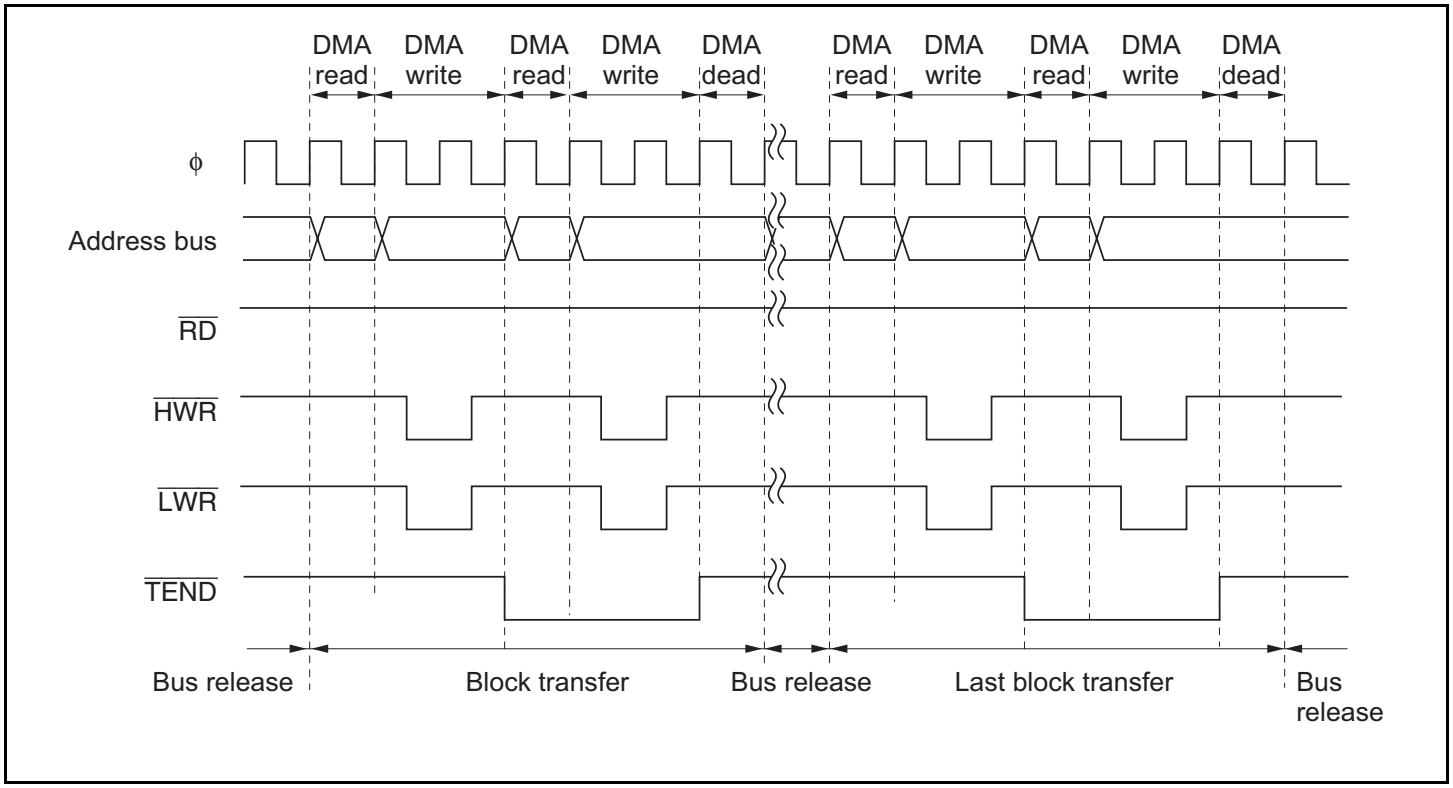


Figure 7.21 Example of Full Address Mode Transfer (Block Transfer Mode)

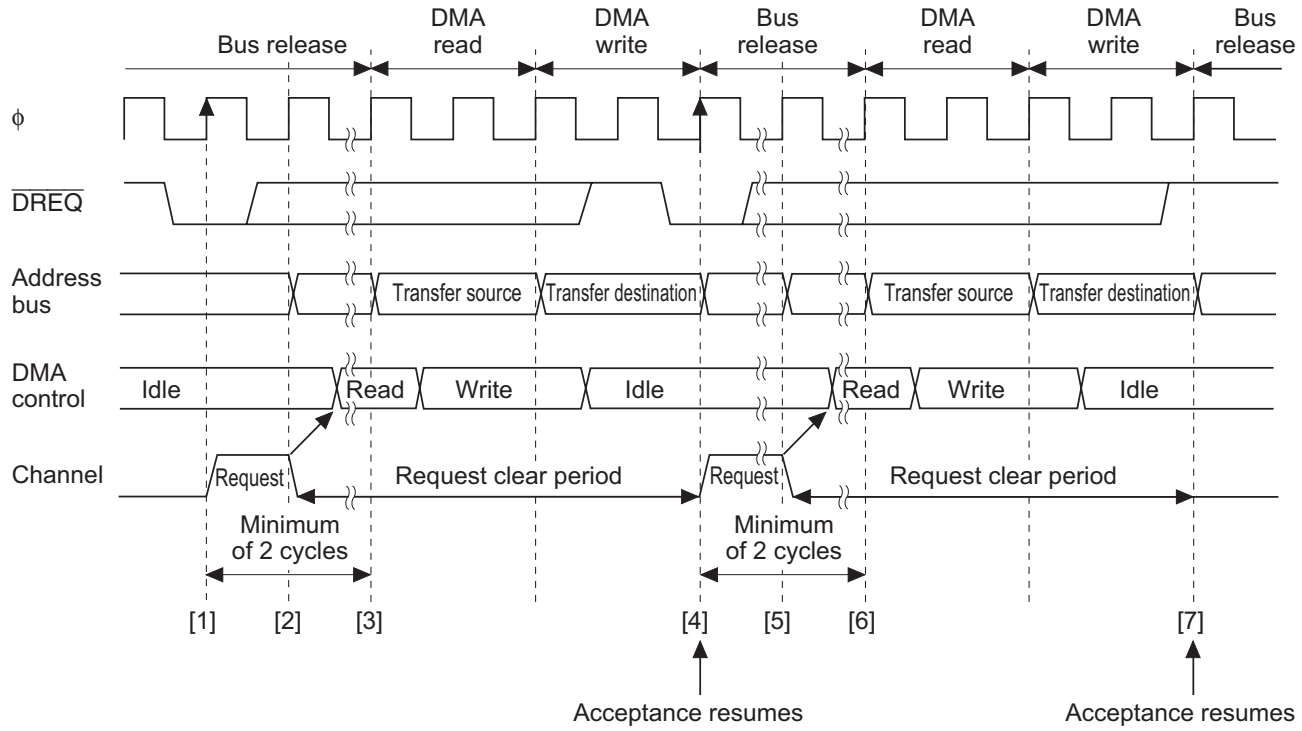
A one-block transfer is performed for a single transfer request, and after the transfer the bus is released. While the bus is released, one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle of each block (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle. Even if an NMI interrupt is generated during data transfer, block transfer operation is not affected until data transfer for one block has ended.

(5) $\overline{\text{DREQ}}$ Pin Falling Edge Activation Timing

Set the DTA bit in DMABCRH to 1 for the channel for which the $\overline{\text{DREQ}}$ pin is selected.

Figure 7.22 shows an example of normal mode transfer activated by the $\overline{\text{DREQ}}$ pin falling edge.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of φ starts.

[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the write cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.)

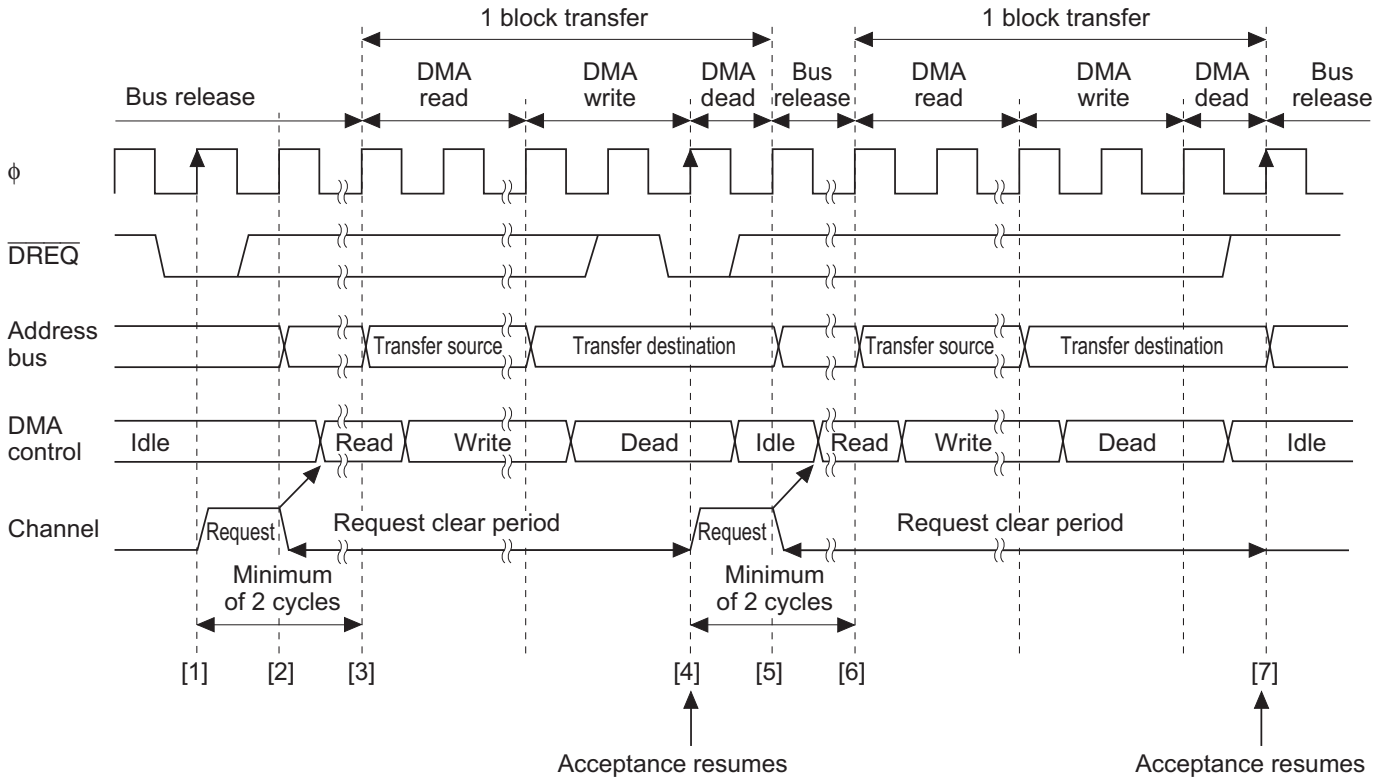
Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.22 Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Normal Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next φ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and $\overline{\text{DREQ}}$ pin high level sampling for edge detection is started. If $\overline{\text{DREQ}}$ pin high level sampling has been completed by the time the DMA write cycle ends, acceptance resumes after the end of the write cycle, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

Figure 7.23 shows an example of block transfer mode transfer activated by the $\overline{\text{DREQ}}$ pin falling edge.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of φ starts.

[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the dead cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.23 Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Block Transfer Mode Transfer

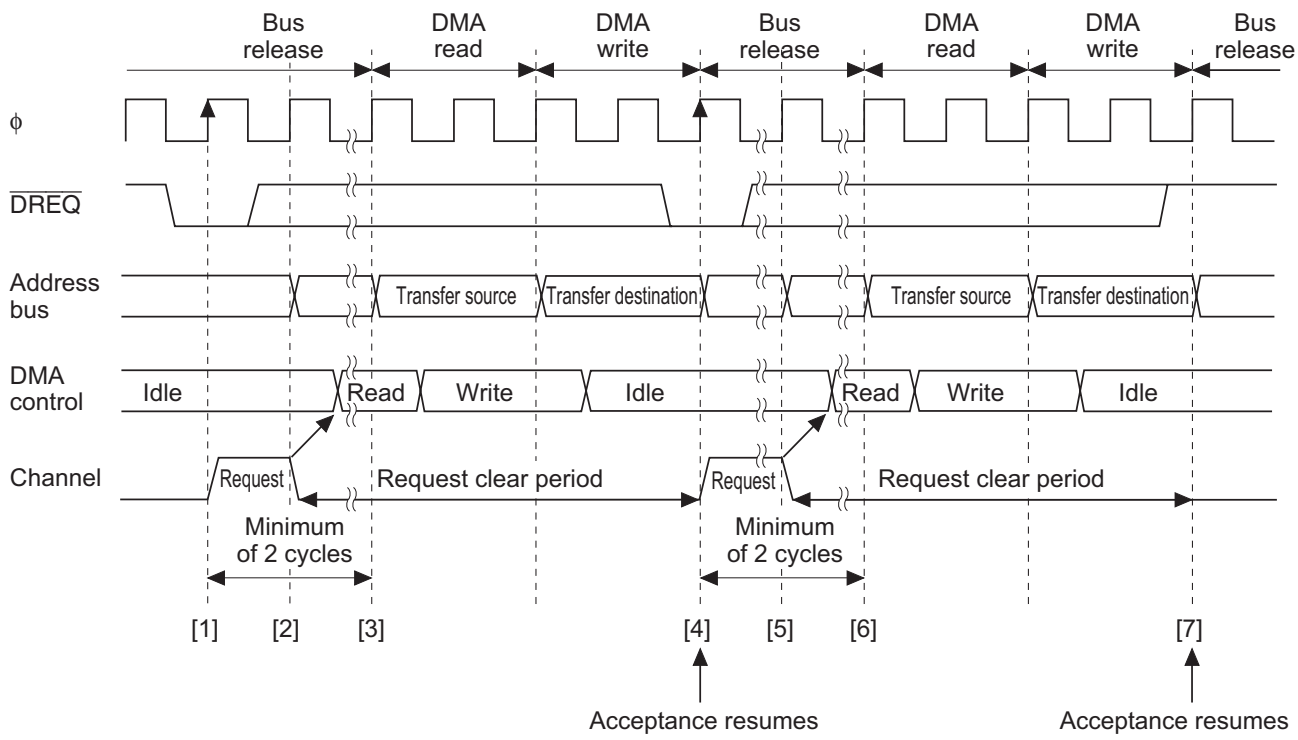
$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next φ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and $\overline{\text{DREQ}}$ pin high level sampling for edge detection is started. If $\overline{\text{DREQ}}$ pin high level sampling has been completed by the time the DMA dead cycle ends, acceptance resumes after the end of the dead cycle, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

(6) $\overline{\text{DREQ}}$ Pin Low Level Activation Timing (Normal Mode)

Set the DTA bit in DMABCRH to 1 for the channel for which the $\overline{\text{DREQ}}$ pin is selected.

Figure 7.24 shows an example of normal mode transfer activated by the $\overline{\text{DREQ}}$ pin low level.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] The DMA cycle is started.

[4] [7] Acceptance is resumed after the write cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.)

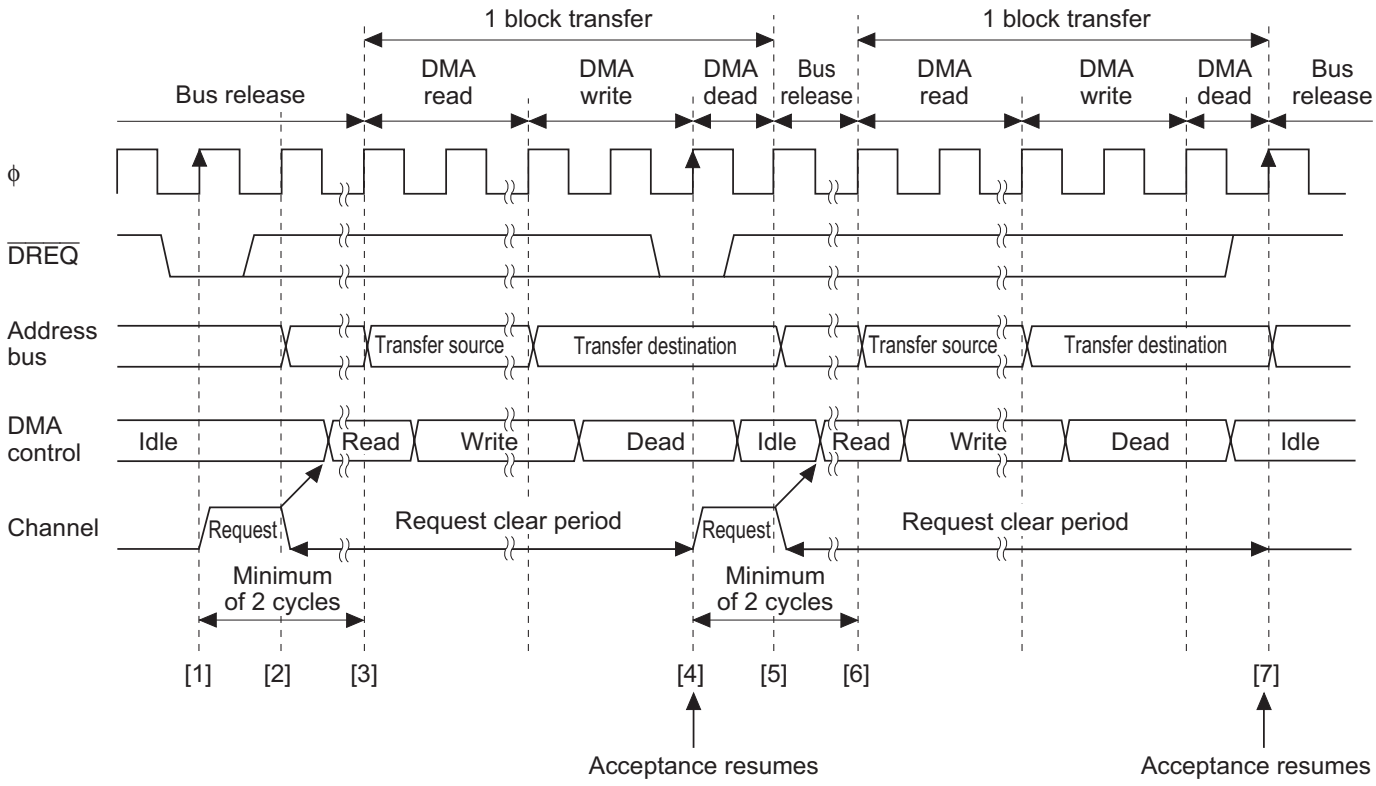
Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.24 Example of $\overline{\text{DREQ}}$ Pin Low Level Activated Normal Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next φ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. After the end of the write cycle, acceptance resumes, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

Figure 7.25 shows an example of block transfer mode transfer activated by $\overline{\text{DREQ}}$ pin low level.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] The DMA cycle is started.

[4] [7] Acceptance is resumed after the dead cycle is completed. (As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.25 Example of $\overline{\text{DREQ}}$ Pin Low Level Activated Block Transfer Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next φ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. After the end of the dead cycle, acceptance resumes, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

7.5.10 DMA Transfer (Single Address Mode) Bus Cycles

(1) Single Address Mode (Read)

Figure 7.26 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and byte-size single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.

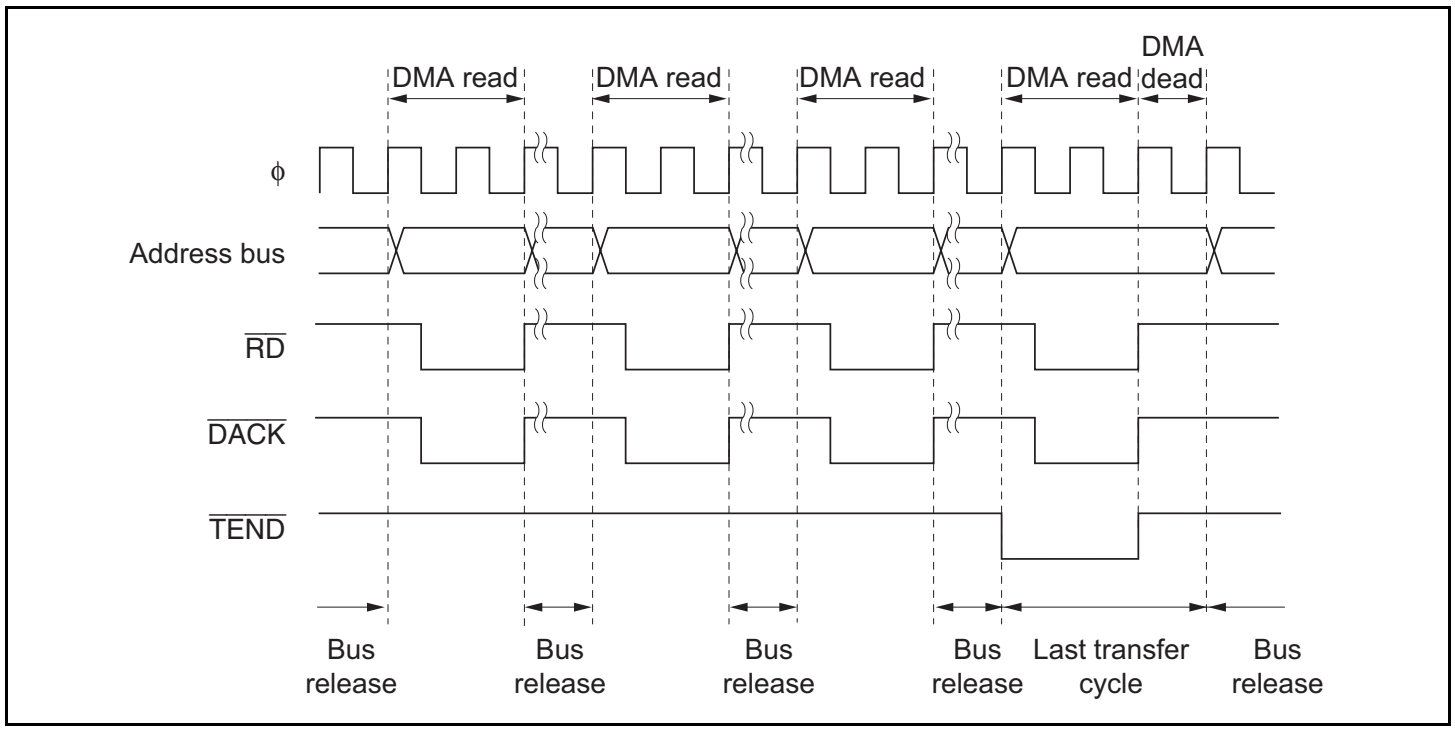


Figure 7.26 Example of Single Address Mode Transfer (Byte Read)

Figure 7.27 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and word-size single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.

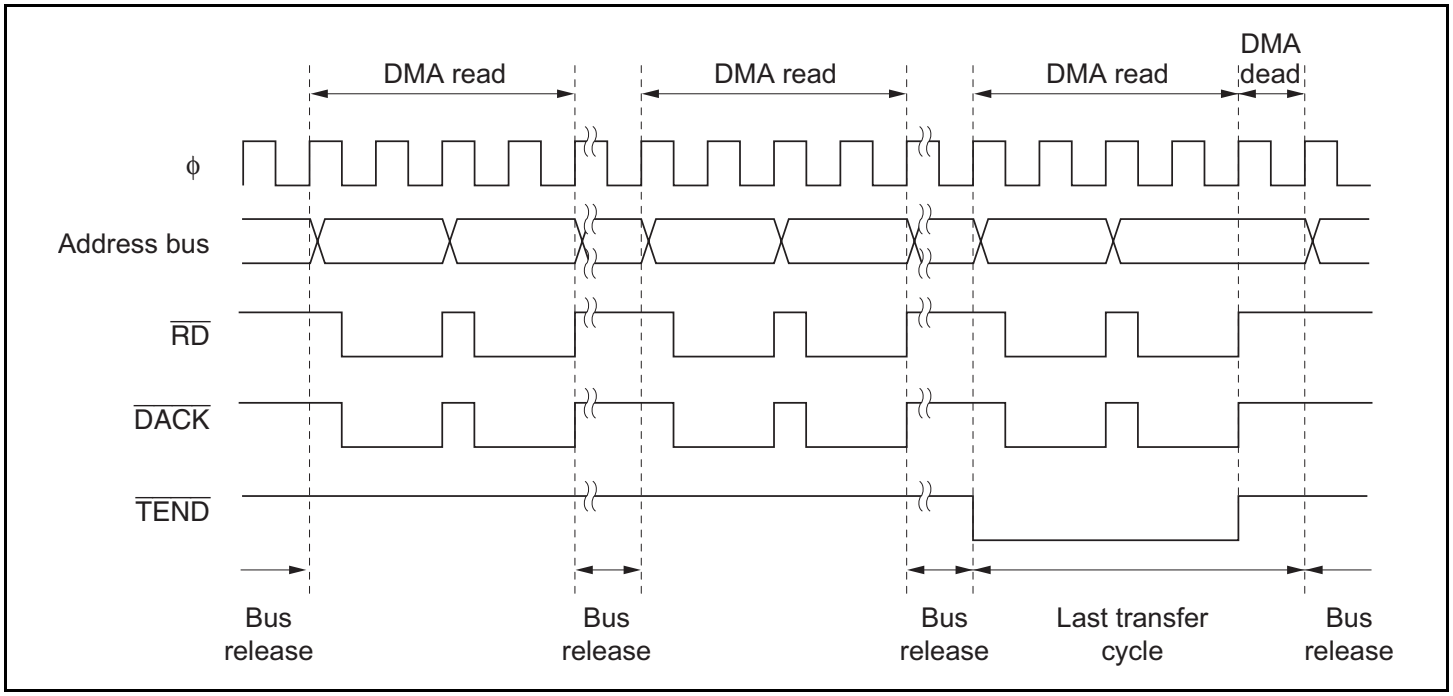


Figure 7.27 Example of Single Address Mode (Word Read) Transfer

A byte or word transfer is performed for a single transfer request, and after the transfer, the bus is released. While the bus is released, one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

(2) Single Address Mode (Write)

Figure 7.28 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and byte-size single address mode transfer (write) is performed from an external device to external 8-bit, 2-state access space.

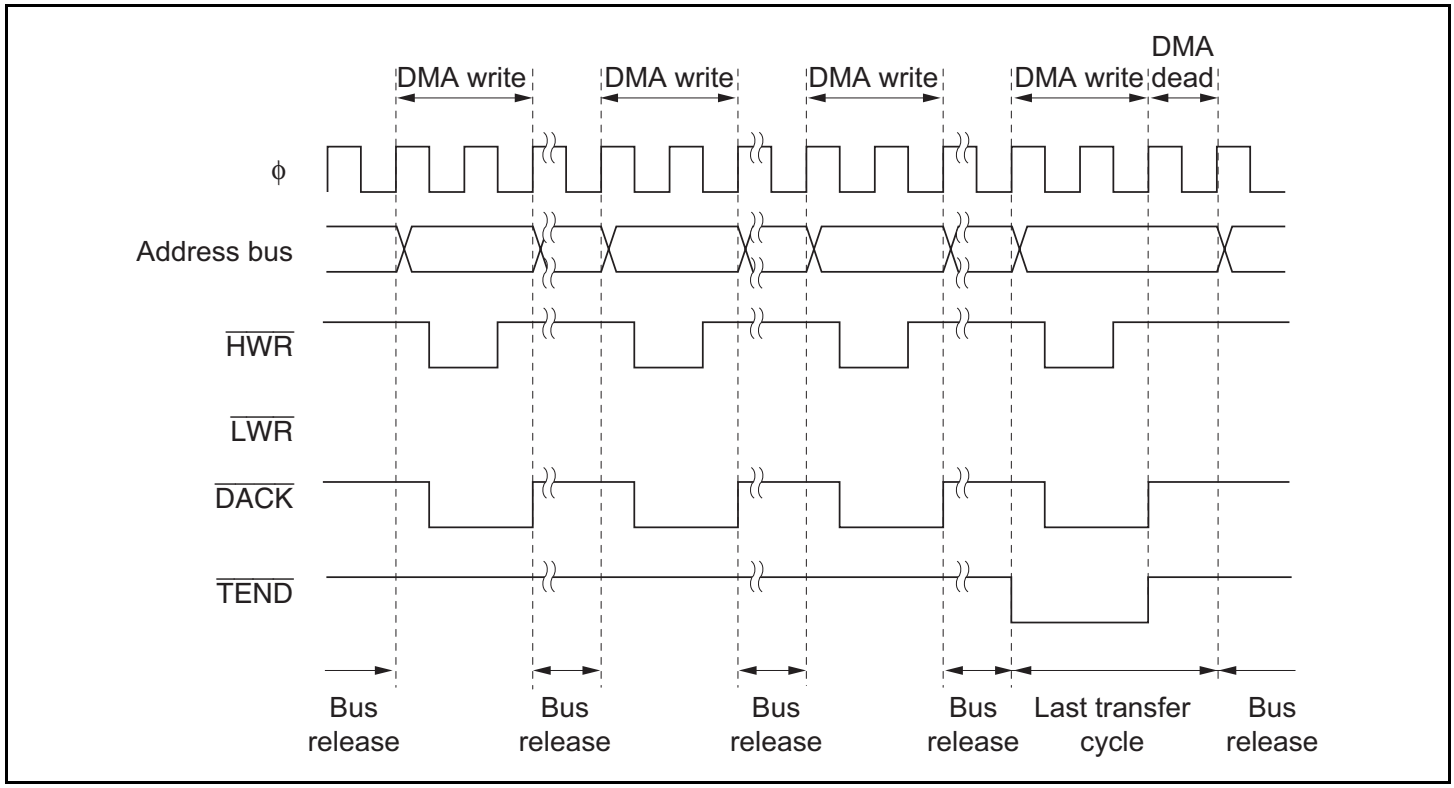


Figure 7.28 Example of Single Address Mode Transfer (Byte Write)

Figure 7.29 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and word-size single address mode transfer (write) is performed from an external device to external 8-bit, 2-state access space.

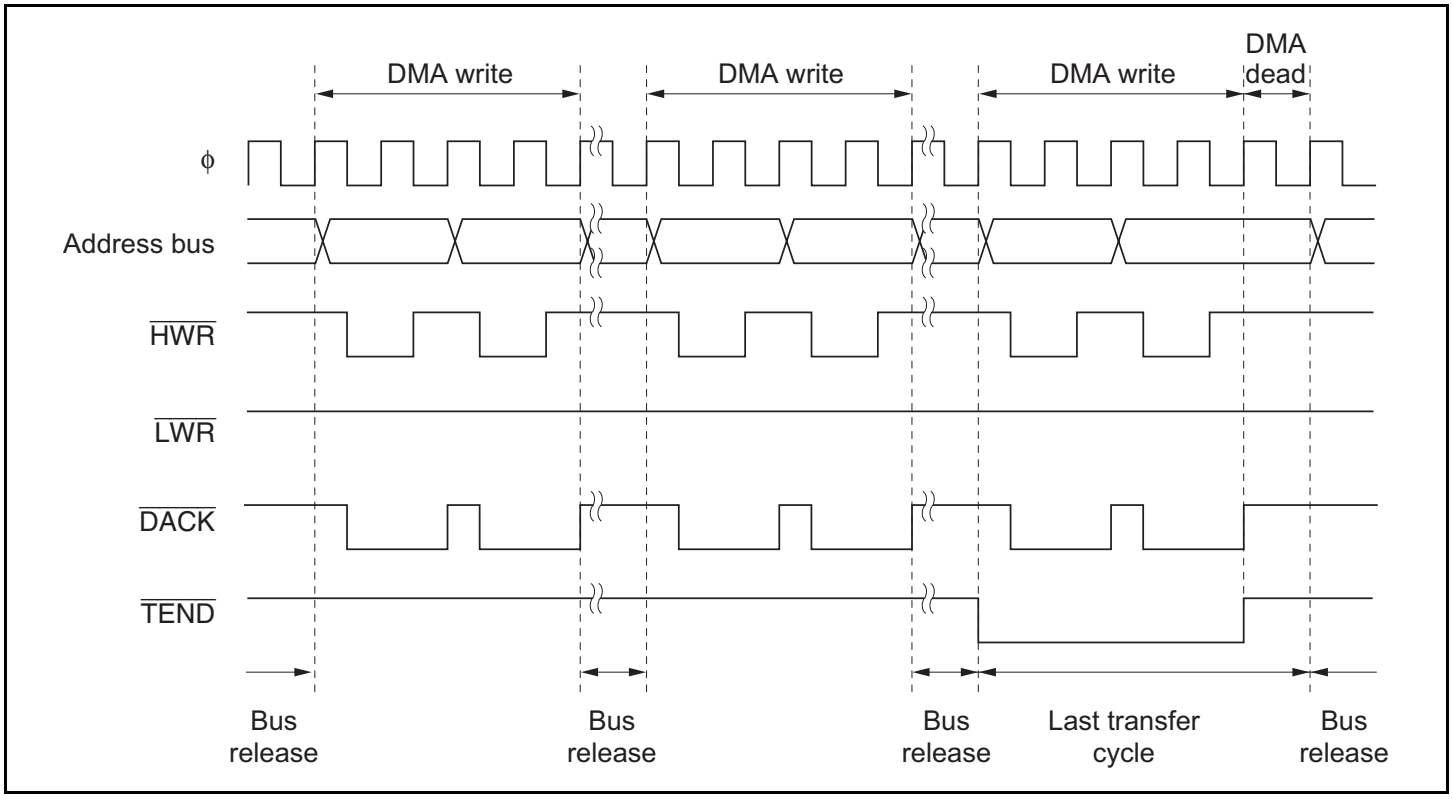


Figure 7.29 Example of Single Address Mode Transfer (Word Write)

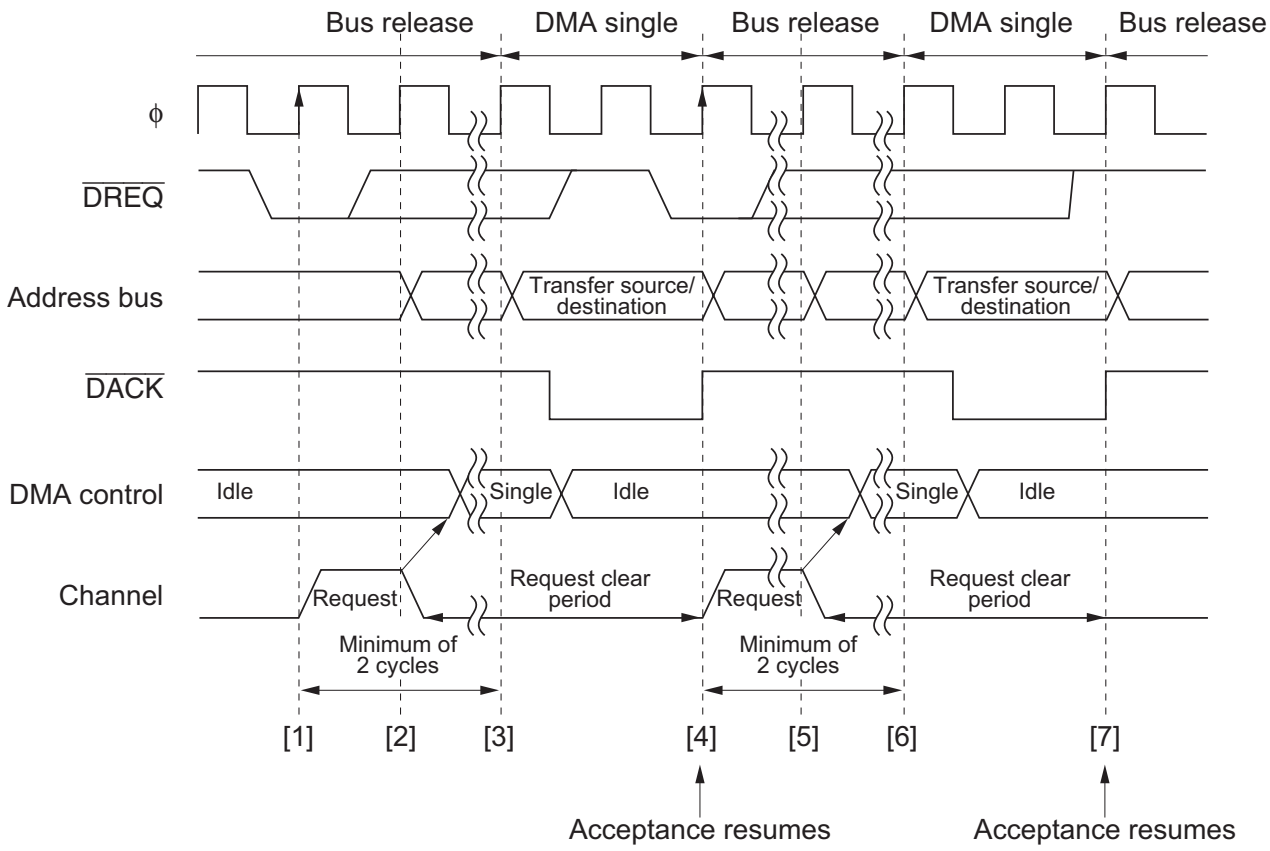
A byte or word transfer is performed for a single transfer request, and after the transfer, the bus is released. While the bus is released, one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

(3) $\overline{\text{DREQ}}$ Pin Falling Edge Activation Timing

Set the DTA bit in DMABCRH to 1 for the channel for which the $\overline{\text{DREQ}}$ pin is selected.

Figure 7.30 shows an example of single address mode transfer activated by the $\overline{\text{DREQ}}$ pin falling edge.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of φ starts.

[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the single cycle is completed. (As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.30 Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Single Address Mode Transfer

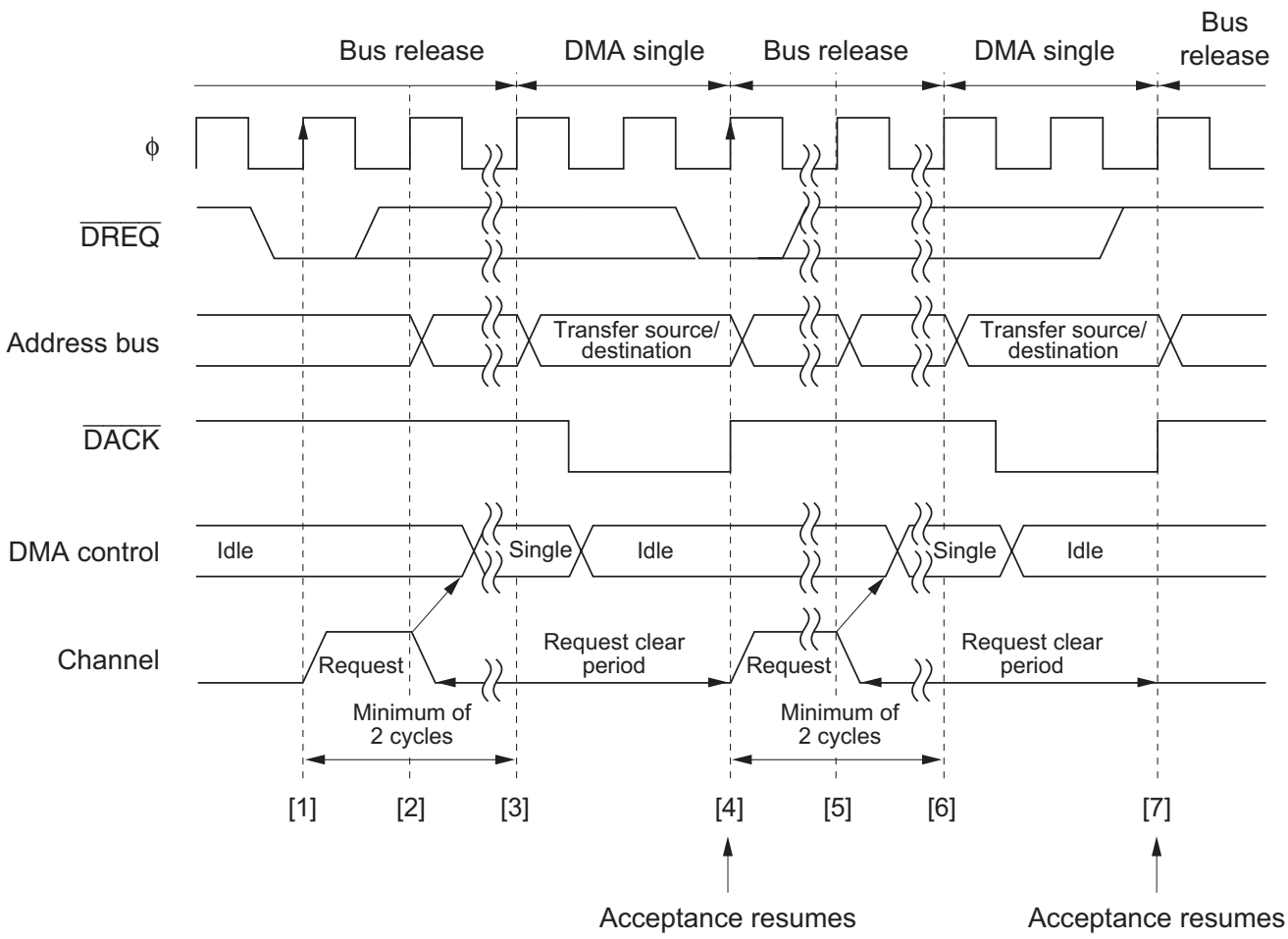
$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next φ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and $\overline{\text{DREQ}}$ pin high level sampling for edge detection is started. If $\overline{\text{DREQ}}$ pin high level sampling has been completed by the time the DMA single cycle ends, acceptance resumes after the end of the single cycle, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

(4) $\overline{\text{DREQ}}$ Pin Low Level Activation Timing

Set the DTA bit in DMABCRH to 1 for the channel for which the $\overline{\text{DREQ}}$ pin is selected.

Figure 7.31 shows an example of single address mode transfer activated by the $\overline{\text{DREQ}}$ pin low level.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.
[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
[3] [6] The DMAC cycle is started.
[4] [7] Acceptance is resumed after the single cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.31 Example of $\overline{\text{DREQ}}$ Pin Low Level Activated Single Address Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next φ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. After the end of the single cycle, acceptance resumes, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

7.5.11 Write Data Buffer Function

DMAC internal-to-external dual address transfers and single address transfers can be executed at high speed using the write data buffer function, enabling system throughput to be improved.

When the WDBE bit of BCR in the bus controller is set to 1, enabling the write data buffer function, dual address transfer external write cycles or single address transfer and internal accesses (on-chip memory or internal I/O registers) are executed in parallel. Internal accesses are independent of the bus mastership, and DMAC dead cycles are regarded as internal accesses.

A low level can always be output from the $\overline{\text{TEND}}$ pin if the bus cycle in which a low level is to be output from the $\overline{\text{TEND}}$ pin is an external bus cycle. However, a low level is not output from the $\overline{\text{TEND}}$ pin if the bus cycle in which a low level is to be output from the $\overline{\text{TEND}}$ pin is an internal bus cycle, and an external write cycle is executed in parallel with this cycle.

Figure 7.32 shows an example of dual address transfer using the write data buffer function. The data is transferred from on-chip RAM to external memory.

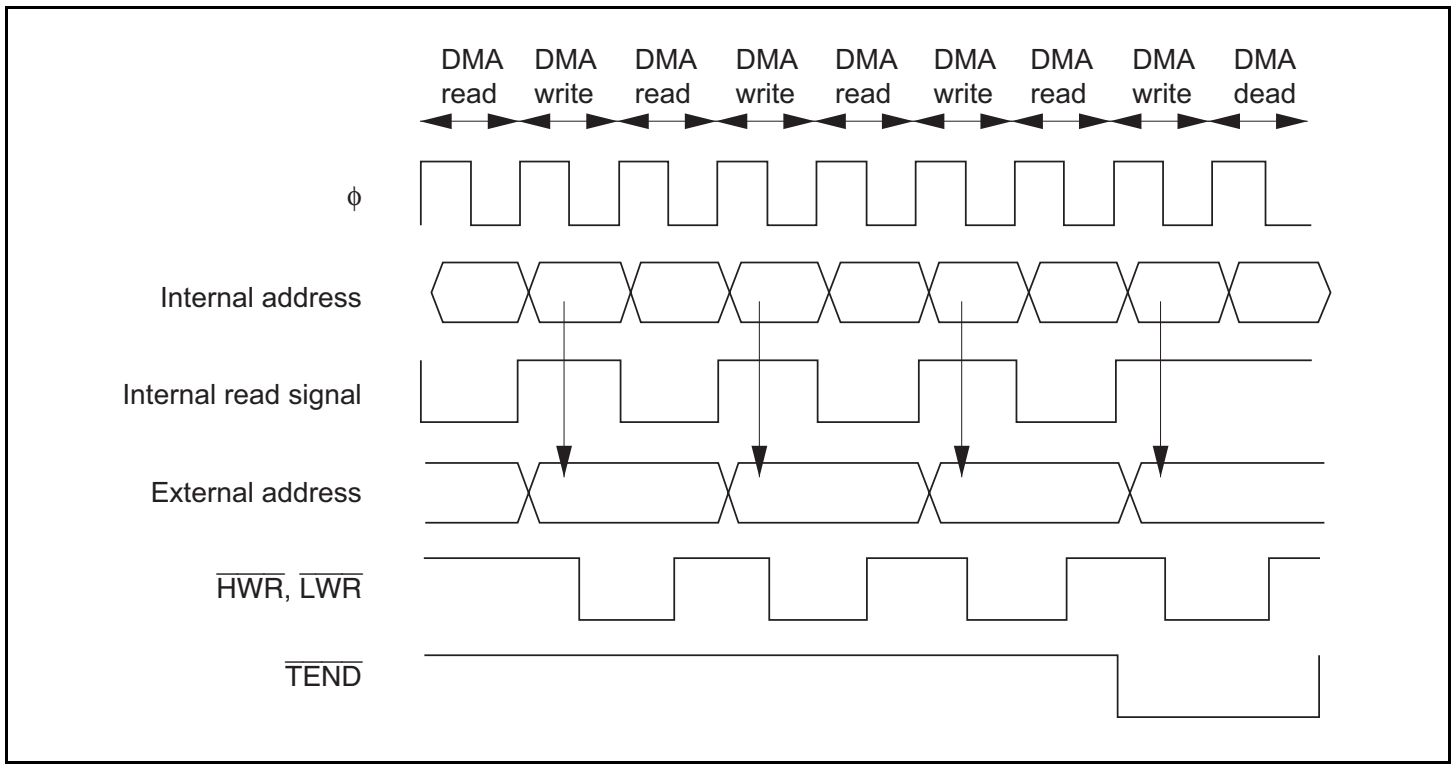


Figure 7.32 Example of Dual Address Transfer Using Write Data Buffer Function

Figure 7.33 shows an example of single address transfer using the write data buffer function. In this example, the CPU program area is in on-chip memory.

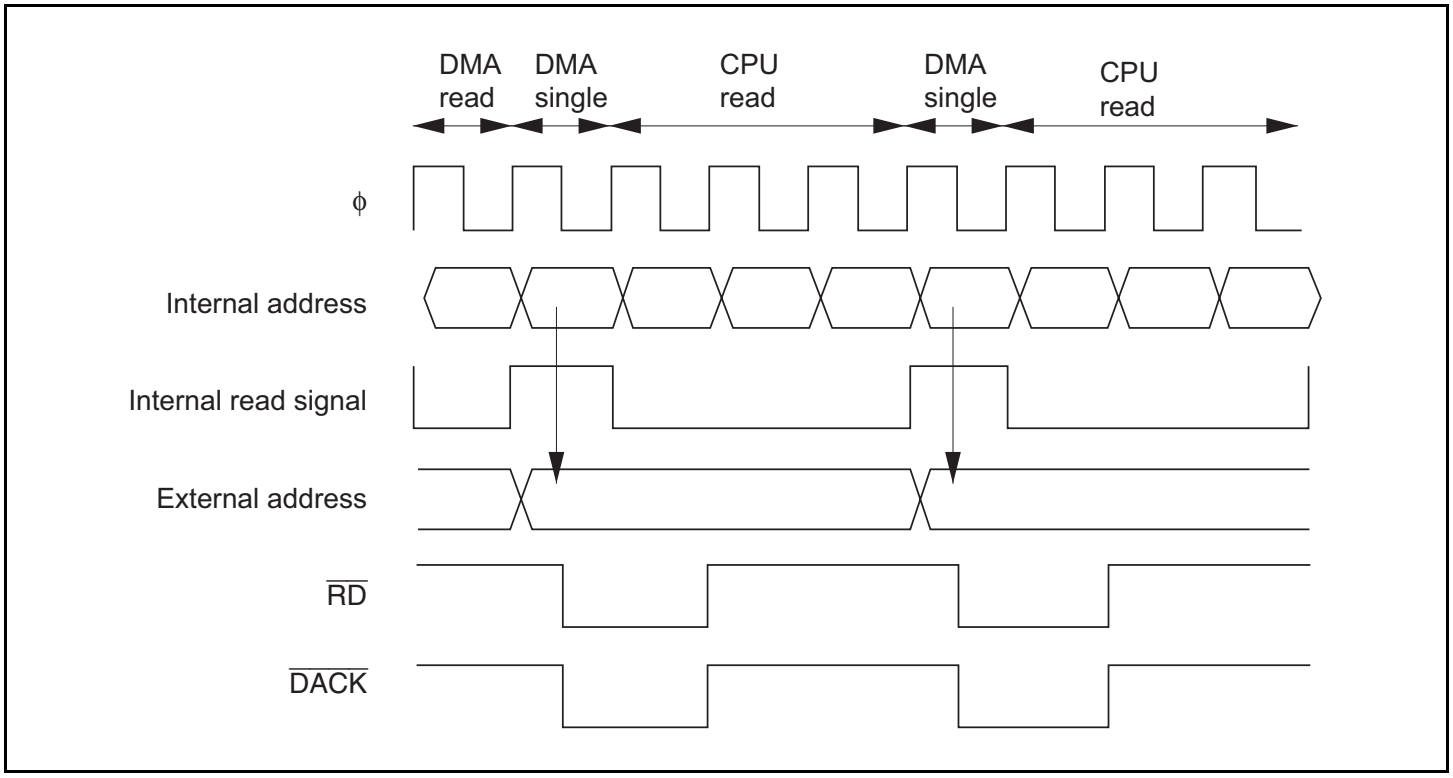


Figure 7.33 Example of Single Address Transfer Using Write Data Buffer Function

When the write data buffer function is activated, the DMAC recognizes that the bus cycle concerned has ended, and starts the next operation. Therefore, $\overline{\text{DREQ}}$ pin sampling is started one state after the start of the DMA write cycle or single address transfer.

7.5.12 Multi-Channel Operation

The DMAC channel priority order is: channel 0 > channel 1, and channel A > channel B. Table 7.11 summarizes the priority order for DMAC channels.

Table 7.11 DMAC Channel Priority Order

Short Address Mode	Full Address Mode	Priority
Channel 0A	Channel 0	High
Channel 0B		↑
Channel 1A	Channel 1	
Channel 1B		Low

If transfer requests are issued simultaneously for more than one channel, or if a transfer request for another channel is issued during a transfer, when the bus is released, the DMAC selects the highest-priority channel from among those issuing a request according to the priority order shown in table 7.11. During burst transfer, or when one block is being transferred in block transfer, the channel will not be changed until the end of the transfer. Figure 7.34 shows a transfer example in which transfer requests are issued simultaneously for channels 0A, 0B, and 1.

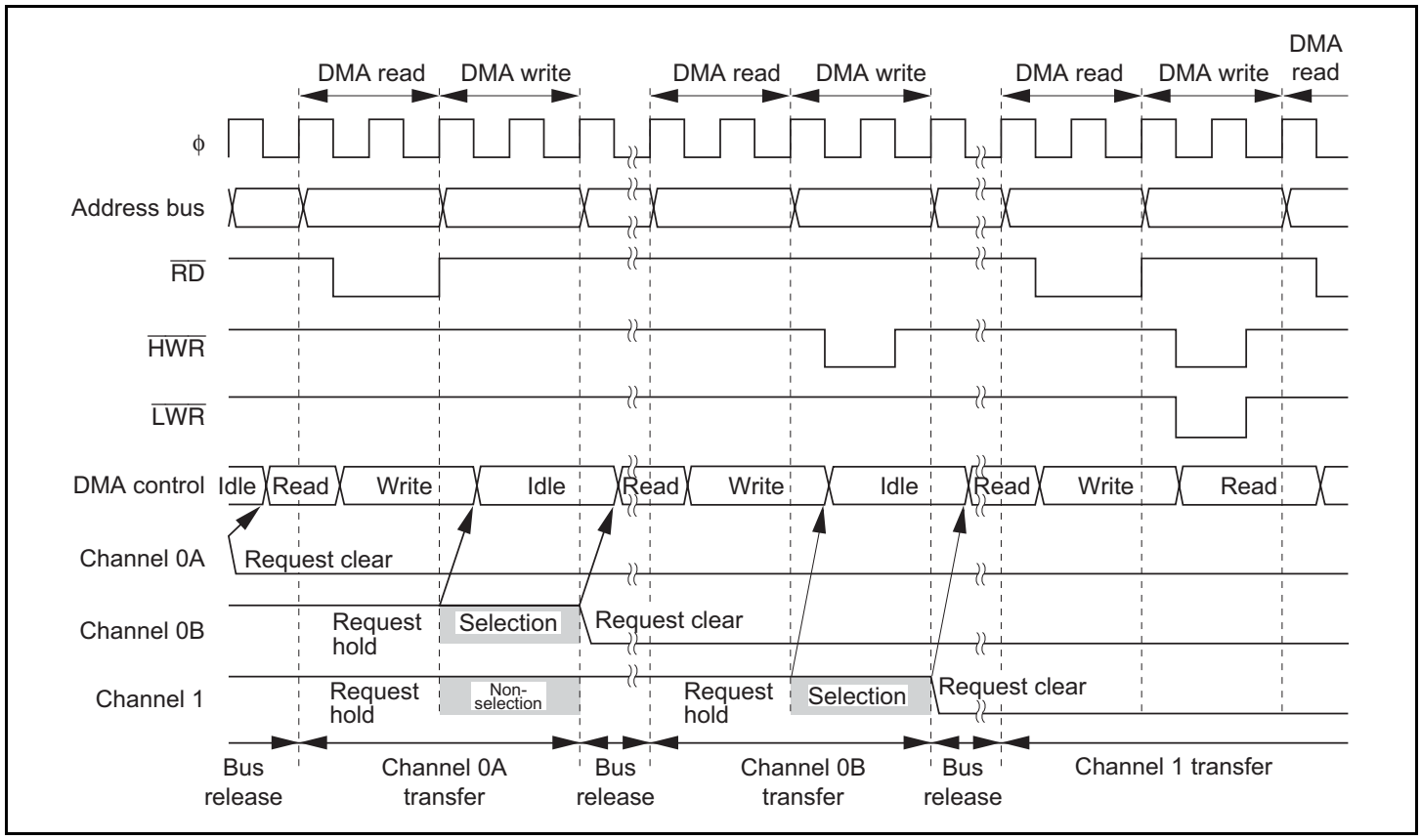


Figure 7.34 Example of Multi-Channel Transfer

7.5.13 Relation between DMAC and External Bus Requests, Refresh Cycles, and EXDMAC

When the DMAC accesses external space, contention with a refresh cycle, EXDMAC cycle, or external bus release cycle may arise. In this case, the bus controller will suspend the transfer and insert a refresh cycle, EXDMAC cycle, or external bus release cycle, in accordance with the external bus priority order, even if the DMAC is executing a burst transfer or block transfer. (An external access by the DTC or CPU, which has a lower priority than the DMAC, is not executed until the DMAC releases the external bus.)

When the DMAC transfer mode is dual address mode, the DMAC releases the external bus after an external write cycle. The external read cycle and external write cycle are inseparable, and so the bus cannot be released between these two cycles.

When the DMAC accesses internal space (on-chip memory or an internal I/O register), the DMAC cycle may be executed at the same time as a refresh cycle, EXDMAC cycle, or external bus release cycle.

7.5.14 DMAC and NMI Interrupts

When an NMI interrupt is requested, burst mode transfer in full address mode is interrupted. An NMI interrupt does not affect the operation of the DMAC in other modes.

In full address mode, transfer is enabled for a channel when both the DTE bit and DTME bit in DMABCRL are set to 1. With burst mode setting, the DTME bit is cleared when an NMI interrupt is requested.

If the DTME bit is cleared during burst mode transfer, the DMAC discontinues transfer on completion of the 1-byte or 1-word transfer in progress, then releases the bus, which passes to the CPU.

The channel on which transfer was interrupted can be restarted by setting the DTME bit to 1 again. Figure 7.35 shows the procedure for continuing transfer when it has been interrupted by an NMI interrupt on a channel designated for burst mode transfer.

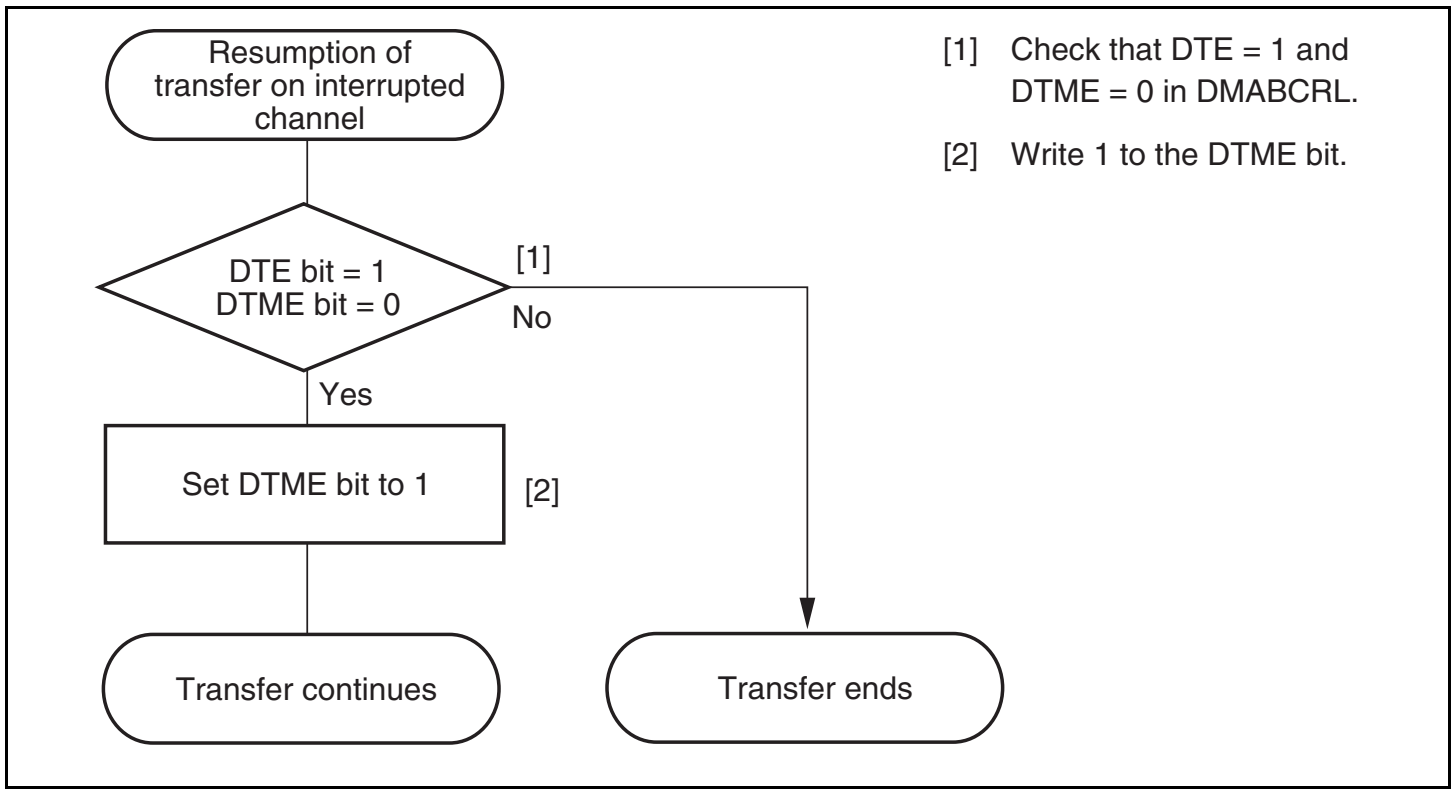


Figure 7.35 Example of Procedure for Continuing Transfer on Channel Interrupted by NMI Interrupt

7.5.15 Forced Termination of DMAC Operation

If the DTE bit in DMABCRL is cleared to 0 for the channel currently operating, the DMAC stops on completion of the 1-byte or 1-word transfer in progress. DMAC operation resumes when the DTE bit is set to 1 again. In full address mode, the same applies to the DTME bit in DMABCRL. Figure 7.36 shows the procedure for forcibly terminating DMAC operation by software.

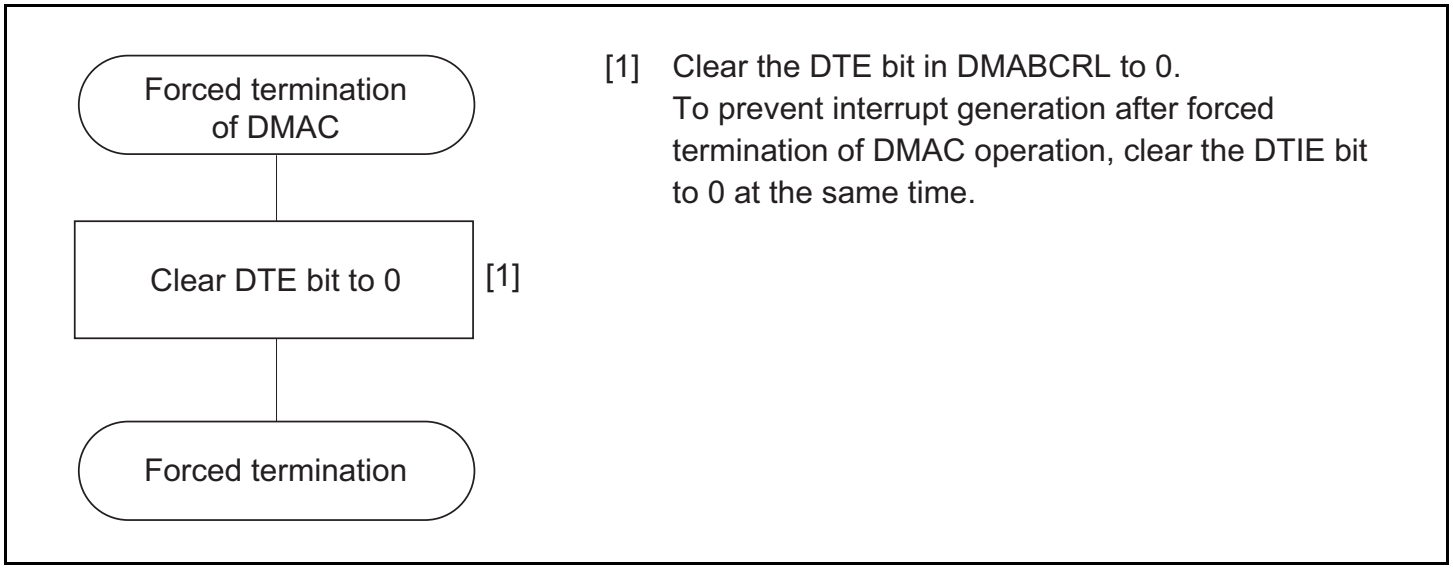


Figure 7.36 Example of Procedure for Forcibly Terminating DMAC Operation

7.5.16 Clearing Full Address Mode

Figure 7.37 shows the procedure for releasing and initializing a channel designated for full address mode. After full address mode has been cleared, the channel can be set to another transfer mode using the appropriate setting procedure.

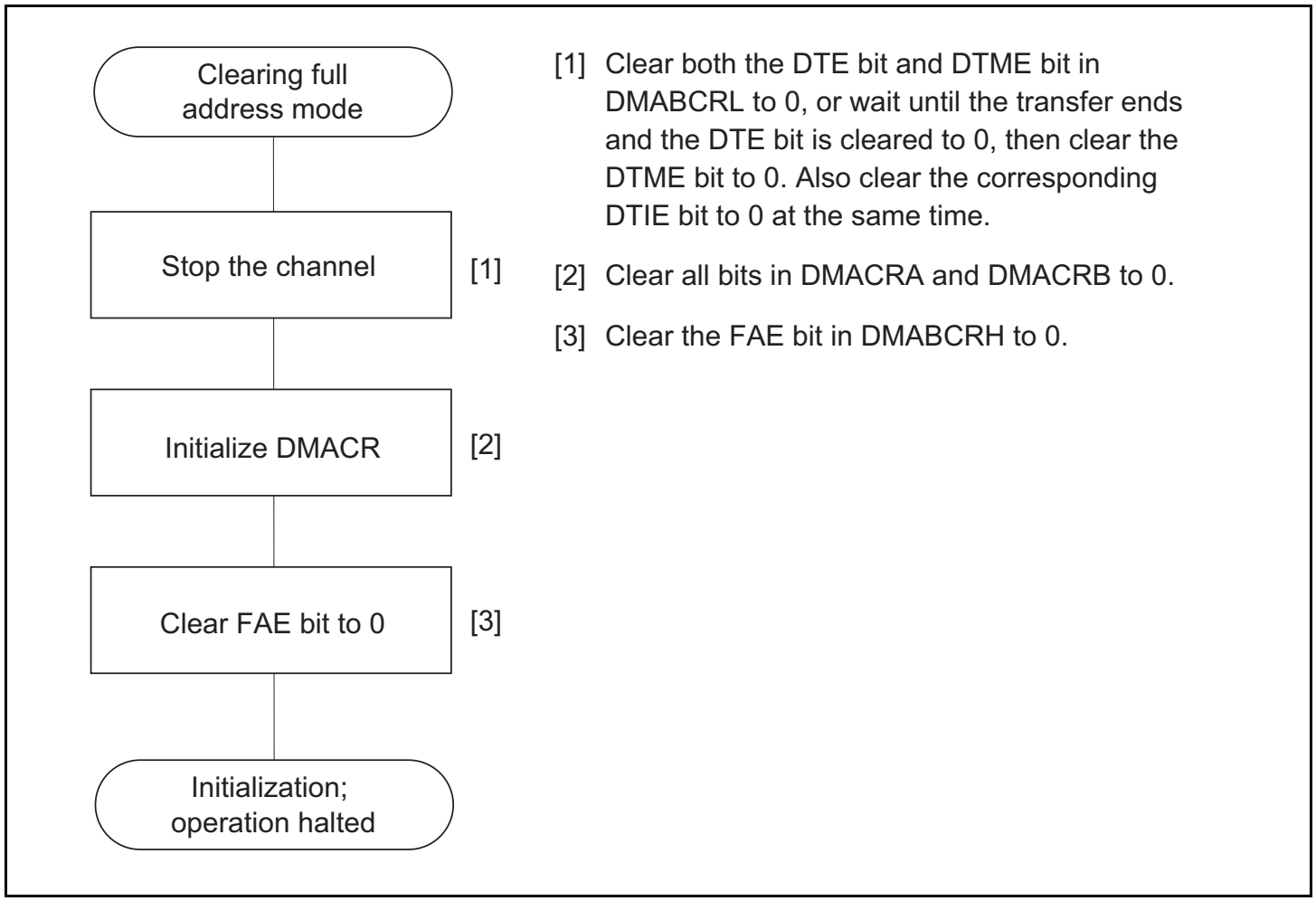


Figure 7.37 Example of Procedure for Clearing Full Address Mode

7.6 Interrupt Sources

The sources of interrupts generated by the DMAC are transfer end and transfer break. Table 7.12 shows the interrupt sources and their priority order.

Table 7.12 Interrupt Sources and Priority Order

Interrupt Name	Interrupt Source		Interrupt Priority Order
	Short Address Mode	Full Address Mode	
DMTEND0A	Interrupt due to end of transfer on channel 0A	Interrupt due to end of transfer on channel 0	High
DMTEND0B	Interrupt due to end of transfer on channel 0B	Interrupt due to break in transfer on channel 0	
DMTEND1A	Interrupt due to end of transfer on channel 1A	Interrupt due to end of transfer on channel 1	
DMTEND1B	Interrupt due to end of transfer on channel 1B	Interrupt due to break in transfer on channel 1	Low

Enabling or disabling of each interrupt source is set by means of the DTIE bit in DMABCRL for the corresponding channel in DMABCRL, and interrupts from each source are sent to the interrupt controller independently. The priority of transfer end interrupts on each channel is decided by the interrupt controller, as shown in table 7.12.

Figure 7.38 shows a block diagram of a transfer end/transfer break interrupt. An interrupt is always generated when the DTIE bit is set to 1 while the DTE bit in DMABCRL is cleared to 0.

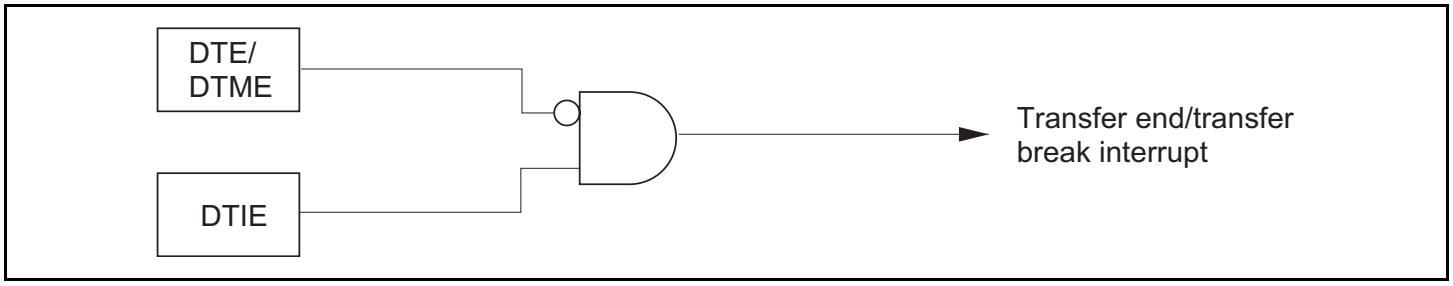


Figure 7.38 Block Diagram of Transfer End/Transfer Break Interrupt

In full address mode, a transfer break interrupt is generated when the DTME bit is cleared to 0 while the DTIE bit is set to 1. In both short address mode and full address mode, DMABCR should be set so as to prevent the occurrence of a combination that constitutes a condition for interrupt generation during setting.

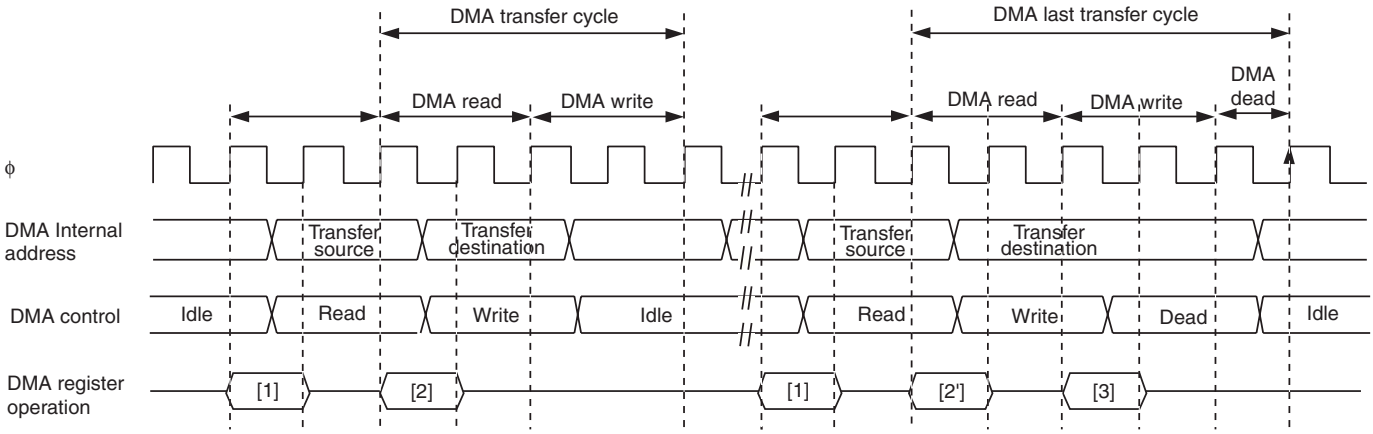
7.7 Usage Notes

(1) DMAC Register Access during Operation

Except for forced termination of the DMAC, the operating (including transfer waiting state) channel setting should not be changed. The operating channel setting should only be changed when transfer is disabled. Also, DMAC registers should not be written to in a DMA transfer.

DMAC register reads during operation (including the transfer waiting state) are described below.

- DMAC control starts one cycle before the bus cycle, with output of the internal address. Consequently, MAR is updated in the bus cycle before DMA transfer. Figure 7.39 shows an example of the update timing for DMAC registers in dual address transfer mode.

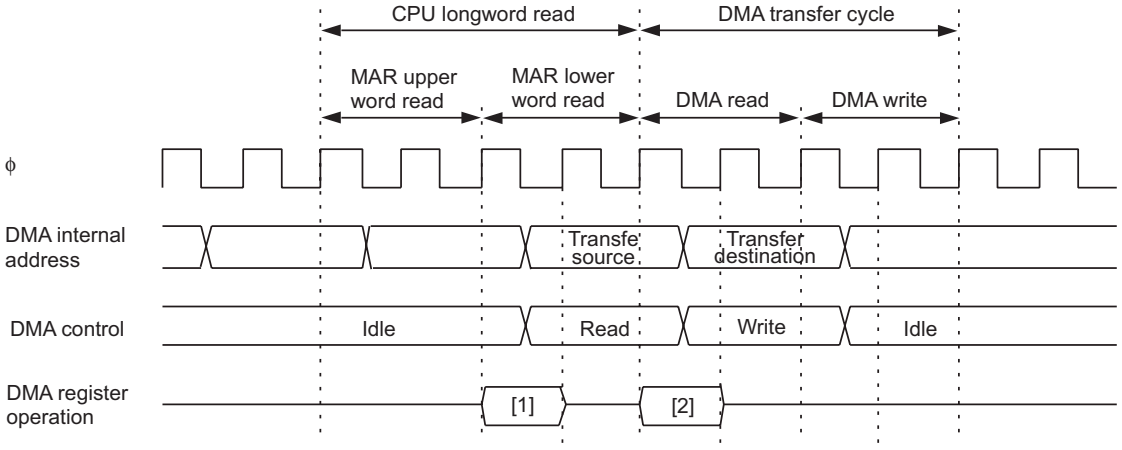


[1] Transfer source address register MAR operation (incremented/decremented/fixed)
Transfer counter ETCR operation (decremented)
Block size counter ETCR operation (decremented in block transfer mode)
[2] Transfer destination address register MAR operation (incremented/decremented/fixed)
[2']Transfer destination address register MAR operation (incremented/decremented/fixed)
Block transfer counter ETCR operation (decremented, in last transfer cycle of a block in block transfer mode)
[3] Transfer address register MAR restore operation (in block or repeat transfer mode)
Transfer counter ETCR restore (in repeat transfer mode)
Block size counter ETCR restore (in block transfer mode)

Note: In single address transfer mode, the update timing is the same as [1].
The MAR operation is post-incrementing/decrementing of the DMA internal address value.

Figure 7.39 DMAC Register Update Timing

- If a DMAC transfer cycle occurs immediately after a DMAC register read cycle, the DMAC register is read as shown in figure 7.40.



Note: The lower word of MAR is the updated value after the operation in [1].

Figure 7.40 Contention between DMAC Register Update and CPU Read

(2) Module Stop

When the MSTP13 bit in MSTPCRH is set to 1, the DMAC clock stops, and the module stop state is entered. However, 1 cannot be written to the MSTP13 bit if any of the DMAC channels is enabled. This setting should therefore be made when DMAC operation is stopped.

When the DMAC clock stops, DMAC register accesses can no longer be made. Since the following DMAC register settings are valid even in the module stop state, they should be invalidated, if necessary, before a module stop.

- Transfer end/break interrupt (DTE = 0 and DTIE = 1)
- $\overline{\text{TEND}}$ pin enable (TEE = 1)
- $\overline{\text{DACK}}$ pin enable (FAE = 0 and SAE = 1)

(3) Write Data Buffer Function

When the WDBE bit of BCR in the bus controller is set to 1, enabling the write data buffer function, dual address transfer external write cycles or single address transfers and internal accesses (on-chip memory or internal I/O registers) are executed in parallel.

- Write data buffer function and DMAC register setting

  If the setting of a register that controls external accesses is changed during execution of an external access by means of the write data buffer function, the external access may not be performed normally. Registers that control external accesses should only be manipulated when external reads, etc., are used with DMAC operation disabled, and the operation is not performed in parallel with external access.
- Write data buffer function and next DMAC operation

  The DMAC can start its next operation during external access using the write data buffer function. Consequently, the $\overline{\text{DREQ}}$ pin sampling timing, $\overline{\text{TEND}}$ output timing, etc., are different from the case in which the write data buffer function is disabled. Also, internal bus cycles maybe hidden, and not visible.

(4) $\overline{\text{TEND}}$ Output

If the last transfer cycle is for an internal address, note that even if low-level output at the $\overline{\text{TEND}}$ pin has been set, a low level may not be output at the $\overline{\text{TEND}}$ pin under the following external bus conditions since the last transfer cycle (internal bus cycle) and the external bus cycle are executed in parallel.

1. EXDMAC cycle
2. Write cycle with write buffer mode enabled
3. DMAC single address cycle for a different channel with write buffer mode enabled
4. Bus release cycle
5. CBR refresh cycle

Figure 7.41 shows an example in which a low level is not output from the $\overline{\text{TEND}}$ pin in case 2 above.

If the last transfer cycle is an external address cycle, a low level is output at the $\overline{\text{TEND}}$ pin in synchronization with the bus cycle.

However, if the last transfer cycle and a CBR refresh occur simultaneously, note that although the CBR refresh and the last transfer cycle may be executed consecutively, $\overline{\text{TEND}}$ may also go low in this case for the refresh cycle.

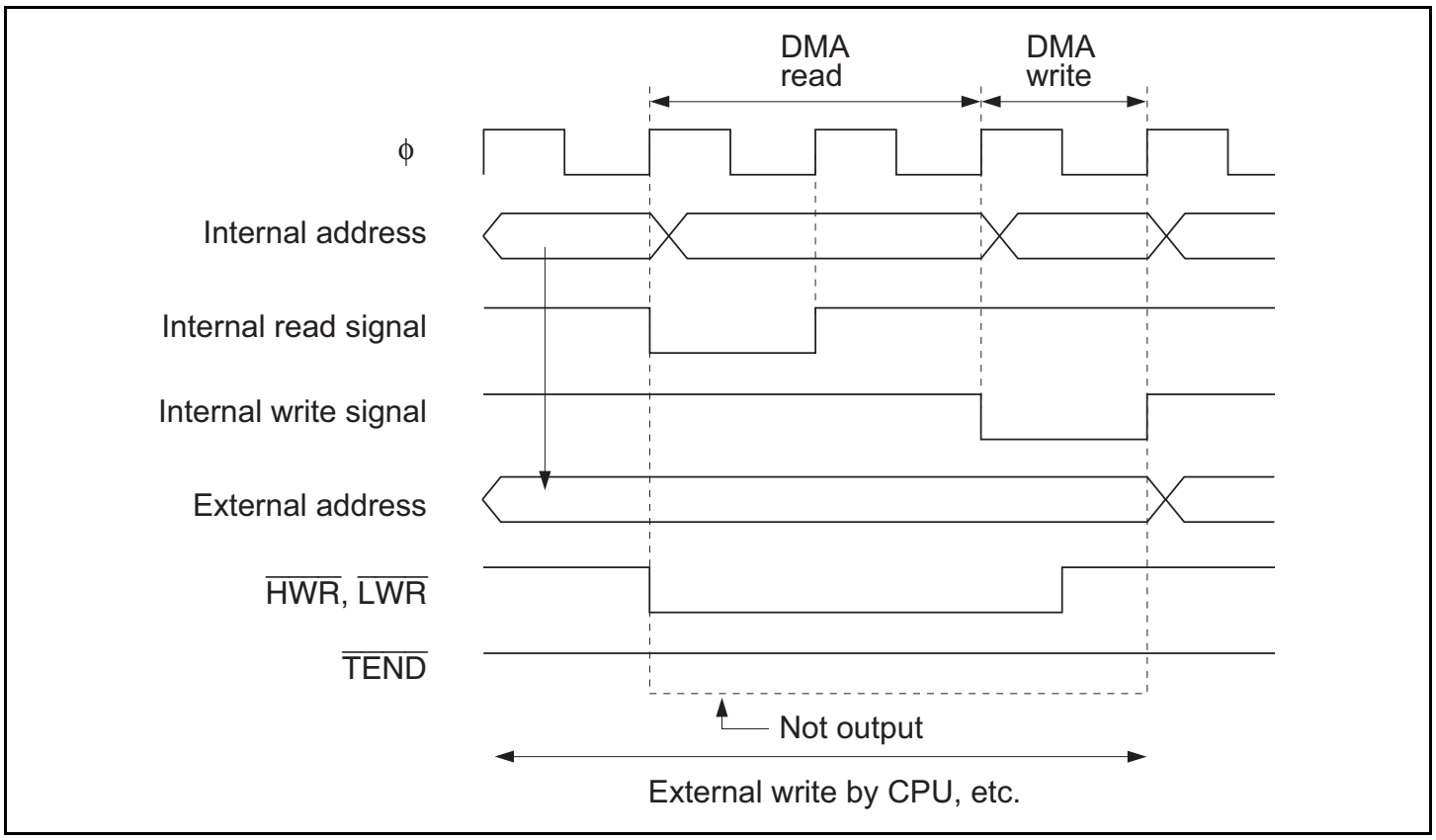


Figure 7.41 Example in which Low Level Is Not Output at $\overline{\text{TEND}}$ Pin

(5) Activation by Falling Edge on $\overline{\text{DREQ}}$ Pin

$\overline{\text{DREQ}}$ pin falling edge detection is performed in synchronization with DMAC internal operations. The operation is as follows:

[1] Activation request wait state: Waits for detection of a low level on the $\overline{\text{DREQ}}$ pin, and switches to [2].

[2] Transfer wait state: Waits for DMAC data transfer to become possible, and switches to [3].

[3] Activation request disabled state: Waits for detection of a high level on the $\overline{\text{DREQ}}$ pin, and switches to [1].

After DMAC transfer is enabled, a transition is made to [1]. Thus, initial activation after transfer is enabled is performed on detection of a low level.

(6) Activation Source Acceptance

At the start of activation source acceptance, a low level is detected in both $\overline{\text{DREQ}}$ pin falling edge sensing and low level sensing. Similarly, in the case of an internal interrupt, the interrupt request is detected. Therefore, a request is accepted from an internal interrupt or $\overline{\text{DREQ}}$ pin low level that occurs before write to DMABCRL to enable transfer.

When the DMAC is activated, take any necessary steps to prevent an internal interrupt or $\overline{\text{DREQ}}$ pin low level remaining from the end of the previous transfer, etc.

(7) Internal Interrupt after End of Transfer

When the DTE bit in DMABCRL is cleared to 0 at the end of a transfer or by a forcible termination, the selected internal interrupt request will be sent to the CPU or DTC even if the DTA bit in DMABCRH is set to 1.

Also, if internal DMAC activation has already been initiated when operation is forcibly terminated, the transfer is executed but flag clearing is not performed for the selected internal interrupt even if the DTA bit is set to 1.

An internal interrupt request following the end of transfer or a forcible termination should be handled by the CPU as necessary.

(8) Channel Re-Setting

To reactivate a number of channels when multiple channels are enabled, use exclusive handling of transfer end interrupts, and perform DMABCR control bit operations exclusively.

Note, in particular, that in cases where multiple interrupts are generated between reading and writing of DMABCR, and a DMABCR operation is performed during new interrupt handling, the DMABCR write data in the original interrupt handling routine will be incorrect, and the write may invalidate the results of the operations by the multiple interrupts. Ensure that overlapping DMABCR operations are not performed by multiple interrupts, and that there is no separation between read and write operations by the use of a bit-manipulation instruction.

Also, when the DTE and DTME bits are cleared by the DMAC or are written with 0, they must first be read while cleared to 0 before the CPU can write 1 to them.

Section 8 EXDMA Controller (EXDMAC)

This LSI has a built-in dual-channel external bus transfer DMA controller (EXDMAC). The EXDMAC can carry out high-speed data transfer, in place of the CPU, to and from external devices and external memory with a DACK (DMA transfer notification) facility.

Note: This EXDMAC is not supported by the H8S/2454 Group.

8.1 Features

- Direct specification of 16-Mbyte address space
- Selection of byte or word transfer data length
- Maximum number of transfers: 16M (16,777,215)/infinite (free-running)
- Selection of dual address mode or single address mode
- Selection of cycle steal mode or burst mode as bus mode
- Selection of normal mode or block transfer mode as transfer mode
- Two kinds of transfer requests: external request and auto-request
- An interrupt request can be sent to the CPU at the end of the specified number of transfers.
- Repeat area designation function:
- Operation in parallel with internal bus master:
- Acceptance of a transfer request and the start of transfer processing can be reported to an external device via the $\overline{\text{EDRAK}}$ pin.
- Module stop mode can be set.

Figure 8.1 shows a block diagram of the EXDMAC.

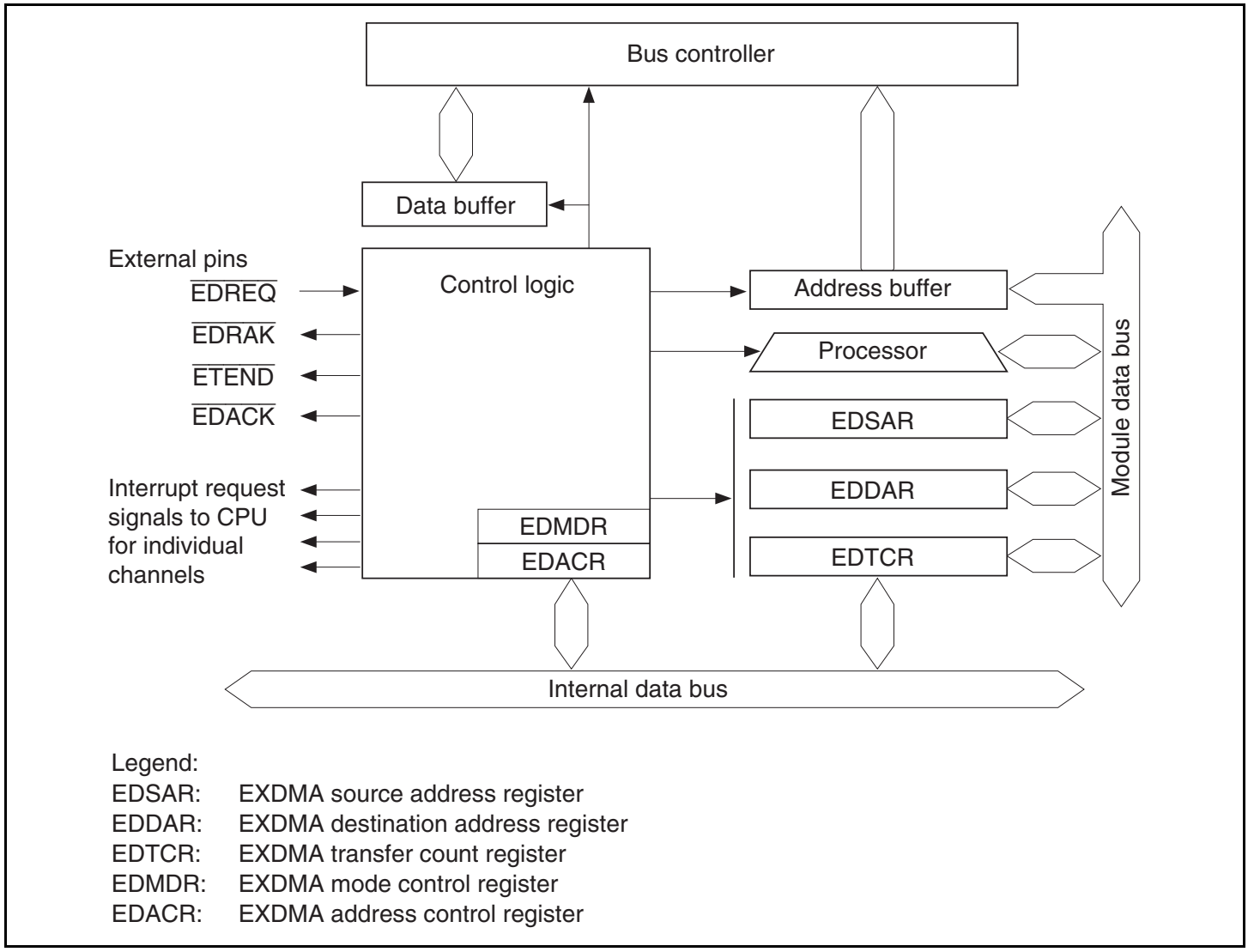


Figure 8.1 Block Diagram of EXDMAC

8.2 Input/Output Pins

Table 8.1 shows the pin configuration of the EXDMAC.

Table 8.1 Pin Configuration

Channel	Name	Abbre-viation	I/O	Function
2	EXDMA transfer request 2	$\overline{\text{EDREQ2}}$	Input	Channel 2 external request
	EXDMA transfer acknowledge 2	$\overline{\text{EDACK2}}$	Output	Channel 2 single address transfer 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 device of channel 2 external request acceptance and start of transfer processing
3	EXDMA transfer request 3	$\overline{\text{EDREQ3}}$	Input	Channel 3 external request
	EXDMA transfer acknowledge 3	$\overline{\text{EDACK3}}$	Output	Channel 3 single address transfer 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 device of channel 3 external request acceptance and start of transfer processing

8.3 Register Descriptions

The EXDMAC has the following registers.

- EXDMA source address register_2 (EDSAR_2)
- EXDMA destination address register_2 (EDDAR_2)
- EXDMA transfer count register_2 (EDTCR_2)
- EXDMA mode control register_2 (EDMDR_2)
- EXDMA address control register_2 (EDACR_2)
- EXDMA source address register_3 (EDSAR_3)
- EXDMA destination address register_3 (EDDAR_3)
- EXDMA transfer count register_3 (EDTCR_3)
- EXDMA mode control register_3 (EDMDR_3)
- EXDMA address control register_3 (EDACR_3)

8.3.1 EXDMA Source Address Register (EDSAR)

EDSAR is a 32-bit readable/writable register that specifies the transfer source address. An address update function is provided that updates the register contents to the next transfer source address each time transfer processing is performed. In single address mode, the EDSAR value is ignored when a device with $\overline{\text{DACK}}$ is specified as the transfer source.

The upper 8 bits of EDSAR are reserved; they are always read as 0 and cannot be modified. Only 0 should be written to these bits.

EDSAR can be read at all times by the CPU. When reading EDSAR for a channel on which EXDMA transfer processing is in progress, a longword-size read must be executed. Do not write to EDSAR for a channel on which EXDMA transfer is in progress. The initial values of EDSAR are undefined.

8.3.2 EXDMA Destination Address Register (EDDAR)

EDDAR is a 32-bit readable/writable register that specifies the transfer destination address. An address update function is provided that updates the register contents to the next transfer destination address each time transfer processing is performed. In single address mode, the EDDAR value is ignored when a device with $\overline{\text{DACK}}$ is specified as the transfer destination.

The upper 8 bits of EDDAR are reserved; they are always read as 0 and cannot be modified. Only 0 should be written to these bits.

EDDAR can be read at all times by the CPU. When reading EDDAR for a channel on which EXDMA transfer processing is in progress, a longword-size read must be executed. Do not write to EDDAR for a channel on which EXDMA transfer is in progress. The initial values of EDDAR are undefined.

8.3.3 EXDMA Transfer Count Register (EDTCR)

EDTCR specifies the number of transfers. The function differs according to the transfer mode. Do not write to EDTCR for a channel on which EXDMA transfer is in progress.

(1) Normal Transfer Mode

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
23 to 0		All 0	R/W	24-Bit Transfer Counter These bits specify the number of transfers. Setting H'000001 specifies one transfer. Setting H'000000 means no specification for the number of transfers, and the transfer counter function is halted. In this case, there is no transfer end interrupt by the transfer counter. Setting H'FFFFFF specifies the maximum number of transfers, that is 16,777,215. During EXDMA transfer, this counter shows the remaining number of transfers. This counter can be read at all times. When reading EDTCR for a channel on which EXDMA transfer processing is in progress, a longword-size read must be executed.

(2) Block Transfer Mode

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
23 to 16		Undefined	R/W	Block Size These bits specify the block size (number of bytes or number of words) for block transfer. Setting H'01 specifies one as the block, while setting H'00 specifies the maximum block size, that is 256. The register value always indicates the specified block size.
15 to 0		Undefined	R/W	16-Bit Transfer Counter These bits specify the number of block transfers. Setting H'0001 specifies one block transfer. Setting H'0000 means no specification for the number of transfers, and the transfer counter function is halted. In this case, there is no transfer end interrupt by the transfer counter. Setting H'FFFF specifies the maximum number of block transfers, that is 65,535. During EXDMA transfer, this counter shows the remaining number of block transfers.

8.3.4 EXDMA Mode Control Register (EDMDR)

EDMDR controls EXDMAC operations.

Bit	Bit Name	Initial Value	R/W	Description
15	EDA	0	R/(W)	EXDMA Active Enables or disables data transfer on the corresponding channel. When this bit is set to 1, this indicates that an EXDMA operation is in progress. When auto request mode is specified (by bits MDS1 and MDS0), transfer processing begins when this bit is set to 1. With external requests, transfer processing begins when a transfer request is issued after this bit has been set to 1. When this bit is cleared to 0 during an EXDMA operation, transfer is halted. If this bit is cleared to 0 during an EXDMA operation in block transfer mode, transfer processing is continued for the currently executing one-block transfer, and the bit is cleared on completion of the currently executing one-block transfer. If an external source that ends (aborts) transfer occurs, this bit is automatically cleared to 0 and transfer is terminated. Do not change the operating mode, transfer method, or other parameters while this bit is set to 1. 0: Data transfer disabled on corresponding channel [Clearing conditions] • When the specified number of transfers end • When operation is halted by a repeat area overflow interrupt • When 0 is written to EDA while EDA = 1 (In block transfer mode, write is effective after end of one-block transfer) • Reset, NMI interrupt, hardware standby mode 1: Data transfer enabled on corresponding channel Note: The value written in the EDA bit may not be effective immediately.

Bit	Bit Name	Initial Value	R/W	Description
14	BEF	0	R/(W)*	Block Transfer Error Flag Flag that indicates the occurrence of an error during block transfer. If an NMI interrupt is generated during block transfer, the EXDMAC immediately terminates the EXDMA operation and sets this bit to 1. The address registers indicate the next transfer addresses, but the data for which transfer has been performed within the block size is lost. 0: No block transfer error [Clearing condition] Writing 0 to BEF after reading BEF = 1 1: Block transfer error [Setting condition] NMI interrupt during block transfer
13	EDRAKE	0	R/W	$\overline{\text{EDRAK}}$ Pin Output Enable Enables output from the $\overline{\text{EDREQ}}$ acknowledge/transfer processing start ($\overline{\text{EDRAK}}$) pin. 0: $\overline{\text{EDRAK}}$ pin output disabled 1: $\overline{\text{EDRAK}}$ pin output enabled
12	ETENDE	0	R/W	$\overline{\text{ETEND}}$ Pin Output Enable Enables output from the EXDMA transfer end ($\overline{\text{ETEND}}$) pin. 0: $\overline{\text{ETEND}}$ pin output disabled 1: $\overline{\text{ETEND}}$ pin output enabled
11	EDREQS	0	R/W	$\overline{\text{EDREQ}}$ Select Specifies low level sensing or falling edge sensing as the sampling method for the $\overline{\text{EDREQ}}$ pin used in external request mode. 0: Low level sensing (Low level sensing is used for the first transfer after transfer is enabled.) 1: Falling edge sensing

Bit	Bit Name	Initial Value	R/W	Description
10	AMS	0	R/W	Address Mode Select Selects single address mode or dual address mode. When single address mode is selected, the $\overline{\text{EDACK}}$ pin is valid. 0: Dual address mode 1: Single address mode
9	MDS1	0	R/W	Mode Select 1 and 0 These bits specify the activation source, bus mode, and transfer mode. 00: Auto request, cycle steal mode, normal transfer mode 01: Auto request, burst mode, normal transfer mode 10: External request, cycle steal mode, normal transfer mode 11: External request, cycle steal mode, block transfer mode
8	MDS0	0	R/W	
7	EDIE	0	R/W	EXDMA Interrupt Enable Enables or disables interrupt requests. When this bit is set to 1, an interrupt is requested when the IRF bit is set to 1. The interrupt request is cleared by clearing this bit or the IRF bit to 0. 0: Interrupt request is not generated 1: Interrupt request is generated

Bit	Bit Name	Initial Value	R/W	Description
6	IRF	0	R/(W)*	Interrupt Request Flag Flag indicating that an interrupt request has occurred and transfer has ended. 0: No interrupt request [Clearing conditions] • Writing 1 to the EDA bit • Writing 0 to IRF after reading IRF = 1 1: Interrupt request occurrence [Setting conditions] • Transfer end interrupt request generated by transfer counter • Source address repeat area overflow interrupt request • Destination address repeat area overflow interrupt request
5	TCEIE	0	R/W	Transfer Counter End Interrupt Enable Enables or disables transfer end interrupt requests by the transfer counter. When transfer ends according to the transfer counter while this bit is set to 1, the IRF bit is set to 1, indicating that an interrupt request has occurred. 0: Transfer end interrupt requests by transfer counter are disabled 1: Transfer end interrupt requests by transfer counter are enabled
4	SDIR	0	R/W	Single Address Direction Specifies the data transfer direction in single address mode. In dual address mode, the specification by this bit is ignored. 0: Transfer direction: EDSAR → external device with $\overline{\text{DACK}}$ 1: Transfer direction: External device with $\overline{\text{DACK}}$→ EDDAR

Bit	Bit Name	Initial Value	R/W	Description
3	DTSIZE	0	R/W	Data Transmit Size Specifies the size of data to be transferred. 0: Byte-size 1: Word-size
2	BGUP	0	R/W	Bus Give-Up When this bit is set to 1, the bus can be transferred to an internal bus master in burst mode or block transfer mode. This setting is ignored in normal mode and cycle steal mode. 0: Bus is not released 1: Bus is transferred if requested by an internal bus master
1	—	0	R/W	Reserved
0	—	0	R/W	These bits are always read as 0. The initial values should not be modified.

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

8.3.5 EXDMA Address Control Register (EDACR)

EDACR specifies address register incrementing/decrementing and use of the repeat area function.

Bit	Bit Name	Initial Value	R/W	Description
15	SAT1	0	R/W	Source Address Update Mode
14	SAT0	0	R/W	These bits specify incrementing/decrementing of the transfer source address (EDSAR). When an external device with DACK is designated as the transfer source in single address mode, the specification by these bits is ignored. 0×: Fixed 10: Incremented (+1 in byte transfer, +2 in word transfer) 11: Decremented (–1 in byte transfer, –2 in word transfer)
13	SARIE	0	R/W	Source Address Repeat Interrupt Enable When this bit is set to 1, in the event of source address repeat area overflow, the IRF bit is set to 1 and the EDA bit cleared to 0 in EDMDR, and transfer is terminated. If the EDIE bit in EDMDR is 1 when the IRF bit in EDMDR is set to 1, an interrupt request is sent to the CPU. When used together with block transfer mode, a source address repeat interrupt is requested at the end of a block-size transfer. If the EDA bit is set to 1 in EDMDR for the channel on which transfer is terminated by a source address repeat interrupt, transfer can be resumed from the state in which it ended. If a source address repeat area has not been designated, this bit is ignored. 0: Source address repeat interrupt is not requested 1: When source address repeat area overflow occurs, the IRF bit in EDMDR is set to 1 and an interrupt is requested

Bit	Bit Name	Initial Value	R/W	Description
12	SARA4	0	R/W	Source Address Repeat Area
11	SARA3	0	R/W	These bits specify the source address (EDSAR) repeat area. The repeat area function updates the specified lower address bits, leaving the remaining upper address bits always the same. A repeat area size of 2 bytes to 8 Mbytes can be specified. The setting interval is a power-of-two number of bytes. When repeat area overflow results from incrementing or decrementing an address, the lower address is the start address of the repeat area in the case of address incrementing, or the last address of the repeat area in the case of address decrementing. If the SARIE bit is set to 1, an interrupt can be requested when repeat area overflow occurs. 00000: Not designated as repeat area 00001: Lower 1 bit (2-byte area) designated as repeat area 00010: Lower 2 bits (4-byte area) designated as repeat area 00011: Lower 3 bits (8-byte area) designated as repeat area 00100: Lower 4 bits (16-byte area) designated as repeat area : : 10011: Lower 19 bits (512-Kbyte area) designated as repeat area 10100: Lower 20 bits (1-Mbyte area) designated as repeat area 10101: Lower 21 bits (2-Mbyte area) designated as repeat area 10110: Lower 22 bits (4-Mbyte area) designated as repeat area 10111: Lower 23 bits (8-Mbyte area) designated as repeat area 11×××: Setting prohibited
10	SARA2	0	R/W	
9	SARA1	0	R/W	
8	SARA0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	DAT1	0	R/W	Destination Address Update Mode
6	DAT0	0	R/W	These bits specify incrementing/decrementing of the transfer destination address (EDDAR). When an external device with DACK is designated as the transfer destination in single address mode, the specification by these bits is ignored. 0×: Fixed 10: Incremented (+1 in byte transfer, +2 in word transfer) 11: Decremented (–1 in byte transfer, –2 in word transfer)
5	DARIE	0	R/W	Destination Address Repeat Interrupt Enable When this bit is set to 1, in the event of destination address repeat area overflow the IRF bit is set to 1 and the EDA bit cleared to 0 in EDMDR, and transfer is terminated. If the EDIE bit in EDMDR is 1 when the IRF bit in EDMDR is set to 1, an interrupt request is sent to the CPU. When used together with block transfer mode, a destination address repeat interrupt is requested at the end of a block-size transfer. If the EDA bit is set to 1 in EDMDR for the channel on which transfer is terminated by a destination address repeat interrupt, transfer can be resumed from the state in which it ended. If a destination address repeat area has not been designated, this bit is ignored. 0: Destination address repeat interrupt is not requested 1: When destination address repeat area overflow occurs, the IRF bit in EDMDR is set to 1 and an interrupt is requested

Bit	Bit Name	Initial Value	R/W	Description
4	DARA4	0	R/W	Destination Address Repeat Area These bits specify the destination address (EDDAR) repeat area. The repeat area function updates the specified lower address bits, leaving the remaining upper address bits always the same. A repeat area size of 2 bytes to 8 Mbytes can be specified. The setting interval is a power-of-two number of bytes. When repeat area overflow results from incrementing or decrementing an address, the lower address is the start address of the repeat area in the case of address incrementing, or the last address of the repeat area in the case of address decrementing. If the DARIE bit is set to 1, an interrupt can be requested when repeat area overflow occurs. 00000: Not designated as repeat area 00001: Lower 1 bit (2-byte area) designated as repeat area 00010: Lower 2 bits (4-byte area) designated as repeat area 00011: Lower 3 bits (8-byte area) designated as repeat area 00100: Lower 4 bits (16-byte area) designated as repeat area : : 10011: Lower 19 bits (512-Kbyte area) designated as repeat area 10100: Lower 20 bits (1-Mbyte area) designated as repeat area 10101: Lower 21 bits (2-Mbyte area) designated as repeat area 10110: Lower 22 bits (4-Mbyte area) designated as repeat area 10111: Lower 23 bits (8-Mbyte area) designated as repeat area 11×××: Setting prohibited
3	DARA3	0	R/W	
2	DARA2	0	R/W	
1	DARA1	0	R/W	
0	DARA0	0	R/W	

Legend:
×: Don't care

8.4 Operation

8.4.1 Transfer Modes

The transfer modes of the EXDMAC are summarized in table 8.2.

Table 8.2 EXDMAC Transfer Modes

Transfer Mode			Transfer Origin	Number of Transfers	Address Registers	
					Source	Destination
Dual address mode	Normal transfer mode	Auto request mode • Burst/cycle steal mode	Auto request	1 to 16,777,215 or no specification	EDSAR	EDDAR
		External request mode • Cycle steal mode	External request			
	Block transfer mode	External request mode • Burst transfer of specified block size for a single transfer request • Block size: 1 to 256 bytes or words	External request	1 to 65,535 or no specification		
Single address mode	• Direct data transfer to/from external device using $\overline{\text{EDACK}}$ 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/ $\overline{\text{EDACK}}$	$\overline{\text{EDACK}}$/ EDDAR

The transfer mode can be set independently for each channel.

In normal transfer mode, a one-byte or one-word transfer is executed in response to one transfer request. With auto requests, burst or cycle steal transfer mode can be set. In burst transfer mode, continuous, high-speed transfer can be performed until the specified number of transfers have been executed or the transfer enable bit is cleared to 0.

In block transfer mode, a transfer of the specified block size is executed in response to one transfer request. The block size can be from 1 to 256 bytes or words. Within a block, transfer can be performed at the same high speed as in block transfer mode.

When the “no specification” setting (EDTCR = H'000000) is made for the number of transfers, the transfer counter is halted and there is no limit on the number of transfers, allowing transfer to be performed endlessly.

Incrementing or decrementing the memory address by 1 or 2, or leaving the address unchanged, can be specified independently for each address register.

In all transfer modes, it is possible to set a repeat area comprising a power-of-two number of bytes.

8.4.2 Address Modes

(1) Dual Address Mode

In dual address mode, both the transfer source and transfer destination are specified by registers in the EXDMAC, and one transfer is executed in two bus cycles.

The transfer source address is set in the source address register (EDSAR), and the transfer destination address is set in the transfer destination address register (EDDAR).

In a transfer operation, the value in external memory specified by the transfer source address is read in the first bus cycle, and is written to the external memory specified by the transfer destination address in the next bus cycle.

These consecutive read and write cycles are indivisible: another bus cycle (external access by an internal bus master, refresh cycle, or external bus release cycle) does not occur between these two cycles.

$\overline{\text{ETEND}}$ 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. The $\overline{\text{EDACK}}$ signal is not output.

Figure 8.2 shows an example of the timing in dual address mode.

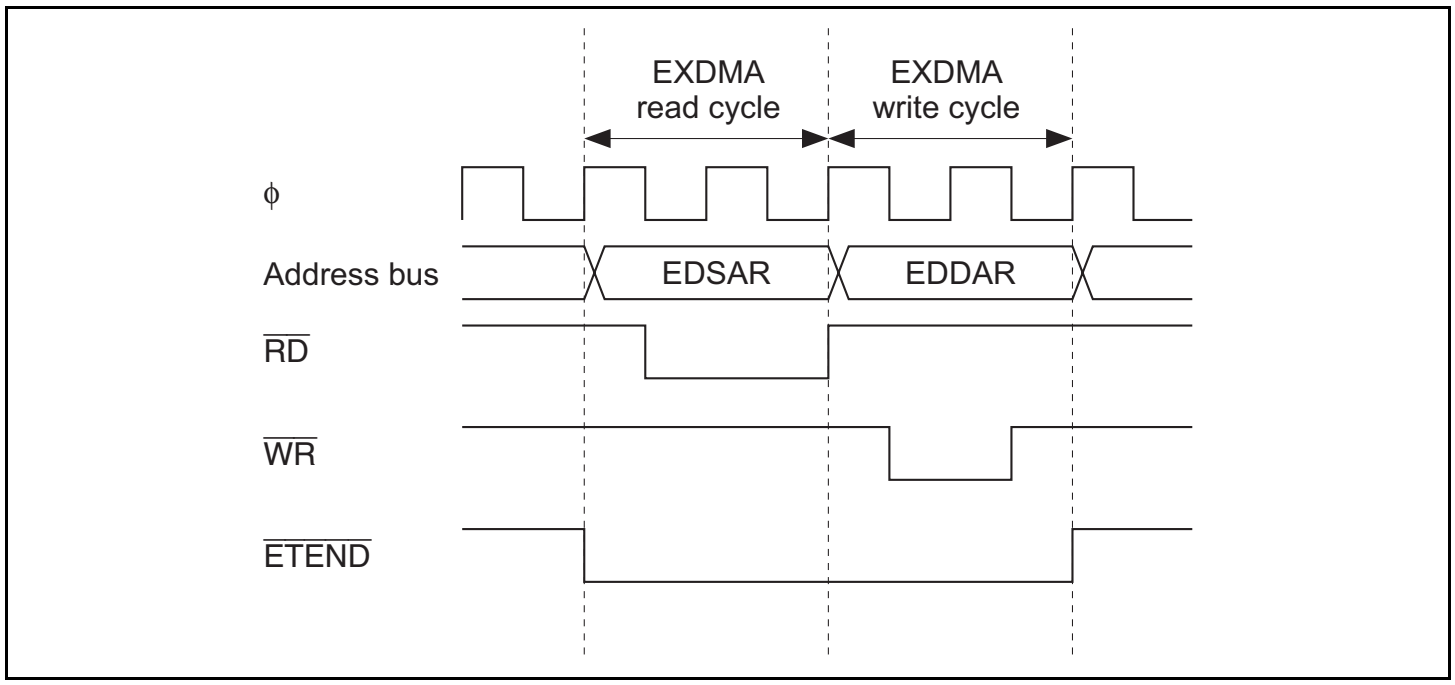


Figure 8.2 Example of Timing in Dual Address Mode

(2) Single Address Mode

In single address mode, the $\overline{\text{EDACK}}$ signal is used instead of the source or destination address register to transfer data directly between an external device and external memory. In this mode, the EXDMAC accesses the transfer source or transfer destination external device by outputting the external I/O strobe signal ($\overline{\text{EDACK}}$), and at the same time accesses the other external device in the transfer by outputting an address. In this way, DMA transfer can be executed in one bus cycle. In the example of transfer between external memory and an external device with DACK shown in figure 8.3, data is output to the data bus by the external device and written to external memory in the same bus cycle.

The transfer direction, that is whether the external device with DACK is the transfer source or transfer destination, can be specified with the SDIR bit in EDMDR. Transfer is performed from the external memory (EDSAR) to the external device with DACK when SDIR = 0, and from the external device with DACK to the external memory (EDDAR) when SDIR = 1.

The setting in the source or destination address register not used in the transfer is ignored.

The $\overline{\text{EDACK}}$ pin becomes valid automatically when single address mode is selected. The $\overline{\text{EDACK}}$ pin 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.

Figure 8.3 shows the data flow in single address mode, and figure 8.4 shows an example of the timing.

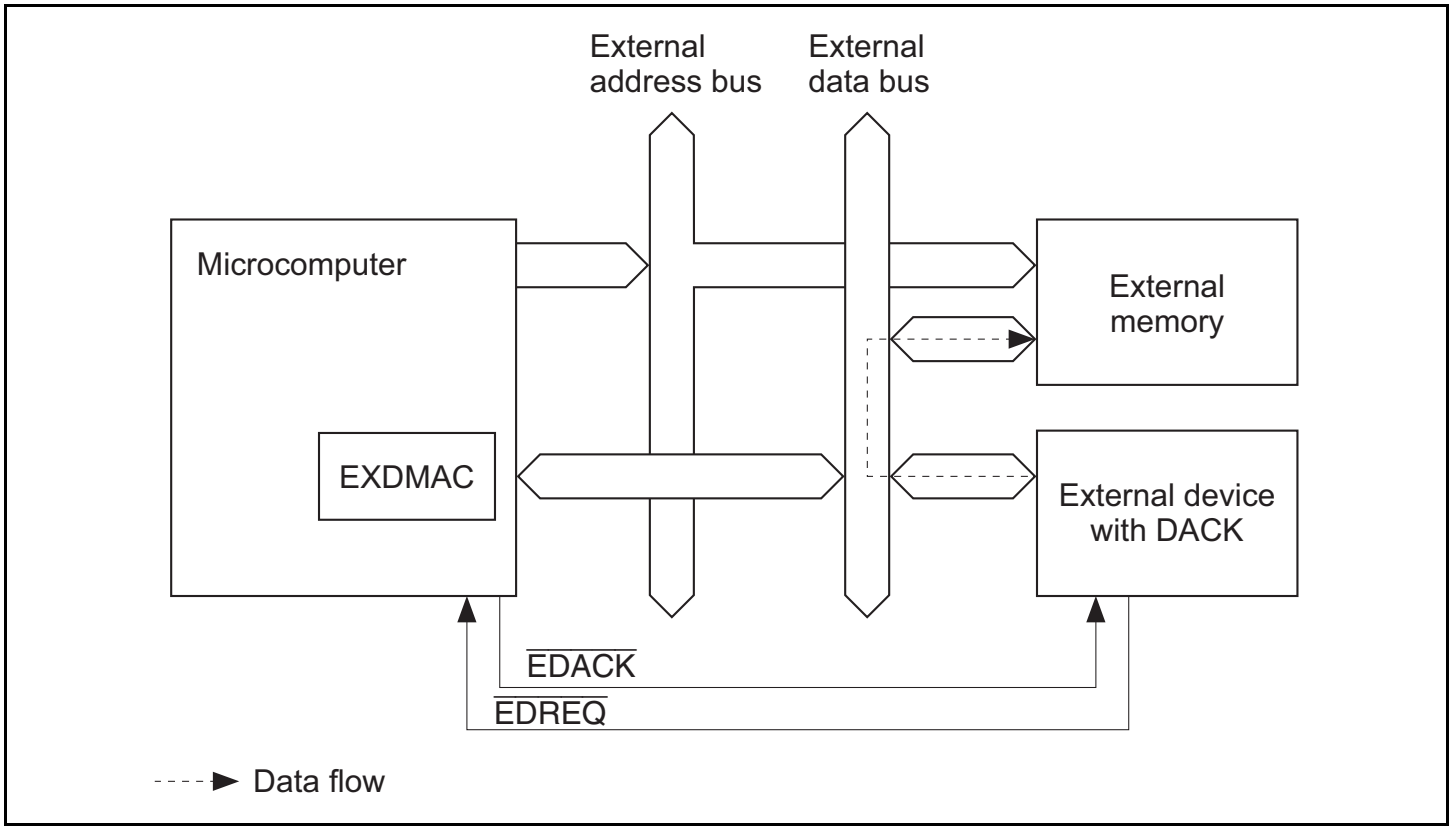


Figure 8.3 Data Flow in Single Address Mode

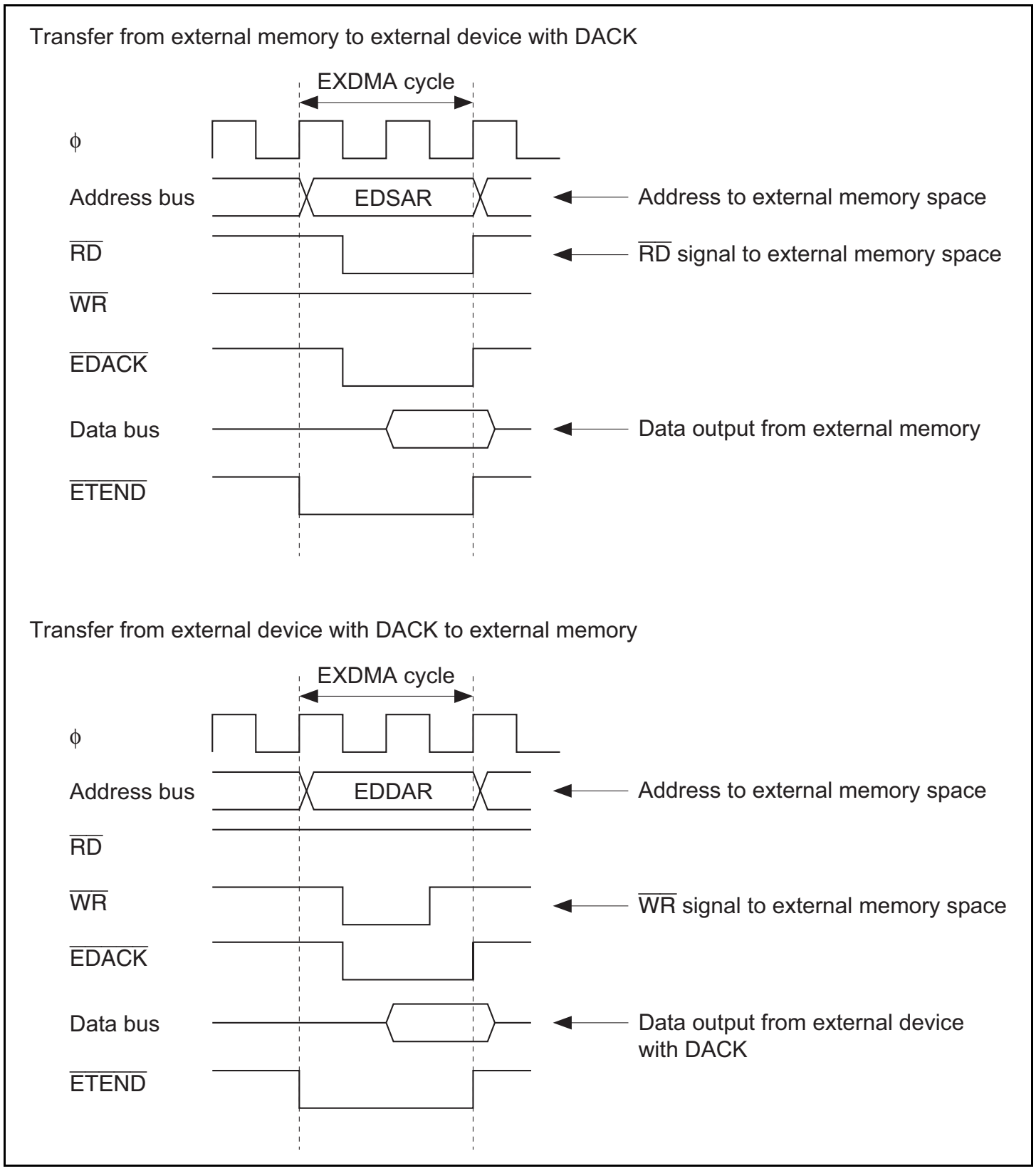


Figure 8.4 Example of Timing in Single Address Mode

8.4.3 DMA Transfer Requests

(1) Auto Request Mode

In auto request mode, transfer request signals are automatically generated within the EXDMAC in cases where a transfer request signal is not issued from outside, such as in transfer between two memories, or between a peripheral module that is not capable of generating transfer requests and memory. In auto request mode, transfer is started when the EDA bit is set to 1 in EDMDR.

In auto request mode, either cycle steal mode or burst mode can be selected as the bus mode. Block transfer mode cannot be used.

(2) External Request Mode

In external request mode, transfer is started by a transfer request signal ($\overline{\text{EDREQ}}$) from a device external to this LSI. DMA transfer is started when $\overline{\text{EDREQ}}$ is input while DMA transfer is enabled (EDA = 1).

The transfer request source need not be the data transfer source or data transfer destination.

The transfer request signal is accepted via the $\overline{\text{EDREQ}}$ pin. Either falling edge sensing or low level sensing can be selected for the $\overline{\text{EDREQ}}$ pin by means of the EDREQS bit in EDMDR (low level sensing when EDREQS = 0, falling edge sensing when EDREQS = 1).

Setting the EDRAKE bit to 1 in EDMDR enables a signal confirming transfer request acceptance to be output from the $\overline{\text{EDRAK}}$ pin. The $\overline{\text{EDRAK}}$ signal is output when acceptance and transfer processing has been started in response to a single external request. The $\overline{\text{EDRAK}}$ signal enables the external device to determine the timing of $\overline{\text{EDREQ}}$ signal negation, and makes it possible to provide handshaking between the transfer request source and the EXDMAC.

In external request mode, block transfer mode can be used instead of burst mode. Block transfer mode allows continuous execution (burst operation) of the specified number of transfers (the block size) in response to a single transfer request. In block transfer mode, the $\overline{\text{EDRAK}}$ signal is output only once for a one-block transfer, since the transfer request via the $\overline{\text{EDREQ}}$ pin is for a block unit.

8.4.4 Bus Modes

There are two bus modes: cycle steal mode and burst mode. When the activation source is an auto request, either cycle steal mode or burst mode can be selected. When the activation source is an external request, cycle steal mode is used.

(1) Cycle Steal Mode

In cycle steal mode, the EXDMAC releases the bus at the end of each transfer of a transfer unit (byte, word, or block). If there is a subsequent transfer request, the EXDMAC takes back the bus, performs another transfer-unit transfer, and then releases the bus again. This procedure is repeated until the transfer end condition is satisfied.

If a transfer request occurs in another channel during DMA transfer, the bus is temporarily released, then transfer is performed on the channel for which the transfer request was issued. If there is no external space bus request from another bus master, a one-cycle bus release interval is inserted. For details on the operation when there are requests for a number of channels, see section 8.4.8, Channel Priority Order.

Figure 8.5 shows an example of the timing in cycle steal mode.

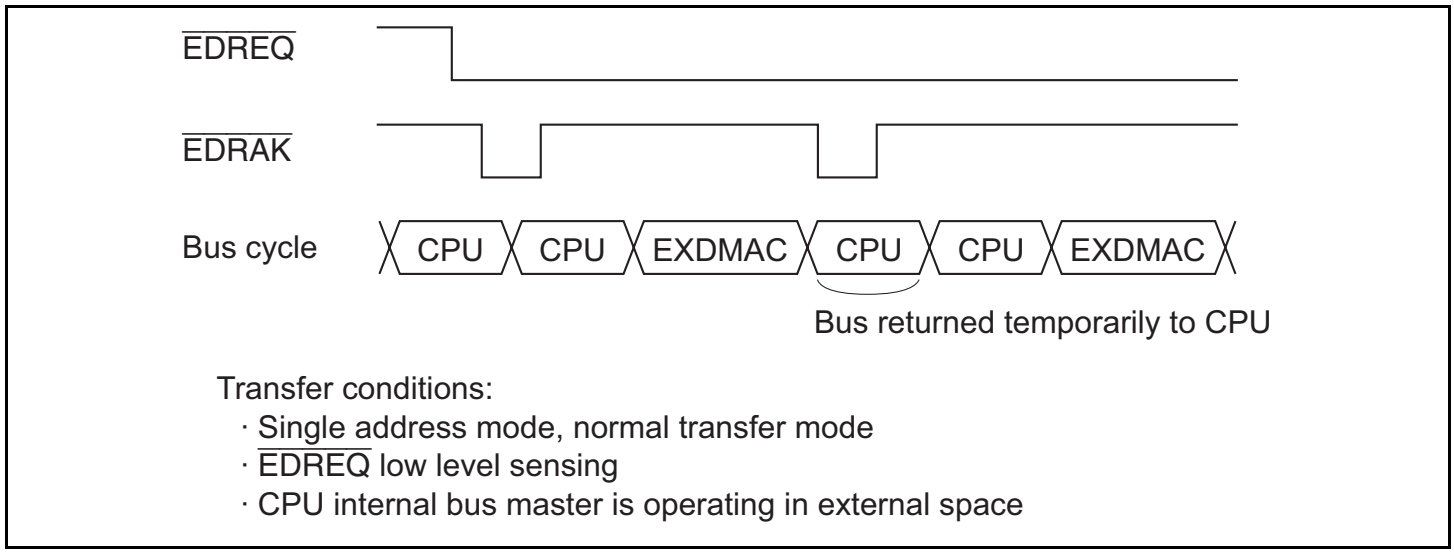


Figure 8.5 Example of Timing in Cycle Steal Mode

(2) Burst Mode

In burst mode, once the EXDMAC acquires the bus it continues transferring data, without releasing the bus, until the transfer end condition is satisfied. There is no burst mode in external request mode.

In burst mode, once transfer is started it is not interrupted even if there is a transfer request from another channel with higher priority. When the burst mode channel finishes its transfer, it releases the bus in the next cycle in the same way as in cycle steal mode.

When the EDA bit is cleared to 0 in EDMDR, DMA transfer is halted. However, DMA transfer is executed for all transfer requests generated within the EXDMAC up until the EDA bit was cleared to 0.

If a repeat area overflow interrupt is generated, the EDA bit is cleared to 0 and transfer is terminated.

When the BGUP bit is set to 1 in EDMDR, the bus is released if a bus request is issued by another bus master during burst transfer. If there is no bus request, burst transfer is executed even if the BGUP bit is set to 1.

Figure 8.6 shows examples of the timing in burst mode.

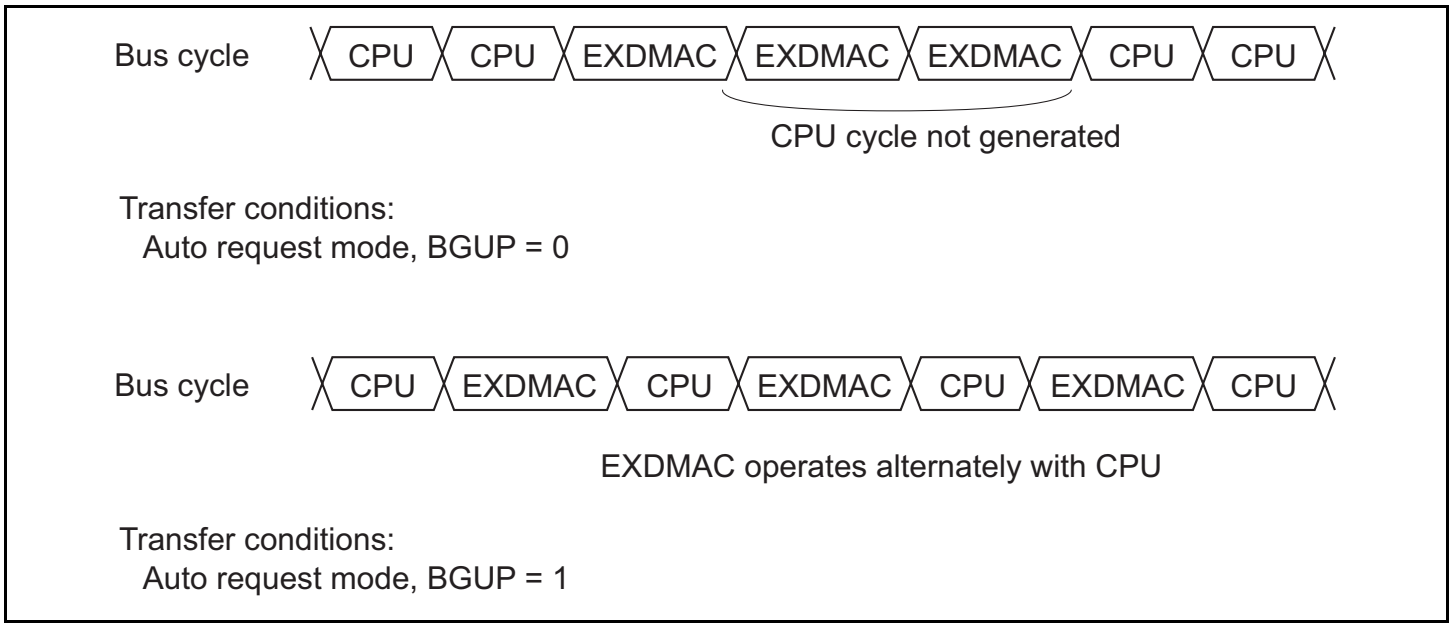


Figure 8.6 Examples of Timing in Burst Mode

8.4.5 Transfer Modes

There are two transfer modes: normal transfer mode and block transfer mode. When the activation source is an external request, either normal transfer mode or block transfer mode can be selected. When the activation source is an auto request, normal transfer mode is used.

(1) Normal Transfer Mode

In normal transfer mode, transfer of one transfer unit is processed in response to one transfer request. EDTCR functions as a 24-bit transfer counter.

The $\overline{\text{ETEND}}$ signal is output only for the last DMA transfer. The $\overline{\text{EDRAK}}$ signal is output each time a transfer request is accepted and transfer processing is started.

Figure 8.7 shows examples of DMA transfer timing in normal transfer mode.

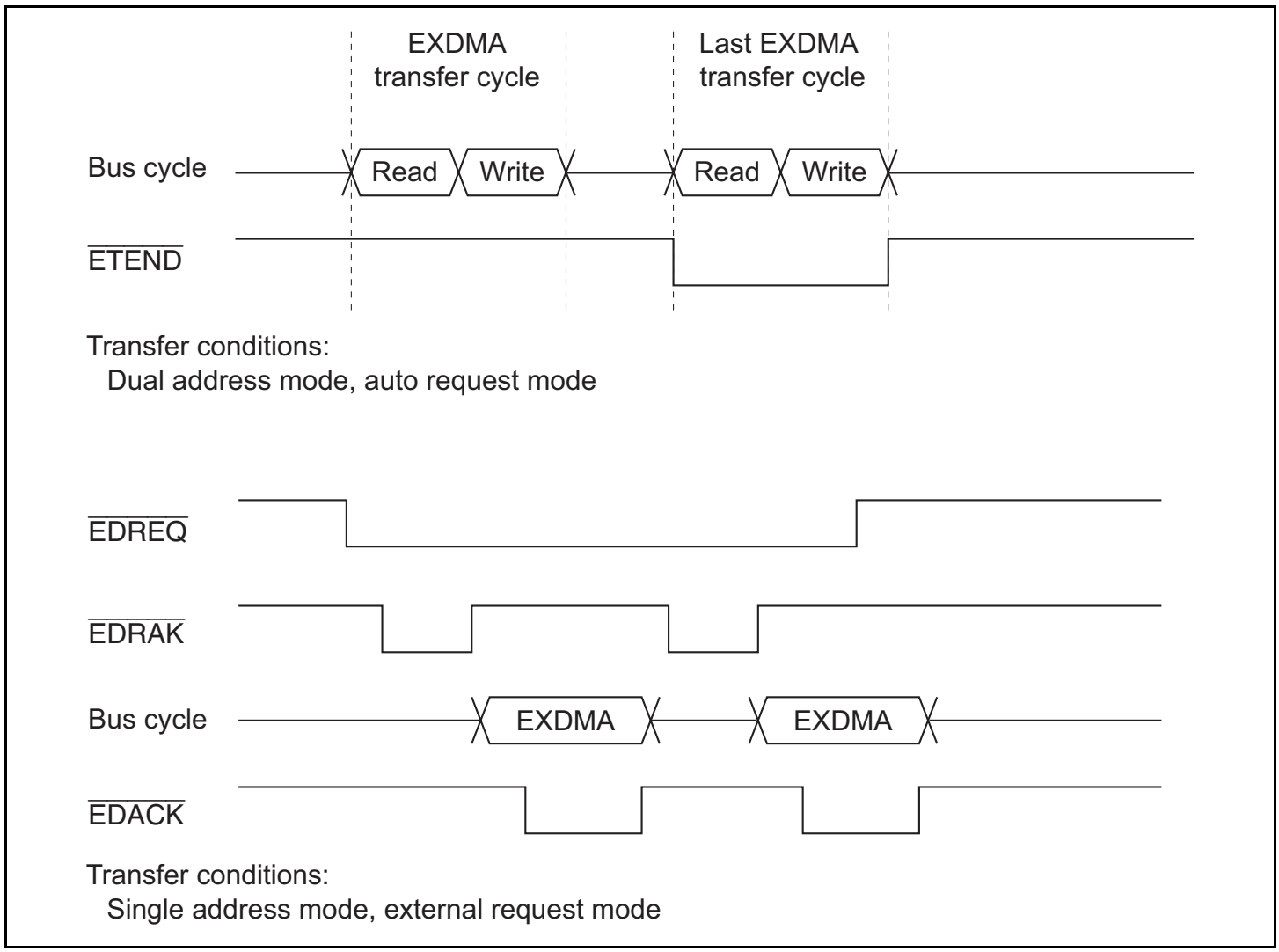


Figure 8.7 Examples of Timing in Normal Transfer Mode

(2) Block Transfer Mode

In block transfer mode, the number of bytes or words specified by the block size is transferred in response to one transfer request. The upper 8 bits of EDTCR specify the block size, and the lower 16 bits function as a 16-bit transfer counter. A block size of 1 to 256 can be specified. During transfer of a block, transfer requests for other higher-priority channels are held pending. When transfer of one block is completed, the bus is released in the next cycle.

When the BGUP bit is set to 1 in EDMDR, the bus is released if a bus request is issued by another bus master during block transfer.

Address register values are updated in the same way as in normal mode. There is no function for restoring the initial address register values after each block transfer.

The $\overline{\text{ETEND}}$ signal is output for each block transfer in the DMA 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 repeat area overflow interrupt of the repeat area function in block transfer mode. See section 8.4.6, Repeat Area Function, for details.

Block transfer is aborted if an NMI interrupt is generated. See section 8.4.12, Ending DMA Transfer, for details.

Figure 8.8 shows an example of DMA transfer timing in block transfer mode.

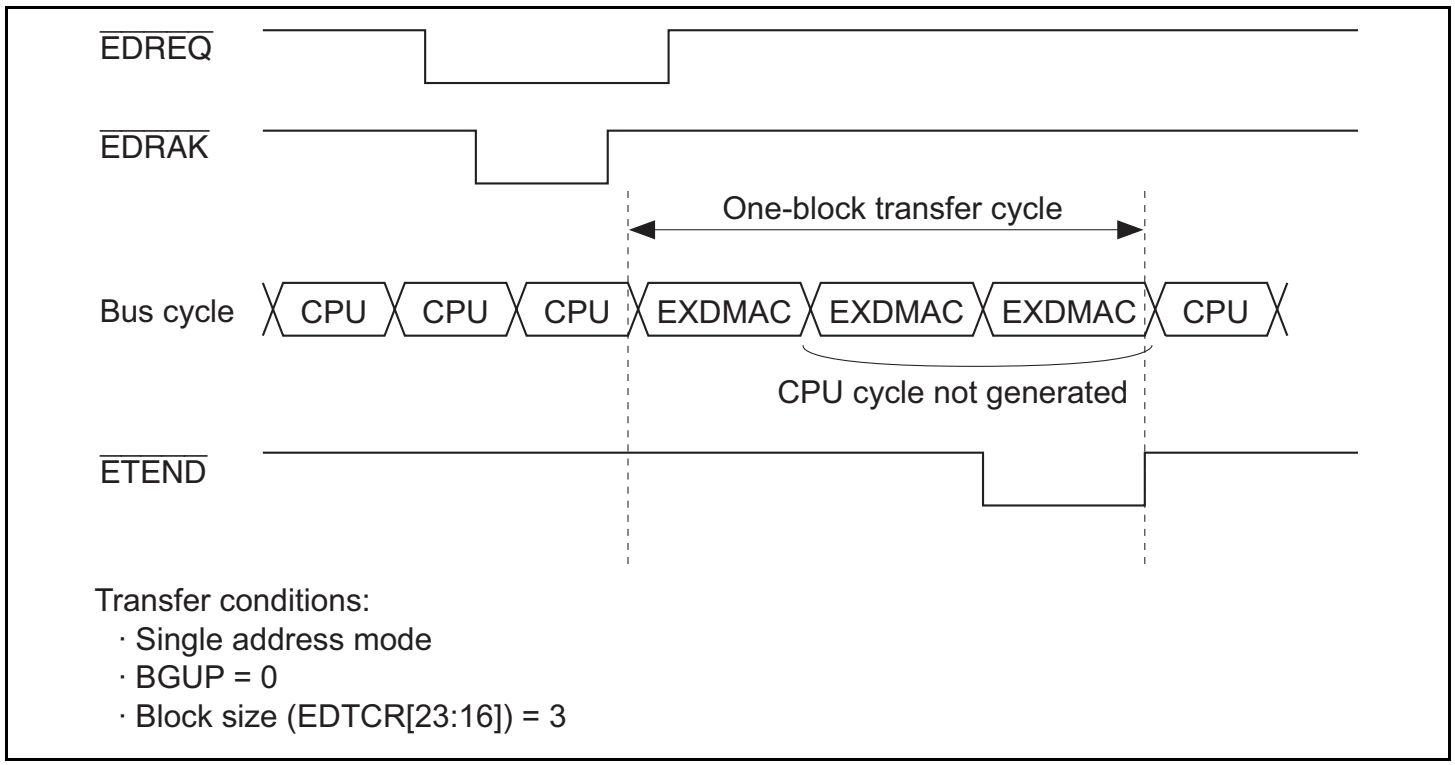


Figure 8.8 Example of Timing in Block Transfer Mode

8.4.6 Repeat Area Function

The EXDMAC has a function for designating a repeat area for source addresses and/or destination addresses. When a repeat area is designated, the address register values repeat within the range specified as the repeat area. Normally, when a ring buffer is involved in a transfer, an operation is required to restore the address register value to the buffer start address each time the address register value is the last address in the buffer (i.e. when ring buffer address overflow occurs), but if the repeat area function is used, the operation that restores the address register value to the buffer start address is performed automatically within the EXDMAC.

The repeat area function can be set independently for the source address register and the destination address register.

The source address repeat area is specified by bits SARA4 to SARA0 in EDACR, and the destination address repeat area by bits DARA4 to DARA0 in EDACR. The size of each repeat area can be specified independently.

When the address register value is the last address in the repeat area and repeat area overflow occurs, DMA transfer can be temporarily halted and an interrupt request sent to the CPU. If the SARIE bit in EDACR is set to 1, when the source address register overflows the repeat area, the IRF bit is set to 1 and the EDA bit cleared to 0 in EDMDR, and transfer is terminated. If EDIE = 1 in EDMDR, an interrupt is requested. If the DARIE bit in EDACR is set to 1, the above applies to the destination address register.

If the EDA bit in EDMDR is set to 1 during interrupt generation, transfer is resumed. Figure 8.9 illustrates the operation of the repeat area function.

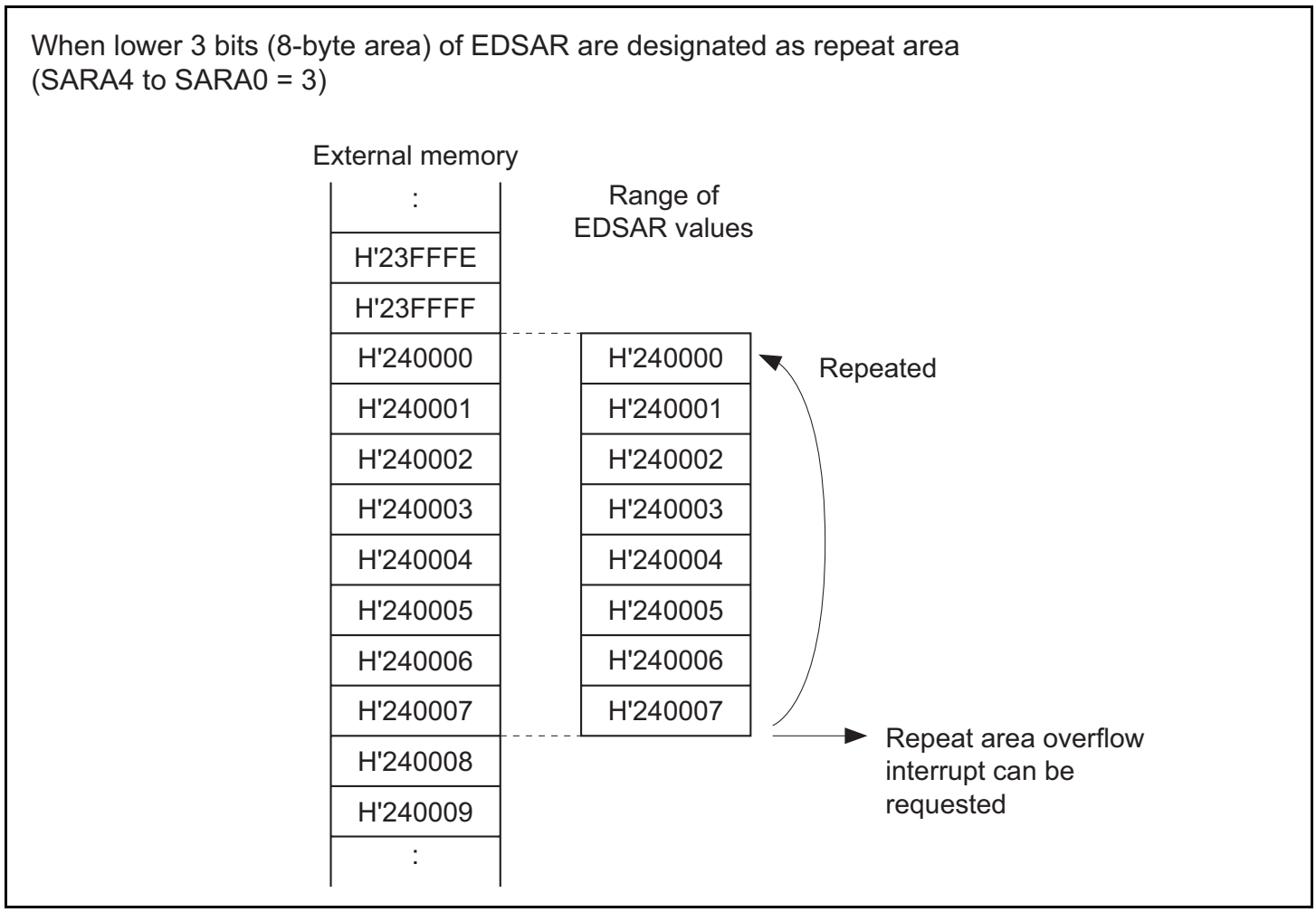


Figure 8.9 Example of Repeat Area Function Operation

Caution is required when the repeat area overflow interrupt function is used together with block transfer mode. If transfer is always terminated when repeat area overflow occurs in block transfer mode, the block size must be a power of two, or alternatively, the address register value must be set so that the end of a block coincides with the end of the repeat area range.

If repeat area overflow occurs while a block is being transferred in block transfer mode, the repeat interrupt request is held pending until the end of the block, and transfer overrun will occur. Figure 8.10 shows an example in which block transfer mode is used together with the repeat area function.

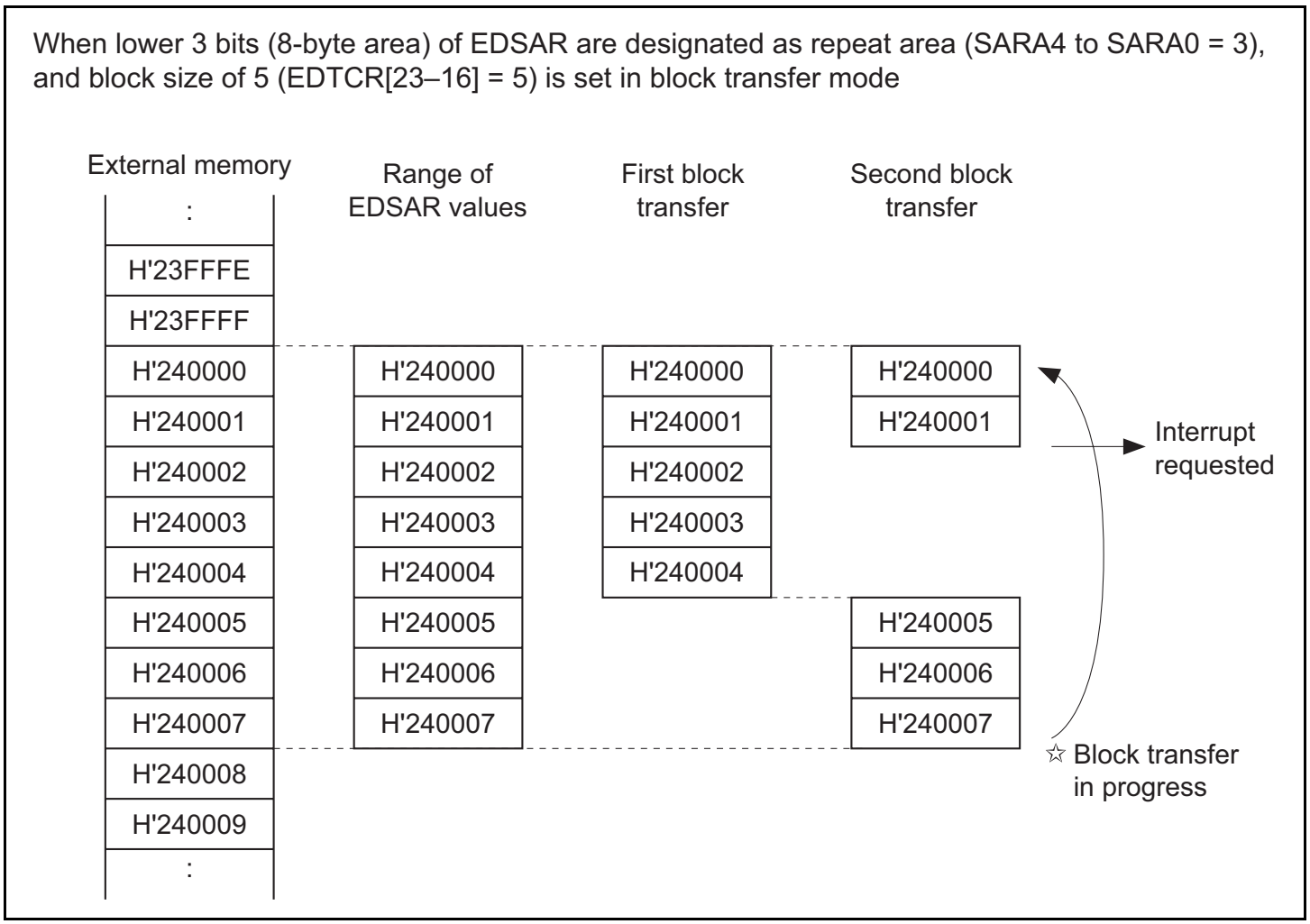


Figure 8.10 Example of Repeat Area Function Operation in Block Transfer Mode

8.4.7 Registers during DMA Transfer Operation

EXDMAC register values are updated as DMA transfer processing is performed. The updated values depend on various settings and the transfer status. The following registers and bits are updated: EDSAR, EDDAR, EDTCR, and bits EDA, BEF, and IRF in EDMDR,

(1) EXDMA Source Address Register (EDSAR)

When the EDSAR address is accessed as the transfer source, after the EDSAR value is output, EDSAR is updated with the address to be accessed next. Bits SAT1 and SAT0 in EDACR specify incrementing or decrementing. The address is fixed when SAT1 = 0, incremented when SAT1 = 1 and SAT0 = 0, and decremented when SAT1 = 1 and SAT0 = 1.

The size of the increment or decrement is determined by the size of the data transferred. When the DTSIZE bit in EDMDR = 0, the data is byte-size and the address is incremented or decremented by 1; when DTSIZE = 1, the data is word-size and the address is incremented or decremented by 2.

When a repeat area setting is made, the operation conforms to that setting. The upper part of the address set for the repeat area function is fixed, and is not affected by address updating.

When EDSAR is read during a transfer operation, a longword access must be used. During a transfer operation, EDSAR may be updated without regard to accesses from the CPU, and the correct values may not be read if the upper and lower words are read separately. In a longword access, the EXDMAC buffers the EDSAR value to ensure that the correct value is output.

Do not write to EDSAR for a channel on which a transfer operation is in progress.

(2) EXDMA Destination Address Register (EDDAR)

When the EDDAR address is accessed as the transfer destination, after the EDDAR value is output, EDDAR is updated with the address to be accessed next. Bits DAT1 and DAT0 in EDACR specify incrementing or decrementing. The address is fixed when DAT1 = 0, incremented when DAT1 = 1 and DAT0 = 0, and decremented when DAT1 = 1 and DAT0 = 1.

The size of the increment or decrement is determined by the size of the data transferred. When the DTSIZE bit in EDMDR = 0, the data is byte-size and the address is incremented or decremented by 1; when DTSIZE = 1, the data is word-size and the address is incremented or decremented by 2.

When a repeat area setting is made, the operation conforms to that setting. The upper part of the address set for the repeat area function is fixed, and is not affected by address updating.

When EDDAR is read during a transfer operation, a longword access must be used. During a transfer operation, EDDAR may be updated without regard to accesses from the CPU, and the correct values may not be read if the upper and lower words are read separately. In a longword access, the EXDMAC buffers the EDDAR value to ensure that the correct value is output.

Do not write to EDDAR for a channel on which a transfer operation is in progress.

(3) EXDMA Transfer Count Register (EDTCR)

When a DMA transfer is performed, the value in EDTCR is decremented by 1. However, when the EDTCR value is 0, transfers are not counted and the EDTCR value does not change.

EDTCR functions differently in block transfer mode. The upper 8 bits, EDTCR[23:16], are used to specify the block size, and their value does not change. The lower 16 bits, EDTCR[15:0], function as a transfer counter, the value of which is decremented by 1 when a DMA transfer is performed. However, when the EDTCR[15:0] value is 0, transfers are not counted and the EDTCR[15:0] value does not change.

In normal transfer mode, all of the lower 24 bits of EDTCR may change, so when EDTCR is read by the CPU during DMA transfer, a longword access must be used. During a transfer operation, EDTCR may be updated without regard to accesses from the CPU, and the correct values may not be read if the upper and lower words are read separately. In a longword access, the EXDMAC buffers the EDTCR value to ensure that the correct value is output.

In block transfer mode, the upper 8 bits are never updated, so there is no problem with using word access.

Do not write to EDTCR for a channel on which a transfer operation is in progress. If there is contention between an address update associated with DMA transfer and a write by the CPU, the CPU write has priority.

In the event of contention between an EDTCR update from 1 to 0 and a write (of a nonzero value) by the CPU, the CPU write value has priority as the EDTCR value, but transfer is terminated. Transfer does not end if the CPU writes 0 to EDTCR.

Figure 8.11 shows EDTCR update operations in normal transfer mode and block transfer mode.

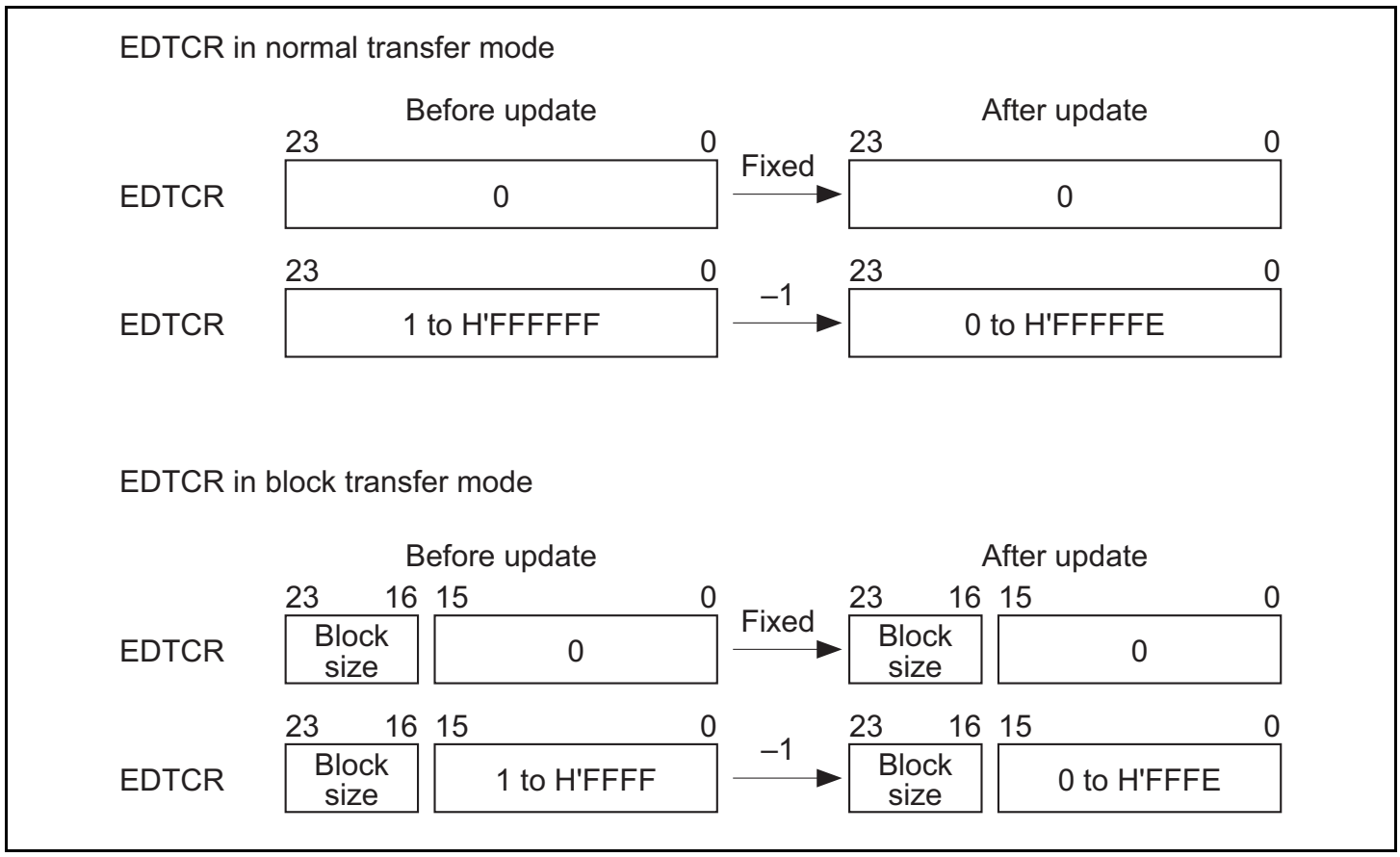


Figure 8.11 EDTCR Update Operations in Normal Transfer Mode and Block Transfer Mode

(4) EDA Bit in EDMDR

The EDA bit in EDMDR is written to by the CPU to control enabling and disabling of data transfer, but may be cleared automatically by the EXDMAC due to the DMA transfer status. There are also periods during transfer when a 0-write to the EDA bit by the CPU is not immediately effective.

Conditions for EDA bit clearing by the EXDMAC include the following:

- When the EDTCR value changes from 1 to 0, and transfer ends
- When a repeat area overflow interrupt is requested, and transfer ends
- When an NMI interrupt is generated, and transfer halts
- A reset
- Hardware standby mode
- When 0 is written to the EDA bit, and transfer halts

When transfer is halted by writing 0 to the EDA bit, the EDA bit remains at 1 during the DMA transfer period. In block transfer mode, since a block-size transfer is carried out without interruption, the EDA bit remains at 1 from the time 0 is written to it until the end of the current block-size transfer.

In burst mode, transfer is halted for up to three DMA transfers following the bus cycle in which 0 is written to the EDA bit. The EDA bit remains set to 1 from the time of the 0-write until the end of the last DMA cycle.

Writes (except to the EDA bit) are prohibited to registers of a channel for which the EDA bit is set to 1. When changing register settings after a 0-write to the EDA bit, it is necessary to confirm that the EDA bit has been cleared to 0.

Figure 8.12 shows the procedure for changing register settings in an operating channel.

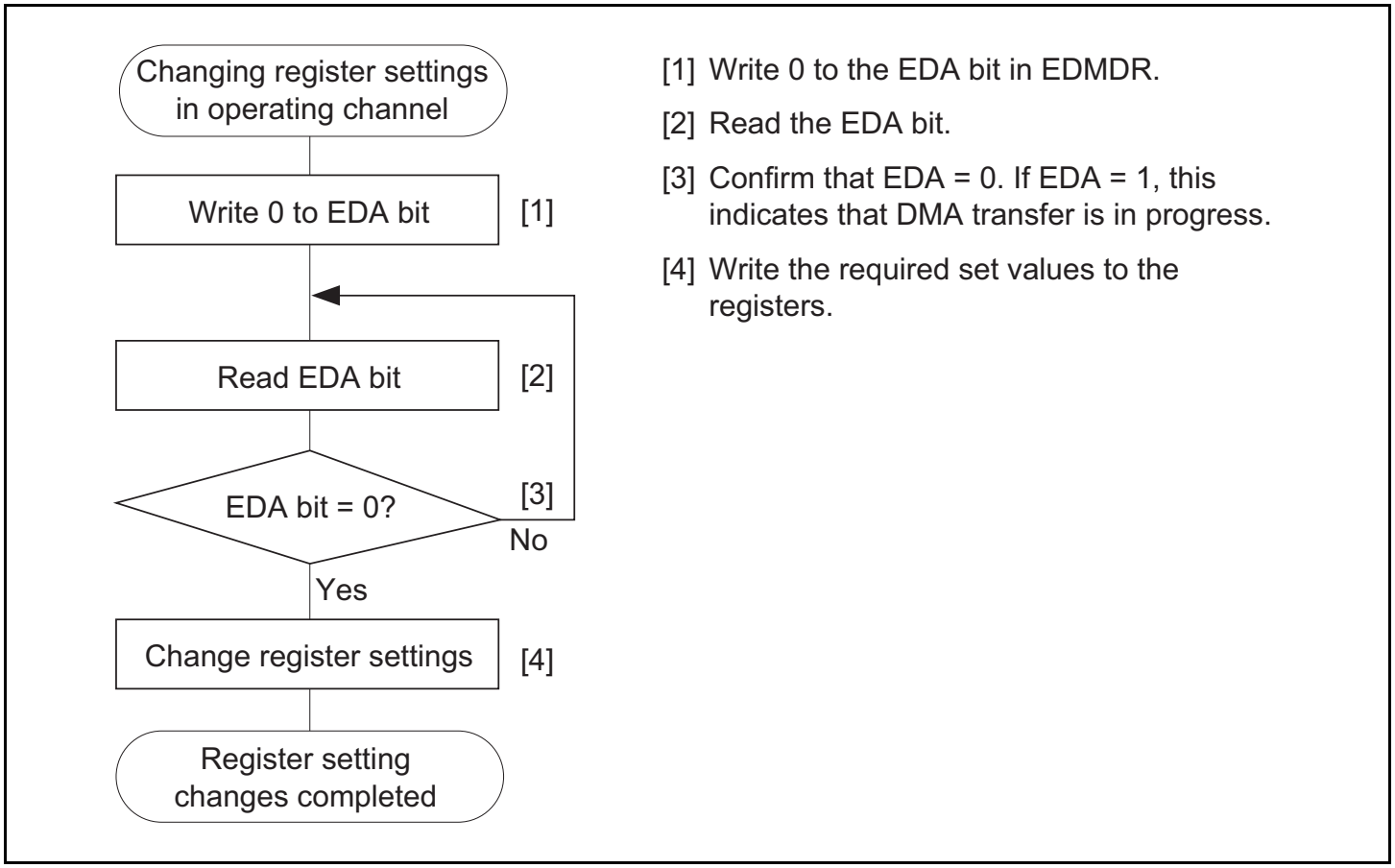


Figure 8.12 Procedure for Changing Register Settings in Operating Channel

(5) BEF Bit in EDMDR

In block transfer mode, the specified number of transfers (equivalent to the block size) is performed in response to a single transfer request. To ensure that the correct number of transfers is carried out, a block-size transfer is always executed, except in the event of a reset, transition to standby mode, or generation of an NMI interrupt.

If an NMI interrupt is generated during block transfer, operation is halted midway through a block-size transfer and the EDA bit is cleared to 0, terminating the transfer operation. In this case the BEF bit, which indicates the occurrence of an error during block transfer, is set to 1.

(6) IRF Bit in EDMDR

The IRF bit in EDMDR is set to 1 when an interrupt request source occurs. If the EDIE bit in EDMDR is 1 at this time, an interrupt is requested.

The timing for setting the IRF bit to 1 is when the EDA bit in EDMDR is cleared to 0 and transfer ends following the end of the DMA transfer bus cycle in which the source generating the interrupt occurred.

If the EDA bit is set to 1 and transfer is resumed during interrupt handling, the IRF bit is automatically cleared to 0 and the interrupt request is cleared.

For details on interrupts, see section 8.5, Interrupt Sources.

8.4.8 Channel Priority Order

The priority order of the EXDMAC channels is: channel 2 > channel 3. Table 8.3 shows the EXDMAC channel priority order.

Table 8.3 EXDMAC Channel Priority Order

Channel	Priority
Channel 2	High ↑
Channel 3	Low

If transfer requests occur simultaneously for a number of channels, the highest-priority channel according to the priority order in table 8.3 is selected for transfer.

(1) Transfer Requests from Multiple Channels (Except Auto Request Cycle Steal Mode)

If transfer requests for different channels are issued during a transfer operation, the highest-priority channel (excluding the currently transferring channel) is selected. The selected channel begins transfer after the currently transferring channel releases the bus. If there is a bus request from a bus master other than the EXDMAC at this time, a cycle for the other bus master is initiated. If there is no other bus request, the bus is released for one cycle.

Channel switching does not take place during a burst transfer or a block transfer of a single block. Figure 8.13 shows a case in which transfer requests for channels 2 and 3 are issued simultaneously. The example shown in the figure illustrates the handling of external requests in the cycle steal mode.

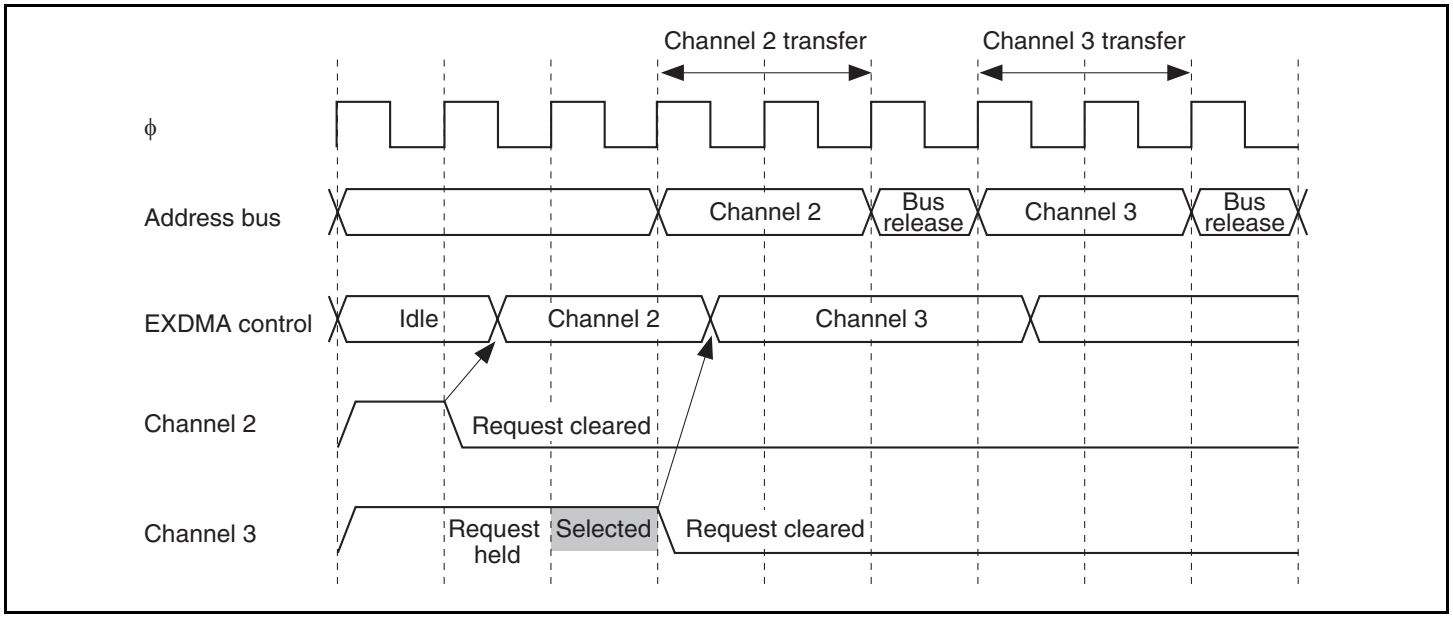


Figure 8.13 Example of Channel Priority Timing

(2) Transfer Requests from Multiple Channels in Auto Request Cycle Steal Mode

If transfer requests for different channels are issued during a transfer in auto request cycle steal mode, the operation depends on the channel priority. If the channel that made the transfer request is of higher priority than the channel currently performing transfer, the channel that made the transfer request is selected.

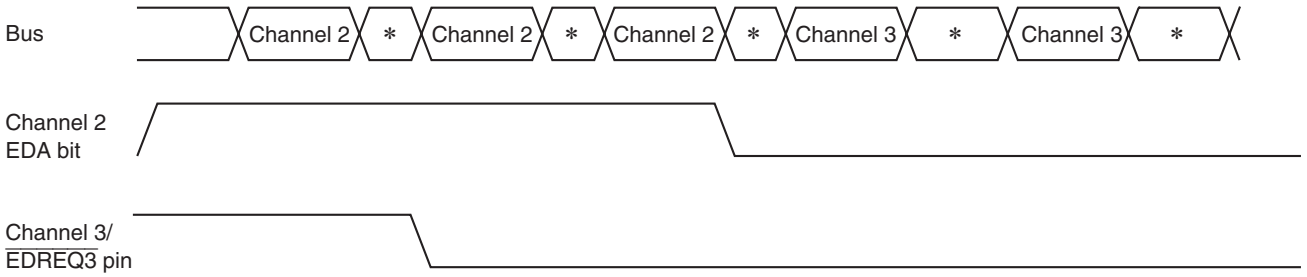
If the channel that made the transfer request is of lower priority than the channel currently performing transfer, that channel's transfer request is held pending, and the currently transferring channel remains selected.

The selected channel begins transfer after the currently transferring channel releases the bus. If there is a bus request from a bus master other than the EXDMAC at this time, a cycle for the other bus master is initiated. If there is no other bus request, the bus is released for one cycle.

Figure 8.14 shows examples of transfer timing in cases that include auto request cycle steal mode.

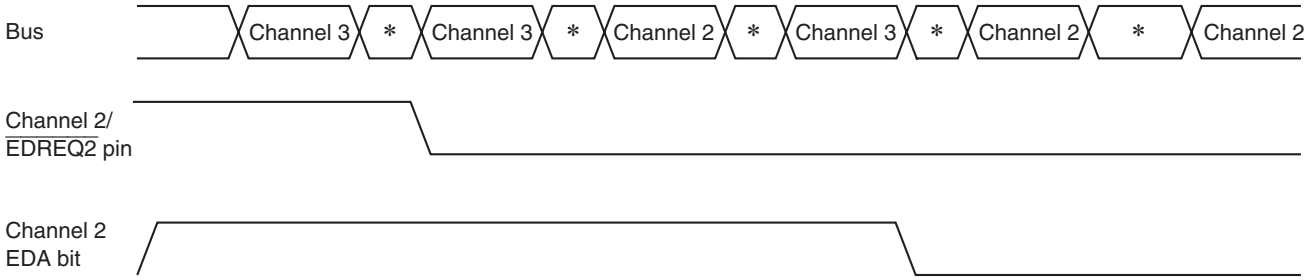
Conditions (1)

Channel 2: Auto request, cycle steal mode
Channel 3: External request, cycle steal mode, low level activation



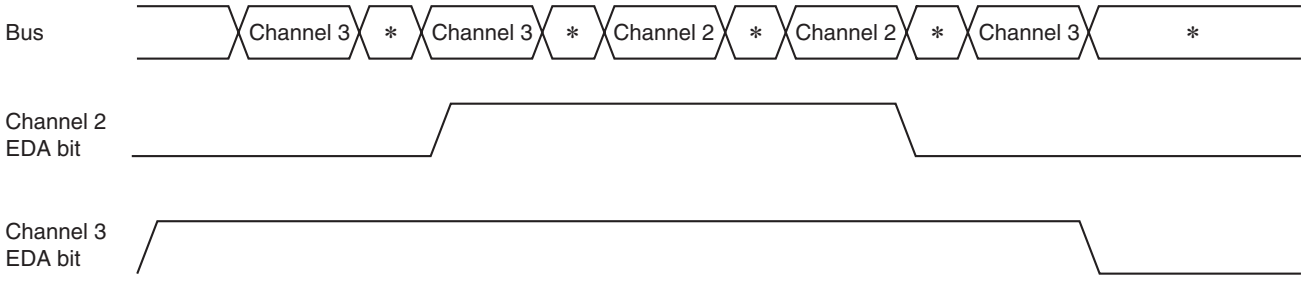
Conditions (2)

Channel 2: External request, cycle steal mode, low level activation
Channel 3: Auto request, cycle steal mode



Conditions (3)

Channel 2: Auto request, cycle steal mode
Channel 3: Auto request, cycle steal mode



*: Bus release

Figure 8.14 Examples of Channel Priority Timing

8.4.9 EXDMAC Bus Cycles (Dual Address Mode)

(1) Normal Transfer Mode (Cycle Steal Mode)

Figure 8.15 shows an example of transfer when $\overline{\text{ETEND}}$ output is enabled, and word-size, normal transfer mode (cycle steal mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

After one byte or word has been transferred, the bus is released. While the bus is released, one CPU, DMAC, or DTC bus cycle is initiated.

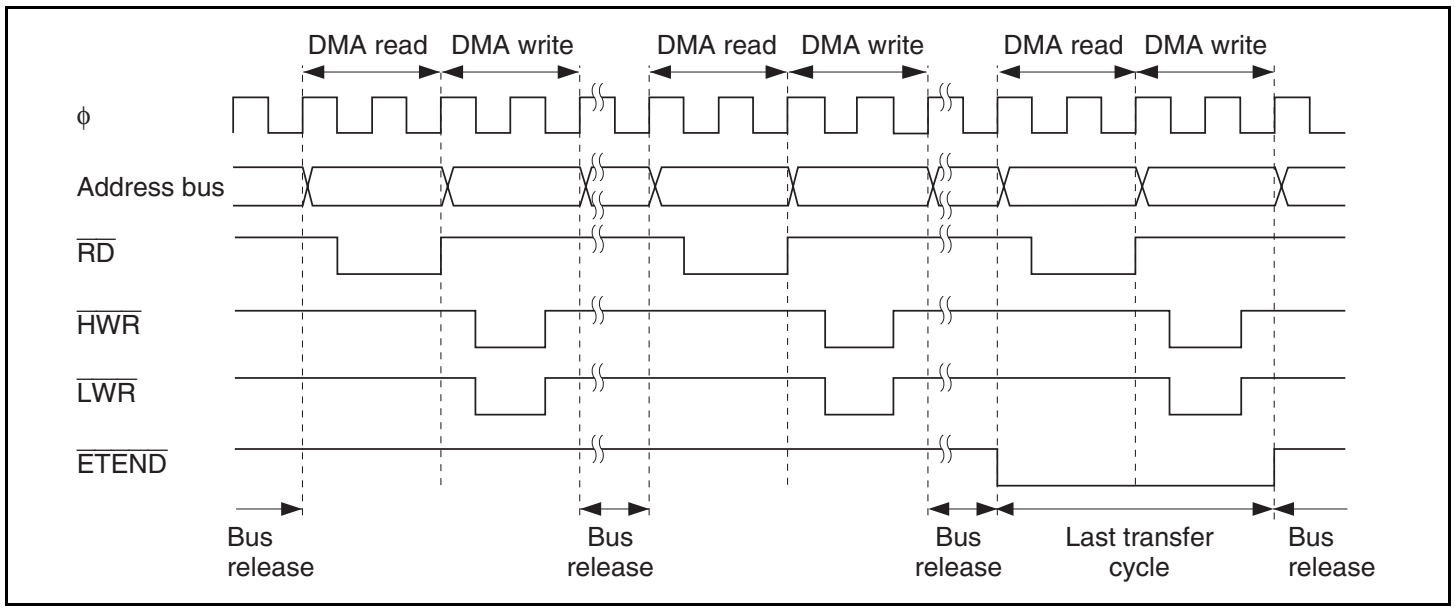


Figure 8.15 Example of Normal Transfer Mode (Cycle Steal Mode) Transfer

(2) Normal Transfer Mode (Burst Mode)

Figure 8.16 shows an example of transfer when $\overline{\text{ETEND}}$ output is enabled, and word-size, normal transfer mode (burst mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

In burst mode, one-byte or one-word transfers are executed continuously until transfer ends.

Once burst transfer starts, requests from other channels, even of higher priority, are held pending until transfer ends.

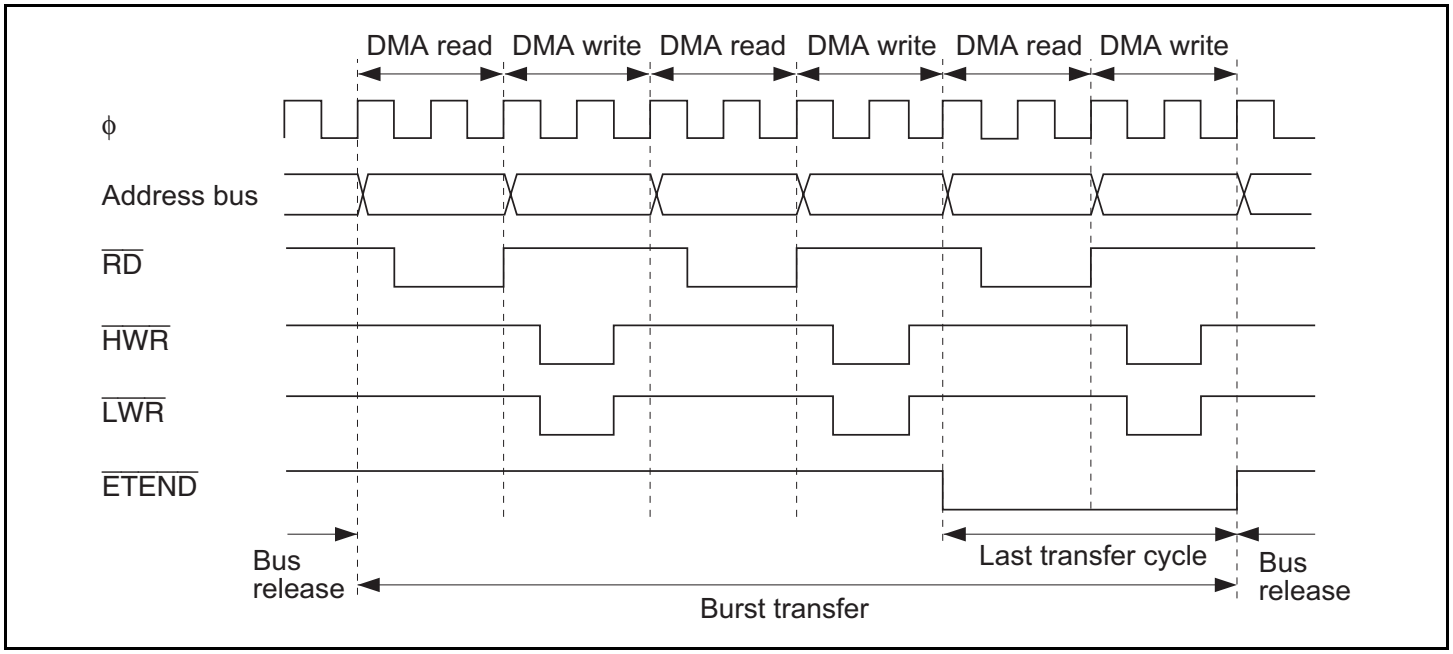


Figure 8.16 Example of Normal Transfer Mode (Burst Mode) Transfer

If an NMI interrupt is generated while a channel designated for burst transfer is enabled for transfer, the EDA bit is cleared and transfer is disabled. If a block transfer has already been initiated within the EXDMAC, the bus is released on completion of the currently executing byte or word transfer, and burst transfer is aborted. If the last transfer cycle in burst transfer has been initiated within the EXDMAC, transfer is executed to the end even if the EDA bit is cleared.

(3) Block Transfer Mode (Cycle Steal Mode)

Figure 8.17 shows an example of transfer when $\overline{\text{ETEND}}$ output is enabled, and word-size, block transfer mode (cycle steal mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

One block is transferred in response to one transfer request, and after the transfer, the bus is released. While the bus is released, one or more CPU, DMAC, or DTC bus cycles are initiated.

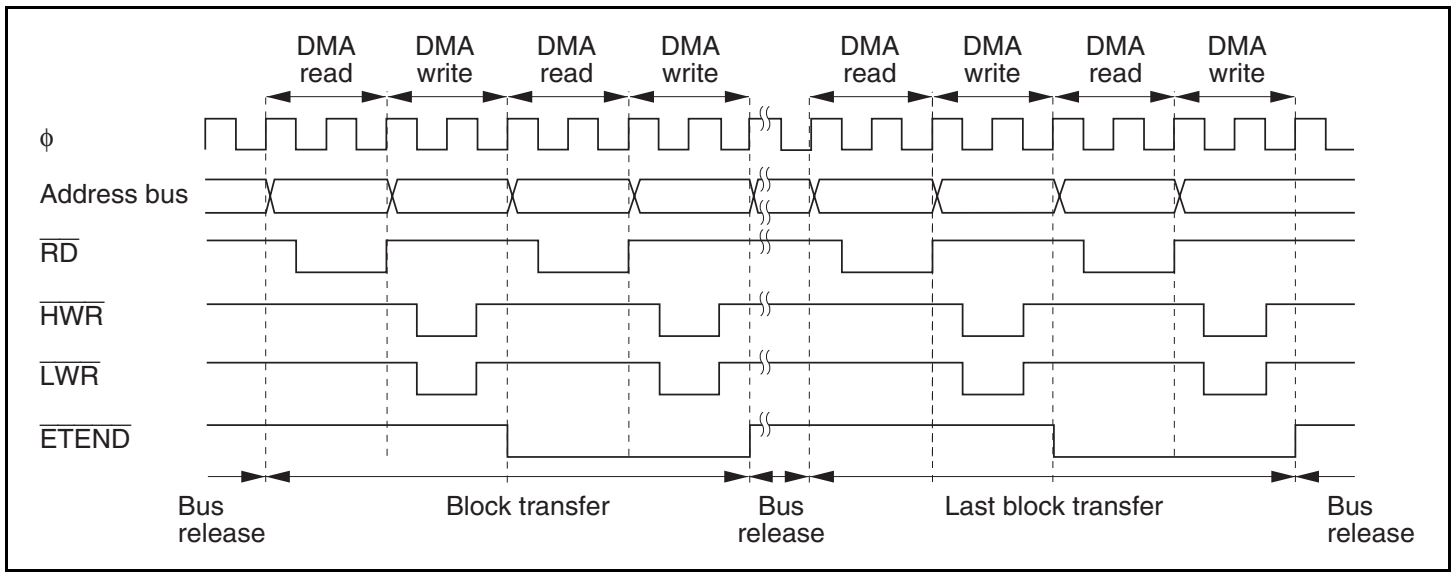
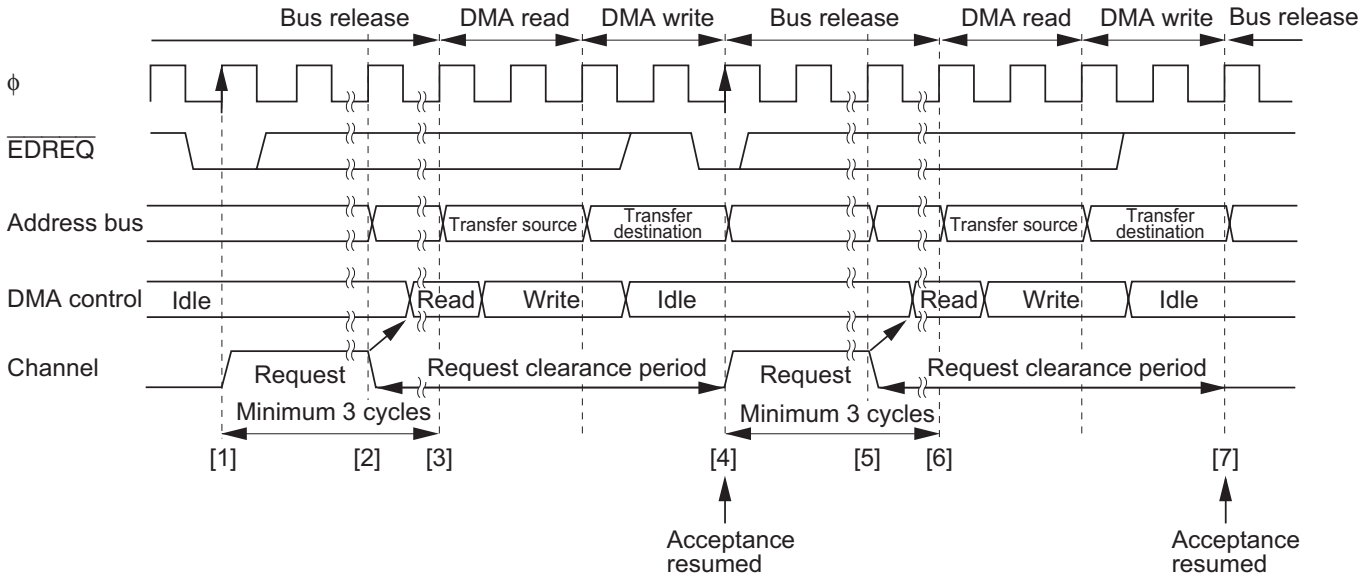


Figure 8.17 Example of Block Transfer Mode (Cycle Steal Mode) Transfer

(4) $\overline{\text{EDREQ}}$ Pin Falling Edge Activation Timing

Figure 8.18 shows an example of normal mode transfer activated by the $\overline{\text{EDREQ}}$ pin falling edge.



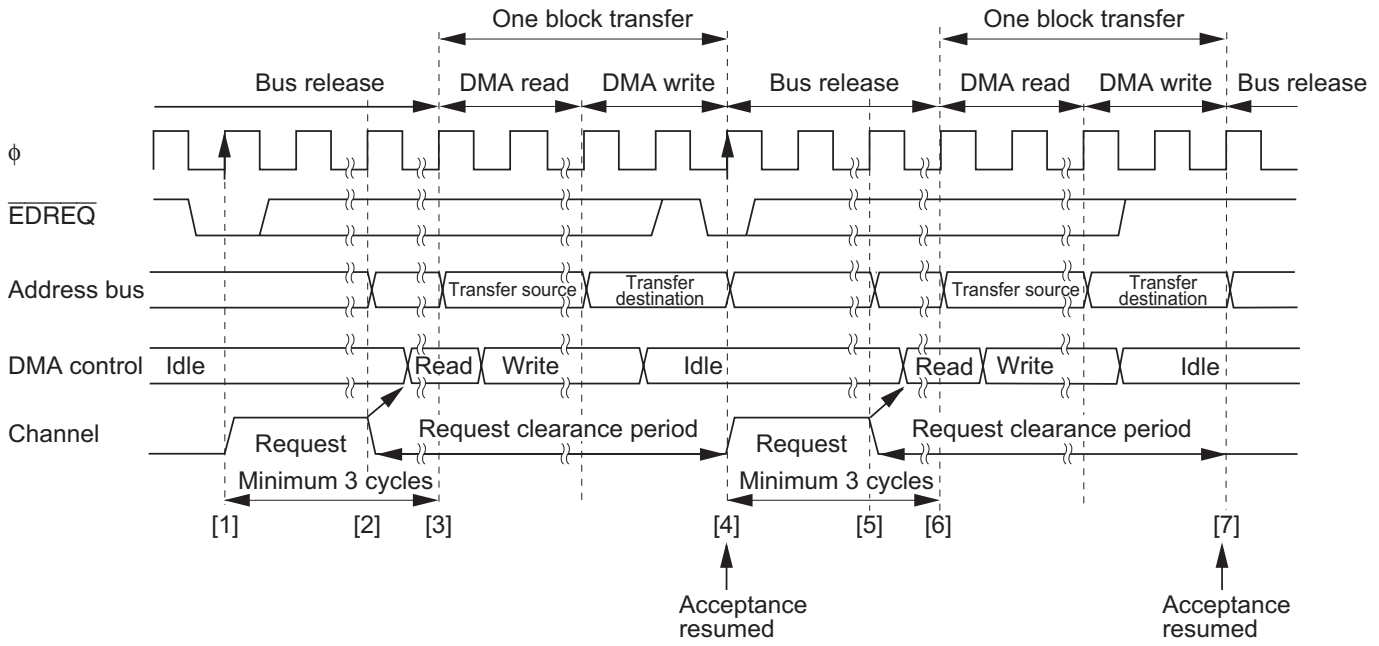
[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ϕ, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] DMA cycle start; $\overline{\text{EDREQ}}$ pin high level sampling is started at rise of ϕ.
[4], [7] When $\overline{\text{EDREQ}}$ pin high level has been sampled, acceptance is resumed after completion of write cycle. (As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ϕ, and request is held.)

Figure 8.18 Example of Normal Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Falling Edge

$\overline{\text{EDREQ}}$ pin sampling is performed in each cycle starting at the next rise of ϕ after the end of the EDMDR write cycle for setting the transfer-enabled state.

When a low level is sampled at the $\overline{\text{EDREQ}}$ pin while acceptance via the $\overline{\text{EDREQ}}$ pin is possible, the request is held within the EXDMAC. Then when activation is initiated within the EXDMAC, the request is cleared, and $\overline{\text{EDREQ}}$ pin high level sampling for edge sensing is started. If $\overline{\text{EDREQ}}$ pin high level sampling is completed by the end of the DMA write cycle, acceptance resumes after the end of the write cycle, and $\overline{\text{EDREQ}}$ pin low level sampling is performed again; this sequence of operations is repeated until the end of the transfer.

Figure 8.19 shows an example of block transfer mode transfer activated by the $\overline{\text{EDREQ}}$ pin falling edge.



[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of φ, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] DMA cycle start; $\overline{\text{EDREQ}}$ pin high level sampling is started at rise of φ.
[4], [7] When $\overline{\text{EDREQ}}$ pin high level has been sampled, acceptance is resumed after completion of dead cycle. (As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of φ, and request is held.)

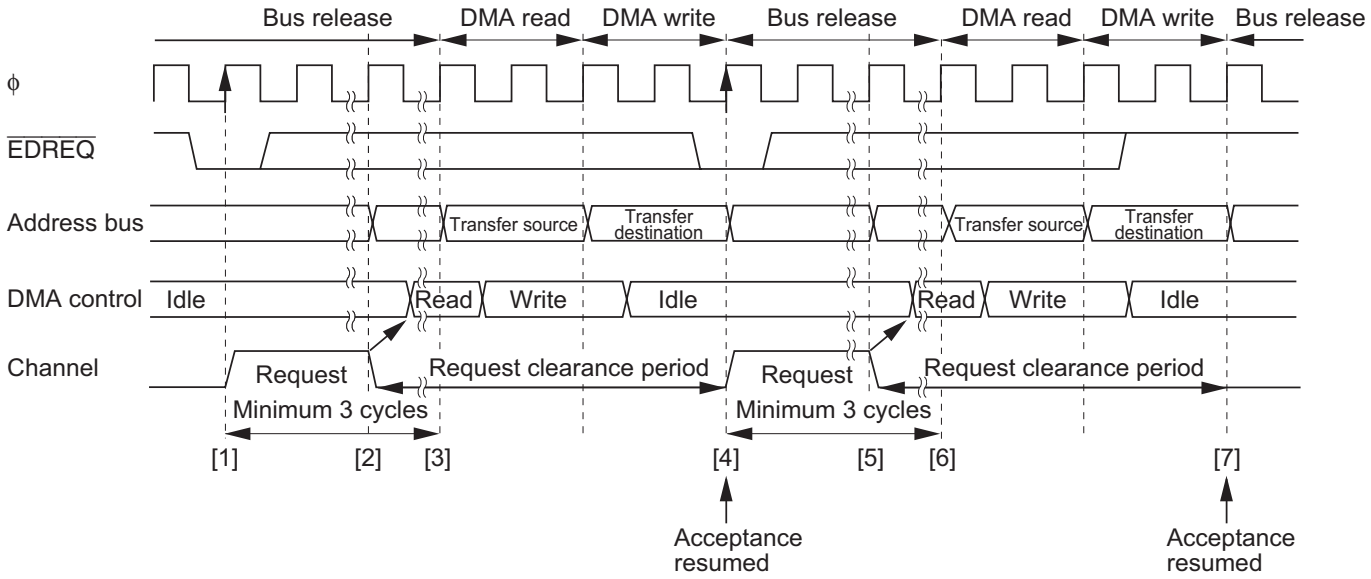
Figure 8.19 Example of Block Transfer Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Falling Edge

$\overline{\text{EDREQ}}$ pin sampling is performed in each cycle starting at the next rise of φ after the end of the EDMDR write cycle for setting the transfer-enabled state.

When a low level is sampled at the $\overline{\text{EDREQ}}$ pin while acceptance via the $\overline{\text{EDREQ}}$ pin is possible, the request is held within the EXDMAC. Then when activation is initiated within the EXDMAC, the request is cleared, and $\overline{\text{EDREQ}}$ pin high level sampling for edge sensing is started. If $\overline{\text{EDREQ}}$ pin high level sampling is completed by the end of the DMA write cycle, acceptance resumes after the end of the write cycle, and $\overline{\text{EDREQ}}$ pin low level sampling is performed again; this sequence of operations is repeated until the end of the transfer.

(5) $\overline{\text{EDREQ}}$ Pin Low Level Activation Timing

Figure 8.20 shows an example of normal mode transfer activated by the $\overline{\text{EDREQ}}$ pin low level.



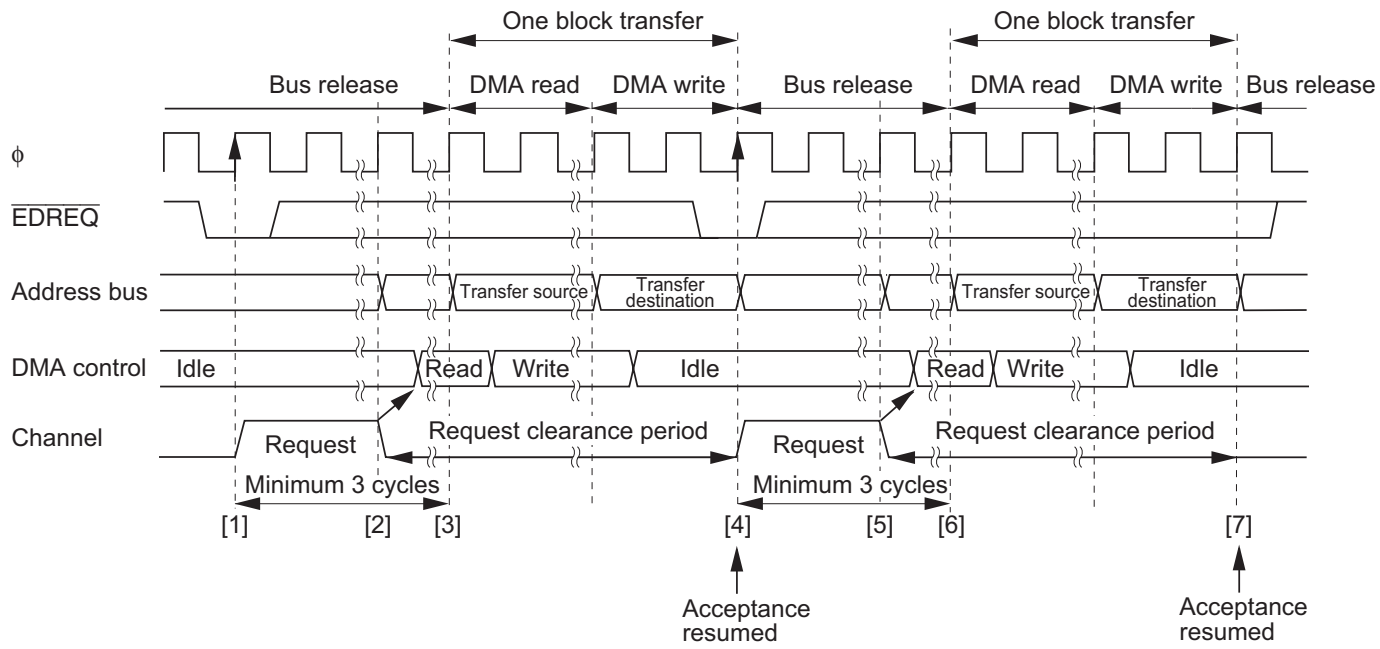
[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of φ, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] DMA cycle is started.
[4], [7] Acceptance is resumed after completion of write cycle.
(As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of φ, and request is held.)

Figure 8.20 Example of Normal Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Low Level

$\overline{\text{EDREQ}}$ pin sampling is performed in each cycle starting at the next rise of φ after the end of the EDMDR write cycle for setting the transfer-enabled state.

When a low level is sampled at the $\overline{\text{EDREQ}}$ pin while acceptance via the $\overline{\text{EDREQ}}$ pin is possible, the request is held within the EXDMAC. Then when activation is initiated within the EXDMAC, the request is cleared. At the end of the write cycle, acceptance resumes and $\overline{\text{EDREQ}}$ pin low level sampling is performed again; this sequence of operations is repeated until the end of the transfer.

Figure 8.21 shows an example of block transfer mode transfer activated by the $\overline{\text{EDREQ}}$ pin low level.



[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ϕ, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] DMA cycle is started.
[4], [7] Acceptance is resumed after completion of dead cycle.
(As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ϕ, and request is held.)

Figure 8.21 Example of Block Transfer Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Low Level

$\overline{\text{EDREQ}}$ pin sampling is performed in each cycle starting at the next rise of ϕ after the end of the EDMDR write cycle for setting the transfer-enabled state.

When a low level is sampled at the $\overline{\text{EDREQ}}$ pin while acceptance via the $\overline{\text{EDREQ}}$ pin is possible, the request is held within the EXDMAC. Then when activation is initiated within the EXDMAC, the request is cleared. At the end of the write cycle, acceptance resumes and $\overline{\text{EDREQ}}$ pin low level sampling is performed again; this sequence of operations is repeated until the end of the transfer.

8.4.10 EXDMAC Bus Cycles (Single Address Mode)

(1) Single Address Mode (Read)

Figure 8.22 shows an example of transfer when $\overline{\text{ETEND}}$ output is enabled, and byte-size, single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.

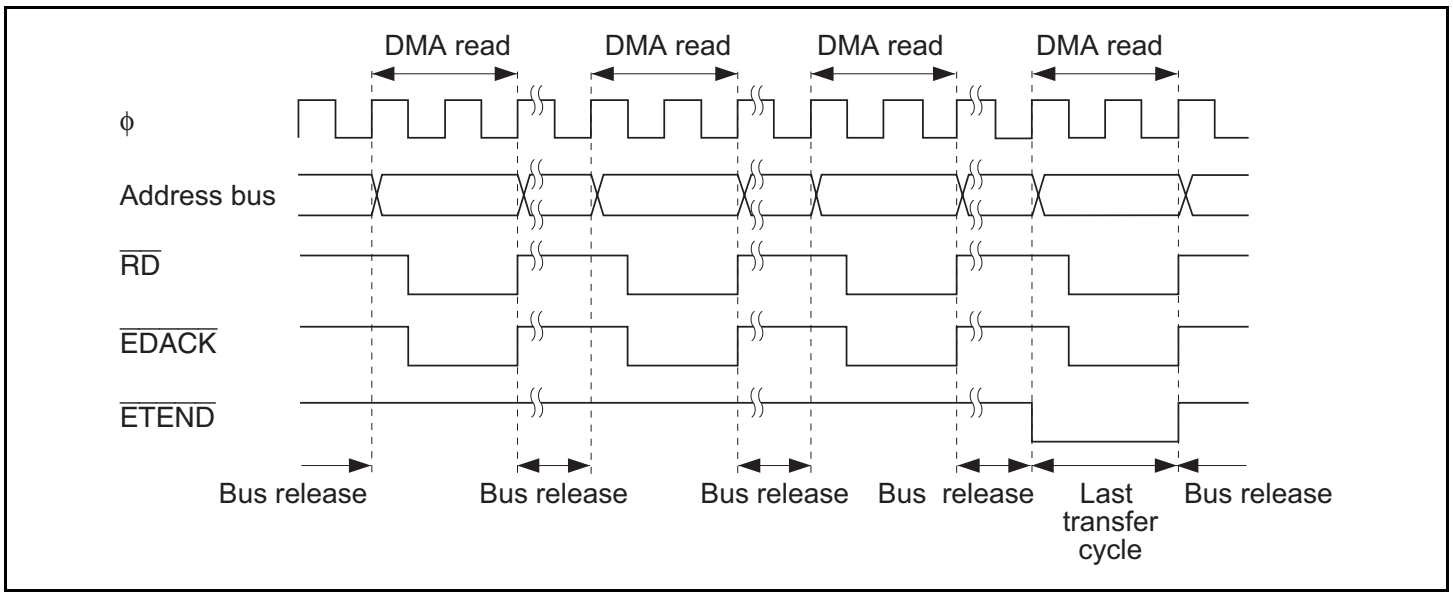


Figure 8.22 Example of Single Address Mode (Byte Read) Transfer

Figure 8.23 shows an example of transfer when $\overline{\text{ETEND}}$ output is enabled, and word-size, single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.

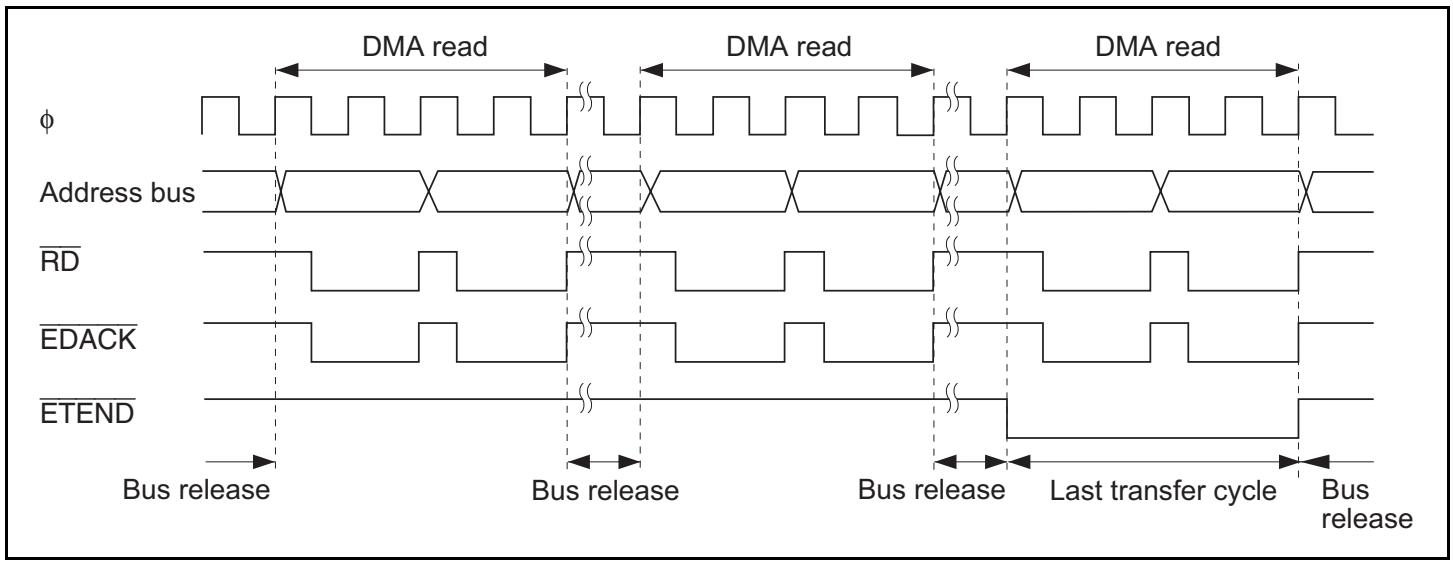


Figure 8.23 Example of Single Address Mode (Word Read) Transfer

After one byte or word has been transferred in response to one transfer request, the bus is released. While the bus is released, one or more CPU, DMAC, or DTC bus cycles are initiated.

(2) Single Address Mode (Write)

Figure 8.24 shows an example of transfer when $\overline{\text{ETEND}}$ output is enabled, and byte-size, single address mode transfer (write) is performed from an external device to external 8-bit, 2-state access space.

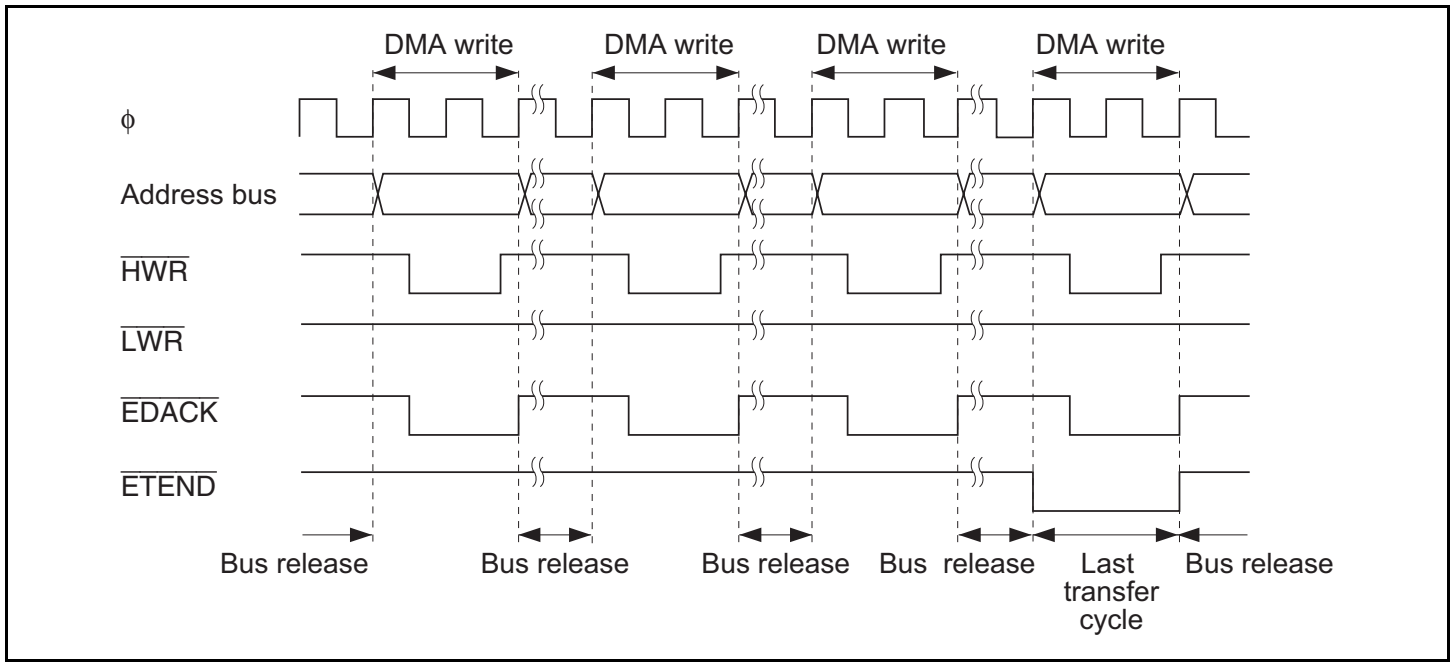


Figure 8.24 Example of Single Address Mode (Byte Write) Transfer

Figure 8.25 shows an example of transfer when $\overline{\text{ETEND}}$ output is enabled, and word-size, single address mode transfer (write) is performed from an external device to external 8-bit, 2-state access space.

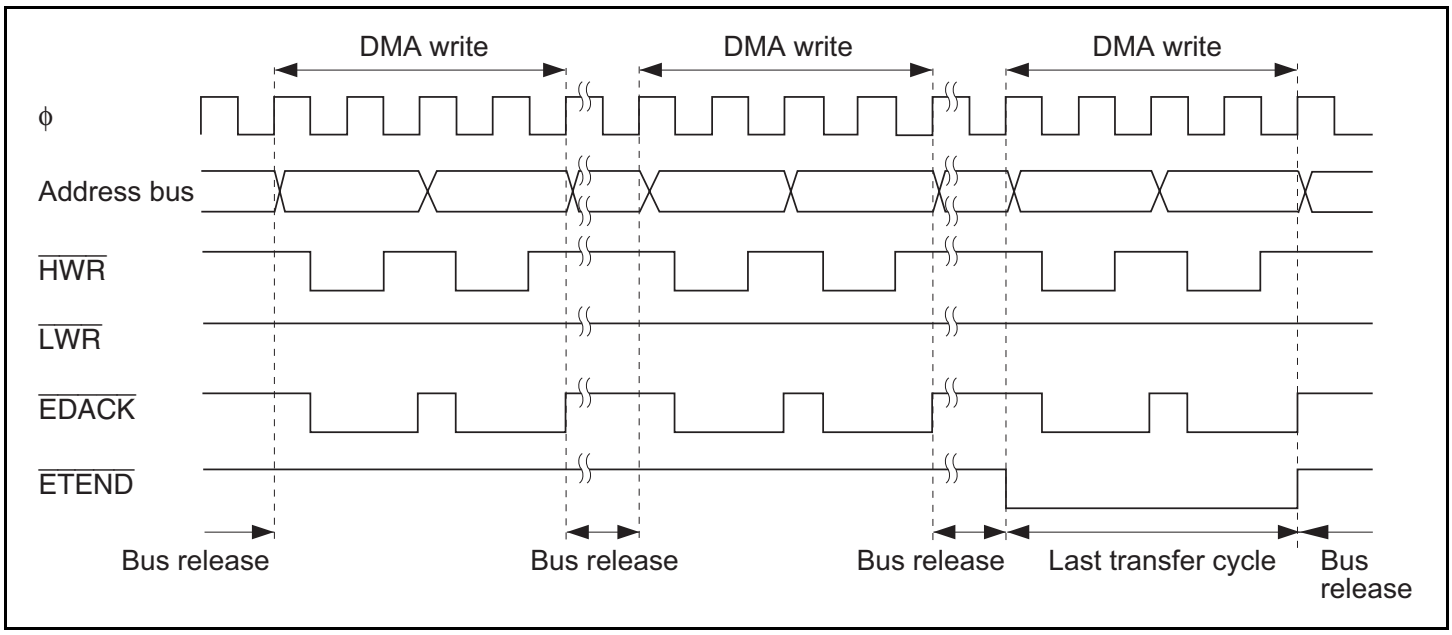
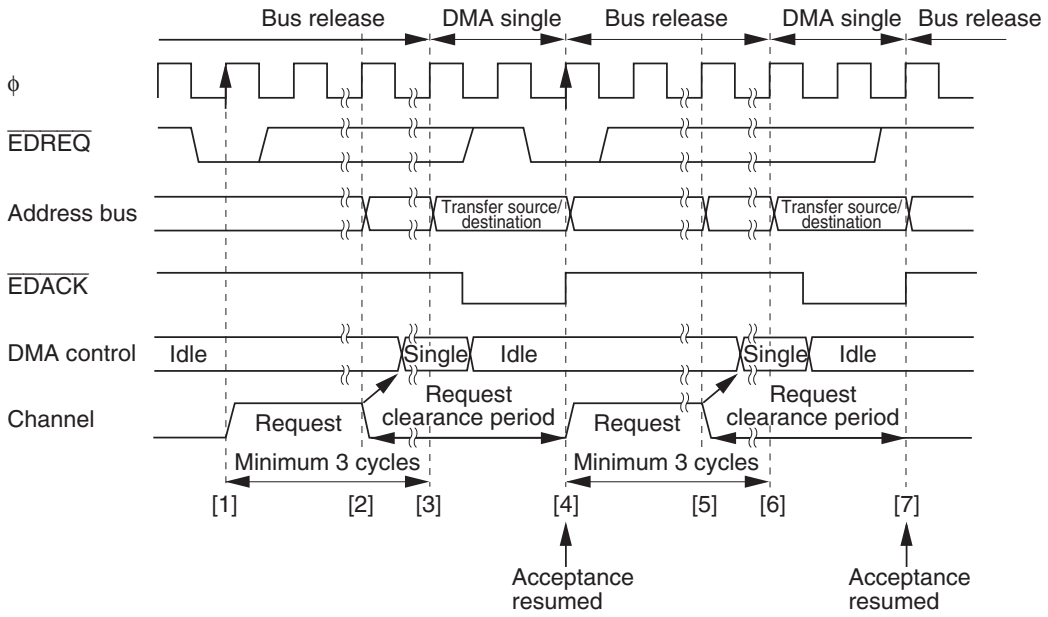


Figure 8.25 Example of Single Address Mode (Word Write) Transfer

After one byte or word has been transferred in response to one transfer request, the bus is released. While the bus is released, one or more CPU, DMAC, or DTC bus cycles are initiated.

(3) $\overline{\text{EDREQ}}$ Pin Falling Edge Activation Timing

Figure 8.26 shows an example of single address mode transfer activated by the $\overline{\text{EDREQ}}$ pin falling edge.



[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ϕ, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] DMA cycle start; $\overline{\text{EDREQ}}$ pin high level sampling is started at rise of ϕ.
[4], [7] When $\overline{\text{EDREQ}}$ pin high level has been sampled, acceptance is resumed after completion of single cycle. (As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ϕ, and request is held.)

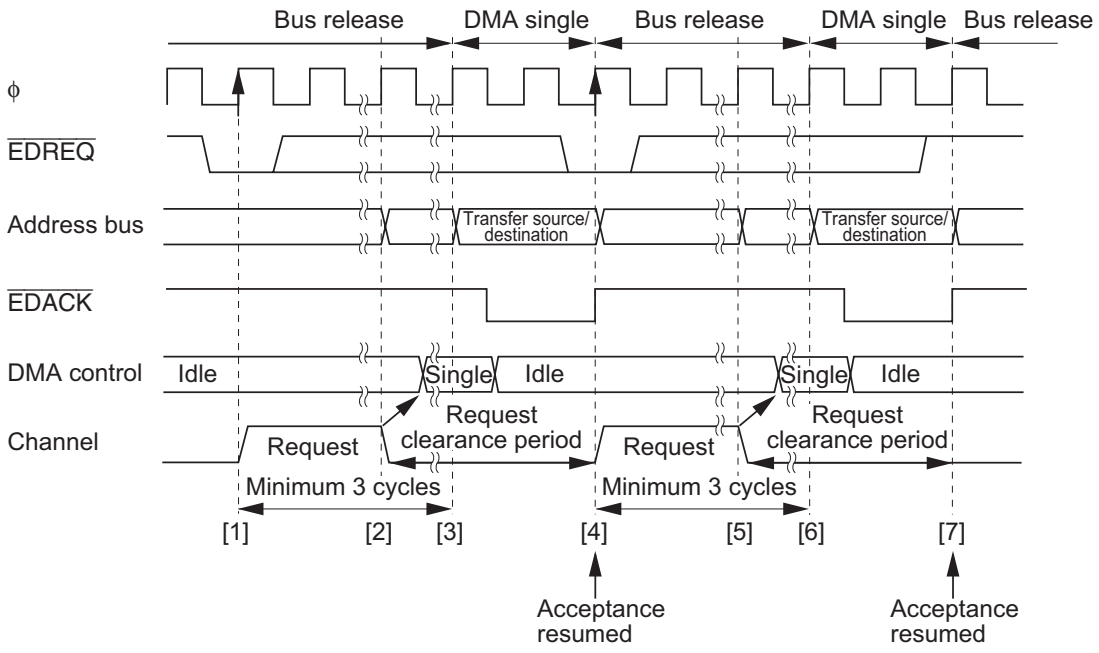
Figure 8.26 Example of Single Address Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Falling Edge

$\overline{\text{EDREQ}}$ pin sampling is performed in each cycle starting at the next rise of ϕ after the end of the EDMDR write cycle for setting the transfer-enabled state.

When a low level is sampled at the $\overline{\text{EDREQ}}$ pin while acceptance via the $\overline{\text{EDREQ}}$ pin is possible, the request is held within the EXDMAC. Then when activation is initiated within the EXDMAC, the request is cleared, and $\overline{\text{EDREQ}}$ pin high level sampling for edge sensing is started. If $\overline{\text{EDREQ}}$ pin high level sampling is completed by the end of the DMA single cycle, acceptance resumes after the end of the single cycle, and $\overline{\text{EDREQ}}$ pin low level sampling is performed again; this sequence of operations is repeated until the end of the transfer.

(4) $\overline{\text{EDREQ}}$ Pin Low Level Activation Timing

Figure 8.27 shows an example of single address mode transfer activated by the $\overline{\text{EDREQ}}$ pin low level.



[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ϕ, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] DMA cycle is started.
[4], [7] Acceptance is resumed after completion of single cycle.
(As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of ϕ, and request is held.)

Figure 8.27 Example of Single Address Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Low Level

$\overline{\text{EDREQ}}$ pin sampling is performed in each cycle starting at the next rise of ϕ after the end of the EDMDR write cycle for setting the transfer-enabled state.

When a low level is sampled at the $\overline{\text{EDREQ}}$ pin while acceptance via the $\overline{\text{EDREQ}}$ pin is possible, the request is held within the EXDMAC. Then when activation is initiated within the EXDMAC, the request is cleared. At the end of the single cycle, acceptance resumes and $\overline{\text{EDREQ}}$ pin low level sampling is performed again; this sequence of operations is repeated until the end of the transfer.

8.4.11 Examples of Operation Timing in Each Mode

(1) Auto Request/Cycle Steal Mode/Normal Transfer Mode

When the EDA bit is set to 1 in EDMDR, an EXDMA transfer cycle is started a minimum of three cycles later. There is a one-cycle bus release interval between the end of a one-transfer-unit EXDMA cycle and the start of the next transfer.

If there is a transfer request for another channel of higher priority, the transfer request by the original channel is held pending, and transfer is performed on the higher-priority channel from the next transfer. Transfer on the original channel is resumed on completion of the higher-priority channel transfer.

Figures 8.28 to 8.30 show operation timing examples for various conditions.

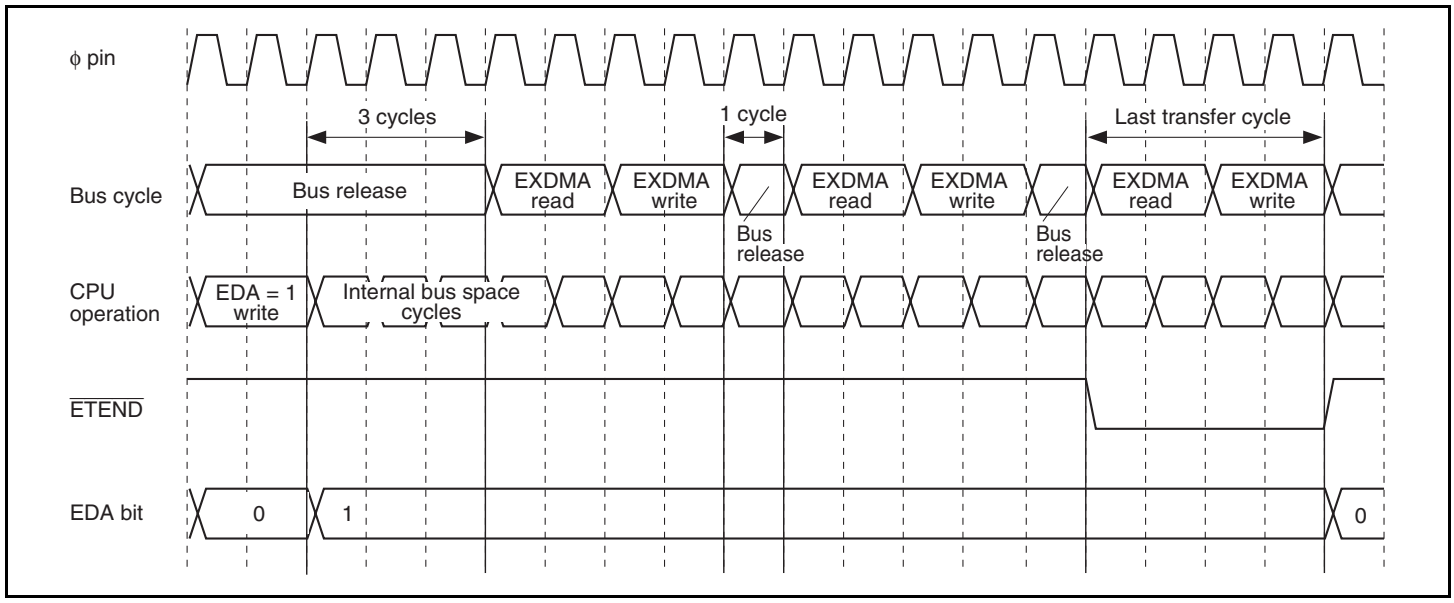


Figure 8.28 Auto Request/Cycle Steal Mode/Normal Transfer Mode (No Contention/Dual Address Mode)

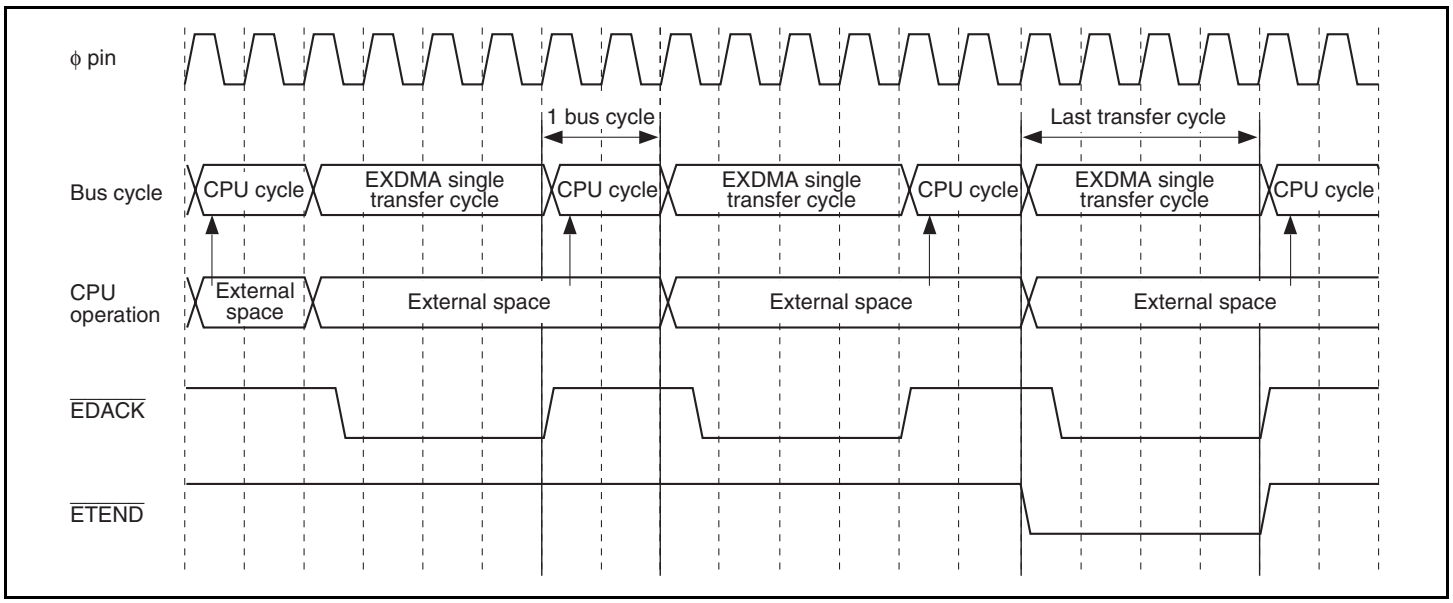


Figure 8.29 Auto Request/Cycle Steal Mode/Normal Transfer Mode (CPU Cycles/Single Address Mode)

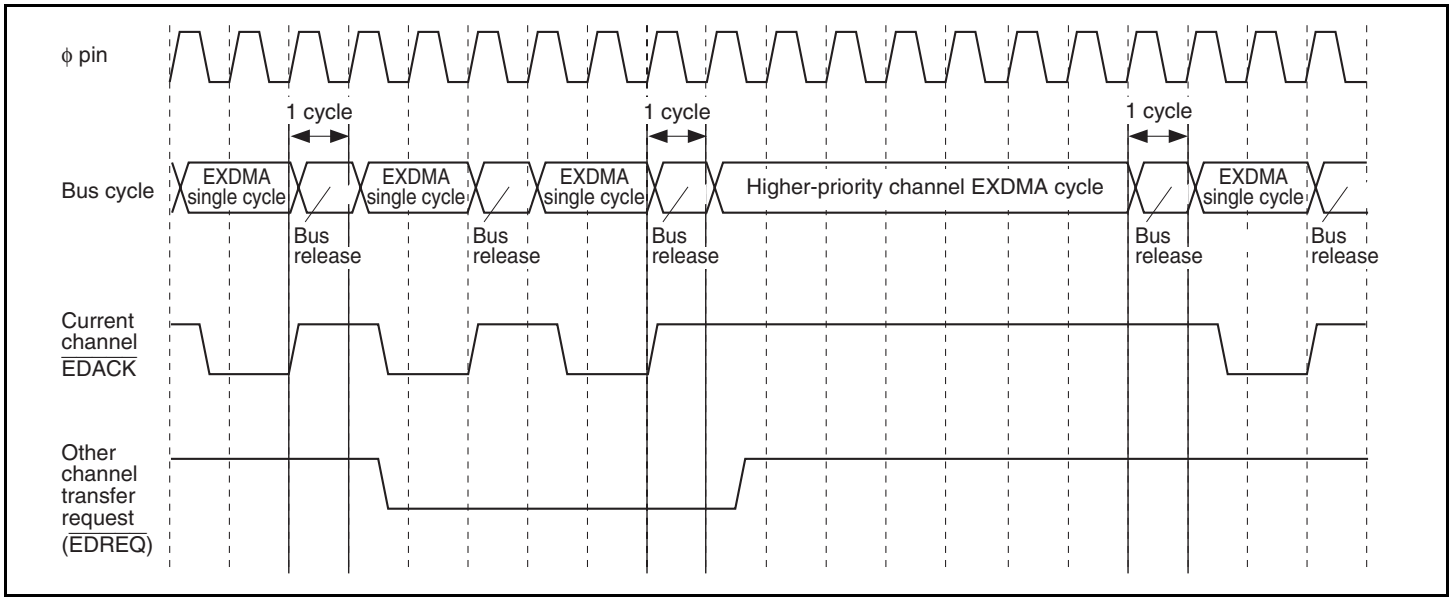


Figure 8.30 Auto Request/Cycle Steal Mode/Normal Transfer Mode (Contention with Another Channel/Single Address Mode)

(2) Auto Request/Burst Mode/Normal Transfer Mode

When the EDA bit is set to 1 in EDMDR, an EXDMA transfer cycle is started a minimum of three cycles later. Once transfer is started, it continues (as a burst) until the transfer end condition is satisfied.

If the BGUP bit is 1 in EDMDR, the bus is transferred in the event of a bus request from another bus master.

Transfer requests for other channels are held pending until the end of transfer on the current channel.

Figures 8.31 to 8.34 show operation timing examples for various conditions.

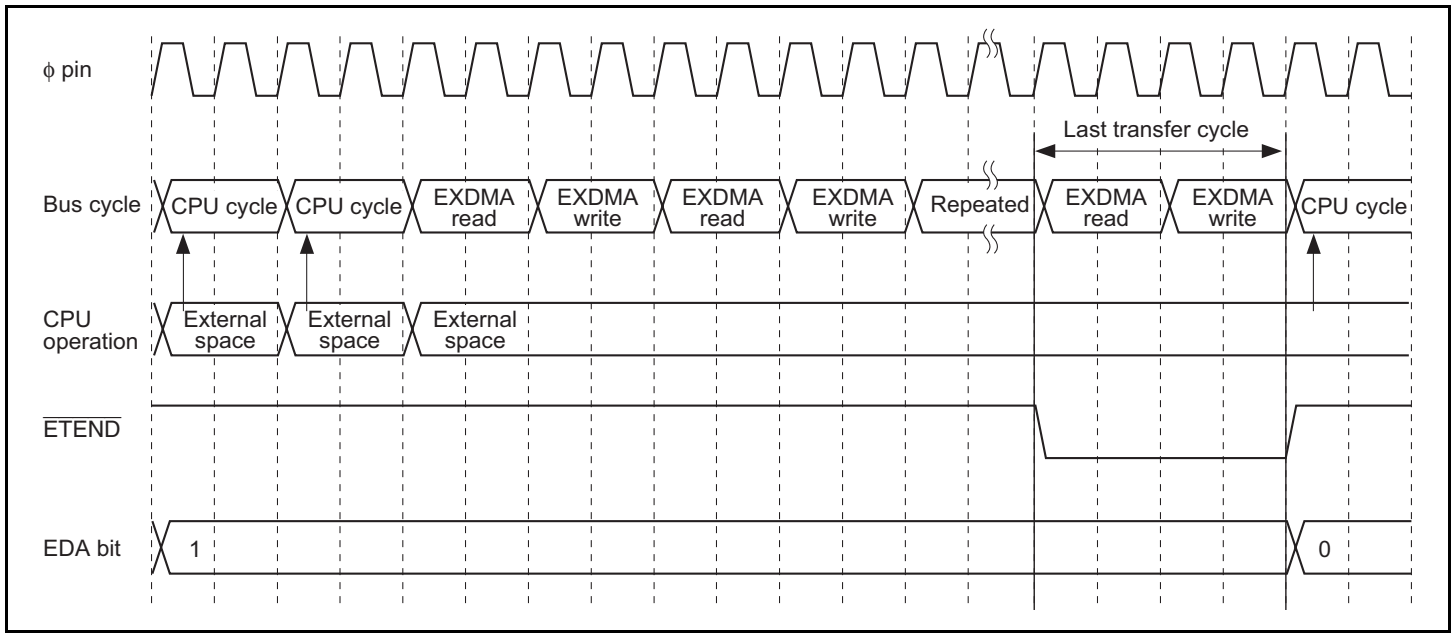


Figure 8.31 Auto Request/Burst Mode/Normal Transfer Mode (CPU Cycles/Dual Address Mode/BGUP = 0)

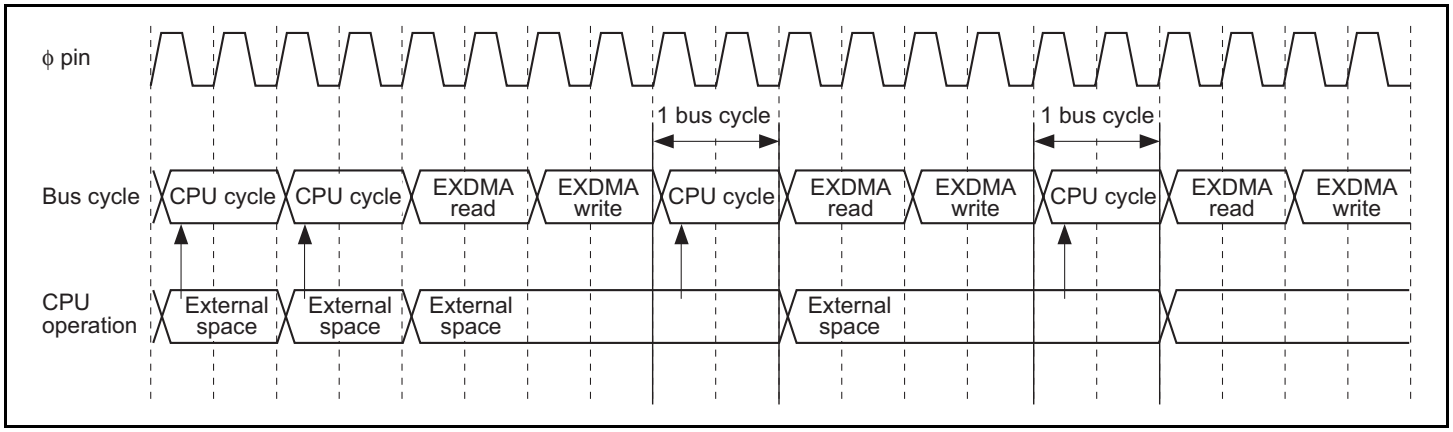


Figure 8.32 Auto Request/Burst Mode/Normal Transfer Mode (CPU Cycles/Dual Address Mode/BGUP = 1)

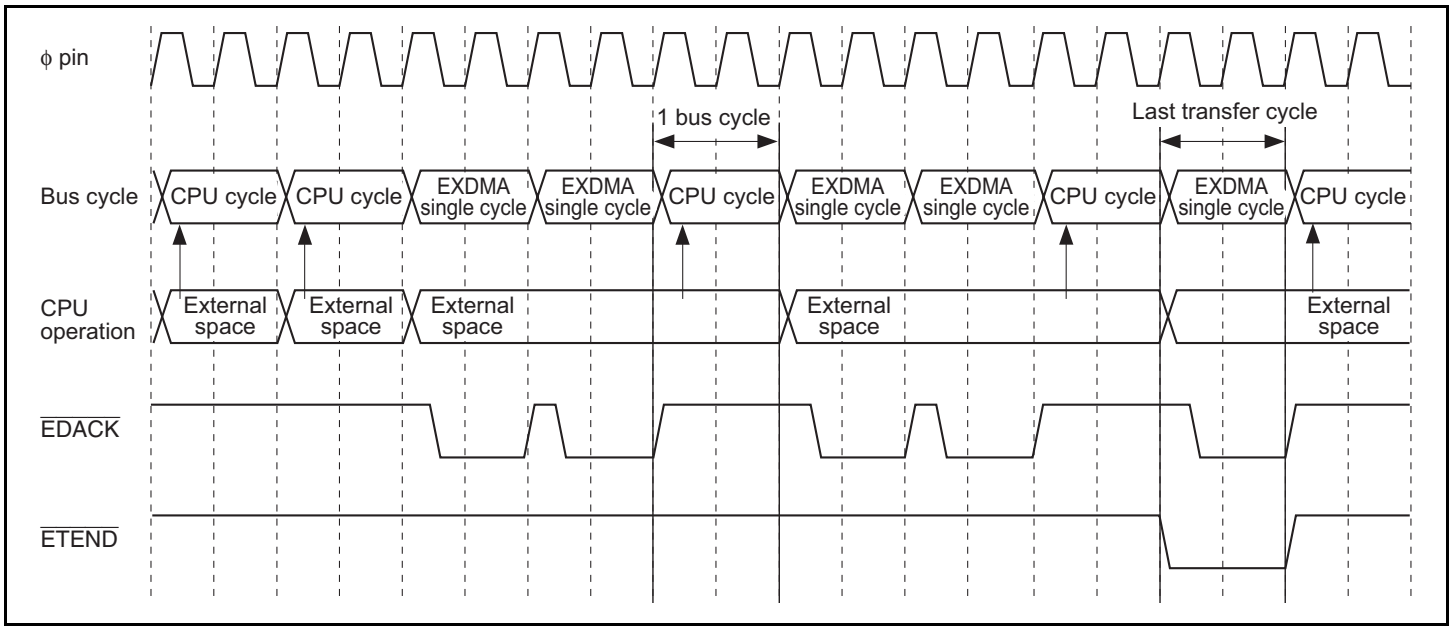


Figure 8.33 Auto Request/Burst Mode/Normal Transfer Mode (CPU Cycles/Single Address Mode/BGUP = 1)

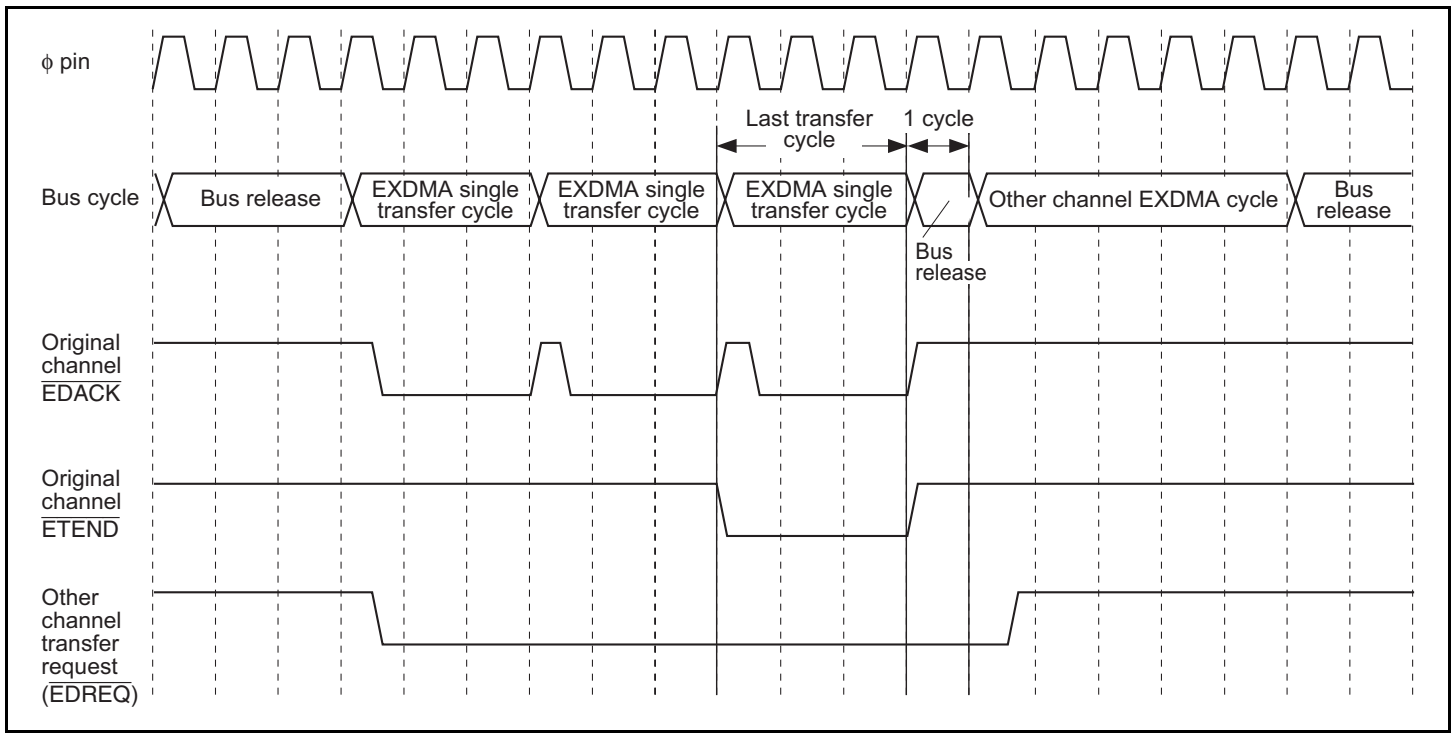


Figure 8.34 Auto Request/Burst Mode/Normal Transfer Mode (Contention with Another Channel/Single Address Mode)

(3) External Request/Cycle Steal Mode/Normal Transfer Mode

In external request mode, an EXDMA transfer cycle is started a minimum of three cycles after a transfer request is accepted. The next transfer request is accepted after the end of a one-transfer-unit EXDMA cycle. For external bus space CPU cycles, at least two bus cycles are generated before the next EXDMA cycle.

If a transfer request is generated for another channel, an EXDMA cycle for the other channel is generated before the next EXDMA cycle.

The $\overline{\text{EDREQ}}$ 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 8.35 to 8.38 show operation timing examples for various conditions.

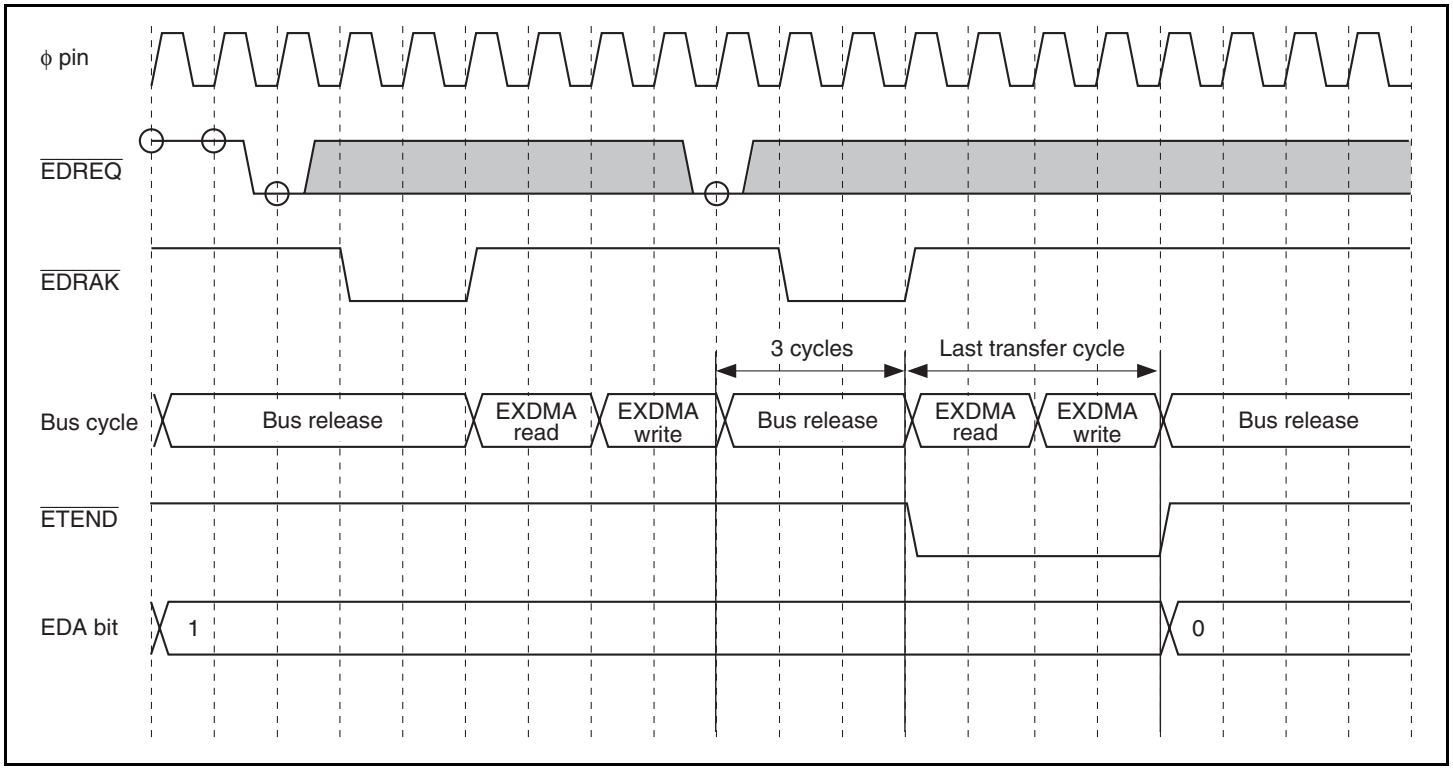


Figure 8.35 External Request/Cycle Steal Mode/Normal Transfer Mode (No Contention/Dual Address Mode/Low Level Sensing)

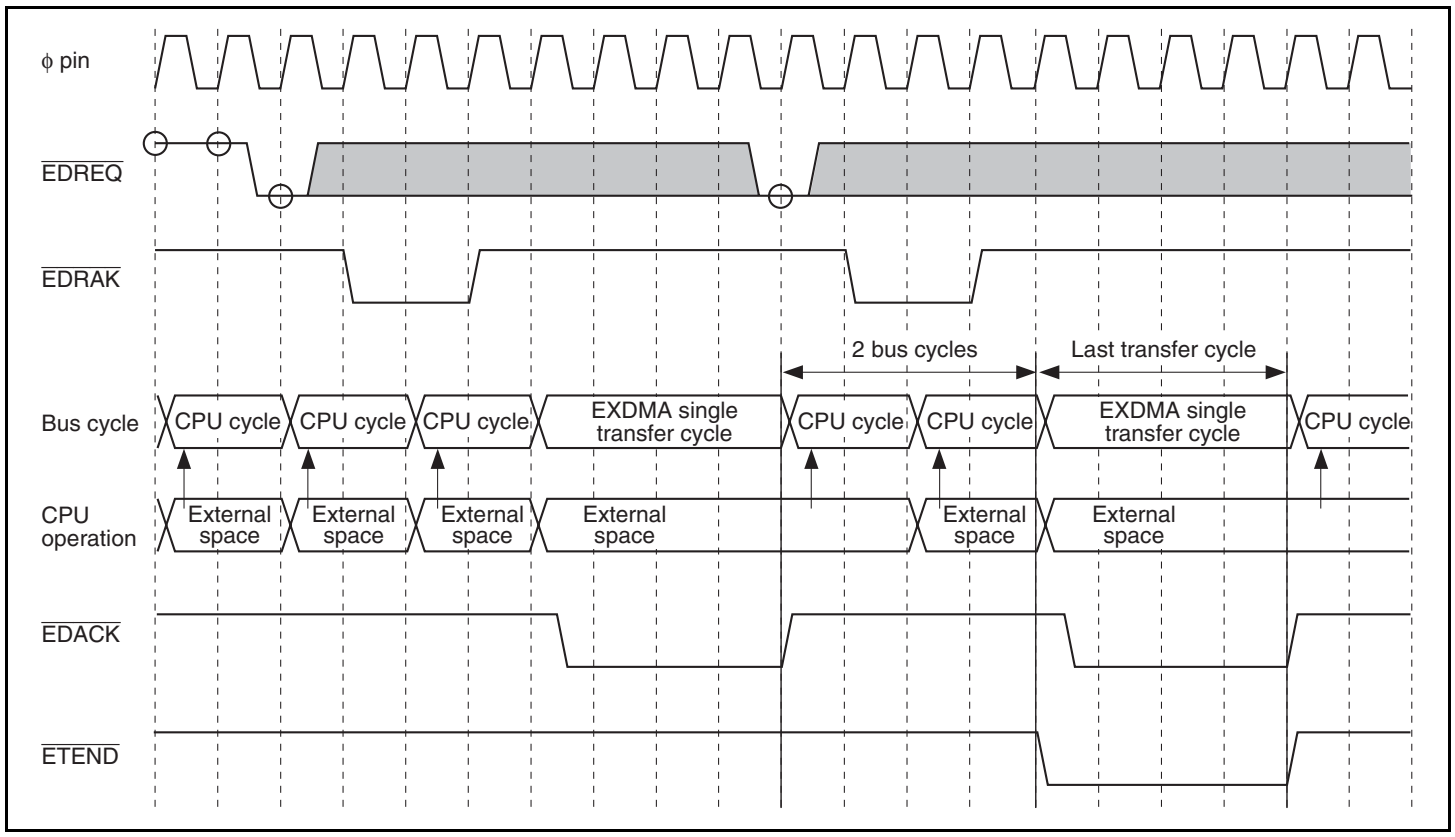


Figure 8.36 External Request/Cycle Steal Mode/Normal Transfer Mode (CPU Cycles/Single Address Mode/Low Level Sensing)

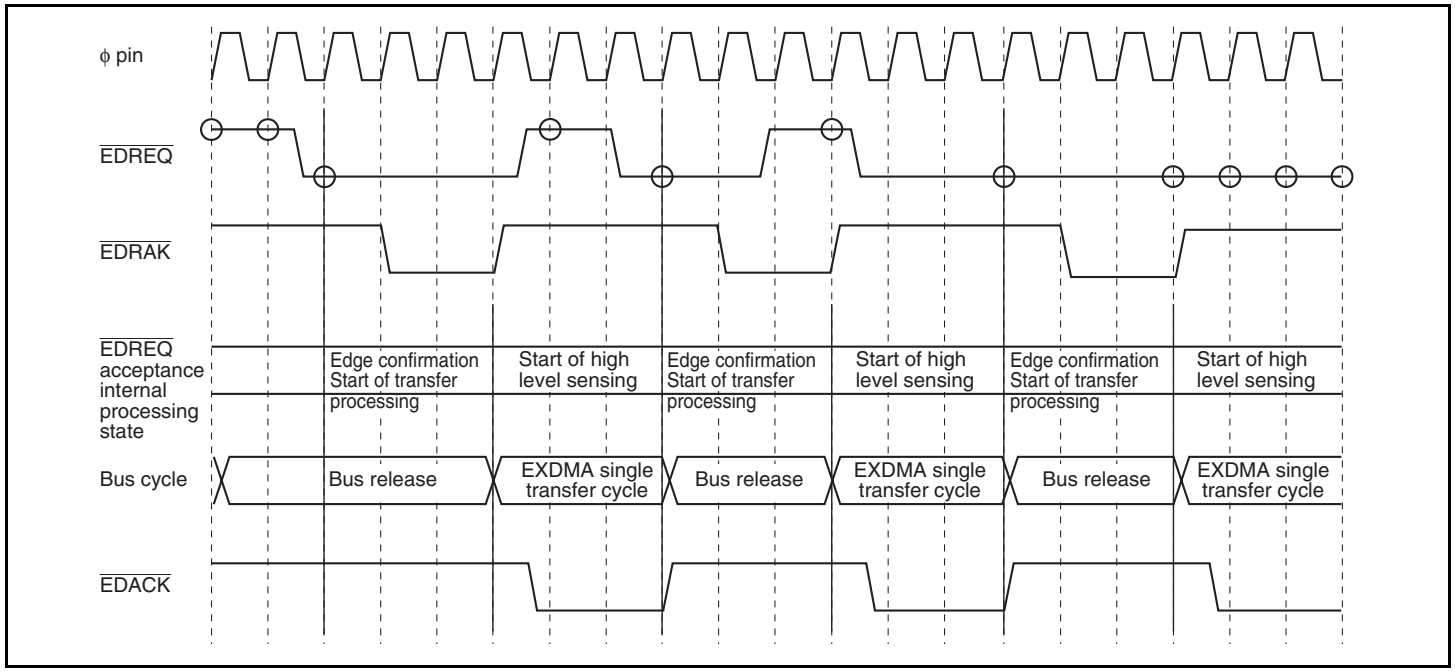


Figure 8.37 External Request/Cycle Steal Mode/Normal Transfer Mode (No Contention/Single Address Mode/Falling Edge Sensing)

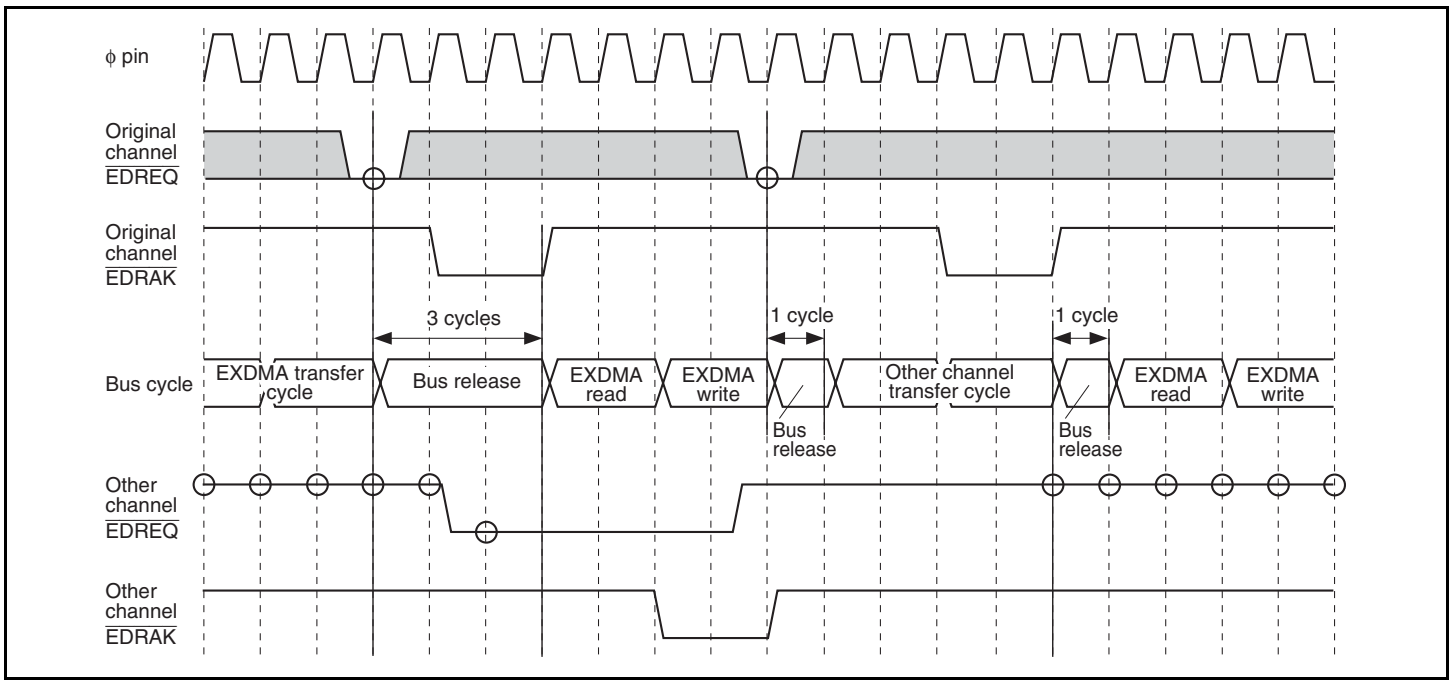


Figure 8.38 External Request/Cycle Steal Mode/Normal Transfer Mode Contention with Another Channel/Dual Address Mode/Low Level Sensing

(4) External Request/Cycle Steal Mode/Block Transfer Mode

In block transfer mode, transfer of one block is performed continuously in the same way as in burst mode. The timing of the start of the next block transfer is the same as in normal transfer mode.

If a transfer request is generated for another channel, an EXDMA cycle for the other channel is generated before the next block transfer.

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 8.39 to 8.44 show operation timing examples for various conditions.

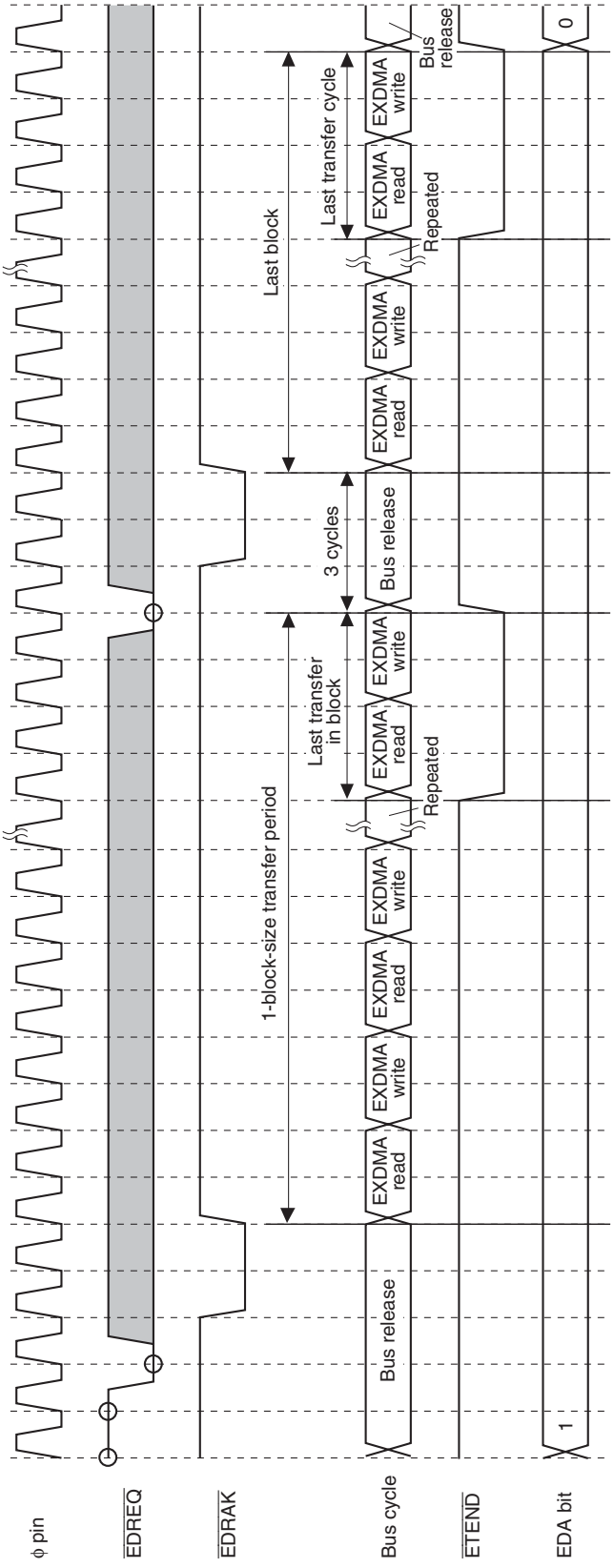


Figure 8.39 External Request/Cycle Steal Mode/Block Transfer Mode (No Contention/Dual Address Mode/Low Level Sensing/BGUP = 0)

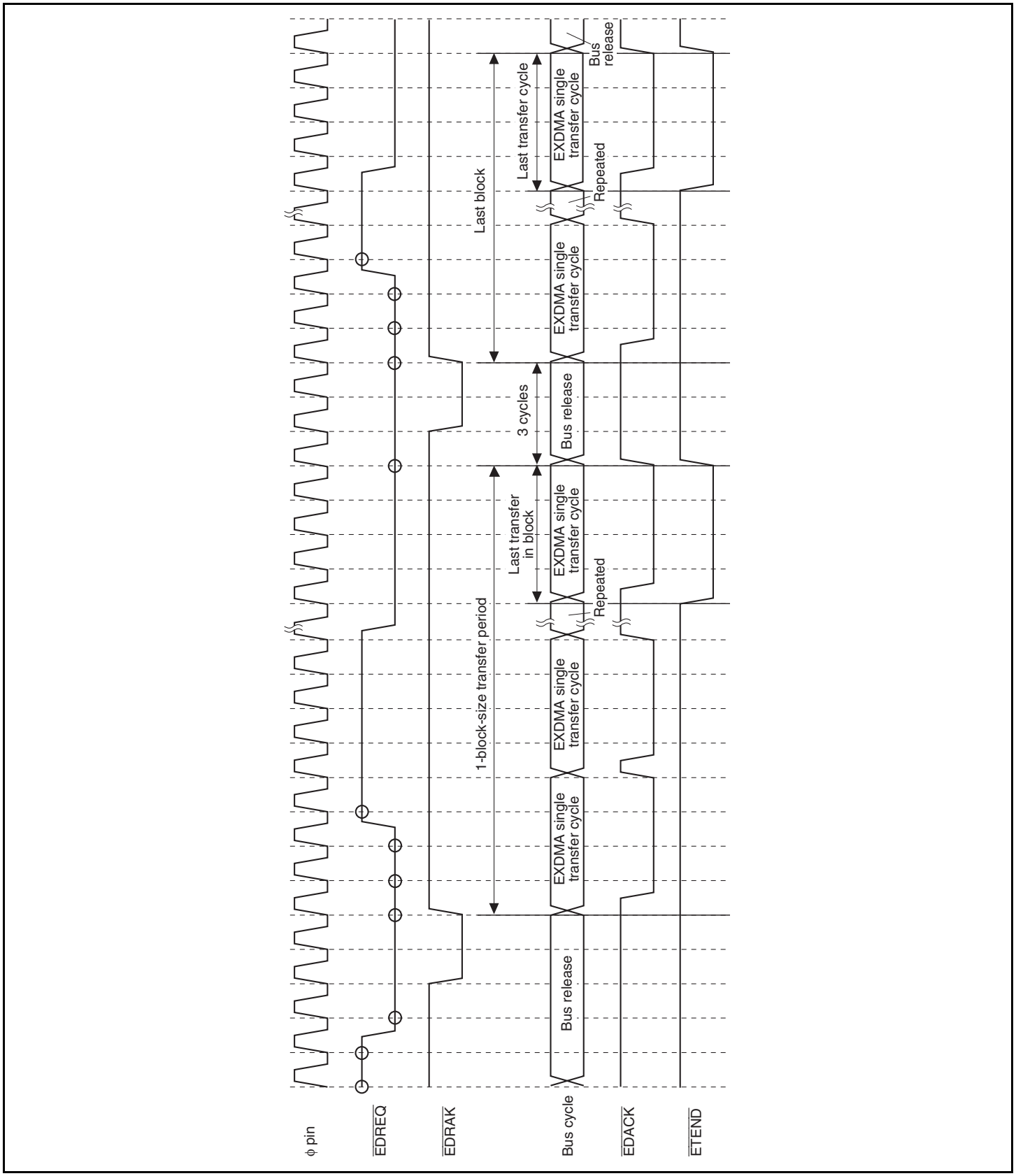


Figure 8.40 External Request/Cycle Steal Mode/Block Transfer Mode (No Contention/Single Address Mode/Falling Edge Sensing/BGUP = 0)

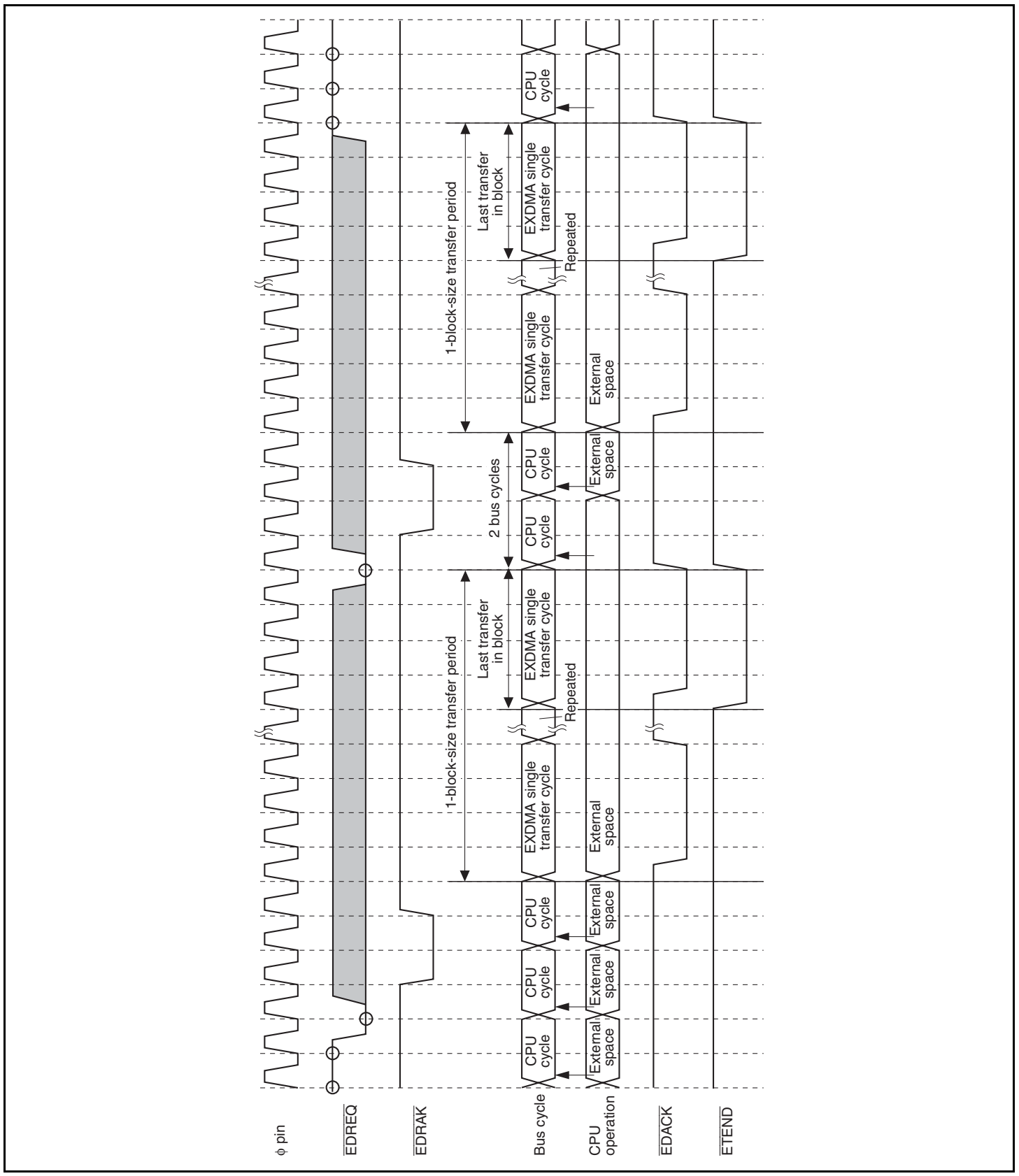


Figure 8.41 External Request/Cycle Steal Mode/Block Transfer Mode (CPU Cycles/Single Address Mode/Low Level Sensing/BGUP = 0)

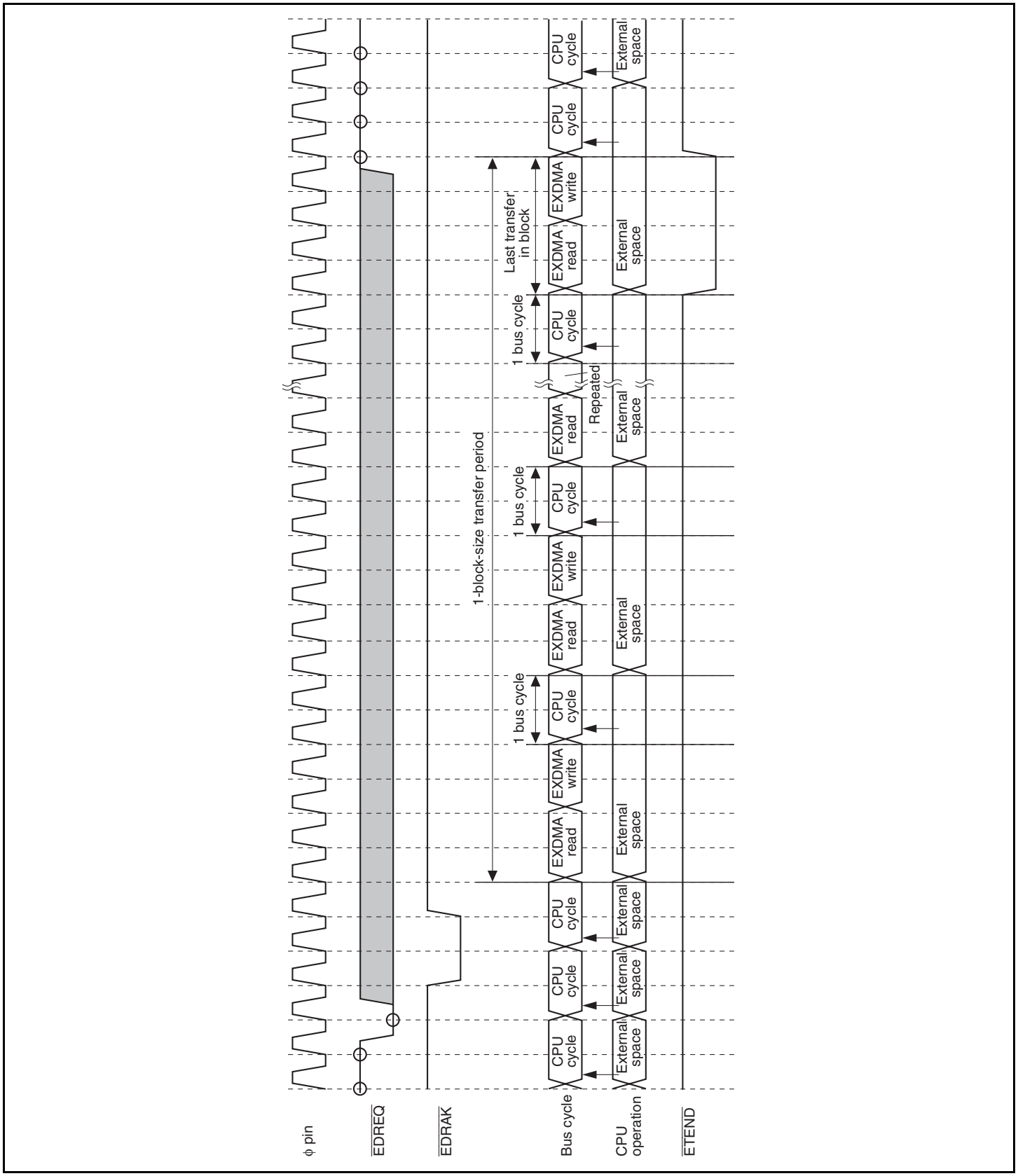


Figure 8.42 External Request/Cycle Steal Mode/Block Transfer Mode (CPU Cycles/Dual Address Mode/Low Level Sensing/BGUP = 1)

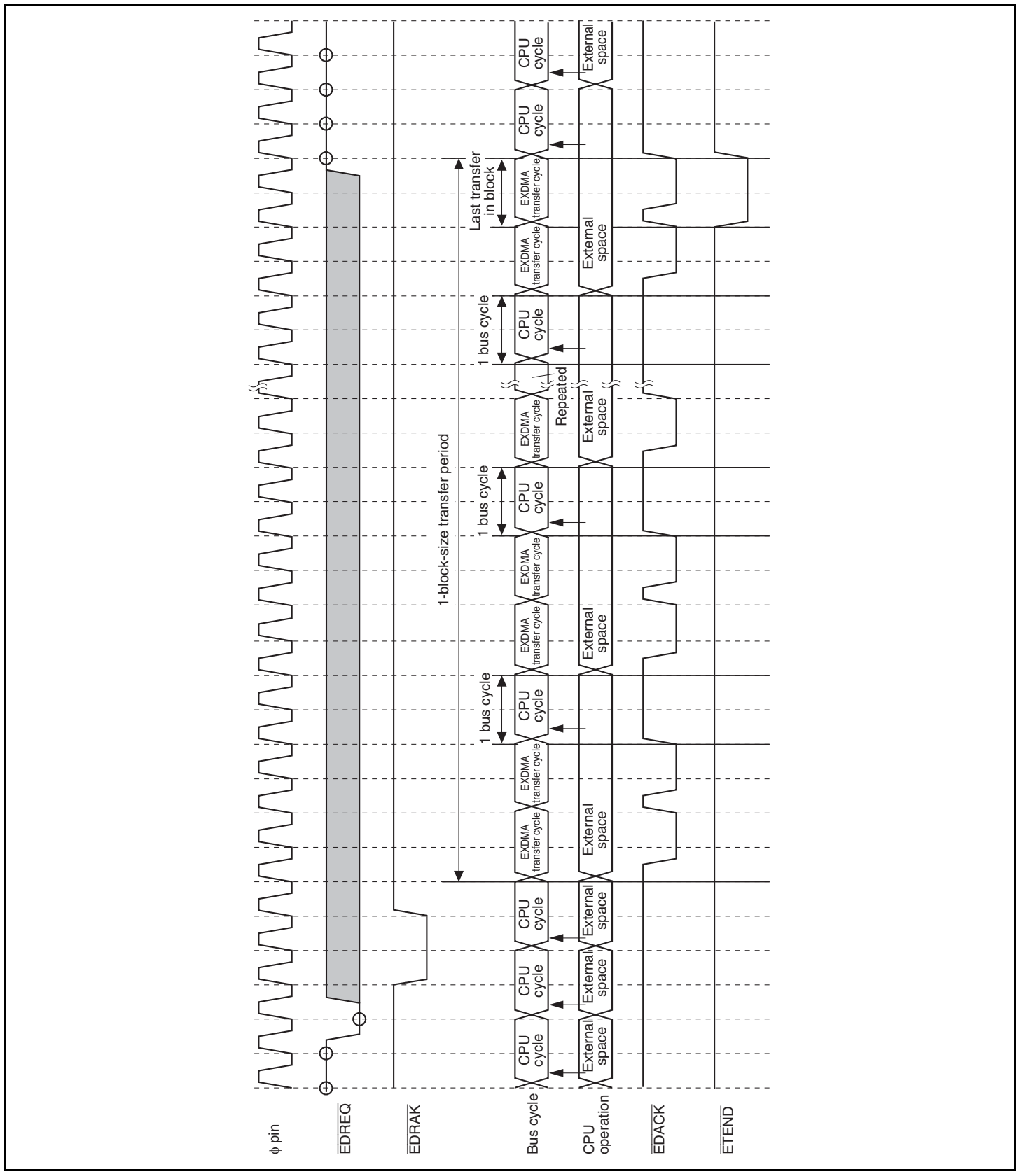


Figure 8.43 External Request/Cycle Steal Mode/Block Transfer Mode (CPU Cycles/Single Address Mode/Low Level Sensing/BGUP = 1)

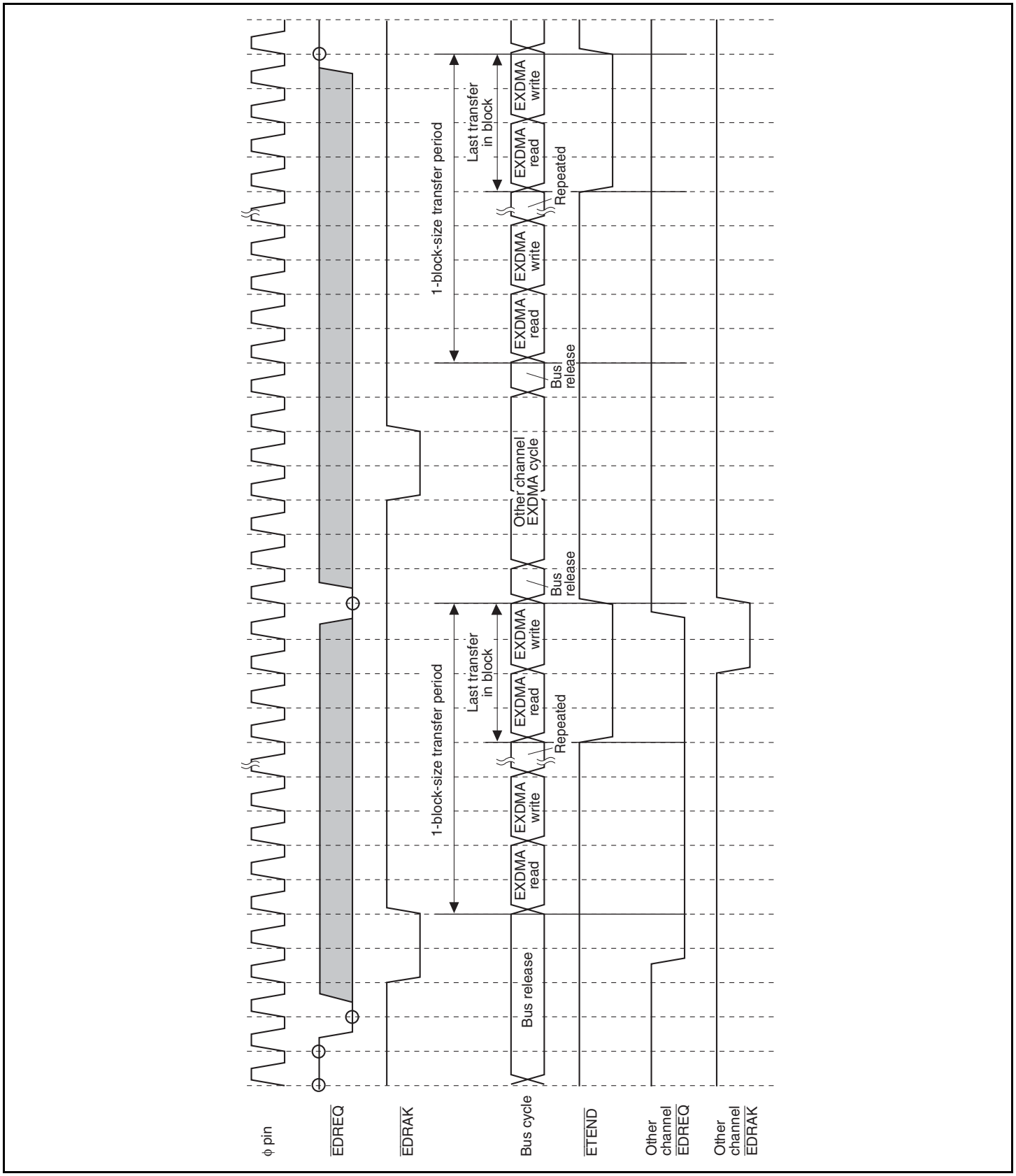


Figure 8.44 External Request/Cycle Steal Mode/Block Transfer Mode (Contention with Another Channel/Dual Address Mode/Low Level Sensing)

8.4.12 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the EDA bit in EDMDR changes from 1 to 0, indicating that DMA transfer has ended.

(1) Transfer End by 1 → 0 Transition of EDTCR

When the value of EDTCR changes from 1 to 0, DMA transfer ends on the corresponding channel and the EDA bit in EDMDR is cleared to 0. If the TCEIE bit in EDMDR is set at this time, a transfer end interrupt request is generated by the transfer counter and the IRF bit in EDMDR is set to 1.

In block transfer mode, DMA transfer ends when the value of bits 15 to 0 in EDTCR changes from 1 to 0.

DMA transfer does not end if the EDTCR value has been 0 since before the start of transfer.

(2) Transfer End by Repeat Area Overflow Interrupt

If an address overflows the repeat area when a repeat area specification has been made and repeat interrupts have been enabled (with the SARIE or DARIE bit in EDACR), a repeat area overflow interrupt is requested. DMA transfer ends, the EDA bit in EDMDR is cleared to 0, and the IRF bit in EDMDR is set to 1.

In dual address mode, if a repeat area overflow interrupt is requested during a read cycle, the following write cycle processing is still executed.

In block transfer mode, if a repeat area overflow interrupt is requested during transfer of a block, transfer continues to the end of the block. Transfer end by means of a repeat area overflow interrupt occurs between block-size transfers.

(3) Transfer End by 0-Write to EDA Bit in EDMDR

When 0 is written to the EDA bit in EDMDR by the CPU, etc., transfer ends after completion of the DMA cycle in which transfer is in progress or a transfer request was accepted.

In block transfer mode, DMA transfer halts after completion of one-block-size transfer.

The EDA bit in EDMDR is not cleared to 0 until all transfer processing has ended. Up to that point, the value of the EDA bit will be read as 1.

(4) Transfer Abort by NMI Interrupt

DMA transfer is aborted when an NMI interrupt is generated. The EDA bit is cleared to 0 in all channels. In external request mode, DMA transfer is performed for all transfer requests for which $\overline{\text{EDRAK}}$ has been output. In dual address mode, processing is executed for the write cycle following the read cycle.

In block transfer mode, operation is aborted even in the middle of a block-size transfer. As the transfer is halted midway through a block, the BEF bit in EDMDR is set to 1 to indicate that the block transfer was not carried out normally.

When transfer is aborted, register values are retained, and as the address registers indicate the next transfer addresses, transfer can be resumed by setting the EDA bit to 1 in EDMDR. If the BEF bit is 1 in EDMDR, transfer can be resumed from midway through a block.

(5) Hardware Standby Mode and Reset Input

The EXDMAC is initialized in hardware standby mode and by a reset. DMA transfer is not guaranteed in these cases.

8.4.13 Relationship between EXDMAC and Other Bus Masters

The read and write operations in a DMA transfer cycle are indivisible, and a refresh cycle, external bus release cycle, or internal bus master (CPU, DTC, or DMAC) external space access cycle never occurs between the two.

When read and write cycles occur consecutively, as in burst transfer or block transfer, a refresh or external bus release state may be inserted after the write cycle. As the internal bus masters are of lower priority than the EXDMAC, external space accesses by internal bus masters are not executed until the EXDMAC releases the bus.

The EXDMAC releases the bus in the following cases:

1. When DMA transfer is performed in cycle steal mode
2. When switching to a different channel
3. When transfer ends in burst transfer mode
4. When transfer of one block ends in block transfer mode
5. When burst transfer or block transfer is performed with the BGUP bit in EDMDR set to 1 (however, the bus is not released between read and write cycles)

8.5 Interrupt Sources

EXDMAC interrupt sources are a transfer end indicated by the transfer counter, and repeat area overflow interrupts. Table 8.4 shows the interrupt sources and their priority order.

Table 8.4 Interrupt Sources and Priority Order

Interrupt	Interrupt source	Interrupt Priority
EXDMTEND2	Transfer end indicated by channel 2 transfer counter	High
	Channel 2 source address repeat area overflow	
	Channel 2 destination address repeat area overflow	
EXDMTEND3	Transfer end indicated by channel 3 transfer counter	
	Channel 3 source address repeat area overflow	
	Channel 3 destination address repeat area overflow	Low

Interrupt sources can be enabled or disabled by means of the EDIE bit in EDMDR for the relevant channel, and can be sent to the interrupt controller independently. The relative priority order of the channels is determined by the interrupt controller (see table 8.4).

Figure 8.45 shows the transfer end interrupt logic. A transfer end interrupt is generated whenever the EDIE bit is set to 1 while the IRF bit is set to 1 in EDMDR.

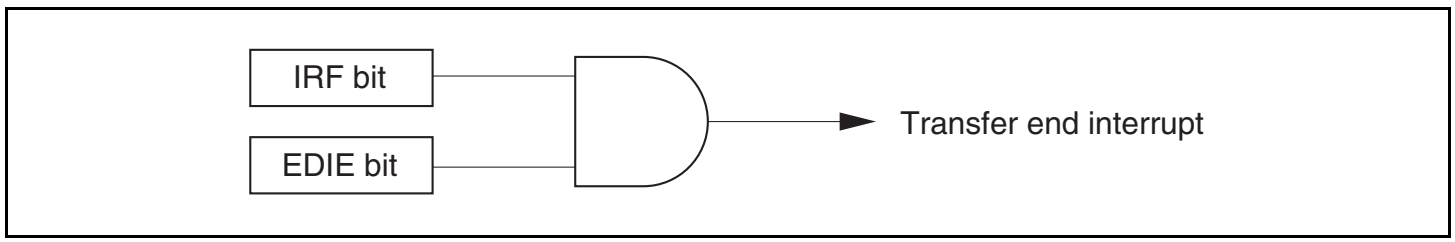
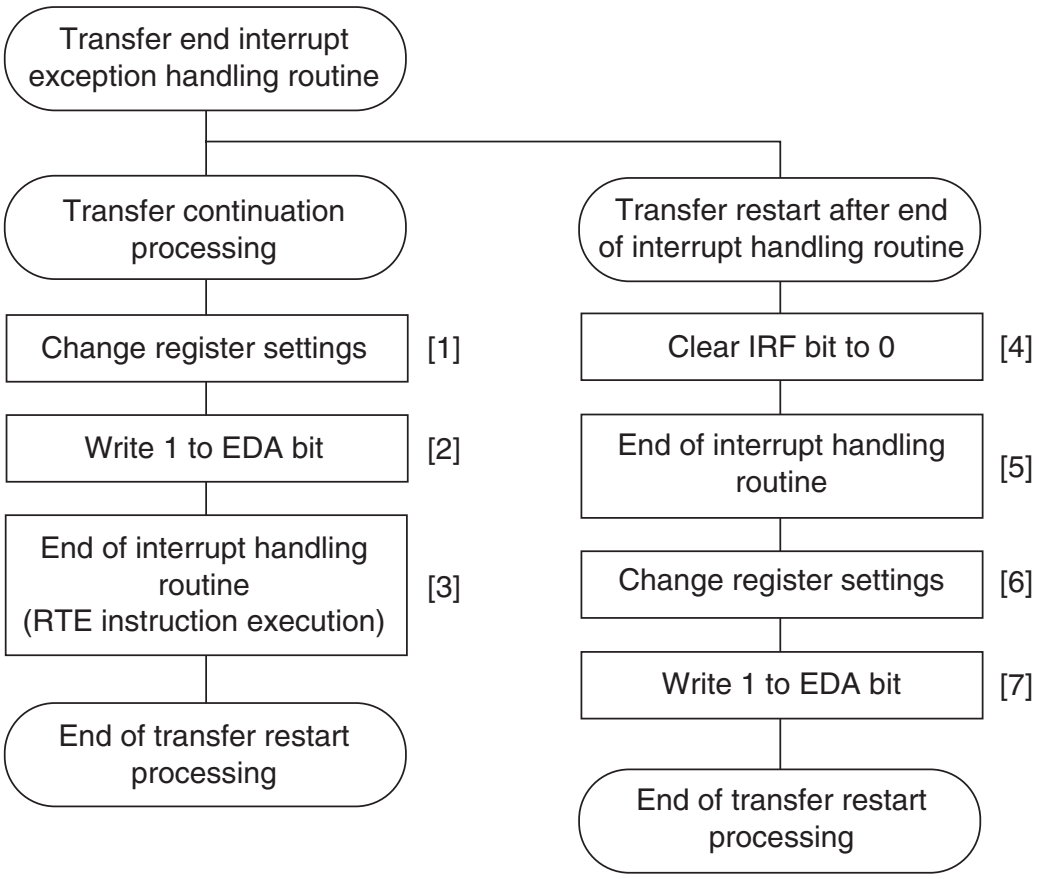


Figure 8.45 Transfer End Interrupt Logic

Interrupt source settings are made individually with the interrupt enable bits in the registers for the relevant channels. The transfer counter's transfer end interrupt is enabled or disabled by means of the TCEIE bit in EDMDR, the source address register repeat area overflow interrupt by means of the SARIE bit in EDACR, and the destination address register repeat area overflow interrupt by means of the DARIE bit in EDACR. When an interrupt source occurs while the corresponding interrupt enable bit is set to 1, the IRF bit in EDMDR is set to 1. The IRF bit is set by all interrupt sources indiscriminately.

The transfer end interrupt can be cleared either by clearing the IRF bit to 0 in EDMDR within the interrupt handling routine, or by re-setting the transfer counter and address registers and then setting the EDA bit to 1 in EDMDR to perform transfer continuation processing. An example of the procedure for clearing the transfer end interrupt and restarting transfer is shown in figure 8.46.



[1] Write set values to the registers (transfer counter, address registers, etc.).

[2] Write 1 to the EDA bit in EDMDR to restart EXDMA operation. When 1 is written to the EDA bit, the IRF bit in EDMDR is automatically cleared to 0 and the interrupt source is cleared.

[3] The interrupt handling routine is ended with an RTE instruction, etc.

[4] Clear the IRF bit to 0 in EDMDR by first reading 1 from it, then writing 0.

[5] After the interrupt handling routine is ended with an RTE instruction, etc., interrupt masking is cleared.

[6] Write set values to the registers (transfer counter, address registers, etc.).

[7] Write 1 to the EDA bit in EDMDR to restart EXDMA operation.

Figure 8.46 Example of Procedure for Restarting Transfer on Channel in which Transfer End Interrupt Occurred

8.6 Usage Notes

(1) EXDMAC Register Access during Operation

Except for clearing the EDA bit to 0 in EDMDR, settings should not be changed for a channel in operation (including the transfer standby state). Transfer must be disabled before changing a setting for an operational channel.

(2) Module Stop State

When the MSTP14 bit is set to 1 in MSTPCRH, the EXDMAC clock stops and the EXDMAC enters the module stop state. However, 1 cannot be written to the MSTP14 bit when any of the EXDMAC's channels is enabled for transfer, or when an interrupt is being requested. Before setting the MSTP14 bit, first clear the EDA bit in EDMDR to 0, then clear the IRF or EDIE bit in EDMDR to 0.

When the EXDMAC clock stops, EXDMAC registers can no longer be accessed. The following EXDMAC register settings remain valid in the module stop state, and so should be changed, if necessary, before making the module stop transition.

- ETENDE = 1 in EDMDR ($\overline{\text{ETEND}}$ pin enable)
- EDRAKE = 1 in EDMDR ($\overline{\text{EDRAK}}$ pin enable)
- AMS = 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 goes to [2].

[2] Transfer standby state: Waits for EXDMAC data transfer to become possible, then goes to [3].

[3] Activation request disabled state: Waits for high level sensing on $\overline{\text{EDREQ}}$ pin, then goes to [1].

After EXDMAC transfer is enabled, the EXDMAC goes to state [1], so low level sensing is used for the initial activation after transfer is enabled.

(4) Activation Source Acceptance

At the start of activation source acceptance, low level sensing is used for both falling edge sensing and low level sensing on the $\overline{\text{EDREQ}}$ pin. Therefore, a request is accepted in the case of a low level at the $\overline{\text{EDREQ}}$ pin that occurs before execution of the EDMDR write for setting the transfer-enabled state.

When the EXDMAC is activated, make sure, if necessary, that a low level does not remain at the $\overline{\text{EDREQ}}$ pin from the previous end of transfer, etc.

(5) Enabling Interrupt Requests when IRF = 1 in EDMDR

When transfer is started while the IRF bit is set to 1 in EDMDR, if the EDIE bit is set to 1 in EDMDR together with the EDA bit in EDMDR, enabling interrupt requests, an interrupt will be requested since EDIE = 1 and IRF = 1. To prevent the occurrence of an erroneous interrupt request when transfer starts, ensure that the IRF bit is cleared to 0 before the EDIE bit is set to 1.

(6) $\overline{\text{ETEND}}$ Pin and CBR Refresh Cycle

If the last EXDMAC transfer cycle and a CBR refresh cycle occur simultaneously, note that although the CBR refresh and the last transfer cycle may be executed consecutively, $\overline{\text{ETEND}}$ may also go low in this case for the refresh cycle.

Section 9 Data Transfer Controller (DTC)

This LSI includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

Figure 9.1 shows a block diagram of the DTC.

9.1 Features

- Transfer possible over any number of channels
- Three transfer modes

1. Normal mode
   One operation transfers one byte or one word of data.
   Memory address is incremented or decremented by 1 or 2.
   From 1 to 65,536 transfers can be specified.
2. Repeat mode
   One operation transfers one byte or one word of data.
   Memory address is incremented or decremented by 1 or 2.
   Once the specified number of transfers (1 to 256) has ended, the initial state is restored, and transfer is repeated.
3. Block transfer mode
   One operation transfers one block of data.
   The block size is 1 to 256 bytes or words.
   From 1 to 65,536 transfers can be specified.
   Either the transfer source or the transfer destination is designated as a block area.

- One activation source can trigger a number of data transfers (chain transfer)
- Direct specification of 16-Mbyte address space possible
- Activation by software is possible
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
- Module stop mode can be set

The DTC's register information is stored in the on-chip RAM. When the DTC is used, the RAME bit in SYSCR must be set to 1. A 32-bit bus connects the DTC to the on-chip RAM (1 Kbyte), enabling 32-bit/1-state reading and writing of the DTC register information.

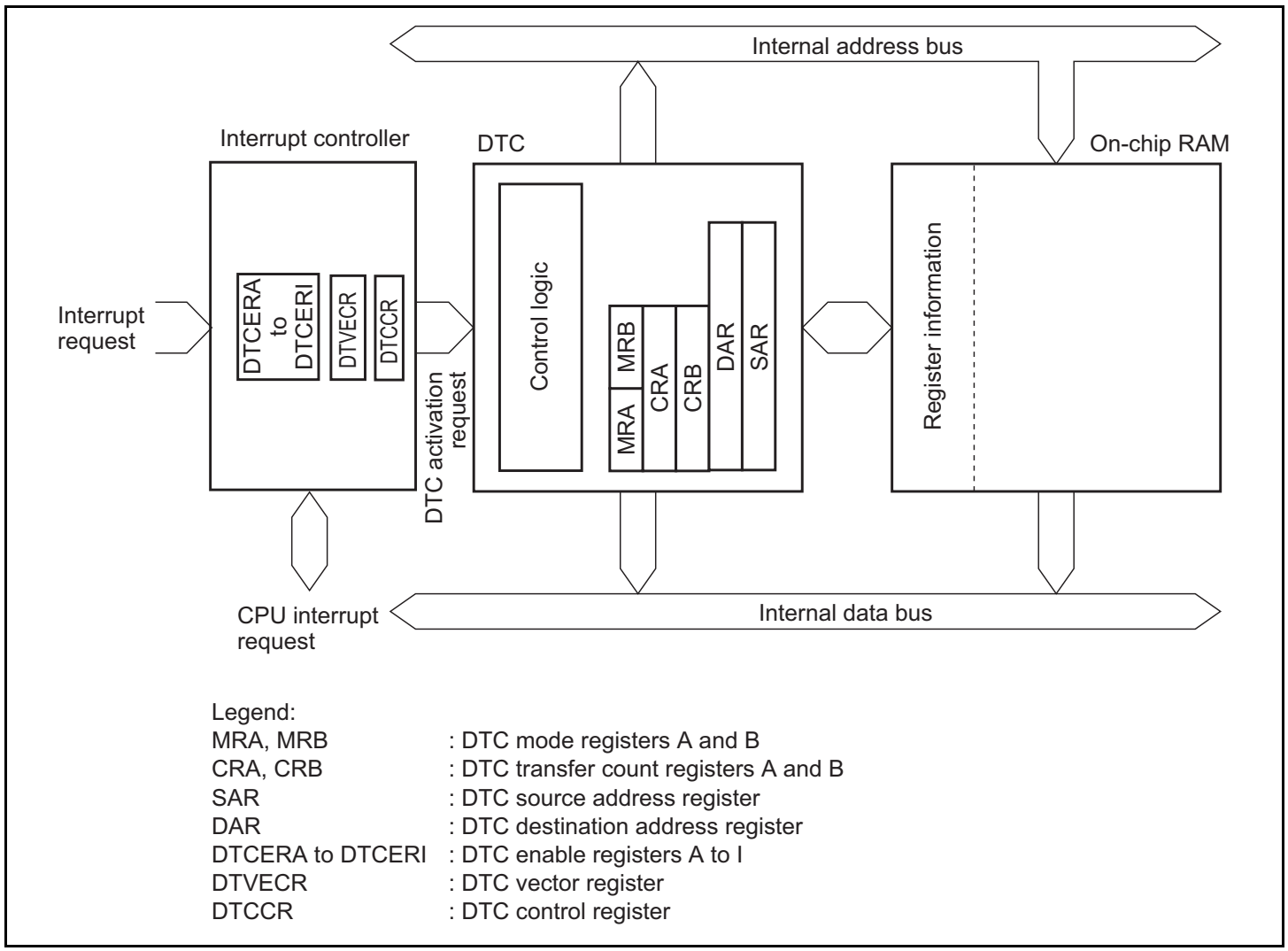


Figure 9.1 Block Diagram of DTC

9.2 Register Descriptions

DTC has the following registers.

- DTC mode register A (MRA)
- DTC mode register B (MRB)
- DTC source address register (SAR)
- DTC destination address register (DAR)
- DTC transfer count register A (CRA)
- DTC transfer count register B (CRB)

These six registers cannot be directly accessed from the CPU. When activated, the DTC reads a set of register information that is stored in an on-chip RAM to the corresponding DTC registers and transfers data. After the data transfer, it writes a set of updated register information back to the RAM.

- DTC enable registers A to I (DTCERA to DTCERI)
- DTC vector register (DTVECR)
- DTC control register (DTCCR)

9.2.1 DTC Mode Register A (MRA)

MRA selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	SM1	Undefined	—	Source Address Mode 1 and 0
6	SM0	Undefined	—	These bits specify an SAR operation after a data transfer. 0×: SAR is fixed 10: SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Bit	Bit Name	Initial Value	R/W	Description
5	DM1	Undefined	—	Destination Address Mode 1 and 0
4	DM0	Undefined	—	These bits specify a DAR operation after a data transfer. 0×: DAR is fixed 10: DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: DAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)
3	MD1	Undefined	—	DTC Mode
2	MD0	Undefined	—	These bits specify the DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: Setting prohibited
1	DTS	Undefined	—	DTC Transfer Mode Select Specifies whether the source side or the destination side is set to be a repeat area or block area, in repeat mode or block transfer mode. 0: Destination side is repeat area or block area 1: Source side is repeat area or block area
0	Sz	Undefined	—	DTC Data Transfer Size Specifies the size of data to be transferred. 0: Byte-size transfer 1: Word-size transfer

Legend:

× : Don't care

9.2.2 DTC Mode Register B (MRB)

MRB selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable When this bit is set to 1, a chain transfer will be performed. For details, refer to section 9.5.4, Chain Transfer. In data transfer with CHNE set to 1, determination of the end of the specified number of transfers, clearing of the activation source flag, and clearing of DTCER is not performed.
6	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, a CPU interrupt request is generated every time after a data transfer ends. When this bit is set to 0, a CPU interrupt request is generated at the time when the specified number of data transfer ends.
5	CHNS	Undefined	—	DTC Chain Transfer Select Specifies the chain transfer condition. 0: Chain transfer every time 1: Chain transfer only when transfer counter = 0
4 to 0	—	Undefined	—	Reserved These bits have no effect on DTC operation, and should always be written with 0.

9.2.3 DTC Source Address Register (SAR)

SAR is a 24-bit register that designates the source address of data to be transferred by the DTC. For word-size transfer, specify an even source address.

9.2.4 DTC Destination Address Register (DAR)

DAR is a 24-bit register that designates the destination address of data to be transferred by the DTC. For word-size transfer, specify an even destination address.

9.2.5 DTC Transfer Count Register A (CRA)

CRA is a 16-bit register that designates the number of times data is to be transferred by the DTC.

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65,536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

In repeat mode or block transfer mode, the CRA is divided into two parts: the upper 8 bits (CRAH) and the lower 8 bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and the contents of CRAH are sent when the count reaches H'00.

9.2.6 DTC Transfer Count Register B (CRB)

CRB is a 16-bit register that designates the number of times data is to be transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65,536) that is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000. The CRB is not available in normal and repeat modes.

9.2.7 DTC Enable Registers A to I (DTCERA to DTCERI)

DTCER which is comprised of registers, DTCERA to DTCERI, is a register that specifies DTC activation interrupt sources. The correspondence between interrupt sources and DTCE bits is shown in table 9.2. For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for reading and writing. If all interrupts are masked, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

Bit	Bit Name	Initial Value	R/W	Description
7	DTCE7	0	R/W	DTC Activation Enable
6	DTCE6	0	R/W	Setting this bit to 1 specifies a relevant interrupt source to a DTC activation source. [Clearing conditions] • When the DISEL bit is 1 and the data transfer has ended • When the specified number of transfers have ended These bits are not automatically cleared when the DISEL bit is 0 and the specified number of transfers have not ended • When 0 is written to DTCE after reading DTCE = 1
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	

9.2.8 DTC Vector Register (DTVECR)

DTVECR sets a vector number for the software activation interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	DTVEC7	0	R/W	DTC Software Activation Vectors 7 to 0
6	DTVEC6	0	R/W	These bits specify a vector number for DTC software activation. The vector address is expressed as H'0400 + (vector number × 2). For example, when DTVEC7 to DTVEC0 = H'10, the vector address is H'0420. These bits can be written to only when the SWDTE bit is 0.
5	DTVEC5	0	R/W	
4	DTVEC4	0	R/W	
3	DTVEC3	0	R/W	
2	DTVEC2	0	R/W	
1	DTVEC1	0	R/W	
0	DTVEC0	0	R/W	

9.2.9 DTC Control Register (DTCCR)

DTCCR enables or disables DTC activation by software.

Bit	Bit Name	Initial Value	R/W	Description
7	SWDTE	0	R/W	DTC Software Activation Enable Setting this bit to 1 activates the DTC. Only 1 can be written to this bit. [Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers have not ended • When 0 is written to the DISEL bit after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU. When the DISEL bit is 1 and data transfer has ended or when the specified number of transfers have ended, this bit will not be cleared.
6 to 0	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

9.3 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTVECR or DTCCR by software. An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTCER bit. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source or corresponding DTCER bit is cleared. The activation source flag, in the case of RXI0, for example, is the RDRF flag of SCI_0.

When an interrupt has been designated a DTC activation source, existing CPU mask level and interrupt controller priorities have no effect. If there is more than one activation source at the same time, the DTC operates in accordance with the default priorities.

Table 9.1 shows a relationship between activation sources and DTCER clear conditions. Figure 9.2 shows a block diagram of activation source control. For details see section 5, Interrupt Controller.

Table 9.1 Relationship between Activation Sources and DTCER Clearing

Activation Source	DISEL = 0 and Specified Number of Transfers Has Not Ended	DISEL = 1 or Specified Number of Transfers Has Ended
Activation by software	SWDTE bit is cleared to 0	• SWDTE bit remains set to 1 • Interrupt request to CPU
Activation by an interrupt	• Corresponding DTCER bit remains set to 1. • Activation source flag is cleared to 0.	• Corresponding DTCER bit is cleared to 0. • Activation source flag remains set to 1. • Interrupt that became the activation source is requested to the CPU.

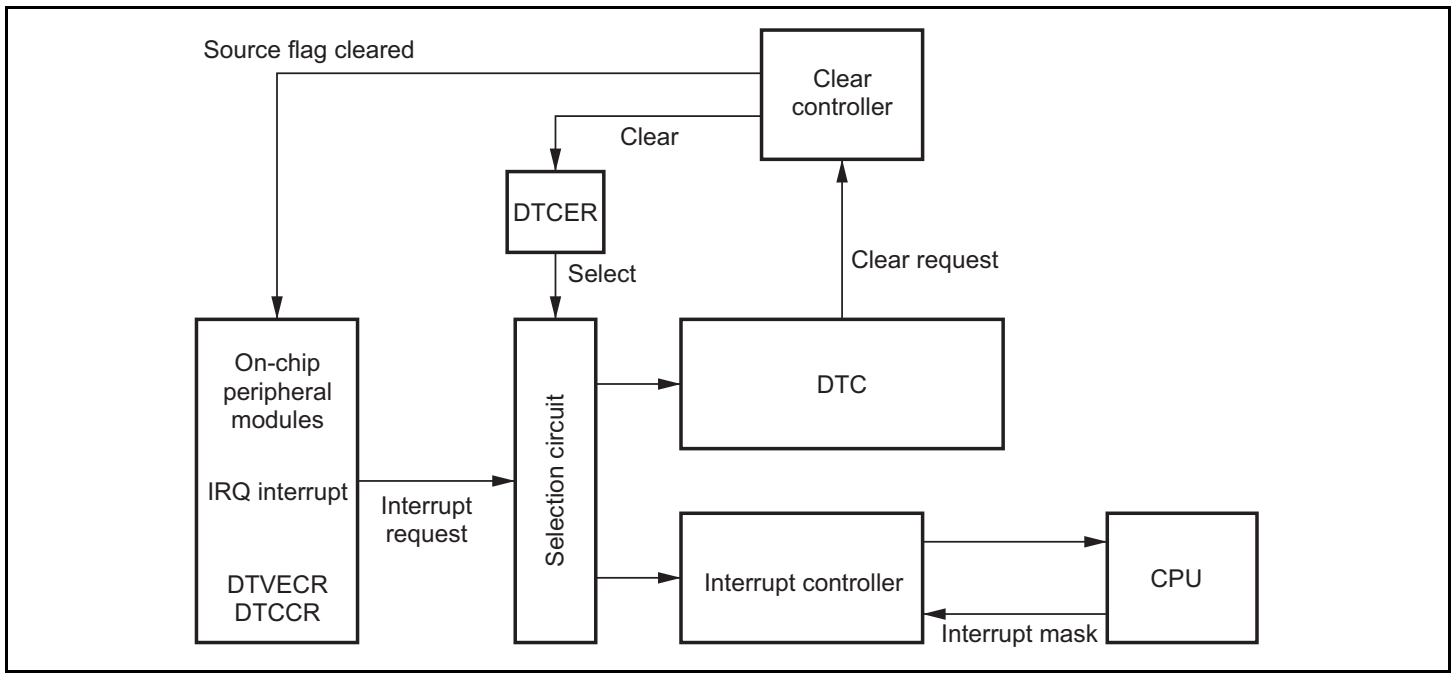


Figure 9.2 Block Diagram of DTC Activation Source Control

9.4 Location of Register Information and DTC Vector Table

Locate the register information in the on-chip RAM (addresses: H'FFBC00 to H'FFBFFF). Register information should be located at the address that is multiple of four within the range. Locating the register information in address space is shown in figure 9.3. Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information. In the case of chain transfer, register information should be located in consecutive areas as shown in figure 9.3 and the register information start address should be located at the corresponding vector address to the activation source. Figure 9.4 shows correspondences between the DTC vector address and register information. The DTC reads the start address of the register information from the vector address set for each activation source, and then reads the register information from that start address.

When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[7:0] × 2). For example, if DTVECR is H'10, the vector address is H'0420.

The configuration of the vector address is the same in both normal* and advanced modes, a 2-byte unit being used in both cases. These two bytes specify the lower bits of the register information start address.

Note: * Not available in this LSI.

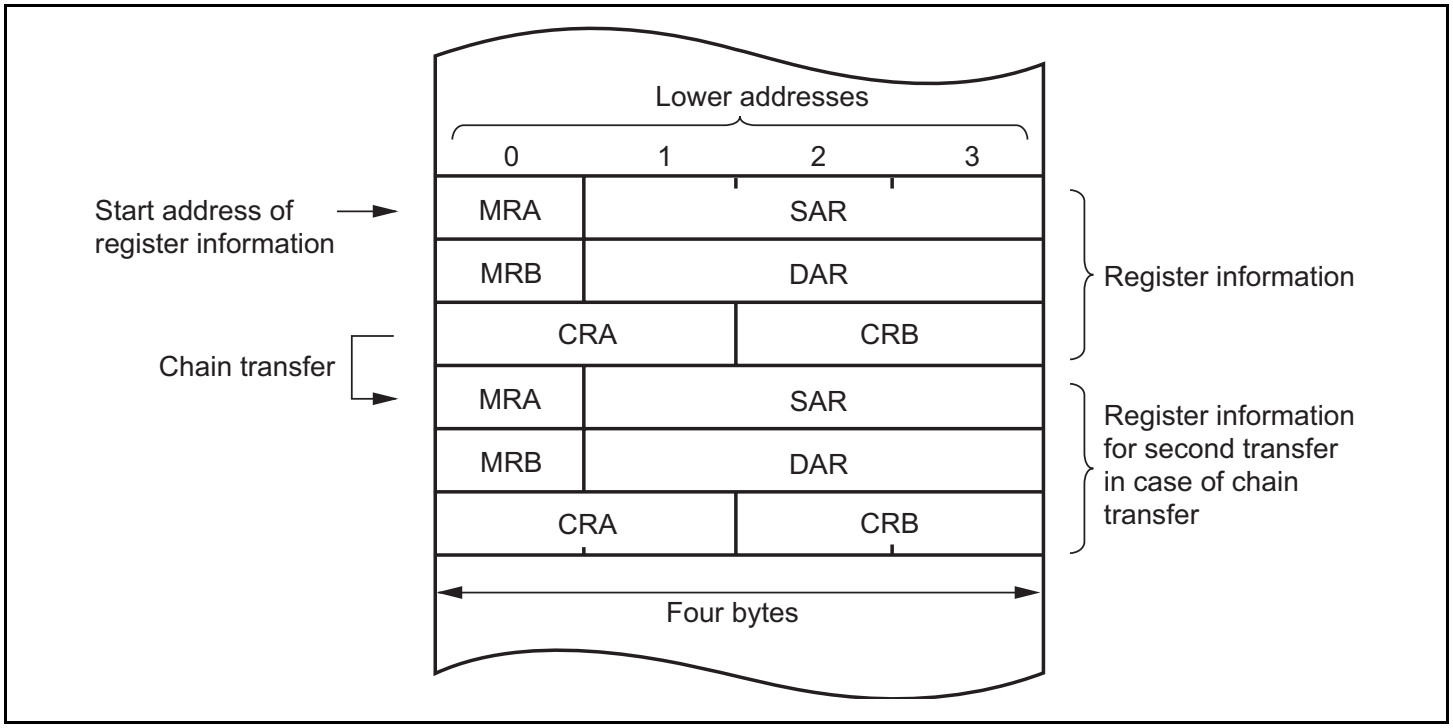


Figure 9.3 Correspondence between DTC Vector Address and Register Information

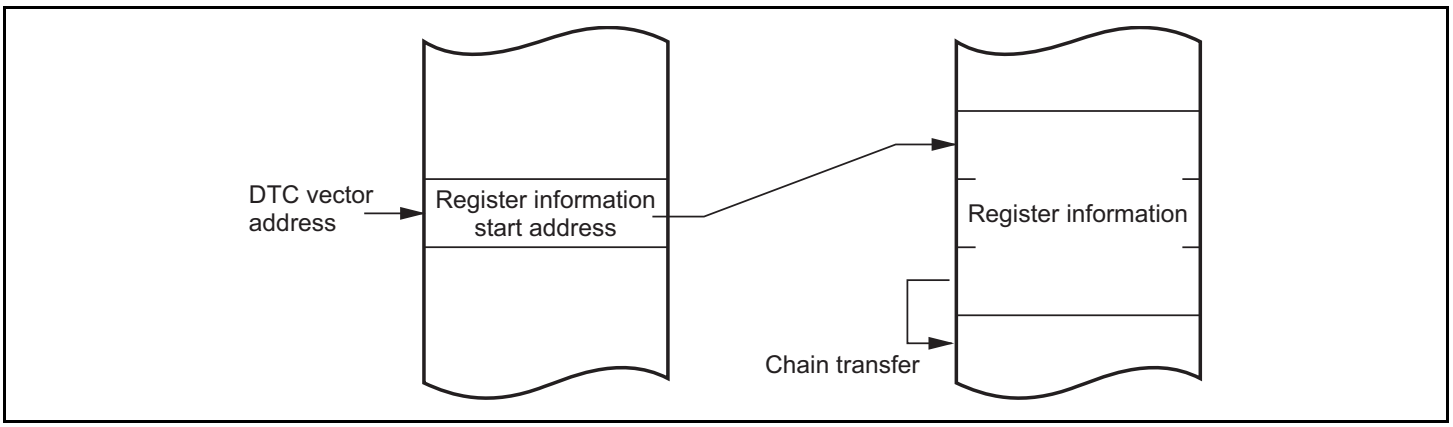


Figure 9.4 Correspondence between DTC Vector Address and Register Information

Table 9.2 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address	DTCE*1	Priority
Software	Write to DTVECR	DTVECR	H'0400 + (DTVECR[7:0] × 2)	—	High
External pin	IRQ0	16	H'0420	DTCEA7	
	IRQ1	17	H'0422	DTCEA6	
	IRQ2	18	H'0424	DTCEA5	
	IRQ3	19	H'0426	DTCEA4	
	IRQ4	20	H'0428	DTCEA3	
	IRQ5	21	H'042A	DTCEA2	
	IRQ6	22	H'042C	DTCEA1	
	IRQ7	23	H'042E	DTCEA0	
	IRQ8*2	24	H'0430	DTCEB7	
	IRQ9*2	25	H'0432	DTCEB6	
	IRQ10*2	26	H'0434	DTCEB5	
	IRQ11*2	17	H'0436	DTCEB4	
	IRQ12*2	18	H'0438	DTCEB3	
	IRQ13*2	19	H'043A	DTCEB2	
	IRQ14*2	30	H'043C	DTCEB1	
	IRQ15*2	31	H'043E	DTCEB0	
A/D_0	ADI0	38	H'044C	DTCEC6	Low

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address	DTCE*1	Priority
TPU_0	TGI0A	40	H'0450	DTCEC5	High
	TGI0B	41	H'0452	DTCEC4	
	TGI0C	42	H'0454	DTCEC3	
	TGI0D	43	H'0456	DTCEC2	
TPU_1	TGI1A	48	H'0460	DTCEC1	
	TGI1B	49	H'0462	DTCEC0	
TPU_2	TGI2A	52	H'0468	DTCED7	
	TGI2B	53	H'046A	DTCED6	
TPU_3	TGI3A	56	H'0470	DTCED5	
	TGI3B	57	H'0472	DTCED4	
	TGI3C	58	H'0474	DTCED3	
	TGI3D	59	H'0476	DTCED2	
TPU_4	TGI4A	64	H'0480	DTCED1	
	TGI4B	65	H'0482	DTCED0	
TPU_5	TGI5A	68	H'0488	DTCEE7	
	TGI5B	69	H'048A	DTCEE6	
TMR_0	CMIA0	72	H'0490	DTCEE3	
	CMIB0	73	H'0492	DTCEE2	
TMR_1	CMIA1	76	H'0498	DTCEE1	
	CMIB1	77	H'049A	DTCEE0	
DMAC	DMTEND0A	80	H'04A0	DTCEF7	
	DMTEND0B	81	H'04A2	DTCEF6	
	DMTEND1A	82	H'04A4	DTCEF5	
	DMTEND1B	83	H'04A6	DTCEF4	
SCI_0	RXI0	89	H'04B2	DTCEF3	
	TXI0	90	H'04B4	DTCEF2	
SCI_1	RXI1	93	H'04BA	DTCEF1	
	TXI1	94	H'04BC	DTCEF0	
SCI_2	RXI2	97	H'04C2	DTCEG7	
	TXI2	98	H'04C4	DTCEG6	Low

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address	DTCE*1	Priority
SCI_3	RXI3	101	H'04CA	DTCEF5	High
	TXI3	102	H'04CC	DTCEF4	
SCI_4	RXI4	105	H'04D2	DTCEG3	
	TXI4	106	H'04D4	DTCEG2	
A/D_1	ADI1	112	H'04E0	DTCEG1	
TPU_6	TGI6A	120	H'04F0	DTCEG0	
	TGI6B	121	H'04F2	DTCEH7	
	TGI6C	122	H'04F4	DTCEH6	
	TGI6D	123	H'04F6	DTCEH5	
TPU_7	TGI7A	125	H'04FA	DTCEH4	
	TGI7B	126	H'04FC	DTCEH3	
TPU_8	TGI8A	129	H'0502	DTCEH2	
	TGI8B	130	H'0504	DTCEH1	
TPU_9	TGI9A	133	H'050A	DTCEH0	
	TGI9B	134	H'050C	DTCEI7	
	TGI9C	135	H'050E	DTCEI6	
	TGI9D	136	H'0510	DTCEI5	
TPU_10	TGI10A	138	H'0514	DTCEI4	
	TGI10B	139	H'0516	DTCEI3	
TPU_11	TGI11A	142	H'051C	DTCEI2	
	TGI11B	143	H'051E	DTCEI1	Low

Notes: 1. DTCE bits with no corresponding interrupt are reserved, and 0 should be written to. When clearing the software standby state or all-module-clocks-stop mode with an interrupt, write 0 to the corresponding DTCE bit.

2. Not supported by the H8S/2454 Group.

9.5 Operation

The DTC stores register information in the on-chip RAM. When activated, the DTC reads register information that is already stored in the on-chip RAM and transfers data on the basis of that register information. After the data transfer, it writes updated register information back to the on-chip RAM. Pre-storage of register information in the on-chip RAM makes it possible to transfer data over any required number of channels. There are three transfer modes: normal mode, repeat mode, and block transfer mode. Setting the CHNE bit to 1 makes it possible to perform a number of transfers with a single activation (chain transfer). A setting can also be made to have chain transfer performed only when the transfer counter value is 0. This enables DTC re-setting to be performed by the DTC itself.

The 24-bit SAR designates the DTC transfer source address and the 24-bit DAR designates the transfer destination address. After each transfer, SAR and DAR are independently incremented, decremented, or left fixed.

Figure 9.5 shows a flowchart of DTC operation, and table 9.3 summarizes the chain transfer conditions (combinations for performing the second and third transfers are omitted).

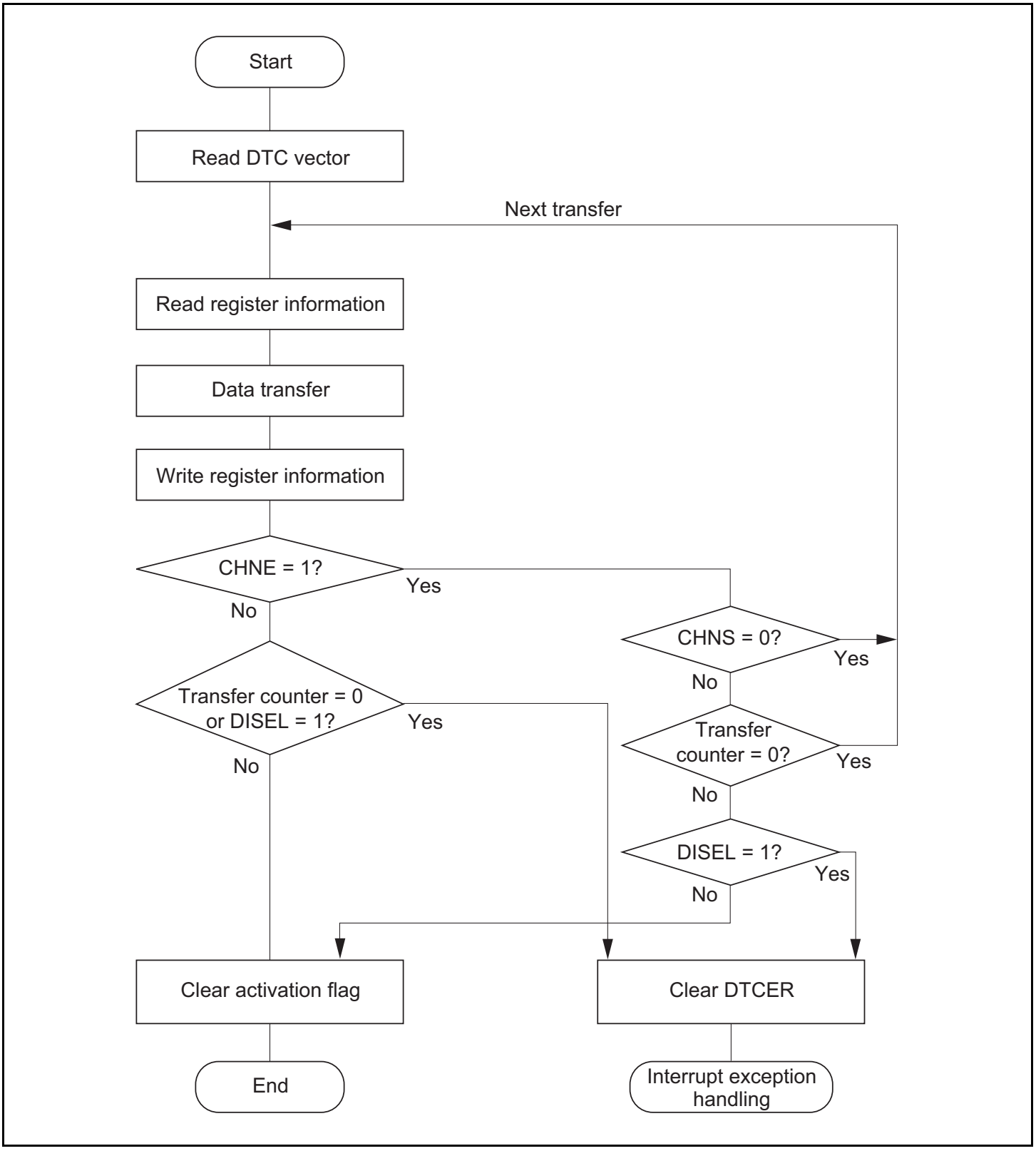


Figure 9.5 Flowchart of DTC Operation

Table 9.3 Chain Transfer Conditions

1st Transfer				2nd Transfer				
CHNE	**CHNS**	**DISEL**	**CR**	**CHNE**	**CHNS**	**DISEL**	**CR**	**DTC Transfer**
0	—	0	Not 0	—	—	—	—	Ends at 1st transfer
0	—	0	0	—	—	—	—	Ends at 1st transfer
0	—	1	—	—	—	—	—	Interrupt request to CPU
1	0	—	—	0	—	0	Not 0	Ends at 2nd transfer
				0	—	0	0	Ends at 2nd transfer
				0	—	1	—	Interrupt request to CPU
1	1	0	Not 0	—	—	—	—	Ends at 1st transfer
1	1	—	0	0	—	0	Not 0	Ends at 2nd transfer
				0	—	0	0	Ends at 2nd transfer
				0	—	1	—	Interrupt request to CPU
1	1	1	Not 0	—	—	—	—	Ends at 1st transfer
								Interrupt request to CPU

9.5.1 Normal Mode

In normal mode, one operation transfers one byte or one word of data. Table 9.4 lists the register function in normal mode. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has ended, a CPU interrupt can be requested.

Table 9.4 Register Function in Normal Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register A	CRA	Designates transfer count
DTC transfer count register B	CRB	Not used

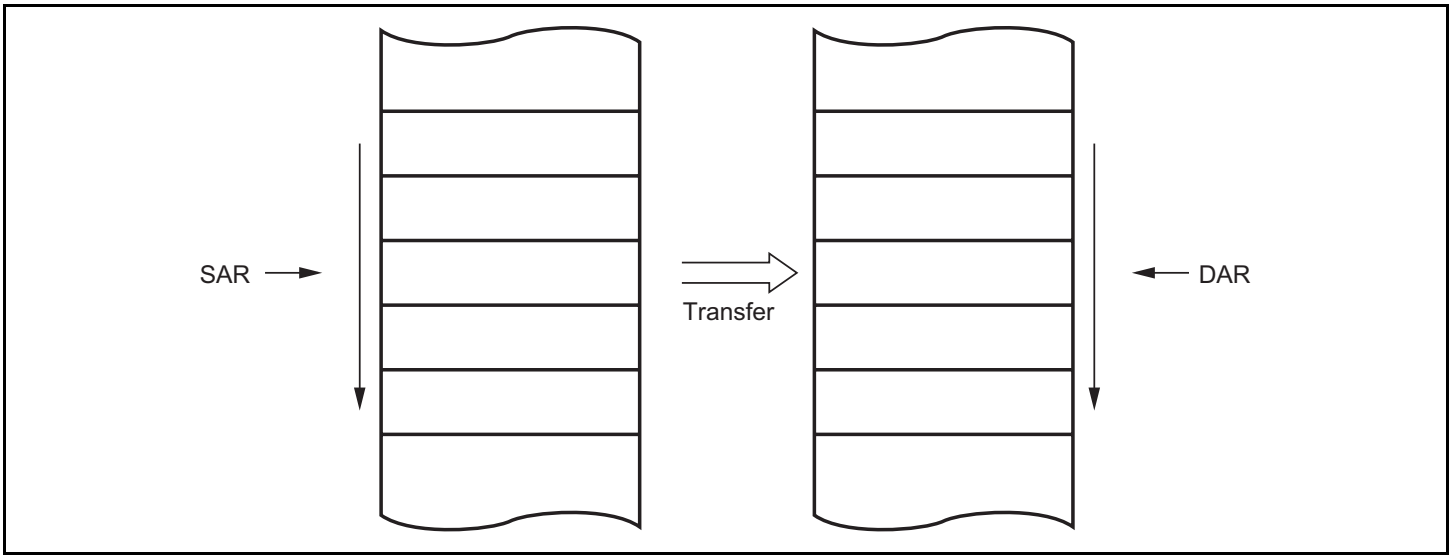


Figure 9.6 Memory Mapping in Normal Mode

9.5.2 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data. Table 9.5 lists the register function in repeat mode. From 1 to 256 transfers can be specified. Once the specified number of transfers has ended, the initial state of the transfer counter and the address register specified as the repeat area is restored, and transfer is repeated. In repeat mode the transfer counter value does not reach H'00, and therefore CPU interrupts cannot be requested when DISEL = 0.

Table 9.5 Register Function in Repeat Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register AH	CRAH	Holds number of transfers
DTC transfer count register AL	CRAL	Designates transfer count
DTC transfer count register B	CRB	Not used

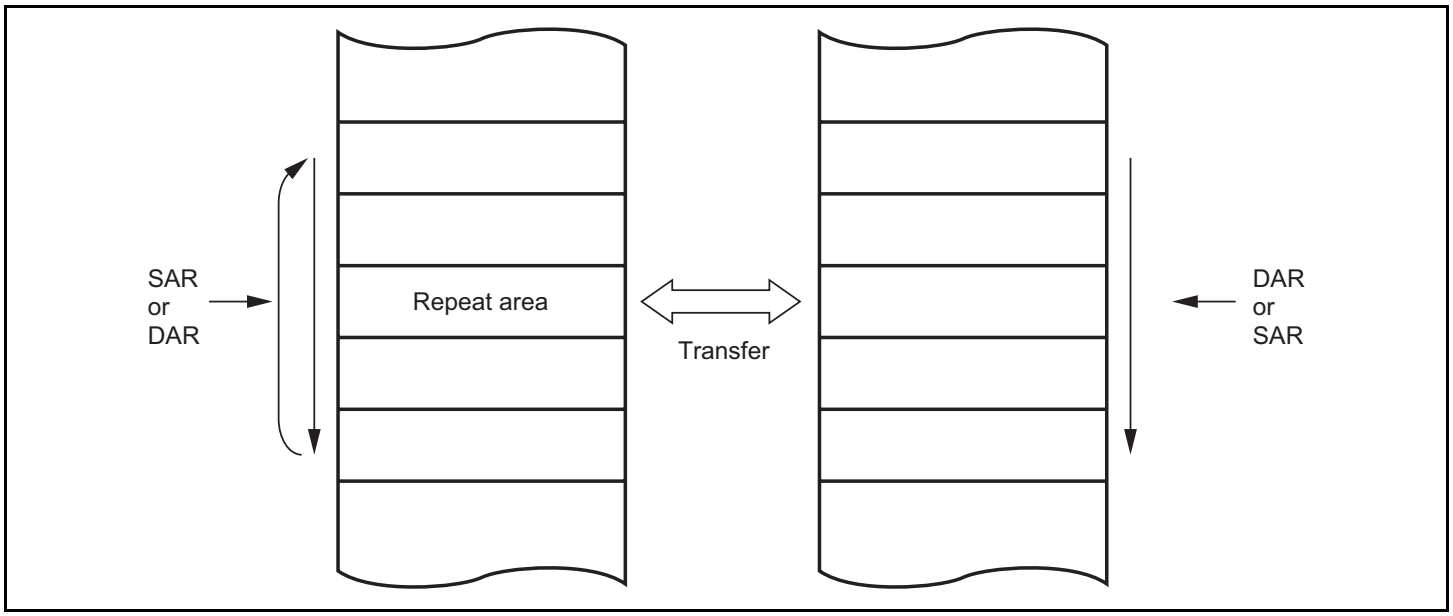


Figure 9.7 Memory Mapping in Repeat Mode

9.5.3 Block Transfer Mode

In block transfer mode, one operation transfers one block of data. Either the transfer source or the transfer destination is designated as a block area. Table 9.6 lists the register function in block transfer mode. The block size is 1 to 256. When the transfer of one block ends, the initial state of the block size counter and the address register specified as the block area is restored. The other address register is then incremented, decremented, or left fixed. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has ended, a CPU interrupt is requested.

Table 9.6 Register Function in Block Transfer Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register AH	CRAH	Holds block size
DTC transfer count register AL	CRAL	Designates block size count
DTC transfer count register B	CRB	Designates transfer count

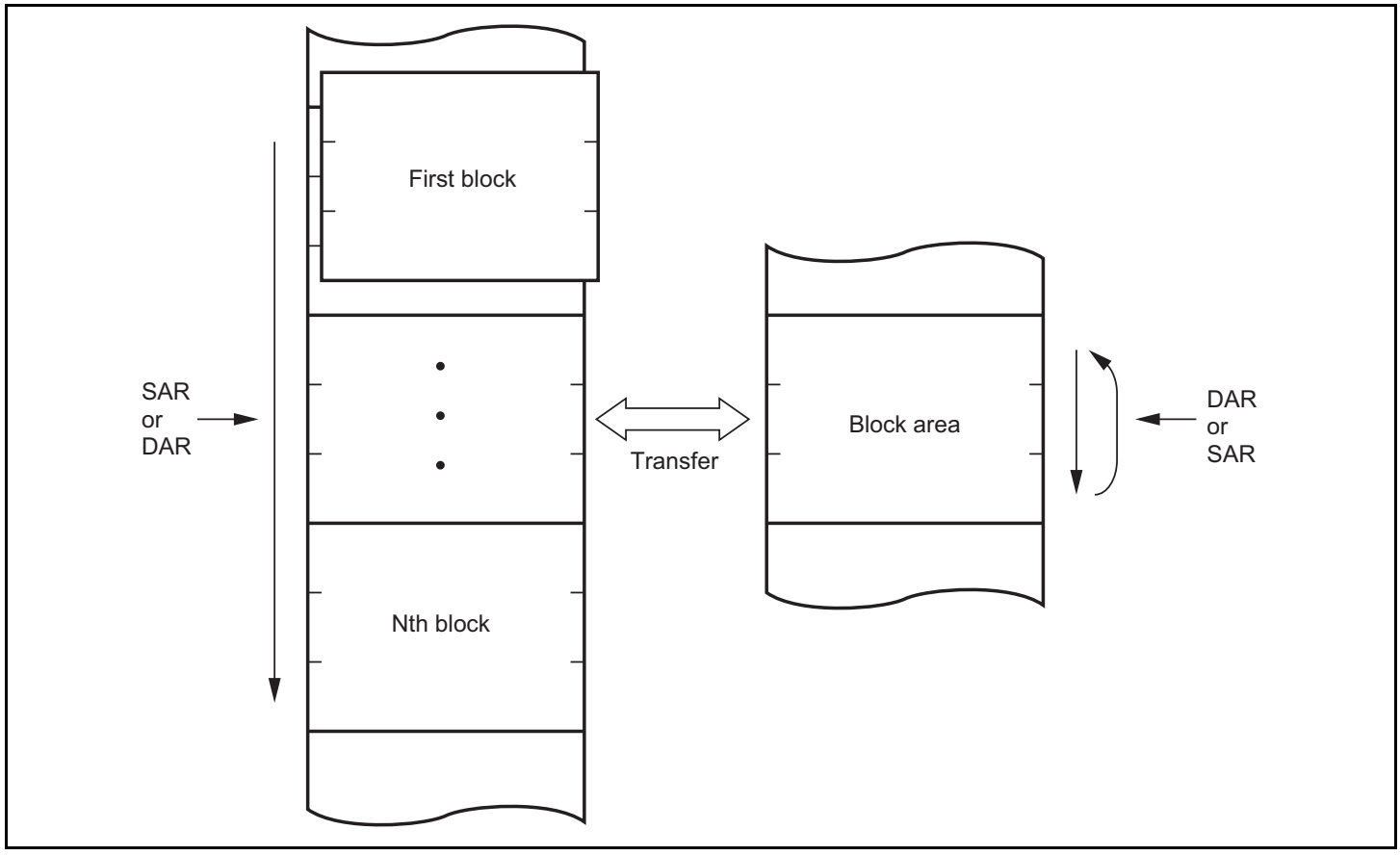


Figure 9.8 Memory Mapping in Block Transfer Mode

9.5.4 Chain Transfer

Setting the CHNE bit to 1 enables a number of data transfers to be performed consecutively in response to a single transfer request. SAR, DAR, CRA, CRB, MRA, and MRB, which define data transfers, can be set independently.

Figure 9.9 shows the operation of chain transfer. When activated, the DTC reads the register information start address stored at the vector address, and then reads the first register information at that start address. The CHNE bit in MRB is checked after the end of data transfer, if the value is 1, the next register information, which is located consecutively, is read and transfer is performed. This operation is repeated until the end of data transfer of register information with CHNE = 0. It is also possible, by setting both the CHNE bit and CHNS bit to 1, to specify execution of chain transfer only when the transfer counter value is 0.

In the case of transfer with CHNE set to 1, an interrupt request to the CPU is not generated at the end of the specified number of transfers or by setting of the DISEL bit to 1, and the interrupt source flag for the activation source is not affected.

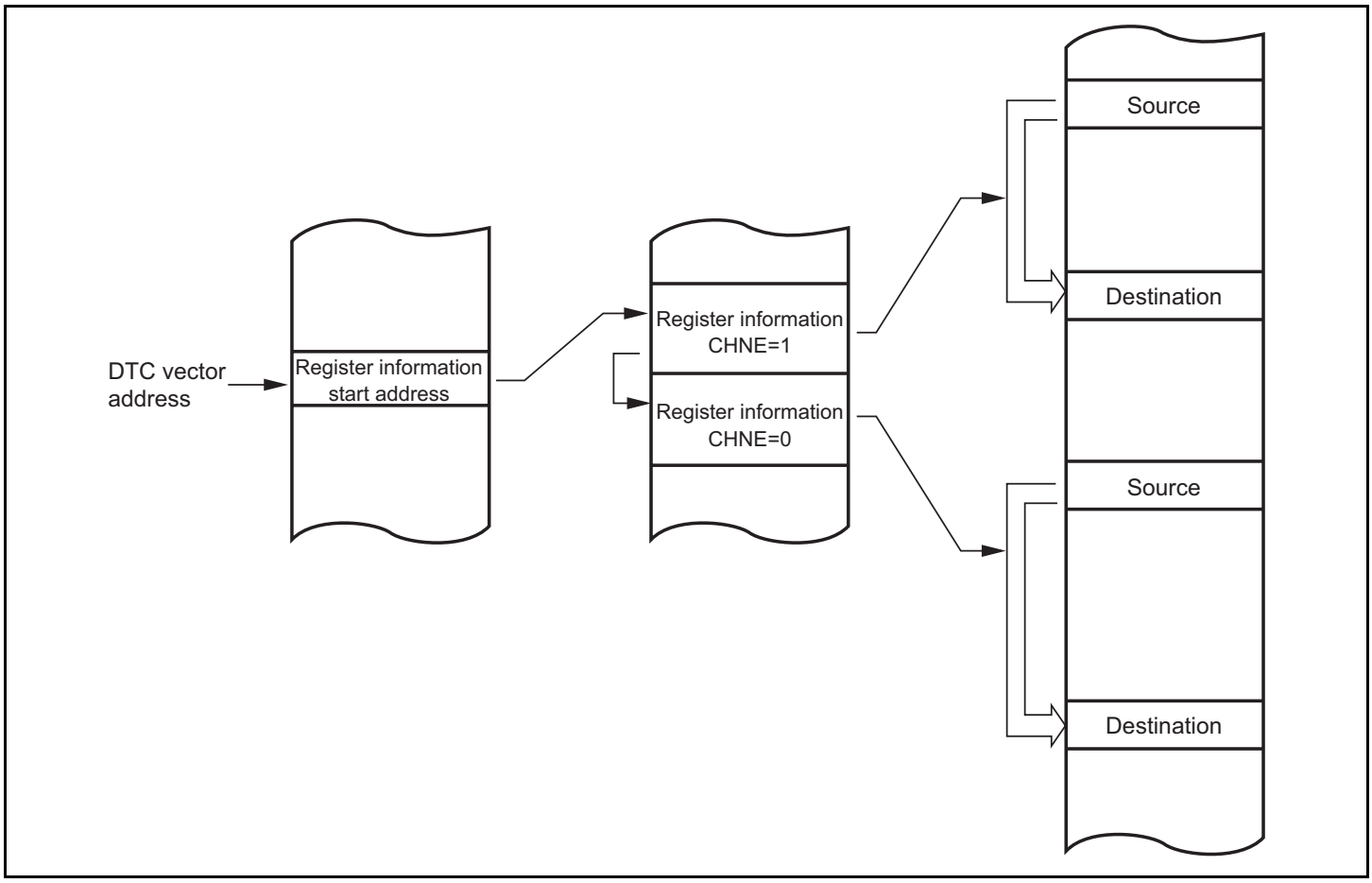


Figure 9.9 Operation of Chain Transfer

9.5.5 Interrupt Sources

An interrupt request is issued to the CPU when the DTC finishes the specified number of data transfers, or a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and interrupt controller priority level control.

In the case of activation by software, a software activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has ended, or the specified number of transfers has ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated. The interrupt handling routine should clear the SWDTE bit to 0.

When the DTC is activated by software, an SWDTEND interrupt is not generated during a data transfer wait or during data transfer even if the SWDTE bit is set to 1.

9.5.6 Operation Timing

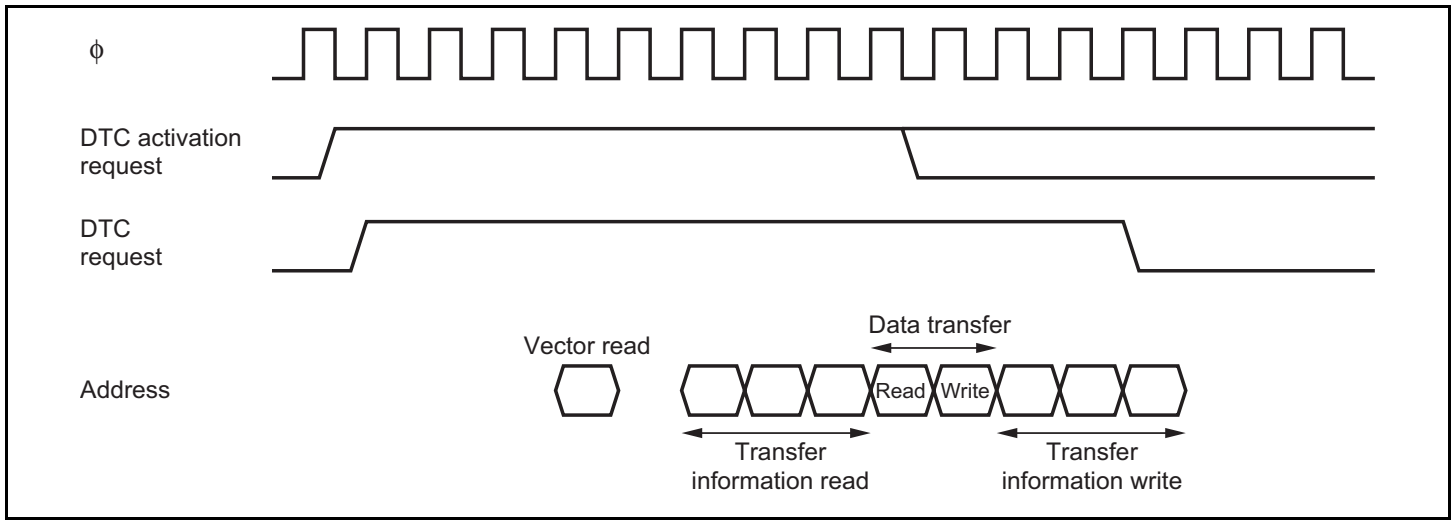


Figure 9.10 DTC Operation Timing (Example in Normal Mode or Repeat Mode)

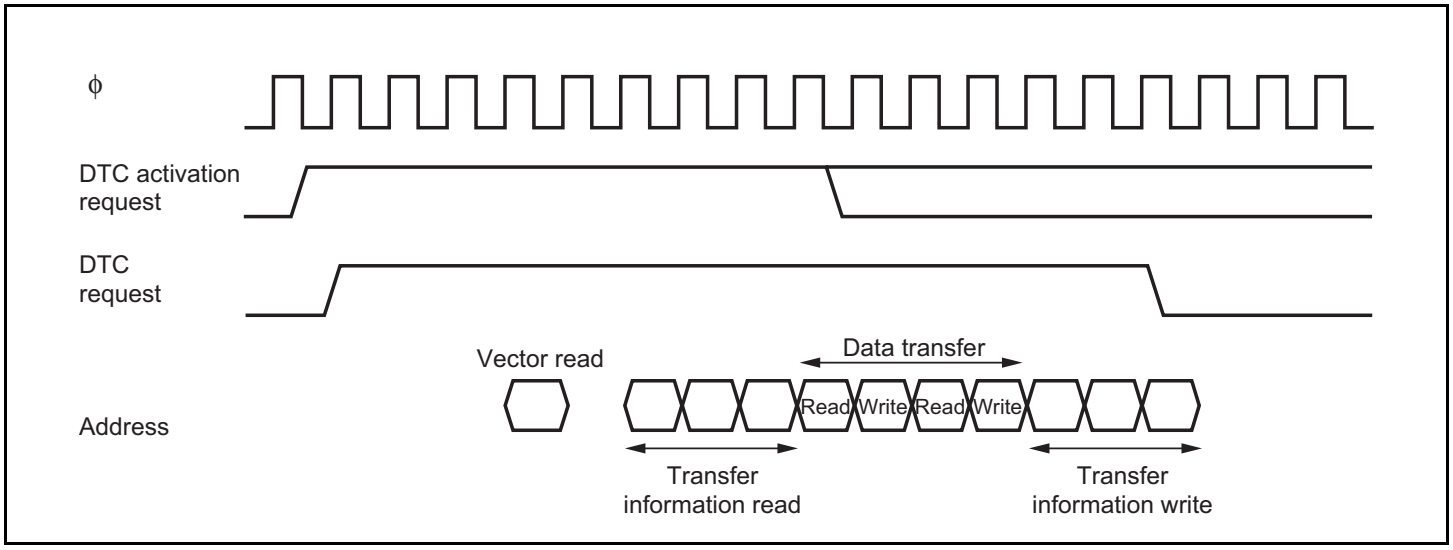


Figure 9.11 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

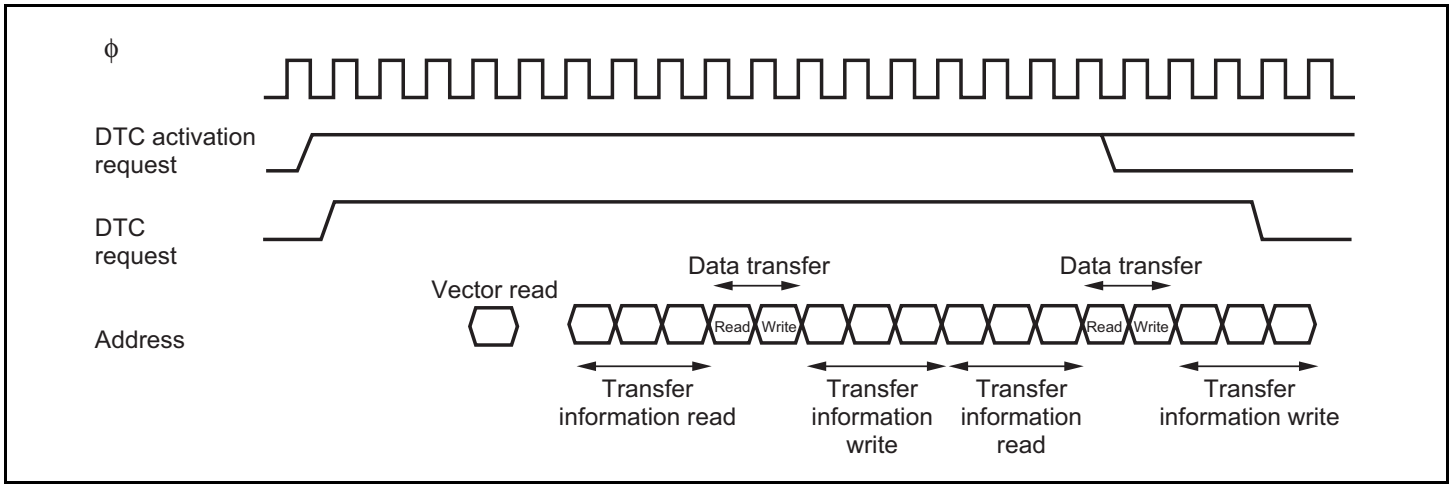


Figure 9.12 DTC Operation Timing (Example of Chain Transfer)

9.5.7 Number of DTC Execution States

Table 9.7 lists execution status for a single DTC data transfer, and table 9.8 shows the number of states required for each execution status.

Table 9.7 DTC Execution Status

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	Internal Operations M
Normal	1	6	1	1	3
Repeat	1	6	1	1	3
Block transfer	1	6	N	N	3

Legend:

N: Block size (initial setting of CRAH and CRAL)

Table 9.8 Number of States Required for Each Execution Status

Object to be Accessed			On-Chip RAM	On-Chip ROM	On-Chip I/O Registers		External Devices			
Bus width			32	16	8	16	8		16	
Access states			1	1	2	2	2	3	2	3
Execution status	Vector read	S_I	—	1	—	—	4	6+2m	2	3+m
	Register information read/write	S_J	1	—	—	—	—	—	—	—
	Byte data read	S_K	1	1	2	2	2	3+m	2	3+m
	Word data read	S_K	1	1	4	2	4	6+2m	2	3+m
	Byte data write	S_L	1	1	2	2	2	3+m	2	3+m
	Word data write	S_L	1	1	4	2	4	6+2m	2	3+m
	Internal operation	S_M	1							

The number of execution states is calculated from the formula below. Note that Σ means the sum of all transfers activated by one activation event (the number in which the CHNE bit is set to 1, plus 1).

$$\text{Number of execution states} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$$

For example, when the DTC vector address table is located in on-chip ROM, normal mode is set, and data is transferred from the on-chip ROM to an internal I/O register, the time required for the DTC operation is 13 states. The time from activation to the end of the data write is 10 states.

9.6 Procedures for Using DTC

9.6.1 Activation by Interrupt

The procedure for using the DTC with interrupt activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the enable bits for the interrupt sources to be used as the activation sources to 1. The DTC is activated when an interrupt used as an activation source is generated.
5. After the end of one data transfer, or after the specified number of data transfers have ended, the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the DTC is to continue transferring data, set the DTCE bit to 1.

9.6.2 Activation by Software

The procedure for using the DTC with software activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Check that the SWDTE bit is 0.
4. Write 1 to SWDTE bit and the vector number to DTVECR.
5. Check the vector number written to DTVECR.
6. After the end of one data transfer, if the DISEL bit is 0 and a CPU interrupt is not requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the SWDTE bit to 1. When the DISEL bit is 1, or after the specified number of data transfers have ended, the SWDTE bit is held at 1 and a CPU interrupt is requested.

9.7 Examples of Use of the DTC

9.7.1 Normal Mode

An example is shown in which the DTC is used to receive 128 bytes of data via the SCI.

1. Set MRA to fixed source address (SM1 = SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0). Set the SCI RDR address in SAR, the start address of the RAM area where the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.
2. Set the start address of the register information at the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The interrupt handling routine should perform wrap-up processing.

9.7.2 Chain Transfer

An example of DTC chain transfer is shown in which pulse output is performed using the PPG. Chain transfer can be used to perform pulse output data transfer and PPG output trigger cycle updating. Repeat mode transfer to NDR of the PPG is performed in the first half of the chain transfer, and normal mode transfer to the TPU's TGR in the second half. This is because clearing of the activation source and interrupt generation at the end of the specified number of transfers are restricted to the second half of the chain transfer (transfer when CHNE = 0).

1. Perform settings for transfer to NDR of the PPG. Set MRA to source address incrementing (SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), repeat mode (MD1 = 0, MD0 = 1), and word size (Sz = 1). Set the source side as a repeat area (DTS = 1). Set MRB to chain mode (CHNE = 1, DISEL = 0). Set the data table start address in SAR, the NDRH address in DAR, and the data table size in CRAH and CRAL. CRB can be set to any value.
2. Perform settings for transfer to the TPU's TGR. Set MRA to source address incrementing (SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), normal mode (MD1 = MD0 = 0), and word size (Sz = 1). Set the data table start address in SAR, the TGRA address in DAR, and the data table size in CRA. CRB can be set to any value.
3. Locate the TPU transfer register information consecutively after the NDR transfer register information.
4. Set the start address of the NDR transfer register information to the DTC vector address.
5. Set the bit corresponding to TGIA in DTCER to 1.
6. Set TGRA as an output compare register (output disabled) with TIOR, and enable the TGIA interrupt with TIER.
7. Set the initial output value in PODR, and the next output value in NDR. Set bits in DDR and NDER for which output is to be performed to 1. Using PCR, select the TPU compare match 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 and the set value of the next output trigger period is transferred to TGRA. The activation source TGFA flag is cleared.
10. When the specified number of transfers are completed (the TPU transfer CRA value is 0), the TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is sent to the CPU. Termination processing should be performed in the interrupt handling routine.

9.7.3 Chain Transfer when Counter = 0

By executing a second data transfer, and performing re-setting of the first data transfer, only when the counter value is 0, it is possible to perform 256 or more repeat transfers.

An example is shown in which a 128-Kbyte input buffer is configured. The input buffer is assumed to have been set to start at lower address H'0000. Figure 9.13 shows the chain transfer when the counter value is 0.

1. For the first transfer, set the normal mode for input data. Set fixed transfer source address (G/A, etc.), CRA = H'0000 (65,536 times), and CHNE = 1, CHNS = 1, and DISEL = 0.
2. Prepare the upper 8-bit addresses of the start addresses for each of the 65,536 transfer start addresses for the first data transfer in a separate area (in ROM, etc.). For example, if the input buffer comprises H'200000 to H'21FFFF, prepare H'21 and H'20.
3. For the second transfer, set repeat mode (with the source side as the repeat area) for re-setting the transfer destination address for the first data transfer. Use the upper 8 bits of DAR in the first register information area as the transfer destination. Set CHNE = DISEL = 0. If the above input buffer is specified as H'200000 to H'21FFFF, set the transfer counter to 2.
4. Execute the first data transfer 65,536 times by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper 8 bits of the transfer source address for the first data transfer to H'21. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
5. Next, execute the first data transfer the 65,536 times specified for the first data transfer by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper 8 bits of the transfer source address for the first data transfer to H'20. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
6. Steps 4 and 5 are repeated endlessly. As repeat mode is specified for the second data transfer, an interrupt request is not sent to the CPU.

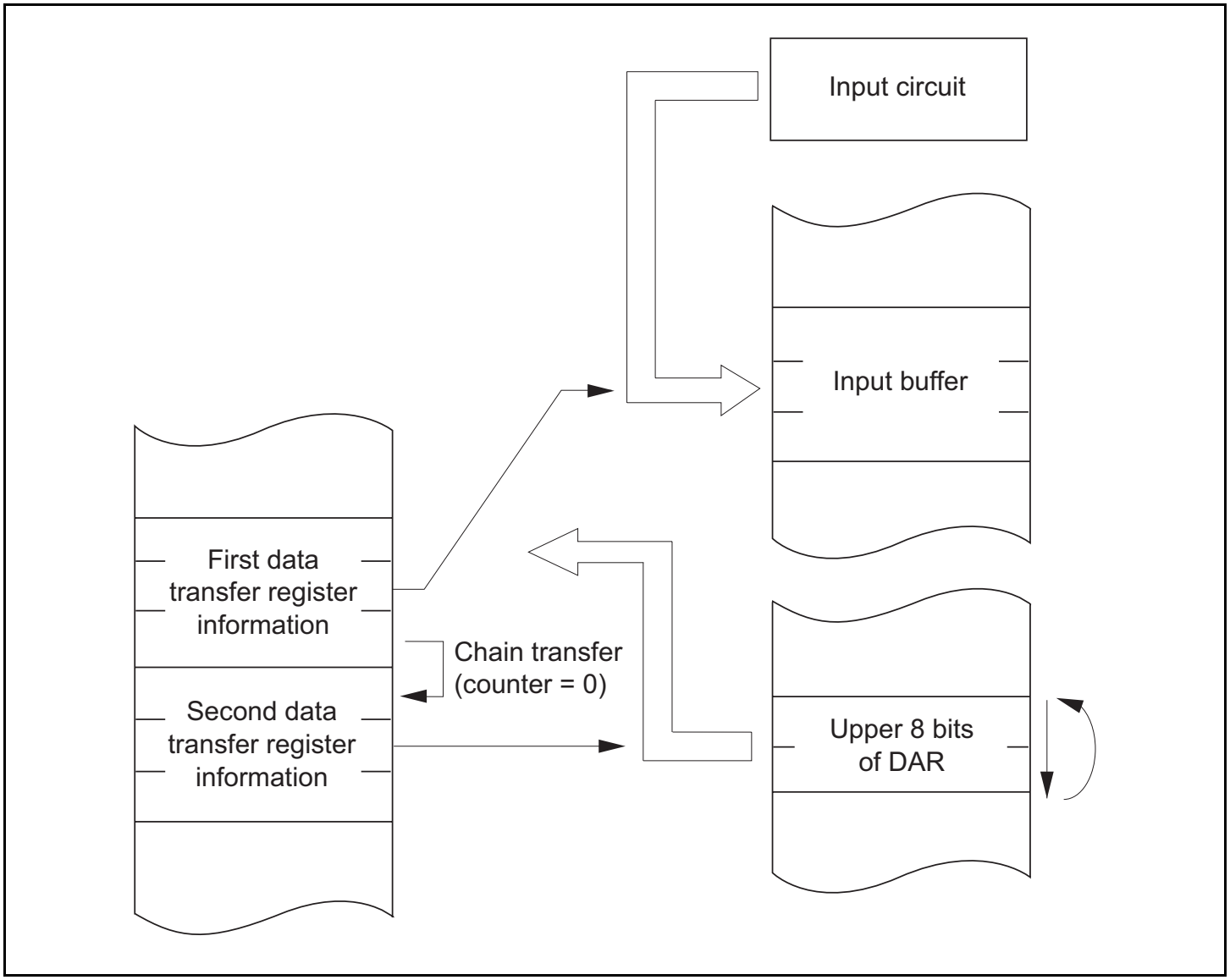


Figure 9.13 Chain Transfer when Counter = 0

9.7.4 Software Activation

An example is shown in which the DTC is used to transfer a block of 128 bytes of data by means of software activation. The transfer source address is H'1000 and the destination address is H'2000. The vector number is H'60, so the vector address is H'04C0.

1. Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt (CHNE = 0). Set the transfer source address (H'1000) in SAR, the destination address (H'2000) in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.
2. Set the start address of the register information at the DTC vector address (H'04C0).
3. Check that the SWDTE bit in DTCCR is 0. Check that there is currently no transfer activated by software.
4. Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data is H'60.
5. Read DTVECR again and check that it is set to the vector number (H'60). If it is not, this indicates that the write failed. This is presumably because an interrupt occurred between steps 3 and 4 and led to a different software activation. To activate this transfer, go back to step 3.
6. If the write was successful, the DTC is activated and a block of 128 bytes of data is transferred.
7. After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine should clear the SWDTE bit to 0 and perform other wrap-up processing.

9.8 Usage Notes

9.8.1 Module Stop Mode Setting

DTC operation can be disabled or enabled using the module stop control register. The initial setting is for DTC operation to be enabled. Register access is disabled by setting module stop mode. Module stop mode cannot be set while the DTC is activated. For details, refer to section 25, Power-Down Modes.

9.8.2 On-Chip RAM

The MRA, MRB, SAR, DAR, CRA, and CRB registers are all located in on-chip RAM. When the DTC is used, the RAME bit in SYSCR must not be cleared to 0.

9.8.3 DTCE Bit Setting

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all interrupts are disabled, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

9.8.4 DMAC Transfer End Interrupt

When DTC transfer is activated by a DMAC transfer end interrupt, regardless of the transfer counter and DISEL bit, the DMAC's DTE bit is not subject to DTC control, and the write data has priority. Consequently, an interrupt request may not be sent to the CPU when the DTC transfer counter reaches 0.

9.8.5 Chain Transfer

When chain transfer is used, clearing of the activation source or DTCER is performed when the last of the chain of data transfers is executed. SCI and high-speed A/D converter interrupt/activation sources, on the other hand, are cleared when the DTC reads or writes to the prescribed register. Therefore, when the DTC is activated by an interrupt or activation source, if a read/write of the relevant register is not included in the last chained data transfer, the interrupt or activation source will be retained.

Section 10 I/O Ports

Table 10.1 summarizes the port functions of the H8S/2456 Group and H8S/2456R Group. Table 10.2 summarizes the port functions of the H8S/2454 Group. The pins of each port also have other functions such as input/output or external interrupt input pins of on-chip peripheral modules. Each I/O port includes a data direction register (DDR) that controls input/output, a data register (DR) that stores output data, a port register (PORT) used to read the pin states, and a port function control register (PFCR) used to set input/output destination. Before enabling each input/output pins, select the input/output destination by PFCR. The input-only ports do not have a DR or DDR register.

Ports A to E have a built-in pull-up MOS function and a pull-up MOS control register (PCR) to control the on/off state of the input pull-up MOS.

Ports 1 to 3, 5 to 8, and A to J include an open-drain control register (ODR) that controls the on/off state of the output buffer PMOS.

Ports 1 to 3, 5 (P50 to P53), 6, and 8 can drive a single TTL load and 30-pF capacitive load. Ports A to H can drive a single TTL load and 50-pF capacitive load.

All of the I/O ports can drive a Darlington transistor when outputting data.

Ports 1 and 2 are Schmitt-triggered inputs.

- H8S/2456 Group and H8S/2456R Group

  Ports 5 (P50 to P52), 8 (P81, P83, and P85), B, and, C are Schmitt-triggered inputs when used as TPU inputs.

  Ports 2 (P20 and P25 to P27), 5, 6, 8, A (PA4 to PA7), F (PF1 and PF2), and H (PH2 and PH3) are Schmitt-triggered inputs when used as IRQ inputs.

  Ports 3 (P32 to P35) and 5 (P50 and P51) are Schmitt-triggered inputs when used as I^2C inputs.
  Ports 5 (P51 and P51), 6 (P60 to P63), and 8 (P81 and P83) are Schmitt-triggered inputs when used as 8-bit timer inputs.
- H8S/2454 Group

  Ports 5 (P50 to P52), 8 (P81, P83, and P85), B, and C are Schmitt-triggered inputs when used as TPU inputs.

  Ports 4, 5, 8, and A (PA4 to PA7) are Schmitt-triggered inputs when used as IRQ inputs.

  Ports 3 (P32 to P35) and 5 (P50 and P51) are Schmitt-triggered inputs when used as I^2C inputs.
  Ports 2 (P20), 5 (P50 and P51), and 8 (P81 and P83) are Schmitt-triggered inputs when used as 8-bit timer inputs.

Table 10.1 Port Functions of H8S/2456 Group and H8S/2456R Group

Port	Description	Mode 1	Mode 2	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port 1	General I/O port also functioning as PPG outputs, TPU I/Os, EXDMAC outputs, and SSU I/Os	P17/PO15/TIOCB2/TCLKD/$\overline{\text{EDRAK3}}$/$\overline{\text{SCS0-A}}$				P17/PO15/TIOCB2/ TCLKD/$\overline{\text{SCS0-A}}$	Schmitt-triggered inputs when used as general input port and TPU inputs. Open-drain output capability.
		P16/PO14/TIOCA2/$\overline{\text{EDRAK2}}$/SSCK0-A				P16/PO14/TIOCA2/ SSCK0-A	
		P15/PO13/TIOCB1/TCLKC/SSI0-A P14/PO12/TIOCA1/SSO0-A P13/PO11/TIOCD0/TCLKB P12/PO10/TIOCC0/TCLKA P11/PO9/TIOCB0 P10/PO8/TIOCA0					
Port 2	General I/O port also functioning as PPG outputs, TPU I/Os, interrupt inputs, SCI I/Os, I^2C I/Os, A/D converter inputs, bus control signal I/Os, and USB I/Os.	P27/$\overline{\text{IRQ15-B}}$/PO7/TIOCB5/SCL2 P26/$\overline{\text{IRQ14-B}}$/PO6/TIOCA5/SDA2/$\overline{\text{ADTRG1}}$					Schmitt-triggered inputs when used as general input port, TPU inputs, interrupt inputs, and I^2C inputs. Open-drain output capability. 5-V tolerance.
		P25/$\overline{\text{WAIT-B}}$/$\overline{\text{IRQ13-B}}$/PO5-A/TIOCB4-A/VBUS				P25/$\overline{\text{IRQ13-B}}$/PO5-A/ TIOCB4-A/VBUS	
		P20/$\overline{\text{IRQ8-B}}$/PO0-A/TIOCA3-A/PUPD+					
Port 3	General I/O port also functioning as SCI I/Os, I^2C I/Os, and bus control signal I/Os	P35/$\overline{\text{OE-B}}$*[1]/CKE-B/SCK1/SCL0				P35/SCK1/SCL0	Open-drain output capability. Only P32 to P35 are Schmitt-triggered inputs when used as I^2C inputs. P32 to P35 have 5-V tolerance.
		P34/SCK0/SCK4-A/SDA0 P33/RxD1/SCL1 P32/RxD0/IrRxD/SDA1 P31/TxD1 P30/TxD0/IrTxD					
Port 4	General I/O port also functioning as A/D converter analog inputs.	P47/AN7_0 P46/AN6_0 P45/AN5_0 P44/AN4_0 P43/AN3_0 P42/AN2_0 P41/AN1_0 P40/AN0_0					

Port	Description	Mode 1	Mode 2	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port 5	General I/O port also functioning as interrupt inputs, A/D converter inputs, SCI I/Os, PPG outputs, TPU I/Os, TMR I/Os, I^2C I/Os, bus control signal I/Os, and JTAG inputs	P53/$\overline{\text{IRQ3-A}}$/$\overline{\text{ADTRG0-A}}$/$\overline{\text{TRST}}$*[3]					Schmitt-triggered inputs when used as IRQ inputs. Only P50 and P51 are Schmitt-triggered inputs when used as I^2C inputs. Open-drain output capability. Only P50 to P52 are Schmitt-triggered inputs when used as TPU inputs. Only P50 and P51 are Schmitt-triggered inputs when used as 8-bit timer inputs. P50 and P51 have 5-V tolerance.
		P52/$\overline{\text{BACK-B}}$/$\overline{\text{IRQ2-A}}$/PO4-B/TIOCA4-B/ TMO0-B/SCK2				P52/$\overline{\text{IRQ2-A}}$/PO4-B/ TIOCA4-B/TMO0-B/ SCK2	
		P51/$\overline{\text{BREQ-B}}$/$\overline{\text{IRQ1-A}}$/PO2-B/TIOCC3-B/ TMCI0-B/RxD2/SCL3				P51/$\overline{\text{IRQ1-A}}$/PO2-B/ TIOCC3-B/TMCI0-B/ RxD2/SCL3	
		P50/$\overline{\text{BREQO-B}}$/$\overline{\text{IRQ0-A}}$/PO0-B/TIOCA3-B/ TMRI0-B/TxD2/SDA3				P50/$\overline{\text{IRQ0-A}}$/PO0-B/ TIOCA3-B/TMRI0-B/ TxD2/SDA3	
Port 6	General I/O port also functioning as interrupt inputs, TMR I/Os, and DMAC I/Os	P65/$\overline{\text{IRQ13-A}}$/$\overline{\text{DACK1}}$/TMO1-A P64/$\overline{\text{IRQ12-A}}$/$\overline{\text{DACK0}}$/TMO0-A P63/$\overline{\text{IRQ11-A}}$/$\overline{\text{TEND1}}$/TMCI1-A P62/$\overline{\text{IRQ10-A}}$/$\overline{\text{TEND0}}$/TMCI0-A P61/$\overline{\text{IRQ9-A}}$/$\overline{\text{DREQ1}}$/TMRI1-A P60/$\overline{\text{IRQ8-A}}$/$\overline{\text{DREQ0}}$/TMRI0-A					Schmitt-triggered inputs when used as IRQ inputs. Open-drain output capability. Only P60 to P63 are Schmitt-triggered inputs when used as 8-bit timer inputs.
Port 8	General I/O port also functioning as EXDMAC I/Os, PPG outputs, TPU I/Os, TMR I/Os, SCI I/Os and interrupt inputs	P85/$\overline{\text{IRQ5-B}}$ /PO5-B/TIOCB4-B/TMO1-B/SCK3/ $\overline{\text{EDACK3}}$				P85/$\overline{\text{IRQ5-B}}$/PO5-B/ TIOCB4-B/TMO1-B/ SCK3	Schmitt-triggered inputs when used as IRQ inputs. Open-drain output capability. Only P81, P83, and P85 are Schmitt-triggered inputs when used as TPU inputs.
		P84/$\overline{\text{IRQ4-B}}$/$\overline{\text{EDACK2}}$				P84/$\overline{\text{IRQ4-B}}$	
		P83/$\overline{\text{IRQ3-B}}$/PO3-B/TIOCD3-B/TMCI1-B/RxD3/ $\overline{\text{ETEND3}}$				P83/$\overline{\text{IRQ3-B}}$/PO3-B/ TIOCD3-B/TMCI1-B/ RxD3	
		P82/$\overline{\text{IRQ2-B}}$/$\overline{\text{ETEND2}}$				P82/$\overline{\text{IRQ2-B}}$	
		P81/$\overline{\text{IRQ1-B}}$/PO1-B/TIOCB3-B/ TMRI1-B/TxD3/ $\overline{\text{EDREQ3}}$				P81/$\overline{\text{IRQ1-B}}$/PO1-B/ TIOCB3-B/TMRI1-B/ TxD3	
		P80/$\overline{\text{IRQ0-B}}$/$\overline{\text{EDREQ2}}$				P80/$\overline{\text{IRQ0-B}}$	
Port 9	Dedicated input port also functioning as A/D converter analog inputs and D/A converter analog outputs	P97/AN15_1 P96/AN14_1 P95/AN13_1/DA3 P94/AN12_1/DA2 P93/AN11_1 P92/AN10_1 P91/AN9_1 P90/AN8_1					

Port	Description	Mode 1	Mode 2	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port A	General I/O port also functioning as address outputs, interrupt inputs, SSU I/Os, and SCI I/Os	PA7/A23/IRQ7-A/SSO0-B		PA7/A23/IRQ7-A/SSO0-B		PA7/IRQ7-A/SSO0-B	Only PA4 to PA7 are Schmitt-triggered inputs when used as IRQ inputs. Built-in input pull-up MOS. Open-drain output capability.
		PA6/A22/IRQ6-A/SSI0-B		PA6/A22/IRQ6-A/SSI0-B		PA6/IRQ6-A/SSI0-B	
		PA5/A21/IRQ5-A/SSCK0-B		PA5/A21/IRQ5-A/SSCK0-B		PA5/IRQ5-A/SSCK0-B	
		A20/IRQ4-A		PA4/A20/IRQ4-A/SCS0-B		PA4/IRQ4-A/SCS0-B	
		A19		PA3/A19/SCK4-B		PA3/SCK4-B	
		A18		PA2/A18/RxD4-B		PA2/RxD4-B	
		A17		PA1/A17/TxD4-B		PA1/TxD4-B	
		A16		PA0/A16		PA0	
Port B	General I/O port also functioning as address outputs and TPU I/Os	A15		PB7/A15		PB7/TIOCB8/TCLKH	Built-in input pull-up MOS. Schmitt-triggered inputs when used as TPU inputs. Open-drain output capability.
		A14		PB6/A14		PB6/TIOCA8	
		A13		PB5/A13		PB5/TIOCB7/TCLKG	
		A12		PB4/A12		PB4/TIOCA7	
		A11		PB3/A11		PB3/TIOCD6/TCLKF	
		A10		PB2/A10		PB2/TIOCC6/TCLKE	
		A9		PB1/A9		PB1/TIOCB6	
		A8		PB0/A8		PB0/TIOCA6	
Port C	General I/O port also functioning as address outputs and TPU I/Os	A7		PC7/A7		PC7/TIOCB11	Built-in input pull-up MOS. Schmitt-triggered inputs when used as TPU inputs. Open-drain output capability.
		A6		PC6/A6		PC6/TIOCA11	
		A5		PC5/A5		PC5/TIOCB10	
		A4		PC4/A4		PC4/TIOCA10	
		A3		PC3/A3		PC3/TIOCD9	
		A2		PC2/A2		PC2/TIOCC9	
		A1		PC1/A1		PC1/TIOCB9	
		A0		PC0/A0		PC0/TIOCA9	
Port D	General I/O port also functioning as data I/Os	D15				PD7	Built-in input pull-up MOS. Open-drain output capability.
		D14				PD6	
		D13				PD5	
		D12				PD4	
		D11				PD3	
		D10				PD2	
		D9				PD1	
		D8				PD0	

Port	Description	Mode 1	Mode 2	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port E	General I/O port also functioning as data I/Os	PE7/D7 PE6/D6 PE5/D5 PE4/D4 PE3/D3 PE2/D2 PE1/D1 PE0/D0	PE7/D7 PE6/D6 PE5/D5 PE4/D4 PE3/D3 PE2/D2 PE1/D1 PE0/D0			PE7 PE6 PE5 PE4 PE3 PE2 PE1 PE0	Built-in input pull-up MOS. Open-drain output capability.
Port F	General I/O port also functioning as interrupt inputs, bus control signal I/Os, SSU I/Os, and A/D converter inputs	PF7/ϕ PF6/$\overline{\text{AS}}$/$\overline{\text{AH}}$ $\overline{\text{RD}}$ $\overline{\text{HWR}}$ PF3/$\overline{\text{LWR}}$/SSO0-C PF2/$\overline{\text{LCAS}}$/DQML/$\overline{\text{IRQ15-A}}$/SSI0-C PF1/$\overline{\text{UCAS}}$/DQMU/$\overline{\text{IRQ14-A}}$/SSCK0-C PF0/$\overline{\text{WAIT-A}}$/$\overline{\text{ADTRG0-B}}$/$\overline{\text{SCS0-C}}$				PF7/ϕ PF6 PF5 PF4 PF3/SSO0-C PF2/$\overline{\text{IRQ15-A}}$/SSI0-C PF1/$\overline{\text{IRQ14-A}}$/ SSCK0-C PF0/$\overline{\text{ADTRG0-B}}$/ $\overline{\text{SCS0-C}}$	Only PF1 and PF2 are Schmitt-triggered inputs when used as IRQ inputs. Open-drain output capability.
Port G	General I/O port also functioning as bus control signal I/Os and JTAG inputs	PG6/$\overline{\text{BREQ-A}}$/TDI*[3] PG5/$\overline{\text{BACK-A}}$/TMS*[3] PG4/$\overline{\text{BREQO-A}}$/TCK*[3] PG3/$\overline{\text{CS3}}$/$\overline{\text{RAS3}}$/$\overline{\text{CAS}}$*[1] PG2/$\overline{\text{CS2}}$/$\overline{\text{RAS2}}$/$\overline{\text{RAS}}$*[1] PG1/$\overline{\text{CS1}}$ PG0/$\overline{\text{CS0}}$				PG6/TDI*[3] PG5/TMS*[3] PG4/TCK*[3] PG3 PG2 PG1 PG0	Open-drain output capability.
Port H	General I/O port also functioning as interrupt inputs and bus control signal I/Os	PH3/$\overline{\text{CS7}}$/$\overline{\text{OE-A}}$/CKE-A*[1]/$\overline{\text{IRQ7-B}}$ PH2/$\overline{\text{CS6}}$/$\overline{\text{IRQ6-B}}$ PH1/$\overline{\text{CS5}}$/$\overline{\text{RAS5}}$/SDRAMϕ*[1] PH0/$\overline{\text{CS4}}$/$\overline{\text{RAS4}}$/$\overline{\text{WE}}$*[1]				PH3/$\overline{\text{IRQ7-B}}$ PH2/$\overline{\text{IRQ6-B}}$ PH1/SDRAMϕ*[1] PH0	Only PH2 and PH3 are Schmitt-triggered inputs when used as IRQ inputs. Open-drain output capability.
Port J	General I/O port	PJ2*[2] PJ1 PJ0					Open-drain output capability for only PJ0 and PJ1. 5-V tolerance.

Notes: 1. Not supported in the H8S/2456 Group.
2. Not supported in the 145-pin package.
3. Supported only in the 145-pin package.

Table 10.2 Port Functions of H8S/2454 Group

Port	Description	Mode 1	Mode 2	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port 1	General I/O port also functioning as PPG outputs, TPU I/Os, DMAC I/Os, and SSU I/Os	P17/PO15/TIOCB2/TCLKD/$\overline{\text{SCS0-A}}$ P16/PO14/TIOCA2/SSCK0-A P15/$\overline{\text{DACK1}}$/PO13/TIOCB1/TCLKC/SSI0-A P14/$\overline{\text{DACK0}}$/PO12/TIOCA1/SSO0-A P13/$\overline{\text{TEND1}}$/PO11/TIOCD0/TCLKB P12/$\overline{\text{TEND0}}$/PO10/TIOCC0/TCLKA P11/$\overline{\text{DREQ1}}$/PO9/TIOCB0 P10/$\overline{\text{DREQ0}}$/PO8/TIOCA0					Schmitt-triggered inputs. Open-drain output capability.
Port 2	General I/O port also functioning as PPG outputs, TPU I/Os, SCI I/Os, TMR I/Os, I^{2}C I/Os, A/D converter inputs, bus control signal I/Os, and USB I/Os.	P27/PO7/TIOCB5/SCL2 P26/PO6/TIOCA5/SDA2/$\overline{\text{ADTRG1}}$					Schmitt-triggered inputs. Open-drain output capability. 5-V tolerance.
		P25/$\overline{\text{WAIT-B}}$/PO5-A/TIOCB4-A/TMO1-A/VBUS				P25/PO5-A/TIOCB4-A/ TMO1-A/VBUS	
		P20/PO0-A/TIOCA3-A/TMRI0/PUPD+					
Port 3	General I/O port also functioning as SCI I/Os, I^{2}C I/Os, and bus control signal I/Os	P35/$\overline{\text{OE-B}}$/SCK1/SCL0				P35/SCK1/SCL0	Open-drain output capability. Only P32 to P35 are Schmitt-triggered inputs when used as I^{2}C inputs. P32 to P35 have 5-V tolerance.
		P34/SCK0/SCK4-A/SDA0 P33/RxD1/SCL1 P32/RxD0/IrRxD/SDA1 P31/TxD1 P30/TxD0/IrTxD					
Port 4	General I/O port also functioning as A/D converter analog inputs and interrupt inputs	P47/$\overline{\text{IRQ7-B}}$/AN7_0 P46/$\overline{\text{IRQ6-B}}$/AN6_0 P45/$\overline{\text{IRQ5-B}}$/AN5_0 P44/$\overline{\text{IRQ4-B}}$/AN4_0 P43/$\overline{\text{IRQ3-B}}$/AN3_0 P42/$\overline{\text{IRQ2-B}}$/AN2_0 P41/$\overline{\text{IRQ1-B}}$/AN1_0 P40/$\overline{\text{IRQ0-B}}$/AN0_0					Schmitt-triggered inputs when used as IRQ inputs.

Port	Description	Mode 1	Mode 2	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port 5	General I/O port also functioning as interrupt inputs, A/D converter inputs, SCI I/Os, PPG outputs, TPU I/Os, TMR I/Os, I^2C I/Os, and bus control signal I/Os	P53/$\overline{\text{IRQ3-A}}$ /$\overline{\text{ADTRG0-A}}$					Schmitt-triggered inputs when used as IRQ inputs. Only P50 and P51 are Schmitt-triggered inputs when used as I^2C inputs. Open-drain output capability. Only P50 to P52 are Schmitt-triggered inputs when used as TPU inputs. Only P50 and P51 are Schmitt-triggered inputs when used as 8-bit timer inputs. P50 and P51 have 5-V tolerance.
		P52/$\overline{\text{BACK-B}}$/$\overline{\text{IRQ2-A}}$ /PO4-B/TIOCA4-B/TMO0-B/ SCK2				P52/$\overline{\text{IRQ2-A}}$/PO4-B/ TIOCA4-B/TMO0-B/ SCK2	
		P51/$\overline{\text{BREQ-B}}$/$\overline{\text{IRQ1-A}}$/PO2-B/TIOCC3-B/TMCI0-B/ RxD2/SCL3				P51/$\overline{\text{IRQ1-A}}$/PO2-B/ TIOCC3-B/TMCI0-B/ RxD2/SCL3	
		P50/$\overline{\text{BREQO-B}}$/$\overline{\text{IRQ0-A}}$/PO0-B/TIOCA3-B/TMRI0-B/ TxD2/SDA3				P50/$\overline{\text{IRQ0-A}}$/PO0-B/ TIOCA3-B/TMRI0-B/ TxD2/SDA3	
Port 8	General I/O port also functioning as PPG outputs, TPU I/Os, TMR I/Os, and SCI I/Os	P85/PO5-B/TIOCB4-B/TMO1-B/SCK3 P83/PO3-B/TIOCD3-B/ TMCI1-B/RxD3 P81/PO1-B/TIOCB3-B/ TMRI1-B/TxD3					Open-drain output capability. Only P81, P83, and P85 are Schmitt-triggered inputs when used as TPU inputs. Only P81 and P83 are Schmitt-triggered inputs when used as 8-bit timer inputs. P81 and P83 have 5-V tolerance.
Port 9	Dedicated input port also functioning as A/D converter analog inputs and D/A converter analog outputs	P95/AN13_1/DA3 P94/AN12_1/DA2					

Port	Description	Mode 1	Mode 2	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port A	General I/O port also functioning as address outputs, SSU I/Os, SCI I/Os, and bus control signal outputs	PA7/A23/CS7/ IRQ7-A/ SSO0-B		PA7/A23/CS7/IRQ7-A/ SSO0-B		PA7/IRQ7-A/SSO0-B	Only PA4 to PA7 are Schmitt-triggered inputs when used as IRQ inputs. Built-in input pull-up MOS. Open-drain output capability.
		PA6/A22/ IRQ6-A/ SSI0-B		PA6/A22/IRQ6-A/SSI0-B		PA6/IRQ6-A/SSI0-B	
		PA5/A21/ IRQ5-A/ SSCK0-B		PA5/A21/IRQ5-A/SSCK0-B		PA5/IRQ5-A/SSCK0-B	
		A20/ IRQ4-A		PA4/A20/IRQ4-A/SCS0-B		PA4/IRQ4-A/SCS0-B	
		A19		PA3/A19/SCK4-B		PA3/SCK4-B	
		A18		PA2/A18/RxD4-B		PA2/RxD4-B	
		A17		PA1/A17/TxD4-B		PA1/TxD4-B	
		A16		PA0/A16		PA0	
Port B	General I/O port also functioning as address outputs and TPU I/Os	A15		PB7/A15		PB7/TIOCB8/TCLKH	Built-in input pull-up MOS. Schmitt-triggered inputs when used as TPU inputs. Open-drain output capability.
		A14		PB6/A14		PB6/TIOCA8	
		A13		PB5/A13		PB5/TIOCB7/TCLKG	
		A12		PB4/A12		PB4/TIOCA7	
		A11		PB3/A11		PB3/TIOCD6/TCLKF	
		A10		PB2/A10		PB2/TIOCC6/TCLKE	
		A9		PB1/A9		PB1/TIOCB6	
		A8		PB0/A8		PB0/TIOCA6	
Port C	General I/O port also functioning as address outputs and TPU I/Os	A7		PC7/A7		PC7/TIOCB11	Built-in input pull-up MOS. Schmitt-triggered inputs when used as TPU inputs. Open-drain output capability.
		A6		PC6/A6		PC6/TIOCA11	
		A5		PC5/A5		PC5/TIOCB10	
		A4		PC4/A4		PC4/TIOCA10	
		A3		PC3/A3		PC3/TIOCD9	
		A2		PC2/A2		PC2/TIOCC9	
		A1		PC1/A1		PC1/TIOCB9	
		A0		PC0/A0		PC0/TIOCA9	
Port D	General I/O port also functioning as data I/Os	D15				PD7	Built-in input pull-up MOS. Schmitt-triggered inputs when used as TPU inputs. Open-drain output capability.
		D14				PD6	
		D13				PD5	
		D12				PD4	
		D11				PD3	
		D10				PD2	
		D9				PD1	
		D8				PD0	

Port	Description	Mode 1	Mode 2	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port E	General I/O port also functioning as data I/Os	PE7/D7	PE7/D7			PE7	Built-in input pull-up MOS. Open-drain output capability.
		PE6/D6	PE6/D6			PE6	
		PE5/D5	PE5/D5			PE5	
		PE4/D4	PE4/D4			PE4	
		PE3/D3	PE3/D3			PE3	
		PE2/D2	PE2/D2			PE2	
		PE1/D1	PE1/D1			PE1	
		PE0/D0	PE0/D0			PE0	
Port F	General I/O port also functioning as, bus control signal I/Os, SSU I/Os, and A/D converter inputs	PF7/ϕ				PF7/ϕ	Open-drain output capability.
		PF6/$\overline{\text{AS}}$/$\overline{\text{AH}}$				PF6	
		$\overline{\text{RD}}$				PF5	
		$\overline{\text{HWR}}$				PF4	
		PF3/$\overline{\text{LWR}}$/SSO0-C				PF3/SSO0-C	
		PF2/$\overline{\text{CS6}}$/$\overline{\text{LCAS}}$/SSI0-C				PF2/SSI0-C	
		PF1/$\overline{\text{CS5}}$/$\overline{\text{UCAS}}$/SSCK0-C				PF1/SSCK0-C	
		PF0/$\overline{\text{WAIT-A}}$/$\overline{\text{OE-A}}$/ADTRG0-B/$\overline{\text{SCS0-C}}$				PF0/$\overline{\text{ADTRG0-B}}$/ $\overline{\text{SCS0-C}}$	
Port G	General I/O port also functioning as bus control signal I/Os	PG6/$\overline{\text{BREQ-A}}$				PG6	Open-drain output capability.
		PG5/$\overline{\text{BACK-A}}$				PG5	
		PG4/$\overline{\text{BREQO-A}}$/$\overline{\text{CS4}}$				PG4	
		PG3/$\overline{\text{CS3}}$/$\overline{\text{RAS3}}$				PG3	
		PG2/$\overline{\text{CS2}}$/$\overline{\text{RAS2}}$				PG2	
		PG1/$\overline{\text{CS1}}$				PG1	
		PG0/$\overline{\text{CS0}}$				PG0	

10.1 Port 1

Port 1 is an 8-bit I/O port that also has other functions. Port 1 has the following registers. For the port function control registers, refer to section 10.18, Port Function Control Registers.

- Port 1 data direction register (P1DDR)
- Port 1 data register (P1DR)
- Port 1 register (PORT1)
- Port 1 open drain control register (P1ODR)
- Port function control register 5 (PFCR5)

10.1.1 Port 1 Data Direction Register (P1DDR)

The individual bits of P1DDR specify input or output for the pins of port 1. P1DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	When a pin function is specified as a general purpose I/O, setting this bit to 1 makes the corresponding pin an output port, while clearing this bit to 0 makes the corresponding pin an input port.
6	P16DDR	0	W	
5	P15DDR	0	W	
4	P14DDR	0	W	
3	P13DDR	0	W	
2	P12DDR	0	W	
1	P11DDR	0	W	
0	P10DDR	0	W	

10.1.2 Port 1 Data Register (P1DR)

P1DR stores output data for the port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
6	P16DR	0	R/W	
5	P15DR	0	R/W	
4	P14DR	0	R/W	
3	P13DR	0	R/W	
2	P12DR	0	R/W	
1	P11DR	0	R/W	
0	P10DR	0	R/W	

10.1.3 Port 1 Register (PORT1)

PORT1 shows the pin states of port 1. PORT1 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	—*	R	If this register is read while a P1DDR bit is set to 1, the corresponding P1DR value is read. If this register is read while a P1DDR bit is cleared to 0, the corresponding pin state is read.
6	P16	—*	R	
5	P15	—*	R	
4	P14	—*	R	
3	P13	—*	R	
2	P12	—*	R	
1	P11	—*	R	
0	P10	—*	R	

Note: * Determined by the states of pins P17 to P10.

10.1.4 Port 1 Open Drain Control Register (P1ODR)

P1ODR specifies the output type of each port 1 pin.

Bit	Bit Name	Initial Value	R/W	Description
7	P17ODR	0	R/W	Setting a P1ODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a P1ODR bit to 0 makes the corresponding pin a CMOS output pin.
6	P16ODR	0	R/W	
5	P15ODR	0	R/W	
4	P14ODR	0	R/W	
3	P13ODR	0	R/W	
2	P12ODR	0	R/W	
1	P11ODR	0	R/W	
0	P10ODR	0	R/W	

10.1.5 Pin Functions

Port 1 pins also function as the pins for PPG outputs, TPU I/Os, EXDMAC I/Os (H8S/2456, H8S/2456R), SSU I/Os, and DMAC I/Os (H8S/2454). The correspondence between the register specification and the pin functions is shown below.

(1) Pin Functions of H8S/2456 Group and H8S/2456R Group

- P17/PO15/TIOCB2/TCLKD/$\overline{\text{EDRAK3}}$/$\overline{\text{SCS0-A}}$

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 2 settings (by bits MD3 to MD0 in TMDR_2, bits IOB3 to IOB0 in TIOR_2, and bits CCLR1 and CCLR0 in TCR_2), bits TPSC2 to TPSC0 in TCR_0 and TCR_5, bit NDER15 in NDERH of PPG, bit EDRAKE in EDMDR_3 of EXDMAC, bits MSS, CSS1, and CSS0 in SSCRH and bit SSUMS in SSCRL of SSU, bits SCS0S1 and SCS0S0 in PFCR5, and bit P17DDR.

- Modes 1, 2, 4, and 7 (EXPE = 1)

SSU settings	Can be used as I/O port					Input state	Output state
EDRAKE	0				1	—	
TPU channel 2 settings	(1) in table below	(2) in table below			—	—	
P17DDR	—	0	1	1	—	0	—
NDER15	—	—	0	1	—	—	
Pin function	TIOCB2 output	P17 input	P17 output	PO15 output	$\overline{\text{EDRAK3}}$ output	$\overline{\text{SCS0-A}}$ input*3	$\overline{\text{SCS0-A}}$ output*4
		TIOCB2 input*1					
	TCLKD input*2						

- Mode 7 (EXPE = 0)

SSU settings	Can be used as I/O port				Input state	Output state
EDRAKE	0				—	
TPU channel 2 settings	(1) in table below	(2) in table below			—	
P17DDR	—	0	1	1	0	—
NDER15	—	—	0	1	—	
Pin function	TIOCB2 output	P17 input	P17 output	PO15 output	$\overline{\text{SCS0-A}}$ input*3	$\overline{\text{SCS0-A}}$ output*4
		TIOCB2 input*1				
	TCLKD input*2					

Notes: 1. TIOCB2 input when MD3 to MD0 = B'0000 or B'01×× and IOB3 = 1.
2. TCLKD input when the setting for either TCR_0 or TCR_5 is TPSC2 to TPSC0 = B'111. TCLKD input when channels 2 and 4 are set to phase counting mode.
3. $\overline{\text{SCSO-A}}$ input when SCS0S1 and SCS0S0 = B'00 in PFCR5, and SSUMS, MSS, CSS1, and CSS0 = B'00××, B'0101, or B'0110. Do not set up for TPU or EXDMAC outputs with $\overline{\text{SCSO-A}}$ input.
4. $\overline{\text{SCSO-A}}$ output when SCS0S1 and SCS0S0 = B'00 in PFCR5, and SSUMS, MSS, CSS1, and CSS0 = B'011×.

TPU channel 2 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

	$\overline{\text{SCS}}$ pin settings					
SSUMS	0					1
MSS	0	1				×
CSS1	×	0		1		×
CSS0	×	0	1	0	1	×
Pin state	Input	—	Input	Automatic I/O	Output	—

Legend:

×: Don't care

—: Pin is not used by the SSU (can be used as I/O port)

- P16/PO14/TIOCA2/$\overline{\text{EDRAK2}}$/SSCK0-A

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 2 settings (by bits MD3 to MD0 in TMDR_2, bits IOA3 to IOA0 in TIOR_2, and bits CCLR1 and CCLR0 in TCR_2), bit NDER14 in NDERH of PPG, bit EDRAKE in EDMDR_2 of EXDMAC, bits MSS and SCKS in SSCRH and bit SSUMS in SSCRL of SSU, bits SSCK0S1 and SSCK0S0 in PFCR5, and bit P16DDR.

- Modes 1, 2, 4, and 7 (EXPE = 1)

SSU settings	Can be used as I/O port					Input state	Output state
EDRAKE	0				1	—	
TPU channel 2 settings	(1) in table below	(2) in table below			—	—	
P16DDR	—	0	1	1	—	0	—
NDER14	—	—	0	1	—	—	
Pin function	TIOCA2 output	P16 input	P16 output	PO14 output	$\overline{\text{EDRAK2}}$ output	SSCK0-A input*3	SSCK0-A output*4
		TIOCA2 input*1					

- Mode 7 (EXPE = 0)

SSU settings	Can be used as I/O port				Input state	Output state
EDRAKE	0				—	
TPU channel 2 settings	(1) in table below	(2) in table below			—	
P16DDR	—	0	1	1	0	—
NDER14	—	—	0	1	—	
Pin function	TIOCA2 output	P16 input	P16 output	PO14 output	SSCK0-A input*3	SSCK0-A output*4
		TIOCA2 input*1				

TPU channel 2 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00		
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA2 input when MD3 to MD0 = B'0000 or B'01×× and IOA3 = 1.

2. TIOCB2 output disabled.

3. SSCK0-A input when SSCK0S1 and SSCK0S0 = B'00 in PFCR5, and SSUMS, MSS, and SCKS = B'001 or B'101. Do not set up for TPU or EXDMAC outputs with SSCK0-A input.

4. SSCK0-A output when SSCK0S1 and SSCK0S0 = B'00 in PFCR5, and SSUMS, MSS, and SCKS = B'×11.

	SSCK pin settings							
SSUMS	0				1			
MSS	0		1		0		1	
SCKS	0	1	0	1	0	1	0	1
Pin state	—	Input	—	Output	—	Input	—	Output

Legend:

—: Pin is not used by the SSU (can be used as I/O port)

- P15/PO13/TIOCB1/TCLKC/SSI0-A

  The pin function is switched as shown below according to the combination of TPU channel 1 settings (by bits MD3 to MD0 in TMDR_1, bits IOB3 to IOB0 in TIOR_1, and bits CCLR1 and CCLR0 in TCR_1), bits TPSC2 to TPSC0 in TCR_0, TCR_2, TCR_4, and TCR_5, bit NDER13 in NDERH of PPG, bits MSS and BIDE in SSCRH, bit SSUMS in SSCRL, and bits TE and RE in SSER of SSU, bits SSI0S1 and SSI0S0 in PFCR5, and bit P15DDR.

SSU settings	Can be used as I/O port				Input state	Output state
TPU channel 1 settings	(1) in table below	(2) in table below			—	
P15DDR	—	0	1	1	0	—
NDER13	—	—	0	1	—	
Pin function	TIOCB1 output	P15 input	P15 output	PO13 output	SSI0-A input*3	SSI0-A output*4
		TIOCB1 input*1				
	TCLKC input*2					

Notes: 1. TIOCB1 input when MD3 to MD0 = B'0000 or B'01×× and IOB3 to IOB0 = B'10××.
2. TCLKC input when the setting for either TCR_0 or TCR_2 is TPSC2 to TPSC0 = B'111, or when the setting for either TCR_4 or TCR_5 is TPSC2 to TPSC0 = B'101. TCLKC input when phase counting mode is set for channels 2 and 4.
3. SSI0-A input when SSI0S1 and SSI0S0 = B'00 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'001×1 or B'10××1. Do not set up for TPU output with SSI0-A input.
4. SSI0-A output when SSI0S1 and SSI0S0 = B'00 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'0001×.

TPU channel 1 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don't care

	SSI pin settings															
SSUMS	0						0				1					
BIDE	0						1				0					
MSS	0			1			0		1		0			1		
TE	0	1		0	1		0	1	0	1	0	1		0	1	
RE	1	0	1	1	0	1	1	0	1	0	1	0	1	1	0	1
Pin state	—	Output	Output	Input	—	Input	—	—	—	—	Input	—	Input	Input	—	Input

Legend:

—: Pin is not used by the SSU (can be used as I/O port)

- P14/PO12/TIOCA1/SSO0-A

  The pin function is switched as shown below according to the combination of TPU channel 1 settings (by bits MD3 to MD0 in TMDR_1, bits IOA3 to IOA0 in TIOR_1, and bits CCLR1 and CCLR0 in TCR_1), bit NDER12 in NDERH of PPG, bits MSS and BIDE in SSCRH, bit SSUMS in SSCRL, and bits TE and RE in SSER of SSU, bits SSO0S1 and SSO0S0 in PFCR5, and bit P14DDR.

SSU settings	Can be used as I/O port				Input state	Output state
TPU channel 1 settings	(1) in table below	(2) in table below			—	
P14DDR	—	0	1	1	0	—
NDER12	—	—	0	1	—	
Pin function	TIOCA1 output	P14 input	P14 output	PO12 output	SSO0-A input*3	SSO0-A output*4
		TIOCA1 input*1				

TPU channel 1 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA1 input when MD3 to MD0 = B'0000 or B'01×× and IOA3 to IOA0 = B'10××.

2. TIOCB1 output disabled.

3. SSO0-A input when SSO0S1 and SSO0S0 = B'00 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'000×1 or B'01×01. Do not set up for TPU output with SSO0-A input.

4. SSO0-A output when SSO0S1 and SSO0S0 = B'00 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'0011×, B'01×10, or B'10×1×.

	SSO pin settings																	
SSUMS	0						0				1							
BIDE	0						1				0							
MSS	0			1			0		1		0			1				
TE	0	1		0	1		0	1	0	1	0	1		0	1			
RE	1	0	1	1	0	1	1	0	1	0	1	0	1	1	0	1		
Pin state	Input	—	Input	—	Output	Output	Input	Output	Input	Output	—	Output	Output	—	Output	Output		

Legend:

—: Pin is not used by the SSU (can be used as I/O port)

- P13/PO11/TIOCD0/TCLKB

  The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR_0, bits IOD3 to IOD0 in TIORL_0, and bits CCLR2 to CCLR0 in TCR_0), bits TPSC2 to TPSC0 in TCR_0 to TCR_2, bit NDER11 in NDERH of PPG, and bit P13DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P13DDR	—	0	1	
NDER11	—	—	0	1
Pin function	TIOCD0 output	P13 input	P13 output	PO11 output
		TIOCD0 input*1		
	TCLKB input*2			

Notes: 1. TIOCD0 input when MD3 to MD0 = B'0000 and IOD3 to IOD0 = B'10××.
2. TCLKB input when the setting for any of TCR_0 to TCR_2 is TPSC2 to TPSC0 = B'101. TCLKB input when phase counting mode is set for channels 1 and 5.

TPU channel 0 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOD3 to IOD0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don’t care

- P12/PO10/TIOCC0/TCLKA

  The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR_0, bits IOC3 to IOC0 in TIORL_0, and bits CCLR2 to CCLR0 in TCR_0), bits TPSC2 to TPSC0 in TCR_0 to TCR_5, bit NDER10 in NDERH of PPG, and bit P12DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P12DDR	—	0	1	
NDER10	—	—	0	1
Pin function	TIOCC0 output	P12 input	P12 output	PO10 output
		TIOCC0 input*1		
	TCLKA input*2			

TPU channel 0 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001×	B'0010	B'0011	
IOC3 to IOC0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM*3 mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCC0 input when MD3 to MD0 = B'0000 and IOC3 to IOC0 = B'10××.

2. TCLKA input when the setting for any of TCR_0 to TCR_5 is TPSC2 to TPSC0 = B'100. TCLKA input when phase counting mode is set for channels 1 and 5.

3. TIOCD0 output disabled. Output disabled and settings (2) effective when BFA = 1 or BFB = 1 in TMDR_0.

- P11/PO9/TIOCB0

  The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR_0, bits IOB3 to IOB0 in TIORH_0, and bits CCLR2 to CCLR0 in TCR_0), bit NDER9 in NDERH of PPG, and bit P11DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P11DDR	—	0	1	
NDER9	—	—	0	1
Pin function	TIOCB0 output	P11 input	P11 output	PO9 output
		TIOCB0 input*		

Note: * TIOCB0 input when MD3 to MD0 = B'0000 and IOB3 to IOB0 = B'10××.

TPU channel 0 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

×: Don't care

- P10/PO8/TIOCA0

  The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR_0, bits IOA3 to IOA0 in TIORH_0, and bits CCLR2 to CCLR0 in TCR_0), bit NDER8 in NDERH of PPG, and bit P10DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P10DDR	—	0	1	
NDER8	—	—	0	1
Pin function	TIOCA0 output	P10 input	P10 output	PO8 output
		TIOCA0 input*1		

TPU channel 0 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA0 input when MD3 to MD0 = B'0000 and IOA3 to IOA0 = B'10××.

2. TIOCB0 output disabled.

(2) Pin Functions of H8S/2454 Group

- P17/PO15/TIOCB2/TCLKD/$\overline{\text{SCS0-A}}$

  The pin function is switched as shown below according to the combination of the TPU channel 2 settings (by bits MD3 to MD0 in TMDR_2, bits IOB3 to IOB0 in TIOR_2, and bits CCLR1 and CCLR0 in TCR_2), bits TPSC2 to TPSC0 in TCR_0 and TCR_5, bit NDER15 in NDERH of PPG, bits MSS, CSS1, and CSS0 in SSCRH and bit SSUMS in SSCRL of SSU, bits SCS0S1 and SCS0S0 in PFCR5, and bit P17DDR.

<table>
<tr><td>SSU settings</td><td colspan="4">Can be used as I/O port</td><td>Input state</td><td>Output state</td></tr>
<tr><td>TPU channel 2 settings</td><td>(1) in table below</td><td colspan="3">(2) in table below</td><td colspan="2">—</td></tr>
<tr><td>P17DDR</td><td>—</td><td>0</td><td>1</td><td>1</td><td>0</td><td>—</td></tr>
<tr><td>NDER15</td><td>—</td><td>—</td><td>0</td><td>1</td><td colspan="2">—</td></tr>
<tr><td rowspan="2">Pin function</td><td rowspan="2">TIOCB2 output</td><td>P17 input</td><td>P17 output</td><td>PO15 output</td><td rowspan="2">$\overline{\text{SCS0-A}}$ input*3</td><td rowspan="2">$\overline{\text{SCS0-A}}$ output*4</td></tr>
<tr><td colspan="3">TIOCB2 input*1</td></tr>
<tr><td colspan="4">TCLKD input*2</td></tr>
</table>

Notes: 1. TIOCB2 input when MD3 to MD0 = B'0000 or B'01×× and IOB3 = 1.
2. TCLKD input when the setting for either TCR_0 or TCR_5 is TPSC2 to TPSC0 = B'111. TCLKD input when channels 2 and 4 are set to phase counting mode.
3. $\overline{\text{SCS0-A}}$ input when SCS0S1 and SCS0S0 = B'00 in PFCR5, and SSUMS, MSS, CSS1, and CSS0 = B'00××, B'0101, or B'0110. Do not set up for TPU output with $\overline{\text{SCS0-A}}$ input.
4. $\overline{\text{SCS0-A}}$ output when SCS0S1 and SCS0S0 = B'00 in PFCR5, and SSUMS, MSS, CSS1, and CSS0 = B'011×.

TPU channel 2 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don't care

	$\overline{SCS}$ pin settings					
SSUMS	0					1
MSS	0	1				×
CSS1	×	0		1		×
CSS0	×	0	1	0	1	×
Pin state	Input	—	Input	Automatic I/O	Output	—

Legend:
×: Don't care
—: Pin is not used by the SSU (can be used as I/O port)

- P16/PO14/TIOCA2/SSCK0-A

  The pin function is switched as shown below according to the combination of the TPU channel 2 settings (by bits MD3 to MD0 in TMDR_2, bits IOA3 to IOA0 in TIOR_2, and bits CCLR1 and CCLR0 in TCR_2), bit NDER14 in NDERH of PPG, bits MSS and SCKS in SSCRH and bit SSUMS in SSCRL of SSU, bits SSCK0S1 and SSCK0S0 in PFCR5, and bit P16DDR.

SSU settings	Can be used as I/O port				Input state	Output state
TPU channel 2 settings	(1) in table below	(2) in table below			—	
P16DDR	—	0	1	1	0	1
NDER14	—	—	0	1	—	
Pin function	TIOCA2 output	P16 input	P16 output	PO14 output	SSCK0-A input*3	SSCK0-A output*4
		TIOCA2 input*1				

TPU channel 2 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00		
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA2 input when MD3 to MD0 = B'0000 or B'01×× and IOA3 = 1.

2. TIOCB2 output disabled.

3. SSCK0-A input when SSCK0S1 and SSCK0S0 = B'00 in PFCR5, and SSUMS, MSS, and SCKS = B'001 or B'101. Do not set up for TPU output with SSCK0-A input.

4. SSCK0-A output when SSCK0S1 and SSCK0S0 = B'00 in PFCR5, and SSUMS, MSS, and SCKS = B'×11.

	SSCK pin settings							
SSUMS	0				1			
MSS	0		1		0		1	
SCKS	0	1	0	1	0	1	0	1
Pin state	—	Input	—	Output	—	Input	—	Output

Legend:
—: Pin is not used by the SSU (can be used as I/O port)

- P15/$\overline{\text{DACK1}}$/PO13/TIOCB1/TCLKC/SSI0-A

  The pin function is switched as shown below according to the combination of bit SAE1 in DMABCRH of DMAC, TPU channel 1 settings (by bits MD3 to MD0 in TMDR_1, bits IOB3 to IOB0 in TIOR_1, and bits CCLR1 and CCLR0 in TCR_1), bits TPSC2 to TPSC0 in TCR_0, TCR_2, TCR_4, and TCR_5, bit NDER13 in NDERH of PPG, bits MSS and BIDE in SSCRH, bit SSUMS in SSCRL, and bits TE and RE in SSER of SSU, bits SSI0S1 and SSI0S0 in PFCR5, and bit P15DDR.

SSU settings						Input state	Output state
SAE1	0				1	—	
TPU channel 1 settings	(1) In table below	(2) In table below			—	—	
P15DDR	—	0	1	1	—	0	—
NDER13	—	—	0	1	—	—	
Pin function	TIOCB1 output	P15 input	P15 output	PO13 output	$\overline{\text{DACK1}}$ output	SSI0-A input*3	SSI0-A output*4
		TIOCB1 input*1					
	TCLKC input*2						

Notes: 1. TIOCB1 input when MD3 to MD0 = B'0000 or B'01×× and IOB3 to IOB0 = B'10××.
2. TCLKC input when the setting for either TCR_0 or TCR_2 is TPSC2 to TPSC0 = B'111, or when the setting for either TCR_4 or TCR_5 is TPSC2 to TPSC0 = B'101. TCLKC input when phase counting mode is set for channels 2 and 4.
3. SSI0-A input when SSI0S1 and SSI0S0 = B'00 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'001×1 or B'10××1. Do not set up for TPU output with SSCK0-A input.
4. SSI0-A output when SSI0S1 and SSI0S0 = B'00 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'0001×.

TPU channel 1 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don't care

	SSI pin settings															
SSUMS	0						0				1					
BIDE	0						1				0					
MSS	0			1			0		1		0			1		
TE	0	1		0	1		0	1	0	1	0	1		0	1	
RE	1	0	1	1	0	1	1	0	1	0	1	0	1	1	0	1
Pin state	—	Output	Output	Input	—	Input	—	—	—	—	Input	—	Input	Input	—	Input

Legend:
—: Pin is not used by the SSU (can be used as I/O port)

- P14/$\overline{\text{DACK0}}$/PO12/TIOCA1/SSO0-A

  The pin function is switched as shown below according to the combination of bit SAE0 in DMABCRH of DMAC, TPU channel 1 settings (by bits MD3 to MD0 in TMDR_1, bits IOA3 to IOA0 in TIOR_1, and bits CCLR1 and CCLR0 in TCR_1), bit NDER12 in NDERH of PPG, bits MSS and BIDE in SSCRH, bit SSUMS in SSCRL, and bits TE and RE in SSER of SSU, bits SSO0S1 and SSO0S0 in PFCR5, and bit P14DDR.

SSU settings	Can be used as I/O port					Input state	Output state
SAE0	0				1	—	
TPU channel 1 settings	(1) in table below	(2) in table below			—	—	
P14DDR	—	0	1	1	—	0	—
NDER12	—	—	0	1	—	—	
Pin function	TIOCA1 output	P14 input	P14 output	PO12 output	$\overline{\text{DACK0}}$ output	SSO0-A input*3	SSO0-A output*4
		TIOCA1 input*1					

TPU channel 1 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA1 input when MD3 to MD0 = B'0000 or B'01×× and IOA3 to IOA0 = B'10××.

2. TIOCB1 output disabled.

3. SSO0-A input when SSO0S1 and SSO0S0 = B'00 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'000×1 or B'01×01. Do not set up for TPU or DMAC output with SSO0-A input.

4. SSO0-A output when SSO0S1 and SSO0S0 = B'00 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'0011×, B'01×10, or B'10×1×.

	SSO pin settings															
SSUMS	0						0				1					
BIDE	0						1				0					
MSS	0			1			0		1		0			1		
TE	0	1		0	1		0	1	0	1	0	1		0	1	
RE	1	0	1	1	0	1	1	0	1	0	1	0	1	1	0	1
Pin state	Input	—	Input	—	Output	Output	Input	Output	Input	Output	—	Output	Output	—	Output	Output

Legend:

—: Pin is not used by the SSU (can be used as I/O port)

- P13/$\overline{\text{TEND1}}$/PO11/TIOCD0/TCLKB

  The pin function is switched as shown below according to the combination of bit TEE1 in DMATCR of DMAC, TPU channel 0 settings (by bits MD3 to MD0 in TMDR_0, bits IOD3 to IOD0 in TIORL_0, and bits CCLR2 to CCLR0 in TCR_0), bits TPSC2 to TPSC0 in TCR_0 to TCR_2, bit NDER11 in NDERH of PPG, and bit P13DDR.

TEE1	0				1
TPU channel 0 settings	(1) in table below	(2) in table below			—
P13DDR	—	0	1	1	—
NDER11	—	—	0	1	—
Pin function	TIOCD0 output	P13 input	P13 output	PO11 output	$\overline{\text{TEND1}}$ output
		TIOCD0 input*1			
	TCLKB input*2				

Notes: 1. TIOCD0 input when MD3 to MD0 = B'0000 and IOD3 to IOD0 = B'10××.
2. TCLKB input when the setting for any of TCR_0 to TCR_2 is TPSC2 to TPSC0 = B'101. TCLKB input when phase counting mode is set for channels 1 and 5.

TPU channel 0 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOD3 to IOD0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don't care

- P12/$\overline{\text{TEND0}}$/PO10/TIOCC0/TCLKA

  The pin function is switched as shown below according to the combination of bit TEE0 in DMATCR of DMAC, TPU channel 0 settings (by bits MD3 to MD0 in TMDR_0, bits IOC3 to IOC0 in TIORL_0, and bits CCLR2 to CCLR0 in TCR_0), bits TPSC2 to TPSC0 in TCR_0 to TCR_5, bit NDER10 in NDERH of PPG, and bit P12DDR.

TEE0	0				1
TPU channel 0 settings	(1) in table below	(2) in table below			—
P12DDR	—	0	1	1	—
NDER10	—	—	0	1	—
Pin function	TIOCC0 output	P12 input	P12 output	PO10 output	$\overline{\text{TEND0}}$ output
		TIOCC0 input*1			
	TCLKA input*2				

TPU channel 0 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001×	B'0010	B'0011	
IOC3 to IOC0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM*3 mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCC0 input when MD3 to MD0 = B'0000 and IOC3 to IOC0 = B'10××.

2. TCLKA input when the setting for any of TCR_0 to TCR_5 is TPSC2 to TPSC0 = B'100. TCLKA input when phase counting mode is set for channels 1 and 5.

3. TIOCD0 output disabled. Output disabled and settings (2) effective when BFA = 1 or BFB = 1 in TMDR_0.

- P11/$\overline{\text{DREQ1}}$/PO9/TIOCB0

  The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR_0, bits IOB3 to IOB0 in TIORH_0, and bits CCLR2 to CCLR0 in TCR_0), bit NDER9 in NDERH of PPG, bit USBDRQE in PFCR3 and bit P11DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P11DDR	—	0	1	
NDER9	—	—	0	1
Pin function	TIOCB0 output	P11 input	P11 output	PO9 output
		TIOCB0 input*1		
	$\overline{\text{DREQ1}}$ input*2			

Note: 1. TIOCB0 input when MD3 to MD0 = B'0000 and IOB3 to IOB0 = B'10××.
2. $\overline{\text{DREQ1}}$ input when bit USBDRQE in PFCR3 is 0.

TPU channel 0 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don't care

- P10/$\overline{\text{DREQ0}}$/PO8/TIOCA0

  The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR_0, bits IOA3 to IOA0 in TIORH_0, and bits CCLR2 to CCLR0 in TCR_0), bit NDER8 in NDERH of PPG, bit USBDRQE in PFCR3, and bit P10DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P10DDR	—	0	1	
NDER8	—	—	0	1
Pin function	TIOCA0 output	P10 input	P10 output	PO8 output
		TIOCA0 input*1		
	$\overline{\text{DREQ0}}$ input*3			

TPU channel 0 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA0 input when MD3 to MD0 = B'0000 and IOA3 to IOA0 = B'10××.

2. TIOCB0 output disabled.

3. $\overline{\text{DREQ0}}$ input when bit USBDRQE in PFCR3 is 0.

10.2 Port 2

Port 2 is an 8-bit I/O port that also has other functions. Port 2 has the following registers. For the port function control registers, refer to section 10.18, Port Function Control Registers.

- Port 2 data direction register (P2DDR)
- Port 2 data register (P2DR)
- Port 2 register (PORT2)
- Port 2 open drain control register (P2ODR)
- Port function control register 3 (PFCR3)

10.2.1 Port 2 Data Direction Register (P2DDR)

The individual bits of P2DDR specify input or output for the pins of port 2. P2DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DDR	0	W	When a pin function is specified as a general purpose I/O, setting this bit to 1 makes the corresponding pin an output port, while clearing this bit to 0 makes the corresponding pin an input port.
6	P26DDR	0	W	
5	P25DDR	0	W	
4	—	0	—	Bits 4 to 0 are reserved.
3	—	0	—	
2	—	0	—	
1	—	0	—	
0	P20DDR	0	W	When a pin function is specified as a general purpose I/O, setting this bit to 1 makes the corresponding pin an output port, while clearing this bit to 0 makes the corresponding pin an input port.

10.2.2 Port 2 Data Register (P2DR)

P2DR stores output data for the port 2 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
6	P26DR	0	R/W	
5	P25DR	0	R/W	
4	—	0	—	Bits 4 to 1 are reserved. These bits are read as 0. When written, the initial value should be written to.
3	—	0	—	
2	—	0	—	
1	—	0	—	
0	P20DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.

10.2.3 Port 2 Register (PORT2)

PORT2 shows the pin states of port 2. PORT2 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P27	—*	R	If this register is read while a P2DDR bit is set to 1, the corresponding P2DR value is read. If this register is read while a P2DDR bit is cleared to 0, the corresponding pin state is read.
6	P26	—*	R	
5	P25	—*	R	
4	—	Undefined	—	Bits 4 to 1 are reserved. The read value is undefined.
3	—	Undefined	—	
2	—	Undefined	—	
1	—	Undefined	—	
0	P20	—*	R	If this register is read while a P2DDR bit is set to 1, the corresponding P2DR value is read. If this register is read while a P2DDR bit is cleared to 0, the corresponding pin state is read.

Note: * Determined by the states of pins P27 to P25 and P20.

10.2.4 Port 2 Open Drain Control Register (P2ODR)

P2ODR specifies the output type of each port 2 pin.

Bit	Bit Name	Initial Value	R/W	Description
7	P27ODR	0	R/W	Setting a P2ODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a P2ODR bit to 0 makes the corresponding pin a CMOS output pin.
6	P26ODR	0	R/W	
5	P25ODR	0	R/W	
4	—	0	—	Bits 4 to 1 are reserved. When written, the initial value should be written to.
3	—	0	—	
2	—	0	—	
1	—	0	—	
0	P20ODR	0	R/W	Setting a P2ODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a P2ODR bit to 0 makes the corresponding pin a CMOS output pin.

10.2.5 Pin Functions

Port 2 pins also function as the pins for PPG outputs, TPU I/Os, interrupt inputs (H8S/2456, H8S/2456R), 8-bit timer I/Os (H8S/2454), I^2C I/Os, USB I/Os, and bus control signal inputs. The correspondence between the register specification and the pin functions is shown below.

(1) Pin Functions of H8S/2456 Group and H8S/2456R Group

- P27/PO7/TIOCB5/$\overline{\text{IRQ15-B}}$/SCL2

  The pin function is switched as shown below according to the combination of the TPU channel 5 settings (by bits MD3 to MD0 in TMDR_5, bits IOB3 to IOB0 in TIOR_5, and bits CCLR1 and CCLR0 in TCR_5), bit NDER7 in NDERL of PPG, bit ICE in ICCRA_2 of I^2C, bit P27DDR, and bit ITS15 in ITSR of the interrupt controller.

ICE	0				1
TPU channel 5 settings	(1) in table below	(2) in table below			—
P27DDR	—	0	1	1	—
NDER7	—	—	0	1	—
Pin function	TIOCB5 output	P27 input	P27 output	PO7 output	SCL2 I/O
		TIOCB5 input*1			
	$\overline{\text{IRQ15-B}}$ interrupt input*2				

Notes: 1. TIOCB5 input when MD3 to MD0 = B'0000 or B'01×× and IOB3 = 1.
2. $\overline{\text{IRQ15-B}}$ input when the ITS15 bit in ITSR is 1.

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don't care

- P26/PO6/TIOCA5/$\overline{\text{IRQ14-B}}$/SDA2/$\overline{\text{ADTRG1}}$

  The pin function is switched as shown below according to the combination of the TPU channel 5 settings (by bits MD3 to MD0 in TMDR_5, bits IOA3 to IOA0 in TIOR_5, and bits CCLR1 and CCLR0 in TCR_5), bit NDER6 in NDERL of PPG, bits TRGS1, TRGS0, and EXTRGS in ADCR_1 of ADC, bit ICE in ICCRA_2 of I^2C, bit P26DDR, and bit ITS14 in ITSR of the interrupt controller.

ICE	0				1
TPU channel 5 settings	(1) in table below	(2) in table below			—
P26DDR	—	0	1	1	—
NDER6	—	—	0	1	—
Pin function	TIOCA5 output	P26 input	P26 output	PO6 output	SDA2 I/O
		TIOCA5 input*1			
	$\overline{\text{IRQ14-B}}$ interrupt input*2				
	$\overline{\text{ADTRG1}}$ input*4				

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*3 mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA5 input when MD3 to MD0 = B'0000 or B'01×× and IOA3 = 1.

2. $\overline{\text{IRQ14-B}}$ input when the ITS14 bit in ITSR is 1.
3. TIOCB5 output disabled.
4. $\overline{\text{ADTRG1}}$ input when EXTRGS = 0 and TRGS1 = TRGS0 = 1.

- P25/PO5-A/TIOCB4-A/$\overline{\text{IRQ13}}$-B/$\overline{\text{WAIT}}$-B/VBUS

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit WAITE in BCR of the bus controller, TPU channel 4 settings by bits MD3 to MD0 in TMDR_4, bits IOB3 to IOB0 in TIOR_4, and bits CCLR1 and CCLR0 in TCR_4, bit NDER5 in NDERL of PPG, bits PPGS, TPUS, and TMRS in PFCR3, bit WAITS in PFCR4, bit P25DDR, and bit ITS13 in ITSR of the interrupt controller.

- Modes 1, 2, 4, and 7 (EXPE = 1)

<table>
<tr><td>WAITE</td><td colspan="4">0</td><td>1</td></tr>
<tr><td>TPU channel 4 settings</td><td>(1) in table below</td><td colspan="3">(2) in table below</td><td>—</td></tr>
<tr><td>P25DDR</td><td>—</td><td>0</td><td>1</td><td>1</td><td>—</td></tr>
<tr><td>NDER5</td><td>—</td><td>—</td><td>0</td><td>1</td><td>—</td></tr>
<tr><td rowspan="3">Pin function</td><td rowspan="2">TIOCB4-A output*4</td><td>P25 input</td><td>P25 output</td><td>PO5-A output*3</td><td rowspan="2">$\overline{\text{WAIT}}$-B input*5</td></tr>
<tr><td colspan="3">TIOCB4-A input*1*4</td></tr>
<tr><td colspan="5">$\overline{\text{IRQ13}}$-B interrupt input*2</td></tr>
<tr><td></td><td colspan="5">VBUS input</td></tr>
</table>

- Mode 7 (EXPE = 0)

WAITE	—			
TPU channel 4 settings	(1) in table below	(2) in table below		
P25DDR	—	0	0	1
NDER5	—	—	—	0
Pin function	TIOCB4-A output*4	P25 input	P25 output	PO5-A output*3
		TIOCB4-A input*1*4		
	$\overline{\text{IRQ13-B}}$ interrupt input*2			
	VBUS input			

Notes: 1. TIOCB4-A input when MD3 to MD0 = B'0000 or B'01×× and IOB3 to IOB0 = B'10××.
2. $\overline{\text{IRQ13-B}}$ input when the ITS13 bit in ITSR is 1.
3. PO5-A output when the PPGS bit in PFCR3 is 0.
4. TIOCB4-A input/output when the TPUS bit in PFCR3 is 0.
5. $\overline{\text{WAIT-B}}$ input when the WAITS bit in PFCR4 is 1.

TPU channel 4 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don't care

- P20/PO0-A/TIOCA3-A/$\overline{\text{IRQ8-B}}$*[2]/PUPD+

  The pin function is switched as shown below according to the combination of the TPU channel 3 settings (by bits MD3 to MD0 in TMDR_3, bits IOA3 to IOA0 in TIORH_3, and bits CCLR2 to CCLR0 in TCR_3), bit NDER0 in NDERL of PPG, bit PULLUP_E in DMA of USB, bit P20DDR, and bit ITS8 in ITSR of the interrupt controller.

PULLUP-E	0				1
TPU channel 3 settings	(1) In table below	(2) In table below			—
P20DDR	—	0	1		—
NDER0	—	—	0	1	—
Pin function	TIOCA3-A output*[5]	P20 input	P20 output	PO0-A output*[4]	—
		TIOCA3-A input*[1]*[5]			PUPD+ output
	$\overline{\text{IRQ8-B}}$ interrupt input*[2]				

TPU channel 3 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011 B'0101 to B'0111	B'××00	Other than B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM*[3] mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA3-A input when MD3 to MD0 = B'0000 and IOA3 to IOA0 = B'10××.

2. $\overline{\text{IRQ8-B}}$ input when the ITS8 bit in ITSR is 1.
3. TIOCB3 output disabled.
4. PO0-A output when the PPGS bit in PFCR3 is 0.
5. TIOCA3-A input/output when the TPUS bit in PFCR3 is 0.

(2) Pin Functions of H8S/2454 Group

- P27/PO7/TIOCB5/SCL2

  The pin function is switched as shown below according to the combination of the TPU channel 5 settings (by bits MD3 to MD0 in TMDR_5, bits IOB3 to IOB0 in TIOR_5, and bits CCLR1 and CCLR0 in TCR_5), bit NDER7 in NDERL of PPG, bit ICE in ICCRA_2 of I^2C, and bit P27DDR.

ICE	0				1
TPU channel 5 settings	(1) in table below	(2) in table below			—
P27DDR	—	0	1	1	—
NDER7	—	—	0	1	—
Pin function	TIOCB5 output	P27 input	P27 output	PO7 output	SCL2 I/O
		TIOCB5 input*			

Note: * TIOCB5 input when MD3 to MD0 = B'0000 or B'01×× and IOB3 = 1.

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

×: Don't care

- P26/PO6/TIOCA5/SDA2/$\overline{\text{ADTRG1}}$

  The pin function is switched as shown below according to the combination of the TPU channel 5 settings (by bits MD3 to MD0 in TMDR_5, bits IOA3 to IOA0 in TIOR_5, and bits CCLR1 and CCLR0 in TCR_5), bit NDER6 in NDERL of PPG, bits TRGS1, TRGS0, and EXTRGS in ADCR_1 of ADC, bit ICE in ICCRA_2 of I^2C, and bit P26DDR.

ICE	0				1
TPU channel 5 settings	(1) in table below	(2) in table below			—
P26DDR	—	0	1	1	—
NDER6	—	—	0	1	—
Pin function	TIOCA5 output	P26 input	P26 output	PO6 output	SDA2 I/O
		TIOCA5 input*1			
	$\overline{\text{ADTRG1}}$ input*3				

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA5 input when MD3 to MD0 = B'0000 or B'01×× and IOA3 = 1.

2. TIOCB5 output disabled.

3. $\overline{\text{ADTRG1}}$ input when EXTRGS = 0 and TRGS1 = TRGS0 = 1.

- P25/$\overline{\text{WAIT-B}}$/PO5-A/TIOCB4-A/TMO1-A

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit WAITE in BCR of the bus controller, TPU channel 4 settings (by bits MD3 to MD0 in TMDR_4, bits IOB3 to IOB0 in TIOR_4, and bits CCLR1 and CCLR0 in TCR_4), bits OS3 to OS0*[8] in TCSR_1 of the 8-bit timer, bit NDER5 in NDERL of PPG, bits PPGS, TPUS, and TMRS in PFCR3, bit WAITS in PFCR4, and bit P25DDR.

- Modes 1, 2, 4, and 7 (EXPE = 1)

WAITE	0					1
TPU channel 4 settings	(1) In table below		(2) In table below			—
OS3 to OS0	—	All 0			Not all 0	—
P25DDR	—	0	1	1	—	—
NDER5	—	—	0	1	—	—
Pin function	TIOCB4-A output*[3]	P25 input	P25 output	PO5-A output*[2]	TMO1-A output*[4]	$\overline{\text{WAIT-B}}$ input*[5]
		TIOCB4-A input*[1]*[3]				
	VBUS input					

- Mode 7 (EXPE = 0)

WAITE	—				
TPU channel 4 settings	(1) In table below	(2) In table below			
OS3 to OS1	—	All 0			Not all 0
P25DDR	—	0	1	1	—
NDER5	—	—	0	1	—
Pin function	TIOCB4-A output*[3]	P25 input	P25 output	PO5-A output*[2]	TMO1-A output*[4]
		TIOCB4-A input*[1]*[3]			
	VBUS input				

Notes: 1. TIOCB4-A input when MD3 to MD0 = B'0000 or B'01×× and IOB3 to IOB0 = B'10××.
2. PO5-A output when the PPGS bit in PFCR3 is 0.
3. TIOCB4-A input/output when the TPUS bit in PFCR3 is 0.
4. TMO1-A output when the TMRS bit in PFCR3 is 0.
5. $\overline{\text{WAIT-B}}$ input when the WAITS bit in PFCR4 is 1.

TPU channel 4 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don't care

- P20/PO0-A/TIOCA3-A/TMRI0-A/PUPD+

  The pin function is switched as shown below according to the combination of the TPU channel 3 settings (by bits MD3 to MD0 in TMDR_3, bits IOA3 to IOA0 in TIORH_3, and bits CCLR2 to CCLR0 in TCR_3), bit NDER0 in NDERL of PPG, bits PPGS, TPUS, and TMRS in PFCR3, and bit P20DDR.

PULLUP_E	0				1
TPU channel 3 settings	(1) in table below	(2) in table below			—
P20DDR	—	0	1		—
NDER0	—	—	0	1	—
Pin function	TIOCA3-A output*[4]	P20 input	P20 output	PO0-A output*[3]	PUPD+ output
		TIOCA3-A input*[1]*[4]			
	TMRI0-A input*[5]				

TPU channel 3 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA3-A input when MD3 to MD0 = B'0000 and IOA3 to IOA0 = B'10××.

2. TIOCB3 output disabled.
3. PO0-A output when the PPGS bit in PFCR3 is 0.
4. TIOCA3-A input/output when the TPUS bit in PFCR3 is 0.
5. TMRI0-A input when the TMRS bit in PFCR3 is 0.

10.3 Port 3

Port 3 is a 6-bit I/O port that also has other functions. Port 3 has the following registers. For the port function control registers, refer to section 10.18, Port Function Control Registers.

- Port 3 data direction register (P3DDR)
- Port 3 data register (P3DR)
- Port 3 register (PORT3)
- Port 3 open drain control register (P3ODR)
- Port function control register 2 (PFCR2)

10.3.1 Port 3 Data Direction Register (P3DDR)

The individual bits of P3DDR specify input or output for the pins of port 3. P3DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved
5	P35DDR	0	W	When a pin function is specified as a general purpose I/O, setting this bit to 1 makes the corresponding pin an output port, while clearing this bit to 0 makes the corresponding pin an input port.
4	P34DDR	0	W	
3	P33DDR	0	W	
2	P32DDR	0	W	
1	P31DDR	0	W	
0	P30DDR	0	W	

10.3.2 Port 3 Data Register (P3DR)

P3DR stores output data for the port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
5	P35DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
4	P34DR	0	R/W	
3	P33DR	0	R/W	
2	P32DR	0	R/W	
1	P31DR	0	R/W	
0	P30DR	0	R/W	

10.3.3 Port 3 Register (PORT3)

PORT3 shows the pin states of port 3. PORT3 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	Undefined	—	Reserved If these bits are read, they will return an undefined value.
5	P35	—*	R	If this register is read while a P3DDR bit is set to 1, the corresponding P3DR value is read. If this register is read while a P3DDR bit is cleared to 0, the corresponding pin state is read.
4	P34	—*	R	
3	P33	—*	R	
2	P32	—*	R	
1	P31	—*	R	
0	P30	—*	R	

Note: * Determined by the states of pins P35 to P30.

10.3.4 Port 3 Open Drain Control Register (P3ODR)

P3ODR specifies the output type of each port 3 pin.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0. Only the initial values should be written to these bits.
5	P35ODR	0	R/W	When $\overline{\text{OE-B}}$/CKE-B output is not selected, setting a P3ODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a P3ODR bit to 0 makes the corresponding pin a CMOS output pin.
4	P34ODR	0	R/W	
3	P33ODR	0	R/W	
2	P32ODR	0	R/W	
1	P31ODR	0	R/W	
0	P30ODR	0	R/W	

10.3.5 Pin Functions

Port 3 pins also function as the pins for SCI I/Os, I^2C I/Os, and bus control signal outputs. The correspondence between the register specification and the pin functions is shown below.

- P35/$\overline{\text{OE-B}}$/CKE-B*[4]/SCK1/SCL0

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit ICE in ICCRA_0 of I^2C, bit C/$\overline{\text{A}}$ in SMR_1 and bits CKE0 and CKE1 in SCR_1 of SCI, bits OEE and RMTS2 to RMTS0 in DRAMCR of the bus controller, bit OES in PFCR2, and bit P35DDR.

- Modes 1, 2, 4, and 7 (EXPE = 1)

OEE	0						1							
OES	—						1						0	
RMTS2 to RMTS0	—						—						Areas 2 to 5 are DRAM space	Areas 2 to 5 are continuous SDRAM space
ICE	0					1	0					1	—	
CKE1	0				1	—	0				1	—	—	
C/$\overline{A}$	0			1	—	—	0			1	—	—	—	
CKE0	0		1	—	—	—	0		1	—	—	—	—	
P35DDR	0	1	—	—	—	—	0	1	—	—	—	—	—	
Pin function	P35 input	P35 output*1	SCK1 output*1	SCK1 output*1	SCK1 input	SCL0 I/O*2	P35 input	P35 output*1	SCK1 output*1	SCK1 output*1	SCK1 input	SCL0 I/O*2	$\overline{OE}$-B output*3	CKE-B output*3*4

- Mode 7 (EXPE = 0)

OEE	0					
OES	—					
RMTS2 to RMTS0	—					
ICE	0					1
CKE1	0				1	—
C/$\overline{A}$	0			1	—	—
CKE0	0		1	—	—	—
P35DDR	0	1	—	—	—	—
Pin function	P35 input	P35 output*1	SCK1 output*1	SCK1 output*1	SCK1 input	SCL0 I/O*2

Notes: 1. NMOS open-drain output when P35ODR = 1.
2. NMOS open-drain output regardless of P35ODR.
3. $\overline{OE}$-B/CKE-B output when the OES bit in PFCR2 is 0.
4. Not supported in the H8S/2456 Group and H8S/2454 Group.

- P34/SCK0/SCK4-A/SDA0

  The pin function is switched as shown below according to the combination of bit ICE in ICCRA_0 of I^2C, bit C/$\overline{A}$ in SMR_0 and bits CKE0 and CKE1 in SCR_0 and SCR_4 of SCI, and bit P34DDR.

ICE	0					1
CKE1	0				1	—
C/$\overline{A}$	0			1	—	—
CKE0	0		1	—	—	—
P34DDR	0	1	—	—	—	—
Pin function	P34 input	P34 output*1	SCK0/SCK4-A output*1*3*4	SCK0/SCK4-A output*1*3*4	SCK0/SCK4-A input*4	SDA0 I/O*2

Notes: 1. NMOS open-drain output when P34ODR = 1.
2. NMOS open-drain output regardless of P34ODR.
3. Simultaneous output of SCK0 and SCK4 cannot be set.
4. SCK4-A input/output when the SCK4S bit in PFCR4 is 0.

- P33/RxD1/SCL1

  The pin function is switched as shown below according to the combination of bit ICE in ICCRA_1 of I^2C, bit RE in SCR_1 of SCI, and bit P33DDR.

ICE	0			1
RE	0		1	—
P33DDR	0	1	—	—
Pin function	P33 input	P33 output*1	RxD1 input	SCL1 I/O*2

Notes: 1. NMOS open-drain output when P33ODR = 1.
2. NMOS open-drain output regardless of P33ODR.

- P32/RxD0/IrRxD/SDA1

  The pin function is switched as shown below according to the combination of bit ICE in ICCRA_1 of I^2C, bit RE in SCR_0 of SCI, and bit P32DDR.

ICE	0			1
RE	0		1	—
P32DDR	0	1	—	—
Pin function	P32 input	P32 output*1	RxD0/IrRxD input	SDA1 I/O*2

Notes: 1. NMOS open-drain output when P32ODR = 1.
2. NMOS open-drain output regardless of P32ODR.

- P31/TxD1

  The pin function is switched as shown below according to the combination of bit TE in SCR_1 of SCI and bit P31DDR.

TE	0		1
P31DDR	0	1	—
Pin function	P31 input	P31 output*	TxD1 output*

Note: * NMOS open-drain output when P31ODR = 1.

- P30/TxD0/IrTxD

  The pin function is switched as shown below according to the combination of bit TE in SCR_0 of SCI and bit P30DDR.

TE	0		1
P30DDR	0	1	—
Pin function	P30 input	P30 output*	TxD0/IrTxD output*

Note: * NMOS open-drain output when P30ODR = 1.

10.4 Port 4

Port 4 is an 8-bit input-only port that also has other functions, such as analog input pins. Port 4 has the following register.

- Port 4 register (PORT4)

10.4.1 Port 4 Register (PORT4)

PORT4 is an 8-bit read-only register that shows the pin states of port 4. PORT4 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P47	—*	R	The pin states are always read from this register.
6	P46	—*	R	
5	P45	—*	R	
4	P44	—*	R	
3	P43	—*	R	
2	P42	—*	R	
1	P41	—*	R	
0	P40	—*	R	

Note: * Determined by the states of pins P47 to P40.

10.4.2 Pin Functions

Port 4 also functions as the pins for A/D converter analog inputs and interrupt inputs (the H8S/2454 Group). The correspondence between pins is as follows.

(1) Pin Functions of H8S/2456 Group and H8S/2456R Group

- P40/AN0_0, P41/AN1_0, P42/AN2_0, P43/AN3_0, P44/AN4_0, P45/AN5_0, P46/AN6_0, P47/AN7_0

Pin function	ANn_0 input

Legend:
n = 7 to 0

(2) Pin Functions of H8S/2454 Group

- P47/$\overline{\text{IRQ7-B}}$/AN7_0

Pin function	AN7_0 input
	$\overline{\text{IRQ7-B}}$ interrupt input*

- P46/$\overline{\text{IRQ6-B}}$/AN6_0

Pin function	AN6_0 input
	$\overline{\text{IRQ6-B}}$ interrupt input*

- P45/$\overline{\text{IRQ5-B}}$/AN5_0

Pin function	AN5_0 input
	$\overline{\text{IRQ5-B}}$ interrupt input*

- P44/$\overline{\text{IRQ4-B}}$/AN4_0

Pin function	AN4_0 input
	$\overline{\text{IRQ4-B}}$ interrupt input*

- P43/$\overline{\text{IRQ3-B}}$/AN3_0

Pin function	AN3_0 input
	$\overline{\text{IRQ3-B}}$ interrupt input*

- P42/$\overline{\text{IRQ2-B}}$/AN2_0

Pin function	AN2_0 input
	$\overline{\text{IRQ2-B}}$ interrupt input*

- P41/$\overline{\text{IRQ1-B}}$/AN1_0

Pin function	AN1_0 input
	$\overline{\text{IRQ1-B}}$ interrupt input*

- P40/$\overline{\text{IRQ0-B}}$/AN0_0

Pin function	AN0_0 input
	$\overline{\text{IRQ0-B}}$ interrupt input*

Note: * $\overline{\text{IRQn}}$ input when the ITSn bit in ITSR is 1. (n = 7 to 0)

10.5 Port 5

Port 5 is a 4-bit I/O port. Port 5 has the following registers. For the port function control registers, refer to section 10.18, Port Function Control Registers.

- Port 5 data direction register (P5DDR)
- Port 5 data register (P5DR)
- Port 5 register (PORT5)
- Port 5 open drain control register (P5ODR)
- Port function control register 4 (PFCR4)

10.5.1 Port 5 Data Direction Register (P5DDR)

The individual bits of P5DDR specify input or output for the pins of port 5. P5DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved
3	P53DDR	0	W	When a pin function is specified as a general purpose I/O, setting this bit to 1 makes the corresponding pin an output port, while clearing this bit to 0 makes the corresponding pin an input port.
2	P52DDR	0	W	
1	P51DDR	0	W	
0	P50DDR	0	W	

10.5.2 Port 5 Data Register (P5DR)

P5DR stores output data for the port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	P53DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
2	P52DR	0	R/W	
1	P51DR	0	R/W	
0	P50DR	0	R/W	

10.5.3 Port 5 Register (PORT5)

PORT5 shows the pin states of port 5. PORT5 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	R	Reserved If these bits are read, they will return an undefined value.
3	P53	—*	R	If the P53 to P50 bits are read while a P5DDR bit is set to 1, the corresponding P5DR value is read. If this register is read while a P5DDR bit is cleared to 0, the corresponding pin state is read.
2	P52	—*	R	
1	P51	—*	R	
0	P50	—*	R	

Note: * Determined by the states of pins P53 to P50.

10.5.4 Port 5 Open Drain Control Register (P5ODR)

P5ODR specifies the output type of each port 5 pin.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0. Only the initial values should be written to these bits.
3	P53ODR	0	R/W	When $\overline{\text{BACK-B}}$/$\overline{\text{BREQO-B}}$ output is not selected, setting a P5ODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a P5ODR bit to 0 makes the corresponding pin a CMOS output pin.
2	P52ODR	0	R/W	
1	P51ODR	0	R/W	
0	P50ODR	0	R/W	

10.5.5 Pin Functions

Port 5 pins also function as the pins for SCI I/Os, A/D converter inputs, interrupt inputs, I^2C I/Os, bus control signal I/Os, JTAG inputs, PPG outputs, TPU I/Os, and 8-bit timer I/Os. The correspondence between the register specification and the pin functions is shown below.

- P53/$\overline{\text{IRQ3-A}}$/$\overline{\text{ADTRG0-A}}$/$\overline{\text{TRST}}$*[3]

  The pin function is switched as shown below according to the combination of bits TRGS1, TRGS0, and EXTRGS in ADCR_0 of ADC, bit P53DDR, and bit ITS3 in ITSR of the interrupt controller.

P53DDR	0	1
Pin function	P53 input	P53 output
	$\overline{\text{ADTRG0-A}}$ input*[1]	
	$\overline{\text{IRQ3-A}}$ interrupt input*[2]	
	$\overline{\text{TRST}}$ input*[3]	

Notes: 1. $\overline{\text{ADTRG0-A}}$ input when the EXTRGS in ADCR0 is 0, and TRGS1 = TRGS0 = 1.
2. $\overline{\text{IRQ3-A}}$ input when the ITS3 bit in ITSR is 0.
3. $\overline{\text{TRST}}$ input when BSCANE pin = 1 and EMLE = 0 in the 145-pin package.

- P52/SCK2/$\overline{\text{IRQ2-A}}$/$\overline{\text{BACK-B}}$/PO4-B/TIOCA4-B/TMO0-B

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit BRLE in BCR of the bus controller, bits OS3 to OS0 in TCSR0 of 8-bit timer, bits MD3 to MD0 in TMDR_4 of TPU, bits IOA3 to IOA0 in TIOR_4, TPU channel 4 settings by bits CCLR1 and CCLR0 in TCR_4, bit NDER4 in NDERL of PPG, bit C/$\overline{\text{A}}$ in SMR_2 and bits CKE0 and CKE1 in SCR_2 of SCI, bits PPGS, TPUS, and TMRS in PFCR3, bit BACKS in PFCR4, bit P52DDR, bit NDER4 in NDERL of PPG, and bit ITS2 in ITSR of the interrupt controller.

- Modes 1, 2, 4, and 7 (EXPE = 1)

BRLE BACKS	BRLE = 0 or BRLE = 1 and BACKS = 0								BRLE = 1 and BACKS = 1
TPU channel 4 settings	(1) in table below	(2) in table below							—
OS3 to OS0	—	All 0						Not all 0	—
CKE1	—	0					1	—	—
C/$\overline{\text{A}}$	—	0				1	—	—	—
CKE0	—	0			1	—	—	—	—
P52DDR	—	0	1	1	—	—	—	—	—
NDER4	—	—	0	1	—	—	—	—	—
Pin function	TIOCA4-B output*[3]	P52 input	P52 output	PO4-B output*[2]	SCK2 output	SCK2 output	SCK2 input	TMO0-B output*[4]	$\overline{\text{BACK-B}}$ output
		TIOCA4-B input*[3]							
	$\overline{\text{IRQ2-A}}$ interrupt input*[1]								

- Mode 7 (EXPE = 0)

BRLE BACKS	—							
TPU channel 4 settings	(1) in table below	(2) in table below						
OS3 to OS0	—	All 0						Not all 0
CKE1	—	0					1	—
C/$\overline{A}$	—	0				1	—	—
CKE0	—	0			1	—	—	—
P52DDR	—	0	1	1	—	—	—	—
NDER4	—	—	0	1	—	—	—	—
Pin function	TIOCA4-B output*[3]	P52 input	P52 output	PO4-B output*[2]	SCK2 output	SCK2 output	SCK2 input	TMO0-B output*[4]
		TIOCA4-B input*[3]						
	$\overline{IRQ2\text{-}A}$ interrupt input*[1]							

Notes: 1. $\overline{IRQ2\text{-}A}$ input when the ITS2 bit in ITSR is 0.
2. PO4-B output when the PPGS bit in PFCR3 is 1.
3. TIOCA4-B input/output when the TPUS bit in PFCR3 is 1.
4. TMO0-B output when the TMRS bit in PFCR3 is 1.

TPU channel 4 settings	(2)	(1)	(1)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'001×	B'0011	
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	Other than B'××00	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	PWM mode 1 output	—	PWM mode 2 output	—

Legend:
×: Don't care

- P51/RxD2/$\overline{\text{IRQ1-A}}$/SCL3/$\overline{\text{BREQ-B}}$/PO2-B/TIOCC3-B/TMCI0-B

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit BRLE in BCR of the bus controller, bit ICE in ICCRA_3 of I^2C, bits MD3 to MD0 in TMDR_3 of TPU, bits IOC3 to IOC0 in TIORL_3, TPU channel 3 settings by bits CCLR2 to CCLR0 in TCR_3, bit NDER2 in NDERL of PPG, bit RE in SCR_2 of SCI, bit P51DDR, and bit ITS1 in ITSR of the interrupt controller.

- Modes 1, 2, 4, and 7 (EXPE = 1)

<table>
<tr><td>BRLE
BREQS</td><td colspan="6">BRLE = 0 or BRLE = 1 and BREQS = 0</td><td>BRLE = 1 and BREQS = 1</td></tr>
<tr><td>ICE</td><td colspan="5">0</td><td>1</td><td>—</td></tr>
<tr><td>TPU channel 3 settings</td><td>(1) in table below</td><td colspan="4">(2) in table below</td><td>—</td><td>—</td></tr>
<tr><td>RE</td><td>—</td><td colspan="3">0</td><td>1</td><td>—</td><td>—</td></tr>
<tr><td>P51DDR</td><td>—</td><td>0</td><td>1</td><td>1</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>NDER2</td><td>—</td><td>—</td><td>0</td><td>1</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td rowspan="3">Pin function</td><td rowspan="2">TIOCC3-B output*3</td><td>P51 input</td><td>P51 output</td><td>PO2-B output*2</td><td>RxD2 input</td><td>SCL3 I/O</td><td>$\overline{\text{BREQ-B}}$ input</td></tr>
<tr><td colspan="6">TIOCC3-B input*3</td></tr>
<tr><td colspan="7">$\overline{\text{IRQ1-A}}$ interrupt input*1</td></tr>
<tr><td></td><td colspan="7">TMCI0-B input*4</td></tr>
</table>

- Mode 7 (EXPE = 0)

BRLE BREQS	—					
ICE	0					1
TPU channel 3 settings	(1) in table below	(2) in table below				—
RE	—	0			1	—
P51DDR	—	0	1	1	—	—
NDER2	—	—	0	1	—	—
Pin function	TIOCC3-B output*[3]	P51 input	P51 output	PO2-B output*[2]	RxD2 input	SCL3 I/O
		TIOCC3-B input*[3]				
	$\overline{\text{IRQ1-A}}$ interrupt input*[1]					
	TMCI0-B input*[4]					

TPU channel 3 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001×	B'0010	B'0011	
IOC3 to IOC0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM*[3] mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. $\overline{\text{IRQ1-A}}$ input when the ITS1 bit in ITSR is 0.

2. PO2-B output when the PPGS bit in PFCR3 is 1.
3. TIOCC3-B input/output when the TPUS bit in PFCR3 is 1.
4. TMCI0-B input when the TMRS bit in PFCR3 is 1.

- P50/TxD2/$\overline{\text{IRQ0-A}}$/SDA3/$\overline{\text{BREQO-B}}$*[2]/PO0-B/TIOCA3-B/TMRI0-B

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit BRLE in BCR of the bus controller, bit ICE in ICCRA_3 of I^2C, bits MD3 to MD0 in TMDR_3 of TPU, bits IOA3 to IOA0 in TIORH_3, TPU channel 3 settings by bits CCLR2 to CCLR0 in TCR_3, bit NDER0 in NDERL of PPG, bit TE in SCR_2 of SCI, bits PPGS, TPUS, and TMRS in PFCR3, bit BREQOS in PFCR4, bit P50DDR, bit NDER0 in NDERL of PPG, and bit ITS0 in ITSR of the interrupt controller.

- Modes 1, 2, 4, and 7 (EXPE = 1)

<table>
<tr><td>BRLE</td><td colspan="7">0</td><td colspan="7">1</td></tr>
<tr><td>BREQOE
BREQOS</td><td colspan="7">—</td><td colspan="6">BREQOE= 0 or BREQOE= 1 and BREQOS = 0</td><td>BREQOE = 1
and
BREQOS = 1</td></tr>
<tr><td>ICE</td><td colspan="6">0</td><td>1</td><td colspan="5">0</td><td>1</td><td>—</td></tr>
<tr><td>TPU
channel 3
settings</td><td>(1) in table below</td><td colspan="5">(2) in table below</td><td>—</td><td>(1) in table below</td><td colspan="4">(2) in table below</td><td>—</td><td>—</td></tr>
<tr><td>TE</td><td>—</td><td colspan="3">0</td><td>1</td><td>—</td><td>—</td><td colspan="3">0</td><td>1</td><td>—</td><td>—</td></tr>
<tr><td>P50DDR</td><td>—</td><td>0</td><td>1</td><td>1</td><td>—</td><td>—</td><td>—</td><td>0</td><td>1</td><td>1</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>NDER0</td><td>—</td><td>—</td><td>0</td><td>1</td><td>—</td><td>—</td><td>—</td><td>—</td><td>0</td><td>1</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td rowspan="3">Pin
function</td><td rowspan="2">TIOCA3-B
output*[3]</td><td>P50
input</td><td>P50
output</td><td>PO0-B
output
*[2]</td><td>TxD2
output</td><td>SDA3
I/O</td><td rowspan="2">TIOCA3-B
output*[3]</td><td>P50
input</td><td>P50
output</td><td>PO0-B
output
*[2]</td><td>TxD2
output</td><td>SDA3
I/O</td><td>$\overline{\text{BREQO-B}}$
output</td></tr>
<tr><td colspan="6">TIOCA3-B input*[3]</td><td colspan="6">TIOCA3-B input*[3]</td></tr>
<tr><td colspan="7">$\overline{\text{IRQ0-A}}$ interrupt input*[1]</td><td colspan="7">TMRI0-B input*[4]</td></tr>
</table>

- Mode 7 (EXPE = 0)

BRLE	—					
BREQOE BREQOS	—					
ICE	0					1
TPU channel 3 settings	(1) in table below	(2) in table below				—
TE	—	0			1	—
P50DDR	—	0	1	1	—	—
NDER0	—	—	0	1	—	—
Pin function	TIOCA3-B output*3	P50 input	P50 output	PO0-B output*2	TxD2 output	SDA3 I/O
		TIOCA3-B input*3				
	$\overline{\text{IRQ0-A}}$ interrupt input*1					
	TMRI0-B input*4					

TPU channel 3 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM*3 mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. $\overline{\text{IRQ0-A}}$ input when the ITS0 bit in ITSR is 0.

2. PO0-B output when the PPGS bit in PFCR3 is 1.
3. TIOCA3-B input/output when the TPUS bit in PFCR3 is 1.
4. TMRI0-B input when the TMRS bit in PFCR3 is 1.

10.6 Port 6

Note: Port 6 is not supported in the H8S/2454 Group.

Port 6 is a 6-bit I/O port that also has other functions. Port 6 has the following registers. For the port function control registers, refer to section 10.18, Port Function Control Registers.

- Port 6 data direction register (P6DDR)
- Port 6 data register (P6DR)
- Port 6 register (PORT6)
- Port 6 open drain control register (P6ODR)
- Port function control register 3 (PFCR3)

10.6.1 Port 6 Data Direction Register (P6DDR)

The individual bits of P6DDR specify input or output for the pins of port 6. P6DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved
5	P65DDR	0	W	When a pin function is specified as a general purpose I/O, setting this bit to 1 makes the corresponding pin an output port, while clearing this bit to 0 makes the corresponding pin an input port.
4	P64DDR	0	W	
3	P63DDR	0	W	
2	P62DDR	0	W	
1	P61DDR	0	W	
0	P60DDR	0	W	

10.6.2 Port 6 Data Register (P6DR)

P6DR stores output data for the port 6 pins.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
5	P65DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
4	P64DR	0	R/W	
3	P63DR	0	R/W	
2	P62DR	0	R/W	
1	P61DR	0	R/W	
0	P60DR	0	R/W	

10.6.3 Port 6 Register (PORT6)

PORT6 shows the pin states of port 6. PORT6 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	Undefined	—	Reserved If these bits are read, they will return an undefined value.
5	P65	—*	R	If this register is read while a P6DDR bit is set to 1, the corresponding P6DR value is read. If this register is read while a P6DDR bit is cleared to 0, the corresponding pin state is read.
4	P64	—*	R	
3	P63	—*	R	
2	P62	—*	R	
1	P61	—*	R	
0	P60	—*	R	

Note: * Determined by the states of pins P65 to P60.

10.6.4 Port 6 Open Drain Control Register (P6ODR)

P6ODR specifies the output type of each port 6 pin.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0. Only the initial values should be written to these bits.
5	P65ODR	0	R/W	Setting a P6ODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a P6ODR bit to 0 makes the corresponding pin a CMOS output pin.
4	P64ODR	0	R/W	
3	P63ODR	0	R/W	
2	P62ODR	0	R/W	
1	P61ODR	0	R/W	
0	P60ODR	0	R/W	

10.6.5 Pin Functions

Port 6 pins also function as 8-bit timer I/Os, interrupt inputs, and DMAC I/Os. The correspondence between the register specification and the pin functions is shown below.

- P65/$\overline{\text{IRQ13-A}}$/$\overline{\text{DACK1}}$/TMO1-A

  The pin function is switched as shown below according to the combination of bit SAE1 in DMABCRH of DMAC, bits OS3 to OS0 in TCSR_1 of the 8-bit timer, bit TMRS in PFCR3, bit P65DDR, and bit ITS13 in ITSR of the interrupt controller.

SAE1	0			1
OS3 to OS0	All 0		Not all 0	—
P65DDR	0	1	—	—
Pin function	P65 input	P65 output	TMO1-A output*[2]	$\overline{\text{DACK1}}$ output
	$\overline{\text{IRQ13-A}}$ interrupt input*[1]			

Notes: 1. $\overline{\text{IRQ13-A}}$ input when the ITS13 bit in ITSR is 0.
2. TMO1-A output when the TMRS bit in PFCR3 is 0.

- P64/$\overline{\text{IRQ12-A}}$/$\overline{\text{DACK0}}$/TMO0-A

  The pin function is switched as shown below according to the combination of bit SAE0 in DMABCRH of DMAC, bits OS3 to OS0 in TCSR_0 of the 8-bit timer, bit TMRS in PFCR3, bit P64DDR, and bit ITS12 in ITSR of the interrupt controller.

SAE0	0			1
OS3 to OS0	All 0		Not all 0	—
P64DDR	0	1	—	—
Pin function	P64 input	P64 output	TMO0-A output*2	$\overline{\text{DACK0}}$ output
	$\overline{\text{IRQ12-A}}$ interrupt input*1			

Notes: 1. $\overline{\text{IRQ12-A}}$ input when the ITS12 bit in ITSR is 0.
2. TMO0-A output when the TMRS bit in PFCR3 is 0.

- P63/$\overline{\text{IRQ11-A}}$/$\overline{\text{TEND1}}$/TMCI1-A

  The pin function is switched as shown below according to the combination of bit TEE1 in DMATCR of DMAC, bit TMRS in PFCR3, bit P63DDR, and bit ITS11 in ITSR of the interrupt controller.

TEE1	0		1
P63DDR	0	1	—
Pin function	P63 input	P63 output	$\overline{\text{TEND1}}$ output
	$\overline{\text{IRQ11-A}}$ interrupt input*1		
	TMCI1-A input*2*3		

Notes: 1. $\overline{\text{IRQ11-A}}$ input when the ITS11 bit in ITSR is 0.
2. When used as the external clock input pin for the TMR, its pin function should be specified to the external clock input by the CKS2 to CKS0 bits in TCR_1.
3. TMCI1-A input when the TMRS bit in PFCR3 is 0.

- P62/$\overline{\text{IRQ10-A}}$/$\overline{\text{TEND0}}$/TMCI0-A

  The pin function is switched as shown below according to the combination of bit TEE0 in DMATCR of DMAC, bit TMRS in PFCR3, bit P62DDR, and bit ITS10 in ITSR of the interrupt controller.

TEE0	0		1
P62DDR	0	1	—
Pin function	P62 input	P62 output	$\overline{\text{TEND0}}$ output
	$\overline{\text{IRQ10-A}}$ interrupt input*[1]		
	TMCI0-A input*[2]*[3]		

Notes: 1. $\overline{\text{IRQ10-A}}$ input when the ITS10 bit in ITSR is 0.
2. When used as the external clock input pin for the TMR, its pin function should be specified to the external clock input by the CKS2 to CKS0 bits in TCR_0.
3. TMCI0-A input when the TMRS bit in PFCR3 is 0.

- P61/$\overline{\text{IRQ9-A}}$/$\overline{\text{DREQ1}}$/TMRI1-A

  The pin function is switched as shown below according to the combination of bits TMRS and USBDRQE in PFCR3, bit P61DDR, and bit ITS9 in ITSR of the interrupt controller.

P61DDR	0	1
Pin function	P61 input	P61 output
	TMRI1-A input*[1]*[3]	
	$\overline{\text{DREQ1}}$ input*[4]	
	$\overline{\text{IRQ9-A}}$ interrupt input*[2]	

Notes: 1. When used as the counter reset input pin for the TMR, both the CCLR1 and CCLR0 bits in TCR_1 should be set to 1.
2. $\overline{\text{IRQ9-A}}$ input when the ITS9 bit in ITSR is 0.
3. TMRI1-A input when the TMRS bit in PFCR3 is 0.
4. $\overline{\text{DREQ1}}$ input when the USBDRQE bit in PFCR3 is 0.

- P60/$\overline{\text{IRQ8-A}}$/$\overline{\text{DREQ0}}$/TMRI0-A

  The pin function is switched as shown below according to the combination of bits TMRS and USBDRQE in PFCR3, bit P60DDR, and bit ITS8 in ITSR of the interrupt controller.

P60DDR	0	1
Pin function	P60 input	P60 output
	TMRI0-A input*[1]*[3]	
	$\overline{\text{DREQ0}}$ input*[4]	
	$\overline{\text{IRQ8-A}}$ interrupt input*[2]	

Notes: 1. When used as the counter reset input pin for the TMR, both the CCLR1 and CCLR0 bits in TCR_0 should be set to 1.
2. $\overline{\text{IRQ8-A}}$ input when the ITS8 bit in ITSR is 0.
3. TMRI0-A input when the TMRS bit in PFCR3 is 0.
4. $\overline{\text{DREQ0}}$ input when the USBDRQE bit in PFCR3 is 0.

10.7 Port 8

Port 8 is a 6-bit I/O port that also has other functions. Port 8 has the following registers. For the port function control registers, refer to section 10.18, Port Function Control Registers.

- Port 8 data direction register (P8DDR)
- Port 8 data register (P8DR)
- Port 8 register (PORT8)
- Port 8 open drain control register (P8ODR)
- Port function control register 3 (PFCR3)

10.7.1 Port 8 Data Direction Register (P8DDR)

The individual bits of P8DDR specify input or output for the pins of port 8. P8DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved
5	P85DDR	0	W	When a pin function is specified as a general purpose I/O, setting this bit to 1 makes the corresponding pin an output port, while clearing this bit to 0 makes the corresponding pin an input port. Bits 4, 2, and 0 are reserved in the H8S/2454 Group.
4	P84DDR	0	W	
3	P83DDR	0	W	
2	P82DDR	0	W	
1	P81DDR	0	W	
0	P80DDR	0	W	

10.7.2 Port 8 Data Register (P8DR)

P8DR stores output data for the port 8 pins.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
5	P85DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O. Bits 4, 2, and 0 are reserved in the H8S/2454 Group.
4	P84DR	0	R/W	
3	P83DR	0	R/W	
2	P82DR	0	R/W	
1	P81DR	0	R/W	
0	P80DR	0	R/W	

10.7.3 Port 8 Register (PORT8)

PORT8 shows the pin states of port 8. PORT8 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	Undefined	—	Reserved If these bits are read, they will return an undefined value.
5	P85	—*	R	If this register is read while a P8DDR bit is set to 1, the corresponding P8DR value is read. If this register is read while a P8DDR bit is cleared to 0, the corresponding pin state is read. Bits 4, 2, and 0 are reserved in the H8S/2454 Group.
4	P84	—*	R	
3	P83	—*	R	
2	P82	—*	R	
1	P81	—*	R	
0	P80	—*	R	

Note: * Determined by the states of pins P85 to P80.

10.7.4 Port 8 Open Drain Control Register (P8ODR)

P8ODR specifies the output type of each port 8 pin.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0. Only the initial values should be written to these bits.
5	P85ODR	0	R/W	Setting a P8ODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a P8ODR bit to 0 makes the corresponding pin a CMOS output pin. Bits 4, 2, and 0 are reserved in the H8S/2454 Group.
4	P84ODR	0	R/W	
3	P83ODR	0	R/W	
2	P82ODR	0	R/W	
1	P81ODR	0	R/W	
0	P80ODR	0	R/W	

10.7.5 Pin Functions

Port 8 pins also function as SCI I/Os, interrupt inputs, EXDMAC I/Os, PPG outputs, TPU I/Os, and 8-bit timer I/Os. The correspondence between the register specification and the pin functions is shown below.

(1) Pin Functions of H8S/2456 Group and H8S/2456R Group

- P85/$\overline{\text{EDACK3}}$/$\overline{\text{IRQ5}}$-B/SCK3/PO5-B/TIOCB4-B/TMO1-B

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 4 settings (by bits MD3 to MD0 in TMDR_4, bits IOB3 to IOB0 in TIOR_4, and bits CCLR1 and CCLR0 in TCR_4), bits OS3 to OS0 in TCSR_1 of the 8-bit timer, bit NDER5 in NDERL of PPG, bit AMS in EDMDR_3 of EXDMAC, bit C/$\overline{\text{A}}$ in SMR_3 and bits CKE0 and CKE1 in SCR_3 of SCI, bits PPGS, TPUS, and TMRS in PFCR3, bit P85DDR, and bit ITS5 in ITSR of the interrupt controller.

- Modes 1, 2, 4, and 7 (EXPE = 1)

TPU channel 4 settings	(1) in table below	(2) in table below							
OS3 to OS0	—	All 0							Not all 0
AMS	—	0						1	—
CKE1	—	0					1	—	—
C/$\overline{A}$	—	0				1	—	—	—
CKE0	—	0			1	—	—	—	—
P85DDR	—	0	1	1	—	—	—	—	—
NDER5	—	—	0	1	—	—	—	—	—
Pin function	TIOCB4-B output*3	P85 input	P85 output	PO5-B output*2	SCK3 output	SCK3 output	SCK3 input	$\overline{\text{EDACK3}}$ output	TMO1-B output*4
		TIOCB4-B input*3							
	$\overline{\text{IRQ5-B}}$ interrupt input*1								

- Mode 7 (EXPE = 0)

TPU channel 4 settings	(1) in table below	(2) in table below						
OS3 to OS0	—	All 0						Not all 0
AMS	—	—						—
CKE1	—	0					1	—
C/$\overline{A}$	—	0				1	—	—
CKE0	—	0			1	—	—	—
P85DDR	—	0	1	1	—	—	—	—
NDER5	—	—	0	1	—	—	—	—
Pin function	TIOCB4-B output*3	P85 input	P85 output	PO5-B output*2	SCK3 output	SCK3 output	SCK3 input	TMO1-B output*4
		TIOCB4-B input*3						
	$\overline{\text{IRQ5-B}}$ interrupt input*1							

Notes: 1. $\overline{\text{IRQ5-B}}$ input when the ITS5 bit in ITSR is 1.
2. PO5-B output when the PPGS bit in PFCR3 is 1.
3. TIOCB4-B input/output when the TPUS bit in PFCR3 is 1.
4. TMO1-B output when the TMRS bit in PFCR3 is 1.

TPU channel 4 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don't care

- P84/$\overline{\text{IRQ4-B}}$/EDACK2

  The pin function is switched as shown below according to the combination of bit AMS in EDMDR_2 of EXDMAC, bit P84DDR, and bit ITS4 in ITSR of the interrupt controller.

Operating mode	1, 2, 4, 7 (EXPE = 1)			7 (EXPE = 0)	
AMS	0		1	—	
P84DDR	0	1	—	0	1
Pin function	P84 input	P84 output	$\overline{\text{EDACK2}}$ output	P84 input	P84 output
	$\overline{\text{IRQ4-B}}$ interrupt input*				

Note: * $\overline{\text{IRQ4-B}}$ input when the ITS4 bit in ITSR is 1.

- P83/$\overline{\text{ETEND3}}$*[5]/$\overline{\text{IRQ3-B}}$*[5]/RxD3/PO3-B/TIOCD3-B/TMCI1-B

  The pin function is switched as shown below according to the combination of bit ETENDE in EDMDR_3 of EXDMAC, bit RE in SCR_3 of SCI, TPU channel 3 settings (by bits MD3 to MD0 in TMDR_3, bits IOD3 to IOD0 in TIORL_3, and bits CCLR2 to CCLR0 in TCR_3), bit NDER3 in NDERL of PPG, bits PPGS, TPUS, and TMRS in PFCR3, bit P83DDR, and bit ITS3 in ITSR of the interrupt controller.
- Modes 1, 2, 4, and 7 (EXPE = 1)

<table>
<tr><td>TPU channel 3 settings</td><td>(1) in table below</td><td colspan="5">(2) in table below</td></tr>
<tr><td>ETENDE</td><td>—</td><td colspan="4">0</td><td>1</td></tr>
<tr><td>RE</td><td>—</td><td colspan="3">0</td><td>1</td><td>—</td></tr>
<tr><td>P83DDR</td><td>—</td><td>0</td><td>1</td><td>1</td><td>—</td><td>—</td></tr>
<tr><td>NDER3</td><td>—</td><td>—</td><td>0</td><td>1</td><td>—</td><td>—</td></tr>
<tr><td rowspan="4">Pin function</td><td rowspan="2">TIOCD3-B output*[3]</td><td>P83 input</td><td>P83 output</td><td>PO3-B output*[2]</td><td>RxD3 input</td><td>$\overline{\text{ETEND3}}$ output</td></tr>
<tr><td colspan="5">TIOCD3-B input*[3]</td></tr>
<tr><td colspan="6">$\overline{\text{IRQ3-B}}$ interrupt input*[1]</td></tr>
<tr><td colspan="6">TMCI1-B input*[4]</td></tr>
</table>

- Mode 7 (EXPE = 0)

TPU channel 3 settings	(1) in table below	(2) in table below			
ETENDE	—	0			1
RE	—	0		1	—
P83DDR	—	0	1		—
NDER3	—	—	0	1	—
Pin function	TIOCD3-B output*3	P83 input	P83 output	PO3-B output*2	RxD3 input
		TIOCD3-B input*3			
	$\overline{\text{IRQ3-B}}$ interrupt input*1				
	TMCI1-B input*4				

Notes: 1. $\overline{\text{IRQ3-B}}$ input when the ITS3 bit in ITSR is 1.
2. PO3-B output when the PPGS bit in PFCR3 is 1.
3. TIOCD3-B input/output when the TPUS bit in PFCR3 is 1.
4. TMCI1-B input when the TMRS bit in PFCR3 is 1.

TPU channel 3 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOD3 to IOD0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don’t care

- P82/$\overline{\text{IRQ2-B}}$/ETEND2

  The pin function is switched as shown below according to the combination of bit ETENDE in EDMDR_2 of EXDMAC, bit P82DDR, and bit ITS2 in ITSR of the interrupt controller.

Operating mode	1, 2, 4, 7 (EXPE = 1)			7 (EXPE = 0)	
ETENDE	0		1	—	
P82DDR	0	1	—	0	1
Pin function	P82 input	P82 output	$\overline{\text{ETEND2}}$ output	P82 input	P82 output
	$\overline{\text{IRQ2-B}}$ interrupt input*				

Note: * $\overline{\text{IRQ2-B}}$ input when the ITS2 bit in ITSR is 1.

- P81/$\overline{\text{EDREQ3}}$/$\overline{\text{IRQ1-B}}$/TxD3/PO1-B/TIOCB3-B/TMRI1-B

  The pin function is switched as shown below according to the combination of bit TE in SCR_3 of SCI, TPU channel 3 settings (by bits MD3 to MD0 in TMDR_3, bits IOB3 to IOB0 in TIORH_3, and bits CCLR2 to CCLR0 in TCR_3), bit NDER1 in NDERL of PPG, bits PPGS, TPUS, and TMRS in PFCR3, bit P81DDR, and bit ITS1 in ITSR of the interrupt controller.

TPU channel 3 settings	(1) in table below	(2) in table below			
TE	—	0			1
P81DDR	—	0	1	1	—
NDER1	—	—	0	1	—
Pin function	TIOCB3-B output*[3]	P81 input	P81 output	PO1-B output*[2]	TxD3 output
		TIOCB3-B input*[3]			
	$\overline{\text{EDREQ3}}$ input				
	$\overline{\text{IRQ1-B}}$ interrupt input*[1]				
	TMRI1-B input*[4]				

Notes: 1. $\overline{\text{IRQ1-B}}$ input when the ITS1 bit in ITSR is 1.
2. PO1-B output when the PPGS bit in PFCR3 is 1.
3. TIOCB3-B input/output when the TPUS bit in PFCR3 is 1.
4. TMRI1-B input when the TMRS bit in PFCR3 is 1.

TPU channel 3 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don't care

- P80/$\overline{\text{IRQ0-B}}$/$\overline{\text{EDREQ2}}$

  The pin function is switched as shown below according to the combination of bit P80DDR and bit ITS0 in ITSR of the interrupt controller.

P80DDR	0	1
Pin function	P80 input	P80 output
	$\overline{\text{EDREQ2}}$ input	
	$\overline{\text{IRQ0-B}}$ interrupt input*	

Note: * $\overline{\text{IRQ0-B}}$ input when the ITS0 bit in ITSR is 1.

(2) Pin Functions of H8S/2454 Group

- P85/SCK3/PO5-B/TIOCB4-B/TMO1-B

  The pin function is switched as shown below according to the combination of the TPU channel 4 settings (by bits MD3 to MD0 in TMDR_4, bits IOB3 to IOB0 in TIOR_4, and bits CCLR1 and CCLR0 in TCR_4), bits OS3 to OS0 in TCSR_1 of the 8-bit timer, bit NDER5 in NDERL of PPG, bit C/$\overline{A}$ in SMR_3 and bits CKE0 and CKE1 in SCR_3 of SCI, bits PPGS, TPUS, and TMRS in PFCR3, and bit P85DDR.

TPU channel 4 settings	(1) in table below	(2) in table below						
OS3 to OS0	—	All 0						Not all 0
CKE1	—	0					1	—
C/$\overline{A}$	—	0				1	—	—
CKE0	—	0			1	—	—	—
P85DDR	—	0	1	1	—	—	—	—
NDER5	—	—	0	1	—	—	—	—
Pin function	TIOCB4-B output*3	P85 input	P85 output	PO5-B output*1	SCK3 output	SCK3 output	SCK3 input	TMO1-B output*3
		TIOCB4-B input*2						

Notes: 1. PO5-B output when the PPGS bit in PFCR3 is 1.
2. TIOCB4-B input/output when the TPUS bit in PFCR3 is 1.
3. TMO1-B output when the TMRS bit in PFCR3 is 1.

TPU channel 4 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don't care

- P83/PO3-B/TIOCD3-B/TMCI1-B/RxD3

  The pin function is switched as shown below according to the combination of the TPU channel 3 settings (by bits MD3 to MD0 in TMDR_3, bits IOD3 to IOD0 in TIORL_3, and bits CCLR2 to CCLR0 in TCR_3), bit NDER3 in NDERL of PPG, bit RE in SCR_3 of SCI, bits PPGS, TPUS, and TMRS in PFCR3, and bit P83DDR.

TPU channel 3 settings	(1) in table below	(2) in table below			
RE	—	0			1
P83DDR	—	0	1		—
NDER3	—	—	0	1	—
Pin function	TIOCD3-B output*[2]	P83 input	P83 output	PO3-B output*[1]	RxD3 input
		TIOCD3-B input*[2]			
	TMCI1-B input*[3]				

Notes: 1. PO3-B output when the PPGS bit in PFCR3 is 1.
2. TIOCD3-B input/output when the TPUS bit in PFCR3 is 1.
3. TMCI1-B input when the TMRS bit in PFCR3 is 1.

TPU channel 3 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOD3 to IOD0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don’t care

- P81/PO1-B/TIOCB3-B/TMRI1-B/TxD3

  The pin function is switched as shown below according to the combination of the TPU channel 3 settings (by bits MD3 to MD0 in TMDR_3, bits IOB3 to IOB0 in TIORH_3, and bits CCLR2 to CCLR0 in TCR_3), bit NDER1 in NDERL of PPG, bit TE in SCR_3 of SCI, bits PPGS, TPUS, and TMRS in PFCR3, and bit P81DDR.

TPU channel 3 settings	(1) in table below	(2) in table below			
TE	—	0			1
P81DDR	—	0	1		—
NDER1	—	—	0	1	—
Pin function	TIOCB3-B output*[2]	P81 input	P81 output	PO1-B output*[1]	TxD3 output
		TIOCB3-B input*[2]			
	TMRI1-B input*[3]				

Notes: 1. PO1-B output when the PPGS bit in PFCR3 is 1.
2. TIOCB3-B input/output when the TPUS bit in PFCR3 is 1.
3. TMRI1-B input when the TMRS bit in PFCR3 is 1.

TPU channel 3 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:
×: Don't care

10.8 Port 9

Port 9 is an 8-bit input-only port that also has other functions. Port 9 has the following register.

- Port 9 register (PORT9)

10.8.1 Port 9 Register (PORT9)

PORT9 is an 8-bit read-only register that shows the pin states of port 9. PORT9 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P97	—*	R	The pin states are always read from this register. Bits 7, 6, and 3 to0 are reserved in the H8S/2454 Group.
6	P96	—*	R	
5	P95	—*	R	
4	P99	—*	R	
3	P93	—*	R	
2	P92	—*	R	
1	P91	—*	R	
0	P90	—*	R	

Note: * Determined by the states of pins P97 to P90.

10.8.2 Pin Functions

Port 9 also functions as the pins for A/D converter analog inputs and D/A converter analog outputs. The correspondence between pins is as follows.

(1) Pin Functions of H8S/2456 Group and H8S/2456R Group

- P97/AN15_1

Pin function	AN15_1 input

- P96/AN14_1

Pin function	AN14_1 input

- P95/AN13_1/DA3

Pin function	AN13_1 input
	DA3 output

- P94/AN12_1/DA2

Pin function	AN12_1 input
	DA2 output

- P93/AN11_1

Pin function	AN11_1 input

- P92/AN10_1

Pin function	AN10_1 input

- P91/AN9_1

Pin function	AN9_1 input

- P90/AN8_1

Pin function	AN8_1 input

(2) Pin Functions of H8S/2454 Group

- P95/AN13_1/DA3

Pin function	AN13_1 input
	DA3 output

- P94/AN12_1/DA2

Pin function	AN12_1 input
	DA2 output

10.9 Port A

Port A is an 8-bit I/O port that also has other functions. Port A has the following registers. For the port function control registers, refer to section 10.18, Port Function Control Registers.

- Port A data direction register (PADDR)
- Port A data register (PADR)
- Port A register (PORTA)
- Port A pull-up MOS control register (PAPCR)
- Port A open-drain control register (PAODR)
- Port function control register 0 (PFCR0)(the H8S/2454 Group)
- Port function control register 1 (PFCR1)
- Port function control register 4 (PFCR4)
- Port function control register 5 (PFCR5)

10.9.1 Port A Data Direction Register (PADDR)

The individual bits of PADDR specify input or output for the pins of port A. PADDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DDR	0	W	• Modes 1 and 2 Pins PA4 to PA0 are address outputs. For pins PA7 to PA5, when the corresponding bit of A23E to A21E is set to 1, setting a PADDR bit to 1 makes the corresponding pin an address output, while clearing the bit to 0 makes the corresponding pin an input port. Clearing one of bits A23E to A21E to 0 makes the corresponding pin an I/O port, and its function can be switched with PADDR. • Modes 7 (when EXPE = 1) and 4 When the corresponding bit of A23E to A16E is set to 1, setting a PADDR bit to 1 makes the corresponding pin an address output, while clearing the bit to 0 makes the corresponding pin an input port. Clearing one of bits A23E to A16E to 0 makes the corresponding pin an I/O port, and its function can be switched with PADDR. • Mode 7 (when EXPE = 0) Port A is an I/O port, and its pin functions can be switched with PADDR.
6	PA6DDR	0	W	
5	PA5DDR	0	W	
4	PA4DDR	0	W	
3	PA3DDR	0	W	
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR	0	W	

10.9.2 Port A Data Register (PADR)

PADR stores output data for the port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
6	PA6DR	0	R/W	
5	PA5DR	0	R/W	
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

10.9.3 Port A Register (PORTA)

PORTA shows the pin states of port A. PORTA cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7	—*	R	If this register is read while a PADDR bit is set to 1, the corresponding PADR value is read. If this register is read while a PADDR bit is cleared to 0, the corresponding pin state is read.
6	PA6	—*	R	
5	PA5	—*	R	
4	PA4	—*	R	
3	PA3	—*	R	
2	PA2	—*	R	
1	PA1	—*	R	
0	PA0	—*	R	

Note: * Determined by the states of pins PA7 to PA0.

10.9.4 Port A Pull-Up MOS Control Register (PAPCR)

PAPCR controls on/off of the input pull-up MOS for port A. Bits 7 to 5 are valid in modes 1 and 2 and all the bits are valid in modes 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7PCR	0	R/W	When in a input port state, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PA6PCR	0	R/W	
5	PA5PCR	0	R/W	
4	PA4PCR	0	R/W	
3	PA3PCR	0	R/W	
2	PA2PCR	0	R/W	
1	PA1PCR	0	R/W	
0	PA0PCR	0	R/W	

10.9.5 Port A Open Drain Control Register (PAODR)

PAODR specifies the output type of each port A pin.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7ODR	0	R/W	When not specified for address output or $\overline{CS7}$ output*, setting a PAODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a PAODR bit to 0 makes the corresponding pin a CMOS output pin.
6	PA6ODR	0	R/W	
5	PA5ODR	0	R/W	
4	PA4ODR	0	R/W	
3	PA3ODR	0	R/W	
2	PA2ODR	0	R/W	
1	PA1ODR	0	R/W	
0	PA0ODR	0	R/W	

Note: * Not supported by the H8S/2456 Group and the H8S/2456R Group.

10.9.6 Pin Functions

Port A pins also function as the pins for address outputs, interrupt inputs, SSU I/Os, SCI I/Os, and bus control signal outputs. The correspondence between the register specification and the pin functions is shown below.

- PA7/A23/$\overline{CS7}$*[4]/$\overline{IRQ7\text{-}A}$/SSO0-B

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits MSS and BIDE in SSCRH, bit SSUMS in SSCRL, and bits TE and RE in SSER of SSU, bits SSO0S1 and SSO0S0 in PFCR5, bit CS7E in PFCR0 (the H8S/2454 Group), bit A23E in PFCR1, bit PA7DDR, and bit ITS7 in ITSR of the interrupt controller.

- Modes 1, 2, and 4

A23E	0						1	
CS7E*[4]	0				1		—	
SSU settings	Can be used as I/O port		Input state	Output state	—		—	
PA7DDR	0	1	0	—	0	1	0	1
Pin function	PA7 input	PA7 output	SSO0-B input*[2]	SSO0-B output*[3]	PA7 input	$\overline{CS7}$ output*[4]	PA7 input	A23 output
	$\overline{IRQ7\text{-}A}$ interrupt input*[1]							

- Mode 7 (EXPE = 1)

A23E	0						1	
CS7E*[4]	0				1		—	
SSU settings	Can be used as I/O port		Input state	Output state	—		—	
PA7DDR	0	1	0	—	0	1	0	1
Pin function	PA7 input	PA7 output	SSO0-B input*[2]	SSO0-B output*[3]	PA7 input	$\overline{CS7}$ output*[4]	PA7 input	A23 output
	$\overline{IRQ7\text{-}A}$ interrupt input*[1]							

- Mode 7 (EXPE = 0)

A23E	—			
CS7E*[4]	0			
SSU settings	Can be used as I/O port		Input state	Output state
PA7DDR	0	1	0	—
Pin function	PA7 input	PA7 output	SSO0-B input*[2]	SSO0-B output*[3]
	$\overline{\text{IRQ7-A}}$ interrupt input*[1]			

Notes: 1. $\overline{\text{IRQ7-A}}$ input when the ITS7 bit in ITSR is 0.
2. SSO0-B input when SSO0S1 and SSO0S0 = B'01 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'000×1 or B'01×01.
3. SSO0-B output when SSO0S1 and SSO0S0 = B'01 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'001××, B'0101×, or B'10×1×.
4. Supported only by the H8S/2454 Group and not supported by the H8S/2456 and H8S/2456R Groups.

	SSO pin settings															
SSUMS	0						0				1					
BIDE	0						1				0					
MSS	0			1			0			1				0		
TE	0	1		0	1		0	1		0	1			0	1	
RE	1	0	1	1	0	1	1	0	1	0	1	0	1	1	0	1
Pin state	Input	—	Input	—	Output	Output	Input	Output	Input	Output	—	Output	Output	—	Output	Output

Legend:
—: Pin is not used by the SSU (can be used as I/O port)

- PA6/A22/$\overline{\text{IRQ6-A}}$/SSI0-B

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits MSS and BIDE in SSCRH, bit SSUMS in SSCRL, and bits TE and RE in SSER of SSU, bit A22E in PFCR1, bits SSO0S1 and SSO0S0 in PFCR5, bit PA6DDR, and bit ITS6 in ITSR of the interrupt controller.

- Modes 1, 2, and 4

A22E	—					
SSU settings	Can be used as I/O port		Input state	Output state	—	
PA6DDR	0	1	0	—	0	1
Pin function	PA6 input	PA6 output	SSI0-B input*2	SSI0-B output*3	PA6 input	A22 output
	$\overline{\text{IRQ6-A}}$ interrupt input*1					

- Mode 7

EXPE	1				0					
A22E	0				1		—			
SSU settings	Can be used as I/O port		Input state	Output state	—		Can be used as I/O port		Input state	Output state
PA6DDR	0	1	0	—	0	1	0	1	0	—
Pin function	PA6 input	PA6 output	SSI0-B input*2	SSI0-B output*3	PA6 input	A22 output	PA6 input	PA6 output	SSI0-B input*2	SSI0-B output*3
	$\overline{\text{IRQ6-A}}$ interrupt input*1									

Notes: 1. $\overline{\text{IRQ6-A}}$ input when the ITS6 bit in ITSR is 0.
2. SSI0-B input when SSI0S1 and SSI0S0 = B'01 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'001×1 or B'10××1.
3. SSI0-B output when SSI0S1 and SSI0S0 = B'01 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'0001×.

	SSI pin settings															
SSUMS	0						0				1					
BIDE	0						1				0					
MSS	0			1			0		1		0			1		
TE	0	1		0	1		0	1	0	1	0	1		0	1	
RE	1	0	1	1	0	1	1	0	1	0	1	0	1	1	0	1
Pin state	—	Output	Output	Input	—	Input	—	—	—	—	Input	—	Input	Input	—	Input

Legend:

—: Pin is not used by the SSU (can be used as I/O port)

- PA5/A21/$\overline{\text{IRQ5-A}}$/SSCK0-B

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits MSS and SCKS in SSCRH and bit SSUMS in SSCRL of SSU, bit A21E in PFCR1, bits SSCK0S1 and SSCK0S0 in PFCR5, bit PA5DDR, and bit ITS5 in ITSR of the interrupt controller

- Modes 1, 2, and 4

A21E	0				1	
SSU settings	Can be used as I/O port		Input state	Output state	—	
PA5DDR	0	1	0	—	0	1
Pin function	PA5 input	PA5 output	SSCK0-B input*2	SSCK0-B output*3	PA5 input	A21 output
	$\overline{\text{IRQ5-A}}$ interrupt input*1					
	SSCK0-B input*2					

- Mode 7

EXPE	1						0			
A21E	0				1		—			
SSU settings	Can be used as I/O port		Input state	Output state	—		Can be used as I/O port		Input state	Output state
PA5DDR	0	1	0	—	0	1	0	1	0	—
Pin function	PA5 input	PA5 output	SSCK0-B input*[2]	SSCK0-B output*[3]	PA5 input	A21 output	PA5 input	PA5 output	SSCK0-B input*[2]	SSCK0-B output*[3]
	$\overline{\text{IRQ5-A}}$ interrupt input*[1]									
	SSCK0-B input*[2]									

Notes: 1. $\overline{\text{IRQ5-A}}$ input when the ITS5 bit in ITSR is 0.
2. SSCK0-B input when SSCK0S1 and SSCK0S0 = B'00 in PFCR5, and SSUMS, MSS, and SCKS = B'001 or B'101.
3. SSCK0-B output when SSCK0S1 and SSCK0S0 = B'00 in PFCR5, and SSUMS, MSS, and SCKS = B'×11.

	SSCK pin settings							
SSUMS	0				1			
MSS	0		1		0		1	
SCKS	0	1	0	1	0	1	0	1
Pin state	—	Input	—	Output	—	Input	—	Output

Legend:
—: Pin is not used by the SSU (can be used as I/O port)

- PA4/A20/$\overline{\text{IRQ4-A}}$/SCS0-B

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits MSS, CSS1, and CSS0 in SSCRH and bit SSUMS in SSCRL of SSU, bit A20E in PFCR1, bit PA4DDR, and bit ITS4 in ITSR of the interrupt controller.

Operating mode	1, 2	4					
EXPE	—	—					
A20E	—	0				1	
SSU settings	—	Can be used as I/O port		Input state	Output state	—	
PA4DDR	—	0	1	0	—	0	1
Pin function	A20 output	PA4 input	PA4 output	$\overline{\text{SCS0-B}}$ input*2	$\overline{\text{SCS0-B}}$ output*3	PA4 input	A20 output
	$\overline{\text{IRQ4-A}}$ interrupt input*1						

Operating mode	7									
EXPE	0				1					
A20E	—				0				1	
SSU settings	Can be used as I/O port		Input state	Output state	Can be used as I/O port		Input state	Output state	—	
PA4DDR	0	1	0	—	0	1	0	—	0	1
Pin function	PA4 input	PA4 output	$\overline{\text{SCS0-B}}$ input*2	$\overline{\text{SCS0-B}}$ output*3	PA4 input	PA4 output	$\overline{\text{SCS0-B}}$ input*2	$\overline{\text{SCS0-B}}$ output*3	PA4 input	A20 output
	$\overline{\text{IRQ4-A}}$ interrupt input*1									

Notes: 1. $\overline{\text{IRQ4-A}}$ input when the ITS4 bit in ITSR is 0.
2. $\overline{\text{SCSO-B}}$ input when SCS0S1 and SCS0S0 = B'01 in PFCR5, and SSUMS, MSS, CSS1, and CSS0 = B'00××, B'0101, or B'0110.
3. $\overline{\text{SCSO-B}}$ output when SCS0S1 and SCS0S0 = B'01 in PFCR5, and SSUMS, MSS, CSS1, and CSS0 = B'011×.

	$\overline{SCS}$ pin settings					
SSUMS	0					1
MSS	0	1				×
CSS1	×	0		1		×
CSS0	×	0	1	0	1	×
Pin state	Input	—	Input	Automatic I/O	Output	—

Legend:

×: Don't care

—: Pin is not used by the SSU (can be used as I/O port)

- PA3/A19/SCK4-B

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit C/$\overline{A}$ in SMR_4 and bits CKE0 and CKE1 in SCR_4 of SCI, bit A19E in PFCR1, bit SCK4S in PFCR4, and bit PA3DDR.

Operating mode	1, 2	4						
EXPE	—	—						
A19E	—	0					1	
CKE1	—	0				1	—	
C/$\overline{A}$	—	0			1	—	—	
CKE0	—	0		1	—	—	—	
PA3DDR	—	0	1	—	—	—	0	1
Pin function	A19 output	PA3 input	PA3 output	SCK4-B output*	SCK4-B output*	SCK4-B input*	PA3 input	A19 output

Operating mode	7											
EXPE	0					1						
A19E	—					0					1	
CKE1	0				1	0				1	—	
C/$\overline{A}$	0			1	—	0			1	—	—	
CKE0	0		1	—	—	0		1	—	—	—	
PA3DDR	0	1	—	—	—	0	1	—	—	—	0	1
Pin function	PA3 input	PA3 output	SCK4-B output*	SCK4-B output*	SCK4-B input*	PA3 input	PA3 output	SCK4-B output*	SCK4-B output*	SCK4-B input*	PA3 input	A19 output

Note: * SCK4-B input/output when the SCK4S bit in PFCR4 is 1.

- PA2/A18/RxD4-B

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit RE in SCR_4 of SCI, bit A18E in PFCR1, bit RXD4S in PFCR4, and bit PA2DDR.

Operating mode	1, 2	4					7							
EXPE	—	—					0			1				
A18E	—	0			1		—			0			1	
RE	—	0		1	—		0		1	0		1	—	
PA2DDR	—	0	1	—	0	1	0	1	—	0	1	—	0	1
Pin function	A18 output	PA2 input	PA2 output	RxD4-B input*	PA2 input	A18 output	PA2 input	PA2 output	RxD4-B input*	PA2 input	PA2 output	RxD4-B input*	PA2 input	A18 output

Note: * RxD4-B input when the RXD4S bit in PFCR4 is 1.

- PA1/A17/TxD4-B

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit TE in SCR_4 of SCI, bit A17E in PFCR1, bit TXD4S in PFCR4, and bit PA1DDR.

Operating mode	1, 2	4					7							
EXPE	—	—					0			1				
A17E	—	0			1		—			0			1	
TE	—	0		1	—		0		1	0		1	—	
PA1DDR	—	0	1	—	0	1	0	1	—	0	1	—	0	1
Pin function	A17 output	PA1 input	PA1 output	TxD4-B output*	PA1 input	A17 output	PA1 input	PA1 output	TxD4-B output*	PA1 input	PA1 output	TxD4-B output*	PA1 input	A17 output

Note: * TxD4-A output when the TXD4S bit in PFCR4 is 1.

- PA0/A16

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit A16E in PFCR1, and bit PA0DDR.

Operating mode	1, 2	4				7					
EXPE	—	—				0		1			
A16E	—	0		1		—		0		1	
PA0DDR	—	0	1	0	1	0	1	0	1	0	1
Pin function	A16 output	PA0 input	PA0 output	PA0 input	A16 output	PA0 input	PA0 output	PA0 input	PA0 output	PA0 input	A16 output

10.9.7 Port A Input Pull-Up MOS States

Port A has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used by pins PA7 to PA5 in modes 1 and 2, and by all pins in modes 4 and 7. The input pull-up MOS can be specified as on or off on a bit-by-bit basis.

Table 10.3 summarizes the input pull-up MOS states.

Table 10.3 Input Pull-Up MOS States for Port A

Mode		Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
4 or 7	PA7 to PA0	Off	Off	On/Off	On/Off
1 or 2	PA7 to PA5			On/Off	On/Off
	PA4 to PA0			Off	Off

Legend:
Off: Input pull-up MOS is always off.
On/Off: Input pull-up MOS is on when in input port register state* and PAPCR = 1; otherwise off.
Note: * Not available with SSU/SCI input

10.10 Port B

Port B is an 8-bit I/O port that also has other functions. Port B has the following registers.

- Port B data direction register (PBDDR)
- Port B data register (PBDR)
- Port B register (PORTB)
- Port B pull-up MOS control register (PBPCR)
- Port B open drain control register (PBODR)

10.10.1 Port B Data Direction Register (PBDDR)

The individual bits of PBDDR specify input or output for the pins of port B. PBDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DDR	0	W	• Modes 1 and 2 Port B pins are address outputs regardless of the PBDDR settings. • Modes 7 (when EXPE = 1) and 4 Setting a PBDDR bit to 1 makes the corresponding pin an address output, while clearing a PBDDR bit to 0 makes the corresponding pin an input port. • Mode 7 (when EXPE = 0) Port B is an I/O port, and its pin functions can be switched with PBDDR.
6	PB6DDR	0	W	
5	PB5DDR	0	W	
4	PB4DDR	0	W	
3	PB3DDR	0	W	
2	PB2DDR	0	W	
1	PB1DDR	0	W	
0	PB0DDR	0	W	

10.10.2 Port B Data Register (PBDR)

PBDR stores output data for the port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

10.10.3 Port B Register (PORTB)

PORTB shows the pin states of port B. PORTB cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7	—*	R	If this register is read while a PBDDR bit is set to 1, the corresponding PBDR value is read. If this register is read while a PBDDR bit is cleared to 0, the corresponding pin state is read.
6	PB6	—*	R	
5	PB5	—*	R	
4	PB4	—*	R	
3	PB3	—*	R	
2	PB2	—*	R	
1	PB1	—*	R	
0	PB0	—*	R	

Note: * Determined by the states of pins PB7 to PB0.

10.10.4 Port B Pull-Up MOS Control Register (PBPCR)

PBPCR controls on/off of the input pull-up MOS for port B. PBPCR is valid in modes 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PCR	0	R/W	When in a input port register state, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PB6PCR	0	R/W	
5	PB5PCR	0	R/W	
4	PB4PCR	0	R/W	
3	PB3PCR	0	R/W	
2	PB2PCR	0	R/W	
1	PB1PCR	0	R/W	
0	PB0PCR	0	R/W	

10.10.5 Port B Open Drain Control Register (PBODR)

PBODR specifies the output type of each port B pin.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7ODR	0	R/W	When not specified for address output, setting a PBODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a PBODR bit to 0 makes the corresponding pin a CMOS output pin.
6	PB6ODR	0	R/W	
5	PB5ODR	0	R/W	
4	PB4ODR	0	R/W	
3	PB3ODR	0	R/W	
2	PB2ODR	0	R/W	
1	PB1ODR	0	R/W	
0	PB0ODR	0	R/W	

10.10.6 Pin Functions

Port B pins also function as the pins for TPU I/Os and address outputs. The correspondence between the register specification and the pin functions is shown below.

- PB7/A15/TIOCB8/TCLKH

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 8 settings (by bits MD3 to MD0 in TMDR_8, bits IOB3 to IOB0 in TIOR_8, and bits CCLR1 and CCLR0 in TCR_8), bits TPSC2 to TPSC0 in TCR_6 and TCR_11, and bit PB7DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 8 settings	—	—		(1) in table below	(2) in table below	
PB7DDR	—	0	1	—	0	1
Pin function	A15 output	PB7 input	A15 output	TIOCB8 output	PB7 input	PB7 output
					TIOCB8 input*[1]	
				TCLKH input*[2]		

TPU channel 8 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCB8 input when MD3 to MD0 = B'0000 or B'01×× and IOB3 = 1.

2. TCLKH input when the setting for either TCR_6 or TCR_11 is TPSC2 to TPSC0 = B'111. TCLKH input when phase counting mode is set for channels 8 and 10.

- PB6/A14/TIOCA8

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 8 settings (by bits MD3 to MD0 in TMDR_8, bits IOA3 to IOA0 in TIOR_8, and bits CCLR1 and CCLR0 in TCR_8), and bit PB6DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 8 settings	—	—		(1) in table below	(2) in table below	
PB6DDR	—	0	1	—	0	1
Pin function	A14 output	PB6 input	A14 output	TIOCA8 output	PB6 input	PB6 output
					TIOCA8 input*[1]	

TPU channel 8 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00		
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA8 input when MD3 to MD0 = B'0000 or B'01×× and IOA3 = 1.

2. TIOCB8 output disabled.

- PB5/A13/TIOCB7/TCLKG

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 7 settings (by bits MD3 to MD0 in TMDR_7, bits IOB3 to IOB0 in TIOR_7, and bits CCLR1 and CCLR0 in TCR_7), bits TPSC2 to TPSC0 in TCR_6, TCR_8, TCR_10, and TCR_11, and bit PB5DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 7 settings	—	—		(1) in table below	(2) in table below	
PB5DDR	—	0	1	—	0	1
Pin function	A13 output	PB5 input	A13 output	TIOCB7 output	PB5 input	PB5 output
					TIOCB7 input*[1]	
				TCLKG input*[2]		

TPU channel 7 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

×: Don’t care

Notes: 1. TIOCB7 input when MD3 to MD0 = B'0000 or B'01×× and IOB3 to IOB0 = B'10××.

2. TCLKG input when the setting for either TCR_6 or TCR_8 is TPSC2 to TPSC0 = B'111, or when the setting for either TCR_10 or TCR_11 is TPSC2 to TPSC0 = B'101. TCLKG input when phase counting mode is set for channels 8 and 10.

- PB4/A12/TIOCA7

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 7 settings (by bits MD3 to MD0 in TMDR_7, bits IOA3 to IOA0 in TIOR_7, and bits CCLR1 and CCLR0 in TCR_7), and bit PB4DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 7 settings	—	—		(1) in table below	(2) in table below	
PB4DDR	—	0	1	—	0	1
Pin function	A12 output	PB4 input	A12 output	TIOCA7 output	PB4 input	PB4 output
					TIOCA7 input*[1]	

TPU channel 7 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00		
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA7 input when MD3 to MD0 = B'0000 or B'01×× and IOA3 to IOA0 = B'10××.

2. TIOCB7 output disabled.

- PB3/A11/TIOCD6/TCLKF

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 6 settings (by bits MD3 to MD0 in TMDR_6, bits IOD3 to IOD0 in TIORL_6, and bits CCLR2 to CCLR0 in TCR_6), bits TPSC2 to TPSC0 in TCR_6 to TCR_8, and bit PB3DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 6 settings	—	—		(1) in table below	(2) in table below	
PB3DDR	—	0	1	—	0	1
Pin function	A11 output	PB3 input	A11 output	TIOCD6 output	PB3 input	PB3 output
					TIOCD6 input*[1]	
				TCLKF input*[2]		

TPU channel 6 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOD3 to IOD0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

×: Don’t care

Notes: 1. TIOCD6 input when MD3 to MD0 = B'0000 and IOD3 to IOD0 = B'10××.

2. TCLKF input when the setting for any of TCR_6 to TCR_8 is TPSC2 to TPSC0 = B'101. TCLKF input when phase counting mode is set for channels 7 and 11.

- PB2/A10/TIOCC6/TCLKE

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 6 settings (by bits MD3 to MD0 in TMDR_6, bits IOC3 to IOC0 in TIORL_6, and bits CCLR2 to CCLR0 in TCR_6), bits TPSC2 to TPSC0 in TCR_6 to TCR_11, and bit PB2DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 6 settings	—	—		(1) in table below	(2) in table below	
PB2DDR	—	0	1	—	0	1
Pin function	A10 output	PB2 input	A10 output	TIOCC6 output	PB2 input	PB2 output
					TIOCC6 input*[1]	
				TCLKE input*[2]		

TPU channel 6 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001×	B'0010	B'0011	
IOC3 to IOC0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00		
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM*[3] mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCC6 input when MD3 to MD0 = B'0000 and IOC3 to IOC0 = B'10××.

2. TCLKE input when the setting for any of TCR_6 to TCR_11 is TPSC2 to TPSC0 = B'100. TCLKE input when phase counting mode is set for channels 7 and 11.

3. TIOCD6 output disabled. Output disabled and settings (2) effective when BFA = 1 or BFB = 1 in TMDR_6.

- PB1/A9/TIOCB6

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 6 settings (by bits MD3 to MD0 in TMDR_6, bits IOB3 to IOB0 in TIORH_6, and bits CCLR2 to CCLR0 in TCR_6), and bit PB1DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 6 settings	—	—		(1) in table below	(2) in table below	
PB1DDR	—	0	1	—	0	1
Pin function	A9 output	PB1 input	A9 output	TIOCB6 output	PB1 input	PB1 output
					TIOCB6 input*	

TPU channel 6 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

×: Don't care

Note: * TIOCB6 input when MD3 to MD0 = B'0000 and IOB3 to IOB0 = B'10××.

- PB0/A8/TIOCA6

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 6 settings (by bits MD3 to MD0 in TMDR_6, bits IOA3 to IOA0 in TIORH_6, and bits CCLR2 to CCLR0 in TCR_6), and bit PB0DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 6 settings	—	—		(1) in table below	(2) in table below	
PB0DDR	—	0	1	—	0	1
Pin function	A8 output	PB0 input	A8 output	TIOCA6 output	PB0 input	PB0 output
					TIOCA6 input*[1]	

TPU channel 6 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00		
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCC6 input when MD3 to MD0 = B'0000 and IOA3 to IOA0 = B'10××.

2. TIOCB6 output disabled.

10.10.7 Port B Input Pull-Up MOS States

Port B has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used in modes 4 and 7. The input pull-up MOS can be specified as on or off on a bit-by-bit basis.

In modes 4 and 7, when a PBDDR bit is cleared to 0, setting the corresponding PBPCR bit to 1 turns on the input pull-up MOS for that pin.

Table 10.4 summarizes the input pull-up MOS states.

Table 10.4 Input Pull-Up MOS States for Port B

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1 or 2	Off	Off	Off	Off
4 or 7			On/Off	On/Off

Legend:
Off: Input pull-up MOS is always off.
On/Off: Input pull-up MOS is on when in a input port state 0 and PBPCR = 1; otherwise off.

10.11 Port C

Port C is an 8-bit I/O port that also has other functions. Port C has the following registers.

- Port C data direction register (PCDDR)
- Port C data register (PCDR)
- Port C register (PORTC)
- Port C pull-up MOS control register (PCPCR)
- Port C open drain control register (PCODR)

10.11.1 Port C Data Direction Register (PCDDR)

The individual bits of PCDDR specify input or output for the pins of port C. PCDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	• Modes 1 and 2 Port C pins are address outputs regardless of the PCDDR settings. • Modes 7 (when EXPE = 1) and 4 Setting a PCDDR bit to 1 makes the corresponding pin an address output, while clearing a PCDDR to 0 makes the corresponding pin an input port. • Mode 7 (when EXPE = 0) Port C is an I/O port, and its pin functions can be switched with PCDDR.
6	PC6DDR	0	W	
5	PC5DDR	0	W	
4	PC4DDR	0	W	
3	PC3DDR	0	W	
2	PC2DDR	0	W	
1	PC1DDR	0	W	
0	PC0DDR	0	W	

10.11.2 Port C Data Register (PCDR)

PCDR stores output data for the port C pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

10.11.3 Port C Register (PORTC)

PORTC shows the pin states of port C. PORTC cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7	—*	R	If this register is read while a PCDDR bit is set to 1, the corresponding PCDR value is read. If this register is read while a PCDDR bit is cleared to 0, the corresponding pin state is read.
6	PC6	—*	R	
5	PC5	—*	R	
4	PC4	—*	R	
3	PC3	—*	R	
2	PC2	—*	R	
1	PC1	—*	R	
0	PC0	—*	R	

Note: * Determined by the states of pins PC7 to PC0.

10.11.4 Port C Pull-Up MOS Control Register (PCPCR)

PCPCR controls on/off of the input pull-up MOS for port C. PCPCR is valid in modes 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PCR	0	R/W	When in a input port state, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PC6PCR	0	R/W	
5	PC5PCR	0	R/W	
4	PC4PCR	0	R/W	
3	PC3PCR	0	R/W	
2	PC2PCR	0	R/W	
1	PC1PCR	0	R/W	
0	PC0PCR	0	R/W	

10.11.5 Port C Open Drain Control Register (PCODR)

PCODR specifies the output type of each port C pin.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7ODR	0	R/W	When not specified for address output, setting a PCODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a PCODR bit to 0 makes the corresponding pin a CMOS output pin.
6	PC6ODR	0	R/W	
5	PC5ODR	0	R/W	
4	PC4ODR	0	R/W	
3	PC3ODR	0	R/W	
2	PC2ODR	0	R/W	
1	PC1ODR	0	R/W	
0	PC0ODR	0	R/W	

10.11.6 Pin Functions

Port C pins also function as the pins for TPU I/Os and address outputs. The correspondence between the register specification and the pin functions is shown below.

- PC7/A7/TIOCB11

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 11 settings (by bits MD3 to MD0 in TMDR_11, bits IOB3 to IOB0 in TIOR_11, and bits CCLR1 and CCLR0 in TCR_11), and bit PC7DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 11 settings	—	—		(1) in table below	(2) in table below	
PC7DDR	—	0	1	—	0	1
Pin function	A7 output	PC7 input	A7 output	TIOCB11 output	PC7 input	PC7 output
					TIOCB11 input*	

TPU channel 11 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

×: Don't care

Note: * TIOCB11 input when MD3 to MD0 = B'0000 or B'01×× and IOB3 = 1.

- PC6/A6/TIOCA11

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 11 settings (by bits MD3 to MD0 in TMDR_11, bits IOA3 to IOA0 in TIOR_11, and bits CCLR1 and CCLR0 in TCR_11), and bit PC6DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 11 settings	—	—		(1) in table below	(2) in table below	
PC6DDR	—	0	1	—	0	1
Pin function	A6 output	PC6 input	A6 output	TIOCA11 output	PC6 input	PC6 output
					TIOCA11 input*[1]	

TPU channel 11 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00		
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA11 input when MD3 to MD0 = B'0000 or B'01×× and IOA3 = 1.

2. TIOCB11 output disabled.

- PC5/A5/TIOCB10

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 10 settings (by bits MD3 to MD0 in TMDR_10, bits IOB3 to IOB0 in TIOR_10, and bits CCLR1 and CCLR0 in TCR_10), and bit PC5DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 10 settings	—	—		(1) in table below	(2) in table below	
PC5DDR	—	0	1	—	0	1
Pin function	A5 output	PC5 input	A5 output	TIOCB10 output	PC5 input	PC5 output
					TIOCB10 input*	

TPU channel 10 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

×: Don't care

Note: * TIOCB10 input when MD3 to MD0 = B'0000 or B'01×× and IOB3 to IOB0 = B'10××.

- PC4/A4/TIOCA10

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 10 settings (by bits MD3 to MD0 in TMDR_10, bits IOA3 to IOA0 in TIOR_10, and bits CCLR1 and CCLR0 in TCR_10), and bit PC4DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 10 settings	—	—		(1) in table below	(2) in table below	
PC4DDR	—	0	1	—	0	1
Pin function	A4 output	PC4 input	A4 output	TIOCA10 output	PC4 input	PC4 output
					TIOCA10 input*[1]	

TPU channel 10 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00		
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA10 input when MD3 to MD0 = B'0000 or B'01×× and IOA3 to IOA0 = B'10××.

2. TIOCB10 output disabled.

- PC3/A3/TIOCD9

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 9 settings (by bits MD3 to MD0 in TMDR_9, bits IOD3 to IOD0 in TIORL_9, and bits CCLR2 to CCLR0 in TCR_9), and bit PC3DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 9 settings	—	—		(1) in table below	(2) in table below	
PC3DDR	—	0	1	—	0	1
Pin function	A3 output	PC3 input	A3 output	TIOCD9 output	PC3 input	PC3 output
					TIOCD9 input*	

TPU channel 9 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOD3 to IOD0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

×: Don't care

Note: * TIOCD9 input when MD3 to MD0 = B'0000 and IOD3 to IOD0 = B'10××.

- PC2/A2/TIOCC9

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 9 settings (by bits MD3 to MD0 in TMDR_9, bits IOC3 to IOC0 in TIORL_9, and bits CCLR2 to CCLR0 in TCR_9), and bit PC2DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 9 settings	—	—		(1) in table below	(2) in table below	
PC2DDR	—	0	1	—	0	1
Pin function	A2 output	PC2 input	A2 output	TIOCC9 output	PC2 input	PC2 output
					TIOCC9 input*[1]	

TPU channel 9 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001×	B'0010	B'0011	
IOC3 to IOC0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00		
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCC9 input when MD3 to MD0 = B'0000 and IOC3 to IOC0 = B'10××.

2. TIOCD9 output disabled. Output disabled and settings (2) effective when BFA = 1 or BFB = 1 in TMDR_9.

- PC1/A1/TIOCB9

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 9 settings (by bits MD3 to MD0 in TMDR_9, bits IOB3 to IOB0 in TIORH_9, and bits CCLR2 to CCLR0 in TCR_9), and bit PC1DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 9 settings	—	—		(1) in table below	(2) in table below	
PC1DDR	—	0	1	—	0	1
Pin function	A1 output	PC1 input	A1 output	TIOCB9 output	PC1 input	PC1 output
					TIOCB9 input*	

TPU channel 9 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	—	B'××00	Other than B'××00	
CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

×: Don't care

Note: * TIOCB9 input when MD3 to MD0 = B'0000 and IOB3 to IOB0 = B'10××.

- PC0/A0/TIOCA9

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, TPU channel 9 settings (by bits MD3 to MD0 in TMDR_9, bits IOA3 to IOA0 in TIORH_9, and bits CCLR2 to CCLR0 in TCR_9), and bit PC0DDR.

Operating mode	1, 2	4, 7 (EXPE = 1)		7 (EXPE = 0)		
TPU channel 9 settings	—	—		(1) in table below	(2) in table below	
PC0DDR	—	0	1	—	0	1
Pin function	A0 output	PC0 input	A0 output	TIOCA9 output	PC0 input	PC0 output
					TIOCA9 input*[1]	

TPU channel 9 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000, B'0100, B'1×××	B'0001 to B'0011, B'0101 to B'0111	B'××00	Other than B'××00		
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

Legend:

×: Don't care

Notes: 1. TIOCA9 input when MD3 to MD0 = B'0000 and IOA3 to IOA0 = B'10××.

2. TIOCB9 output disabled.

10.11.7 Port C Input Pull-Up MOS States

Port C has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used in modes 4 and 7. The input pull-up MOS can be specified as on or off on a bit-by-bit basis.

In modes 4 and 7, when a PCDDR bit is cleared to 0, setting the corresponding PCPCR bit to 1 turns on the input pull-up MOS for that pin.

Table 10.5 summarizes the input pull-up MOS states.

Table 10.5 Input Pull-Up MOS States for Port C

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1 or 2	Off	Off	Off	Off
4 or 7			On/Off	On/Off

Legend:
Off: Input pull-up MOS is always off.
On/Off: Input pull-up MOS is on when in a input port state and PCPCR = 1; otherwise off.

10.12 Port D

Port D is an 8-bit I/O port that also has other functions. Port D has the following registers.

- Port D data direction register (PDDDR)
- Port D data register (PDDR)
- Port D register (PORTD)
- Port D pull-up MOS control register (PDPCR)
- Port D open drain control register (PDODR)

10.12.1 Port D Data Direction Register (PDDDR)

The individual bits of PDDDR specify input or output for the pins of port D. PDDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DDR	0	W	• Modes 7 (when EXPE = 1), 1, 2, and 4 Port D is automatically designated for data input/output. • Mode 7 (when EXPE = 0) Port D is an I/O port, and its pin functions can be switched with PDDDR.
6	PD6DDR	0	W	
5	PD5DDR	0	W	
4	PD4DDR	0	W	
3	PD3DDR	0	W	
2	PD2DDR	0	W	
1	PD1DDR	0	W	
0	PD0DDR	0	W	

10.12.2 Port D Data Register (PDDR)

PDDR stores output data for the port D pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

10.12.3 Port D Register (PORTD)

PORTD shows the pin states of port D. PORTD cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7	—*	R	If this register is read while a PDDDR bit is set to 1, the corresponding PDDR value is read. If this register is read while a PDDDR bit is cleared to 0, the corresponding pin state is read.
6	PD6	—*	R	
5	PD5	—*	R	
4	PD4	—*	R	
3	PD3	—*	R	
2	PD2	—*	R	
1	PD1	—*	R	
0	PD0	—*	R	

Note: * Determined by the states of pins PD7 to PD0.

10.12.4 Port D Pull-Up MOS Control Register (PDPCR)

PDPCR controls on/off of the input pull-up MOS for port D. PDPCR is valid in mode 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PCR	0	R/W	When PDDDR = 0 (input port), setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PD6PCR	0	R/W	
5	PD5PCR	0	R/W	
4	PD4PCR	0	R/W	
3	PD3PCR	0	R/W	
2	PD2PCR	0	R/W	
1	PD1PCR	0	R/W	
0	PD0PCR	0	R/W	

10.12.5 Port D Open Drain Control Register (PDODR)

PDODR specifies the output type of each port D pin.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7ODR	0	R/W	When not specified for data output, setting a PDODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a PDODR bit to 0 makes the corresponding pin a CMOS output pin.
6	PD6ODR	0	R/W	
5	PD5ODR	0	R/W	
4	PD4ODR	0	R/W	
3	PD3ODR	0	R/W	
2	PD2ODR	0	R/W	
1	PD1ODR	0	R/W	
0	PD0ODR	0	R/W	

10.12.6 Pin Functions

Port D pins also function as the pins for data I/Os. The correspondence between the register specification and the pin functions is shown below.

- PD7/D15, PD6/D14, PD5/D13, PD4/D12, PD3/D11, PD2/D10, PD1/D9, PD0/D8
  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, and bit PDnDDR.

Operating mode	1, 2, 4	7		
EXPE	—	0		1
PDnDDR	—	0	1	—
Pin function	Data I/O	PDn input	PDn output	Data I/O

Legend:
n = 7 to 0

10.12.7 Port D Input Pull-Up MOS States

Port D has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used in mode 7. The input pull-up MOS can be specified as on or off on a bit-by-bit basis.

In mode 7, when a PDDDR bit is cleared to 0, setting the corresponding PDPCR bit to 1 turns on the input pull-up MOS for that pin.

Table 10.6 summarizes the input pull-up MOS states.

Table 10.6 Input Pull-Up MOS States for Port D

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1, 2, or 4	Off	Off	Off	Off
7			On/Off	On/Off

Legend:
Off: Input pull-up MOS is always off.
On/Off: Input pull-up MOS is on when PDDDR = 0 and PDPCR = 1; otherwise off.

10.13 Port E

Port E is an 8-bit I/O port that also has other functions. Port E has the following registers.

- Port E data direction register (PEDDR)
- Port E data register (PEDR)
- Port E register (PORTE)
- Port E pull-up MOS control register (PEPCR)
- Port E open drain control register (PEODR)

10.13.1 Port E Data Direction Register (PEDDR)

The individual bits of PEDDR specify input or output for the pins of port E. PEDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	• Modes 1, 2, and 4 When 8-bit bus mode is selected, port E is an I/O port, and its pin functions can be switched with PEDDR. When 16-bit bus mode is selected, port E is designated for data input/output. For details on 8-bit and 16-bit bus modes, see section 6, Bus Controller (BSC). • Mode 7 (when EXPE = 1) When 8-bit bus mode is selected, port E is an I/O port. Setting a PEDDR bit to 1 makes the corresponding pin an output port, while clearing a PEDDR bit to 0 makes the corresponding pin an input port. When 16-bit bus mode is selected, port E is designated for data input/output. • Mode 7 (when EXPE = 0) Port E is an I/O port, and its pin functions can be switched with PEDDR.
6	PE6DDR	0	W	
5	PE5DDR	0	W	
4	PE4DDR	0	W	
3	PE3DDR	0	W	
2	PE2DDR	0	W	
1	PE1DDR	0	W	
0	PE0DDR	0	W	

10.13.2 Port E Data Register (PEDR)

PEDR stores output data for the port E pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
6	PE6DR	0	R/W	
5	PE5DR	0	R/W	
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

10.13.3 Port E Register (PORTE)

PORTE shows the pin states of port E. PORTE cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7	—*	R	If this register is read while a PEDDR bit is set to 1, the corresponding PEDR value is read. If this register is read while a PEDDR bit is cleared to 0, the corresponding pin state is read.
6	PE6	—*	R	
5	PE5	—*	R	
4	PE4	—*	R	
3	PE3	—*	R	
2	PE2	—*	R	
1	PE1	—*	R	
0	PE0	—*	R	

Note: * Determined by the states of pins PE7 to PE0.

10.13.4 Port E Pull-Up MOS Control Register (PEPCR)

PEPCR controls on/off of the input pull-up MOS for port E. PEPCR is valid in 8-bit bus mode.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7PCR	0	R/W	When PEDDR = 0 (input port), setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PE6PCR	0	R/W	
5	PE5PCR	0	R/W	
4	PE4PCR	0	R/W	
3	PE3PCR	0	R/W	
2	PE2PCR	0	R/W	
1	PE1PCR	0	R/W	
0	PE0PCR	0	R/W	

10.13.5 Port E Open Drain Control Register (PEODR)

PEODR specifies the output type of each port E pin.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7ODR	0	R/W	When not specified for data output, setting a PEODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a PEODR bit to 0 makes the corresponding pin a CMOS output pin.
6	PE6ODR	0	R/W	
5	PE5ODR	0	R/W	
4	PE4ODR	0	R/W	
3	PE3ODR	0	R/W	
2	PE2ODR	0	R/W	
1	PE1ODR	0	R/W	
0	PE0ODR	0	R/W	

10.13.6 Pin Functions

Port E pins also function as the pins for data I/Os. The correspondence between the register specification and the pin functions is shown below.

- PE7/D7, PE6/D6, PE5/D5, PE4/D4, PE3/D3, PE2/D2, PE1/D1, PE0/D0
  The pin function is switched as shown below according to the combination of the operating mode, bus mode, bit EXPE, and bit PEnDDR.

Operating mode	1, 2, 4			7				
Bus mode	All areas are 8-bit space		At least one area is 16-bit space	—		All areas are 8-bit space		At least one area is 16-bit space
EXPE	—		—	0		1		1
PEnDDR	0	1	—	0	1	0	1	—
Pin function	PEn input	PEn output	Data I/O	PEn input	PEn output	PEn input	PEn output	Data I/O

Legend:
n = 7 to 0

10.13.7 Port E Input Pull-Up MOS States

Port E has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used in 8-bit bus mode. The input pull-up MOS can be specified as on or off on a bit-by-bit basis. In 8-bit bus mode, when a PEDDR bit is cleared to 0, setting the corresponding PEPCR bit to 1 turns on the input pull-up MOS for that pin.

Table 10.7 summarizes the input pull-up MOS states.

Table 10.7 Input Pull-Up MOS States for Port E

Mode		Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1, 2, or 4	8-bit bus	Off	Off	On/Off	On/Off
	16-bit bus			Off	Off

Legend:
Off: Input pull-up MOS is always off.
On/Off: Input pull-up MOS is on when PEDDR = 0 and PEPCR = 1; otherwise off.

10.14 Port F

Port F is an 8-bit I/O port that also has other functions. Port F has the following registers. For the port function control registers, refer to section 10.18, Port Function Control Registers.

- Port F data direction register (PFDDR)
- Port F data register (PFDR)
- Port F register (PORTF)
- Port function control register 0 (PFCR0)
- Port function control register 2 (PFCR2)
- Port function control register 4 (PFCR4)
- Port function control register 5 (PFCR5)
- Port F open drain control register (PFODR)

10.14.1 Port F Data Direction Register (PFDDR)

The individual bits of PFDDR specify input or output for the pins of port F. PFDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DDR	1/0*	W	• Modes 7 (when EXPE = 1), 1, 2, and 4 Pin PF7 functions as the ϕ output pin when the corresponding PFDDR bit is set to 1, and as an input port when the bit is cleared to 0. Pin PF6 functions as the $\overline{AS}$ output pin when the ASOE bit is set to 1. When the ASOE bit is cleared to 0, pin PF6 is an I/O port and its function can be switched with PF6DDR. Pins PF5 and PF4 are automatically designated as bus control outputs ($\overline{RD}$ and $\overline{HWR}$). Pin PF3 functions as the $\overline{LWR}$ output pin when the LWROE bit is set to 1. When the LWROE bit is cleared to 0, pin PF3 is an I/O port and its function can be switched with PF3DDR. Pins PF2 to PF0 function as bus control input/output pins ($\overline{LCAS}$, $\overline{UCAS}$, and $\overline{WAIT}$) when the appropriate bus controller settings are made. Otherwise, these pins are output ports when the corresponding PFDDR bits are set to 1 and are input ports when the bits are cleared to 0. • Mode 7 (when EXPE = 0) Pin PF7 functions as the ϕ output pin when the corresponding PFDDR bit is set to 1, and as an input port when the bit is cleared to 0. Pins PF6 to PF0 are I/O ports, and their functions can be switched with PFDDR.
6	PF6DDR	0	W	
5	PF5DDR	0	W	
4	PF4DDR	0	W	
3	PF3DDR	0	W	
2	PF2DDR	0	W	
1	PF1DDR	0	W	
0	PF0DDR	0	W	

10.14.2 Port F Data Register (PFDR)

PFDR stores output data for the port F pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
6	PF6DR	0	R/W	
5	PF5DR	0	R/W	
4	PF4DR	0	R/W	
3	PF3DR	0	R/W	
2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

10.14.3 Port F Register (PORTF)

PORTF shows the pin states of port F. PORTF cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7	—*	R	If this register is read while a PFDDR bit is set to 1, the corresponding PFDR value is read. If this register is read while a PFDDR bit is cleared to 0, the corresponding pin state is read.
6	PF6	—*	R	
5	PF5	—*	R	
4	PF4	—*	R	
3	PF3	—*	R	
2	PF2	—*	R	
1	PF1	—*	R	
0	PF0	—*	R	

Note: * Determined by the states of pins PF7 to PF0.

10.14.4 Port F Open Drain Control Register (PFODR)

PFODR specifies the output type of each port F pin.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7ODR	0	R/W	When not specified for $\overline{AS}$, $\overline{AH}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$, $\overline{LCAS}$, $\overline{UCAS}$, DQML, DQMU, $\overline{CS5}$, $\overline{CS6}$, or $\overline{OE}$-A output, setting a PFODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a PFODR bit to 0 makes the corresponding pin a CMOS output pin.
6	PF6ODR	0	R/W	
5	PF5ODR	0	R/W	
4	PF4ODR	0	R/W	
3	PF3ODR	0	R/W	
2	PF2ODR	0	R/W	
1	PF1ODR	0	R/W	
0	PF0ODR	0	R/W	

10.14.5 Pin Functions

Port F pins also function as the pins for SSU I/Os, A/D converter inputs, interrupt inputs, bus control signal I/Os, and system clock outputs. The correspondence between the register specification and the pin functions is shown below.

- PF7/ϕ

  The pin function is switched as shown below according to bit PF7DDR.

Operating mode	1, 2, 4, 7	
PF7DDR	0	1
Pin function	PF7 input	ϕ output

- PF6/$\overline{AS}$/$\overline{AH}$

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit MPXE in MPXCR of the bus controller, bit ASOE in PFCR2, and bit PF6DDR.

Operating mode	1, 2, 4			7				
EXPE	—			0		1		
ASOE	1	0		—		1	0	
PF6DDR	—	0	1	0	1	—	0	1
Pin function	$\overline{AS}$/$\overline{AH}$* output	PF6 input	PF6 output	PF6 input	PF6 output	$\overline{AS}$/$\overline{AH}$* output	PF6 input	PF6 output

Note: * $\overline{AH}$ output when MPXE = 1, and $\overline{AS}$ output when MPXE = 0.

- PF5/$\overline{RD}$

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, and bit PF5DDR.

Operating mode	1, 2, 4	7		
EXPE	—	0		1
PF5DDR	—	0	1	—
Pin function	$\overline{RD}$ output	PF5 input	PF5 output	$\overline{RD}$ output

- PF4/$\overline{HWR}$

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, and bit PF4DDR.

Operating mode	1, 2, 4	7		
EXPE	—	0		1
PF4DDR	—	0	1	—
Pin function	$\overline{HWR}$ output	PF4 input	PF4 output	$\overline{HWR}$ output

- PF3/$\overline{\text{LWR}}$/SSO0-C

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits MSS and BIDE in SSCRH, bit SSUMS in SSCRL, and bits TE and RE in SSER of SSU, bit LWROE in PFCR2, bits SSOS1 and SSOS0 in PFCR5, and bit PF3DDR.

Operating mode	1, 2, 4, 7 (EXPE = 1)					7 (EXPE = 0)			
LWROE	1	0				0			
SSU settings	—	Can be used as I/O port		Input state	Output state	Can be used as I/O port		Input state	Output state
PF3DDR	—	0	1	0	—	0	1	0	—
Pin function	$\overline{\text{LWR}}$ output	PF3 input	PF3 output	SSO0-C input*1	SSO0-C output*2	PF3 input	PF3 output	SSO0-C input*1	SSO0-C output*2

Notes: 1. SSO0-C input when SSO0S1 and SSO0S0 = B'10 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'000×1 or B'01×01.

2. SSO0-C output when SSO0S1 and SSO0S0 = B'10 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'0011×, B'01×10, or B'10×1×.

	SSO pin settings																	
SSUMS	0						0				1							
BIDE	0						1				0							
MSS	0			1			0		1		0			1				
TE	0	1		0	1		0	1	0	1	0	1		0	1			
RE	1	0	1	1	0	1	1	0	1	0	1	0	1	1	0	1		
Pin state	Input	—	Input	—	Output	Output	Input	Output	Input	Output	—	Output	Output	—	Output	Output		

Legend:

—: Pin is not used by the SSU (can be used as I/O port)

- PF2/$\overline{\text{LCAS}}$/DQML/$\overline{\text{IRQ15-A}}$/SSI0-C (H8S/2456 Group and H8S/2456R Group)

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits MSS and BIDE in SSCRH, bit SSUMS in SSCRL, and bits TE and RE in SSER of SSU, bits RMTS2 to RMTS0 in DRAMCR of the bus controller, bits ABW5 to ABW2 in ABWCR, bits SSI0S1 and SSI0S0 in PFCR5, and bit PF2DDR.

- Modes 1, 2, 4, and 7 (EXPE = 1)

Areas 2 to 5	Any DRAM/ synchronous DRAM space area is 16-bit bus space	All DRAM/synchronous DRAM space areas are 8-bit bus space, or areas 2 to 5 are all normal space			
SSU settings	—	Can be used as I/O port		Input state	Output state
PF2DDR	—	0	1	0	—
Pin function	$\overline{\text{LCAS}}$ output DQML*[3] output	PF2 input	PF2 output	SSI0-C input*[2]	SSI0-C output*[3]
	$\overline{\text{IRQ15-A}}$ interrupt input*[1]				

- Mode 7 (EXPE = 0)

Areas 2 to 5	—			
SSU settings	Can be used as I/O port		Input state	Output state
PF2DDR	0	1	0	—
Pin function	PF2 input	PF2 output	SSI0-C input*[2]	SSI0-C output*[3]
	$\overline{\text{IRQ15-A}}$ interrupt input*[1]			

Notes: 1. $\overline{\text{IRQ15}}$ input when the ITS15 bit in ITSR is 0.

2. SSI0-C input when SSI0S1 and SSI0S0 = B'10 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'001×1 or B'10××1.

3. SSI0-C output when SSI0S1 and SSI0S0 = B'10 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'0001×.

	SSI pin settings															
SSUMS	0						0				1					
BIDE	0						1				0					
MSS	0			1			0		1		0			1		
TE	0	1		0	1		0	1	0	1	0	1		0	1	
RE	1	0	1	1	0	1	1	0	1	0	1	0	1	1	0	1
Pin state	—	Output	Output	Input	—	Input	—	—	—	—	Input	—	Input	Input	—	Input

Legend:

—: Pin is not used by the SSU (can be used as I/O port)

- PF2/$\overline{\text{CS6}}$/$\overline{\text{LCAS}}$/SSI0-C (H8S/2454 Group)

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits MSS and BIDE in SSCRH, bit SSUMS in SSCRL, and bits TE and RE in SSER of SSU, bits RMTS2 to RMTS0 in DRAMCR of the bus controller, bit CS6E in PFCR0, bits SSI0S1 and SSI0S0 in PFCR5, bits ABW5 to ABW2 in ABWCR, and bit PF2DDR.

- Modes 1, 2, 4, and 7 (EXPE = 1)

Areas 2 to 5	Any DRAM/ synchronous DRAM space area is 16-bit bus space	All DRAM/synchronous DRAM space areas are 8-bit bus space, or areas 2 to 5 are all normal space					
CS6E	—	0				1	
SSU settings	—	Can be used as I/O port		Input state	Output state	—	
PF2DDR	—	0	1	0	—	0	1
Pin function	$\overline{\text{LCAS}}$ output	PF2 input	PF2 output	SSI0-C input*[1]	SSI0-C output*[2]	PF2 input	$\overline{\text{CS6}}$ output

- Mode 7 (EXPE = 0)

Areas 2 to 5	—			
CS6E	—			
SSU settings	Can be used as I/O port		Input state	Output state
PF2DDR	0	1	0	—
Pin function	PF2 input	PF2 output	SSI0-C input*[1]	SSI0-C output*[2]

Notes: 1. SSI0-C input when SSI0S1 and SSI0S0 = B'10 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'001×1 or B'10××1.
2. SSI0-C output when SSI0S1 and SSI0S0 = B'10 in PFCR5, and SSUMS, BIDE, MSS, TE, and RE = B'0001×.

	SSI pin settings															
SSUMS	0						0				1					
BIDE	0						1				0					
MSS	0			1			0		1		0			1		
TE	0	1		0	1		0	1	0	1	0	1		0	1	
RE	1	0	1	1	0	1	1	0	1	0	1	0	1	1	0	1
Pin state	—	Output	Output	Input	—	Input	—	—	—	—	Input	—	Input	Input	—	Input

Legend:
—: Pin is not used by the SSU (can be used as I/O port)

- PF1/$\overline{\text{UCAS}}$/DQMU/$\overline{\text{IRQ14-A}}$/SSCK0-C (H8S/2456 Group and H8S/2456R Group)

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits MSS and SCKS in SSCRH and bit SSUMS in SSCRL of SSU, bits RMTS2 to RMTS0 in DRAMCR of the bus controller, bits SSCK0S1 and SSCK0S0 in PFCR5, and bit PF1DDR.

- Modes 1, 2, 4, and 7 (EXPE = 1)

Areas 2 to 5	Any of areas 2 to 5 is DRAM/ synchronous DRAM space	Areas 2 to 5 are all normal space			
SSU settings	—	Can be used as I/O port		Input state	Output state
PF1DDR	—	0	1	0	—
Pin function	$\overline{\text{UCAS}}$ output DQMU*[3] output	PF1 input	PF1 output	SSCK0-C input*[2]	SSCK0-C output*[3]
	$\overline{\text{IRQ14-A}}$ interrupt input*[1]				

- Mode 7 (EXPE = 0)

Areas 2 to 5	—			
SSU settings	Can be used as I/O port		Input state	Output state
PF1DDR	0	1	0	—
Pin function	PF1 input	PF1 output	SSCK0-C input*[2]	SSCK0-C output*[3]
	$\overline{\text{IRQ14-A}}$ interrupt input*[1]			

Notes: 1. $\overline{\text{IRQ14}}$ input when the ITS14 bit in ITSR is 0.
2. SSCK0-C input when SSCK0S1 and SSCK0S0 = B'10 in PFCR5, and SSUMS, MSS, and SCKS = B'001 or B'101.
3. SSCK0-C output when SSCK0S1 and SSCK0S0 = B'10 in PFCR5, and SSUMS, MSS, and SCKS = B'×11.

	SSCK pin settings							
SSUMS	0				1			
MSS	0		1		0		1	
SCKS	0	1	0	1	0	1	0	1
Pin state	—	Input	—	Output	—	Input	—	Output

Legend:
—: Pin is not used by the SSU (can be used as I/O port)

- PF1/$\overline{CS5}$/$\overline{UCAS}$/SSCK0-C (H8S/2454 Group)

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits MSS and SCKS in SSCRH and bit SSUMS in SSCRL of SSU, bits RMTS2 to RMTS0 in DRAMCR of the bus controller, bit CS5E in PFCR0, bits SSCK0S1 and SSCK0S0 in PFCR5, and bit PF1DDR.

- Modes 2, 4, and 7 (EXPE = 1)

Areas 2 to 5	DRAM space	Areas 2 to 5 are all normal space					
CS5E	—	0				1	
SSU settings	—	Can be used as I/O port		Input state	Output state	—	
PF1DDR	—	0	1	0	—	0	1
Pin function	$\overline{UCAS}$ output	PF1 input	PF1 output	SSCK0-C input*[1]	SSCK0-C output*[2]	PF1 input	$\overline{CS5}$ output

- Mode 7 (EXPE = 0)

Areas 2 to 5	—			
CS5E	—			
SSU settings	Can be used as I/O port		Input state	Output state
PF1DDR	0	1	0	—
Pin function	PF1 input	PF1 output	SSCK0-C input*[1]	SSCK0-C output*[2]

Notes: 1. SSCK0-C input when SSCK0S1 and SSCK0S0 = B'10 in PFCR5, and SSUMS, MSS, and SCKS = B'001 or B'101.

2. SSCK0-C output when SSCK0S1 and SSCK0S0 = B'10 in PFCR5, and SSUMS, MSS, and SCKS = B'×11.

	SSCK pin settings							
SSUMS	0				1			
MSS	0		1		0		1	
SCKS	0	1	0	1	0	1	0	1
Pin state	—	Input	—	Output	—	Input	—	Output

Legend:

—: Pin is not used by the SSU (can be used as I/O port)

- PF0/$\overline{\text{WAIT-A}}$/$\overline{\text{ADTRG0-B}}$/$\overline{\text{SCS0-C}}$ (H8S/2456 Group and H8S/2456R Group)

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit WAITE in BCR of the bus controller, bits MSS, CSS1, and CSS0 in SSCRH and bit SSUMS in SSCRL of SSU, bits TRGS1, TRGS0, and EXTRGS in ADCR_0 of ADC, bits ADTRG0S and WAITS in PFCR4, bits SCS0S1 and SCS0S0 in PFCR5, and bit PF0DDR.

- Modes 1, 2, 4, and 7 (EXPE = 1)

WAITE	0				1
SSU settings	Can be used as I/O port		Input state	Output state	—
PF0DDR	0	1	0	—	—
Pin function	PF0 input	PF0 output	$\overline{\text{SCS0-C}}$ input*3	$\overline{\text{SCS0-C}}$ output*4	$\overline{\text{WAIT-A}}$ input*2
	$\overline{\text{ADTRG0-B}}$ input*1				

- Mode 7 (EXPE = 0)

WAITE	—			
SSU settings	Can be used as I/O port		Input state	Output state
PF0DDR	0	1	0	—
Pin function	PF0 input	PF0 output	$\overline{\text{SCS0-C}}$ input*3	$\overline{\text{SCS0-C}}$ output*4
	$\overline{\text{ADTRG0-B}}$ input*1			

Notes: 1. $\overline{\text{ADTRG0-B}}$ input when the ADTRG0S bit in PFCR4 is 1, TRGS1 = TRGS0 = 0, and EXTRGS = 1

2. $\overline{\text{WAIT-A}}$ input when the WAITS bit in PFCR4 is 0.
3. $\overline{\text{SCSO-C}}$ input when SCS0S1 and SCS0S0 = B'10 in PFCR5, and SSUMS, MSS, CSS1, and CSS0 = B'00××, B'0101, or B'0110.
4. $\overline{\text{SCSO-C}}$ output when SCS0S1 and SCS0S0 = B'10 in PFCR5, and SSUMS, MSS, CSS1, and CSS0 = B'011×.

	$\overline{\text{SCS}}$ pin settings					
SSUMS	0					1
MSS	0	1				×
CSS1	×	0		1		×
CSS0	×	0	1	0	1	×
Pin state	Input	—	Input	Automatic I/O	Output	—

Legend:

×: Don't care

—: Pin is not used by the SSU (can be used as I/O port)

- PF0/$\overline{\text{WAIT-A}}$/ADTRG0-B/$\overline{\text{SCS0-C}}$/$\overline{\text{OE-A}}$ (H8S/2454 Group)

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit WAITE in BCR of the bus controller, bit OEE in DRAMCR, bits MSS, CSS1, and CSS0 in SSCRH and bit SSUMS in SSCRL of SSU, bits TRGS1, TRGS0, and EXTRGS in ADCR_0 of ADC, bit OES in PFCR2, bits ADTRG0S and WAITS in PFCR4, bits SCS0S1 and SCS0S0 in PFCR5, and bit PF0DDR.

- Modes 1, 2, 4, and 7 (EXPE = 1)

OEE	0					1					
RMTS2 to RMTS0	—					—					Areas 2 to 5 are DRAM space
WAITE	0				1	0				1	—
SSU settings	Can be used as I/O port		Input state	Output state	—	Can be used as I/O port		Input state	Output state	—	—
PF0DDR	0	1	0	—	—	0	1	0	—	—	—
Pin function	PF0 input	PF0 output	$\overline{\text{SCS0-C}}$ input*2	$\overline{\text{SCS0-C}}$ output*4	$\overline{\text{WAIT-A}}$ input*3	PF0 input	PF0 output	$\overline{\text{SCS0-C}}$ input*2	$\overline{\text{SCS0-C}}$ output*4	$\overline{\text{WAIT-A}}$ input*3	$\overline{\text{OE-A}}$ output
	ADTRG0-B input*1										

- Mode 7 (EXPE = 0)

OEE	—			
Area 2	—			
WAITE	—			
SSU settings	—		Input state	Output state
PF0DDR	0	1	0	—
Pin function	PF0 input	PF0 output	$\overline{\text{SCS0-C}}$ input*2	$\overline{\text{SCS0-C}}$ output*4
	$\overline{\text{ADTRG0-B}}$ input*1			

Notes: 1. $\overline{\text{ADTRG0-B}}$ input when the ADTRG0S bit in PFCR4 is 1, TRGS1 = TRGS0 = 0, and EXTRGS = 1
2. $\overline{\text{WAIT-A}}$ input when the WAITS bit in PFCR4 is 0.
3. $\overline{\text{SCSO-C}}$ input when SCS0S1 and SCS0S0 = B'10 in PFCR5, and SSUMS, MSS, CSS1, and CSS0 = B'00××, B'0101, or B'0110.
4. $\overline{\text{SCSO-C}}$ output when SCS0S1 and SCS0S0 = B'10 in PFCR5, and SSUMS, MSS, CSS1, and CSS0 = B'011×.

	$\overline{\text{SCS}}$ pin settings					
SSUMS	0					1
MSS	0	1				×
CSS1	×	0		1		×
CSS0	×	0	1	0	1	×
Pin state	Input	—	Input	Automatic I/O	Output	—

Legend:

×: Don’t care

—: Pin is not used by the SSU (can be used as I/O port)

10.15 Port G

Port G is a 7-bit I/O port that also has other functions. Port G has the following registers.

- Port G data direction register (PGDDR)
- Port G data register (PGDR)
- Port G register (PORTG)
- Port function control register 0 (PFCR0)
- Port function control register 4 (PFCR4)
- Port G open drain control register (PGODR)

10.15.1 Port G Data Direction Register (PGDDR)

The individual bits of PGDDR specify input or output for the pins of port G. PGDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	PG6DDR	0	W	• Modes 7 (when EXPE = 1), 1, 2, and 4 Pins PG6 to PG4 function as bus control input/output pins ($\overline{\text{BREQO}}$, $\overline{\text{BACK}}$, and $\overline{\text{BREQ}}$) when the appropriate bus controller settings are made. Otherwise, these pins are I/O ports, and their functions can be switched with PGDDR. When the $\overline{\text{CS}}$ output enable bits (CS3E to CS0E) are set to 1, pins PG3 to PG0 function as $\overline{\text{CS}}$ output pins when the corresponding PGDDR bit is set to 1, and as input ports when the bit is cleared to 0. When the $\overline{\text{CS}}$ output enable bits (CS3E to CS0E) are cleared to 0, pins PG3 to PG0 are I/O ports, and their functions can be switched with PGDDR. • Mode 7 (when EXPE = 0) Pins PG6 to PG0 are I/O ports, and their functions can be switched with PGDDR.
5	PG5DDR	0	W	
4	PG4DDR	0	W	
3	PG3DDR	0	W	
2	PG2DDR	0	W	
1	PG1DDR	0	W	
0	PG0DDR	1/0*	W	

Note: * PG0DDR is initialized to 1 in modes 1 and 2, and to 0 in modes 4 and 7.

10.15.2 Port G Data Register (PGDR)

PGDR stores output data for the port G pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0, and cannot be modified.
6	PG6DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
5	PG5DR	0	R/W	
4	PG4DR	0	R/W	
3	PG3DR	0	R/W	
2	PG2DR	0	R/W	
1	PG1DR	0	R/W	
0	PG0DR	0	R/W	

10.15.3 Port G Register (PORTG)

PORTG shows the pin states of port G. PORTG cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved If this bit is read, it will return an undefined value.
6	PG6	—*	R	If this register is read while a PGDDR bit is set to 1, the corresponding PGDR value is read. If this register is read while a PGDDR bit is cleared to 0, the corresponding pin state is read.
5	PG5	—*	R	
4	PG4	—*	R	
3	PG3	—*	R	
2	PG2	—*	R	
1	PG1	—*	R	
0	PG0	—*	R	

Note: * Determined by the states of pins PG6 to PG0.

10.15.4 Port G Open Drain Control Register (PGODR)

PGODR specifies the output type of each port G pin.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0. Only the initial value should be written to this bit.
6	PG6ODR	0	R/W	When not specified for $\overline{\text{BACK-A}}$, $\overline{\text{BREQO-A}}$, $\overline{\text{CS0}}$, $\overline{\text{CS1}}$, $\overline{\text{CS2}}$, $\overline{\text{CS3}}$, $\overline{\text{CS4}}$, $\overline{\text{RAS2}}$, $\overline{\text{RAS3}}$, $\overline{\text{RAS}}$, or $\overline{\text{CAS}}$ output, setting a PGODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a PGODR bit to 0 makes the corresponding pin a CMOS output pin.
5	PG5ODR	0	R/W	
4	PG4ODR	0	R/W	
3	PG3ODR	0	R/W	
2	PG2ODR	0	R/W	
1	PG1ODR	0	R/W	
0	PG0ODR	0	R/W	

10.15.5 Pin Functions

Port G pins also function as the pins for JTAG inputs and bus control signal I/Os. The correspondence between the register specification and the pin functions is shown below.

- PG6/$\overline{\text{BREQ-A}}$/TDI*[1]

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit BRLE in BCR of the bus controller, bit BREQS in PFCR4, and bit PG6DDR.

Operating mode	1, 2, 4			7				
EXPE	—			0		1		
BRLE BREQS	BRLE = 0 or BRLE = 1 and BREQS = 1		BRLE = 1 and BREQS = 0	—		BRLE = 0 or BRLE = 1 and BREQS = 1		BRLE = 1 and BREQS = 0
PG6DDR	0	1	—	0	1	0	1	—
Pin function	PG6 input	PG6 output	$\overline{\text{BREQ-A}}$ input	PG6 input	PG6 output	PG6 input	PG6 output	$\overline{\text{BREQ-A}}$ input
	TDI input*[2]							

Notes: 1. Supported only in the 145-pin package.
2. TDI input when BSCANE pin = 1 in the 145-pin package.

- PG5/$\overline{\text{BACK-A}}$/TMS*[1]

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit BRLE in BCR of the bus controller, bit BACKS in PFCR4, and bit PG5DDR.

Operating mode	1, 2, 4			7				
EXPE	—			0		1		
BRLE BACKS	BRLE = 0 or BRLE = 1 and BACKS = 1		BRLE = 1 and BACKS = 0	—		BRLE = 0 or BRLE = 1 and BACKS = 1		BRLE = 1 and BACKS = 0
PG5DDR	0	1	—	0	1	0	1	—
Pin function	PG5 input	PG5 output	$\overline{\text{BACK-A}}$ output	PG5 input	PG5 output	PG5 input	PG5 output	$\overline{\text{BACK-A}}$ output
	TMS input*[2]							

Notes: 1. Supported only in the 145-pin package.
2. TMS input when BSCANE pin = 1 in the 145-pin package.

- PG4/$\overline{\text{BREQO-A}}$/$\overline{\text{CS4}}$*[1]/TCK*[2]

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit BRLE in BCR of the bus controller, bit BREQOE, bit BREQOS in PFCR4, and bit PG4DDR.

- Modes 1, 2, 4, and 7 (EXPE = 1)

BRLE	0			1			
BREQOE BREQOS	—			BREQOE = 0 or BREQOE = 1 and BREQOS = 1			BREQOR = 1 and BREQOS = 0
CS4E	0		1	0		1	—
PG4DDR	0	1	—	0	1	—	—
Pin function	PG4 input	PG4 output	$\overline{\text{CS4}}$ output*[1]	PG4 input	PG4 output	$\overline{\text{CS4}}$ output*[1]	$\overline{\text{BREQO-A}}$ output
	TCK input*[3]						

- Mode 7 (EXPE = 0)

BRLE	—		
BREQOE BREQOS	—		
CS4E	0		1
PG4DDR	0	1	—
Pin function	PG4 input	PG4 output	$\overline{CS4}$ output*[1]
	TCK input*[3]		

Notes: 1. Not supported in the H8S/2456 Group and H8S/2456R Group.
2. Supported only in the 145-pin package.
3. TCK input when BSCANE pin = 1 in the 145-pin package.

- PG3/$\overline{CS3}$/$\overline{RAS3}$/$\overline{CAS}$*

  The pin function is switched as shown below according to the combination of the operating mode, bits RMTS2 to RMTS0 in DRAMCR of the bus controller, bit CS3E in PFCR0, and bit PG3DDR.

Operating mode	1, 2, 4						7							
EXPE	—						0		1					
CS3E	0		1				—		0		1			
RMTS2 to RMTS0	—		Area 3 is in normal space		Area 3 is in DRAM space	Areas 2 to 5 are in synchronous DRAM* space	—		—		Area 3 is in normal space		Area 3 is in DRAM space	Areas 2 to 5 are in synchronous DRAM* space
PG3DDR	0	1	0	1	—	—	0	1	0	1	0	1	—	—
Pin function	PG3 input	PG3 output	PG3 input	$\overline{CS3}$ output	$\overline{RAS3}$ output	$\overline{CAS}$* output	PG3 input	PG3 output	PG3 input	PG3 output	PG3 input	$\overline{CS3}$ output	$\overline{RAS3}$ output	$\overline{CAS}$* output

Note: * Not supported in the H8S/2456 Group and H8S/2454 Group.

- PG2/$\overline{CS2}$/$\overline{RAS2}$/$\overline{RAS}$*

  The pin function is switched as shown below according to the combination of the operating mode, bits RMTS2 to RMTS0 in DRAMCR of the bus controller, bit CS2E in PFCR0, and bit PG2DDR.

Operating mode	1, 2, 4						7							
EXPE	—						0		1					
CS2E	0		1				—		0		1			
RMTS2 to RMTS0	—		Area 2 is in normal space		Area 2 is in DRAM space	Areas 2 to 5 are in synchronous DRAM* space	—		—		Area 2 is in normal space		Area 2 is in DRAM space	Areas 2 to 5 are in synchronous DRAM* space
PG2DDR	0	1	0	1	—	—	0	1	0	1	0	1	—	—
Pin function	PG2 input	PG2 output	PG2 input	$\overline{CS2}$ output	$\overline{RAS2}$ output	$\overline{RAS}$* output	PG2 input	PG2 output	PG2 input	PG2 output	PG2 input	$\overline{CS2}$ output	$\overline{RAS2}$ output	$\overline{RAS}$* output

Note: * Not supported in the H8S/2456 Group and H8S/2454 Group.

- PG1/$\overline{CS1}$, PG0/$\overline{CS0}$

  The pin function is switched as shown below according to the combination of the operating mode, bit CSnE in PFCR0, and bit PGnDDR.

Operating mode	1, 2, 4				7					
EXPE	—				0		1			
CSnE	0		1		—		0		1	
PGnDDR	0	1	0	1	0	1	0	1	0	1
Pin function	PGn input	PGn output	PGn input	$\overline{CSn}$ output	PGn input	PGn output	PGn input	PGn output	PGn input	$\overline{CSn}$ output

Legend:
n = 1 or 0

10.16 Port H

Note: Port H is not supported in the H8S/2454 Group.

Port H is a 4-bit I/O port that also has other functions. Port H has the following registers. For the port function control registers, refer to section 10.18, Port Function Control Registers.

- Port H data direction register (PHDDR)
- Port H data register (PHDR)
- Port H register (PORTH)
- Port function control register 0 (PFCR0)
- Port function control register 2 (PFCR2)
- Port H open drain control register (PHODR)

10.16.1 Port H Data Direction Register (PHDDR)

The individual bits of PHDDR specify input or output for the pins of port H. PHDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved
3	PH3DDR	0	W	• Modes 7 (when EXPE = 1), 1, 2, and 4 When the $\overline{OE}$ output enable bit (OEE) and $\overline{OE}$ output select bit (OES) are set to 1, pin PH3 functions as the $\overline{OE}$ output pin. Otherwise, when bit CS7E is set to 1, pin PH3 functions as the $\overline{CS7}$ output pin when bit PH3DDR is set to 1, and as an input port when the bit is cleared to 0. When bit CS7E is cleared to 0, pin PH3 is an I/O port, and its function can be switched with bit PH3DDR. When areas 2 to 5 are specified as continuous SDRAM space*, $\overline{OE}$ output is CKE output. When bit CS6E is set to 1, setting bit PH2DDR to 1 makes pin PH2 function as the $\overline{CS6}$ output pin, and clearing the bit to 0 makes the pin function as an I/O port. When bit CS6E is cleared to 0, pin PH2 is an I/O port, and its function can be switched with bit PH2DDR.
2	PH2DDR	0	W	
1	PH1DDR	0	W	
0	PH0DDR	0	W	

Bit	Bit Name	Initial Value	R/W	Description
0	PH0DDR	0	W	Pin PH1 functions as the SDRAMϕ* output pin when the SDPSTP bit is 0 in a product supporting the SDRAM interface. In a product not supporting the SDRAM interface or when the SDPSTP bit is 1, if bit CS5E is set to 1 while area 5 is specified as normal space, pin PH1 functions as the $\overline{CS5}$ output pin when bit PH1DDR is set to 1, and functions as an I/O port when the bit is cleared to 0. When bit CS5E is cleared to 0, pin PH1 is an I/O port, and its function can be switched with bit PH1DDR. When area 5 is specified as DRAM space and bit CS5E is set to 1, pin PH1 functions as the $\overline{RAS5}$ output pin and as an I/O port when the bit is cleared to 0. Pin PH0 functions as the $\overline{CS4}$ output pin when area 4 is specified as normal space and bit PH0DDR is set to 1. If bit PH0DDR is cleared to 0, pin PH0 functions as an I/O port. When bit CS4E is cleared to 0, pin PH0 is an I/O port, and its function can be switched with bit PH0DDR. When area 4 is specified as DRAM space and bit CS4E is set to 1, pin PH0 functions as the $\overline{RAS4}$ output pin and as an I/O port when the bit is cleared to 0. When areas 2 to 5 are specified as continuous SDRAM space*, pin PH0 functions as the $\overline{WE}$ output pin when bit CS4E is set to 1, and as an I/O port when the bit is cleared to 0. • Mode 7 (when EXPE = 0) Pins PH3 to PH0 are I/O ports, and their functions can be switched with PHDDR. Pin PH1 functions as the SDRAMϕ output pin when the SDPSTP bit is 0 in a product supporting the SDRAM interface. In a product not supporting the SDRAM interface or when the SDPSTP bit is 1, pin PH1 is an I/O port and its function can be switched with PHDDR.

Note: * Not supported in the H8S/2456 Group.

10.16.2 Port H Data Register (PHDR)

PHDR stores output data for the port H pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	PH3DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
2	PH2DR	0	R/W	
1	PH1DR	0	R/W	
0	PH0DR	0	R/W	

10.16.3 Port H Register (PORTH)

PORTH shows the pin states of port H. PORTH cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved If these bits are read, they will return an undefined value.
3	PH3	—*	R	If this register is read while a PHDDR bit is set to 1, the corresponding PHDR value is read. If this register is read while a PHDDR bit is cleared to 0, the corresponding pin state is read.
2	PH2	—*	R	
1	PH1	—*	R	
0	PH0	—*	R	

Note: * Determined by the states of pins PH3 to PH0.

10.16.4 Port H Open Drain Control Register (PHODR)

PHODR specifies the output type of each port H pin.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0. Only the initial values should be written to these bits.
3	PH3ODR	0	R/W	When not specified for $\overline{CS4}$, $\overline{CS5}$, $\overline{CS6}$, $\overline{CS7}$, $\overline{OE\text{-}A}$, CKE-A, $\overline{RAS4}$, $\overline{RAS5}$, $\overline{WE}$, or SDRAMϕ* output, setting a PHODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a PHODR bit to 0 makes the corresponding pin a CMOS output pin.
2	PH2ODR	0	R/W	
1	PH1ODR	0	R/W	
0	PH0ODR	0	R/W	

Note: * Not supported in the H8S/2456 Group.

10.16.5 Pin Functions

Port H pins also function as bus control signal I/Os and interrupt inputs. The correspondence between the register specification and the pin functions is shown below.

- PH3/$\overline{\text{CS7}}$/$\overline{\text{OE-A}}$/CKE-A/$\overline{\text{IRQ7-B}}$

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bit OEE of the bus controller, bit OES in PFCR2, bit CS7E in PFCR0, and bit PH3DDR.

- Modes 1, 2, 4, and 7 (EXPE = 1)

OEE	0				1					
OES	0				0				1	
RMTS2 to RMTS0	—				—				Areas 2 to 5 are DRAM space	Areas 2 to 5 are synchronous DRAM*[3] space
CS7E	0		1		0		1		—	—
PH3DDR	0	1	0	1	0	1	0	1	—	—
Pin function	PH3 input	PH3 output	PH3 input	$\overline{\text{CS7}}$ output	PH3 input	PH3 output	PH3 input	$\overline{\text{CS7}}$ output	$\overline{\text{OE-A}}$ output*[2]	CKE-A*[3] output*[2]
	$\overline{\text{IRQ7-B}}$ input*[1]									

- Mode 7 (EXPE = 0)

OEE	—	
OES	—	
RMTS2 to RMTS0	—	
CS7E	—	
PH3DDR	0	1
Pin function	PH3 input	PH3 output
	$\overline{\text{IRQ7-B}}$ input*[1]	

Notes: 1. $\overline{\text{IRQ7-B}}$ input when the ITS7 bit in ITSR is 1.
2. $\overline{\text{OE-A}}$/CKE-A output when the OES bit in PFCR2 is 1.
3. Not supported in the H8S/2456 Group.

- PH2/$\overline{\text{CS6}}$/$\overline{\text{IRQ6-B}}$

  The pin function is switched as shown below according to the combination of the operating mode, bit CS6E in PFCR0, and bit PH2DDR.

Operating mode	1, 2, 4				7					
EXPE	—				0		1			
CS6E	0		1		—		0		1	
PH2DDR	0	1	0	1	0	1	0	1	0	1
Pin function	PH2 input	PH2 output	PH2 input	$\overline{\text{CS6}}$ output	PH2 input	PH2 output	PH2 input	PH2 output	PH2 input	$\overline{\text{CS6}}$ output
	$\overline{\text{IRQ6-B}}$ interrupt input*									

Note: * $\overline{\text{IRQ6-B}}$ input when the ITS6 bit in ITSR is 1.

- PH1/$\overline{CS5}$/$\overline{RAS5}$/SDRAMϕ

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits RMTS2 to RMTS0 in DRAMCR of the bus controller, bit SDPSTP in SCKCR of the clock pulse generator, bit CS5E in PFCR0 and bit PH1DDR.

SDPSTP	1																0
Operating mode	1, 2, 4							7									—
EXPE	—							0		1							—
RMTS2 to RMTS0	Area 5 is normal space				Area 5 is DRAM space			—		Area 5 is normal space				Area 5 is DRAM space			—
CS5E	0		1		0		1	—		0		1		0		1	—
PH1DDR	0	1	0	1	0	1	—	0	1	0	1	0	1	0	1	—	—
Pin function	PH1 input	PH1 output	PH1 input	$\overline{CS5}$ output	PH1 input	PH1 output	$\overline{RAS5}$ output	PH1 input	PH1 output	PH1 input	PH1 output	PH1 input	$\overline{CS5}$ output	PH1 input	PH1 output	$\overline{RAS5}$ output	SDRAMϕ output*

Notes: * Not supported in the H8S/2456 Group.

- PH0/$\overline{CS4}$/$\overline{RAS4}$/$\overline{WE}$*

  The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits RMTS2 to RMTS0 in DRAMCR of the bus controller, bit CS4E in PFCR0, and bit PH0DDR.

Operating mode	1, 2, 4						7							
EXPE	—						0		1					
CS4E	0		1				—		0		1			
RMTS2 to RMTS0	—		Area 4 is normal space		Area 4 is DRAM space	Areas 2 to 5 are syn-chronous DRAM* space	—		—		Area 4 is normal space		Area 4 is DRAM space	Areas 2 to 5 are syn-chronous DRAM* space
PH0DDR	0	1	0	1	—	—	0	1	0	1	0	1	—	—
Pin function	PH0 input	PH0 output	PH0 input	$\overline{CS4}$ output	$\overline{RAS4}$ output	$\overline{WE}$* output	PH0 input	PH0 output	PH0 input	PH0 output	PH0 input	$\overline{CS4}$ output	$\overline{RAS4}$ output	$\overline{WE}$* output

Note: * Not supported in the H8S/2456 Group.

10.17 Port J

Note: Port J is not supported in the H8S/2454 Group and in the 145-pin package.

Port J is a 3-bit I/O port. Port J has the following registers.

- Port J data direction register (PJDDR)
- Port J data register (PJDR)
- Port J register (PORT3)
- Port J open drain control register (PJODR)

10.17.1 Port J Data Direction Register (PJDDR)

The individual bits of PJDDR specify input or output for the pins of port J. PJDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	—	Reserved
1	PJ1DDR	0	W	When a pin function is specified as a general purpose I/O, setting this bit to 1 makes the corresponding pin an output port, while clearing this bit to 0 makes the corresponding pin an input port.
0	PJ0DDR	0	W	

10.17.2 Port J Data Register (PJDR)

PJDR stores output data for the port J pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
1	PJ1DR	0	R/W	Output data for a pin is stored when the pin function is specified as a general purpose I/O.
0	PJ0DR	0	R/W	

10.17.3 Port J Register (PORTJ)

PORTJ shows the pin states of port J. PORTJ cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	—	Reserved If these bits are read, they will return an undefined value.
2	PJ2	—*	R	The pin state is always read from this register. Bit 2 is reserved for the 145-pin version.
1	PJ1	—*	R	If this register is read, the PJDR values are read for the bits with the corresponding PJDDR bits set to 1. For the bits with the corresponding PJDDR bits cleared to 0, the pin states are read.
0	PJ0	—*	R	

Note: * Determined by the state of pins PJ0 to PJ2.

10.17.4 Port J Open Drain Control Register (PJODR)

PJODR specifies the output type of each port J pin.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	—	Reserved These bits are always read as 0. Only the initial values should be written to these bits.
1	PJ1ODR	0	R/W	Setting a PJODR bit to 1 makes the corresponding pin an NMOS open-drain output pin, while clearing a PJODR bit to 0 makes the corresponding pin a CMOS output pin.
0	PJ0ODR	0	R/W	

10.17.5 Pin Functions

Port J pins function only as I/O ports. The correspondence between the register specification and the pin functions is shown below.

- PJ2*

  The PJ2 pin is an input-only pin.

Pin function	PJ2 input

Note: * Not supported in the 145-pin package.

- PJ1, PJ0

  The pin function is switched as shown below according to bit PJnDDR.

PJnDDR	0	1
Pin function	PJn input	PJn output

Legend:
n = 1 or 0

10.18 Port Function Control Registers

The port function controller performs I/O port control. The setting of input or output for each pin should be enabled only after the input or output destination has been selected.

The port function controller has the following registers.

- Port function control register 0 (PFCR0)
- Port function control register 1 (PFCR1)
- Port function control register 2 (PFCR2)
- Port function control register 3 (PFCR3)
- Port function control register 4 (PFCR4)
- Port function control register 5 (PFCR5)

10.18.1 Port Function Control Register 0 (PFCR0)

PFCR0 switches the functions of the chip select output pins.

Bit	Bit Name	Initial Value	R/W	Description
7	CS7E	1	R/W	CS7 to CS0 Enable These bits enable or disable the corresponding $\overline{\text{CSn}}$ output. 0: Pin is designated as I/O port 1: Pin is designated as $\overline{\text{CSn}}$ output pin (n = 7 to 0)
6	CS6E	1	R/W	
5	CS5E	1	R/W	
4	CS4E	1	R/W	
3	CS3E	1	R/W	
2	CS2E	1	R/W	
1	CS1E	1	R/W	
0	CS0E	1	R/W	

10.18.2 Port Function Control Register 1 (PFCR1)

PFCR1 enables or disables address output (A23 to A16).

Bits 7 to 5 are valid in modes 1 and 2 and all the bits are valid in modes 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	A23E	1	R/W	Address 23 Enable Enables or disables output for address output 23 (A23). 0: DR output when PA7DDR = 1 1: A23 output when PA7DDR = 1
6	A22E	1	R/W	Address 22 Enable Enables or disables output for address output 22 (A22). 0: DR output when PA6DDR = 1 1: A22 output when PA6DDR = 1
5	A21E	1	R/W	Address 21 Enable Enables or disables output for address output 21 (A21). 0: DR output when PA5DDR = 1 1: A21 output when PA5DDR = 1
4	A20E	1	R/W	Address 20 Enable Enables or disables output for address output 20 (A20). 0: DR output when PA4DDR = 1 1: A20 output when PA4DDR = 1
3	A19E	1	R/W	Address 19 Enable Enables or disables output for address output 19 (A19). 0: DR output when PA3DDR = 1 1: A19 output when PA3DDR = 1
2	A18E	1	R/W	Address 18 Enable Enables or disables output for address output 18 (A18). 0: DR output when PA2DDR = 1 1: A18 output when PA2DDR = 1

Bit	Bit Name	Initial Value	R/W	Description
1	A17E	1	R/W	Address 17 Enable Enables or disables output for address output 17 (A17). 0: DR output when PA1DDR = 1 1: A17 output when PA1DDR = 1
0	A16E	1	R/W	Address 16 Enable Enables or disables output for address output 16 (A16). 0: DR output when PA0DDR = 1 1: A16 output when PA0DDR = 1

10.18.3 Port Function Control Register 2 (PFCR2)

PFCR2 enables or disables $\overline{AS}$ output, $\overline{LWR}$ output, and $\overline{OE}$ output.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	ASOE	1	R/W	$\overline{AS}$ Output Enable Enables or disables the $\overline{AS}$ output pin. 0: PF6 is designated as I/O port 1: PF6 is designated as $\overline{AS}$ output pin
2	LWROE	1	R/W	$\overline{LWR}$ Output Enable Enables or disables the $\overline{LWR}$ output pin. 0: PF3 is designated as I/O port 1: PF3 is designated as $\overline{LWR}$ output pin
1	OES	1	R/W	$\overline{OE}$ Output Select Selects the $\overline{OE}$/CKE*[1] output pin port when the OEE bit in DRAMCR is set to 1 (enabling $\overline{OE}$/CKE*[1] output). 0: P35 is designated as $\overline{OE\text{-}B}$/CKE-B*[1] output pin. 1: PH3*[2] is designated as $\overline{OE\text{-}A}$/CKE-A*[1] output pin.

Bit	Bit Name	Initial Value	R/W	Description
0	—	0	—	Reserved This bit is always read as 0. Only the initial value should be written to this bit.

Notes: 1. Not supported in the H8S/2454 Group.
2. PH3 becomes PF0 in the H8S/2454 Group.

10.18.4 Port Function Control Register 3 (PFCR3)

PFCR3 enables or disables DMAC activation interrupts from the USB, and switches the functions of the PPG output pin, TPU input/output pin, and TMR input/output pin.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1. Only the initial value should be written to this bit.
6	PPGS	0	R/W	PPG Pin Select Selects the output pins of PO5 to PO0. 0: P25/PO5-A and P20/PO0-A are selected. 1: P85/PO5-B, P52/PO4-B, P83/PO3-B, P51/PO2-B, P81/PO1-B, and P50/PO0-B are selected.
5	TPUS	0	R/W	TPU Pin Select Selects the output pins of TIOCA3, TIOCB3, TIOCC3, TIOCD3, TIOCA4, and TIOCB4. 0: P25/TIOCB4-A and P20/TIOCA3-A are selected. 1: P85/TIOCB4-B, P52/TIOCA4-B, P83/TIOCD3-B, P51/TIOCC3-B, P81/TIOCB3-B, and P50/TIOCA3-B are selected.

Bit	Bit Name	Initial Value	R/W	Description
4	TMRS	0	R/W	TMR Pin Select Selects the output pins of TMO1 and TMO0 and input pins of TMCI1, TMCI0, TMRI1, and TMRI0. 0: [For H8S/2454] P25/TMO1-A, P24/TMO0-A, P23/TMCI1-A, P22/TMCI0-A, P21/TMRI1-A, and P20/TMRI0-A are selected. [For H8S/2456, H8S/2456R] P65/TMO1-A, P64/TMO0-A, P63/TMCI1-A, P62/TMCI0-A, P61/TMRI1-A, and P60/TMRI0-A are selected. 1: P85/TMO1-B, P52/TMO0-B, P83/TMCI1-B, P51/TMCI0-B, P81/TMRI1-B, and P50/TMRI0-B are selected.
3, 2	—	All 0	—	Reserved These bits are always read as 0. Only the initial values should be written to these bits.
1	USBDRQE	0	R/W	USB-DMAC Activation Interrupt Enable Enables or disables interrupt to activate the DMAC (USBINTN) from the USB. 0: The $\overline{\text{DREQ}}$ signal from the $\overline{\text{DREQ}}$ pin is set as a data transfer activation source. 1: The DMAC activation interrupt signal from the USB is set as a data transfer activation source.
0	—	1	—	Reserved This bit is always read as 1. Only the initial value should be written to this bit.

10.18.5 Port Function Control Register 4 (PFCR4)

PFCR4 switches the functions of the $\overline{\text{WAIT}}$ input pin, $\overline{\text{BREQ}}$ input pin, $\overline{\text{BACK}}$ output pin, $\overline{\text{BREQO}}$ output pin, TxD4 output pin, RxD4 input pin, and SCK4 input/output pin.

Bit	Bit Name	Initial Value	R/W	Description
7	WAITS	0	R/W	$\overline{\text{WAIT}}$ Pin Select Selects the $\overline{\text{WAIT}}$ input pin. 0: PF0/$\overline{\text{WAIT-A}}$ is selected 1: P25/$\overline{\text{WAIT-B}}$ is selected
6	BREQS	0	R/W	$\overline{\text{BREQ}}$ Pin Select Selects the $\overline{\text{BREQ}}$ input pin. 0: PG6/$\overline{\text{BREQ-A}}$ is selected 1: P51/$\overline{\text{BREQ-B}}$ is selected
5	BACKS	0	R/W	$\overline{\text{BACK}}$ Pin Select Selects the $\overline{\text{BACK}}$ output pin. 0: PG5/$\overline{\text{BACK-A}}$ is selected 1: P52/$\overline{\text{BACK-B}}$ is selected
4	BREQOS	0	R/W	$\overline{\text{BREQO}}$ Pin Select Selects the $\overline{\text{BREQO}}$ output pin. 0: PG4/$\overline{\text{BREQO-A}}$ is selected 1: P50/$\overline{\text{BREQO-B}}$ is selected
3	—	0	—	Reserved This bit is read as 0. When written, the initial value should be written to.
2	TXD4S	0	R/W	TxD4 Pin Select Selects the TxD4 output pin. 0: P23/TxD4-A is selected 1: PA1/TxD4-B is selected

Bit	Bit Name	Initial Value	R/W	Description
1	RXD4S	0	R/W	RxD4 Pin Select Selects the RxD4 input pin. 0: P24/RxD4-A is selected 1: PA2/RxD4-B is selected
0	SCK4S	0	R/W	SCK4 Pin Select Selects the SCK4 input/output pin. 0: P34/SCK4-A is selected 1: PA3/SCK4-B is selected

10.18.6 Port Function Control Register 5 (PFCR5)

PFCR5 switches the functions of the SSU input/output pins.

Bit	Bit Name	Initial Value	R/W	Description
7	SSO0S1	0	R/W	SSO0 Pin Select
6	SSO0S0	0	R/W	Selects the SSO0 input/output pin. 00: P14/SSO0-A is selected 01: PA7/SSO0-B is selected 10: PF3/SSO0-C is selected 11: Setting prohibited
5	SSI0S1	0	R/W	SSI0 Pin Select
4	SSI0S0	0	R/W	Selects the SSI0 input/output pin. 00: P15/SSI0-A is selected 01: PA6/SSI0-B is selected 10: PF2/SSI0-C is selected 11: Setting prohibited
3	SSCK0S1	0	R/W	SSCK0 Pin Select
2	SSCK0S0	0	R/W	Selects the SSCK0 input/output pin. 00: P16/SSCK0-A is selected 01: PA5/SSCK0-B is selected 10: PF1/SSCK0-C is selected 11: Setting prohibited
1	SCS0S1	0	R/W	$\overline{\text{SCS0}}$ Pin Select
0	SCS0S0	0	R/W	Selects the $\overline{\text{SCS0}}$ input/output pin. 00: P17/$\overline{\text{SCS0-A}}$ is selected 01: PA4/$\overline{\text{SCS0-B}}$ is selected 10: PF0/$\overline{\text{SCS0-C}}$ is selected 11: Setting prohibited

Section 11 16-Bit Timer Pulse Unit (TPU)

This LSI has two on-chip 16-bit timer pulse units (TPU: unit 0 and unit 1) which each comprises six 16-bit timer channels, resulting in a total of 12 channels. The functions of unit 0 are listed in table 11.1, and the functions of unit 1 are listed in table 11.2. The block diagram of unit 0 is shown in figure 11.1 and the block diagram of unit 1 is shown in figure 11.2.

The descriptions in this section refer to unit 0.

11.1 Features

- Maximum 16-pulse input/output
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - Synchronous operations:
    Multiple timer counters (TCNT) can be written to simultaneously
    Simultaneous clearing by compare match and input capture possible
    Register simultaneous input/output possible by counter synchronous operation
  - Maximum of 15-phase PWM output possible by combination with synchronous operation
- Buffer operation settable for channels 0 and 3
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
- Cascaded operation
- Fast access via internal 16-bit bus
- 26 interrupt sources
- Automatic transfer of register data
- Programmable pulse generator (PPG) output trigger can be generated
- A/D converter conversion start trigger can be generated
- Module stop mode can be set

Table 11.1 TPU (Unit 0) Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock		φ/1 φ/4 φ/16 φ/64 TCLKA TCLKB TCLKC TCLKD	φ/1 φ/4 φ/16 φ/64 φ/256 TCLKA TCLKB	φ/1 φ/4 φ/16 φ/64 φ/1024 TCLKA TCLKB TCLKC	φ/1 φ/4 φ/16 φ/64 φ/256 φ/1024 φ/4096 TCLKA	φ/1 φ/4 φ/16 φ/64 φ/1024 TCLKA TCLKC	φ/1 φ/4 φ/16 φ/64 φ/256 TCLKA TCLKC TCLKD
General registers (TGR)		TGRA_0 TGRB_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4	TGRA_5 TGRB_5
General registers/ buffer registers		TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TIOCA5 TIOCB5
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	○	○	○	○	○	○
	1 output	○	○	○	○	○	○
	Toggle output	○	○	○	○	○	○
Input capture function		○	○	○	○	○	○
Synchronous operation		○	○	○	○	○	○
PWM mode		○	○	○	○	○	○
Phase counting mode		—	○	○	—	○	○
Buffer operation		○	—	—	○	—	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
DMAC activation	TGRA_0 compare match or input capture	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture	TGRA_3 compare match or input capture	TGRA_4 compare match or input capture	TGRA_5 compare match or input capture
A/D converter trigger	TGRA_0 compare match or input capture	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture	TGRA_3 compare match or input capture	TGRA_4 compare match or input capture	TGRA_5 compare match or input capture
PPG trigger	TGRA_0/ TGRB_0 compare match or input capture	TGRA_1/ TGRB_1 compare match or input capture	TGRA_2/ TGRB_2 compare match or input capture	TGRA_3/ TGRB_3 compare match or input capture	—	—
Interrupt sources	5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	4 sources • Compare match or input capture 4A • Compare match or input capture 4B • Overflow • Underflow	4 sources • Compare match or input capture 5A • Compare match or input capture 5B • Overflow • Underflow

Legend:

○: Possible

—: Not possible

Table 11.2 TPU (Unit 1) Functions

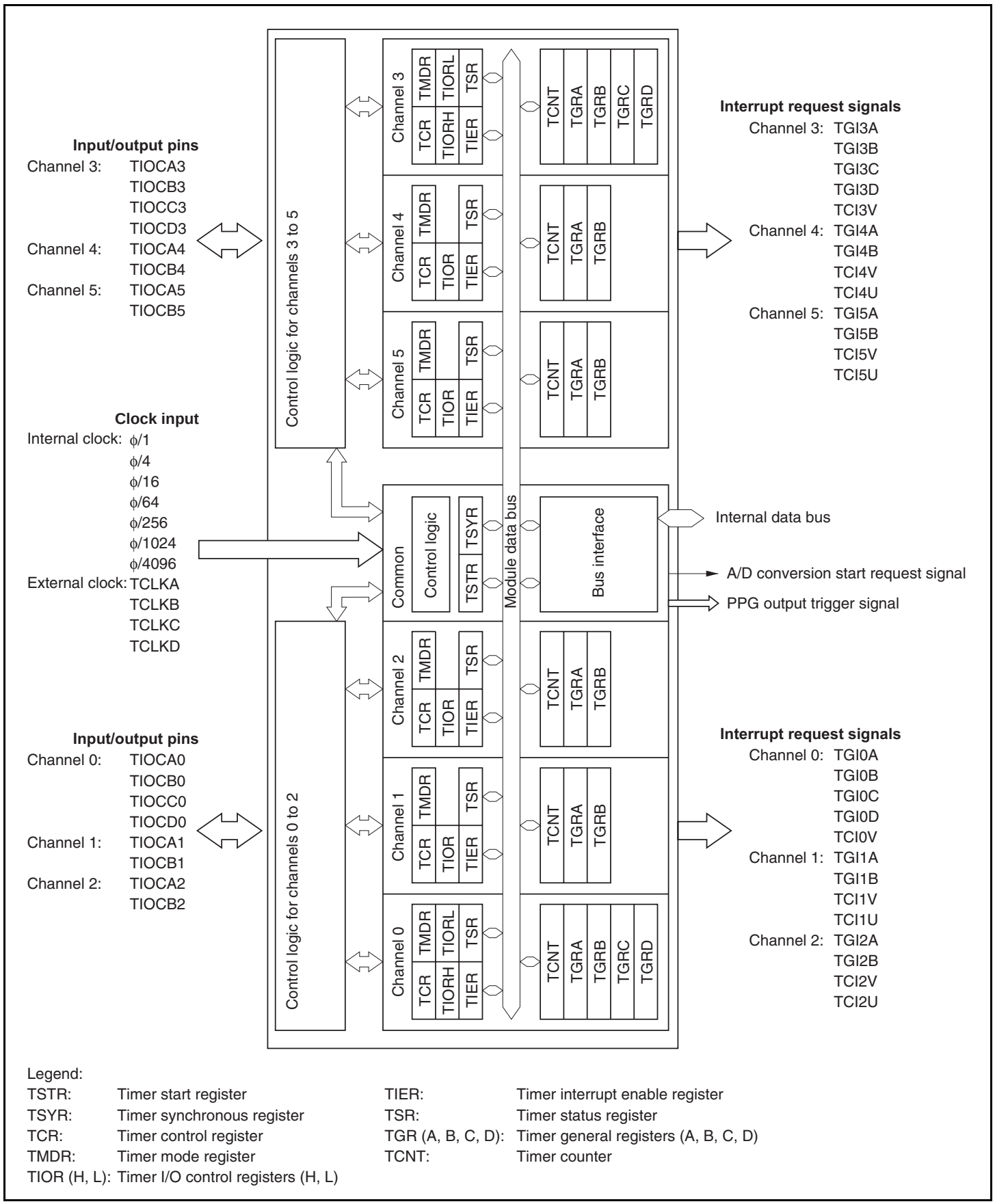
Item		Channel 6	Channel 7	Channel 8	Channel 9	Channel 10	Channel 11
Count clock		φ/1 φ/4 φ/16 φ/64 TCLKE TCLKF TCLKG TCLKH	φ/1 φ/4 φ/16 φ/64 φ/256 TCLKE TCLKF	φ/1 φ/4 φ/16 φ/64 φ/1024 TCLKE TCLKF TCLKG	φ/1 φ/4 φ/16 φ/64 φ/256 φ/1024 φ/4096 TCLKE	φ/1 φ/4 φ/16 φ/64 φ/1024 TCLKE TCLKG	φ/1 φ/4 φ/16 φ/64 φ/256 TCLKE TCLKG TCLKH
General registers (TGR)		TGRA_6 TGRB_6	TGRA_7 TGRB_7	TGRA_8 TGRB_8	TGRA_9 TGRB_9	TGRA_10 TGRB_10	TGRA_11 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	○	○	○	○	○	○
	1 output	○	○	○	○	○	○
	Toggle output	○	○	○	○	○	○
Input capture function		○	○	○	○	○	○
Synchronous operation		○	○	○	○	○	○
PWM mode		○	○	○	○	○	○
Phase counting mode		—	○	○	—	○	○
Buffer operation		○	—	—	○	—	—

Item	Channel 6	Channel 7	Channel 8	Channel 9	Channel 10	Channel 11
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
DMAC activation	—	—	—	—	—	—
A/D converter trigger	TGRA_6 compare match or input capture	TGRA_7 compare match or input capture	TGRA_8 compare match or input capture	TGRA_9 compare match or input capture	TGRA_10 compare match or input capture	TGRA_11 compare match or input capture
PPG trigger	TGRA_6/ TGRB_6 compare match or input capture	TGRA_7/ TGRB_7 compare match or input capture	TGRA_8/ TGRB_8 compare match or input capture	TGRA_9/ TGRB_9 compare match or input capture	—	—
Interrupt sources	5 sources • Compare match or input capture 6A • Compare match or input capture 6B • Compare match or input capture 6C • Compare match or input capture 6D • Overflow	4 sources • Compare match or input capture 7A • Compare match or input capture 7B • Overflow • Underflow	4 sources • Compare match or input capture 8A • Compare match or input capture 8B • Overflow • Underflow	5 sources • Compare match or input capture 9A • Compare match or input capture 9B • Compare match or input capture 9C • Compare match or input capture 9D • Overflow	4 sources • Compare match or input capture 10A • Compare match or input capture 10B • Overflow • Underflow	4 sources • Compare match or input capture 11A • Compare match or input capture 11B • Overflow • Underflow

Legend:

○: Possible

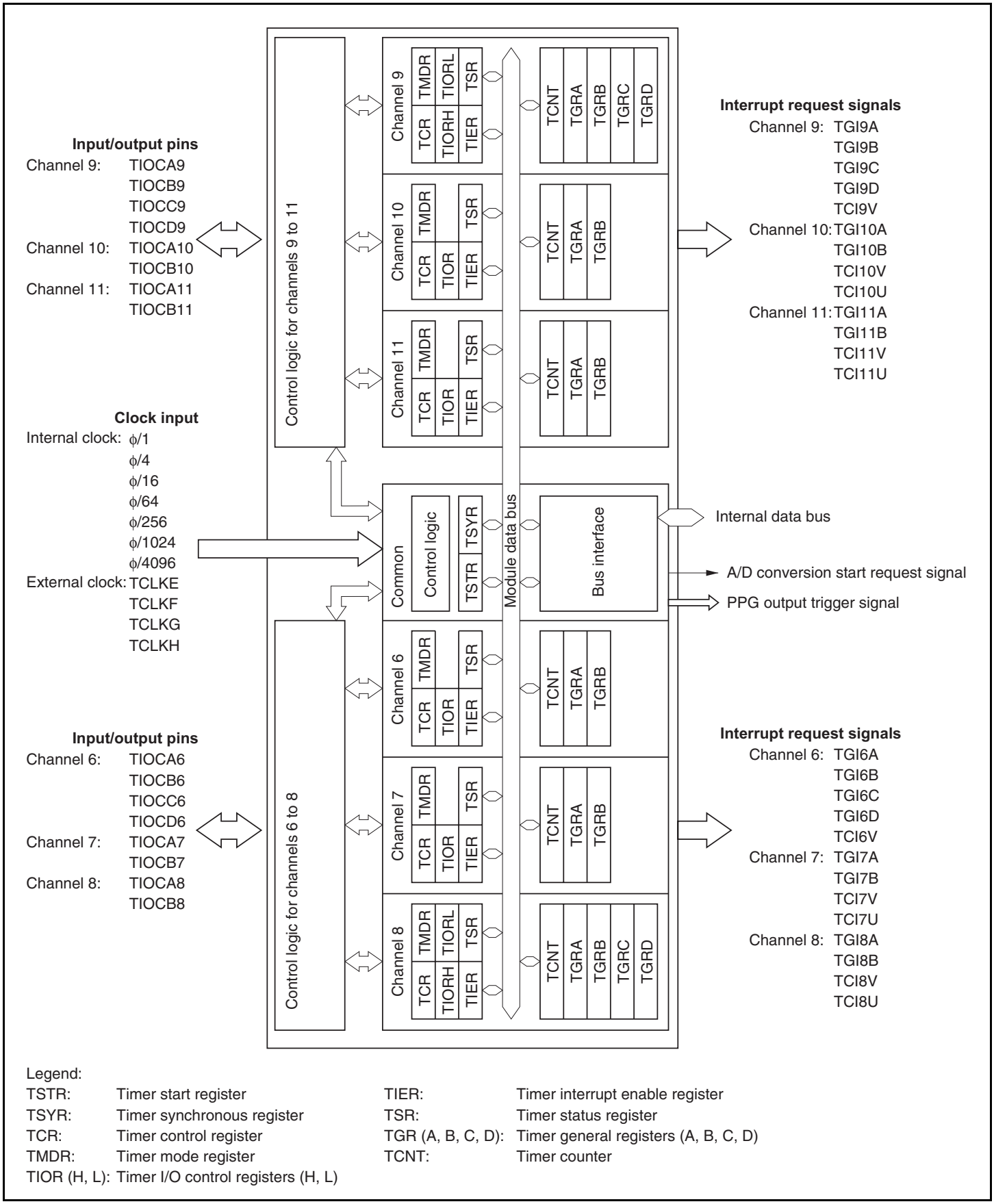
—: Not possible



Legend:

TSTR:	Timer start register	TIER:	Timer interrupt enable register
TSYR:	Timer synchronous register	TSR:	Timer status register
TCR:	Timer control register	TGR (A, B, C, D):	Timer general registers (A, B, C, D)
TMDR:	Timer mode register	TCNT:	Timer counter
TIOR (H, L):	Timer I/O control registers (H, L)		

Figure 11.1 Block Diagram of TPU (Unit 0)



Legend:

TSTR:	Timer start register	TIER:	Timer interrupt enable register
TSYR:	Timer synchronous register	TSR:	Timer status register
TCR:	Timer control register	TGR (A, B, C, D):	Timer general registers (A, B, C, D)
TMDR:	Timer mode register	TCNT:	Timer counter
TIOR (H, L):	Timer I/O control registers (H, L)		

Figure 11.2 Block Diagram of TPU (Unit 1)

11.2 Input/Output Pins

Table 11.3 Pin Configuration

Unit	Channel	Symbol	I/O	Function
0	All	TCLKA	Input	External clock A input pin (Channel 1 and 5 phase counting mode A phase input)
		TCLKB	Input	External clock B input pin (Channel 1 and 5 phase counting mode B phase input)
		TCLKC	Input	External clock C input pin (Channel 2 and 4 phase counting mode A phase input)
		TCLKD	Input	External clock D input pin (Channel 2 and 4 phase counting mode B phase input)
	0	TIOCA0	I/O	TGRA_0 input capture input/output compare output/PWM output pin
		TIOCB0	I/O	TGRB_0 input capture input/output compare output/PWM output pin
		TIOCC0	I/O	TGRC_0 input capture input/output compare output/PWM output pin
		TIOCD0	I/O	TGRD_0 input capture input/output compare output/PWM output pin
	1	TIOCA1	I/O	TGRA_1 input capture input/output compare output/PWM output pin
		TIOCB1	I/O	TGRB_1 input capture input/output compare output/PWM output pin
	2	TIOCA2	I/O	TGRA_2 input capture input/output compare output/PWM output pin
		TIOCB2	I/O	TGRB_2 input capture input/output compare output/PWM output pin
	3	TIOCA3	I/O	TGRA_3 input capture input/output compare output/PWM output pin
		TIOCB3	I/O	TGRB_3 input capture input/output compare output/PWM output pin
		TIOCC3	I/O	TGRC_3 input capture input/output compare output/PWM output pin
		TIOCD3	I/O	TGRD_3 input capture input/output compare output/PWM output pin

Unit	Channel	Symbol	I/O	Function
0	4	TIOCA4	I/O	TGRA_4 input capture input/output compare output/PWM output pin
		TIOCB4	I/O	TGRB_4 input capture input/output compare output/PWM output pin
	5	TIOCA5	I/O	TGRA_5 input capture input/output compare output/PWM output pin
		TIOCB5	I/O	TGRB_5 input capture input/output compare output/PWM output pin
1	All	TCLKE	Input	External clock E input pin (Channel 7 and 11 phase counting mode A phase input)
		TCLKF	Input	External clock F input pin (Channel 7 and 11 phase counting mode B phase input)
		TCLKG	Input	External clock G input pin (Channel 8 and 10 phase counting mode A phase input)
		TCLKH	Input	External clock H input pin (Channel 8 and 10 phase counting mode B phase input)
	6	TIOCA6	I/O	TGRA_6 input capture input/output compare output/PWM output pin
		TIOCB6	I/O	TGRB_6 input capture input/output compare output/PWM output pin
		TIOCC6	I/O	TGRC_6 input capture input/output compare output/PWM output pin
		TIOCD6	I/O	TGRD_6 input capture input/output compare output/PWM output pin
	7	TIOCA7	I/O	TGRA_7 input capture input/output compare output/PWM output pin
		TIOCB7	I/O	TGRB_7 input capture input/output compare output/PWM output pin
	8	TIOCA8	I/O	TGRA_8 input capture input/output compare output/PWM output pin
		TIOCB8	I/O	TGRB_8 input capture input/output compare output/PWM output pin

Unit	Channel	Symbol	I/O	Function
1	9	TIOCA9	I/O	TGRA_9 input capture input/output compare output/PWM output pin
		TIOCB9	I/O	TGRB_9 input capture input/output compare output/PWM output pin
		TIOCC9	I/O	TGRC_9 input capture input/output compare output/PWM output pin
		TIOCD9	I/O	TGRD_9 input capture input/output compare output/PWM output pin
	10	TIOCA10	I/O	TGRA_10 input capture input/output compare output/PWM output pin
		TIOCB10	I/O	TGRB_10 input capture input/output compare output/PWM output pin
	11	TIOCA11	I/O	TGRA_11 input capture input/output compare output/PWM output pin
		TIOCB11	I/O	TGRB_11 input capture input/output compare output/PWM output pin

11.3 Register Descriptions

The TPU has the following registers in each channel. The descriptions in this section refer to the registers of unit 0.

Unit 0:

Channel 0

- Timer control register_0 (TCR_0)
- Timer mode register_0 (TMDR_0)
- Timer I/O control register H_0 (TIORH_0)
- Timer I/O control register L_0 (TIORL_0)
- Timer interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)

Channel 1

- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register_1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)

Channel 2

- Timer control register_2 (TCR_2)
- Timer mode register_2 (TMDR_2)
- Timer I/O control register_2 (TIOR_2)
- Timer interrupt enable register_2 (TIER_2)
- Timer status register_2 (TSR_2)
- Timer counter_2 (TCNT_2)
- Timer general register A_2 (TGRA_2)
- Timer general register B_2 (TGRB_2)

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)

Channel 5

- Timer control register_5 (TCR_5)
- Timer mode register_5 (TMDR_5)
- Timer I/O control register_5 (TIOR_5)
- Timer interrupt enable register_5 (TIER_5)
- Timer status register_5 (TSR_5)
- Timer counter_5 (TCNT_5)
- Timer general register A_5 (TGRA_5)
- Timer general register B_5 (TGRB_5)

Common Registers of Unit 0

- 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 7

- Timer control register_7 (TCR_7)
- Timer mode register_7 (TMDR_7)
- Timer I/O control register_7 (TIOR_7)
- Timer interrupt enable register_7 (TIER_7)
- Timer status register_7 (TSR_7)
- Timer counter_7 (TCNT_7)
- Timer general register A_7 (TGRA_7)
- Timer general register B_7 (TGRB_7)

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 10

- Timer control register_10 (TCR_10)
- Timer mode register_10 (TMDR_10)
- Timer I/O control register_10 (TIOR_10)
- Timer interrupt enable register_10 (TIER_10)
- Timer status register_10 (TSR_10)
- Timer counter_10 (TCNT_10)
- Timer general register A_10 (TGRA_10)
- Timer general register B_10 (TGRB_10)

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 of Unit 1

- Timer start register B (TSTRB)
- Timer synchronous register B (TSYRB)

11.3.1 Timer Control Register (TCR)

The TCR registers control the TCNT operation for each channel. The TPU has a total of six TCR registers, one for each channel. TCR register settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial Value	R/W	Description
7	CCLR2	0	R/W	Counter Clear 2 to 0
6	CCLR1	0	R/W	These bits select the TCNT counter clearing source. See tables 11.4 and 11.5 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. When the input clock is counted using both edges, the input clock period is halved (e.g. ϕ/4 both edges = ϕ/2 rising edge). If phase counting mode is used on channels 1, 2, 4, and 5, this setting is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is ϕ/4 or slower. This setting is ignored if the input clock is ϕ/1, or when overflow/underflow of another channel is selected. 00: Count at rising edge 01: Count at falling edge 1×: Count at both edges Legend: ×: Don't care
2	TPSC2	0	R/W	Time Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 11.6 to 11.11 for details.
0	TPSC0	0	R/W	

Table 11.4 CCLR2 to CCLR0 (Channels 0 and 3)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*2
		1	0	TCNT cleared by TGRD compare match/input capture*2
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

Table 11.5 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
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot be modified.

Table 11.6 TPSC2 to TPSC0 (Channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	External clock: counts on TCLKD pin input

Table 11.7 TPSC2 to TPSC0 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on ϕ/256
			1	Counts on TCNT2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

Table 11.8 TPSC2 to TPSC0 (Channel 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	Internal clock: counts on ϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 11.9 TPSC2 to TPSC0 (Channel 3)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	Internal clock: counts on ϕ/1024
		1	0	Internal clock: counts on ϕ/256
			1	Internal clock: counts on ϕ/4096

Table 11.10 TPSC2 to TPSC0 (Channel 4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
4	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKC pin input
		1	0	Internal clock: counts on ϕ/1024
			1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

Table 11.11 TPSC2 to TPSC0 (Channel 5)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKC pin input
		1	0	Internal clock: counts on ϕ/256
			1	External clock: counts on TCLKD pin input

Note: This setting is ignored when channel 5 is in phase counting mode.

11.3.2 Timer Mode Register (TMDR)

TMDR registers are used to set the operating mode for each channel. The TPU has six TMDR registers, one for each channel. TMDR register settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved
6	—	1	—	These bits are always read as 1 and cannot be modified.
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: TGRB operates normally 1: TGRB and TGRD used together for buffer operation
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified. 0: TGRA operates normally 1: TGRA and TGRC used together for buffer operation
3	MD3	0	R/W	Modes 3 to 0 These bits are used to set the timer operating mode. MD3 is a reserved bit. The write value should always be 0. See table 11.12 for details.
2	MD2	0	R/W	
1	MD1	0	R/W	
0	MD0	0	R/W	

Table 11.12 MD3 to MD0

Bit 3 MD3*1	Bit 2 MD2*2	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	×	×	×	—

Legend: ×: Don't care

Notes: 1. MD3 is a reserved bit. In a write, it should always be written with 0.
2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 should always be written to MD2.

11.3.3 Timer I/O Control Register (TIOR)

TIOR registers control the TGR registers. The TPU has eight TIOR registers, two each for channels 0 and 3, and one each for channels 1, 2, 4, and 5. Care is required since TIOR is affected by the TMDR setting.

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIOR_4, TIOR_5

Bit	Bit Name	Initial Value	R/W	Description
7	IOB3	0	R/W	I/O Control B3 to B0
6	IOB2	0	R/W	Specify the function of TGRB.
5	IOB1	0	R/W	For details, see tables 11.13, 11.15, 11.16, 11.17, 11.19, and 11.20.
4	IOB0	0	R/W	
3	IOA3	0	R/W	I/O Control A3 to A0
2	IOA2	0	R/W	Specify the function of TGRA.
1	IOA1	0	R/W	For details, see tables 11.21, 11.23, 11.24, 11.25, 11.27, and 11.28.
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 11.14 and 11.18.
4	IOD0	0	R/W	
3	IOC3	0	R/W	I/O Control C3 to C0
2	IOC2	0	R/W	Specify the function of TGRC.
1	IOC1	0	R/W	For details, see tables 11.22 and 11.26.
0	IOC0	0	R/W	

Table 11.13 TIORH_0

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_0 Function**	**TIOCB0 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB0 pin Input capture at rising edge
			1		Capture input source is TIOCB0 pin Input capture at falling edge
		1	×		Capture input source is TIOCB0 pin Input capture at both edges
	1	×	×		Capture input source is channel 1/count clock Input capture at TCNT_1 count- up/count-down*

Legend: ×: Don’t care

Note: * When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and φ/1 is used as the TCNT_1 count clock, this setting is invalid and input capture is not generated.

Table 11.14 TIORL_0

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_0 Function**	**TIOCD0 Pin Function**
0	0	0	0	Output compare register*2	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD0 pin Input capture at rising edge
			1		Capture input source is TIOCD0 pin Input capture at falling edge
		1	×		Capture input source is TIOCD0 pin Input capture at both edges
	1	×	×		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down*1

Legend: ×: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and 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.

Table 11.15 TIOR_1

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_1 Function**	**TIOCB1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 pin Input capture at rising edge
			1		Capture input source is TIOCB1 pin Input capture at falling edge
		1	×		Capture input source is TIOCB1 pin Input capture at both edges
	1	×	×		TGRC_0 compare match/input capture Input capture at generation of TGRC_0 compare match/input capture

Legend: ×: Don’t care

Table 11.16 TIOR_2

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_2 Function**	**TIOCB2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	×	0	0	Input capture register	Capture input source is TIOCB2 pin Input capture at rising edge
			1		Capture input source is TIOCB2 pin Input capture at falling edge
		1	×		Capture input source is TIOCB2 pin Input capture at both edges

Legend: ×: Don’t care

Table 11.17 TIORH_3

				Description	
Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_3 Function	TIOCB3 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB3 pin Input capture at rising edge
			1		Capture input source is TIOCB3 pin Input capture at falling edge
		1	×		Capture input source is TIOCB3 pin Input capture at both edges
	1	×	×		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down*

Legend: ×: Don't care

Note: * When bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and φ/1 is used as the TCNT_4 count clock, this setting is invalid and input capture is not generated.

Table 11.18 TIORL_3

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_3 Function**	**TIOCD3 Pin Function**
0	0	0	0	Output compare register*2	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD3 pin Input capture at rising edge
			1		Capture input source is TIOCD3 pin Input capture at falling edge
		1	×		Capture input source is TIOCD3 pin Input capture at both edges
	1	×	×		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down*1

Legend: ×: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and 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.

Table 11.19 TIOR_4

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_4 Function**	**TIOCB4 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB4 pin Input capture at rising edge
			1		Capture input source is TIOCB4 pin Input capture at falling edge
		1	×		Capture input source is TIOCB4 pin Input capture at both edges
	1	×	×		Capture input source is TGRC_3 compare match/input capture Input capture at generation of TGRC_3 compare match/input capture

Legend: ×: Don't care

Table 11.20 TIOR_5

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_5 Function**	**TIOCB5 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	×	0	0	Input capture register	Capture input source is TIOCB5 pin Input capture at rising edge
			1		Capture input source is TIOCB5 pin Input capture at falling edge
		1	×		Capture input source is TIOCB5 pin Input capture at both edges

Legend: ×: Don’t care

Table 11.21 TIORH_0

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_0 Function**	**TIOCA0 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin Input capture at rising edge
			1		Capture input source is TIOCA0 pin Input capture at falling edge
		1	×		Capture input source is TIOCA0 pin Input capture at both edges
	1	×	×		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

Legend: ×: Don’t care

Table 11.22 TIORL_0

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_0 Function**	**TIOCC0 Pin Function**
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCC0 pin Input capture at rising edge
			1		Capture input source is TIOCC0 pin Input capture at falling edge
		1	×		Capture input source is TIOCC0 pin Input capture at both edges
	1	×	×		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

Legend: ×: Don't care

Note: * When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 11.23 TIOR_1

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_1 Function**	**TIOCA1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA1 pin Input capture at rising edge
			1		Capture input source is TIOCA1 pin Input capture at falling edge
		1	×		Capture input source is TIOCA1 pin Input capture at both edges
	1	×	×		Capture input source is TGRA_0 compare match/input capture Input capture at generation of channel 0/TGRA_0 compare match/input capture

Legend: ×: Don't care

Table 11.24 TIOR_2

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_2 Function**	**TIOCA2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	×	0	0	Input capture register	Capture input source is TIOCA2 pin Input capture at rising edge
			1		Capture input source is TIOCA2 pin Input capture at falling edge
		1	×		Capture input source is TIOCA2 pin Input capture at both edges

Legend: ×: Don't care

Table 11.25 TIORH_3

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_3 Function**	**TIOCA3 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA3 pin Input capture at rising edge
			1		Capture input source is TIOCA3 pin Input capture at falling edge
		1	×		Capture input source is TIOCA3 pin Input capture at both edges
	1	×	×		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down

Legend: ×: Don’t care

Table 11.26 TIORL_3

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_3 Function**	**TIOCC3 Pin Function**
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCC3 pin Input capture at rising edge
			1		Capture input source is TIOCC3 pin Input capture at falling edge
		1	×		Capture input source is TIOCC3 pin Input capture at both edges
	1	×	×		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down

Legend: ×: Don't care

Note: * When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 11.27 TIOR_4

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_4 Function**	**TIOCA4 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA4 pin Input capture at rising edge
			1		Capture input source is TIOCA4 pin Input capture at falling edge
		1	×		Capture input source is TIOCA4 pin Input capture at both edges
	1	×	×		Capture input source is TGRA_3 compare match/input capture Input capture at generation of TGRA_3 compare match/input capture

Legend: ×: Don't care

Table 11.28 TIOR_5

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_5 Function**	**TIOCA5 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	×	0	0	Input capture register	Input capture source is TIOCA5 pin Input capture at rising edge
			1		Input capture source is TIOCA5 pin Input capture at falling edge
		1	×		Input capture source is TIOCA5 pin Input capture at both edges

Legend: ×: Don't care

11.3.4 Timer Interrupt Enable Register (TIER)

TIER registers control enabling or disabling of interrupt requests for each channel. The TPU has six TIER registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TTGE	0	R/W	A/D Conversion Start Request Enable Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match. 0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled
6	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1, 2, 4, and 5. In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0 and 3. In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled

Bit	Bit Name	Initial value	R/W	Description
2	TGIEC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channels 0 and 3. In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGIC) by TGFC bit disabled 1: Interrupt requests (TGIC) by TGFC bit enabled
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

11.3.5 Timer Status Register (TSR)

TSR registers indicate the status of each channel. The TPU has six TSR registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in channels 1, 2, 4, and 5. In channels 0 and 3, bit 7 is reserved. It is always read as 1 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1, 2, 4, and 5 are set to phase counting mode. In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified. [Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF) [Clearing condition] When 0 is written to TCFU after reading TCFU = 1
4	TCFV	0	R/(W)*	Overflow Flag Status flag that indicates that TCNT overflow has occurred. [Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000) [Clearing condition] When 0 is written to TCFV after reading TCFV = 1

Bit	Bit Name	Initial value	R/W	Description
3	TGFD	0	R/(W)*	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of TGRD input capture or compare match in channels 0 and 3. In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register [Clearing conditions] • When DTC is activated by TGID interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFD after reading TGFD = 1
2	TGFC	0	R/(W)*	Input Capture/Output Compare Flag C Status flag that indicates the occurrence of TGRC input capture or compare match in channels 0 and 3. In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIC interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFC after reading TGFC = 1

Bit	Bit Name	Initial value	R/W	Description
1	TGFB	0	R/(W)*	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGRB input capture or compare match. [Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
0	TGFA	0	R/(W)*	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA input capture or compare match. [Setting conditions] When TCNT = TGRA while TGRA is functioning as output compare register When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA interrupt while DTE bit of DMABCR in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1

Note: * Only 0 can be written, for flag clearing.

11.3.6 Timer Counter (TCNT)

The TCNT registers are 16-bit readable/writable counters. The TPU has six TCNT counters, one for each channel.

The TCNT counters are initialized to H'0000 by a reset, or in hardware standby mode.

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

11.3.7 Timer General Register (TGR)

The TGR registers are 16-bit readable/writable registers with a dual function as output compare and input capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two each for channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for operation as buffer registers. The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit. TGR buffer register combinations are TGRA–TGRC and TGRB–TGRD.

11.3.8 Timer Start Register (TSTR)

TSTR selects operation/stoppage for channels 0 to 5. When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bit	Bit Name	Initial value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	The write value should always be 0.
5	CST5	0	R/W	Counter Start 5 to 0 These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_5 to TCNT_0 count operation is stopped 1: TCNT_5 to TCNT_0 performs count operation
4	CST4	0	R/W	
3	CST3	0	R/W	
2	CST2	0	R/W	
1	CST1	0	R/W	
0	CST0	0	R/W	

11.3.9 Timer Synchronous Register (TSYR)

TSYR selects independent operation or synchronous operation for the TCNT counters of channels 0 to 5. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

Bit	Bit Name	Initial value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	The write value should always be 0.
5	SYNC5	0	R/W	Timer Synchronization 5 to 0 These bits select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, synchronous presetting of multiple channels, and synchronous clearing through counter clearing on another channel are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR2 to CCLR0 in TCR. 0: TCNT_5 to TCNT_0 operates independently (TCNT presetting /clearing is unrelated to other channels) 1: TCNT_5 to TCNT_0 performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is possible)
4	SYNC4	0	R/W	
3	SYNC3	0	R/W	
2	SYNC2	0	R/W	
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

11.3.10 Timer Start Register B (TSTRB)

TSTRB selects operation/stoppage for channels 6 to 11. When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bit	Bit Name	Initial value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	The write value should always be 0.
5	CST11	0	R/W	Counter Start 11 to 6
4	CST10	0	R/W	These bits select operation or stoppage for TCNT.
3	CST9	0	R/W	If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value.
2	CST8	0	R/W	
1	CST7	0	R/W	
0	CST6	0	R/W	
				0: TCNT_11 to TCNT_6 count operation is stopped
				1: TCNT_11 to TCNT_6 performs count operation

11.3.11 Timer Synchronous Register B (TSYRB)

TSYRB selects independent operation or synchronous operation for the TCNT counters of channels 6 to 11. A channel performs synchronous operation when the corresponding bit in TSYRB is set to 1.

Bit	Bit Name	Initial value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	The write value should always be 0.
5	SYNC11	0	R/W	Timer Synchronization 11 to 6 These bits select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, synchronous presetting of multiple channels, and synchronous clearing through counter clearing on another channel are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR2 to CCLR0 in TCR. 0: TCNT_11 to TCNT_6 operates independently (TCNT presetting /clearing is unrelated to other channels) 1: TCNT_11 to TCNT_6 performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is possible)
4	SYNC10	0	R/W	
3	SYNC9	0	R/W	
2	SYNC8	0	R/W	
1	SYNC7	0	R/W	
0	SYNC6	0	R/W	

11.4 Operation

11.4.1 Basic Functions

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

(1) Counter Operation

When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

(a) Example of count operation setting procedure

Figure 11.3 shows an example of the count operation setting procedure.

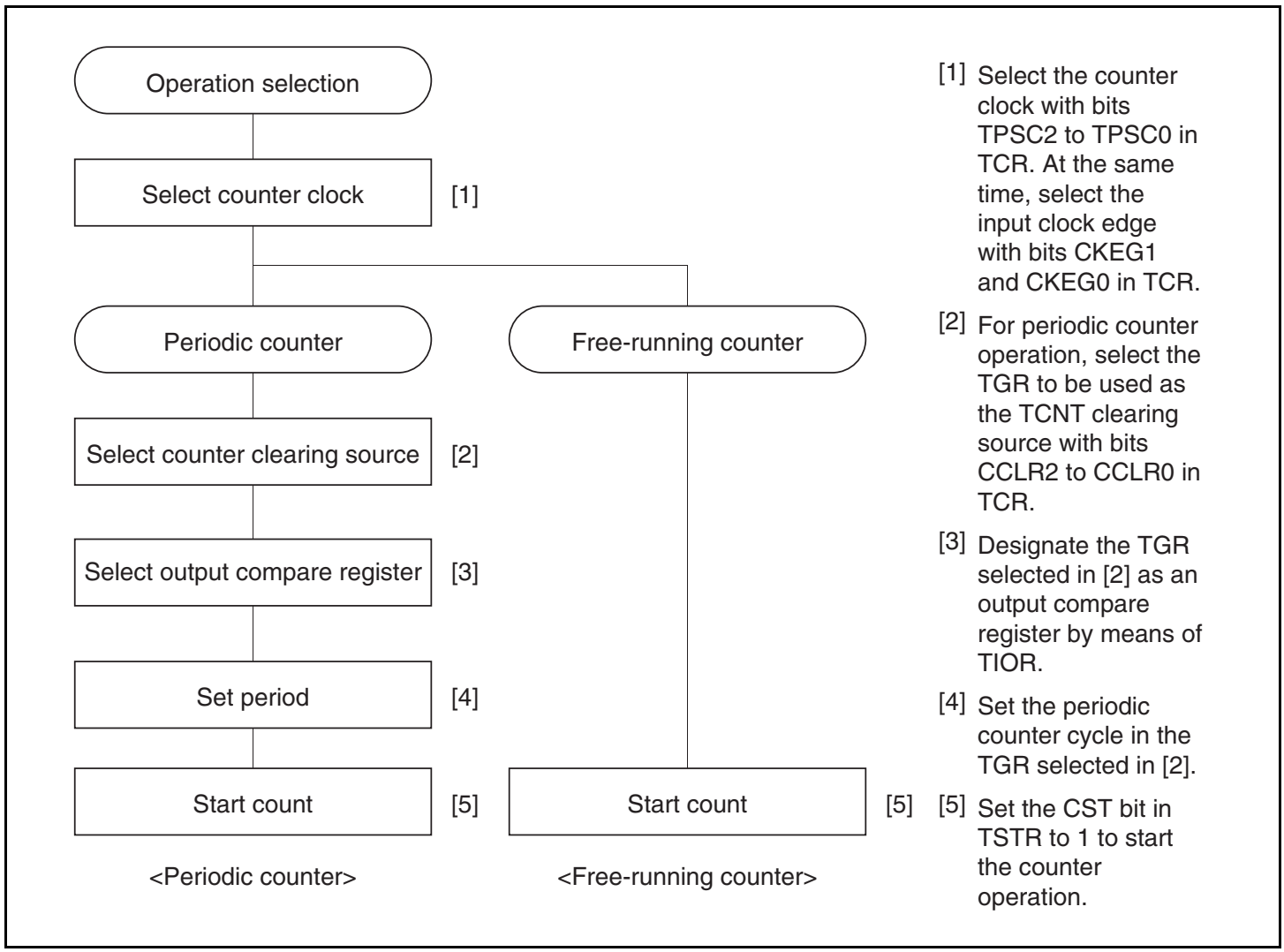


Figure 11.3 Example of Counter Operation Setting Procedure

(b) Free-running count operation and periodic count operation

Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (changes from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

Figure 11.4 illustrates free-running counter operation.

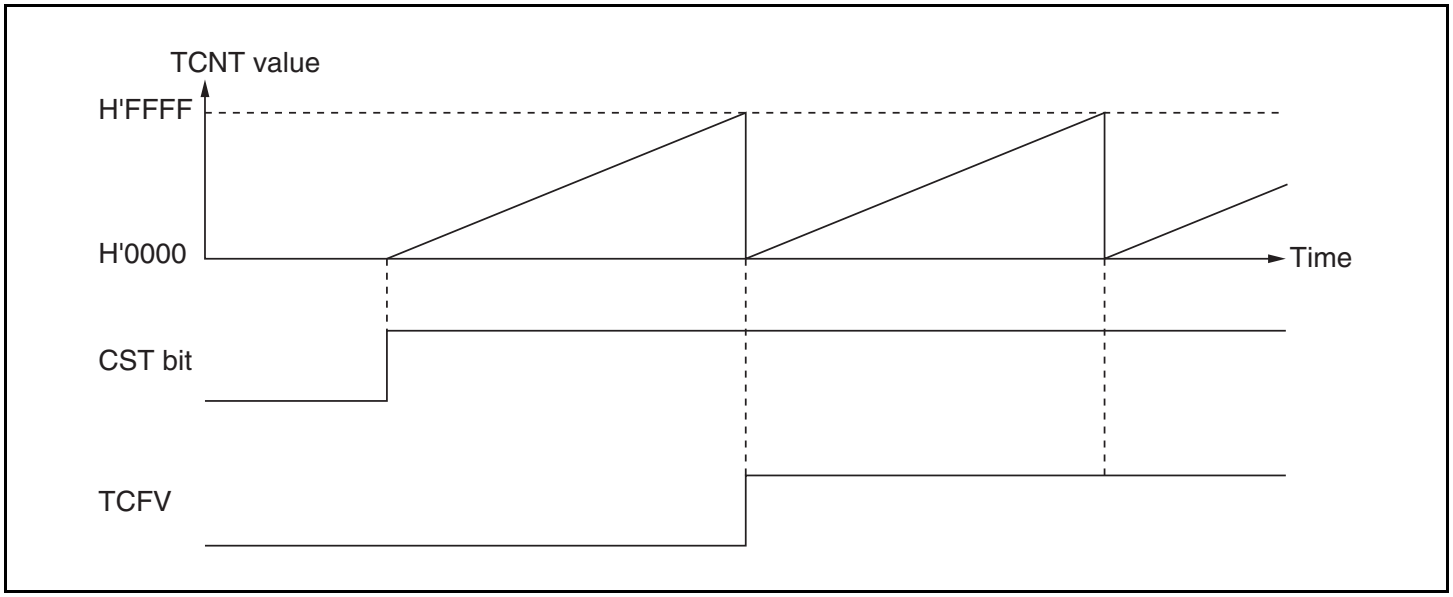


Figure 11.4 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts count-up operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 11.5 illustrates periodic counter operation.

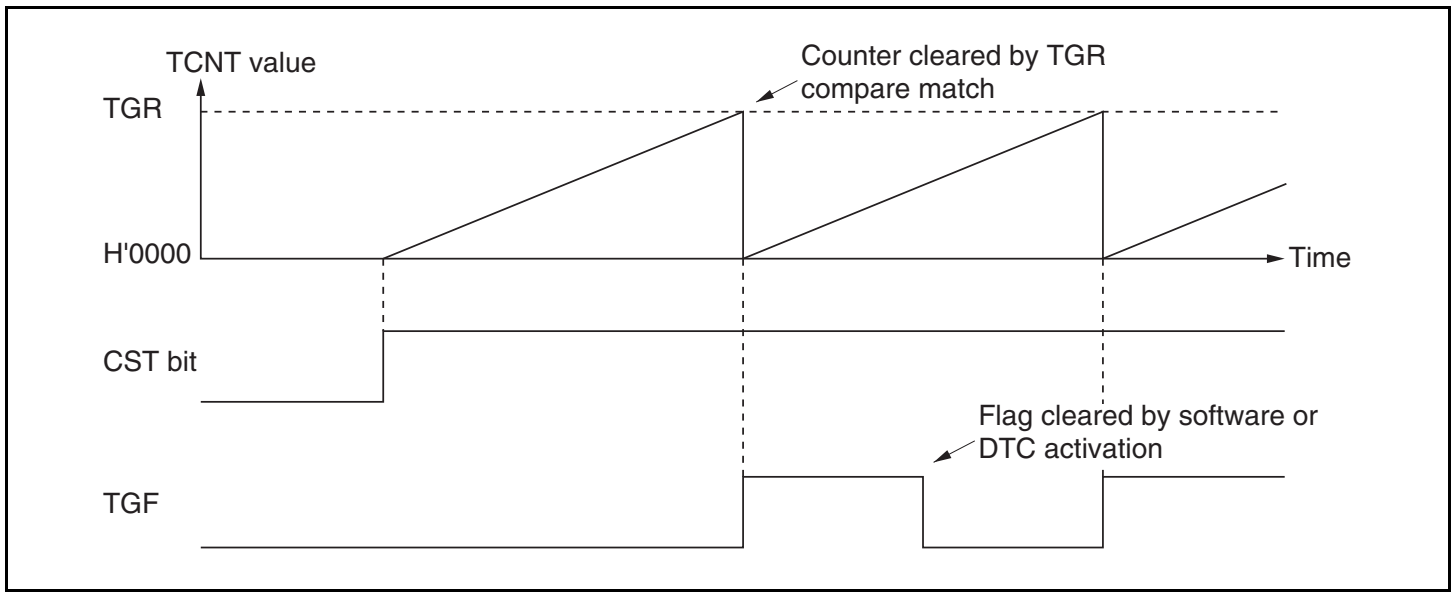


Figure 11.5 Periodic Counter Operation

(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 11.6 shows an example of the setting procedure for waveform output by a compare match.

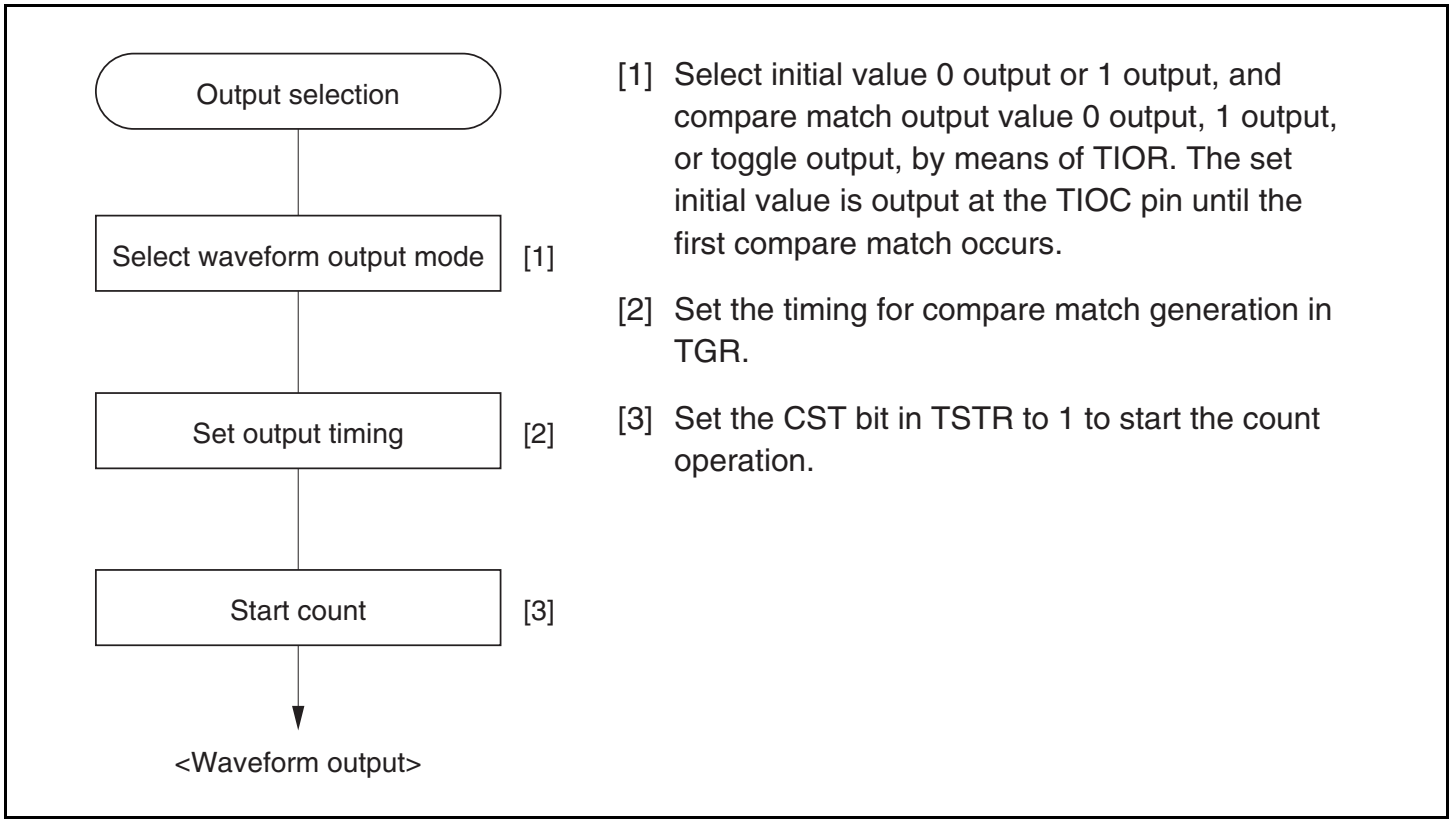


Figure 11.6 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of waveform output operation

Figure 11.7 shows an example of 0 output/1 output.

In this example, TCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level match, the pin level does not change.

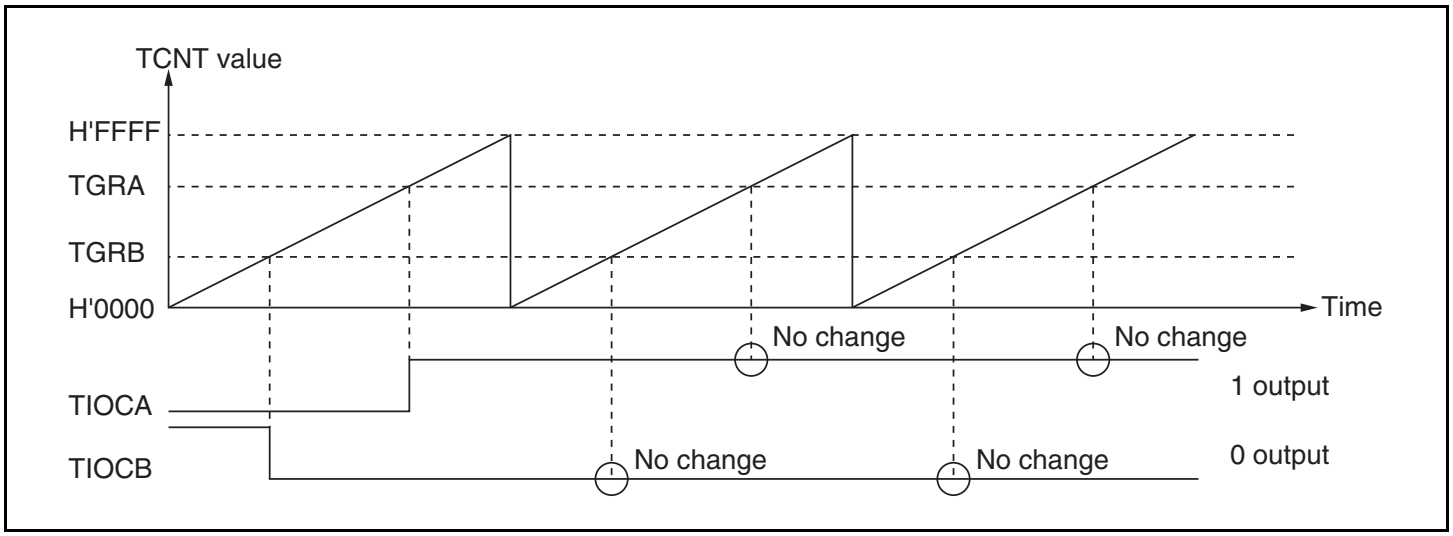


Figure 11.7 Example of 0 Output/1 Output Operation

Figure 11.8 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

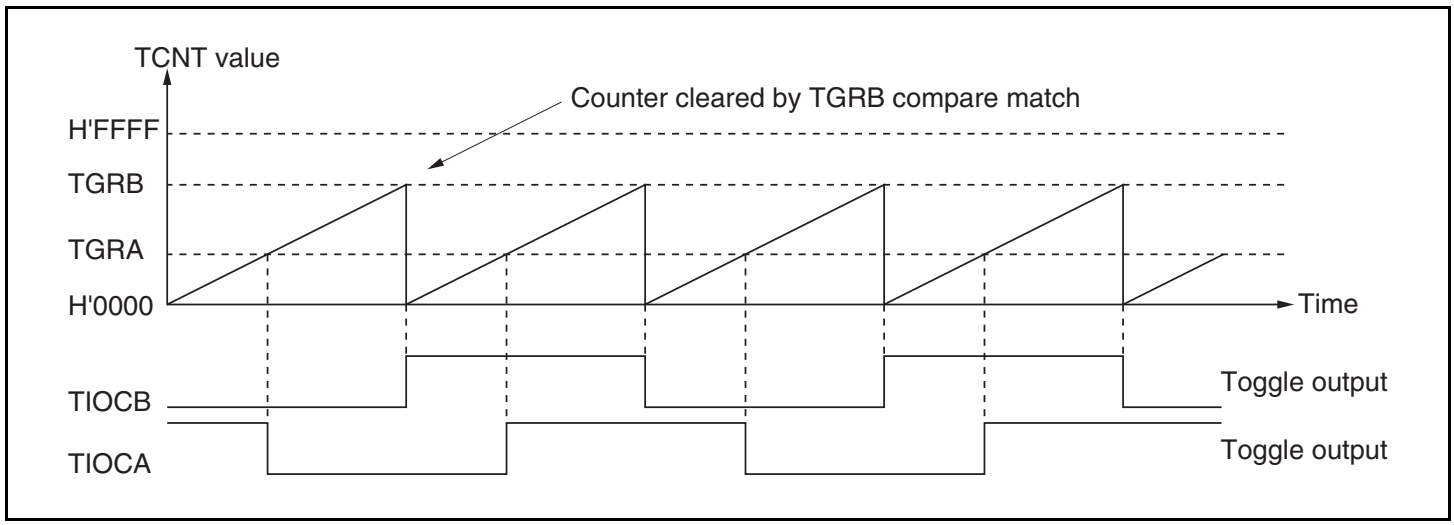


Figure 11.8 Example of Toggle Output Operation

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detection edge. For channels 0, 1, 3, 4, 6, 7, 9, and 10 it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0, 3, 6, and 9, ϕ/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if ϕ/1 is selected.

(a) Example of setting procedure for input capture operation

Figure 11.9 shows an example of the setting procedure for input capture operation.

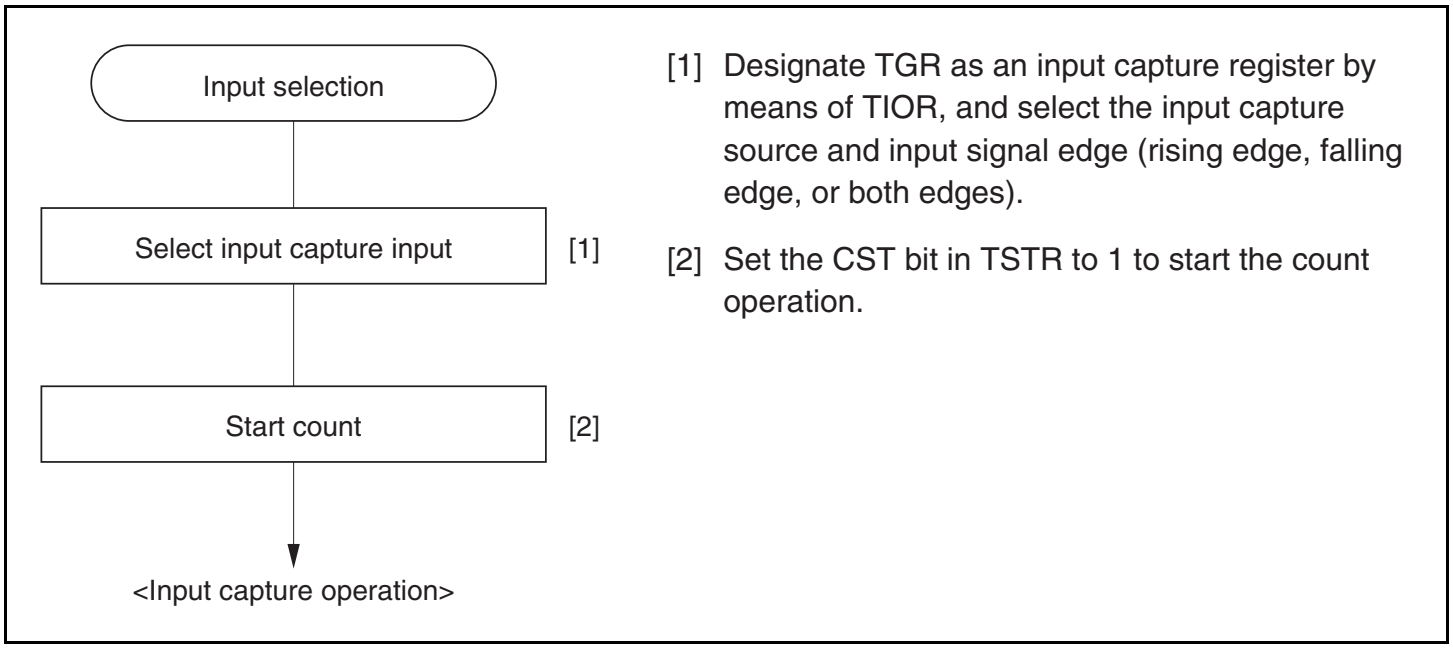


Figure 11.9 Example of Setting Procedure for Input Capture Operation

(b) Example of input capture operation

Figure 11.10 shows an example of input capture operation.

In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

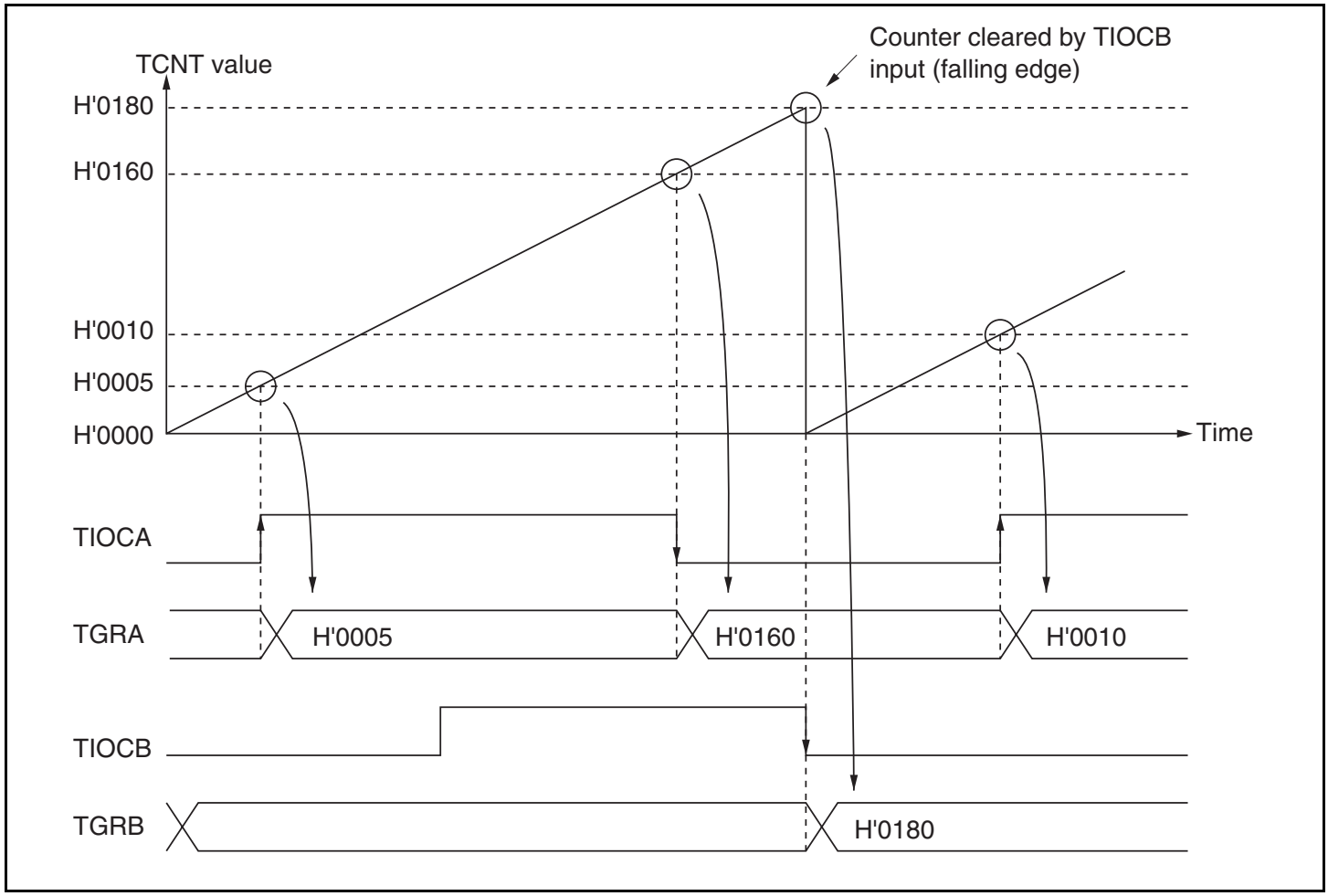


Figure 11.10 Example of Input Capture Operation

11.4.2 Synchronous Operation

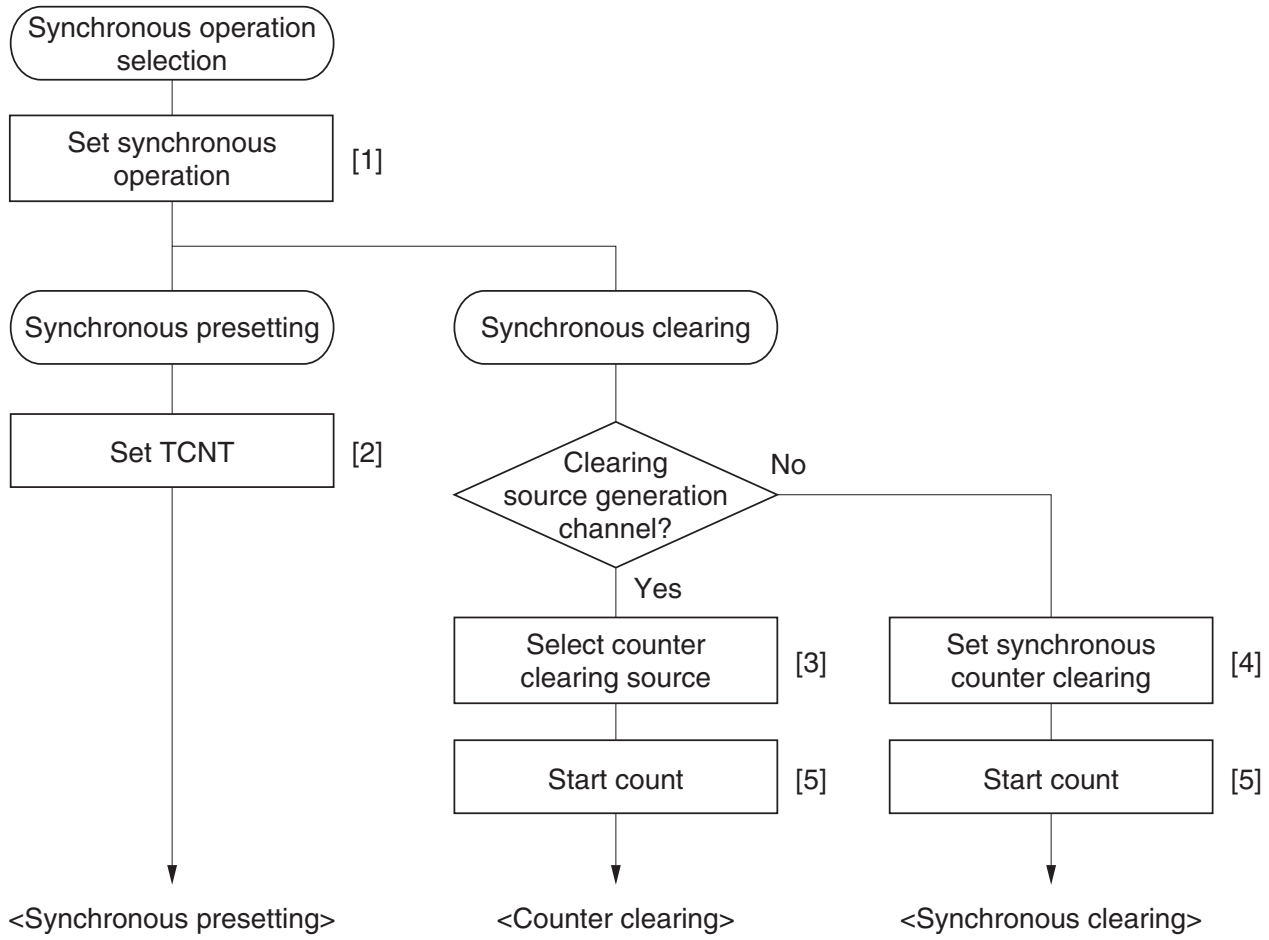
In synchronous operation, the values in multiple TCNT counters can be rewritten simultaneously (synchronous presetting). Also, multiple of TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 5 and 6 to 11 can all be designated for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 11.11 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.

[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.

[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.

[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.

[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 11.11 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 11.12 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOCA0, TIOCA1, and TIOCA2. At this time, synchronous presetting, and synchronous clearing by TGRB_0 compare match, is performed for channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the PWM cycle.

For details on PWM modes, see section 11.4.5, PWM Modes.

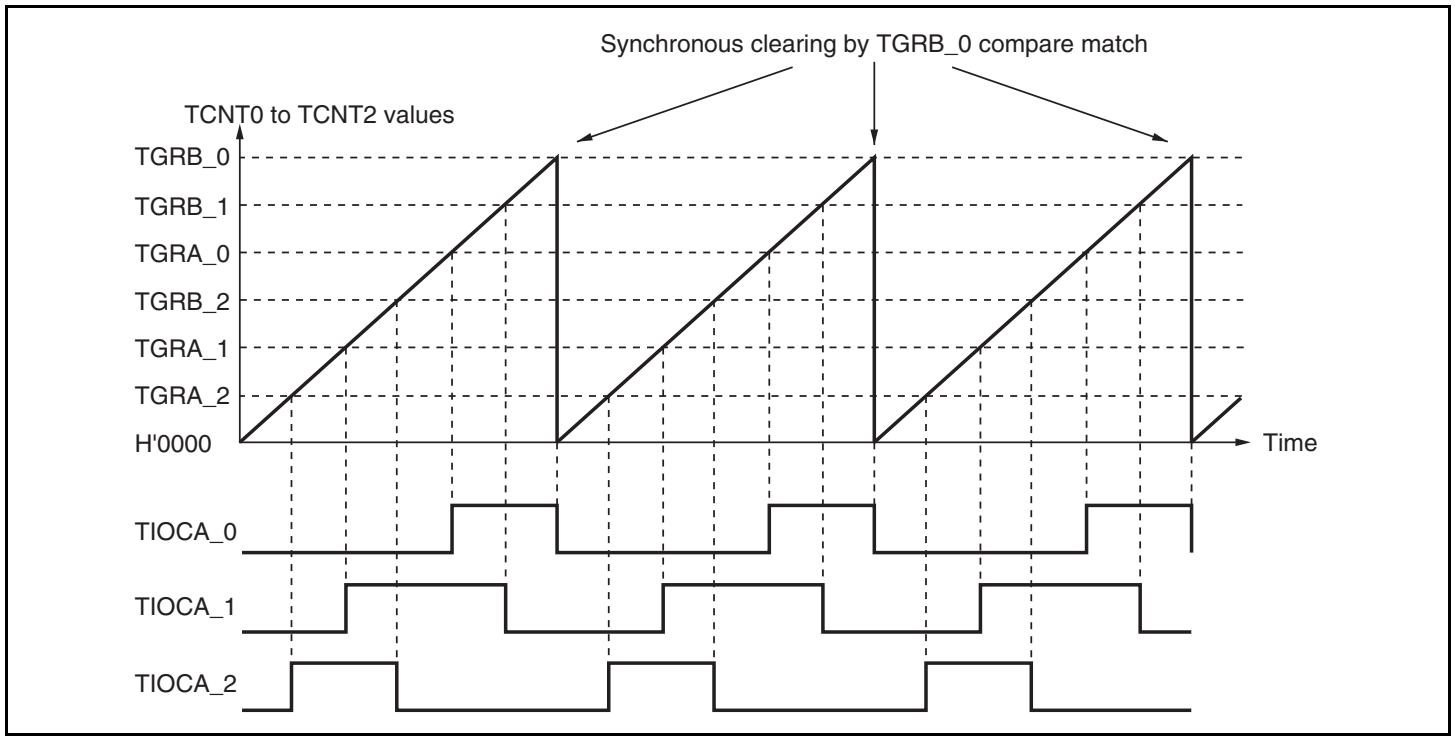


Figure 11.12 Example of Synchronous Operation

11.4.3 Buffer Operation

Buffer operation, provided for channels 0, 3, 6, and 9, enables TGRC and TGRD to be used as buffer registers.

Buffer operation differs depending on whether TGR has been designated as an input capture register or a compare match register.

Table 11.29 shows the register combinations used in buffer operation.

Table 11.29 Register Combinations in Buffer Operation

Unit	Channel	Timer General Register	Buffer Register
0	0	TGRA_0	TGRC_0
		TGRB_0	TGRD_0
	3	TGRA_3	TGRC_3
		TGRB_3	TGRD_3
1	6	TGRA_6	TGRC_6
		TGRB_6	TGRD_6
	9	TGRA_9	TGRC_9
		TGRB_9	TGRD_9

- 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 11.13.

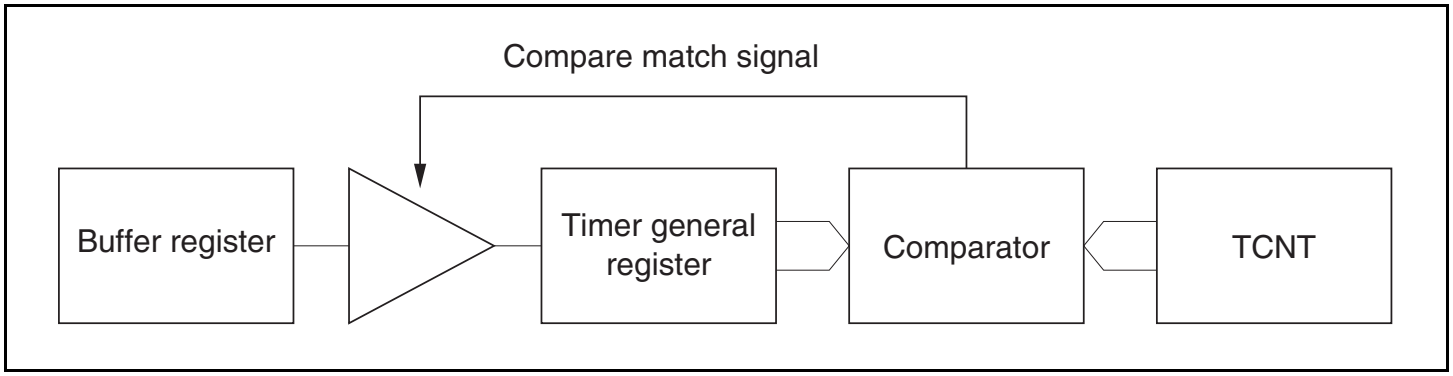


Figure 11.13 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

  This operation is illustrated in figure 11.14.

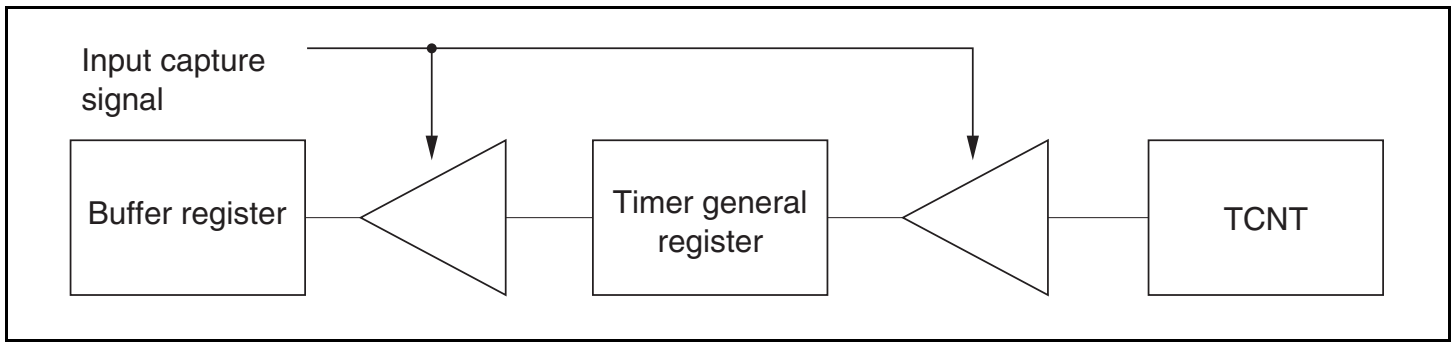


Figure 11.14 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 11.15 shows an example of the buffer operation setting procedure.

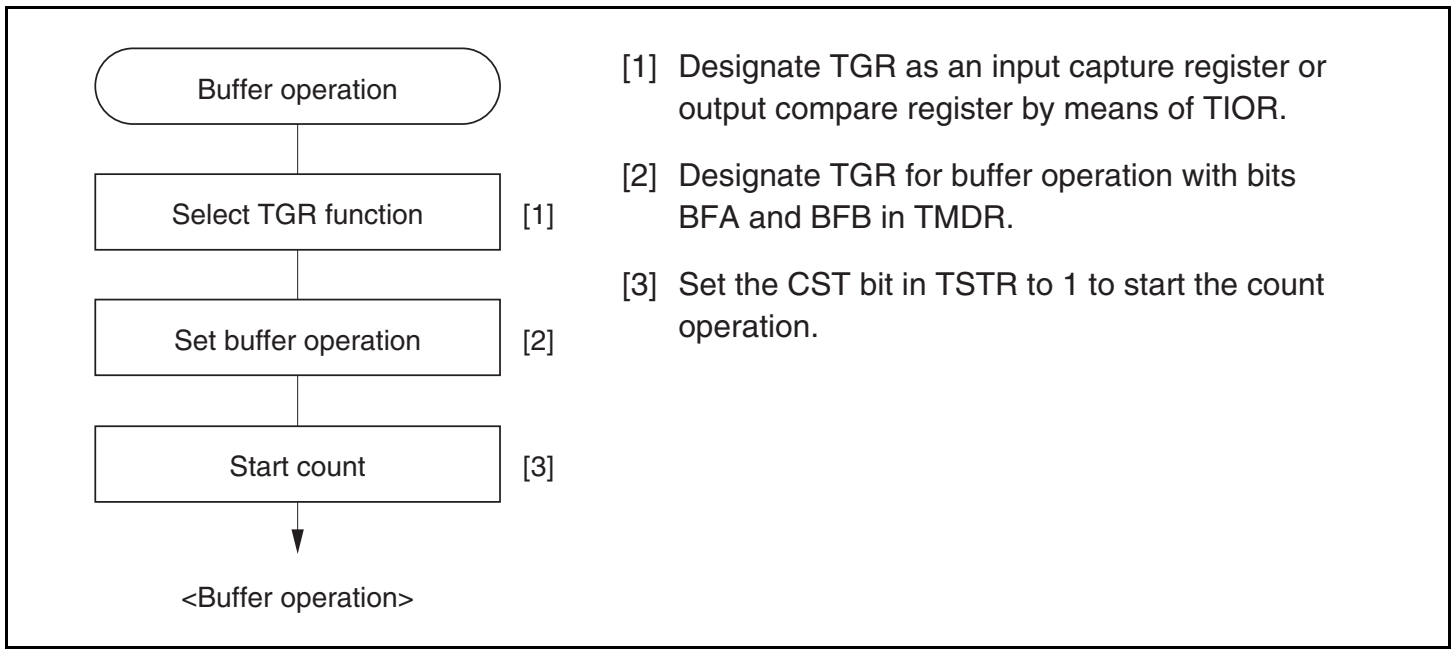


Figure 11.15 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an output compare register

Figure 11.16 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B.

As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

For details on PWM modes, see section 11.4.5, PWM Modes.

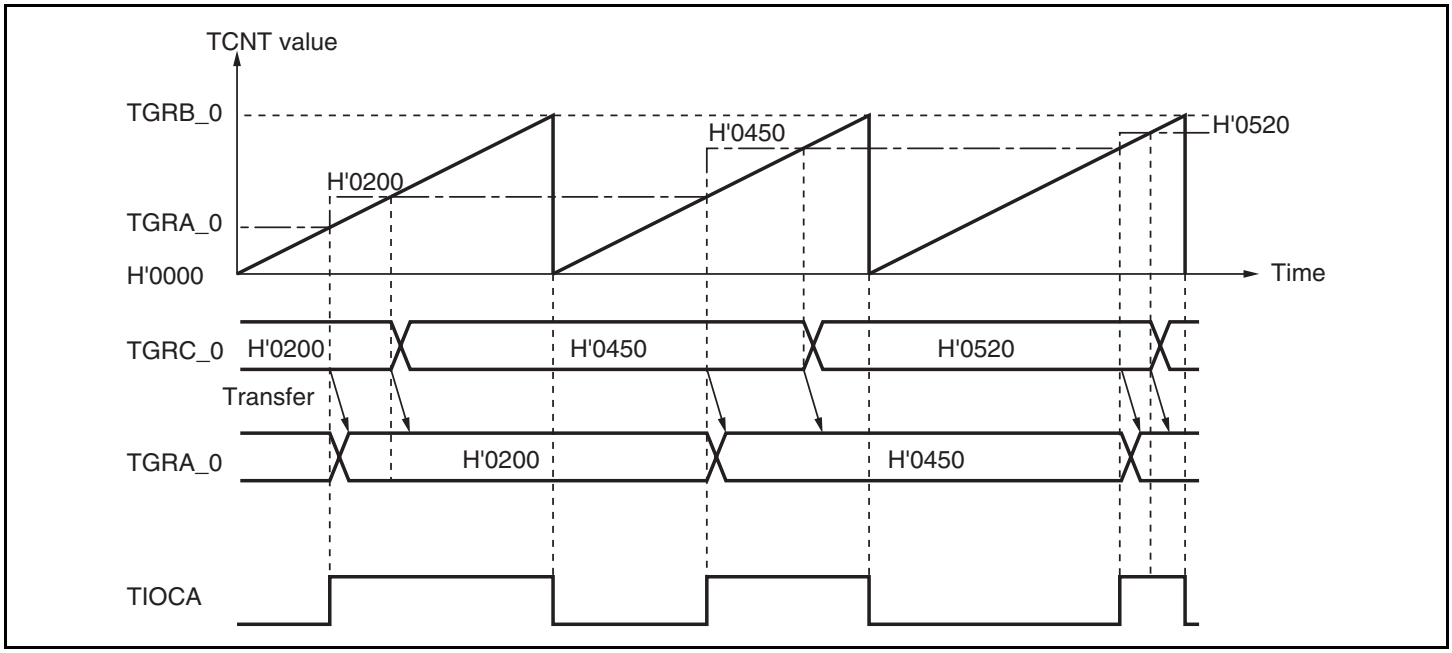


Figure 11.16 Example of Buffer Operation (1)

(b) When TGR is an input capture register

Figure 11.17 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

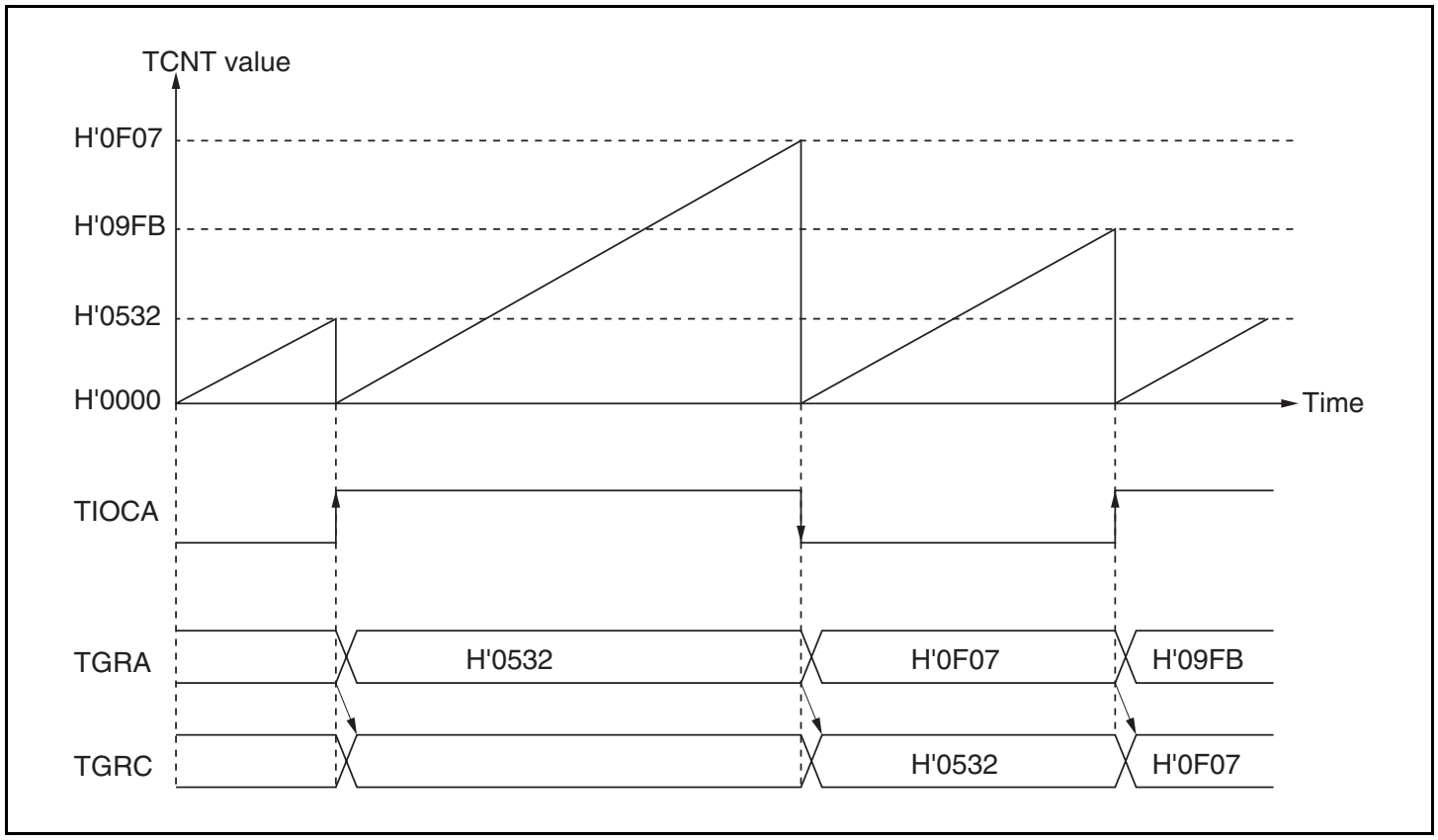


Figure 11.17 Example of Buffer Operation (2)

11.4.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 (channel 4, channel 7, or channel 10) counter clock at overflow/underflow of TCNT_2 (TCNT_5, TCNT_8, or TCNT_11) as set in bits TPSC2 to TPSC0 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase-counting mode.

Table 11.30 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is invalid and the counter operates independently in phase counting mode.

Table 11.30 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2
Channels 4 and 5	TCNT_4	TCNT_5
Channels 7 and 8	TCNT_7	TCNT_8
Channels 10 and 11	TCNT_10	TCNT_11

(1) Example of Cascaded Operation Setting Procedure

Figure 11.18 shows an example of the setting procedure for cascaded operation.

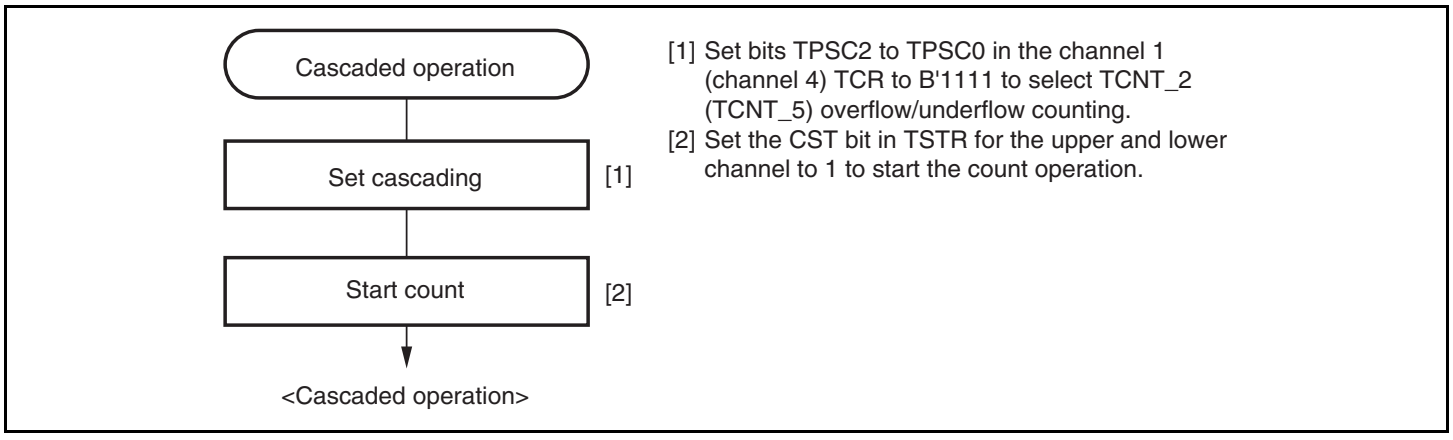


Figure 11.18 Cascaded Operation Setting Procedure

(2) Examples of Cascaded Operation

Figure 11.19 illustrates the operation when counting upon TCNT_2 overflow/underflow has been set for TCNT_1, TGRA_1 and TGRA_2 have been designated as input capture registers, and the TIOC pin rising edge has been selected.

When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TGRA_1, and the lower 16 bits to TGRA_2.

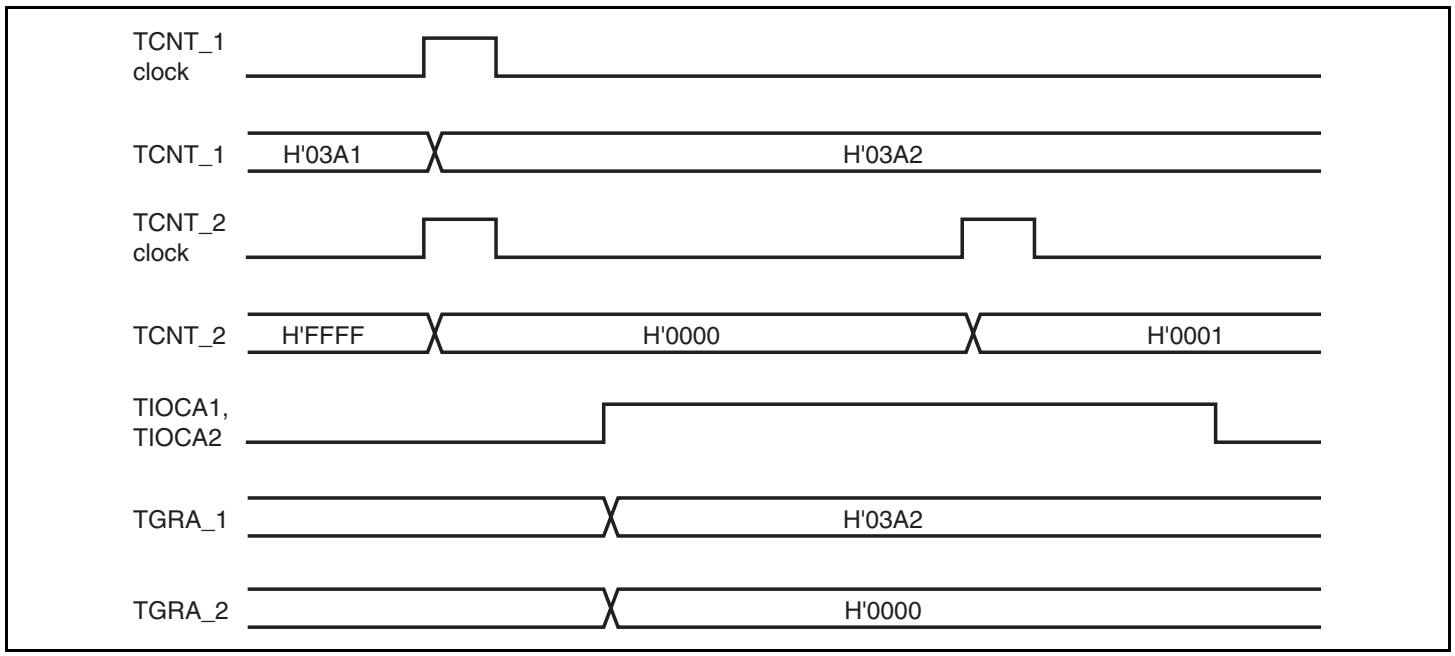


Figure 11.19 Example of Cascaded Operation (1)

Figure 11.20 illustrates the operation when counting upon TCNT_2 overflow/underflow has been set for TCNT_1, and phase counting mode has been designated for channel 2.

TCNT_1 is incremented by TCNT_2 overflow and decremented by TCNT_2 underflow.

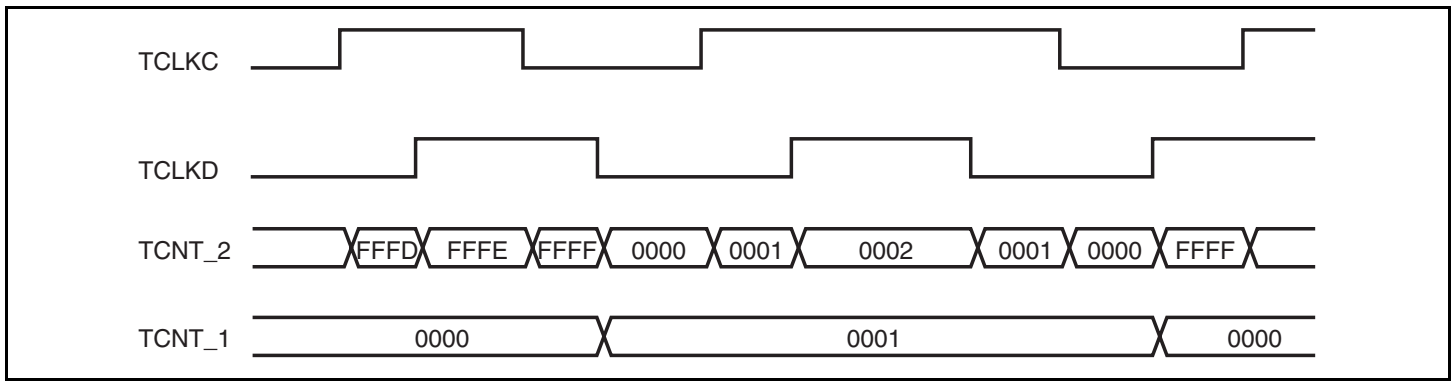


Figure 11.20 Example of Cascaded Operation (2)

11.4.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0, 1, or toggle output can be selected as the output level in response to compare match of each TGR.

Settings of TGR registers can output a PWM waveform in the range of 0–% to 100–% duty cycle.

Designating TGR compare match as the counter clearing source enables the cycle to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The outputs specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR are output from the TIOCA and TIOCC pins at compare matches A and C, respectively. The outputs specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR are output at compare matches B and D, respectively. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.

  In PWM mode 1, a maximum 8-phase PWM output is possible.
- PWM mode 2

  PWM output is generated using one TGR as the cycle register and the others as duty cycle registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a synchronization register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty cycle registers are identical, the output value does not change when a compare match occurs.

  In PWM mode 2, a maximum 15-phase PWM output is possible by combined use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 11.31.

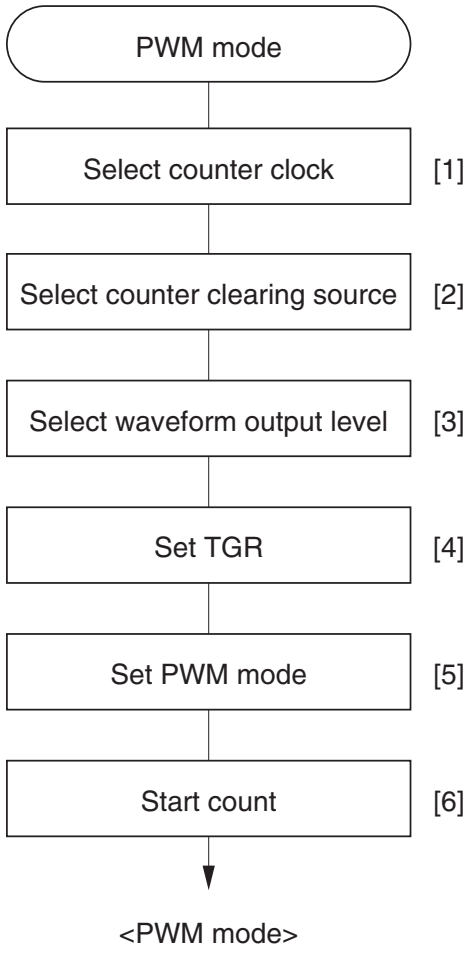
Table 11.31 PWM Output Registers and Output Pins

			Output Pins	
Unit	**Channel**	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	0	TGRA_0	TIOCA0	TIOCA0
		TGRB_0		TIOCB0
		TGRC_0	TIOCC0	TIOCC0
		TGRD_0		TIOCD0
	1	TGRA_1	TIOCA1	TIOCA1
		TGRB_1		TIOCB1
	2	TGRA_2	TIOCA2	TIOCA2
		TGRB_2		TIOCB2
	3	TGRA_3	TIOCA3	TIOCA3
		TGRB_3		TIOCB3
		TGRC_3	TIOCC3	TIOCC3
		TGRD_3		TIOCD3
	4	TGRA_4	TIOCA4	TIOCA4
		TGRB_4		TIOCB4
	5	TGRA_5	TIOCA5	TIOCA5
		TGRB_5		TIOCB5
1	6	TGRA_6	TIOCA6	TIOCA6
		TGRB_6		TIOCB6
		TGRC_6	TIOCC6	TIOCC6
		TGRD_6		TIOCD6
	7	TGRA_7	TIOCA7	TIOCA7
		TGRB_7		TIOCB7
	8	TGRA_8	TIOCA8	TIOCA8
		TGRB_8		TIOCB8
	9	TGRA_9	TIOCA9	TIOCA9
		TGRB_9		TIOCB9
		TGRC_9	TIOCC9	TIOCC9
		TGRD_9		TIOCD9
	10	TGRA_10	TIOCA10	TIOCA10
		TGRB_10		TIOCB10
	11	TGRA_11	TIOCA11	TIOCA11
		TGRB_11		TIOCB11

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the cycle is set.

(1) Example of PWM Mode Setting Procedure

Figure 11.21 shows an example of the PWM mode setting procedure.



[1] Select the counter clock with bits TPSC2 to TPSC0 in TCR. At the same time, select the input clock edge with bits CKEG1 and CKEG0 in TCR.

[2] Use bits CCLR2 to CCLR0 in TCR to select the TGR to be used as the TCNT clearing source.

[3] Use TIOR to designate the TGR as an output compare register, and select the initial value and output value.

[4] Set the cycle in the TGR selected in [2], and set the duty in the other TGRs.

[5] Select the PWM mode with bits MD3 to MD0 in TMDR.

[6] Set the CST bit in TSTR to 1 to start the count operation.

Figure 11.21 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 11.22 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 TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the cycle, and the values set in TGRB registers as the duty cycle.

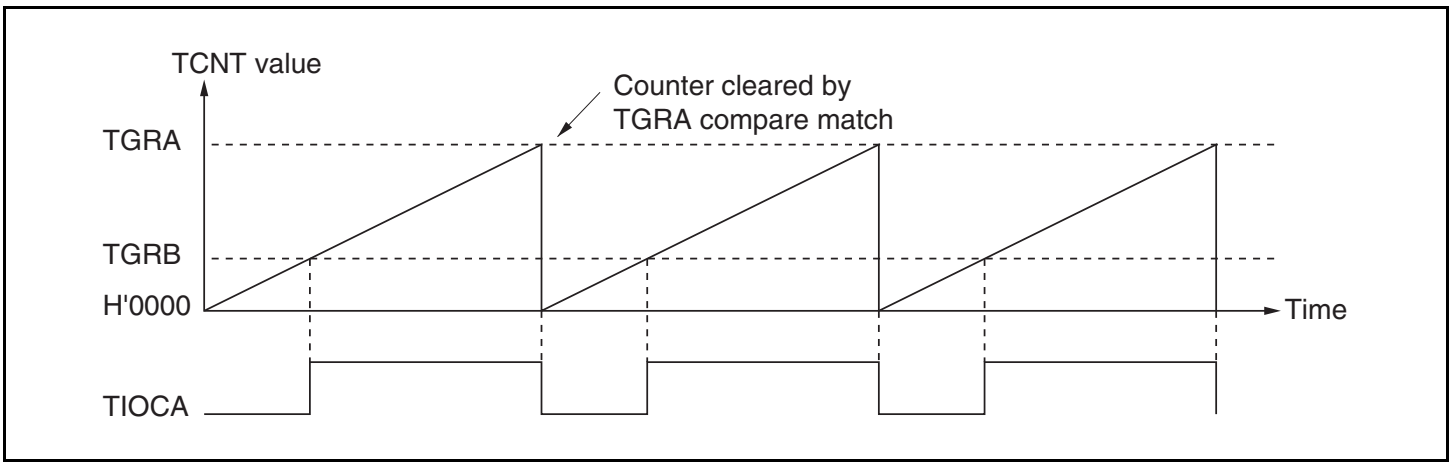


Figure 11.22 Example of PWM Mode Operation (1)

Figure 11.23 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), to output a 5-phase PWM waveform.

In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other TGRs as the duty cycle.

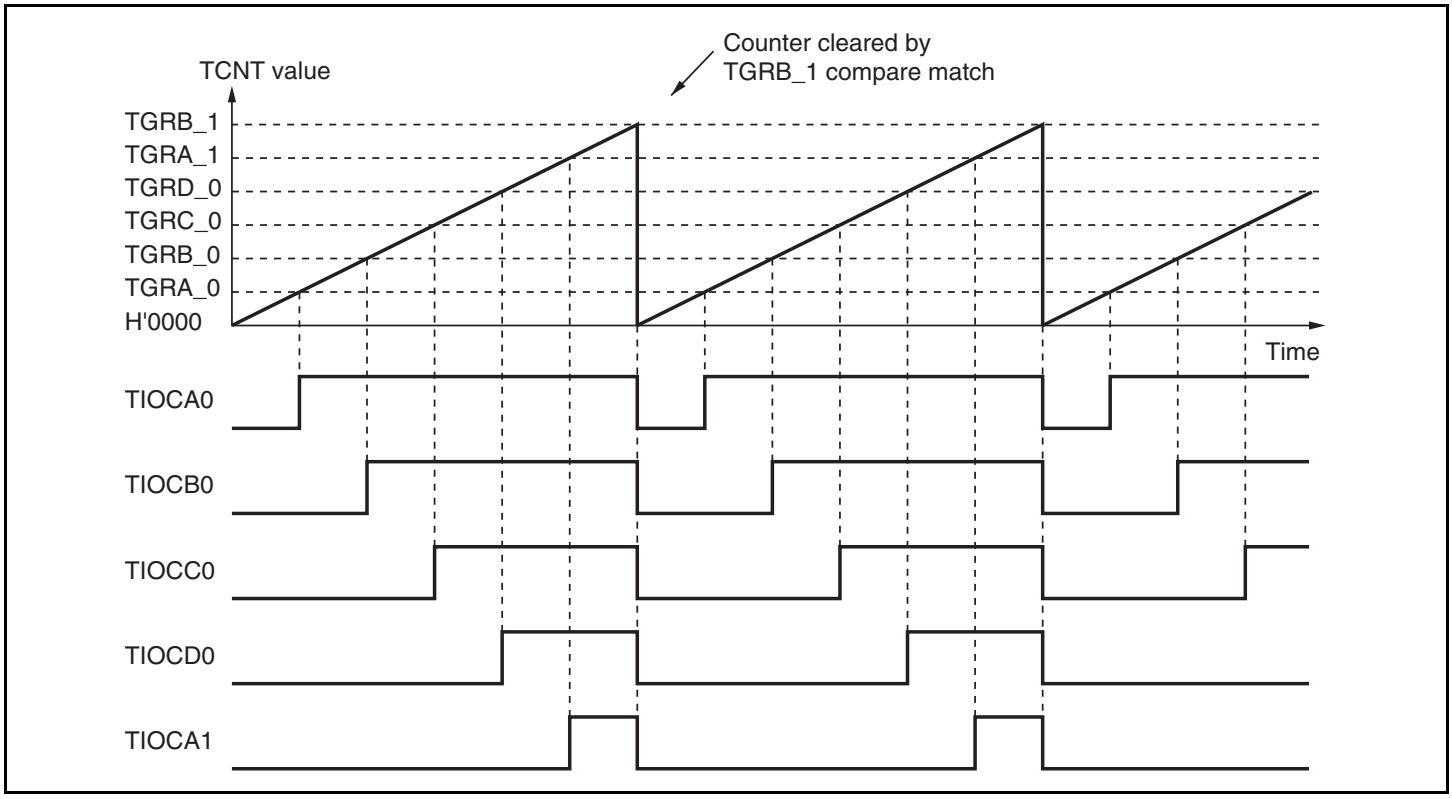


Figure 11.23 Example of PWM Mode Operation (2)

Figure 11.24 shows examples of PWM waveform output with 0% duty cycle and 100% duty cycle in PWM mode.

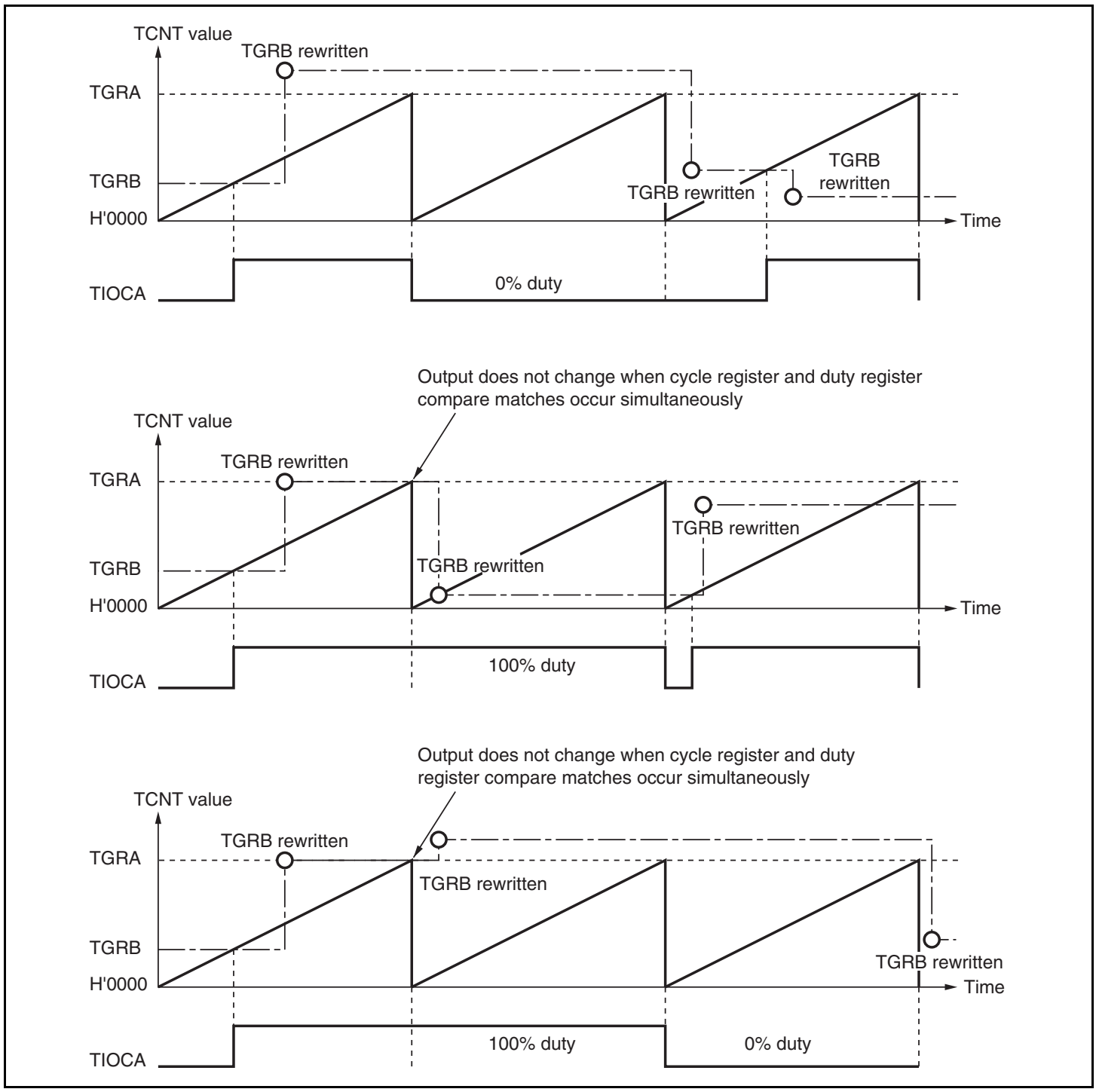


Figure 11.24 Example of PWM Mode Operation (3)

11.4.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1, 2, 4, 5, 7, 8, 10, and 11.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, the functions of bits CCLR1 and CCLR0 in TCR, and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.

This can be used for two-phase encoder pulse input.

When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when underflow occurs while TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.

Table 11.32 shows the correspondence between external clock pins and channels.

Table 11.32 Clock Input Pins in Phase Counting Mode

		External Clock Pins	
Unit	**Channels**	**A-Phase**	**B-Phase**
0	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
1	When channel 7 or 11 is set to phase counting mode	TCLKE	TCLKF
	When channel 8 or 10 is set to phase counting mode	TCLKG	TCLKH

(1) Example of Phase Counting Mode Setting Procedure

Figure 11.25 shows an example of the phase counting mode setting procedure.

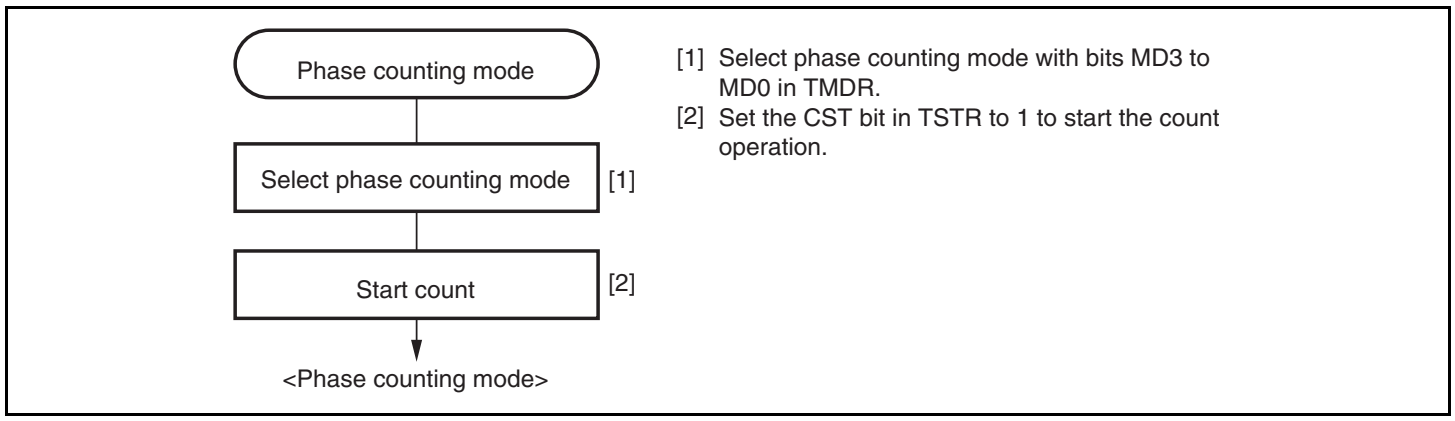


Figure 11.25 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

a. Phase counting mode 1

Figure 11.26 shows an example of phase counting mode 1 operation, and table 11.33 summarizes the TCNT up/down-count conditions.

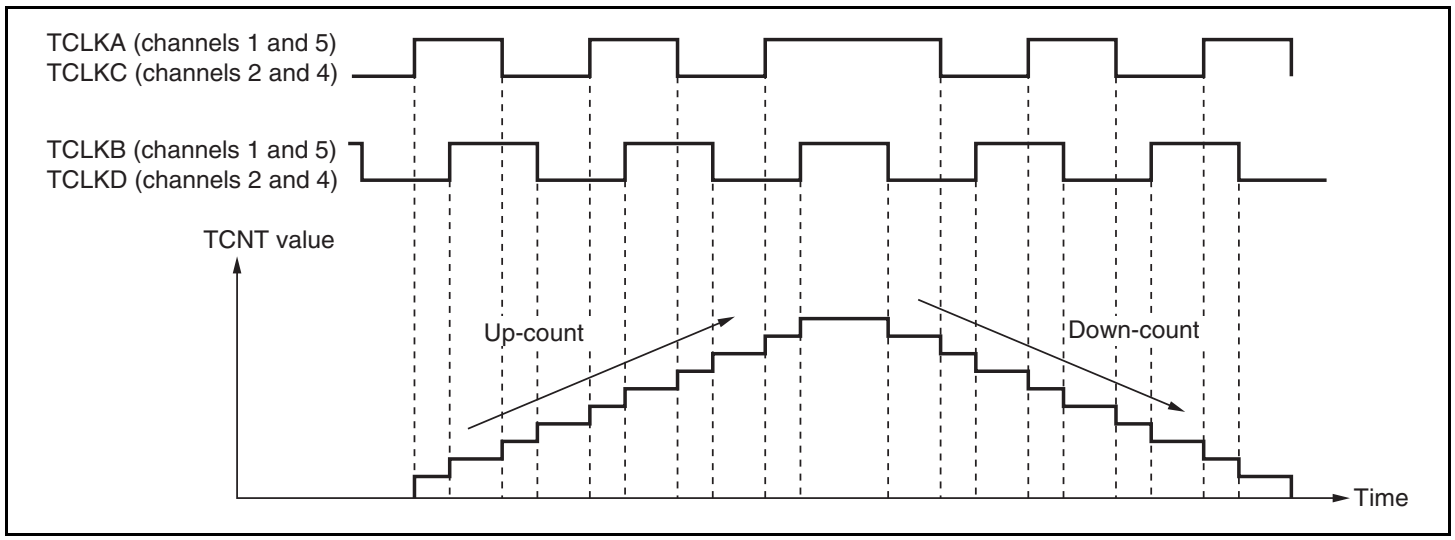


Figure 11.26 Example of Phase Counting Mode 1 Operation

Table 11.33 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4) TCLKE (Channels 7 and 11) TCLKG (Channels 8 and 10)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4) TCLKF (Channels 7 and 11) TCLKH (Channels 8 and 10)	Operation
High level		Up-count
Low level		
	Low level	
	High level	
High level		Down-count
Low level		
	High level	
	Low level	

Legend:

 : Rising edge

 : Falling edge

b. Phase counting mode 2

Figure 11.27 shows an example of phase counting mode 2 operation, and table 11.34 summarizes the TCNT up/down-count conditions.

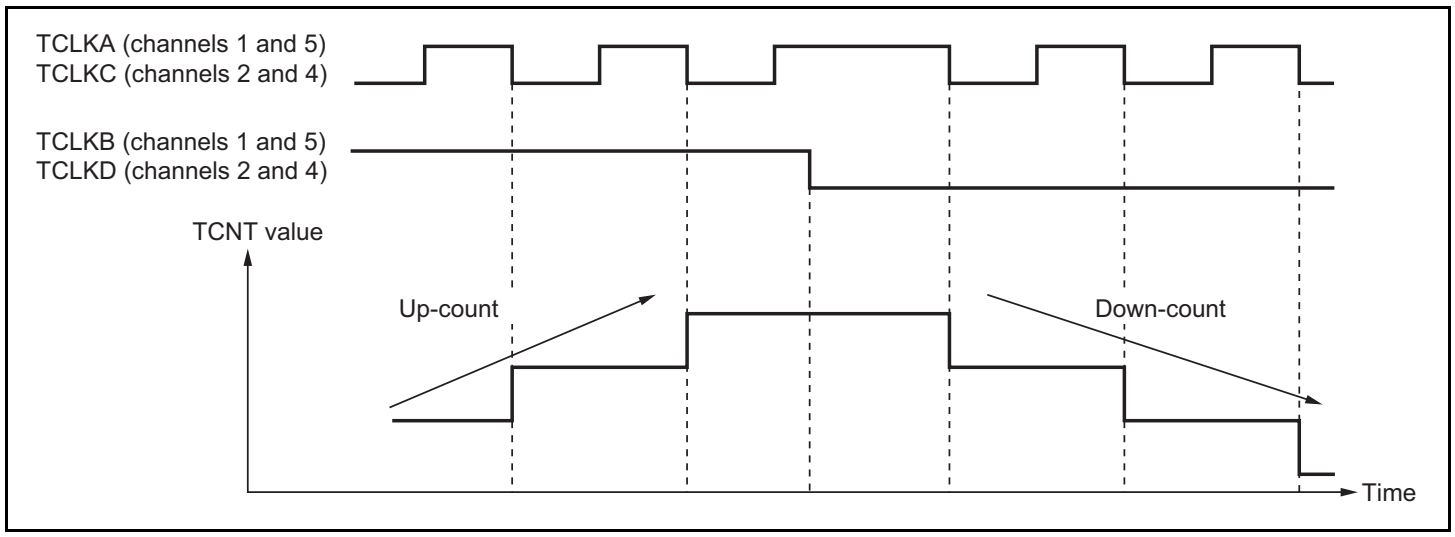


Figure 11.27 Example of Phase Counting Mode 2 Operation

Table 11.34 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4) TCLKE (Channels 7 and 11) TCLKG (Channels 8 and 10)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4) TCLKF (Channels 7 and 11) TCLKH (Channels 8 and 10)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

Legend:

↑ : Rising edge

↓ : Falling edge

c. Phase counting mode 3

Figure 11.28 shows an example of phase counting mode 3 operation, and table 11.35 summarizes the TCNT up/down-count conditions.

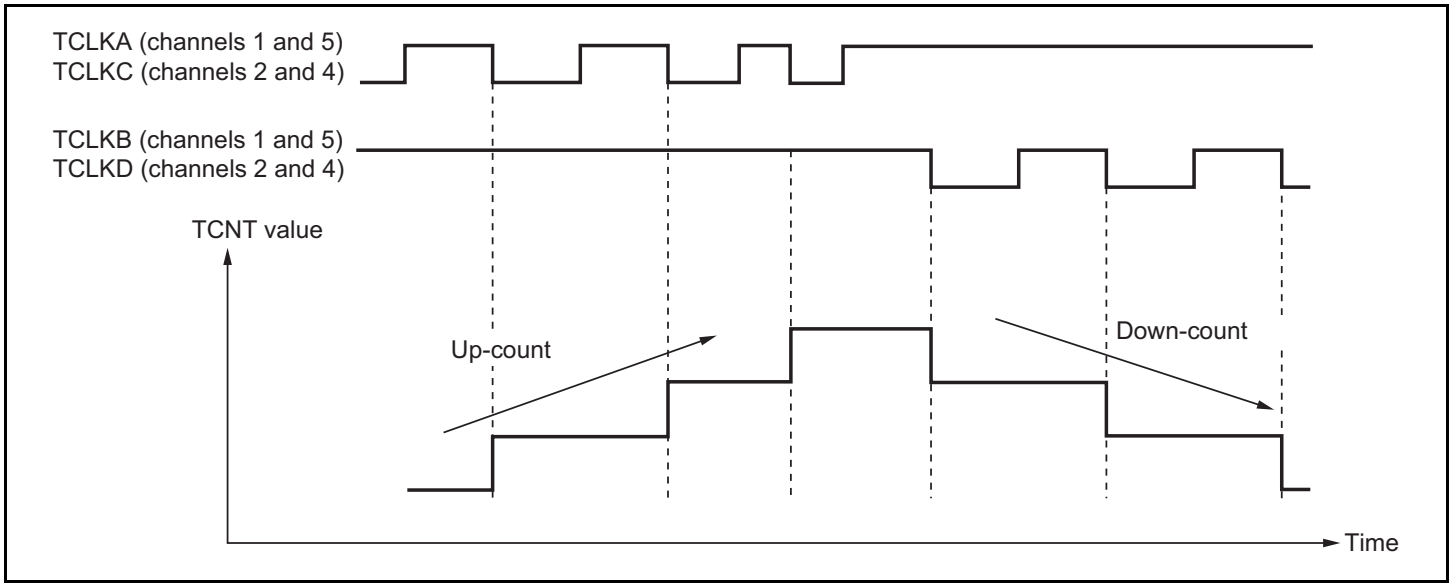


Figure 11.28 Example of Phase Counting Mode 3 Operation

Table 11.35 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4) TCLKE (Channels 7 and 11) TCLKG (Channels 8 and 10)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4) TCLKF (Channels 7 and 11) TCLKH (Channels 8 and 10)	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

d. Phase counting mode 4

Figure 11.29 shows an example of phase counting mode 4 operation, and table 11.36 summarizes the TCNT up/down-count conditions.

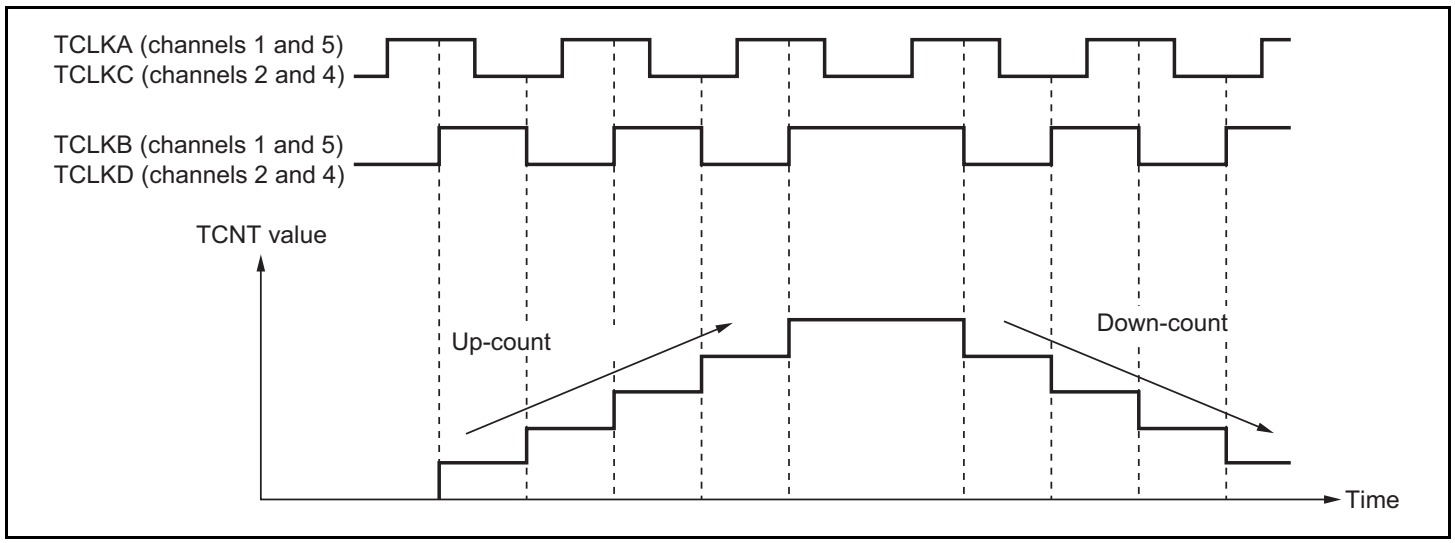


Figure 11.29 Example of Phase Counting Mode 4 Operation

Table 11.36 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4) TCLKE (Channels 7 and 11) TCLKG (Channels 8 and 10)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4) TCLKF (Channels 7 and 11) TCLKH (Channels 8 and 10)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

Legend:
↑ : Rising edge
↓ : Falling edge

(3) Phase Counting Mode Application Example

Figure 11.30 shows an example in which phase counting mode is designated for channel 1, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect the position or speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to TCLKA and TCLKB.

Channel 0 operates with TCNT counter clearing by TGRC_0 compare match; TGRA_0 and TGRC_0 are used for the compare match function, and are set with the speed control cycle and position control cycle. TGRB_0 is used for input capture, with TGRB_0 and TGRD_0 operating in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input capture source, and detection of the pulse width of 2-phase encoder 4-multiplication pulses is performed.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, channel 0 TGRA_0 and TGRC_0 compare matches are selected as the input capture source, and the up/down-counter values for the control cycles are stored.

This procedure enables accurate position/speed detection to be achieved.

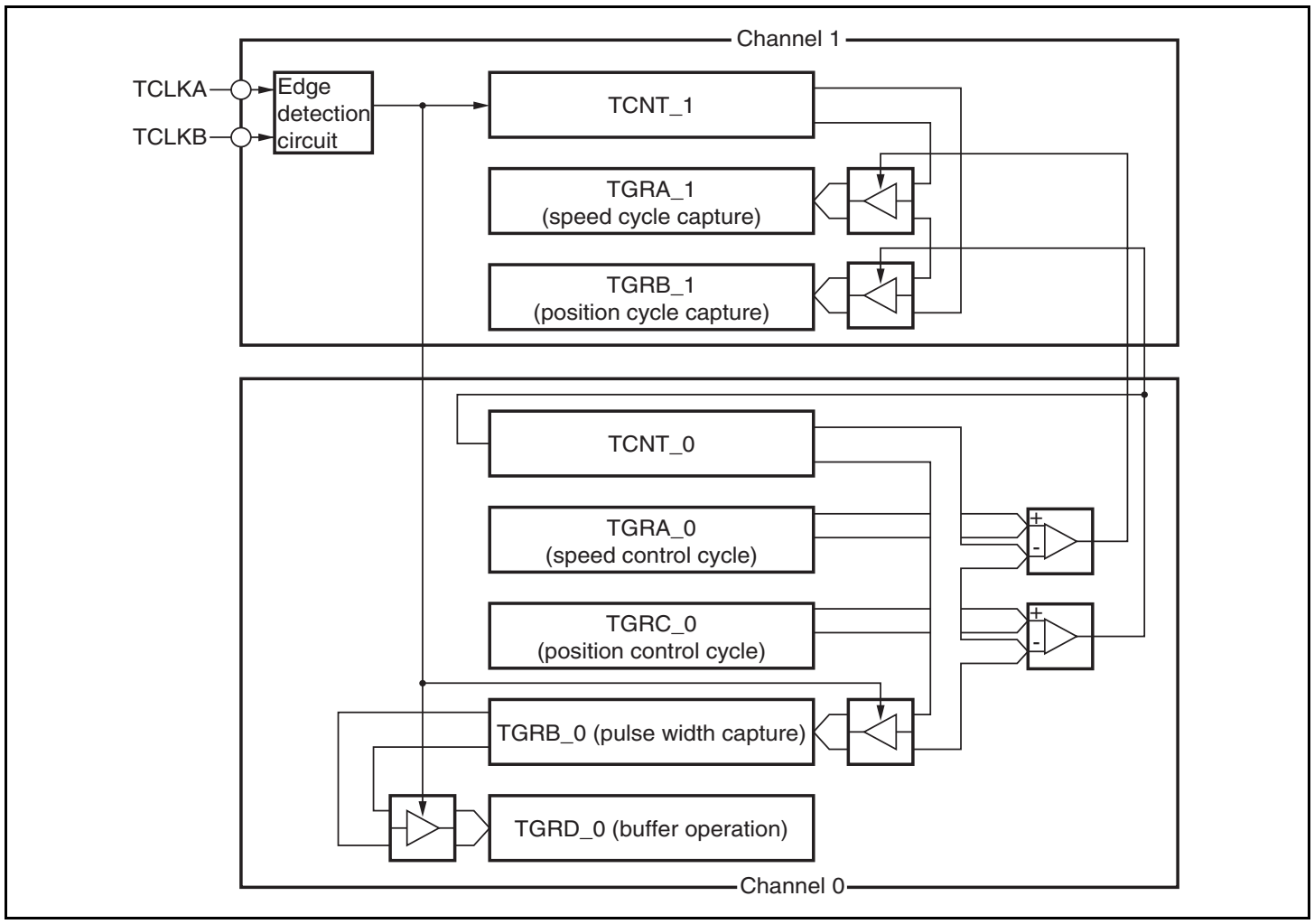


Figure 11.30 Phase Counting Mode Application Example

11.5 Interrupt Sources

There are three kinds of TPU interrupt source: TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disable bit, allowing generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, but the priority order within a channel is fixed. For details, see section 5, Interrupt Controller.

Table 11.37 lists the TPU interrupt sources.

Table 11.37 TPU Interrupts

Unit	Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
0	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
	3	TGI3A	TGRA_3 input capture/compare match	TGFA_3	Possible	Possible
		TGI3B	TGRB_3 input capture/compare match	TGFB_3	Possible	Not possible
		TGI3C	TGRC_3 input capture/compare match	TGFC_3	Possible	Not possible
		TGI3D	TGRD_3 input capture/compare match	TGFD_3	Possible	Not possible
		TCI3V	TCNT_3 overflow	TCFV_3	Not possible	Not possible
	4	TGI4A	TGRA_4 input capture/compare match	TGFA_4	Possible	Possible
		TGI4B	TGRB_4 input capture/compare match	TGFB_4	Possible	Not possible
		TCI4V	TCNT_4 overflow	TCFV_4	Not possible	Not possible
		TCI4U	TCNT_4 underflow	TCFU_4	Not possible	Not possible
	5	TGI5A	TGRA_5 input capture/compare match	TGFA_5	Possible	Possible
		TGI5B	TGRB_5 input capture/compare match	TGFB_5	Possible	Not possible
		TCI5V	TCNT_5 overflow	TCFV_5	Not possible	Not possible
		TCI5U	TCNT_5 underflow	TCFU_5	Not possible	Not possible

Unit	Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
1	6	TGI6A	TGRA_6 input capture/compare match	TGFA_6	Possible	Not possible
		TGI6B	TGRB_6 input capture/compare match	TGFB_6	Possible	Not possible
		TGI6C	TGRC_6 input capture/compare match	TGFC_6	Possible	Not possible
		TGI6D	TGRD_6 input capture/compare match	TGFD_6	Possible	Not possible
		TCI6V	TCNT_6 overflow	TCFV_6	Not possible	Not possible
	7	TGI7A	TGRA_7 input capture/compare match	TGFA_7	Possible	Not possible
		TGI7B	TGRB_7 input capture/compare match	TGFB_7	Possible	Not possible
		TCI7V	TCNT_7 overflow	TCFV_7	Not possible	Not possible
		TCI7U	TCNT_7 underflow	TCFU_7	Not possible	Not possible
	8	TGI8A	TGRA_8 input capture/compare match	TGFA_8	Possible	Not possible
		TGI8B	TGRB_8 input capture/compare match	TGFB_8	Possible	Not possible
		TCI8V	TCNT_8 overflow	TCFV_8	Not possible	Not possible
		TCI8U	TCNT_8 underflow	TCFU_8	Not possible	Not possible
	9	TGI9A	TGRA_9 input capture/compare match	TGFA_9	Possible	Not possible
		TGI9B	TGRB_9 input capture/compare match	TGFB_9	Possible	Not possible
		TGI9C	TGRC_9 input capture/compare match	TGFC_9	Possible	Not possible
		TGI9D	TGRD_9 input capture/compare match	TGFD_9	Possible	Not possible
		TCI9V	TCNT_9 overflow	TCFV_9	Not possible	Not possible
	10	TGI10A	TGRA_10 input capture/compare match	TGFA_10	Possible	Not possible
		TGI10B	TGRB_10 input capture/compare match	TGFB_10	Possible	Not possible
		TCI10V	TCNT_10 overflow	TCFV_10	Not possible	Not possible
		TCI10U	TCNT_10 underflow	TCFU_10	Not possible	Not possible
	11	TGI11A	TGRA_11 input capture/compare match	TGFA_11	Possible	Not possible
		TGI11B	TGRB_11 input capture/compare match	TGFB_11	Possible	Not possible
		TCI11V	TCNT_11 overflow	TCFV_11	Not possible	Not possible
		TCI11U	TCNT_11 underflow	TCFU_11	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 particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has 32 input capture/compare match interrupts, four each for channels 0, 3, 6, and 9, and two each for channels 1, 2, 4, 5, 7, 8, 10, and 11.

(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 TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The TPU has 12 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 TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The TPU has eight underflow interrupts, one each for channels 1, 2, 4, 5, 7, 8, 10, and 11.

11.6 DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt for a channel. For details, see section 9, Data Transfer Controller (DTC).

A total of 32 TPU input capture/compare match interrupts can be used as DTC activation sources, four each for channels 0, 3, 6, and 9, and two each for channels 1, 2, 4, 5, 7, 8, 10, and 11.

11.7 DMAC Activation

In unit 0 of the TPU, the DMAC can be activated by the TGRA input capture/compare match interrupt for a channel. For details, see section 7, DMA Controller (DMAC). (The DMAC cannot be activated by unit 1.)

In unit 0 of the TPU, a total of six TGRA input capture/compare match interrupts can be used as DMAC activation sources, one for each channel.

11.8 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a channel.

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

In the TPU, a total of 12 TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

11.9 Operation Timing

11.9.1 Input/Output Timing

(1) TCNT Count Timing

Figure 11.31 shows TCNT count timing in internal clock operation, and figure 11.32 shows TCNT count timing in external clock operation.

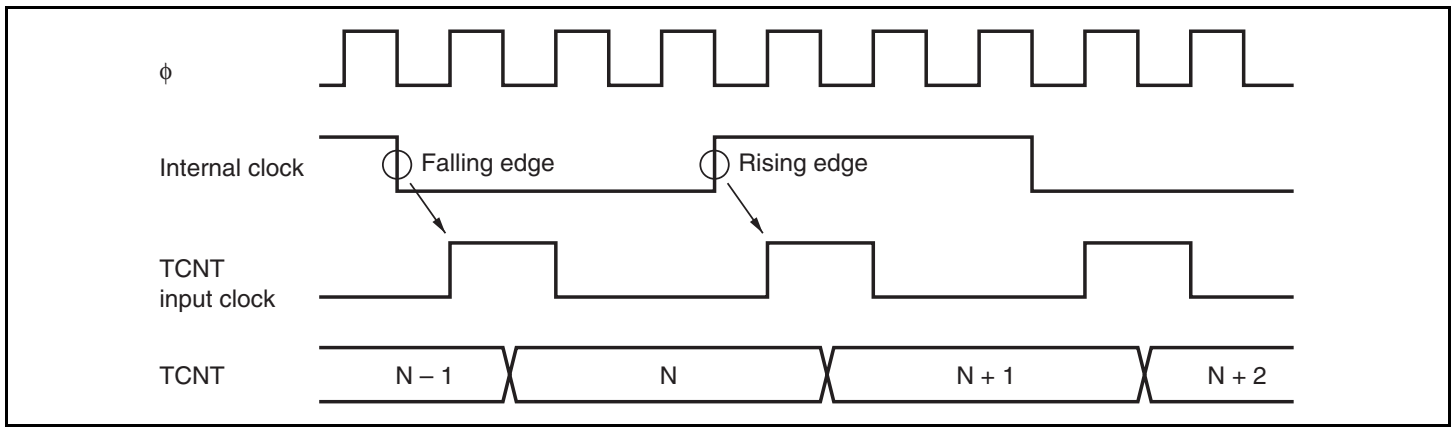


Figure 11.31 Count Timing in Internal Clock Operation

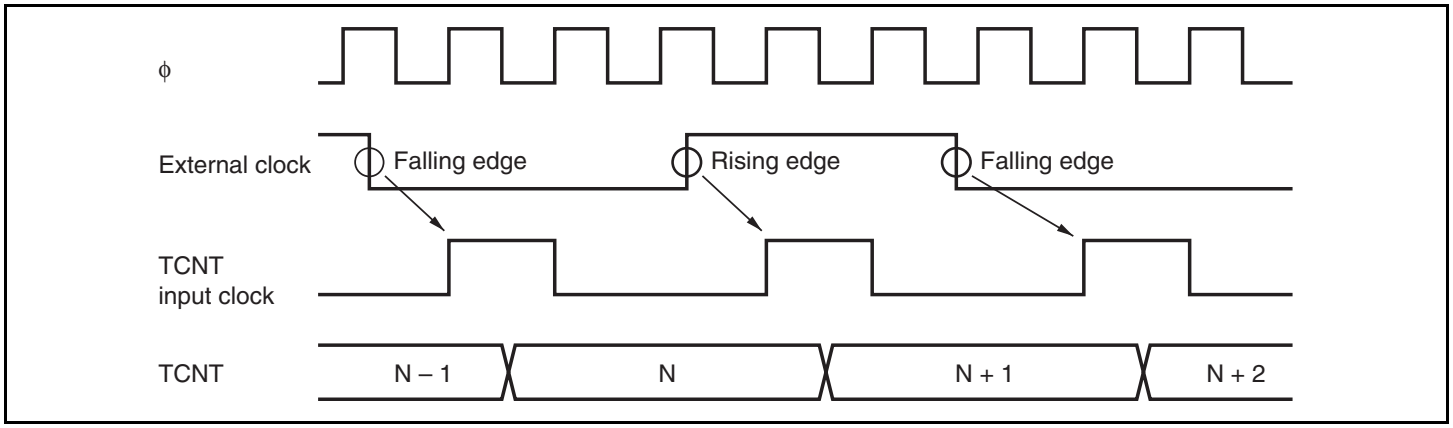


Figure 11.32 Count Timing in External Clock Operation

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin. After a match between TCNT and TGR, the compare match signal is not generated until the (TIOC pin) TCNT input clock is generated.

Figure 11.33 shows output compare output timing.

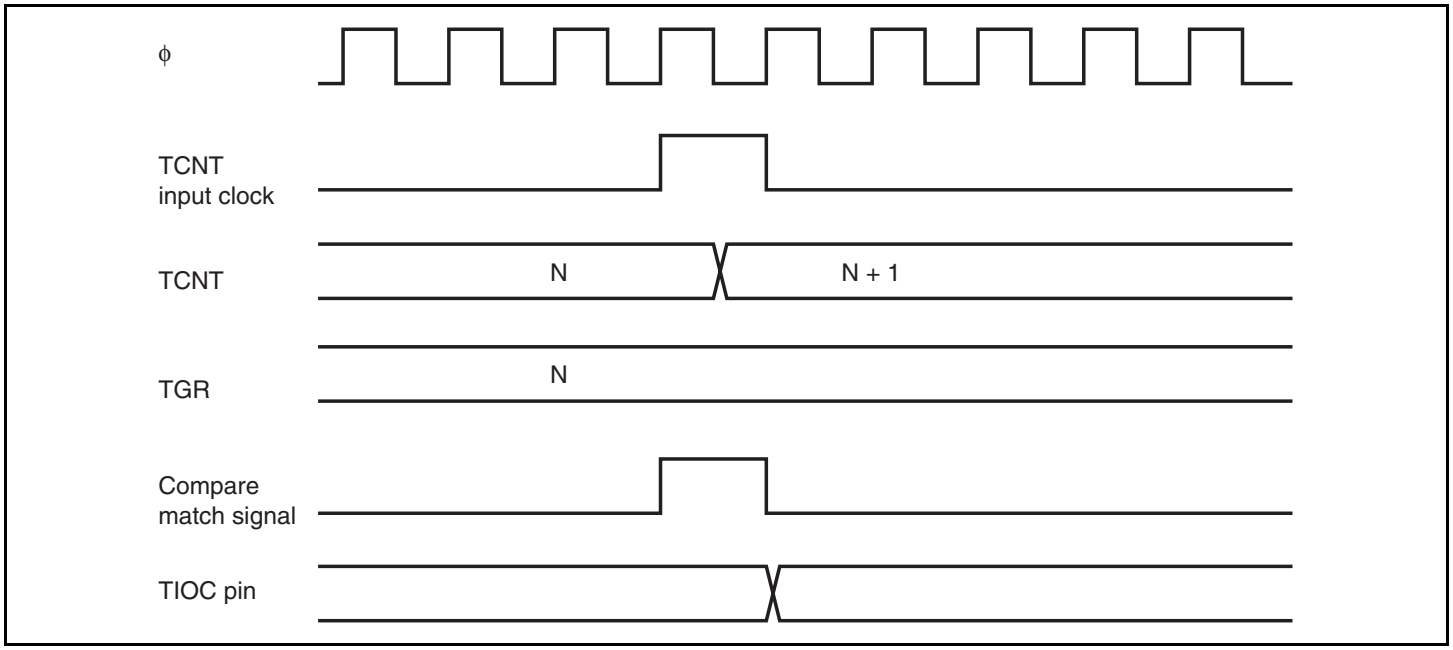


Figure 11.33 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 11.34 shows input capture signal timing.

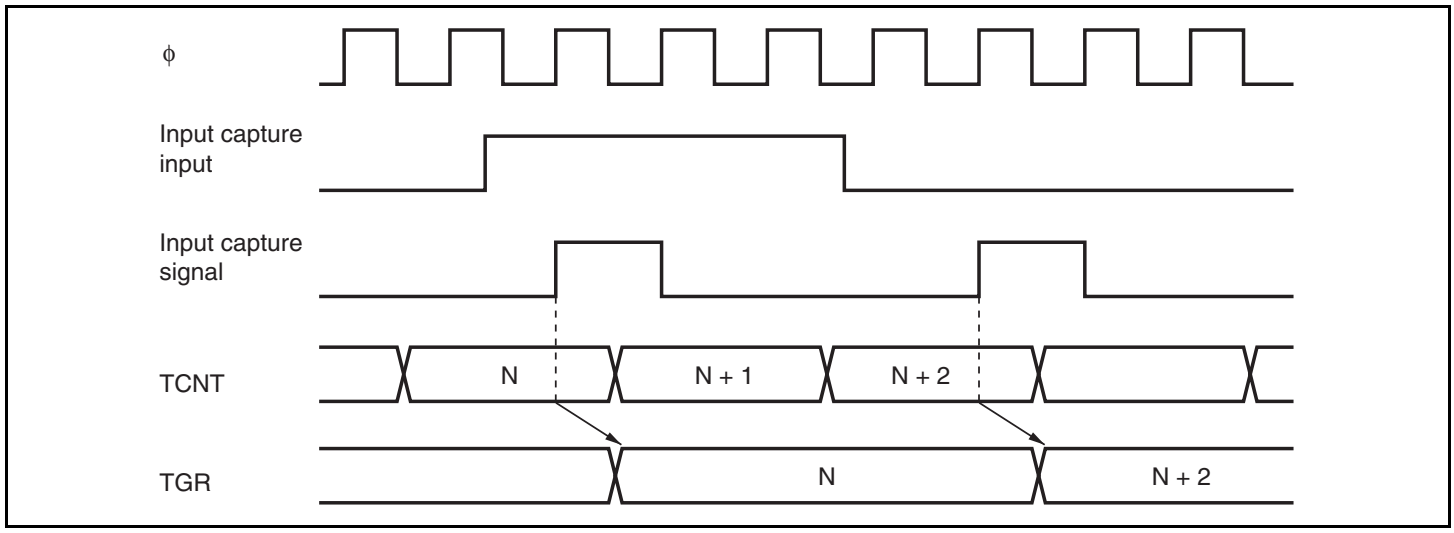


Figure 11.34 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 11.35 shows the timing when counter clearing by compare match occurrence is specified, and figure 11.36 shows the timing when counter clearing by input capture occurrence is specified.

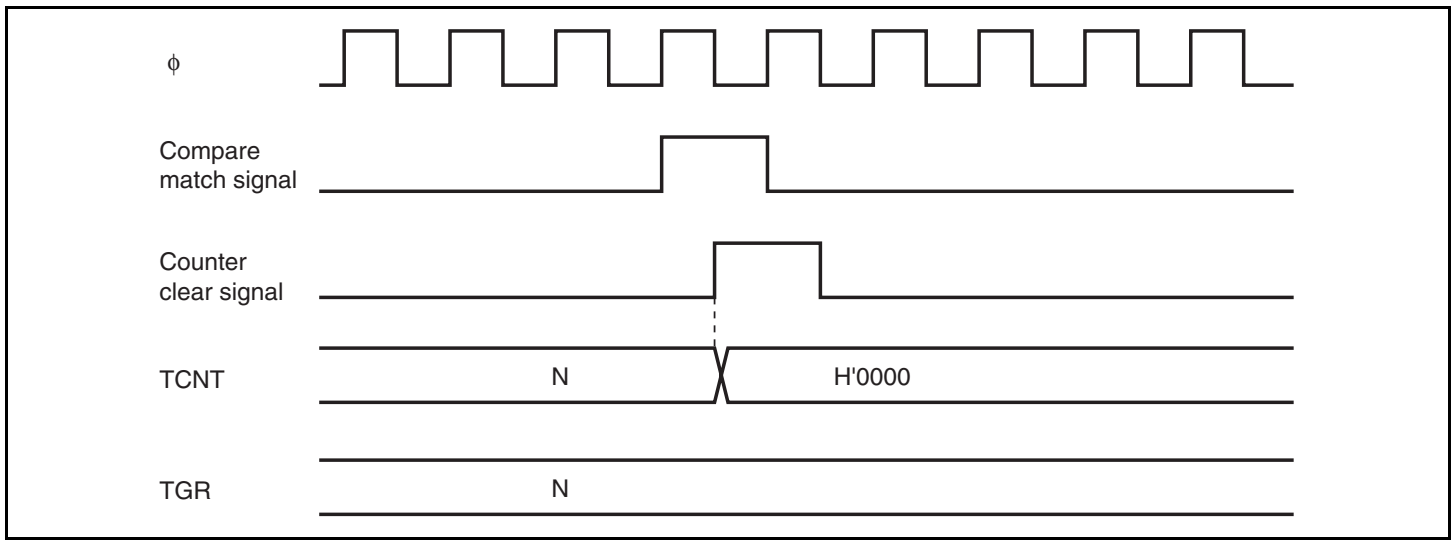


Figure 11.35 Counter Clear Timing (Compare Match)

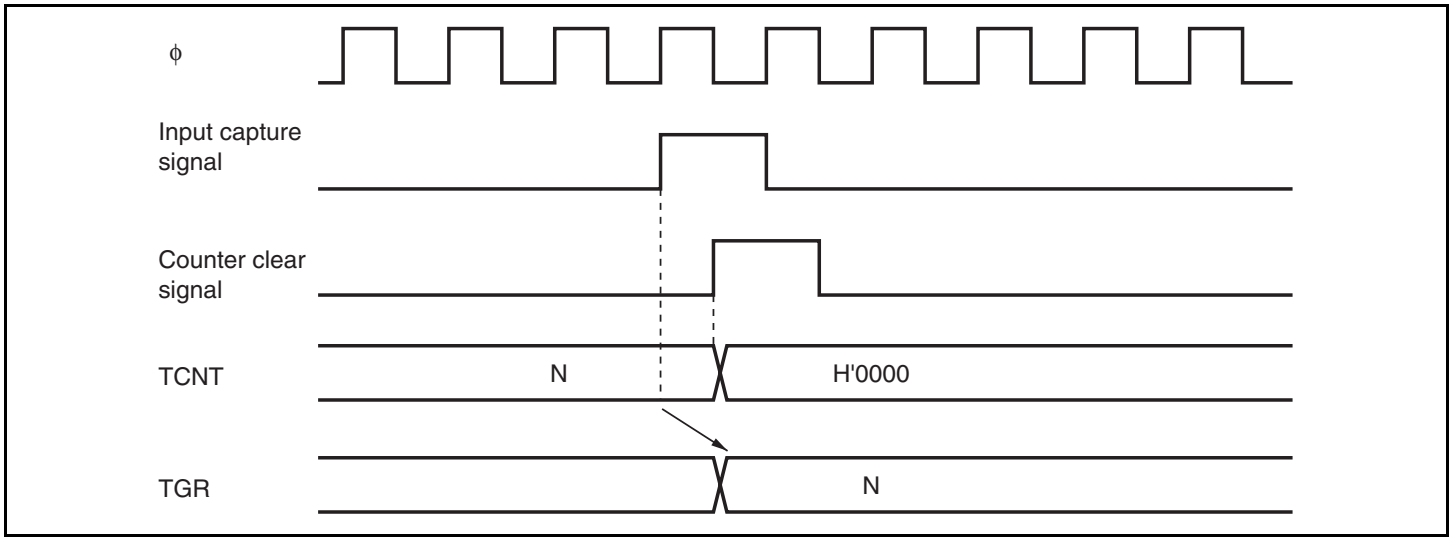


Figure 11.36 Counter Clear Timing (Input Capture)

(5) Buffer Operation Timing

Figures 11.37 and 11.38 show the timings in buffer operation.

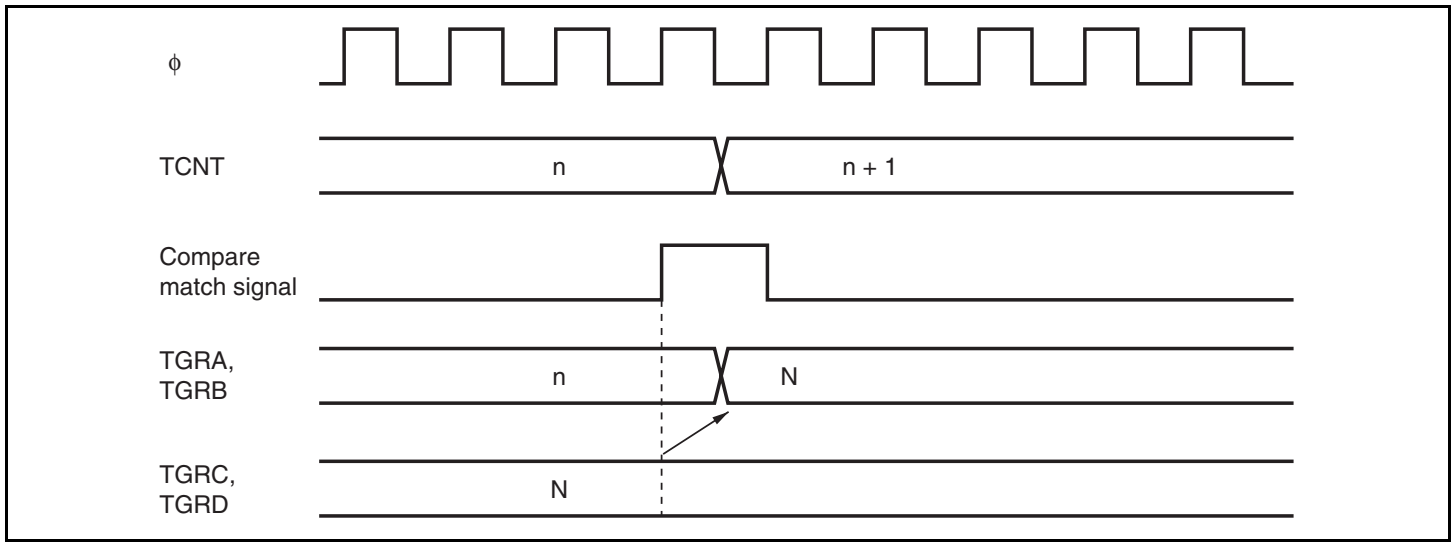


Figure 11.37 Buffer Operation Timing (Compare Match)

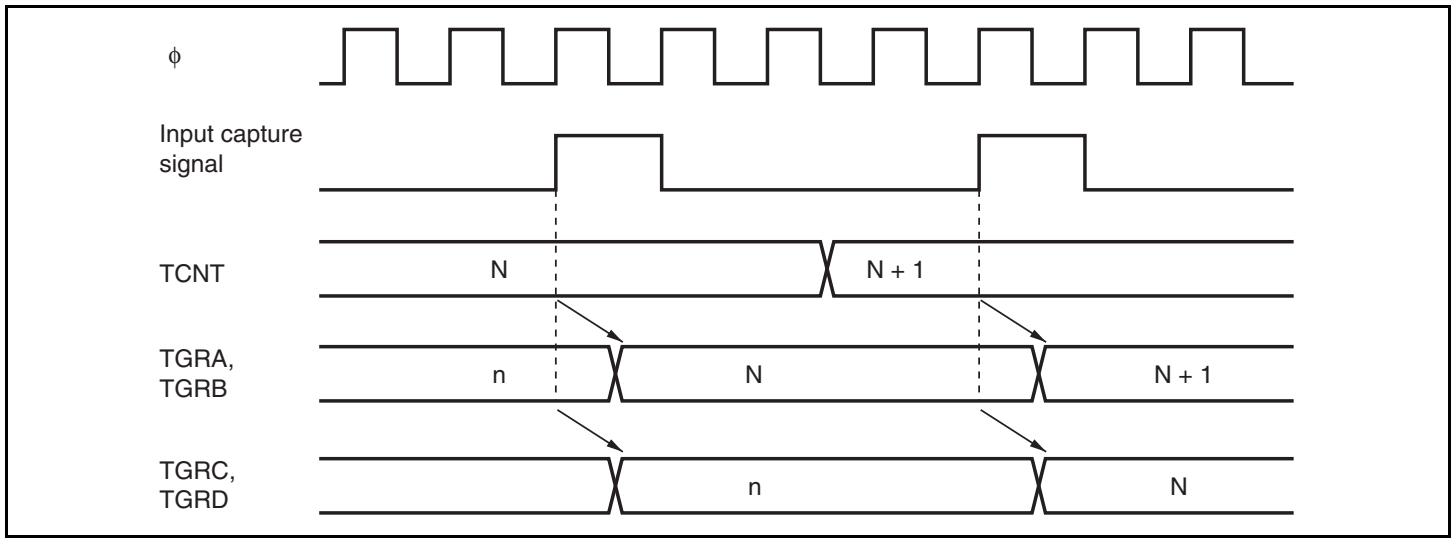


Figure 11.38 Buffer Operation Timing (Input Capture)

11.9.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figure 11.39 shows the timing for setting of the TGF flag in TSR by compare match occurrence, and the TGI interrupt request signal timing.

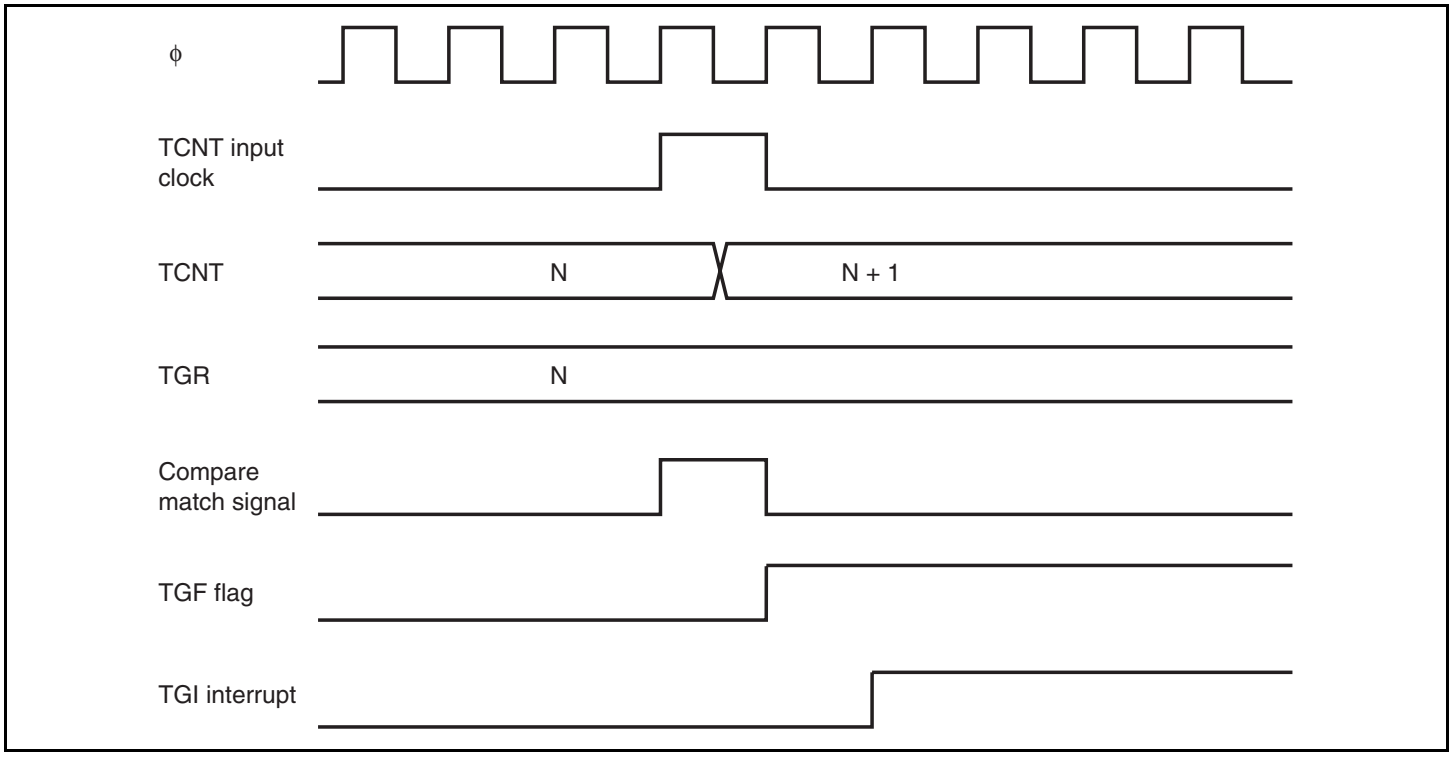


Figure 11.39 TGI Interrupt Timing (Compare Match)

(2) TGF Flag Setting Timing in Case of Input Capture

Figure 11.40 shows the timing for setting of the TGF flag in TSR by input capture occurrence, and the TGI interrupt request signal timing.

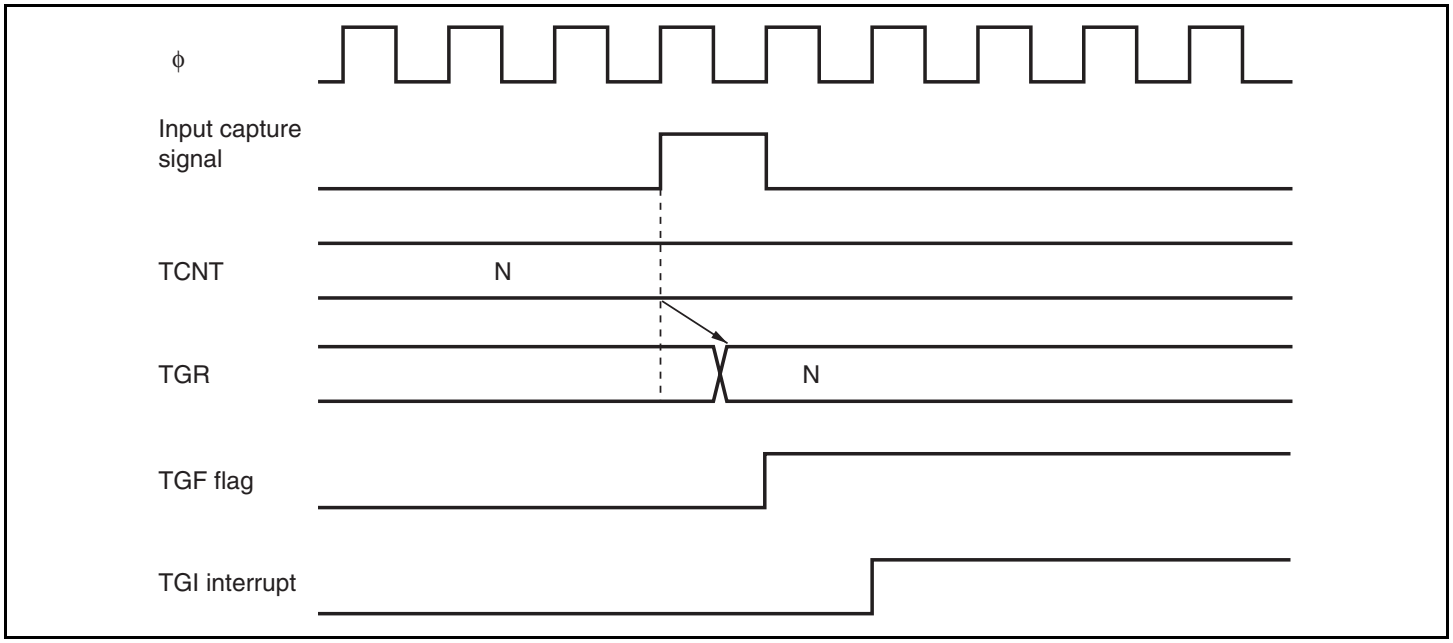


Figure 11.40 TGI Interrupt Timing (Input Capture)

(3) TCFV Flag/TCFU Flag Setting Timing

Figure 11.41 shows the timing for setting of the TCFV flag in TSR by overflow occurrence, and the TCIV interrupt request signal timing.

Figure 11.42 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and the TCIU interrupt request signal timing.

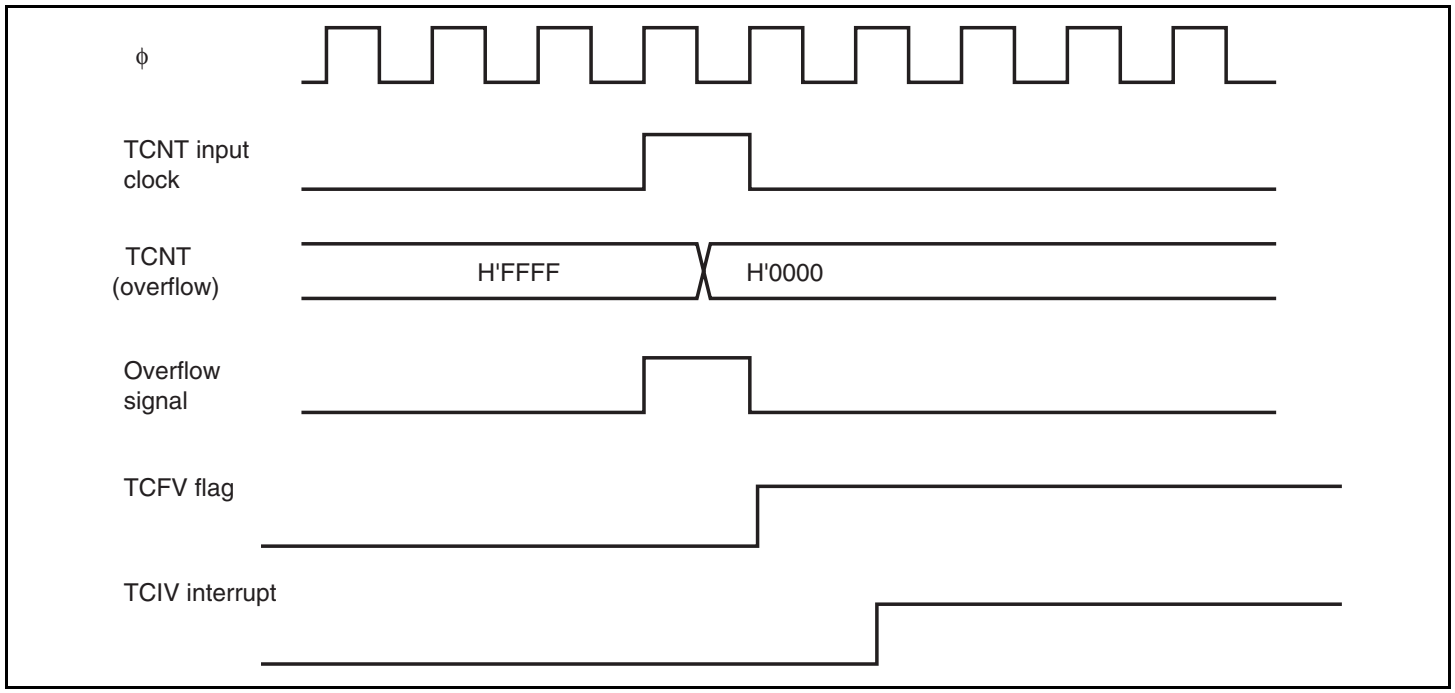


Figure 11.41 TCIV Interrupt Setting Timing

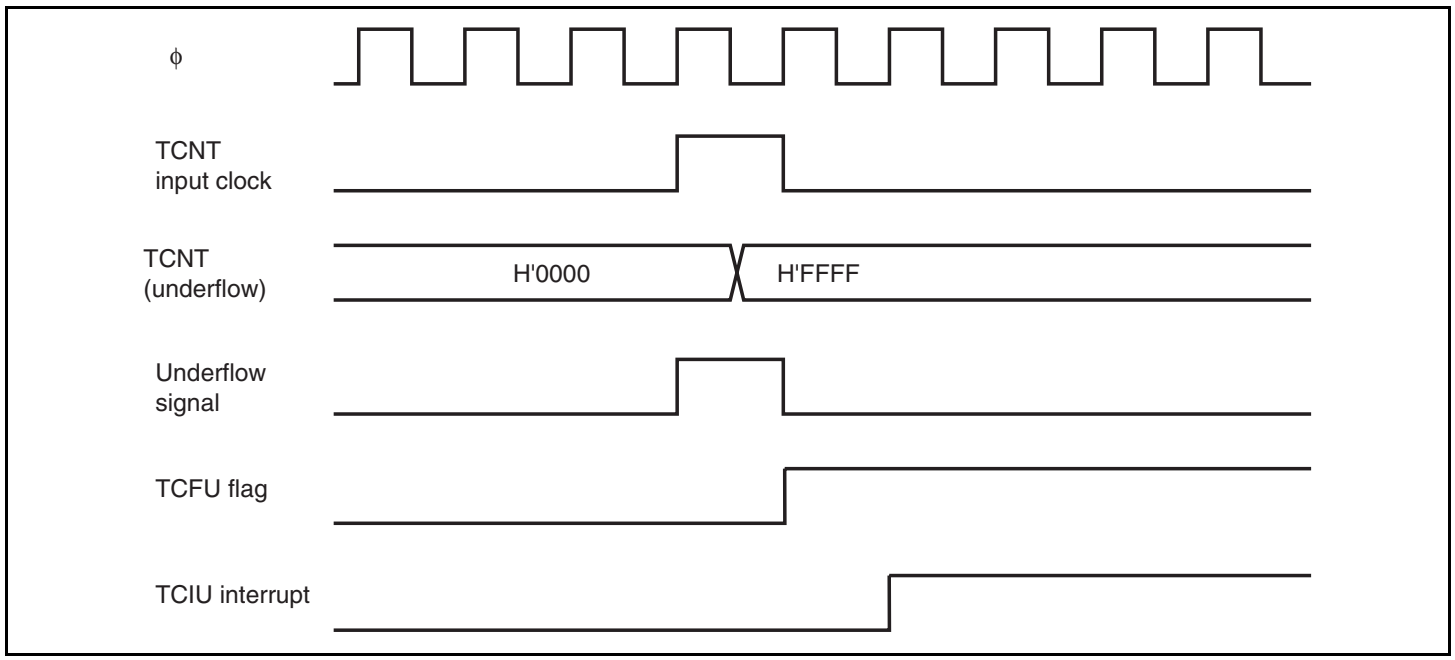


Figure 11.42 TCIU Interrupt Setting Timing

(4) Status Flag Clearing Timing

After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DTC or DMAC is activated, the flag is cleared automatically. Figure 11.43 shows the timing for status flag clearing by the CPU, and figure 11.44 shows the timing for status flag clearing by the DTC or DMAC.

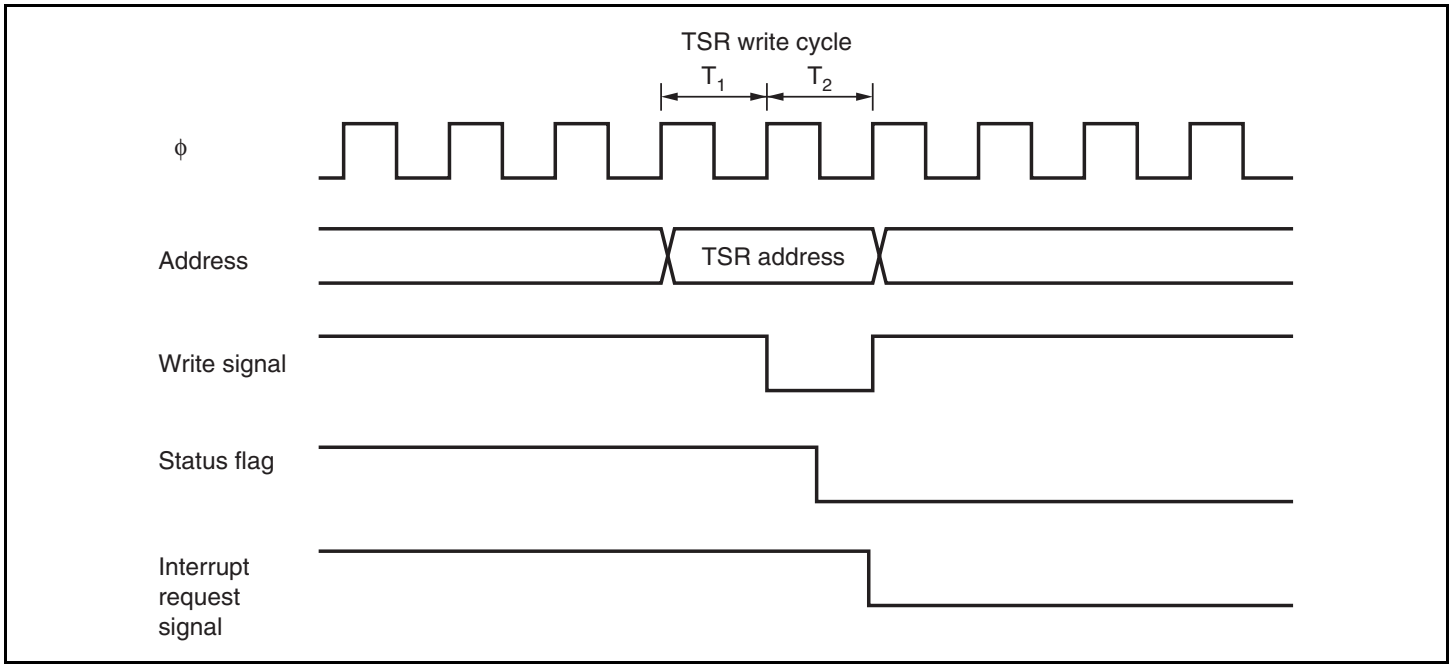


Figure 11.43 Timing for Status Flag Clearing by CPU

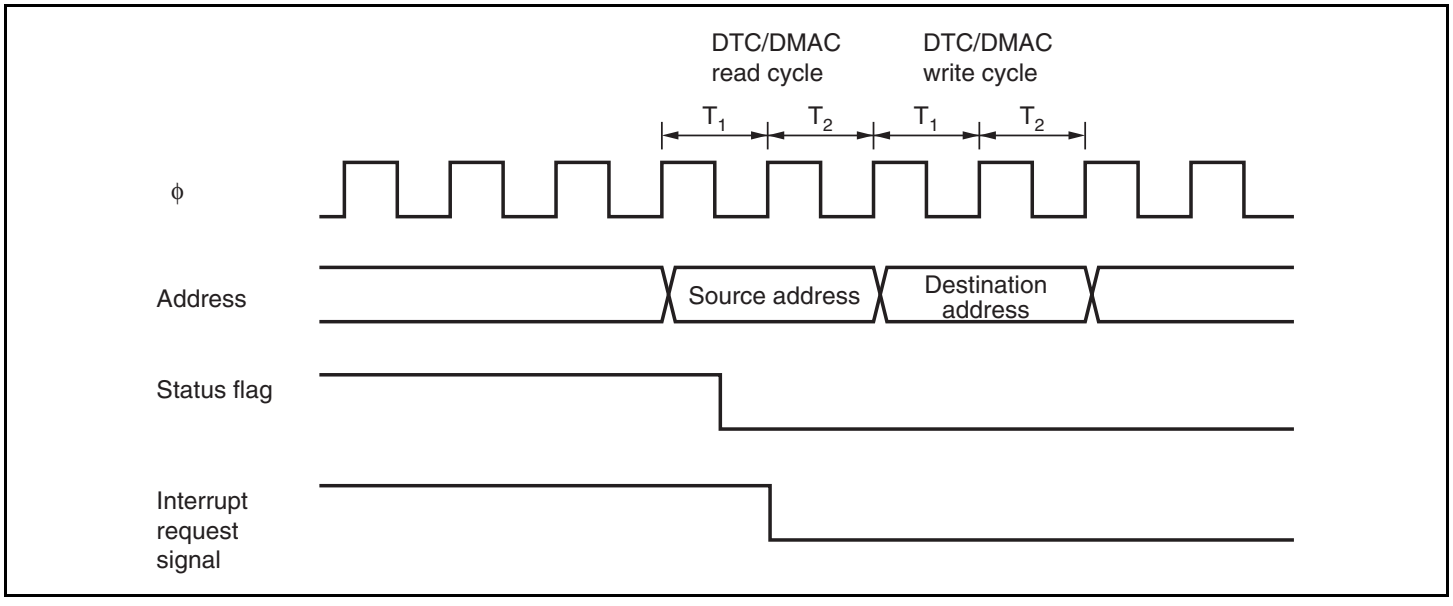


Figure 11.44 Timing for Status Flag Clearing by DTC/DMAC Activation

11.10 Usage Notes

11.10.1 Module Stop Mode Setting

TPU operation can be disabled or enabled using the module stop control register. The initial setting is for TPU operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 25, Power-Down Modes.

11.10.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 11.45 shows the input clock conditions in phase counting mode.

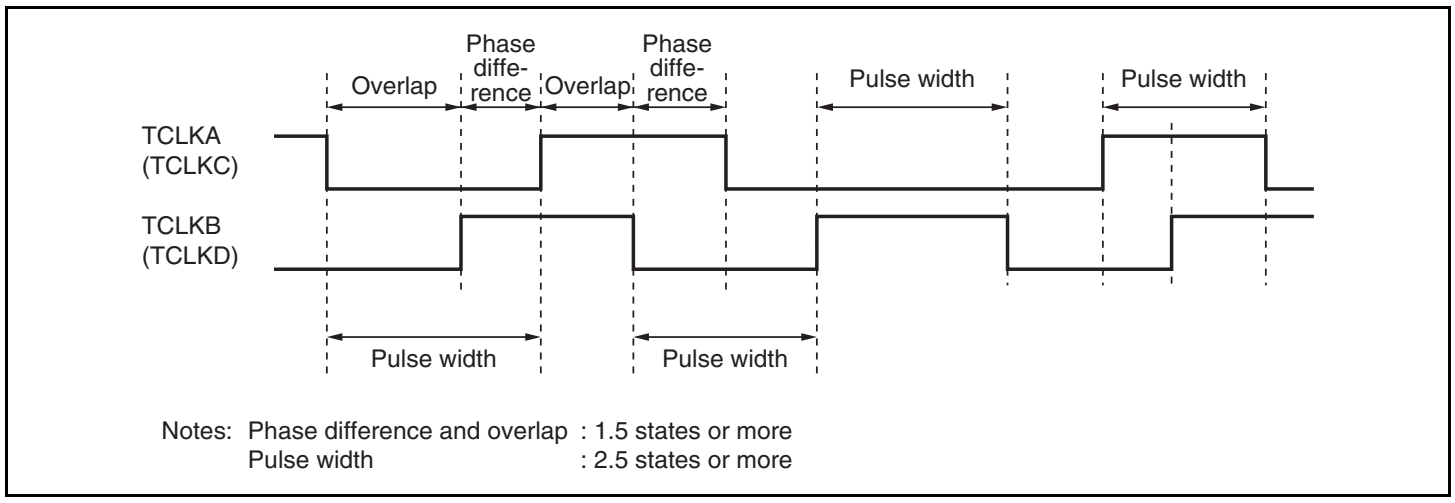


Figure 11.45 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

11.10.3 Caution on Cycle Setting

When counter clearing by compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{\phi}{(N + 1)}$$

Where f: Counter frequency
ϕ: Operating frequency
N: TGR set value

11.10.4 Contention between TCNT Write and Clear Operations

If the counter clearing signal is generated in the T_2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed. Figure 11.46 shows the timing in this case.

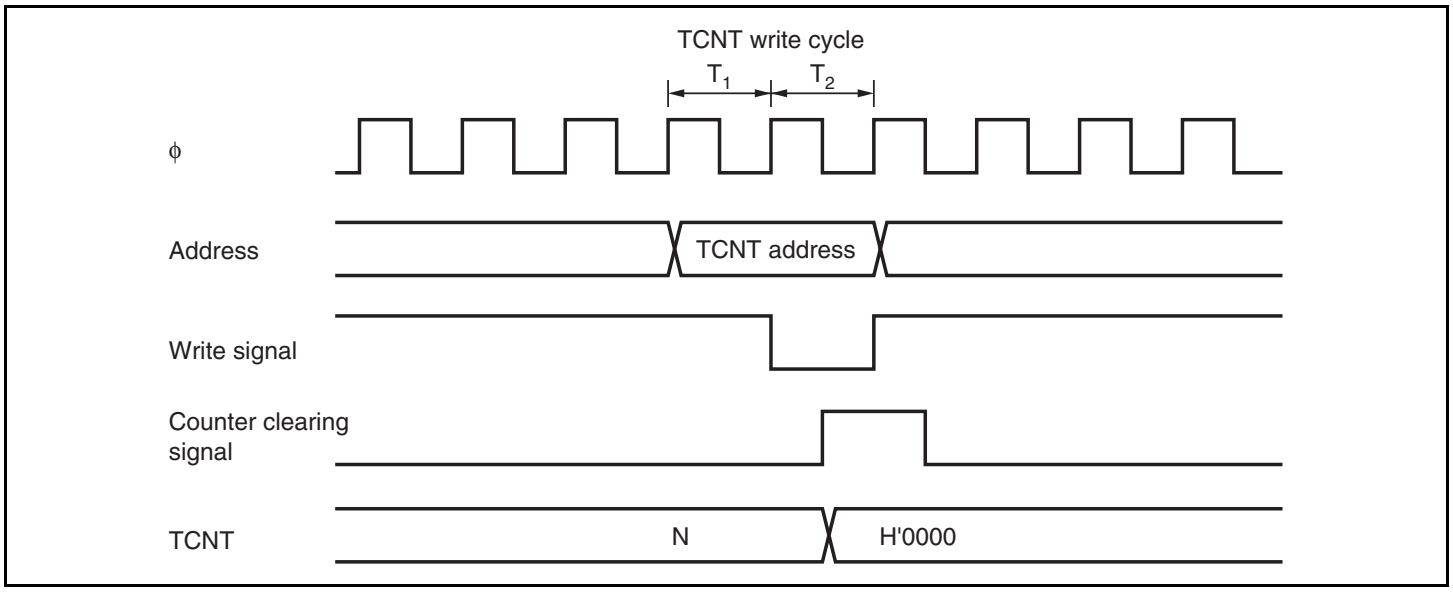


Figure 11.46 Contention between TCNT Write and Clear Operations

11.10.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in the T_2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented. Figure 11.47 shows the timing in this case.

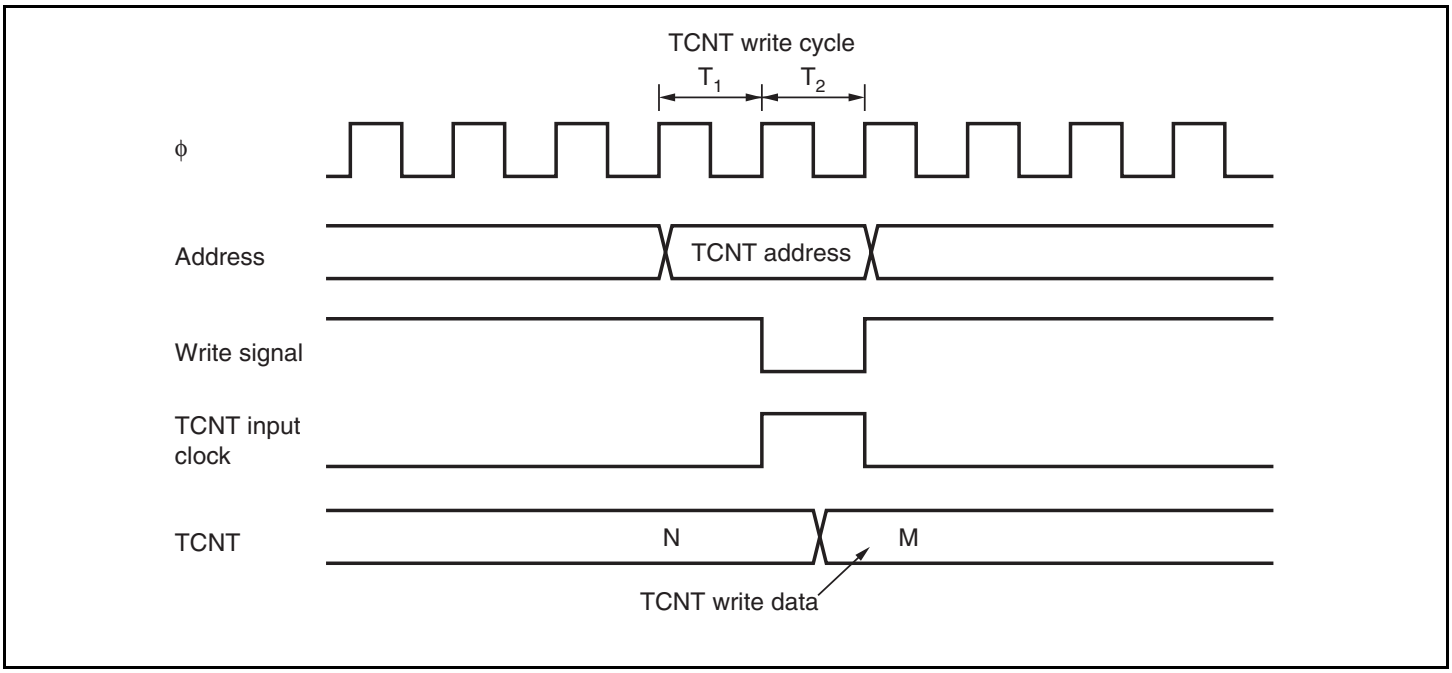


Figure 11.47 Contention between TCNT Write and Increment Operations

11.10.6 Contention between TGR Write and Compare Match

If a compare match occurs in the T_2 state of a TGR write cycle, the TGR write takes precedence and the compare match signal is disabled. A compare match also does not occur when the same value as before is written.

Figure 11.48 shows the timing in this case.

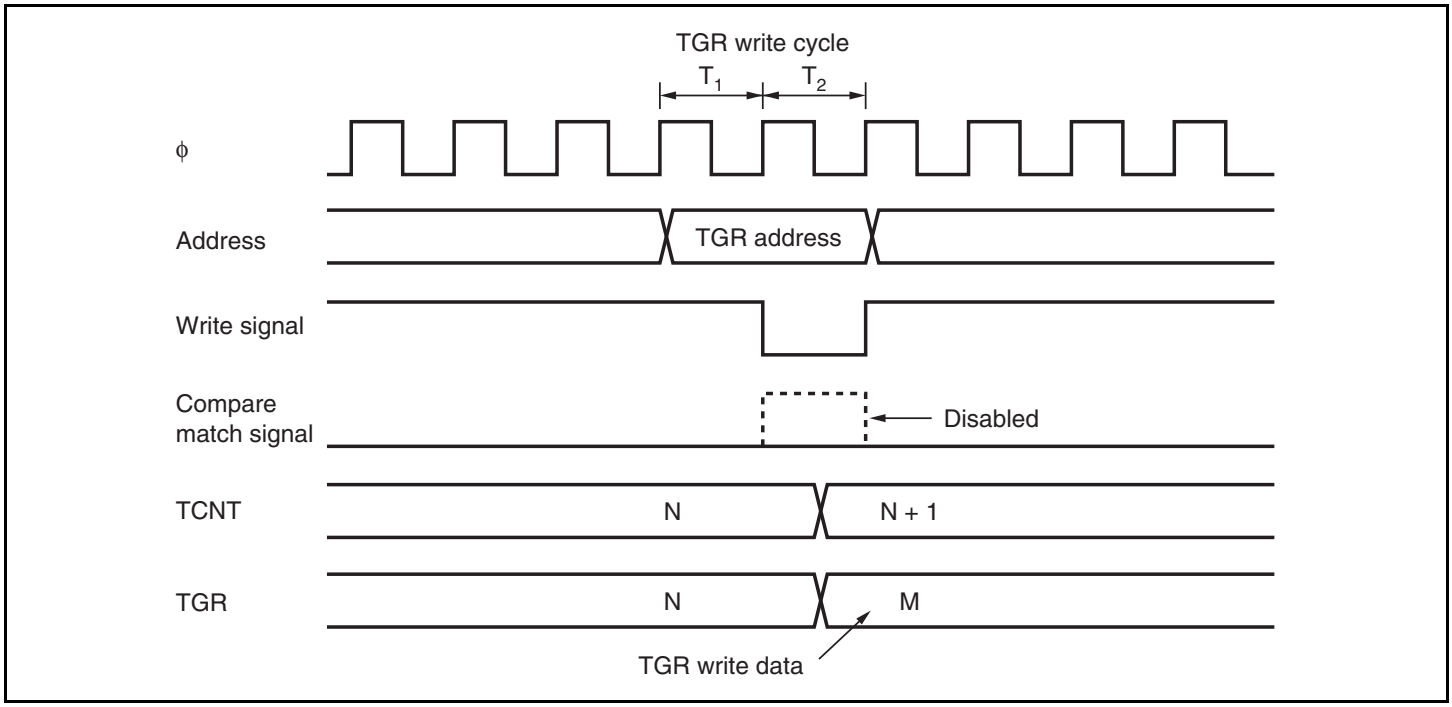


Figure 11.48 Contention between TGR Write and Compare Match

11.10.7 Contention between Buffer Register Write and Compare Match

If a compare match occurs in the T_2 state of a TGR write cycle, the data transferred to TGR by the buffer operation will be the data prior to the write.

Figure 11.49 shows the timing in this case.

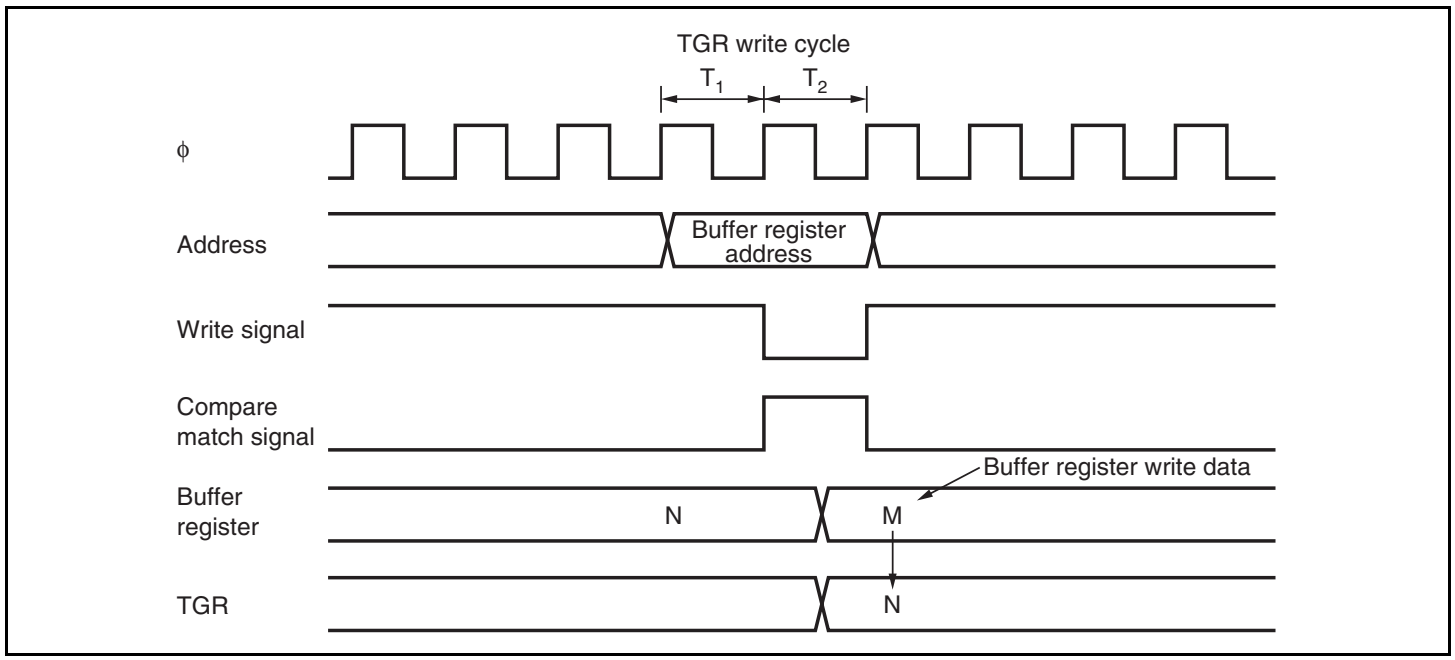


Figure 11.49 Contention between Buffer Register Write and Compare Match

11.10.8 Contention between TGR Read and Input Capture

If the input capture signal is generated in the T_1 state of a TGR read cycle, the data that is read will be the data after input capture transfer.

Figure 11.50 shows the timing in this case.

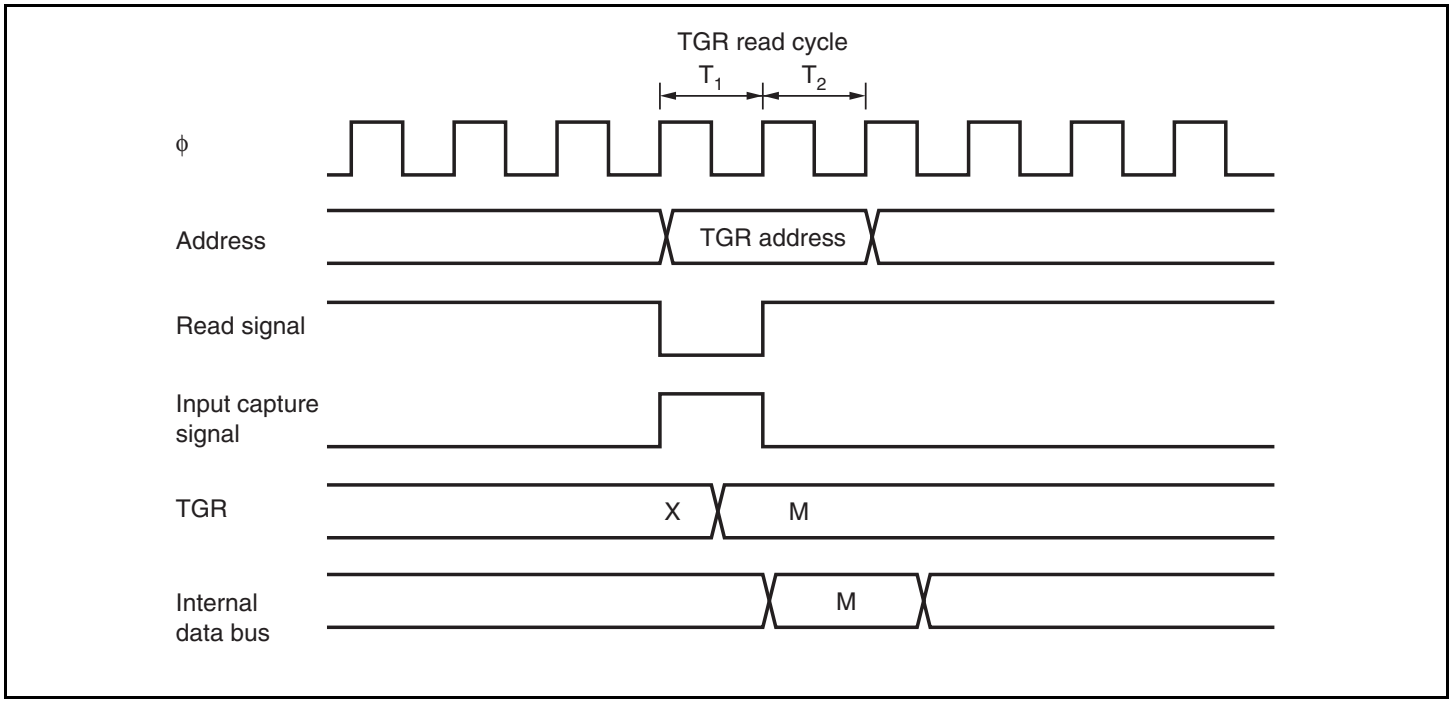


Figure 11.50 Contention between TGR Read and Input Capture

11.10.9 Contention between TGR Write and Input Capture

If the input capture signal is generated in the T_2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed.

Figure 11.51 shows the timing in this case.

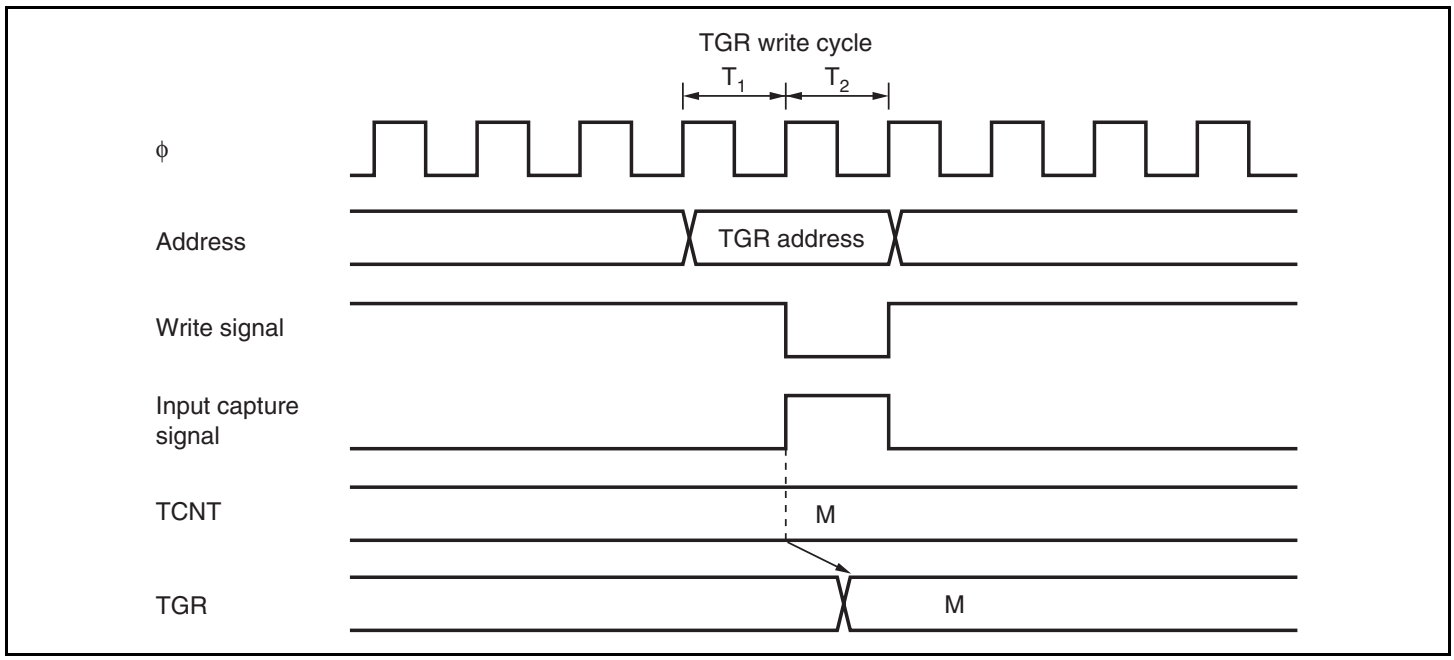


Figure 11.51 Contention between TGR Write and Input Capture

11.10.10 Contention between Buffer Register Write and Input Capture

If the input capture signal is generated in the T_2 state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 11.52 shows the timing in this case.

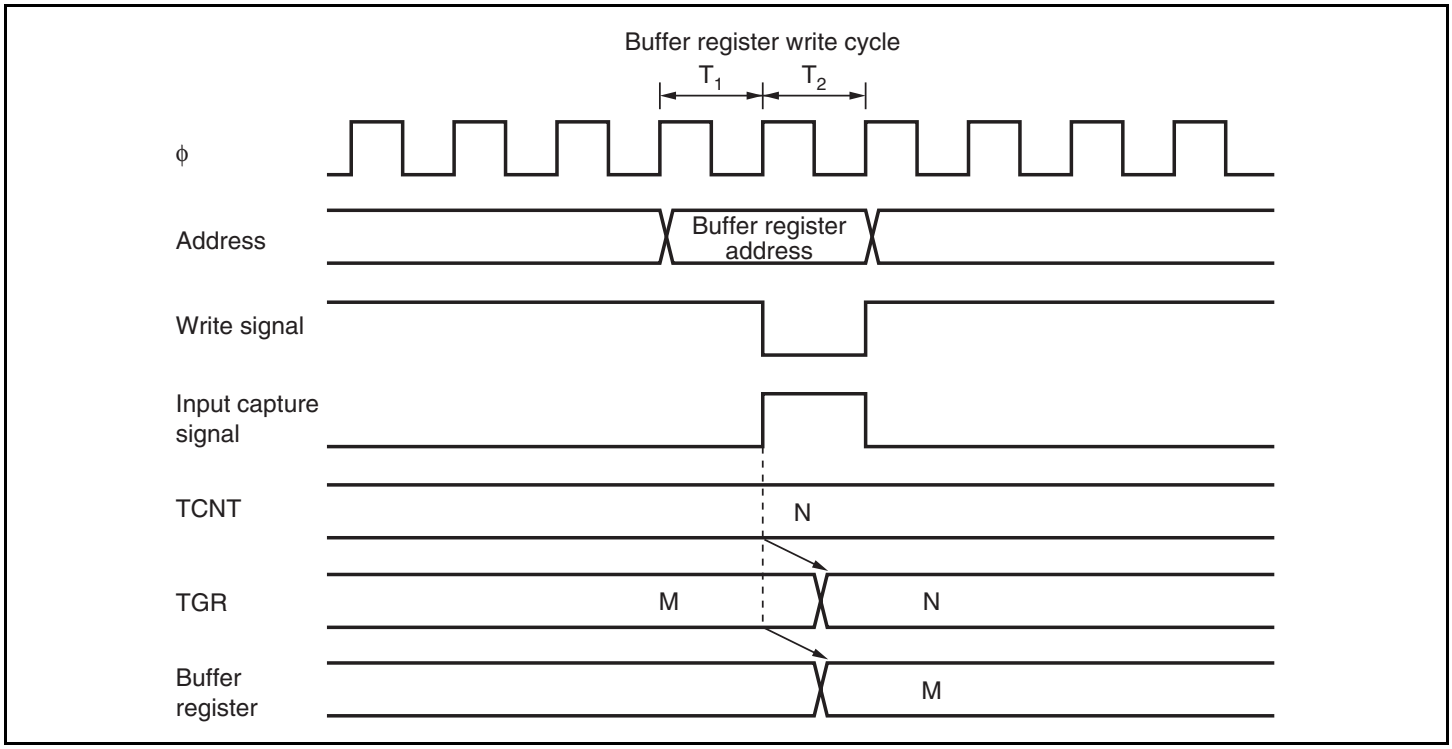


Figure 11.52 Contention between Buffer Register Write and Input Capture

11.10.11 Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 11.53 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.

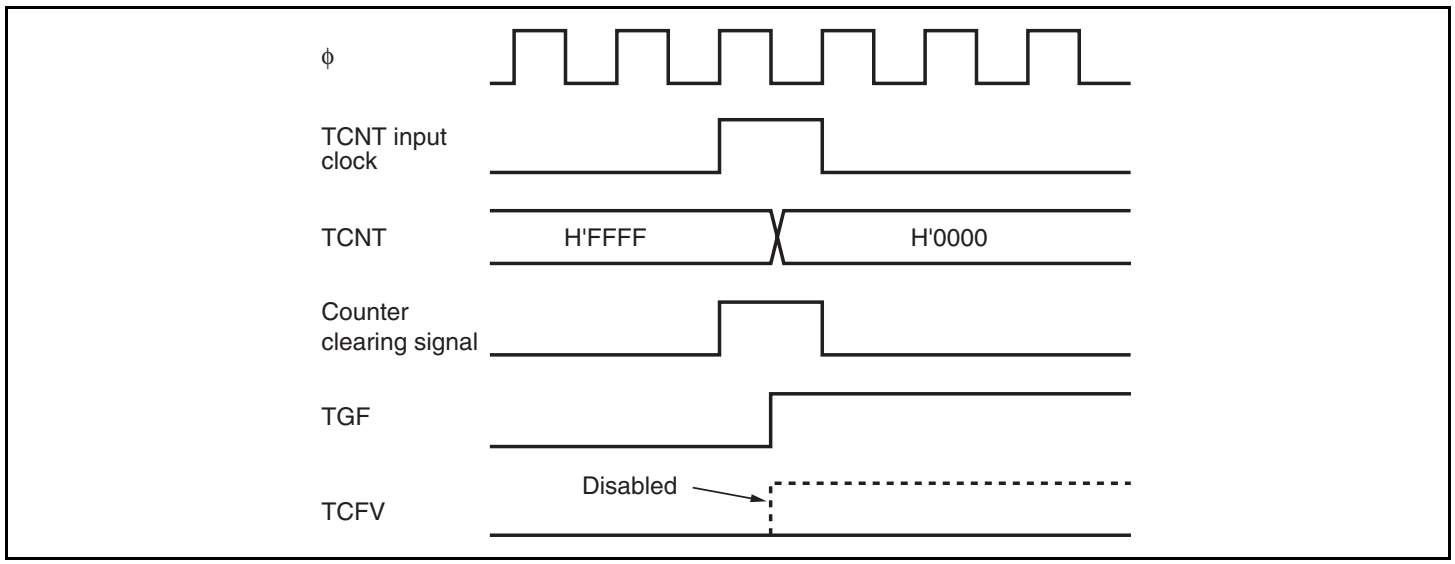


Figure 11.53 Contention between Overflow and Counter Clearing

11.10.12 Contention between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T_2 state of a TCNT write cycle, when overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 11.54 shows the operation timing when there is contention between TCNT write and overflow.

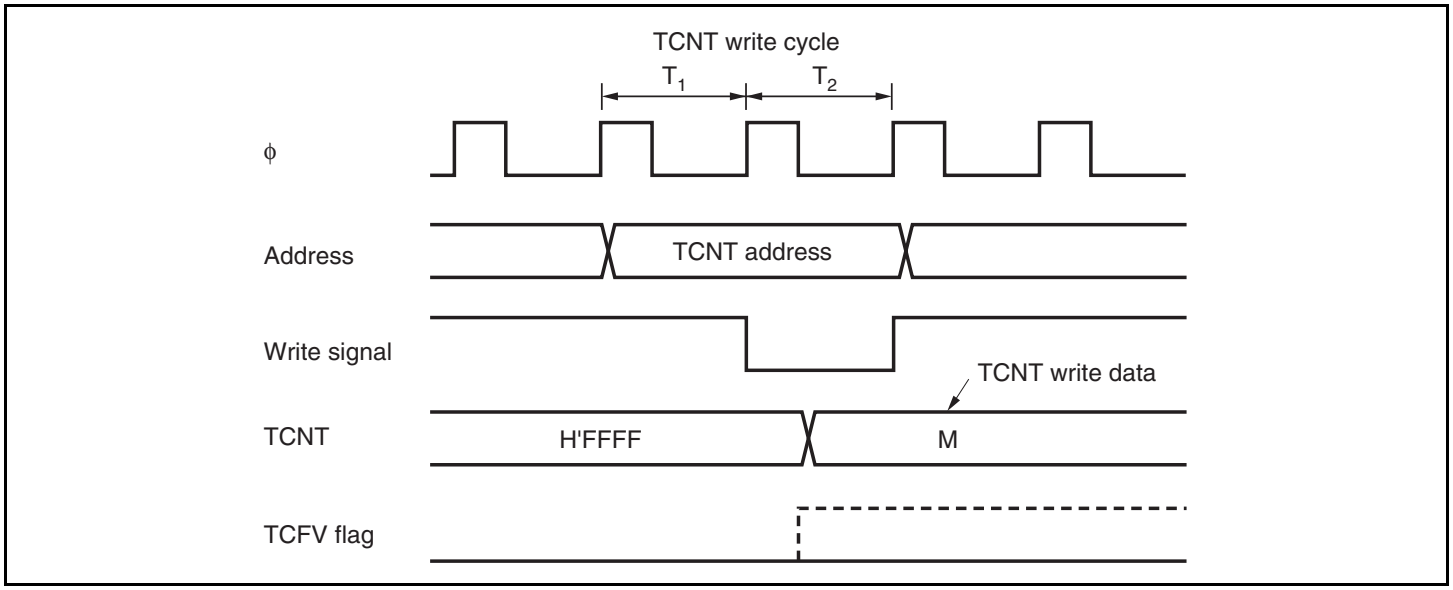


Figure 11.54 Contention between TCNT Write and Overflow

11.10.13 Multiplexing of I/O Pins

In this LSI, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLKD input pin with the TIOCB2 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

11.10.14 Interrupts and Module Stop Mode

If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DMAC or DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

Section 12 Programmable Pulse Generator (PPG)

The programmable pulse generator (PPG) provides pulse outputs by using the 16-bit timer pulse unit (TPU) as a time base. The PPG pulse outputs are divided into 4-bit groups (groups 3 to 0) that can operate both simultaneously and independently. The block diagram of PPG is shown in figure 12.1.

12.1 Features

- 16-bit output data
- Four output groups
- Selectable output trigger signals
- Non-overlap mode
- Can operate together with the data transfer controller (DTC) and the DMA controller (DMAC)
- Settable inverted output
- Module stop mode can be set

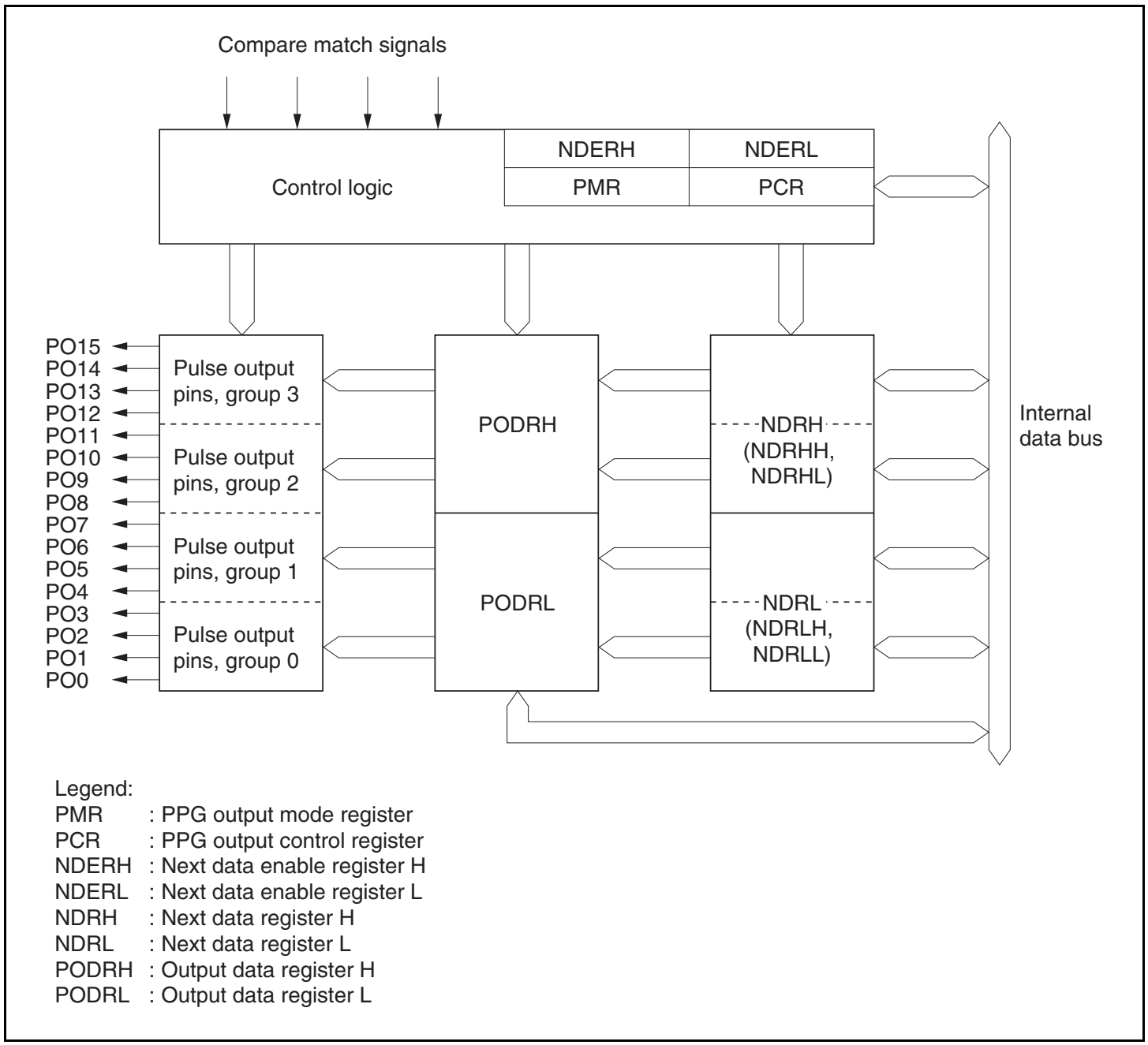


Figure 12.1 Block Diagram of PPG

12.2 Input/Output Pins

Table 12.1 shows the PPG pin configuration.

Table 12.1 Pin Configuration

Pin Name	I/O	Function
PO15	Output	Group 3 pulse output
PO14	Output	
PO13	Output	
PO12	Output	
PO11	Output	Group 2 pulse output
PO10	Output	
PO9	Output	
PO8	Output	
PO7	Output	Group 1 pulse output
PO6	Output	
PO5	Output	
PO4	Output	
PO3	Output	Group 0 pulse output
PO2	Output	
PO1	Output	
PO0	Output	

12.3 Register Descriptions

The PPG has the following registers.

- Next data enable register H (NDERH)
- Next data enable register L (NDERL)
- Output data register H (PODRH)
- Output data register L (PODRL)
- Next data register H (NDRH)
- Next data register L (NDRL)
- PPG output control register (PCR)
- PPG output mode register (PMR)

12.3.1 Next Data Enable Registers H and L (NDERH, NDERL)

NDERH and NDERL enable or disable pulse output on a bit-by-bit basis. For outputting pulse by the PPG, set the corresponding DDR to 1.

- **NDERH**

Bit	Bit Name	Initial Value	R/W	Description
7	NDER15	0	R/W	Next Data Enable 15 to 8
6	NDER14	0	R/W	When a bit is set to 1, the value in the corresponding NDRH bit is transferred to the PODRH bit by the selected output trigger. Values are not transferred from NDRH to PODRH for cleared bits.
5	NDER13	0	R/W	
4	NDER12	0	R/W	
3	NDER11	0	R/W	
2	NDER10	0	R/W	
1	NDER9	0	R/W	
0	NDER8	0	R/W	

- **NDERL**

Bit	Bit Name	Initial Value	R/W	Description
7	NDER7	0	R/W	Next Data Enable 7 to 0
6	NDER6	0	R/W	When a bit is set to 1, the value in the corresponding NDRL bit is transferred to the PODRL bit by the selected output trigger. Values are not transferred from NDRL to PODRL for cleared bits.
5	NDER5	0	R/W	
4	NDER4	0	R/W	
3	NDER3	0	R/W	
2	NDER2	0	R/W	
1	NDER1	0	R/W	
0	NDER0	0	R/W	

12.3.2 Output Data Registers H and L (PODRH, PODRL)

PODRH and PODRL store output data for use in pulse output. A bit that has been set for pulse output by NDER is read-only and cannot be modified.

- **PODRH**

Bit	Bit Name	Initial Value	R/W	Description
7	POD15	0	R/W	Output Data Register 15 to 8
6	POD14	0	R/W	For bits which have been set to pulse output by NDERH, the output trigger transfers NDRH values to this register during PPG operation. While NDERH is set to 1, the CPU cannot write to this register. While NDERH is cleared, the initial output value of the pulse can be set.
5	POD13	0	R/W	
4	POD12	0	R/W	
3	POD11	0	R/W	
2	POD10	0	R/W	
1	POD9	0	R/W	
0	POD8	0	R/W	

- **PODRL**

Bit	Bit Name	Initial Value	R/W	Description
7	POD7	0	R/W	Output Data Register 7 to 0
6	POD6	0	R/W	For bits which have been set to pulse output by NDERL, the output trigger transfers NDRL values to this register during PPG operation. While NDERL is set to 1, the CPU cannot write to this register. While NDERL is cleared, the initial output value of the pulse can be set.
5	POD5	0	R/W	
4	POD4	0	R/W	
3	POD3	0	R/W	
2	POD2	0	R/W	
1	POD1	0	R/W	
0	POD0	0	R/W	

12.3.3 Next Data Registers H and L (NDRH, NDRL)

NDRH and NDRL store the next data for pulse output. The NDR addresses differ depending on whether pulse output groups have the same output trigger or different output triggers.

- **NDRH (NDRHH, NDRHL)***

If pulse output groups 2 and 3 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time, as shown below.

Note: * When pulse output groups 2 and 3 have the same output trigger by PCR settings, the NDRH address is H′FF4C. When they have different output triggers, the NDRH addresses corresponding to the groups 2 and 3 are NDRHH (H′FF4E) and NDRHL (H′FF4C), respectively. Also, when pulse output groups 0 and 1 have the same output trigger by PCR settings, the NDRL address is NDRLH (H′FF4D). When they have different output triggers, the NDRL addresses corresponding to the groups 0 and 1 are NDRLL (H′FF4F) and H′FF4D respectively.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 15 to 8
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3	NDR11	0	R/W	
2	NDR10	0	R/W	
1	NDR9	0	R/W	
0	NDR8	0	R/W	

If pulse output groups 2 and 3 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 15 to 12
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3 to 0	—	All 1	—	Reserved 1 is always read and write is disabled.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved 1 is always read and write is disabled.
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 specified with PCR.
1	NDR9	0	R/W	
0	NDR8	0	R/W	

- **NDRL (NDRLH, NDRLL)***

If pulse output groups 0 and 1 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time, as shown below.

Note: * When pulse output groups 2 and 3 have the same output trigger by PCR settings, the NDRH address is H′FF4C. When they have different output triggers, the NDRH addresses corresponding to the groups 2 and 3 are NDRHH (H′FF4E) and NDRHL (H′FF4C), respectively. Also, when pulse output groups 0 and 1 have the same output trigger by PCR settings, the NDRL address is NDRLH (H′FF4D). When they have different output triggers, the NDRL addresses corresponding to the groups 0 and 1 are NDRLL (H′FF4F) and H′FF4D respectively.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 7 to 0
6	NDR6	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3	NDR3	0	R/W	
2	NDR2	0	R/W	
1	NDR1	0	R/W	
0	NDR0	0	R/W	

If pulse output groups 0 and 1 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 7 to 4
6	NDR6	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3 to 0	—	All 1	—	Reserved 1 is always read and write is disabled.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved 1 is always read and write is disabled.
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 specified with PCR.
1	NDR1	0	R/W	
0	NDR0	0	R/W	

12.3.4 PPG Output Control Register (PCR)

PCR selects output trigger signals on a group-by-group basis. For details on output trigger selection, refer to section 12.3.5, PPG Output Mode Register (PMR).

Bit	Bit Name	Initial Value	R/W	Description
7	G3CMS1	1	R/W	Group 3 Compare Match Select 1 and 0
6	G3CMS0	1	R/W	Select output trigger of pulse output group 3. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
5	G2CMS1	1	R/W	Group 2 Compare Match Select 1 and 0
4	G2CMS0	1	R/W	Select output trigger of pulse output group 2. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
3	G1CMS1	1	R/W	Group 1 Compare Match Select 1 and 0
2	G1CMS0	1	R/W	Select output trigger of pulse output group 1. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
1	G0CMS1	1	R/W	Group 0 Compare Match Select 1 and 0
0	G0CMS0	1	R/W	Select output trigger of pulse output group 0. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3

12.3.5 PPG Output Mode Register (PMR)

PMR selects the pulse output mode of the PPG for each group. If inverted output is selected, a low-level pulse is output when PODRH is 1 and a high-level pulse is output when PODRH is 0. If non-overlapping operation is selected, PPG updates its output values at compare match A or B of the TPU that becomes the output trigger. For details, refer to section 12.4.4, Non-Overlapping Pulse Output.

Bit	Bit Name	Initial Value	R/W	Description
7	G3INV	1	R/W	Group 3 Inversion Selects direct output or inverted output for pulse output group 3. 0: Inverted output 1: Direct output
6	G2INV	1	R/W	Group 2 Inversion Selects direct output or inverted output for pulse output group 2. 0: Inverted output 1: Direct output
5	G1INV	1	R/W	Group 1 Inversion Selects direct output or inverted output for pulse output group 1. 0: Inverted output 1: Direct output
4	G0INV	1	R/W	Group 0 Inversion Selects direct output or inverted output for pulse output group 0. 0: Inverted output 1: Direct output

Bit	Bit Name	Initial Value	R/W	Description
3	G3NOV	0	R/W	Group 3 Non-Overlap Selects normal or non-overlapping operation for pulse output group 3. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)
2	G2NOV	0	R/W	Group 2 Non-Overlap Selects normal or non-overlapping operation for pulse output group 2. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)
1	G1NOV	0	R/W	Group 1 Non-Overlap Selects normal or non-overlapping operation for pulse output group 1. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)
0	G0NOV	0	R/W	Group 0 Non-Overlap Selects normal or non-overlapping operation for pulse output group 0. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)

12.4 Operation

Figure 12.2 shows an overview diagram of the PPG. PPG pulse output is enabled when the corresponding bits in P1DDR, P2DDR, and NDER are set to 1. An initial output value is determined by its corresponding PODR initial setting. When the compare match event specified by PCR occurs, the corresponding NDR bit contents are transferred to PODR to update the output values. Sequential output of data of up to 16 bits is possible by writing new output data to NDR before the next compare match.

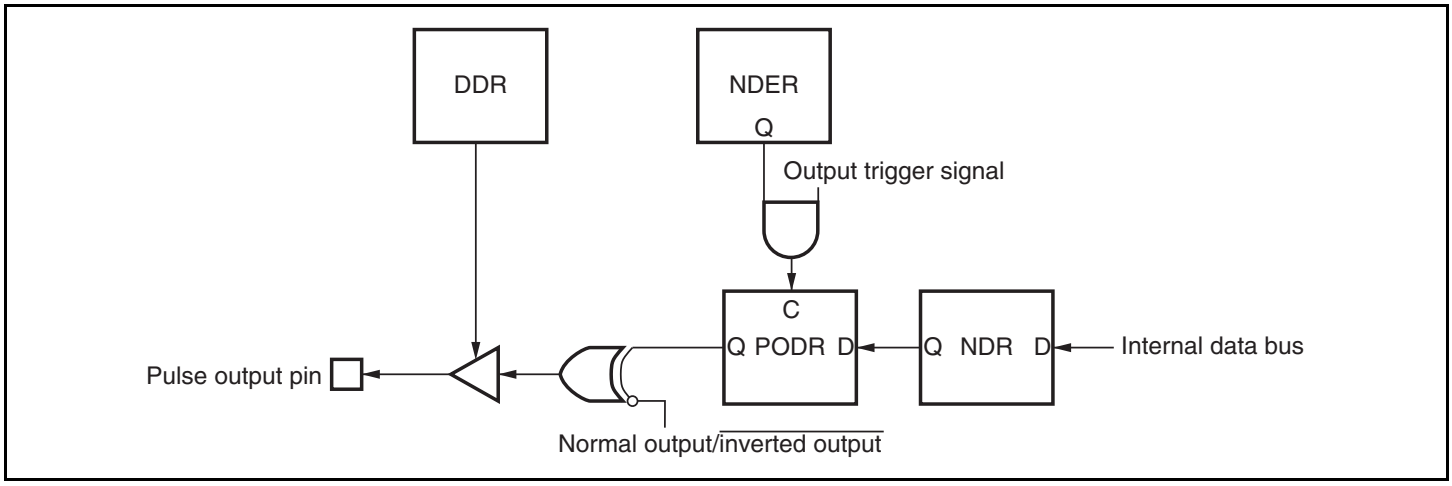


Figure 12.2 Overview Diagram of PPG

12.4.1 Output Timing

If pulse output is enabled, NDR contents are transferred to PODR and output when the specified compare match event occurs. Figure 12.3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare match A.

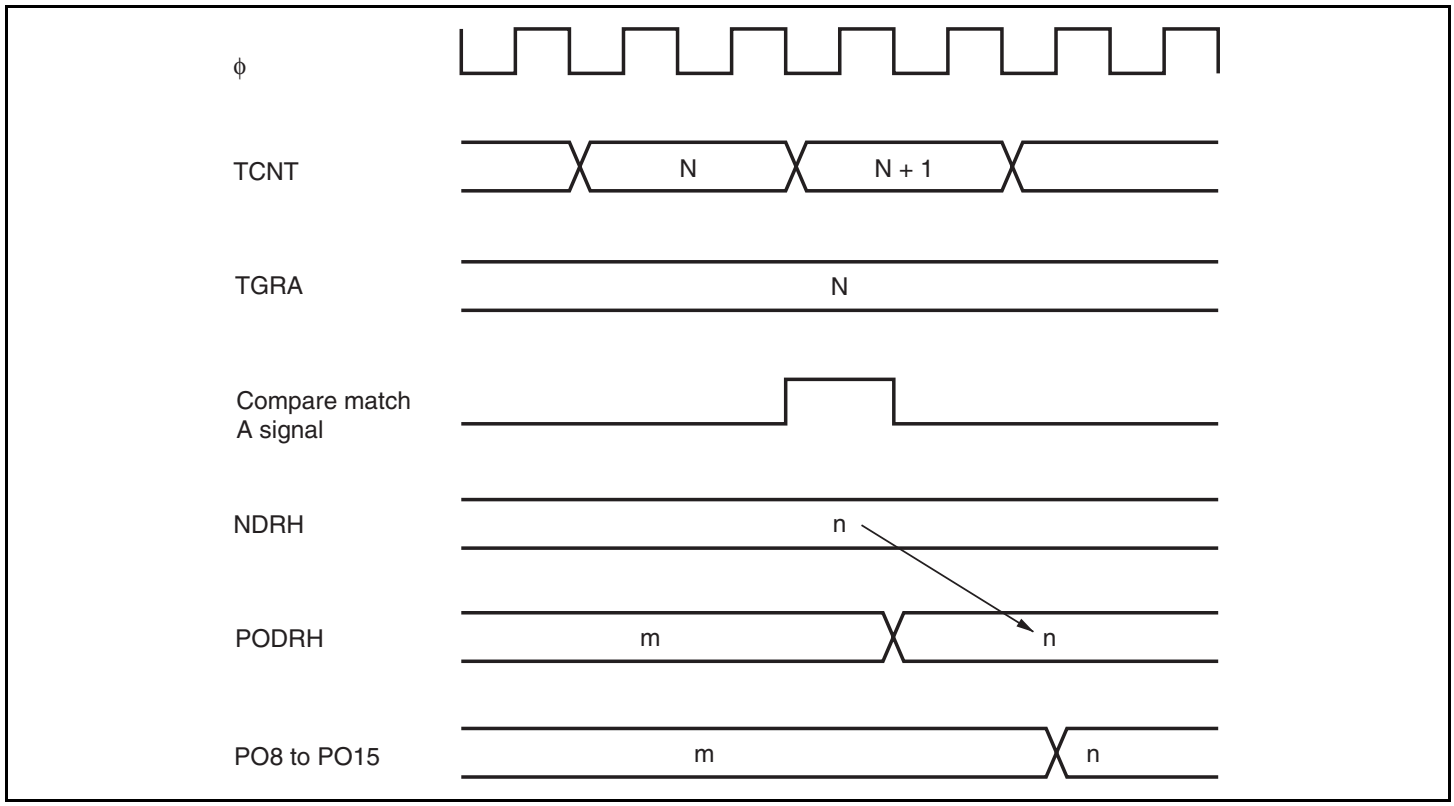


Figure 12.3 Timing of Transfer and Output of NDR Contents (Example)

12.4.2 Sample Setup Procedure for Normal Pulse Output

Figure 12.4 shows a sample procedure for setting up normal pulse output.

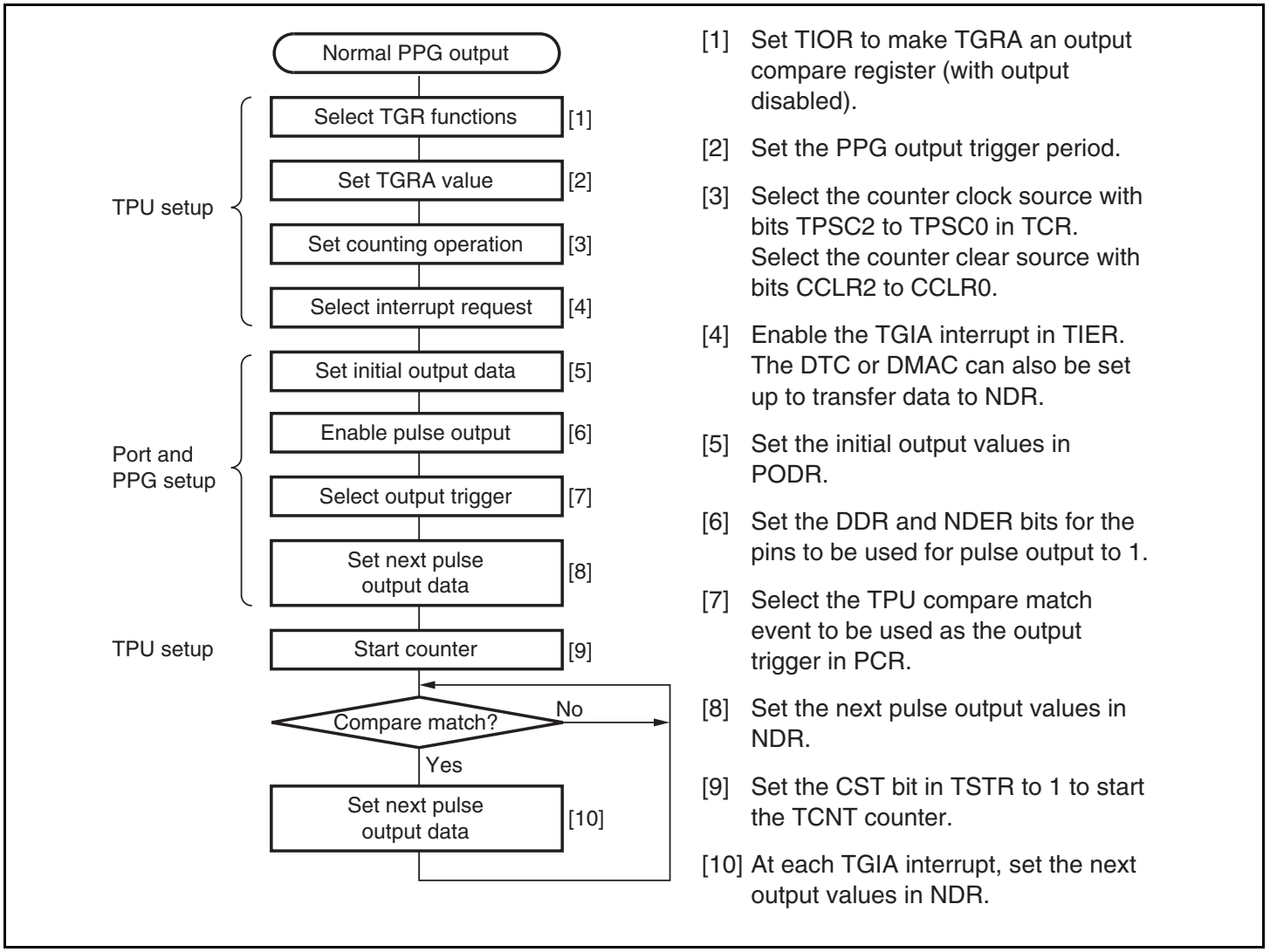


Figure 12.4 Setup Procedure for Normal Pulse Output (Example)

12.4.3 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

Figure 12.5 shows an example in which pulse output is used for cyclic five-phase pulse output.

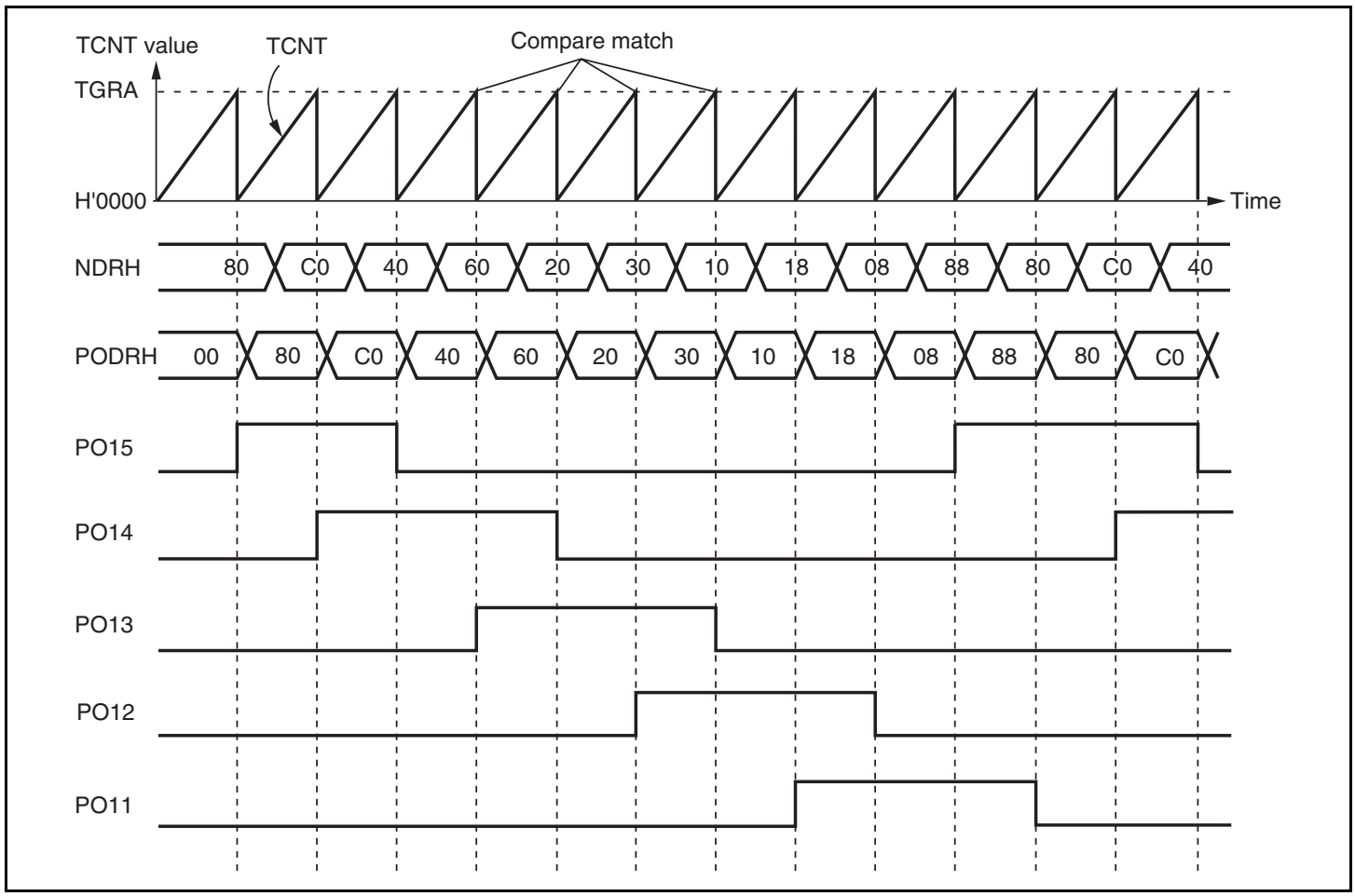


Figure 12.5 Normal Pulse Output Example (Five-Phase Pulse Output)

1. Set up TGRA in TPU which is used as the output trigger to be an output compare register. Set a cycle in TGRA so that the counter will be cleared by compare match A. Set the TGIEA bit in TIER to 1 to enable the compare match/input capture A (TGIA) interrupt.
2. Write H'F8 in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Write output data H'80 in NDRH.
3. The timer counter in the TPU channel starts. When compare match A occurs, the NDRH contents are transferred to PODRH and output. The TGIA interrupt handling routine writes the next output data (H'C0) in NDRH.
4. Five-phase pulse output (one or two phases active at a time) can be obtained subsequently by writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88... at successive TGIA interrupts.

   If the DTC or DMAC is set for activation by the TGIA interrupt, pulse output can be obtained without imposing a load on the CPU.

12.4.4 Non-Overlapping Pulse Output

During non-overlapping operation, transfer from NDR to PODR is performed as follows:

- NDR bits are always transferred to PODR bits at compare match A.
- At compare match B, NDR bits are transferred only if their value is 0. Bits are not transferred if their value is 1.

Figure 12.6 illustrates the non-overlapping pulse output operation.

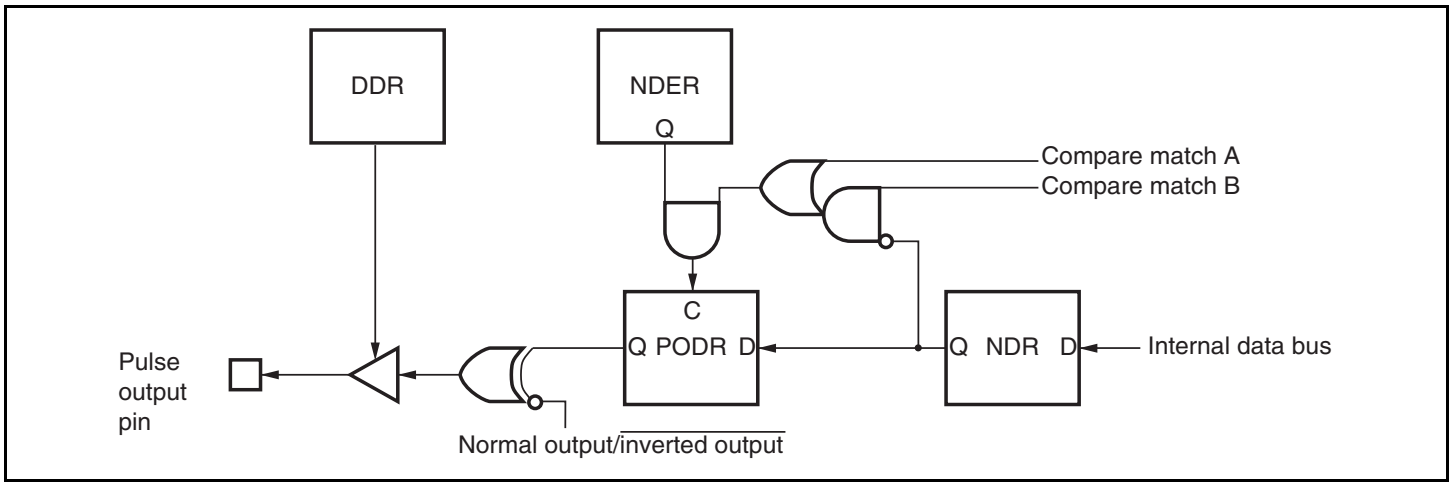


Figure 12.6 Non-Overlapping Pulse Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur before compare match A.

The NDR contents should not be altered during the interval from compare match B to compare match A (the non-overlap margin).

This can be accomplished by having the TGIA interrupt handling routine write the next data in NDR, or by having the TGIA interrupt activate the DTC or DMAC. Note, however, that the next data must be written before the next compare match B occurs.

Figure 12.7 shows the timing of this operation.

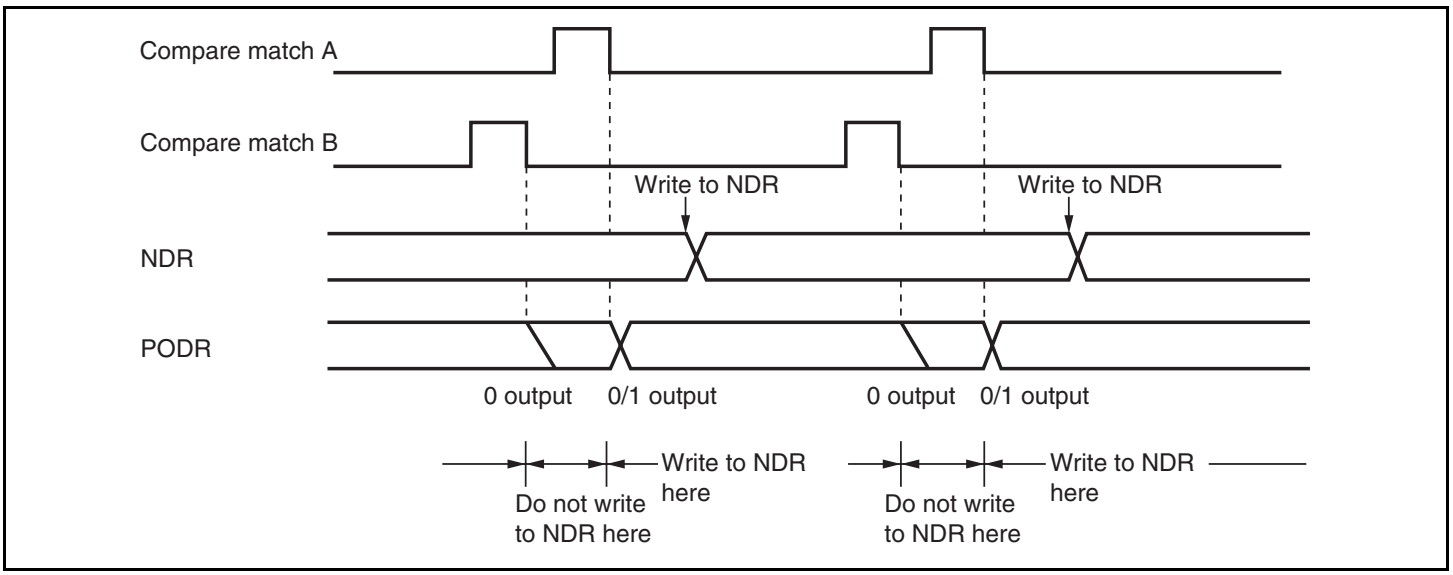


Figure 12.7 Non-Overlapping Operation and NDR Write Timing

12.4.5 Sample Setup Procedure for Non-Overlapping Pulse Output

Figure 12.8 shows a sample procedure for setting up non-overlapping pulse output.

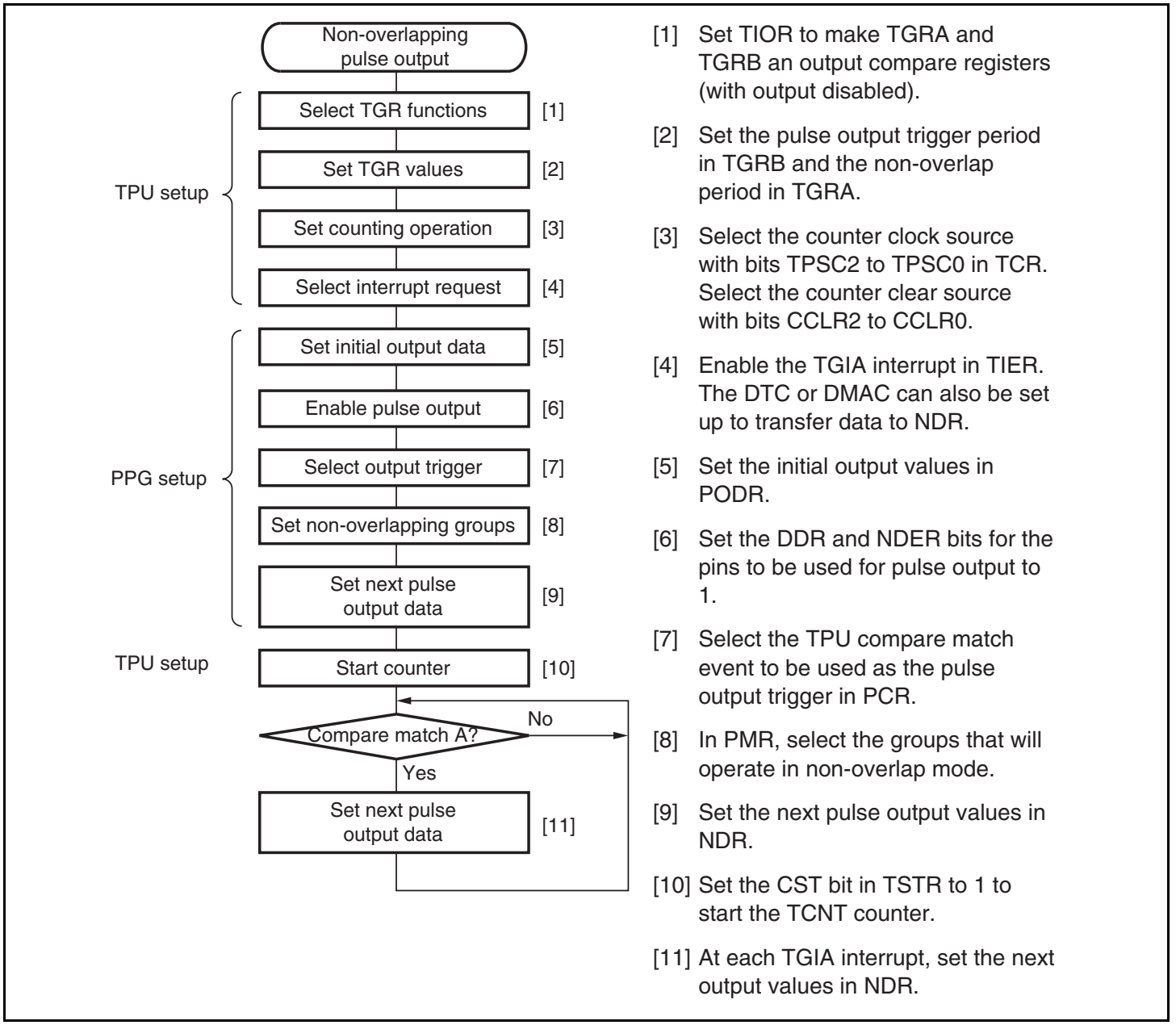


Figure 12.8 Setup Procedure for Non-Overlapping Pulse Output (Example)

12.4.6 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)

Figure 12.9 shows an example in which pulse output is used for four-phase complementary non-overlapping pulse output.

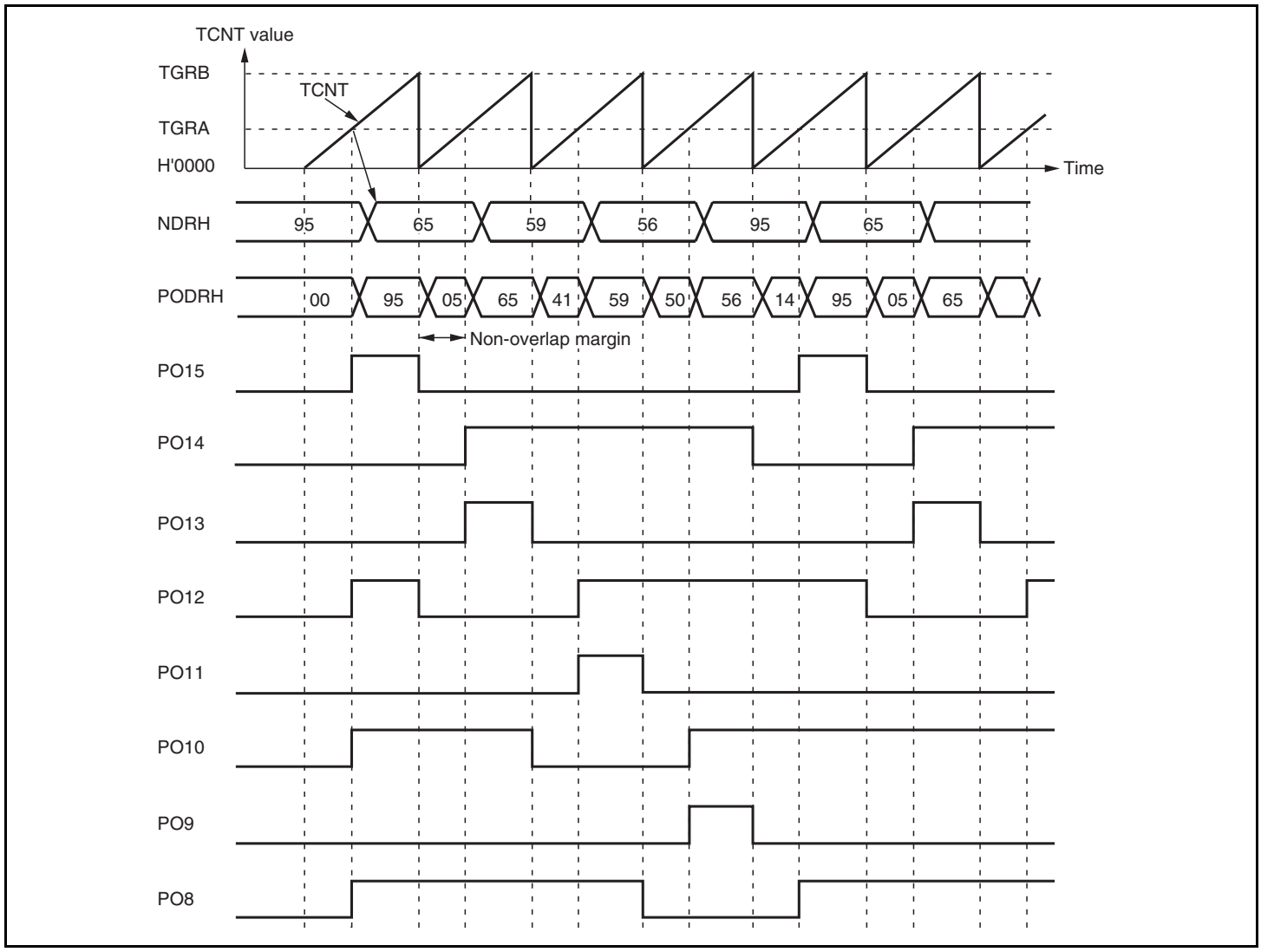


Figure 12.9 Non-Overlapping Pulse Output Example (Four-Phase Complementary)

1. Set up the TPU channel to be used as the output trigger channel so that TGRA and TGRB are output compare registers. Set the trigger period in TGRB and the non-overlap margin in TGRA, and set the counter to be cleared by compare match B. Set the TGIEA bit in TIER to 1 to enable the TGIA interrupt.

2. Write H'FF in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Set the G3NOV and G2NOV bits in PMR to 1 to select non-overlapping output. Write output data H'95 in NDRH.

3. The timer counter in the TPU channel starts. When a compare match with TGRB occurs, outputs change from 1 to 0. When a compare match with TGRA occurs, outputs change from 0 to 1 (the change from 0 to 1 is delayed by the value set in TGRA). The TGIA interrupt handling routine writes the next output data (H'65) in NDRH.

4. Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing H'59, H'56, H'95... at successive TGIA interrupts.

   If the DTC or DMAC is set for activation by the TGIA interrupt, pulse output can be obtained without imposing a load on the CPU.

12.4.7 Inverted Pulse Output

If the G3INV, G2INV, G1INV, and G0INV bits in PMR are cleared to 0, values that are the inverse of the PODR contents can be output.

Figure 12.10 shows the outputs when G3INV and G2INV are cleared to 0, in addition to the settings of figure 12.9.

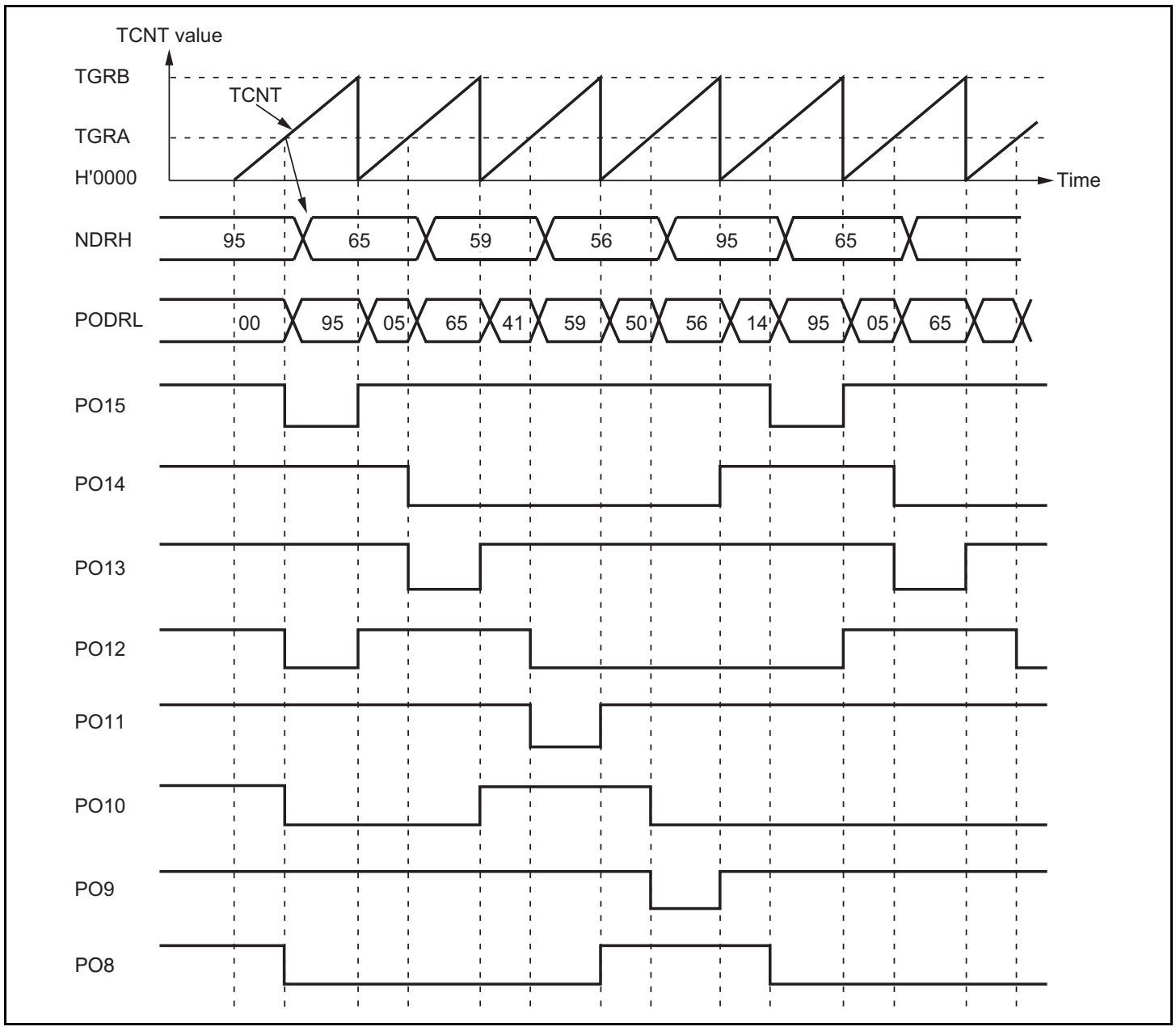


Figure 12.10 Inverted Pulse Output (Example)

12.4.8 Pulse Output Triggered by Input Capture

Pulse output can be triggered by TPU input capture as well as by compare match. If TGRA functions as an input capture register in the TPU channel selected by PCR, pulse output will be triggered by the input capture signal.

Figure 12.11 shows the timing of this output.

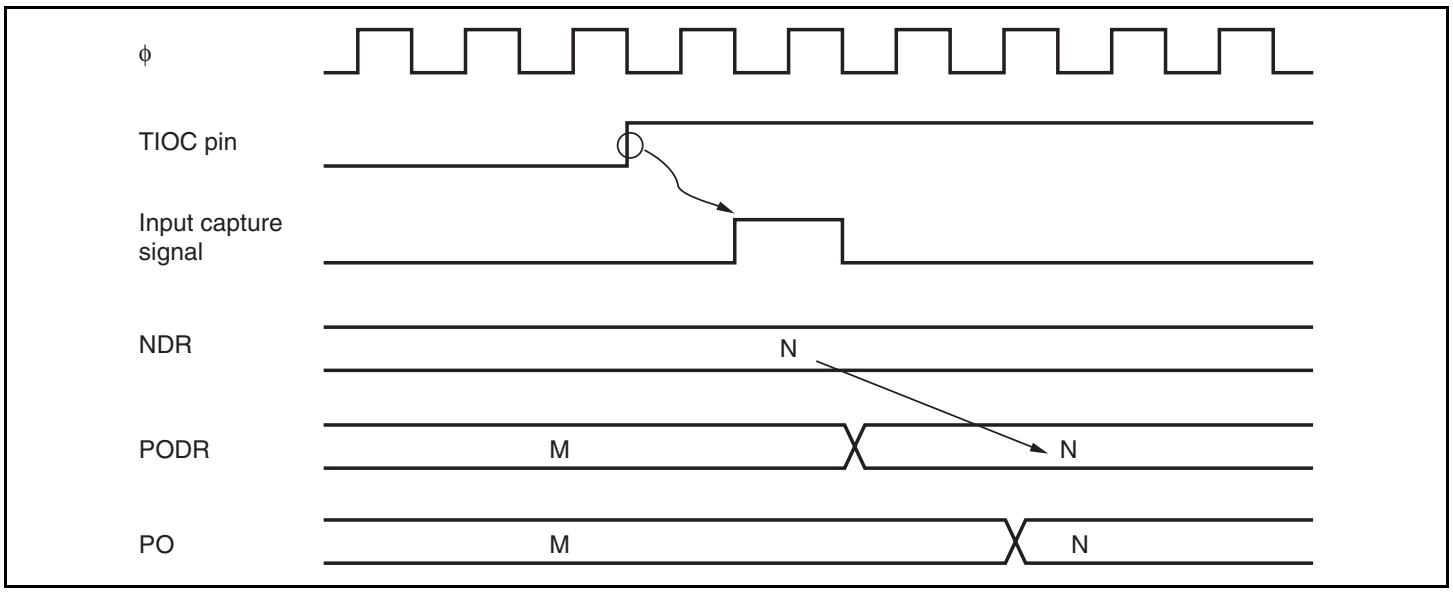


Figure 12.11 Pulse Output Triggered by Input Capture (Example)

12.5 Usage Notes

12.5.1 Module Stop Mode Setting

PPG operation can be disabled or enabled using the module stop control register. The initial value is for PPG operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 25, Power-Down Modes.

12.5.2 Operation of Pulse Output Pins

Pins PO0 to PO15 are also used for other peripheral functions such as the TPU. When output by another peripheral function is enabled, the corresponding pins cannot be used for pulse output. Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of the usage of the pins.

Pin functions should be changed only under conditions in which the output trigger event will not occur.

Section 13 8-Bit Timers (TMR)

This LSI has an on-chip 8-bit timer module with two channels operating on the basis of an 8-bit counter. The 8-bit timer module can be used to count external events and be used as a multifunction timer in a variety of applications, such as generation of counter reset, interrupt requests, and pulse output with an arbitrary duty cycle using a compare-match signal with two registers.

13.1 Features

- Selection of seven clock sources

  The counters can be driven by one of six internal clock signals (ϕ/8, ϕ/64, ϕ/8192, ϕ/2, ϕ/32, or ϕ/1024) or an external clock input
- 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
- Timer output control by a combination of two compare match signals

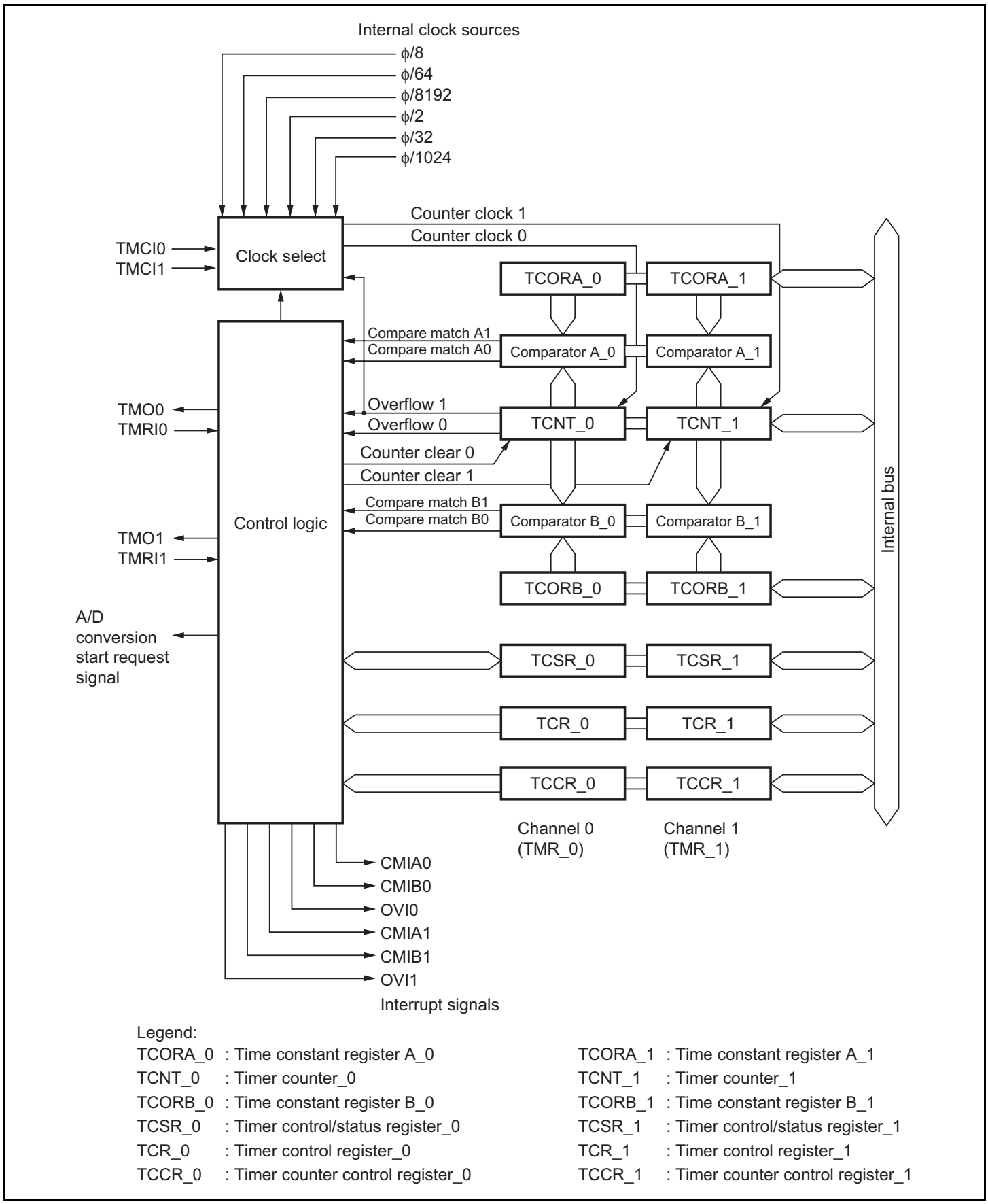
  The timer output signal in each channel is controlled by a combination of two independent compare match signals, enabling the timer to generate output waveforms with an arbitrary duty cycle or PWM output
- Provision for cascading of two channels (TMR_0 and TMR_1)

  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 independent interrupts

  Compare match A and B and overflow interrupts can be requested independently
- A/D converter conversion start trigger can be generated

Figure 13.1 shows a block diagram of the 8-bit timer module (TMR_0 and TMR_1).



Legend:

TCORA_0	: Time constant register A_0	TCORA_1	: Time constant register A_1
TCNT_0	: Timer counter_0	TCNT_1	: Timer counter_1
TCORB_0	: Time constant register B_0	TCORB_1	: Time constant register B_1
TCSR_0	: Timer control/status register_0	TCSR_1	: Timer control/status register_1
TCR_0	: Timer control register_0	TCR_1	: Timer control register_1
TCCR_0	: Timer counter control register_0	TCCR_1	: Timer counter control register_1

Figure 13.1 Block Diagram of 8-Bit Timer Module

13.2 Input/Output Pins

Table 13.1 shows the pin configuration of the 8-bit timer module.

Table 13.1 Pin Configuration

Channel	Name	Symbol	I/O	Function
0	Timer output pin	TMO0	Output	Outputs at compare match
	Timer clock input pin	TMCI0	Input	Inputs external clock for counter
	Timer reset input pin	TMRI0	Input	Inputs external reset to counter
1	Timer output pin	TMO1	Output	Outputs at compare match
	Timer clock input pin	TMCI1	Input	Inputs external clock for counter
	Timer reset input pin	TMRI1	Input	Inputs external reset to counter

13.3 Register Descriptions

The 8-bit timer module has the following registers. For details on the module stop control register, see section 25.1.2, Module Stop Control Registers H and L (MSTPCRH, MSTPCRL).

- Timer counter_0 (TCNT_0)
- Time constant register A_0 (TCORA_0)
- Time constant register B_0 (TCORB_0)
- Timer control register_0 (TCR_0)
- Timer control/status register_0 (TCSR_0)
- Timer counter control register_0 (TCCR_0)
- Timer counter_1 (TCNT_1)
- Time constant register A_1 (TCORA_1)
- Time constant register B_1 (TCORB_1)
- Timer control register_1 (TCR_1)
- Timer control/status register_1 (TCSR_1)
- Timer counter control register_1 (TCCR_1)

13.3.1 Timer Counter (TCNT)

TCNT is 8-bit up-counter. TCNT_0 and TCNT_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction. Bits CKS2 to CKS0 in TCR are used to select a clock. TCNT can be cleared by an external reset input or by a compare match signal A or B. Which signal is to be used for clearing is selected by bits CCLR1 and CCLR0 in TCR. When TCNT overflows from H'FF to H'00, OVF in TCSR is set to 1. TCNT is initialized to H'00.

13.3.2 Time Constant Register A (TCORA)

TCORA is 8-bit readable/writable register. TCORA_0 and TCORA_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction. The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding CMFA flag in TCSR is set to 1. Note, however, that comparison is disabled during the T_2 state of a TCORA write cycle. The timer output from the TMO pin can be freely controlled by this compare match signal (compare match A) and the settings of bits OS1 and OS0 in TCSR. TCORA is initialized to H'FF.

13.3.3 Time Constant Register B (TCORB)

TCORB is 8-bit readable/writable register. TCORB_0 and TCORB_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction. TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding CMFB flag in TCSR is set to 1. Note, however, that comparison is disabled during the T_2 state of a TCOBR write cycle. The timer output from the TMO pin can be freely controlled by this compare match signal (compare match B) and the settings of bits OS3 and OS2 in TCSR. TCORB is initialized to H'FF.

13.3.4 Timer Control Register (TCR)

TCR selects the clock source and the time at which TCNT is cleared, and controls interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare Match Interrupt Enable B Selects whether CMFB interrupt requests (CMIB) are enabled or disabled when the CMFB flag in TCSR is set to 1. 0: CMFB interrupt requests (CMIB) are disabled 1: CMFB interrupt requests (CMIB) are enabled
6	CMIEA	0	R/W	Compare Match Interrupt Enable A Selects whether CMFA interrupt requests (CMIA) are enabled or disabled when the CMFA flag in TCSR is set to 1. 0: CMFA interrupt requests (CMIA) are disabled 1: CMFA interrupt requests (CMIA) are enabled
5	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether OVF interrupt requests (OVI) are enabled or disabled when the OVF flag in TCSR is set to 1. 0: OVF interrupt requests (OVI) are disabled 1: OVF interrupt requests (OVI) are enabled
4	CCLR1	0	R/W	Counter Clear 1 and 0 These bits select the method by which TCNT is cleared, in combination with the TMRIS bit in TCCR. See table 13.2.
3	CCLR0	0	R/W	
2	CKS2	0	R/W	Clock Select 2 to 0 These bits select the clock input to TCNT and the count condition, in combination with the ICKS1 and ICKS0 bits in TCCR. See table 13.3.
1	CKS1	0	R/W	
0	CKS0	0	R/W	

13.3.5 Timer Counter Control Register (TCCR)

TCCR selects the TCNT internal clock source and controls the external reset input.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
3	TMRIS	0	R/W	Timer Reset Input Select Selects the external reset input, in combination with the CCLR1 and CCLR0 bits in TCR. See table 13.2.
2	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
1	ICKS1	0	R/W	Internal Clock Select 1, 0
0	ICKS0	0	R/W	These bits select the internal clock source, in combination with the CKS2 to CKS0 bits in TCR. See table 13.3.

Table 13.2 Reset Input to TCNT and Clearing Condition

TCR		TCCR	
Bit 1 CCLR1	Bit 0 CCLR0	Bit 3 TMRIS	Description
0	0	0	Clearing is disabled
0	1	0	Clear by compare match A
1	0	0	Clear by compare match B
1	1	0	Clear by rising edge of external reset input
0	0	1	Clear by both rising and falling edges of external reset input
0	1	1	Clear by falling edge of external reset input
1	0	1	Clear by low level of external reset input
1	1	1	Clear by high level of external reset input

Table 13.3 Clock Input to TCNT and Count Condition

	TCR			TCCR		
Channel	Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Bit 1 ICKS1	Bit 0 ICKS0	Description
TMR_0	0	0	0	—	—	Clock input disabled
	0	0	1	0	0	Internal clock, counted at rising edge of φ/8
				0	1	Internal clock, counted at rising edge of φ/2
				1	0	Internal clock, counted at falling edge of φ/8
				1	1	Internal clock, counted at falling edge of φ/2
	0	1	0	0	0	Internal clock, counted at rising edge of φ/64
				0	1	Internal clock, counted at rising edge of φ/32
				1	0	Internal clock, counted at falling edge of φ/64
				1	1	Internal clock, counted at falling edge of φ/32
	0	1	1	0	0	Internal clock, counted at rising edge of φ/8192
				0	1	Internal clock, counted at rising edge of φ/1024
				1	0	Internal clock, counted at falling edge of φ/8192
				1	1	Internal clock, counted at falling edge of φ/1024
	1	0	0	—	—	Counted at TCNT_1 overflow signal*
TMR_1	0	0	0	—	—	Clock input disabled
	0	0	1	0	0	Internal clock, counted at rising edge of φ/8
				0	1	Internal clock, counted at rising edge of φ/2
				1	0	Internal clock, counted at falling edge of φ/8
				1	1	Internal clock, counted at falling edge of φ/2
	0	1	0	0	0	Internal clock, counted at rising edge of φ/64
				0	1	Internal clock, counted at rising edge of φ/32
				1	0	Internal clock, counted at falling edge of φ/64
				1	1	Internal clock, counted at falling edge of φ/32
	0	1	1	0	0	Internal clock, counted at rising edge of φ/8192
				0	1	Internal clock, counted at rising edge of φ/1024
				1	0	Internal clock, counted at falling edge of φ/8192
				1	1	Internal clock, counted at falling edge of φ/1024
	1	0	0	—	—	Counted at TCNT_0 compare match A*

	TCR			TCCR		
Channel	Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Bit 1 ICKS1	Bit 0 ICKS0	Description
All	1	0	1	—	—	External clock, counted at rising edge
		1	0	—	—	External clock, counted at falling edge
		1	1	—	—	External clock, counted at both rising and falling edges

Note: * If the count input of TMR_0 is the TCNT_1 overflow signal and that of TMR_1 is the TCNT_0 compare match signal, no incrementing clock is generated. Do not use this setting.

13.3.6 Timer Control/Status Register (TCSR)

TCSR displays status flags, and controls compare match output.

- TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare Match Flag B [Setting condition] • Set when TCNT matches TCORB [Clearing conditions] • Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB • When DTC is activated by CMIB interrupt while DISEL bit of MRB in DTC is 0
6	CMFA	0	R/(W)*	Compare Match Flag A [Setting condition] • Set when TCNT matches TCORA [Clearing conditions] • Cleared by reading CMFA when CMFA = 1, then writing 0 to CMFA • When DTC is activated by CMIA interrupt while DISEL bit of MRB in DTC is 0

Bit	Bit Name	Initial Value	R/W	Description
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] Set when TCNT overflows from H'FF to H'00 [Clearing condition] Cleared by reading OVF when OVF = 1, then writing 0 to OVF
4	ADTE	0	R/W	A/D Trigger Enable Selects enabling or disabling of A/D converter start requests by compare match A. 0: A/D converter start requests by compare match A are disabled 1: A/D converter start requests by compare match A are enabled
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits select a method of TMO pin output when compare match B of TCORB and TCNT occurs. 00: No change when compare match B occurs 01: 0 is output when compare match B occurs 10: 1 is output when compare match B occurs 11: Output is inverted when compare match B occurs (toggle output)
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits select a method of TMO pin output when compare match A of TCORA and TCNT occurs. 00: No change when compare match A occurs 01: 0 is output when compare match A occurs 10: 1 is output when compare match A occurs 11: Output is inverted when compare match A occurs (toggle output)

Note: Only 0 can be written to, to clear these flags.

- TCSR_1

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare Match Flag B [Setting condition] • Set when TCNT matches TCORB [Clearing conditions] • Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB • When DTC is activated by CMIB interrupt while DISEL bit of MRB in DTC is 0
6	CMFA	0	R/(W)*	Compare Match Flag A [Setting condition] • Set when TCNT matches TCORA [Clearing conditions] • Cleared by reading CMFA when CMFA = 1, then writing 0 to CMFA • When DTC is activated by CMIA interrupt while DISEL bit of MRB in DTC is 0
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] • Set when TCNT overflows from H'FF to H'00 [Clearing condition] • Cleared by reading OVF when OVF = 1, then writing 0 to OVF
4	—	1	R	Reserved This bit is always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits select a method of TMO pin output when compare match B of TCORB and TCNT occurs. 00: No change when compare match B occurs 01: 0 is output when compare match B occurs 10: 1 is output when compare match B occurs 11: Output is inverted when compare match B occurs (toggle output)
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits select a method of TMO pin output when compare match A of TCORA and TCNT occurs. 00: No change when compare match A occurs 01: 0 is output when compare match A occurs 10: 1 is output when compare match A occurs 11: Output is inverted when compare match A occurs (toggle output)

Note: * Only 0 can be written to, to clear these flags.

13.4 Operation

13.4.1 Pulse Output

Figure 13.2 shows an example in which the 8-bit timer is used to generate a pulse output with a selected duty cycle. The control bits are set as follows:

[1] In TCR, the CCLR1 bit is cleared to 0 and the CCLR0 bit is set to 1 so that TCNT is cleared at a TCORA compare match.

[2] In TCSR, the OS3 to OS0 bits are set 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 output of pulses at a rate determined by TCORA with a pulse width determined by TCORB. No software intervention is required.

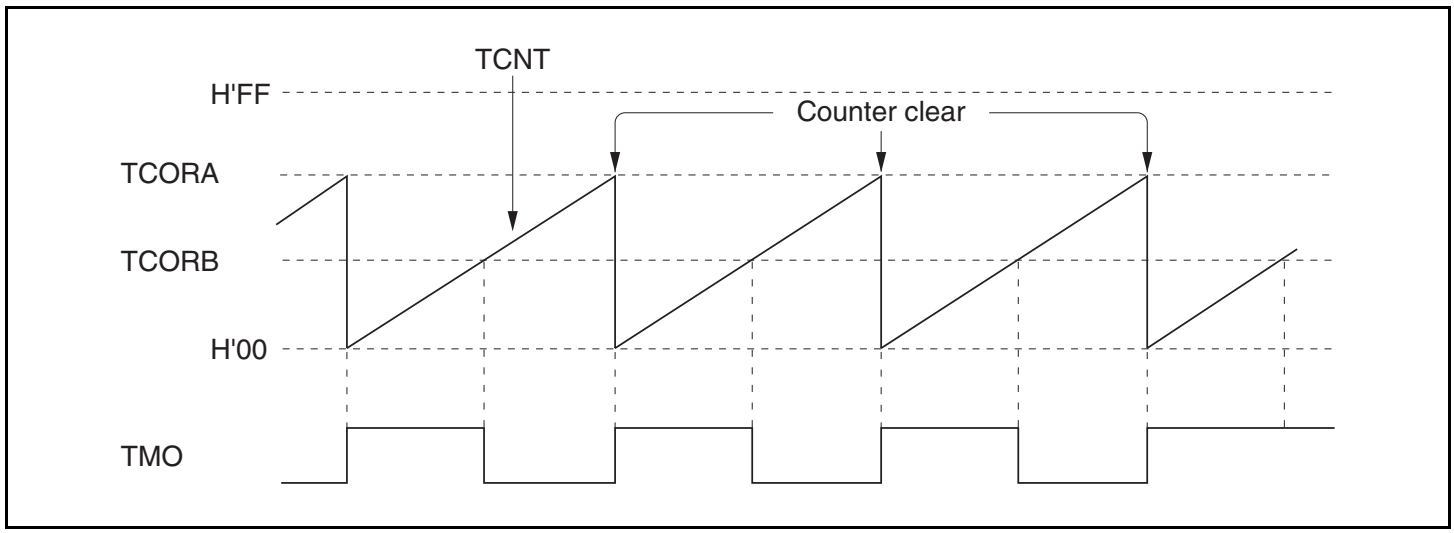


Figure 13.2 Example of Pulse Output

13.4.2 Reset Input

Figure 13.3 shows an example in which the 8-bit timer is used to generate a pulse output with a selected delay in response to the TMRI input. The control bits are set as follows:

[1] The CCLR0 bit in TCR is set to 1 and the TMRIS bit in TCCR is set to 1 so that TCNT is cleared at the high level of the TMRI input.

[2] In TCSR, bits OS3 to OS0 are set 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 output of pulses whose delay from the TMRI input is determined by TCORA and the pulse width determined by (TCORB – TCORA).

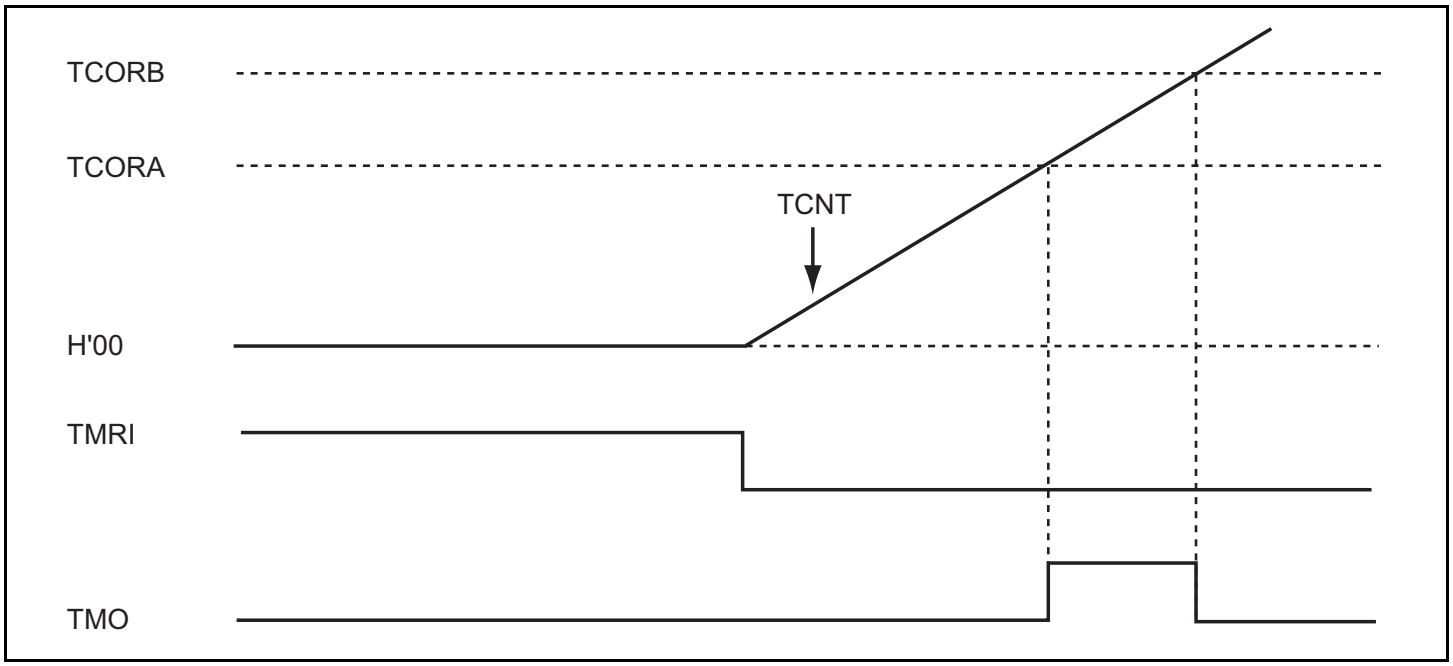


Figure 13.3 Example of Reset Input

13.5 Operation Timing

13.5.1 TCNT Incrementation Timing

Figure 13.4 shows the count timing for internal clock input. Figure 13.5 shows the count timing for external clock signal. Note that the external clock pulse width must be at least 1.5 states for incrementation at a single edge, and at least 2.5 states for incrementation at both edges. The counter will not increment correctly if the pulse width is less than these values.

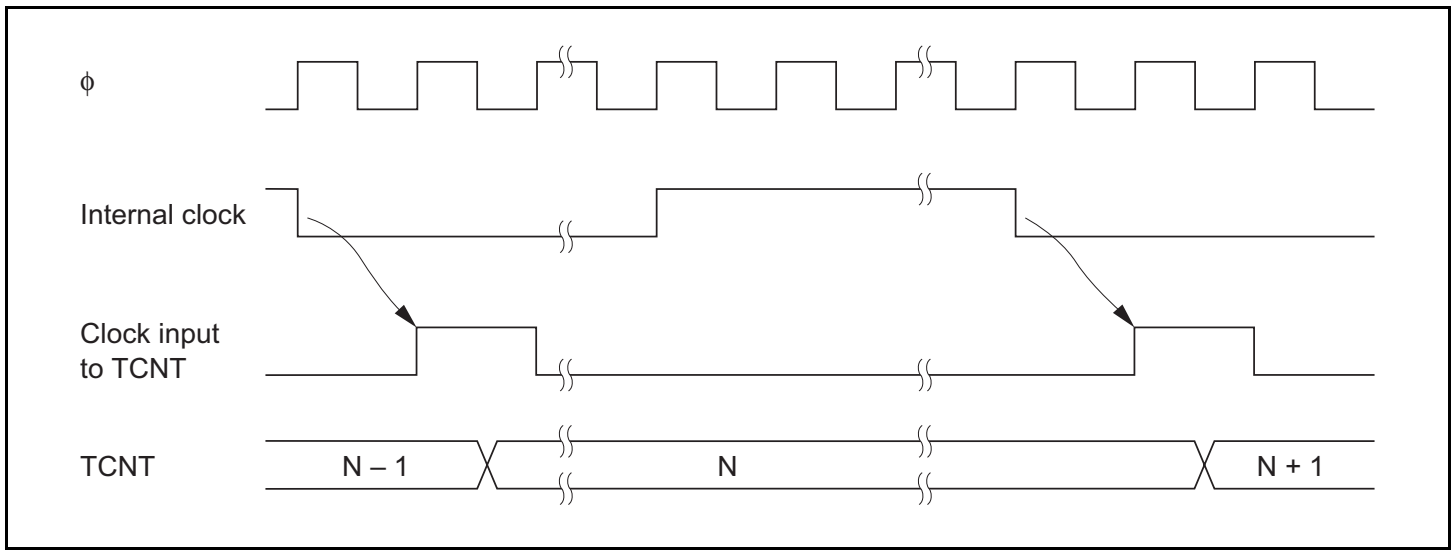


Figure 13.4 Count Timing for Internal Clock Input

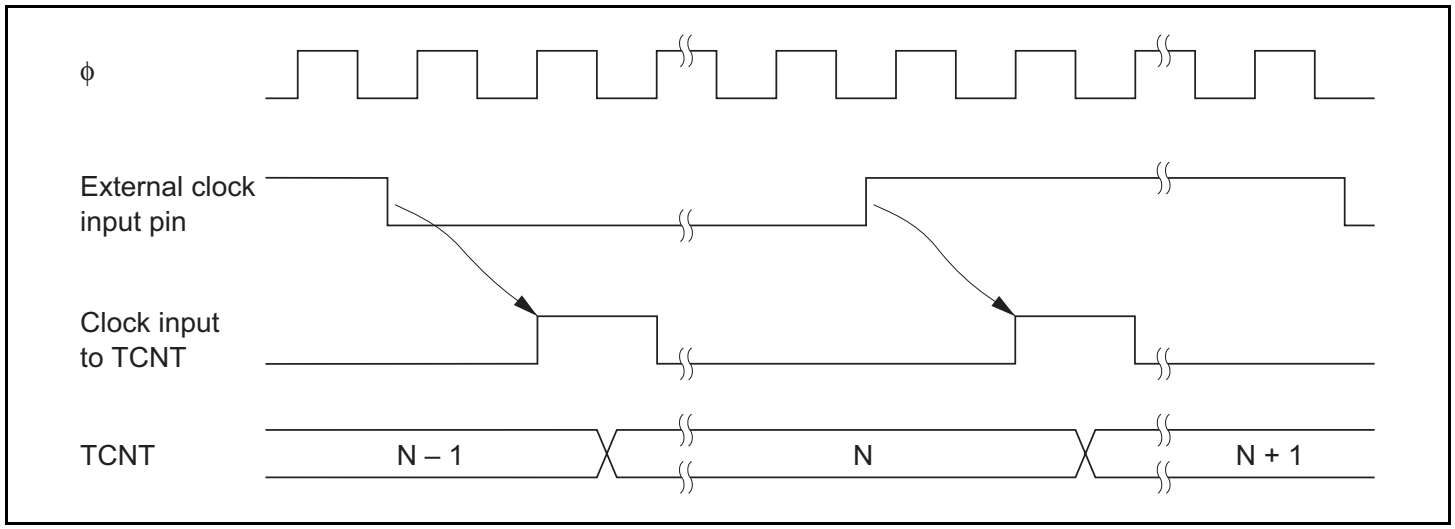


Figure 13.5 Count Timing for External Clock Input

13.5.2 Timing of CMFA and CMFB Setting when Compare-Match Occurs

The CMFA and CMFB flags in TCSR are set to 1 by a compare match signal generated when the TCOR and TCNT values match. The compare match signal is generated at the last state in which the match is true, just before the timer counter is updated. Therefore, when TCOR and TCNT match, the compare match signal is not generated until the next incrementation clock input. Figure 13.6 shows this timing.

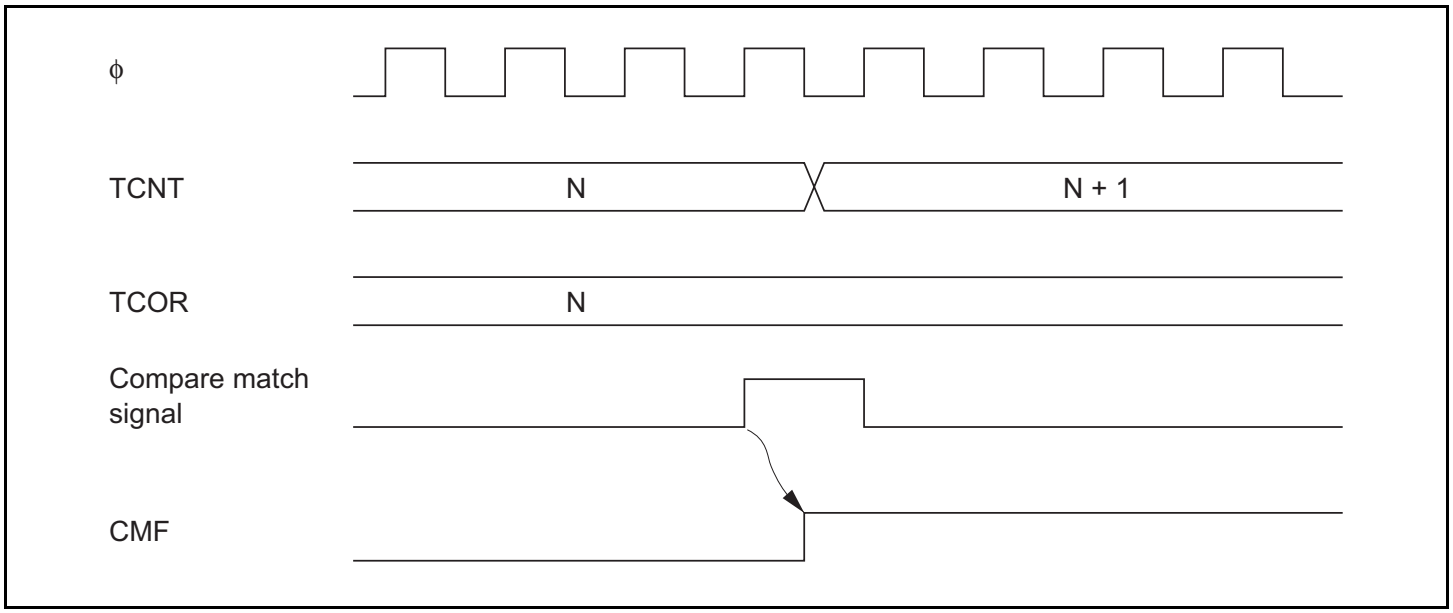


Figure 13.6 Timing of CMF Setting

13.5.3 Timing of Timer Output when Compare-Match Occurs

When compare match A or B occurs, the timer output changes as specified by bits OS3 to OS0 in TCSR. Figure 13.7 shows the timing when the output is set to toggle at compare match A.

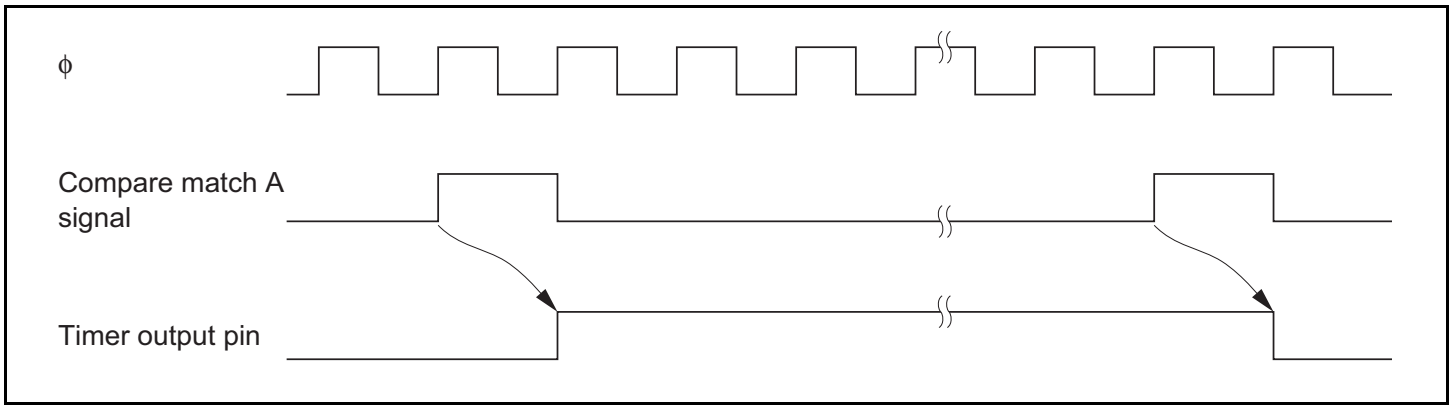


Figure 13.7 Timing of Timer Output

13.5.4 Timing of Compare Match Clear

TCNT is cleared when compare match A or B occurs, depending on the settings of the CCLR1 and CCLR0 bits in TCR and the TMRIS bit in TCCR. Figure 13.8 shows the timing of this operation.

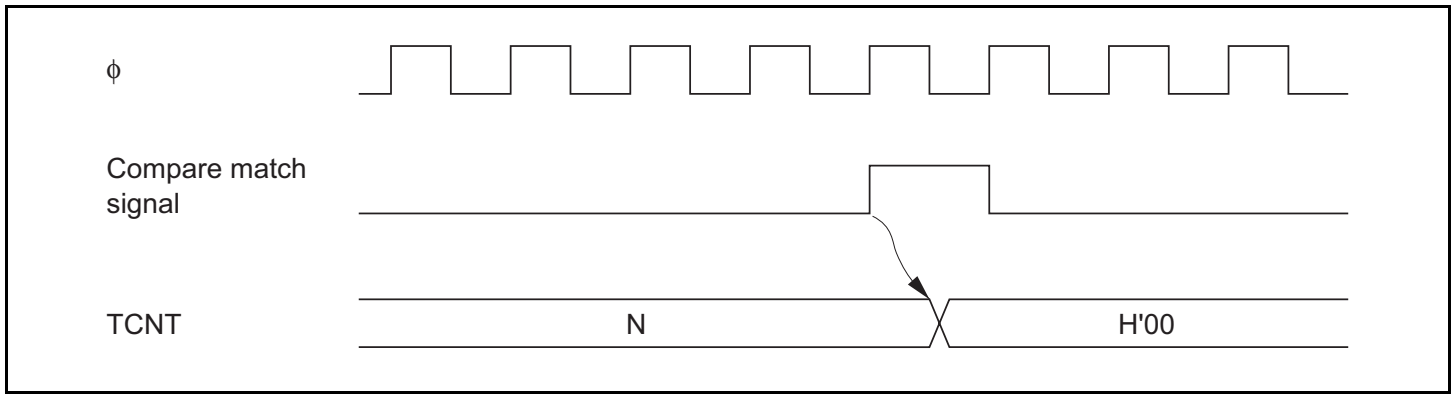


Figure 13.8 Timing of Compare Match Clear

13.5.5 Timing of TCNT External Reset

TCNT is cleared at the rising edge, falling edge, low level, or high level of an external reset input, depending on the settings of the CCLR1 and CCLR0 bits in TCR and the TMRIS bit in TCCR. The clear pulse width must be at least 1.5 states for a single edge and at least 2.5 states for both edges. Figure 13.9 shows the timing of this operation.

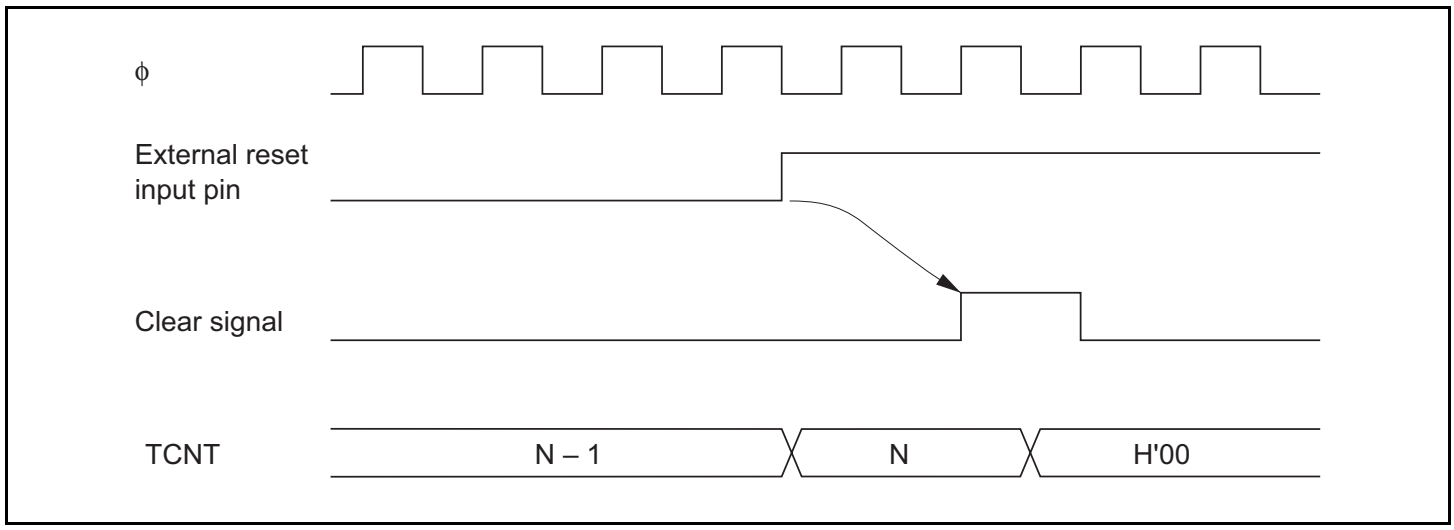


Figure 13.9 Timing of Clearance by External Reset

13.5.6 Timing of Overflow Flag (OVF) Setting

The OVF in TCSR is set to 1 when TCNT overflows (changes from H'FF to H'00). Figure 13.10 shows the timing of this operation.

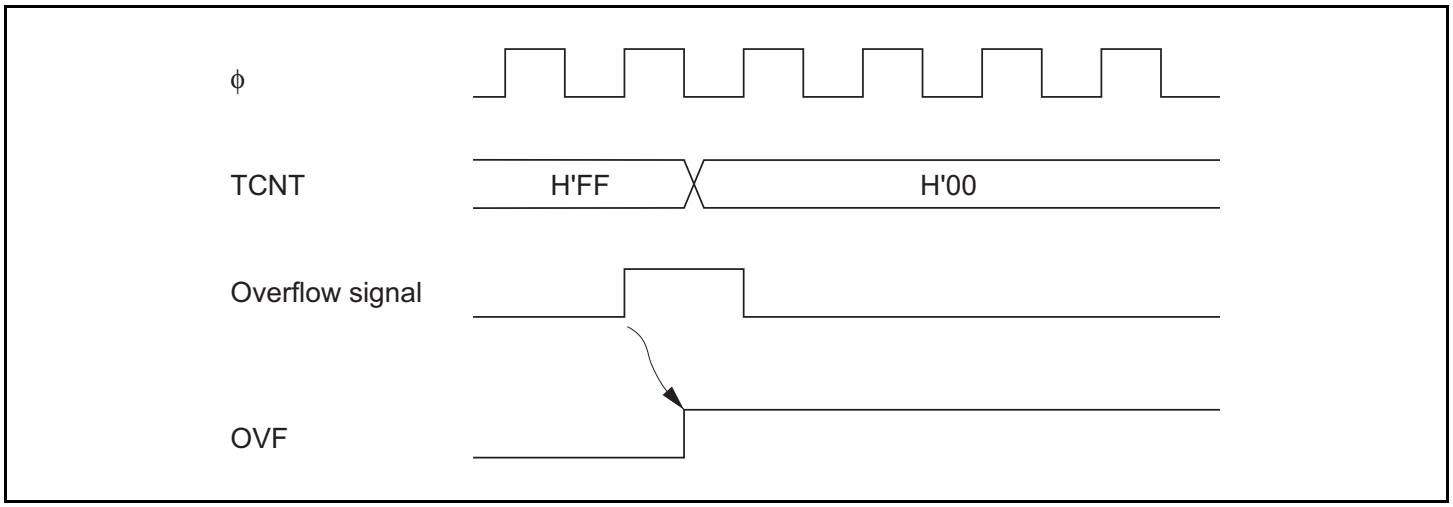


Figure 13.10 Timing of OVF Setting

13.6 Operation with Cascaded Connection

If bits CKS2 to CKS0 in either TCR_0 or TCR_1 are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, a single 16-bit timer could be used (16-bit counter mode) or compare matches of the 8-bit channel 0 could be counted by the timer of channel 1 (compare match count mode). In this case, the timer operates as below.

13.6.1 16-Bit Counter Mode

When bits CKS2 to CKS0 in TCR_0 are set to B'100, the timer functions as a single 16-bit timer with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bits.

[1] Setting of compare match flags

- The CMF flag in TCSR_0 is set to 1 when a 16-bit compare match event occurs.
- The CMF flag in TCSR_1 is set to 1 when a lower 8-bit compare match event occurs.

[2] Counter clear specification

- If the CCLR1 and CCLR0 bits in TCR_0 have been set for counter clear at compare match, the 16-bit counters (TCNT_0 and TCNT_1 together) are cleared when a 16-bit compare match event occurs. The 16-bit counters (TCNT0 and TCNT1 together) are cleared even if counter clear by the TMRI0 pin has also been set.
- The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bits cannot be cleared independently.

[3] Pin output

- Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR_0 is in accordance with the 16-bit compare match conditions.
- Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR_1 is in accordance with the lower 8-bit compare match conditions.

13.6.2 Compare Match Count Mode

When bits CKS2 to CKS0 in TCR_1 are B'100, TCNT_1 counts compare match A’s for channel 0. Channels 0 and 1 are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clear are in accordance with the settings for each channel.

13.7 Interrupt Sources

13.7.1 Interrupt Sources and DTC Activation

There are three 8-bit timer interrupt sources: CMIA, CMIB, and OVI. Their relative priorities are shown in table 13.4. Each interrupt source is set as enabled or disabled by the corresponding interrupt enable bit in TCR or TCSR, and independent interrupt requests are sent for each to the interrupt controller. It is also possible to activate the DTC by means of CMIA and CMIB interrupts.

Table 13.4 8-Bit Timer Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority
CMIA0	TCORA_0 compare match	CMFA	Possible	High
CMIB0	TCORB_0 compare match	CMFB	Possible	↑
OVI0	TCNT_0 overflow	OVF	Not possible	Low
CMIA1	TCORA_1 compare match	CMFA	Possible	High
CMIB1	TCORB_1 compare match	CMFB	Possible	↑
OVI1	TCNT_1 overflow	OVF	Not possible	Low

13.7.2 A/D Converter Activation

The A/D converter can be activated only by TMR_0 compare match A. If the ADTE bit in TCSR0 is set to 1 when the CMFA flag is set to 1 by the occurrence of TMR_0 compare match A, a request to start A/D conversion is sent to the A/D converter. If the 8-bit timer conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

13.8 Usage Notes

13.8.1 Contention between TCNT Write and Clear

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the clear takes priority, so that the counter is cleared and the write is not performed. Figure 13.11 shows this operation.

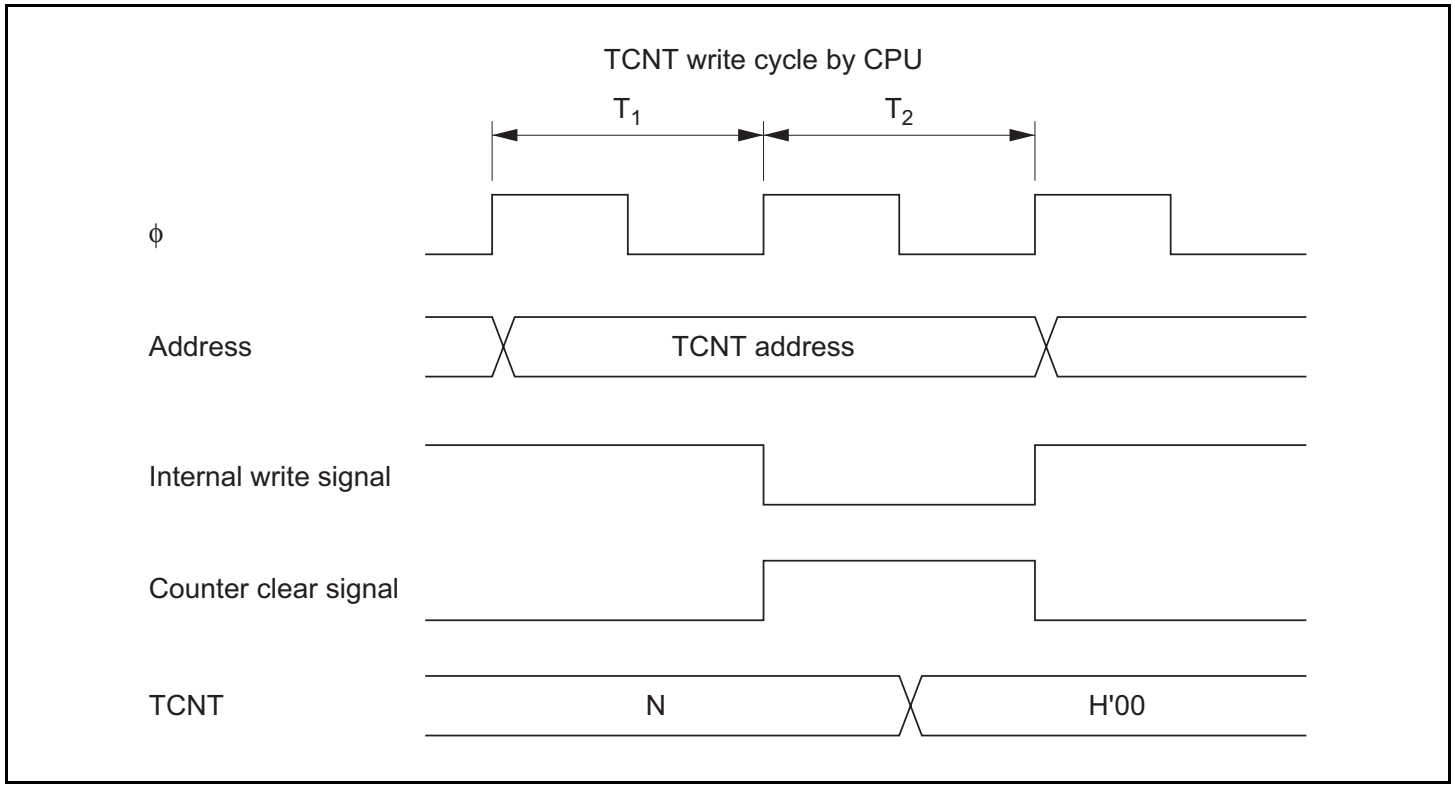


Figure 13.11 Contention between TCNT Write and Clear

13.8.2 Contention between TCNT Write and Increment

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the write takes priority and the counter is not incremented. Figure 13.12 shows this operation.

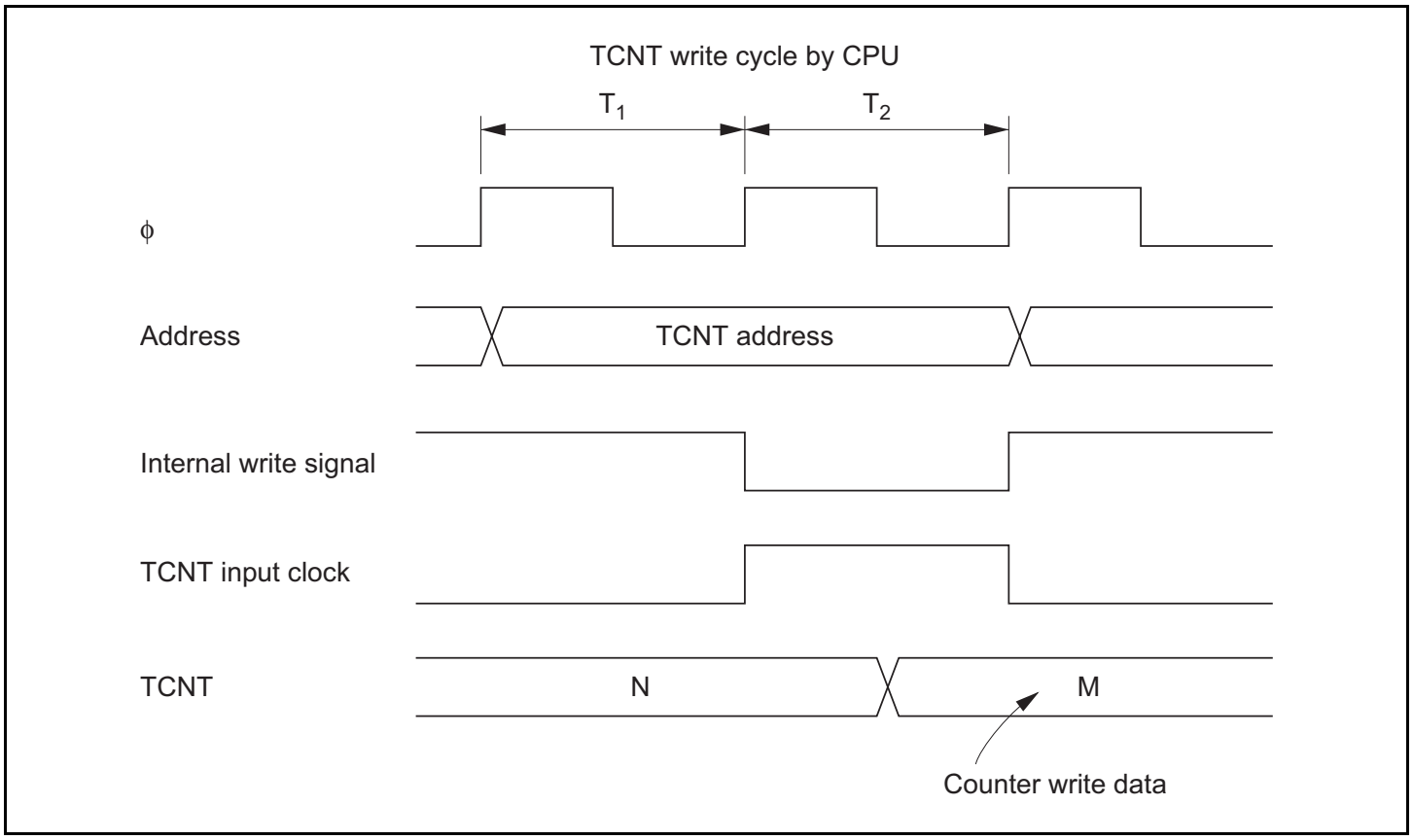


Figure 13.12 Contention between TCNT Write and Increment

13.8.3 Contention between TCOR Write and Compare Match

During the T_2 state of a TCOR write cycle, the TCOR write has priority and the compare match signal is inhibited even if a compare match event occurs as shown in figure 13.13.

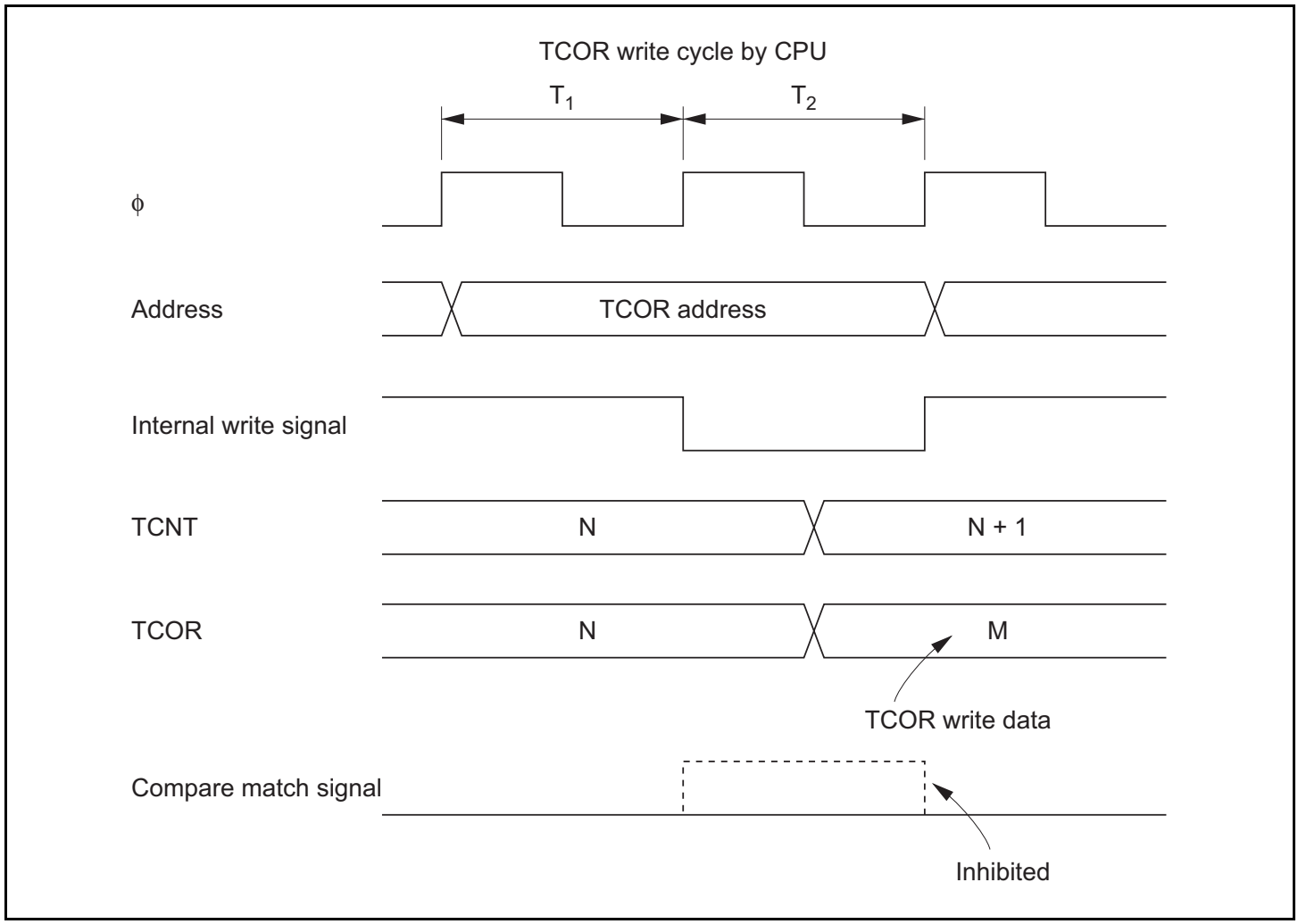


Figure 13.13 Contention between TCOR Write and Compare Match

13.8.4 Contention between Compare Matches A and B

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output statuses set for compare match A and compare match B, as shown in table 13.5.

Table 13.5 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1 output	↑
0 output	
No change	Low

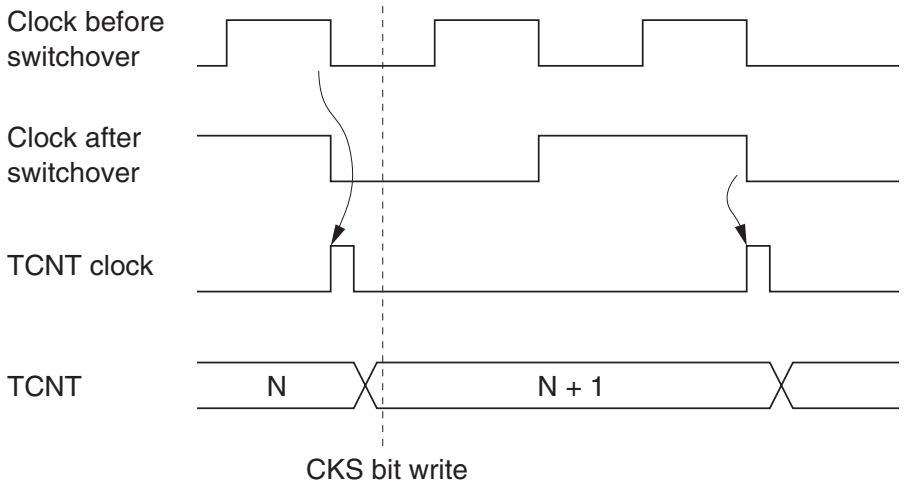
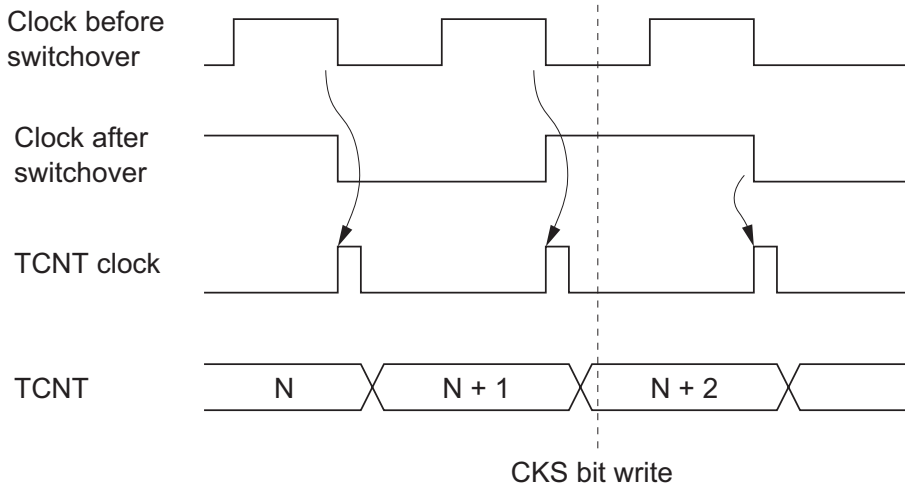
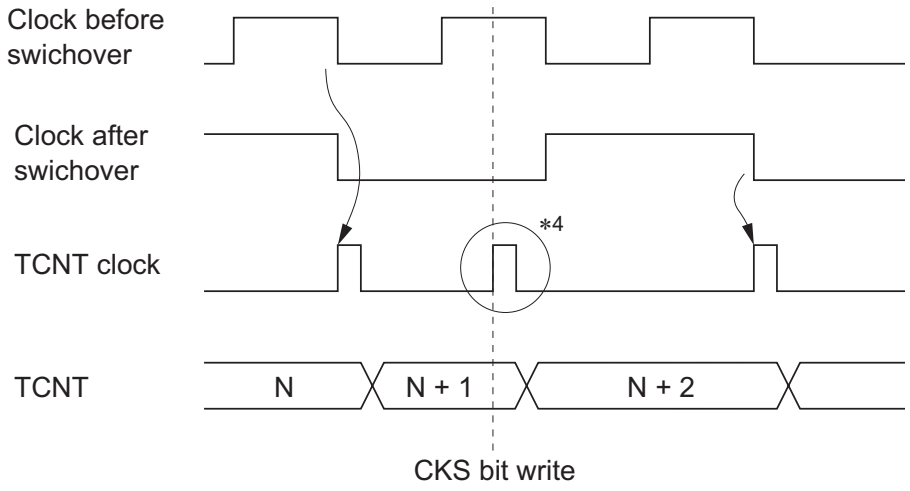
13.8.5 Switching of Internal Clocks and TCNT Operation

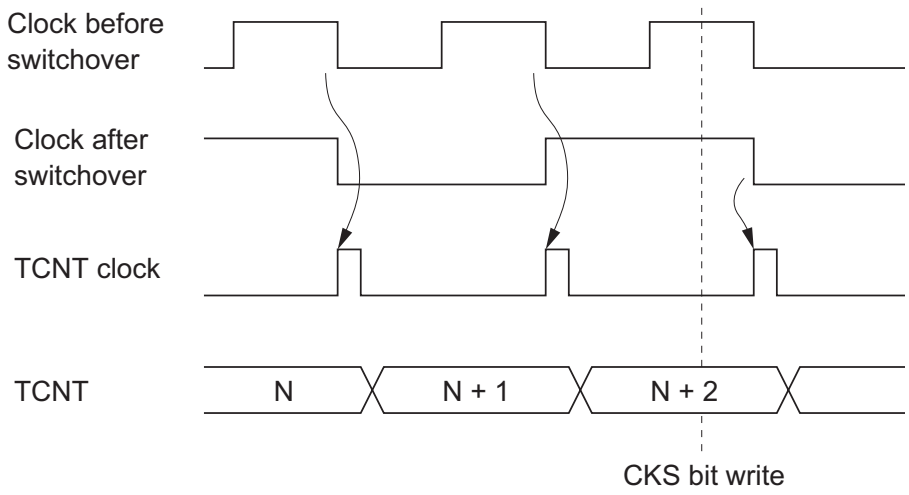
TCNT may increment erroneously when the internal clock is switched over. Table 13.6 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1, CKS0, ICKS1, and ICKS0 bits) and the TCNT operation.

When the TCNT clock is generated from an internal clock, the rising edge or falling edge of the internal clock pulse is detected. Therefore, when the falling edge is selected, if clock switching causes a change from high to low level, as shown in case 3 in table 13.6, a TCNT clock pulse is generated and the TCNT incremented on the assumption that the switchover is a falling edge. This is the same as when the rising edge is selected.

The erroneous incrementation can also happen when switching between the rising edge and falling edge of an internal clock or switching between internal and external clocks.

Table 13.6 Switching of Internal Clock and TCNT Operation

No.	Timing of Switchover by Means of Modifying CKS1, CKS0, ICKS1, and ICKS0 Bits	TCNT Clock Operation
1	Switching from low to low*1	
2	Switching from low to high*2	
3	Switching from high to low*3	

No.	Timing of Switchover by Means of Modifying CKS1, CKS0, ICKS1, and ICKS0 Bits	TCNT Clock Operation
4	Switching from high to high	

Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated on the assumption that the switchover is a falling edge; TCNT is incremented.

13.8.6 Mode Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, input clocks for TCNT_0 and TCNT_1 are not generated, and the counter stops. Do not specify 16-bit counter and compare match count modes simultaneously.

13.8.7 Module Stop Mode Setting

Operation of the TMR can be disabled or enabled using the module stop control register. The initial setting is for operation of the TMR to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 25, Power-Down Modes.

13.8.8 Interrupts in 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 source. Interrupts should therefore be disabled before entering module stop mode.

Section 14 Watchdog Timer (WDT)

The watchdog timer (WDT) is an 8-bit timer that outputs an overflow signal ($\overline{\text{WDTOVF}}$) if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow. At the same time, the WDT can also generate an internal reset signal.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

The block diagram of the WDT is shown in figure 14.1.

14.1 Features

- Selectable from eight counter input clocks
- Switchable between watchdog timer mode and interval timer mode

Watchdog Timer Mode

- If the counter overflows, the WDT outputs $\overline{\text{WDTOVF}}$. It is possible to select whether or not the entire chip is reset at the same time.

Interval Timer Mode

- If the counter overflows, the WDT generates an interval timer interrupt (WOVI).

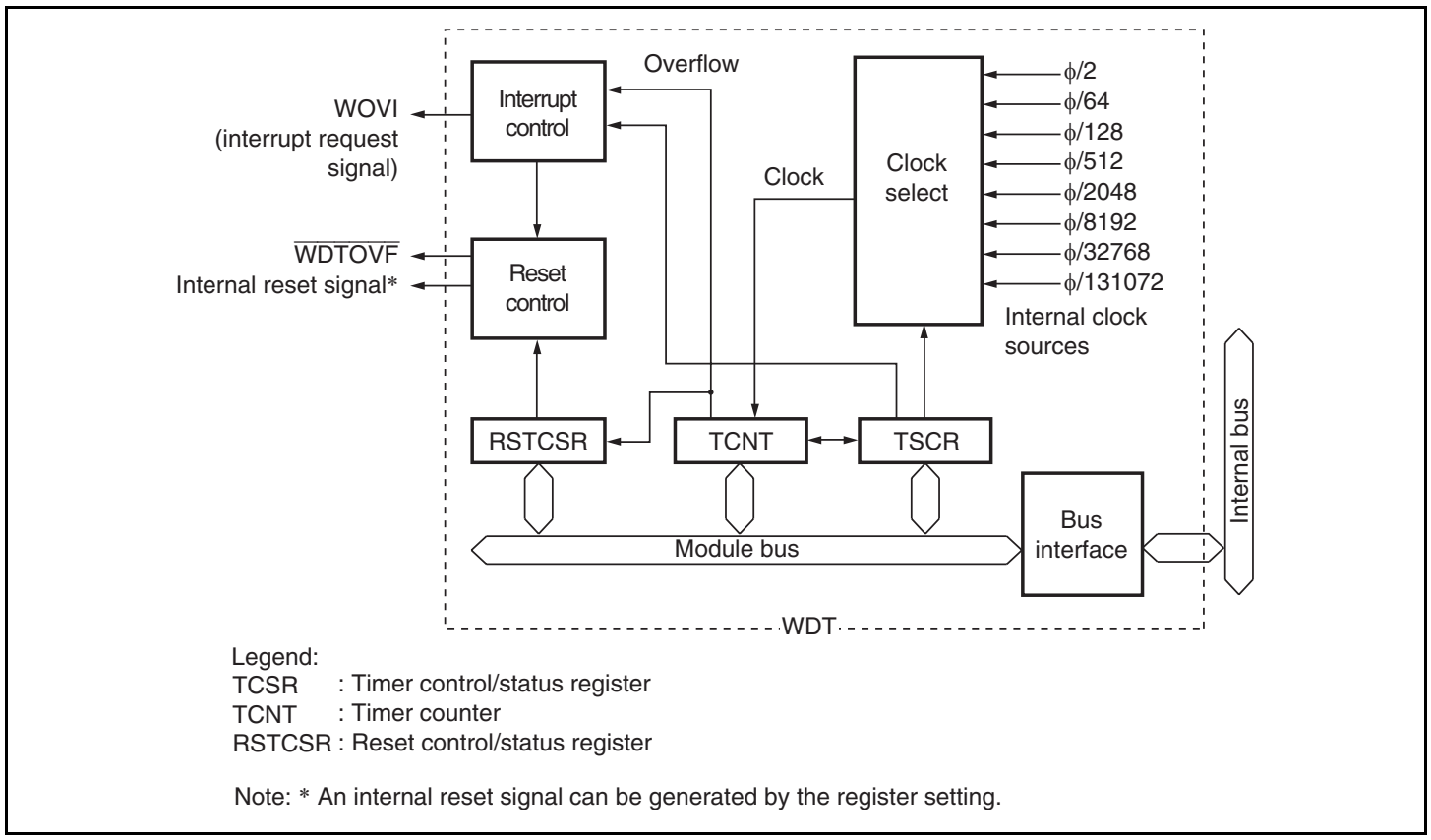


Figure 14.1 Block Diagram of WDT

14.2 Input/Output Pin

Table 14.1 shows the WDT pin configuration.

Table 14.1 Pin Configuration

Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs counter overflow signal in watchdog timer mode

14.3 Register Descriptions

The WDT has the following three registers. To prevent accidental overwriting, TCSR, TCNT, and RSTCSR have to be written to in a method different from normal registers. For details, refer to section 14.6.1, Notes on Register Access.

- Timer counter (TCNT)
- Timer control/status register (TCSR)
- Reset control/status register (RSTCSR)

14.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT is initialized to H'00 when the TME bit in TCSR is cleared to 0.

14.3.2 Timer Control/Status Register (TCSR)

TCSR selects the clock source to be input to TCNT, and the timer mode.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Overflow Flag Indicates that TCNT has overflowed in interval timer mode. Only a write of 0 is permitted, to clear the flag. [Setting condition] When TCNT overflows in interval timer mode (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing conditions] Cleared by reading TCSR when OVF = 1, then writing 0 to OVF

Bit	Bit Name	Initial Value	R/W	Description
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. 0: Interval timer mode When TCNT overflows, an interval timer interrupt (WOVI) is requested. 1: Watchdog timer mode When TCNT overflows, the $\overline{WDTOVF}$ signal is output.
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.
4, 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	CKS2	0	R/W	Clock Select 2 to 0 Selects the clock source to be input to TCNT. The overflow frequency for ϕ = 20 MHz is enclosed in parentheses. 000: Clock ϕ/2 (frequency: 25.6 μs) 001: Clock ϕ/64 (frequency: 819.2 μs) 010: Clock ϕ/128 (frequency: 1.6 ms) 011: Clock ϕ/512 (frequency: 6.6 ms) 100: Clock ϕ/2048 (frequency: 26.2 ms) 101: Clock ϕ/8192 (frequency: 104.9 ms) 110: Clock ϕ/32768 (frequency: 419.4 ms) 111: Clock ϕ/131072 (frequency: 1.68 s)
1	CKS1	0	R/W	
0	CKS0	0	R/W	

Note: * Only a write of 0 is permitted, to clear the flag.

14.3.3 Reset Control/Status Register (RSTCSR)

RSTCSR controls the generation of the internal reset signal when TCNT overflows, and selects the type of internal reset signal. RSTCSR is initialized to H'1F by a reset signal from the $\overline{\text{RES}}$ pin, but not by the WDT internal reset signal caused by overflows.

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)*	Watchdog Timer Overflow Flag This bit is set when TCNT overflows in watchdog timer mode. This bit cannot be set in interval timer mode, and only 0 can be written. [Setting condition] Set when TCNT overflows (changed from H'FF to H'00) in watchdog timer mode [Clearing condition] Cleared by reading RSTCSR when WOVF = 1, and then writing 0 to WOVF
6	RSTE	0	R/W	Reset Enable Specifies whether or not a reset signal is generated in the chip if TCNT overflows during watchdog timer operation. 0: Reset signal is not generated even if TCNT overflows (Though this LSI is not reset, TCNT and TCSR in WDT are reset) 1: Reset signal is generated if TCNT overflows
5	—	0	R/W	Reserved Can be read and written, but does not affect operation.
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Note: * Only a write of 0 is permitted, to clear the flag.

14.4 Operation

14.4.1 Watchdog Timer Mode

To use the WDT as a watchdog timer mode, set the WT/$\overline{IT}$ and TME bits in TCSR to 1.

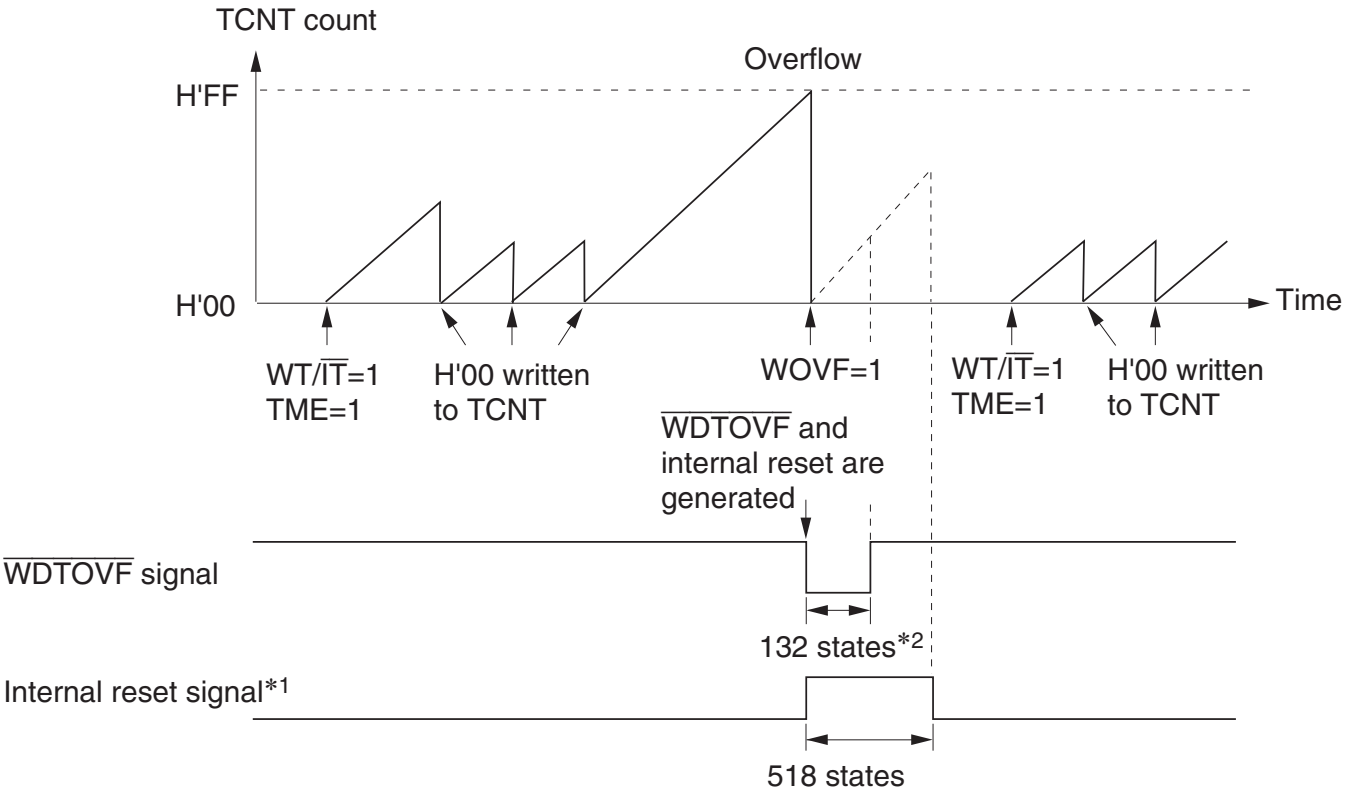
If TCNT overflows without being rewritten because of a system crash or other error, the $\overline{WDTOVF}$ signal is output.

This ensures that TCNT does not overflow while the system is operating normally. Software must prevent TCNT overflows by rewriting the TCNT value (normally be writing H'00) before overflow occurs. This $\overline{WDTOVF}$ signal can be used to reset the chip internally in watchdog timer mode.

If TCNT overflows when 1 is set in the RSTE bit in RSTCSR, a signal that resets this LSI internally is generated at the same time as the $\overline{WDTOVF}$ signal. If a reset caused by a signal input to the $\overline{RES}$ pin occurs at the same time as a reset caused by a WDT overflow, the $\overline{RES}$ pin reset has priority and the WOVF bit in RSTCSR is cleared to 0.

The $\overline{WDTOVF}$ signal is output for 132 states when RSTE = 1, and for 130 states when RSTE = 0. The internal reset signal is output for 518 states.

When TCNT overflows in watchdog timer mode, the WOVF bit in RSTCSR is set to 1. If TCNT overflows when 1 is set in the RSTE bit in RSTCSR, an internal reset signal is generated to the entire chip.



Notes: 1. If TCNT overflows when the RSTE bit is set to 1, an internal reset signal is generated.
2. 130 states when the RSTE bit is cleared to 0.

Figure 14.2 Operation in Watchdog Timer Mode

14.4.2 Interval Timer Mode

To use the WDT as an interval timer, set the WT/$\overline{IT}$ bit to 0 and TME bit in TCSR to 1.

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is generated each time the TCNT overflows. Therefore, an interrupt can be generated at intervals.

When the TCNT overflows in interval timer mode, an interval timer interrupt (WOVI) is requested at the same time the OVF bit in the TCSR is set to 1.

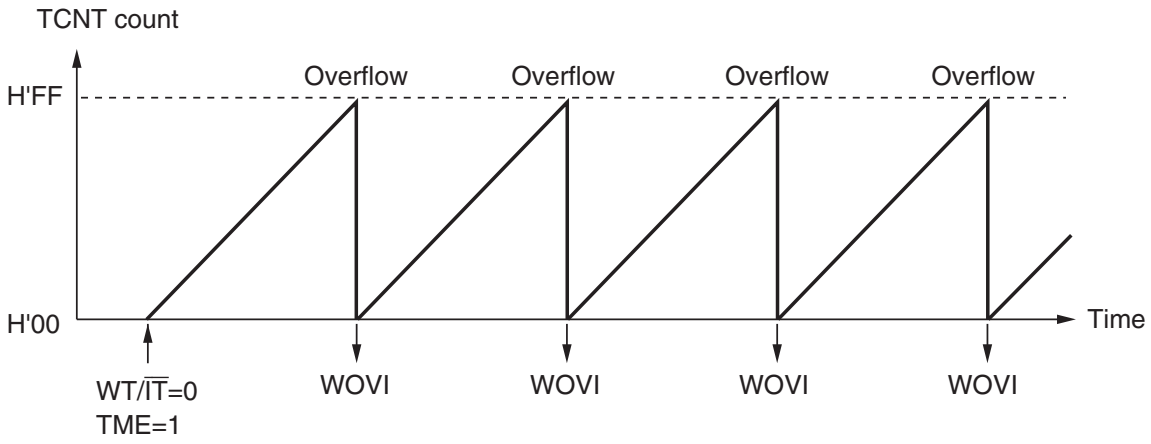


Figure 14.3 Operation in Interval Timer Mode

14.5 Interrupt Source

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. OVF must be cleared to 0 in the interrupt handling routine.

Table 14.2 WDT Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation
WOVI	TCNT overflow	OVF	Impossible

14.6 Usage Notes

14.6.1 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in being more difficult to write to. The procedures for writing to and reading these registers are given below.

(1) Writing to TCNT, TCSR, and RSTCSR

TCNT and TCSR must be written to by a word transfer instruction. They cannot be written to by a byte transfer instruction.

TCNT and TCSR both have the same write address. Therefore, satisfy the relative condition shown in figure 14.4 to write to TCNT or TCSR. The transfer instruction writes the lower byte data to TCNT or TCSR according to the satisfied condition.

To write to RSTCSR, execute a word transfer instruction for address H'FFBE. A byte transfer instruction cannot perform writing to RSTCSR.

The method of writing 0 to the WOVF bit differs from that of writing to the RSTE bit. To write 0 to the WOVF bit, satisfy the lower condition shown in figure 14.4.

If satisfied, the transfer instruction clears the WOVF bit to 0, but has no effect on the RSTE bit. To write to the RSTE bit, satisfy the above condition shown in figure 14.4. If satisfied, the transfer instruction writes the value in bit 6 of the lower byte into the RSTE bit, but has no effect on the WOVF bit.

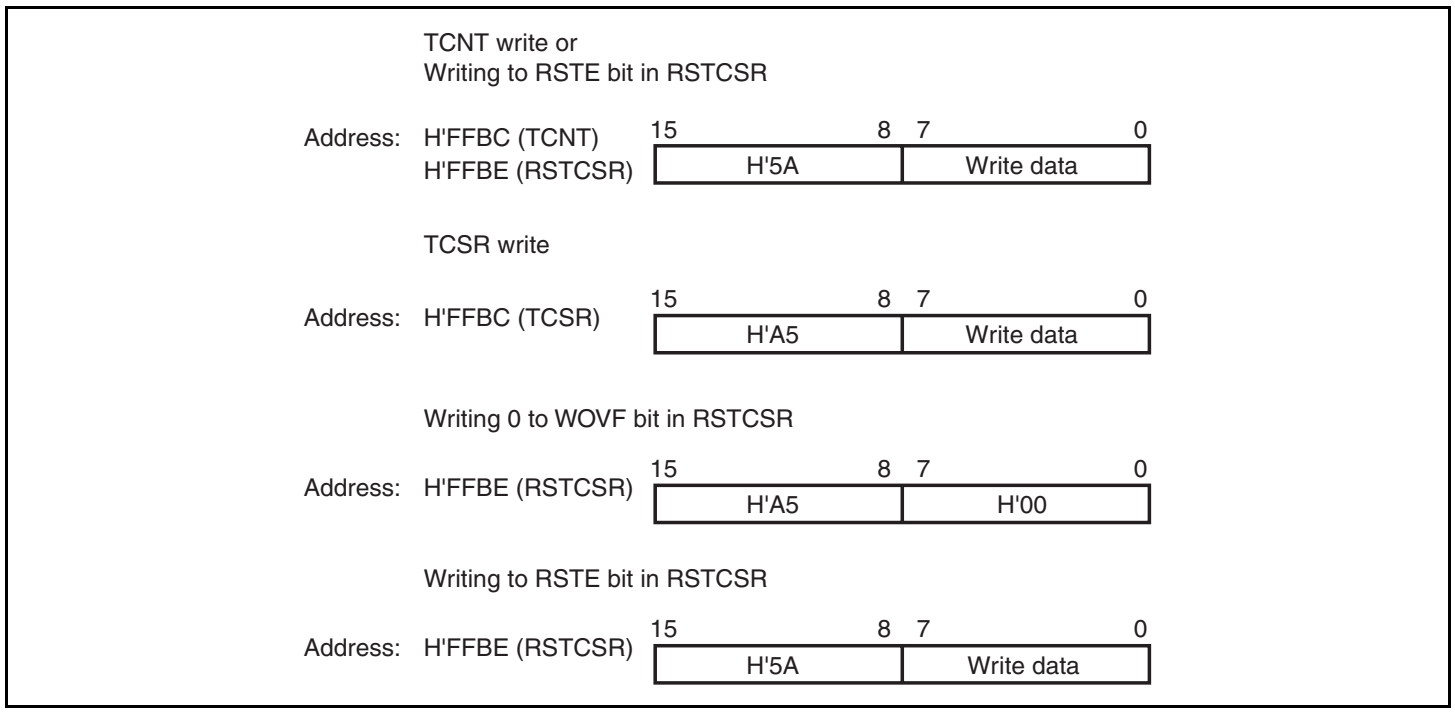


Figure 14.4 Writing to TCNT, TCSR, and RSTCSR

(2) Reading TCNT, TCSR, and RSTCSR

These registers are read in the same way as other registers. The read addresses are H'FFBC for TCSR, H'FFBD for TCNT, and H'FFBF for RSTCSR.

14.6.2 Contention between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the next cycle after the T_2 state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 14.5 shows this operation.

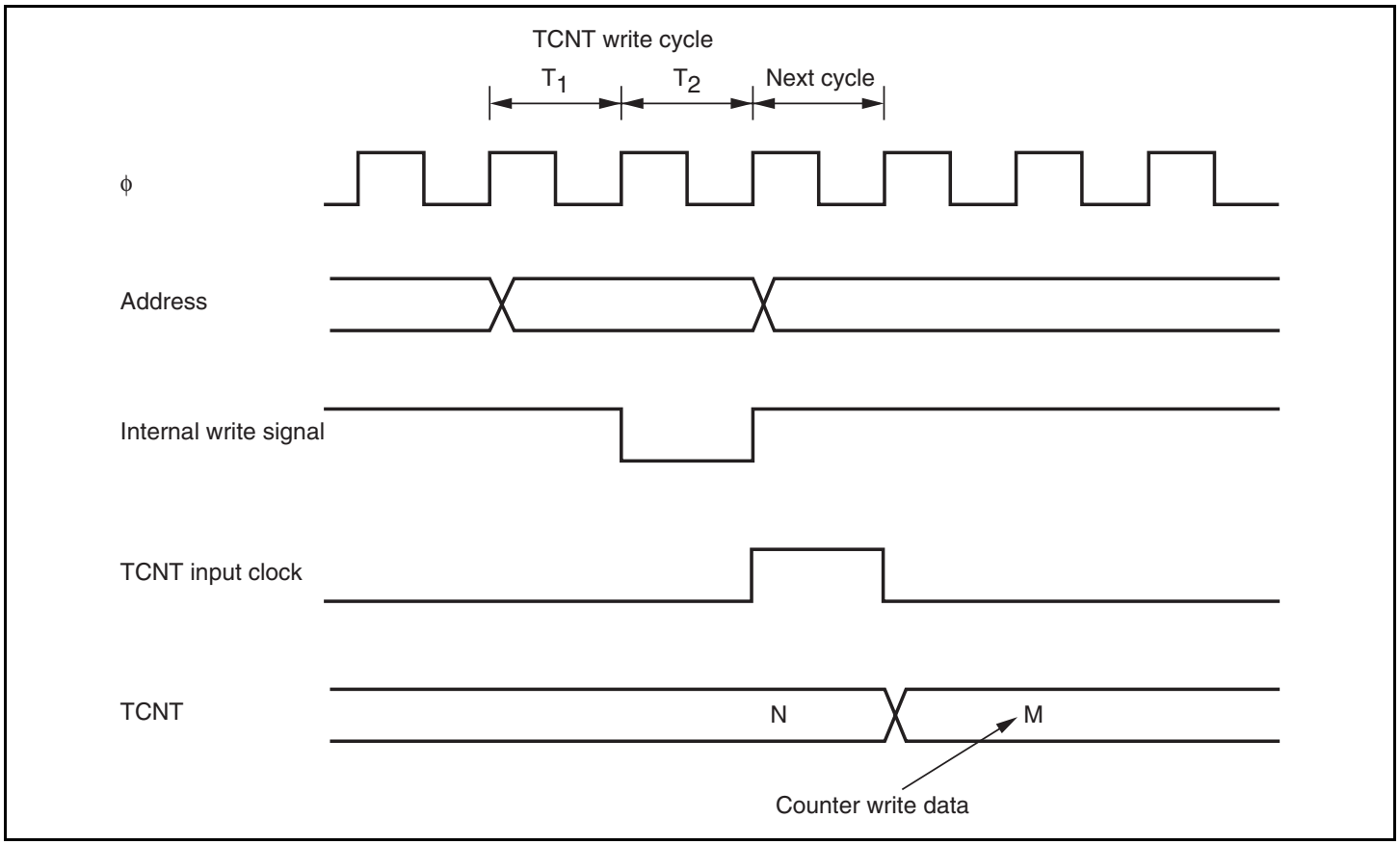


Figure 14.5 Contention between TCNT Write and Increment

14.6.3 Changing Value of CKS2 to CKS0

If bits CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before changing the value of bits CKS2 to CKS0.

14.6.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer, while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

14.6.5 Internal Reset in Watchdog Timer Mode

This LSI is not reset internally if TCNT overflows while the RSTE bit is cleared to 0 during watchdog timer mode operation, but TCNT and TCSR of the WDT are reset.

TCNT, TCSR, and RSTCR cannot be written to while the $\overline{\text{WDTOVF}}$ signal is low. Also note that a read of the WOVF flag is not recognized during this period. To clear the WOVF flag, therefore, read TCSR after the $\overline{\text{WDTOVF}}$ signal goes high, then write 0 to the WOVF flag.

14.6.6 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ output signal is input to the $\overline{\text{RES}}$ pin, the chip will not be initialized correctly. Make sure 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 the circuit shown in figure 14.6.

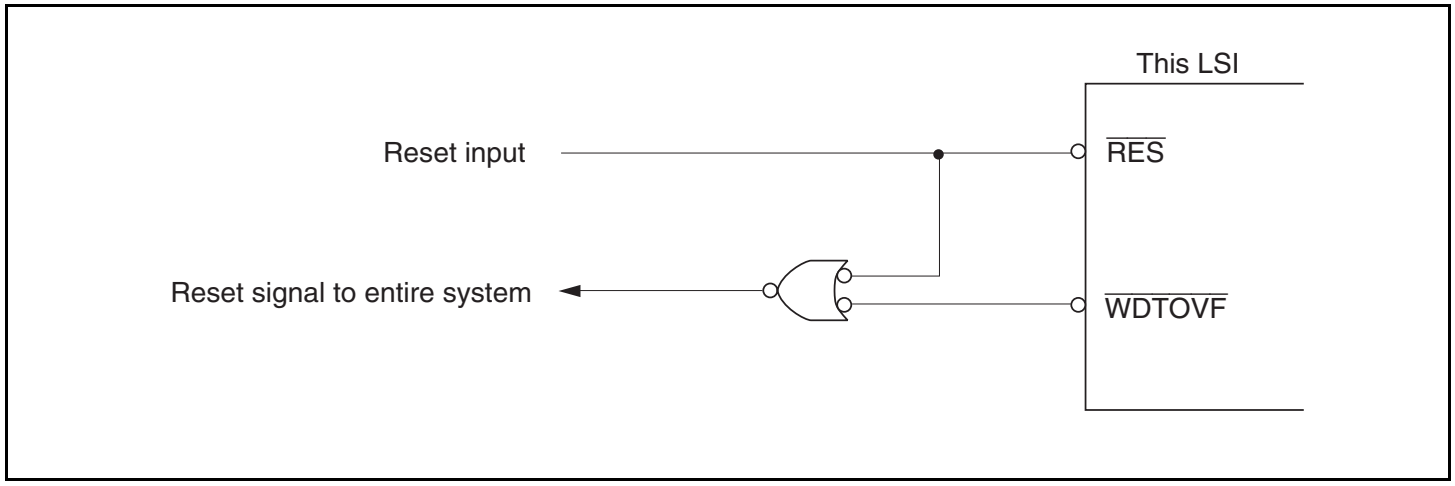


Figure 14.6 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example)

Section 15 Serial Communication Interface (SCI, IrDA)

This LSI has five independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function) in asynchronous mode. The SCI also supports an IC card (Smart Card) interface conforming to ISO/IEC 7816-3 (Identification Card) as an asynchronous serial communication interface extension function. One of the five SCI channels (SCI_0) can generate an IrDA communication waveform conforming to IrDA specification version 1.0.

Figure 15.1 shows a block diagram of the SCI.

15.1 Features

- Choice of asynchronous or clocked synchronous serial communication mode
- Full-duplex communication capability

  The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously. Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected

  External clock can be selected as a transfer clock source (except for in Smart Card interface mode).
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode 7-bit data)
- Four interrupt sources

  Four interrupt sources — transmit-end, transmit-data-empty, receive-data-full, and receive error — that can issue requests. The transmit-data-empty interrupt and receive data full interrupts can activate the data transfer controller (DTC) or DMA controller (DMAC).
- Module stop mode can be set

Asynchronous Mode

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors
- Break detection: Break can be detected by reading the RxD pin level directly in case of a framing error
- Average transfer rate generator (SCI_2 only):
  115.152 or 460.606 kbps at 10.667-MHz operation
  115.196, 460.784, or 720 kbps at 16-MHz operation
  720 kbps at 32-MHz operation

Clocked Synchronous Mode

- Data length: 8 bits
- Receive error detection: Overrun errors detected

Smart Card Interface

- Automatic transmission of error signal (parity error) in receive mode
- Error signal detection and automatic data retransmission in transmit mode
- Direct convention and inverse convention both supported

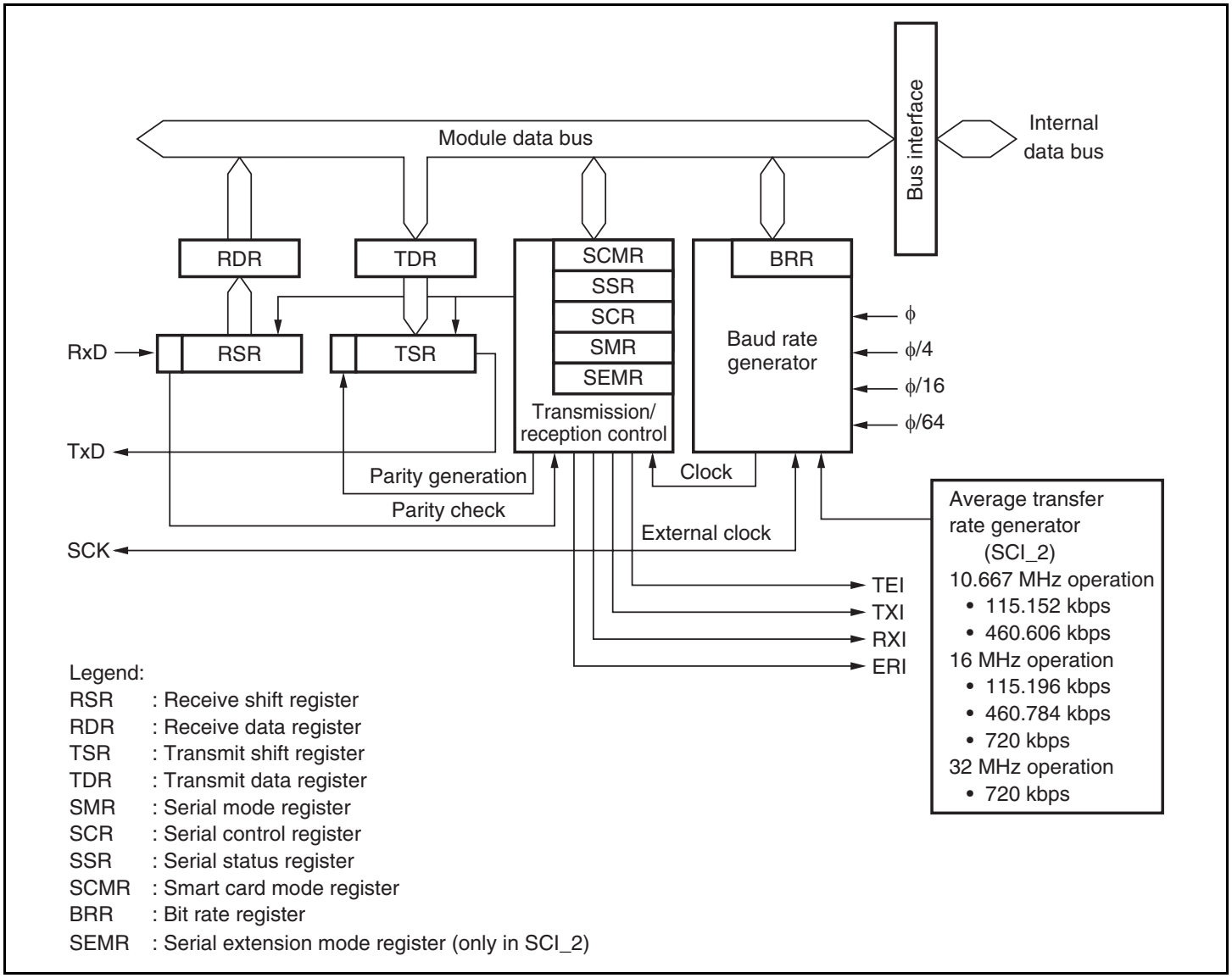


Figure 15.1 Block Diagram of SCI

15.2 Input/Output Pins

Table 15.1 shows the pin configuration of the serial communication interface.

Table 15.1 Pin Configuration

Channel	Pin Name*	I/O	Function
0	SCK0	I/O	Channel 0 clock input/output
	RxD0/IrRxD	Input	Channel 0 receive data input (normal/IrDA)
	TxD0/IrTxD	Output	Channel 0 transmit data output (normal/IrDA)
1	SCK1	I/O	Channel 1 clock input/output
	RxD1	Input	Channel 1 receive data input
	TxD1	Output	Channel 1 transmit data output
2	SCK2	I/O	Channel 2 clock input/output
	RxD2	Input	Channel 2 receive data input
	TxD2	Output	Channel 2 transmit data output
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

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting the channel designation.

15.3 Register Descriptions

The SCI has the following registers. The serial mode register (SMR), serial status register (SSR), and serial control register (SCR) are described separately for normal serial communication interface mode and Smart Card interface mode because their bit functions partially differ.

- Receive shift register_0 (RSR_0)
- Transmit shift register_0 (TSR_0)
- Receive data register_0 (RDR_0)
- Transmit data register_0 (TDR_0)
- Serial mode register_0 (SMR_0)
- Serial control register_0 (SCR_0)
- Serial status register_0 (SSR_0)
- Smart card mode register_0 (SCMR_0)
- Bit rate register_0 (BRR_0)
- IrDA control register_0 (IrCR_0)
- Receive shift register_1 (RSR_1)
- Transmit shift register_1 (TSR_1)
- Receive data register_1 (RDR_1)
- Transmit data register_1 (TDR_1)
- Serial mode register_1 (SMR_1)
- Serial control register_1 (SCR_1)
- Serial status register_1 (SSR_1)
- Smart card mode register_1 (SCMR_1)
- Bit rate register_1 (BRR_1)
- Receive shift register_2 (RSR_2)
- Transmit shift register_2 (TSR_2)
- Receive data register_2 (RDR_2)
- Transmit data register_2 (TDR_2)
- Serial mode register_2 (SMR_2)
- Serial control register_2 (SCR_2)
- Serial status register_2 (SSR_2)
- Smart card mode register_2 (SCMR_2)
- Bit rate register_2 (BRR_2)
- Serial extension mode register_2 (SEMR_2)

- 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)
- 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)

15.3.1 Receive Shift Register (RSR)

RSR is a shift register used to receive serial data that is input to the RxD pin and convert it into parallel data. When one byte of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

15.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores receive data. When the SCI has received one byte of serial data, it transfers the received serial data from RSR to RDR where it is stored. After this, RSR is receive-enabled. Since RSR and RDR function as a double buffer in this way, enables continuous receive operations to be performed. After confirming that the RDRF bit in SSR is set to 1, read RDR for only once. RDR cannot be written to by the CPU.

15.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structures of TDR and TSR enable continuous serial transmission. If the next transmit data has already been written to TDR during serial transmission, the SCI transfers the written data to TSR to continue transmission. Although TDR can be read or written to by the CPU at all times, to achieve reliable serial transmission, write transmit data to TDR for only once after confirming that the TDRE bit in SSR is set to 1.

15.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, then sends the data to the TxD pin starting. TSR cannot be directly accessed by the CPU.

15.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the on-chip baud rate generator clock source. Some bit functions of SMR differ in normal serial communication interface mode and Smart Card interface mode.

Normal Serial Communication Interface Mode (When SMIF bit in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	C/$\overline{A}$	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is fixed and the MSB (bit 7) of TDR is not transmitted in transmission. In clocked synchronous mode, a fixed data length of 8 bits is used.

Bit	Bit Name	Initial Value	R/W	Description
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. For a multiprocessor format, parity bit addition and checking are not performed regardless of the PE bit setting.
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous mode) Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked regardless of the STOP bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit character.
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynchronous mode) When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and O/$\overline{E}$ bit settings are invalid in multiprocessor mode.
1	CKS1	0	R/W	Clock Select 1 and 0:
0	CKS0	0	R/W	These bits select the clock source for the on-chip baud rate generator. 00: ϕ clock (n = 0) 01: ϕ/4 clock (n = 1) 10: ϕ/16 clock (n = 2) 11: ϕ/64 clock (n = 3) For the relation between the bit rate register setting and the baud rate, see section 15.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 15.3.9, Bit Rate Register (BRR)).

Smart Card Interface Mode (When SMIF bit in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	GSM Mode When this bit is set to 1, the SCI operates in GSM mode. In GSM mode, the timing of the TEND setting is advanced by 11.0 etu (Elementary Time Unit: the time for transfer of 1 bit), and clock output control mode addition is performed. For details, refer to section 15.7.8, Clock Output Control.
6	BLK	0	R/W	When this bit is set to 1, the SCI operates in block transfer mode. For details on block transfer mode, refer to section 15.7.3, Block Transfer Mode.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. In Smart Card interface mode, this bit must be set to 1.
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity. For details on setting this bit in Smart Card interface mode, refer to section 15.7.2, Data Format (Except for Block Transfer Mode).

Bit	Bit Name	Initial Value	R/W	Description
3	BCP1	0	R/W	Basic Clock Pulse 1 and 0
2	BCP0	0	R/W	These bits, in combination with the BCP2 bit in SCMR, select the number of basic clock cycles in a 1-bit transfer interval in Smart Card interface mode. BCP2 to BCP0 Settings: 000: 93 clock cycles (S = 93) 001: 128 clock cycles (S = 128) 010: 186 clock cycles (S = 186) 011: 512 clock cycles (S = 512) 100: 32 clock cycles (S = 32) (initial value) 101: 64 clock cycles (S = 64) 110: 372 clock cycles (S = 372) 111: 256 clock cycles (S = 256) For details, refer to section 15.7.4, Receive Data Sampling Timing and Reception Margin. S stands for the value of S in BRR (see section 15.3.9, Bit Rate Register (BRR)).
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the on-chip baud rate generator. 00: ϕ clock (n = 0) 01: ϕ/4 clock (n = 1) 10: ϕ/16 clock (n = 2) 11: ϕ/64 clock (n = 3) For the relation between the bit rate register setting and the baud rate, see section 15.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 15.3.9, Bit Rate Register (BRR)).

15.3.6 Serial Control Register (SCR)

SCR performs enabling or disabling of SCI transfer operations and interrupt requests, and selection of the transfer/receive clock source. For details on interrupt requests, refer to section 15.9, Interrupt Sources. Some bit functions of SCR differ in normal serial communication interface mode and Smart Card interface mode.

Normal Serial Communication Interface Mode (When SMIF bit in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, TXI interrupt request is enabled. TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag, then clearing it to 0, or clearing the TIE bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled. RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF flag, or the FER, PER, or ORER flag, then clearing the flag to 0, or by clearing the RIE bit to 0.
5	TE	0	R/W	Transmit Enable When this bit s set to 1, transmission is enabled. In this state, serial transmission is started when transmit data is written to TDR and the TDRE flag in SSR is cleared to 0. SMR setting must be performed to decide the transfer format before setting the TE bit to 1. The TDRE flag in SSR is fixed at 1 if transmission is disabled by clearing this bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in clocked synchronous mode. SMR setting must be performed to decide the transfer format before setting the RE bit to 1. Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and ORER status flags in SSR is prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, refer to section 15.5, Multiprocessor Communication Function. When receive data including MPB = 0 in SSR is received, receive data transfer from RSR to RDR, receive error detection, and setting of the RDRF, FER, and ORER flags in SSR , is not performed. When receive data including MPB = 1 is received, the MPB bit in SSR is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI and ERI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and ORER flag setting is enabled.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, TEI interrupt request is enabled. TEI cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0 and clearing the TEND flag to 0, or by clearing the TEIE bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	Selects the clock source and SCK pin function. Asynchronous mode 00: On-chip baud rate generator SCK pin functions as I/O port 01: On-chip baud rate generator (Outputs a clock of the same frequency as the bit rate from the SCK pin.) 1×: External clock (Inputs a clock with a frequency 16 times the bit rate from the SCK pin.) Clocked synchronous mode 0×: Internal clock (SCK pin functions as clock output) 1×: External clock (SCK pin functions as clock input)

Legend: ×: Don't care

Smart Card Interface Mode (When SMIF bit in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, TXI interrupt request is enabled. TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag, then clearing it to 0, or clearing the TIE bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled. RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF flag, or the FER, PER, or ORER flag, then clearing the flag to 0, or by clearing the RIE bit to 0.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled. In this state, serial transmission is started when transmit data is written to TDR and the TDRE flag in SSR is cleared to 0. SMR setting must be performed to decide the transfer format before setting the TE bit to 1. The TDRE flag in SSR is fixed at 1 if transmission is disabled by clearing this bit to 0.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in clocked synchronous mode. SMR setting must be performed to decide the transfer format before setting the RE bit to 1. Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode) Write 0 to this bit in Smart Card interface mode.

Bit	Bit Name	Initial Value	R/W	Description
2	TEIE	0	R/W	Transmit End Interrupt Enable Write 0 to this bit in Smart Card interface mode.
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	Enables or disables clock output from the SCK pin. The clock output can be dynamically switched in GSM mode. For details, refer to section 15.7.8, Clock Output Control. When the GM bit in SMR is 0: 00: Output disabled (SCK pin can be used as an I/O port pin) 01: Clock output 1×: Reserved When the GM bit in SMR is 1: 00: Output fixed low 01: Clock output 10: Output fixed high 11: Clock output

Legend: ×: Don't care

15.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER; they can only be cleared. Some bit functions of SSR differ in normal serial communication interface mode and Smart Card interface mode.

Normal Serial Communication Interface Mode (When SMIF bit in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR, and data writing to TDR is enabled. [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt request and transfers data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or DTC is activated by an RXI interrupt and transferred data from RDR The RDRF flag is not affected and retains its previous value when the RE bit in SCR is cleared to 0. Exercise care because if reception of the next data is completed while the RDRF flag is set to 1, an overrun error occurs and receive data will be lost.

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W)*	Overrun Error Indicates that an overrun error occurred while receiving and the reception has ended abnormally. [Setting condition] • When the next serial reception is completed while RDRF = 1 The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to ORER after reading ORER = 1 The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
4	FER	0	R/(W)*	Framing Error Indicates that a framing error occurred while receiving in asynchronous mode and the reception has ended abnormally. [Setting condition] • When the stop bit is 0 In 2-stop-bit mode, only the first stop bit is checked for a value of 0; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the FER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to FER after reading FER = 1 The FER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
3	PER	0	R/(W)*	Parity Error Indicates that a parity error occurred while receiving in asynchronous mode and the reception has ended abnormally. [Setting condition] • When a parity error is detected during reception If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to PER after reading PER = 1 The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR
1	MPB	0	R	Multiprocessor Bit MPB stores the multiprocessor bit in the receive data. When the RE bit in SCR is cleared to 0 its previous state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT sets the multiprocessor bit to be added to the transmit data.

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

Smart Card Interface Mode (When SMIF bit in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR, and data writing to TDR is enabled. [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt request and transfers data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or DTC is activated by an RXI interrupt and transferred data from RDR The RDRF flag is not affected and retains its previous value when the RE bit in SCR is cleared to 0. Exercise care because if reception of the next data is completed while the RDRF flag is set to 1, an overrun error occurs and receive data will be lost.

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W)*	Overrun Error Indicates that an overrun error occurred while receiving and the reception has ended abnormally. [Setting condition] • When the next serial reception is completed while RDRF = 1 The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to ORER after reading ORER = 1 The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
4	ERS	0	R/(W)*	Error Signal Status [Setting condition] • When the low level of the error signal is sampled [Clearing conditions] • When 0 is written to ERS after reading ERS = 1

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/(W)*	Parity Error Indicates that a parity error occurred while receiving in asynchronous mode and the reception has ended abnormally. [Setting condition] • When a parity error is detected during reception If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to PER after reading PER = 1 The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	Transmit End This bit is set to 1 when no error signal has been sent back from the receiving end and the next transmit data is ready to be transferred to TDR. [Setting conditions] • When the TE bit in SCR is 0 and the ERS bit is also 0 • If the ERS bit is 0 and the TDRE bit is 1 after the specified interval after transmission of 1-byte data Timing to set this bit differs according to the register settings. GM = 0, BLK = 0: 2.5 etu after transmission GM = 0, BLK = 1: 1.5 etu after transmission GM = 1, BLK = 0: 1.0 etu after transmission GM = 1, BLK = 1: 1.0 etu after transmission [Clearing conditions] • When 0 is written to TEND after reading TEND = 1 • When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR
1	MPB	0	R	Multiprocessor Bit This bit is not used in Smart Card interface mode.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in Smart Card interface mode.

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

15.3.8 Smart Card Mode Register (SCMR)

SCMR selects Smart Card interface mode and its format.

Bit	Bit Name	Initial Value	R/W	Description
7	BCP2	1	R/W	Basic Clock Pulse 2 Selects, in combination with the BCP1 and BCP0 bits in SMR, the number of basic clock cycles in a 1-bit transfer interval in Smart Card interface mode. For the settings, refer to section 15.3.5, Serial Mode Register (SMR).
6 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: LSB-first in transfer 1: MSB-first in transfer The bit setting is valid only when the transfer data format is 8 bits. For 7-bit data, LSB-first is fixed.
2	SINV	0	R/W	Smart Card Data Invert Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit. To invert the parity bit, invert the O/$\overline{E}$ 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 This bit is set to 1 to make the SCI operate in Smart Card interface mode. 0: Normal asynchronous mode or clocked synchronous mode 1: Smart Card interface mode

15.3.9 Bit Rate Register (BRR)

BRR is an 8-bit register that adjusts the bit rate. As the SCI performs baud rate generator control independently for each channel, different bit rates can be set for each channel. Table 15.2 shows the relationships between the N setting in BRR and bit rate B for normal asynchronous mode, clocked synchronous mode, and Smart Card interface mode. The initial value of BRR is H'FF, and it can be read or written to by the CPU at all times.

Table 15.2 Relationships between N Setting in BRR and Bit Rate B

Mode	Bit Rate	Error
Asynchronous Mode	$B = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N+1)}$	
Clocked Synchronous Mode	$B = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N+1)}$	
Smart Card Interface Mode	$B = \frac{\phi \times 10^6}{S \times 2^{2n+1} \times (N+1)}$	$\text{Error (\%)} = \{ \frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \} \times 100$

Note: B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ϕ: Operating frequency (MHz)
n and S: Determined by the SMR settings shown in the following tables.

SMR Setting		
CKS1	CKS0	n
0	0	0
0	1	1
1	0	2
1	1	3

SCMR Setting			SMR Setting
BCP2	BCP1	BCP0	S
0	0	0	93
0	0	1	128
0	1	0	186
0	1	1	512
1	0	0	32
1	0	1	64
1	1	0	372
1	1	1	256

Table 15.3 shows sample N settings in BRR in normal asynchronous mode. Table 15.4 shows the maximum bit rate for each frequency in normal asynchronous mode. Table 15.6 shows sample N settings in BRR in clocked synchronous mode. Table 15.8 shows sample N settings in BRR in Smart Card interface mode. In Smart Card interface mode, S (the number of basic clock cycles in a 1-bit transfer interval) can be selected. For details, refer to section 15.7.4, Receive Data Sampling Timing and Reception Margin. Tables 15.5 and 15.7 show the maximum bit rates with external clock input.

Table 15.3 BRR Settings for Various Bit Rates (Asynchronous Mode)

	Operating Frequency ϕ (MHz)											
	8			9.8304			10			12		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	25	0.16	0	31	0.00	0	32	–1.38	0	38	0.16
19200	0	12	0.16	0	15	0.00	0	15	1.70	0	19	–2.40
31250	0	7	0.00	0	9	–1.73	0	9	0.00	0	11	0.00
38400	—	—	—	0	7	0.00	0	7	1.70	0	9	–2.40

	Operating Frequency ϕ (MHz)											
	12.288			14			14.7456			16		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	2	248	–0.17	3	64	0.69	3	70	0.03
150	2	159	0.00	2	181	0.16	2	191	0.00	2	207	0.16
300	2	79	0.00	2	90	0.16	2	95	0.00	2	103	0.16
600	1	159	0.00	1	181	0.16	1	191	0.00	1	207	0.16
1200	1	79	0.00	1	90	0.16	1	95	0.00	1	103	0.16
2400	0	159	0.00	0	181	0.16	0	191	0.00	0	207	0.16
4800	0	79	0.00	0	90	0.16	0	95	0.00	0	103	0.16
9600	0	39	0.00	0	45	–0.94	0	47	0.00	0	51	0.16
19200	0	19	0.00	0	22	–0.94	0	23	0.00	0	25	0.16
31250	0	11	2.34	0	13	0.00	0	14	–1.73	0	15	0.00
38400	0	9	0.00	—	—	—	0	11	0.00	0	12	0.16

	Operating Frequency ϕ (MHz)											
	17.2032			18			19.6608			20		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	75	0.48	3	79	–0.12	3	86	0.31	3	88	–0.25
150	2	223	0.00	2	233	0.16	2	255	0.00	3	64	0.16
300	2	111	0.00	2	116	0.16	2	127	0.00	2	129	0.16
600	1	223	0.00	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	111	0.00	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	223	0.00	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	111	0.00	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	55	0.00	0	58	–0.69	0	63	0.00	0	64	0.16
19200	0	27	0.00	0	28	1.01	0	31	0.00	0	32	–1.38
31250	0	16	1.20	0	17	0.00	0	19	–1.73	0	19	0.00
38400	0	13	0.00	0	14	–2.40	0	15	0.00	0	15	1.70

	Operating Frequency ϕ (MHz)								
	25			30			33		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	110	–0.02	3	132	0.13	3	145	0.33
150	3	80	0.47	3	97	–0.35	3	106	0.39
300	2	162	–0.15	2	194	0.16	2	214	–0.07
600	2	80	0.47	2	97	–0.35	2	106	0.39
1200	1	162	–0.15	1	194	0.16	1	214	–0.07
2400	1	80	0.47	1	97	–0.35	1	106	0.39
4800	0	162	–0.15	0	194	0.16	0	214	–0.07
9600	0	80	0.47	0	97	–0.35	0	106	0.39
19200	0	40	–0.76	0	48	–0.35	0	53	–0.54
31250	0	24	0.00	0	29	0.00	0	32	0.00
38400	0	19	1.70	0	23	1.70	0	26	–0.54

Table 15.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0
25	781250	0	0
30	937500	0	0
33	1031250	0	0

Table 15.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500
25	6.2500	390625
30	7.5000	468750
33	8.2500	515625

Table 15.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate (bit/s)	Operating Frequency ϕ (MHz)													
	8		10		16		20		25		30		33	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110														
250	3	124	—	—	3	249								
500	2	249	—	—	3	124	—	—			3	233		
1 k	2	124	—	—	2	249	—	—	3	97	3	116	3	128
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187	2	205
5 k	1	99	1	124	1	199	1	249	2	77	2	93	2	102
10 k	0	199	0	249	1	99	1	124	1	155	1	187	1	205
25 k	0	79	0	99	0	159	0	199	0	249	1	74	1	82
50 k	0	39	0	49	0	79	0	99	0	124	0	149	0	164
100 k	0	19	0	24	0	39	0	49	0	62	0	74	0	82
250 k	0	7	0	9	0	15	0	19	0	24	0	29	0	32
500 k	0	3	0	4	0	7	0	9	—	—	0	14	—	—
1 M	0	1			0	3	0	4	—	—	—	—	—	—
2.5 M			0	0*			0	1	—	—	0	2	—	—
5 M							0	0*	—	—	—	—	—	—

Legend:

Blank: Cannot be set.

—: Can be set, but there will be a degree of error.

*: Continuous transfer is not possible.

Table 15.7 Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
8	1.3333	1333333.3	18	3.0000	3000000.0
10	1.6667	1666666.7	20	3.3333	3333333.3
12	2.0000	2000000.0	25	4.1667	4166666.7
14	2.3333	2333333.3	30	5.0000	5000000.0
16	2.6667	2666666.7	33	5.5000	5500000.0

Table 15.8 Examples of Bit Rate for Various BRR Settings (Smart Card Interface Mode) (when n = 0 and S = 372)

	Operating Frequency ϕ (MHz)											
	10.00			10.7136			13.00			14.2848		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	30.00	0	1	25.00	0	1	8.99	0	1	0.00

	Operating Frequency ϕ (MHz)											
	16.00			18.00			20.00			25.00		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	12.01	0	2	15.99	0	2	6.66	0	3	12.49

	Operating Frequency ϕ (MHz)					
	30.00			33.00		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)
9600	0	3	5.01	0	4	7.59

Table 15.9 Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode) (when S = 372)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
10.00	13441	0	0	18.00	24194	0	0
10.7136	14400	0	0	20.00	26882	0	0
13.00	17473	0	0	25.00	33602	0	0
14.2848	19200	0	0	30.00	40323	0	0
16.00	21505	0	0	33.00	44355	0	0

15.3.10 IrDA Control Register (IrCR)

IrCR selects the function of SCI_0.

Bit	Bit Name	Initial Value	R/W	Description
7	IrE	0	R/W	IrDA Enable Specifies normal SCI mode or IrDA mode for SCI_0 input/output. 0: Pins TxD0/IrTxD and RxD0/IrRxD function as TxD0 and RxD0 1: Pins TxD0/IrTxD and RxD0/IrRxD function as IrTxD and IrRxD
6	IrCKS2	0	R/W	IrDA Clock Select 2 to 0 Specifies the high pulse width in IrTxD output pulse encoding when the IrDA function is enabled. 000: Pulse width = B × 3/16 (3/16 of bit rate) 001: Pulse width = ϕ/2 010: Pulse width = ϕ/4 011: Pulse width = ϕ/8 100: Pulse width = ϕ/16 101: Pulse width = ϕ/32 110: Pulse width = ϕ/64 111: Pulse width = ϕ/128
5	IrCKS1	0	R/W	
4	IrCKS0	0	R/W	
3	IrTxINV	0	R/W	IrTx Data Invert Specifies the logic level of the IrTxD output to be inverted. When inversion is performed, the high pulse width specified by bits 6 to 4 becomes the low pulse width. 0: Transmit data is used as IrTxD output without change 1: Transmit data is inverted before used as IrTxD output

Bit	Bit Name	Initial Value	R/W	Description
2	IrRxINV	0	R/W	IrRx Data Invert Specifies the logic level of the IrRxD output to be inverted. When inversion is performed, the high pulse width specified by bits 6 to 4 becomes the low pulse width. 0: Transmit data is used as IrRxD output without change 1: Transmit data is inverted before used as IrRxD output
1, 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

15.3.11 Serial Extension Mode Register (SEMR)

SEMR selects the clock source in asynchronous mode. The basic clock can be automatically set by selecting the average transfer rate.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved If these bits are read, an undefined value will be returned and cannot be modified.
3	ABCS	0	R/W	Asynchronous basic clock selection (valid only in asynchronous mode) Selects the basic clock for 1-bit period in asynchronous mode. 0: Operates on a basic clock with a frequency of 16 times the transfer rate. 1: Operates on a basic clock with a frequency of 8 times the transfer rate.

Bit	Bit Name	Initial Value	R/W	Description
2	ACS2	0	R/W	Asynchronous clock source selection (valid when CKE1 = 1 in asynchronous mode) Selects the clock source for the average transfer rate. The basic clock can be automatically set by selecting the average transfer rate in spite of the value of ABCS. 000: External clock input 001: Selects 115.152 kbps which is the average transfer rate dedicated for ϕ= 10.667 MHz. (Operates on a basic clock with a frequency of 16 times the transfer rate.) 010: Selects 460.606 kbps which is the average transfer rate dedicated for ϕ= 10.667 MHz. (Operates on a basic clock with a frequency of 8 times the transfer rate.) 011: Selects 720 kbps which is the average transfer rate dedicated for ϕ = 32 MHz. (Operates on a basic clock with a frequency of 16 times the transfer rate.) 100: Reserved 101: Selects 115.196 kbps which is the average transfer rate dedicated for ϕ = 16 MHz (Operates on a basic clock with a frequency of 16 times the transfer rate.) 110: Selects 460.784 kbps which is the average transfer rate dedicated for ϕ = 16 MHz (Operates on a basic clock with a frequency of 16 times the transfer rate.) 111: Selects 720 kbps which is the average transfer rate dedicated for ϕ = 16 MHz (Operates on a basic clock with a frequency of 8 times the transfer rate.) Note that the average transfer rate does not correspond to the frequency other than 10.667, 16, or 32 MHz.
1	ACS1	0	R/W	
0	ACS0	0	R/W	

15.4 Operation in Asynchronous Mode

Figure 15.2 shows the general format for asynchronous serial communication. One frame consists of a start bit (low level), followed by transfer data, a parity bit, and finally stop bits (high level). In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. In asynchronous serial communication, the communication line is usually held in the mark state (high level). The SCI monitors the communication line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

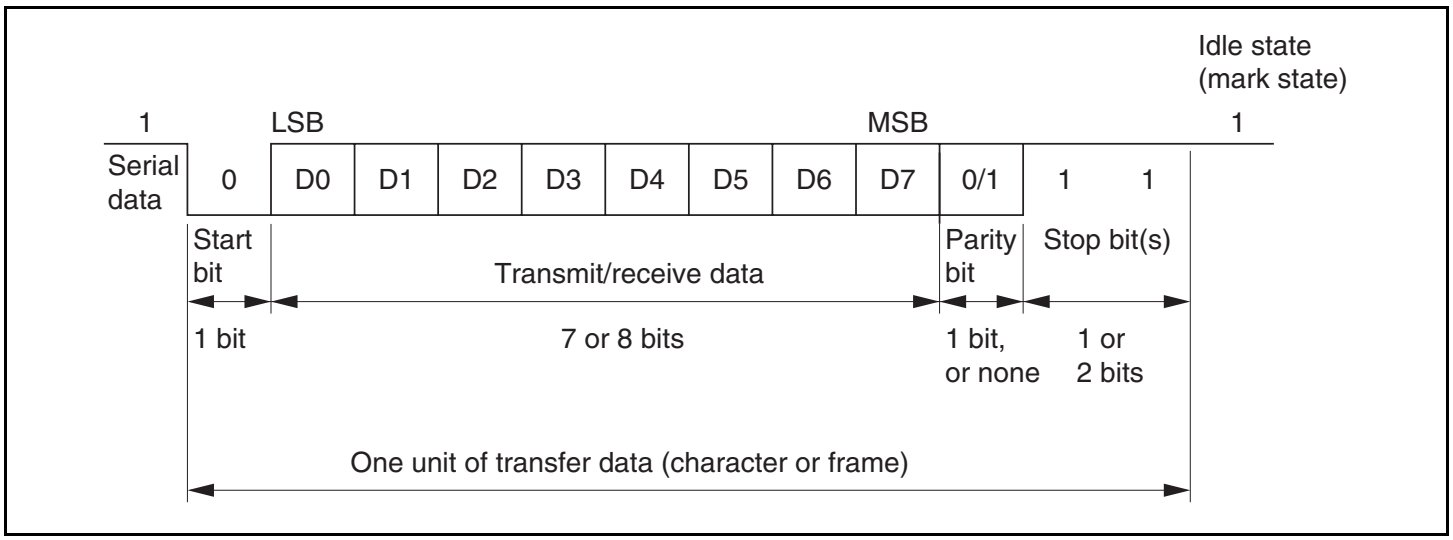


Figure 15.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

15.4.1 Data Transfer Format

Table 15.10 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected according to the SMR setting. For details on the multiprocessor bit, refer to section 15.5, Multiprocessor Communication Function.

Table 15.10 Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STOP
0	1	0	0	S 8-bit data P STOP
0	1	0	1	S 8-bit data P STOP STOP
1	0	0	0	S 7-bit data STOP
1	0	0	1	S 7-bit data STOP STOP
1	1	0	0	S 7-bit data P STOP
1	1	0	1	S 7-bit data P STOP STOP
0	—	1	0	S 8-bit data MPB STOP
0	—	1	1	S 8-bit data MPB STOP STOP
1	—	1	0	S 7-bit data MPB STOP
1	—	1	1	S 7-bit data MPB STOP STOP

Legend:

S : Start bit

STOP : Stop bit

P : Parity bit

MPB : Multiprocessor bit

15.4.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a basic clock with a frequency of 16 times the bit rate. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched at the middle of each bit by sampling the data at the rising edge of the 8th pulse of the basic clock as shown in figure 15.3. Thus the reception margin in asynchronous mode is given by formula (1) below.

$$M = \{ (0.5 - \frac{1}{2N}) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \} \times 100 \ [\%] \quad \text{... Formula (1)}$$

Where M: Reception Margin

N: Ratio of bit rate to clock (N = 16)

D: Clock duty cycle (D = 0.5 to 1.0)

L: Frame length (L = 9 to 12)

F: Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in formula (1), a reception margin is given by formula below.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100 \ [\%] = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

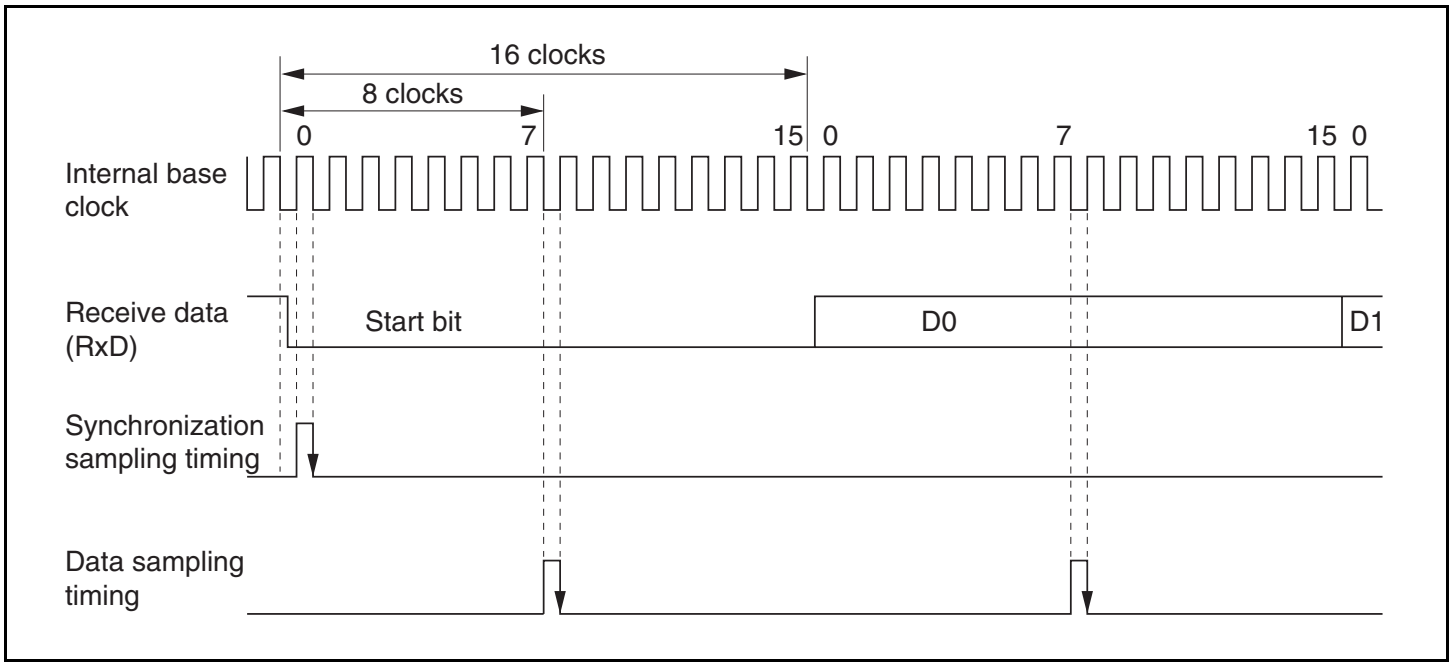


Figure 15.3 Receive Data Sampling Timing in Asynchronous Mode

15.4.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 15.4.

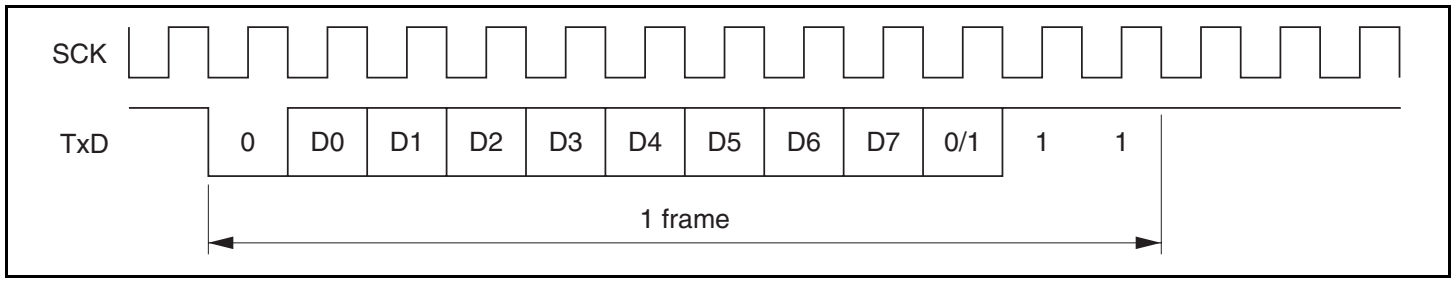
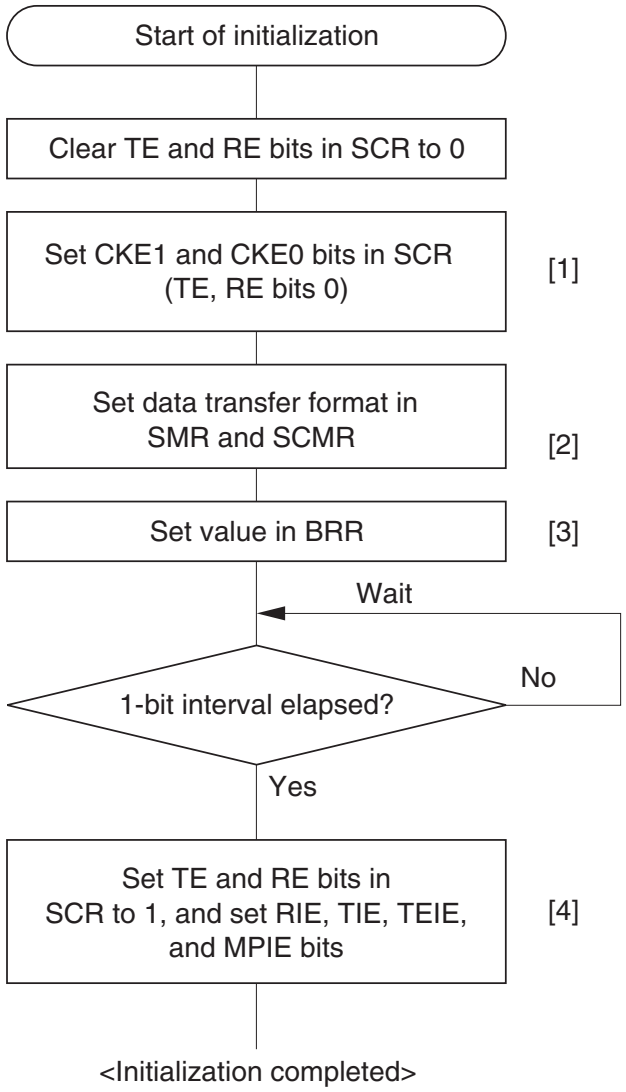


Figure 15.4 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)

15.4.4 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as shown in figure 15.5. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock is selected in asynchronous mode, it is output immediately after SCR settings are made.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. (Not necessary if an external clock is used.)

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.

Setting the TE and RE bits enables the TxD and RxD pins to be used.

Figure 15.5 Sample SCI Initialization Flowchart

15.4.5 Data Transmission (Asynchronous Mode)

Figure 15.6 shows an example of the operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if is cleared to 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit is set to 1 at this time, a transmit data empty interrupt request (TXI) is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit or multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.
6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the “mark state” is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 15.7 shows a sample flowchart for transmission in asynchronous mode.

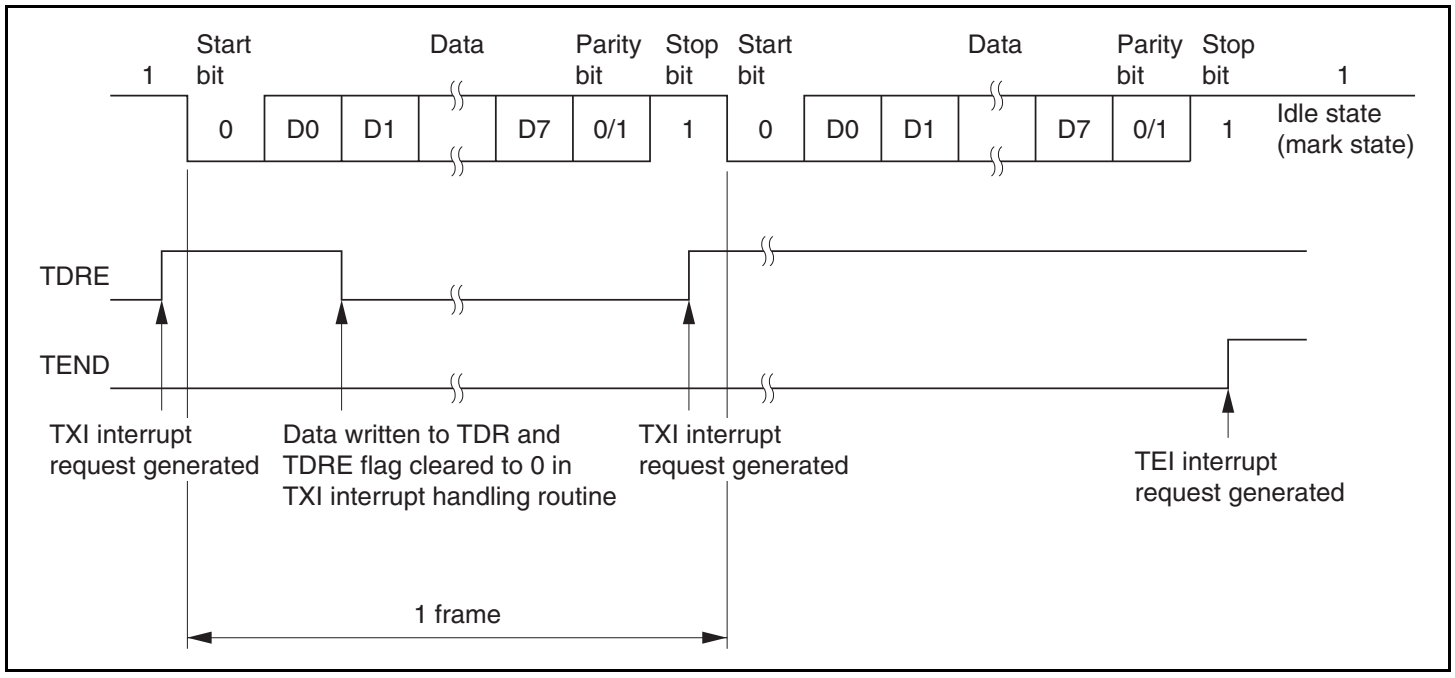
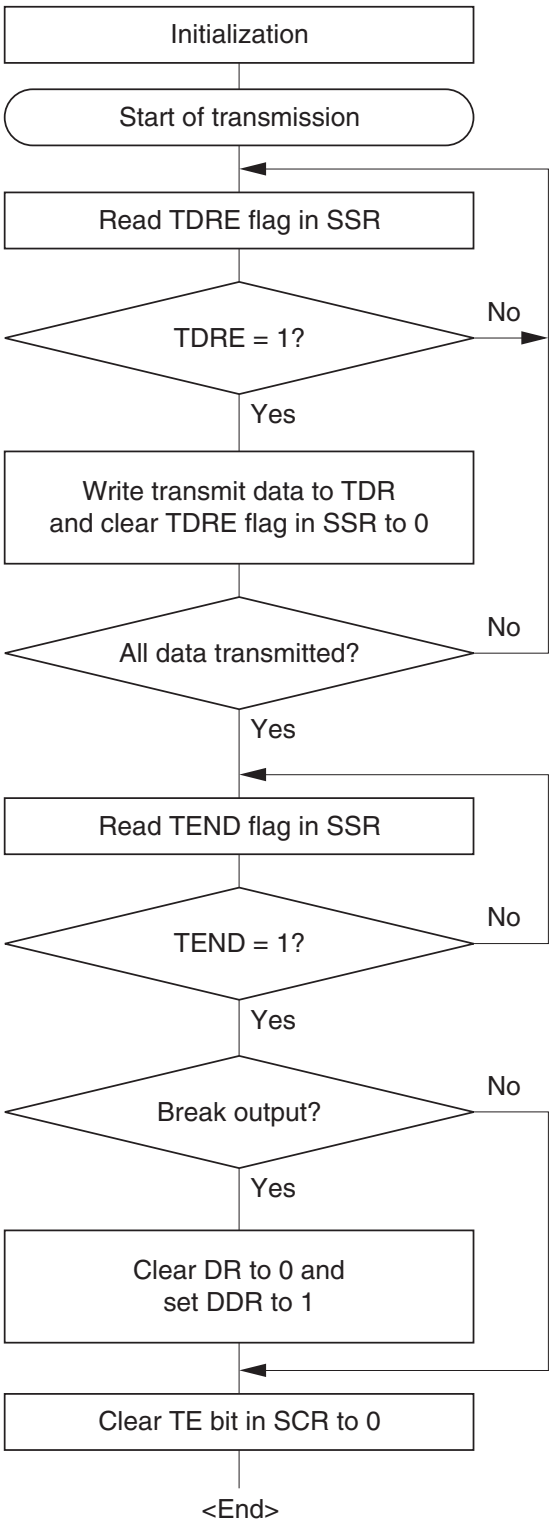


Figure 15.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit-data-empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 15.7 Sample Serial Transmission Flowchart

15.4.6 Serial Data Reception (Asynchronous Mode)

Figure 15.8 shows an example of the operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI monitors the communication line, and if a start bit is detected, performs internal synchronization, receives receive data in RSR, and checks the parity bit and stop bit.
2. If an overrun error (when reception of the next data is completed while the RDRF flag is still set to 1) occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
5. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

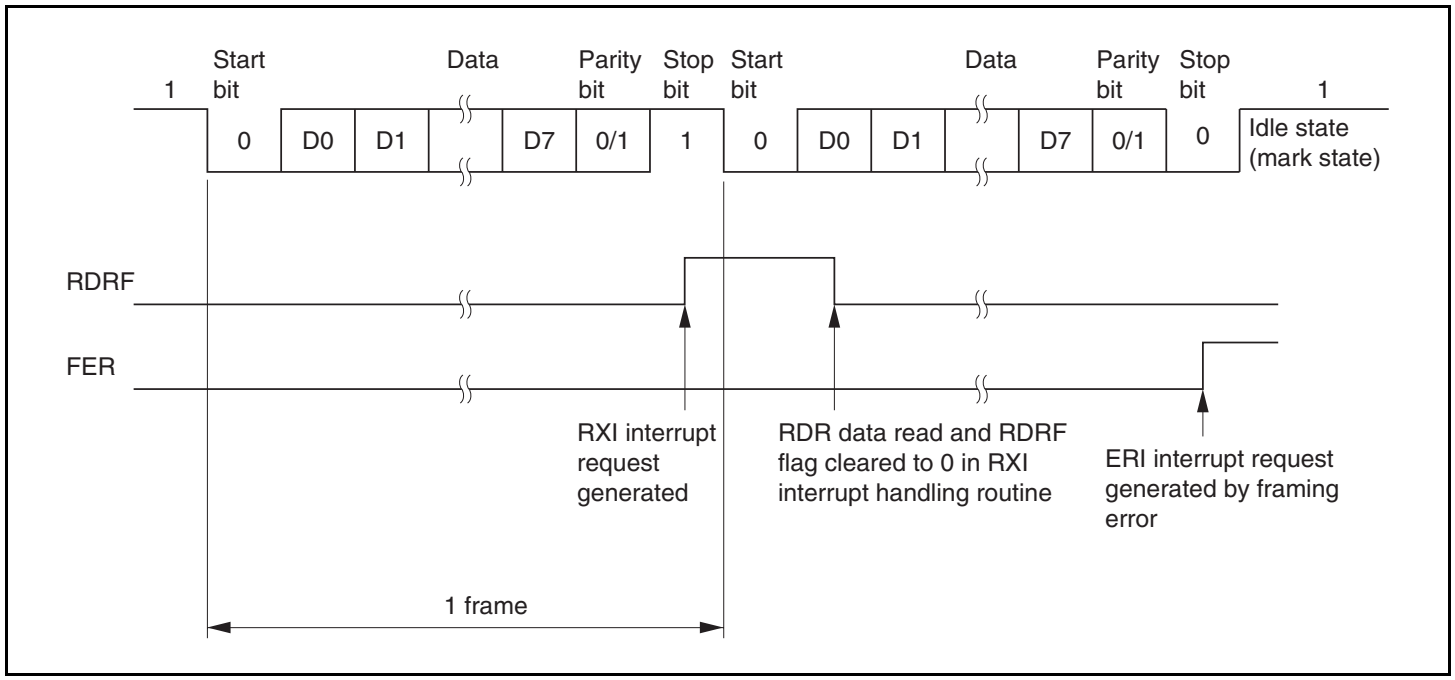


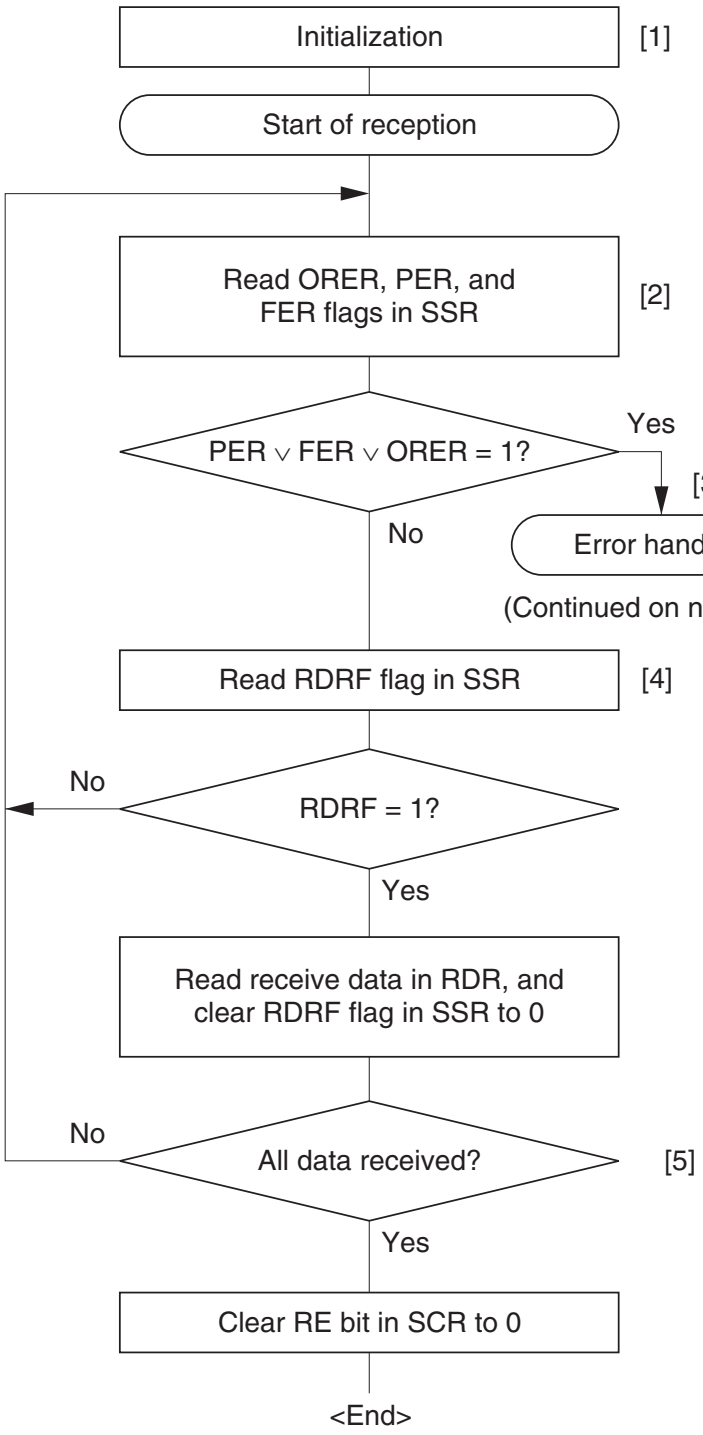
Figure 15.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

Table 15.11 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 15.9 shows a sample flowchart for serial data reception.

Table 15.11 SSR Status Flags and Receive Data Handling

SSR Status Flag					
RDRF*	**ORER**	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	1	0	0	Lost	Overrun error
0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + framing error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

Note: * The RDRF flag retains its state before data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error handling and break detection:
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

[4] SCI status check and receive data read :
Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0. The RDRF flag is cleared automatically when the DMAC or DTC is activated by an RXI interrupt and the RDR value is read.

Figure 15.9 Sample Serial Reception Data Flowchart (1)

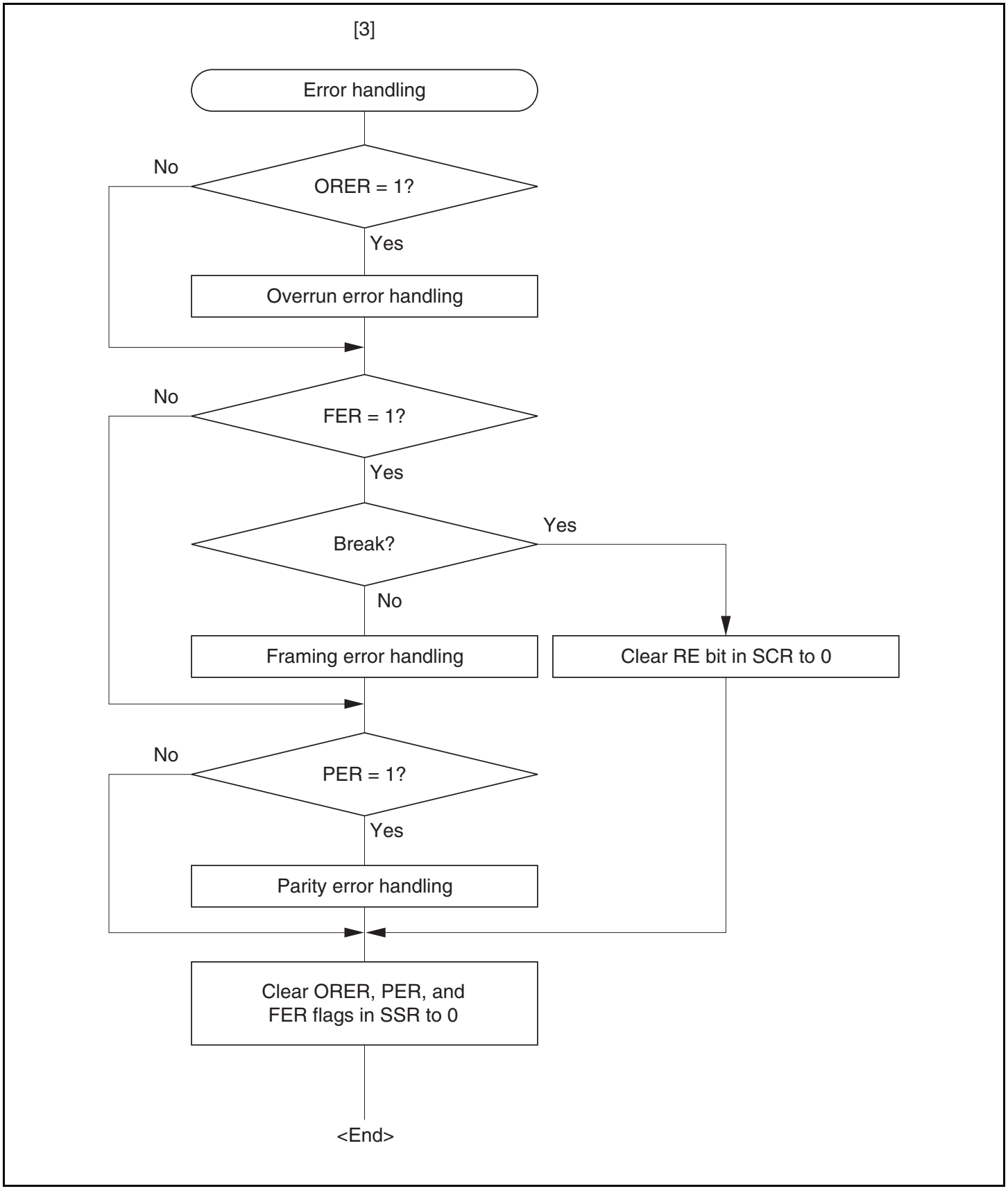


Figure 15.9 Sample Serial Reception Data Flowchart (2)

15.5 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer to be performed among a number of processors sharing communication lines by means of asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle to the specified receiving station. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle, and if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 15.10 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends communication data with a 1 multiprocessor bit added to the ID code of the receiving station. It then sends transmit data as data with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set to 1, transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status flags, RDRF, FER, and ORER to 1 are inhibited until data with a 1 multiprocessor bit is received. On reception of receive character with a 1 multiprocessor bit, the MPBR bit in SSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is invalid. All other bit settings are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

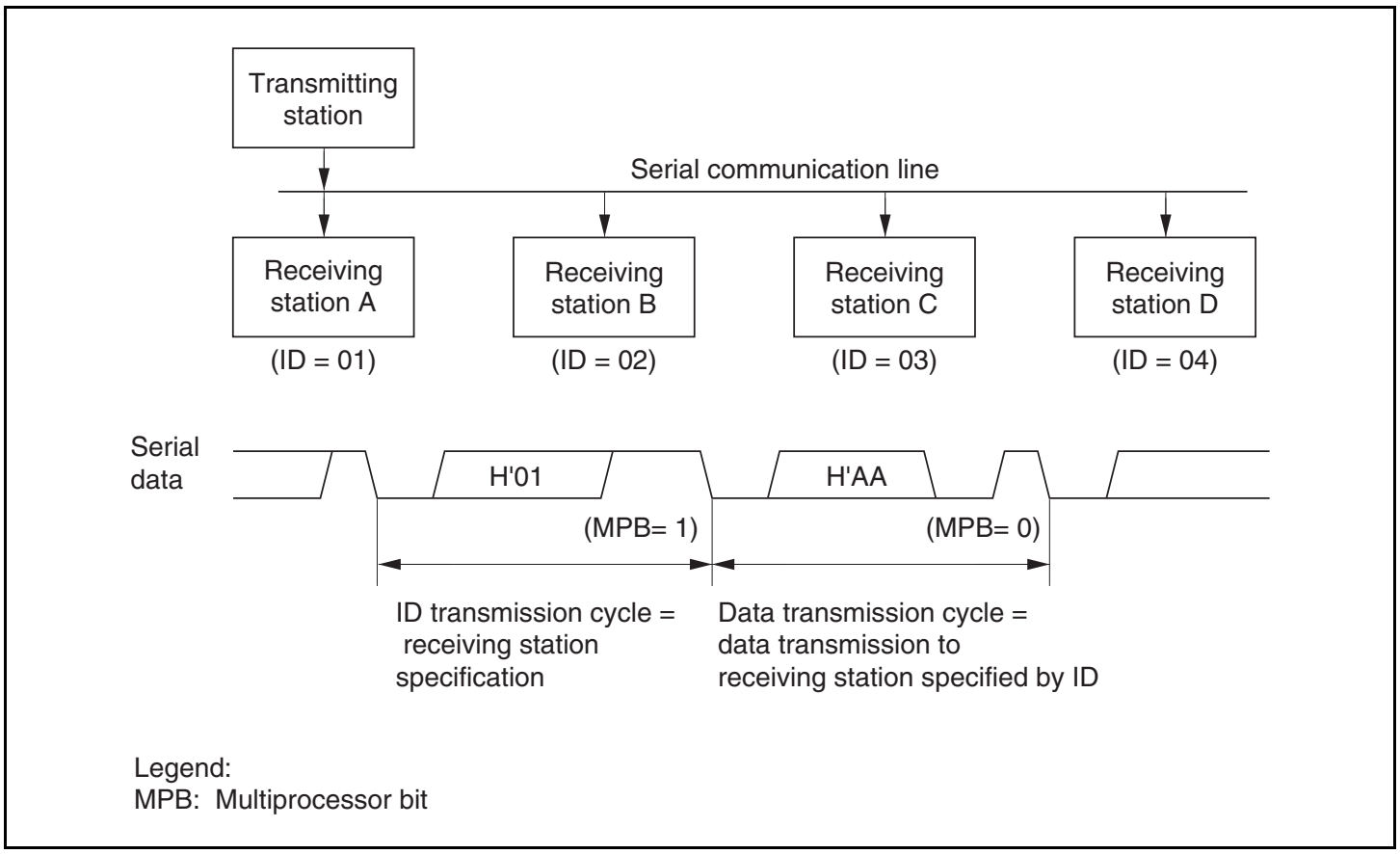
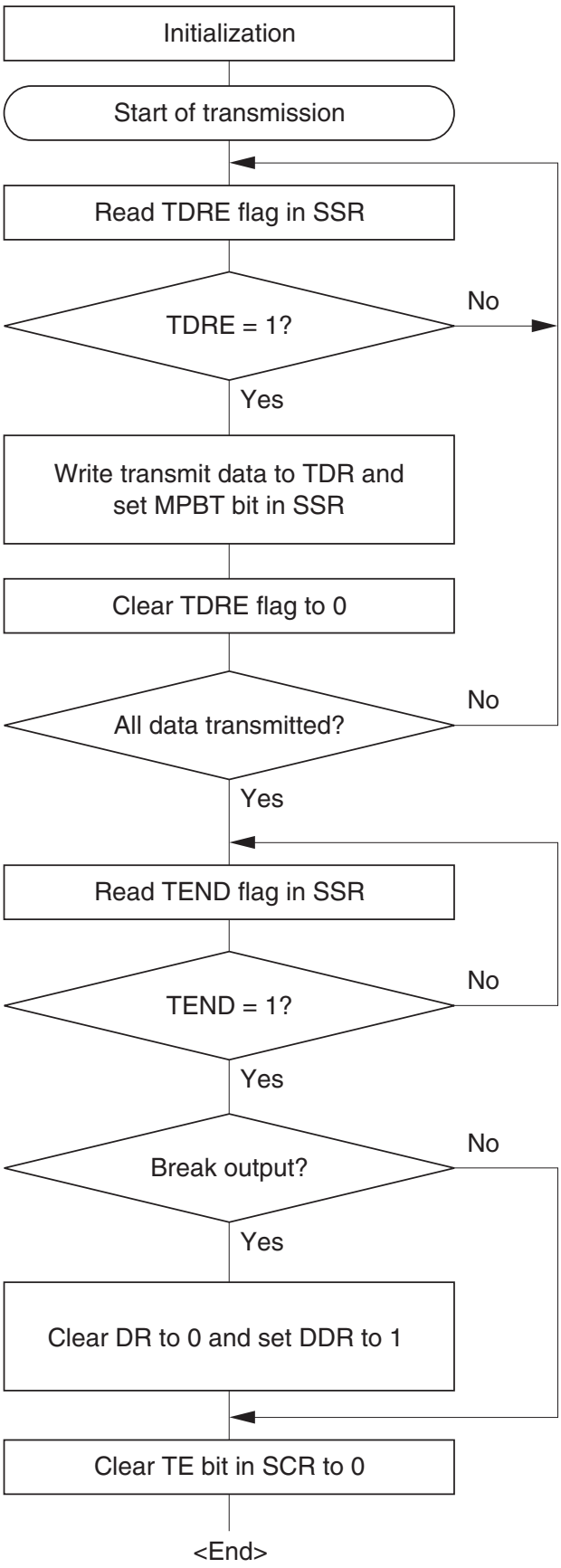


Figure 15.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

15.5.1 Multiprocessor Serial Data Transmission

Figure 15.11 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI operations are the same as those in asynchronous mode.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit-data-empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the port DDR to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 15.11 Sample Multiprocessor Serial Transmission Flowchart

15.5.2 Multiprocessor Serial Data Reception

Figure 15.13 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR is set to 1, data is skipped until data with a 1 multiprocessor bit is received. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 15.12 shows an example of SCI operation for multiprocessor format reception.

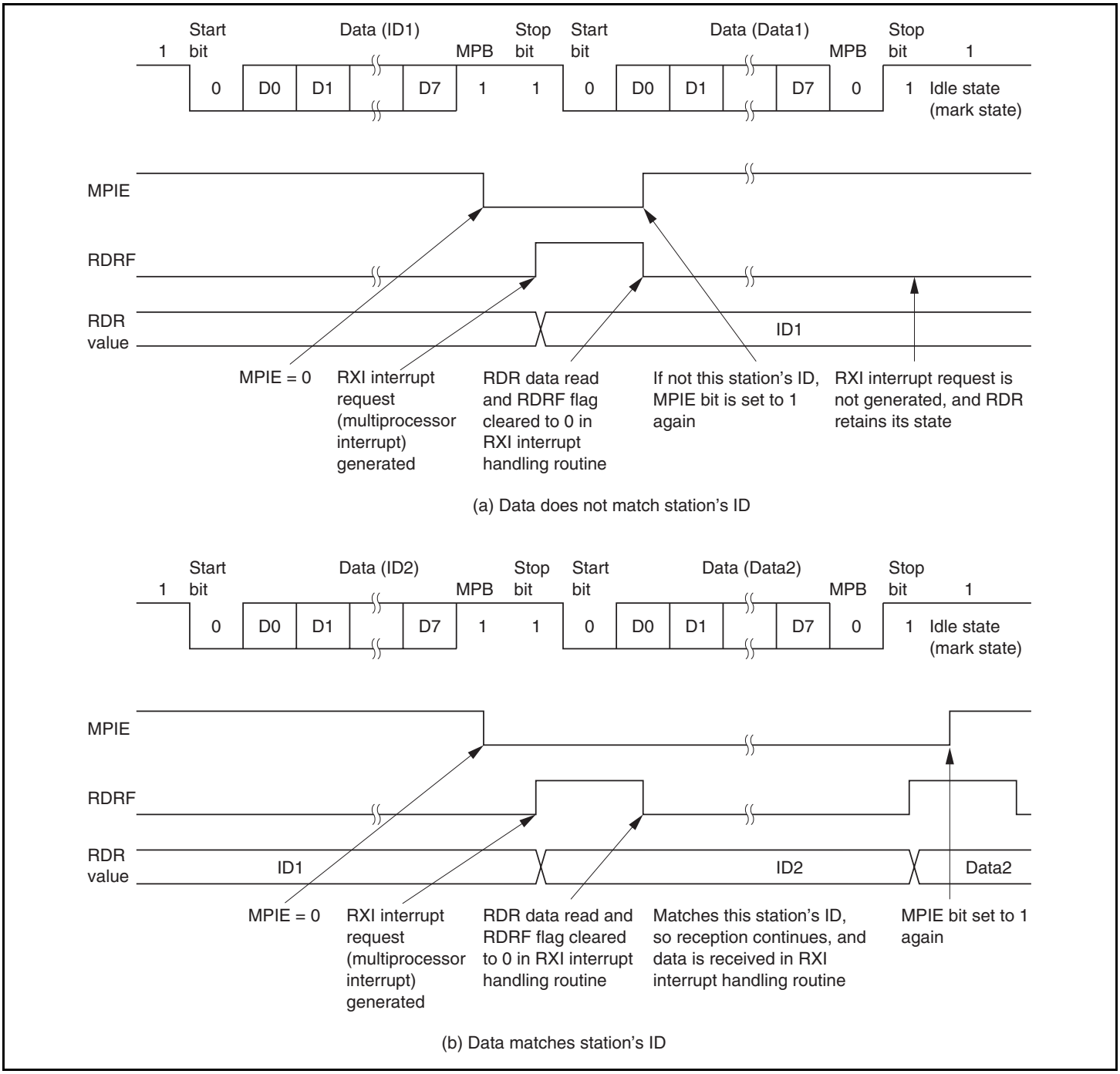
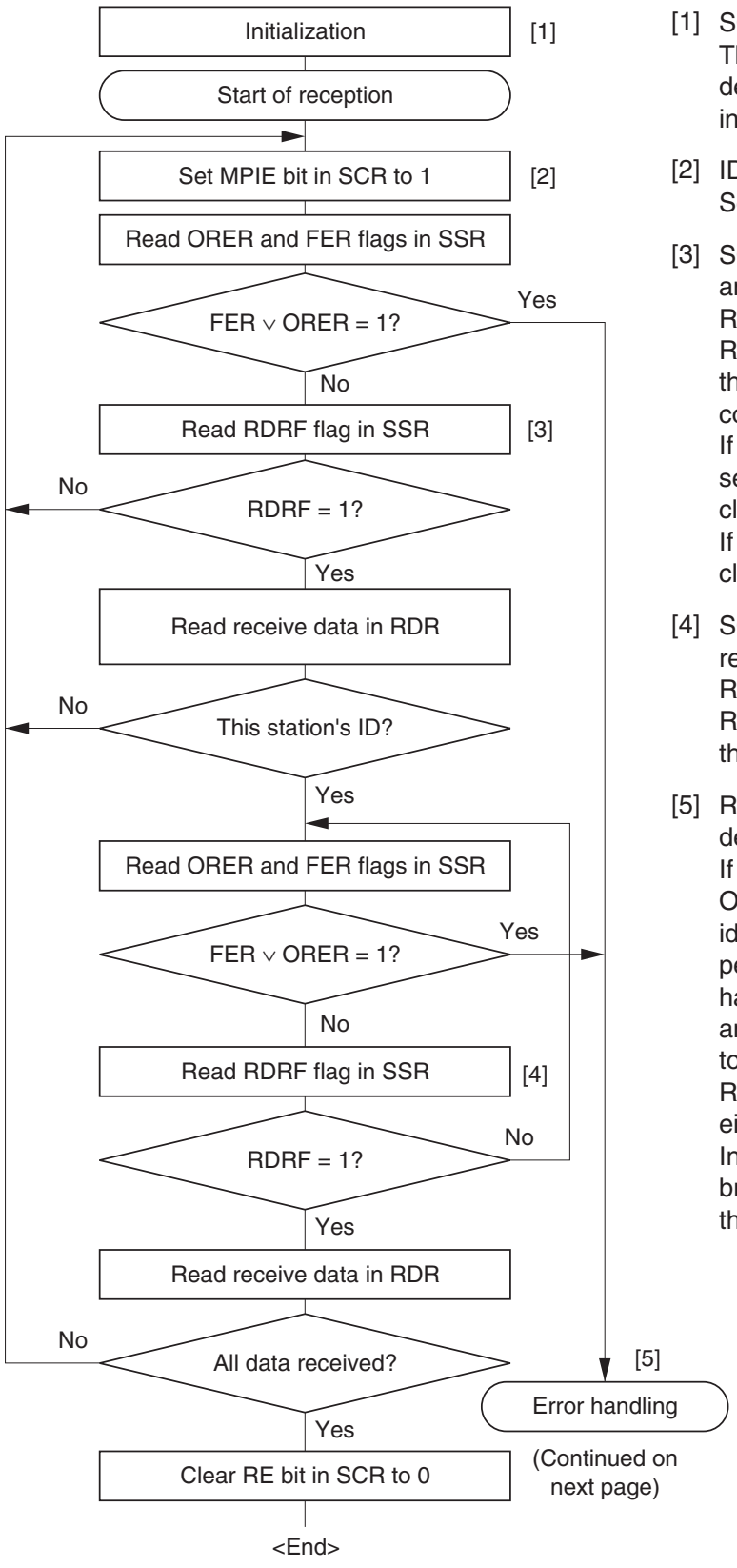


Figure 15.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] ID reception cycle:
Set the MPIE bit in SCR to 1.

[3] SCI status check, ID reception and comparison:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and compare it with this station's ID.
If the data is not this station's ID, set the MPIE bit to 1 again, and clear the RDRF flag to 0.
If the data is this station's ID, clear the RDRF flag to 0.

[4] SCI status check and data reception:
Read SSR and check that the RDRF flag is set to 1, then read the data in RDR.

[5] Receive error handling and break detection:
If a receive error occurs, read the ORER and FER flags in SSR to identify the error. After performing the appropriate error handling, ensure that the ORER and FER flags are both cleared to 0.
Reception cannot be resumed if either of these flags is set to 1.
In the case of a framing error, a break can be detected by reading the RxD pin value.

Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (1)

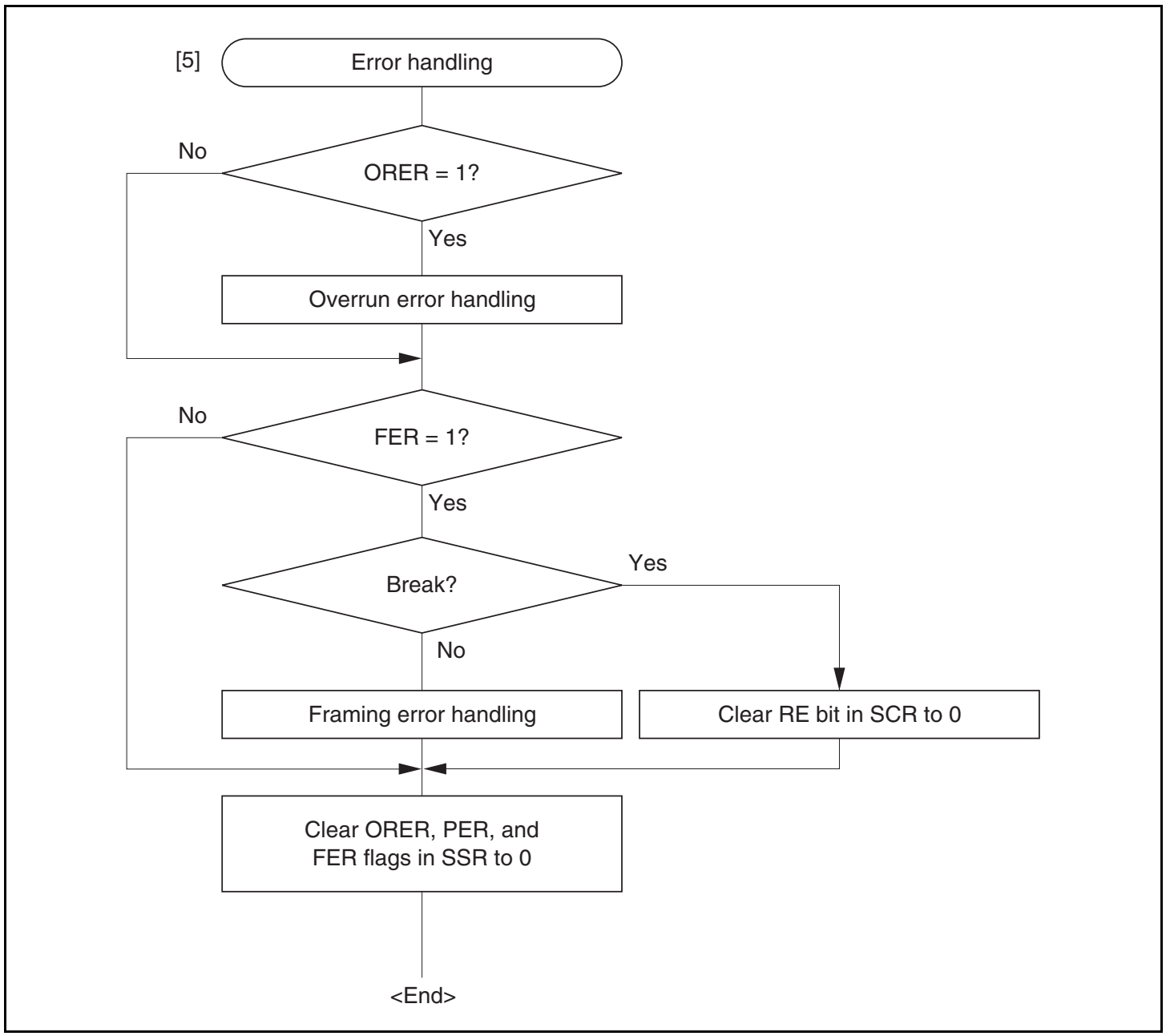


Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (2)

15.6 Operation in Clocked Synchronous Mode

Figure 15.14 shows the general format for clocked synchronous communication. In clocked synchronous mode, data is transmitted or received in synchronization with clock pulses. One character of communication data consists of 8-bit data. In clocked synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. In clocked synchronous mode, the SCI receives data in synchronization with the rising edge of the serial clock. After 8-bit data is output, the transmission line holds the MSB state. In clocked synchronous mode, no parity or multiprocessor bit is added. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

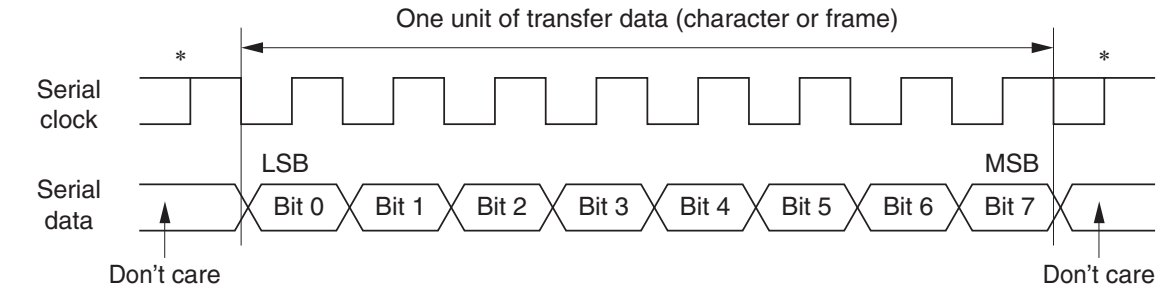


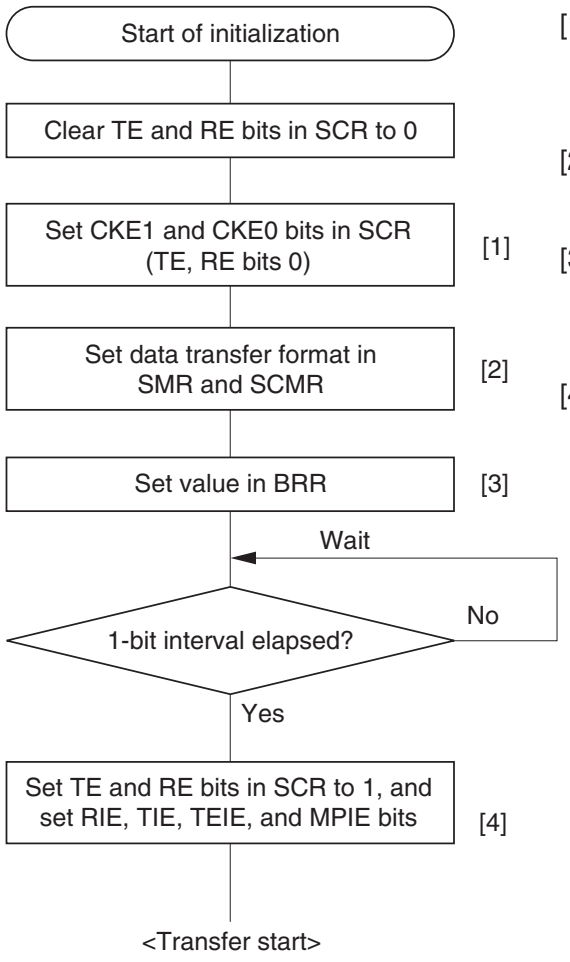
Figure 15.14 Data Format in Clocked Synchronous Communication (For LSB-First)

15.6.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK pin can be selected, according to the setting of CKE1 and CKE0 bits in SCR. When the SCI is operated on an internal clock, the serial clock is output from the SCK pin. Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high.

15.6.2 SCI Initialization (Clocked Synchronous Mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described in a sample flowchart in figure 15.15. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, TE and RE, to 0.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. (Not necessary if an external clock is used.)

[4] Wait at least one bit interval, then set the TE and RE bits in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.
Setting the TE and RE bits enable the TxD and RxD pins to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 15.15 Sample SCI Initialization Flowchart

15.6.3 Serial Data Transmission (Clocked Synchronous Mode)

Figure 15.16 shows an example of SCI operation for transmission in clocked synchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a TXI interrupt request is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. 8-bit data is sent from the TxD pin synchronized with the output clock when output clock mode has been specified and synchronized with the input clock when use of an external clock has been specified.
4. The SCI checks the TDRE flag at the timing for sending the MSB.
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TxD pin maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated. The SCK pin is fixed high.

Figure 15.17 shows a sample flowchart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) is set to 1. Make sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

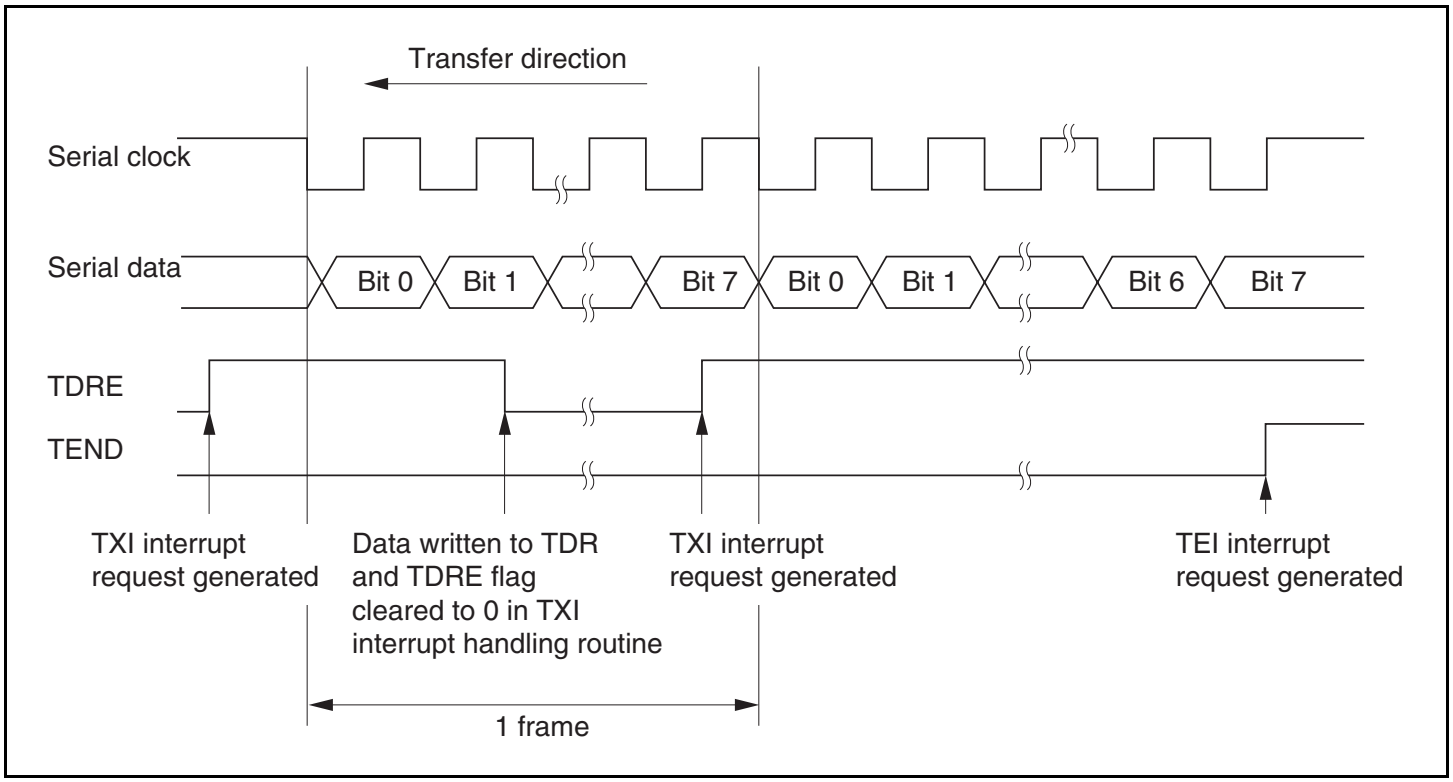
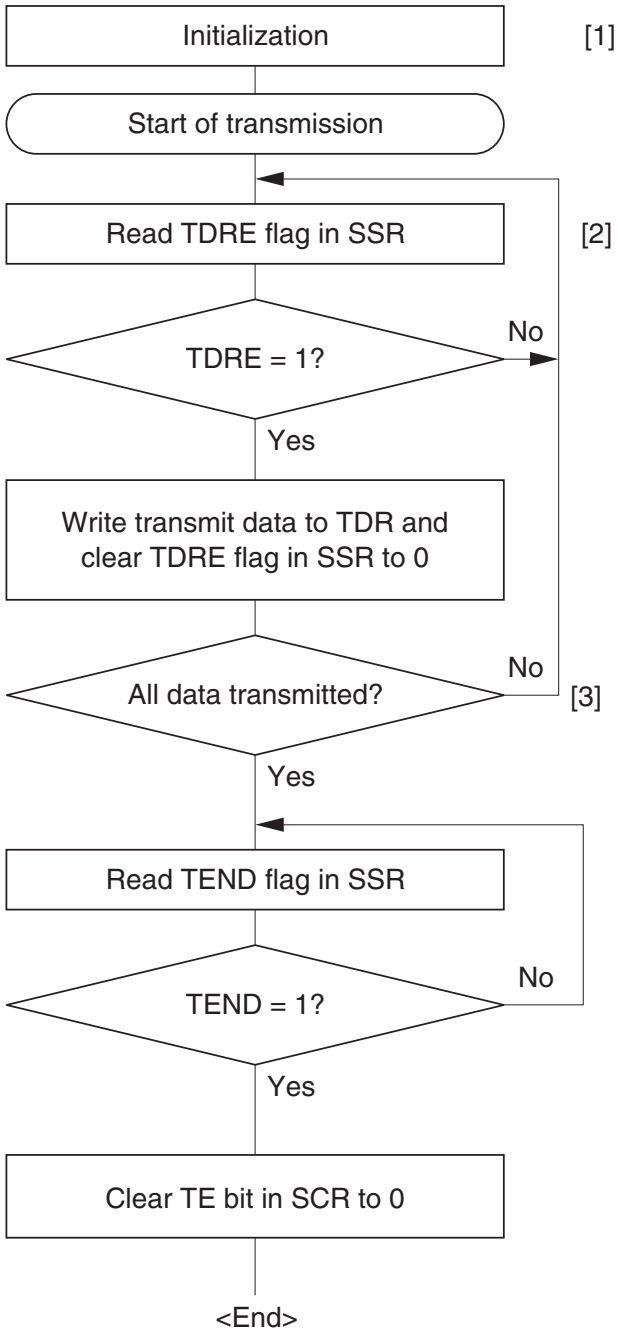


Figure 15.16 Sample SCI Transmission Operation in Clocked Synchronous Mode



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit-data-empty interrupt (TXI) request and data is written to TDR.

Figure 15.17 Sample Serial Transmission Flowchart

15.6.4 Serial Data Reception (Clocked Synchronous Mode)

Figure 15.18 shows an example of SCI operation for reception in clocked synchronous mode. In serial reception, the SCI operates as described below.

1. The SCI performs internal initialization in synchronization with a synchronization clock input or output, starts receiving data, and stores the received data in RSR.
2. If an overrun error (when reception of the next data is completed while the RDRF flag is still set to 1) occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
3. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

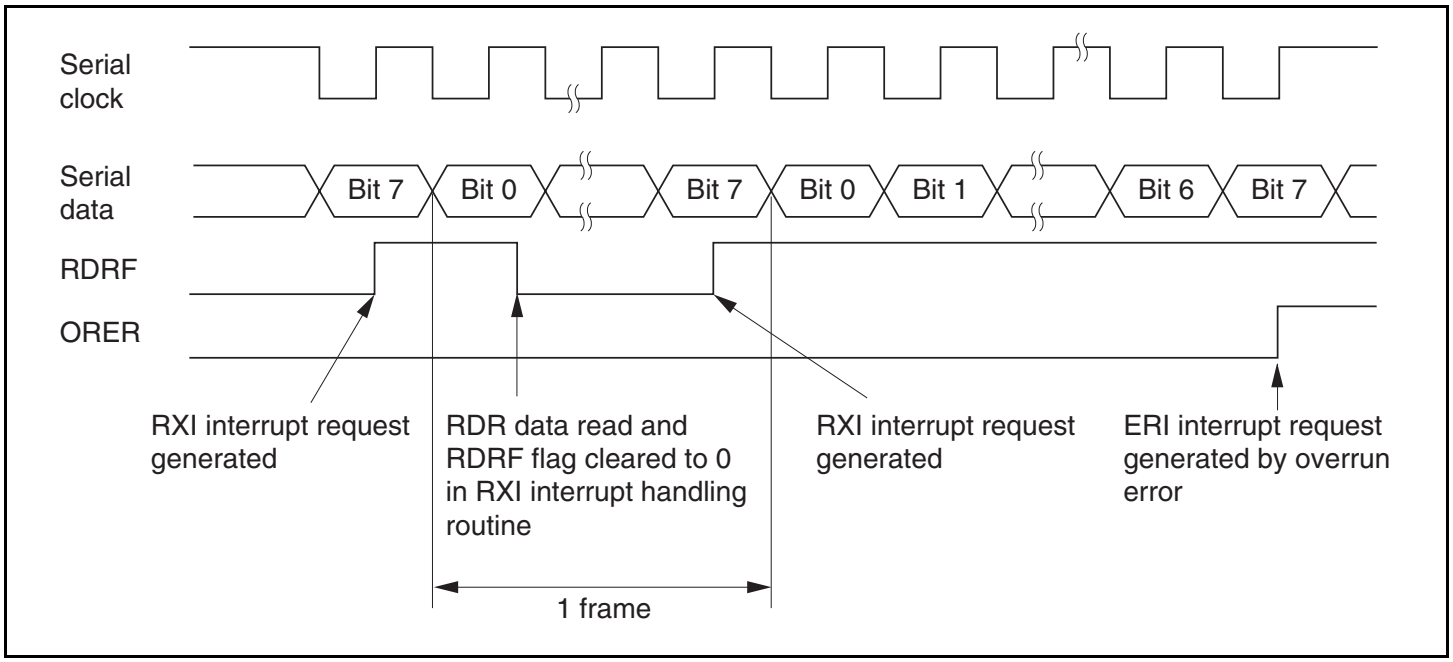
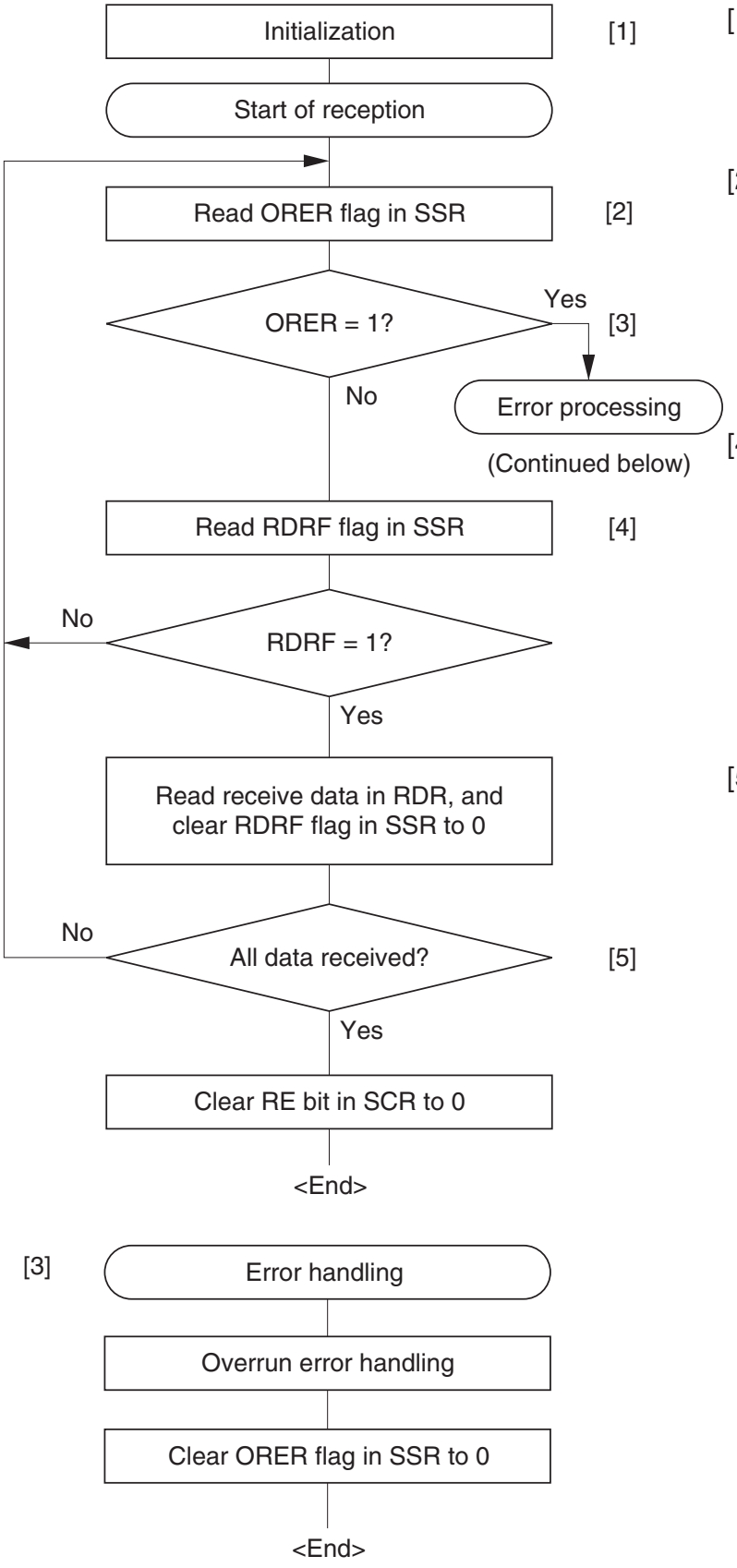


Figure 15.18 Example of SCI Operation in Reception

Transfer cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 15.19 shows a sample flowchart for serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error handling:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error handling, clear the ORER flag to 0. Transfer cannot be resumed if the ORER flag is set to 1.

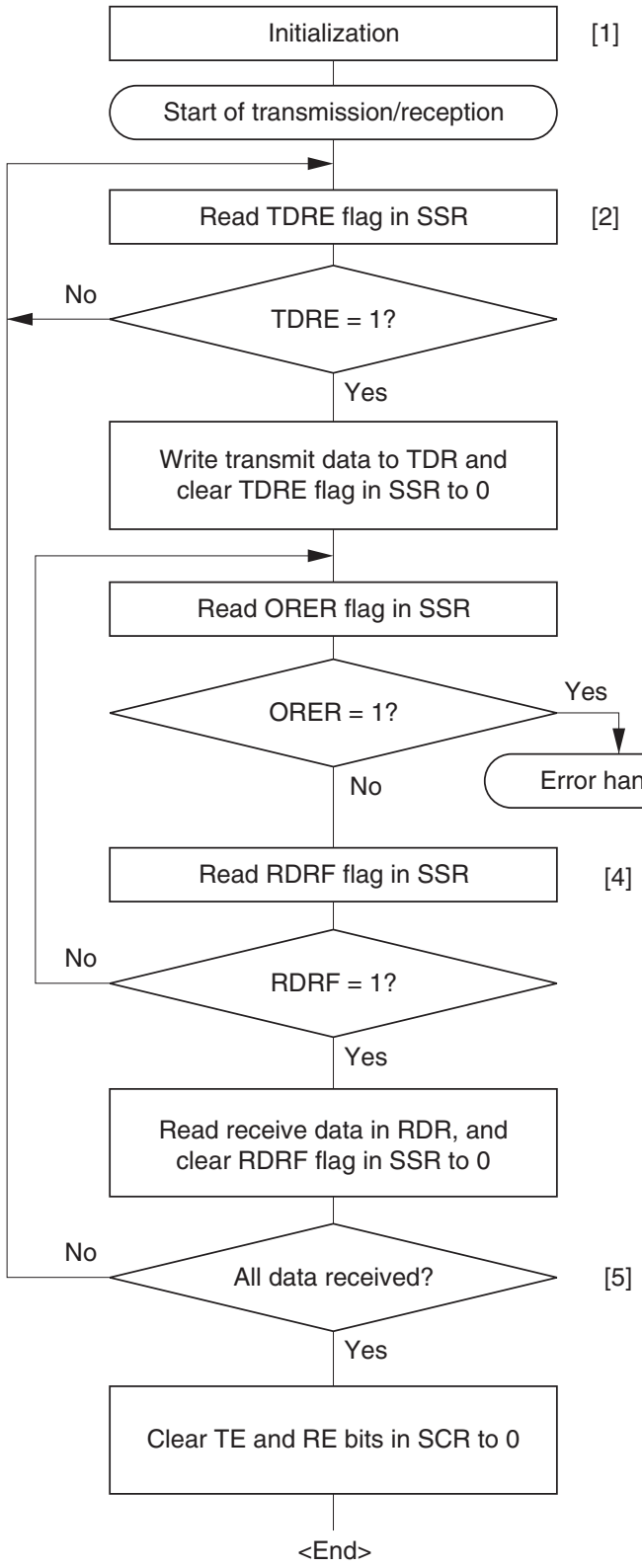
[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. The RDRF flag is cleared automatically when the DMAC or DTC is activated by a receive-data-full interrupt (RXI) request and the RDR value is read.

Figure 15.19 Sample Serial Reception Flowchart

15.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 15.20 shows a sample flowchart for simultaneous serial transmit and receive operations. The following procedure should be used for simultaneous serial data transmit and receive operations after the SCI is initialized. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.



Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE and RE bits to 0, then set both these bits to 1 simultaneously.

[1] SCI initialization:
The TxD pin is designated as the transmit data output pin, and the RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.
Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error handling:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error handling, clear the ORER flag to 0.
Transmission/reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit-data-empty interrupt (TXI) request and data is written to TDR. Also, the RDRF flag is cleared automatically when the DMAC or DTC is activated by a receive-data-full interrupt (RXI) request and the RDR value is read.

Figure 15.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

15.7 Operation in Smart Card Interface Mode

The SCI supports an IC card (Smart Card) interface conforming to ISO/IEC 7816-3 (Identification Card) as a serial communication interface extension function. Switching between the normal serial communication interface and the Smart Card interface is carried out by means of a register setting.

15.7.1 Pin Connection Example

Figure 15.21 shows an example of connection with the Smart Card. In communication with an IC card, since both transmission and reception are carried out on a single data transmission line, the TxD pin and RxD pin should be connected with the LSI pin. The data transmission line should be pulled up to the V_{CC} power supply with a resistor. If an IC card is not connected, and the TE and RE bits are both set to 1, closed transmission/reception is possible, enabling self-diagnosis to be carried out. When the clock generated on the SCI is used by an IC card, the SCK pin output is input to the CLK pin of the IC card. This LSI port output is used as the reset signal.

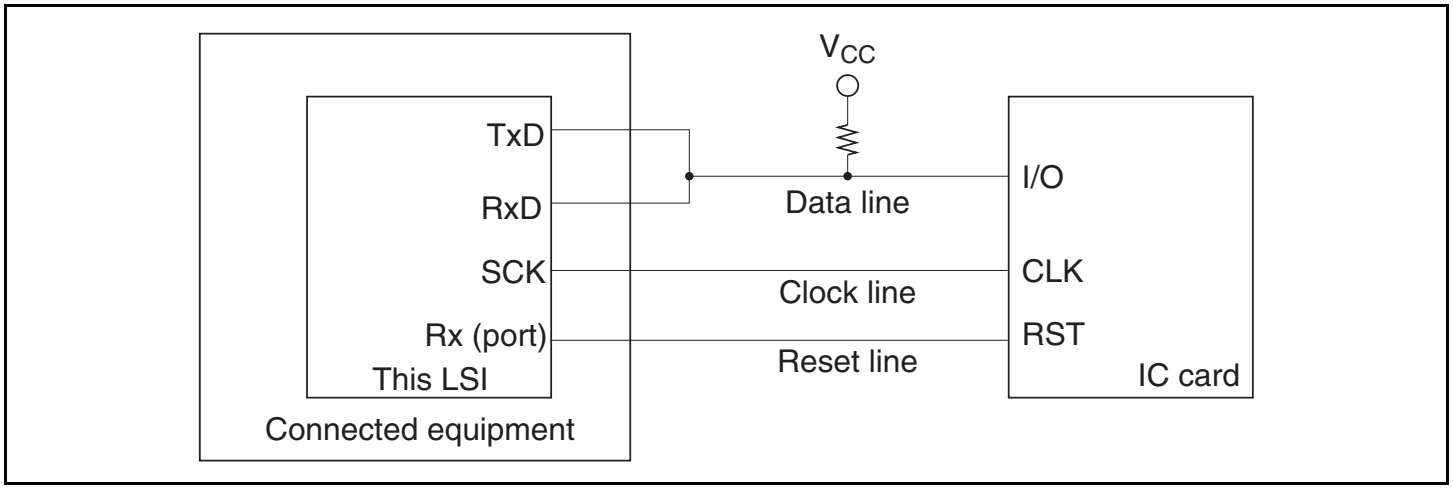


Figure 15.21 Schematic Diagram of Smart Card Interface Pin Connections

15.7.2 Data Format (Except for Block Transfer Mode)

Figure 15.22 shows the transfer data format in Smart Card interface mode.

- One frame consists of 8-bit data plus a parity bit in asynchronous mode.
- In transmission, a guard time of at least 2 etu (Elementary Time Unit: time for transfer of 1 bit) is left between the end of the parity bit and the start of the next frame.
- If a parity error is detected during reception, a low error signal level is output for one etu period, 10.5 etu after the start bit.
- If an error signal is sampled during transmission, the same data is retransmitted automatically after the elapse of 2 etu or longer.

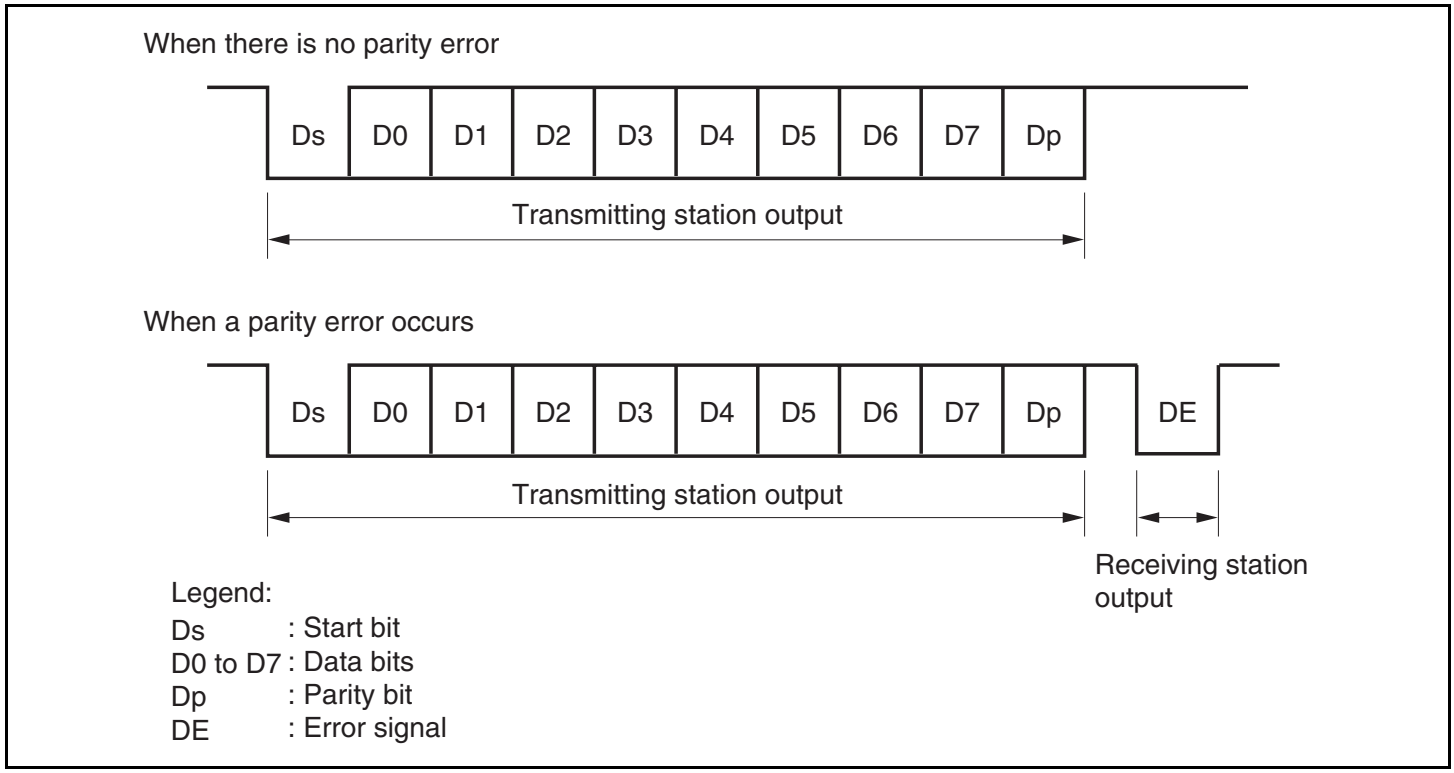


Figure 15.22 Normal Smart Card Interface Data Format

Data transfer with the types of IC cards (direct convention and inverse convention) are performed as described in the following.

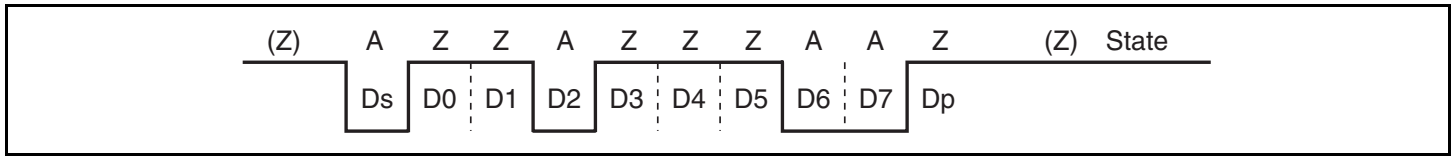


Figure 15.23 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

As in the above sample start character, with the direct convention type, the logic 1 level corresponds to state Z and the logic 0 level to state A, and transfer is performed in LSB-first order. The start character data above is H'3B. For the direct convention type, clear the SDIR and SINV bits in SCMR to 0. According to the Smart Card regulations, clear the O/$\overline{E}$ bit in SMR to 0 to select even parity mode.

Figure 15.24 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1)

With the inverse convention type, the logic 1 level corresponds to state A and the logic 0 level to state Z, and transfer is performed in MSB-first order. The start character data above is H'3F. For the inverse convention type, set the SDIR and SINV bits in SCMR to 1. According to the Smart Card regulations, even parity mode is the logic 0 level of the parity bit, and corresponds to state Z. In this LSI, the SINV bit inverts only data bits D7 to D0. Therefore, set the O/$\overline{E}$ bit in SMR to 1 to invert the parity bit for both transmission and reception.

15.7.3 Block Transfer Mode

Operation in block transfer mode is the same as that in normal Smart Card interface, except for the following points.

- In reception, though the parity check is performed, no error signal is output even if an error is detected. However, the PER bit in SSR is set to 1 and must be cleared before receiving the parity bit of the next frame.
- In transmission, a guard time of at least 1 etu is left between the end of the parity bit and the start of the next frame.
- In transmission, because retransmission is not performed, the TEND flag is set to 1, 11.5 etu after transmission start.
- As with the normal Smart Card interface, the ERS flag indicates the error signal status, but since error signal transfer is not performed, this flag is always cleared to 0.

15.7.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator is used as transmit/receive clock in Smart Card interface. In Smart Card interface mode, the SCI operates on a basic clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate (fixed at 16 times in normal asynchronous mode) as determined by bits BCP2 to BCP0. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. As shown in figure 15.25, by sampling receive data at the rising-edge of the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, or 256th pulse of the basic clock, data can be latched at the middle of the bit. The reception margin is given by the following formula.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}\,(1 + F) \right| \times 100\ [\%]$$

Where M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, 256, 93, 128, 186, or 512)

D: Clock duty cycle (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5 and N = 372 in the above formula, the reception margin formula is as follows.

$$M = (0.5 - 1/2 \times 372) \times 100\%$$
$$= 49.866\%$$

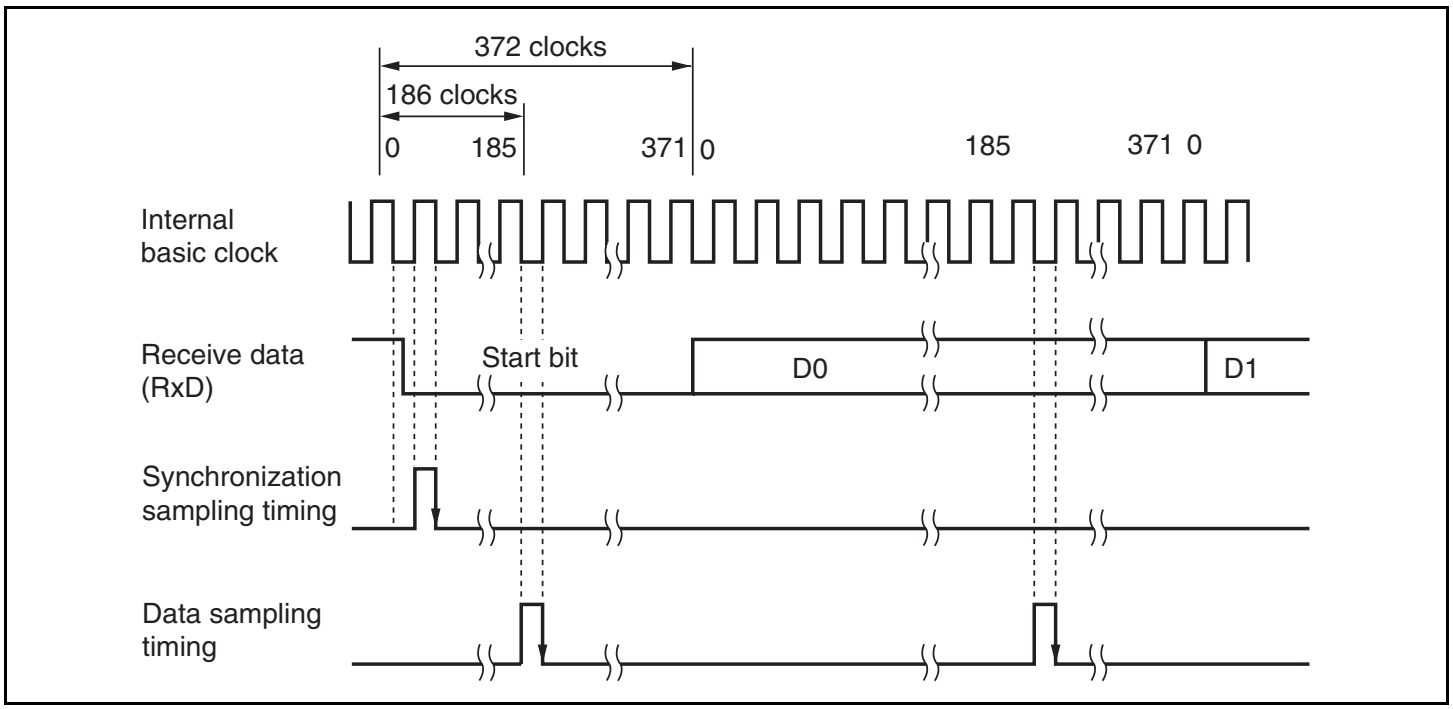


Figure 15.25 Receive Data Sampling Timing in Smart Card Mode (Using Clock of 372 Times the Bit Rate)

15.7.5 Initialization

Before transmitting and receiving data, initialize the SCI as described below. Initialization is also necessary when switching from transmit mode to receive mode, or vice versa.

1. Clear the TE and RE bits in SCR to 0.
2. Clear the error flags ERS, PER, and ORER in SSR to 0.
3. Set the GM, BLK, O/$\overline{E}$, BCP1, BCP0, CKS1, and CKS0 bits in SMR, and the BCP2 bit in SCMR. Set the PE bit to 1.
4. Set the SMIF, SDIR, and SINV bits in SCMR.

   When the SMIF bit is set to 1, the TxD and RxD pins are both switched from ports to SCI pins, and are placed in the high-impedance state.
5. Set the value corresponding to the bit rate in BRR.
6. Set the CKE0 and CKE1 bits in SCR. Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0.

   If the CKE0 bit is set to 1, the clock is output from the SCK pin.
7. Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do not set the TE bit and RE bit at the same time, except for self-diagnosis.

To switch from receive mode to transmit mode, after checking that the SCI has finished reception, initialize the SCI, and clear RE to 0 and set TE to 1. Whether SCI has finished reception can be checked with the RDRF, PER, or ORER flag. To switch from transmit mode to receive mode, after checking that the SCI has finished transmission, initialize the SCI, and clear TE to 0 and set RE to 1. Whether SCI has finished transmission can be checked with the TEND flag.

15.7.6 Data Transmission (Except for Block Transfer Mode)

As data transmission in Smart Card interface mode involves error signal sampling and retransmission processing, the operations are different from those in normal serial communication interface mode (except for block transfer mode). Figure 15.26 illustrates the retransfer operation when the SCI is in transmit mode.

1. If an error signal is sampled from the receiving end after transmission of one frame is completed, the ERS bit in SSR is set to 1. If the RIE bit in SCR is set at this time, an ERI interrupt request is generated. The ERS bit in SSR should be cleared to 0 before the next parity bit is sampled.
2. The TEND bit in SSR is not set for a frame for which an error signal is received. Data is retransferred from TDR to TSR, and retransmitted automatically.
3. If an error signal is not sent back from the receiving end, the ERS bit in SSR is not set.
4. Transmission of one frame, including a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1. If the TIE bit in SCR is set at this time, a TXI interrupt request is generated. Writing transmit data to TDR transfers the next transmit data.

Figure 15.28 shows a flowchart for transmission. The sequence of transmit operations can be performed automatically by specifying the DTC or DMAC to be activated with a TXI interrupt source. In a transmit operation, the TDRE flag is also set to 1 at the same time as the TEND flag in SSR, and a TXI interrupt will be generated if the TIE bit in SCR has been set to 1. If the TXI request is designated beforehand as a DTC or DMAC activation source, the DTC or DMAC will be activated by the TXI request, and transfer of the transmit data will be carried out. The TDRE and TEND flags are automatically cleared to 0 when data transfer is performed by the DTC or DMAC. In the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC will automatically transmit the specified number of bytes in the event of an error, including retransmission. However, the ERS flag is not cleared automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When performing transfer using the DTC or DMAC, it is essential to set and enable the DTC or DMAC before carrying out SCI setting. For details on the DTC or DMAC setting procedures, refer to section 9, Data Transfer Controller (DTC) or section 7, DMA Controller (DMAC).

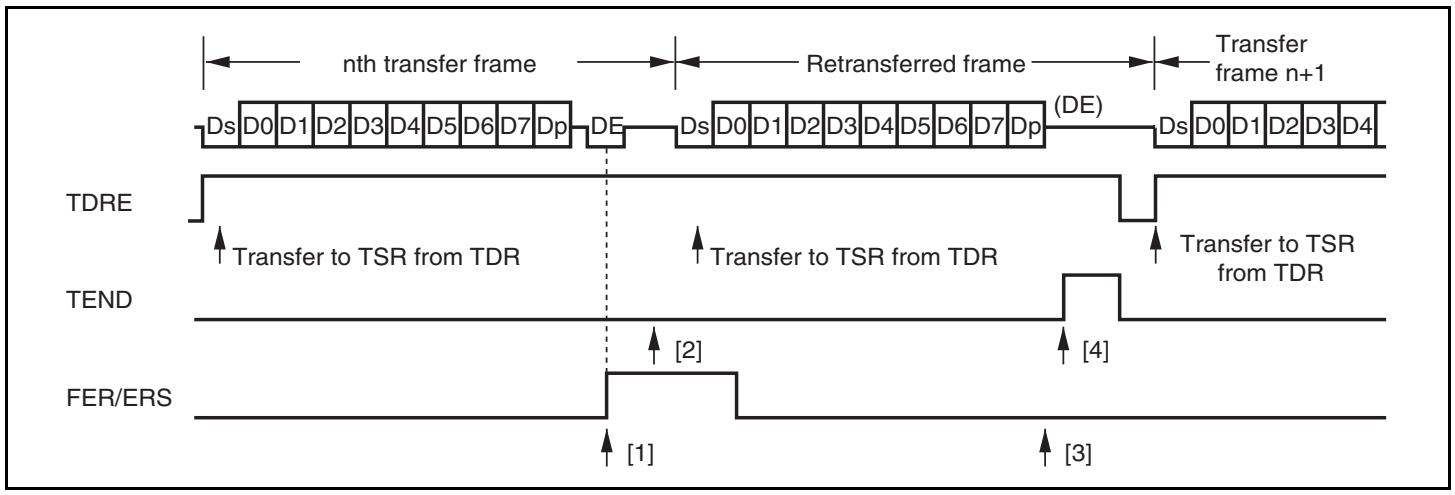


Figure 15.26 Retransfer Operation in SCI Transmit Mode

The timing for setting the TEND flag depends on the value of the GM bit in SMR. The TEND flag generation timing is shown in figure 15.27.

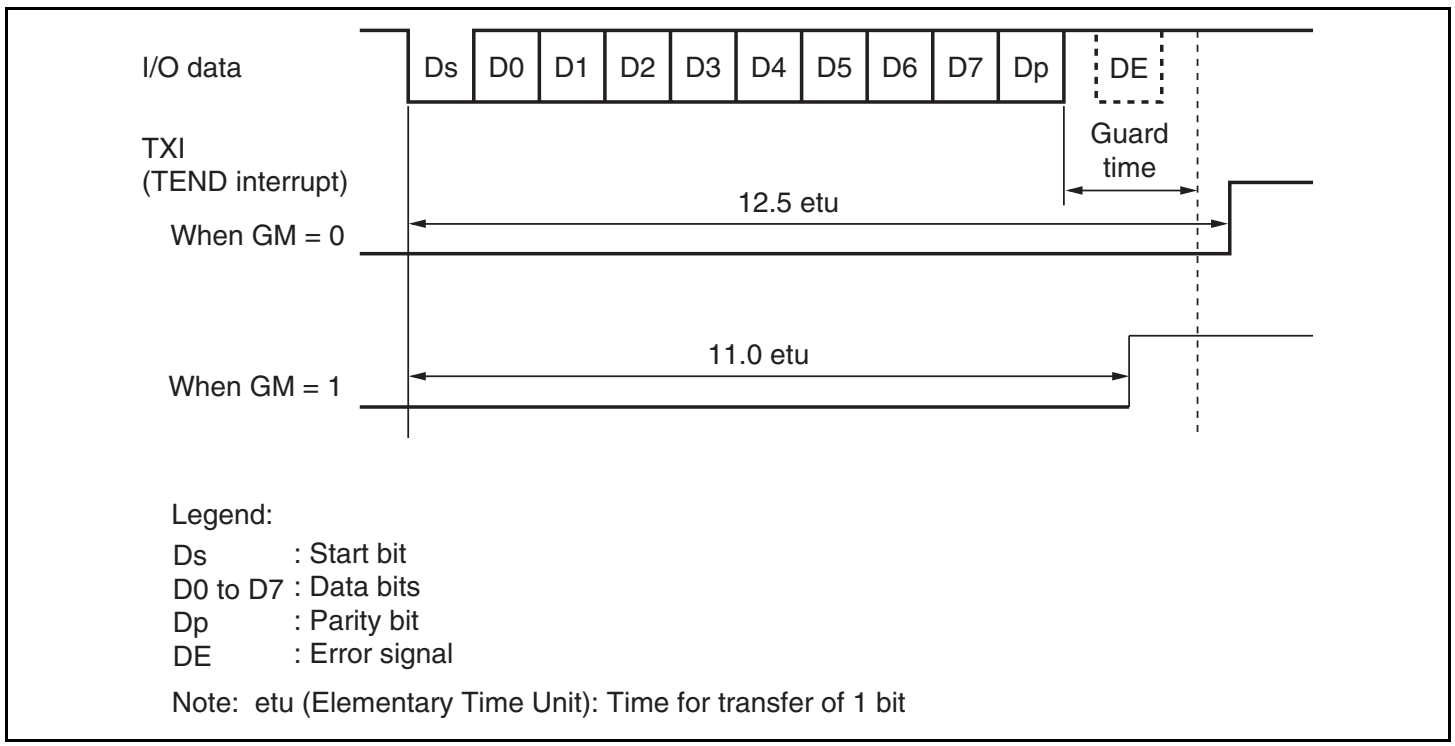


Figure 15.27 TEND Flag Generation Timing in Transmission Operation

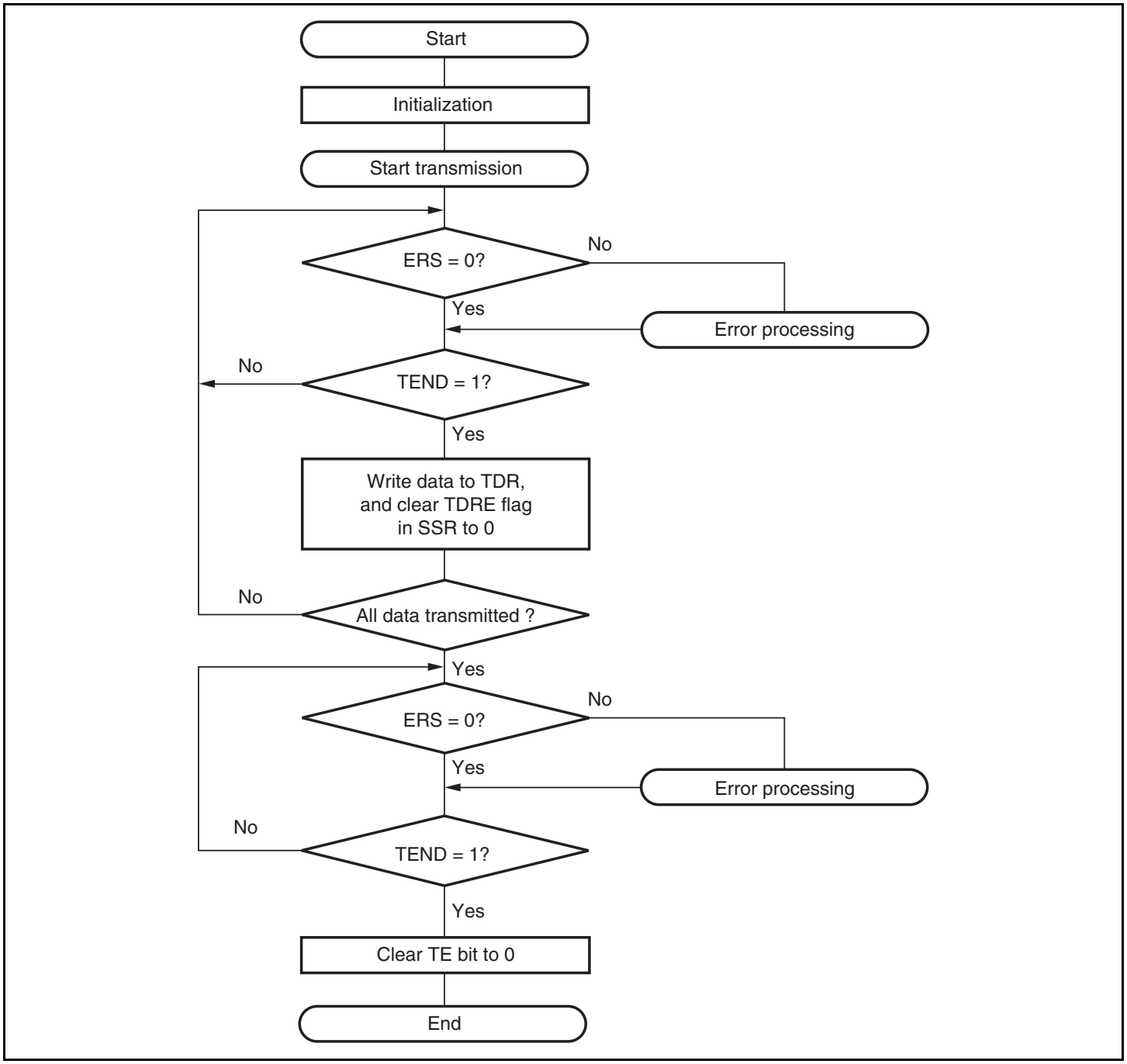


Figure 15.28 Example of Transmission Processing Flow

15.7.7 Serial Data Reception (Except for Block Transfer Mode)

Data reception in Smart Card interface mode uses the same operation procedure as for normal serial communication interface mode. Figure 15.29 illustrates the retransfer operation when the SCI is in receive mode.

1. If an error is found when the received parity bit is checked, the PER bit in SSR is automatically set to 1. If the RIE bit in SCR is set at this time, an ERI interrupt request is generated. The PER bit in SSR should be cleared to 0 before the next parity bit is sampled.
2. The RDRF bit in SSR is not set for a frame in which an error has occurred.
3. If no error is found when the received parity bit is checked, the PER bit in SSR is not set to 1.
4. The receive operation is judged to have been completed normally, and the RDRF flag in SSR is automatically set to 1. If the RIE bit in SCR is set at this time, an RXI interrupt request is generated.

Figure 15.30 shows a flowchart for reception. The sequence of receive operations can be performed automatically by specifying the DTC or DMAC to be activated with an RXI interrupt source. In a receive operation, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DTC or DMAC activation source, the DTC or DMAC will be activated by the RXI request, and transfer of the receive data will be carried out. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC or DMAC. If an error occurs in receive mode and the ORER or PER flag is set to 1, a transfer error interrupt (ERI) request will be generated, and so the error flag must be cleared to 0. In the event of an error, the DTC or DMAC is not activated and receive data is skipped. Therefore, receive data is transferred for only the specified number of bytes in the event of an error. Even when a parity error occurs in receive mode and the PER flag is set to 1, the data that has been received is transferred to RDR and can be read from there.

Note: For details on receive operations in block transfer mode, refer to section 15.4, Operation in Asynchronous Mode.

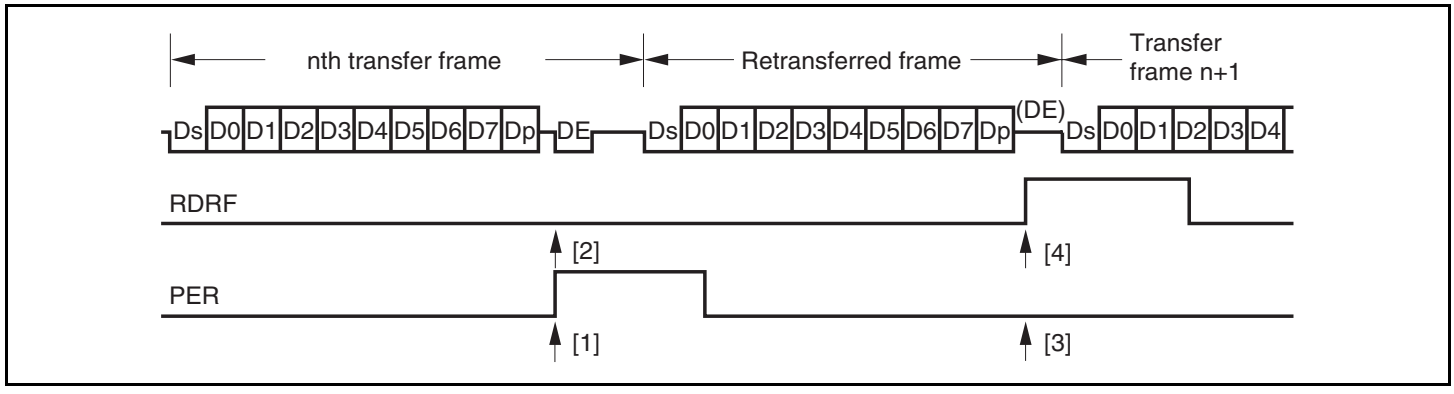


Figure 15.29 Retransfer Operation in SCI Receive Mode

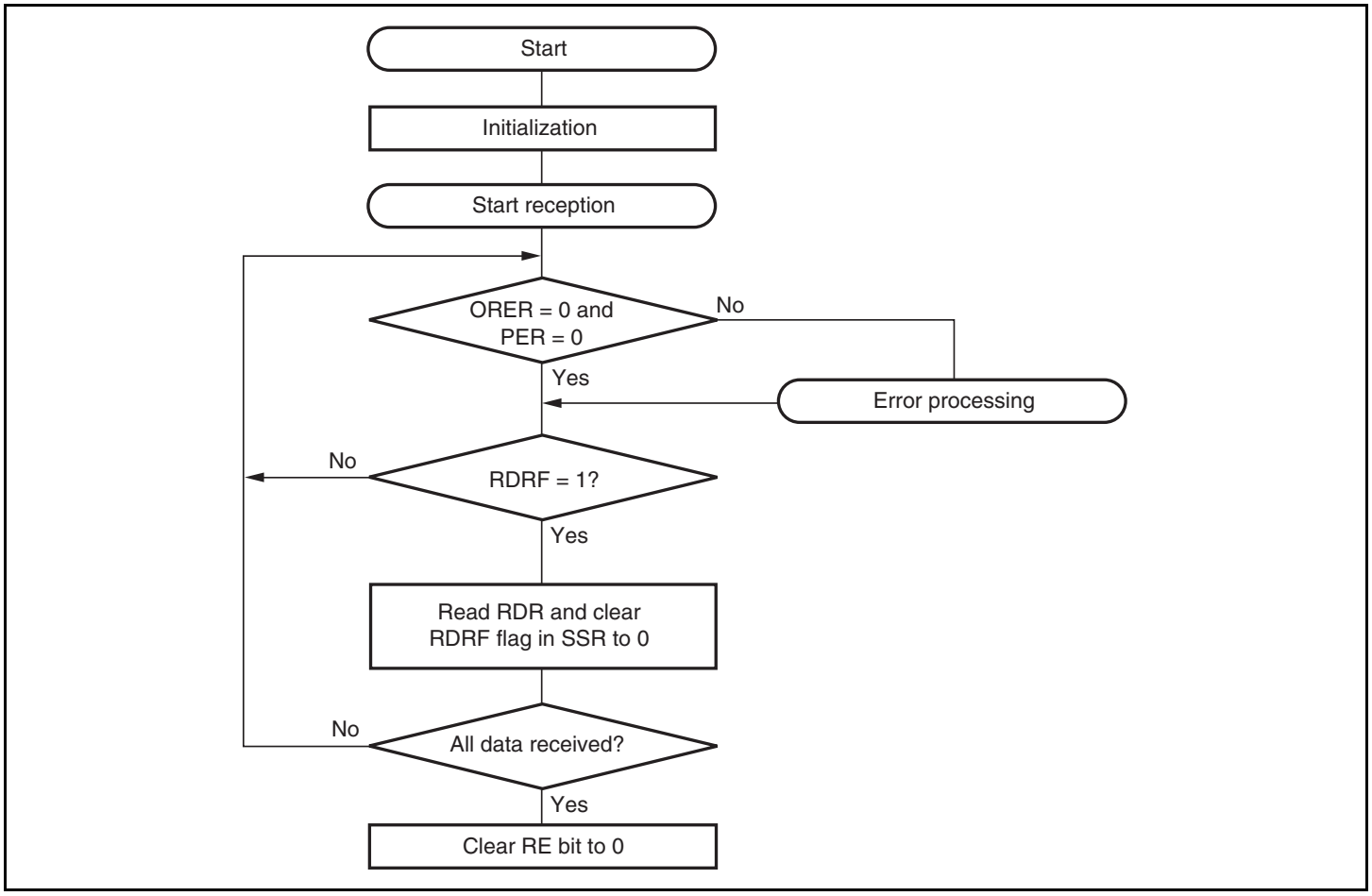


Figure 15.30 Example of Reception Processing Flow

15.7.8 Clock Output Control

When the GM bit in SMR is set to 1, the clock output level can be fixed with bits CKE1 and CKE0 in SCR. At this time, the minimum clock pulse width can be made the specified width.

Figure 15.31 shows the timing for fixing the clock output level. In this example, GM is set to 1, CKE1 is cleared to 0, and the CKE0 bit is controlled.

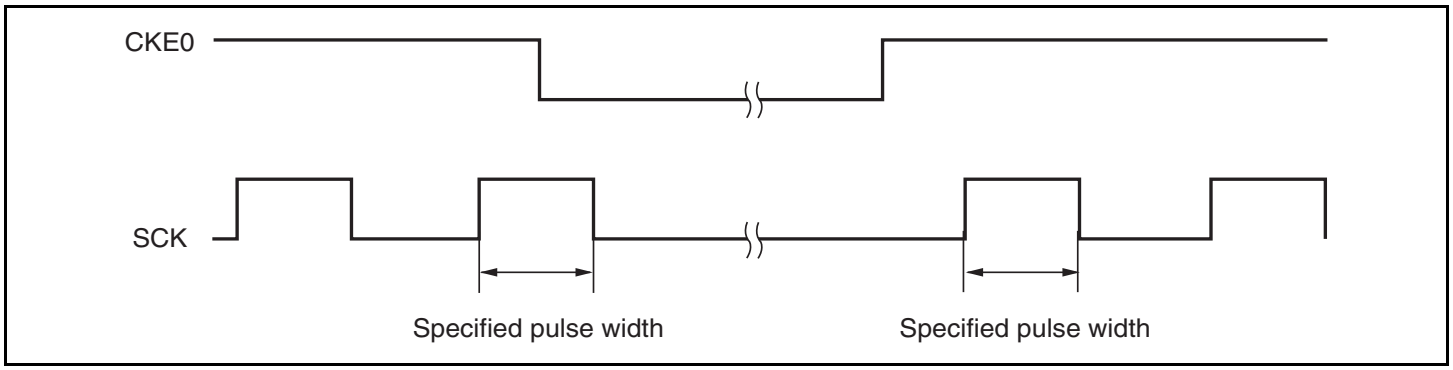


Figure 15.31 Timing for Fixing Clock Output Level

When turning on the power or switching between Smart Card interface mode and software standby mode, the following procedures should be followed in order to maintain the clock duty cycle.

Powering On: To secure the clock duty cycle from power-on, the following switching procedure should be followed.

1. The initial state is port input and high impedance. Use a pull-up resistor or pull-down resistor to fix the potential.
2. Fix the SCK pin to the specified output level with the CKE1 bit in SCR.
3. Set SMR and SCMR, and switch to Smart Card mode operation.
4. Set the CKE0 bit in SCR to 1 to start clock output.

When Changing from Smart Card Interface Mode to Software Standby Mode:

1. Set the data register (DR) and data direction register (DDR) corresponding to the SCK pin to the value for the fixed output state in software standby mode.
2. Write 0 to the TE bit and RE bit in the serial control register (SCR) to halt transmit/receive operation. At the same time, set the CKE1 bit to the value for the fixed output state in software standby mode.
3. Write 0 to the CKE0 bit in SCR to halt the clock.
4. Wait for one serial clock cycle.

   During this interval, clock output is fixed at the specified level, with the duty cycle preserved.
5. Make the transition to the software standby state.

When Returning to Smart Card Interface Mode from Software Standby Mode:

6. Exit the software standby state.
7. Write 1 to the CKE0 bit in SCR and output the clock. Signal generation is started with the normal duty cycle.

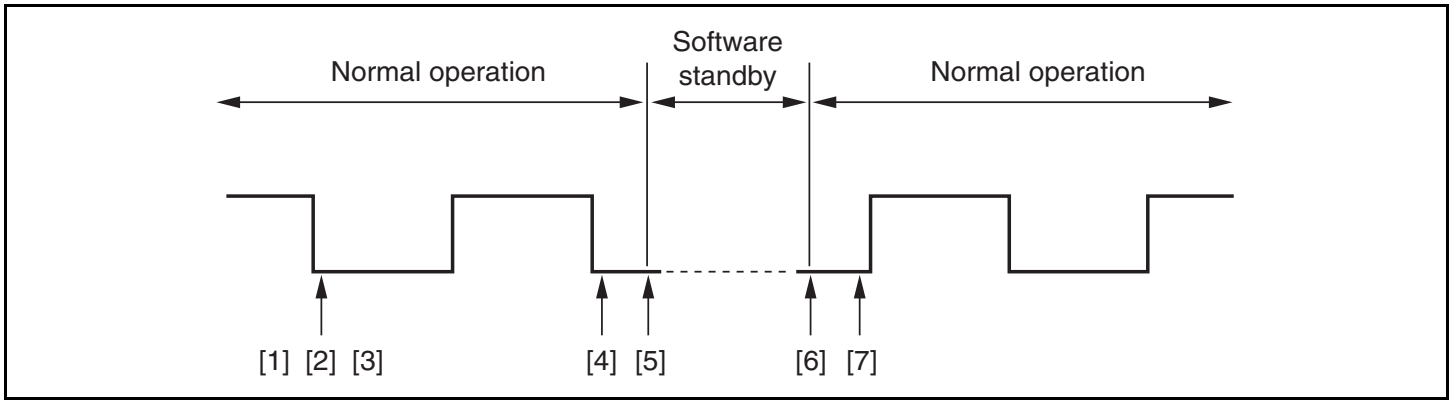


Figure 15.32 Clock Halt and Restart Procedure

15.8 IrDA Operation

When the IrDA function is enabled with bit IrE in IrCR, the SCI_0 TxD0 and RxD0 signals are subjected to waveform encoding/decoding conforming to IrDA specification version 1.0 (IrTxD and IrRxD pins). By connecting these pins to an infrared transceiver/receiver, it is possible to implement infrared transmission/reception conforming to the IrDA specification version 1.0 system.

In the IrDA specification version 1.0 system, communication is started at a transfer rate of 9600 bps, and subsequently the transfer rate can be varied as necessary. As the IrDA interface in this LSI does not include a function for varying the transfer rate automatically, the transfer rate setting must be changed by software.

Figure 15.33 shows a block diagram of the IrDA function.

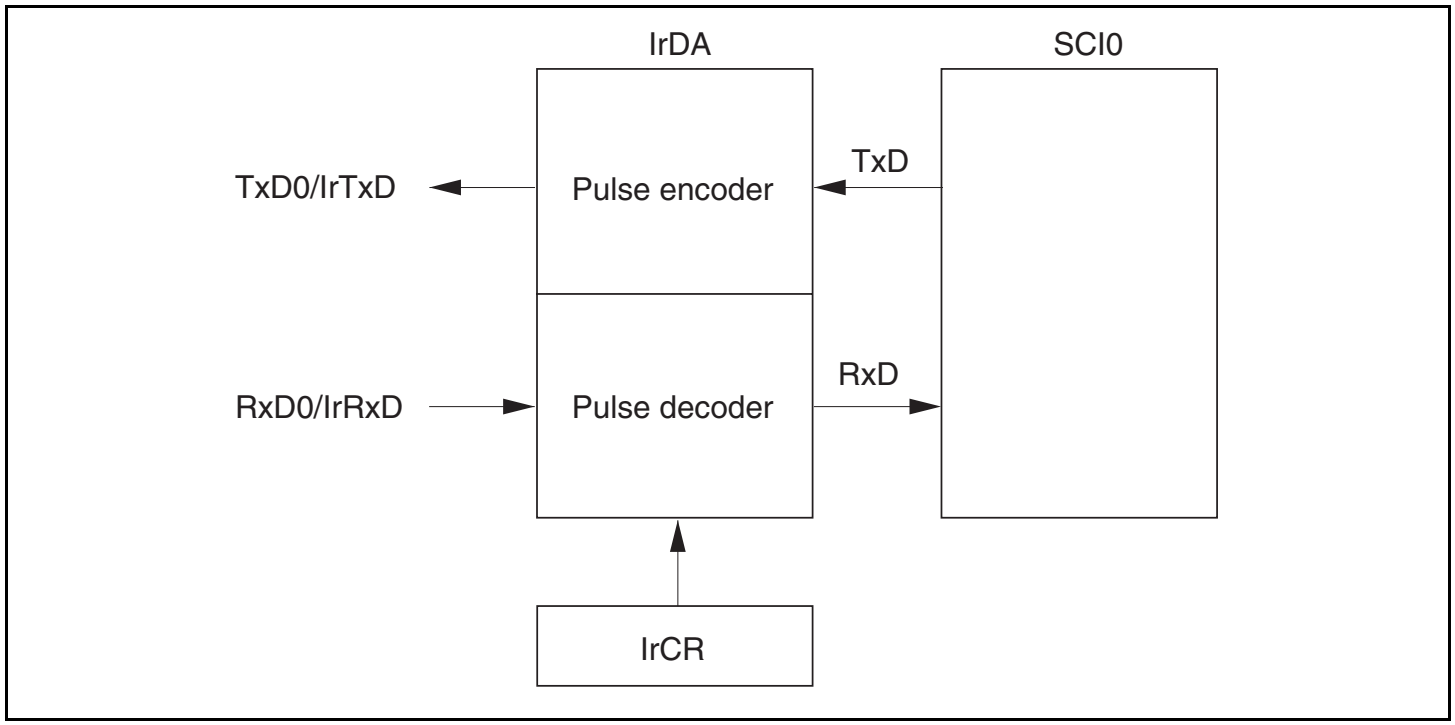


Figure 15.33 Block Diagram of IrDA

(1) Transmission

In transmission, the output signal (UART frame) from the SCI is converted to an IR frame by the IrDA interface (see figure 15.34).

When the serial data is 0, a high pulse of 3/16 the bit rate (interval equivalent to the width of one bit) is output (initial value). The high-level pulse can be varied according to the setting of bits IrCKS2 to IrCKS0 in IrCR.

In the specification, the high pulse width is fixed at a minimum of 1.41 µs, and a maximum of (3/16 + 2.5%) × bit rate or (3/16 × bit rate) + 1.08 µs. When system clock ϕ is 20 MHz, 1.6 µs can be set for a high pulse width with a minimum value of 1.41 µs.

When the serial data is 1, no pulse is output.

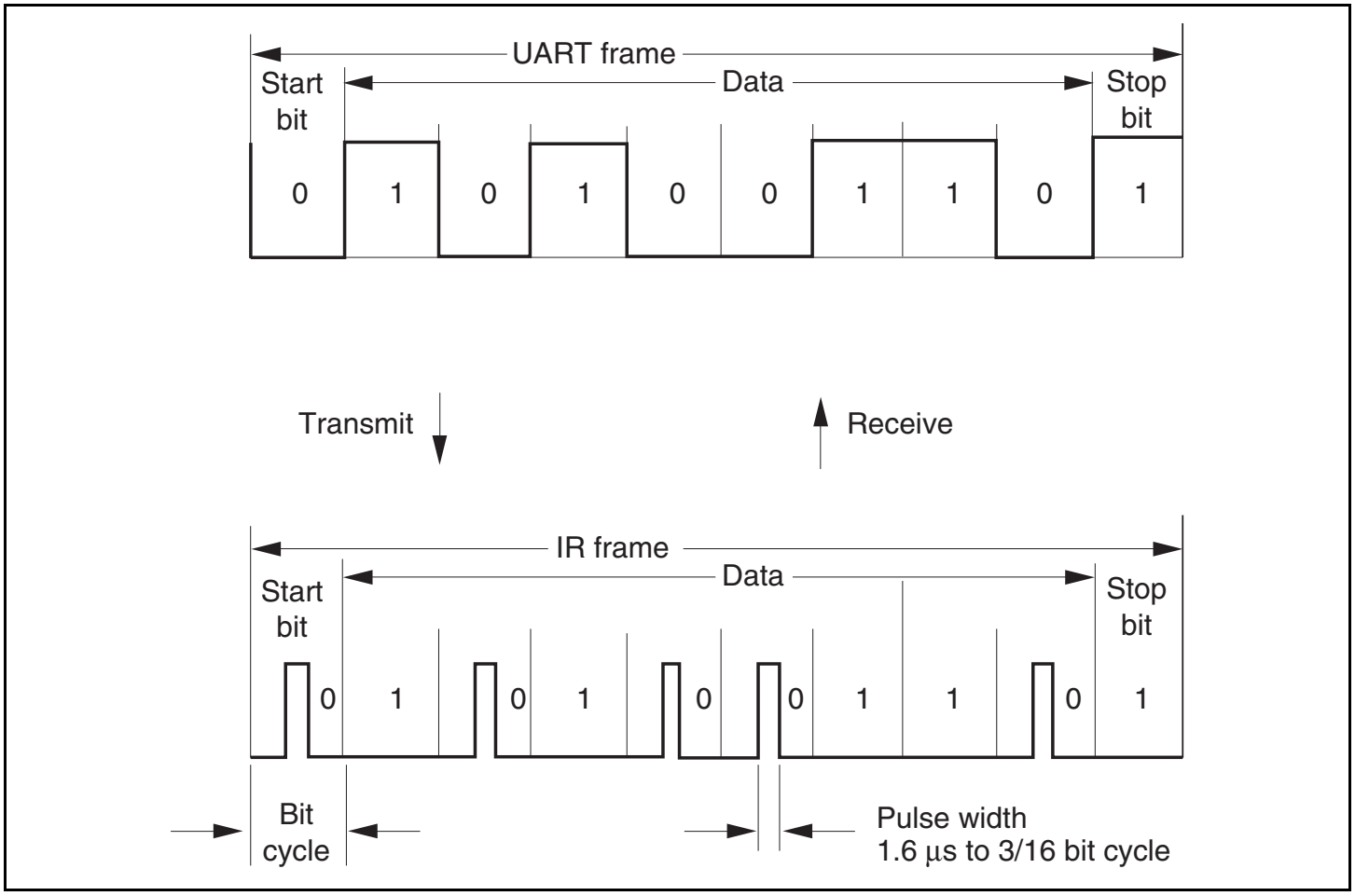


Figure 15.34 IrDA Transmit/Receive Operations

(2) Reception

In reception, IR frame data is converted to a UART frame by the IrDA interface, and input to the SCI.

When a high pulse is detected, 0 data is output, and if there is no pulse during a one-bit interval, 1 data is output. Note that a pulse shorter than the minimum pulse width of 1.41 µs will be identified as a 0 signal.

(3) High Pulse Width Selection

Table 15.12 shows possible settings for bits IrCKS2 to IrCKS0 (minimum pulse width), and operating frequencies of this LSI and bit rates, for making the pulse width shorter than 3/16 times the bit rate in transmission.

Table 15.12 Settings of IrCKS2 to IrCKS0 Bits

Operating Frequency ϕ (MHz)	Bit Rate (bps) (Above)/Bit Period × 3/16 (µs) (Below)					
	2400	9600	19200	38400	57600	115200
	78.13	19.53	9.77	4.88	3.26	1.63
8	100	100	100	100	100	100
9.8304	100	100	100	100	100	100
10	100	100	100	100	100	100
12	101	101	101	101	101	101
12.288	101	101	101	101	101	101
14	101	101	101	101	101	101
14.7456	101	101	101	101	101	101
16	101	101	101	101	101	101
16.9344	101	101	101	101	101	101
17.2032	101	101	101	101	101	101
18	101	101	101	101	101	101
19.6608	101	101	101	101	101	101
20	101	101	101	101	101	101
25	110	110	110	110	110	—
30	110	110	110	110	110	—
33	110	110	110	110	110	—

Legend:

—: A bit rate setting cannot be made on the SCI side.

15.9 Interrupt Sources

15.9.1 Interrupts in Normal Serial Communication Interface Mode

Table 15.13 shows the interrupt sources in normal serial communication interface mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled using the enable bits in SCR.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the TEND flag in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the DTC or DMAC to perform data transfer. The TDRE flag is cleared to 0 automatically when data transfer is performed by the DTC or DMAC.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interrupt request can activate the DTC or DMAC to perform data transfer. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC or DMAC.

A TEI interrupt is generated when the TEND flag is set to 1 while the TEIE bit is set to 1. If a TEI interrupt and a TXI interrupt are generated simultaneously, the TXI interrupt has priority for acceptance. However, note that if the TDRE and TEND flags are cleared simultaneously by the TXI interrupt routine, the SCI cannot branch to the TEI interrupt routine later.

Table 15.13 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority
0	ERI0	Receive Error	ORER, FER, PER	Not possible	Not possible	High
	RXI0	Receive Data Full	RDRF	Possible	Possible	
	TXI0	Transmit Data Empty	TDRE	Possible	Possible	
	TEI0	Transmission End	TEND	Not possible	Not possible	
1	ERI1	Receive Error	ORER, FER, PER	Not possible	Not possible	
	RXI1	Receive Data Full	RDRF	Possible	Possible	
	TXI1	Transmit Data Empty	TDRE	Possible	Possible	
	TEI1	Transmission End	TEND	Not possible	Not possible	
2	ERI2	Receive Error	ORER, FER, PER	Not possible	Not possible	
	RXI2	Receive Data Full	RDRF	Possible	Not possible	
	TXI2	Transmit Data Empty	TDRE	Possible	Not possible	
	TEI2	Transmission End	TEND	Not possible	Not possible	
3	ERI3	Receive Error	ORER, FER, PER	Not possible	Not possible	
	RXI3	Receive Data Full	RDRF	Possible	Not possible	
	TXI3	Transmit Data Empty	TDRE	Possible	Not possible	
	TEI3	Transmission End	TEND	Not possible	Not possible	
4	ERI4	Receive Error	ORER, FER, PER	Not possible	Not possible	
	RXI4	Receive Data Full	RDRF	Possible	Not possible	
	TXI4	Transmit Data Empty	TDRE	Possible	Not possible	
	TEI4	Transmission End	TEND	Not possible	Not possible	Low

15.9.2 Interrupts in Smart Card Interface Mode

Table 15.14 shows the interrupt sources in Smart Card interface mode. The transmit end interrupt (TEI) request cannot be used in this mode.

Table 15.14 Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority
0	ERI0	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	High
	RXI0	Receive Data Full	RDRF	Possible	Possible	
	TXI0	Transmit Data Empty	TEND	Possible	Possible	
1	ERI1	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	
	RXI1	Receive Data Full	RDRF	Possible	Possible	
	TXI1	Transmit Data Empty	TEND	Possible	Possible	
2	ERI2	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	
	RXI2	Receive Data Full	RDRF	Possible	Not possible	
	TXI2	Transmit Data Empty	TEND	Possible	Not possible	
3	ERI3	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	
	RXI3	Receive Data Full	RDRF	Possible	Not possible	
	TXI3	Transmit Data Empty	TEND	Possible	Not possible	
4	ERI4	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	
	RXI4	Receive Data Full	RDRF	Possible	Not possible	
	TXI4	Transmit Data Empty	TEND	Possible	Not possible	Low

In Smart Card interface mode, as in normal serial communication interface mode, transfer can be carried out using the DTC or DMAC. In transmit operations, the TDRE flag is also set to 1 at the same time as the TEND flag in SSR, and a TXI interrupt is generated. If the TXI request is designated beforehand as a DTC or DMAC activation source, the DTC or DMAC will be activated by the TXI request, and transfer of the transmit data will be carried out. The TDRE and TEND flags are automatically cleared to 0 when data transfer is performed by the DTC or DMAC. In the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC will automatically transmit the specified number of bytes in the event of an error, including retransmission. However, the ERS flag is not cleared automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When performing transfer using the DTC or DMAC, it is essential to set and enable the DTC or DMAC before carrying out SCI setting. For details on the DTC or DMAC setting procedures, refer to section 9, Data Transfer Controller (DTC) or section 7, DMA Controller (DMAC).

In receive operations, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DTC or DMAC activation source, the DTC or DMAC will be activated by the RXI request, and transfer of the receive data will be carried out. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC or DMAC. If an error occurs, an error flag is set but the RDRF flag is not. Consequently, the DTC or DMAC is not activated, but instead, an ERI interrupt request is sent to the CPU. Therefore, the error flag should be cleared.

15.10 Usage Notes

15.10.1 Module Stop Mode Setting

SCI operation can be disabled or enabled using the module stop control register. The initial setting is for SCI operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 25, Power-Down Modes.

15.10.2 Break Detection and Processing

When framing error detection is performed, a break can be detected by reading the RxD pin value directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag is set, and the PER flag may also be set. Note that, since the SCI continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

15.10.3 Mark State and Break Sending

When TE is 0, the TxD pin is used as an I/O port whose direction (input or output) and level are determined by DR and DDR. This can be used to set the TxD pin to mark state or send a break during serial data transmission. To maintain the communication line at mark state until TE is set to 1, set both DDR and DR to 1. Since TE is cleared to 0 at this point, the TxD pin becomes an I/O port, and 1 is output from the TxD pin. To send a break during serial transmission, first set DDR to 1 and clear DR to 0, and then clear TE to 0. When TE is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TxD pin becomes an I/O port, and 0 is output from the TxD pin.

15.10.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only)

Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

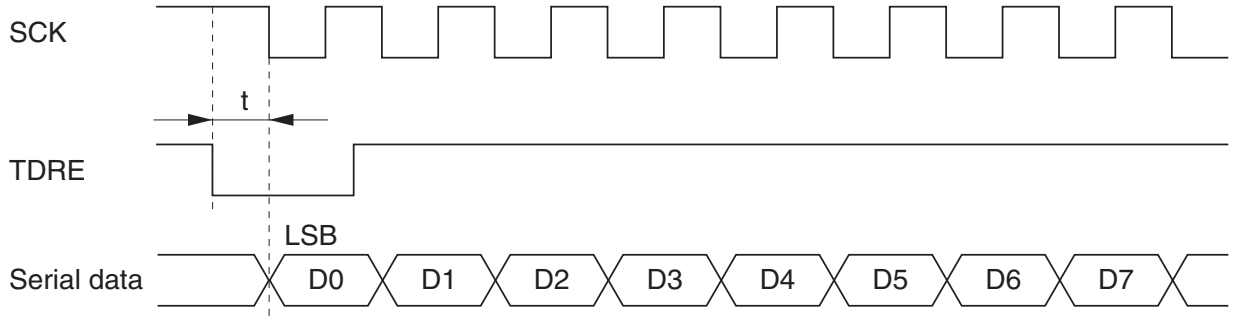
15.10.5 Relation between Writes to TDR and the TDRE Flag

The TDRE flag in SSR is a status flag that indicates that transmit data has been transferred from TDR to TSR. When the SCI transfers data from TDR to TSR, the TDRE flag is set to 1.

Data can be written to TDR regardless of the state of the TDRE flag. However, if new data is written to TDR when the TDRE flag is cleared to 0, the data stored in TDR will be lost since it has not yet been transferred to TSR. It is therefore essential to check that the TDRE flag is set to 1 before writing transmit data to TDR.

15.10.6 Restrictions on Use of DMAC or DTC

- When an external clock source is used as the serial clock, the transmit clock should not be input until at least 5 ϕ clock cycles after TDR is updated by the DMAC or DTC. Incorrect operation may occur if the transmit clock is input within 4 ϕ clocks after TDR is updated. (Figure 15.35)
- When RDR is read by the DMAC or DTC, be sure to set the activation source to the relevant SCI receive-data-full interrupt (RXI).



Note: When operating on an external clock, set t > 4 clocks.

Figure 15.35 Example of Synchronous Transmission Using DTC

15.10.7 Operation in Case of Mode Transition

- Transmission

  Operation should be stopped (by clearing TE, TIE, and TEIE to 0) before making a module stop mode or software standby mode transition. TSR, TDR, and SSR are reset. The output pin states in module stop mode or software standby mode depend on the port settings, and become high-level output after the relevant mode is cleared. If a transition is made during transmission, the data being transmitted will be undefined.

  When transmitting without changing the transmit mode after the relevant mode is cleared, transmission can be started by setting TE to 1 again, and performing the following sequence: SSR read → TDR write → TDRE clearance. To transmit with a different transmit mode after clearing the relevant mode, the procedure must be started again from initialization.

  Figure 15.36 shows a sample flowchart for mode transition during transmission. Port pin states during mode transition are shown in figures 15.37 and 15.38.

  Operation should also be stopped (by clearing TE, TIE, and TEIE to 0) before making a transition from transmission by DTC transfer to module stop mode or software standby mode transition. To perform transmission with the DTC after the relevant mode is cleared, setting TE and TIE to 1 will set the TXI flag and start DTC transmission.

- Reception

  Receive operation should be stopped (by clearing RE to 0) before making a module stop mode or software standby mode transition. RSR, RDR, and SSR are reset. If a transition is made during reception, the data being received will be invalid.

  To continue receiving without changing the reception mode after the relevant mode is cleared, set RE to 1 before starting reception. To receive with a different receive mode, the procedure must be started again from initialization.

  Figure 15.39 shows a sample flowchart for mode transition during reception.

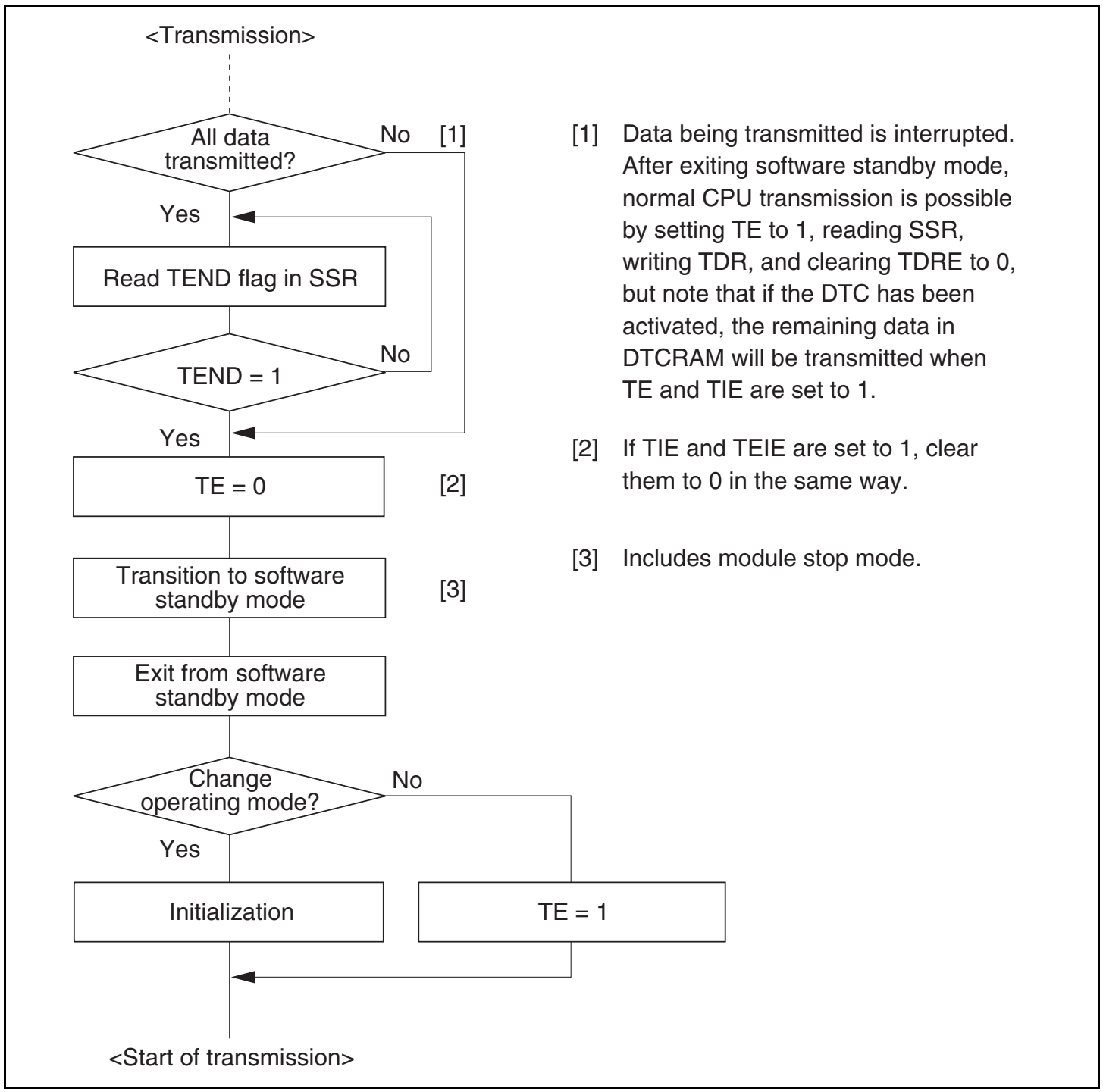


Figure 15.36 Sample Flowchart for Mode Transition during Transmission

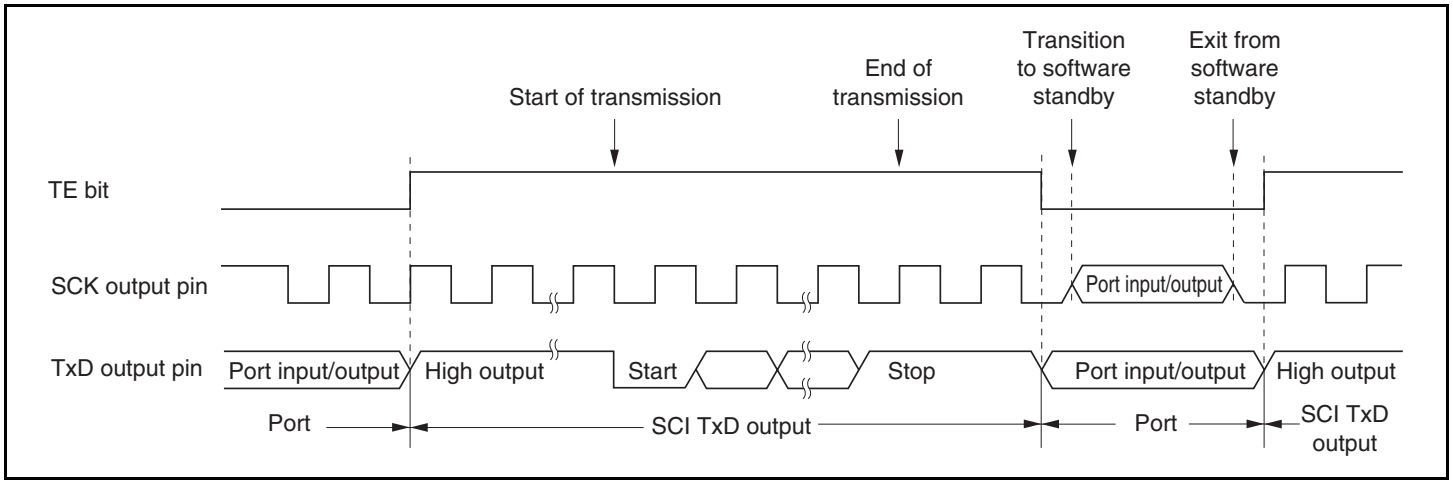


Figure 15.37 Port Pin States during Mode Transition (Internal Clock, Asynchronous Transmission)

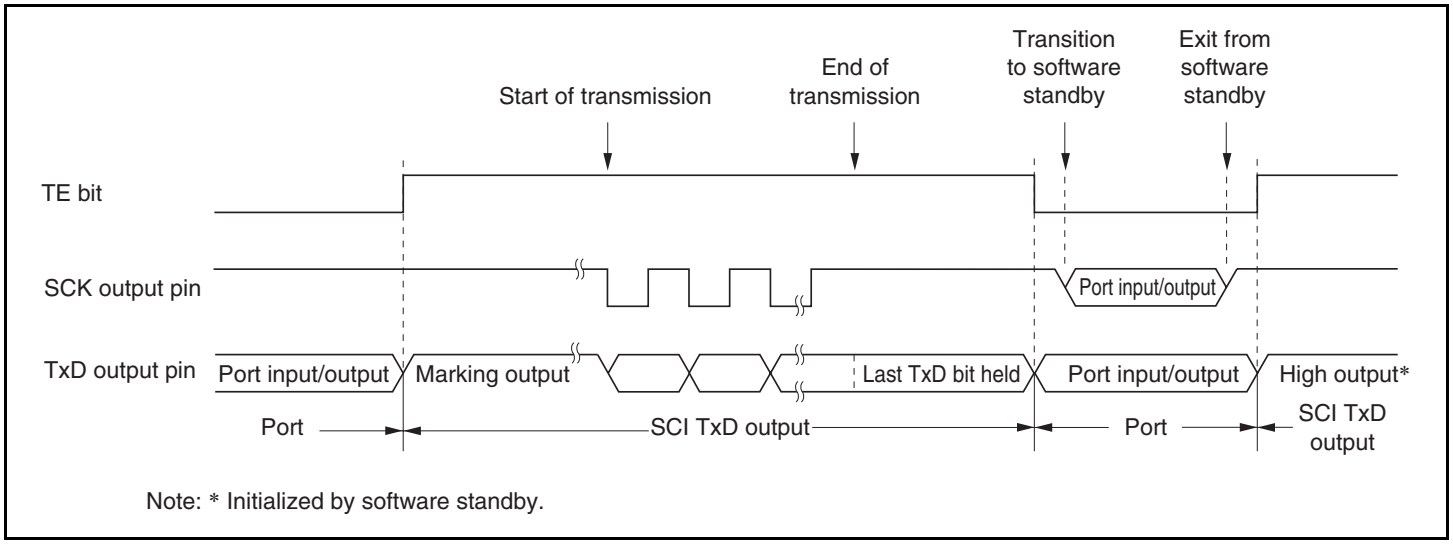


Figure 15.38 Port Pin States during Mode Transition (Internal Clock, Synchronous Transmission)

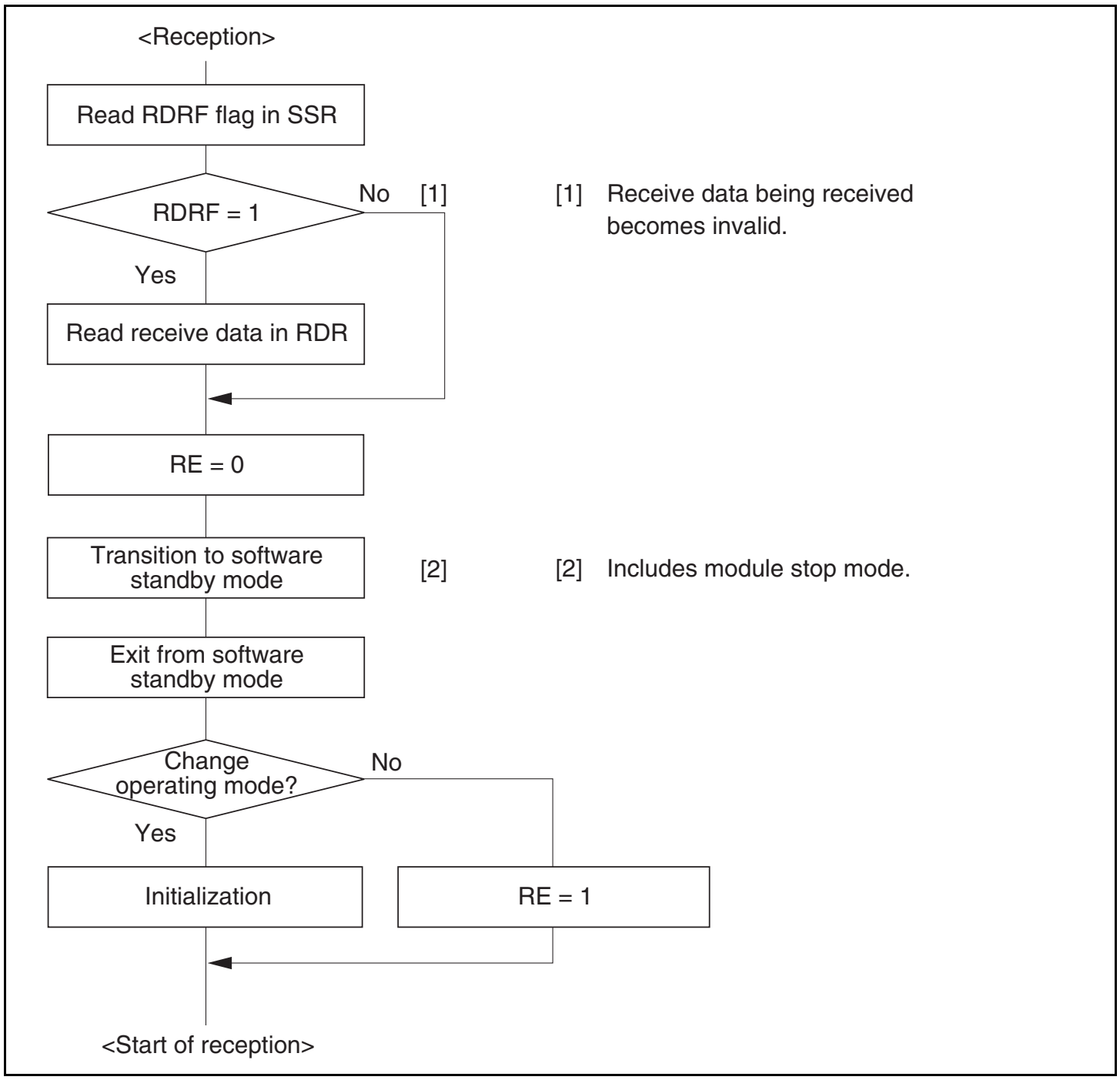


Figure 15.39 Sample Flowchart for Mode Transition during Reception

Section 16 USB Function Module (USB)

This LSI incorporates a USB function module (USB).

16.1 Features

- The protocol block conforming to USB2.0 and transceiver process USB protocol automatically.

  Automatic processing of USB standard commands for endpoint 0 (some commands and class/vendor commands require decoding and processing by firmware)
- Transfer speed: Supports full-speed (12 Mbps)
- Endpoint configuration:

Endpoint Name	Abbreviation	Transfer Type	Maximum Packet Size	FIFO Buffer Capacity (Byte)	DMA Transfer
Endpoint 0	EP0s	Setup	8	8	—
	EP0i	Control-in	16	16	—
	EP0o	Control-out	16	16	—
Endpoint 1	EP1	Bulk-out	64	128	Possible
Endpoint 2	EP2	Bulk-in	64	128	Possible
Endpoint 3	EP3	Interrupt-in	16	16	—

Configuration1-Interface0 to 3-AlternateSetting0- ◄——— EndPoint1 to 3

- Interrupt requests: Generates various interrupt signals necessary for USB transmission/reception
- Power mode: Self power mode or bus power mode can be selected by the power mode bit (PWMD) in the control register (CTLR).

Figure 16.1 shows the block diagram of the USB.

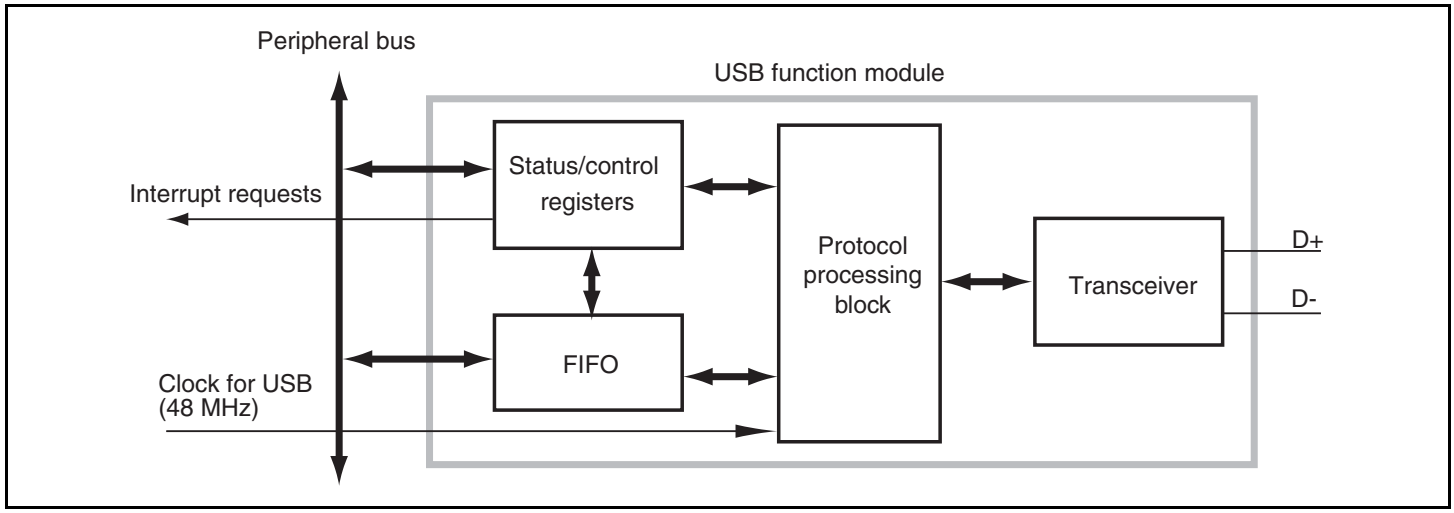


Figure 16.1 Block Diagram of USB

16.2 Input/Output Pins

Table 16.1 shows the USB pin configuration.

Table 16.1 Pin Configuration

Pin Name	I/O	Function
VBUS	Input	USB cable connection monitor pin
USD+	I/O	USB data I/O pin
USD-	I/O	USB data I/O pin
DrVcc	Input	Power supply pin for USB on-chip transceiver
DrVss	Input	Ground pin for USB on-chip transceiver

16.3 Register Descriptions

The USB has following registers. For the information on the addresses of these registers and the state of the register in each processing condition, see section 26, List of Registers.

- Interrupt flag register 0 (IFR0)
- Interrupt flag register 1 (IFR1)
- Interrupt flag register 2 (IFR2)
- Interrupt enable register 0 (IER0)
- Interrupt enable register 1 (IER1)
- Interrupt enable register 2 (IER2)
- Interrupt select register 0 (ISR0)
- Interrupt select register 1 (ISR1)
- Interrupt select register 2 (ISR2)
- EP0i data register (EPDR0i)
- EP0o data register (EPDR0o)
- EP0s data register (EPDR0s)
- EP1 data register (EPDR1)
- EP2 data register (EPDR2)
- EP3 data register (EPDR3)
- EP0o receive data size register (EPSZ0o)
- EP1 receive data size register (EPSZ1)
- Data status register 0 (DASTS0)
- Data status register 1 (DASTS1)
- Trigger register 0 (TRG0)
- Trigger register 1 (TRG1)
- FIFO clear register 0 (FCLR0)
- FIFO clear register 1 (FCLR1)
- Endpoint stall register 0 (EPSTL0)
- Endpoint stall register 1 (EPSTL1)
- Stall status register 1 (STLSR1)
- DMA transfer setting register (DMAR)
- Configuration value register (CVR)
- Control register (CTLR)
- Endpoint information register (EPIR)

- Transceiver test register 0 (TRNTREG0)
- Transceiver test register 1 (TRNTREG1)

16.3.1 Interrupt Flag Register 0 (IFR0)

IFR0, together with interrupt flag registers 1 and 2 (IFR1 and IFR2), indicates interrupt status information required by the application. When an interrupt source is generated, the corresponding bit is set to 1. And then this bit, in combination with interrupt enable register 0 (IER0), generates an interrupt request to the CPU. To clear, write 0 to the bit to be cleared and 1 to the other bits. However, since SURSS and VBUSMN are status bits, these bits cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	BRST	0	R/W	Bus Reset This bit is set to 1 when a bus reset signal is detected on the USB bus.
6	CFDN	0	R/W	End Point Information Load End This bit is set to 1 when writing data in the endpoint information register to the EPIR register ends (load end). This module starts the USB operation after the endpoint information is completely set.
5	SURSS	0	R	Suspend/Resume Status This is a status bit that describes bus state. 0: Normal state 1: Suspended state This is a status bit and cannot be cleared. It generates no interrupt request.
4	SURSF	0	R/W	Suspend/Resume Detection This bit is set to 1 when the state changed from normal to suspended state or vice versa. The corresponding interrupt output is RESUME, USBINTN2, and USBINTN3.
3	SETC	0	R/W	Set_Configuration Command Detection When the Set_Configuration command is detected, this bit is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
2	SETI	0	R/W	Set_Interface Command Detection When the Set_Interface command is detected, this bit is set to 1.
1	VBUSMN	0	R	VBUS Pin State Monitor This is a status bit that monitors the state of the VBUS pin. 0: VBUS pin = 0 1: VUBS pin = 1 This is a status bit and cannot be cleared. It generates no interrupt request. This bit is always 0 when the PULLUPE bit in CTLR is 0.
0	VBUSF	0	R/W	USB BUS Connection/Disconnection Detection When the function is connected to the USB bus or disconnected from it, this bit is set to 1. The VBUS pin of this module is used for detecting connection or disconnection.

16.3.2 Interrupt Flag Register 1 (IFR1)

IFR1, together with interrupt flag registers 0 and 2 (IFR0 and IFR2), indicates interrupt status information required by the application. When an interrupt source is generated, the corresponding bit is set to 1. And then this bit, in combination with interrupt enable register 1 (IER1), generates an interrupt request to the CPU. To clear, write 0 to the bit to be cleared and 1 to the other bits.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0. The write value should always be 0.
5	—	0	—	
4	SOF	0	R/W	SOF Packet Detection This bit is set to 1 when the Start Of Frame (SOF) packet is detected.
3	SETUP TS	0	R/W	Setup Command Receive Complete This bit is set to 1 when endpoint 0 receives successfully a setup command requiring decoding on the application side, and returns an ACK handshake to the host.
2	EP0o TS	0	R/W	EP0o Receive Complete This bit is set to 1 when endpoint 0 receives data from the host successfully, stores the data in the FIFO buffer, and returns an ACK handshake to the host.
1	EP0i TR	0	R/W	EP0i Transfer Request This bit is set if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 0 is received from the host. A NAK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
0	EP0i TS	0	R/W	EP0i Transmit Complete This bit is set when data is transmitted to the host from endpoint 0 and an ACK handshake is returned.

16.3.3 Interrupt Flag Register 2 (IFR2)

IFR2, together with interrupt flag registers 0 and 1, (IFR0 and IFR1), indicates interrupt status information required by the application. When an interrupt source is generated, the corresponding bit is set to 1. And then this bit, in combination with interrupt enable register 2 (IER2), generates an interrupt request to the CPU. To clear, write 0 to the bit to be cleared and 1 to the other bits. However, since EP2 EMPTY, EP2 ALLEMP, and EP1 FULL are status bits, these bits cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0. The write value should always be 0.
5	EP3 TR	0	R/W	EP3 Transfer Request This bit is set if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 3 is received from the host. A NAK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
4	EP3 TS	0	R/W	EP3 Transmit Complete This bit is set when data is transmitted to the host from endpoint 3 and an ACK handshake is returned.
3	EP2 TR	0	R/W	EP2 Transfer Request This bit is set if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 2 is received from the host. A NAK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
2	EP2 EMPTY	1	R	EP2 FIFO Empty This bit is set when at least one of the dual endpoint 2 transmit FIFO buffers is ready for transmit data to be written. This is a status bit and cannot be cleared.
1	EP2 ALLEMP	1	R	EP2 FIFO All Empty This bit is set when both of the dual endpoint 2 transmit FIFO buffers are empty. This is a status bit and cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
0	EP1 FULL	0	R	EP1 FIFO Full This bit is set when endpoint 1 receives one packet of data successfully from the host, and holds a value of 1 as long as there is valid data in the FIFO buffer. This is a status bit and cannot be cleared.

16.3.4 Interrupt Enable Register 0 (IER0)

IER0 enables the interrupt requests of interrupt flag register 0 (IFR0). When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, an interrupt request is sent to the CPU. The interrupt vector number is determined by the contents of interrupt select register 0 (ISR0).

Bit	Bit Name	Initial Value	R/W	Description
7	BRSTE	0	R/W	Bus Reset
6	CFDNE	0	R/W	End Point Information Load End
5	SSRSME	0	R/W	Resume Detection for Software Standby Cancel For details of the operation, see section 16.5.4, Suspend and Resume Operations.
4	SURSFE	0	R/W	Suspend/Resume Detection For details of the operation, see section 16.5.4, Suspend and Resume Operations.
3	SETCE	0	R/W	Set_Configuration Command Detection
2	SETIE	0	R/W	Set_Interface Command Detection
1	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
0	VBUSFE	0	R/W	USB Bus Connection/Disconnection

16.3.5 Interrupt Enable Register 1 (IER1)

IER1 enables the interrupt requests of interrupt flag register 1 (IFR1). When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, an interrupt request is sent to the CPU. The interrupt vector number is determined by the contents of interrupt select register 1 (ISR1).

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0. The write value should always be 0.
5	—	0	—	
4	SOFE	0	R/W	SOF Packet Detection
3	SETUP TSE	0	R/W	Setup Command Receive Complete
2	EP0o TSE	0	R/W	EP0o Receive Complete
1	EP0i TRE	0	R/W	EP0i Transfer Request
0	EP0i TSE	0	R/W	EP0i Transmission Complete

16.3.6 Interrupt Enable Register 2 (IER2)

IER2 enables the interrupt requests of interrupt flag register 2 (IFR2). When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, an interrupt request is sent to the CPU. The interrupt vector number is determined by the contents of interrupt select register 2 (ISR2).

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0. The write value should always be 0.
5	EP3 TRE	0	R/W	EP3 Transfer Request
4	EP3 TSE	0	R/W	EP3 Transmission Complete
3	EP2 TRE	0	R/W	EP2 Transfer Request
2	EP2 EMPTYE	0	R/W	EP2 FIFO Empty
1	EP2 ALLEMPE	0	R/W	EP2 FIFO All Empty
0	EP1 FULLE	0	R/W	EP1 FIFO Full

16.3.7 Interrupt Select Register 0 (ISR0)

ISR0 selects the vector numbers of the interrupt requests indicated in interrupt flag register 0 (IFR0). If the USB issues an interrupt request to the INTC when a bit in ISR0 is cleared to 0, the interrupt corresponding to the bit will be USBINTN2. If the USB issues an interrupt request to the INTC when a bit in ISR0 is set to 1, the corresponding interrupt will be USBINTN3.

Bit	Bit Name	Initial Value	R/W	Description
7	BRSTS	0	R/W	Bus Reset
6	CFDNS	0	R/W	End Point Information Load End
5	—	0	—	Reserved This bit is always read as 1. The write value should always be 1.
4	SURSFS	0	R/W	Suspend/Resume Detection
3	SETCS	0	R/W	Set_Configuration Command Detection
2	SETIS	0	R/W	Set_Interface Command Detection
1	—	0	—	Reserved This bit is always read as 1. The write value should always be 1.
0	VBUSFS	1	R/W	USB Bus Connection/Disconnection

16.3.8 Interrupt Select Register 1 (ISR1)

ISR1 selects the vector numbers of the interrupt requests indicated in interrupt flag register 1 (IFR1). If the USB issues an interrupt request to the INTC when a bit in ISR1 is cleared to 0, the interrupt corresponding to the bit will be USBINTN2. If the USB issues an interrupt request to the INTC when a bit in ISR1 is set to 1, the corresponding interrupt will be USBINTN3.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0. The write value should always be 0.
5	—	0	—	
4	SOFS	0	R/W	SOF Packet Detection
3	SETUP TSS	0	R/W	Setup Command Receive Complete
2	EP0o TSS	0	R/W	EP0o Receive Complete
1	EP0i TRS	0	R/W	EP0i Transfer Request
0	EP0i TSS	0	R/W	EP0i Transmission Complete

16.3.9 Interrupt Select Register 2 (ISR2)

ISR2 selects the vector numbers of the interrupt requests indicated in interrupt flag register 2 (IFR2). If the USB issues an interrupt request to the INTC when a bit in ISR2 is cleared to 0, the interrupt corresponding to the bit will be USBINTN2. If the USB issues an interrupt request to the INTC when a bit in ISR2 is set to 1, the corresponding interrupt will be USBINTN3.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0. The write value should always be 0.
5	EP3 TRS	0	R/W	EP3 Transfer Request
4	EP3 TSS	0	R/W	EP3 Transmission Complete
3	EP2 TRS	0	R/W	EP2 Transfer Request
2	EP2 EMPTYS	0	R/W	EP2 FIFO Empty
1	EP2 ALLEMPS	0	R/W	EP2 FIFO All Empty
0	EP1 FULLS	0	R/W	EP1 FIFO Full

16.3.10 EP0i Data Register (EPDR0i)

EPDR0i is a 16-byte transmit FIFO buffer for endpoint 0. EPDR0i holds one packet of transmit data for control-in. Transmit data is fixed by writing one packet of data and setting EP0i PKTE in trigger register 0. When an ACK handshake is returned from the host after the data has been transmitted, EP0i TS in interrupt flag register 1 is set. This FIFO buffer can be initialized by means of EP0i CLR in FCLR register 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for control-in transfer

16.3.11 EP0o Data Register (EPDR0o)

EPDR0o is a 16-byte receive FIFO buffer for endpoint 0. EPDR0o holds endpoint 0 receive data other than setup commands. When data is received successfully, EP0o TS in interrupt flag register 1 is set, and the number of receive bytes is indicated in the EP0o receive data size register. After the data has been read, setting EP0o RDFN in trigger register 0 enables the next packet to be received. This FIFO buffer can be initialized by means of EP0o CLR in FCLR register 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	R	Data register for control-out transfer

16.3.12 EP0s Data Register (EPDR0s)

EPDR0s is an 8-byte FIFO buffer specifically for receiving endpoint 0 setup commands. Only the setup command to be processed by the application is received. When command data is received successfully, the SETUPTS bit in interrupt flag register 1 is set.

As a latest setup command must be received in high priority, if data is left in this buffer, it will be overwritten with new data. If reception of the next command is started while the current command is being read, command reception has priority, the read by the application is forcibly stopped, and the read data is invalid.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	R	Data register for storing the setup command at the control-out transfer

16.3.13 EP1 Data Register (EPDR1)

EPDR1 is a 128-byte receive FIFO buffer for endpoint 1. EPDR1 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. When one packet of data is received successfully, EP1 FULL in interrupt flag register 2 is set, and the number of receive bytes is indicated in the EP1 receive data size register. After the data has been read, the buffer that was read is enabled to receive data again by writing 1 to the EP1 RDFN bit in trigger register 1. The receive data in this FIFO buffer can be transferred by DMA. This FIFO buffer can be initialized by means of EP1 CLR in FCLR register 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	R	Data register for endpoint 1 transfer

16.3.14 EP2 Data Register (EPDR2)

EPDR2 is a 128-byte transmit FIFO buffer for endpoint 2. EPDR2 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. When transmit data is written to this FIFO buffer and EP2 PKTE in trigger register 1 is set, one packet of transmit data is fixed, and the dual-FIFO buffer is switched over. The transmit data for this FIFO buffer can be transferred by DMA. This FIFO buffer can be initialized by means of EP2 CLR in FCLR register 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for endpoint 2 transfer

16.3.15 EP3 Data Register (EPDR3)

EPDR3 is a 16-byte transmit FIFO buffer for endpoint 3. EPDR3 holds one packet of transmit data for the interrupt transfer of endpoint 3. Transmit data is fixed by writing one packet of data and setting EP3 PKTE in trigger register 1. This FIFO buffer can be initialized by means of EP3 CLR in FCLR register 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for endpoint 3 transfer

16.3.16 EP0o Receive Data Size Register (EPSZ0o)

EPSZ0o indicates the number of bytes received at endpoint 0 from the host.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits are always read as 0.
4 to 0	D4 to D0	All 0	R	Number of receive data for endpoint 0

16.3.17 EP1 Receive Data Size Register (EPSZ1)

EPSZ1 is a receive data size resister for endpoint 1. EPSZ1 indicates the number of bytes received from the host. The FIFO for endpoint 1 has a dual-buffer configuration. The size of the received data indicated by this register is the size of the currently selected side (can be read by CPU).

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6 to 0	D6 to D0	All 0	R	Number of received bytes for endpoint 1

16.3.18 Data Status Register 0 (DASTS0)

DASTS0 indicates whether the transmit FIFO buffers contain valid data. A bit is set when data is written to the corresponding FIFO buffer and the packet enable state is set, and cleared when all data has been transmitted to the host.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0. The write value should always be 0.
5	—	0	—	
4	—	0	—	
3	—	0	—	
2	—	0	—	
1	—	0	—	
0	EP0i DE	0	R	EP0i Data Present This bit is set when the endpoint 0i FIFO buffer contains valid data.

16.3.19 Data Status Register 1 (DASTS1)

DASTS1 indicates whether the transmit FIFO buffers contain valid data. A bit is set when data is written to the corresponding FIFO buffer and the packet enable state is set, and cleared when all data has been transmitted to the host.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0. The write value should always be 0.
5	—	0	—	
4	—	0	—	
3	—	0	—	
2	EP3 DE	0	R	EP3 Data Present This bit is set when the endpoint 3 FIFO buffer contains valid data.
1	EP2 DE	0	R	EP2 Data Present This bit is set when the endpoint 2 FIFO buffer contains valid data.
0	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.

16.3.20 Trigger Register 0 (TRG0)

TRG0 generates one-shot triggers to control the transfer sequence for endpoint 0.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0. The write value should always be 0.
5	—	0	—	
4	—	0	—	
3	—	0	—	
2	EP0s RDFN	0	W	EP0s Read Complete Write 1 to this bit after data for the EP0s command FIFO has been read. Writing 1 to this bit enables transfer of data in the following data stage. A NAK handshake is returned in response to transfer requests from the host in the data stage until 1 is written to this bit.
1	EP0o RDFN	0	W	EP0o Read Complete Writing 1 to this bit after one packet of data has been read from the endpoint 0 transmit FIFO buffer initializes the FIFO buffer, enabling the next packet to be received.
0	EP0i PKTE	0	W	EP0i Packet Enable After one packet of data has been written to the endpoint 0 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.

16.3.21 Trigger Register 1 (TRG1)

TRG1 generates one-shot triggers to control the transfer sequence for each endpoint.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0. The write value should always be 0.
5	—	0	—	
4	—	0	—	
3	—	0	—	
2	EP3 PKTE	0	W	EP3 Packet Enable After one packet of data has been written to the endpoint 3 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.
1	EP2 PKTE	0	W	EP2 Packet Enable After one packet of data has been written to the endpoint 2 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.
0	EP1 RDFN	0	W	EP1 Read Complete Write 1 to this bit after one packet of data has been read from the endpoint 1 FIFO buffer. The endpoint 1 receive FIFO buffer has a dual-buffer configuration. Writing 1 to this bit initializes the FIFO that was read, enabling the next packet to be received.

16.3.22 FIFO Clear Register 0 (FCLR0)

FCLR0 is a register to initialize the FIFO buffers for endpoint 0. Writing 1 to a bit clears all the data in the corresponding FIFO buffer. Note that the corresponding interrupt flag is not cleared. Do not clear a FIFO buffer during transfer.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	The write value should always be 0.
5	—	0	—	
4	—	0	—	
3	—	0	—	
2	—	0	—	
1	EP0o CLR	0	W	EP0o Clear Writing 1 to this bit initializes the endpoint 0 receive FIFO buffer.
0	EP0i CLR	0	W	EP0i Clear Writing 1 to this bit initializes the endpoint 0 transmit FIFO buffer.

16.3.23 FIFO Clear Register 1 (FCLR1)

FCLR1 is a register to initialize the FIFO buffers for each endpoint. Writing 1 to a bit clears all the data in the corresponding FIFO buffer. Note that the corresponding interrupt flag is not cleared. Do not clear a FIFO buffer during transfer.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	The write value should always be 0.
5	—	0	—	
4	—	0	—	
3	—	0	—	
2	EP3 CLR	0	W	EP3 Clear Writing 1 to this bit initializes the endpoint 3 transmit FIFO buffer.
1	EP2 CLR	0	W	EP2 Clear Writing 1 to this bit initializes both sides of the endpoint 2 transmit FIFO buffer.
0	EP1 CLR	0	W	EP1 Clear Writing 1 to this bit initializes both sides of the endpoint 1 receive FIFO buffer.

16.3.24 Endpoint Stall Register 0 (EPSTL0)

Bit 0 in EPSTL0 is used to forcibly stall endpoint 0 on the application side. While the bit is set to 1, the corresponding endpoint returns a stall handshake to the host. Bit 4 is used to clear the stall setting in bit 0. Writing 1 to the EP0 stall setting bit and stall clear bit at the same time is prohibited.

The stall bit for endpoint 0 is cleared automatically on reception of 8-byte setup command data for which decoding is performed by firmware the EP0 STLS bit is cleared. When the SETUPTS flag in the IFR1 is set to 1, writing 1 to the EP0 STLS bit is ignored. For detailed operation, see section 16.7, Stall Operations.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0. The write value should always be 0.
5	—	0	—	
4	EP0 STLC	0	W	EP0 Stall Clear Writing 1 to this bit clears the EP0 STLS bit to 0. Writing 0 is ignored.
3	—	0	—	Reserved
2	—	0	—	These bits are always read as 0. The write value should always be 0.
1	—	0	—	
0	EP0 STLS	0	R/W	EP0 Stall Setting Writing 1 to this bit specifies a stall for EP0. Writing 0 is ignored.

16.3.25 Endpoint Stall Register 1 (EPSTL1)

Bits 2 to 0 in EPSTL1 are used to forcibly stall the corresponding endpoints on the application side. While a bit is set to 1, the corresponding endpoint returns a stall handshake to the host. Bits 6 to 4 are used to clear the stall settings for the endpoints (bits 2 to 0). Writing 1 to the stall setting bit and stall clear bit for an endpoint at the same time is prohibited.

For detailed operation, see section 16.7, Stall Operations.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
6	EP3 STLC	0	W	EP3 Stall Clear Writing 1 to this bit clears the EP3 STLS bit to 0. Writing 0 is ignored.
5	EP2 STLC	0	W	EP2 Stall Clear Writing 1 to this bit clears the EP2 STLS bit to 0. Writing 0 is ignored.
4	EP1 STLC	0	W	EP1 Stall Clear Writing 1 to this bit clears the EP1 STLS bit to 0. Writing 0 is ignored.
3	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
2	EP3 STLS	0	R/W	EP3 Stall Setting Writing 1 to this bit specifies a stall for EP3. Writing 0 is ignored.
1	EP2 STLS	0	R/W	EP2 Stall Setting Writing 1 to this bit specifies a stall for EP2. Writing 0 is ignored.
0	EP1 STLS	0	R/W	EP1 Stall Setting Writing 1 to this bit specifies a stall for EP1. Writing 0 is ignored.

16.3.26 Stall Status Register 1 (STLSR1)

Bits 2 to 0 in STLSR1 are status bits that indicate the internal stall state of each endpoint (internal status bits shown in figures 16.19 and 16.20). When a bit is 1, the corresponding endpoint is in stall state. When a bit is 0, the corresponding endpoint is in normal operation state. Since these bits are status bits, they cannot be cleared.

Bits 6 to 4 in STLSR1 are used to enable automatic stall clear for each endpoint.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
6	EP3 ASCE	0	R/W	EP3 Automatic Stall Clear Enable Setting the EP3 ASCE bit to 1 automatically clears the EP3 stall setting bit (the EP3 STLS bit in EPSTL1) after the stall handshake is returned to the host. When the EP3 ASCE bit is set to 0, the stall setting bit is not automatically cleared and must be cleared by the users. To enable the automatic stall clear function, make sure that the EP3 ASCE bit should be set to 1 before the EP3 STLS bit in EPSTL1 is set to 1.
5	EP2 ASCE	0	R/W	EP2 Automatic Stall Clear Enable Setting the EP2 ASCE bit to 1 automatically clears the EP2 stall setting bit (the EP2 STLS bit in EPSTL1) after the stall handshake is returned to the host. When the EP2 ASCE bit is set to 0, the stall setting bit is not automatically cleared and must be cleared by the users. To enable the automatic stall clear function, make sure that the EP2 ASCE bit should be set to 1 before the EP2 STLS bit in EPSTL1 is set to 1.
4	EP1 ASCE	0	R/W	EP1 Automatic Stall Clear Enable Setting the EP1 ASCE bit to 1 automatically clears the EP1 stall setting bit (the EP1 STLS bit in EPSTL1) after the stall handshake is returned to the host. When the EP1 ASCE bit is set to 0, the stall setting bit is not automatically cleared and must be cleared by the users. To enable the automatic stall clear function, make sure that the EP1 ASCE bit should be set to 1 before the EP1 STLS bit in EPSTL1 is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
2	EP3 STLST	0	R	EP3 internal stall state
1	EP2 STLST	0	R	EP2 internal stall state
0	EP1 STLST	0	R	EP1 internal stall state

16.3.27 DMA Transfer Setting Register (DMAR)

DMA transfer can be carried out between the data registers for endpoints 1 and 2 and memory by means of the on-chip direct memory access controller (DMAC). Dual address transfer is performed in bytes. To start DMA transfer, DMAC settings must be made in addition to the settings in this register.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0. The write value should always be 0.
5	—	0	—	
4	—	0	R/W	Reserved The write value should always be 0.
3	—	0	R/W	Reserved The write value should always be 0.
2	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	EP2 DMAE	0	R/W	EP2 DMA Transfer Enable When this bit is set, DMA transfer is enabled from memory to the endpoint 2 transmit FIFO buffer. If there is at least one byte of open space in the FIFO buffer, a DMA transfer request signal (USB INTN1) is asserted. In DMA transfer, when 64 bytes are written to the FIFO buffer the EP2 packet enable bit is set automatically, allowing 64 bytes of data to be transferred, and if there is still space in the other side of the two FIFOs, the DMA transfer request signal (USB INTN1) is asserted again. However, if the size of the data packet to be transmitted is less than 64 bytes, the EP2 packet enable bit is not set automatically, and so should be set by the CPU with a DMA transfer end interrupt. As EP2-related interrupt requests to the CPU are not automatically masked, interrupt requests should be masked as necessary in the interrupt enable register. • Operating procedure 1. Write of 1 to the EP2 DMAE bit in DMAR 2. Set the DMAC to activate through $\overline{\text{DREQ1}}$ (USB INTN1) 3. Transfer count setting in the DMAC 4. DMAC activation 5. DMA transfer 6. DMA transfer end interrupt generated See section 16.8.4, DMA Transfer for Endpoints 2.

Bit	Bit Name	Initial Value	R/W	Description
0	EP1 DMAE	0	R/W	EP1 DMA Transfer Enable When this bit is set, a DMA transfer request (USB INTN0) is asserted and DMA transfer is enabled from the endpoint 1 receive FIFO buffer to memory. If there is at least one byte of receive data in the FIFO buffer, the DMA transfer request (USB INTN0) is asserted. In DMA transfer, when all the received data is read, EP1 is automatically read and the completion trigger operates. EP1-related interrupt requests to the CPU are not automatically masked. • Operating procedure: 1. Write of 1 to the EP1 DMAE bit in DMA 2. Set the DMAC to activate through $\overline{\text{DREQ0}}$ (USB INTN0) 3. Transfer count setting in the DMAC 4. DMAC activation 5. DMA transfer 6. DMA transfer end interrupt generated See section 16.8.3, DMA Transfer for Endpoints 1and 4.

16.3.28 Configuration Value Register (CVR)

This register stores the Configuration, Interface, or Alternate set value when the Set Configuration or Set Interface command from the host is correctly received.

Bit	Bit Name	Initial Value	R/W	Description
7 6	CNFV1 CNFV0	All 0	R	These bits store Configuration Setting value when they receive Set Configuration command. CNFV is updated when the SETC bit in IFR0 is set to 1.
5 4	INTV1 INTV0	All 0	R	These bits store Interface Setting value when they receive Set Interface command. INTV is updated when the SETI bit in IFR0 is set to 1.
3	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
2	ALTV2	0	R	These bits store Alternate Setting value when they receive Set Interface command. ALTV2 to ALTV0 are updated when the SETI bit in IFR0 is set to 1.
1	ALTV1	0	R	
0	ALTV0	0	R	

16.3.29 Control Register (CTLR)

This register sets functions for bits PRTRST, ASCE, PWMD, RSME, PWUPS, and PULLUP_E.

Bit	Bit Name	Initial Value	R/W	Description
7	PULLUPE	0	R/W	Pull-up Enable This bit controls whether to pull up the D+ pin. P20 is used as the pull-up control pin. 0: D+ is not pulled up. 1: D+ is pulled up.
6, 5	—	0	R/W	Reserved The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4	RWUPS	0	R	Remote Wakeup Status This status bit indicates remote wakeup command from USB host is enabled or disabled. This bit is set to 0 when remote wakeup command from UBM host is disabled by Device_Remote_Wakeup due to Set Feature or Clear Feature request. This bit is set to 1 when remote wakeup command is enabled.
3	RSME	0	R/W	Resume Enable This bit releases the suspend state (or executes remote wakeup). When RSME is set to 1, resume request starts. If RSME is once set to 1, clear this bit to 0 again afterwards. In this case, the value 1 set to RSME must be kept for at least one clock period of 12-MHz clock.
2	PWMD	0	R/W	Bus Power Mode This bit specifies the USB power mode. When PWMD is set to 0, the self-power mode is selected for this module. When set to 1, the bus-power mode is selected.
1	EP0 ASCE	0	R/W	EP0 Automatic Stall Clear Enable Setting the EP0 ASCE bit to 1 automatically clears the EP0 stall setting bit (the EP0 STLS bit in EPSTL0) after the stall handshake is returned to the host. When the EP0 ASCE bit is set to 0, the stall setting bit is not automatically cleared and must be cleared by the users. To enable the automatic stall clear function, make sure that the EP0 ASCE bit should be set to 1 before the EP0 STLS bit in EPSTL0 is set to 1.
0	PRTRST	1	R/W	Protocol Processing Block Reset 0: The protocol processing block is placed in operation state. 1: The protocol processing block is placed in reset state.

16.3.30 Endpoint Information Register (EPIR)

This register sets the information for each endpoint. Each endpoint needs five bytes to store the information. Writing data should be done in sequence starting at logical endpoint 0. Make sure to write data of 20 bytes (five bytes multiplied by four endpoints) to this register. The information should be written to this register only once at a power-on reset and no data should be written after that. Description of writing data for one endpoint is shown below.

Although this register consists of one register to which data is written sequentially for one address, the write data for the endpoint 0 is described as EPIR00 to EPIR04 (EPIR endpoint number in write order) to make the explanation understood easier. Write should start at EPIR00.

The endpoint numbers should not be overlapped except the one not in use.

- EPIR00

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	D7 to D4	Undefined	W	Endpoint Number [Enable setting range] 0 to 3
3, 2	D3, D2	Undefined	W	Endpoint Configuration Number [Enable setting range] 0 or 1
1, 0	D1, D0	Undefined	W	Endpoint Interface Number [Enable setting range] 0 to 3

- EPIR01

Bit	Bit Name	Initial Value	R/W	Description
7, 6	D7, D6	Undefined	W	Endpoint Alternate Number [Possible setting range] 0 or 1
5, 4	D5, D4	Undefined	W	Endpoint Transmission [Possible setting range] 0: Control 1: Setting prohibited 2: Bulk 3: Interrupt
3	D3	Undefined	W	Endpoint Transmission Direction [Possible setting range] 0: Out 1: In
2 to 0	D2 to D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

- EPIR02

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	D7 to D1	Undefined	W	Endpoint Maximum Packet Size [Possible setting range] 0 to 64
0	D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

- EPIR03

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

- EPIR04

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Endpoint FIFO Number [Possible setting range] 0 to 3

The endpoint number is the endpoint number the USB host uses. The endpoint FIFO number corresponds to the endpoint number described in this manual. Thus data transfer between the USB host and the endpoint FIFO can be enabled by putting the endpoint number and the endpoint FIFO number in one-to-one correspondence. Note that the setting value is subject to a limitation described below.

Since each endpoint FIFO number is optimized by the exclusive software that corresponds to the transfer system, direction, and the maximum packet size, make sure to set the endpoint FIFO number to the data described in table 16.2.

1. Endpoint number 0 and endpoint FIFO number 0 must have one-on-one relationship.
2. The maximum packet size for endpoint FIFO number 0 is limited to 16 bytes.
3. For endpoint FIFO number 0, only the maximum packet size can be specified and the data for the rest should be all 0.
4. The maximum packet size for endpoint FIFO numbers 1 and 2 is limited to 64 bytes.
5. Only the bulk transfer method and out transfer direction can be specified for endpoint FIFO numbers 1.
6. Only the bulk transfer method and in transfer direction can be specified for endpoint FIFO numbers 2.
7. The maximum packet size for endpoint FIFO numbers 3 is limited to 16 bytes.
8. Only the interrupt transfer method and in transfer direction can be specified for endpoint FIFO numbers 3.
9. The maximum number of endpoint information settings is four.
10. Four endpoint information settings should be made.
11. Write 0 to the endpoints not in use.

Table 16.2 shows the limitations for the maximum packet size, the transfer method, and the transfer direction.

Table 16.2 Limitations for Setting Values

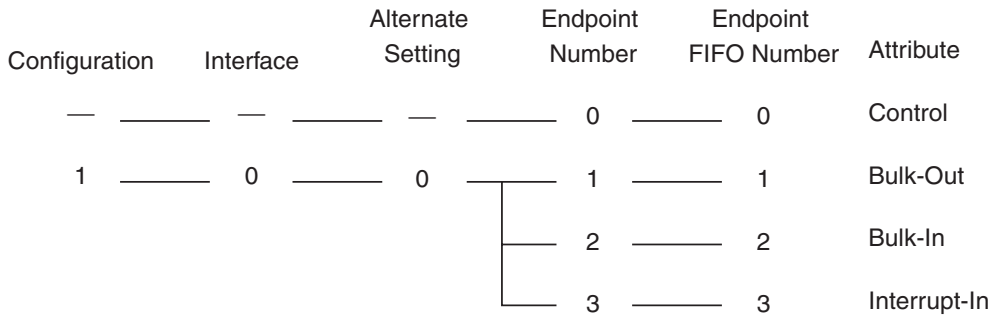
Endpoint FIFO Number	Maximum Packet Size	Transfer Method	Transfer Direction
0	16 bytes	Control	In/Out
1	64 bytes	Bulk	Out
2	64 bytes	Bulk	In
3	16 bytes	Interrupt	In

Table 16.3 shows a specific example of setting.

Table 16.3 Example of Setting

Endpoint Number	Conf.	Int.	Alt.	Transfer Method	Transfer Direction	Maximum Packet Size	Endpoint FIFO Number
0	—	—	—	Control	In/Out	16 bytes	0
1	1	0	0	Bulk	Out	64 bytes	1
2	1	0	0	Bulk	In	64 bytes	2
3	1	0	0	Interrupt	In	16 bytes	3

N	EPIR[N]0	EPIR[N]1	EPIR[N]2	EPIR[N]3	EPIR[N]4
0	00	00	20	00	00
1	14	20	80	00	01
2	24	28	80	00	02
3	34	38	20	00	03



16.3.31 Transceiver Test Register 0 (TRNTREG0)

TRNTREG0 controls the on-chip transceiver output signals. Setting the PTSTE bit to 1 specifies the transceiver output signals (USD+ and USD-) arbitrarily. Table 16.4 shows the relationship between TRNTREG0 setting and pin output.

Bit	Bit Name	Initial Value	R/W	Description
7	PTSTE	0	R/W	Pin Test Enable Enables the test control for the on-chip transceiver output pins (USD+ and USD-).
6 to 4	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
3	SUSPEND	0	R/W	On-Chip Transceiver Output Signal Setting
2	txenl	0	R/W	SUSPEND: Sets the (SUSPEND) signal of the on-chip transceiver.
1	txse0	0	R/W	
0	txdata	0	R/W	txenl: Sets the output enable (txenl) signal of the on-chip transceiver. txse0: Sets the Signal-ended 0 (txse0) signal of the on-chip transceiver. txdata: Sets the (txdata) signal of the on-chip transceiver.

Table 16.4 Relationship between TRNTREG0 Setting and Pin Output

Register Setting				Pin Output	
PTSTE	txenl	txse0	txdata	USD+	USD-
0	X	X	X	—	—
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	X	0	0
1	1	X	X	Hi-Z	Hi-Z

[Legend]

X: Don't care.

—: Cannot be controlled. Indicates state in normal operation according to the USB operation and port settings.

16.3.32 Transceiver Test Register 1 (TRNTREG1)

TRNTREG1 is a test register that can monitor the on-chip transceiver input signal.

Setting bits PTSTE and txenl in TRNTREG0 to 1 enables monitoring the on-chip transceiver input signal. Table 16.5 shows the relationship between pin input and TRNTREG1 monitoring value.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
2	xver_data	0	R	On-Chip Transceiver Input Signal Monitor
1	dpls	0	R	xver_data: Monitors the differential input level (xver_data) signal of the on-chip transceiver.
0	dmns	0	R	dpls: Monitors the USD+ (dpls) signal of the on-chip transceiver. dmns: Monitors the USD- (dmns) signal of the on-chip transceiver.

Table 16.5 Relationship between Pin Input and TRNTREG1 Monitoring Value

Register Setting		Pin Input		TRNTREG1 Monitoring Value			
PTSTE	**SUSPEND**	**USD+**	**USD-**	**xver_data**	**dpls**	**dmns**	**Remarks**
0	X	X	X	0	0	0	Cannot be monitored when VBUS = 0 or PTSTE = 0 (initial value)
1	0	0	0	X	0	0	Can be monitored when VBUS = 1 and PTSTE = 1
1	0	0	1	0	0	1	
1	0	1	0	1	1	0	
1	0	1	1	X	1	1	
1	1	0	0	0	0	0	
1	1	0	1	0	0	1	
1	1	1	0	0	1	0	
1	1	1	1	0	1	1	

[Legend]
X: Don't care.

16.4 Interrupt Sources

This module has five interrupt signals. Table 16.6 shows the interrupt sources and their corresponding interrupt request signals. The USBINTN interrupt signals are activated at low level. The USBINTN interrupt requests can only be detected at low level (specified as level sensitive).

Table 16.6 Interrupt Sources

Register	Bit	Transfer Mode	Interrupt Source	Description	Interrupt Request Signal	DMAC Activation
IFR0	0	Status	VBUSF	USB bus connection/ disconnection detection	USBINTN2 or USBINTN3	×
	1		VBUSMN	VBUS connection status	—	×
	2		SETI	Set_Interface command detection	USBINTN2 or USBINTN3	×
	3		SETC	Set_Configuration command detection	USBINTN2 or USBINTN3	×
	4		SURSF	Suspend/resume detection	USBINTN2, USBINTN3 or RESUME	×
	5		SURSS	Suspend/resume status	—	×
	6		CFDN	Endpoint information load end	USBINTN2 or USBINTN3	×
	7		BRST	Bus reset	USBINTN2 or USBINTN3	×
IFR1	0	Control transfer (EP0)	EP0i_TS*	EP0i transmission complete	USBINTN2 or USBINTN3	×
	1		EP0i_TR*	EP0i transfer request	USBINTN2 or USBINTN3	×
	2		EP0o_TS*	EP0o receive complete	USBINTN2 or USBINTN3	×
	3		SETUP_TS*	Setup command receive complete	USBINTN2 or USBINTN3	×
	4	Status	SOF	SOF packet detection	USBINTN2 or USBINTN3	×
	5	—	Reserved	—	—	—
	6	—	Reserved	—	—	—
	7	—	Reserved	—	—	—

Register	Bit	Transfer Mode	Interrupt Source	Description	Interrupt Request Signal	DMAC Activation
IFR2	0	Bulk_out transfer (EP1)	EP1_FULL	EP1 FIFO full	USBINTN2 or USBINTN3	USBINTN0 ($\overline{\text{DREQ0}}$)
	1	Bulk_in transfer (EP2)	EP2_ALLEMP	EP2 FIFO all empty	USBINTN2 or USBINTN3	×
	2		EP2_EMPTY	EP2 FIFO empty	USBINTN2 or USBINTN3	USBINTN1 ($\overline{\text{DREQ1}}$)
	3		EP2_TR	EP2 transfer request	USBINTN2 or USBINTN3	×
	4	Interrupt_in transfer (EP3)	EP3_TS	EP3 transmission complete	USBINTN2 or USBINTN3	×
	5		EP3_TR	EP3 transfer request	USBINTN2 or USBINTN3	×
	6	—	Reserved	—	—	—
	7	—	Reserved	—	—	—

Note: * EP0 interrupts must be assigned to the same interrupt request signal.

- USB INTN0 signal

  DMA transfer request signal only for EP1. See section 16.8, DMA Transfer.
- USB INTN1 signal

  DMA transfer request signal only for EP2. See section 16.8, DMA Transfer.
- USB INTN2 signal

  The USB INTN2 signal requests interrupt sources for which the corresponding bits in interrupt select registers 0 and 2 (ISR0 and ISR2) are cleared to 0. The USB INTN2 is driven low if a corresponding bit in the interrupt flag register is set to 1.
- USB INTN3 signal

  The USBINTN3 signal requests interrupt sources for which the corresponding bits in interrupt select registers 0 to 2 (ISR0 and ISR2) are cleared to 0. The USB INTN3 is driven low if a corresponding bit in the interrupt flag register is set to 1.
- RESUME signal

  The RESUME signal is a resume interrupt signal for canceling software standby mode. The RESUME signal is driven low at the transition to the resume state for canceling software standby mode.

16.5 Operation

16.5.1 Initial Settings

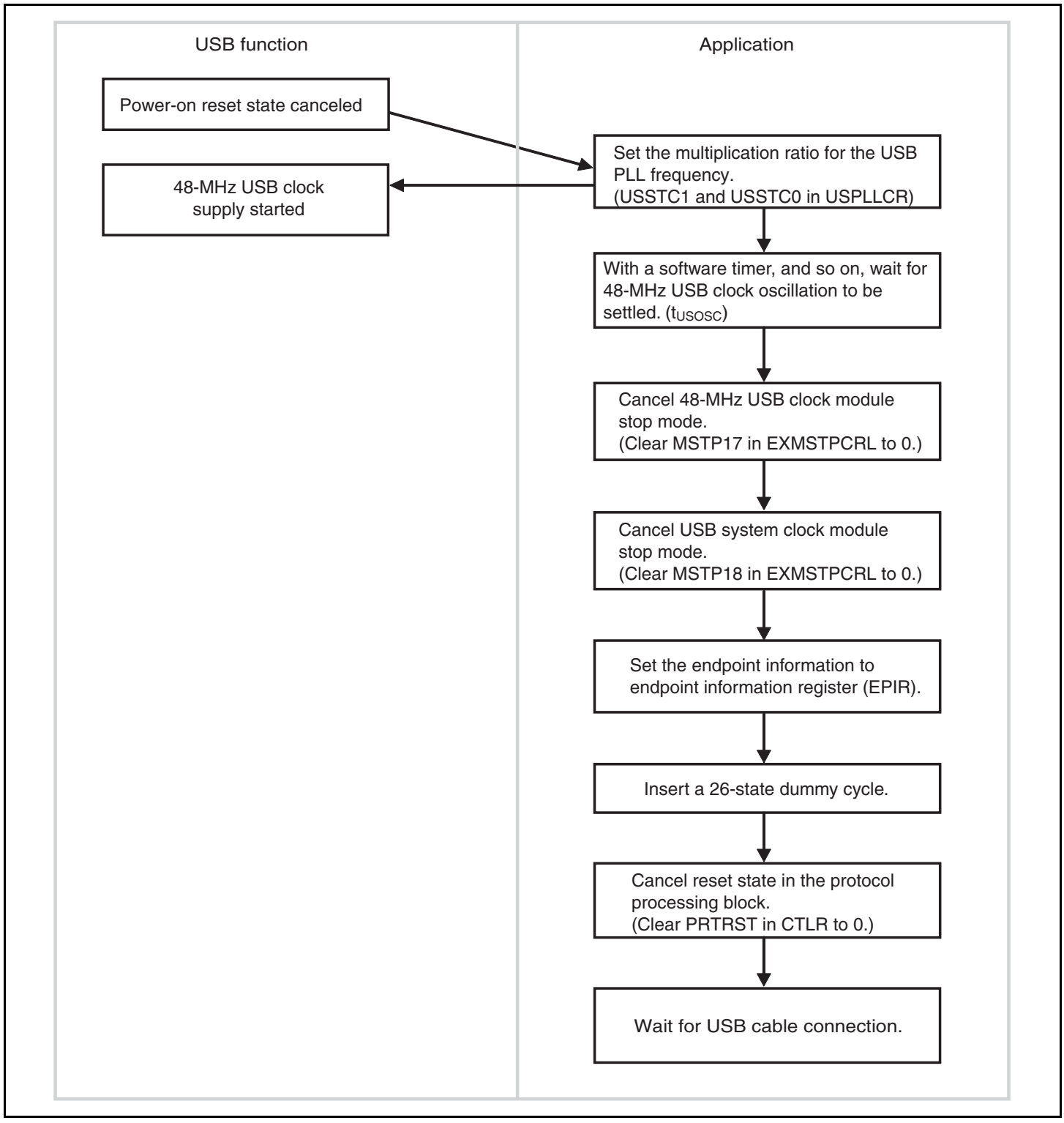


Figure 16.2 Initial Setting Operation

16.5.2 Cable Connection

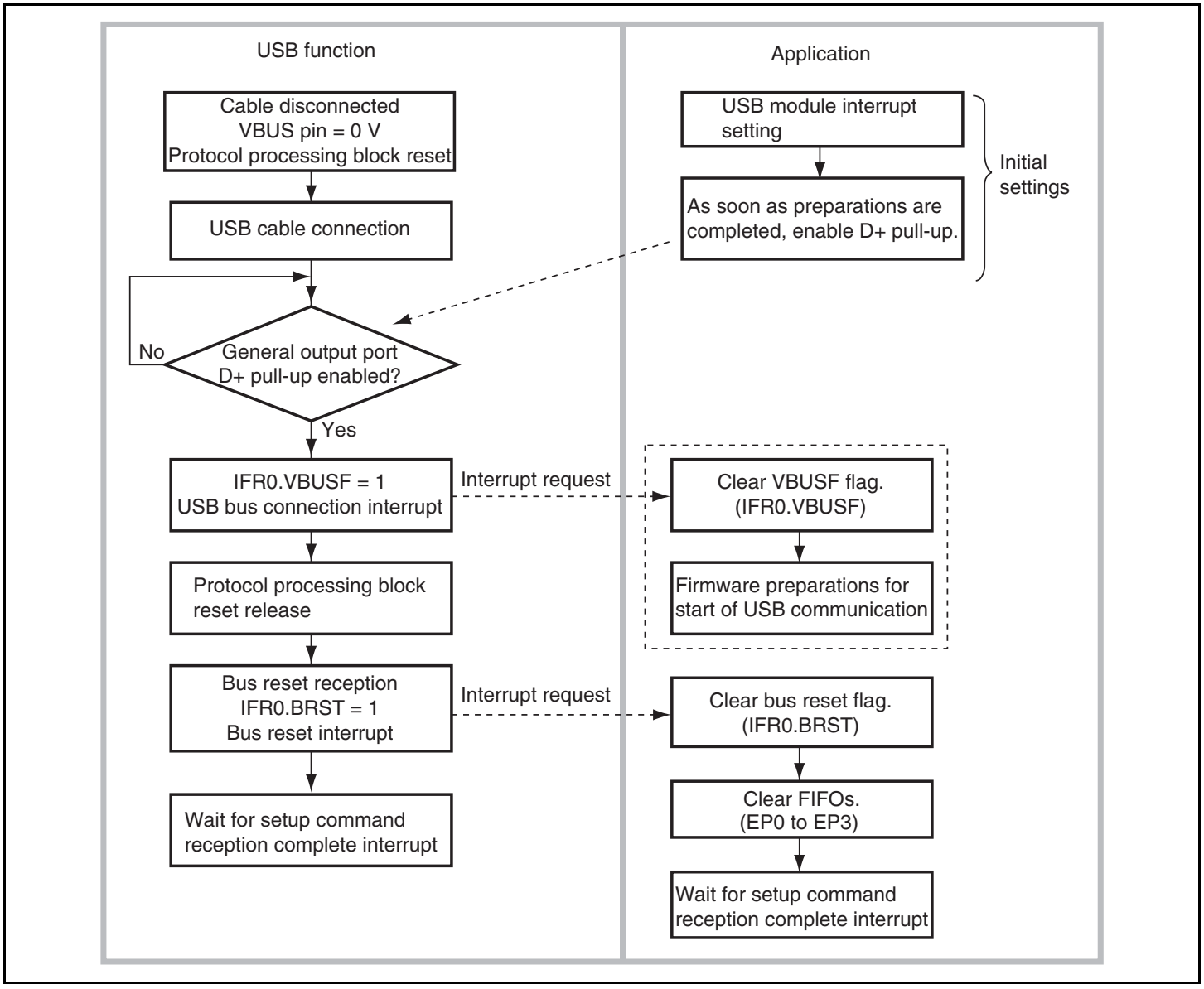


Figure 16.3 Cable Connection Operation

The above flowchart shows the operation in the case of in section 16.9, Example of USB External Circuitry.

In applications that do not require USB cable connection to be detected, processing by the USB bus connection interrupt is not necessary. Preparations should be made with the bus-reset interrupt.

16.5.3 Cable Disconnection

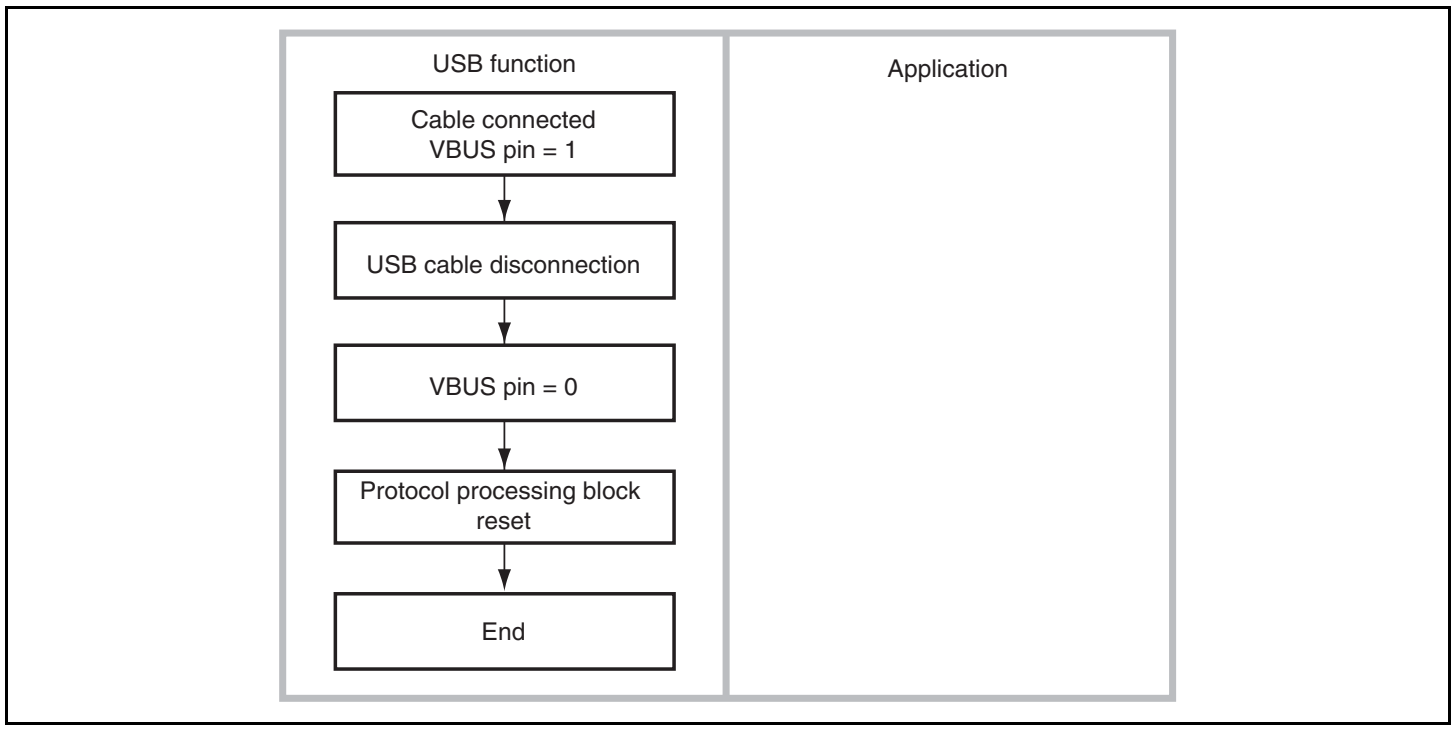


Figure 16.4 Cable Disconnection Operation

The above flowchart shows the operation in section 16.9, Example of USB External Circuitry.

16.5.4 Suspend and Resume Operations

(1) Suspend Operation

If the USB bus enters the suspend state from the non-suspend state, perform the operation as shown in figure 16.5.

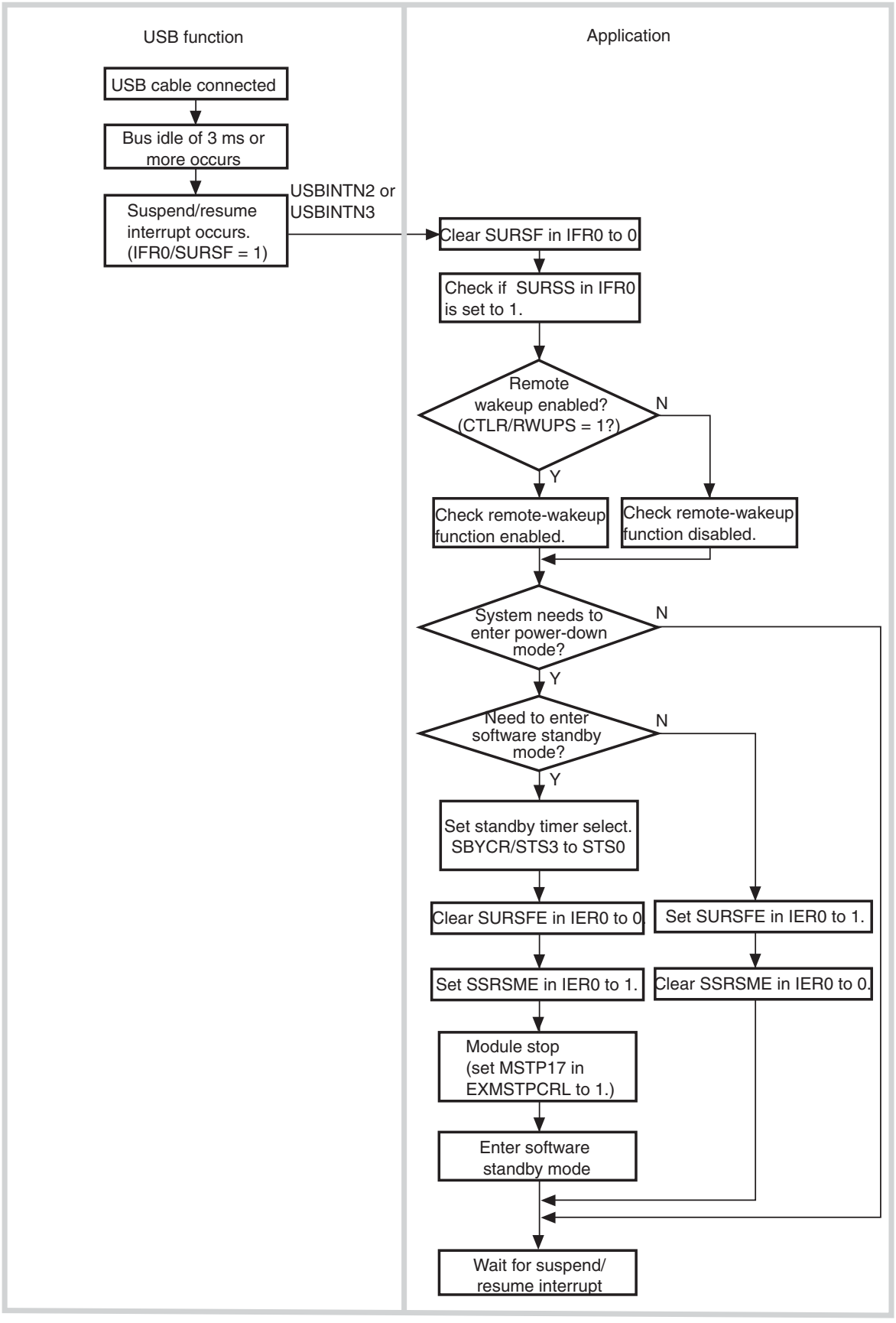


Figure 16.5 Suspend Operation

(2) Resume Operation from Up-Stream

If the USB bus enters the non-suspend state from the suspend state by resume signal output from up-stream, perform the operation as shown in figure 16.6.

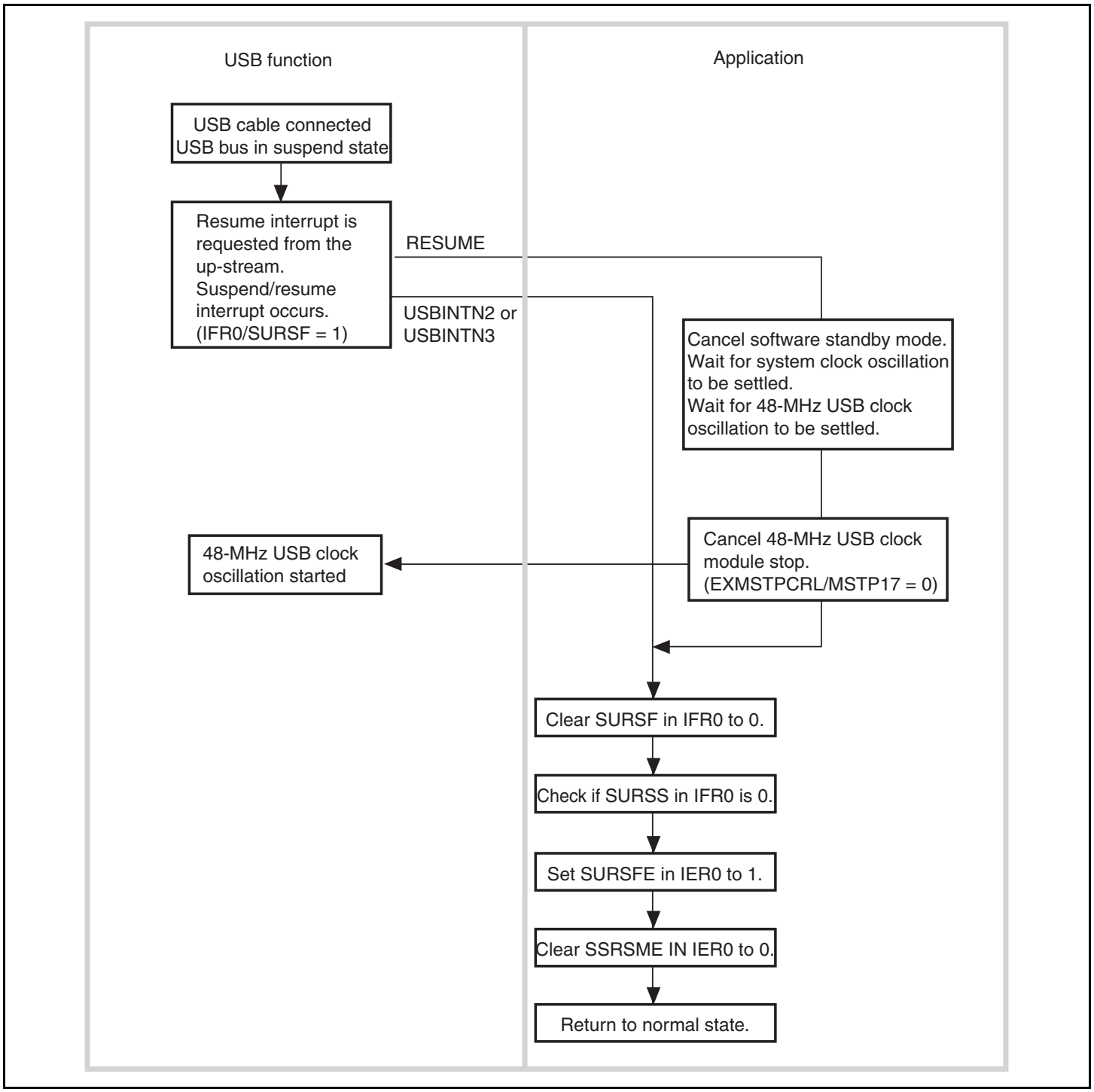


Figure 16.6 Resume Operation from Up-Stream

(3) Transition from Suspend State to Software Standby Mode and Canceling Software Standby Mode

If the USB bus enters from the suspend state to software standby mode, perform the operation as shown in figure 16.7. When canceling software standby mode, ensure enough time for the system clock oscillation to be settled.

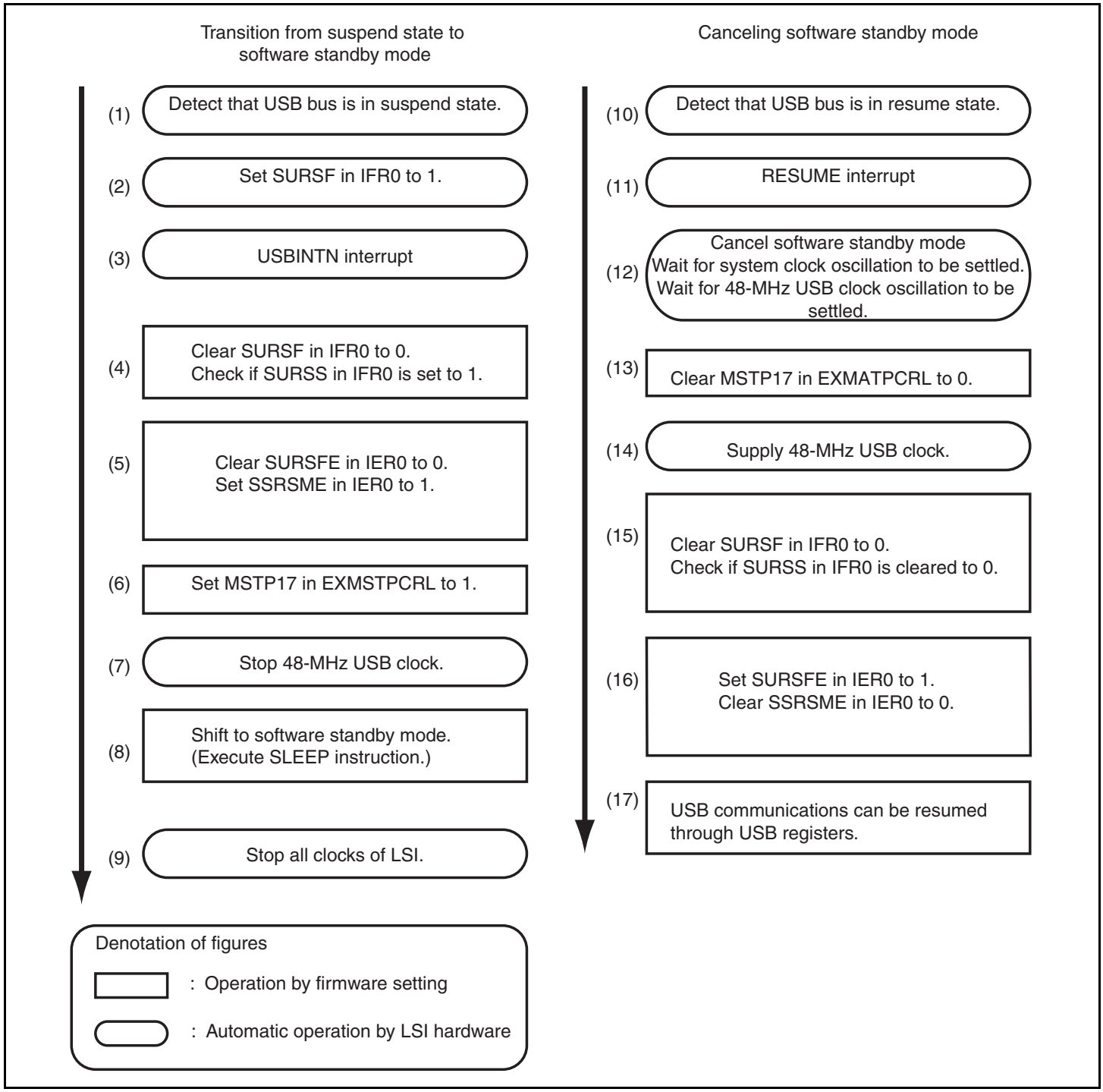


Figure 16.7 Flow of Transition to and Canceling Software Standby Mode

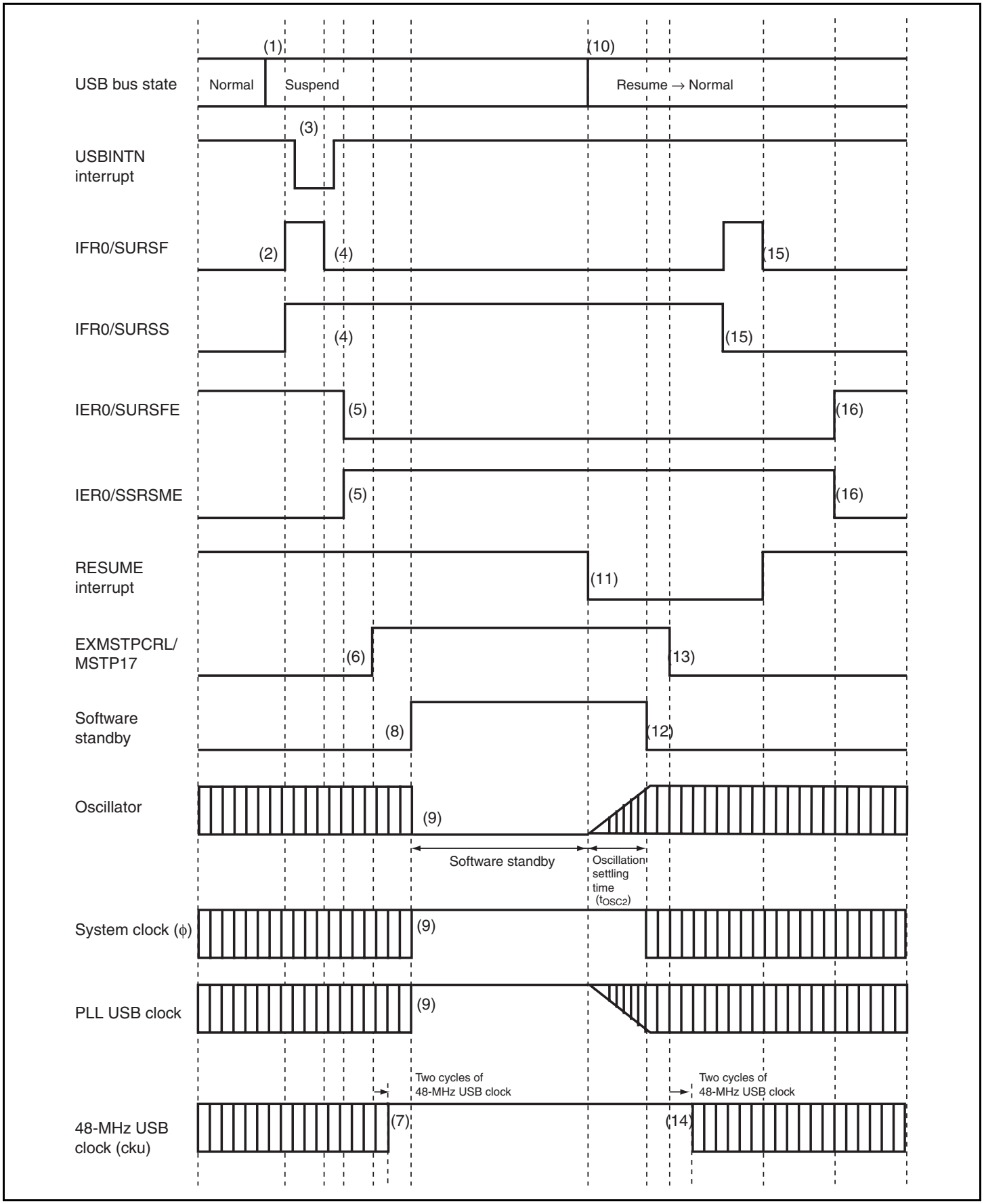


Figure 16.8 Timing of Transition to and Canceling Software Standby Mode

(4) Remote-Wakeup Operation

If the USB bus enters the non-suspend (resume) state from the suspend state by the remote-wakeup signal output from this function, perform the operation as shown in figure 16.9.

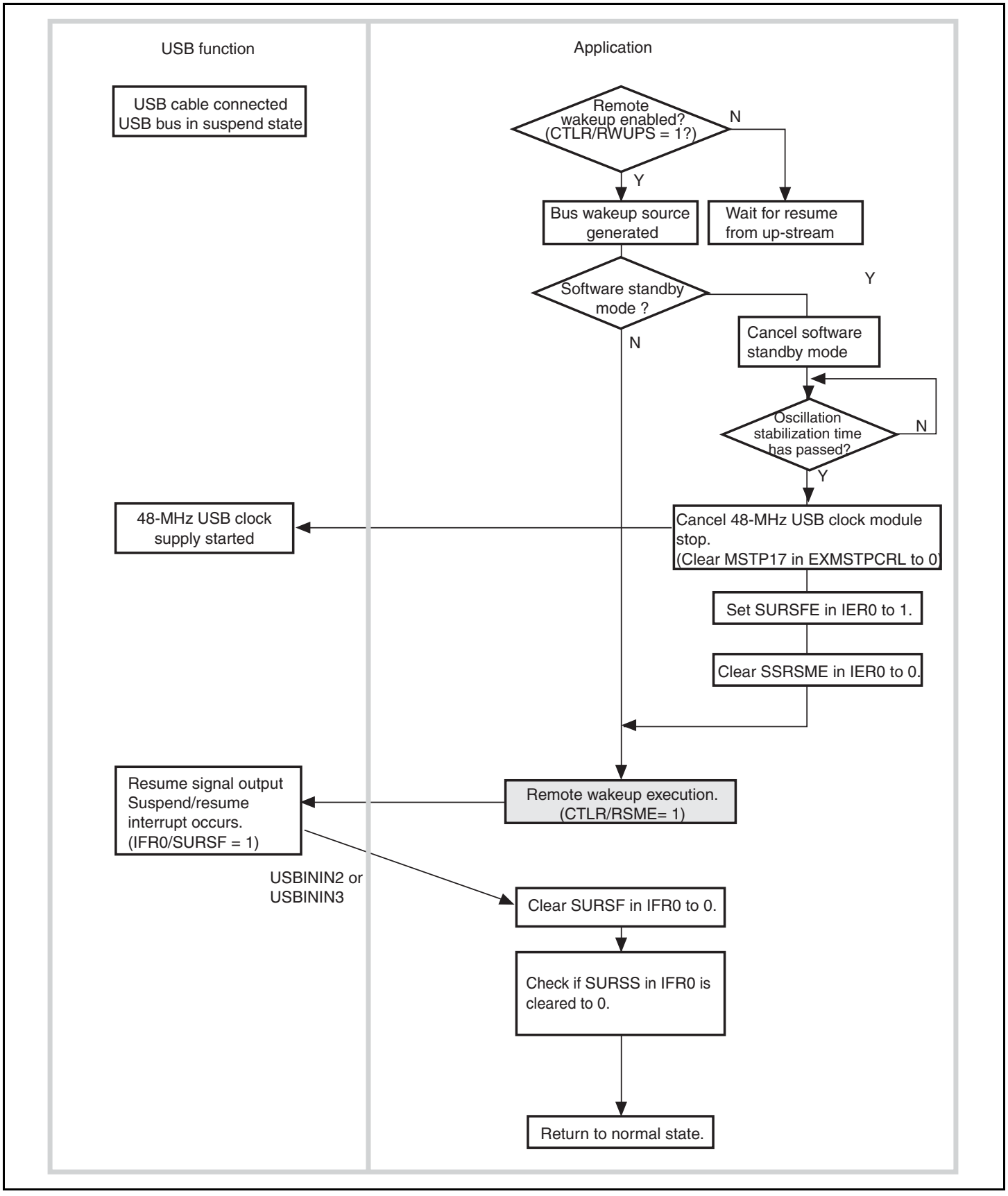


Figure 16.9 Remote-Wakeup

16.5.5 Control Transfer

Control transfer consists of three stages: setup, data (not always included), and status (figure 16.10). The data stage comprises a number of bus transactions. Operation flowcharts for each stage are shown below.

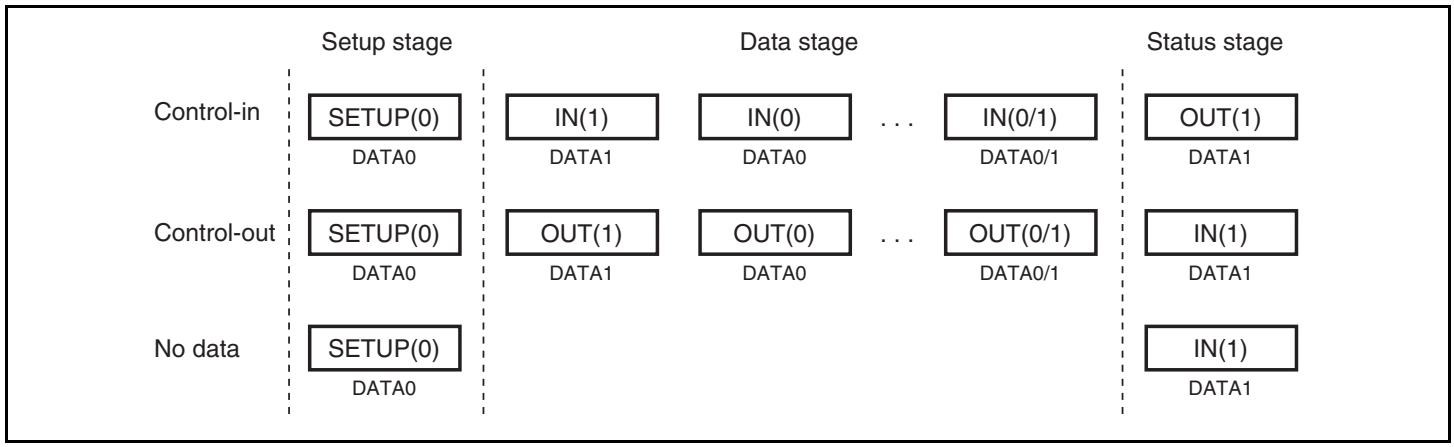
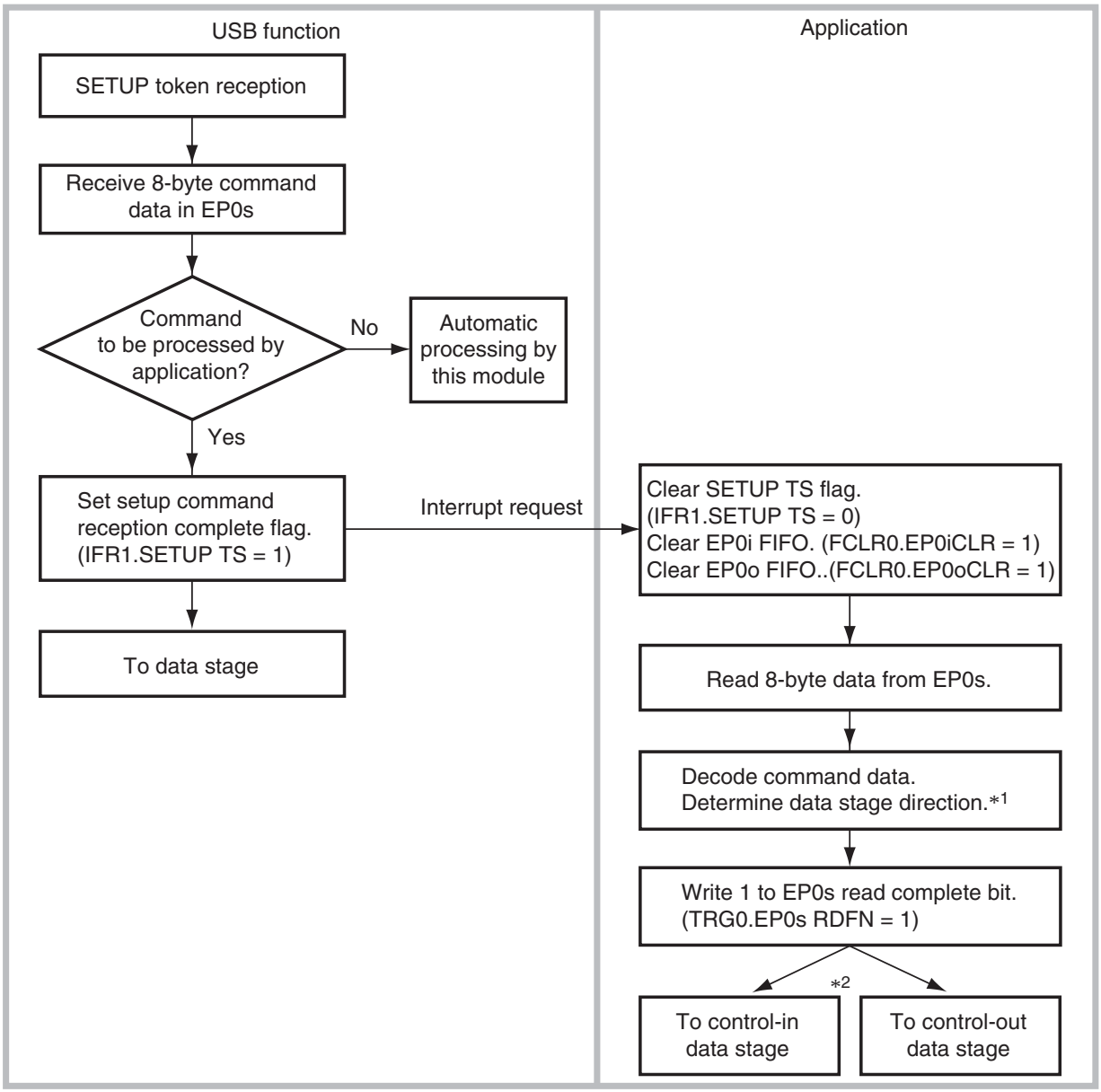


Figure 16.10 Transfer Stages in Control Transfer

(1) Setup Stage



Notes: 1. In the setup stage, the application analyzes command data from the host requiring processing by the application, and determines the subsequent processing (for example, data stage direction, etc.).
2. When the transfer direction is control-out, the EP0i transfer request interrupt required in the status stage should be enabled here. When the transfer direction is control-in, this interrupt is not required and should be disabled.

Figure 16.11 Setup Stage Operation

(2) Data Stage (Control-In)

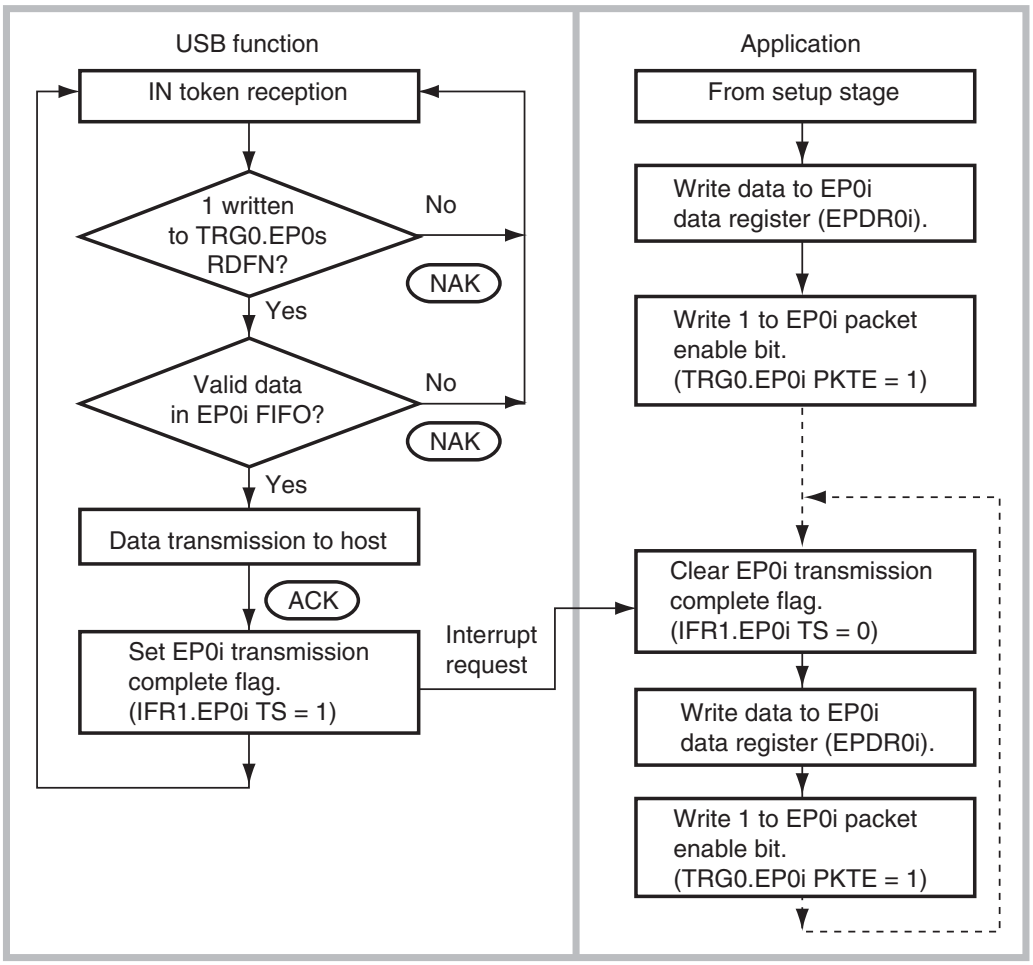


Figure 16.12 Data Stage (Control-In) Operation

The application first analyzes command data from the host in the setup stage, and determines the subsequent data stage direction. If the result of command data analysis is that the data stage is in-transfer, one packet of data to be sent to the host is written to the FIFO. If there is more data to be sent, this data is written to the FIFO after the data written first has been sent to the host (EP0i TS bit in IFR1 = 1).

The end of the data stage is identified when the host transmits an OUT token and the status stage is entered.

Note: If the size of the data transmitted by the function is smaller than the data size requested by the host, the function indicates the end of the data stage by returning to the host a packet shorter than the maximum packet size. If the size of the data transmitted by the function is an integral multiple of the maximum packet size, the function indicates the end of the data stage by transmitting a zero-length packet.

(3) Data Stage (Control-Out)

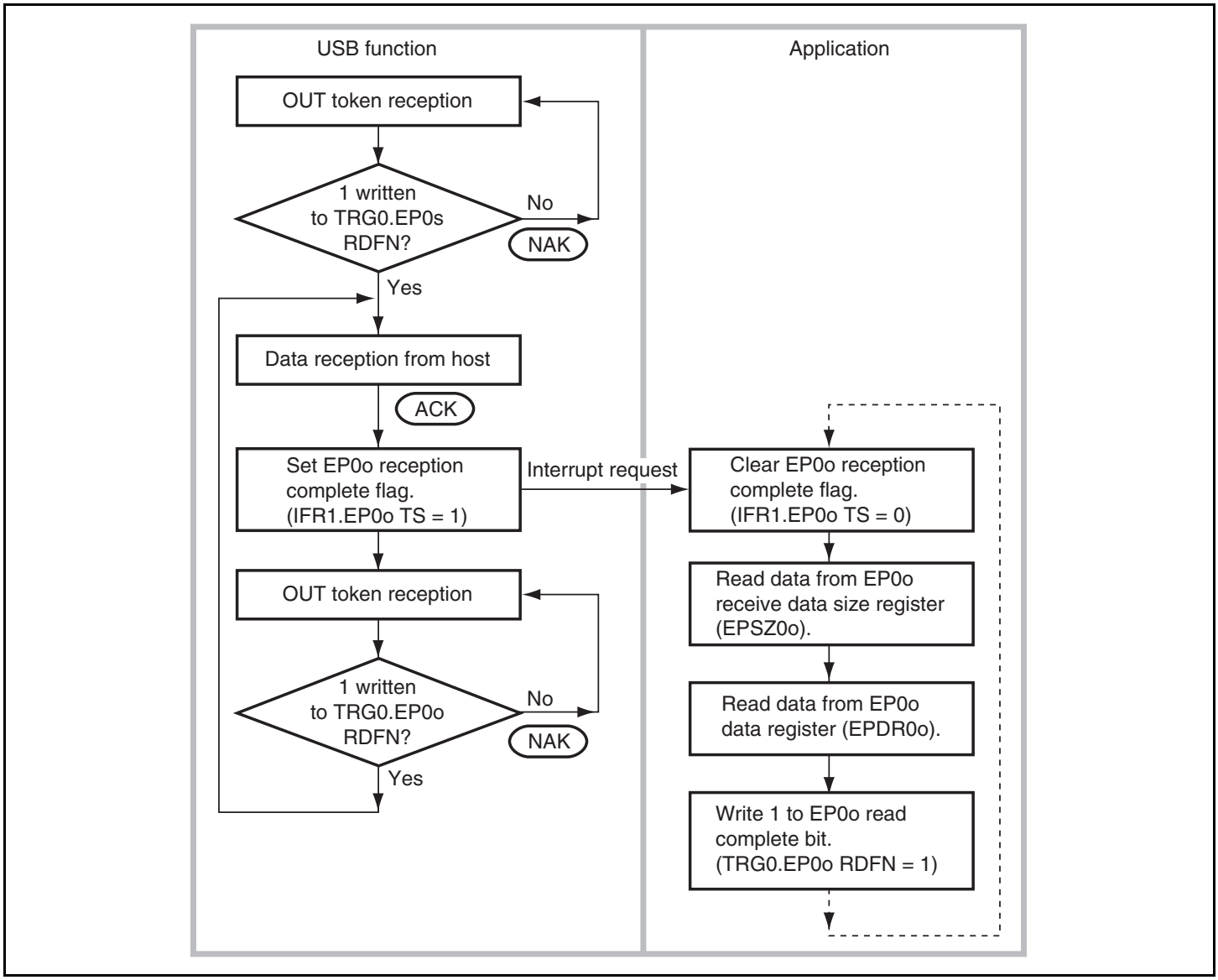


Figure 16.13 Data Stage (Control-Out) Operation

The application first analyzes command data from the host in the setup stage, and determines the subsequent data stage direction. If the result of command data analysis is that the data stage is out-transfer, the application waits for data from the host, and after data is received (EP0o TS bit in IFR1 = 1), reads data from the FIFO. Next, the application writes 1 to the EP0o read complete bit, empties the receive FIFO, and waits for reception of the next data.

The end of the data stage is identified when the host transmits an IN token and the status stage is entered.

(4) Status Stage (Control-In)

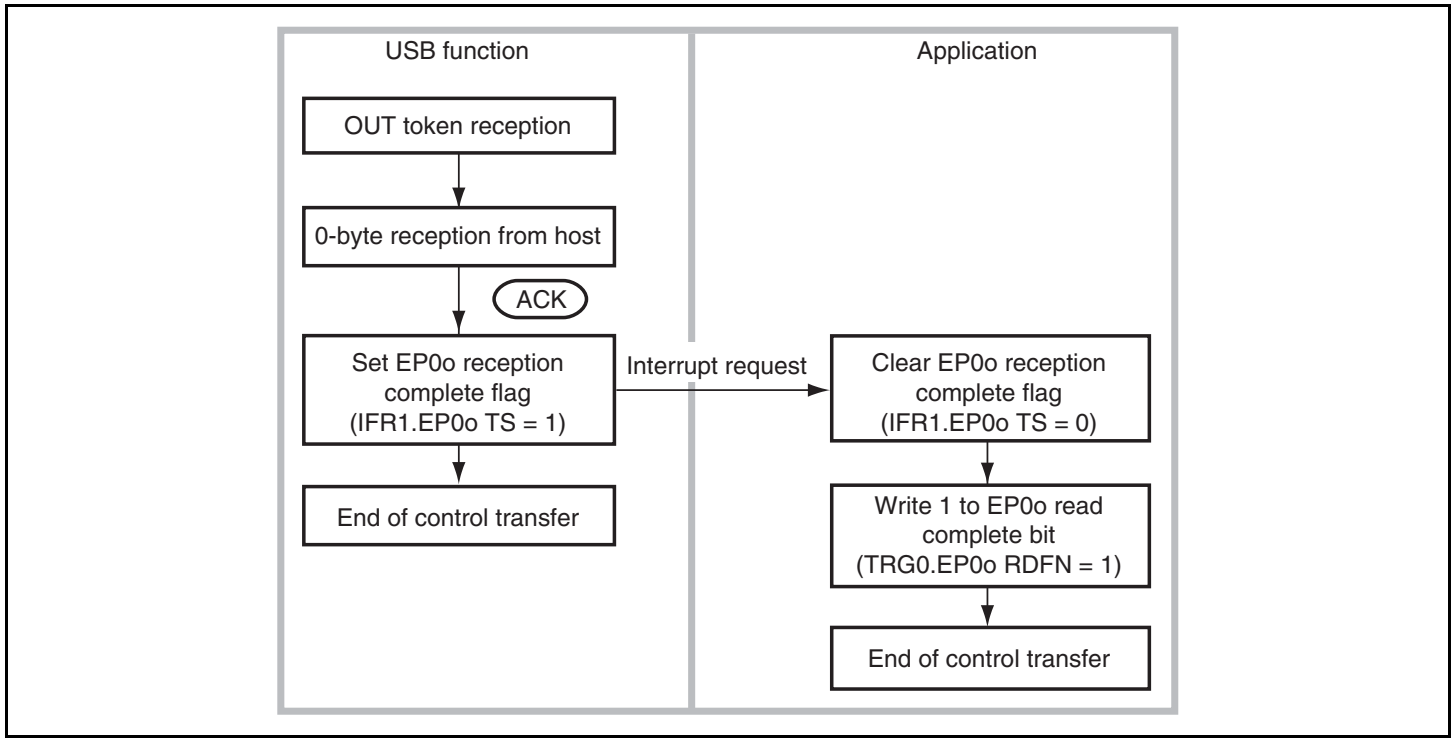


Figure 16.14 Status Stage (Control-In) Operation

The control-in status stage starts with an OUT token from the host. The application receives 0-byte data from the host, and ends control transfer.

(5) Status Stage (Control-Out)

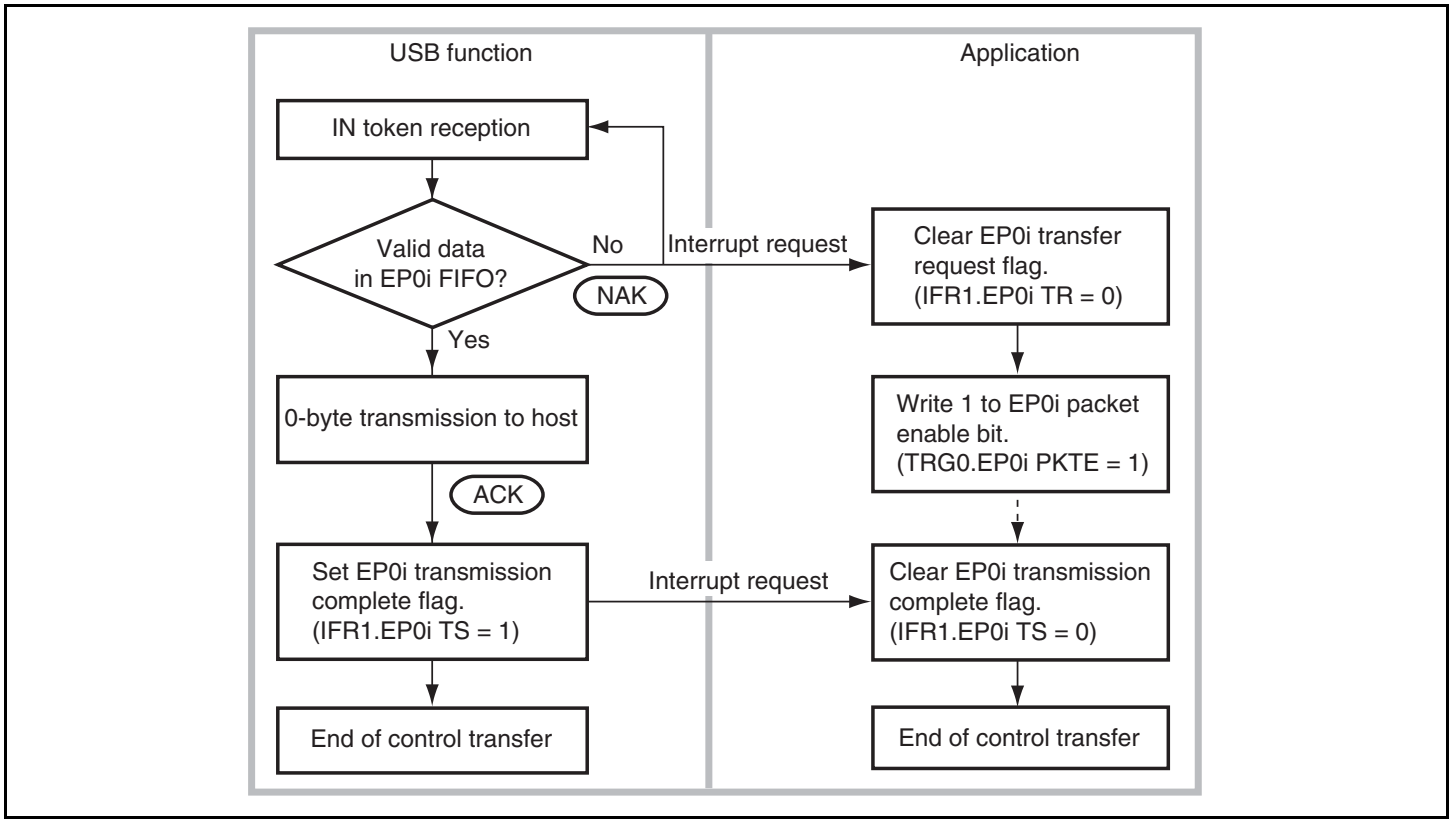


Figure 16.15 Status Stage (Control-Out) Operation

The control-out status stage starts with an IN token from the host. When an IN-token is received at the start of the status stage, there is not yet any data in the EP0i FIFO, and so an EP0i transfer request interrupt is generated. The application recognizes from this interrupt that the status stage has started. Next, in order to transmit 0-byte data to the host, 1 is written to the EP0i packet enable bit but no data is written to the EP0i FIFO. As a result, the next IN token causes 0-byte data to be transmitted to the host, and control transfer ends.

After the application has finished all processing relating to the data stage, 1 should be written to the EP0i packet enable bit.

16.5.6 EP1 Bulk-Out Transfer

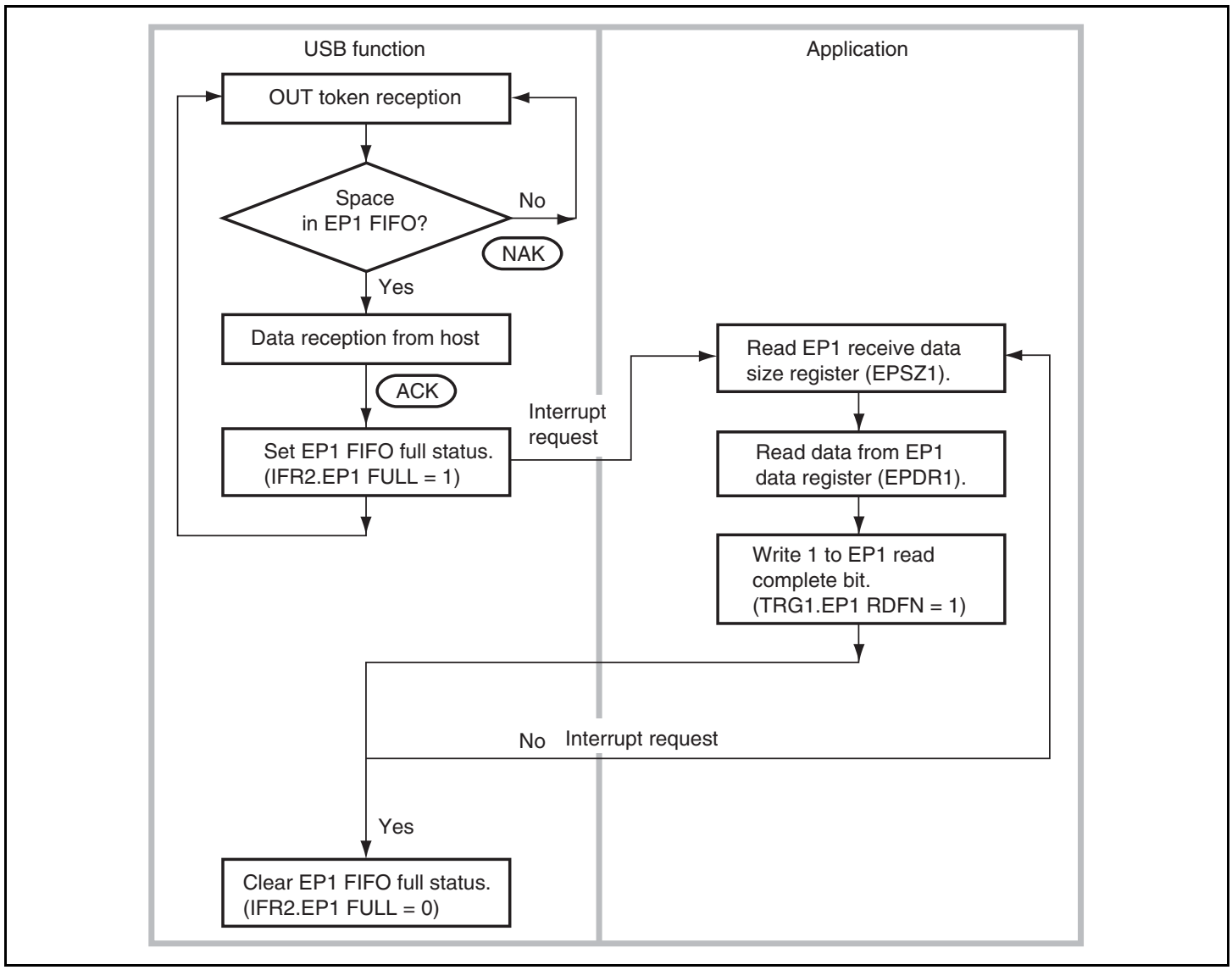


Figure 16.16 EP1 Bulk-Out Transfer Operation

- Dual FIFOs (EP1)

EP1 has two 64-byte FIFOs, but the user can receive data and read receive data without being aware of this dual-FIFO configuration.

When one FIFO is full after reception is completed, the EP1 FULL bit in IFR2 is set. After the first receive operation into one of the FIFOs when both FIFOs are empty, the other FIFO is empty, and so the next packet can be received immediately. When both FIFOs are full, NAK is returned to the host automatically. When reading of the receive data is completed following data reception, 1 is written to the EP1 RDFN bit in TRG1. This operation empties the FIFO that has just been read, and makes it ready to receive the next packet.

16.5.7 EP2 Bulk-In Transfer

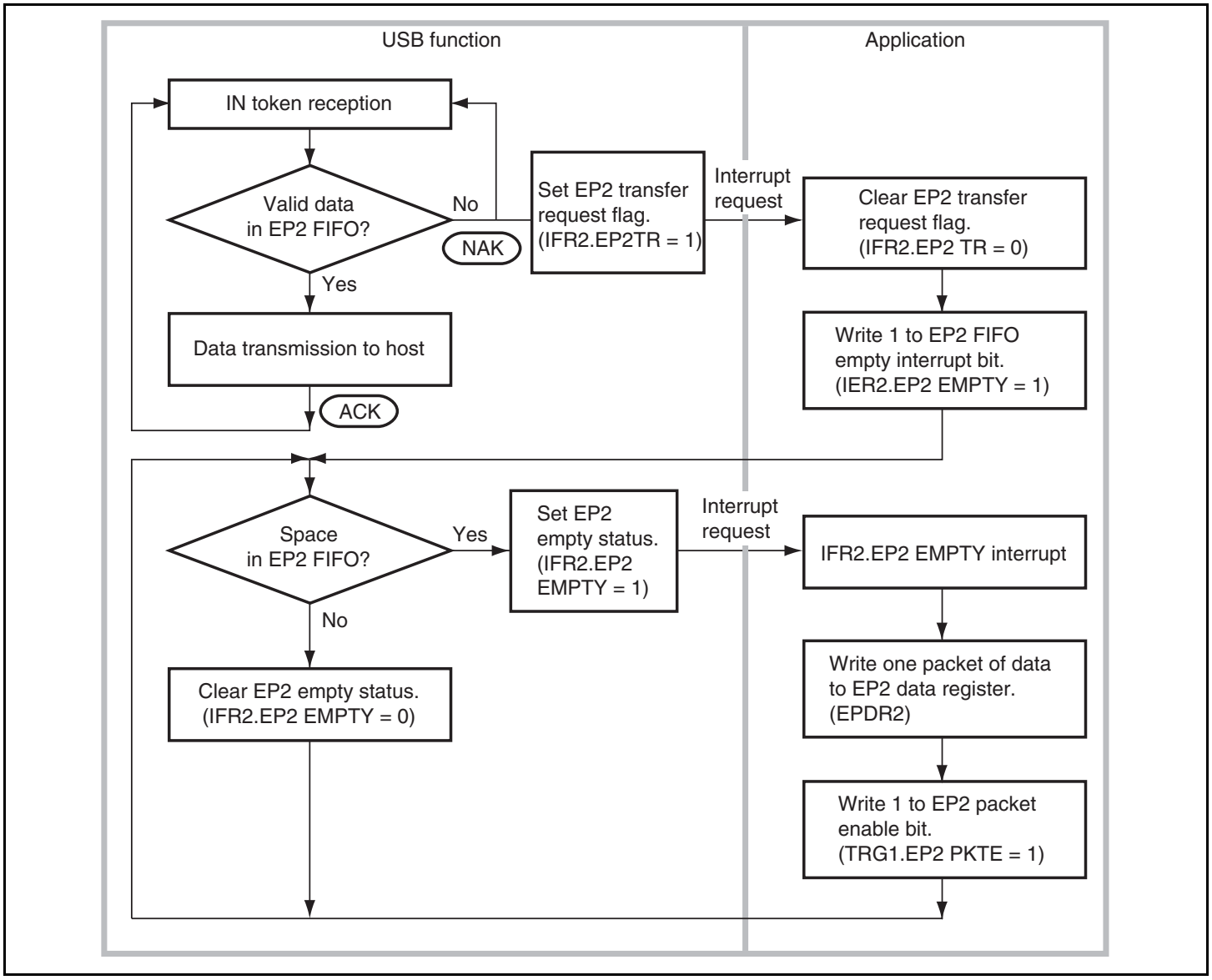


Figure 16.17 EP2 Bulk-In Transfer Operation

(1) Dual FIFOs (EP2)

EP2 has two 64-byte FIFOs, but the user can transmit data and write transmit data without being aware of this dual-FIFO configuration. However, one data write is performed for one FIFO. For example, even if both FIFOs are empty, it is not possible to perform EP2 PKTE at one time after consecutively writing 128 bytes of data. EP2 PKTE must be performed for each 64-byte write.

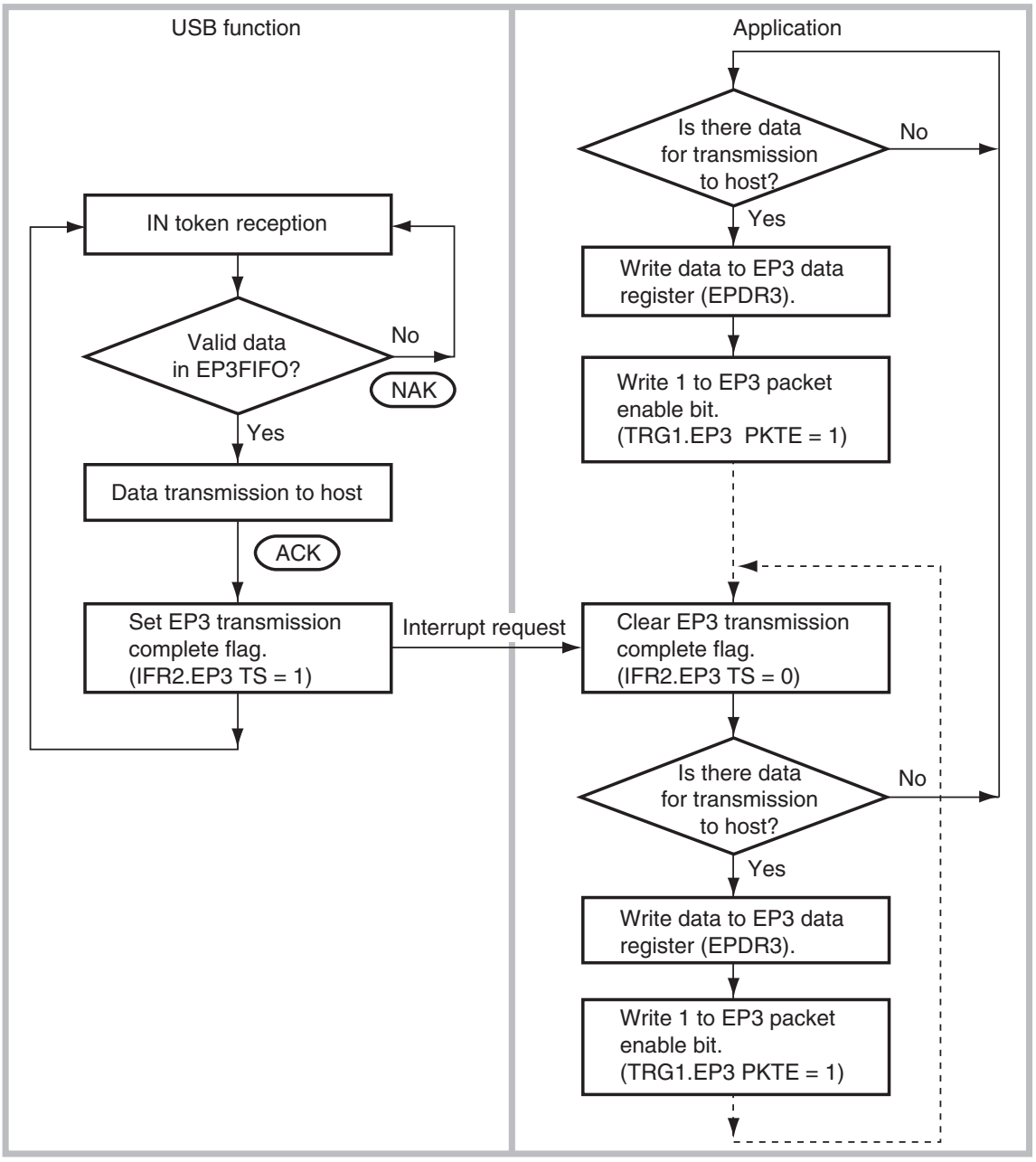
When performing bulk-in transfer, as there is no valid data in the FIFOs on reception of the first IN token, an EP2 TR bit interrupts in IFR2 is requested. With this interrupt, 1 is written to the EP2

EMPTYE bit in IER2, and the EP2 FIFO empty interrupt is enabled. At first, both EP2 FIFOs are empty, and so an EP2 FIFO empty interrupt is generated immediately.

The data to be transmitted is written to the data register using this interrupt. After the first transmit data write for one FIFO, the other FIFO is empty, and so the next transmit data can be written to the other FIFO immediately. When both FIFOs are full, EP2 EMPTYE is cleared to 0. If at least one FIFO is empty, the EP2 EMPTY bit in IFR2 is set to 1. When ACK is returned from the host after data transmission is completed, the FIFO used in the data transmission becomes empty. If the other FIFO contains valid transmit data at this time, transmission can be continued.

When transmission of all data has been completed, write 0 to the EP2 EMPTYE bit in IER2 and disable interrupt requests.

16.5.8 EP3 Interrupt-In Transfer



Note: This flowchart shows just one example of interrupt transfer processing. Other possibilities include an operation flow in which, if there is data to be transferred, the EP3 DE bit in the data status register is referenced to confirm that the FIFO is empty, and then data is written to the FIFO.

Figure 16.18 Operation of EP3 Interrupt-In Transfer

16.6 Processing of USB Standard Commands and Class/Vendor Commands

16.6.1 Processing of Commands Transmitted by Control Transfer

A command transmitted from the host by control transfer may require decoding and execution of command processing on the application side. Whether command decoding is required on the application side is indicated in table 16.7 below.

Table 16.7 Command Decoding on Application Side

Decoding not Necessary on Application Side	Decoding Necessary on Application Side
Clear Feature	Get Descriptor
Get Configuration	Class/Vendor command
Get Interface	Set Descriptor
Get Status	Sync Frame
Set Address	
Set Configuration	
Set Feature	
Set Interface	

If decoding is not necessary on the application side, command decoding and data stage and status stage processing are performed automatically. No processing is necessary by the user. An interrupt is not generated in this case.

If decoding is necessary on the application side, this module stores the command in the EP0s FIFO. After reception is completed successfully, the IFR1/SETUP TS flag is set and an interrupt request is generated. In the interrupt routine, eight bytes of data must be read from the EP0s data register (EPDR0s) and decoded by firmware. The necessary data stage and status stage processing should then be carried out according to the result of the decoding operation.

16.7 Stall Operations

16.7.1 Overview

This section describes stall operations in this module. There are two cases in which the USB function module stall function is used:

- When the application forcibly stalls an endpoint for some reason
- When a stall is performed automatically within the USB function module due to a USB specification violation

The USB function module has internal status bits that hold the status (stall or non-stall) of each endpoint. When a transaction is sent from the host, the module references these internal status bits and determines whether to return a stall to the host. These bits cannot be cleared by the application; they must be cleared with a Clear Feature command from the host.

However, the internal status bit for EP0 is automatically cleared only when the setup command is received.

16.7.2 Forcible Stall by Application

The application uses the EPSTL register to issue a stall request for the USB function module. When the application wishes to stall a specific endpoint, it sets the corresponding bit in EPSTL (1-1 in figure 16.19). The internal status bits are not changed at this time. When a transaction is sent from the host for the endpoint for which the EPSTL bit was set, the USB function module references the internal status bit, and if this is not set, references the corresponding bit in EPSTL (1-2 in figure 16.19). If the corresponding bit in EPSTL is set, the USB function module sets the internal status bit and returns a stall handshake to the host (1-3 in figure 16.19). If the corresponding bit in EPSTL is not set, the internal status bit is not changed and the transaction is accepted.

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to the EPSTL register. Even after a bit is cleared by the Clear Feature command (3-1 in figure 16.19), the USB function module continues to return a stall handshake while the bit in EPSTL is set, since the internal status bit is set each time a transaction is executed for the corresponding endpoint (1-2 in figure 16.19). To clear a stall, therefore, it is necessary for the corresponding bit in EPSTL to be cleared by the application, and also for the internal status bit to be cleared with a Clear Feature command (2-1, 2-2, and 2-3 in figure 16.19).

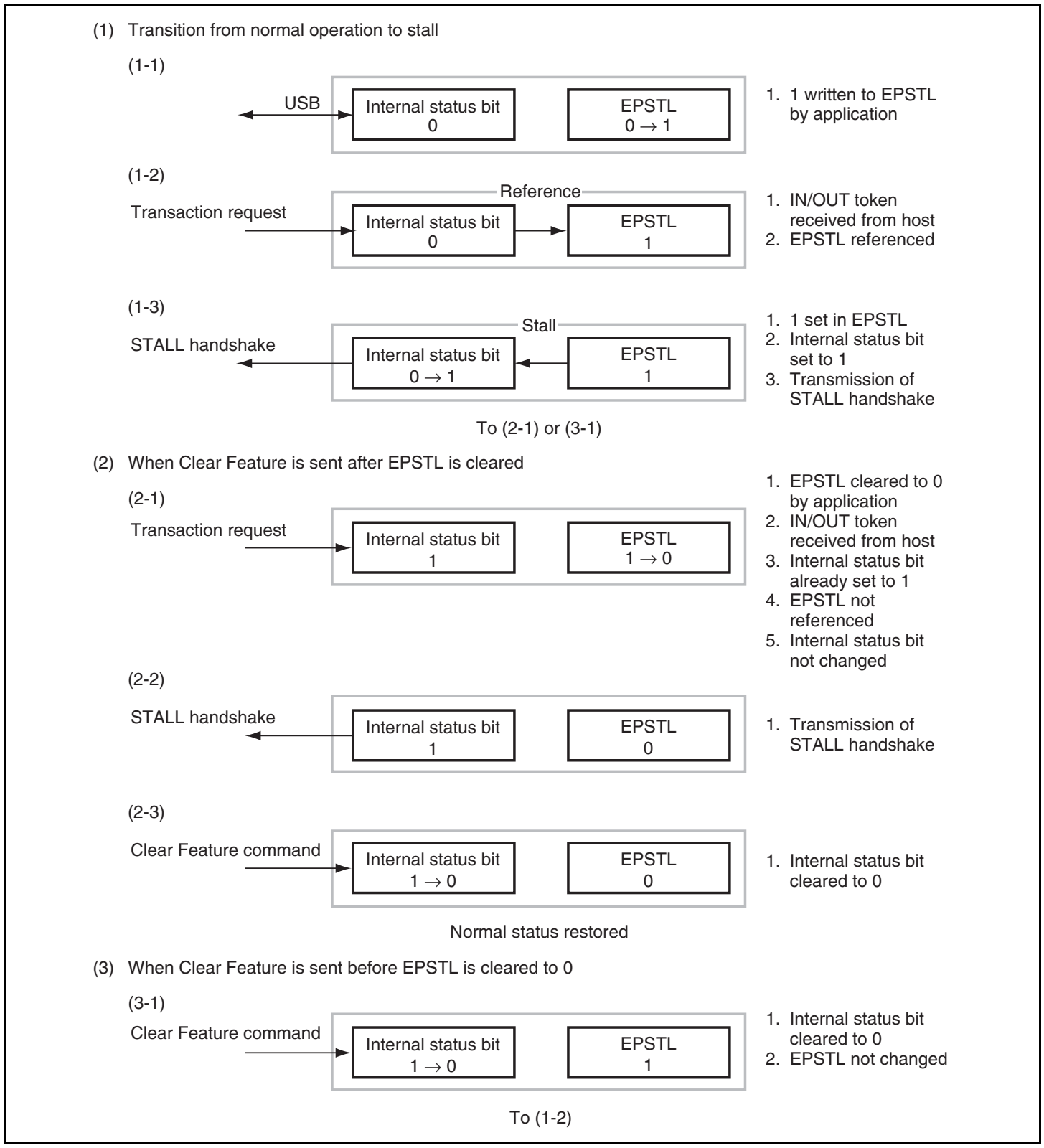


Figure 16.19 Forcible Stall by Application

16.7.3 Automatic Stall by USB Function Module

When a stall setting is made with the Set Feature command, or in the event of a USB specification violation, the USB function module automatically sets the internal status bit for the relevant endpoint without regard to the EPSTL register, and returns a stall handshake (1-1 in figure 16.20).

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to the EPSTL register. After a bit is cleared by the Clear Feature command, EPSTL is referenced (3-1 in figure 16.20). The USB function module continues to return a stall handshake while the internal status bit is set, since the internal status bit is set even if a transaction is executed for the corresponding endpoint (2-1 and 2-2 in figure 16.20). To clear a stall, therefore, the internal status bit must be cleared with a Clear Feature command (3-1 in figure 16.20). If set by the application, EPSTL should also be cleared (2-1 in figure 16.20).

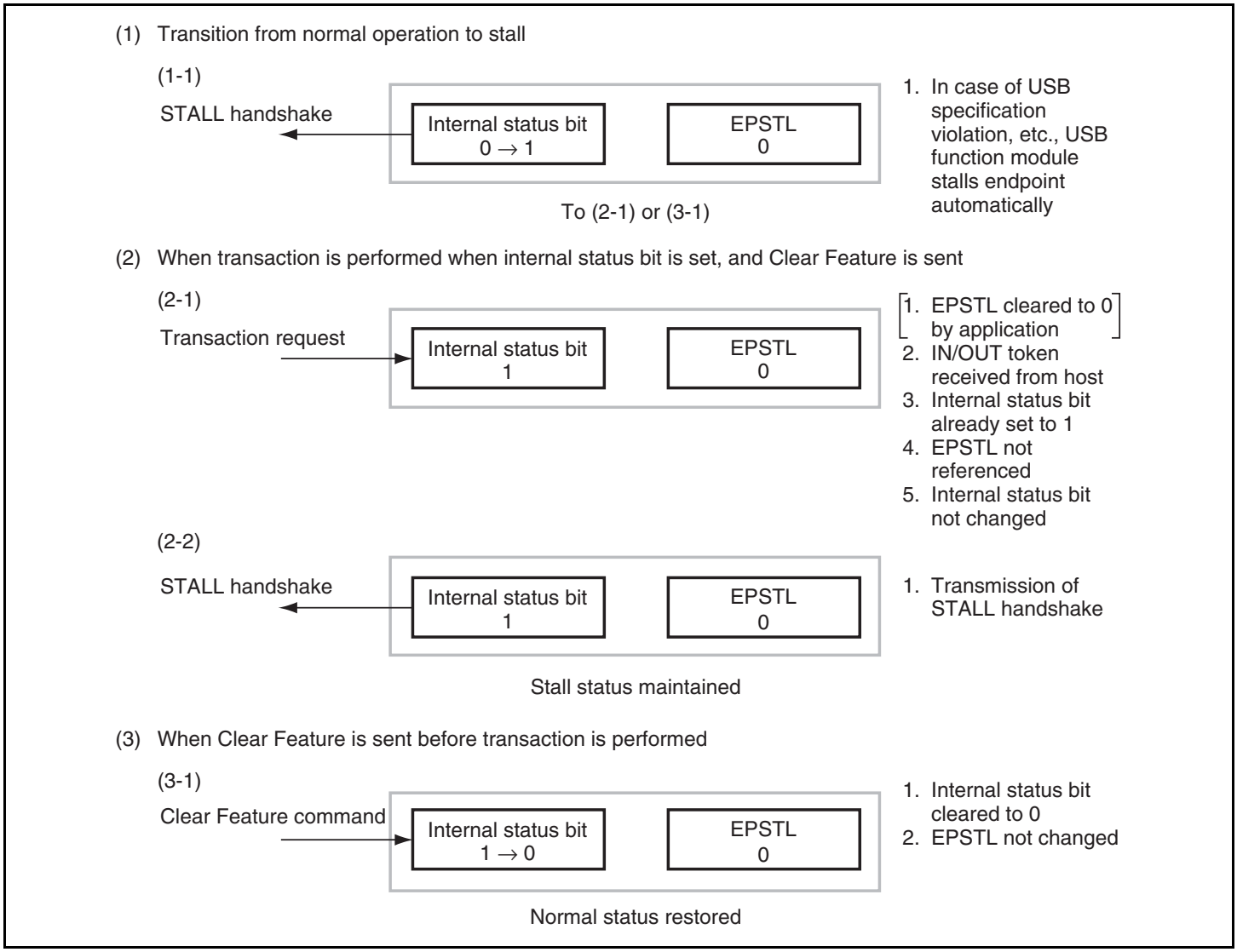


Figure 16.20 Automatic Stall by USB Function Module

16.8 DMA Transfer

16.8.1 Overview

DMA transfer can be performed for endpoints 1 and 2 in this module. Note that word or longword data cannot be transferred.

When endpoint 1 holds at least one byte of valid receive data, a DMA request for endpoint 1 is generated. When endpoint 2 holds no valid data, a DMA request for endpoint 2 is generated.

If the DMA transfer is enabled by setting the EP1 DMAE bit in the DMA transfer setting register to 1, zero-length data reception at endpoint 1 is ignored. When the DMA transfer is enabled, the EP1 RDFN bit and EP2 PKTE bit do not need to be set to 1 in TRG1. (Note that the PKTE bit in TRG1 must be set to 1 when the transfer data is less than the maximum number of bytes). When all the data received at EP1 is read, the FIFO automatically enters the EMPTY state. When the maximum number of bytes (64 bytes) are written to the EP2 FIFO, the FIFO automatically enters the FULL state, and the data in the FIFO can be transmitted (see figures 16.21 and 16.22).

16.8.2 Setting for the On-chip DMAC

The on-chip DMAC should be set for USB requests (using the $\overline{\text{DREQ}}$ signal), low-level input activation, byte size, full-address mode transfer, and the DTA bit = 1 in the DMABCR register. The on-chip DMAC will then be stopped after transfer has been completed the specified number of times. However, note that the $\overline{\text{DREQ}}$ signal continues to be asserted (held at the low level) regardless of the state of the DMAC when the DMA transfer requests still remains in this module.

16.8.3 DMA Transfer for Endpoints 1 and 4

When the data received at EP1 is transferred by the DMAC, the USB function module automatically performs the same processing as writing 1 to the RDFN bit in TRG1 if the currently selected FIFO becomes empty. Accordingly, in DMA transfer, do not write 1 to the RDFN bit in TRG1. If the user writes 1 to the RDFN bit in DMA transfer, correct operation cannot be guaranteed.

Figure 16.21 shows an example of receiving 150 bytes of data from the host. In this case, internal processing which is the same as writing 1 to the RDFN bit in TRG1 is automatically performed three times. This internal processing is performed when the currently selected data FIFO becomes empty. Accordingly, this processing is automatically performed both when 64-byte data is sent and when data less than 64 bytes is sent.

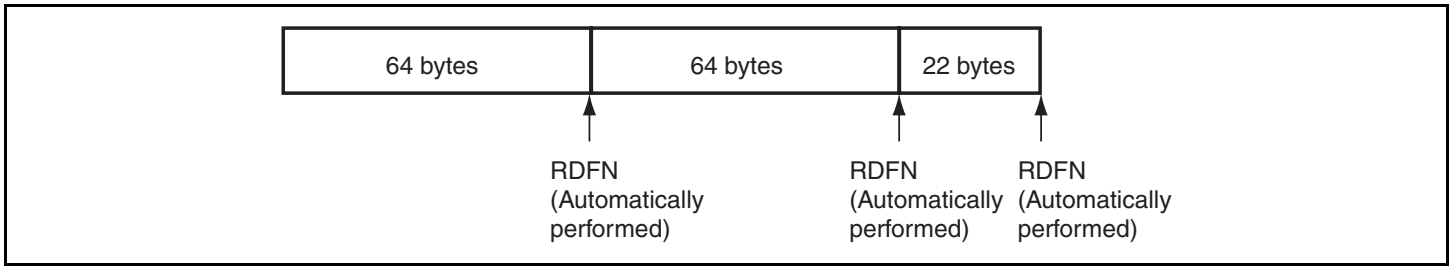


Figure 16.21 RDFN Bit Operation for EP1

16.8.4 DMA Transfer for Endpoints 2

When the transmit data at EP2 is transferred by the DMAC, the USB function module automatically performs the same processing as writing 1 to the PKTE bit in TRG1 if the currently selected FIFO (64 bytes) becomes full. Accordingly, to transfer data of a multiple of 64 bytes, the user need not write 1 to the PKTE bit in TRG1. To transfer data of less than 64 bytes, the user must write 1 to the PKTE bit using the DMA transfer end interrupt of the on-chip DMAC. If the user writes 1 to the PKTE bit in TRG1 when the maximum number of bytes (64 bytes) are transferred, correct operation cannot be guaranteed.

Figure 16.22 shows an example for transmitting 150 bytes of data to the host. In this case, internal processing which is the same as writing 1 to the PKTE bit in TRG1 is automatically performed twice. This internal processing is performed when the currently selected data FIFO becomes full. Accordingly, this processing is automatically performed only when 64-byte data is sent.

When the last 22 bytes are sent, the internal processing for writing 1 to the PKTE bit in TRG1 is not performed, and the user must write 1 to the PKTE bit by software. In this case, the application has no more data to transfer but the USB function module continues to output DMA requests for EP2 as long as the FIFO has an empty space. When all data has been transferred, write 0 to the EP2 DMAE bit in DMAR to cancel DMA requests for EP2.

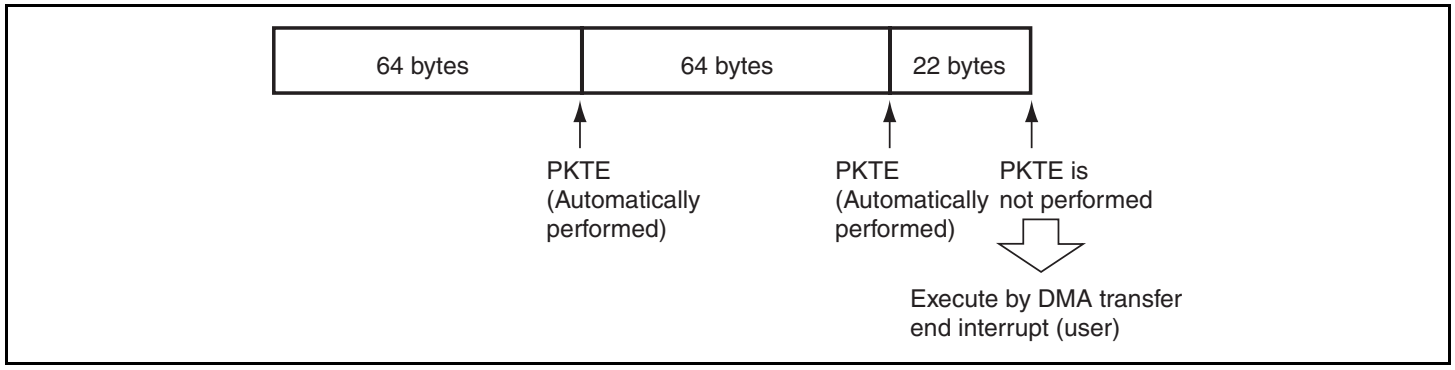


Figure 16.22 PKTE Bit Operation for EP2

16.9 Example of USB External Circuitry

1. USB Transceiver

   This module supports the on-chip transceiver only, not the external transceiver.

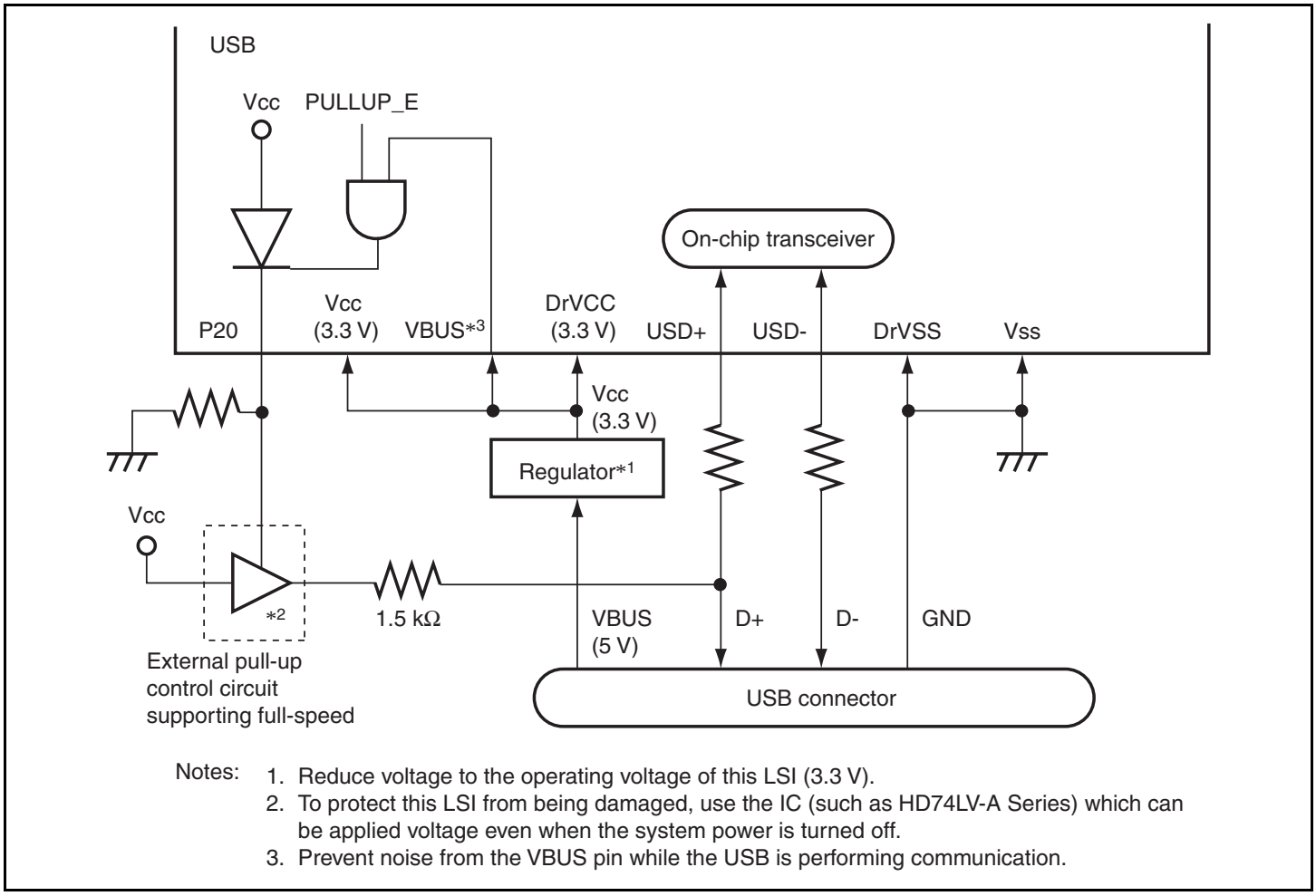
2. D+ Pull-Up Control

   The general output port (P20) is used for D+ pull-up control pin. The P20 pin is driven high by the PULLUP_E bit of CTLR when the USB cable VBUS is connected.

   Thus, USB host/hub connection notification (D+ pill-up) is enabled.

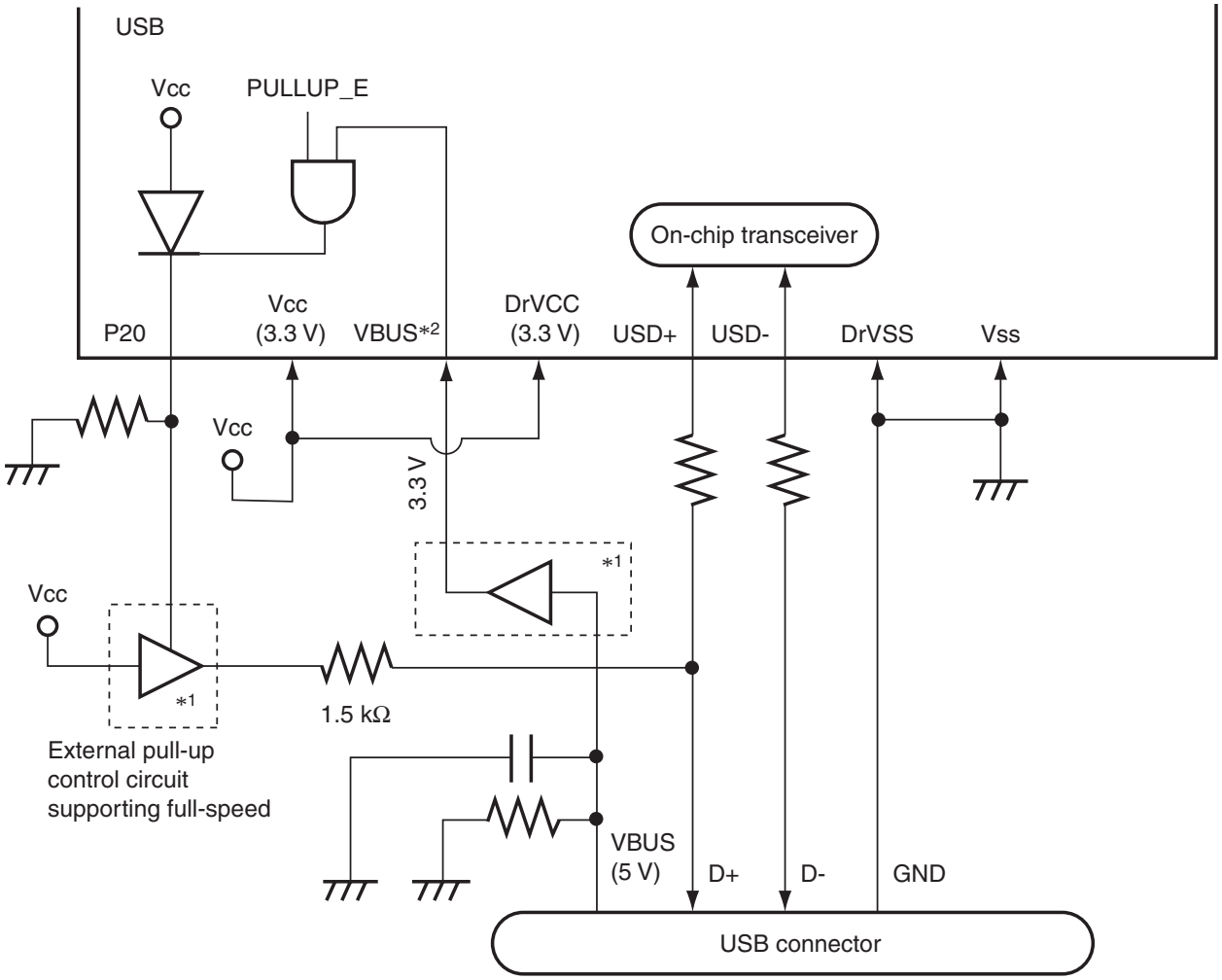
3. Detection of USB Cable Connection/Disconnection

   As USB states, etc., are managed by hardware in this module, a VBUS signal that recognizes connection/disconnection is necessary. The power supply signal (VBUS) in the USB cable is used for this purpose. However, if the cable is connected to the USB host/hub when the function (system installing this LSI) power is off, a voltage (5 V) will be applied from the USB host/hub. Therefore, an IC (such as an HD74LV1G08A or 2G08A) that allows voltage application when the system power is off should be connected externally.



Notes:
1. Reduce voltage to the operating voltage of this LSI (3.3 V).
2. To protect this LSI from being damaged, use the IC (such as HD74LV-A Series) which can be applied voltage even when the system power is turned off.
3. Prevent noise from the VBUS pin while the USB is performing communication.

Figure 16.23 Example of Circuitry in Bus Power Mode



Notes:
1. To protect this LSI from being damaged, use the IC (such as HD74LV-A Series) which can be applied voltage even when the system power is turned off.
2. Prevent noise from the VBUS pin while the USB is performing communication.

Figure 16.24 Example of Circuitry in Self Power Mode

16.10 Usage Notes

16.10.1 Receiving Setup Data

Note the following for EPDR0s that receives 8-byte setup data:

1. As a latest setup command must be received in high priority, the write from the USB bus takes priority over the read from the CPU. If the next setup command reception is started while the CPU is reading data after the data is received, the read from the CPU is forcibly terminated. Therefore, the data read after reception is started becomes invalid.
2. EPDR0s must always be read in 8-byte units. If the read is terminated at a midpoint, the data received at the next setup cannot be read correctly.

16.10.2 Clearing the FIFO

If a USB cable is disconnected during data transfer, the data being received or transmitted may remain in the FIFO. When disconnecting a USB cable, clear the FIFO.

While a FIFO is transferring data, it must not be cleared.

16.10.3 Overreading and Overwriting the Data Registers

Note the following when reading or writing to a data register of this module.

(1) Receive data registers

The receive data registers must not be read exceeding the valid amount of receive data, that is, the number of bytes indicated by the receive data size register. Even for EPDR1, which has double FIFO buffers, the maximum data to be read at one time is 64 bytes. After the data is read from the current valid FIFO buffer, be sure to write 1 to EPx RDFN in TRGx, which switches the valid buffer, updates the receive data size to the new number of bytes, and enables the next data to be received.

(2) Transmit data registers

The transmit data registers must not be written to exceeding the maximum packet size. Even for EPDR2, which has double FIFO buffers, write data within the maximum packet size at one time. After the data is written, write 1 to EPx PKTE in TRGx to switch the valid buffer and enable the next data to be written. Data must not be continuously written to the two FIFO buffers.

16.10.4 Assigning Interrupt Sources to EP0

The EP0-related interrupt sources indicated by the interrupt source bits (bits 0 to 3) in IFR0 must be assigned to the same interrupt signal with ISR0. The other interrupt sources have no limitations.

16.10.5 Clearing the FIFO When DMA Transfer is Enabled

EPDR1 cannot be cleared when DMA transfer for endpoint 1 is enabled (EP1DMAE in DMAR = 1). Cancel DMA transfer before clearing the register.

16.10.6 Notes on TR Interrupt

Note the following when using the transfer request interrupt (TR interrupt) for IN transfer to EP0i, EP2, and EP3.

The TR interrupt flag is set if the FIFO for the target EP has no data when the IN token is sent from the USB host. However, at the timing shown in figure 16.25, multiple TR interrupts occur successively. Take appropriate measures against malfunction in such a case.

Note: This module determines whether to return NAK if the FIFO of the target EP has no data when receiving the IN token, but the TR interrupt flag is set after a NAK handshake is sent. If the next IN token is sent before PKTE of TRG is written to, the TR interrupt flag is set again.

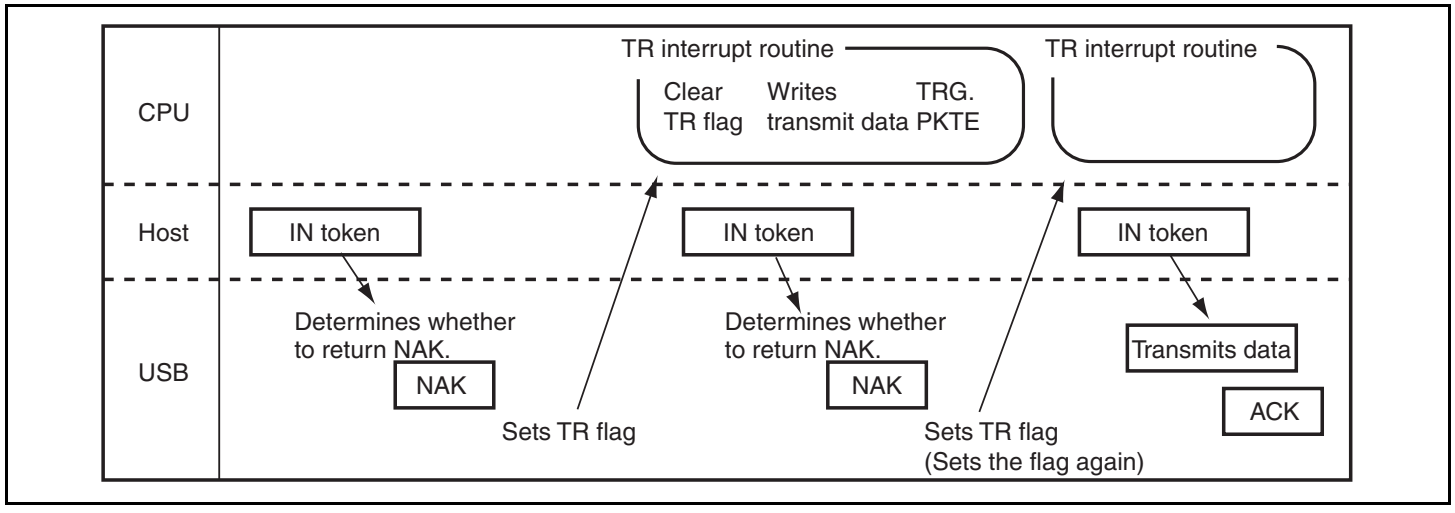


Figure 16.25 TR Interrupt Flag Set Timing

16.10.7 Module Stop Function Setting

Operation of the USB function module can be disabled or enabled using the module stop control register. The initial setting is for operation of the USB function module to be halted. Register access is enabled by clearing the module stop state. After clearing the module stop state, set the register after executing a 26-state dummy read. For details of the module stop control register, see section 25, Power-Down Modes.

Section 17 I²C Bus Interface 2 (IIC2)

This LSI has a four-channel I²C bus interface.

The I²C bus interface conforms to and provides a subset of the Philips I²C bus (inter-IC bus) interface functions. The register configuration that controls the I²C bus differs partly from the Philips configuration, however.

Figure 17.1 shows a block diagram of the I²C bus interface 2. Figure 17.2 shows an example of I/O pin connections to external circuits.

17.1 Features

- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independent from each other, the continuous transmission/reception can be performed.
- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically. If transmission/reception is not yet possible, set the SCL to 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, SCL and SDA pins function as NMOS open-drain outputs.

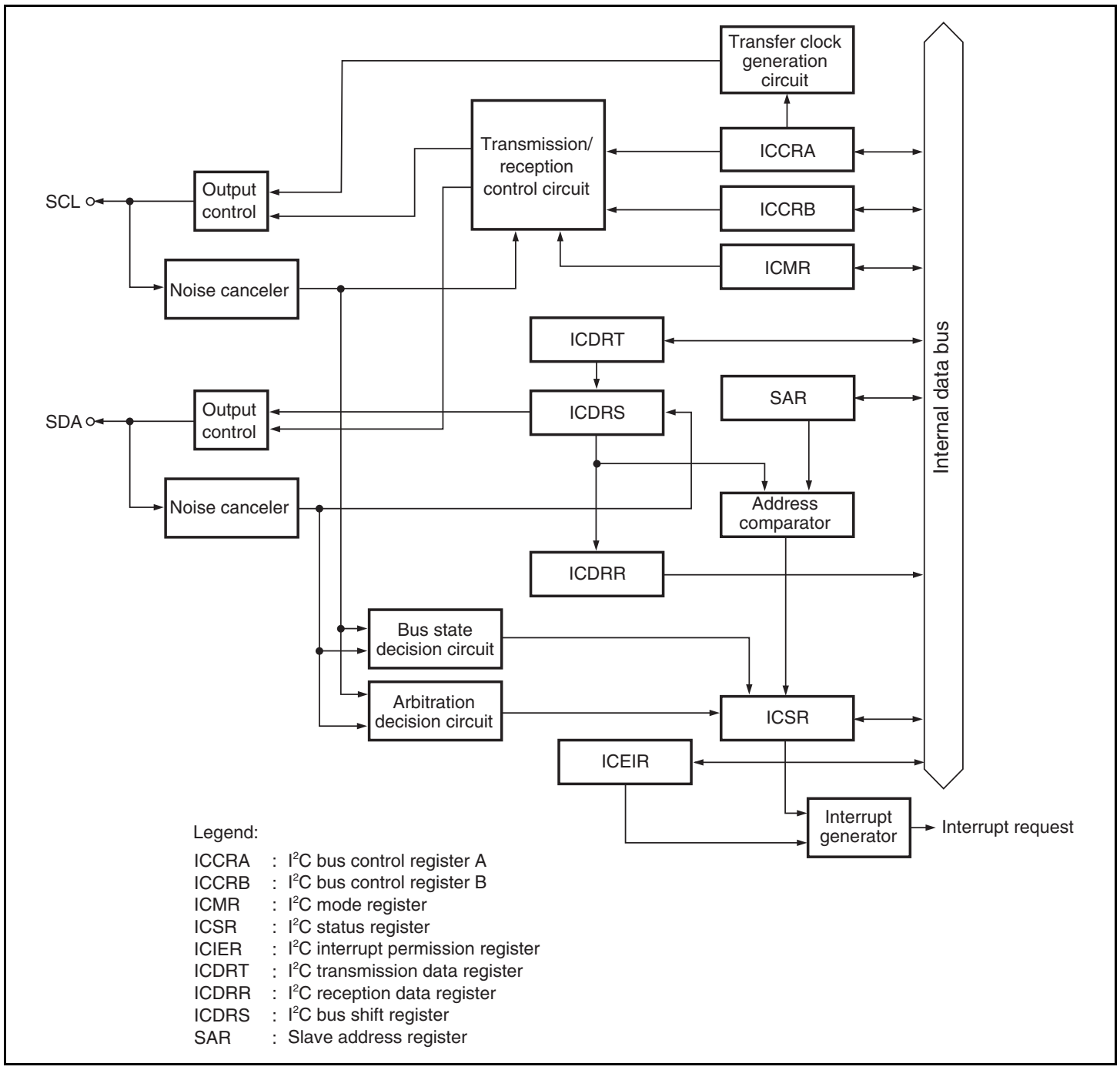


Figure 17.1 Block Diagram of I²C Bus Interface 2

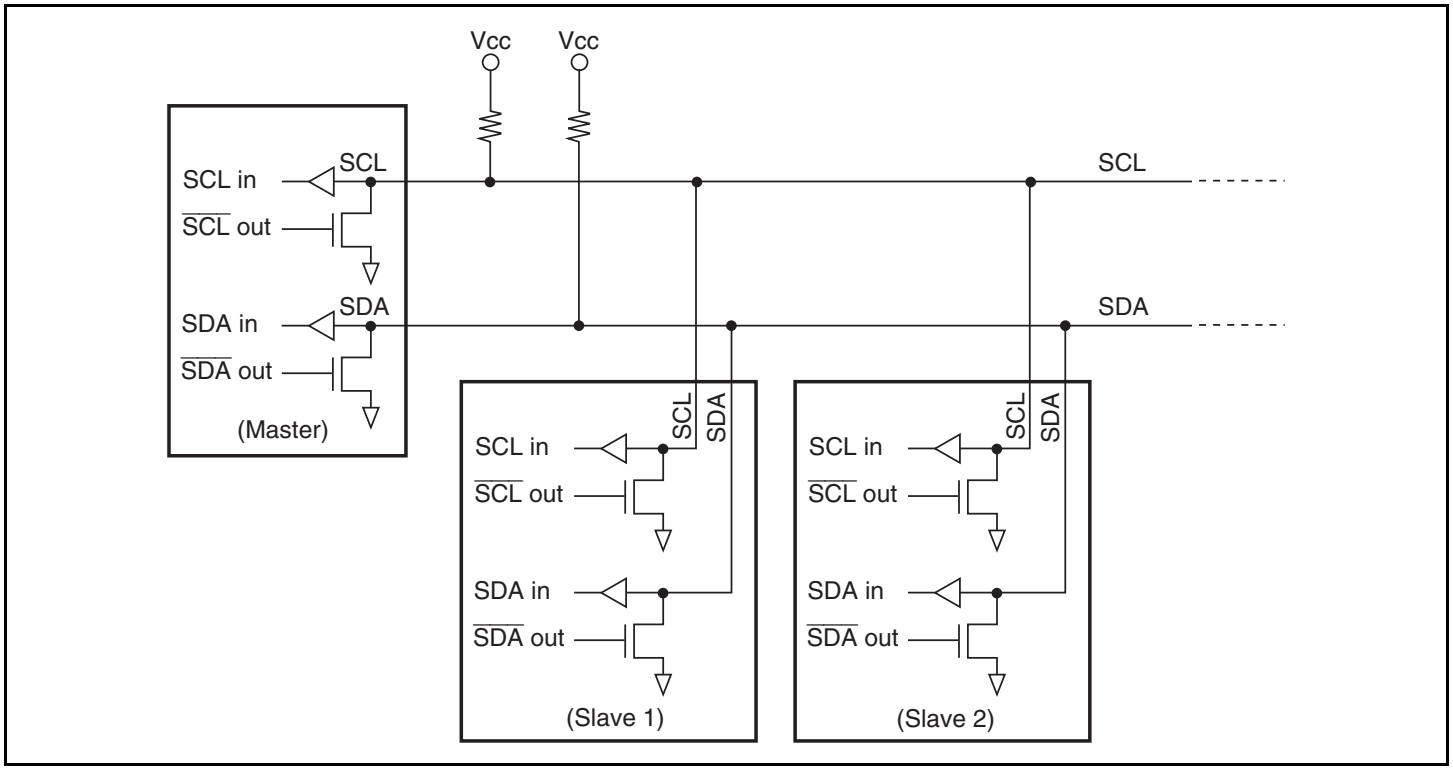


Figure 17.2 External Circuit Connections of I/O Pins

17.2 Input/Output Pins

Table 17.1 shows the pin configuration of the I²C bus interface 2.

Table 17.1 Pin Configuration

Name	Abbreviation	I/O	Function
Serial clock pin	SCL0	I/O	IIC2_0 serial clock input/output
Serial data pin	SDA0	I/O	IIC2_0 serial data input/output
Serial clock pin	SCL1	I/O	IIC2_1 serial clock input/output
Serial data pin	SDA1	I/O	IIC2_1 serial data input/output
Serial clock pin	SCL2	I/O	IIC2_2 serial clock input/output
Serial data pin	SDA2	I/O	IIC2_2 serial data input/output
Serial clock pin	SCL3	I/O	IIC2_3 serial clock input/output
Serial data pin	SDA3	I/O	IIC2_3 serial data input/output

Note: The pin symbols are represented as SCL and SDA; channel numbers are omitted in this manual.

17.3 Register Descriptions

The I²C bus interface has the following registers.

Channel 0

- I²C bus control register A_0 (ICCRA_0)
- I²C bus control register B_0 (ICCRB_0)
- I²C bus mode register_0 (ICMR_0)
- 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)

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)
- I²C bus transmit data register_2 (ICDRT_2)
- I²C bus receive data register_2 (ICDRR_2)
- I²C bus shift register_2 (ICDRS_2)

Channel 3

- I²C bus control register A_3 (ICCRA_3)
- I²C bus control register B_3 (ICCRB_3)
- I²C bus mode register_3 (ICMR_3)
- I²C bus interrupt enable register_3 (ICIER_3)
- I²C bus status register_3 (ICSR_3)
- 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)

17.3.1 I²C Bus Control Register A (ICCRA)

ICCRA is an 8-bit readable/writable register that enables or disables the I²C bus interface, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface Enable 0: This module is halted. 1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable This bit enables or disables the next operation when 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 bit should be made between transfer frames. In addition, TRS is set to 1 automatically in slave receive mode if the seventh bit of the start condition matches the slave address set in SAR and the eighth bit is set to 1. 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 In the master mode, these bits should be set according to the necessary transfer rate (see table 17.2). In the slave mode, they are used to secure the data setup time in transmit mode. The data setup time is 10 tcyc if CKS3 is cleared to 0 and 20 tcyc if CKS3 is set to 1.
2	CKS2	0	R/W	
1	CKS1	0	R/W	
0	CKS0	0	R/W	

Table 17.2 Transfer Rate

Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate				
CKS3	CKS2	CKS1	CKS0	Clock	φ = 8 MHz	φ = 10 MHz	φ = 20 MHz	φ = 25 MHz	φ = 33 MHz
0	0	0	0	φ/28	286 kHz	357 kHz	714 kHz*	893 kHz*	1179 kHz*
			1	φ/40	200 kHz	250 kHz	500 kHz*	625 kHz*	825 kHz*
		1	0	φ/48	167 kHz	208 kHz	417 kHz*	521 kHz*	688 kHz*
			1	φ/64	125 kHz	156 kHz	313 kHz	391 kHz	516 kHz*
	1	0	0	φ/168	47.6 kHz	59.5 kHz	119 kHz	149 kHz	196 kHz
			1	φ/100	80.0 kHz	100 kHz	200 kHz	250 kHz	330 kHz
		1	0	φ/112	71.4 kHz	89.3 kHz	179 kHz	223 kHz	295 kHz
			1	φ/128	62.5 kHz	78.1 kHz	156 kHz	195 kHz	258 kHz
1	0	0	0	φ/56	143 kHz	179 kHz	357 kHz	446 kHz*	589 kHz*
			1	φ/80	100 kHz	125 kHz	250 kHz	313 kHz	413 kHz*
		1	0	φ/96	83.3 kHz	104 kHz	208 kHz	260 kHz	344 kHz
			1	φ/128	62.5 kHz	78.1 kHz	156 kHz	195 kHz	258 kHz
	1	0	0	φ/336	23.8 kHz	29.8 kHz	59.5 kHz	74.4 kHz	98.2 kHz
			1	φ/200	40.0 kHz	50.0 kHz	100 kHz	125 kHz	165 kHz
		1	0	φ/224	35.7 kHz	44.6 kHz	89.3 kHz	112 kHz	147 kHz
			1	φ/256	31.3 kHz	39.1 kHz	78.1 kHz	97.7 kHz	129 kHz

Note: * Correct operation cannot be guaranteed since the transfer rate is beyond the I²C bus interface specification (normal mode: maximum 100 kHz, high-speed mode: maximum 400 kHz).

17.3.2 I²C Bus Control Register B (ICCRB)

ICCRB is an 8-bit readable/writable register that issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls reset in I²C control.

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy This bit enables to confirm whether the I²C bus is occupied or released and to issue start and stop conditions in master mode. This bit is set to 1 when the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Follow this procedure when also re-transmitting a start condition. Write 0 to BBSY and 0 to SCP to issue a stop condition. To issue a start/stop condition, use the MOV instruction.
6	SCP	1	R/W	Start Condition/Stop Condition Prohibit The SCP bit controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. If 1 is written, the data is not stored.
5	SDAO	1	R	This bit monitors SDA output level. When reading and SDA0 is 1, the SDA pin outputs high. When reading and SDA0 is 0, the SDA pin outputs low. The write value should always be 1.
4	—	1	R/W	Reserved The write value should always be 1.
3	SCLO	1	R	This bit monitors SCL output level. When reading and SCLO is 1, the SCL pin outputs high. When reading and SCLO is 0, the SCL pin outputs low.
2	—	1	—	Reserved This bit is always read as 1.

Bit	Bit Name	Initial Value	R/W	Description
1	IICRST	0	R/W	IIC Control Part Reset This bit resets control parts except for I²C registers. If this bit is set to 1 when hang-up is occurred because of communication failure during I²C operation, I²C control part can be reset without setting ports and initializing registers.
0	—	1	—	Reserved This bit is always read as 1.

17.3.3 I²C Bus Mode Register (ICMR)

ICMR controls the master mode wait and selects the number of transfer bits.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved The write value should always be 0.
6	WAIT	0	R/W	Wait Insertion This bit selects whether to insert a wait after data transfer except for the acknowledge bit. When WAIT is set to 1, after the fall of the clock for the final data bit, low period is extended for two transfer clocks. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted. The setting of this bit is invalid in slave mode.
5, 4	—	All 1	—	Reserved These bits are always read as 1.
3	BCWP	1	R/W	BC Write Protect This bit controls the BC2 to BC0 modifications. When modifying BC2 to BC0, this bit should be cleared to 0 and use the MOV instruction. 0: When writing, values of BC2 to BC0 are set. 1: When reading, 1 is always read. When writing, settings of BC2 to BC0 are invalid.

Bit	Bit Name	Initial Value	R/W	Description
2	BC2	0	R/W	Bit Counter 2 to 0
1	BC1	0	R/W	These bits specify the number of bits to be transferred next. When read, the remaining number of transfer bits is indicated. The data is transferred with one addition acknowledge bit. Bit BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL line is low. The value returns to 000 at the end of a data transfer, including the acknowledge bit. 000: 9 bits 001: 2 bits 010: 3 bits 011: 4 bits 100: 5 bits 101: 6 bits 110: 7 bits 111: 8 bits
0	BC0	0	R/W	

17.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER is an 8-bit readable/writable register that enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transferred, and confirms acknowledge bits to be received.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, this bit enables or disables the transmit data empty interrupt (TXI). 0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable This bit enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by clearing the TEND bit or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled. 1: Transmit end interrupt request (TEI) is enabled.
5	RIE	0	R/W	Receive Interrupt Enable This bit enables or disables the receive data full interrupt request (RXI) when a received data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) is disabled. 1: Receive data full interrupt request (RXI) is enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	NAKIE	0	R/W	NACK Receive Interrupt Enable This bit enables or disables the NACK receive interrupt request (NAKI) when the NACKF and AL bits in ICSR are set to 1. NAKI can be canceled by clearing the NACKF, AL, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is disabled. 1: NACK receive interrupt request (NAKI) is enabled.
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (STPI) is disabled. 1: Stop condition detection interrupt request (STPI) is enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgement Select 0: The value of the acknowledge bit is ignored, and continuous transfer is performed. 1: If the acknowledge bit is 1, continuous transfer is interrupted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowledge data that are returned by the receive device. This bit cannot be modified. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be sent at the acknowledge timing. 0: 0 is sent at the acknowledge timing. 1: 1 is sent at the acknowledge timing.

17.3.5 I²C Bus Status Register (ICSR)

ICSR is an 8-bit readable/writable register that performs confirmation of interrupt request flags and status.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Setting condition] • When data is transferred from ICDRT to ICDRS and ICDRT becomes empty. • When TRS has been set. • When a start condition (including retransmission) has been issued. • When a transition from the receive mode to the transmit mode has been made in the slave mode. [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1. • When data is written in ICDRT.
6	TEND	0	R/W	Transmit End [Setting conditions] • When the ninth clock of SCL is rose while the TDRE flag is 1. [Clearing conditions] • When 0 is written in TEND after reading TEND = 1. • When data is written in ICDRT.
5	RDRF	0	R/W	Receive Data Register Full [Setting condition] • When a received data is transferred from ICDRS to ICDRR. [Clearing conditions] • When 0 is written in RDRF after reading RDRF = 1. • When data is read from ICDRR.

Bit	Bit Name	Initial Value	R/W	Description
4	NACKF	0	R/W	No Acknowledge Detection Flag [Setting condition] • When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1. [Clearing condition] • When 0 is written in NACKF after reading NACKF = 1.
3	STOP	0	R/W	Stop Condition Detection Flag [Setting condition] • In master mode, when a stop condition is detected after frame transfer. • In slave mode, when a stop condition is detected after the general call address or the first byte slave address, next to detection of start condition, accords with the address set in SAR. [Clearing condition] • When 0 is written in STOP after reading STOP = 1.
2	AL	0	R/W	Arbitration Lost Flag This flag indicates that arbitration was lost in master mode. When two or more master devices attempt to seize the bus at nearly the same time, if the I²C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been taken by another master. [Setting conditions] • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode. • When the SDA pin outputs high in master mode while a start condition is detected. [Clearing condition] • When 0 is written in AL after reading AL =1.

Bit	Bit Name	Initial Value	R/W	Description
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR. [Setting condition] • When the slave address is detected in slave receive mode. • When the general call address is detected in slave receive mode. [Clearing condition] • When 0 is written in AAS after reading AAS=1
0	ADZ	0	R/W	General Call Address Recognition Flag This bit is valid in slave receive mode. [Setting condition] • When the general call address is detected in slave receive mode. [Clearing conditions] • When 0 is written in ADZ after reading ADZ=1.

17.3.6 Slave Address Register (SAR)

SAR is an 8-bit readable/writable register that sets slave address. When the chip is 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 chip operates as the slave device.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA6 to SVA0	All 0	R/W	Slave Address 6 to 0 These bits set a unique address in bits SVA6 to SVA0, differing from the addresses of other slave devices connected to the I²C bus.
0	—	0	R/W	Reserved This bit is readable/writable. The write value should always be 0.

17.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the I²C bus shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. The initial value of ICDRT is H'FF.

17.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received, ICDRR transfers the received data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot be written to this register. The initial value of ICDRR is H'FF.

17.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after data of one byte is received. This register cannot be read from the CPU.

17.4 Operation

17.4.1 I²C Bus Format

Figure 17.3 shows the I²C bus formats. Figure 17.4 shows the I²C bus timing. The first frame following a start condition always consists of 8 bits.

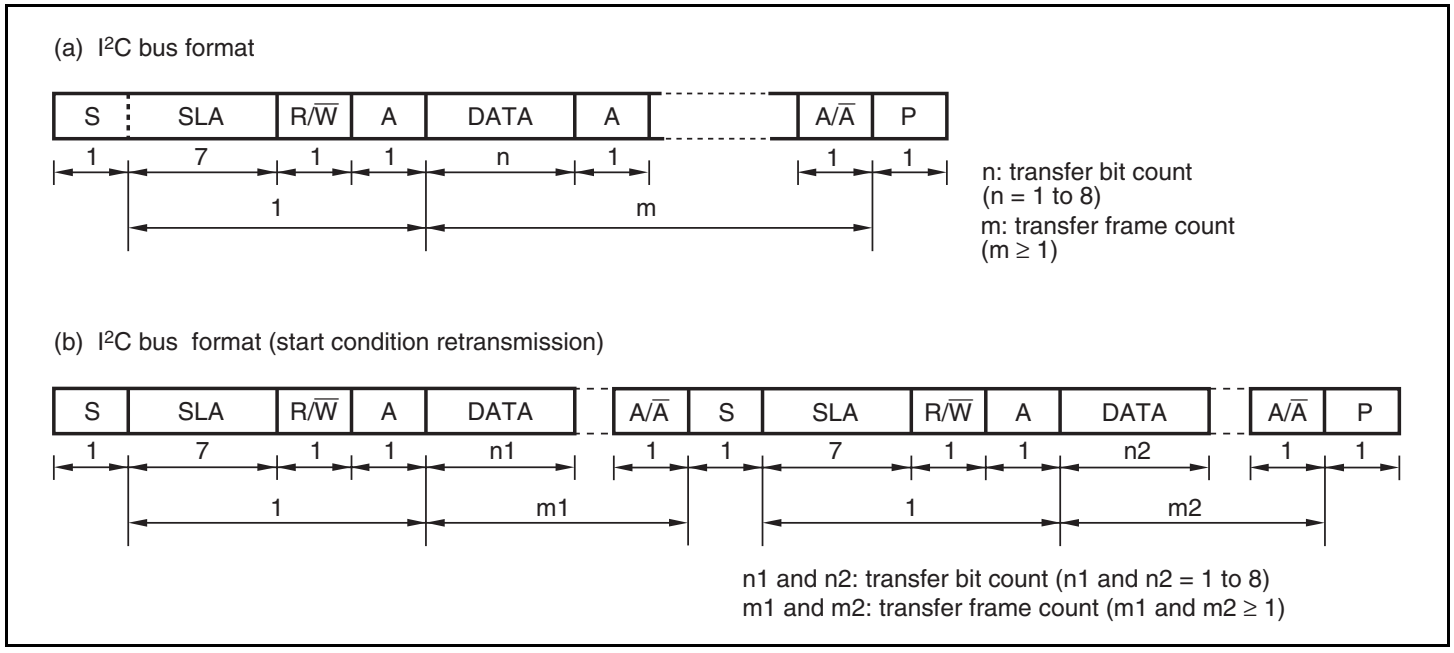


Figure 17.3 I²C Bus Formats

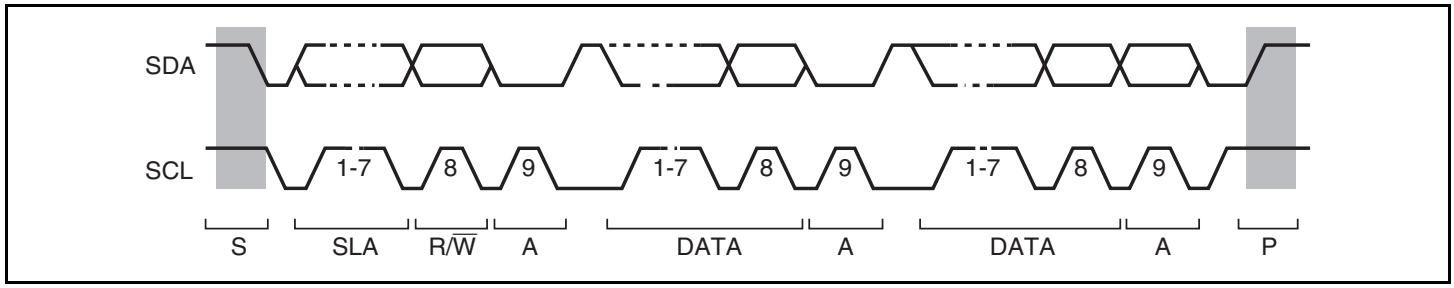


Figure 17.4 I²C Bus Timing

Legend:

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/$\overline{W}$: Indicates the direction of data transfer: from the slave device to the master device when R/$\overline{W}$ is 1, or from the master device to the slave device when R/$\overline{W}$ is 0.

A: Acknowledge. The receiving device drives SDA to low.

DATA: Transferred data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

17.4.2 Master Transmit Operation

In I²C bus format master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCRA to 1. Set the WAIT bit in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. Read the BBSY flag in ICCRB to confirm that the bus is free. Set the MST and TRS bits in ICCRA to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using MOV instruction. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first byte data show the slave address and R/W) to ICDRT. After this, when TDRE is cleared to 0, 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 to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT, and clear TDRE and TEND. When ACKBR is 1, the slave device has not been acknowledged, so issue the stop condition. To issue the stop condition, write 0 to BBSY and SCP using MOV instruction. SCL is fixed low 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, thus clearing TDRE.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TEND or NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode.

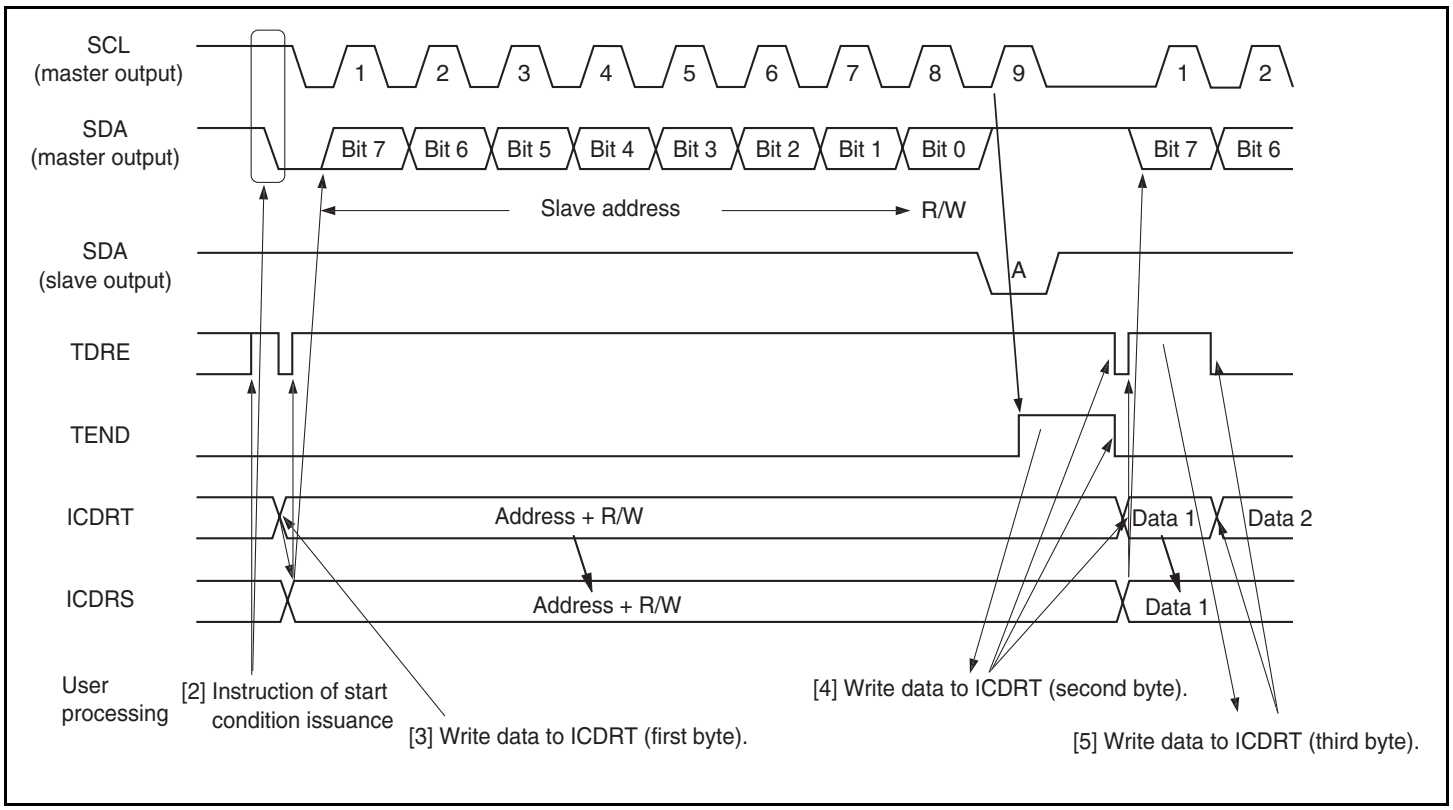


Figure 17.5 Master Transmit Mode Operation Timing 1

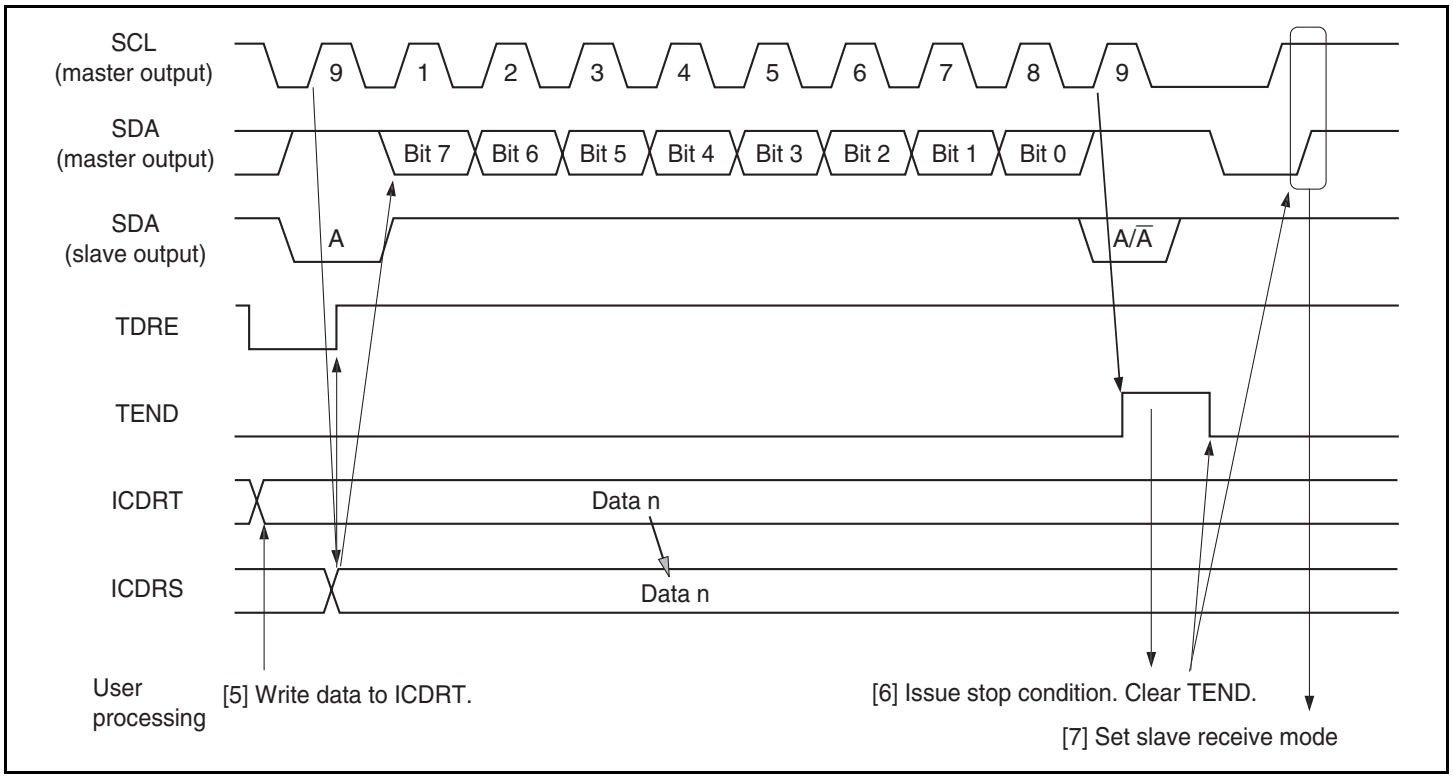


Figure 17.6 Master Transmit Mode Operation Timing 2

17.4.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal. The reception procedure and operations in master receive mode are shown below.

1. Clear the TEND bit in ICSR to 0, then clear the TRS bit in ICCRA to 0 to switch from master transmit mode to master receive mode. Then, clear the TDRE bit to 0.
2. When ICDRR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. The master device outputs the level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.
3. After the reception of first frame data is completed, the RDRF bit in ICST is set to 1 at the rise of 9th receive clock pulse. At this time, the received data is read by reading ICDRR.
4. The continuous reception is performed by reading ICDRR and clearing RDRF to 0 every time RDRF is set. If 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 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 rise of the 9th receive clock pulse, read ICDRR. Then, clear RCVD.
7. When the STOP bit in ICSR is set to 1, read ICDRR and clear RDRF to 0. Then clear the RCVD bit to 0.
8. The operation returns to the slave receive mode.

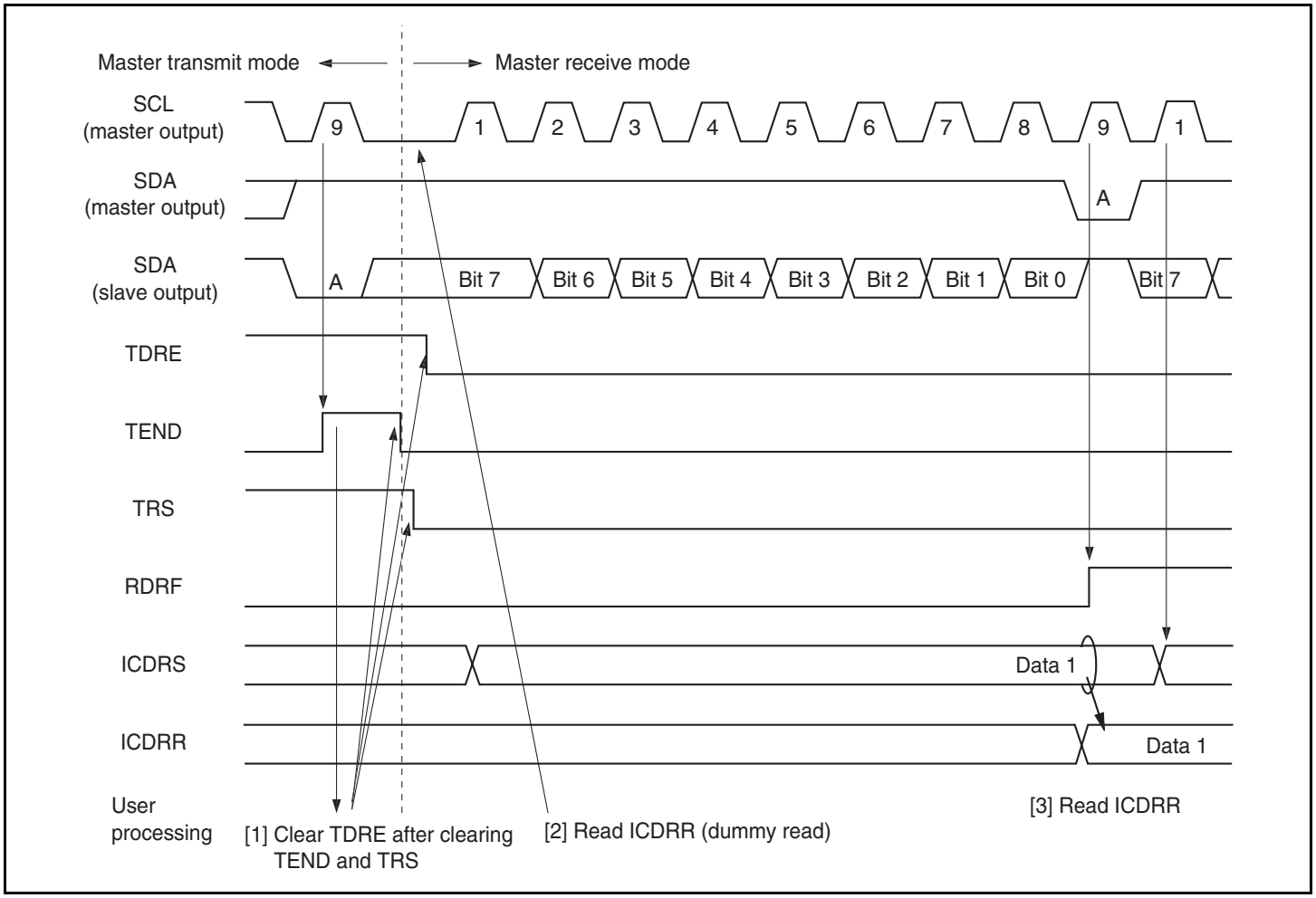


Figure 17.7 Master Receive Mode Operation Timing 1

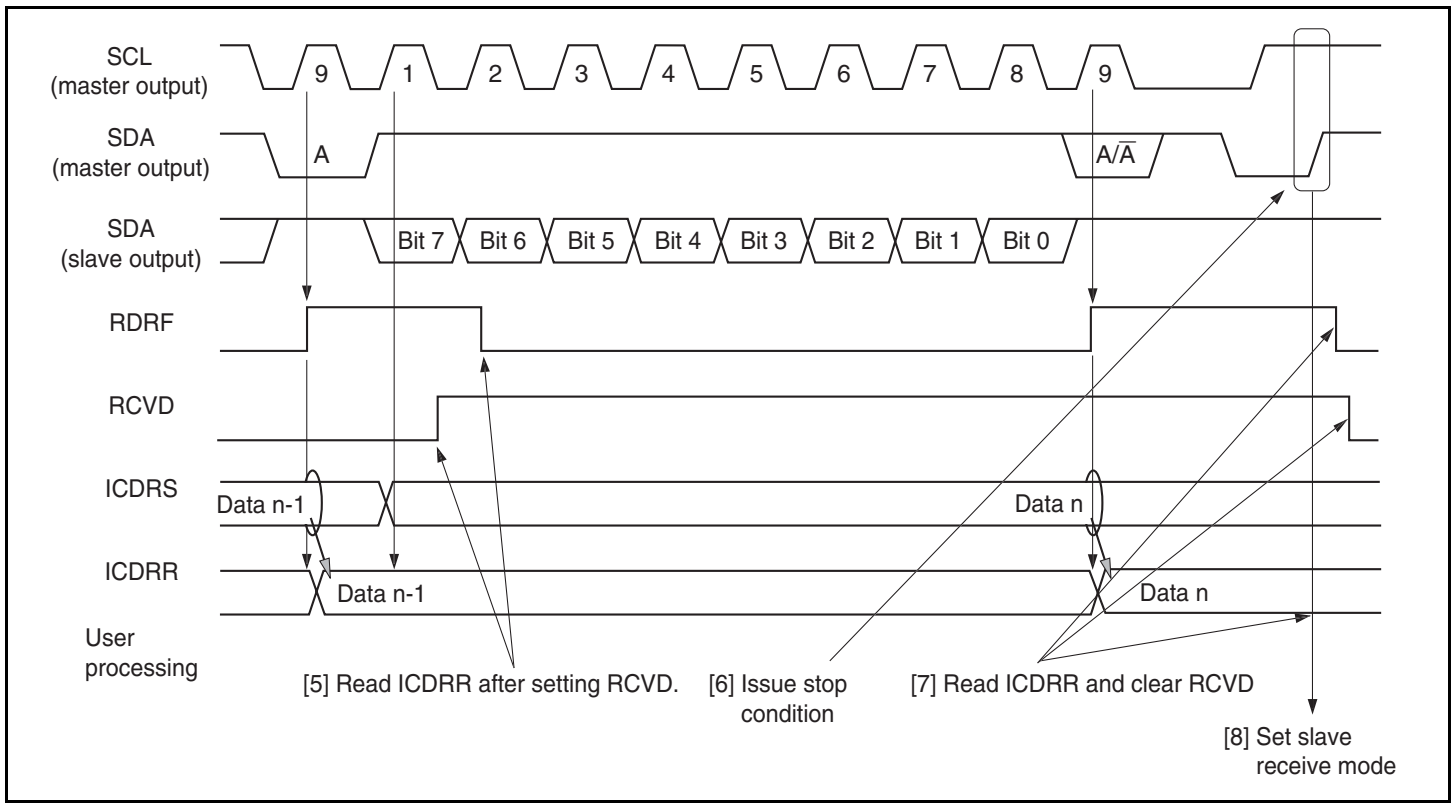


Figure 17.8 Master Receive Mode Operation Timing 2

17.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCRA to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCRA to 1. (Initial setting) 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 detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At this time, if the 8th bit data (R/$\overline{W}$) is 1, the TRS in ICCRA and TDRE in ICSR are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by clearing TDRE after writing transmit data to ICDRT every time TDRE is set.
3. If TDRE is set after writing 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 the end processing, and read ICDRR (dummy read). SCL is free.
5. Clear TDRE.

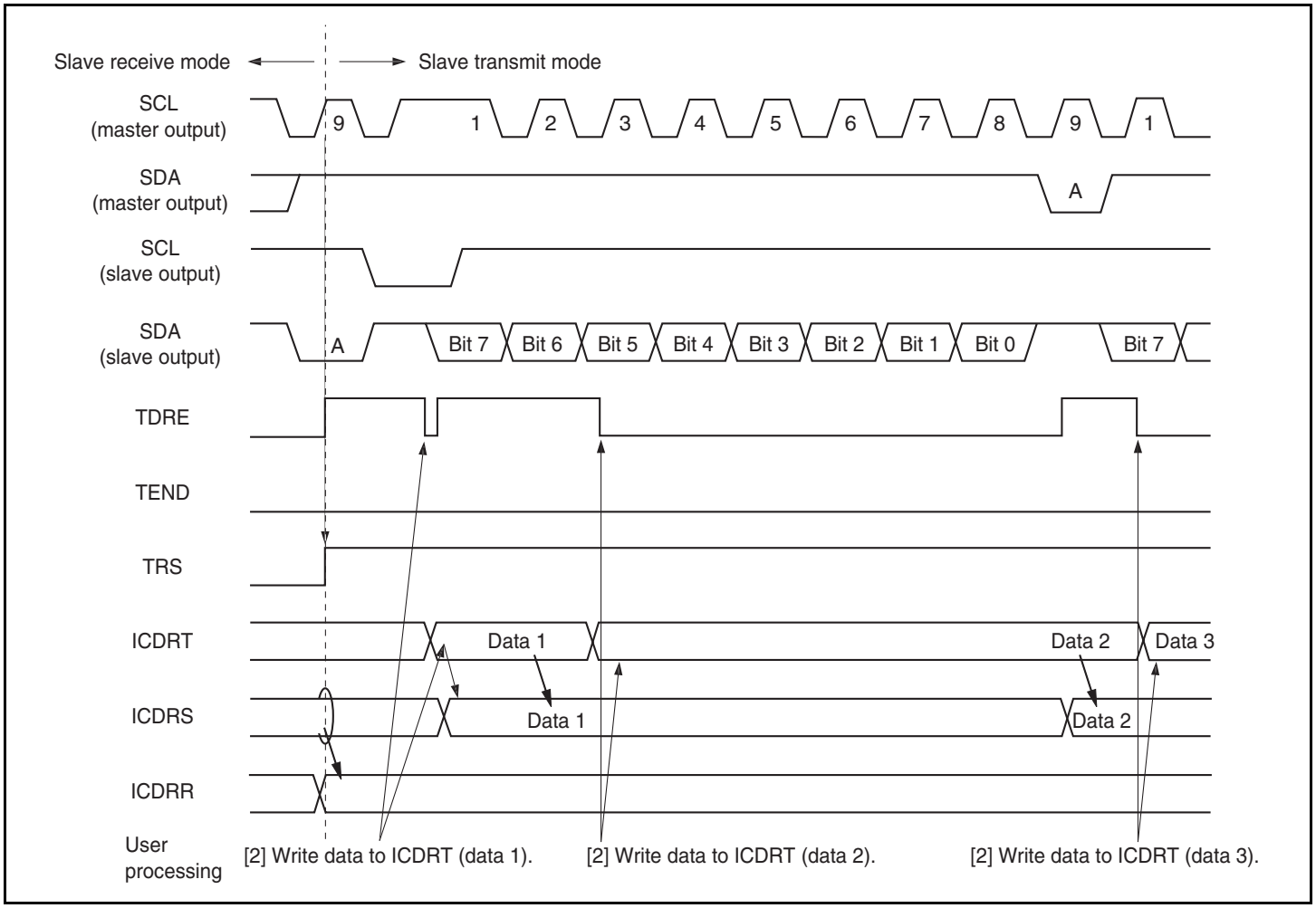


Figure 17.9 Slave Transmit Mode Operation Timing 1

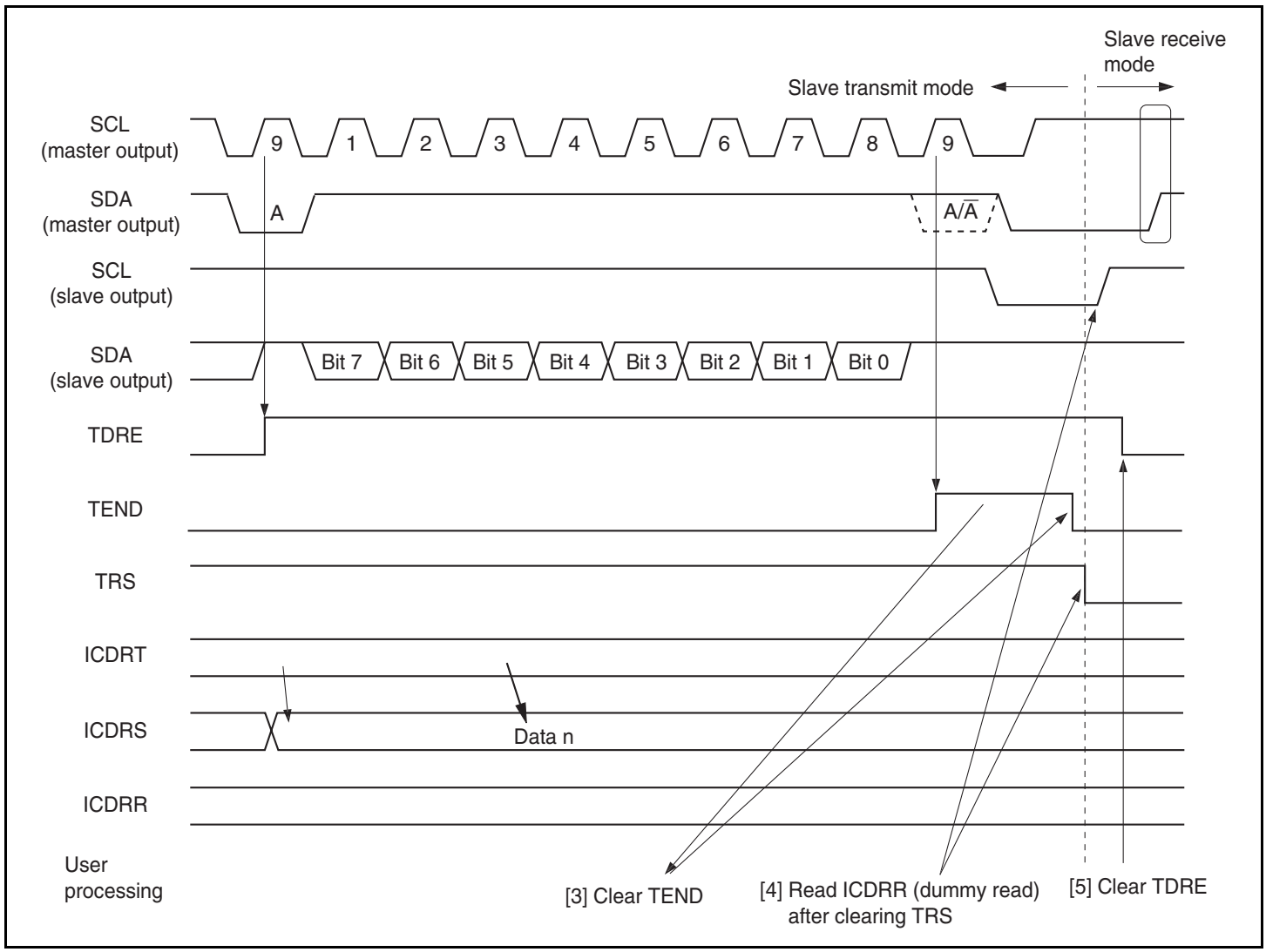


Figure 17.10 Slave Transmit Mode Operation Timing 2

17.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCRA to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCRA to 1. (Initial setting) 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 detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read) and RDRF is cleared. (Since the read data show the slave address and R/$\overline{W}$, it is not used.)
3. Clear RDRF after reading ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is 1, SCL is fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR, to be returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

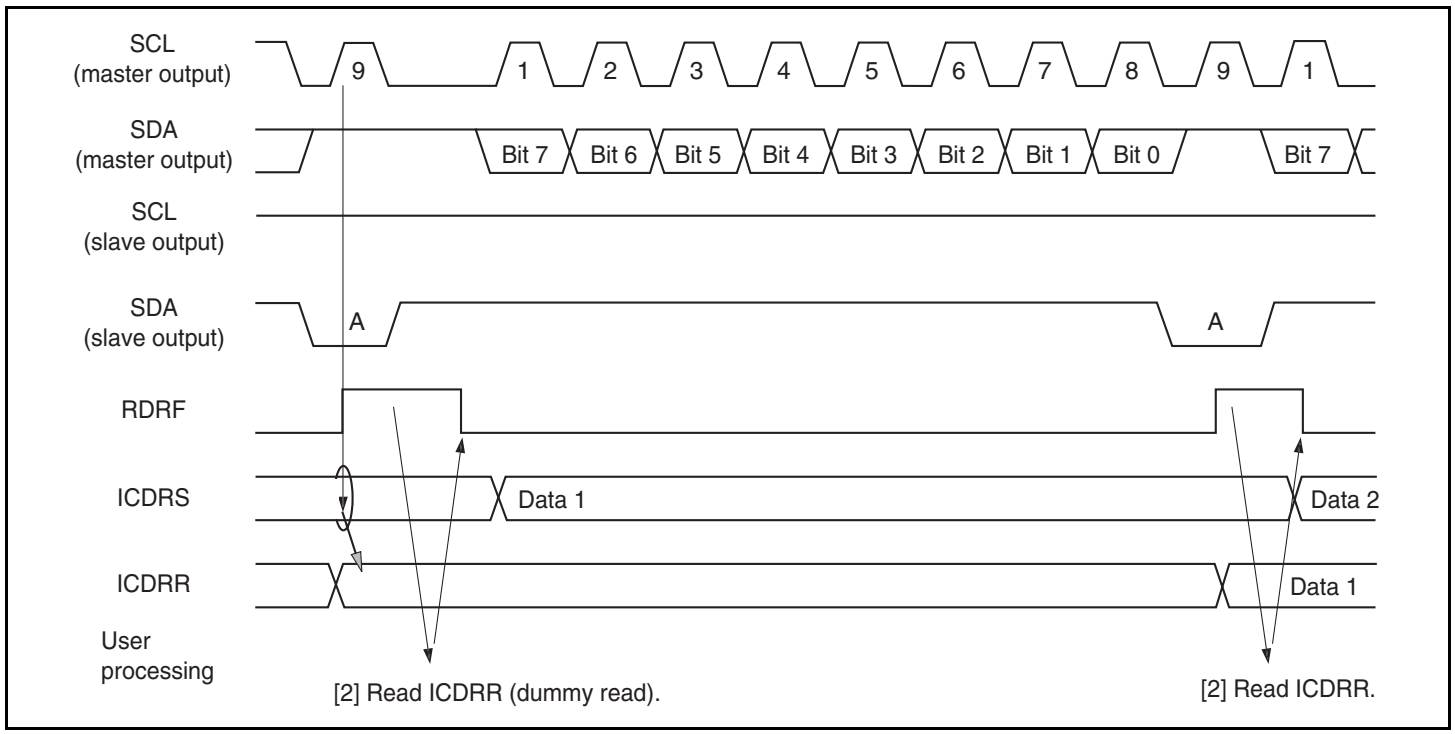


Figure 17.11 Slave Receive Mode Operation Timing 1

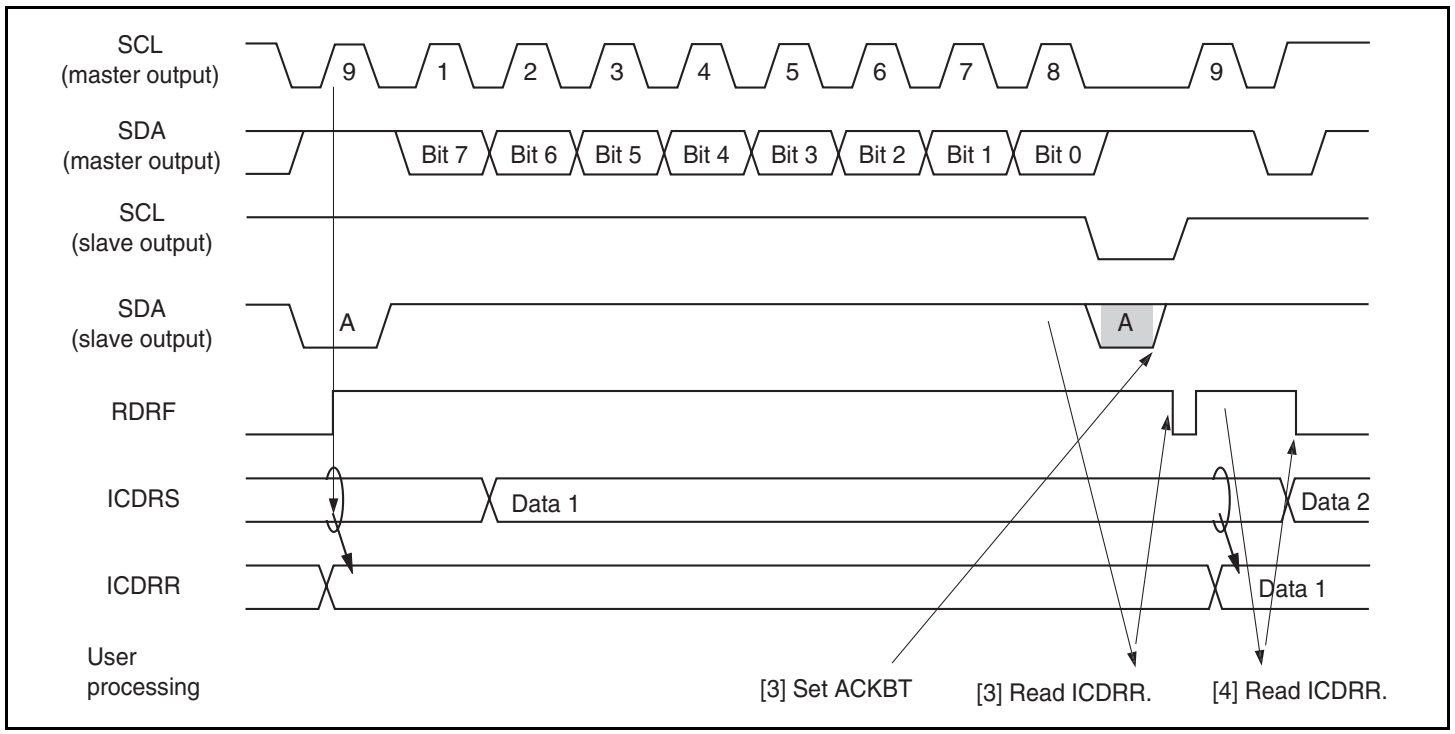


Figure 17.12 Slave Receive Mode Operation Timing 2

17.4.6 Noise Canceler

The logic levels at the SCL and SDA pins are routed through noise cancelers before being latched internally. Figure 17.13 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the system clock, but is not passed forward to the next circuit unless the outputs of both latches agree. If they do not agree, the previous value is held.

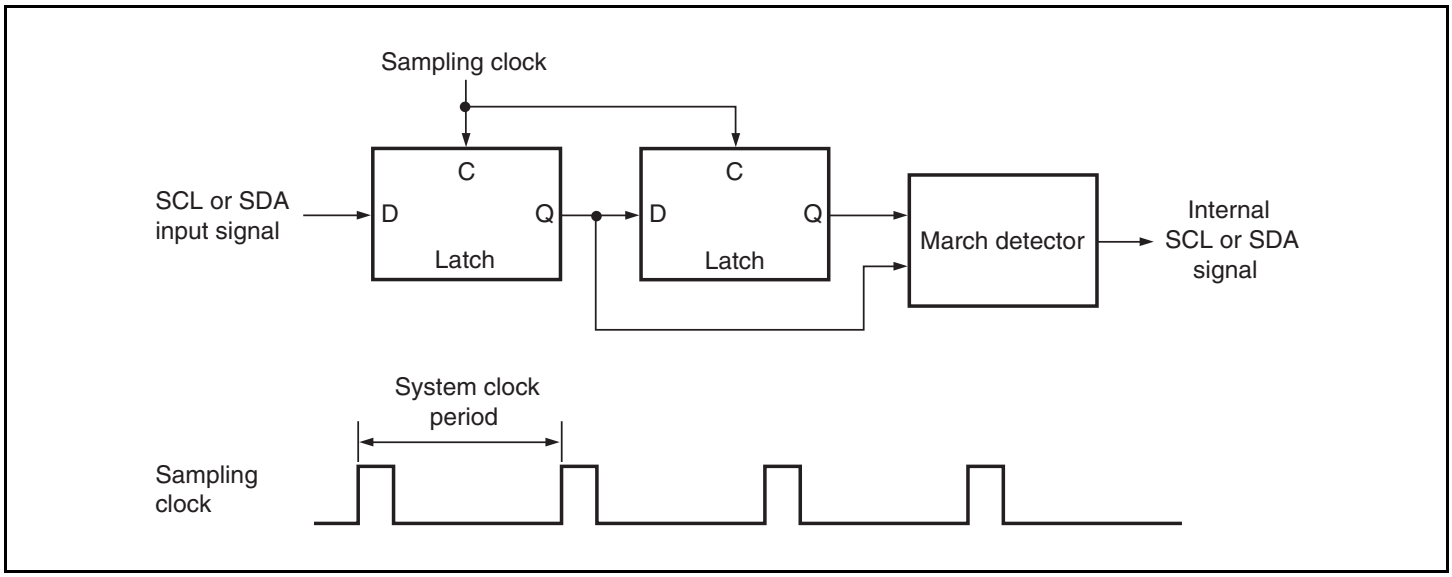


Figure 17.13 Block Diagram of Noise Canceler

17.4.7 Example of Use

Flowcharts in respective modes that use the I²C bus interface are shown in figures 17.14 to 17.17.

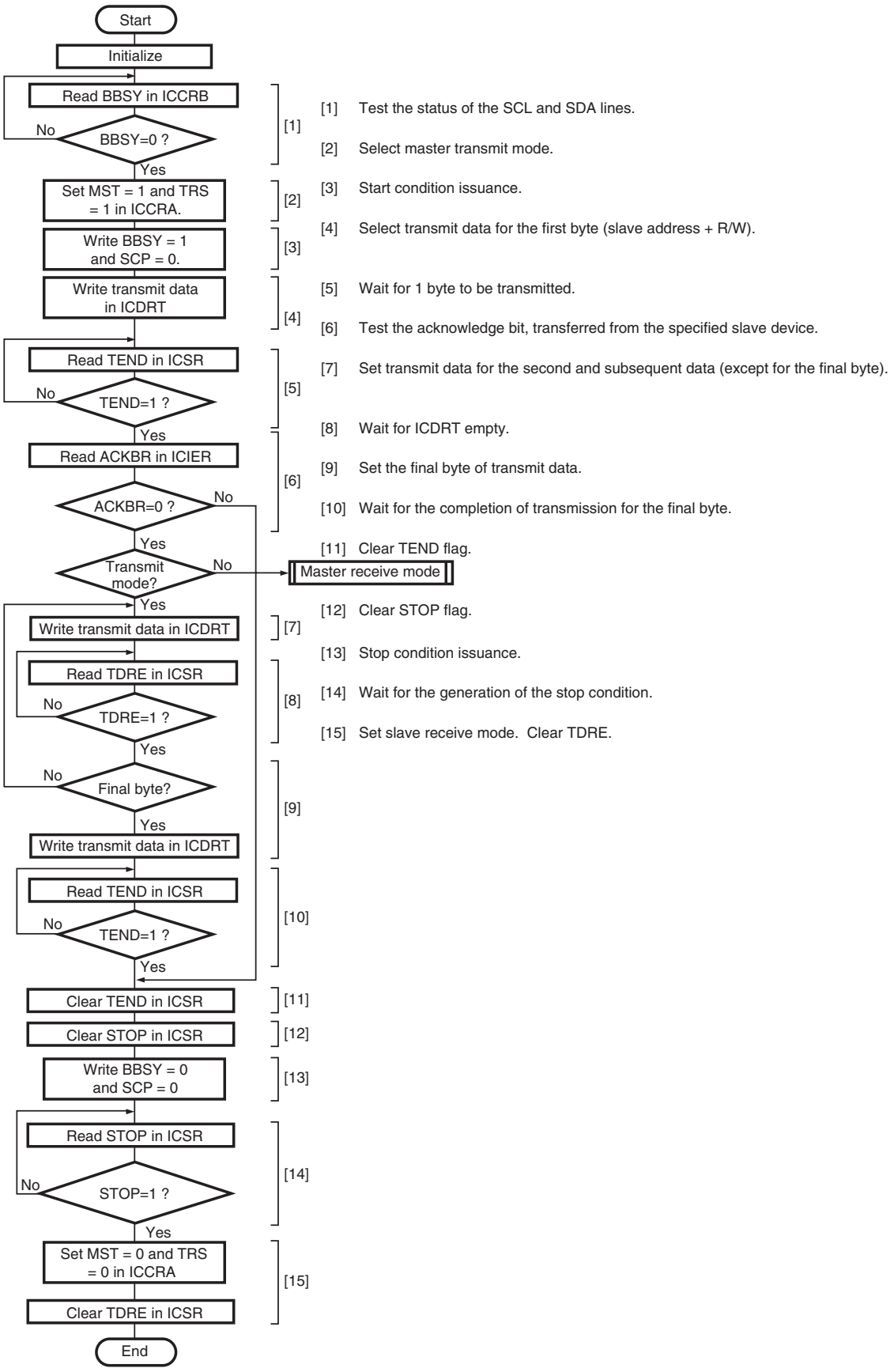
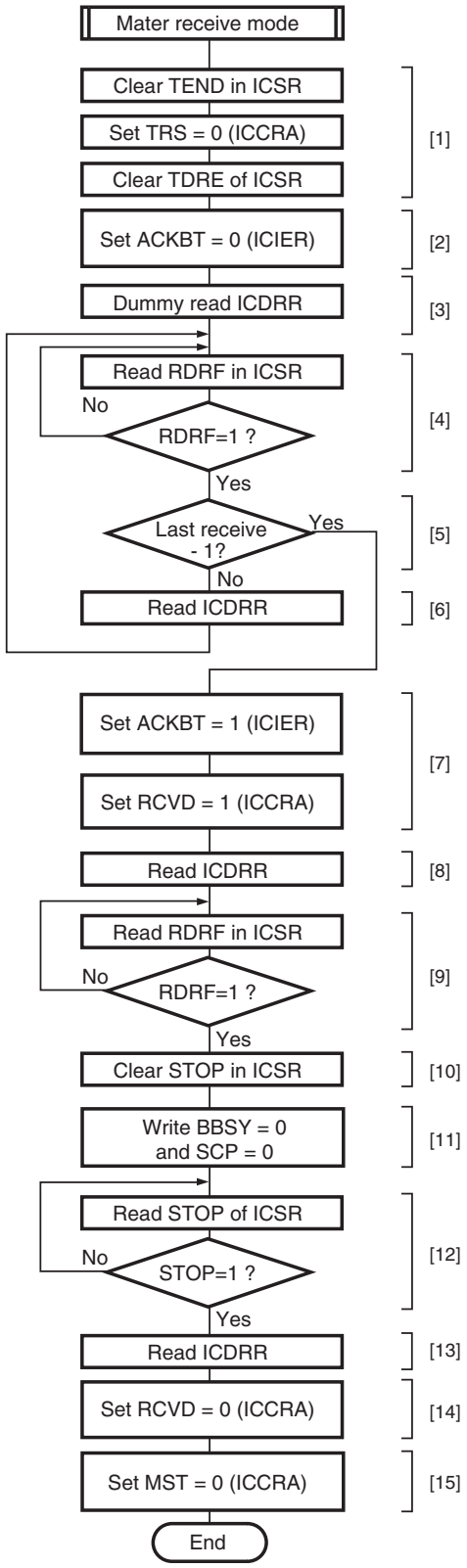


Figure 17.14 Sample Flowchart for Master Transmit Mode



[1] Clear TEND, select master receive mode, and then clear TDRE.*

[2] Set acknowledge to the transmitting device.*

[3] Dummy read ICDDR.*

[4] Wait for 1 byte to be received.

[5] Check if the (last receive - 1).

[6] Read the receive data.

[7] Set acknowledge of the final byte. Disable continuous receive (RCVD = 1).

[8] Read receive data of (final byte - 1).

[9] Wait for the final byte to be received.

[10] Clear STOP flag.

[11] Stop condition issuance.

[12] Wait for the generation of stop condition.

[13] Read the receive data of the final byte.

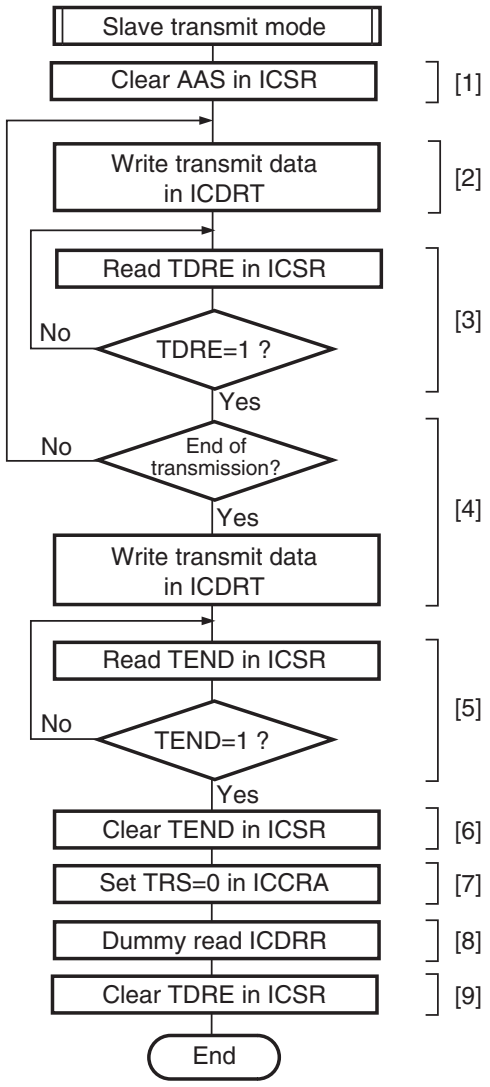
[14] Clear RCVD to 0.

[15] Set slave receive mode.

Note: * Ensure that no interrupts are received while steps [1] through [3] are being processed.

Additional information: If only one byte is received, steps [2] through [6] are omitted following step [1], and processing jumps to step [7]. Step [8] is ICDDR dummy read.

Figure 17.15 Sample Flowchart for Master Receive Mode



[1] Clear the flag AAS.

[2] Set transmit data for ICDRT (except for the last data).

[3] Wait for ICDRT empty.

[4] Set the last byte of the transmit data.

[5] Wait the transmission end of the last byte.

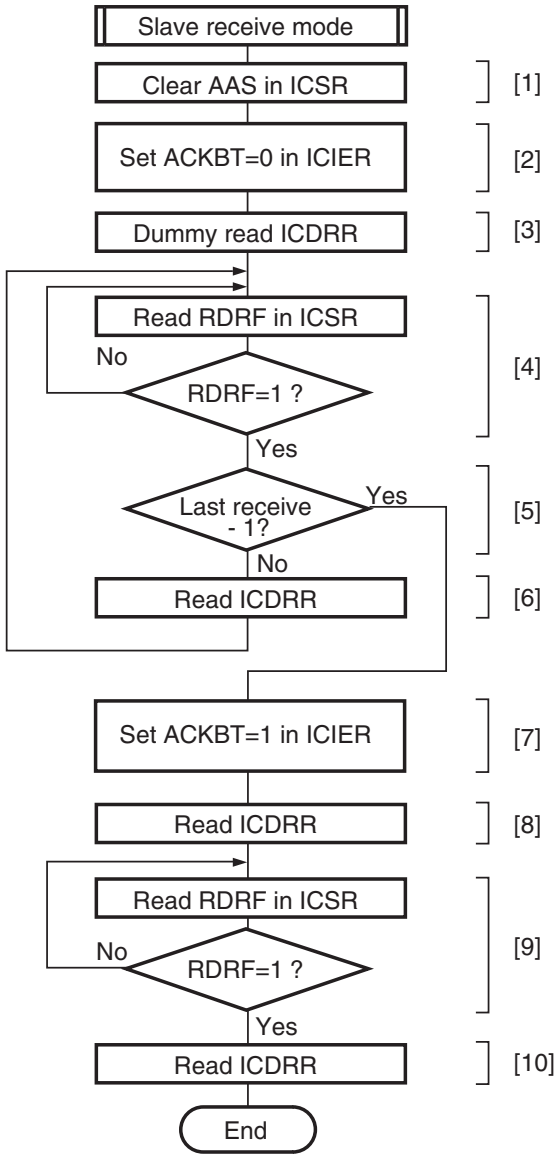
[6] Clear the flag TEND.

[7] Set slave receive mode.

[8] Dummy read ICDRR to release the SCL line.

[9] Clear the flag TDRE.

Figure 17.16 Sample Flowchart for Slave Transmit Mode



[1] Clear the flag AAS.

[2] Set the acknowledge for the transmit device.

[3] Dummy read ICDRR.

[4] Wait the reception end of 1 byte.

[5] Check if the (last receive - 1).

[6] Read the received data.

[7] Set the acknowledge for the last byte.

[8] Read the received data of the (last byte - 1).

[9] Wait the reception end of the last byte.

[10] Read the received data of the last byte.

Additional information: If only one byte is received, steps [2] through [6] are omitted following step [1], and processing jumps to step [7]. Step [8] is ICDRR dummy read.

Figure 17.17 Sample Flowchart for Slave Receive Mode

17.5 Interrupt Request

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK detection, STOP recognition, and arbitration lost. Table 17.3 shows the contents of each interrupt request.

Table 17.3 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition
Transmit Data Empty	TXI	(TDRE=1) • (TIE=1)
Transmit End	TEI	(TEND=1) • (TEIE=1)
Receive Data Full	RXI	(RDRF=1) • (RIE=1)
STOP Recognition	STPI	(STOP=1) • (STIE=1)
NACK Detection	NAKI	{(NACKF=1)+(AL=1)} • (NAKIE=1)
Arbitration Lost		

Interrupt exception handling is performed when the interrupt conditions listed in table 17.3 are set to 1 and the CPU is ready to accept interrupts. During exception handling, the interrupt sources should be cleared. Note, however, that TDRE and TEND are automatically cleared by writing transmit data to ICDRT, and RDRF is automatically cleared by reading data from ICDRR. In particular, if TDRE is set at the same time transmit data is written to ICDRT, and then TDRE is cleared again, an extra byte of data may be transmitted.

17.6 Bit Synchronous Circuit

In master mode,

- When SCL is driven to low by the slave device
- When the rising speed of SCL is lower by the load of the SCL line (load capacitance or pull-up resistance)

This module has a possibility that high level period may be short in the two states described above. Therefore it monitors SCL and communicates by bit with synchronization. Figure 17.18 shows the timing of the bit synchronous circuit and table 17.4 shows the time when SCL output changes from low to Hi-Z then SCL is monitored.

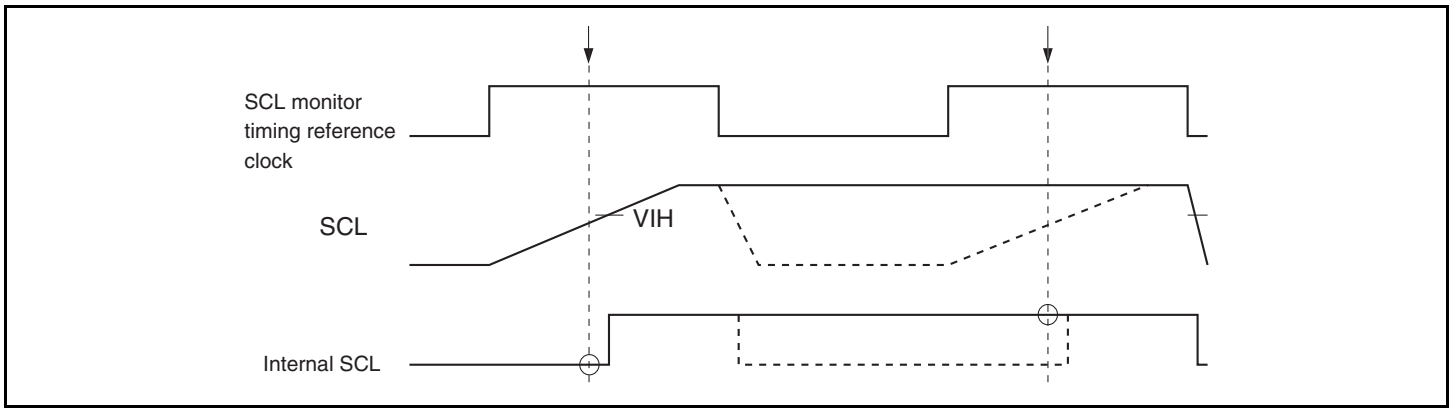


Figure 17.18 Timing of the Bit Synchronous Circuit

Table 17.4 Time for monitoring SCL

CKS3	CKS2	Time for monitoring SCL
0	0	7.5 tcyc
	1	19.5 tcyc
1	0	17.5 tcyc
	1	41.5 tcyc

17.7 Usage Notes

1. Issue (retransmit) the start/stop conditions after the fall of the ninth clock is confirmed.

   Check SCLO in the I²C control register B (IICRB) to confirm the fall of the ninth clock.

   When the start/stop conditions are issued (retransmitted) at the specific timing under the following condition (i) or (ii), such conditions may not be output successfully. This does not occur in other cases.

   (i) When the rising of SCL falls behind the time specified in section 17.6, Bit Synchronous Circuit, by the load of the SCL bus (load capacitance or pull-up resistance)

   (ii) When the bit synchronous circuit is activated by extending the low period of eighth and ninth clocks, that is driven by the slave device

2. Control WAIT in the I²C bus mode register (ICMR) to be set to 0.

   When WAIT is set to 1, and SCL is driven low for two or more transfer clocks by the slave device at the eighth and ninth clocks, the high period of ninth clock may be shortened. This does not occur in other cases.

Section 18 A/D Converter

This LSI includes two units (units 0 and 1) of successive approximation type 10-bit A/D converter. In the H8S/2456 group and H8S/2456R group, the A/D converter units 0 and 1 allow up to eight analog input channels to be selected. In the H8S/2454 group, unit 0 allows up to eight analog input channels to be selected while unit 1 allows up to two channels.

Figures 18.1 and 18.2 show block diagrams of the A/D converter units 0 and 1, respectively.

18.1 Features

- 10-bit resolution
- Input channels:

  H8S/2456 group and H8S/2456R group: Eight channels (total of 16 channels for the two units)

  H8S/2454 group: Eight channels for unit 0 and two channels for unit 1 (total of 10 channels for the two units)
- Conversion cycle: 64 cycles or 40 cycles (A/D conversion clock)
- 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]
- Separate A/D conversion clock specifiable for each unit (Pϕ, Pϕ/2, Pϕ/4, or Pϕ/8)
- Eight data registers for A/D converter unit 0 and eight data registers for unit 1*[2] (total of 16 data registers for the two 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 pulse unit (TPU) or 8-bit timer (TMR), or an external trigger signal.
- Interrupt source

  A/D conversion end interrupt (ADI) request can be generated.
- Module stop state specifiable

Notes: 1. Continuous A/D conversion on 1 to 2 channels in the H8S/2454 group.
2. Two data registers for unit 1 (total of ten data registers for the two units) in the H8S/2454 group.

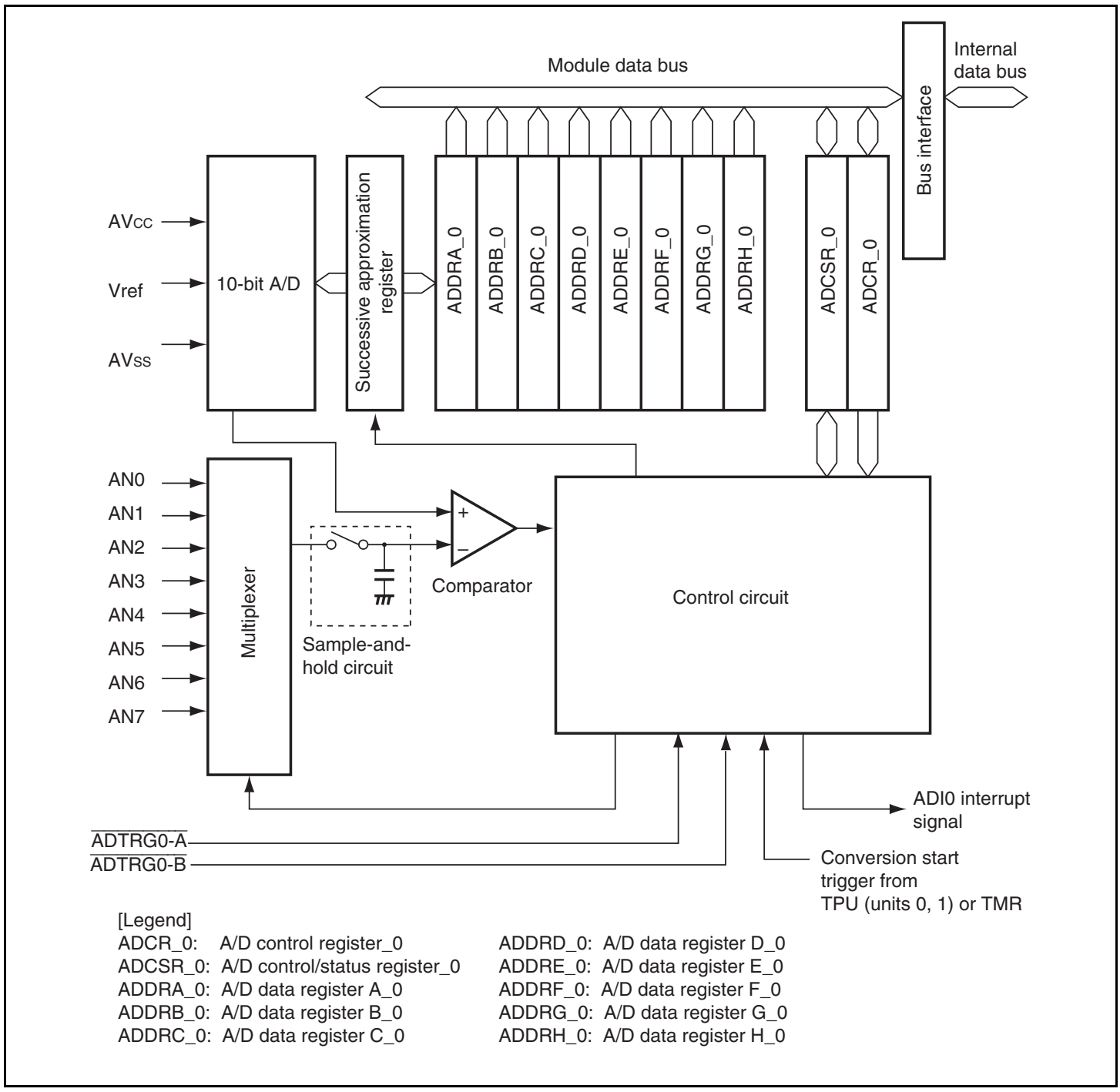


Figure 18.1 Block Diagram of A/D Converter Unit 0 (AD_0)

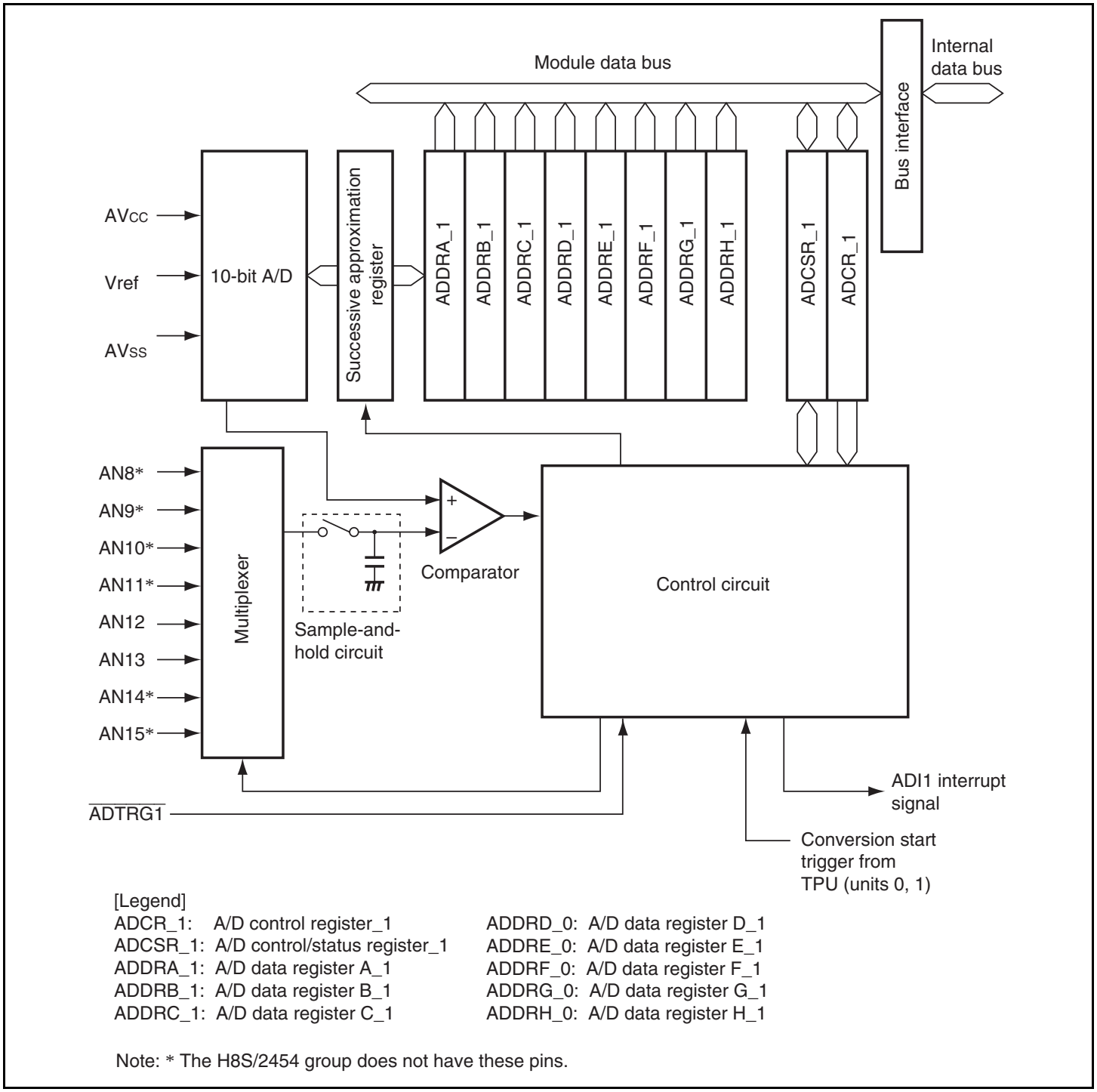


Figure 18.2 Block Diagram of A/D Converter Unit 1 (AD_1)

18.2 Input/Output Pins

Tables 18.1 and 18.2 show the pin configuration of the A/D converter.

Table 18.1 Pin Configuration (H8S/2456 Group and H8S/2456R Group)

Unit	Symbol	Pin Name	Symbol	I/O	Function
0	AD_0	Analog input pin 0	AN0	Input	Analog inputs
		Analog input pin 1	AN1	Input	
		Analog input pin 2	AN2	Input	
		Analog input pin 3	AN3	Input	
		Analog input pin 4	AN4	Input	
		Analog input pin 5	AN5	Input	
		Analog input pin 6	AN6	Input	
		Analog input pin 7	AN7	Input	
		A/D external trigger input pin 0_A	$\overline{\text{ADTRG0}}$-A	Input	External trigger input pin 0_A for starting A/D conversion*
		A/D external trigger input pin 0_B	$\overline{\text{ADTRG0}}$-B	Input	External trigger input pin 0_B for starting A/D conversion*
1	AD_1	Analog input pin 8	AN8	Input	Analog inputs
		Analog input pin 9	AN9	Input	
		Analog input pin 10	AN10	Input	
		Analog input pin 11	AN11	Input	
		Analog input pin 12	AN12	Input	
		Analog input pin 13	AN13	Input	
		Analog input pin 14	AN14	Input	
		Analog input pin 15	AN15	Input	
		A/D external trigger input pin 1	$\overline{\text{ADTRG1}}$	Input	External trigger input pin A for starting A/D conversion
Common		Analog power supply pin	AV_{CC}	Input	Analog block power supply
		Analog ground pin	AV_{SS}	Input	Analog block ground
		Reference voltage pin	Vref	Input	A/D conversion reference voltage

Note: * Selectable by setting of the TRGS1, TRGS0, and EXTRGS bits in ADCR.

Table 18.2 Pin Configuration (H8S/2454 Group)

Unit	Abbr.	Pin Name	Symbol	I/O	Function
0	AD_0	Analog input pin 0	AN0	Input	Analog inputs
		Analog input pin 1	AN1	Input	
		Analog input pin 2	AN2	Input	
		Analog input pin 3	AN3	Input	
		Analog input pin 4	AN4	Input	
		Analog input pin 5	AN5	Input	
		Analog input pin 6	AN6	Input	
		Analog input pin 7	AN7	Input	
		A/D external trigger input pin 0_A	$\overline{\text{ADTRG0}}$-A	Input	External trigger input pin 0_A for starting A/D conversion*
		A/D external trigger input pin 0_B	$\overline{\text{ADTRG0}}$-B	Input	External trigger input pin 0_B for starting A/D conversion*
1	AD_1	Analog input pin 12	AN12	Input	Analog inputs
		Analog input pin 13	AN13	Input	
		A/D external trigger input pin 1	$\overline{\text{ADTRG1}}$	Input	External trigger input pin A for starting A/D conversion
Common		Analog power supply pin	AV_{CC}	Input	Analog block power supply
		Analog ground pin	AV_{SS}	Input	Analog block ground
		Reference voltage pin	Vref	Input	A/D conversion reference voltage

Note: * Selectable by setting of the TRGS1, TRGS0, and EXTRGS bits in ADCR.

18.3 Register Descriptions

The A/D converter has the following registers.

Unit 0 (A/D_0) registers:

- A/D data register A_0 (ADDRA_0)
- A/D data register B_0 (ADDRB_0)
- A/D data register C_0 (ADDRC_0)
- A/D data register D_0 (ADDRD_0)
- 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)

18.3.1 A/D Data Registers A to H (ADDRA to ADDRH)

There are eight 16-bit read-only ADDR registers, ADDRA to ADDRH, used to store the results of A/D conversion. The ADDR registers, which store a conversion result for each channel, are shown in tables 18.3 and 18.4.

The converted 10-bit data is stored in bits 15 to 6. The lower 6-bit data is always read as 0.

The data bus between the CPU and the A/D converter has a 16-bit width. The data can be read directly from the CPU. ADDR must not be accessed in 8-bit units and must be accessed in 16-bit units.

Table 18.3 Analog Input Channels and Corresponding ADDR Registers (H8S/2456 Group and H8S/2456R Group)

Analog Input Channel		Analog Input Channel	
Channel Set 0 (CH3 = 0)	Data Register Storing Conversion Result	Channel Set 1 (CH3 = 1)	Data Register Storing Conversion Result
AN0	ADDRA_0	AN8	ADDRA_1
AN1	ADDRB_0	AN9	ADDRB_1
AN2	ADDRC_0	AN10	ADDRC_1
AN3	ADDRD_0	AN11	ADDRD_1
AN4	ADDRE_0	AN12	ADDRE_1
AN5	ADDRF_0	AN13	ADDRF_1
AN6	ADDRG_0	AN14	ADDRG_1
AN7	ADDRH_0	AN15	ADDRH_1

Table 18.4 Analog Input Channels and Corresponding ADDR Registers (H8S/2454 Group)

Analog Input Channel		Analog Input Channel	
Channel Set 0 (CH3 = 0)	Data Register Storing Conversion Result	Channel Set 1 (CH3 = 1)	Data Register Storing Conversion Result
AN0	ADDRA_0	—	—
AN1	ADDRB_0	—	—
AN2	ADDRC_0	—	—
AN3	ADDRD_0	—	—
AN4	ADDRE_0	AN12	ADDRE_1
AN5	ADDRF_0	AN13	ADDRF_1
AN6	ADDRG_0	—	—
AN7	ADDRH_0	—	—

18.3.2 A/D Control/Status Register for Unit 0 (ADCSR_0)

ADCSR_0 controls A/D conversion operations.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conversion. [Setting conditions] • Completion of A/D conversion in single mode • Completion of A/D conversion on all specified channels in scan mode [Clearing conditions] • Writing of 0 after reading ADF = 1 • Reading from ADDR after activation of the DMAC or DTC by an ADI interrupt
6	ADIE	0	R/W	A/D Interrupt Enable Setting this bit to 1 enables ADI interrupts by ADF.
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion, and the A/D converter enters wait state. Setting this bit to 1 starts 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, a reset, or hardware standby mode. While the ADSTCLR bit in ADCR is set to 1, the ADST bit is cleared to 0 automatically when A/D conversion on all selected channels ends, and then A/D conversion stops. The timing to clear the ADST bit automatically differs from that of ADF setting; the ADST bit is cleared before the ADF bit is set.
4	EXCKS	0	R/W	Clock Extension Select Specifies the A/D conversion time in combination with the CKS1 and CKS0 bits in ADCR. Be sure to set these three bits at one time. For details, see the description of the ADCR resisters.

Bit	Bit Name	Initial Value	R/W	Description
3	CH3	0	R/W	Channel Select 3 to 0
2	CH2	0	R/W	Selects analog input together with bits SCANE and SCANS in ADCR.
1	CH1	0	R/W	
0	CH0	0	R/W	• When SCANE = 0 and SCANS = x 0000: AN0 0001: AN1 0010: AN2 0011: AN3 0100: AN4 0101: AN5 0110: AN6 0111: AN7 1xxx: Setting prohibited • When SCANE = 1 and SCANS = 0 0000: AN0 0001: AN0 and AN1 0010: AN0 to AN2 0011: AN0 to AN3 0100: AN4 0101: AN4 and AN5 0110: AN4 to AN6 0111: AN4 to AN7 1xxx: Setting prohibited • When SCANE = 1 and SCANS = 1 0000: AN0 0001: AN0 and AN1 0010: AN0 to AN2 0011: AN0 to AN3 0100: AN0 to AN4 0101: AN0 to AN5 0110: AN0 to AN6 0111: AN0 to AN7 1xxx: Setting prohibited

[Legend]

x: Don't care

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

18.3.3 A/D Control/Status Register for Unit 1 (ADCSR_1)

ADCSR_1 controls A/D conversion operations.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conversion. [Setting conditions] • Completion of A/D conversion in single mode • Completion of A/D conversion on all specified channels in scan mode [Clearing conditions] • Writing of 0 after reading ADF = 1 • Reading from ADDR after activation of the DTC by an ADI interrupt
6	ADIE	0	R/W	A/D Interrupt Enable Setting this bit to 1 enables ADI interrupts by ADF.
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion, and the A/D converter enters wait state. Setting this bit to 1 starts 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, a reset, or hardware standby mode. While the ADSTCLR bit in ADCR is set to 1, the ADST bit is cleared to 0 automatically when A/D conversion on all selected channels ends, and then A/D conversion stops. The timing to clear the ADST bit automatically differs from that of ADF setting; the ADST bit is cleared before the ADF bit is set.
4	EXCKS	0	R/W	Clock Extension Select Specifies the A/D conversion time in combination with the CKS1 and CKS0 bits in ADCR. Be sure to set these three bits at one time. For details, see the description of the ADCR resisters.

- H8S/2456 Group and H8S/2456R Group

Bit	Bit Name	Initial Value	R/W	Description
3	CH3	0	R/W	Channel Select 3 to 0
2	CH2	0	R/W	Selects analog input together with bits SCANE and SCANS in ADCR.
1	CH1	0	R/W	
0	CH0	0	R/W	• When SCANE = 0 and SCANS = x 0XXX: Setting prohibited 1000: AN8 1001: AN9 1010: AN10 1011: AN11 1100: AN12 1101: AN13 1110: AN14 1111: AN15 • When SCANE = 1 and SCANS = 0 0XXX: Setting prohibited 1000: AN8 1001: AN8 and AN9 1010: AN8 to AN10 1011: AN8 to AN11 1100: AN12 1101: AN12 and AN13 1110: AN12 to AN14 1111: AN12 to AN15 • When SCANE = 1 and SCANS = 1 0XXX: Setting prohibited 1000: AN8 1001: AN8 and AN9 1010: AN8 to AN10 1011: AN8 to AN11 1100: AN8 to AN12 1101: AN8 to AN13 1110: AN8 to AN14 1111: AN8 to AN15

[Legend]

x: Don't care

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

- H8S/2454 Group

Bit	Bit Name	Initial Value	R/W	Description
3	CH3	0	R/W	Channel Select 3 to 0
2	CH2	0	R/W	Selects analog input together with bits SCANE and SCANS in ADCR.
1	CH1	0	R/W	
0	CH0	0	R/W	• When SCANE = 0 and SCANS = x 0XXX: Setting prohibited 10XX: Setting prohibited 1100: AN12 1101: AN13 111X: Setting prohibited • When SCANE = 1 and SCANS = 0 0XXX: Setting prohibited 10XX: Setting prohibited 1100: AN12 1101: AN12 and AN13 111X: Setting prohibited • Setting SCANE = 1 and SCANS = 1 are prohibited.

[Legend]

x: Don't care

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

18.3.4 A/D Control Register (ADCR_0) Unit 0

ADCR enables A/D conversion to be started by an external trigger input.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0 and Extended Trigger Select
6	TRGS0	0	R/W	These bits enable or disable the start of A/D conversion by a trigger signal.
0	EXTRGS	0	R/W	000: Disables A/D conversion start by external trigger 010: Enables A/D conversion start by external trigger from TPU (unit 0) 100: Enables A/D conversion start by external trigger from TMR 110: Enables A/D conversion start by the $\overline{\text{ADTRG0-A}}$ pin 001: Enables A/D conversion start by the $\overline{\text{ADTRG0-B}}$ pin 011: Enables simultaneous A/D conversion start in multiple units by external trigger from TPU (units 0 and 1) 101: Enables simultaneous A/D conversion start in multiple units by external trigger from TMR 111: Enables simultaneous A/D conversion start in multiple units by the $\overline{\text{ADTRG0-B}}$ pin

Bit	Bit Name	Initial Value	R/W	Description
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 for channels 1 to 4. 11: Scan mode. A/D conversion is performed continuously for channels 1 to 8.
3	CKS1	0	R/W	Clock Select 1 and 0
2	CKS0	0	R/W	These bits select the A/D conversion clock (ADCLK) and specify the A/D conversion time in combination with the EXCKS bit. First select the A/D conversion time while ADST = 0 in ADCSR and then set the mode of A/D conversion. Before entering software standby mode or module stop mode, set these bits to B'11. Set CKS1 and CKS0 bits appropriately so that the ADCLK frequency is 10 MHz or less. EXCKS, CKS1, and CKS0 000: Setting prohibited 001: A/D conversion time = 268 states (max.) at ADCLK = ϕ/4 010: A/D conversion time = 138 states (max.) at ADCLK = ϕ/2 011: A/D conversion time = 73 states (max.) at ADCLK = ϕ 100: Setting prohibited 101: A/D conversion time = 172 states (max.) at ADCLK = ϕ/4 110: A/D conversion time = 90 states (max.) at ADCLK = ϕ/2 111: A/D conversion time = 49 states (max.) at ADCLK = ϕ
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: The ADST bit is cleared to 0 upon completion of the A/D conversion for all of the selected channels in scan mode.

[Legend]

x: Don't care

18.3.5 A/D Control Register (ADCR_1) Unit 1

ADCR enables A/D conversion to be started by an external trigger input.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0 and Extended Trigger Select
6	TRGS0	0	R/W	These bits enable or disable the start of A/D conversion by a trigger signal.
0	EXTRGS	0	R/W	000: Disables A/D conversion start by external trigger 010: Enables A/D conversion start by external trigger from TPU (units 0 and 1) 100: Enables A/D conversion start by external trigger from TMR 110: Enables A/D conversion start by the $\overline{\text{ADTRG1}}$ pin 001: Setting prohibited 011: Enables simultaneous A/D conversion start in multiple units by external trigger from TPU (units 0 and 1) 101: Enables simultaneous A/D conversion start in multiple units by external trigger from TMR 111: Enables simultaneous A/D conversion start in multiple units by the $\overline{\text{ADTRG0-B}}$ pin
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 for channels 1 to 4. 11: Scan mode. A/D conversion is performed continuously for channels 1 to 8.*

Bit	Bit Name	Initial Value	R/W	Description
3	CKS1	0	R/W	Clock Select 1 and 0
2	CKS0	0	R/W	These bits select the A/D conversion clock (ADCLK) and specify the A/D conversion time in combination with the EXCKS bit. First select the A/D conversion time while ADST = 0 in ADCSR and then set the mode of A/D conversion. Before entering software standby mode or module stop mode, set these bits to B'11. Set CKS1 and CKS0 bits appropriately so that the ADCLK frequency is 10 MHz or less. EXCKS, CKS1, and CKS0 000: Setting prohibited 001: A/D conversion time = 268 states (max.) at ADCLK = ϕ/4 010: A/D conversion time = 138 states (max.) at ADCLK = ϕ/2 011: A/D conversion time = 73 states (max.) at ADCLK = ϕ 100: Setting prohibited 101: A/D conversion time = 172 states (max.) at ADCLK = ϕ/4 110: A/D conversion time = 90 states (max.) at ADCLK = ϕ/2 111: A/D conversion time = 49 states (max.) at ADCLK = ϕ
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: The ADST bit is cleared to 0 upon completion of the A/D conversion for all of the selected channels in scan mode.

[Legend]

x: Don't care

Note: * Setting prohibited in the H8S/2454 group.

18.4 Operation

The A/D converter has two operating modes: single mode and scan mode. First select the clock for A/D conversion (ADCLK). When changing the operating mode or analog input channel, to prevent incorrect operation, first clear the ADST bit in ADCSR to 0. The ADST bit can be set to 1 at the same time as the operating mode or analog input channel is changed.

18.4.1 Single Mode

In single mode, A/D conversion is to be performed only once on the analog input of the specified 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, TMR, 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 to 0 during A/D conversion, A/D conversion stops and the A/D converter enters a wait state.

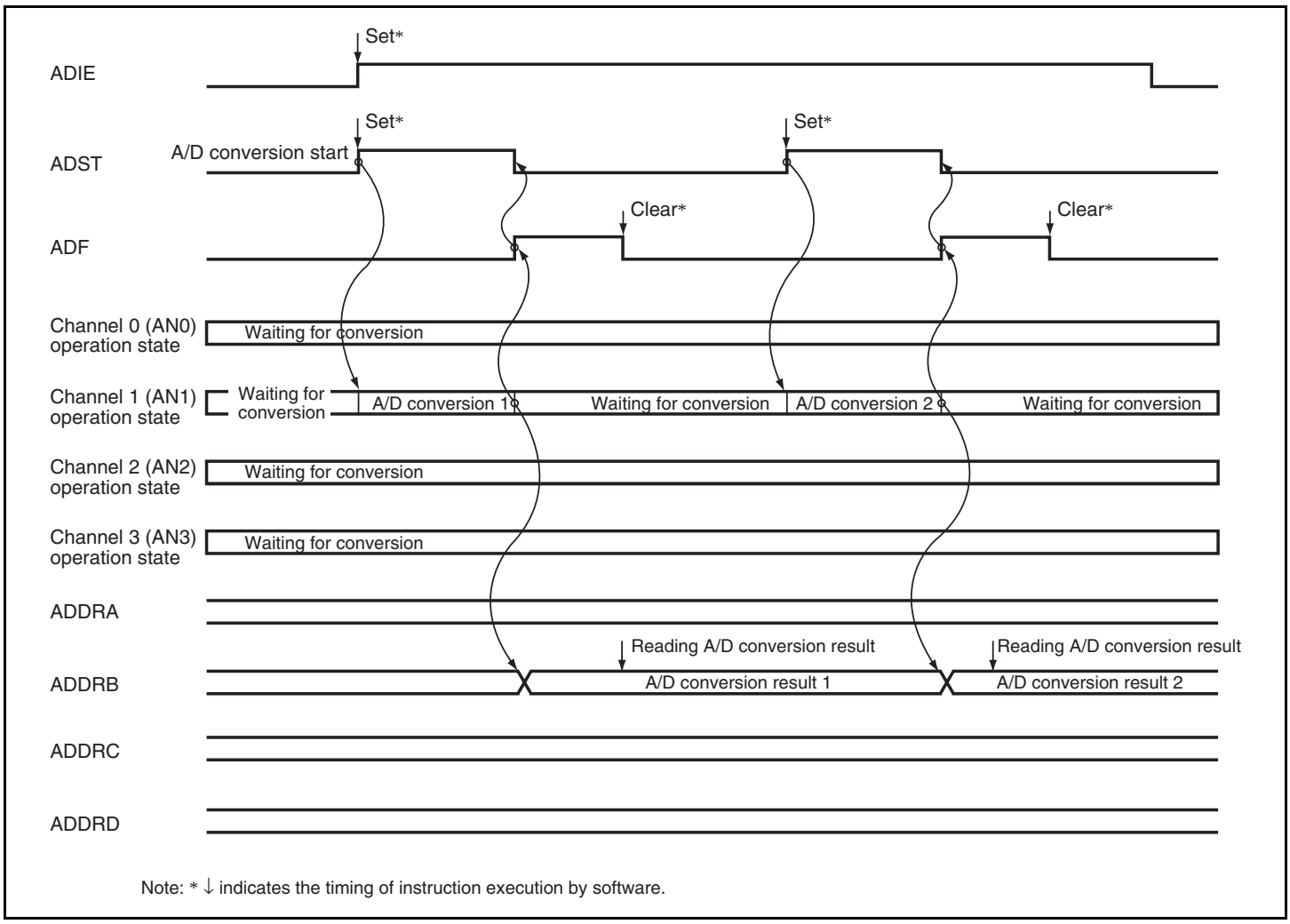


Figure 18.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

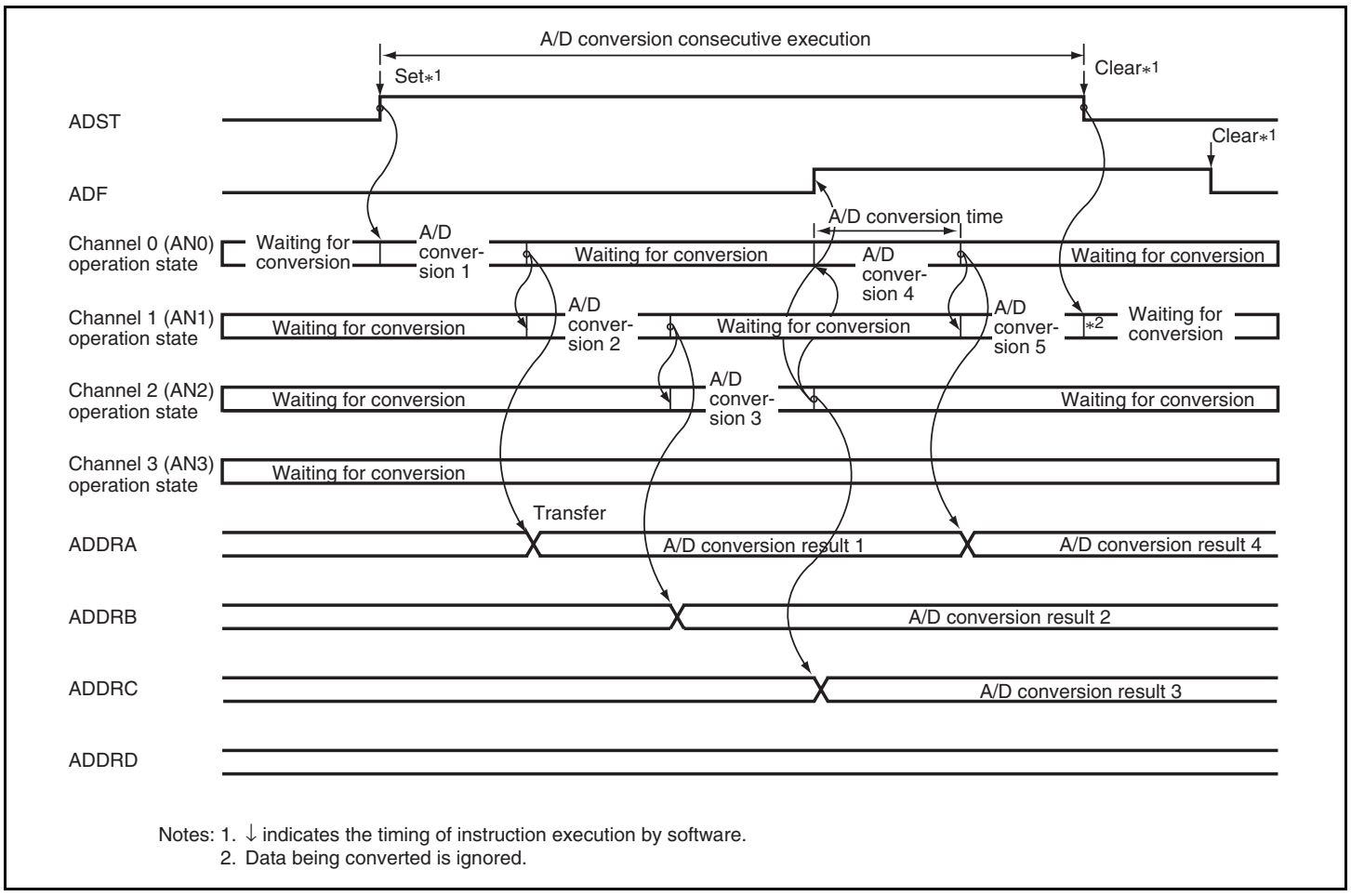
18.4.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the analog inputs of the specified channels up to four or eight* 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, TMR, or an external trigger input, A/D conversion starts on the first channel in the specified channel group. Consecutive A/D conversion on a maximum of four channels (SCANE and SCANS = B'10) or on a maximum of eight channels (SCANE and SCANS = B'11) can be selected. When consecutive 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 unit 1 = B'01, on AN8* when CH3 and CH2 of unit 1 = B'10, or on AN12 when CH3 and CH2 of unit 1= B'11. When consecutive A/D conversion is performed on eight channels, A/D conversion starts on AN0 when CH3 = B'0 or on AN8* when CH3 = B'1.
2. When A/D conversion for each channel is completed, the A/D conversion result is sequentially transferred to the corresponding ADDR of each channel.
3. When A/D conversion of all selected channels is completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. A/D conversion of the first channel in the group starts again.
4. The ADST bit is not cleared automatically, and steps 2 to 3 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter enters wait state. If the ADST bit is later set to 1, A/D conversion starts again from the first channel in the group.

Note: * Only possible in the H8S/2456 group and H8S/2456R group.



Notes: 1. ↓ indicates the timing of instruction execution by software.
2. Data being converted is ignored.

Figure 18.4 Example of A/D Conversion (Continuous Scan Mode, Three Channels (AN0 to AN2) Selected)

(2) One-Cycle Scan Mode

1. Set the ADSTCLR bit in ADCR to 1.
2. When the ADST bit in ADCSR is set to 1 by software, TPU, TMR, or an external trigger input, A/D conversion starts on the first channel in the specified channel group. Consecutive A/D conversion on a maximum of four channels (SCANE and SCANS = B'10) or on a maximum of eight channels (SCANE and SCANS = B'11) can be selected. Four-channel consecutive A/D conversion starts on AN0 when CH3 and CH2 = B'00 of unit 0, on AN4 when CH3 and CH2 = B'01, on AN8* when CH3 and CH2 of unit 1 = B'10, or on AN12 when CH3 and CH2 of unit 1= B'11. Eight-channel consecutive A/D conversion starts on AN0 when CH3 = B'0 or on AN8* when CH3 = B'1.
3. When A/D conversion for each channel is completed, the A/D conversion result is sequentially transferred to the corresponding ADDR of each channel.
4. When A/D conversion of all selected channels is completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
5. The ADST bit is automatically cleared when A/D conversion is completed for all of the channels that have been selected. A/D conversion stops and the A/D converter enters a wait state.

Note: * Only possible in the H8S/2456 group and H8S/2456R group.

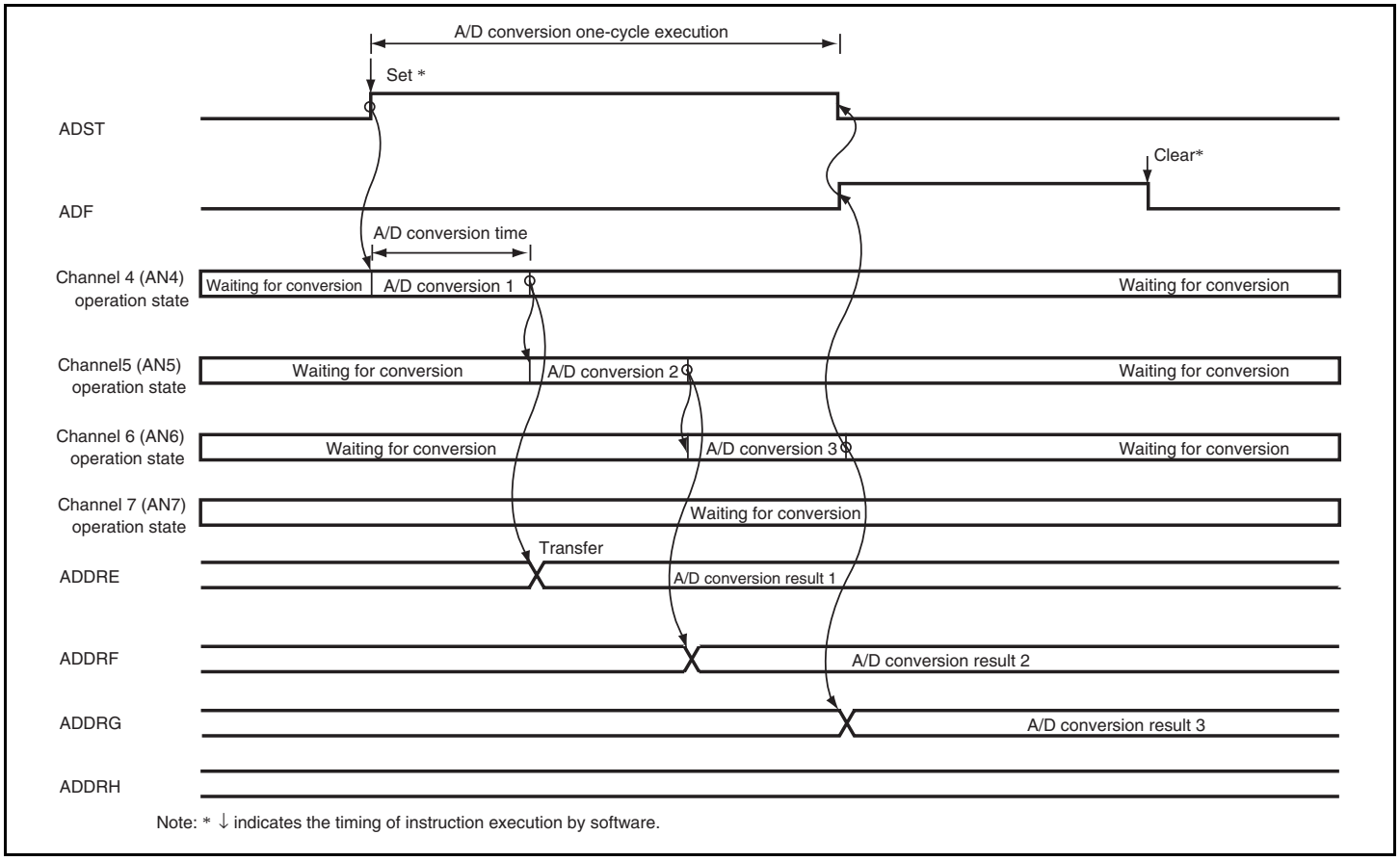


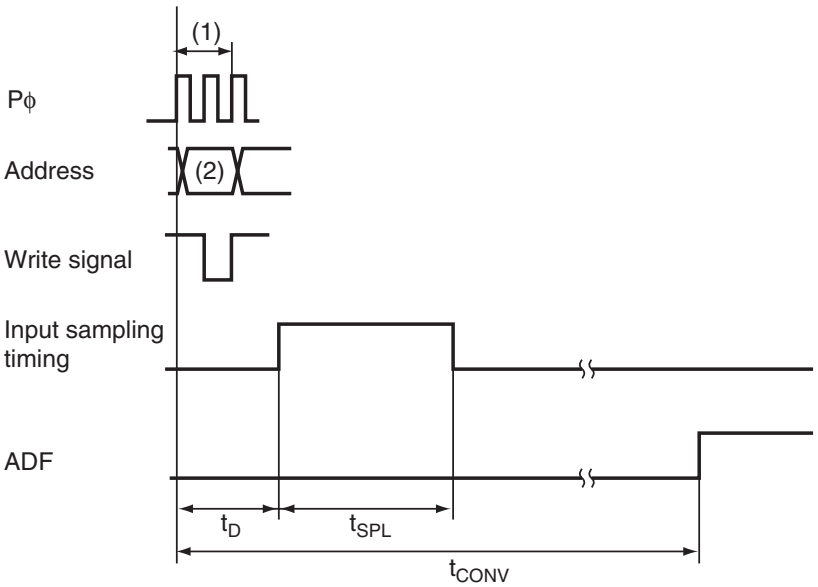
Figure 18.5 Example of A/D Conversion (One-Cycle Scan Mode, Three Channels (AN4 to AN6) Selected)

18.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input when the A/D conversion start delay time (t_D) passes after the ADST bit in ADCSR is set to 1, then starts A/D conversion. Figure 18.6 shows the A/D conversion timing. Tables 18.5 and 18.6 show the A/D conversion time.

As shown in figure 18.6, the A/D conversion time (t_{CONV}) includes the A/D conversion start delay time (t_D) and the input sampling time (t_{SPL}). The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in tables 18.5 and 18.6.

In scan mode, the values given in tables 18.5 and 18.6 apply to the first conversion time. The values given in table 18.7 apply to the second and subsequent conversions. In either case, bit EXCKS in ADCSR, and bits CKS1 and CKS0 in ADCR should be set so that the conversion time is within the ranges indicated by the A/D conversion characteristics.



[Legend]
(1): ADCSR write cycle
(2): ADCSR address
$t_{D:}$ A/D conversion start delay time
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 18.6 A/D Conversion Timing

Table 18.5 A/D Conversion Characteristics (EXCKS = 0)

		CKS1 = 0						CKS1 = 1					
		CKS = 0			CKS = 1			CKS = 0			CKS = 1		
Item	Symbol	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.
A/D conversion start delay time	t_D	4	—	14	4	—	10	4	—	8	3	—	7
Input sampling time	t_{SPL}	—	312	—	—	156	—	—	78	—	—	39	—
A/D conversion time	t_{CONV}	518	—	528	262	—	268	134	—	138	69	—	73

Note: Values in the table are the number of states.

Table 18.6 A/D Conversion Characteristics (EXCKS = 1)

		CKS1 = 0						CKS1 = 1					
		CKS = 0			CKS = 1			CKS = 0			CKS = 1		
Item	Symbol	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.
A/D conversion start delay time	t_D	4	—	14	4	—	10	4	—	8	3	—	7
Input sampling time	t_{SPL}	—	120	—	—	60	—	—	30	—	—	15	—
A/D conversion time	t_{CONV}	326	—	336	166	—	172	86	—	90	45	—	49

Note: Values in the table are the number of states.

Table 18.7 A/D Conversion Time (Scan Mode)

EXCKS	CKS1	CKS0	Conversion Time (Number of States)
0	0	0	512 (fixed)
		1	256 (fixed)
	1	0	128 (fixed)
		1	64 (fixed)
1	0	0	512 (fixed)
		1	256 (fixed)
	1	0	128 (fixed)
		1	64 (fixed)

18.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. 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 or B'001 in ADCR_0. For unit 1, an external trigger is input from the $\overline{\text{ADTRG1}}$ pin when the TRGS1, TRGS0, and EXTRGS bits are set to B'110 in ADCR_1. For multiple-unit simultaneous start, 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. A/D conversion starts when the ADST bit in ADCSR is set to 1 on the falling edge of the $\overline{\text{ADTRG0}}$ 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 18.7 shows the timing. Figure 18.8 shows the timing of multiple-unit simultaneous start.

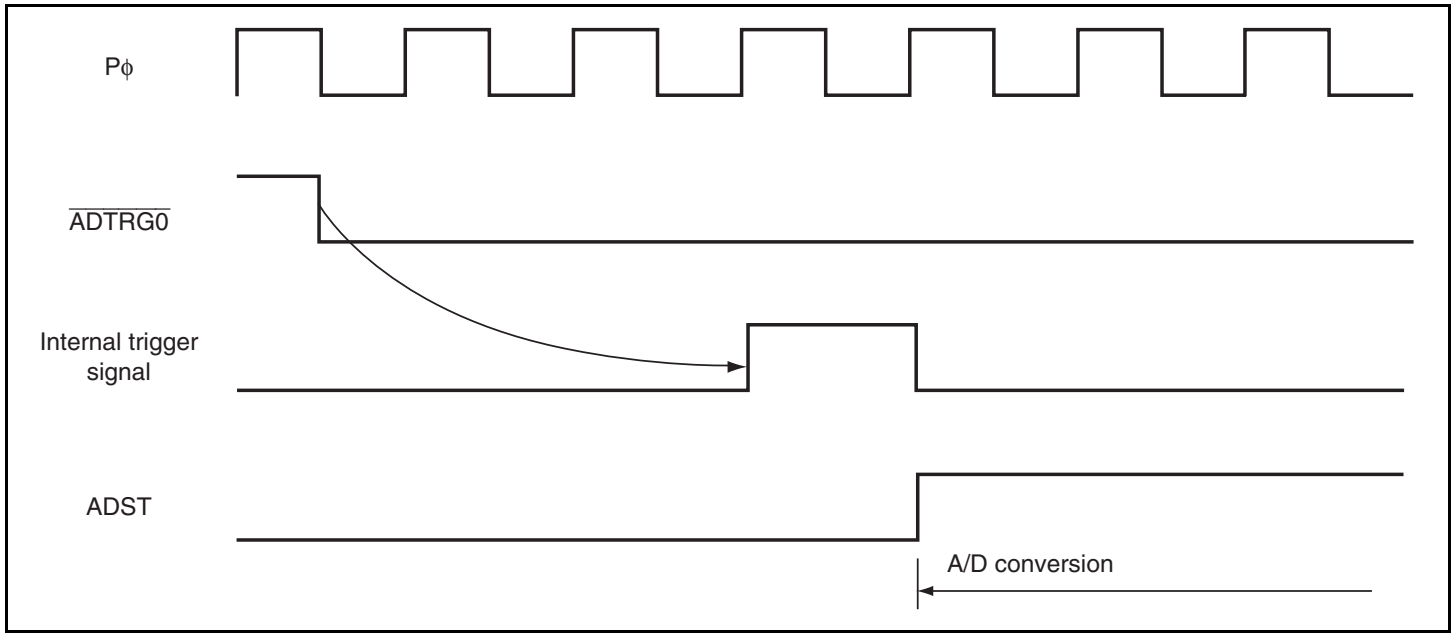


Figure 18.7 External Trigger Input Timing (TRGS1, TRGS0, and EXTRGS ≠ B'111)

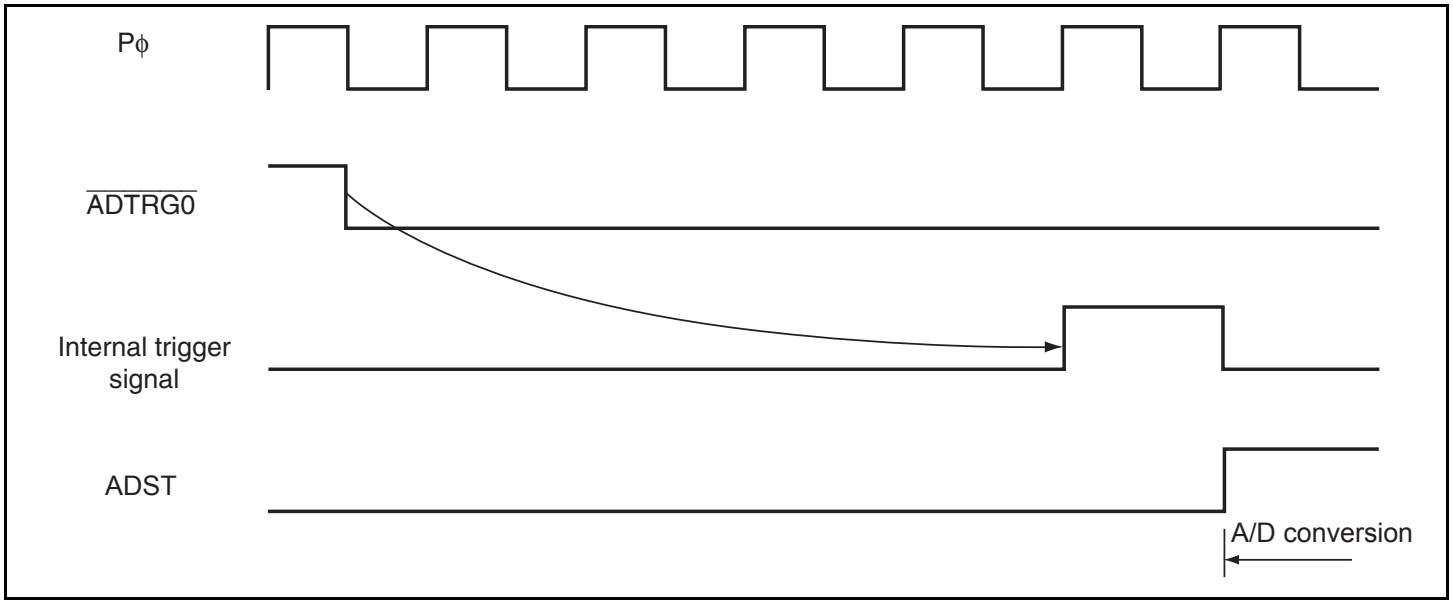


Figure 18.8 External Trigger Input Timing when Multiple Units Start Simultaneously (TRSG1, TRGS0, and EXTRGS = B'111)

18.5 Interrupt Source

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. Setting the ADIE bit to 1 when the ADF bit in ADCSR is set to 1 after A/D conversion is completed enables ADI interrupt requests. The data transfer controller (DTC) and DMA controller (DMAC) * can be activated by an ADI interrupt. Having the converted data read by the DTC or DMAC* in response to an ADI interrupt enables continuous conversion to be achieved without imposing a load on software.

Note: * Only possible in unit 0.

Table 18.8 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ADI0	A/D conversion end	ADF	Possible*	Possible

Note: * Only possible in unit 0.

18.6 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes.
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 18.9).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value B'0000000000 (H'000) to B'0000000001 (H'001) (see figure 18.10).
- 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 18.10).
- 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 18.10).
- 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.

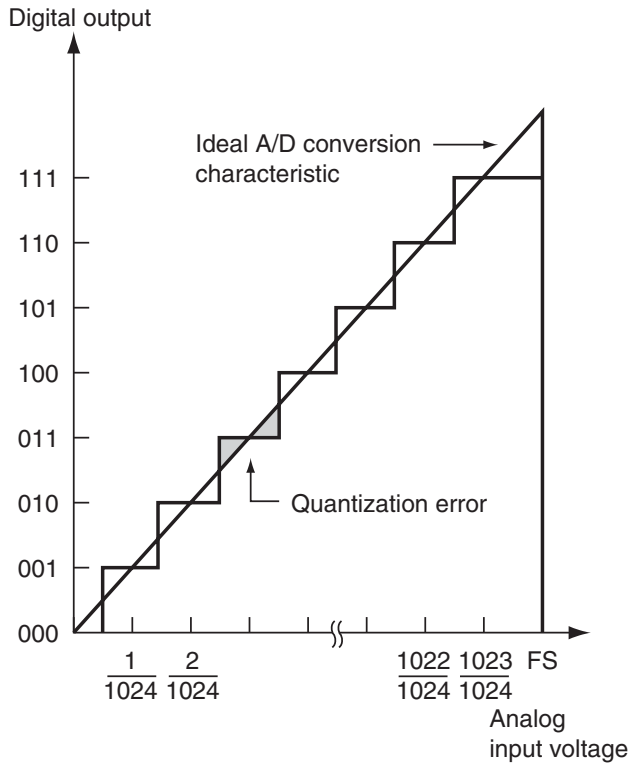


Figure 18.9 A/D Conversion Accuracy Definitions

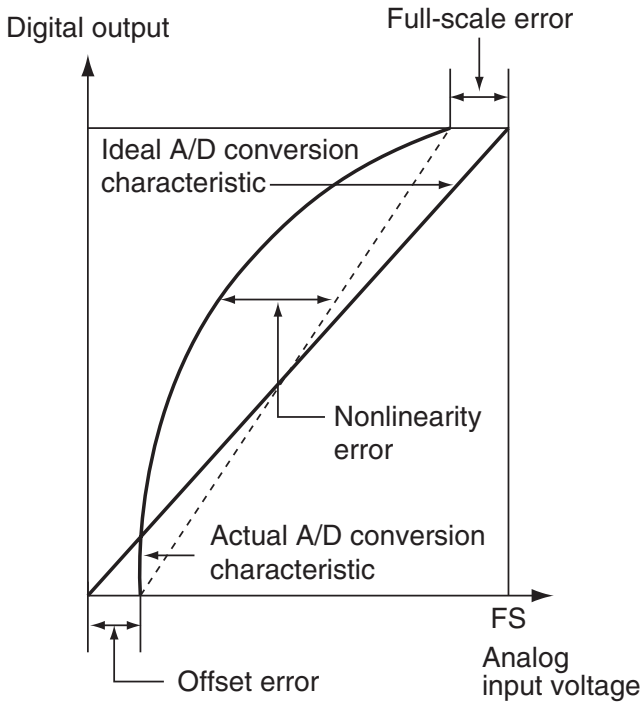


Figure 18.10 A/D Conversion Accuracy Definitions

18.7 Usage Notes

18.7.1 Module Stop Function Setting

Operation of the A/D converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the A/D converter to be halted. Register access is enabled by clearing the module stop state. Set the CKS1 and CKS2 bits to 1 to set ADCLK to ϕ, and clear the ADST, TRGS1, TRGS0, and EXTRGS bits all to 0 to disable A/D conversion when entering module stop state after operation of the A/D converter. After that, set the module stop control register after executing a dummy read by one word. For details, see section 25, Power-Down Modes.

18.7.2 A/D Input Hold Function in Software Standby Mode

When this LSI enters software standby mode with A/D conversion enabled, the analog inputs are retained, and the analog power supply current is equal to as during A/D conversion. If the analog power supply current needs to be reduced in software standby mode, set the CKS1 and CKS2 bits to 1 to set ADCLK to ϕ, 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.

18.7.3 Restarting the A/D Converter

When the ADST bit has been cleared to 0, A/D converter stops in synchronization with the ADCLK and then enters the standby sate. After the ADST bit has been cleared, the converter may not actually make the transition to the standby state for up to 10 cycles (ϕ), so do not change the channels of the ADCLK, motion mode, or analog input at this time.

When restarting the A/D converter right after the ADST bit has been cleared to 0, read the 16 bytes from ADDRA to ADDRH and then start the A/D converter by setting the ADST bit to 1. If the converter is in single mode or one-cycle scan mode, however, the ADST bit can be set to 1 by clearing the ADF bit to 0 after confirming that the ADF bit had been set to 1 on completion of the previous round of conversion.

18.7.4 Permissible Signal Source Impedance

This LSI's analog input is designed so that the conversion accuracy is guaranteed for an input signal for which the signal source impedance is 5 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficient and it may not be possible to guarantee the A/D conversion accuracy. However, if a large capacitance is provided externally for conversion in single mode, the input load will essentially comprise only the internal input resistance of 5 kΩ, and the signal source impedance is ignored. However, since a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/μs or greater) (see figure 18.11). When converting a high-speed analog signal or conversion in scan mode, a low-impedance buffer should be inserted.

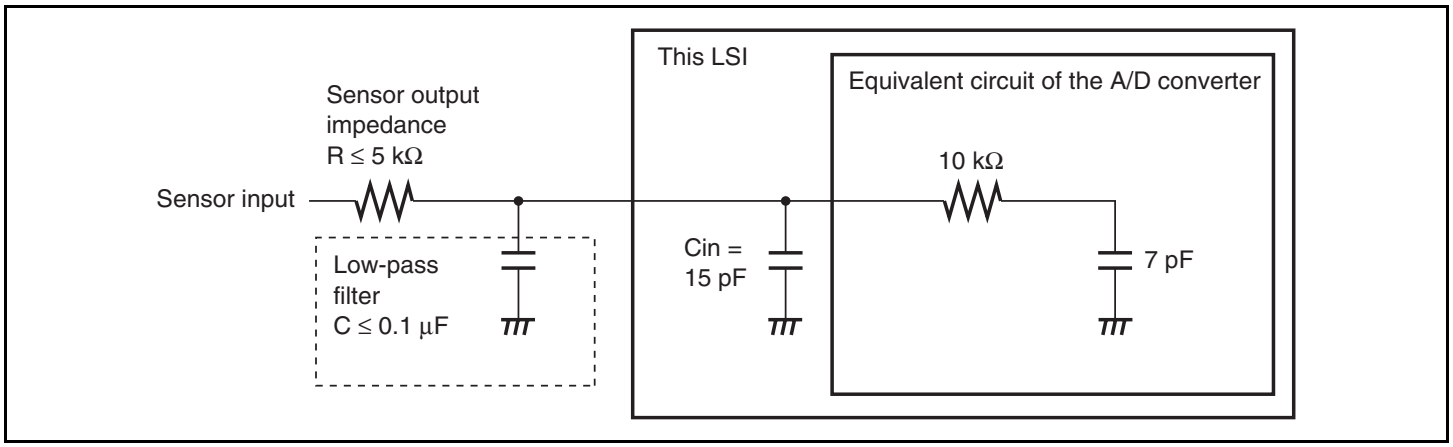


Figure 18.11 Example of Analog Input Circuit

18.7.5 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.

18.7.6 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
  The voltage applied to analog input pin ANn during A/D conversion should be in the range AVss ≤ V_{AN} ≤ Vref.
- Relation between AVcc, AVss and Vcc, Vss
  As the relationship between AVcc, AVss and Vcc, Vss, set AVcc = Vcc ± 0.3 V and AVss = Vss. If the A/D converter is not used, set AVcc = Vcc and AVss = Vss.
- Vref setting range
  The reference voltage at the Vref pin should be set in the range Vref ≤ AVcc.

18.7.7 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Digital circuitry must be isolated from the analog input pins (AN0 to AN15*), analog reference power supply (Vref), and analog power supply (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable ground (Vss) on the board.

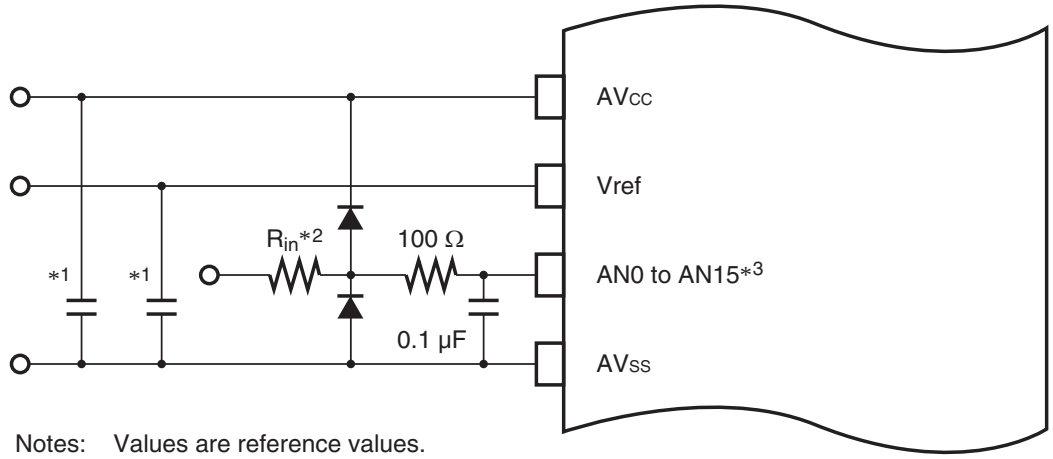
Note: * In the H8S/2454 group, only AN0 to AN7, AN12, and AN13 are available as analog input pins.

18.7.8 Notes on Noise Countermeasures

A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (AN0 to AN15*) should be connected between AVcc and AVss as shown in figure 18.12. Also, the bypass capacitors connected to AVcc and the filter capacitor connected to the AN0 to AN11 pins must be connected to AVss.

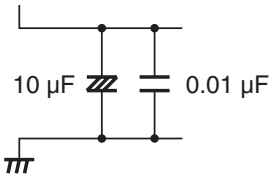
If a filter capacitor is connected, the input currents at the AN0 to AN15* pins are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (R_{in}), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding the circuit constants.

Note: * In the H8S/2454 group, only AN0 to AN7, AN11, and AN12 are available as analog input pins.



Notes: Values are reference values.

1.



2. R_{in}: Input impedance
3. The H8S/2454 group has only AN0 to AN7, AN11, and AN12 as analog input pins.

Figure 18.12 Example of Analog Input Protection Circuit

Table 18.9 Analog Pin Specifications

Item	Min.	Max.	Unit
Analog input capacitance	—	15	pF
Permissible signal source impedance	—	5	kΩ

Section 19 D/A Converter

19.1 Features

D/A converter features are listed below.

- 8-bit resolution
- Output channels: Two channels
- Maximum conversion time of 10 µs (with 20 pF load)
- Output voltage of 0 V to Vref
- D/A output hold function in software standby mode
- Setting the module stop mode

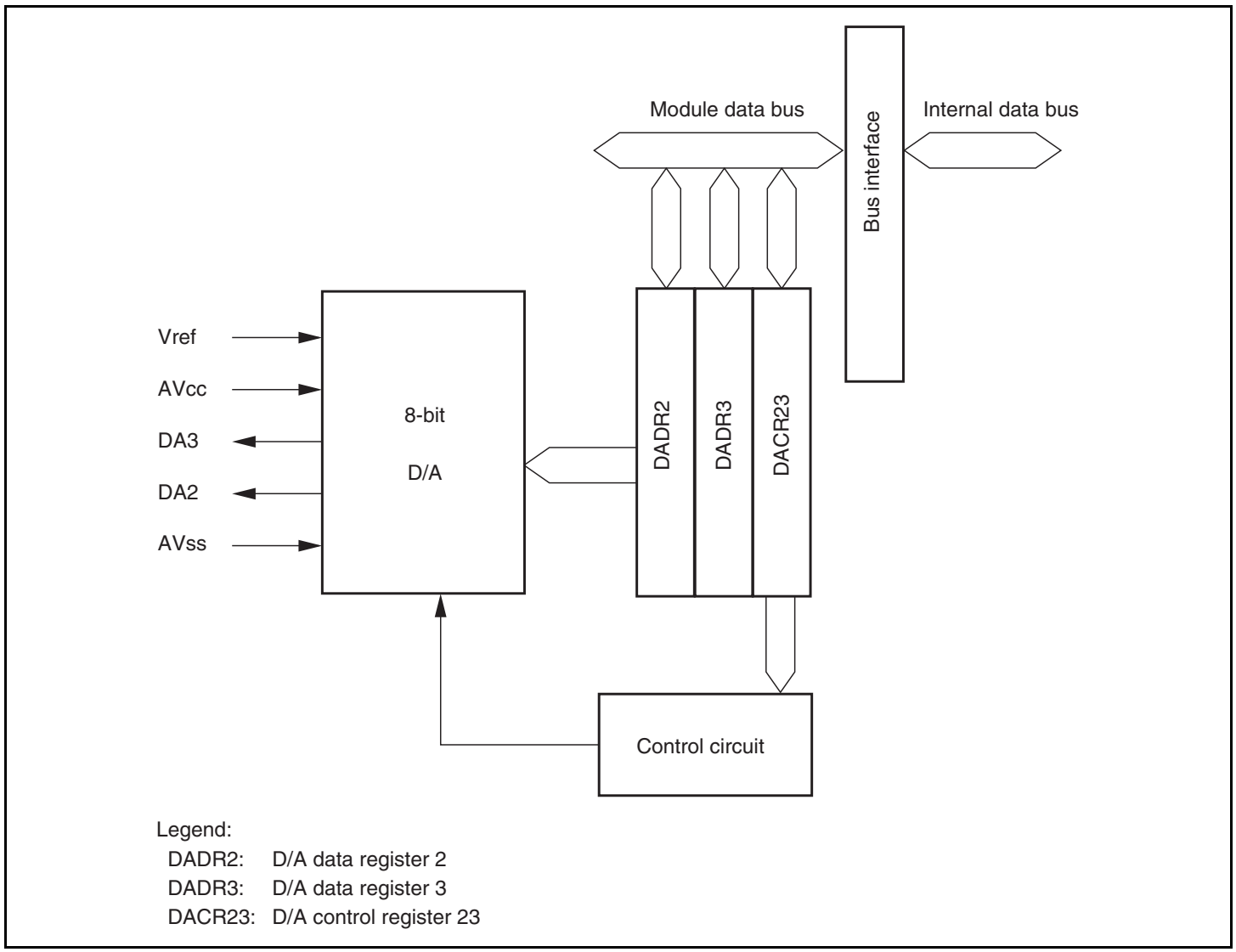


Figure 19.1 Block Diagram of D/A Converter

19.2 Input/Output Pins

Table 19.1 shows the pin configuration of the D/A converter.

Table 19.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power pin	AV_{CC}	Input	Analog power
Analog ground pin	AV_{SS}	Input	Analog ground
Reference voltage pin	Vref	Input	Reference voltage of D/A converter
Analog output pin 2	DA2	Output	Channel 2 analog output
Analog output pin 3	DA3	Output	Channel 3 analog output

19.3 Register Descriptions

The D/A converter has the following registers.

- D/A data register 2 (DADR2)
- D/A data register 3 (DADR3)
- D/A control register 23 (DACR23)

19.3.1 D/A Data Registers 2 and 3 (DADR2 and DADR3)

DADR2 and DADR3 are 8-bit readable/writable registers that store data for conversion.

Whenever analog output is enabled, the values in DADR are converted and output to the analog output pins.

19.3.2 D/A Control Register 23 (DACR23)

DACR23 controls the operation of channels 2 and 3 in the D/A converter.

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE3	0	R/W	D/A Output Enable 3 Controls D/A conversion and analog output. 0: Channel 3 analog output (DA3) is disabled. 1: Channel 3 D/A conversion is enabled; channel 3 analog output (DA3) is enabled.
6	DAOE2	0	R/W	D/A Output Enable 2 Controls D/A conversion and analog output. 0: Channel 2 analog output (DA2) is disabled. 1: Channel 2 D/A conversion is enabled; channel 2 analog output (DA2) is enabled.
5	DAE	0	R/W	D/A Enable This bit is used together with the DAOE2 and DAOE3 bits to control D/A conversion. When the DAE bit is cleared to 0, channel 2 and 3 D/A conversions are controlled independently. When the DAE bit is set to 1, channel 2 and 3 D/A conversions are controlled together. Output of conversion results is always controlled independently by the DAOE2 and DAOE3 bits. For details, see table 19.2.
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Table 19.2 Control of D/A Conversion

Bit 5 DAE	Bit 7 DAOE3	Bit 6 DAOE2	Description
0	0	0	D/A conversion disabled
		1	Channel 2 D/A conversion enabled, and channel 3 D/A conversion disabled. Channel 2 analog output (DA2) enabled, and channel 3 analog output (DA3) disabled.
	1	0	Channel 2 D/A conversion disabled, and channel 3 D/A conversion enabled. Channel 2 analog output (DA2) disabled, channel 3 analog output (DA3) enabled.
		1	Channel 2 and 3 D/A conversions enabled. Channel 2 and 3 analog outputs (DA2 and DA3) enabled.
1	0	0	Channel 2 and 3 D/A conversions enabled. Channel 2 and 3 analog outputs (DA2 and DA3) disabled.
		1	Channel 2 and 3 D/A conversions enabled. Channel 2 analog output (DA2) enabled, and channel 3 analog output (DA3) disabled.
	1	0	Channel 2 and 3 D/A conversions enabled. Channel 2 analog output (DA2) disabled, and channel 3 analog output (DA3) enabled.
		1	Channel 2 and 3 D/A conversions enabled. Channel 2 and 3 analog outputs (DA2 and DA3) enabled.

19.4 Operation

The D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When DAOE bit in DACR23 is set to 1, D/A conversion is enabled and the conversion result is output. The following shows an example of D/A conversion on channel 2. Figure 19.2 shows the timing of this operation.

1. Write the conversion data to DADR2.

2. Set the DAOE2 bit in DACR23 to 1. D/A conversion is started. The conversion result is output from the analog output pin DA2 after the conversion time t_{DCONV} has elapsed. The conversion result is continued to output until DADR2 is written to again or the DAOE2 bit is cleared to 0. The output value is expressed by the following formula:

$$\frac{\text{DADR contents}}{256} \times \text{Vref}$$

3. If DADR2 is written to again, the conversion is immediately started. The conversion result is output after the conversion time t_{DCONV} has elapsed.

4. If the DAOE2 bit is cleared to 0, analog output is disabled.

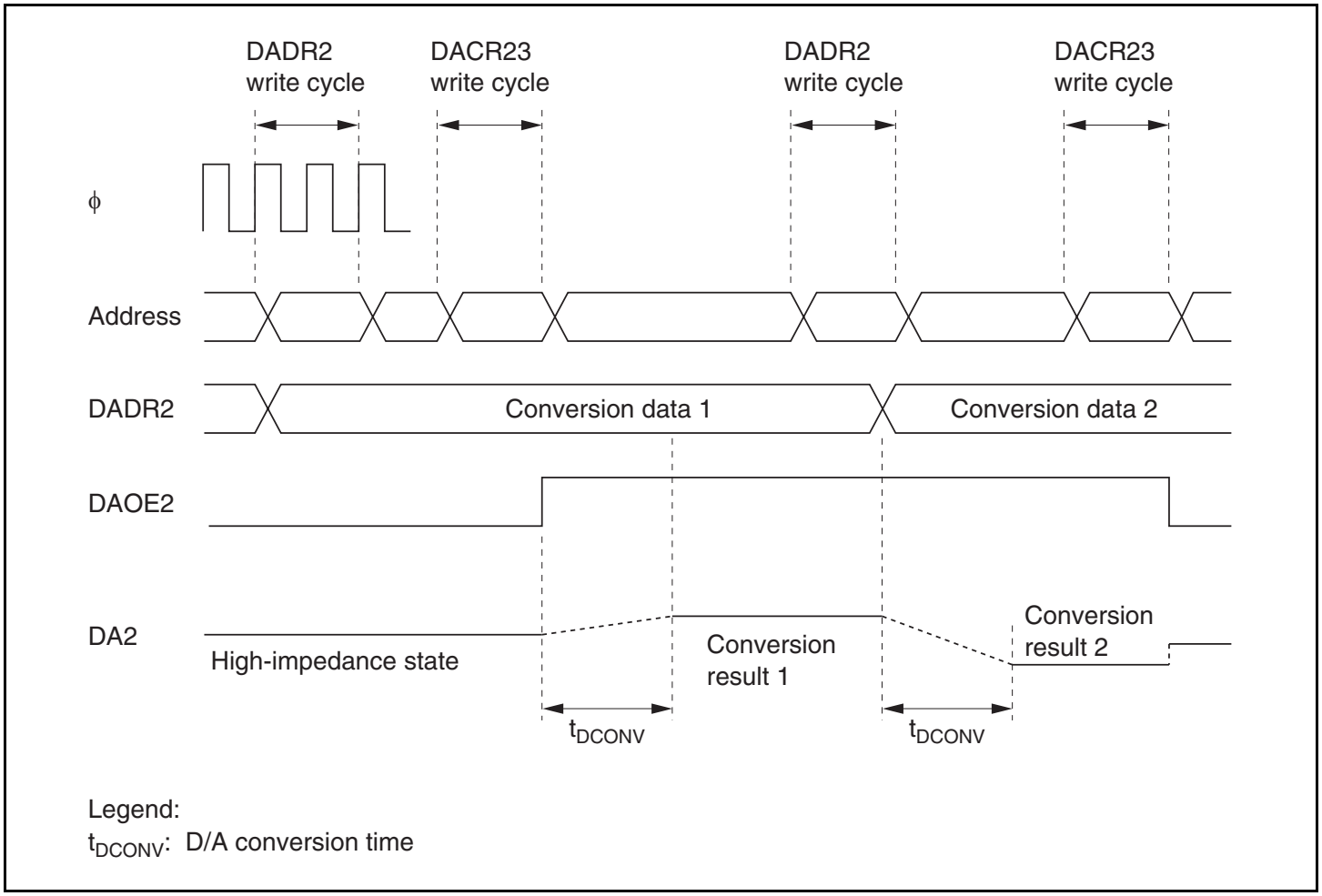


Figure 19.2 Example of D/A Converter Operation

19.5 Usage Notes

19.5.1 Setting for Module Stop Mode

It is possible to enable/disable the D/A converter operation using the module stop control register; the D/A converter does not operate by the initial value of the register. The register can be accessed by releasing the module stop mode. For details, see section 25, Power-Down Modes.

19.5.2 D/A Output Hold Function in Software Standby Mode

If D/A conversion is enabled and this LSI enters software standby mode, D/A output is held and analog power supply current remains at the same level during D/A conversion. When the analog power supply current is required to go low in software standby mode, bits DAOE and DAE should be cleared to 0, and D/A output should be disabled.

Section 20 Synchronous Serial Communication Unit (SSU)

This LSI has one channel of synchronous serial communication unit (SSU). The SSU has master mode in which this LSI outputs clocks as a master device for synchronous serial communication and slave mode in which clocks are input from an external device for synchronous serial communication. Synchronous serial communication can be performed with devices having different clock polarity and clock phase. Figure 20.1 is a block diagram of the SSU.

20.1 Features

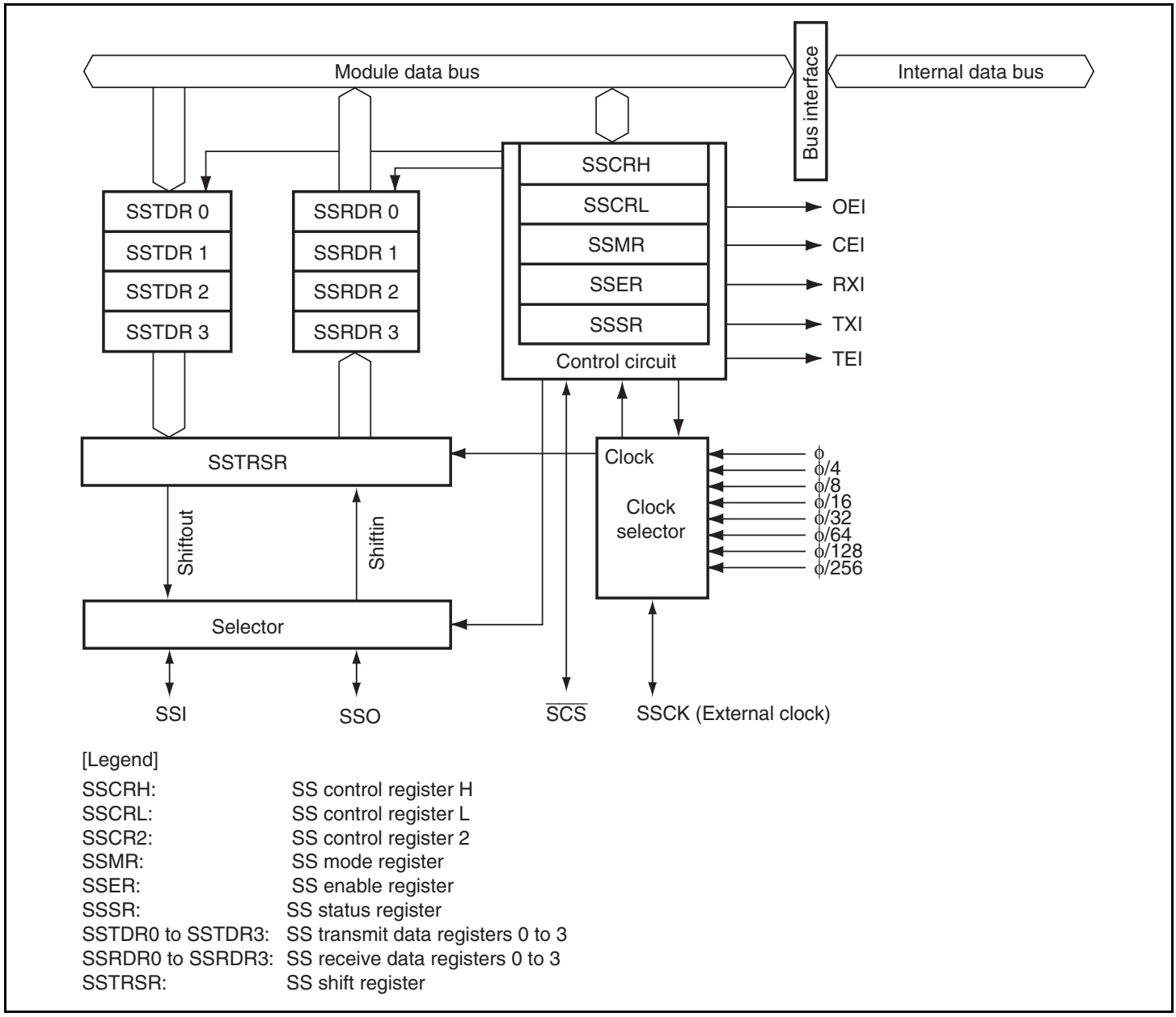
- Choice of SSU mode and clock synchronous mode
- Choice of master mode and slave mode
- Choice of standard mode and bidirectional mode
- Synchronous serial communication with devices with different clock polarity and clock phase
- Choice of 8/16/24/32-bit width of transmit/receive data
- Full-duplex communication capability

  The shift register is incorporated, enabling transmission and reception to be executed simultaneously.
- Consecutive serial communication
- Choice of LSB-first or MSB-first transfer
- Choice of a clock source

  Seven internal clocks (ϕ/4, ϕ/8, ϕ/16, ϕ/32, ϕ/64, ϕ/128, ϕ/256) or an external clock
- Five interrupt sources

  Transmit-end, transmit-data-register-empty, receive-data-full, overrun-error, and conflict error
- Module stop mode can be set

Figure 20.1 shows a block diagram of the SSU.



[Legend]

SSCRH:	SS control register H
SSCRL:	SS control register L
SSCR2:	SS control register 2
SSMR:	SS mode register
SSER:	SS enable register
SSSR:	SS status register
SSTDR0 to SSTDR3:	SS transmit data registers 0 to 3
SSRDR0 to SSRDR3:	SS receive data registers 0 to 3
SSTRSR:	SS shift register

Figure 20.1 Block Diagram of SSU

20.2 Input/Output Pins

Table 20.1 shows the SSU pin configuration.

Table 20.1 Pin Configuration

Channel	Symbol	I/O	Function
0	SSCK0	I/O	SSU clock input/output
	SSI0	I/O	SSU data input/output
	SSO0	I/O	SSU data input/output
	$\overline{\text{SCS0}}$	I/O	SSU chip select input/output

Note: * Because channel numbers are omitted in later descriptions, these are shown SSCK, SSI, SSO, and $\overline{\text{SCS}}$.

20.3 Register Descriptions

The SSU has the following registers.

- SS control register H_0 (SSCRH_0)
- SS control register L_0 (SSCRL_0)
- SS mode register_0 (SSMR_0)
- SS enable register_0 (SSER_0)
- SS status register_0 (SSSR_0)
- SS control register 2_0 (SSCR2_0)
- SS transmit data register 0_0 (SSTDR0_0)
- SS transmit data register 1_0 (SSTDR1_0)
- SS transmit data register 2_0 (SSTDR2_0)
- SS transmit data register 3_0 (SSTDR3_0)
- SS receive data register 0_0 (SSRDR0_0)
- SS receive data register 1_0 (SSRDR1_0)
- SS receive data register 2_0 (SSRDR2_0)
- SS receive data register 3_0 (SSRDR3_0)
- SS shift register_0 (SSTRSR_0)

20.3.1 SS Control Register H (SSCRH)

SSCRH specifies master/slave device selection, bidirectional mode enable, SSO pin output value selection, SSCK pin selection, and $\overline{\text{SCS}}$ pin selection.

Bit	Bit Name	Initial Value	R/W	Description
7	MSS	0	R/W	Master/Slave Device Select Selects that this module is used in master mode or slave mode. When master mode is selected, transfer clocks are output from the SSCK pin. When the CE bit in SSSR is set, this bit is automatically cleared. 0: Slave mode is selected. 1: Master mode is selected.
6	BIDE	0	R/W	Bidirectional Mode Enable Selects that both serial data input pin and output pin are used or one of them is used. However, transmission and reception are not performed simultaneously when bidirectional mode is selected. For details, section 20.4.3, Relationship between Data Input/Output Pins and Shift Register. 0: Standard mode (two pins are used for data input and output) 1: Bidirectional mode (one pin is used for data input and output)
5	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4	SOL	0	R/W	Serial Data Output Value Select The serial data output retains its level of the last bit after completion of transmission. The output level before or after transmission can be specified by setting this bit. When specifying the output level, use the MOV instruction after clearing the SOLP bit to 0. Since writing to this bit during data transmission causes malfunctions, this bit should not be changed. 0: Serial data output is changed to low. 1: Serial data output is changed to high.
3	SOLP	1	R/W	SOL Bit Write Protect When changing the output level of serial data, set the SOL bit to 1 or clear the SOL bit to 0 after clearing the SOLP bit to 0 using the MOV instruction. 0: Output level can be changed by the SOL bit 1: Output level cannot be changed by the SOL bit. This bit is always read as 1.
2	SCKS	0	R/W	SSCK Pin Select Selects that the SSCK pin functions as a port or a serial clock pin. When the SSCK pin is used as a serial clock pin, this bit must be set to 1. 0: Functions as an I/O port. 1: Functions as a serial clock.
1	CSS1	0	R/W	$\overline{\text{SCS}}$ Pin Select
0	CSS0	0	R/W	Select that the $\overline{\text{SCS}}$ pin functions as a port or $\overline{\text{SCS}}$ input or output. However, when MSS = 0, the $\overline{\text{SCS}}$ pin functions as an input pin regardless of the CSS1 and CSS0 settings. 00: I/O port 01: Function as $\overline{\text{SCS}}$ input 10: Function as $\overline{\text{SCS}}$ automatic input/output (function as $\overline{\text{SCS}}$ input before and after transfer and output a low level during transfer) 11: Function as $\overline{\text{SCS}}$ automatic output (outputs a high level before and after transfer and outputs a low level during transfer)

20.3.2 SS Control Register L (SSCRL)

SSCRL selects operating mode, software reset, and transmit/receive data length.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
6	SSUMS	0	R/W	Selects transfer mode from SSU mode and clock synchronous mode. 0: SSU mode 1: Clock synchronous mode
5	SRES	0	R/W	Software Reset Setting this bit to 1 forcibly resets the SSU internal sequencer. After that, this bit is automatically cleared. The ORER, TEND, TDRE, RDRF, and CE bits in SSSR and the TE and RE bits in SSER are also initialized. Values of other bits for SSU registers are held. To stop transfer, set this bit to 1 to reset the SSU internal sequencer.
4 to 2	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
1	DATS1	0	R/W	Transmit/Receive Data Length Select
0	DATS0	0	R/W	Select serial data length. 00: 8 bits 01: 16 bits 10: 32 bits 11: 24 bits

20.3.3 SS Mode Register (SSMR)

SSMR selects the MSB first/LSB first, clock polarity, clock phase, and clock rate of synchronous serial communication.

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB First/LSB First Select Selects that the serial data is transmitted in MSB first or LSB first. 0: LSB first 1: MSB first
6	CPOS	0	R/W	Clock Polarity Select Selects the SSCK clock polarity. 0: High output in idle mode, and low output in active mode 1: Low output in idle mode, and high output in active mode
5	CPHS	0	R/W	Clock Phase Select (Only for SSU Mode) Selects the SSCK clock phase. 0: Data changes at the first edge. 1: Data is latched at the first edge.
4, 3	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
2	CKS2	0	R/W	Transfer Clock Rate Select
1	CKS1	0	R/W	Select the transfer clock rate when an internal clock is selected.
0	CKS0	0	R/W	000: Reserved 100: ϕ/32 001: ϕ/4 101: ϕ/64 010: ϕ/8 110: ϕ/128 011: ϕ/16 111: ϕ/256

20.3.4 SS Enable Register (SSER)

SSER performs transfer/receive control of synchronous serial communication and setting of interrupt enable.

Bit	Bit Name	Initial Value	R/W	Description
7	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled.
6	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.
5, 4	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
3	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt request is enabled.
2	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a TXI interrupt request is enabled.
1	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, an RXI interrupt request and an OEI interrupt request are enabled.
0	CEIE	0	R/W	Conflict Error Interrupt Enable When this bit is set to 1, a CEI interrupt request is enabled.

20.3.5 SS Status Register (SSSR)

SSSR is a status flag register for interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
6	ORER	0	R/W	Overrun Error If the next data is received while RDRF = 1, an overrun error occurs, indicating abnormal termination. SSRDR stores 1-frame receive data before an overrun error occurs and loses data to be received later. While ORER = 1, consecutive serial reception cannot be continued. Serial transmission cannot be continued, either. [Setting condition] When one byte of the next reception is completed with RDRF = 1 [Clearing condition] When writing 0 after reading ORER = 1 (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.)
5, 4	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	TEND	1	R	Transmit End [Setting condition] • When the last bit of transmit data is transmitted while the TENDSTS bit in SSCR2 is cleared to 0 and the TDRE bit is set to 1 • After the last bit of transmit data is transmitted while the TENDSTS bit in SSCR2 is set to 1 and the TDRE bit is set to 1 [Clearing conditions] • When writing 0 after reading TEND = 1 (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.) • When writing data to SSTDR
2	TDRE	1	R/W	Transmit Data Empty Indicates whether or not SSTDR contains transmit data. [Setting conditions] • When the TE bit in SSER is 0 • When data is transferred from SSTDR to SSTRSR and SSTDR is ready to be written to. [Clearing conditions] • When writing 0 after reading TDRE = 1 (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.) • When writing data to SSTDR with TE = 1
1	RDRF	0	R/W	Receive Data Register Full Indicates whether or not SSRDR contains receive data. [Setting condition] • When receive data is transferred from SSTRSR to SSRDR after successful serial data reception [Clearing conditions] • When writing 0 after reading RDRF = 1 (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.) • When reading receive data from SSRDR

Bit	Bit Name	Initial Value	R/W	Description
0	CE	0	R/W	Conflict/Incomplete Error Indicates that a conflict error has occurred when 0 is externally input to the $\overline{SCS}$ pin with SSUMS = 0 (SSU mode) and MSS = 1 (master device). If the $\overline{SCS}$ pin level changes to 1 with SSUMS = 0 (SSU mode) and MSS = 0 (slave device), an incomplete error occurs because it is determined that a master device has terminated the transfer. Data reception does not continue while the CE bit is set to 1. Serial transmission also does not continue. Reset the SSU internal sequencer by setting the SRES bit in SSCRL to 1 before resuming transfer after incomplete error. [Setting condition] • When a low level is input to the $\overline{SCS}$ pin in master device (the MSS bit in SSCRH is set to 1) • When the $\overline{SCS}$ pin is changed to 1 during transfer in slave device (the MSS bit in SSCRH is cleared to 0) [Clearing condition] • When writing 0 after reading CE = 1 (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.)

20.3.6 SS Control Register 2 (SSCR2)

SSCR2 is a register that enables/disables the open-drain outputs of the SSO, SSI, SSCK, and $\overline{\text{SCS}}$ pins, selects the assert timing of the $\overline{\text{SCS}}$ pin, data output timing of the SSO pin, and set timing of the TEND bit.

Bit	Bit Name	Initial Value	R/W	Description
7	SDOS	0	R/W	Serial Data Pin Open Drain Select Selects whether the serial data output pin is used as a CMOS or an NMOS open drain output. Pins to output serial data differ according to the register setting. For details, 20.4.3, Relationship between Data Input/Output Pins and Shift Register. 0: CMOS output 1: NMOS open drain output
6	SSCKOS	0	R/W	SSCK Pin Open Drain Select Selects whether the SSCK pin is used as a CMOS or an NMOS open drain output. 0: CMOS output 1: NMOS open drain output
5	SCSOS	0	R/W	$\overline{\text{SCS}}$ Pin Open Drain Select Selects whether the $\overline{\text{SCS}}$ pin is used as a CMOS or an NMOS open drain output. 0: CMOS output 1: NMOS open drain output
4	TENDSTS	0	R/W	Selects the timing of setting the TEND bit (valid in SSU and master mode). 0: Sets the TEND bit when the last bit is being transmitted 1: Sets the TEND bit after the last bit is transmitted

Bit	Bit Name	Initial Value	R/W	Description
3	SCSATS	0	R/W	Selects the assertion timing of the $\overline{SCS}$ pin (valid in SSU and master mode). 0: Min. values of t_{LEAD} and t_{LAG} are 1/2 × t_{SUcyc} 1: Min. values of t_{LEAD} and t_{LAG} are 3/2 × t_{SUcyc}
2	SSODTS	0	R/W	Selects the data output timing of the SSO pin (valid in SSU and master mode) 0: While BIDE = 0, MSS = 1, and TE = 1 or while BIDE = 1, TE = 1, and RE = 0, the SSO pin outputs data 1: While BIDE = 0, MSS = 1, and TE = 1 or while BIDE = 1, TE = 1, and RE = 0, the SSO pin outputs data while the SCS pin is driven low
1, 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

20.3.7 SS Transmit Data Registers 0 to 3 (SSTDR0 to SSTDR3)

SSTDR is an 8-bit register that stores transmit data. When 8-bit data length is selected by bits DATS1 and DATS0 in SSCRL, SSTDR0 is valid. When 16-bit data length is selected, SSTDR0 and SSTDR1 are valid. When 24-bit data length is selected, SSTDR0, SSTDR1, and SSTDR2 are valid. When 32-bit data length is selected, SSTDR0 to SSTDR3 are valid. Be sure not to access to invalid SSTDRs.

When the SSU detects that SSTRSR is empty, it transfers the transmit data written in SSTDR to SSTRSR and starts serial transmission. If the next transmit data has already been written to SSTDR during serial transmission, the SSU performs consecutive serial transmission.

Although SSTDR can always be read from or written to by the CPU and DMAC, to achieve reliable serial transmission, write transmit data to SSTDR after confirming that the TDRE bit in SSSR is set to 1.

Table 20.2 Correspondence Between DATS Bit Setting and SSTDR

	DATS[1:0] (SSCRL[1:0])			
SSTDR	**00**	**01**	**10**	**11 (Setting Invalid)**
0	Valid	Valid	Valid	Valid
1	Invalid	Valid	Valid	Valid
2	Invalid	Invalid	Valid	Valid
3	Invalid	Invalid	Valid	Invalid

20.3.8 SS Receive Data Registers 0 to 3 (SSRDR0 to SSRDR3)

SSRDR is an 8-bit register that stores receive data. When 8-bit data length is selected by bits DATS1 and DATS0 in SSCRL, SSRDR0 is valid. When 16-bit data length is selected, SSRDR0 and SSRDR1 are valid. When 24-bit data length is selected, SSRDR0, SSRDR1, and SSRDR2 are valid. When 32-bit data length is selected, SSRDR0 to SSRDR3 are valid. Be sure not to access to invalid SSRDR.

When the SSU has received 1-byte data, it transfers the received serial data from SSTRSR to SSRDR where it is stored. After this, SSTRSR is ready for reception. Since SSTRSR and SSRDR function as a double buffer in this way, consecutive receive operations can be performed.

Read SSRDR after confirming that the RDRF bit in SSSR is set to 1.

SSRDR is a read-only register, therefore, cannot be written to by the CPU.

Table 20.3 Correspondence Between DATS Bit Setting and SSRDR

	DATS[1:0] (SSCRL[1:0])			
SSRDR	**00**	**01**	**10**	**11 (Setting Invalid)**
0	Valid	Valid	Valid	Valid
1	Invalid	Valid	Valid	Valid
2	Invalid	Invalid	Valid	Valid
3	Invalid	Invalid	Valid	Invalid

20.3.9 SS Shift Register (SSTRSR)

SSTRSR is a shift register that transmits and receives serial data.

When data is transferred from SSTDR to SSTRSR, bit 0 of transmit data is bit 0 in the SSTDR contents (MLS = 0: LSB first communication) and is bit 7 in the SSTDR contents (MLS = 1: MSB first communication). The SSU transfers data from the LSB (bit 0) in SSTRSR to the SSO pin to perform serial data transmission.

In reception, the SSU sets serial data that has been input via the SSI pin in SSTRSR from the LSB (bit 0). When 1-byte data has been received, the SSTRSR contents are automatically transferred to SSRDR. SSTRSR cannot be directly accessed by the CPU.

20.4 Operation

20.4.1 Transfer Clock

A transfer clock can be selected from eight internal clocks and an external clock. When using this module, set the SCKS bit in SSCRH to 1 to select the SSCK pin as a serial clock. When the MSS bit in SSCRH is 1, an internal clock is selected and the SSCK pin is used as an output pin. When transfer is started, the clock with the transfer rate set by bits CKS2 to CKS0 in SSMR is output from the SSCK pin. When MSS = 0, an external clock is selected and the SSCK pin is used as an input pin.

20.4.2 Relationship of Clock Phase, Polarity, and Data

The relationship of clock phase, polarity, and transfer data depends on the combination of the CPOS and CPHS bits in SSMR. Figure 20.2 shows the relationship. When SSUMS = 1, the CPHS setting is invalid although the CPOS setting is valid.

Setting the MLS bit in SSMR selects that MSB or LSB first communication. When MLS = 0, data is transferred from the LSB to the MSB. When MLS = 1, data is transferred from the MSB to the LSB.

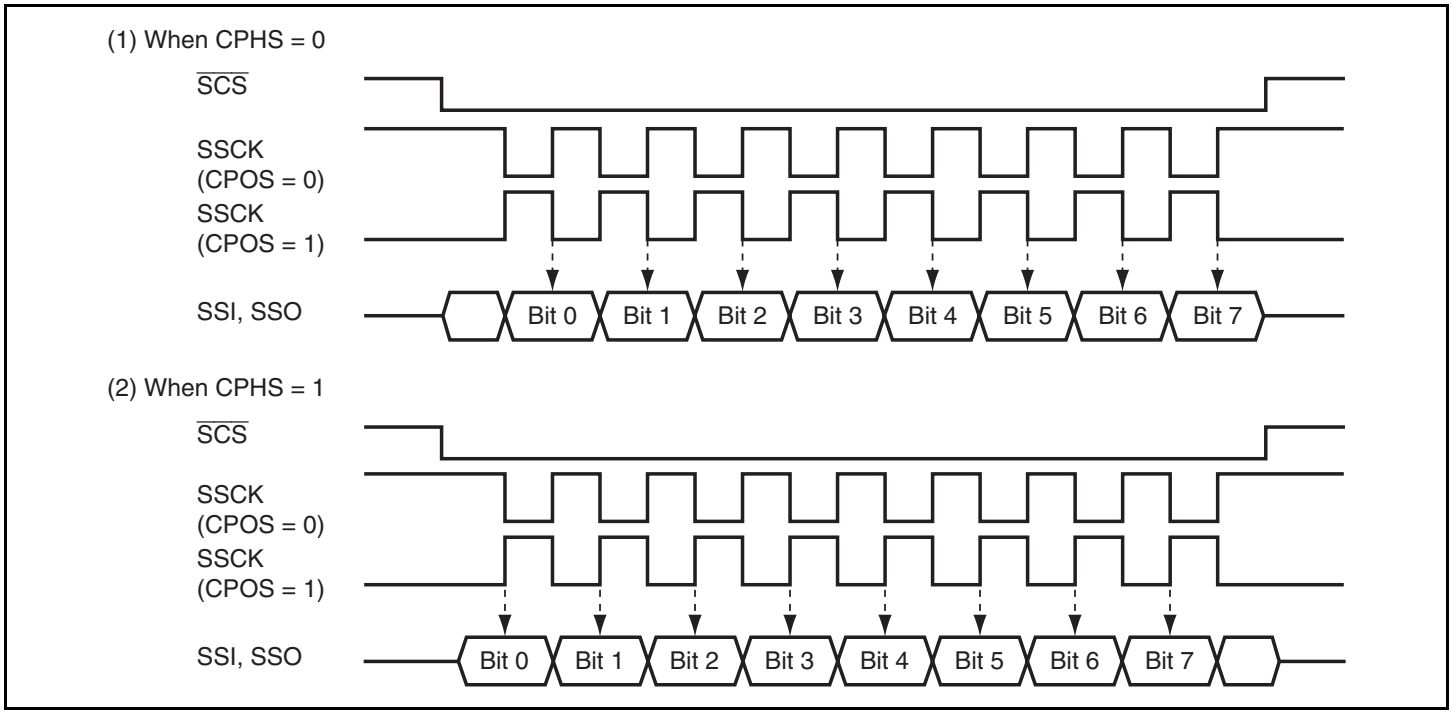


Figure 20.2 Relationship of Clock Phase, Polarity, and Data

20.4.3 Relationship between Data Input/Output Pins and Shift Register

The connection between data input/output pins and the SS shift register (SSTRSR) depends on the combination of the MSS and BIDE bits in SSCRH and the SSUMS bit in SSCRL. Figure 20.3 shows the relationship.

The SSU transmits serial data from the SSO pin and receives serial data from the SSI pin when operating with BIDE = 0 and MSS = 1 (standard, master mode) (see figure 20.3 (1)). The SSU transmits serial data from the SSI pin and receives serial data from the SSO pin when operating with BIDE = 0 and MSS = 0 (standard, slave mode) (see figure 20.3 (2)).

The SSU transmits and receives serial data from the SSO pin regardless of master or slave mode when operating with BIDE = 1 (bidirectional mode) (see figures 20.3 (3) and (4)).

However, even if both the TE and RE bits are set to 1, transmission and reception are not performed simultaneously. Either the TE or RE bit must be selected.

The SSU transmits serial data from the SSO pin and receives serial data from the SSI pin when operating with SSUMS = 1. The SSCK pin outputs the internal clock when MSS = 1 and function as an input pin when MSS = 0 (see figures 20.3 (5) and (6)).

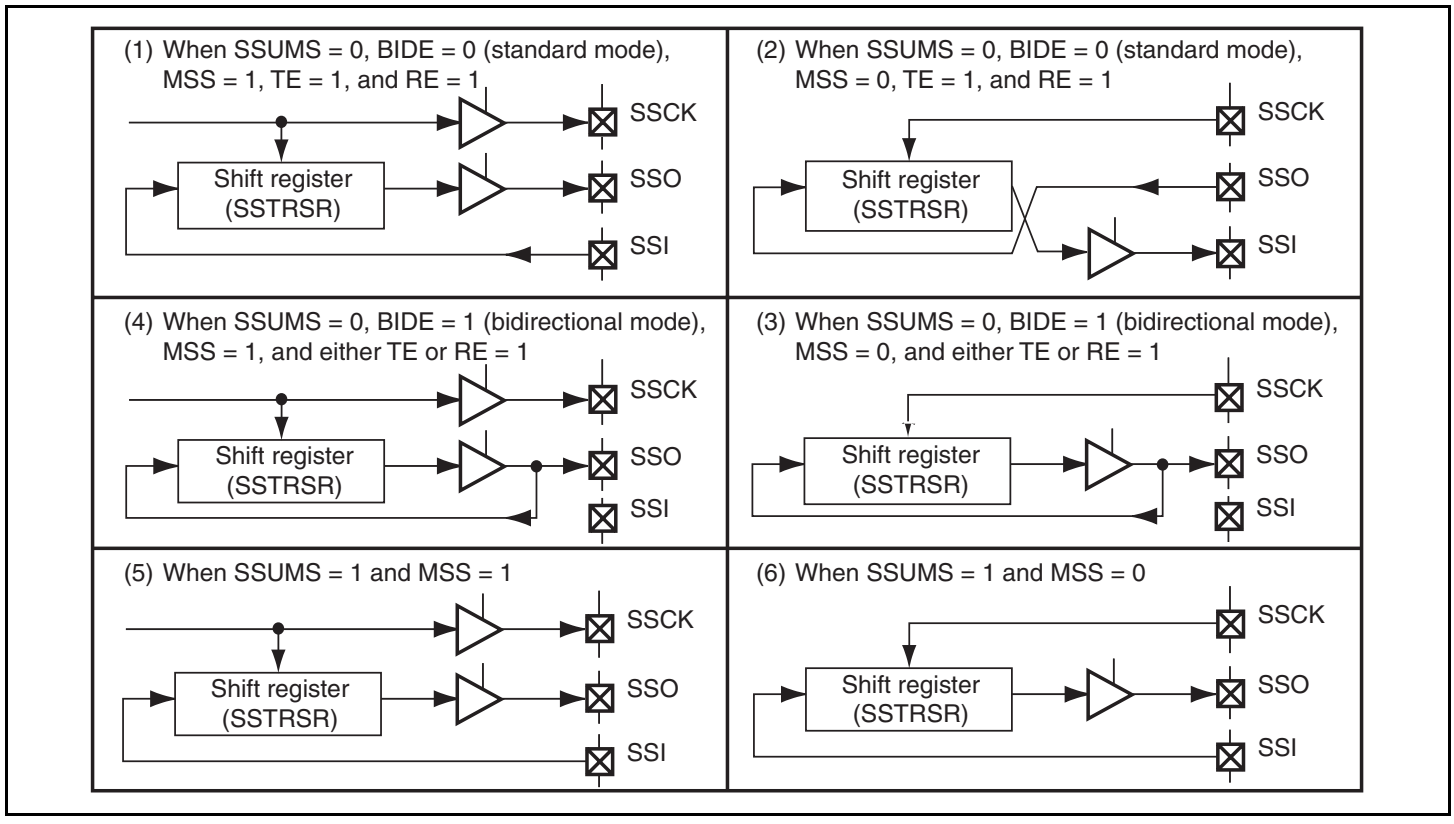


Figure 20.3 Relationship between Data Input/Output Pins and the Shift Register

20.4.4 Communication Modes and Pin Functions

The SSU switches the input/output pin (SSI, SSO, SSCK, and $\overline{SCS}$) functions according to the communication modes and register settings. When a pin is used as an input pin, set the corresponding bit in the input buffer control register (ICR) to 1. The relationship of communication modes and input/output pin functions are shown in tables 20.4 to 20.6.

Table 20.4 Communication Modes and Pin States of SSI and SSO Pins

Communication Mode	Register Setting					Pin State	
	SSUMS	BIDE	MSS	TE	RE	SSI	SSO
SSU communication mode	0	0	0	0	1	—	Input
				1	0	Output	—
					1	Output	Input
			1	0	1	Input	—
				1	0	—	Output
					1	Input	Output
SSU (bidirectional) communication mode	0	1	0	0	1	—	Input
				1	0	—	Output
			1	0	1	—	Input
				1	0	—	Output
Clock synchronous communication mode	1	0	0	0	1	Input	—
				1	0	—	Output
					1	Input	Output
			1	0	1	Input	—
				1	0	—	Output
					1	Input	Output

[Legend]

—: Not used as SSU pin (can be used as I/O port)

Table 20.5 Communication Modes and Pin States of SSCK Pin

Communication Mode	Register Setting			Pin State
	SSUMS	MSS	SCKS	SSCK
SSU communication mode	0	0	0	—
			1	Input
		1	0	—
			1	Output
Clock synchronous communication mode	1	0	0	—
			1	Input
		1	0	—
			1	Output

[Legend]

—: Not used as SSU pin

Table 20.6 Communication Modes and Pin States of $\overline{\text{SCS}}$ Pin

Communication Mode	Register Setting				Pin State
	SSUMS	MSS	CSS1	CSS0	$\overline{\text{SCS}}$
SSU communication mode	0	0	×	×	Input
		1	0	0	—
			0	1	Input
			1	0	Automatic input/output
			1	1	Output
Clock synchronous communication mode	1	×	×	×	—

[Legend]

×: Don't care

—: Not used as SSU pin

20.4.5 SSU Mode

In SSU mode, data communications are performed via four lines: clock line (SSCK), data input line (SSI or SSO), data output line (SSI or SSO), and chip select line ($\overline{SCS}$).

In addition, the SSU supports bidirectional mode in which a single pin functions as data input and data output lines.

(1) Initial Settings in SSU Mode

Figure 20.4 shows an example of the initial settings in SSU mode. Before data transfer, clear both the TE and RE bits in SSER to 0 to set the initial values.

Note: Before changing operating modes and communications formats, clear both the TE and RE bits to 0. Although clearing the TE bit to 0 sets the TDRE bit to 1, clearing the RE bit to 0 does not change the values of the RDRF and ORER bits and SSRDR. Those bits retain the previous values.

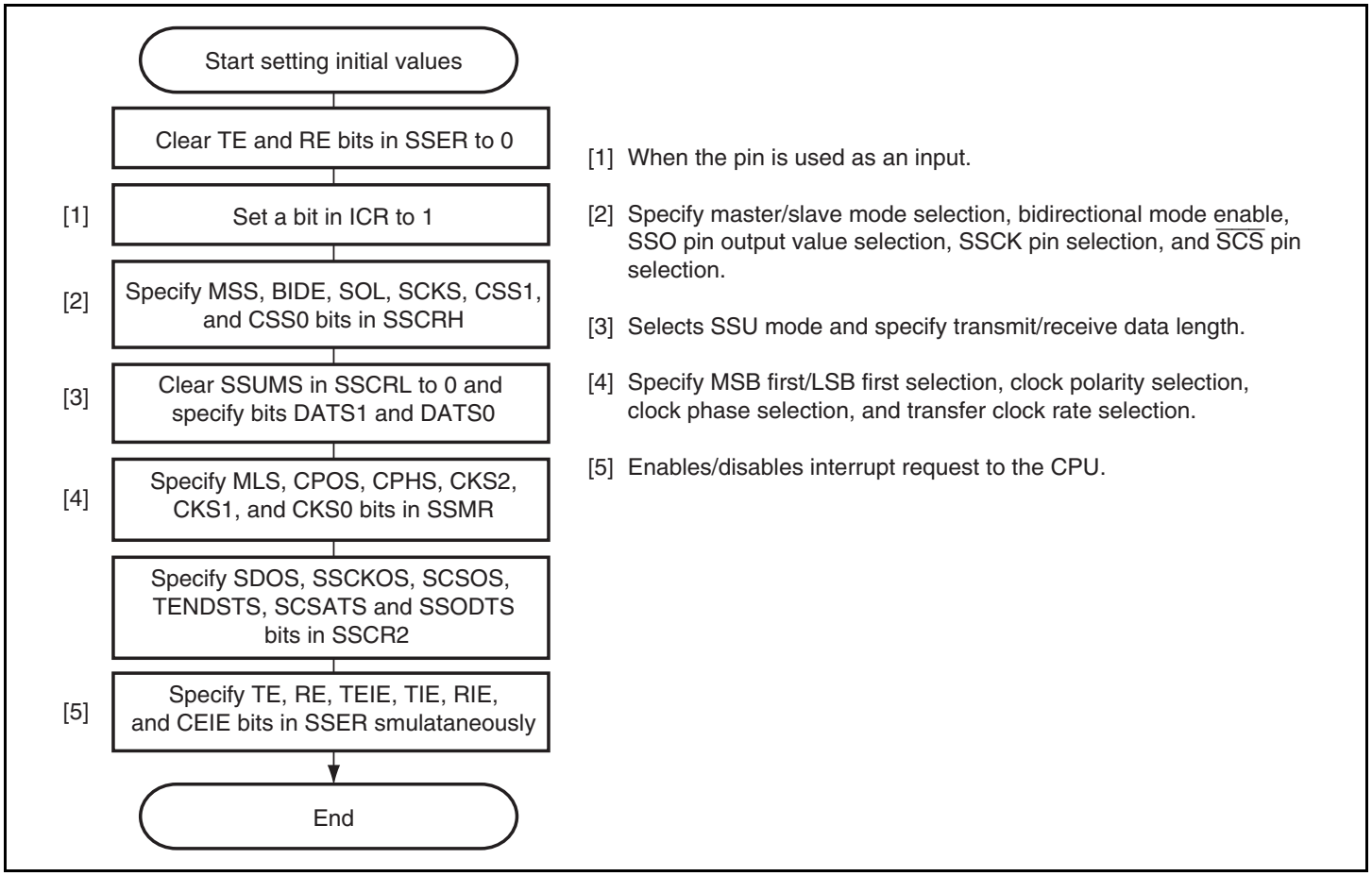


Figure 20.4 Example of Initial Settings in SSU Mode

(2) Data Transmission

Figure 20.5 shows an example of transmission operation, and figure 20.6 shows a flowchart example of data transmission.

When transmitting data, the SSU operates as shown below.

In master mode, the SSU outputs a transfer clock and data. In slave mode, when a low level signal is input to the $\overline{\text{SCS}}$ pin and a transfer clock is input to the SSCK pin, the SSU outputs data in synchronization with the transfer clock.

Writing transmit data to SSTDR after the TE bit is set to 1 clears the TDRE bit in SSSR to 0, and the SSTDR contents are transferred to SSTRSR. After that, the SSU sets the TDRE bit to 1 and starts transmission. At this time, if the TIE bit in SSER is set to 1, a TXI interrupt is generated.

When 1-frame data has been transferred with TDRE = 0, the SSTDR contents are transferred to SSTRSR to start the next frame transmission. When the 8th bit of transmit data has been transferred with TDRE = 1, the TEND bit in SSSR is set to 1 and the state is retained. At this time, if the TEIE bit is set to 1, a TEI interrupt is generated. After transmission, the output level of the SSCK pin is fixed high when CPOS = 0 and low when CPOS = 1.

While the ORER bit in SSSR is set to 1, transmission is not performed. Check that the ORER bit is cleared to 0.

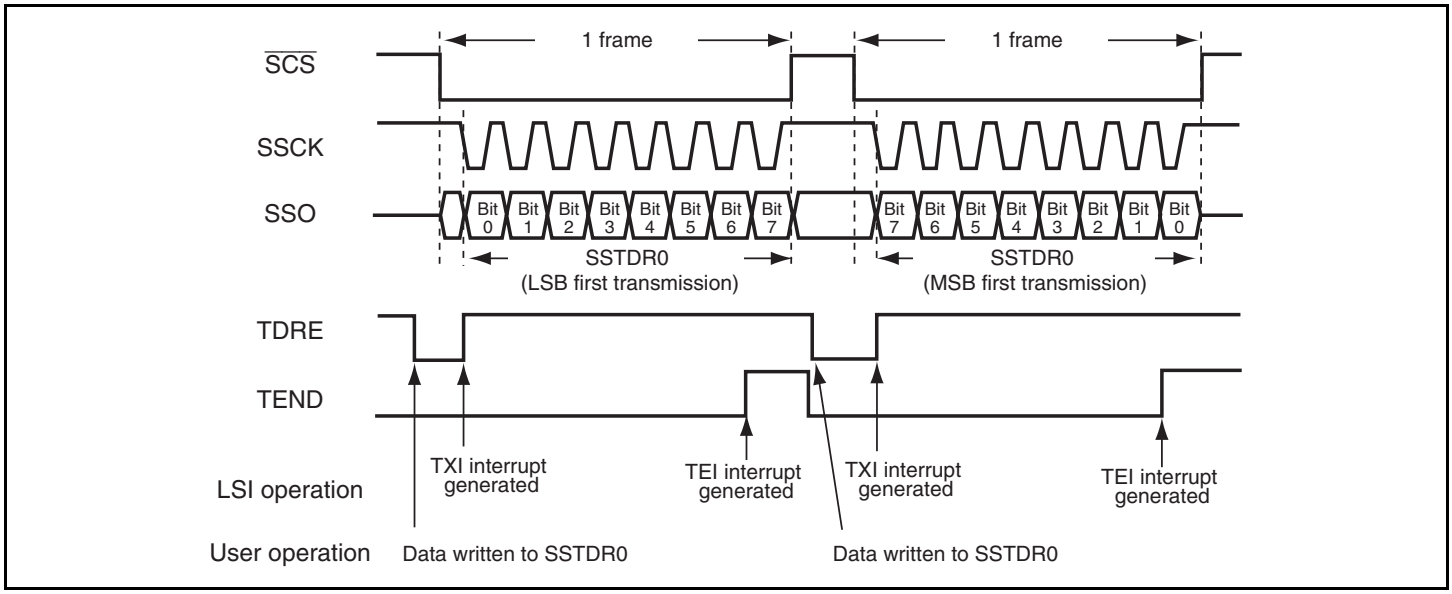


Figure 20.5 (1) Example of Transmission Operation (SSU Mode)
When 8-bit data length is selected (SSTDR0 is valid) with CPOS = 0 and CPHS = 0

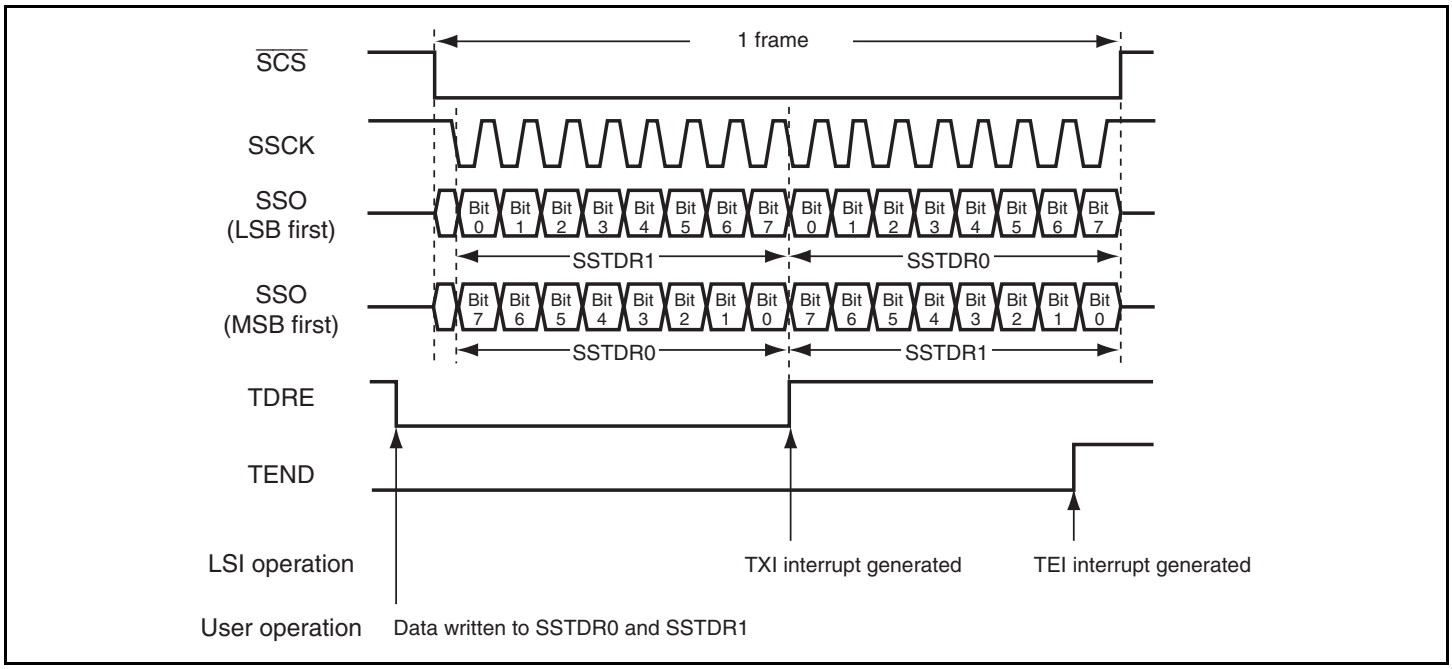


Figure 20.5 (2) Example of Transmission Operation (SSU Mode) When 16-bit data length is selected (SSTDR0 and SSTDR1 are valid) with CPOS = 0 and CPHS = 0

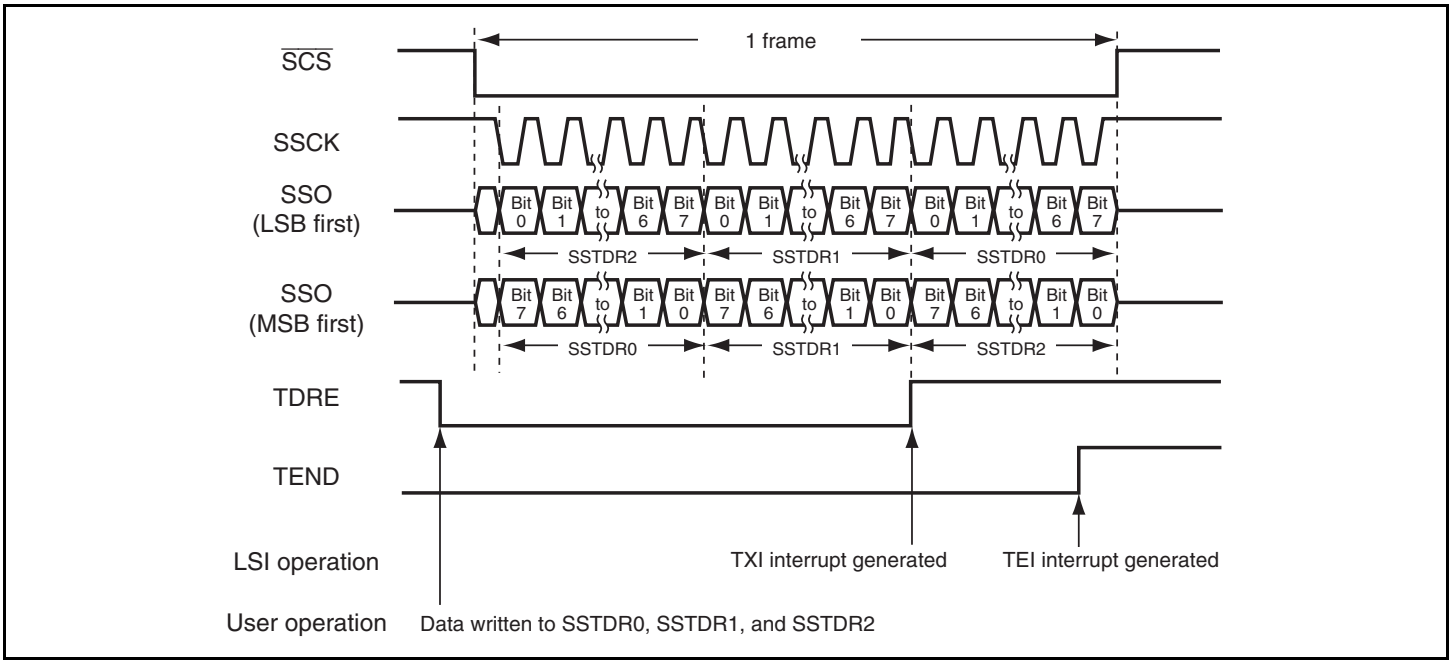


Figure 20.5 (3) Example of Transmission Operation (SSU Mode) When 24-bit data length is selected (SSTDR0, SSTDR1, and SSTDR2 are valid) with CPOS = 0 and CPHS = 0

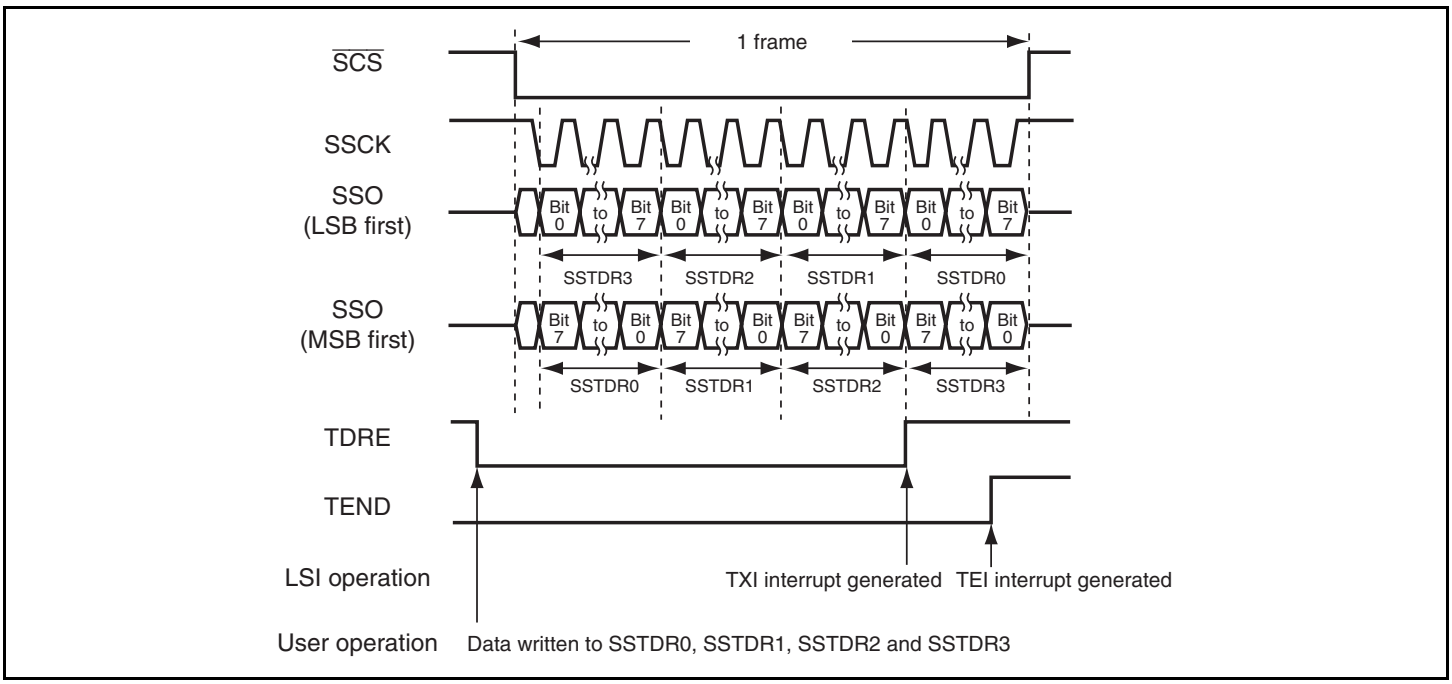
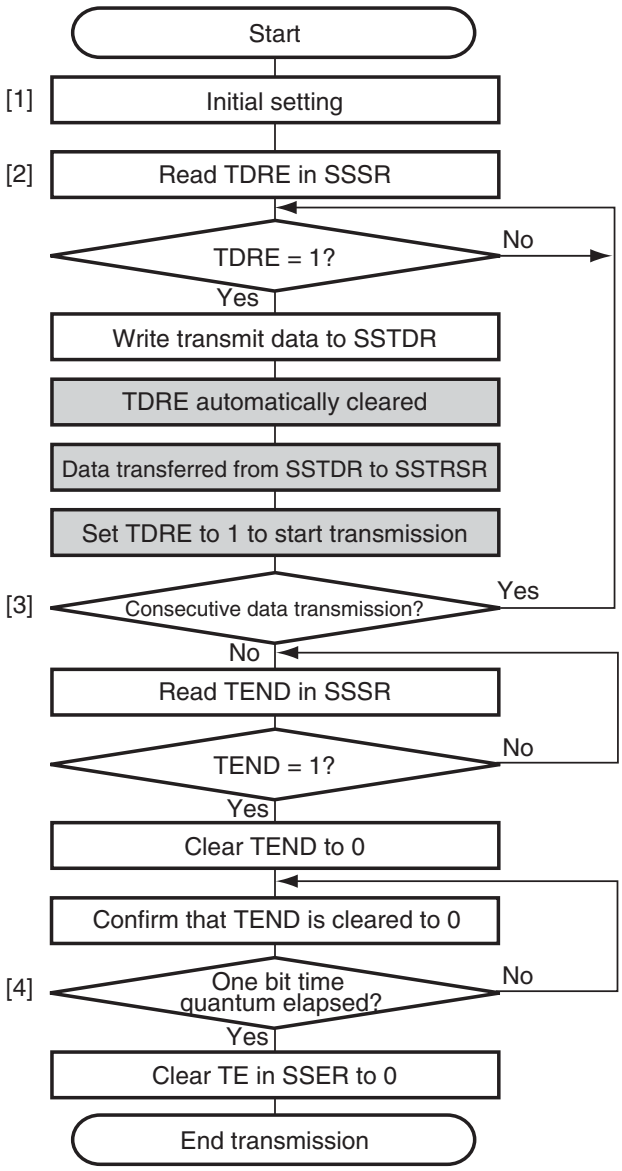


Figure 20.5 (4) Example of Transmission Operation (SSU Mode) When 32-bit data length is selected (SSTDR0, SSTDR1, SSTDR2 and SSTDR3 are valid) with CPOS = 0 and CPHS = 0



[1] Initial setting:
Specify the transmit data format.

[2] Check that the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit is 1. The TDRE bit is automatically cleared to 0 and transmission is started by writing data to SSTDR.

[3] Procedure for consecutive data transmission:
To continue data transmission, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

[4] Procedure for data transmission end:
To end data transmission, confirm that the TEND bit is cleared to 0. After completion of transmitting the last bit, clear the TE bit to 0.

Note: Hatching boxes represent SSU internal operations.

Figure 20.6 Flowchart Example of Data Transmission (SSU Mode)

(3) Data Reception

Figure 20.7 shows an example of reception operation, and figure 20.8 shows a flowchart example of data reception. When receiving data, the SSU operates as shown below.

After setting the RE bit to 1 and dummy-reading SSRDR, the SSU starts data reception.

In master mode, the SSU outputs a transfer clock and receives data. In slave mode, when a low level signal is input to the $\overline{\text{SCS}}$ pin and a transfer clock is input to the SSCK pin, the SSU receives data in synchronization with the transfer clock.

When 1-frame data has been received, the RDRF bit in SSSR is set to 1 and the receive data is stored in SSRDR. At this time, if the RIE bit in SSER is set to 1, an RXI interrupt is generated. The RDRF bit is automatically cleared to 0 by reading SSRDR.

When the RDRF bit has been set to 1 at the 8th rising edge of the transfer clock, the ORER bit in SSSR is set to 1. This indicates that an overrun error (OEI) has occurred. At this time, data reception is stopped. While the ORER bit in SSSR is set to 1, reception is not performed. To resume the reception, clear the ORER bit to 0.

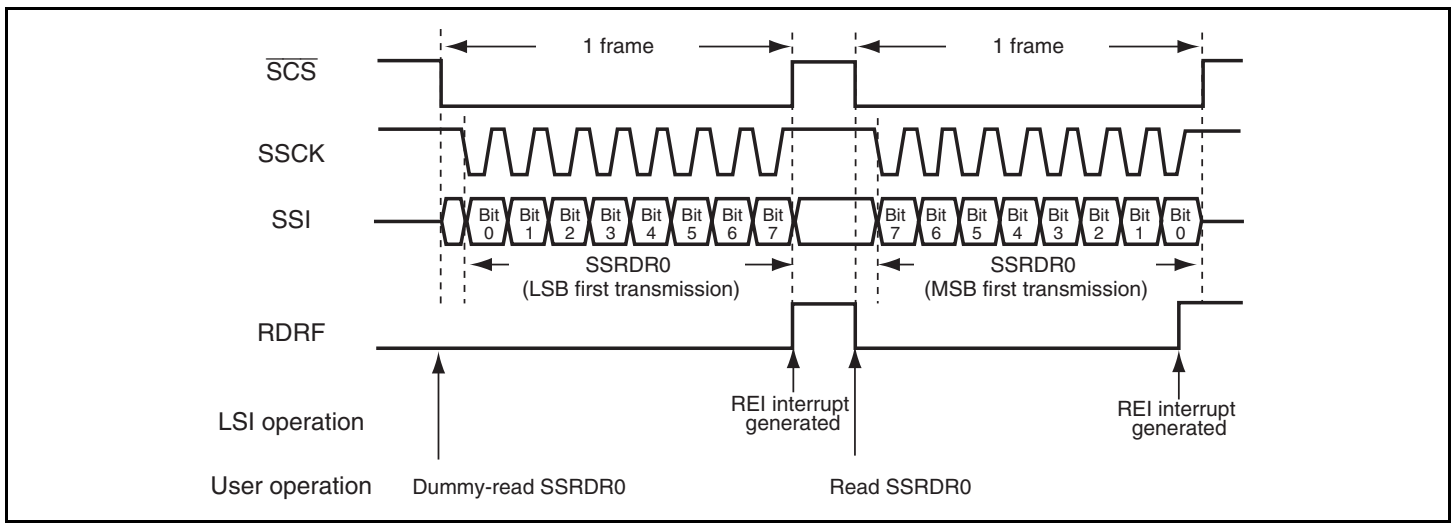


Figure 20.7 (1) Example of Reception Operation (SSU Mode)
When 8-bit data length is selected (SSRDR0 is valid) with CPOS = 0 and CPHS = 0

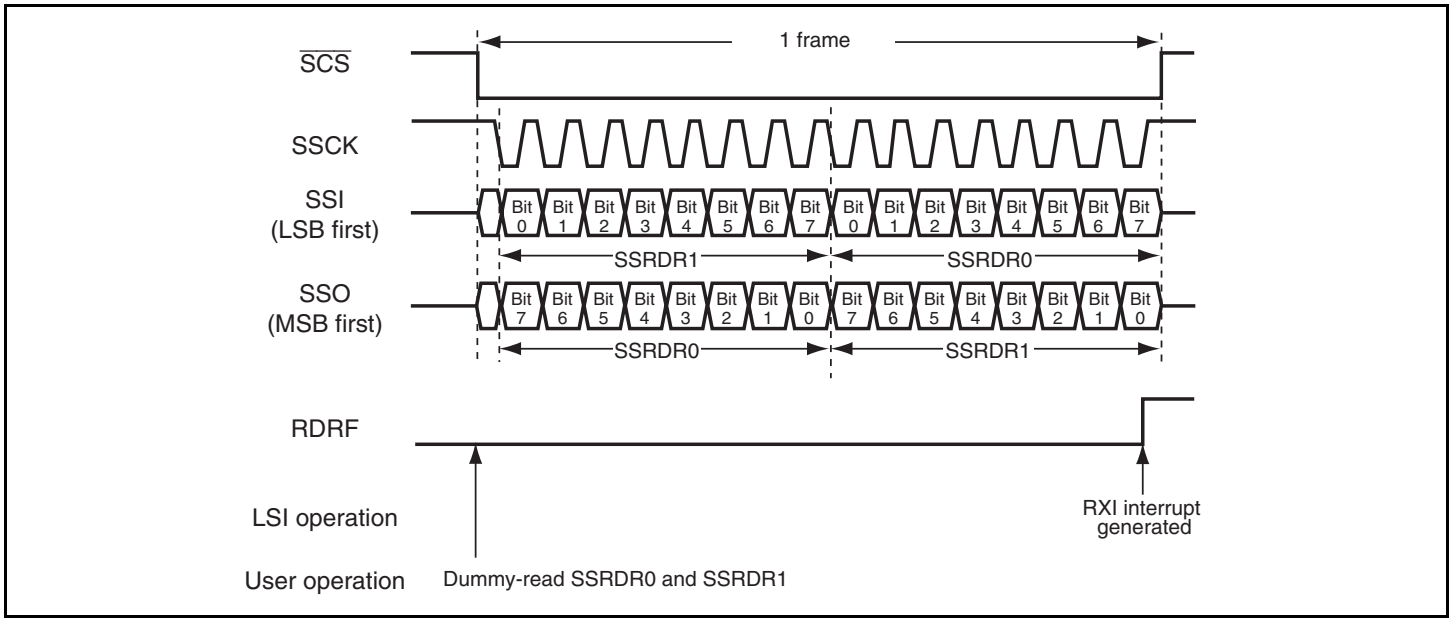


Figure 20.7 (2) Example of Reception Operation (SSU Mode) When 16-bit data length is selected (SSRDR0 and SSRDR1 are valid) with CPOS = 0 and CPHS = 0

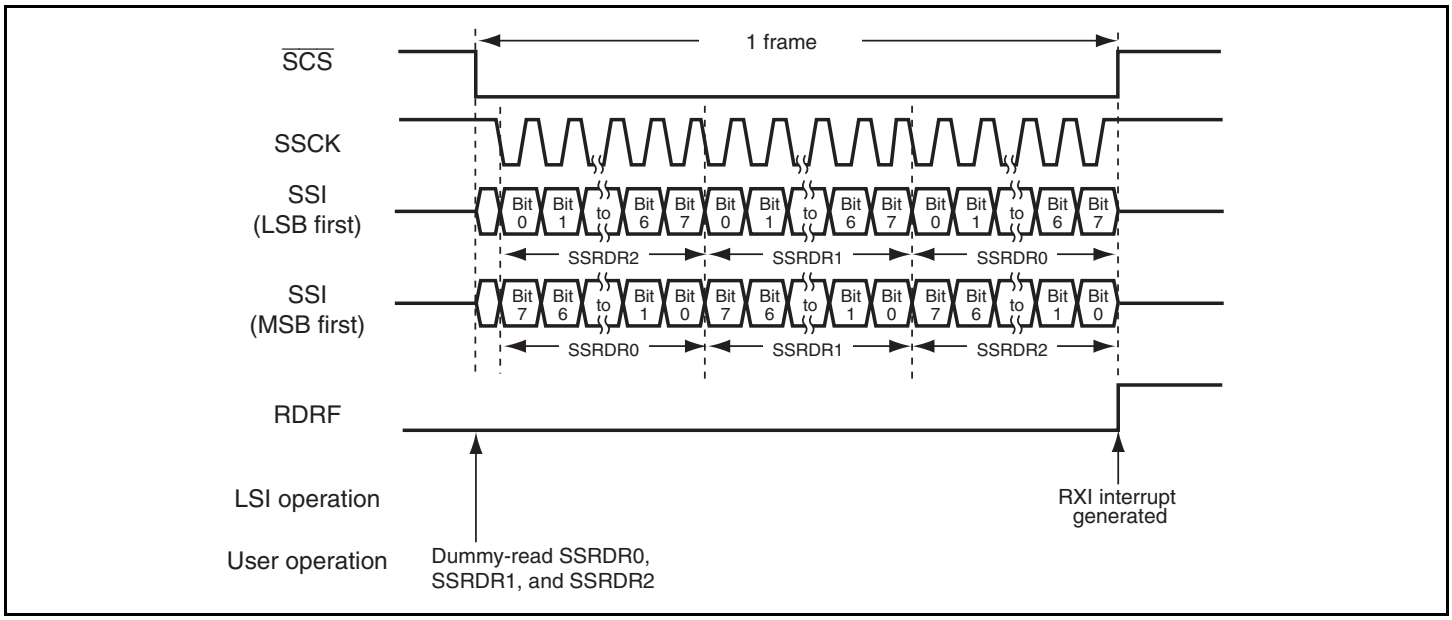


Figure 20.7 Example of Reception Operation (SSU Mode) When 24-bit data length is selected (SSRDR0, SSRDR1, and SSRDR2 are valid) with CPOS = 0 and CPHS = 0 (3)

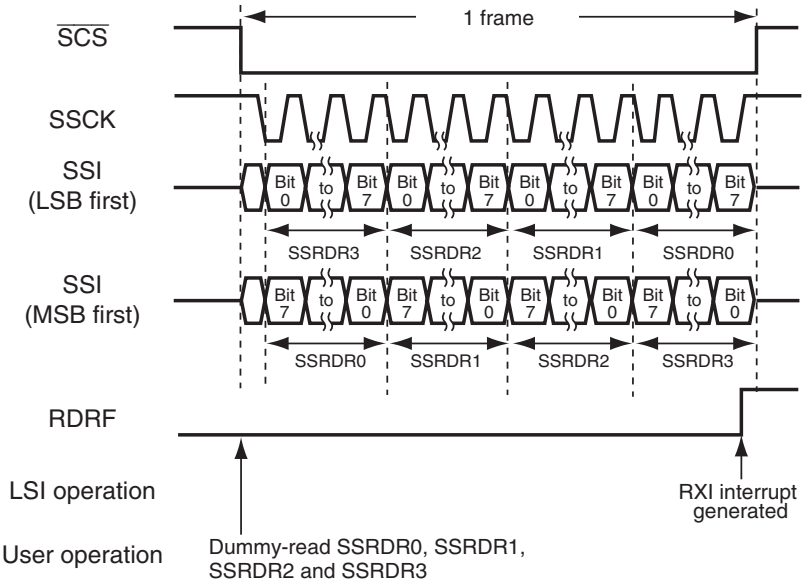
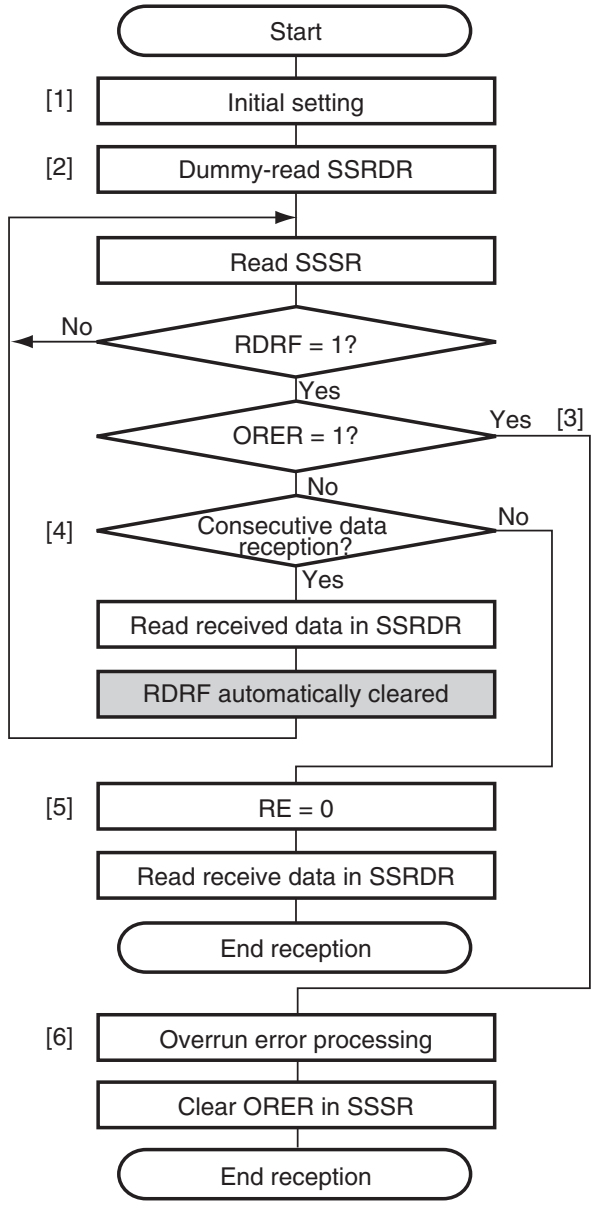


Figure 20.7 Example of Reception Operation (SSU Mode)
When 32-bit data length is selected (SSRDR0, SSRDR1, SSRDR2 and SSRDR3 are valid) with CPOS = 0 and CPHS = 0 (4)



[1] Initial setting:
Specify the receive data format.

[2] Start reception:
When SSRDR is dummy-read with RE = 1, reception is started.

[3], [6] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[4] To continue single reception:
When continuing single reception, wait for time of t_{SUcyc} while the RDRF flag is set to 1 and then read receive data in SSRDR.
The next single reception starts after reading receive data in SSRDR.

[5] To complete reception:
To complete reception, read receive data after clearing the RE bit to 0. When reading SSRDR without clearing the RE bit, reception is resumed.

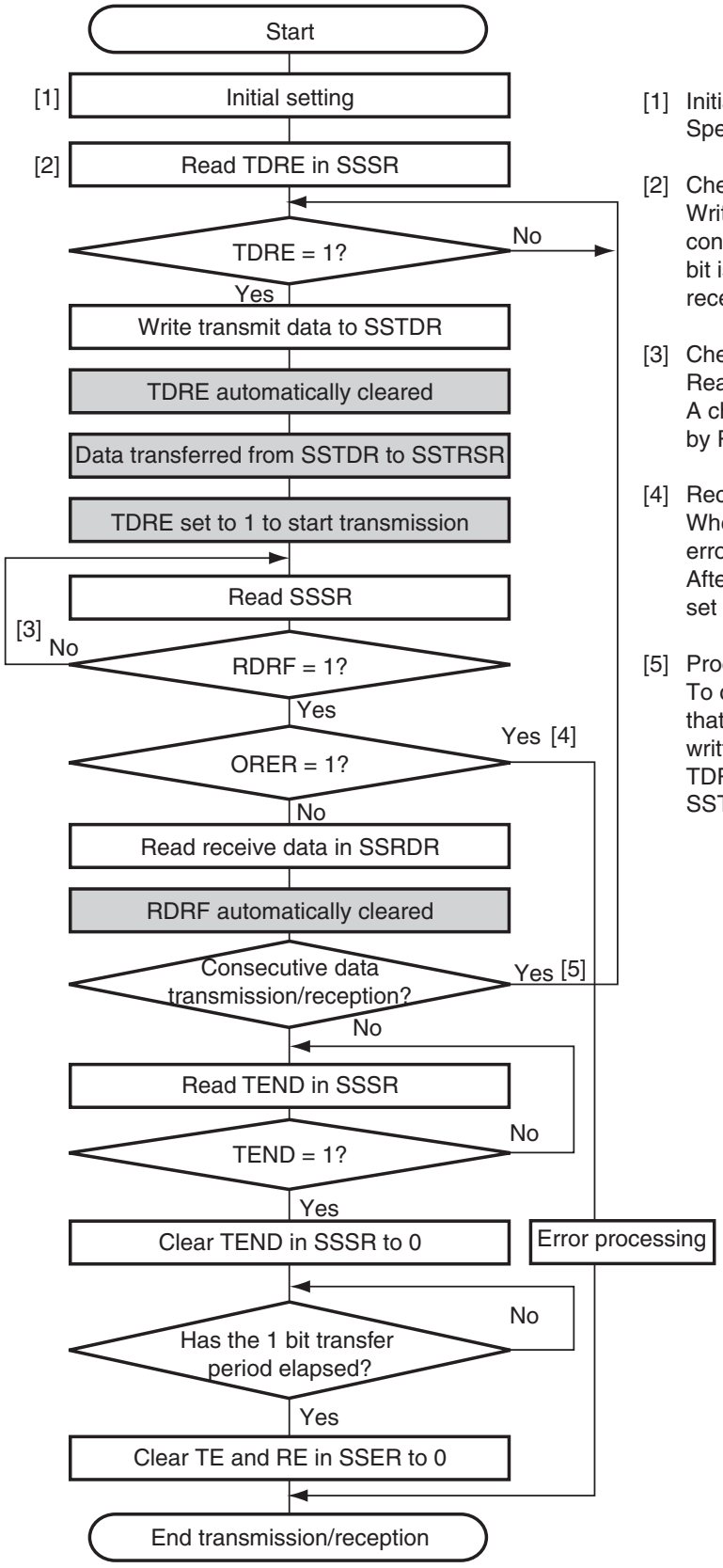
Note: Hatching boxes represent SSU internal operations.

Figure 20.8 Flowchart Example of Data Reception (SSU Mode)

(4) Data Transmission/Reception

Figure 20.9 shows a flowchart example of simultaneous transmission/reception. The data transmission/reception is performed combining the data transmission and data reception as mentioned above. The data transmission/reception is started by writing transmit data to SSTDR with TE = RE = 1.

Before switching transmission mode (TE = 1) or reception mode (RE = 1) to transmission/reception mode (TE = RE = 1), clear the TE and RE bits to 0. When starting the transfer, confirm that the TEND, RDRF, and ORER bits are cleared to 0 before setting the TE or RE bit to 1.



[1] Initial setting:
Specify the transmit/receive data format.

[2] Check the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit in SSSR is 1. The TDRE bit is automatically cleared to 0 and transmission/reception is started by writing data to SSTDR.

[3] Check the SSU state:
Read SSSR confirming that the RDRF bit is 1.
A change of the RDRF bit (from 0 to 1) can be notified by RXI interrupt.

[4] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[5] Procedure for consecutive data transmission/reception:
To continue serial data transmission/reception, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

Note: Hatching boxes represent SSU internal operations.

Figure 20.9 Flowchart Example of Simultaneous Transmission/Reception (SSU Mode)

20.4.6 $\overline{\text{SCS}}$ Pin Control and Conflict Error

When bits CSS1 and CSS0 in SSCRH are specified to B'10 and the SSUMS bit in SSCRL is cleared to 0, the $\overline{\text{SCS}}$ pin functions as an input (Hi-Z) to detect conflict error. The conflict detection period is from setting the MSS bit in SSCRH to 1 to starting serial transfer and after transfer ends. When a low level signal is input to the $\overline{\text{SCS}}$ pin within the period, a conflict error occurs. At this time, the CE bit in SSSR is set to 1 and the MSS bit is cleared to 0.

Note: While the CE bit is set to 1, transmission or reception is not resumed. Clear the CE bit to 0 before resuming the transmission or reception.

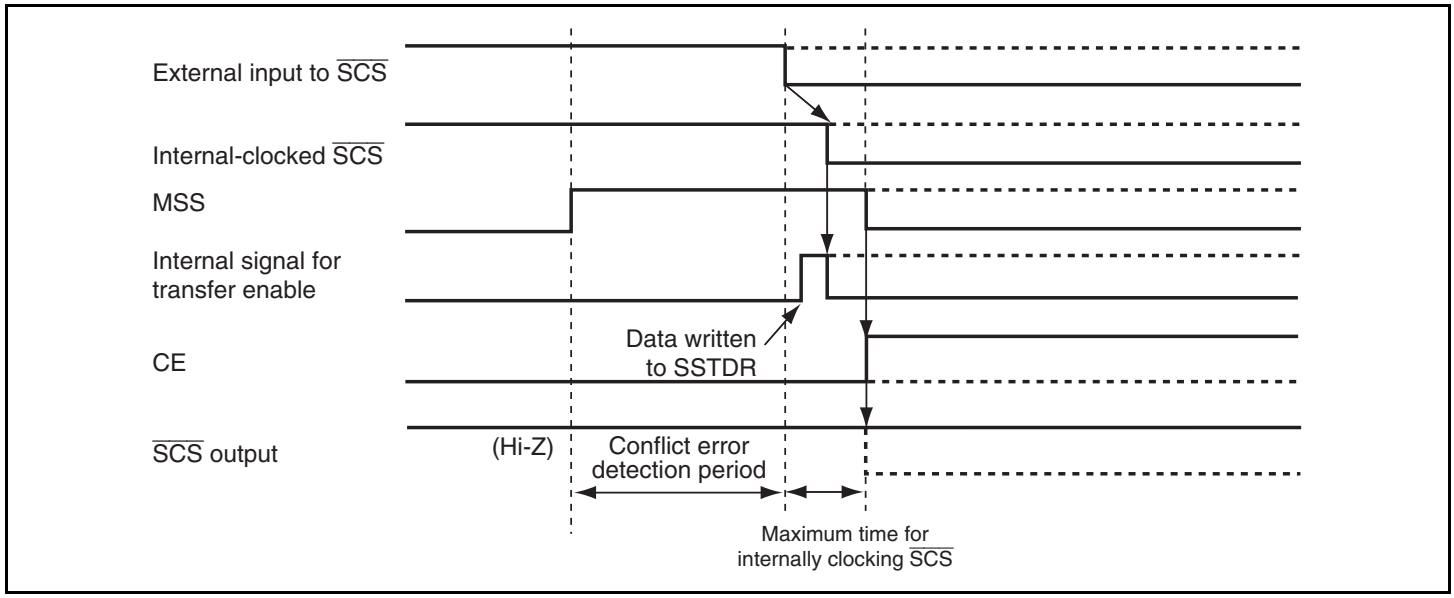


Figure 20.10 Conflict Error Detection Timing (Before Transfer)

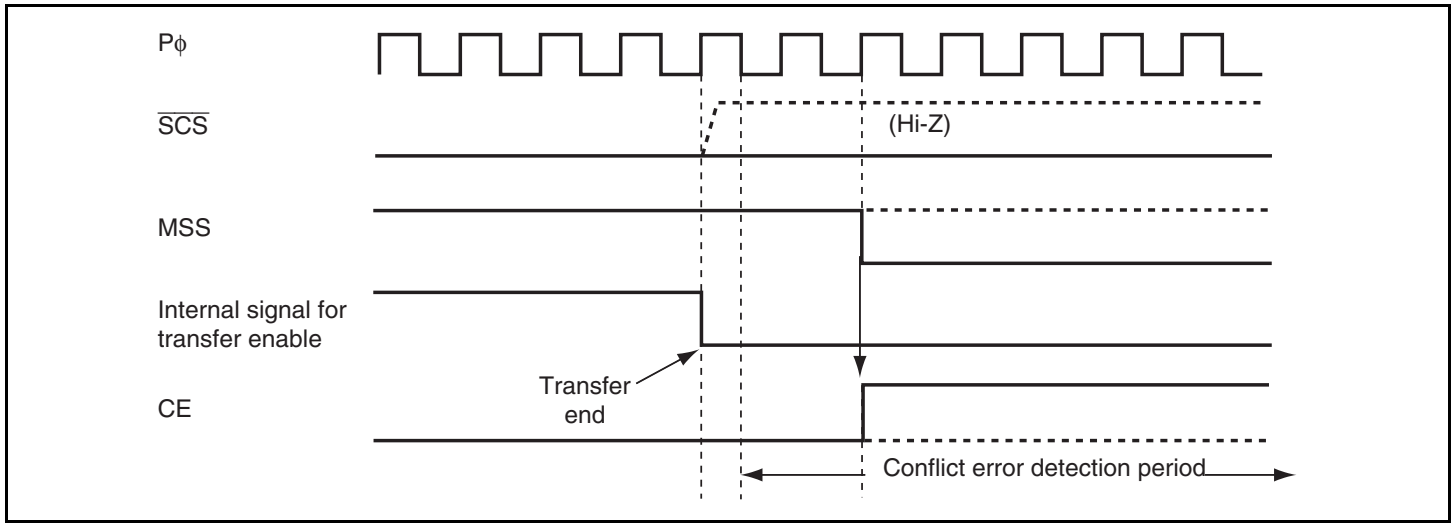


Figure 20.11 Conflict Error Detection Timing (After Transfer End)

20.4.7 Clock Synchronous Communication Mode

In clock synchronous communication mode, data communications are performed via three lines: clock line (SSCK), data input line (SSI), and data output line (SSO).

(1) Initial Settings in Clock Synchronous Communication Mode

Figure 20.12 shows an example of the initial settings in clock synchronous communication mode. Before data transfer, clear both the TE and RE bits in SSER to 0 to set the initial values.

Note: Before changing operating modes and communications formats, clear both the TE and RE bits to 0. Although clearing the TE bit to 0 sets the TDRE bit to 1, clearing the RE bit to 0 does not change the values of the RDRF and ORER bits and SSRDR. Those bits retain the previous values.

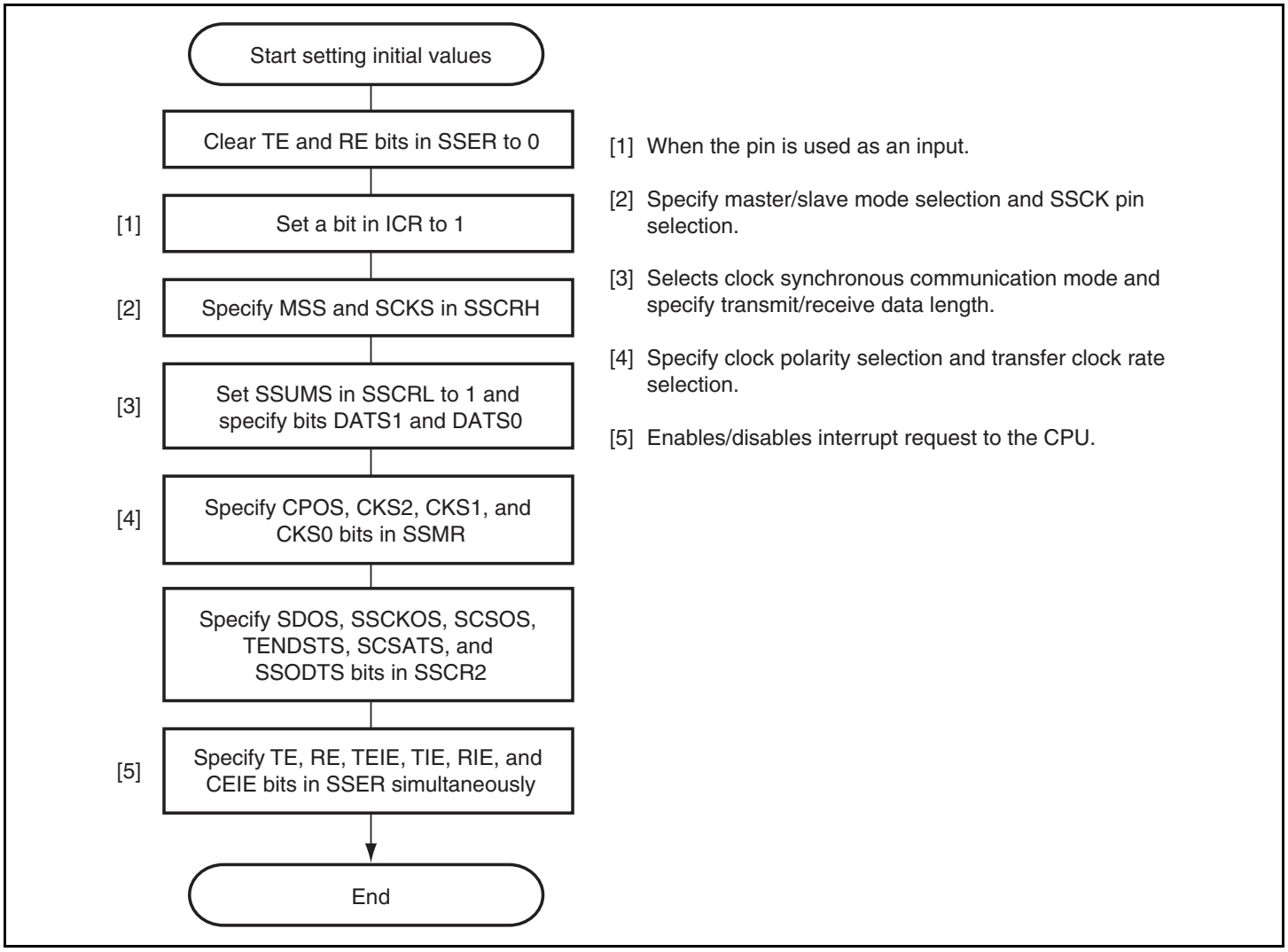


Figure 20.12 Example of Initial Settings in Clock Synchronous Communication Mode

(2) Data Transmission

Figure 20.13 shows an example of transmission operation, and figure 20.14 shows a flowchart example of data transmission. When transmitting data in clock synchronous communication mode, the SSU operates as shown below.

In master mode, the SSU outputs a transfer clock and data. In slave mode, when a transfer clock is input to the SSCK pin, the SSU outputs data in synchronization with the transfer clock.

Writing transmit data to SSTDR after the TE bit is set to 1 clears the TDRE bit in SSSR to 0, and the SSTDR contents are transferred to SSTRSR. After that, the SSU sets the TDRE bit to 1 and starts transmission. At this time, if the TIE bit in SSER is set to 1, a TXI interrupt is generated.

When 1-frame data has been transferred with TDRE = 0, the SSTDR contents are transferred to SSTRSR to start the next frame transmission. When the 8th bit of transmit data has been transferred with TDRE = 1, the TEND bit in SSSR is set to 1 and the state is retained. At this time, if the TEIE bit is set to 1, a TEI interrupt is generated.

While the ORER bit in SSSR is set to 1, transmission is not performed. Check that the ORER bit is cleared to 0.

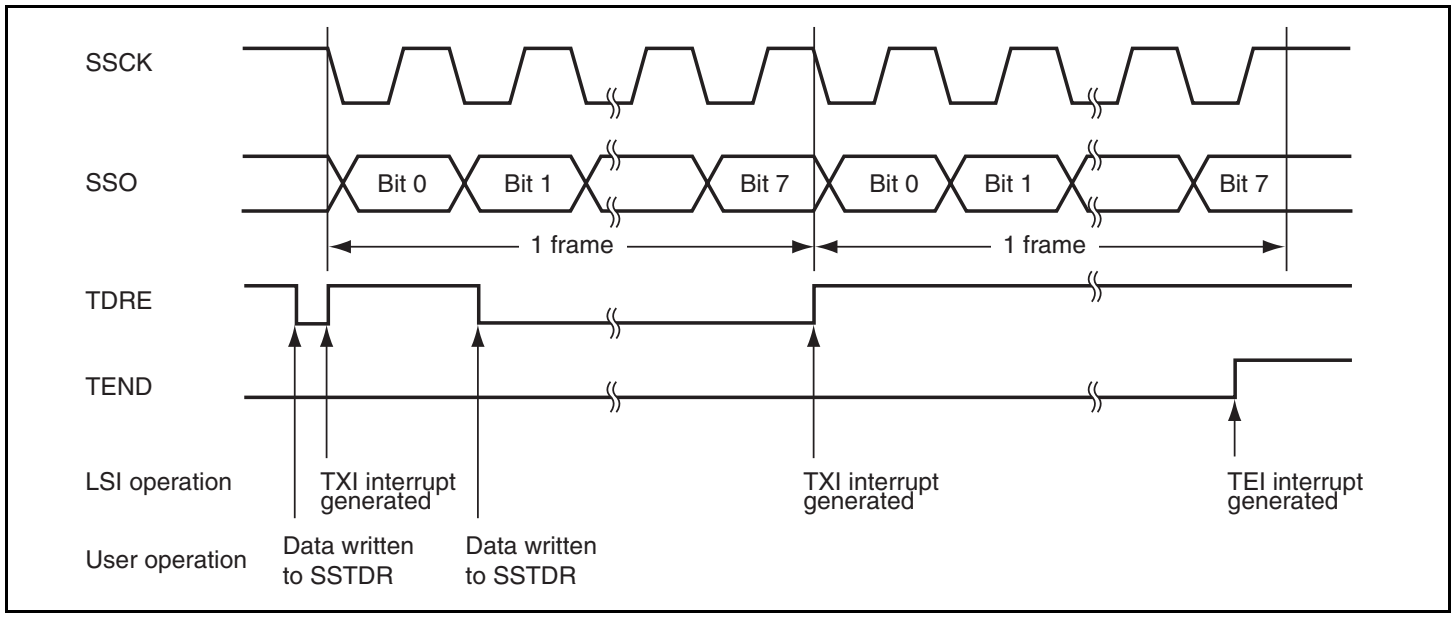
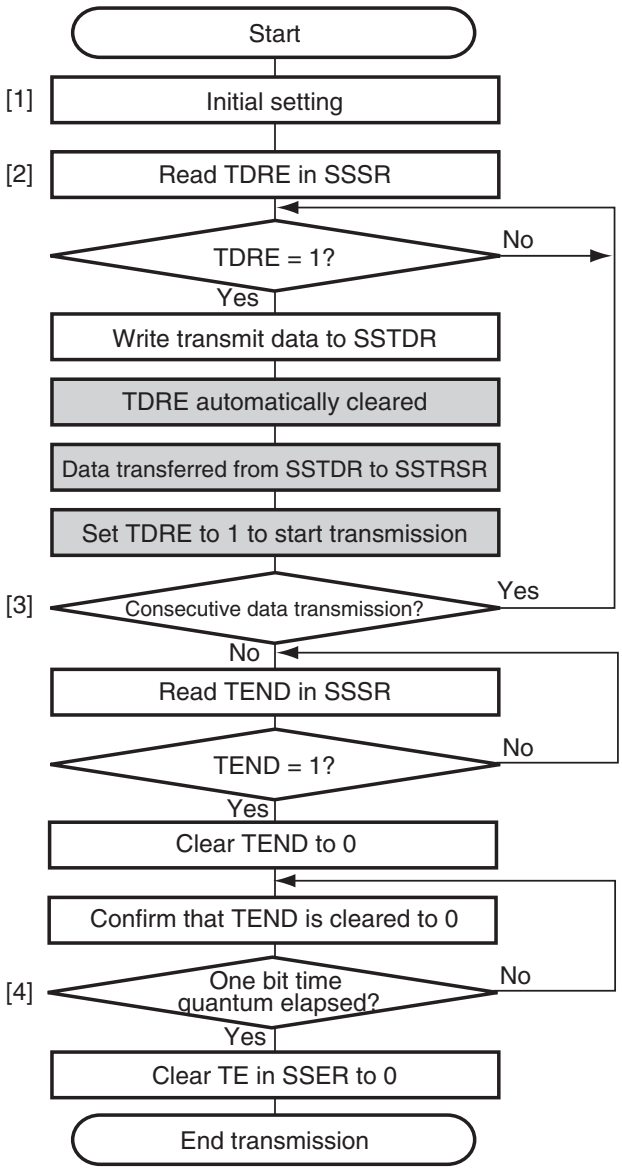


Figure 20.13 Example of Transmission Operation (Clock Synchronous Communication Mode)



[4][1] Initial setting:
Specify the transmit data format.

[2] Check that the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit is 1. The TDRE bit is automatically cleared to 0 and transmission is started by writing data to SSTDR.

[3] Procedure for consecutive data transmission:
To continue data transmission, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

[4] Procedure for data transmission end:
To end data transmission, confirm that the TEND bit is cleared to 0. After completion of transmitting the last bit, clear the TE bit to 0.

Note: Hatching boxes represent SSU internal operations.

Figure 20.14 Flowchart Example of Transmission Operation (Clock Synchronous Communication Mode)

(3) Data Reception

Figure 20.15 shows an example of reception operation, and figure 20.16 shows a flowchart example of data reception. When receiving data, the SSU operates as shown below.

After setting the RE bit in SSER to 1, the SSU starts data reception.

In master mode, the SSU outputs a transfer clock and receives data. In slave mode, when a transfer clock is input to the SSCK pin, the SSU receives data in synchronization with the transfer clock.

When 1-frame data has been received, the RDRF bit in SSSR is set to 1 and the receive data is stored in SSRDR. At this time, if the RIE bit is set to 1, an RXI interrupt is generated. The RDRF bit is automatically cleared to 0 by reading SSRDR.

When the RDRF bit has been set to 1 at the 8th rising edge of the transfer clock, the ORER bit in SSSR is set to 1. This indicates that an overrun error (OEI) has occurred. At this time, data reception is stopped. While the ORER bit in SSSR is set to 1, reception is not performed. To resume the reception, clear the ORER bit to 0.

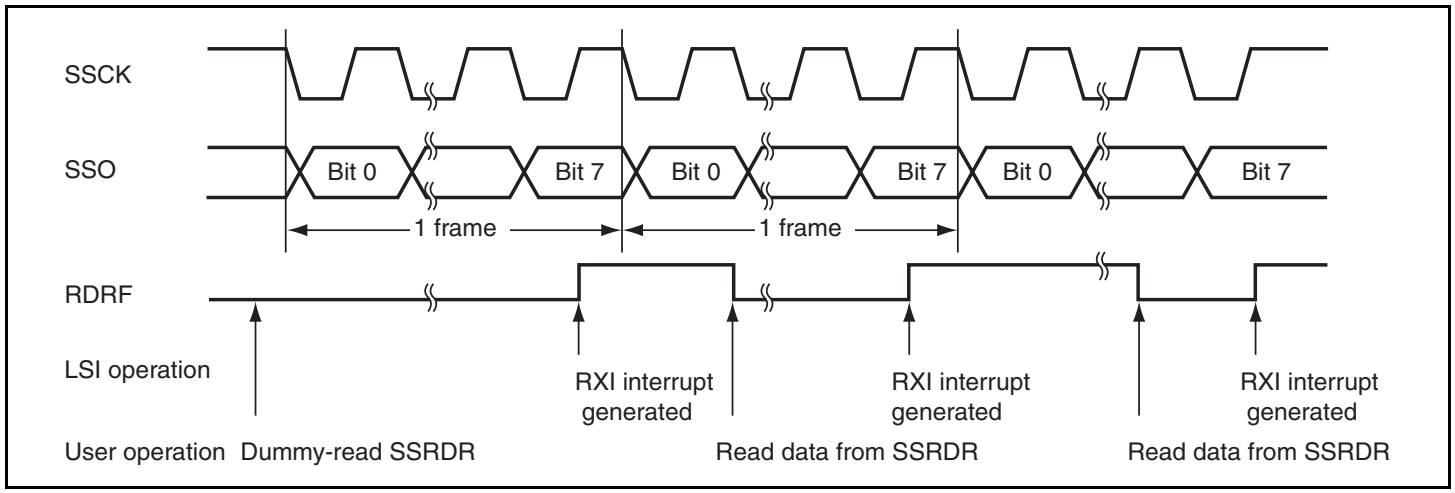
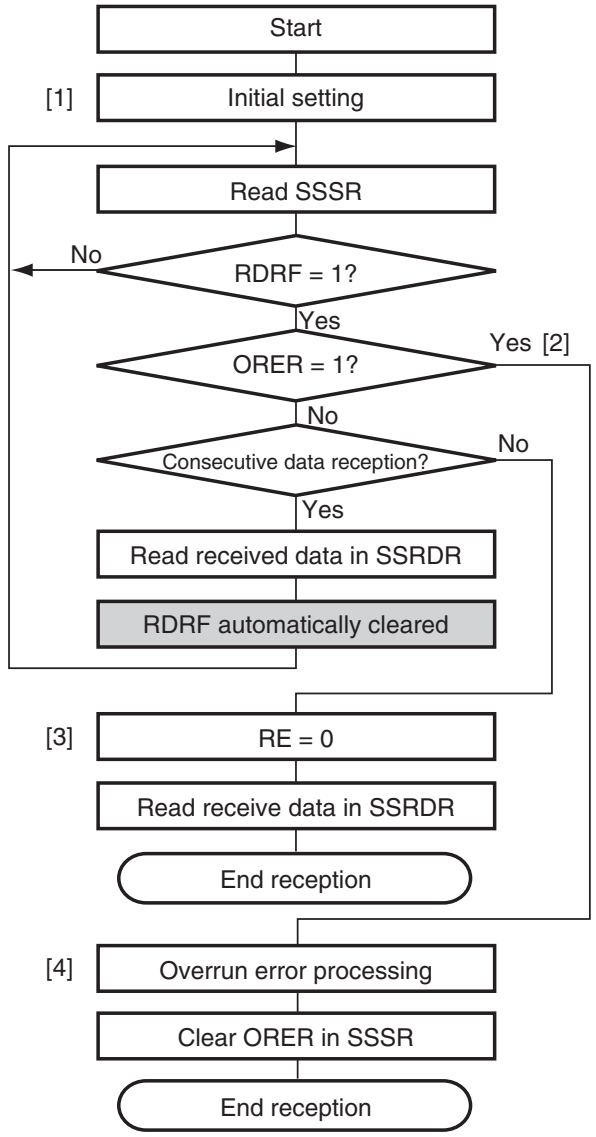


Figure 20.15 Example of Reception Operation (Clock Synchronous Communication Mode)



[1] Initial setting:
Specify the receive data format.

[2], [4] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[3] To complete reception:
To complete reception, read receive data after clearing the RE bit to 0. When reading SSRDR without clearing the RE bit, reception is resumed.

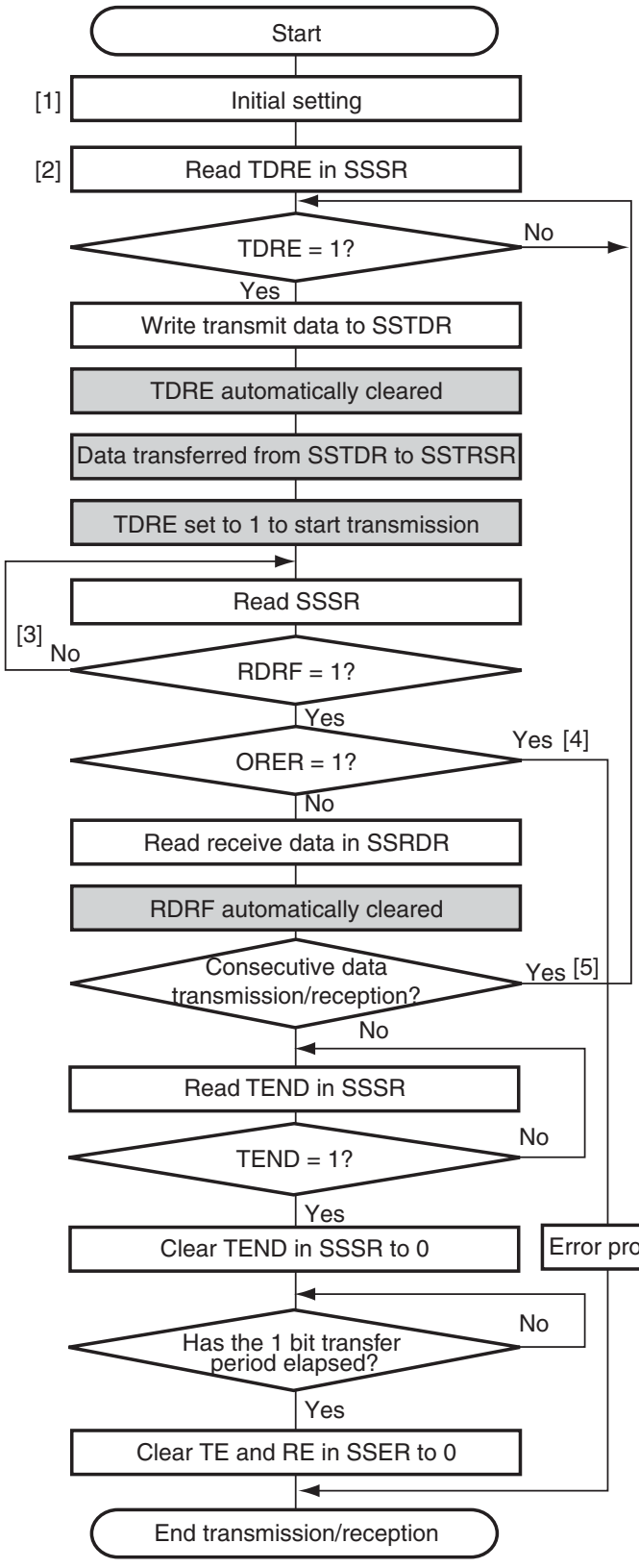
Note: Hatching boxes represent SSU internal operations.

Figure 20.16 Flowchart Example of Data Reception (Clock Synchronous Communication Mode)

(4) Data Transmission/Reception

Figure 20.17 shows a flowchart example of simultaneous transmission/reception. The data transmission/reception is performed combining the data transmission and data reception as mentioned above. The data transmission/reception is started by writing transmit data to SSTDR with TE = RE = 1.

Before switching transmission mode (TE = 1) or reception mode (RE = 1) to transmission/reception mode (TE = RE = 1), clear the TE and RE bits to 0. When starting the transfer, confirm that the TEND, RDRF, and ORER bits are cleared to 0 before setting the TE or RE bits to 1.



Note: Hatching boxes represent SSU internal operations.

[1] Initial setting:
Specify the transmit/receive data format.

[2] Check the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit in SSSR is 1. The TDRE bit is automatically cleared to 0 and transmission is started by writing data to SSTDR.

[3] Check the SSU state:
Read SSSR confirming that the RDRF bit is 1.
A change of the RDRF bit (from 0 to 1) can be notified by RXI interrupt.

[4] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[5] Procedure for consecutive data transmission/reception:
To continue serial data transmission/reception, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

Figure 20.17 Flowchart Example of Simultaneous Transmission/Reception (Clock Synchronous Communication Mode)

20.5 Interrupt Requests

The SSU interrupt requests are an overrun error, a conflict error, a receive data register full, transmit data register empty, and a transmit end interrupts.

Since both an overrun error and a conflict error interrupts are allocated to the SSERI vector address, and both a transmit data register empty and a transmit end interrupts are allocated to the SSTXI vector address, the interrupt source should be decided by their flags. Table 20.7 lists the interrupt sources.

When an interrupt condition shown in table 20.7 is satisfied, an interrupt is requested. Clear the interrupt source by CPU or DMAC data transfer.

Table 20.7 Interrupt Sources

Channel	Abbreviation	Interrupt Source	Symbol	Interrupt Condition	DMAC Activation
0	SSERI0	Overrun error	OEI0	(RIE = 1) • (ORER = 1)	—
		Conflict error	CEI0	(CEIE = 1) • (CE = 1)	—
	SSRXI0	Receive data register full	RXI0	(RIE = 1) • (RDRF = 1)	—
	SSTXI0	Transmit data register empty	TXI0	(TIE = 1) • (TDRE = 1)	—
		Transmit end	TEI0	(TEIE = 1) • (TEND = 1)	—

20.6 Usage Note

20.6.1 Setting of Module Stop Mode

The SSU can be enabled/disabled by setting the module stop control register setting and is disabled by the initial value. Canceling module stop mode enables to access the SSU register. For details, see section 25, Power-Down Modes.

Section 21 RAM

This LSI has an on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling one-state access by the CPU to both byte data and word data.

The on-chip RAM can be enabled or disabled by means of the RAME bit in the system control register (SYSCR). For details on the system control register (SYSCR), see section 3.2.2, System Control Register (SYSCR).

Product Type		ROM Type	RAM Capacity	RAM Address
H8S/24569	R4F24569	Flash memory version	64 Kbytes*	H'FEC000 to H'FFBFFF
H8S/24569R	R4F24569R			
H8S/24549	R4F24549			
H8S/24568	R4F24568		48 Kbytes	H'FF0000 to H'FFBFFF
H8S/24568R	R4F24568R			
H8S/24565	R4F24565			
H8S/24565R	R4F24565R			
H8S/24548	R4F24548			
H8S/24545	R4F24545			
H8S/24562	R4S24562	ROM-less version	64 Kbytes*	H'FEC000 to H'FFBFFF
H8S/24562R	R4S24562R			
H8S/24542	R4S24542			
H8S/24561	R4S24561		48 Kbytes	H'FF0000 to H'FFBFFF
H8S/24561R	R4S24561R			
H8S/24541	R4S24541			

Note: * In planning.

Section 22 Flash Memory

The flash memory in this LSI can be accessed in three programming modes: user programming mode, on-board programming mode, and programmer mode.

Table 22.1 gives an overview of the flash memory specifications (see section 1, Overview, for items that are not shown in table 22.1).

Table 22.1 Overview of Flash Memory Specifications

Item		Description
Flash memory programming modes		Three modes (user programming mode, boot mode, and programmer mode)
Erase block division	User ROM	See figure 22.1.
Programming method		Word or byte units*[1]
Erase method		Block units
Programming and erase control method		Programming and erasure are controlled by software commands
Commands		Six commands
Programming and erase count		100 times*[2]
Data retention		Ten years

Notes: 1. The flash memory can be programmed in byte units only in parallel I/O mode.

2. The programming and erase count determine the number of times the erase operation can be performed in each block.

   For example, if 1-word programming is done 2,048 times, each at a different address in a 4-Kbyte block and then the block is erased, this is counted as one erase count. If the allowed programming and erase count are 100 times, each block can be erased 100 times.

Table 22.2 Overview of Flash Memory Programming Modes

	Flash Memory Programming Mode		
Item	**User Programming Mode**	**On-board Programming Mode**	**Programmer Mode**
Functional overview	The user ROM is programmed by the CPU through execution of software commands. EW0 mode: Programming can be done from outside of the flash memory.	The user ROM is programmed through the on-chip SCI interface. Standard serial I/O mode 1: Clock-synchronous serial I/O Standard serial I/O mode 2: Asynchronous serial I/O	The user ROM is programmed through a dedicated parallel programmer.
Programmable area	User ROM, Data flash*	User ROM, Data flash*	User ROM, Data flash*
Operating mode	Single-chip mode, memory-expanded mode (EW0 mode)	Boot mode	Programmer mode
ROM programmer	—	—	Parallel programmer

Note: * Data flash is in planning.

22.1 Memory Map

This ROM is divided into the user ROM and the data flash.

Figure 22.1 shows a block diagram of the flash memory.

The user ROM and data flash are divided into multiple blocks. The user ROM can be programmed in user programming mode, on-board programming mode, or programmer mode.

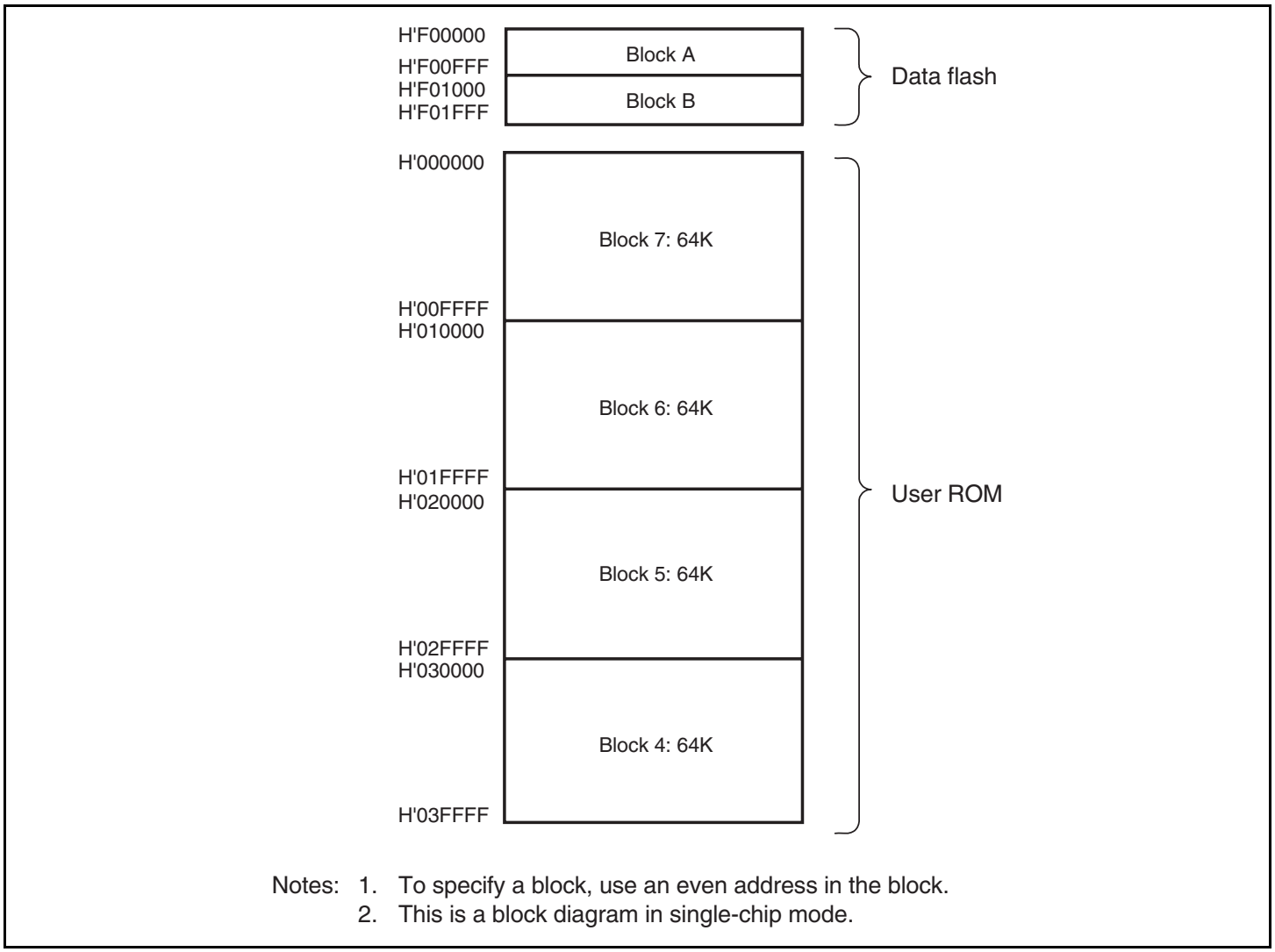


Figure 22.1 Block Diagram of Flash Memory

22.1.1 Boot Mode

Setting the mode pins to mode 3 and resetting the hardware shifts the flash memory into boot mode. In this mode, the embedded standard program is executed.

22.2 Register Descriptions

The flash memory has the following registers.

- Flash memory control register 1 (FLMCR1)
- Flash memory data block protect register (DFPR)
- Flash memory status register (FLMSTR)

22.2.1 Flash Memory Control Register 1 (FLMCR1)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved The initial value should not be changed.
6	CBIDB	1	R/W	CPU Programming Mode Select Setting this bit to 0 (CPU programming mode) enables command acceptance. 0: CPU programming mode enabled 1: CPU programming mode disabled
5	—	0	—	Reserved The initial value should not be changed.
4	—	0	—	Reserved The initial value should not be changed.
3	—	0	—	Reserved The initial value should not be changed.
2	—	1	—	Reserved The initial value should not be changed.
1	—	0	—	Reserved The initial value should not be changed.
0	FMCMDEN	0	R/W	Flash Memory Software Command Enable Setting this bit to 1 (CPU programming mode) enables command acceptance. 0: Flash memory software commands disabled 1: Flash memory software commands enabled To set this bit to 1, be sure to write 0 and then write 1 in a row.

Note: * To set the FMCMDEN bit to 1, write 0 to FMCMDEN and then write 1 in a row.

22.2.2 Flash Memory Data Block Protect Register (DFPR)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved The initial value should not be changed.
6	—	0	—	Reserved The initial value should not be changed.
5	—	0	—	Reserved The initial value should not be changed.
4	—	0	—	Reserved The initial value should not be changed.
3	—	0	—	Reserved The initial value should not be changed.
2	—	0	—	Reserved The initial value should not be changed.
1	—	0	—	Reserved The initial value should not be changed.
0	FMDBPT0	0	R/W	Data Flash E/W Protect* 0: Data flash E/W enabled 1: Data flash E/W disabled To set this bit to 1, be sure to write 0 and then write 1 in a row.

Note: * To set the FMDBPT0 bit to 1, set the FMCMDEN bit to 1.

22.2.3 Flash Memory Status Register (FLMSTR)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved The initial value should not be changed.
6	—	0	—	Reserved The initial value should not be changed.
5	FMERSF*	0	R	Erase or Blank Check Status Flag 0: Successfully completed 1: Ended with an error
4	FMERCF	0	R	Erase Suspend Flag 0: Other than erase-suspended state 1: Erase-suspended state
3	FMPRSF*	0	R	Program Status Flag 0: Successfully completed 1: Ended with an error
2	FMPRCF	0	R	Program Suspend Flag 0: Other than program-suspended state 1: Program-suspended state
1	—	1	—	Reserved The initial value should not be set.
0	FMRDY	1	R	Flash Memory Ready/Busy Status 0: Busy (Interrupt processing or erasure is in progress.) 1: Ready

Note: * The FMERSF and FMPRSF bits are cleared to 0 by a clear status command.

22.3 On-Board Programming Mode

When the mode pins (MD0, MD1, and MD2) are set to on-board programming mode and the reset start is executed, a transition is made to on-board programming mode in which the on-chip flash memory can be programmed/erased. On-board programming mode has three operating modes: SCI boot mode by P27 and P26 settings, USB boot mode, and user program mode.

Table 22.3 shows the pin setting for each operating mode.

Table 22.3 On-Board Programming Mode Setting

Mode Setting	EMLE	MD2	MD1	MD0	P27	P26	Input clock frequencies for oscillators
SCI boot mode	0	0	1	1	0	0	
USB boot mode		0	1	1	0	1	16MHz
		0	1	1	1	0	12MHz
		0	1	1	1	1	8MHz
User programming mode		Mode 4, Mode 7			—	—	

22.3.1 SCI Boot Mode

SCI boot mode executes programming/erasing of the user MAT by means of the control command and program data transmitted from the externally connected host via the on-chip SCI_1.

In SCI boot mode, the tool for transmitting the control command and program data, and the program data must be prepared in the host. The serial communication mode is set to asynchronous mode. The system configuration in SCI boot mode is shown in figure 22.2. Interrupts are ignored in SCI boot mode. Configure the user system so that interrupts do not occur.

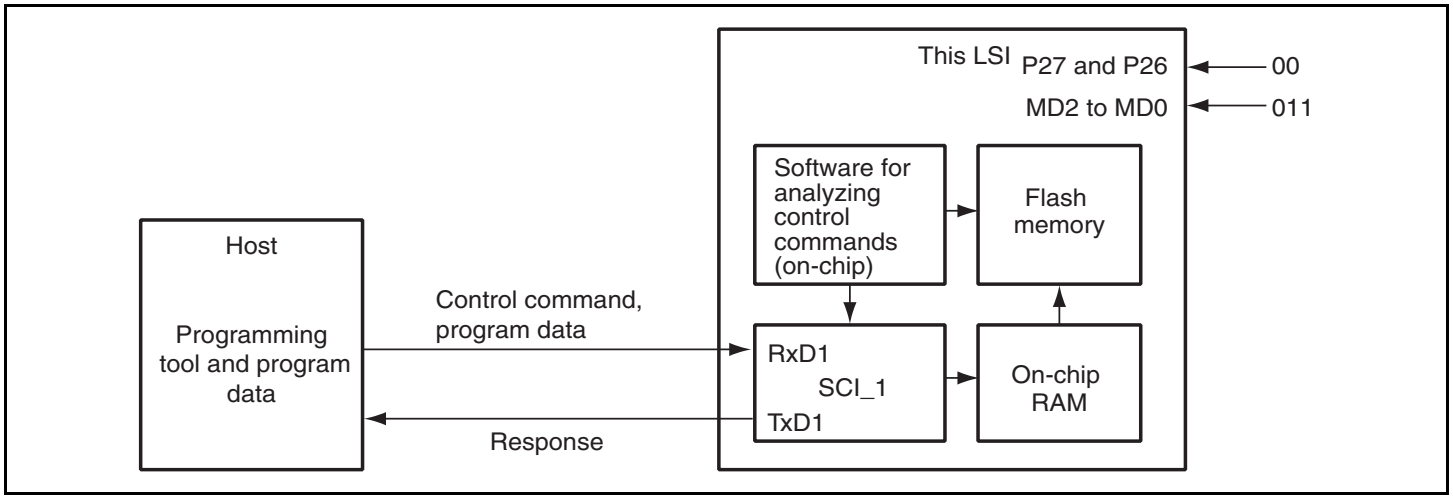


Figure 22.2 System Configuration in SCI Boot Mode

22.3.2 USB Boot Mode

USB boot mode executes programming/erasing of the user MAT by means of the control command and program data transmitted from the externally connected host via the USB.

In USB boot mode, the tool for transmitting the control command and program data, and the program data must be prepared in the host. The system configuration in USB boot mode is shown in figure 22.3. Interrupts are ignored in USB boot mode. Configure the user system so that interrupts do not occur.

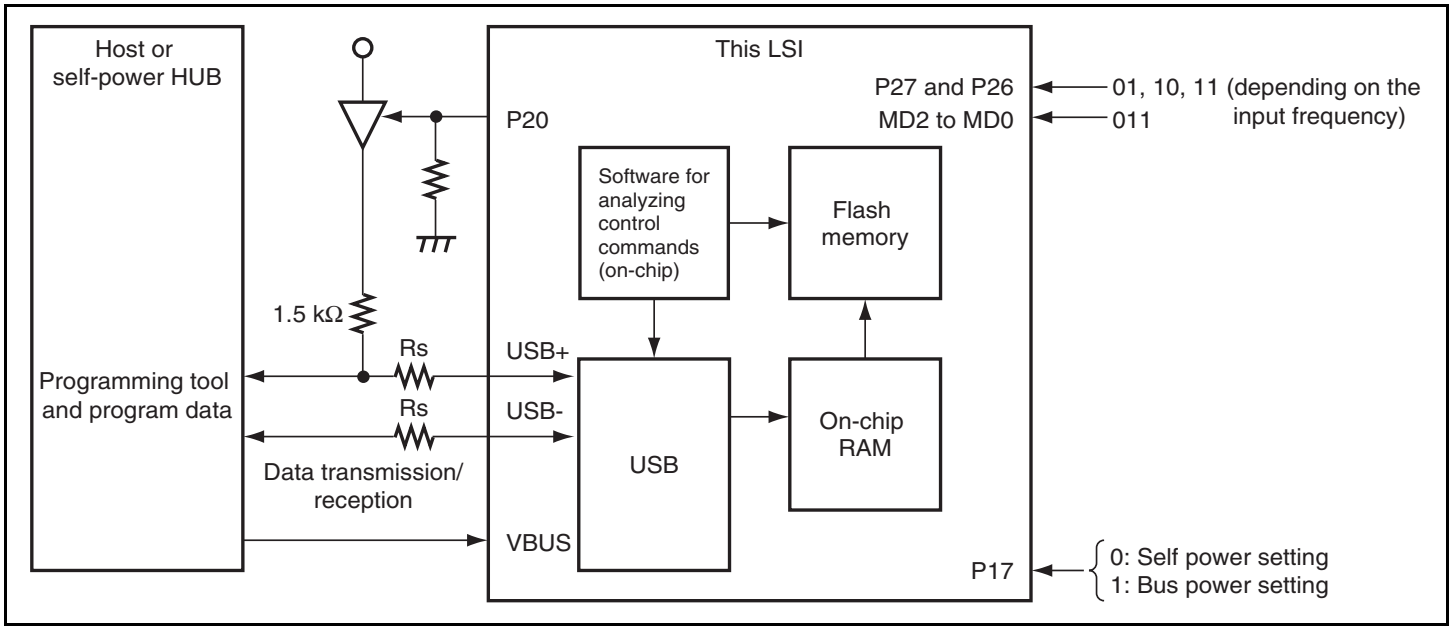


Figure 22.3 System Configuration in USB Boot Mode

(1) Features

- Bus power mode and self-power mode are selectable.
- The P20 pin supports the D+ pull-up control connection.
- For enumeration information, refer to table 22.4.

Table 22.4 Enumeration Information

USB standard	Ver.2.0 (Full speed)	
Transfer mode	Transfer mode Control (in, out), Bulk (in, out)	
Maximum power consumption	For self power mode (PM3 = 0)	100 mA
	For bus power mode (PM3 = 1)	500 mA
Endpoint configuration	EP0 Control (in out) 8 bytes Configuration 1 └ InterfaceNumber0 └ AlternateSetting0 ├ EP1 Bulk (out) 64 bytes └ EP2 Bulk (in) 64 bytes	

(2) State Transition Diagram

The state transition after USB boot mode is initiated is shown in figure 22.4.

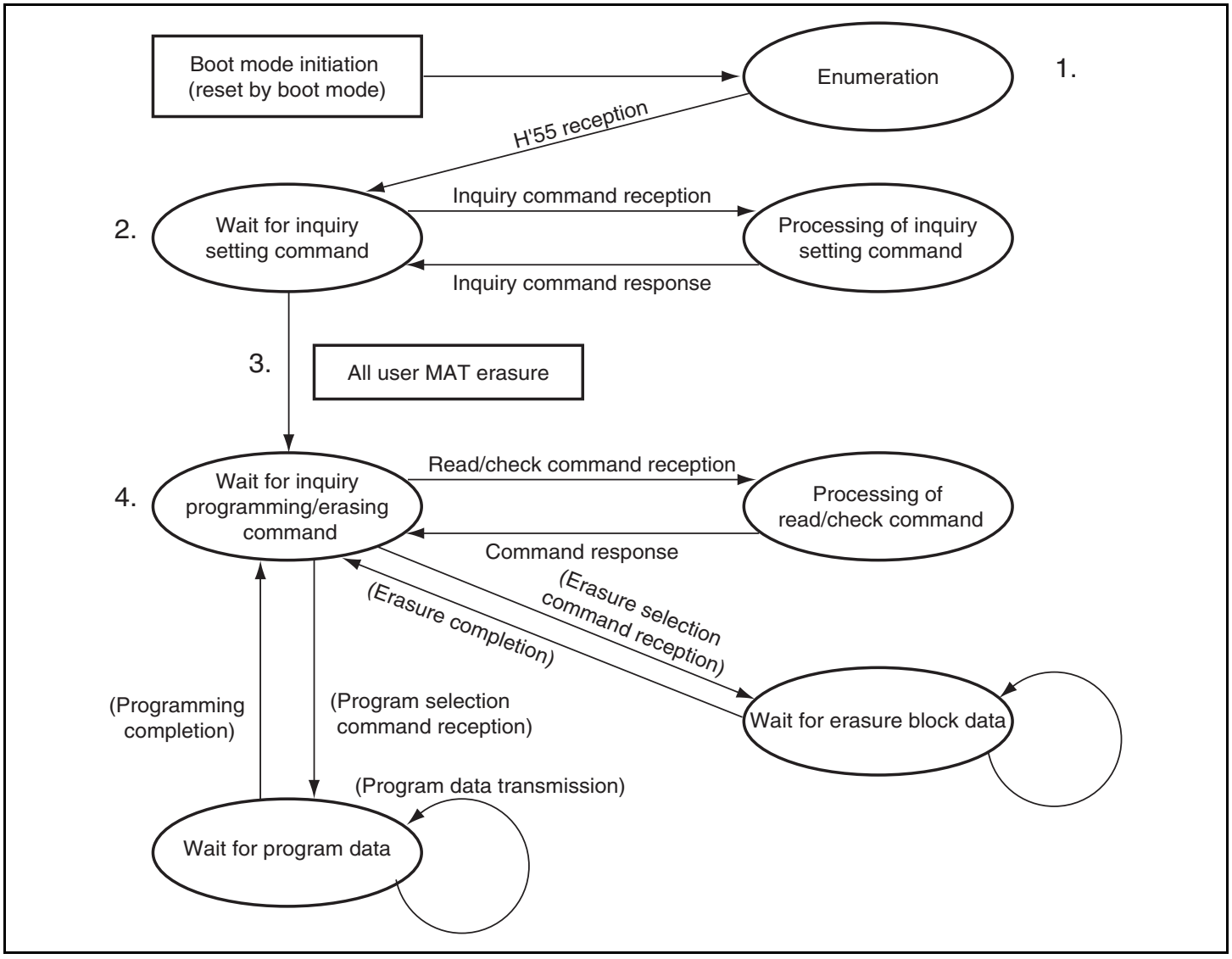


Figure 22.4 USB Boot Mode State Transition Diagram

1. After a transition to the USB boot mode is made, the boot program embedded in this LSI is initialized. This LSI performs enumeration to the host after the USB boot program is initialized.
2. Inquiry information about the size, configuration, start address, and support status of the user MAT is transmitted to the host.
3. After inquiries have finished, all user MAT are automatically erased.
4. After all user MAT are automatically erased, the state of waiting for programming/erasing command is entered. When the programming command is received, the state shifts to the state of waiting for programming data. The same applies to erasing. In addition to the commands for programming/erasing, there are commands for performing sum check, blank check (erasure check), and memory read of the user MAT, and acquiring the current status information.

(3) Notes on USB Boot Mode Execution

- The clock of 48 MHz needs to be supplied to the USB module. Set the external clock frequency and clock pulse generator so as to supply 48 MHz as the clock for the USB (cku). For details, refer to section 24, Clock Pulse Generator.
- Use the P20 pin for the D+ pull-up control connection.
- For the stable supply of the power during the flash memory programming and erasing, the cable should not be connected via the bus powered HUB.
- If the bus powered HUB is disconnected during the flash memory programming and erasing, permanent damage to the LSI may result.
- If the USB bus in the bus power mode enters the suspend mode, this does not make the transition to the software standby mode of the power-down mode.

22.3.3 User Programming Mode

In the user programming mode, the flash memory can be programmed by the CPU through execution of software commands. In this mode, the user ROM can be programmed without using a ROM programmer with the microcomputer mounted on a system board.

The programming and block erase commands should be executed only in each block area of the user program.

The user programming mode provides the erase/write 0 mode (EW0 mode). Table 22.5 gives an overview of the EW0 mode specifications.

Table 22.5 EW0 Mode Specifications

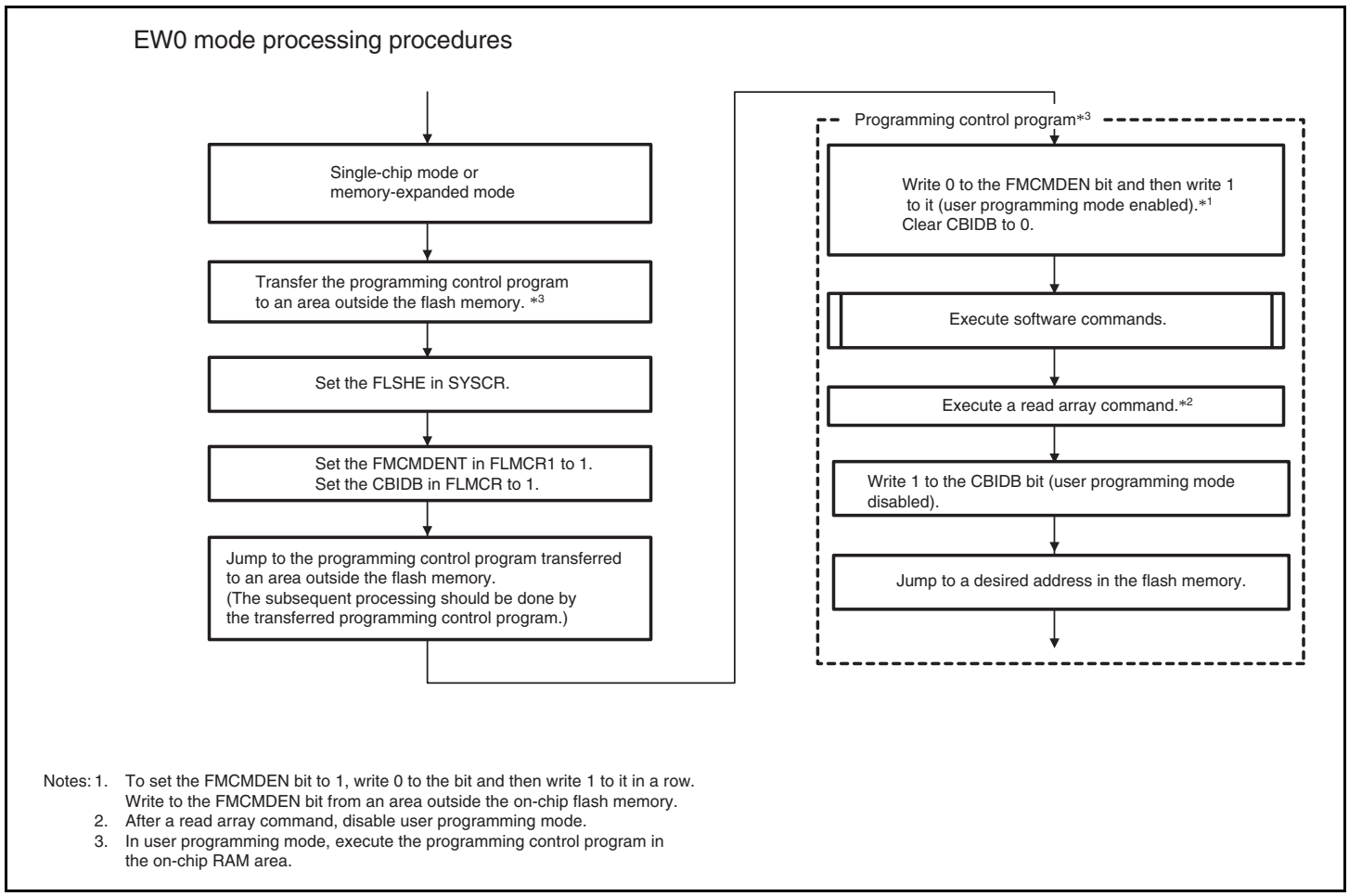
Item	Description
Operating mode	• Single-chip mode • Memory-expanded mode
Area for storing the programming control program	User ROM
Area for executing the programming control program	The programming control program should be transferred to an area outside the flash memory (such as RAM) before execution*[2]
Programmable area	User ROM
Limitations on software commands	None
Mode after programming or erasure	Read status register mode
CPU state during automatic programming or erasure	Operating*[1]
Flash memory status detection	• Reading the FMPRSF and FMERSF bits in FLMSTR by a program. • Executing a read status register command to read the SR7, SR5, and SR4 bits in the status register.

Notes: 1. Make sure that no interrupt (except NMI) or DMA transfer is generated.
2. In the user programming mode, the programming control program should be executed in the on-chip RAM or an external area.

22.3.4 EW0 Mode

Setting the FMCMDEN bit in FLMCR1 to 1 shifts the flash memory into the user programming mode, in which commands can be accepted. Figure 22.5 shows how to set and clear the EW0 mode.

Programming and erasure are controlled through software commands. The flash memory state after programming or erasure can be checked through FLMSTR or the status register.



Notes: 1. To set the FMCMDEN bit to 1, write 0 to the bit and then write 1 to it in a row. Write to the FMCMDEN bit from an area outside the on-chip flash memory.
2. After a read array command, disable user programming mode.
3. In user programming mode, execute the programming control program in the on-chip RAM area.

Figure 22.5 Setting and Clearing EW0 Mode

22.4 Notes on User Programming Mode

22.4.1 Prohibited Interrupts (EW0 Mode)

- The NMI and watchdog timer interrupts can be used because FLMCR1 is forcibly initialized when an interrupt is generated; specify the destination address of each interrupt routine in the fixed vector table. Flash memory programming is terminated when an NMI interrupt or a watchdog timer interrupt occurs. In this case, reexecute the programming program after the interrupt routine is completed.
- The address-match interrupt cannot be used because the interrupt processing accesses data in the flash memory.

22.4.2 Access Method

To set the FMCMDEN bit to 1, be sure to write 0 to the bit and then write 1 in a row. Make sure that no interrupt, EXDMAC transfer, DTC transfer, or DMA transfer is generated between writing 0 and 1.

22.4.3 Programming (EW0 Mode)

If the power-supply voltage falls during programming of the block that stores the programming control program, the programming control program cannot be correctly modified and the flash memory may not be programmed after that. In this case, use the on-board programming mode or programmer mode instead.

22.4.4 Writing Commands or Data

The address to write a command code or data should be a multiple of four (0, 4, 8, C, ...).

22.4.5 Software Standby Mode

Before entering the stop mode, set the FMCMDEN bit to 0 (CPU programming mode disabled), disable the DMA transfer, and then make a transition to the software standby mode.

22.5 Software Commands

The following describes the software commands. A command or data should be read or written in 16-bit units at an even address in the user ROM or data flash area. When a command code is written, the upper eight bits (D15 to D8) are ignored.

Table 22.6 List of Software Commands

	First Bus Cycle			Second Bus Cycle			Third Bus Cycle		
Software Command	**Mode**	**Address**	**Data (D15 to D0)**	**Mode**	**Address**	**Data (D15 to D0)**	**Mode**	**Address**	**Data (D15 to D0)**
Read array	Write	×	H'xxFF						
Read status register	Write	×	H'xx70	Read	×	SRD			
Clear status register	Write	×	H'xx50						
Program	Write	WA0	H'xx41	Write	WA0	WD0	Write	WA1	WD1
Block erase	Write	×	H'xx20	Write	BA	H'xxD0			
Block blank check	Write	×	H'Xx25	Write	BA	H'xxD0			

Legend:

SRD: Status register data (D7 to D0)
WA0: Address to write the lower word (the address for the first bus cycle must be the same even address as that for the second bus cycle).
WA1: Address to write the upper word
WD0: Lower word of write data (16 bits)
WD1: Upper word of write data (16 bits)
BA: Highest address of the block (note that this should be an even address).
×: A desired even address in program ROM1, program ROM2, or data flash.
xx: Upper eight bits of command code (ignored)

22.5.1 Read Array

This command reads the flash memory.

Write H'xxFF in the first bus cycle to shift the flash memory into the read array mode. Specify the target read address in the next bus cycle, and data is read from the address in 16-bit units.

As the flash memory stays in the read array mode until another command is issued, multiple addresses can be read in sequence.

22.5.2 Read Status Register

This command reads the status register.

Write H'xx70 in the first bus cycle, and the status register can be read in the second bus cycle (see section 22.6, Status Register). Specify an even address in the program ROM or data flash to read the status register.

Do not issue this command in the EW1 mode.

22.5.3 Clear Status Register

This command clears the status register.

Write H'xx50 in the first bus cycle, and the FMERSF and FMPRSF bits in FLMSTR are cleared to 0.

22.5.4 Program

This command writes data to the flash memory in 2-word (4-byte) units.

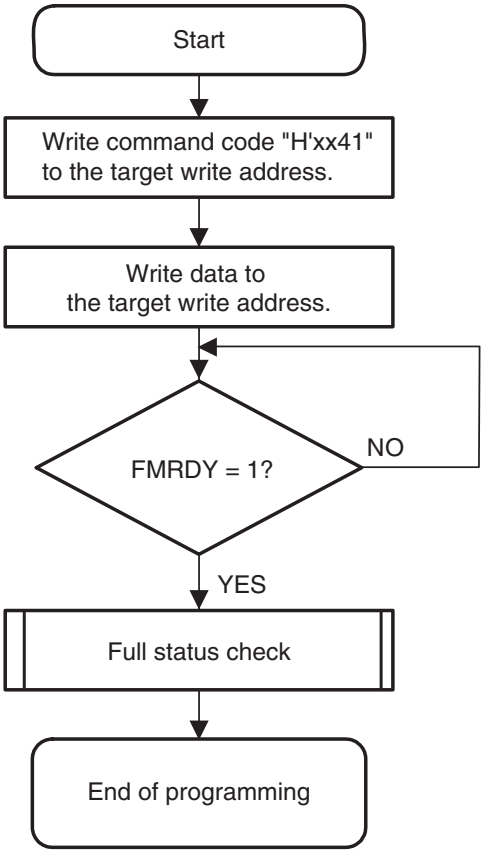
Write H'xx41 in the first bus cycle and write data to the target address in the second and third bus cycles; the flash memory starts automatic writing (programming and verifying data). The address value specified in the first bus cycle should be the same even address as that specified in the second bus cycle.

Completion of automatic writing can be checked through the FMRDY bit in FLMSTR. The FMRDY bit is 0 (busy) during automatic writing and becomes 1 (ready) when writing is completed.

After automatic writing is completed, the result can be checked through the FMPRSF bit in FMRSTR (see section 22.7, Full Status Check).

Once an address is programmed, no additional data can be written to the address. Figure 22.6 shows a flowchart of the program command processing.

In the EW0 mode, the read status register mode is entered as soon as automatic writing starts, and the status register can be read. The SR7 bit in the status register becomes 0 when automatic writing starts and returns to 1 when writing is completed. In this case, the flash memory stays in the read status register mode until a read array command is issued. After automatic writing is completed, the result of writing can be checked by reading the status register.



Note: Write the command code and data to even addresses.

Figure 22.6 Flowchart of Program Command Processing

22.5.5 Block Erase

Write H'xx20 in the first bus cycle and H'xxD0 to the highest address (an even address) of the target block in the second cycle; automatic erasure (erasing data and verifying the erased status) starts in the specified block.

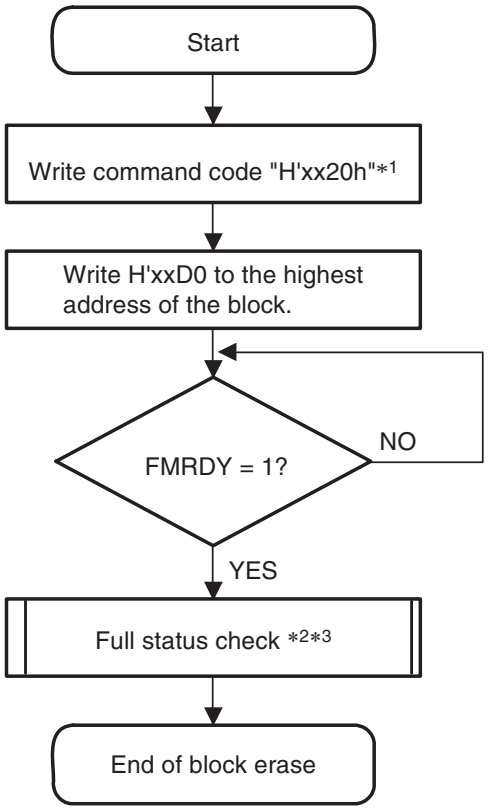
Completion of automatic erasure can be checked through the FMRDY bit in FLMSTR.

The FMRDY bit is 0 (busy) during automatic erasure and becomes 1 (ready) when erasure is completed.

After automatic erasure is completed, the result can be checked through the FMERSF bit in FLMSTR (see section 22.7, Full Status Check).

Figure 22.7 shows a flowchart of the block erase command processing.

In the EW0 mode, the read status register mode is entered as soon as automatic erasure starts, and the status register can be read. The SR7 bit in the status register becomes 0 when automatic erasure starts and returns to 1 when erasure is completed. In this case, the flash memory stays in the read status register mode until a read array command is issued. If an erase error occurs, repeat a sequence of the clear status register command to block erase command at least three times until no erase error occurs.



Notes:1. Write the command code and data to even addresses.
2. See figure 22.8.
3. If an erase error occurs, repeat a sequence of the clear status register command -> block erase command at least three times until no erase error occurs.

Figure 22.7 Flowchart of Block Erase Command Processing

22.5.6 Block Blank Check

This command checks if a block is blank (the erased state).

Write H'xx25 in the first bus cycle and H'xxD0 to the highest address (an even address) of the target block in the second cycle; the check result will be stored in the FMERSF bit in FLMSTR. After the FMRDY bit in FLMSTR has become 1 (ready), read the FMERSF bit.

Figure 22.8 shows a flowchart of the block blank check command processing.

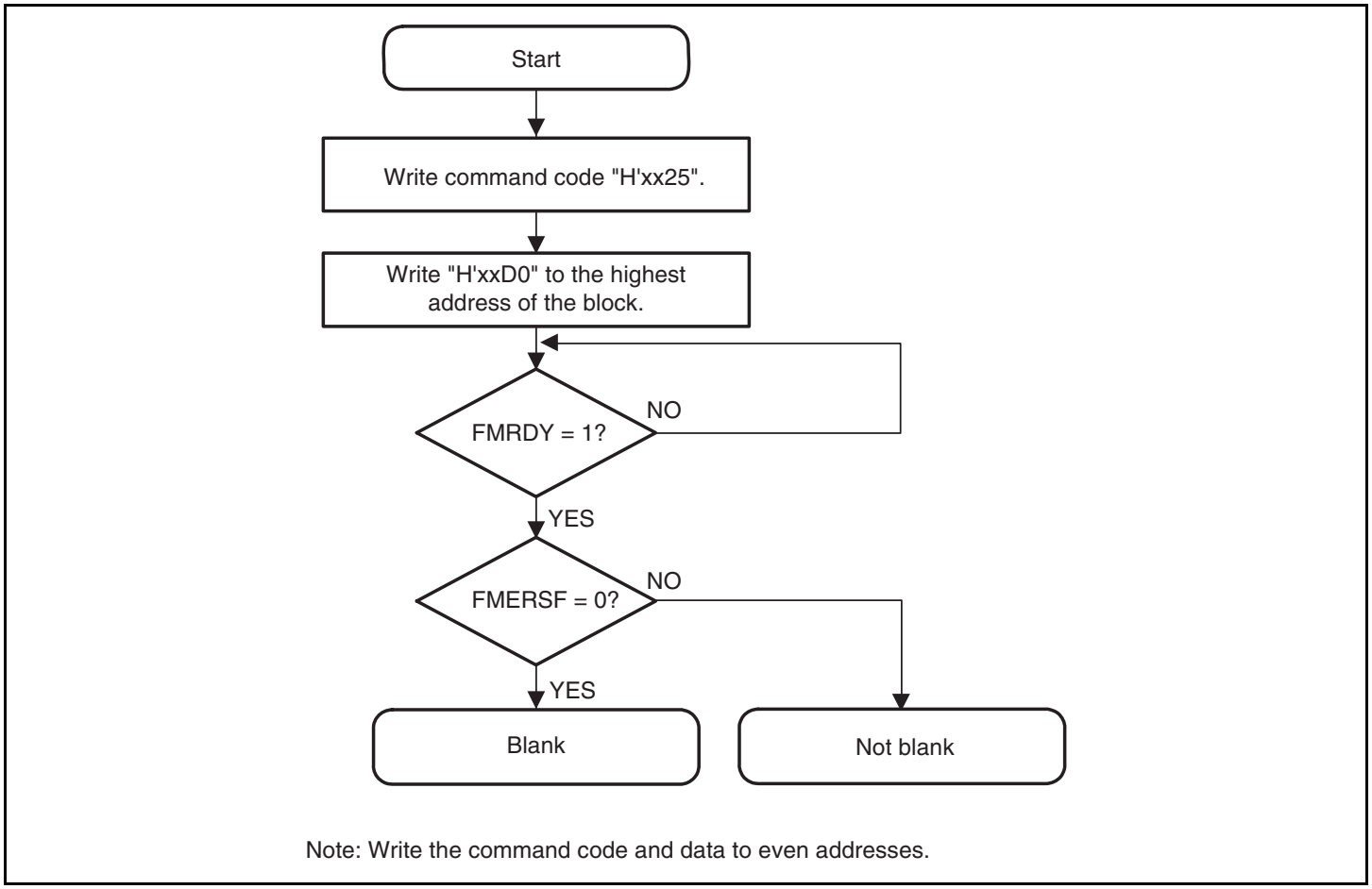


Figure 22.8 Flowchart of Block Blank Check Command Processing

22.6 Status Register

The status register indicates the state of flash memory operation and whether erasure or programming has ended successfully or with an error. The status register contents can be read through the FMRDY, FMPRSF, and FMERSF bits in FLMSTR.

Table 22.7 shows the status register.

In the EW0 mode, the status register can be read with the following timing.

- When a read status register command is issued and then an even address in the user ROM or data flash is read
- When a program command, a block erase command, or a block blank check command is issued and then an even address in the user ROM or data flash is read before a read array command is issued

Table 22.7 Status Register

Bits in Status Register	Bits in FMLSTR	Status Name	Status		Value after Reset
			0	1	
SR0 (D0)	—	Reserved	—	—	—
SR1 (D1)	—	Reserved	—	—	—
SR2 (D2)	—	Reserved	—	—	—
SR3 (D3)	—	Reserved	—	—	—
SR4 (D4)	FMPRSF	Programming status	Completed successfully	Ended with error	0
SR5 (D5)	FMERSF	Erase status	Completed successfully	Ended with error	0
SR6 (D6)	—	Reserved	—	—	
SR7 (D7)	FMRDY	Sequencer status	Busy	Ready	1

Legend:

D0 to D7: Data bus from which the bit is read when a read status register command is issued.

Note: The FMERSF (SR5) and FMPRSF (SR4) bits are cleared to 0 by a clear status register command.

When the FMERSF (SR5) or FMPRSF (SR4) bit is 1, the program, block erase, and block blank check commands are not accepted.

22.6.1 Sequencer Status (FMRDY Bit)

The sequencer status bit indicates the state of flash memory operation. Its value is 0 during execution of a program, block erase, or block blank check, and 1 in other cases.

22.6.2 Erase Status (FMERSF Bit)

See section 22.7, Full Status Check.

22.6.3 Programming Status (FMPRSF Bit)

See section 22.7, Full Status Check.

22.7 Full Status Check

When an error occurs, the FMERSF or FMPRSF bit in FLMSTR becomes 1 to indicate occurrence of the error. Read these status bits (full status check) to check the operation results.

Table 22.8 shows the errors and FLMSTR status and figure 22.9 shows a flowchart of full status check processing and corrective actions for each error.

Table 22.8 Errors and Register Status

State of FLMSTR (Status Register)			
FMERSF Bit (SR5)	**FMPRSF Bit (SR4)**	**Error**	**Error Conditions**
1	1	Command sequence error	• When a command is not issued correctly • When an invalid value (a value other than H'xxD0 or H'xxFF) is written in the second bus cycle of a block erase command*
1	0	Erase error	• When a block erase command is issued but the block is not erased correctly • When a block blank check command is issued and the checked block is not blank
0	1	Programming error	• When a program command is issued but automatic writing is not done correctly

Note: * When H'xxFF is written in the second bus cycle of this command, the flash memory enters the read array mode and the command code written in the first bus cycle is ignored.

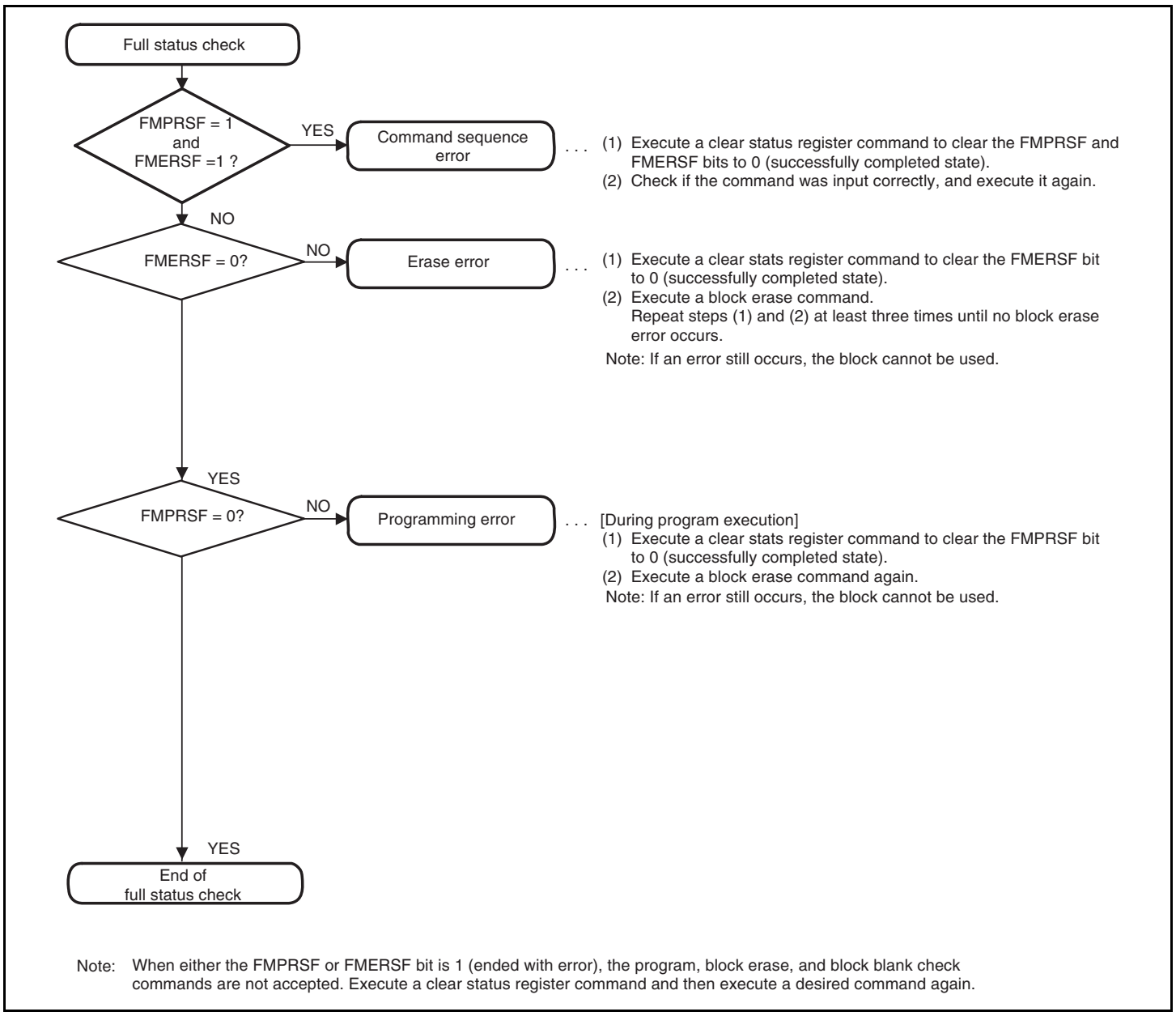


Figure 22.9 Flowchart of Full Status Check Processing and Corrective Actions for Each Error

22.8 Programmer Mode

Along with the on-board programming mode, this LSI also has a programmer mode as a further mode for the writing and erasing of programs and data. In the programmer mode, a general-purpose PROM programmer can be used to freely write programs to the on-chip ROM. Program/erase is possible on the user MAT. The PROM programmer must support Renesas microcomputers with 256-Kbyte flash memory as a device type.

A status-polling system is adopted for operation in automatic program, automatic erase, and status-read modes. In the status-read mode, details of the system's internal signals are output after execution of automatic programming or automatic erasure. In the PROM mode, provide a 12-MHz input-clock signal.

22.9 Serial Communication Interface Specification for Boot Mode

Initiating boot mode enables the boot program to communicate with the host by using the on-chip SCI_1. The serial communication interface specification is shown below.

(1) Status

The boot program has three states.

1. Bit-Rate-Adjustment State

   In this state, the boot program adjusts the bit rate to communicate 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 name, clock mode, and bit rate are selected. After selection of these settings, the program is made 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 RAM and erases the user MATs before the transition.
3. Programming/erasing state

   Programming and erasure by the boot program take place in this state. The boot program is made to transfer the programming/erasing programs to the RAM by commands from the host. Sum checks and blank checks are executed by sending these commands from the host.

These boot program states are shown in figure 22.10.

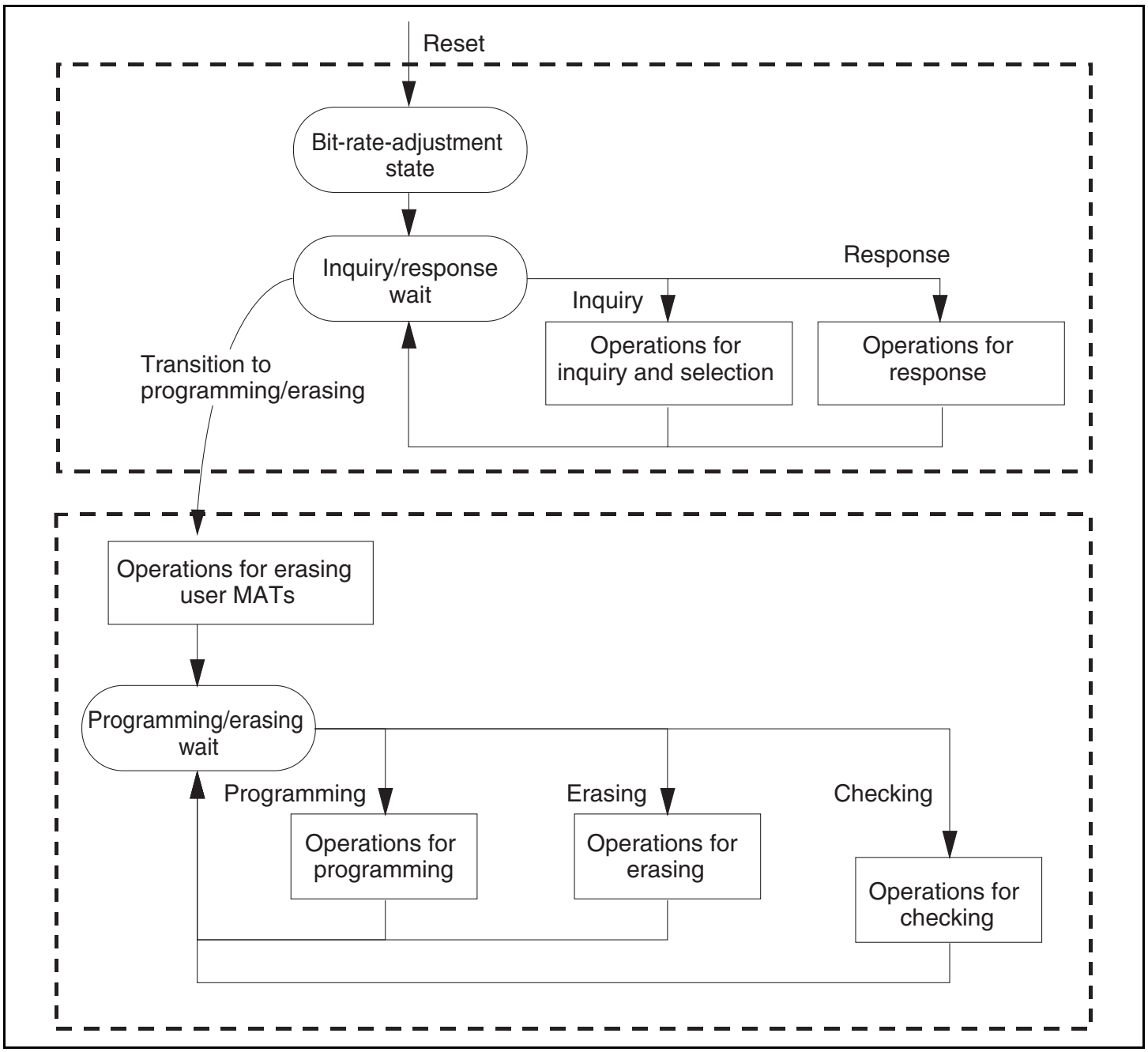


Figure 22.10 Boot Program States

(2) Bit-Rate-Adjustment State

The bit rate is calculated by measuring the period of transfer of a low-level byte (H'00) from the host. The bit rate can be changed by the command for a new bit rate selection. After the bit rate has been adjusted, the boot program enters the inquiry and selection state. The bit-rate-adjustment sequence is shown in figure 22.11.

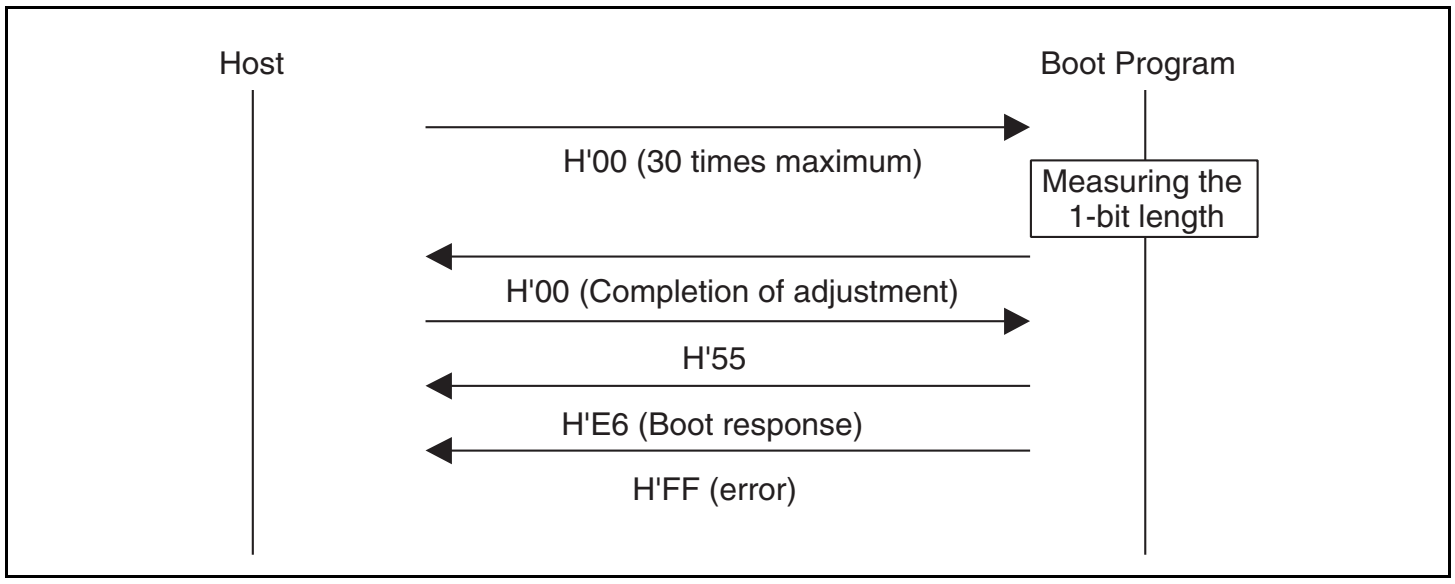


Figure 22.11 Bit-Rate-Adjustment Sequence

(3) Communications Protocol

After adjustment of the bit rate, the protocol for communications between the host and the boot program is as shown below.

1. One-byte commands and one-byte responses

   These commands and responses are comprised of a single byte. These are consists of the inquiries and the ACK for successful completion.
2. n-byte commands or n-byte responses

   These commands and responses are comprised of n bytes of data. These are selections and responses to inquiries.

   The amount of programming data is not included under this heading because it is determined in another command.
3. Error response

   The error response is a response to inquiries. It consists of an error response and an error code 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.
5. Memory read response

   This response consists of 4 bytes of data.

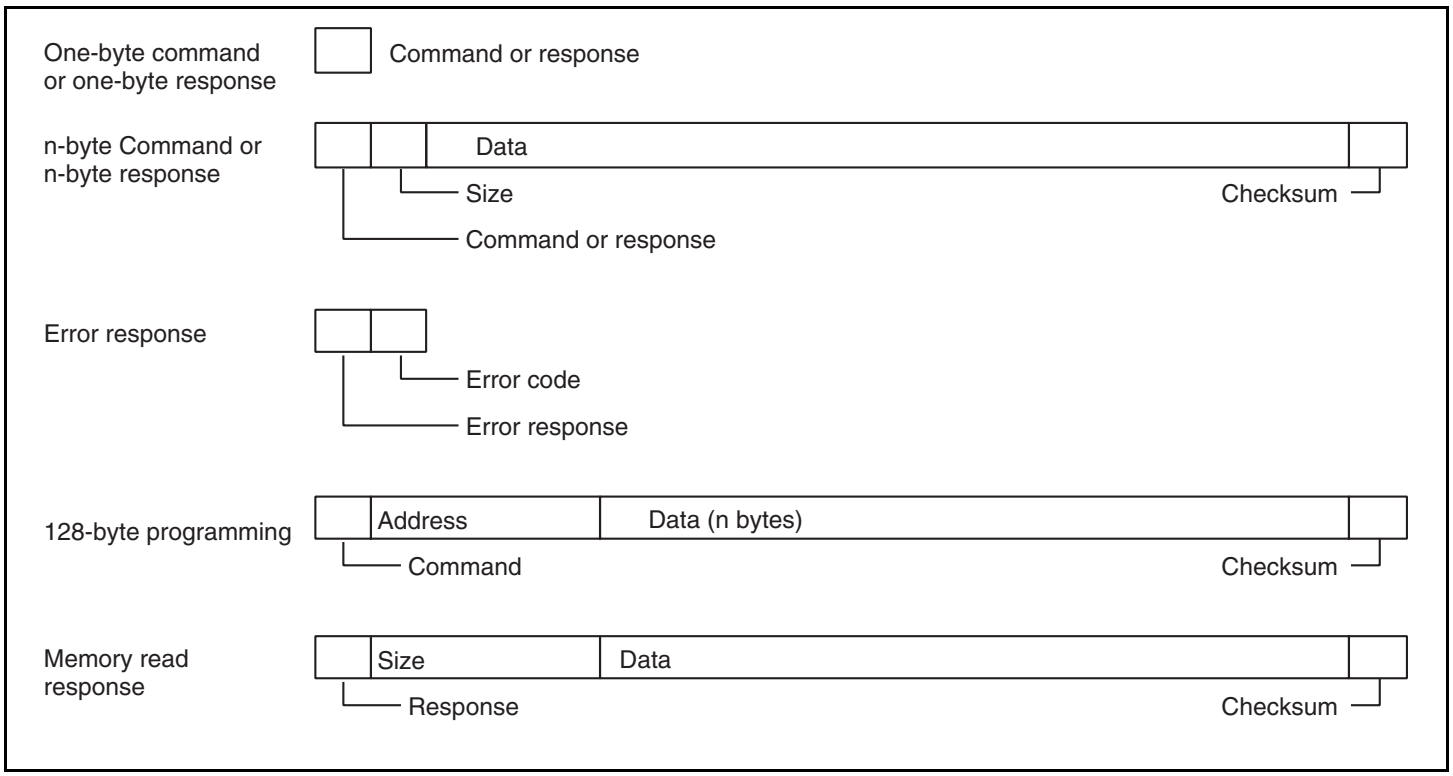


Figure 22.12 Communication Protocol Format

- Command (one byte): Commands including inquiries, selection, programming, erasing, and checking
- Response (one byte): Response to an inquiry
- Size (one byte): The amount of data for transmission excluding the command, data, and checksum
- Data (n bytes): Detailed data of a command or response
- 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.
- 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

(4) Inquiry/Selection State

The boot program returns information from the flash memory in response to the host's inquiry commands and sets the device code, clock mode, and bit rate in response to the host's selection command.

Inquiry and selection commands are listed below.

Table 22.9 Inquiry and Selection Commands

Command	Command Name	Description
H'20	Supported Device Inquiry	Inquiry regarding device codes and product name
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 each multiple
H'23	Operating Clock Frequency Inquiry	Inquiry regarding the maximum and minimum values of the main clock and peripheral clocks
H'25	User MAT Information Inquiry	Inquiry regarding the number of user MATs and the start and last addresses of each MAT
H'26	Block for Erasing Information Inquiry	Inquiry regarding the number of blocks and the start and last addresses of each block
H'27	Programming Unit Inquiry	Inquiry regarding the unit of programming data
H'3F	New Bit Rate Selection	Selection of new bit rate
H'40	Transition to Programming/Erasing State	Erasing of user MAT and entry to programming/erasing state
H'4F	Boot Program Status Inquiry	Inquiry regarding the operated status of the boot program

The selection commands, which are device selection (H'10), clock mode selection (H'11), and new bit rate selection (H'3F), should be sent from the host in that order. These commands will certainly be needed. When two or more selection commands are sent at once, the last command will be valid.

All of these commands, except for the boot program status inquiry command (H'4F), will be valid until the boot program receives the programming/erasing transition command (H'40). The host can choose the needed commands out of the commands and inquiries listed above. The boot program status inquiry command (H'4F) is valid even after the boot program has received the programming/erasing transition command (H'40).

(a) Supported Device Inquiry

The boot program will return the device codes of supported devices and the product name in 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 contributes by the number of devices, characters, device 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 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 of all values from the command byte to the SUM byte becomes H'00.

(b) Device Selection

The boot program will set the supported device to the specified device code. The program will return the selected device code in response to the inquiry after this setting has been made.

Command	H'10	Size	Device code	SUM

- Command, H'10, (one byte): Device selection
- Size (one byte): Amount of device-code data
  This is fixed at 2.
- Device code (four bytes): Device code (ASCII code) returned in response to the supported device inquiry
- 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: Checksum 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 modes
- Mode (one byte): Values of the supported clock modes (i.e. H'01 means clock mode 1.)
- SUM (one byte): Checksum

(d) Clock Mode Selection

The boot program will set the specified clock mode. The program will return the selected clock-mode information after this setting has been made.

The clock-mode selection command should be sent after the device-selection commands.

Command	H'11	Size	Mode	SUM

- Command, H'11, (one byte): Selection of clock mode
- Size (one byte): Amount of data that represents the modes
  This is fixed at 1.
- Mode (one byte): A clock mode returned in reply to the supported clock mode inquiry.
- 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 must be selected using these respective values.

(e) Multiplication Ratio Inquiry

The boot program will return the supported multiplication and division ratios.

Command	H'22

- Command, H'22, (one byte): Inquiry regarding multiplication ratio

Response	H'32	Size	Number of types					
	Number of multiplication ratios	Multiplica-tion 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 clock types and multiplication ratios and the multiplication ratios
- Number of types (one byte): The number of supported multiplied clock types (e.g. when there are two multiplied clock types, which are the main and peripheral clocks, the number of types will be H'02.)
- Number of multiplication ratios (one byte): The number 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 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.
- SUM (one byte): Checksum

(f) Operating Clock Frequency Inquiry

The boot program will return the number of operating clock frequencies, and the maximum and minimum values.

Command	H'23

- Command, H'23, (one byte): Inquiry regarding operating clock frequencies

Response	H'33	Size	Number of operating clock frequencies
	Minimum value of operating clock frequency		Maximum value of operating clock frequency
	...		
	SUM		

- Response, H'33, (one byte): Response to operating clock frequency inquiry
- Size (one byte): The number of bytes that represents the minimum values, maximum 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 represent the values in MHz, valid to the hundredths place of MHz, and multiplied by 100 (e.g. when the value is 64 MHz, it will be 6400 and H'1900).
- Maximum value (two bytes): Maximum value of the multiplied or divided clock frequencies.

  There are as many pairs of minimum and maximum values as there are operating clock frequency.
- SUM (one byte): Checksum

(g) 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	
	Area-start address			Area-last address
	...			
	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 returned is H'01.
- Area-start address (four bytes): 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) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses.

Command	H'26

- Command, H'26, (one byte): Inquiry regarding erased block information

Response	H'36	Size	Number of blocks	
	Block-start address			Block-last address
	...			
	SUM			

- Response, H'36, (one byte): Response to the number of erased blocks and addresses
- Size (two 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

(i) 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 to 2
- Programming unit (two bytes): A unit for programming
  This is the unit for reception of programming data.
- SUM (one byte): Checksum

(j) New Bit-Rate Selection

The boot program will set a new bit rate and return the new bit rate.

This selection should be sent after sending the clock mode selection command.

Command	H'3F	Size	Bit rate	Input frequency
	Number of multiplication ratios	Multiplication ratio 1	Multiplication ratio 2	
	SUM			

- Command, H'3F, (one byte): Selection of new bit rate
- Size (one byte): The number of bytes that represents the bit rate, input frequency, number of multiplication ratios, and multiplication ratio
- Bit rate (two bytes): New bit rate
  One hundredth of the value (e.g. when the value is 19,200 bps, the bit rate is H'00C0, which is 192.)
- 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 64 MHz, the input frequency is H'1900 (= 6400)).
- Number of multiplication ratios (one byte): The number of multiplication ratios to which the device can be set.

- Multiplication ratio 1 (one byte): The value of multiplication or division ratios for the main 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 = [-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 is divided by two, the value of division ratio will be H'FE. H'FE = [-2])
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to selection of a new bit rate
  When it is possible to set the bit rate, the response will be ACK.

Error Response	H'BF	ERROR

- Error response, H'BF, (one byte): Error response to selection of new bit rate
- ERROR: (one byte): Error code
  - H'11: Checksum error
  - H'24: Bit-rate selection error
    The rate is not available.
  - H'25: Error in input frequency
    This input frequency is not within the specified range.
  - H'26: Multiplication-ratio error
    The ratio does not match an available ratio.
  - H'27: Operating frequency error
    The frequency is not within the specified range.

(5) Received Data Check

The methods for checking of received data are listed below.

1. Input frequency

The received value of the input frequency is checked to ensure that it is within the range of 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 a multiplication or division ratio for the clock modes of the specified device. When the value is out of this range, an input-frequency error is generated.

3. Operating frequency error

The 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 is operated at the operating frequency. The expression is given below.

Operating frequency = Input frequency × Multiplication ratio, or
Operating frequency = Input frequency ÷ Division ratio

The calculated operating frequency should be checked to ensure that it is within the range of 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.

4. Bit rate

To facilitate error checking, the value (n) of clock select (CKS) in the serial mode register (SMR), and the value (N) in the bit rate register (BRR), which are found from the peripheral operating clock frequency (ϕ) and bit rate (B), are used to calculate the error rate to ensure that it is less than 4%. If the error is 4% or more, a bit rate error is generated. The error is calculated using the following expression:

$$\text{Error (\%)} = \{[\frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{(2 \times n - 1)}}] - 1\} \times 100$$

When the new bit rate is selectable, the rate will be set in the register after sending ACK in response. The host will send an ACK with the new bit rate for confirmation and the boot program 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 22.13.

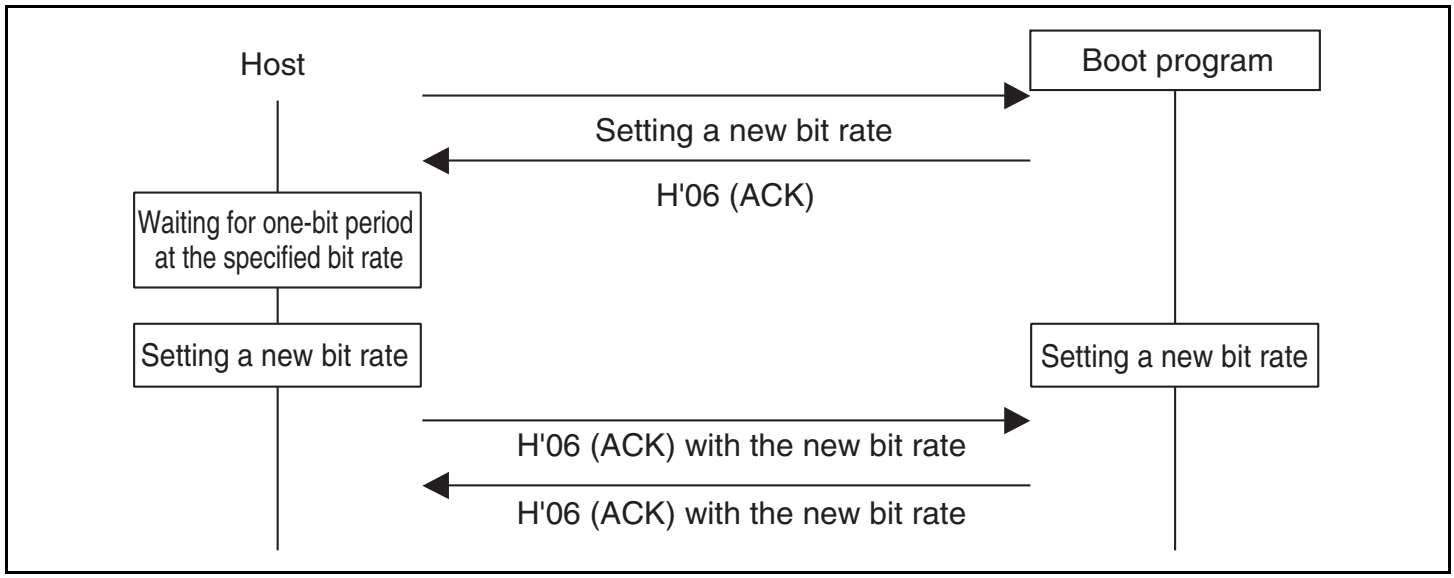


Figure 22.13 New Bit-Rate Selection Sequence

(6) Transition to Programming/Erasing State

The boot program will transfer the erasing program, and erase data of the user MATs. On completion of this erasure, ACK will be returned and the programming/erasing state will be entered.

The host should select the device code, clock mode, and new bit rate with device selection, clock-mode selection, and new bit-rate selection commands, and then send the command for the transition to programming/erasing state. These procedures should be carried out before sending of the programming selection command or program data.

Command	H'40

- Command, H'40, (one byte): Transition to programming/erasing state

Response	H'06

- Response, H'06, (one byte): Response to transition to programming/erasing state
  The boot program will send ACK when the user MAT has been erased by the transferred erasing program.

Error Response	H'C0	H'51

- Error code, H'51, (one byte): Erasing error
  An error occurred and erasure was not completed.

(7) 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 selection or an inquiry command after the transition to programming/erasing state command, are examples.

Error Response	H'80	H'xx

- Error response, H'80, (one byte): Command error
- Command, H'xx, (one byte): Received command

(8) Command Order

The order for commands in the inquiry/selection state is shown below.

1. A supported device inquiry (H'20) should be made to inquire about the supported devices.
2. The device should be selected from among those described by the returned information and set with a device-selection (H'10) command.
3. A clock-mode inquiry (H'21) should be made to inquire about the supported clock modes.
4. The clock mode should be selected from among those described by the returned information and set.
5. After selection of the device and clock mode, inquiries for other required information should 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, inquiries for the information of programming/erasing to the user MAT should be made by the user MATs information inquiry (H'25), erased block information inquiry (H'26), and programming unit inquiry (H'27).
8. After making inquiries and selecting a new bit rate, issue the transition to programming/erasing state command (H'40). The boot program will then enter the programming/erasing state.

(9) Programming/Erasing State

A programming selection command makes the boot program select the programming method, an 128-byte programming command makes it program the memory with data, and an erasing selection command and block erasing command make it erase the block. The programming/erasing commands are listed below.

Table 22.10 Programming/Erasing Commands

Command	Command Name	Description
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'4B	User MAT sum check	Checks the checksum of the user MAT
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

- Programming

Programming is executed by a programming-selection command and a 128-byte programming command.

Firstly, the host should send the programming-selection command and select the programming method and programming MATs. There is a programming selection command according to the area and method for programming.

- User MAT programming selection

After issuing the programming selection command, the host should send the 128-byte programming command. The 128-byte programming command that follows the selection 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'FFFFFFFF as the 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 with another method or of another MAT, the procedure must be repeated from the programming selection command.

The sequence for programming-selection and 128-byte programming commands is shown in figure 22.14.

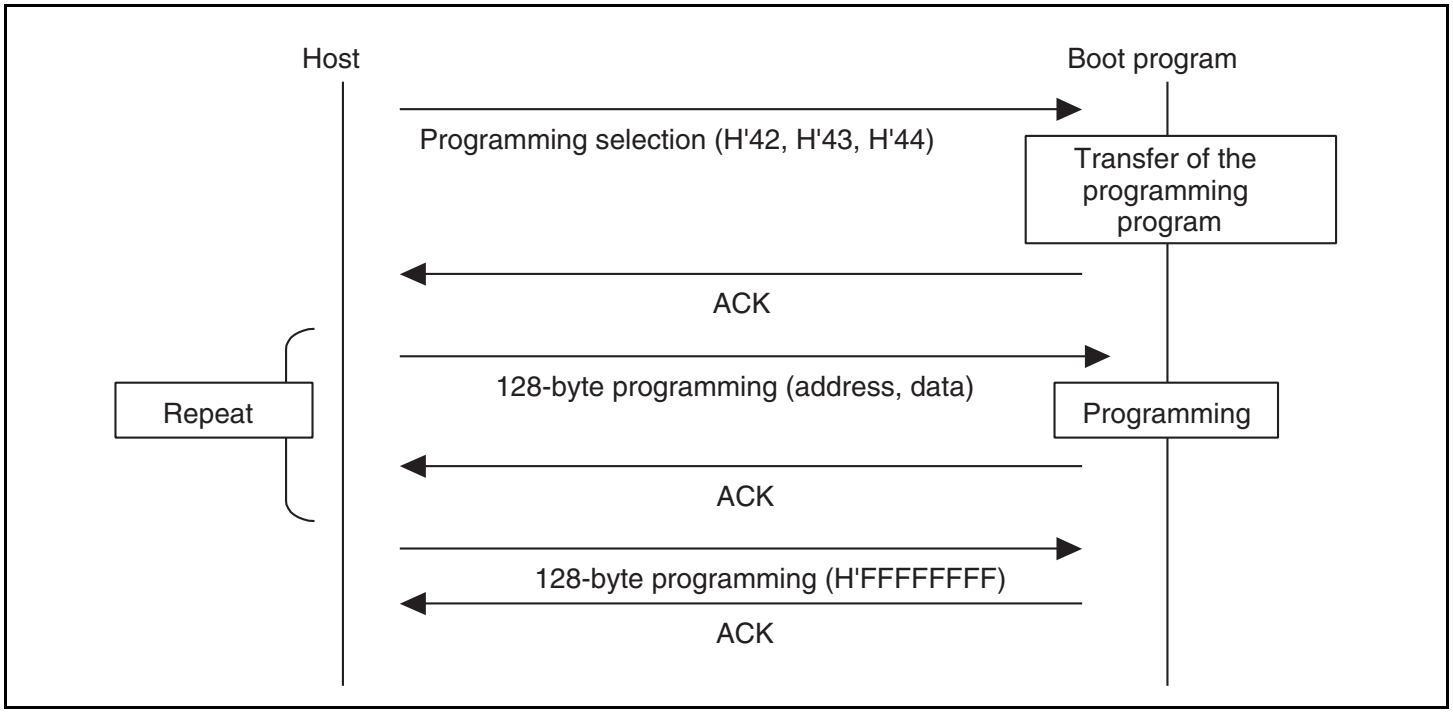


Figure 22.14 Programming Sequence

(a) User-program programming selection

The boot program will transfer a program for programming. The data is programmed to the user MATs by the transferred program for programming.

Command	H'43

- Command, H'43, (one byte): User-program programming selection

Response	H'06

- Response, H'06, (one byte): Response to user-program programming selection
  When the programming program has been transferred, the boot program will return ACK.

(b) 128-byte programming

The boot program will use the programming program transferred by the programming selection to program the user MATs in response to 128-byte programming.

Command	H'50	Address						
	Data	...						
	...							
	SUM							

- Command, H'50, (one byte): 128-byte programming
- Programming Address (four bytes): Start address for programming
  Multiple of the size specified in response to the programming unit inquiry
  (i.e. H'00, H'01, H'00, H'00: H'01000000)
- Programming 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 within 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 the programming is in 128-byte units, the lower byte 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.

Command	H'50	Address	SUM

- Command, H'50, (one byte): 128-byte programming
- Programming Address (four bytes): End code is H'FF, H'FF, H'FF, H'FF.
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, the boot program will return ACK.

Error Response	H'D0	ERROR

- Error Response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code
  H'11: Checksum error
  H'53: Programming error
  An error has occurred in programming and programming cannot be continued.

(10) Erasure

Erasure is performed with the erasure selection and block erasure command.

Firstly, erasure is selected by the erasure selection command and the boot program then erases the specified block. The command should be repeatedly executed if two or more blocks are to be erased. Sending a block-erasure command from the host with the block number H'FF will stop the erasure operating. On completion of erasing, the boot program will wait for selection of programming or erasing.

The sequences of the issuing of erasure selection commands and the erasure of data are shown in figure 22.15.

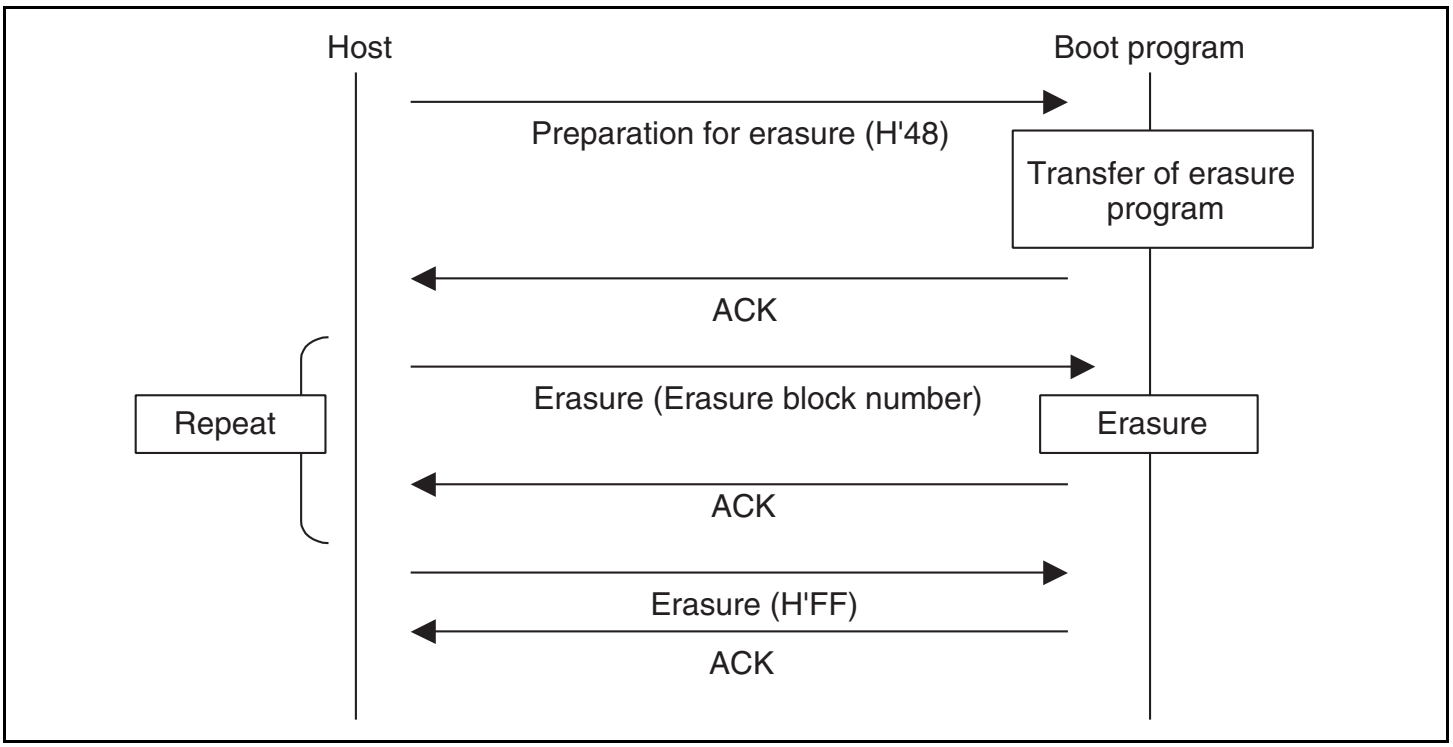


Figure 22.15 Erasure Sequence

(a) Erasure Selection

The boot program will transfer the erasure program. User MAT data is erased by the transferred 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.

(b) Block Erasure

The boot program will erase the contents of the specified block.

Command	H'58	Size	Block number	SUM

- Command, H'58, (one byte): Erasure
- Size (one byte): The number of bytes that represents the erasure block number
  This is fixed to 1.
- Block number (one byte): Number of the block to be erased
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to Erasure
  After erasure has been completed, the boot program will return ACK.

Error Response	H'D8	ERROR

- Error Response, H'D8, (one byte): Response to Erasure
- ERROR (one byte): Error code

  H'11: Checksum error
  H'29: Block number error
  Block number is incorrect.
  H'51: Erasure error
  An error has occurred during erasure.

Section 23 Boundary Scan (JTAG)

For details, contact your Renesas Technology sales agency.

Section 24 Clock Pulse Generator

This LSI has an on-chip clock pulse generator (CPG) that generates the system clock (ϕ) and internal clocks. The clock pulse generator consists of an oscillator circuit, a system-clock PLL circuit and a divider.

Figure 24.1 shows a block diagram of the clock pulse generator.

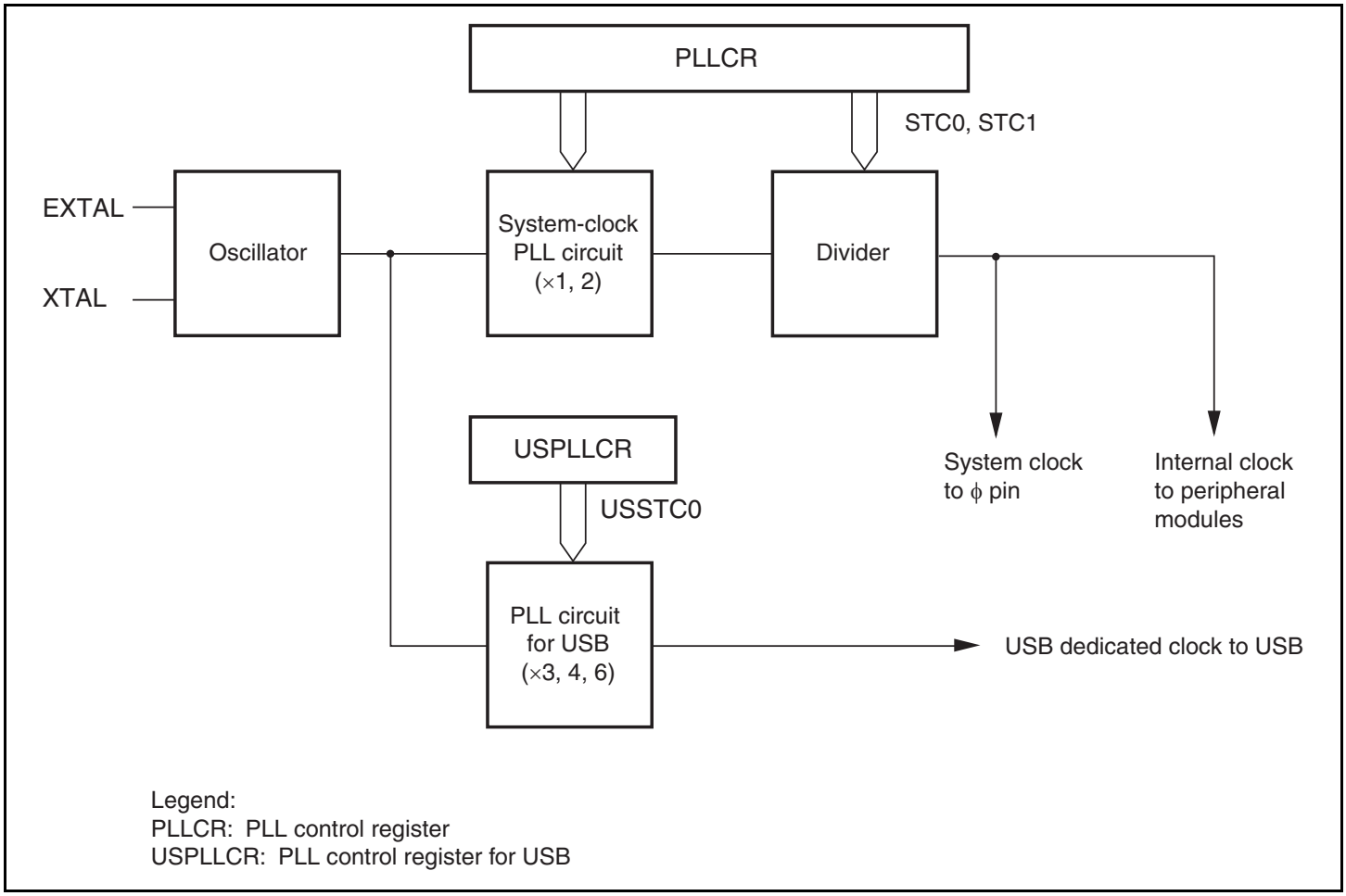


Figure 24.1 Block Diagram of Clock Pulse Generator

The frequency of the system clock from the oscillator can be changed by means of the system-clock PLL circuit and divider. Frequency changes are made by software by means of settings in the PLL control register (PLLCR).

The USB module requires a 48-MHz clock. Set the frequency of the USB dedicated clock (cku toe 48 MHz. Changes to the frequency of the USB dedicated clock are made by software by means of settings in the USB PLL control register (USPLLCR).

24.1 Register Descriptions

The clock pulse generator has the following registers.

- System clock control register (SCKCR)
- PLL control register (PLLCR)
- USB PLL control register (USPLLCR)

24.1.1 System Clock Control Register (SCKCR)

SCKCR controls φ clock output and selects operation when the PLLCR register setting is changed.

Bit	Bit Name	Initial Value	R/W	Description
7	PSTOP	0	R/W	φ Clock Output Disable Controls φ output. Normal Operation 0: φ output 1: Fixed high Sleep Mode 0: φ output 1: Fixed high Software Standby Mode 0: Fixed high 1: Fixed high Hardware Standby Mode 0: High impedance 1: High impedance All module clock stop mode 0: φ output 1: Fixed high
6	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	SDPSTP*	0	R/W	SDRAMϕ Output Disable Controls SDRAMϕ. 0: SDRMϕ output. 1: Can be used as PH1/$\overline{CS5}$/$\overline{RAS5}$. When the SDRAMϕ output is selected, the pin functions as follows in each power-down mode. Normal operation: SDRAMϕ output Sleep mode: SDRAMϕ output Software standby mode: Fixed at a low level Hardware standby mode: High-impedance state All module clock stop mode: SDRAMϕ output
4	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
3	STCS	0	R/W	Frequency Multiplication Factor Switching Mode Select Selects the operation when the PLLCR register setting is changed. 0: Specified multiplication factor is valid after transition to software standby mode. 1: Specified multiplication factor is valid immediately after STC1 and STC0 bits are rewritten.
2	—	0	R/W	Reserved
1	—	0	R/W	This bit can be read from or written to. However, the write value should always be 0.
0	—	0	R/W	

Note: * The H8S/2456 group and H8S/2454 group do not have this bit. The pin always functions as an I/O port regardless of this bit setting.

24.1.2 PLL Control Register (PLLCR)

PLLCR sets the frequency multiplication factor used by the system-clock PLL circuit.

Care must be taken when writing to this register. For details, see section 24.3, System-Clock PLL Circuit and Divider.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.
2	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
1	STC1	0	R/W	Frequency Multiplication Factor for System-Clock PLL Circuit and System Clock Divider Setting The STC bits specify the frequency multiplication factor and dividing ratio with respect to the oscillator frequency. 00: × 1 01: × 2 10: Setting prohibited 11: divided by 2
0	STC0	0	R/W	

24.1.3 USB PLL Control Register (USPLLCR)

USPLLCR selects multiplication factor used by the PLL circuit.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
1	USSTC1	0	R/W	Frequency Multiplication Factor for USB PLL Circuit Setting The USSTC bits specify the frequency multiplication factor for USB PLL circuit. 00: USB PLL circuit operation halted 01: USB PLL in operation with frequency x 3 10: USB PLL in operation with frequency x4 11: USB PLL in operation with frequency x6
0	USSTC0	0	R/W	

24.2 Oscillator

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock.

24.2.1 Connecting a Crystal Resonator

A crystal resonator can be connected as shown in the example in figure 24.2. Select the damping resistance R_d according to table 24.1. An AT-cut parallel-resonance type should be used. When a crystal resonator is used, the range of its frequencies is from 8 to 20 MHz.

Figure 24.3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 24.2.

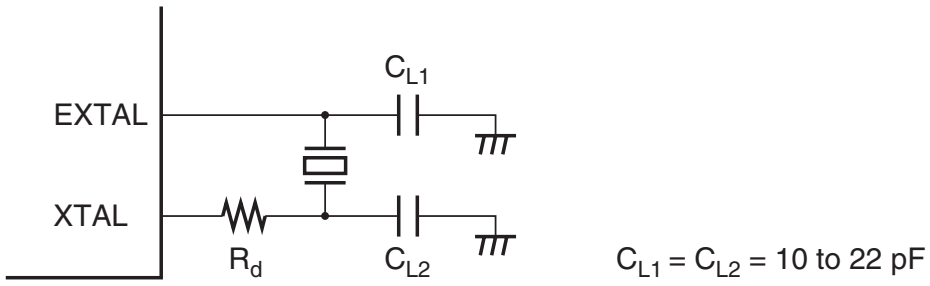


Figure 24.2 Connection of Crystal Resonator (Example)

Table 24.1 Damping Resistance Value

Frequency (MHz)	8	12	16	20
R_d (Ω)	200	0	0	0

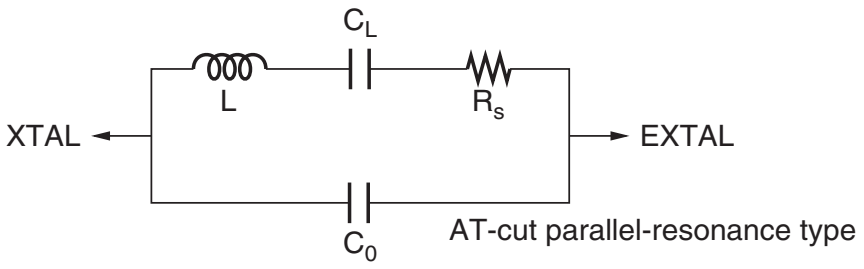


Figure 24.3 Crystal Resonator Equivalent Circuit

Table 24.2 Crystal Resonator Characteristics

Frequency (MHz)	8	12	16	20
R_S max (Ω)	80	60	50	40
C_0 max (pF)	7	7	7	7

24.2.2 External Clock Input

An external clock signal can be input as shown in the examples in figure 24.4. If the XTAL pin is left open, make sure that parasitic capacitance is no more than 10 pF. When the counter clock is input to the XTAL pin, make sure that the external clock is held high in standby mode.

Table 24.3 shows the input conditions for the external clock. When an external clock is used, the range of its frequencies is from 8 to 20 MHz.

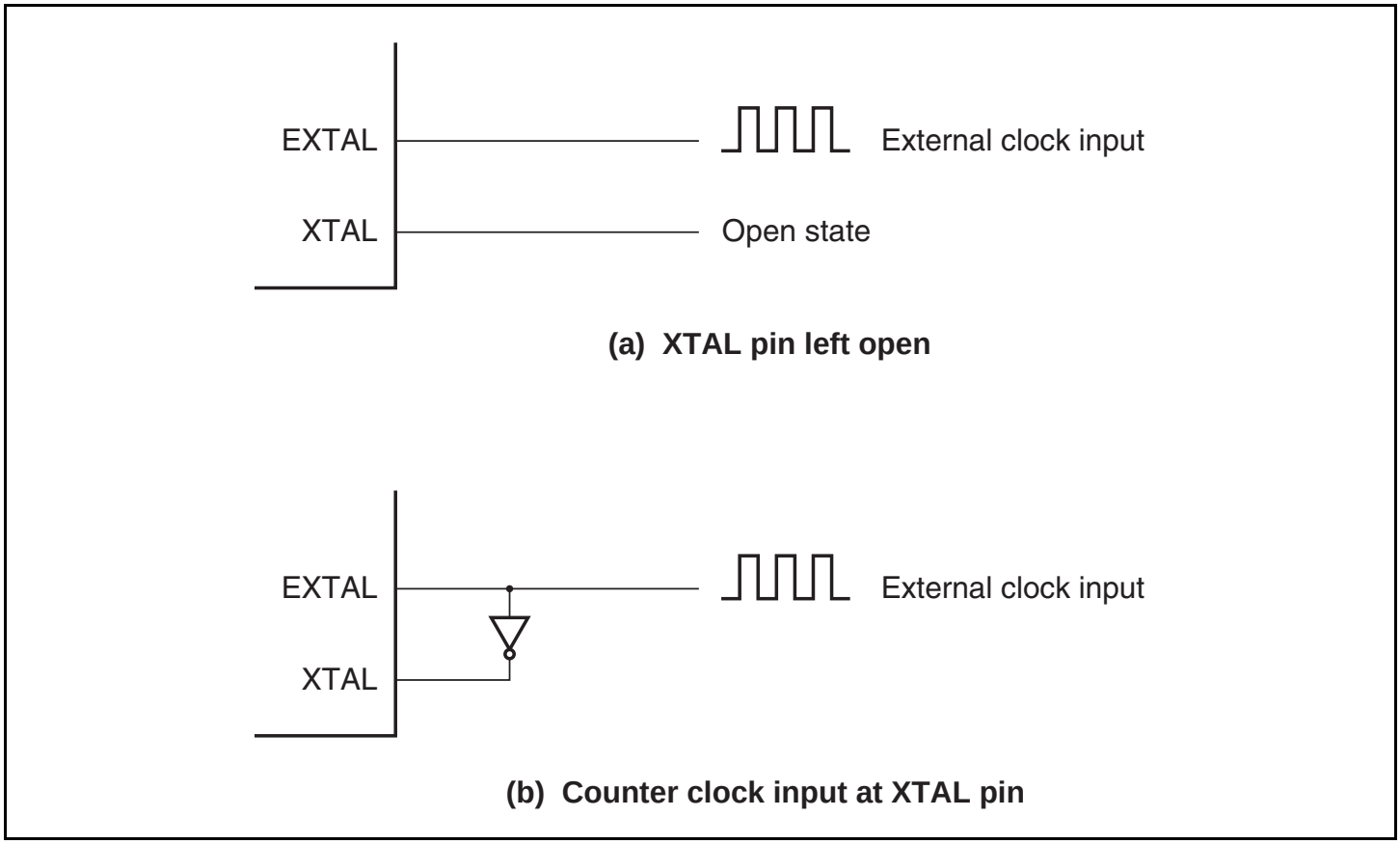


Figure 24.4 Connection of External Clock Input (Examples)

Table 24.3 External Clock Input Conditions

		V_{cc} = 3.0 V to 3.6 V			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
External clock input low pulse width	t_{EXL}	20	—	ns	Figure 24.5
External clock input high pulse width	t_{EXH}	20	—	ns	
External clock rise time	t_{EXr}	—	5	ns	
External clock fall time	t_{EXf}	—	5	ns	
Clock low pulse width	t_{CL}	0.4	0.6	t_{cyc}	
Clock high pulse width	t_{CH}	0.4	0.6	t_{cyc}	

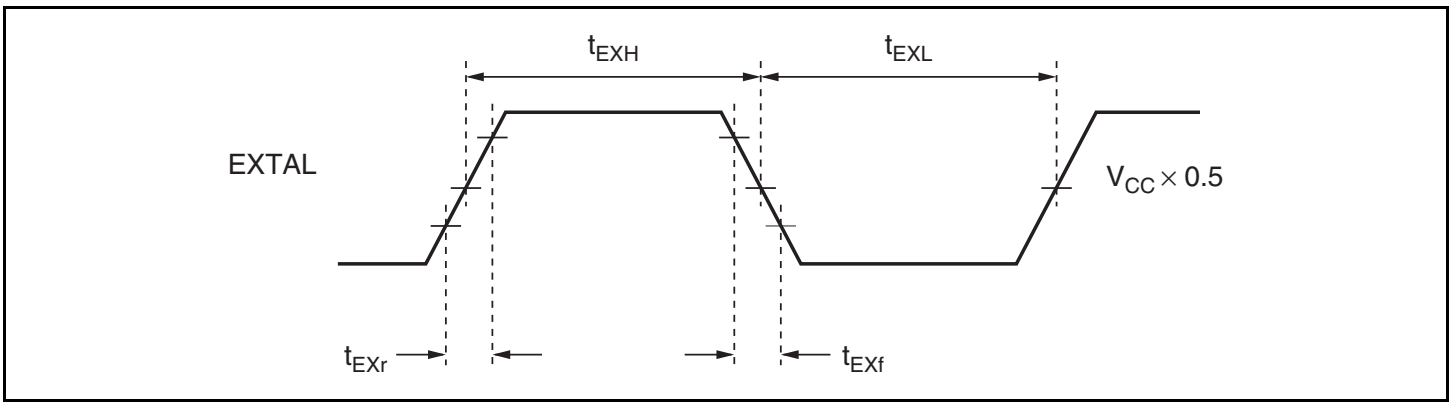


Figure 24.5 External Clock Input Timing

24.3 System-Clock PLL Circuit and Divider

The system-clock PLL circuit and divider have the function of multiplying the frequency of the clock from the oscillator by a factor of 1, 2, or dividing by 2. The system clock frequency is set with the STC1 and STC0 bits in PLLCR. The phase of the rising edge of the internal clock is controlled so as to match that of the rising edge of the EXTAL pin.

When the frequency is changed with the system-clock PLL circuit and divider, the operation varies according to the setting of the STCS bit in SCKCR.

When STCS = 0, the setting of the changed frequency becomes valid after a transition to software standby mode. The transition time count is performed in accordance with the setting of bits STS3 to STS0 in SBYCR. For details on SBYCR, see section 25.1.1, Standby Control Register (SBYCR).

1. The initial PLL circuit multiplication factor is 1.
2. A value is set in bits STS3 to STS0 to give the specified transition time.
3. The target value is set in bits STC1 and STC0, and a transition is made to software standby mode.
4. The clock pulse generator stops and the value set in STC1 and STC0 becomes valid.
5. Software standby mode is cleared, and a transition time is secured in accordance with the setting in STS3 to STS0.
6. After the set transition time has elapsed, this LSI resumes operation using the target multiplication factor.

When STCS = 1, a change to the frequency setting becomes effective a maximum of four cycles after the setting is changed. If the clock frequency is changed during access to an external address space, correct operation cannot be guaranteed. Therefore, be sure to store instructions that change the STC1 and STC0 bits and other instructions to be executed within a maximum of four cycles after the change to the frequency setting in on-chip ROM or on-chip RAM, so that instructions do not access an external address space before the frequency clock is switched over.

24.4 PLL Circuit for the USB Module

The PLL circuit for the USB module takes 8, 12, and 16 MHz clock signals from an oscillator and generates the 48-MHz clock for the USB module through frequency-multiplication by 3,4, or 6.

The frequency-multiplication factor is set by bits USSTC1 and USSTC0 in the USPLLCR. For details on the USPLLCR, see section 24.1.3, USB PLL Control Register (USPLLCR).

When the USB is in use, make settings so that the system clock runs at or above 14 MHz. The settings listed below (table 23.4) produce a USB dedicated clock at 48 MHz. Operation at other frequencies cannot be guaranteed.

Table 24.4 Clock Selection when the USB is to be used

Input clock frequency from the oscillator (MHz)	USB dedicated clock (cku: 48 MHz)	System clock (ϕ)
8 MHz	EXTALx6	EXTALx2 (16MHz)
12 MHz	EXTALx4	EXTALx2 (24MHz)
16 MHz	EXTALx3	EXTALx2 (32MHz)
		EXTALx1 (16MHz)

24.5 Usage Notes

24.5.1 Notes on Clock Pulse Generator

1. The following points should be noted since the frequency of ϕ changes according to the settings of PLLCR.

   Select a clock division ratio that is within the operation guaranteed range of clock cycle time tcyc shown in the AC timing of the Electrical Characteristics. In other words, ϕ must be set to a value between 8 MHz (minimum) and 33 MHz (maximum). The setting of ϕ must not be less than 8 MHz or greater than 33 MHz.
2. All the on-chip peripheral modules operate on the ϕ. Therefore, note that the time processing of modules such as a timer and SCI differ before and after changing the clock division ratio. In addition, wait time for clearing software standby mode differs by changing the clock division ratio. See the description, Setting Oscillation Stabilization Time after Clearing Software Standby Mode in section 25.2.3, Software Standby Mode, for details.
3. Note that the frequency of ϕ will be changed when setting PLLCR while executing the external bus cycle with the write-data-buffer function.

24.5.2 Notes on Resonator

Since various characteristics related to the resonator are closely linked to the user's board design, thorough evaluation is necessary on the user's part, using the resonator connection examples shown in this section as a guide. As the parameters for the oscillation circuit will depend on the floating capacitance of the resonator and the user board, the parameters should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is not applied to the resonator pin.

24.5.3 Notes on Board Design

When using the crystal resonator, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins. Other signal lines should be routed away from the oscillation circuit to prevent induction from interfering with correct oscillation. See figure 24.6.

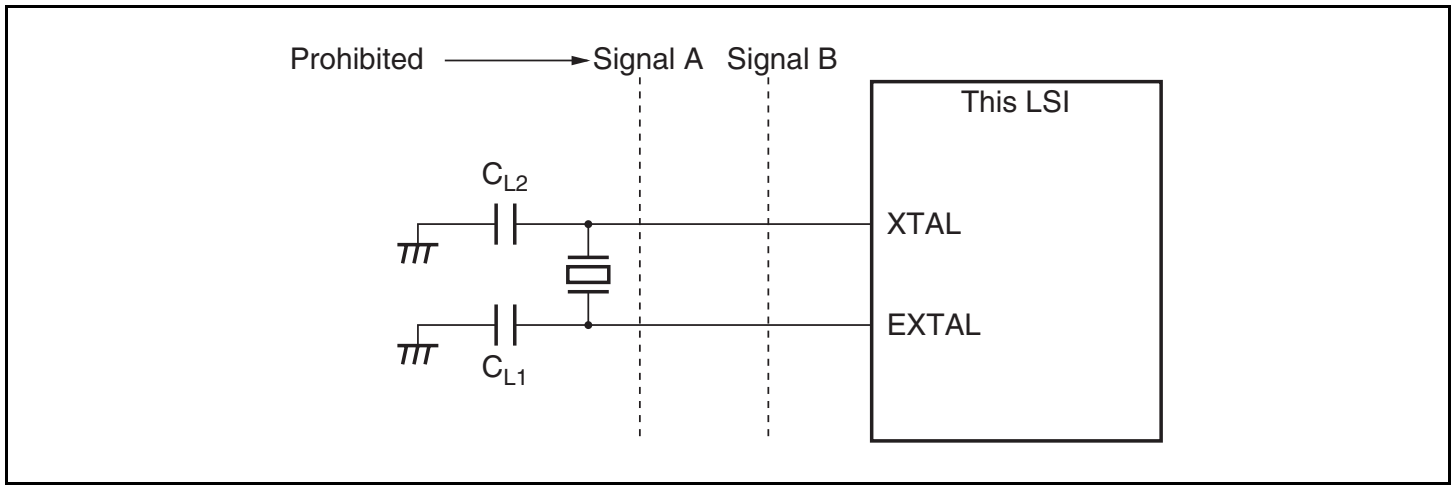


Figure 24.6 Note on Board Design for Oscillation Circuit

Figure 24.7 shows the external circuitry recommended for the PLL circuit. Separate PLLVcc and PLLVss from the other Vcc and Vss lines at the board power supply source, and be sure to insert bypass capacitors CPB and CB close to the pins.

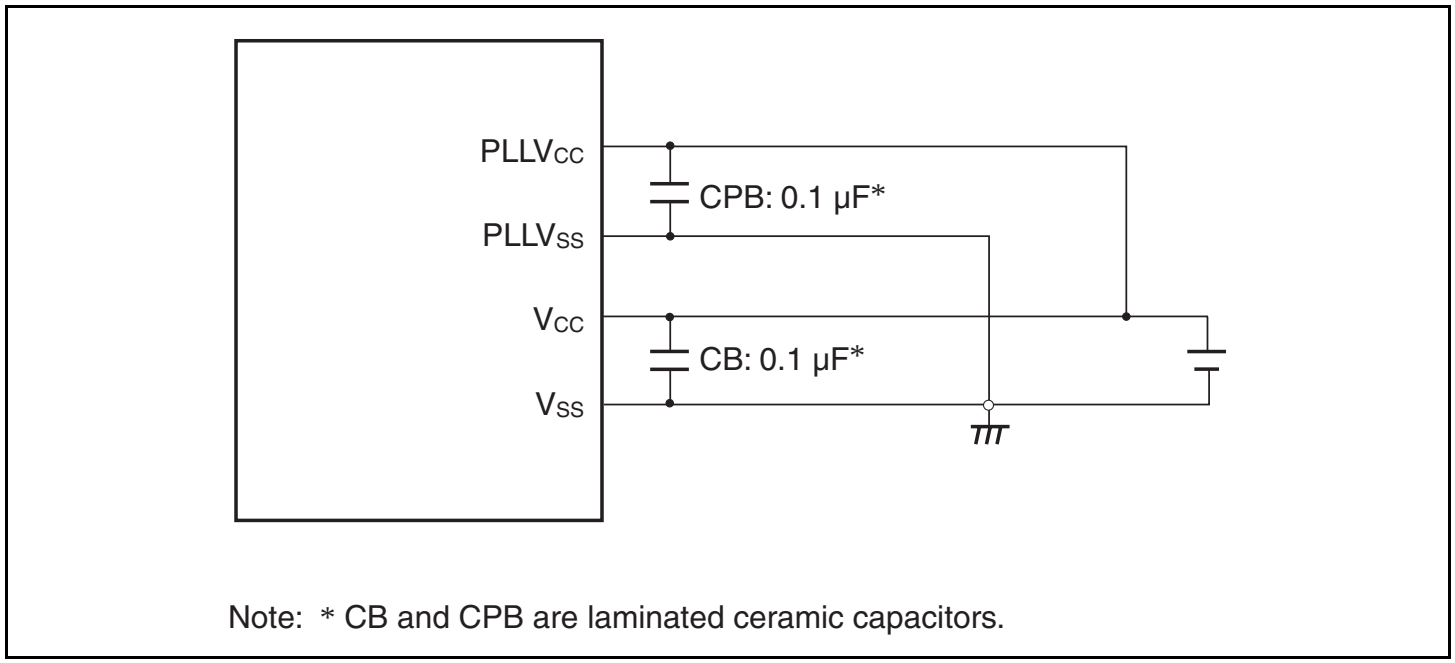


Figure 24.7 Recommended External Circuitry for PLL Circuit

Section 25 Power-Down Modes

In addition to the normal program execution state, this LSI has power-down modes in which operation of the CPU and oscillator is halted and power consumption is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip peripheral modules, and so on.

This LSI's operating modes are high-speed mode and six power down modes:

- Clock division mode
- Sleep mode
- Module stop function
- All module clocks stop mode
- Software standby mode
- Hardware standby mode

Sleep mode is a CPU state, clock division mode is an on-chip peripheral function (including bus masters and the CPU) state, and module stop function is an on-chip peripheral function (including bus masters other than the CPU) state. A combination of these modes can be set.

After a reset, this LSI is in high-speed mode.

Table 25.1 shows the internal states of this LSI in each mode. Figure 25.1 shows the mode transition diagram.

Table 25.1 Operating Modes and Internal states of the LSI

Operating State		High Speed Mode	Clock Division Mode	Sleep Mode	Module Stop Function	All Module Clocks Stop Mode	Software Standby Mode	Hardware Standby Mode
Clock pulse generator		Operating	Operating	Operating	Operating	Operating	Stopped	Stopped
CPU	Instruction execution	Operating	Operating	Stopped	Operating	Stopped	Stopped	Stopped
	Register			Retained			Retained	Undefined
External interrupts	NMI	Operating	Operating	Operating	Operating	Operating	Operating	Stopped
	IRQ0 to 15*[1]							
Peripheral functions	WDT	Operating	Operating	Operating	Operating	Operating	Stopped (Retained)	Stopped (Reset)
	TMR	Operating	Operating	Operating	Stopped (Retained)	Operating/ Stopped (Retained)*[2]	Stopped (Retained)	Stopped (Reset)
	EXDMAC*[3]	Operating	Operating	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Reset)
	DMAC	Operating	Operating	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Reset)
	DTC	Operating	Operating	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Reset)
	TPU	Operating	Operating	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Reset)
	PPG	Operating	Operating	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Reset)
	D/A	Operating	Operating	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Reset)
	A/D	Operating	Operating	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Reset)
	SCI	Operating	Operating	Operating	Stopped*[4] (Reset/ retained)	Stopped*[4] (Reset/ retained)	Stopped*[4] (Reset/ retained)	Stopped (Reset)
	IIC2	Operating	Operating	Operating	Stopped*[4] *[5] (Reset/ retained)	Stopped*[4] *[5] (Reset/ retained)	Stopped*[4] *[5] (Reset/ retained)	Stopped (Reset)
	SSU	Operating	Operating	Operating	Stopped (Reset)	Stopped (Reset)	Stopped (Reset)	Stopped (Reset)
	USB	Operating	Operating	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Reset)
	RAM	Operating	Operating	Operating	Stopped (Retained)	Operating	Retained	Retained

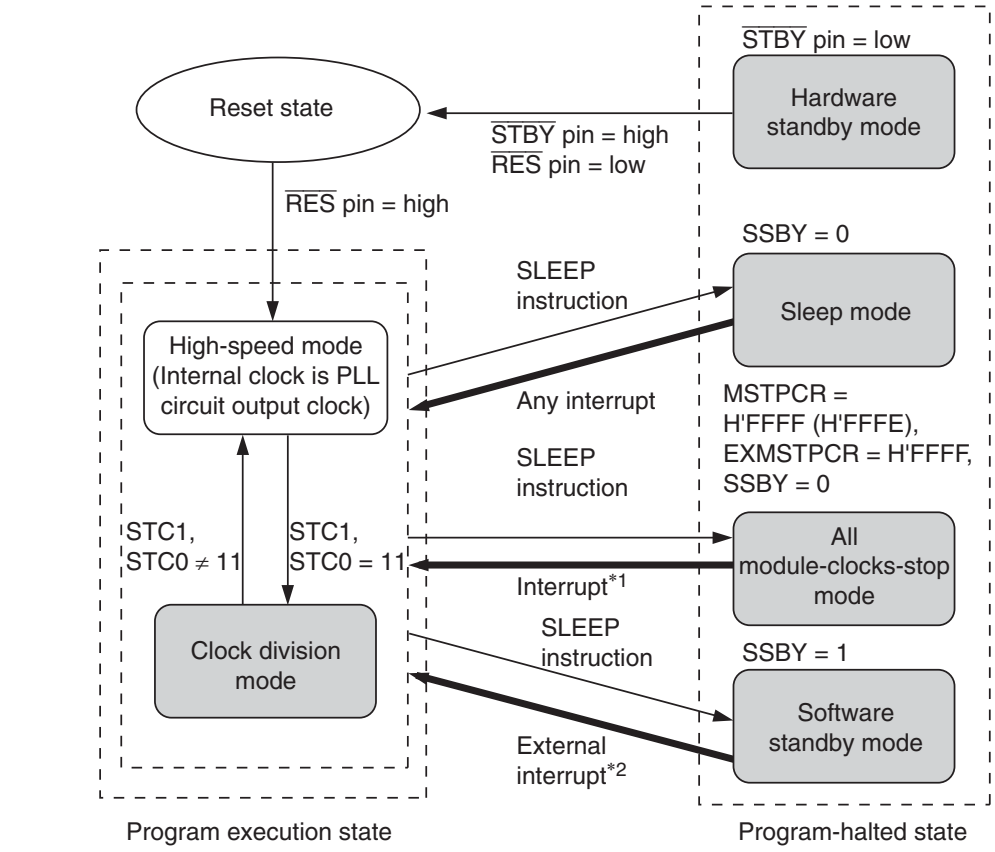
Operating State		High Speed Mode	Clock Division Mode	Sleep Mode	Module Stop Function	All Module Clocks Stop Mode	Software Standby Mode	Hardware Standby Mode
Peripheral functions	I/O	Operating	Operating	Operating	Operating	Retained	Retained	High impedance

Notes: Stopped (Retained) in the table means that internal register values are retained and internal operations are suspended.

Stopped (Reset) in the table means that internal register values and internal states are initialized.

In module stop function, only modules for which a stop setting has been made are stopped (reset or retained).

1. IRQ8 to IRQ15 are not supported by the H8S/2454 group.
2. The active or stopped state can be selected by means of the MSTP0 bit in MSTPCR.
3. Not supported by the H8S/2454 group.
4. TDR, SSR, and RDR are stopped (reset) and other registers are stopped (retained).
5. BC2 to BC0 are stopped (reset) and other registers are stopped (retained).



: Transition after exception handling : Power- down mode

Notes:
- 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.

1. NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$*3, 8-bit timer interrupts, watchdog timer interrupts. (8-bit timer interrupts are valid when MSTP0 = 0.)
2. NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$*3, and resume ($\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$*3 are valid when the corresponding bit in SSIER is 1.)
3. $\overline{\text{IRQ8}}$ to $\overline{\text{IRQ15}}$ are not supported by the H8S/2454 group.

Figure 25.1 Mode Transitions

25.1 Register Descriptions

The registers relating to the power-down mode are shown below. For details on the PLL control register (PLLCR), see section 24.1.2, PLL Control Register (PLLCR).

- PLL control register (PLLCR)
- Standby control register (SBYCR)
- Module stop control register H (MSTPCRH)
- Module stop control register L (MSTPCRL)
- Extension module stop control register H (EXMSTPCRH)
- Extension module stop control register L (EXMSTPCRL)
- RAM module stop control register H (RMMSTPCRH)
- RAM module stop control register L (RMMSTPCRL)

25.1.1 Standby Control Register (SBYCR)

SBYCR performs software standby mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby This bit specifies the transition mode after executing the SLEEP instruction 0: Shifts to sleep mode after the SLEEP instruction is executed 1: Shifts to software standby mode after the SLEEP instruction is executed This bit does not change from 1 when clearing the software standby mode by using external interrupts and shifting to normal operation. This bit should be written 0 when clearing.
6	OPE	1	R/W	Output Port Enable Specifies whether the output of the address bus and bus control signals ($\overline{CS0}$ to $\overline{CS7}$, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$, $\overline{UCAS}$, $\overline{LCAS}$) is retained or set to the high-impedance state in software standby mode. 0: In software standby mode, address bus and bus control signals are high-impedance 1: In software standby mode, address bus and bus control signals retain output state

Bit	Bit Name	Initial Value	R/W	Description
5	—	0	—	Reserved This bit is always read as 0. The initial value should not be changed.
4	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
3	STS3	1	R/W	Standby Timer Select 3 to 0 These bits select the time the MCU waits for the clock to stabilize when software standby mode is cleared by an external interrupt. With crystal oscillation, see table 25.2 and make a selection according to the operating frequency so that the standby time is at least the oscillation stabilization time. With an external clock, a PLL circuit stabilization time is necessary. See table 25.2 to set the standby time. When DRAM is used and self-refreshing in the software standby state is selected, note that the DRAM's tRAS (self-refresh RAS pulse width) specification must be satisfied. 0000: Setting prohibited 0001: Setting prohibited 0010: Setting prohibited 0011: Setting prohibited 0100: Setting prohibited 0101: Standby time = 64 states 0110: Standby time = 512 states 0111: Standby time = 1024 states 1000: Standby time = 2048 states 1001: Standby time = 4096 states 1010: Standby time = 16384 states 1011: Standby time = 32768 states 1100: Standby time = 65536 states 1101: Standby time = 131072 states 1110: Standby time = 262144 states 1111: Standby time = 524288 states
2	STS2	1	R/W	
1	STS1	1	R/W	
0	STS0	1	R/W	

25.1.2 Module Stop Control Registers H and L (MSTPCRH, MSTPCRL)

MSTPCR performs module stop mode control. Setting a bit to 1, the corresponding module enters module stop mode, while clearing the bit to 0 clears the module stop mode.

- MSTPCRH

Bit	Bit Name	Initial Value	R/W	Module
15	ACSE	0	R/W	All Module Clocks Stop Mode Enable Enables or disables all module clocks stop mode, in which, when the CPU executes a SLEEP instruction after module stop mode has been set for all the on-chip peripheral functions controlled by MSTPCR or the on-chip peripheral functions except the TMR. 0: All module clocks stop mode disabled 1: All module clocks stop mode enabled
14	MSTP14	0	R/W	EXDMA controller (EXDMAC)*
13	MSTP13	0	R/W	DMA controller (DMAC)
12	MSTP12	0	R/W	Data transfer controller (DTC)
11	MSTP11	1	R/W	16-bit timer pulse unit 0 (TPU_0)
10	MSTP10	1	R/W	Programmable pulse generator (PPG)
9	MSTP9	1	R/W	16-bit timer pulse unit 1 (TPU_1)
8	MSTP8	1	R/W	D/A converter (channels 2 and 3)

Note: * Not supported by the H8S/2454 group.

- MSTPCRL

Bit	Bit Name	Initial Value	R/W	Module
7	MSTP7	1	R/W	A/D converter unit 1
6	MSTP6	1	R/W	A/D converter unit 0
5	MSTP5	1	R/W	Serial communication interface 4 (SCI_4)
4	MSTP4	1	R/W	Serial communication interface 3 (SCI_3)
3	MSTP3	1	R/W	Serial communication interface 2 (SCI_2)
2	MSTP2	1	R/W	Serial communication interface 1 (SCI_1)
1	MSTP1	1	R/W	Serial communication interface 0 (SCI_0)
0	MSTP0	1	R/W	8-bit timer (TMR)

25.1.3 Extension Module Stop Control Registers H and L (EXMSTPCRH, EXMSTPCRL)

EXMSTPCR performs all module clocks stop mode control with MSTPCR. When entering all module clocks stop mode, set EXMSTPCR to H'FFFF.

- EXMSTPCRH

Bit	Bit Name	Initial Value	R/W	Module
15	MSTP31	1	R/W	—
14	MSTP30	1	R/W	—
13	MSTP29	1	R/W	—
12	MSTP28	1	R/W	—
11	MSTP27	1	R/W	—
10	MSTP26	1	R/W	—
9	MSTP25	1	R/W	—
8	MSTP24	1	R/W	—

- EXMSTPCRL

Bit	Bit Name	Initial Value	R/W	Module
7	MSTP23	1	R/W	Synchronous serial communication unit (SSU)
6	MSTP22	1	R/W	I^2C bus interface 2_3 (IIC2_3)
5	MSTP21	1	R/W	I^2C bus interface 2_2 (IIC2_2)
4	MSTP20	1	R/W	I^2C bus interface 2_1 (IIC2_1)
3	MSTP19	1	R/W	I^2C bus interface 2_0 (IIC2_0)
2	MSTP18	1	R/W	USB function module (USB) (system clock)
1	MSTP17	1	R/W	USB function module (USB) (48 MHz clock)
0	MSTP16	1	R/W	—

25.1.4 RAM Module Stop Control Registers H and L (RMMSTPCRH, RMMSTPCRL)

Setting bits MSTP32 to MSTP39 to 1 stops the corresponding on-chip RAM area. During access to an on-chip RAM area, do not set bits MSTP32 to MSTP39 corresponding to the area to 1. While bit RAME in SYSCR is 1, and bits MSTP32 to MSTP39 are 1, do not access the corresponding RAM area.

- RMMSTPCRH

Bit	Bit Name	Initial Value	R/W	Module
15	MSTP47	0	R/W	—
14	MSTP46	0	R/W	—
13	MSTP45	0	R/W	—
12	MSTP44	0	R/W	—
11	MSTP43	0	R/W	—
10	MSTP42	0	R/W	—
9	MSTP41	0	R/W	—
8	MSTP40	0	R/W	—

- RMMSTPCRL

Bit	Bit Name	Initial Value	R/W	Module
7	MSTP39	0	R/W	On-chip RAM_7 (H'FEC000 to H'FEDFFF)*
6	MSTP38	0	R/W	On-chip RAM_6 (H'FEE000 to H'FEFFFF)*
5	MSTP37	0	R/W	On-chip RAM_5 (H'FF0000 to H'FF1FFF)
4	MSTP36	0	R/W	On-chip RAM_4 (H'FF2000 to H'FF3FFF)
3	MSTP35	0	R/W	On-chip RAM_3 (H'FF4000 to H'FF5FFF)
2	MSTP34	0	R/W	On-chip RAM_2 (H'FF6000 to H'FF7FFF)
1	MSTP33	0	R/W	On-chip RAM_1 (H'FF8000 to H'FF9FFF)
0	MSTP32	0	R/W	On-chip RAM_0 (H'FFA000 to H'FFBFFF)

Note: * Not supported by the H8S/24568R, H8S/24568, H8S/24565R, H8S/24565, H8S/24561R, H8S/24561, H8S/24548, H8S/24545, and H8S/24541 Groups. Although these bits are readable/writable, only 1 should be written to.

25.2 Operation

25.2.1 Clock Division Mode

When bits STC1 and STC0 in PLLCR are set to 11, a transition is made to clock division mode, and the system clock frequency is divided with respect to the oscillator frequency. Clock division mode is cancelled by clearing bits STC1 and STC0 to a value other than 11. The timings of transition and clearing depend on the STCS bit setting in SCKCR. For the operation at transition and clearing, see section 24.3, System-Clock PLL Circuit and Divider.

If a SLEEP instruction is executed while the SSBY bit in SBYCR is cleared to 0, the chip enters sleep mode. When sleep mode is cleared by an interrupt, clock division mode is restored.

If a SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1, the chip enters software standby mode. When software standby mode is cleared by an external or internal interrupt, clock division mode is restored.

When the $\overline{\text{RES}}$ pin is driven low, the reset state is entered and clock division mode is cleared. The same applies to a reset caused by watchdog timer overflow.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

25.2.2 Sleep Mode

(1) Transition to Sleep Mode

When the SLEEP instruction is executed while the SSBY bit is 0 in SBYCR, the CPU enters the sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registers are retained. Other peripheral functions do not stop.

(2) Exiting Sleep Mode

Sleep mode is exited by any interrupt, or signals at the $\overline{\text{RES}}$, or $\overline{\text{STBY}}$ pins.

- Exiting Sleep Mode by Interrupts:
  When an interrupt occurs, sleep mode is exited and interrupt exception processing starts. Sleep mode is not exited if the interrupt is disabled, or interrupts other than NMI are masked by the CPU.
- Exiting Sleep Mode by $\overline{\text{RES}}$ 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 starts the CPU performing reset exception processing.
- Exiting Sleep Mode by $\overline{\text{STBY}}$ Pin:
  When the $\overline{\text{STBY}}$ pin level is driven low, a transition is made to hardware standby mode.

25.2.3 Software Standby Mode

(1) Transition to Software Standby Mode

If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, software standby mode is entered. In this mode, the CPU, on-chip peripheral functions, and oscillator all stop. However, the contents of the CPU's internal registers, RAM data, and the states of on-chip peripheral functions other than the SCI, IIC, and SSU, and the states of I/O ports, are retained. Whether the address bus and bus control signals are placed in the high-impedance state or retain the output state can be specified by the OPE bit in SBYCR.

In this mode the oscillator stops, and therefore power dissipation is significantly reduced.

(2) Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$*), an internal interrupt (resume of USB), or by means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin. Setting the SSI bit in SSIER to 1 enables $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$* to be used as software standby mode clearing sources.

- Clearing with an Interrupt:

  When an NMI or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$* interrupt request signal is input, or if the USB module receives the resume signal from up-stream in the suspended state, USB clock oscillation starts, and stable clocks are supplied to the entire LSI after the elapse of the time set in bits STS3 to STS0 in SBYCR. Then, software standby mode is cleared, and interrupt exception handling is started.

  When clearing software standby mode with an $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$* interrupt, set the corresponding enable bit to 1 and ensure that no interrupt with a higher priority than interrupts $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$* is generated. Software standby mode cannot be cleared if the interrupt has been masked on the CPU side or has been designated as a DTC activation source.

Note: * $\overline{\text{IRQ8}}$ to $\overline{\text{IRQ15}}$ are not supported by the H8S/2454 group.

- Clearing with the $\overline{\text{RES}}$ Pin:

  When the $\overline{\text{RES}}$ pin is driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire LSI. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin goes high, the CPU begins reset exception handling.

- Clearing with the $\overline{\text{STBY}}$ Pin:

  When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Note: * The $\overline{\text{IRQ8}}$ to $\overline{\text{IRQ15}}$ are not supported by the H8S/2454 group.

(3) Setting Oscillation Stabilization Time after Clearing Software Standby Mode

Bits STS3 to STS0 in SBYCR should be set as described below.

- Using a Crystal Resonator:

  Set bits STS3 to STS0 so that the standby time is more than the oscillation stabilization time.

  Table 25.2 shows the standby times for operating frequencies and settings of bits STS3 to STS0.

- Using an External Clock:

  A PLL circuit stabilization time is necessary. See table 25.2 to set the wait time.

Table 25.2 Oscillation Stabilization Time Settings

STS3	STS2	STS1	STS0	Standby Time	φ* [MHz] 33	25	20	13	10	8	Unit
0	0	0	0	Reserved	—	—	—	—	—	—	μs
			1	Reserved	—	—	—	—	—	—	
		1	0	Reserved	—	—	—	—	—	—	
			1	Reserved	—	—	—	—	—	—	
	1	0	0	Reserved	—	—	—	—	—	—	
			1	64	1.9	2.6	3.2	4.9	6.4	8.0	
		1	0	512	15.5	20.5	25.6	39.4	51.2	64.0	
			1	1024	31.0	41.0	51.2	78.8	102.4	128.0	
1	0	0	0	2048	62.1	81.9	102.4	157.5	204.8	256.0	
			1	4096	0.12	0.16	0.20	0.32	0.41	0.51	ms
		1	0	16384	0.50	0.66	0.82	1.26	1.64	2.05	
			1	32765	0.99	1.31	1.64	2.52	3.28	4.10	
	1	0	0	65536	1.99	2.62	3.28	5.04	6.55	8.19	
			1	131072	3.97	5.24	6.55	10.08	13.11	16.38	
		1	0	262144	7.94	10.49	13.11	20.16	26.21	32.77	
			1	524288	15.89	20.97	26.21	40.33	52.43	65.54	

Note: * φ is the frequency divider output.

(4) Software Standby Mode Application Example

Figure 25.2 shows an example in which a transition is made to software standby mode at the falling edge on the NMI pin, and software standby mode is cleared at the rising edge on the NMI pin.

In this example, after an NMI interrupt is accepted with the NMIEG bit in INTCR cleared to 0 (falling edge specification), the NMIEG bit is set to 1 (rising edge specification). And after the SSBY bit is set to 1, a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then cleared at the rising edge on the NMI pin.

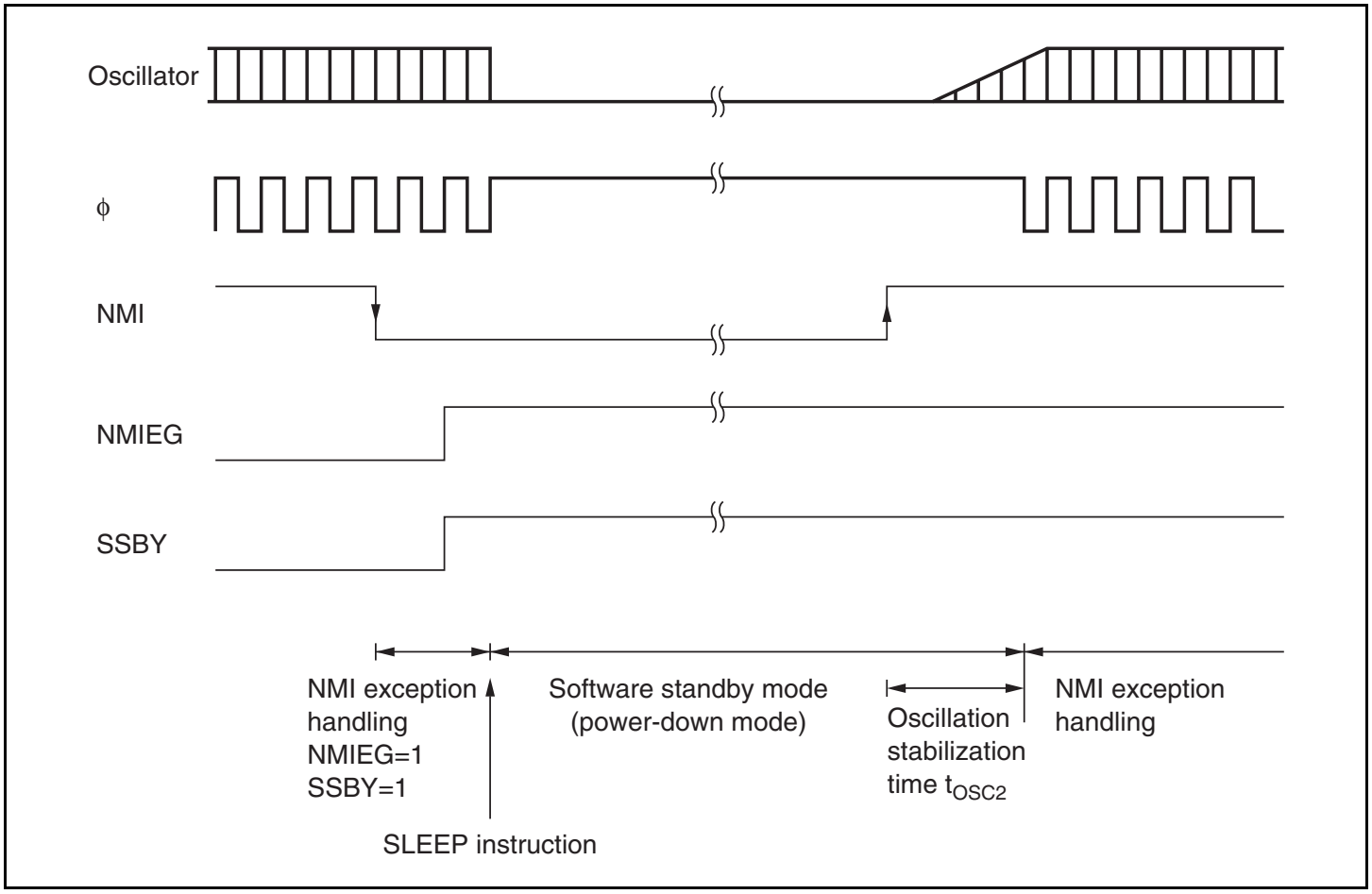


Figure 25.2 Software Standby Mode Application Example

25.2.4 Hardware Standby Mode

(1) Transition to Hardware Standby Mode

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode from any mode.

In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power dissipation. As long as the prescribed voltage is supplied, on-chip RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the $\overline{\text{STBY}}$ pin low. Do not change the state of the mode pins (MD2 to MD0) while this LSI is in hardware standby mode.

(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 set and clock oscillation is started. Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator stabilizes (for details on the oscillation stabilization time, see table 25.2). 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.

(3) Hardware Standby Mode Timing

Figure 25.3 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, waiting for the oscillation stabilization time, then changing the $\overline{\text{RES}}$ pin from low to high.

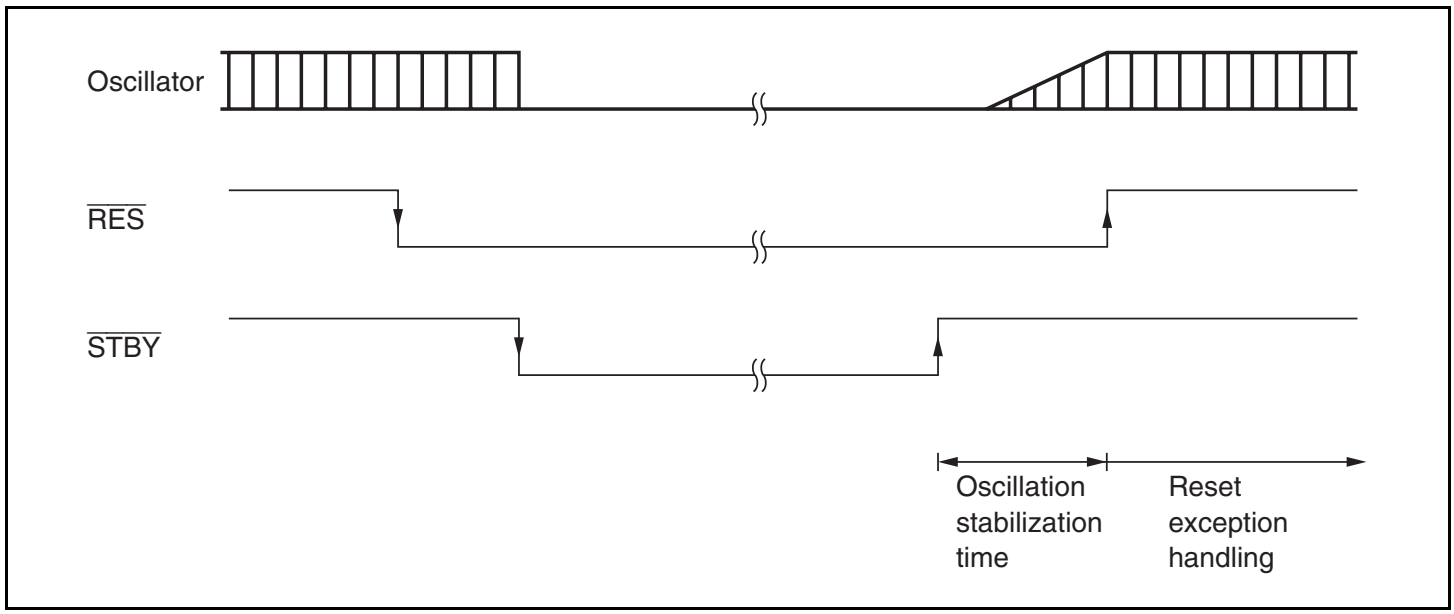


Figure 25.3 Hardware Standby Mode Timing

(4) Hardware Standby Mode Timing when Power Is Supplied

When entering hardware standby mode immediately after the power is supplied, the $\overline{\text{RES}}$ signal must be driven low for a given period with retaining the $\overline{\text{STBY}}$ signal high. After the $\overline{\text{RES}}$ signal is canceled, drive the $\overline{\text{STBY}}$ signal low.

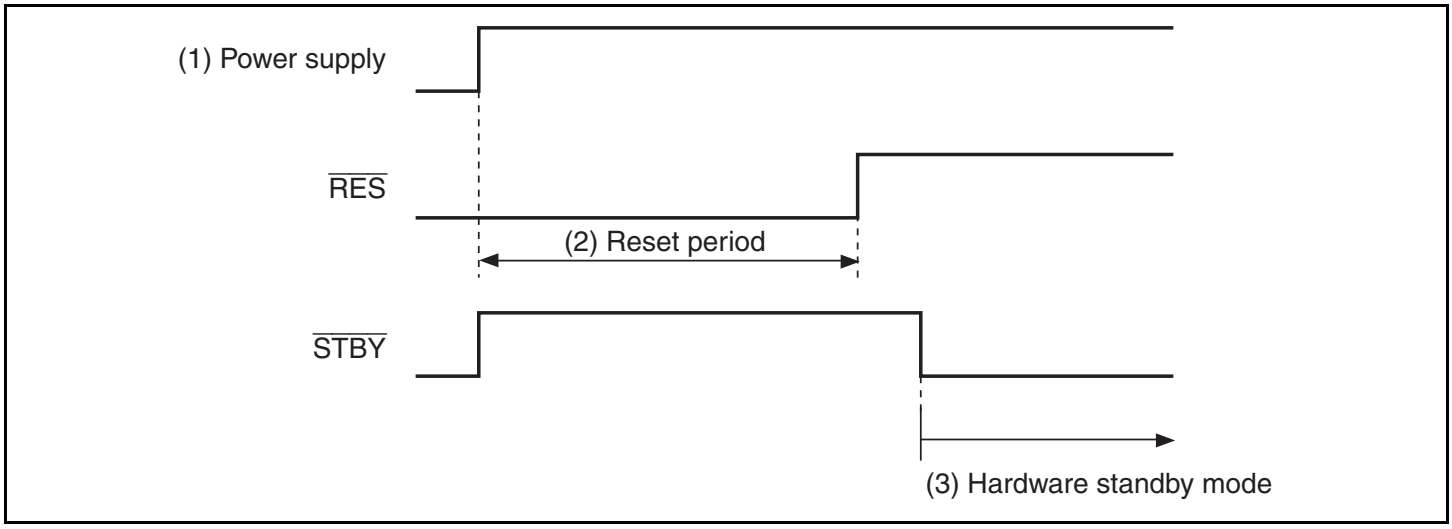


Figure 25.4 Hardware Standby Mode Timing when Power Is Supplied

25.2.5 Module Stop Function

Module stop function can be set for individual on-chip peripheral modules.

When an MSTP bit in MSTPCR, EXMSTPCR, or RMMSTPCR is set to 1, the corresponding module stops operation at the end of the bus cycle and a transition is made to module stop state. The CPU continues operating independently.

When an MSTP bit is cleared to 0, the corresponding module stop state is cleared and the module starts operating at the end of the bus cycle. In module stop state, part of SCI registers and the internal state of SSU are reset but the internal states of the other modules are retained.

After reset clearance, all modules other than the EXDMAC*, DMAC, DTC, and on-chip RAM are in module stop state.

The module registers that are set in module stop state cannot be read or written to.

The module-stop function for RAM is only effective for on-chip RAM. When an area of on-chip RAM is set up as an external address space by bits RAME and EXPE in SYSCR, the resulting external space is accessible regardless of the module-stop setting. Table 25.3 lists the kinds of operation in case of access to the on-chip RAM area.

Note: * The EXDMAC is not supported by the H8S/2454 Group.

Table 25.3 Combinations of SYSCR Settings and Operation in Access to On-Chip RAM

Register Settings				
RAME	**EXPE**	**mstp**	**Target for Access**	**Description**
1	X	1	—	This area is not readable/writable and access is prohibited.
		0	On-chip RAM	
0	1	X	External address space	
	0	X	—	This area is not readable/writable and access is prohibited.

25.2.6 All Module Clocks Stop Mode

When the ACSE bit in MSTPCRH is set to 1 and module stop mode is set for all the on-chip peripheral functions controlled by MSTPCR or EXMSTPCR (MSTPCR = H'FFFF, EXMSTPCR = H'FFFF), or for all the on-chip peripheral functions except the 8-bit timer (MSTPCR = H'FFFE, EXMSTPCR = H'FFFF), executing a SLEEP instruction while the SSBY bit in SBYCR is cleared to 0 will cause all the on-chip peripheral functions (except the 8-bit timer and watchdog timer), the bus controller, and the I/O ports to stop operating, and a transition to be made to all module clocks stop mode at the end of the bus cycle.

Operation or stopping of the 8-bit timer can be selected by means of the MSTP0 bit.

To further reduce the current consumption in all module clocks stop mode, stop the modules controlled by RMMSTPCR (RMMSTPCR = H'FFFF).

All module clocks stop mode is cleared by an external interrupt (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ15}}$* pins), $\overline{\text{RES}}$ pin input, or an internal interrupt (8-bit timer, watchdog timer), and the CPU returns to the normal program execution state via the exception handling state. All module clocks stop mode is not cleared if interrupts are disabled, if interrupts other than NMI are masked by the CPU, or if the relevant interrupt is designated as a DTC activation source.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Note: * $\overline{\text{IRQ8}}$ to $\overline{\text{IRQ15}}$ are not supported by the H8S/2454 group.

25.3 ϕ Clock Output Control

Output of the ϕ clock can be controlled by means of the PSTOP bit in SCKCR, and DDR for the corresponding port. When the PSTOP bit is set to 1, the ϕ clock stops at the end of the bus cycle, and ϕ output goes high. ϕ clock output is enabled when the PSTOP bit is cleared to 0. When DDR for the corresponding port is cleared to 0, ϕ clock output is disabled and input port mode is set. Table 25.4 shows the state of the ϕ pin in each processing state.

Table 25.4 ϕ Pin State in Each Processing State

Register Setting						
DDR	PSTOP	Normal Operating State	Sleep Mode	Software Standby Mode	Hardware Standby Mode	All Module Clocks Stop Mode
0	X	High impedance	High impedance	High impedance	High impedance	High impedance
1	0	ϕ output	ϕ output	Fixed high	High impedance	ϕ output
1	1	Fixed high	Fixed high	Fixed high	High impedance	Fixed high

25.4 SDRAMϕ Clock Output Control

Output of the SDRAMϕ clock can be controlled by the SDPSTP bit in SCKCR. When the SDPSTP bit is set to 1, the SDRAMϕ clock stops at the end of the bus cycle and the pin can be used as a general port. SDRAMϕ clock output is enabled when the SDPSTP bit is cleared to 0 regardless of the DDR value. Table 25.5 shows the state of the SDRAMϕ pin in each processing state.

Note: The SDRAM interface is not supported by the H8S/2456 group and H8S/2454 group.

Table 25.5 SDRAMϕ Pin State in Each Processing State

Register Setting						
SDPSTP	DDR	Normal Operating State	Sleep Mode	Software Standby Mode	Hardware Standby Mode	All Module Clocks Stop Mode
0	×	SDRAMϕ output	SDRAMϕ output	Fixed low	High impedance	SDRAMϕ output
1	0	High impedance	High impedance	High impedance	High impedance	High impedance
1	1	PH1/CS5/RAS5 output	H1/CS5/RAS5 output	H1/CS5/RAS5 output	High impedance	H1/CS5/RAS5 output

Note: SDRAM is not available in the H8S/2456 and H8S/2454 Groups.
In these products, this pin functions as a general pin regardless of the SDPSTP bit setting.

25.5 Usage Notes

25.5.1 I/O Port Status

In software standby mode, I/O port states are retained. Therefore, there is no reduction in current dissipation for the output current when a high-level signal is output.

25.5.2 Current Dissipation during Oscillation Stabilization Standby Period

Current dissipation increases during the oscillation stabilization standby period.

25.5.3 EXDMAC, DMAC, and DTC Module Stop

Depending on the operating status of the EXDMAC, DMAC, or DTC, the MSTP14 to MSTP13 and may not be set to 1. Setting of the EXDMAC, DMAC, or DTC module stop mode should be carried out only when the respective module is not activated.

For details, see section 8, EXDMA Controller (EXDMAC), section 7, DMA Controller (DMAC), and section 9, Data Transfer Controller (DTC).

Note: The EXDMAC is not supported by the H8S/2454 group.

25.5.4 On-Chip Peripheral Module Interrupts

Relevant interrupt operations cannot be performed in module stop mode. Consequently, if module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DMAC or DTC activation source.

Interrupts should therefore be disabled before entering module stop mode.

Note: The EXDMAC is not supported by the H8S/2454 group.

25.5.5 Writing to MSTPCR, EXMSTPCR, and RMMSTPCR

MSTPCR, EXMSTPCR, and RMMSTPCR should only be written to by the CPU.

25.5.6 Notes on Clock Division Mode

The following points should be noted in clock division mode.

- Select the clock division ratio by the STC1 and STC0 bits so that the frequency of ϕ is within the operation guaranteed range of clock cycle time tcyc shown in the Electrical Characteristics. In other words, the frequency of ϕ must be 8 MHz or higher; be careful not so specify ϕ < 8 MHz.
- All the on-chip peripheral modules operate on the ϕ. Therefore, note that the time processing of modules such as a timer and SCI differ before and after changing the clock division ratio. In addition, the wait time for clearing software standby mode differs by changing the clock division ratio.
- Note that the frequency of ϕ will be changed by changing the clock division ratio.

Section 26 List of Registers

The address list gives information on the on-chip register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register addresses (address order)

- Registers are listed from the lower allocation addresses.
- Registers are classified by functional modules.
- The access size is indicated.

2. Register bits

- Bit configurations of the registers are described in the same order as the register addresses.
- Reserved bits are indicated by — in the bit name column.
- For the registers of 16 or 32 bits, the MSB is described first.

3. Register states in each operating mode

- Register states are described in the same order as the register addresses.
- The register states described here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, see the section on that on-chip peripheral module.

26.1 Register Addresses (Address Order)

The data bus width indicates the numbers of bits by which the register is accessed.

The number of access states indicates the number of states based on the specified reference clock.

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
DTC mode register A	MRA	8	H'BC00 to H'BFFF	DTC	16/32	2
DTC source address register	SAR	24		DTC	16/32	2
DTC mode register B	MRB	8		DTC	16/32	2
DTC destination address register	DAR	24		DTC	16/32	2
DTC transfer count register A	CRA	16		DTC	16/32	2
DTC transfer count register B	CRB	16		DTC	16/32	2
Interrupt flag register 0	IFR0	8	H'FB00	USB	8	3
Interrupt flag register 1	IFR1	8	H'FB01	USB	8	3
Interrupt flag register 2	IFR2	8	H'FB02	USB	8	3
Interrupt enable register 0	IER0	8	H'FB08	USB	8	3
Interrupt enable register 1	IER1	8	H'FB09	USB	8	3
Interrupt enable register 2	IER2	8	H'FB0A	USB	8	3
Interrupt select register 0	ISR0	8	H'FB10	USB	8	3
Interrupt select register 1	ISR1	8	H'FB11	USB	8	3
Interrupt select register 2	ISR2	8	H'FB12	USB	8	3
EP0i data register	EPDR0i	32	H'FB20	USB	8	3
EP0o data register	EPDR0o	32	H'FB24	USB	8	3
EP0s data register	EPDR0s	32	H'FB28	USB	8	3
EP1 data register	EPDR1	32	H'FB30	USB	8	3
EP2 data register	EPDR2	32	H'FB34	USB	8	3
EP3 data register	EPDR3	32	H'FB38	USB	8	3
EP0o receive data size register	EPSZ0o	8	H'FB80	USB	8	3
EP1 receive data size register	EPSZ1	8	H'FB81	USB	8	3
Data status register 0	DASTS0	8	H'FB88	USB	8	3
Data status register 1	DASTS1	8	H'FB89	USB	8	3
Trigger register 0	TRG0	8	H'FB90	USB	8	3
Trigger register 1	TRG1	8	H'FB91	USB	8	3

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
FIFO clear register 0	FCLR0	8	H'FB98	USB	8	3
FIFO clear register 1	FCLR1	8	H'FB99	USB	8	3
Endpoint stall register 0	EPSTL0	8	H'FBA0	USB	8	3
Endpoint stall register 1	EPSTL1	8	H'FBA1	USB	8	3
Stall status register 1	STLSR1	8	H'FBA9	USB	8	3
DMA transfer setting register	DMAR	8	H'FBB0	USB	8	3
Configuration value register	CVR	8	H'FBB4	USB	8	3
Control register	CTLR	8	H'FBB8	USB	8	3
Endpoint information register	EPIR	32	H'FBC0	USB	8	3
Transceiver test register 0	TRNTREG0	8	H'FBD0	USB	8	3
Transceiver test register 1	TRNTREG1	8	H'FBD1	USB	8	3
RAM module stop control register H	RMMSTPCRH	8	H'FC80	SYSTEM	8	2
RAM module stop control register L	RMMSTPCRL	8	H'FC81	SYSTEM	8	2
USB PLL control register	USPLLCR	8	H'FC82	SYSTEM	8	2
Interrupt priority register L	IPRL	16	H'FC90	INT	16	2
Interrupt priority register M	IPRM	16	H'FC92	INT	16	2
Interrupt priority register N	IPRN	16	H'FC94	INT	16	2
DTC enable register I	DTCERI	8	H'FC96	DTC	16	2
DTC control register	DTCCR	8	H'FC98	DTC	16	2
A/D data register A_1	ADDRA_1	16	H'FCA0	A/D_1	16	2
A/D data register B_1	ADDRB_1	16	H'FCA2	A/D_1	16	2
A/D data register C_1	ADDRC_1	16	H'FCA4	A/D_1	16	2
A/D data register D_1	ADDRD_1	16	H'FCA6	A/D_1	16	2
A/D data register E_1	ADDRE_1	16	H'FCA8	A/D_1	16	2
A/D data register F_1	ADDRF_1	16	H'FCAA	A/D_1	16	2
A/D data register G_1	ADDRG_1	16	H'FCAC	A/D_1	16	2
A/D data register H_1	ADDRH_1	16	H'FCAE	A/D_1	16	2
A/D control/status register_1	ADCSR_1	8	H'FCB0	A/D_1	16	2
A/D control register_1	ADCR_1	8	H'FCB1	A/D_1	16	2
Timer start register_1	TSTR_1	8	H'FCC0	TPU	16	2
Timer synchronous register_1	TSYR_1	8	H'FCC1	TPU	16	2
Timer control register_6	TCR_6	8	H'FCD0	TPU_6	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Timer mode register_6	TMDR_6	8	H'FCD1	TPU_6	16	2
Timer I/O control register H_6	TIORH_6	8	H'FCD2	TPU_6	16	2
Timer I/O control register L_6	TIORL_6	8	H'FCD3	TPU_6	16	2
Timer interrupt enable register_6	TIER_6	8	H'FCD4	TPU_6	16	2
Timer status register_6	TSR_6	8	H'FCD5	TPU_6	16	2
Timer counter_6	TCNT_6	16	H'FCD6	TPU_6	16	2
Timer general register A_6	TGRA_6	16	H'FCD8	TPU_6	16	2
Timer general register B_6	TGRB_6	16	H'FCDA	TPU_6	16	2
Timer general register C_6	TGRC_6	16	H'FCDC	TPU_6	16	2
Timer general register D_6	TGRD_6	16	H'FCDE	TPU_6	16	2
Timer control register_7	TCR_7	8	H'FCE0	TPU_7	16	2
Timer mode register_7	TMDR_7	8	H'FCE1	TPU_7	16	2
Timer I/O control register_7	TIOR_7	8	H'FCE2	TPU_7	16	2
Timer interrupt enable register_7	TIER_7	8	H'FCE4	TPU_7	16	2
Timer status register_7	TSR_7	8	H'FCE5	TPU_7	16	2
Timer counter_7	TCNT_7	16	H'FCE6	TPU_7	16	2
Timer general register A_7	TGRA_7	16	H'FCE8	TPU_7	16	2
Timer general register B_7	TGRB_7	16	H'FCEA	TPU_7	16	2
Timer control register_8	TCR_8	8	H'FCF0	TPU_8	16	2
Timer mode register_8	TMDR_8	8	H'FCF1	TPU_8	16	2
Timer I/O control register_8	TIOR_8	8	H'FCF2	TPU_8	16	2
Timer interrupt enable register_8	TIER_8	8	H'FCF4	TPU_8	16	2
Timer status register_8	TSR_8	8	H'FCF5	TPU_8	16	2
Timer counter_8	TCNT_8	16	H'FCF6	TPU_8	16	2
Timer general register A_8	TGRA_8	16	H'FCF8	TPU_8	16	2
Timer general register B_8	TGRB_8	16	H'FCFA	TPU_8	16	2
Timer control register_9	TCR_9	8	H'FD00	TPU_9	16	2
Timer mode register_9	TMDR_9	8	H'FD01	TPU_9	16	2
Timer I/O control register H_9	TIORH_9	8	H'FD02	TPU_9	16	2
Timer I/O control register L_9	TIORL_9	8	H'FD03	TPU_9	16	2
Timer interrupt enable register_9	TIER_9	8	H'FD04	TPU_9	16	2
Timer status register_9	TSR_9	8	H'FD05	TPU_9	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Timer counter_9	TCNT_9	16	H'FD06	TPU_9	16	2
Timer general register A_9	TGRA_9	16	H'FD08	TPU_9	16	2
Timer general register B_9	TGRB_9	16	H'FD0A	TPU_9	16	2
Timer general register C_9	TGRC_9	16	H'FD0C	TPU_9	16	2
Timer general register D_9	TGRD_9	16	H'FD0E	TPU_9	16	2
Timer control register_10	TCR_10	8	H'FD10	TPU_10	16	2
Timer mode register_10	TMDR_10	8	H'FD11	TPU_10	16	2
Timer I/O control register_10	TIOR_10	8	H'FD12	TPU_10	16	2
Timer interrupt enable register_10	TIER_10	8	H'FD14	TPU_10	16	2
Timer status register_10	TSR_10	8	H'FD15	TPU_10	16	2
Timer counter_10	TCNT_10	16	H'FD16	TPU_10	16	2
Timer general register A_10	TGRA_10	16	H'FD18	TPU_10	16	2
Timer general register B_10	TGRB_10	16	H'FD1A	TPU_10	16	2
Timer control register_11	TCR_11	8	H'FD20	TPU_11	16	2
Timer mode register_11	TMDR_11	8	H'FD21	TPU_11	16	2
Timer I/O control register_11	TIOR_11	8	H'FD22	TPU_11	16	2
Timer interrupt enable register_11	TIER_11	8	H'FD24	TPU_11	16	2
Timer status register_11	TSR_11	8	H'FD25	TPU_11	16	2
Timer counter_11	TCNT_11	16	H'FD26	TPU_11	16	2
Timer general register A_11	TGRA_11	16	H'FD28	TPU_11	16	2
Timer general register B_11	TGRB_11	16	H'FD2A	TPU_11	16	2
Port 1 open drain control register	P1ODR	8	H'FD40	PORT	8	2
Port 2 open drain control register	P2ODR	8	H'FD41	PORT	8	2
Port 5 open drain control register	P5ODR	8	H'FD42	PORT	8	2
Port 6 open drain control register	P6ODR	8	H'FD43	PORT	8	2
Port 8 open drain control register	P8ODR	8	H'FD44	PORT	8	2
Port B open drain control register	PBODR	8	H'FD45	PORT	8	2
Port C open drain control register	PCODR	8	H'FD46	PORT	8	2
Port D open drain control register	PDODR	8	H'FD47	PORT	8	2
Port E open drain control register	PEODR	8	H'FD48	PORT	8	2
Port F open drain control register	PFODR	8	H'FD49	PORT	8	2
Port G open drain control register	PGODR	8	H'FD4A	PORT	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Port H open drain control register	PHODR	8	H'FD4B	PORT	8	2
Port J open drain control register	PJODR	8	H'FD4C	PORT	8	2
I²C bus control register A_0	ICCRA_0	8	H'FD58	IIC2_0	8	2
I²C bus control register B_0	ICCRB_0	8	H'FD59	IIC2_0	8	2
I²C bus mode register_0	ICMR_0	8	H'FD5A	IIC2_0	8	2
I²C bus interrupt enable register_0	ICIER_0	8	H'FD5B	IIC2_0	8	2
I²C bus status register_0	ICSR_0	8	H'FD5C	IIC2_0	8	2
Slave address register_0	SAR_0	8	H'FD5D	IIC2_0	8	2
I²C transfer data register_0	ICDRT_0	8	H'FD5E	IIC2_0	8	2
I²C receive data register_0	ICDRR_0	8	H'FD5F	IIC2_0	8	2
I²C bus control register A_1	ICCRA_1	8	H'FD60	IIC2_1	8	2
I²C bus control register B_1	ICCRB_1	8	H'FD61	IIC2_1	8	2
I²C bus mode register_1	ICMR_1	8	H'FD62	IIC2_1	8	2
I²C bus interrupt enable register_1	ICIER_1	8	H'FD63	IIC2_1	8	2
I²C bus status register_1	ICSR_1	8	H'FD64	IIC2_1	8	2
Slave address register_1	SAR_1	8	H'FD65	IIC2_1	8	2
I²C transfer data register_1	ICDRT_1	8	H'FD66	IIC2_1	8	2
I²C receive data register_1	ICDRR_1	8	H'FD67	IIC2_1	8	2
I²C bus control register A_2	ICCRA_2	8	H'FD68	IIC2_2	8	2
I²C bus control register B_2	ICCRB_2	8	H'FD69	IIC2_2	8	2
I²C bus mode register_2	ICMR_2	8	H'FD6A	IIC2_2	8	2
I²C bus interrupt enable register_2	ICIER_2	8	H'FD6B	IIC2_2	8	2
I²C bus status register_2	ICSR_2	8	H'FD6C	IIC2_2	8	2
Slave address register_2	SAR_2	8	H'FD6D	IIC2_2	8	2
I²C transfer data register_2	ICDRT_2	8	H'FD6E	IIC2_2	8	2
I²C receive data register_2	ICDRR_2	8	H'FD6F	IIC2_2	8	2
I²C bus control register A_3	ICCRA_3	8	H'FD70	IIC2_3	8	2
I²C bus control register B_3	ICCRB_3	8	H'FD71	IIC2_3	8	2
I²C bus mode register_3	ICMR_3	8	H'FD72	IIC2_3	8	2
I²C bus interrupt enable register_3	ICIER_3	8	H'FD73	IIC2_3	8	2
I²C bus status register_3	ICSR_3	8	H'FD74	IIC2_3	8	2
Slave address register_3	SAR_3	8	H'FD75	IIC2_3	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
I^2C transfer data register_3	ICDRT_3	8	H'FD76	IIC2_3	8	2
I^2C receive data register_3	ICDRR_3	8	H'FD77	IIC2_3	8	2
Serial expansion mode register_2	SEMR_2	8	H'FDA8	SCI_2	8	2
SS control register H	SSCRH	8	H'FDB0	SSU	16	2
SS control register L	SSCRL	8	H'FDB1	SSU	16	2
SS mode register	SSMR	8	H'FDB2	SSU	16	2
SS enable register	SSER	8	H'FDB3	SSU	16	2
SS status register	SSSR	8	H'FDB4	SSU	16	2
SS control register 2	SSCR2	8	H'FDB5	SSU	16	2
SS transmit data register 0	SSTDR0	8	H'FDB6	SSU	16	2
SS transmit data register 1	SSTDR1	8	H'FDB7	SSU	16	2
SS transmit data register 2	SSTDR2	8	H'FDB8	SSU	16	2
SS transmit data register 3	SSTDR3	8	H'FDB9	SSU	16	2
SS receive data register 0	SSRDR0	8	H'FDBA	SSU	16	2
SS receive data register 1	SSRDR1	8	H'FDBB	SSU	16	2
SS receive data register 2	SSRDR2	8	H'FDBC	SSU	16	2
SS receive data register 3	SSRDR3	8	H'FDBD	SSU	16	2
EXDMA source address register_2	EDSAR_2	32	H'FDE0	EXDMAC_2*[3]	16	2
EXDMA destination address register_2	EDDAR_2	32	H'FDE4	EXDMAC_2*[3]	16	2
EXDMA transfer count register_2	EDTCR_2	32	H'FDE8	EXDMAC_2*[3]	16	2
EXDMA mode control register_2	EDMDR_2	16	H'FDEC	EXDMAC_2*[3]	16	2
EXDMA address control register_2	EDACR_2	16	H'FDEE	EXDMAC_2*[3]	16	2
EXDMA source address register_3	EDSAR_3	32	H'FDF0	EXDMAC_3*[3]	16	2
EXDMA destination address register_3	EDDAR_3	32	H'FDF4	EXDMAC_3*[3]	16	2
EXDMA transfer count register_3	EDTCR_3	32	H'FDF8	EXDMAC_3*[3]	16	2
EXDMA mode control register_3	EDMDR_3	16	H'FDFC	EXDMAC_3*[3]	16	2
EXDMA address control register_3	EDACR_3	16	H'FDFE	EXDMAC_3*[3]	16	2
Interrupt priority register A	IPRA	16	H'FE00	INT	16	2
Interrupt priority register B	IPRB	16	H'FE02	INT	16	2
Interrupt priority register C	IPRC	16	H'FE04	INT	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Interrupt priority register D	IPRD	16	H'FE06	INT	16	2
Interrupt priority register E	IPRE	16	H'FE08	INT	16	2
Interrupt priority register F	IPRF	16	H'FE0A	INT	16	2
Interrupt priority register G	IPRG	16	H'FE0C	INT	16	2
Interrupt priority register H	IPRH	16	H'FE0E	INT	16	2
Interrupt priority register I	IPRI	16	H'FE10	INT	16	2
Interrupt priority register J	IPRJ	16	H'FE12	INT	16	2
Interrupt priority register K	IPRK	16	H'FE14	INT	16	2
IRQ pin select register	ITSR	16	H'FE16	INT	16	2
Software standby release IRQ enable register	SSIER	16	H'FE18	INT	16	2
IRQ sense control register H	ISCRH	16	H'FE1A	INT	16	2
IRQ sense control register L	ISCRL	16	H'FE1C	INT	16	2
IrDA control register_0	IrCR_0	8	H'FE1E	IrDA	8	2
Port 1 data direction register	P1DDR	8	H'FE20	PORT	8	2
Port 2 data direction register	P2DDR	8	H'FE21	PORT	8	2
Port 3 data direction register	P3DDR	8	H'FE22	PORT	8	2
Port 5 data direction register	P5DDR	8	H'FE24	PORT	8	2
Port 6 data direction register	P6DDR	8	H'FE25	PORT	8	2
Port 8 data direction register	P8DDR	8	H'FE27	PORT	8	2
Port A data direction register	PADDR	8	H'FE29	PORT	8	2
Port B data direction register	PBDDR	8	H'FE2A	PORT	8	2
Port C data direction register	PCDDR	8	H'FE2B	PORT	8	2
Port D data direction register	PDDDR	8	H'FE2C	PORT	8	2
Port E data direction register	PEDDR	8	H'FE2D	PORT	8	2
Port F data direction register	PFDDR	8	H'FE2E	PORT	8	2
Port G data direction register	PGDDR	8	H'FE2F	PORT	8	2
Port function control register 0	PFCR0	8	H'FE32	PORT	8	2
Port function control register 1	PFCR1	8	H'FE33	PORT	8	2
Port function control register 2	PFCR2	8	H'FE34	PORT	8	2
Port A pull-up MOS control register	PAPCR	8	H'FE36	PORT	8	2
Port B pull-up MOS control register	PBPCR	8	H'FE37	PORT	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Port C pull-up MOS control register	PCPCR	8	H'FE38	PORT	8	2
Port D pull-up MOS control register	PDPCR	8	H'FE39	PORT	8	2
Port E pull-up MOS control register	PEPCR	8	H'FE3A	PORT	8	2
Port 3 open drain control register	P3ODR	8	H'FE3C	PORT	8	2
Port A open drain control register	PAODR	8	H'FE3D	PORT	8	2
Serial mode register_3	SMR_3	8	H'FE40	SCI_3	8	2
Bit rate register_3	BRR_3	8	H'FE41	SCI_3	8	2
Serial control register_3	SCR_3	8	H'FE42	SCI_3	8	2
Transmit data register_3	TDR_3	8	H'FE43	SCI_3	8	2
Serial status register_3	SSR_3	8	H'FE44	SCI_3	8	2
Receive data register_3	RDR_3	8	H'FE45	SCI_3	8	2
Smart card mode register_3	SCMR_3	8	H'FE46	SCI_3	8	2
Serial mode register_4	SMR_4	8	H'FE48	SCI_4	8	2
Bit rate register_4	BRR_4	8	H'FE49	SCI_4	8	2
Serial control register_4	SCR_4	8	H'FE4A	SCI_4	8	2
Transmit data register_4	TDR_4	8	H'FE4B	SCI_4	8	2
Serial status register_4	SSR_4	8	H'FE4C	SCI_4	8	2
Receive data register_4	RDR_4	8	H'FE4D	SCI_4	8	2
Smart card mode register_4	SCMR_4	8	H'FE4E	SCI_4	8	2
Timer control register_3	TCR_3	8	H'FE80	TPU_3	16	2
Timer mode register_3	TMDR_3	8	H'FE81	TPU_3	16	2
Timer I/O control register H_3	TIORH_3	8	H'FE82	TPU_3	16	2
Timer I/O control register L_3	TIORL_3	8	H'FE83	TPU_3	16	2
Timer interrupt enable register_3	TIER_3	8	H'FE84	TPU_3	16	2
Timer status register_3	TSR_3	8	H'FE85	TPU_3	16	2
Timer counter_3	TCNT_3	16	H'FE86	TPU_3	16	2
Timer general register A_3	TGRA_3	16	H'FE88	TPU_3	16	2
Timer general register B_3	TGRB_3	16	H'FE8A	TPU_3	16	2
Timer general register C_3	TGRC_3	16	H'FE8C	TPU_3	16	2
Timer general register D_3	TGRD_3	16	H'FE8E	TPU_3	16	2
Timer control register_4	TCR_4	8	H'FE90	TPU_4	16	2
Timer mode register_4	TMDR_4	8	H'FE91	TPU_4	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Timer I/O control register_4	TIOR_4	8	H'FE92	TPU_4	16	2
Timer interrupt enable register_4	TIER_4	8	H'FE94	TPU_4	16	2
Timer status register_4	TSR_4	8	H'FE95	TPU_4	16	2
Timer counter_4	TCNT_4	16	H'FE96	TPU_4	16	2
Timer general register A_4	TGRA_4	16	H'FE98	TPU_4	16	2
Timer general register B_4	TGRB_4	16	H'FE9A	TPU_4	16	2
Timer control register_5	TCR_5	8	H'FEA0	TPU_5	16	2
Timer mode register_5	TMDR_5	8	H'FEA1	TPU_5	16	2
Timer I/O control register_5	TIOR_5	8	H'FEA2	TPU_5	16	2
Timer interrupt enable register_5	TIER_5	8	H'FEA4	TPU_5	16	2
Timer status register_5	TSR_5	8	H'FEA5	TPU_5	16	2
Timer counter_5	TCNT_5	16	H'FEA6	TPU_5	16	2
Timer general register A_5	TGRA_5	16	H'FEA8	TPU_5	16	2
Timer general register B_5	TGRB_5	16	H'FEAA	TPU_5	16	2
Flash memory control register 1	FLMCR1	8	H'FEB0	FLASH	8	2
Flash memory data block protect register	DFPR	8	H'FEB2	FLASH	8	2
Flash memory status register	FLMSTR	8	H'FEB3	FLASH	8	2
Bus width control register	ABWCR	8	H'FEC0	BSC	16	2
Access state control register	ASTCR	8	H'FEC1	BSC	16	2
Wait control register AH	WTCRAH	8	H'FEC2	BSC	16	2
Wait control register AL	WTCRAL	8	H'FEC3	BSC	16	2
Wait control register BH	WTCRBH	8	H'FEC4	BSC	16	2
Wait control register BL	WTCRBL	8	H'FEC5	BSC	16	2
Read strobe timing control register	RDNCR	8	H'FEC6	BSC	16	2
$\overline{CS}$ assertion period control register H	CSACRH	8	H'FEC8	BSC	16	2
$\overline{CS}$ assertion period control register L	CSACRL	8	H'FEC9	BSC	16	2
Burst ROM interface control register H	BROMCRH	8	H'FECA	BSC	16	2
Burst ROM interface control register L	BROMCRL	8	H'FECB	BSC	16	2
Bus control register	BCR	16	H'FECC	BSC	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Address/data multiplexed I/O control register	MPXCR	8	H'FECF	BSC	16	2
DRAM control register L	DRAMCR	16	H'FED0	BSC	16	2
DRAM access control register H	DRACCRH	8	H'FED2	BSC	16	2
DRAM access control register L	DRACCRL	8	H'FED3	BSC	16	2
Refresh control register	REFCR	16	H'FED4	BSC	16	2
Refresh timer counter	RTCNT	8	H'FED6	BSC	16	2
Refresh time constant register	RTCOR	8	H'FED7	BSC	16	2
Memory address register_0AH	MAR_0AH	16	H'FEE0	DMAC	16	2
Memory address register_0AL	MAR_0AL	16	H'FEE2	DMAC	16	2
I/O address register_0A	IOAR_0A	16	H'FEE4	DMAC	16	2
Transfer count register_0A	ETCR_0A	16	H'FEE6	DMAC	16	2
Memory address register_0BH	MAR_0BH	16	H'FEE8	DMAC	16	2
Memory address register_0BL	MAR_0BL	16	H'FEEA	DMAC	16	2
I/O address register_0B	IOAR_0B	16	H'FEEC	DMAC	16	2
Transfer count register_0B	ETCR_0B	16	H'FEEE	DMAC	16	2
Memory address register_1AH	MAR_1AH	16	H'FEF0	DMAC	16	2
Memory address register_1AL	MAR_1AL	16	H'FEF2	DMAC	16	2
I/O address register_1A	IOAR_1A	16	H'FEF4	DMAC	16	2
Transfer count register_1A	ETCR_1A	16	H'FEF6	DMAC	16	2
Memory address register_1BH	MAR_1BH	16	H'FEF8	DMAC	16	2
Memory address register_1BL	MAR_1BL	16	H'FEFA	DMAC	16	2
I/O address register_1B	IOAR_1B	16	H'FEFC	DMAC	16	2
Transfer count register_1B	ETCR_1B	16	H'FEFE	DMAC	16	2
DMA write enable register	DMAWER	8	H'FF20	DMAC	8	2
DMA terminal control register	DMATCR	8	H'FF21	DMAC	8	2
DMA control register_0A	DMACR_0A	8	H'FF22	DMAC	16	2
DMA control register_0B	DMACR_0B	8	H'FF23	DMAC	16	2
DMA control register_1A	DMACR_1A	8	H'FF24	DMAC	16	2
DMA control register_1B	DMACR_1B	8	H'FF25	DMAC	16	2
DMA band control register H	DMABCRH	8	H'FF26	DMAC	16	2
DMA band control register L	DMABCRL	8	H'FF27	DMAC	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
DTC enable register A	DTCERA	8	H'FF28	DTC	16	2
DTC enable register B	DTCERB	8	H'FF29	DTC	16	2
DTC enable register C	DTCERC	8	H'FF2A	DTC	16	2
DTC enable register D	DTCERD	8	H'FF2B	DTC	16	2
DTC enable register E	DTCERE	8	H'FF2C	DTC	16	2
DTC enable register F	DTCERF	8	H'FF2D	DTC	16	2
DTC enable register G	DTCERG	8	H'FF2E	DTC	16	2
DTC enable register H	DTCERH	8	H'FF2F	DTC	16	2
DTC vector register	DTVECR	8	H'FF30	DTC	16	2
Interrupt control register	INTCR	8	H'FF31	INT	16	2
IRQ enable register	IER	16	H'FF32	INT	16	2
IRQ status register	ISR	16	H'FF34	INT	16	2
Standby control register	SBYCR	8	H'FF3A	SYSTEM	8	2
System clock control register	SCKCR	8	H'FF3B	SYSTEM	8	2
System control register	SYSCR	8	H'FF3D	SYSTEM	8	2
Mode control register	MDCR	8	H'FF3E	SYSTEM	8	2
Module stop control register H	MSTPCRH	8	H'FF40	SYSTEM	8	2
Module stop control register L	MSTPCRL	8	H'FF41	SYSTEM	8	2
Extension module stop control register H	EXMSTPCRH	8	H'FF42	SYSTEM	8	2
Extension module stop control register L	EXMSTPCRL	8	H'FF43	SYSTEM	8	2
PLL control register	PLLCR	8	H'FF45	SYSTEM	8	2
PPG output control register	PCR	8	H'FF46	PPG	8	2
PPG output mode register	PMR	8	H'FF47	PPG	8	2
Next data enable register H	NDERH	8	H'FF48	PPG	8	2
Next data enable register L	NDERL	8	H'FF49	PPG	8	2
Output data register H	PODRH	8	H'FF4A	PPG	8	2
Output data register L	PODRL	8	H'FF4B	PPG	8	2
Next data register H*[1]	NDRHH	8	H'FF4C	PPG	8	2
Next data register L*[1]	NDRLH	8	H'FF4D	PPG	8	2
Next data register H*[1]	NDRHL	8	H'FF4E	PPG	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Next data register L*[1]	NDRLL	8	H'FF4F	PPG	8	2
Port 1 register	PORT1	8	H'FF50	PORT	8	2
Port 2 register	PORT2	8	H'FF51	PORT	8	2
Port 3 register	PORT3	8	H'FF52	PORT	8	2
Port 4 register	PORT4	8	H'FF53	PORT	8	2
Port 5 register	PORT5	8	H'FF54	PORT	8	2
Port 6 register	PORT6	8	H'FF55	PORT	8	2
Port 8 register	PORT8	8	H'FF57	PORT	8	2
Port 9 register	PORT9	8	H'FF58	PORT	8	2
Port A register	PORTA	8	H'FF59	PORT	8	2
Port B register	PORTB	8	H'FF5A	PORT	8	2
Port C register	PORTC	8	H'FF5B	PORT	8	2
Port D register	PORTD	8	H'FF5C	PORT	8	2
Port E register	PORTE	8	H'FF5D	PORT	8	2
Port F register	PORTF	8	H'FF5E	PORT	8	2
Port G register	PORTG	8	H'FF5F	PORT	8	2
Port 1 data register	P1DR	8	H'FF60	PORT	8	2
Port 2 data register	P2DR	8	H'FF61	PORT	8	2
Port 3 data register	P3DR	8	H'FF62	PORT	8	2
Port 5 data register	P5DR	8	H'FF64	PORT	8	2
Port 6 data register	P6DR	8	H'FF65	PORT	8	2
Port 8 data register	P8DR	8	H'FF67	PORT	8	2
Port A data register	PADR	8	H'FF69	PORT	8	2
Port B data register	PBDR	8	H'FF6A	PORT	8	2
Port C data register	PCDR	8	H'FF6B	PORT	8	2
Port D data register	PDDR	8	H'FF6C	PORT	8	2
Port E data register	PEDR	8	H'FF6D	PORT	8	2
Port F data register	PFDR	8	H'FF6E	PORT	8	2
Port G data register	PGDR	8	H'FF6F	PORT	8	2
Port H register	PORTH	8	H'FF70	PORT	8	2
Port J register	PORTJ	8	H'FF71	PORT	8	2
Port H data register	PHDR	8	H'FF72	PORT	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Port J data register	PJDR	8	H'FF73	PORT	8	2
Port H data direction register	PHDDR	8	H'FF74	PORT	8	2
Port J data direction register	PJDDR	8	H'FF75	PORT	8	2
Serial mode register_0	SMR_0	8	H'FF78	SCI_0	8	2
Bit rate register_0	BRR_0	8	H'FF79	SCI_0	8	2
Serial control register_0	SCR_0	8	H'FF7A	SCI_0	8	2
Transmit data register_0	TDR_0	8	H'FF7B	SCI_0	8	2
Serial status register_0	SSR_0	8	H'FF7C	SCI_0	8	2
Receive data register_0	RDR_0	8	H'FF7D	SCI_0	8	2
Smart card mode register_0	SCMR_0	8	H'FF7E	SCI_0	8	2
Serial mode register_1	SMR_1	8	H'FF80	SCI_1	8	2
Bit rate register_1	BRR_1	8	H'FF81	SCI_1	8	2
Serial control register_1	SCR_1	8	H'FF82	SCI_1	8	2
Transmit data register_1	TDR_1	8	H'FF83	SCI_1	8	2
Serial status register_1	SSR_1	8	H'FF84	SCI_1	8	2
Receive data register_1	RDR_1	8	H'FF85	SCI_1	8	2
Smart card mode register_1	SCMR_1	8	H'FF86	SCI_1	8	2
Serial mode register_2	SMR_2	8	H'FF88	SCI_2	8	2
Bit rate register_2	BRR_2	8	H'FF89	SCI_2	8	2
Serial control register_2	SCR_2	8	H'FF8A	SCI_2	8	2
Transmit data register_2	TDR_2	8	H'FF8B	SCI_2	8	2
Serial status register_2	SSR_2	8	H'FF8C	SCI_2	8	2
Receive data register_2	RDR_2	8	H'FF8D	SCI_2	8	2
Smart card mode register_2	SCMR_2	8	H'FF8E	SCI_2	8	2
A/D data register A	ADDRA	16	H'FF90	A/D_0	16	2
A/D data register B	ADDRB	16	H'FF92	A/D_0	16	2
A/D data register C	ADDRC	16	H'FF94	A/D_0	16	2
A/D data register D	ADDRD	16	H'FF96	A/D_0	16	2
A/D data register E	ADDRE	16	H'FF98	A/D_0	16	2
A/D data register F	ADDRF	16	H'FF9A	A/D_0	16	2
A/D data register G	ADDRG	16	H'FF9C	A/D_0	16	2
A/D data register H	ADDRH	16	H'FF9E	A/D_0	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
A/D control/status register	ADCSR	8	H'FFA0	A/D_0	16	2
A/D control register	ADCR	8	H'FFA1	A/D_0	16	2
D/A data register 2	DADR2	8	H'FFA8	D/A	8	2
D/A data register 3	DADR3	8	H'FFA9	D/A	8	2
D/A control register 23	DACR23	8	H'FFAA	D/A	8	2
Timer control register_0	TCR_0	8	H'FFB0	TMR_0	16	2
Timer control register_1	TCR_1	8	H'FFB1	TMR_1	16	2
Timer control/status register_0	TCSR_0	8	H'FFB2	TMR_0	16	2
Timer control/status register_1	TCSR_1	8	H'FFB3	TMR_1	16	2
Time constant register A_0	TCORA_0	8	H'FFB4	TMR_0	16	2
Time constant register A_1	TCORA_1	8	H'FFB5	TMR_1	16	2
Time constant register B_0	TCORB_0	8	H'FFB6	TMR_0	16	2
Time constant register B_1	TCORB_1	8	H'FFB7	TMR_1	16	2
Timer counter_0	TCNT_0	8	H'FFB8	TMR_0	16	2
Timer counter_1	TCNT_1	8	H'FFB9	TMR_1	16	2
Timer counter control register_0	TCCR_0	8	H'FFBA	TMR	16	2
Timer counter control register_1	TCCR_1	8	H'FFBB	TMR	16	2
Timer control/status register	TCSR	8	H'FFBC*[2] (Write) H'FFBC (Read)	WDT	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Timer counter	TCNT	8	H'FFBC*[2] (Write) H'FFBD (Read)	WDT	16	2
Reset control/status register	RSTCSR	8	H'FFBE*[2] (Write) H'FFBF (Read)	WDT	16	2
Timer start register	TSTR	8	H'FFC0	TPU	16	2
Timer synchronous register	TSYR	8	H'FFC1	TPU	16	2
Port function control register 3	PFCR3	8	H'FFC8	PORT	8	2
Port function control register 4	PFCR4	8	H'FFC9	PORT	8	2
Port function control register 5	PFCR5	8	H'FFCA	PORT	8	2
Timer control register_0	TCR_0	8	H'FFD0	TPU_0	16	2
Timer mode register_0	TMDR_0	8	H'FFD1	TPU_0	16	2
Timer I/O control register H_0	TIORH_0	8	H'FFD2	TPU_0	16	2
Timer I/O control register L_0	TIORL_0	8	H'FFD3	TPU_0	16	2
Timer interrupt enable register_0	TIER_0	8	H'FFD4	TPU_0	16	2
Timer status register_0	TSR_0	8	H'FFD5	TPU_0	16	2
Timer counter_0	TCNT_0	16	H'FFD6	TPU_0	16	2
Timer general register A_0	TGRA_0	16	H'FFD8	TPU_0	16	2
Timer general register B_0	TGRB_0	16	H'FFDA	TPU_0	16	2
Timer general register C_0	TGRC_0	16	H'FFDC	TPU_0	16	2
Timer general register D_0	TGRD_0	16	H'FFDE	TPU_0	16	2
Timer control register_1	TCR_1	8	H'FFE0	TPU_1	16	2
Timer mode register_1	TMDR_1	8	H'FFE1	TPU_1	16	2
Timer I/O control register_1	TIOR_1	8	H'FFE2	TPU_1	16	2
Timer interrupt enable register_1	TIER_1	8	H'FFE4	TPU_1	16	2
Timer status register_1	TSR_1	8	H'FFE5	TPU_1	16	2
Timer counter_1	TCNT_1	16	H'FFE6	TPU_1	16	2
Timer general register A_1	TGRA_1	16	H'FFE8	TPU_1	16	2
Timer general register B_1	TGRB_1	16	H'FFEA	TPU_1	16	2
Timer control register_2	TCR_2	8	H'FFF0	TPU_2	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Timer mode register_2	TMDR_2	8	H'FFF1	TPU_2	16	2
Timer I/O control register_2	TIOR_2	8	H'FFF2	TPU_2	16	2
Timer interrupt enable register_2	TIER_2	8	H'FFF4	TPU_2	16	2
Timer status register_2	TSR_2	8	H'FFF5	TPU_2	16	2
Timer counter_2	TCNT_2	16	H'FFF6	TPU_2	16	2
Timer general register A_2	TGRA_2	16	H'FFF8	TPU_2	16	2
Timer general register B_2	TGRB_2	16	H'FFFA	TPU_2	16	2

Notes: 1. If the pulse output group 2 and pulse output group 3 output triggers are the same according to the PCR setting, the NDRH address will be H'FF4C, and if different, the address of NDRH for group 2 will be H'FF4E, and that for group 3 will be H'FF4C. Similarly, if the pulse output group 0 and pulse output group 1 output triggers are the same according to the PCR setting, the NDRL address will be H'FF4D, and if different, the address of NDRL for group 0 will be H'FF4F, and that for group 1 will be H'FF4D.

2. For writing, see section 14.6.1, Notes on Register Access.
3. Not supported by the H8S/2454 Group.

26.2 Register Bits

Register addresses and bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, and 16-bit and 32-bit registers are shown as 2 or 4 lines, respectively.

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MRA	SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz	DTC*[1]
SAR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
MRB	CHNE	DISEL	CHNS	—	—	—	—	—	
DAR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CRA	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CRB	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
IFR0	BRST	CFDN	SURSS	SURSF	SETC	SET1	VBUSMN	VBUSF	USB
IFR1	—	—	—	SOF	SETUPTS	EPOoTS	EPOiTR	EPOiTS	
IFR2	—	—	EP3TR	EP3TS	EP2TR	EP2EMPTY	EP2ALLEMP	EP1FULL	
IER0	BRSTE	CFDNE	SSRSME	SURSFE	SETCE	SETIE	—	VBUSFE	
IER1	—	—	—	SOFE	SETUPTSE	EPOoTSE	EPOiTRE	EPOiTSE	
IER2	—	—	EP3TRE	EP3TSE	EP2TRE	EP2EMPTYE	EP2ALLEMPE	EP1FULLE	
ISR0	BRSTS	CFDNS	—	SURSFS	SETCS	SETIS	—	VBUSFS	
ISR1	—	—	—	SOFS	SETUPTSS	EPOoTSS	EPOiTRS	EPOiTSS	
ISR2	—	—	EP3TRS	EP3TSS	EP2TRS	EP2EMPTYS	EP2ALLEMPS	EP1FULLS	
EPDR0i	D7	D6	D5	D4	D3	D2	D1	D0	
EPDR0o	D7	D6	D5	D4	D3	D2	D1	D0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
EPDR0s	D7	D6	D5	D4	D3	D2	D1	D0	USB
EPDR1	D7	D6	D5	D4	D3	D2	D1	D0	
EPDR2	D7	D6	D5	D4	D3	D2	D1	D0	
EPDR3	D7	D6	D5	D4	D3	D2	D1	D0	
EPSZ0o	—	—	—	D4	D3	D2	D1	D0	
EPSZ1	—	D6	D5	D4	D3	D2	D1	D0	
DASTS0	—	—	—	—	—	—	—	EP0iDE	
DASTS1	—	—	—	—	—	EP3DE	EP2DE	—	
TRG0	—	—	—	—	—	EP0sRDFN	EP0oRDFN	EP0iPKTE	
TRG1	—	—	—	—	—	EP3PKTE	EP2PKTE	EP1RDFN	
FCLR0	—	—	—	—	—	—	EP0oCLR	EP0iCLR	
FCLR1	—	—	—	—	—	EP3CLR	EP2CLR	EP1CLR	
EPSTL0	—	—	—	EP0STLC	—	—	—	EP0STLS	
EPSTL1	—	EP3STLC	EP2STLC	EP1STLC	—	EP3STLS	EP2STLS	EP1STLS	
STLSR1	—	EP3ASCE	EP2ASCE	EP1ASCE	—	EP3STLST	EP2STLST	EP1STLST	
DMAR	—	—	—	—	—	—	EP1DMAE	EP2DMAE	
CVR	CNFV1	CNFV0	INTV1	INTV0	—	ALTV2	ALTV1	ALTV0	
CTLR	PULLUPE	—	—	RWUPS	RSME	PWMD	EP0ASCE	PRTRST	
EPIR	D7	D6	D5	D4	D3	D2	D1	D0	
	D7	D6	D5	D4	D3	D2	D1	D0	
	D7	D6	D5	D4	D3	D2	D1	D0	
	D7	D6	D5	D4	D3	D2	D1	D0	
TRNTREG0	PTSTE	—	—	—	SUSPEND	txenl	txse0	txdata	
TRNTREG1	—	—	—	—	—	xver_data	dpls	dmns	
RMMSTPCRH	MSTP47	MSTP46	MSTP45	MSTP44	MSTP43	MSTP42	MSTP41	MSTP40	SYSTEM
RMMSTPCRL	MSTP39	MSTP38	MSTP37	MSTP36	MSTP35	MSTP34	MSTP33	MSTP32	
USPLLCR	—	—	—	—	—	—	USSTC1	USSTC0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
IPRL	—	IPR14	IPR13	IPR12	—	IPR10	IPR9	IPR8	INTC
	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRM	—	IPR14	IPR13	IPR12	—	IPR10	IPR9	IPR8	
	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRN	—	IPR14	IPR13	IPR12	—	IPR10	IPR9	IPR8	
	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
DTCERL	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0	
DTCCR	SWDTE	—	—	—	—	—	—	—	
ADDRA_1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D_1
	AD1	AD0	—	—	—	—	—	—	
ADDRB_1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRC_1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRD_1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRE_1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRF_1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRG_1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRH_1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR_1	ADF	ADIE	ADST	EXCKS	CH3	CH2	CH1	CH0	
ADCR_1	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	EXTRGS	
TSTR_1	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU
TSYR_1	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	
TCR_6	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_6
TMDR_6	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_6	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TIORL_6	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	TPU_6
TIER_6	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_6	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_6	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_6	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_6	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRC_6	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRD_6	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_7	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_7
TMDR_7	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_7	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_7	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_7	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_7	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_7	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_7	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_8
TMDR_8	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_8	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_8	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_8	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_8	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TGRA_8	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TPU_8
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_8	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_9	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_9
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	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_9	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_9	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_9	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_9	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRC_9	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRD_9	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_10	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_10
TMDR_10	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_10	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_10	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_10	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_10	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_10	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_10	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCR_11	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_11
TMDR_11	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_11	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_11	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_11	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_11	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_11	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_11	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
P1ODR	P17ODR	P16ODR	P15ODR	P14ODR	P13ODR	P12ODR	P11ODR	P10ODR	PORT
P2ODR	P27ODR	P26ODR	P25ODR	—	—	—	—	P20ODR	
P5ODR	—	—	—	—	P53ODR	P52ODR	P51ODR	P500DR	
P6ODR	—	—	P65ODR	P64ODR	P63ODR	P62ODR	P61ODR	P60ODR	
P8ODR	—	—	P85ODR	P84ODR	P83ODR	P82ODR	P81ODR	P80ODR	
PBODR	PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR	
PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR	
PDODR	PD7ODR	PD6ODR	PD5ODR	PD4ODR	PD3ODR	PD2ODR	PD1ODR	PD0ODR	
PEODR	PE7ODR	PE6ODR	PE5ODR	PE4ODR	PE3ODR	PE2ODR	PE1ODR	PE0ODR	
PFODR	PF7ODR	PF6ODR	PF5ODR	PF4ODR	PF3ODR	PF2ODR	PF1ODR	PF0ODR	
PGODR	—	PG6ODR	PG5ODR	PG4ODR	PG3ODR	PG2ODR	PG1ODR	PG0ODR	
PHODR	—	—	—	—	PH3ODR	PH2ODR	PH1ODR	PH0ODR	
PJODR	—	—	—	—	—	—	PJ1ODR	PJ0ODR	
ICCRA_0	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC2_0
ICCRB_0	BBSY	SCP	SDAO	SDAOP	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	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_0	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ICCRA_1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC2_1
ICCRB_1	BBSY	SCP	SDAO	SDAOP	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	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_1	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	
ICCRA_2	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC2_2
ICCRB_2	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—	
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	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_2	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	
ICCRA_3	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC2_3
ICCRB_3	BBSY	SCP	SDAO	SDAOP	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	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_3	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	
SEMR_2	SSE	—	—	—	ABCS	ACS2	ACS1	ACS0	SCI_2
SSCRH	MSS	BIDE	—	SOL	SOLP	SCKS	CSS1	CSS0	SSU
SSCRL	—	SSUMS	SRES	—	—	—	DATS1	DATS0	
SSMR	MLS	CPOS	CPHS	—	—	CKS2	CKS1	CKS0	
SSER	TE	RE	—	—	TEIE	TIE	RIE	CEIE	
SSSR	—	ORER	—	—	TEND	TDRE	RDRF	CE	
SSCR2	SDOS	SSCKOS	SCSOS	TENDSTS	SCSATS	SSODTS	—	—	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SSTDR0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SSU
SSTDR1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSTDR2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSTDR3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSRDR0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSRDR1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSRDR2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSRDR3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
EDSAR_2	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	EXDMAC_2*[7]
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
EDDAR_2	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
EDTCR_2	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
EDMDR_2	EDA	BEF	EDRAKE	ETENDE	EDREQS	AMS	MDS1	MDS0	
	EDIE	IRF	TCEIE	SDIR	DTSIZE	BGUP	—	—	
EDACR_2	SAT1	SAT0	SARIE	SARA4	SARA3	SARA2	SARA1	SARA0	
	DAT1	DAT0	DARIE	DARA4	DARA3	DARA2	DARA1	DARA0	
EDSAR_3	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	EXDMAC_3*[7]
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
EDDAR_3	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	EXDMAC_3*[7]
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	
EDTCR_3	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
EDMDR_3	EDA	BEF	EDRAKE	ETENDE	EDREQS	AMS	MDS1	MDS0	
	EDIE	IRF	TCEIE	SDIR	DTSIZE	BGUP	—	—	
EDACR_3	SAT1	SAT0	SARIE	SARA4	SARA3	SARA2	SARA1	SARA0	
	DAT1	DAT0	DARIE	DARA4	DARA3	DARA2	DARA1	DARA0	
IPRA	—	IPRA14	IPRA13	IPRA12	—	IPRA10	IPRA9	IPRA8	INT
	—	IPRA6	IPRA5	IPRA4	—	IPRA2	IPRA1	IPRA0	
IPRB	—	IPRB14	IPRB13	IPRB12	—	IPRB10	IPRB9	IPRB8	
	—	IPRB6	IPRB5	IPRB4	—	IPRB2	IPRB1	IPRB0	
IPRC	—	IPRC14	IPRC13	IPRC12	—	IPRC10	IPRC9	IPRC8	
	—	IPRC6	IPRC5	IPRC4	—	IPRC2	IPRC1	IPRC0	
IPRD	—	IPRD14	IPRD13	IPRD12	—	IPRD10	IPRD9	IPRD8	
	—	IPRD6	IPRD5	IPRD4	—	IPRD2	IPRD1	IPRD0	
IPRE	—	IPRE14	IPRE13	IPRE12	—	IPRE10	IPRE9	IPRE8	
	—	IPRE6	IPRE5	IPRE4	—	IPRE2	IPRE1	IPRE0	
IPRF	—	IPRF14	IPRF13	IPRF12	—	IPRF10	IPRF9	IPRF8	
	—	IPRF6	IPRF5	IPRF4	—	IPRF2	IPRF1	IPRF0	
IPRG	—	IPRG14	IPRG13	IPRG12	—	IPRG10	IPRG9	IPRG8	
	—	IPRG6	IPRG5	IPRG4	—	IPRG2	IPRG1	IPRG0	
IPRH	—	IPRH14	IPRH13	IPRH12	—	IPRH10	IPRH9	IPRH8	
	—	IPRH6	IPRH5	IPRH4	—	IPRH2	IPRH1	IPRH0	
IPRI	—	IPRI14	IPRI13	IPRI12	—	IPRI10	IPRI9	IPRI8	
	—	IPRI6	IPRI5	IPRI4	—	IPRI2	IPRI1	IPRI0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
IPRJ	—	IPRJ14	IPRJ13	IPRJ12	—	IPRJ10	IPRJ9	IPRJ8	INT
	—	IPRJ6	IPRJ5	IPRJ4	—	IPRJ2	IPRJ1	IPRJ0	
IPRK	—	IPRK14	IPRK13	IPRK12	—	IPRK10	IPRK9	IPRK8	
	—	IPRK6	IPRK5	IPRK4	—	IPRK2	IPRK1	IPRK0	
ITSR	ITS15	ITS14	ITS13	—	—	—	—	ITS8	
	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	ITS0	
SSIER	SSI15	SSI14	SSI13	SSI12	SSI11	SSI10	SSI9	SSI8	
	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	SSI0	
ISCRH	IRQ15SCB	IRQ15SCA	IRQ14SCB	IRQ14SCA	IRQ13SCB	IRQ13SCA	IRQ12SCB	IRQ12SCA	
	IRQ11SCB	IRQ11SCA	IRQ10SCB	IRQ10SCA	IRQ9SCB	IRQ9SCA	IRQ8SCB	IRQ8SCA	
ISCRL	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	
	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA	
IrCR_0	IrE	IrCKS2	IrCKS1	IrCKS0	IrTxINV	IrRxINV	—	—	IrDA
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	PORT
P2DDR	P27DDR	P26DDR	P25DDR	—	—	—	—	P20DDR	
P3DDR	—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR	
P5DDR	—	—	—	—	P53DDR	P52DDR	P51DDR	P50DDR	
P6DDR	—	—	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR	
P8DDR	—	—	P85DDR	P84DDR	P83DDR	P82DDR	P81DDR	P80DDR	
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR	
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR	
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR	
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR	
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR	
PGDDR	—	PG6DDR	PG5DDR	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR	
PFCR0	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E	
PFCR1	A23E	A22E	A21E	A20E	A19E	A18E	A17E	A16E	
PFCR2	—	—	—	—	ASOE	LWROE	OES	—	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PAPCR	PA7PCR	PA6PCR	PA5PCR	PA4PCR	PA3PCR	PA2PCR	PA1PCR	PA0PCR	PORT
PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR	
PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR	
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR	
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR	
P3ODR	—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR	
PAODR	PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	PA0ODR	
SMR_3*[4]	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI_3, Smartcard interface_3
SMR_3*[5]	GM	BLK	PE	O/E	BCP1	BCP0	CKS1	CKS0	
BRR_3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCR_3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSR_3*[4]	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SSR_3*[5]	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	
RDR_3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_3	BCP2	—	—	—	SDIR	SINV	—	SMIF	
SMR_4*[4]	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI_4, Smartcard interface_4
SMR_4*[5]	GM	BLK	PE	O/E	BCP1	BCP0	CKS1	CKS0	
BRR_4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCR_4	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSR_4*[4]	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SSR_4*[5]	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	
RDR_4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_4	BCP2	—	—	—	SDIR	SINV	—	SMIF	
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_3
TMDR_3	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_3	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_3	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCNT_3	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TPU_3
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_3	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_3	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRC_3	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRD_3	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_4
TMDR_4	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_4	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_4	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_4	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_5
TMDR_5	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_5	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_5	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_5	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
FLMCR1	—	CBIDB	—	—	—	—	—	FMCMDEN	FLASH
DFPR	—	—	—	—	—	—	—	FMDBPT0	
FLMSTR	—	—	FMERSF	FMERCF	FMPRSF	FMPRCF	—	FMRDY	
ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	BSC
ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	
WTCRAH	—	W72	W71	W70	—	W62	W61	W60	
WTCRAL	—	W52	W51	W50	—	W42	W41	W40	
WTCRBH	—	W32	W31	W30	—	W22	W21	W20	
WTCRBL	—	W12	W11	W10	—	W02	W01	W00	
RDNCR	RDN7	RDN6	RDN5	RDN4	RDN3	RDN2	RDN1	RDN0	
CSACRH	CSXH7	CSXH6	CSXH5	CSXH4	CSXH3	CSXH2	CSXH1	CSXH0	
CSACRL	CSXT7	CSXT6	CSXT5	CSXT4	CSXT3	CSXT2	CSXT1	CSXT0	
BROMCRH	BSRM0	BSTS02	BSTS01	BSTS00	—	—	BSWD01	BSWD00	
BROMCRL	BSRM1	BSTS12	BSTS11	BSTS10	—	—	BSWD11	BSWD10	
BCR	BRLE	BREQOE	—	IDLC	ICIS1	ICIS0	WDBE	WAITE	
	—	—	—	—	—	ICIS2	—	—	
MPXCR	MPXE	—	—	—	—	—	—	ADDEX	
DRAMCR	OEE	RAST	—	CAST	—	RMTS2	RMTS1	RMTS0	
	BE	RCDM	DDS	EDDS	—	MXC2	MXC1	MXC0	
DRACCR	DRMI	—	TPC1	TPC0	SDWCD	—	RCD1	RCD0	
	—	—	—	—	CKSPE	—	RDXC1	RDXC0	
REFCR	CMF	CMIE	RCW1	RCW0	—	RTCK2	RTCK1	RTCK0	
	RFSHE	CBRM	RLW1	RLW0	SLFRF	TPCS2	TPCS1	TPCS0	
RTCNT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
RTCOR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MAR_0AH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	DMAC
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MAR_0AL	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
IOAR_0A	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ETCR_0A	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MAR_0BH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MAR_0BL	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
IOAR_0B	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ETCR_0B	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MAR_1AH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MAR_1AL	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
IOAR_1A	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ETCR_1A	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MAR_1BH	—	—	—	—	—	—	—	—	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MAR_1BL	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
IOARV1B	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ETCR_1B	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
DMAWER	—	—	—	—	WE1B	WE1A	WE0B	WE0A	DMAC
DMATCR	—	—	TEE1	TEE0	—	—	—	—	
DMACR_0A*[2]	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR_0A*[3]	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—	
DMACR_0B*[2]	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR_0B*[3]	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	
DMACR_1A*[2]	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR_1A*[3]	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—	
DMACR_1B*[2]	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR_1B*[3]	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	
DMABCRH*[2]	FAE1	FAE0	SAE1	SAE0	DTA1B	DTA1A	DTA0B	DTA0A	
DMABCRH*[3]	FAE1	FAE0	—	—	DTA1	—	DTA0	—	
DMABCRL*[2]	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A	
DMABCRL*[3]	DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B	DTIE0A	
DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC
DTCERB	DTCEB7	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0	
DTCERC	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0	
DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0	
DTCERE	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0	
DTCERF	DTCEF7	DTCEF6	DTCEF5	DTCEF4	DTCEF3	DTCEF2	DTCEF1	DTCEF0	
DTCERG	DTCEG7	DTCEG6	DTCEG5	DTCEG4	DTCEG3	DTCEG2	DTCEG1	DTCEG0	
DTCERH	DTCEH7	DTCEH6	DTCEH5	DTCEH4	DTCEH3	DTCEH2	DTCEH1	DTCEH0	
DTCERI	DTCEI7	DTCEI6	DTCEI5	DTCEI4	DTCEI3	DTCEI2	DTCEI1	DTCEI0	
DTVECR	DTVEC7	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
INTCR	—	—	INTM1	INTM0	NMIEG	—	—	—	INT
IER	IRQ15E	IRQ14E	IRQ13E	IRQ12E	IRQ11E	IRQ10E	IRQ9E	IRQ8E	
	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
ISR	IRQ15F	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	IRQ8F	
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SBYCR	SSBY	OPE	—	—	STS3	STS2	STS1	STS0	SYSTEM
SCKCR	PSTOP	—	SDPSTP	—	STCS	—	—	—	
SYSCR	—	—	MACS	—	FLSHE	—	EXPE	RAME	
MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	
MSTPCRH	ACSE	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	
MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0	
EXMSTPCRH	MSTP31	MSTP30	MSTP29	MSTP28	MSTP27	MSTP26	MSTP25	MSTP24	
EXMSTPCRL	MSTP23	MSTP22	MSTP21	MSTP20	MSTP19	MSTP18	MSTP17	MSTP16	
PLLCR	—	—	—	—	—	—	STC1	STC0	
PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	PPG
PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV	
NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8	
NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0	
PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8	
PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0	
NDRHH*6	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8	
NDRLH*6	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0	
NDRHL*6	NDRL15	NDRL14	NDRL13	NDRL12	NDR11	NDR10	NDR9	NDR8	
NDRLL*6	—	—	—	—	—	—	—	—	
PORT1	P17	P16	P15	P14	P13	P12	P11	P10	PORT
PORT2	P27	P26	P25	—	—	—	—	P20	
PORT3	—	—	P35	P34	P33	P32	P31	P30	
PORT4	P47	P46	P45	P44	P43	P42	P41	P40	
PORT5	—	—	—	—	P53	P52	P51	P50	
PORT6	—	—	P65	P64	P63	P62	P61	P60	
PORT8	—	—	P85	P84	P83	P82	P81	P80	
PORT9	P97	P96	P95	P94	P93	P92	P91	P90	
PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	
PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	PORT
PORTG	—	PG6	PG5	PG4	PG3	PG2	PG1	PG0	
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	
P2DR	P27DR	P26DR	P25DR	—	—	—	—	P20DR	
P3DR	—	—	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR	
P5DR	—	—	—	—	P53DR	P52DR	P51DR	P50DR	
P6DR	—	—	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR	
P8DR	—	—	P85DR	P84DR	P83DR	P82DR	P81DR	P80DR	
PADR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
PGDR	—	PG6DR	PG5DR	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR	
PORTH	—	—	—	—	PH3	PH2	PH1	PH0	
PORTJ	—	—	—	—	—	PJ2	PJ1	PJ0	
PHDR	—	—	—	—	PH3DR	PH2DR	PH1DR	PH0DR	
PJDR	—	—	—	—	—	—	PJ1DR	PJ0DR	
PHDDR	—	—	—	—	PH3DDR	PH2DDR	PH1DDR	PH0DDR	
PJDDR	—	—	—	—	—	—	PJ1DDR	PJ0DDR	
SMR_0*[4]	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI_0, Smart card interface_0
SMR_0*[5]	GM	BLK	PE	O/E	BCP1	BCP0	CKS1	CKS0	
BRR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSR_0*[4]	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SSR_0*[5]	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	
RDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_0	BCP2	—	—	—	SDIR	SINV	—	SMIF	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SMR_1*[4]	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI_1, Smart card interface_1
SMR_1*[5]	GM	BLK	PE	O/E	BCP1	BCP0	CKS1	CKS0	
BRR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSR_1*[4]	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SSR_1*[5]	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	
RDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_1	BCP2	—	—	—	SDIR	SINV	—	SMIF	
SMR_2*[4]	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI_2, Smart card interface_2
SMR_2*[5]	GM	BLK	PE	O/E	BCP1	BCP0	CKS1	CKS0	
BRR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSR_2*[4]	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SSR_2*[5]	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	
RDR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_2	BCP2	—	—	—	SDIR	SINV	—	SMIF	
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D_0
	AD1	AD0	—	—	—	—	—	—	
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRE	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRF	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRG	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ADDRH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D_0
	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	—	—	
DADR2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	D/A
DADR3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
DACR23	DAOE1	DAOE0	DAE	—	—	—	—	—	
TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_0
TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_1
TCSR_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	
TCSR_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
TCORA_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORA_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORB_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORB_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCNT_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCNT_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCCR0	—	—	—	—	TMRIS	—	ICKS1	ICKS0	8-bit
TCCR1	—	—	—	—	TMRIS	—	ICKS1	ICKS0	TMR
TCSR	OVF	WT/IT	TME	—	—	CKS2	CKS1	CKS0	WDT
TCNT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
RSTCSR	WOVF	RSTE	—	—	—	—	—	—	
TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU
TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	
PFCR3	—	PPGS	TPUS	TMRS	—	—	USBDRQE	—	PORT
PFCR4	WAITS	BREQS	BACKS	BREQOS	ADTRG0S	TXD4S	RXD4S	SCK4S	
PFCR5	SSO0S1	SSO0S0	SSI0S1	SSI0S0	SSCK0S1	SSCK0S0	SCS0S1	SCS0S0	
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_0
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	TPU_0
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRC_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRD_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_1
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_2
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TGRA_2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TPU_2
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Notes: 1. Loaded in on-chip RAM. The bus width is 32 bits when the DTC accesses this area as register information, and 16 bits otherwise.
2. For short address mode
3. For full address mode
4. For normal mode
5. For smart card interface mode
6. If the pulse output group 2 and pulse output group 3 output triggers are the same according to the PCR setting, the NDRH address will be H'FF4C, and if different, the address of NDRH for group 2 will be H'FF4E, and that for group 3 will be H'FF4C. Similarly, if the pulse output group 0 and pulse output group 1 output triggers are the same according to the PCR setting, the NDRL address will be H'FF4D, and if different, the address of NDRL for group 0 will be H'FF4F, and that for group 1 will be H'FF4D.
7. Not supported by the H8S/2454 Group.

26.3 Register States in Each Operating Mode

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
MRA	Initialized	—	—	—	—	—	—	Initialized	DTC
SAR	Initialized	—	—	—	—	—	—	Initialized	
MRB	Initialized	—	—	—	—	—	—	Initialized	
DAR	Initialized	—	—	—	—	—	—	Initialized	
CRA	Initialized	—	—	—	—	—	—	Initialized	
CRB	Initialized	—	—	—	—	—	—	Initialized	
IFR0	Initialized	—	—	—	—	—	—	Initialized	USB
IFR1	Initialized	—	—	—	—	—	—	Initialized	
IFR2	Initialized	—	—	—	—	—	—	Initialized	
IER0	Initialized	—	—	—	—	—	—	Initialized	
IER1	Initialized	—	—	—	—	—	—	Initialized	
IER2	Initialized	—	—	—	—	—	—	Initialized	
ISR0	Initialized	—	—	—	—	—	—	Initialized	
ISR1	Initialized	—	—	—	—	—	—	Initialized	
ISR2	Initialized	—	—	—	—	—	—	Initialized	
EPDR0i	Initialized	—	—	—	—	—	—	Initialized	
EPDR0o	Initialized	—	—	—	—	—	—	Initialized	
EPDR0s	Initialized	—	—	—	—	—	—	Initialized	
EPDR1	Initialized	—	—	—	—	—	—	Initialized	
EPDR2	Initialized	—	—	—	—	—	—	Initialized	
EPDR3	Initialized	—	—	—	—	—	—	Initialized	
EPSZ0o	Initialized	—	—	—	—	—	—	Initialized	
EPSZ1	Initialized	—	—	—	—	—	—	Initialized	
DASTS0	Initialized	—	—	—	—	—	—	Initialized	
DASTS1	Initialized	—	—	—	—	—	—	Initialized	
TRG0	Initialized	—	—	—	—	—	—	Initialized	
TRG1	Initialized	—	—	—	—	—	—	Initialized	
FCLR0	Initialized	—	—	—	—	—	—	Initialized	
FCLR1	Initialized	—	—	—	—	—	—	Initialized	
EPSTL0	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
EPSTL1	Initialized	—	—	—	—	—	—	Initialized	USB
STLSR1	Initialized	—	—	—	—	—	—	Initialized	
DMAR	Initialized	—	—	—	—	—	—	Initialized	
CVR	Initialized	—	—	—	—	—	—	Initialized	
CTLR	Initialized	—	—	—	—	—	—	Initialized	
EPIR	Initialized	—	—	—	—	—	—	Initialized	
TRNTREG0	Initialized	—	—	—	—	—	—	Initialized	
TRNTREG1	Initialized	—	—	—	—	—	—	Initialized	
RMMSTPCRH	Initialized	—	—	—	—	—	—	Initialized	SYSTEM
RMMSTPCRL	Initialized	—	—	—	—	—	—	Initialized	
USPLLCR	Initialized	—	—	—	—	—	—	Initialized	
IPRL	Initialized	—	—	—	—	—	—	Initialized	INT
IPRM	Initialized	—	—	—	—	—	—	Initialized	
IPRN	Initialized	—	—	—	—	—	—	Initialized	
DTCERI	Initialized	—	—	—	—	—	—	Initialized	DTC
DTCCR	Initialized	—	—	—	—	—	—	Initialized	
ADDRA_1	Initialized	—	—	—	—	—	—	Initialized	A/D_1
ADDRB_1	Initialized	—	—	—	—	—	—	Initialized	
ADDRC_1	Initialized	—	—	—	—	—	—	Initialized	
ADDRD_1	Initialized	—	—	—	—	—	—	Initialized	
ADDRE_1	Initialized	—	—	—	—	—	—	Initialized	
ADDRF_1	Initialized	—	—	—	—	—	—	Initialized	
ADDRG_1	Initialized	—	—	—	—	—	—	Initialized	
ADDRH_1	Initialized	—	—	—	—	—	—	Initialized	
ADCSR_1	Initialized	—	—	—	—	—	—	Initialized	
ADCR_1	Initialized	—	—	—	—	—	—	Initialized	
TSTRB	Initialized	—	—	—	—	—	—	Initialized	TPU
TSYRB	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
TCR_6	Initialized	—	—	—	—	—	—	Initialized	TPU_6
TMDR_6	Initialized	—	—	—	—	—	—	Initialized	
TIORH_6	Initialized	—	—	—	—	—	—	Initialized	
TIORL_6	Initialized	—	—	—	—	—	—	Initialized	
TIER_6	Initialized	—	—	—	—	—	—	Initialized	
TSR_6	Initialized	—	—	—	—	—	—	Initialized	
TCNT_6	Initialized	—	—	—	—	—	—	Initialized	
TGRA_6	Initialized	—	—	—	—	—	—	Initialized	
TGRB_6	Initialized	—	—	—	—	—	—	Initialized	
TGRC_6	Initialized	—	—	—	—	—	—	Initialized	
TGRD_6	Initialized	—	—	—	—	—	—	Initialized	
TCR_7	Initialized	—	—	—	—	—	—	Initialized	TPU_7
TMDR_7	Initialized	—	—	—	—	—	—	Initialized	
TIOR_7	Initialized	—	—	—	—	—	—	Initialized	
TIER_7	Initialized	—	—	—	—	—	—	Initialized	
TSR_7	Initialized	—	—	—	—	—	—	Initialized	
TCNT_7	Initialized	—	—	—	—	—	—	Initialized	
TGRA_7	Initialized	—	—	—	—	—	—	Initialized	
TGRB_7	Initialized	—	—	—	—	—	—	Initialized	
TCR_8	Initialized	—	—	—	—	—	—	Initialized	TPU_8
TMDR_8	Initialized	—	—	—	—	—	—	Initialized	
TIOR_8	Initialized	—	—	—	—	—	—	Initialized	
TIER_8	Initialized	—	—	—	—	—	—	Initialized	
TSR_8	Initialized	—	—	—	—	—	—	Initialized	
TCNT_8	Initialized	—	—	—	—	—	—	Initialized	
TGRA_8	Initialized	—	—	—	—	—	—	Initialized	
TGRB_8	Initialized	—	—	—	—	—	—	Initialized	
TCR_9	Initialized	—	—	—	—	—	—	Initialized	TPU_9
TMDR_9	Initialized	—	—	—	—	—	—	Initialized	
TIORH_9	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
TIORL_9	Initialized	—	—	—	—	—	—	Initialized	TPU_9
TIER_9	Initialized	—	—	—	—	—	—	Initialized	
TSR_9	Initialized	—	—	—	—	—	—	Initialized	
TCNT_9	Initialized	—	—	—	—	—	—	Initialized	
TGRA_9	Initialized	—	—	—	—	—	—	Initialized	
TGRB_9	Initialized	—	—	—	—	—	—	Initialized	
TGRC_9	Initialized	—	—	—	—	—	—	Initialized	
TGRD_9	Initialized	—	—	—	—	—	—	Initialized	
TCR_10	Initialized	—	—	—	—	—	—	Initialized	TPU_10
TMDR_10	Initialized	—	—	—	—	—	—	Initialized	
TIOR_10	Initialized	—	—	—	—	—	—	Initialized	
TIER_10	Initialized	—	—	—	—	—	—	Initialized	
TSR_10	Initialized	—	—	—	—	—	—	Initialized	
TCNT_10	Initialized	—	—	—	—	—	—	Initialized	
TGRA_10	Initialized	—	—	—	—	—	—	Initialized	
TGRB_10	Initialized	—	—	—	—	—	—	Initialized	
TCR_11	Initialized	—	—	—	—	—	—	Initialized	TPU_11
TMDR_11	Initialized	—	—	—	—	—	—	Initialized	
TIOR_11	Initialized	—	—	—	—	—	—	Initialized	
TIER_11	Initialized	—	—	—	—	—	—	Initialized	
TSR_11	Initialized	—	—	—	—	—	—	Initialized	
TCNT_11	Initialized	—	—	—	—	—	—	Initialized	
TGRA_11	Initialized	—	—	—	—	—	—	Initialized	
TGRB_11	Initialized	—	—	—	—	—	—	Initialized	
P1ODR	Initialized	—	—	—	—	—	—	Initialized	PORT
P2ODR	Initialized	—	—	—	—	—	—	Initialized	
P5ODR	Initialized	—	—	—	—	—	—	Initialized	
P6ODR	Initialized	—	—	—	—	—	—	Initialized	
P8ODR	Initialized	—	—	—	—	—	—	Initialized	
PBODR	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
PCODR	Initialized	—	—	—	—	—	—	Initialized	PORT
PDODR	Initialized	—	—	—	—	—	—	Initialized	
PEODR	Initialized	—	—	—	—	—	—	Initialized	
PFODR	Initialized	—	—	—	—	—	—	Initialized	
PGODR	Initialized	—	—	—	—	—	—	Initialized	
PHODR	Initialized	—	—	—	—	—	—	Initialized	
PJODR	Initialized	—	—	—	—	—	—	Initialized	
ICCRA_0	Initialized	—	—	—	—	—	—	Initialized	IIC2_0
ICCRB_0	Initialized	—	—	—	—	—	—	Initialized	
ICMR_0	Initialized	—	—	—	—	—	—	Initialized	
ICIER_0	Initialized	—	—	—	—	—	—	Initialized	
ICSR_0	Initialized	—	—	—	—	—	—	Initialized	
SAR_0	Initialized	—	—	—	—	—	—	Initialized	
ICDRT_0	Initialized	—	—	—	—	—	—	Initialized	
ICDRR_0	Initialized	—	—	—	—	—	—	Initialized	
ICCRA_1	Initialized	—	—	—	—	—	—	Initialized	IIC2_1
ICCRB_1	Initialized	—	—	—	—	—	—	Initialized	
ICMR_1	Initialized	—	—	—	—	—	—	Initialized	
ICIER_1	Initialized	—	—	—	—	—	—	Initialized	
ICSR_1	Initialized	—	—	—	—	—	—	Initialized	
SAR_1	Initialized	—	—	—	—	—	—	Initialized	
ICDRT_1	Initialized	—	—	—	—	—	—	Initialized	
ICDRR_1	Initialized	—	—	—	—	—	—	Initialized	
ICCRA_2	Initialized	—	—	—	—	—	—	Initialized	IIC2_2
ICCRB_2	Initialized	—	—	—	—	—	—	Initialized	
ICMR_2	Initialized	—	—	—	—	—	—	Initialized	
ICIER_2	Initialized	—	—	—	—	—	—	Initialized	
ICSR_2	Initialized	—	—	—	—	—	—	Initialized	
SAR_2	Initialized	—	—	—	—	—	—	Initialized	
ICDRT_2	Initialized	—	—	—	—	—	—	Initialized	
ICDRR_2	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
ICCRA_3	Initialized	—	—	—	—	—	—	Initialized	IIC2_3
ICCRB_3	Initialized	—	—	—	—	—	—	Initialized	
ICMR_3	Initialized	—	—	—	—	—	—	Initialized	
ICIER_3	Initialized	—	—	—	—	—	—	Initialized	
ICSR_3	Initialized	—	—	—	—	—	—	Initialized	
SAR_3	Initialized	—	—	—	—	—	—	Initialized	
ICDRT_3	Initialized	—	—	—	—	—	—	Initialized	
ICDRR_3	Initialized	—	—	—	—	—	—	Initialized	
SEMR_2	Initialized	—	—	—	—	—	—	Initialized	SCI_2
SSCRH	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	SSU
SSCRL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSMR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSER	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSSR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSCR2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSTDR0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSTDR1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSTDR2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSTDR3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSRDR0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSRDR1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSRDR2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSRDR3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
EDSAR_2	Initialized	—	—	—	—	—	—	Initialized	EXDMAC_2*
EDDAR_2	Initialized	—	—	—	—	—	—	Initialized	
EDTCR_2	Initialized	—	—	—	—	—	—	Initialized	
EDMDR_2	Initialized	—	—	—	—	—	—	Initialized	
EDACR_2	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
EDSAR_3	Initialized	—	—	—	—	—	—	Initialized	EXDMAC_3*
EDDAR_3	Initialized	—	—	—	—	—	—	Initialized	
EDTCR_3	Initialized	—	—	—	—	—	—	Initialized	
EDMDR_3	Initialized	—	—	—	—	—	—	Initialized	
EDACR_3	Initialized	—	—	—	—	—	—	Initialized	
IPRA	Initialized	—	—	—	—	—	—	Initialized	INT
IPRB	Initialized	—	—	—	—	—	—	Initialized	
IPRC	Initialized	—	—	—	—	—	—	Initialized	
IPRD	Initialized	—	—	—	—	—	—	Initialized	
IPRE	Initialized	—	—	—	—	—	—	Initialized	
IPRF	Initialized	—	—	—	—	—	—	Initialized	
IPRG	Initialized	—	—	—	—	—	—	Initialized	
IPRH	Initialized	—	—	—	—	—	—	Initialized	
IPRI	Initialized	—	—	—	—	—	—	Initialized	
IPRJ	Initialized	—	—	—	—	—	—	Initialized	
IPRK	Initialized	—	—	—	—	—	—	Initialized	
ITSR	Initialized	—	—	—	—	—	—	Initialized	
SSIER	Initialized	—	—	—	—	—	—	Initialized	
ISCRH	Initialized	—	—	—	—	—	—	Initialized	
ISCRL	Initialized	—	—	—	—	—	—	Initialized	
IrCR_0	Initialized	—	—	—	—	—	—	Initialized	IrDA
P1DDR	Initialized	—	—	—	—	—	—	Initialized	PORT
P2DDR	Initialized	—	—	—	—	—	—	Initialized	
P3DDR	Initialized	—	—	—	—	—	—	Initialized	
P5DDR	Initialized	—	—	—	—	—	—	Initialized	
P6DDR	Initialized	—	—	—	—	—	—	Initialized	
P8DDR	Initialized	—	—	—	—	—	—	Initialized	
PADDR	Initialized	—	—	—	—	—	—	Initialized	
PBDDR	Initialized	—	—	—	—	—	—	Initialized	
PCDDR	Initialized	—	—	—	—	—	—	Initialized	
PDDDR	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
PEDDR	Initialized	—	—	—	—	—	—	Initialized	PORT
PFDDR	Initialized	—	—	—	—	—	—	Initialized	
PGDDR	Initialized	—	—	—	—	—	—	Initialized	
PFCR0	Initialized	—	—	—	—	—	—	Initialized	
PFCR1	Initialized	—	—	—	—	—	—	Initialized	
PFCR2	Initialized	—	—	—	—	—	—	Initialized	
PAPCR	Initialized	—	—	—	—	—	—	Initialized	
PBPCR	Initialized	—	—	—	—	—	—	Initialized	
PCPCR	Initialized	—	—	—	—	—	—	Initialized	
PDPCR	Initialized	—	—	—	—	—	—	Initialized	
PEPCR	Initialized	—	—	—	—	—	—	Initialized	
P3ODR	Initialized	—	—	—	—	—	—	Initialized	
PAODR	Initialized	—	—	—	—	—	—	Initialized	
SMR_3	Initialized	—	—	—	—	—	—	Initialized	SCI_3
BRR_3	Initialized	—	—	—	—	—	—	Initialized	
SCR_3	Initialized	—	—	—	—	—	—	Initialized	
TDR_3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSR_3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RDR_3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCMR_3	Initialized	—	—	—	—	—	—	Initialized	
SMR_4	Initialized	—	—	—	—	—	—	Initialized	SCI_4
BRR_4	Initialized	—	—	—	—	—	—	Initialized	
SCR_4	Initialized	—	—	—	—	—	—	Initialized	
TDR_4	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSR_4	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RDR_4	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCMR_4	Initialized	—	—	—	—	—	—	Initialized	
TCR_3	Initialized	—	—	—	—	—	—	Initialized	TPU_3
TMDR_3	Initialized	—	—	—	—	—	—	Initialized	
TIORH_3	Initialized	—	—	—	—	—	—	Initialized	
TIORL_3	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
TIER_3	Initialized	—	—	—	—	—	—	Initialized	TPU_3
TSR_3	Initialized	—	—	—	—	—	—	Initialized	
TCNT_3	Initialized	—	—	—	—	—	—	Initialized	
TGRA_3	Initialized	—	—	—	—	—	—	Initialized	
TGRB_3	Initialized	—	—	—	—	—	—	Initialized	
TGRC_3	Initialized	—	—	—	—	—	—	Initialized	
TGRD_3	Initialized	—	—	—	—	—	—	Initialized	
TCR_4	Initialized	—	—	—	—	—	—	Initialized	TPU_4
TMDR_4	Initialized	—	—	—	—	—	—	Initialized	
TIOR_4	Initialized	—	—	—	—	—	—	Initialized	
TIER_4	Initialized	—	—	—	—	—	—	Initialized	
TSR_4	Initialized	—	—	—	—	—	—	Initialized	
TCNT_4	Initialized	—	—	—	—	—	—	Initialized	
TGRA_4	Initialized	—	—	—	—	—	—	Initialized	
TGRB_4	Initialized	—	—	—	—	—	—	Initialized	
TCR_5	Initialized	—	—	—	—	—	—	Initialized	TPU_5
TMDR_5	Initialized	—	—	—	—	—	—	Initialized	
TIOR_5	Initialized	—	—	—	—	—	—	Initialized	
TIER_5	Initialized	—	—	—	—	—	—	Initialized	
TSR_5	Initialized	—	—	—	—	—	—	Initialized	
TCNT_5	Initialized	—	—	—	—	—	—	Initialized	
TGRA_5	Initialized	—	—	—	—	—	—	Initialized	
TGRB_5	Initialized	—	—	—	—	—	—	Initialized	
FLMCR1	Initialized	—	—	—	—	—	—	Initialized	FLASH
DFPR	Initialized	—	—	—	—	—	—	Initialized	
FLMSTR	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
ABWCR	Initialized	—	—	—	—	—	—	Initialized	BSC
ASTCR	Initialized	—	—	—	—	—	—	Initialized	
WTCRAH	Initialized	—	—	—	—	—	—	Initialized	
WTCRAL	Initialized	—	—	—	—	—	—	Initialized	
WTCRBH	Initialized	—	—	—	—	—	—	Initialized	
WTCRBL	Initialized	—	—	—	—	—	—	Initialized	
RDNCR	Initialized	—	—	—	—	—	—	Initialized	
CSACRH	Initialized	—	—	—	—	—	—	Initialized	
CSACRL	Initialized	—	—	—	—	—	—	Initialized	
BROMCRH	Initialized	—	—	—	—	—	—	Initialized	
BROMCRL	Initialized	—	—	—	—	—	—	Initialized	
BCR	Initialized	—	—	—	—	—	—	Initialized	
MPXCR	Initialized	—	—	—	—	—	—	Initialized	
DRAMCR	Initialized	—	—	—	—	—	—	Initialized	
DRACCRH	Initialized	—	—	—	—	—	—	Initialized	
DRACCRL	Initialized	—	—	—	—	—	—	Initialized	
REFCR	Initialized	—	—	—	—	—	—	Initialized	
RTCNT	Initialized	—	—	—	—	—	—	Initialized	
RTCOR	Initialized	—	—	—	—	—	—	Initialized	
MAR_0AH	Initialized	—	—	—	—	—	—	Initialized	DMAC
MAR_0AL	Initialized	—	—	—	—	—	—	Initialized	
IOAR_0A	Initialized	—	—	—	—	—	—	Initialized	
ETCR_0A	Initialized	—	—	—	—	—	—	Initialized	
MAR_0BH	Initialized	—	—	—	—	—	—	Initialized	
MAR_0BL	Initialized	—	—	—	—	—	—	Initialized	
IOAR_0B	Initialized	—	—	—	—	—	—	Initialized	
ETCR_0B	Initialized	—	—	—	—	—	—	Initialized	
MAR_1AH	Initialized	—	—	—	—	—	—	Initialized	
MAR_1AL	Initialized	—	—	—	—	—	—	Initialized	
IOAR_1A	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
ETCR_1A	Initialized	—	—	—	—	—	—	Initialized	DMAC
MAR_1BH	Initialized	—	—	—	—	—	—	Initialized	
MAR_1BL	Initialized	—	—	—	—	—	—	Initialized	
IOAR_1B	Initialized	—	—	—	—	—	—	Initialized	
ETCR_1B	Initialized	—	—	—	—	—	—	Initialized	
DMAWER	Initialized	—	—	—	—	—	—	Initialized	
DMATCR	Initialized	—	—	—	—	—	—	Initialized	
DMACR_0A	Initialized	—	—	—	—	—	—	Initialized	
DMACR_0B	Initialized	—	—	—	—	—	—	Initialized	
DMACR_1A	Initialized	—	—	—	—	—	—	Initialized	
DMACR_1B	Initialized	—	—	—	—	—	—	Initialized	
DMABCRH	Initialized	—	—	—	—	—	—	Initialized	
DMABCRL	Initialized	—	—	—	—	—	—	Initialized	
DTCERA	Initialized	—	—	—	—	—	—	Initialized	DTC
DTCERB	Initialized	—	—	—	—	—	—	Initialized	
DTCERC	Initialized	—	—	—	—	—	—	Initialized	
DTCERD	Initialized	—	—	—	—	—	—	Initialized	
DTCERE	Initialized	—	—	—	—	—	—	Initialized	
DTCERF	Initialized	—	—	—	—	—	—	Initialized	
DTCERG	Initialized	—	—	—	—	—	—	Initialized	
DTCERH	Initialized	—	—	—	—	—	—	Initialized	
DTVECR	Initialized	—	—	—	—	—	—	Initialized	
INTCR	Initialized	—	—	—	—	—	—	Initialized	INT
IER	Initialized	—	—	—	—	—	—	Initialized	
ISR	Initialized	—	—	—	—	—	—	Initialized	
SBYCR	Initialized	—	—	—	—	—	—	Initialized	SYSTEM
SCKCR	Initialized	—	—	—	—	—	—	Initialized	
SYSCR	Initialized	—	—	—	—	—	—	Initialized	
MDCR	Initialized	—	—	—	—	—	—	Initialized	
MSTPCRH	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
MSTPCRL	Initialized	—	—	—	—	—	—	Initialized	SYSTEM
EXMSTPCRH	Initialized	—	—	—	—	—	—	Initialized	
EXMSTPCRL	Initialized	—	—	—	—	—	—	Initialized	
PLLCR	Initialized	—	—	—	—	—	—	Initialized	
PCR	Initialized	—	—	—	—	—	—	Initialized	PPG
PMR	Initialized	—	—	—	—	—	—	Initialized	
NDERH	Initialized	—	—	—	—	—	—	Initialized	
NDERL	Initialized	—	—	—	—	—	—	Initialized	
PODRH	Initialized	—	—	—	—	—	—	Initialized	
PODRL	Initialized	—	—	—	—	—	—	Initialized	
NDRHH	Initialized	—	—	—	—	—	—	Initialized	
NDRLH	Initialized	—	—	—	—	—	—	Initialized	
NDRHL	Initialized	—	—	—	—	—	—	Initialized	
NDRLL	Initialized	—	—	—	—	—	—	Initialized	
PORT1	—	—	—	—	—	—	—	—	PORT
PORT2	—	—	—	—	—	—	—	—	
PORT3	—	—	—	—	—	—	—	—	
PORT4	—	—	—	—	—	—	—	—	
PORT5	—	—	—	—	—	—	—	—	
PORT6	—	—	—	—	—	—	—	—	
PORT8	—	—	—	—	—	—	—	—	
PORT9	—	—	—	—	—	—	—	—	
PORTA	—	—	—	—	—	—	—	—	
PORTB	—	—	—	—	—	—	—	—	
PORTC	—	—	—	—	—	—	—	—	
PORTD	—	—	—	—	—	—	—	—	
PORTE	—	—	—	—	—	—	—	—	
PORTF	—	—	—	—	—	—	—	—	
PORTG	—	—	—	—	—	—	—	—	
P1DR	Initialized	—	—	—	—	—	—	Initialized	
P2DR	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
P3DR	Initialized	—	—	—	—	—	—	Initialized	PORT
P5DR	Initialized	—	—	—	—	—	—	Initialized	
P6DR	Initialized	—	—	—	—	—	—	Initialized	
P8DR	Initialized	—	—	—	—	—	—	Initialized	
PADR	Initialized	—	—	—	—	—	—	Initialized	
PBDR	Initialized	—	—	—	—	—	—	Initialized	
PCDR	Initialized	—	—	—	—	—	—	Initialized	
PDDR	Initialized	—	—	—	—	—	—	Initialized	
PEDR	Initialized	—	—	—	—	—	—	Initialized	
PFDR	Initialized	—	—	—	—	—	—	Initialized	
PGDR	Initialized	—	—	—	—	—	—	Initialized	
PORTH	Initialized	—	—	—	—	—	—	Initialized	
PHDR	Initialized	—	—	—	—	—	—	Initialized	
PJDR	Initialized	—	—	—	—	—	—	Initialized	
PHDDR	Initialized	—	—	—	—	—	—	Initialized	
PJDDR	Initialized	—	—	—	—	—	—	Initialized	
SMR_0	Initialized	—	—	—	—	—	—	Initialized	SCI_0
BRR_0	Initialized	—	—	—	—	—	—	Initialized	
SCR_0	Initialized	—	—	—	—	—	—	Initialized	
TDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCMR_0	Initialized	—	—	—	—	—	—	Initialized	
SMR_1	Initialized	—	—	—	—	—	—	Initialized	SCI_1
BRR_1	Initialized	—	—	—	—	—	—	Initialized	
SCR_1	Initialized	—	—	—	—	—	—	Initialized	
TDR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RDR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCMR_1	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
SMR_2	Initialized	—	—	—	—	—	—	Initialized	SCI_2
BRR_2	Initialized	—	—	—	—	—	—	Initialized	
SCR_2	Initialized	—	—	—	—	—	—	Initialized	
TDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCMR_2	Initialized	—	—	—	—	—	—	Initialized	
ADDRA	Initialized	—	—	—	—	—	—	Initialized	A/D_0
ADDRB	Initialized	—	—	—	—	—	—	Initialized	
ADDRC	Initialized	—	—	—	—	—	—	Initialized	
ADDRD	Initialized	—	—	—	—	—	—	Initialized	
ADDRE	Initialized	—	—	—	—	—	—	Initialized	
ADDRF	Initialized	—	—	—	—	—	—	Initialized	
ADDRG	Initialized	—	—	—	—	—	—	Initialized	
ADDRH	Initialized	—	—	—	—	—	—	Initialized	
ADCSR	Initialized	—	—	—	—	—	—	Initialized	
ADCR	Initialized	—	—	—	—	—	—	Initialized	
DADR2	Initialized	—	—	—	—	—	—	Initialized	D/A
DADR3	Initialized	—	—	—	—	—	—	Initialized	
DACR23	Initialized	—	—	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	—	—	Initialized	TMR_0
TCR_1	Initialized	—	—	—	—	—	—	Initialized	TMR_1
TCSR_0	Initialized	—	—	—	—	—	—	Initialized	
TCSR_1	Initialized	—	—	—	—	—	—	Initialized	
TCORA_0	Initialized	—	—	—	—	—	—	Initialized	
TCORA_1	Initialized	—	—	—	—	—	—	Initialized	
TCORB_0	Initialized	—	—	—	—	—	—	Initialized	
TCORB_1	Initialized	—	—	—	—	—	—	Initialized	
TCNT_0	Initialized	—	—	—	—	—	—	Initialized	
TCNT_1	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
TCCR_0	Initialized	—	—	—	—	—	—	Initialized	TMR
TCCR_1	Initialized	—	—	—	—	—	—	Initialized	
TCSR	Initialized	—	—	—	—	—	—	Initialized	WDT
TCNT	Initialized	—	—	—	—	—	—	Initialized	
RSTCSR	Initialized	—	—	—	—	—	—	—	
TSTR	Initialized	—	—	—	—	—	—	Initialized	TPU
TSYR	Initialized	—	—	—	—	—	—	Initialized	
PFCR3	Initialized	—	—	—	—	—	—	Initialized	PORT
PFCR4	Initialized	—	—	—	—	—	—	Initialized	
PFCR5	Initialized	—	—	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	—	—	Initialized	TPU_0
TMDR_0	Initialized	—	—	—	—	—	—	Initialized	
TIORH_0	Initialized	—	—	—	—	—	—	Initialized	
TIORL_0	Initialized	—	—	—	—	—	—	Initialized	
TIER_0	Initialized	—	—	—	—	—	—	Initialized	
TSR_0	Initialized	—	—	—	—	—	—	Initialized	
TCNT_0	Initialized	—	—	—	—	—	—	Initialized	
TGRA_0	Initialized	—	—	—	—	—	—	Initialized	
TGRB_0	Initialized	—	—	—	—	—	—	Initialized	
TGRC_0	Initialized	—	—	—	—	—	—	Initialized	
TGRD_0	Initialized	—	—	—	—	—	—	Initialized	
TCR_1	Initialized	—	—	—	—	—	—	Initialized	
TMDR_1	Initialized	—	—	—	—	—	—	Initialized	
TIOR_1	Initialized	—	—	—	—	—	—	Initialized	
TIER_1	Initialized	—	—	—	—	—	—	Initialized	
TSR_1	Initialized	—	—	—	—	—	—	Initialized	
TCNT_1	Initialized	—	—	—	—	—	—	Initialized	
TGRA_1	Initialized	—	—	—	—	—	—	Initialized	TPU_1
TGRB_1	Initialized	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
TCR_2	Initialized	—	—	—	—	—	—	Initialized	TPU_2
TMDR_2	Initialized	—	—	—	—	—	—	Initialized	
TIOR_2	Initialized	—	—	—	—	—	—	Initialized	
TIER_2	Initialized	—	—	—	—	—	—	Initialized	
TSR_2	Initialized	—	—	—	—	—	—	Initialized	
TCNT_2	Initialized	—	—	—	—	—	—	Initialized	
TGRA_2	Initialized	—	—	—	—	—	—	Initialized	
TGRB_2	Initialized	—	—	—	—	—	—	Initialized	

Note: * Not supported by the H8S/2454 Group.

Section 27 Electrical Characteristics

27.1 Electrical Characteristics for H8S/2456 Group and H8S/2456R Group

27.1.1 Absolute Maximum Ratings

Table 27.1 lists the absolute maximum ratings.

Table 27.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC} $PLLV_{CC}$ D_rV_{CC}	–0.3 to +4.3	V
Input voltage (except ports 4, 9, 2, P32 to P35, P50, P51, and PJ0 to PJ2)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (ports 2, P32 to P35, P50, P51, and PJ0 to PJ2)	V_{in}	–0.3 to +7.0	V
Input voltage (ports 4 and 9)	V_{in}	–0.3 to AV_{CC} +0.3	V
Reference power supply voltage	V_{ref}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +4.3	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75*	°C
		Wide-range specifications: –40 to +85*	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

Note: * Ranges of operating temperature when flash memory is programmed/erased:
Regular specifications: 0 to +75°C
Wide-range specifications: 0 to +85°C

27.1.2 DC Characteristics

Table 27.2 DC Characteristics (1)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, $V_{SS} = AV_{SS}$ = 0 V*1, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	Port 1, port 2, P32 to P35*2, P50 to P53*2, port 6*2, port 8*2, PA4 to PA7*2, port B*2, port C*2, PF1*2, PF2*2, PH2*2, PH3*2, PJ0*2, PJ1*2	VT^-	$V_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.07$	—	—	V	
Input high voltage	$\overline{STBY}$, MD2 to MD0	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} +0.3	V	
	$\overline{RES}$, NMI, FWE		$V_{CC} \times 0.9$	—	V_{CC} +0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	P14 to P17*5, P24 to P26*6, port 3, P50 to P53*3, ports 6 and 8*3, ports A to J*3		2.2	—	V_{CC} +0.3	V	
	Port 4, Port 9		2.2	—	AV_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD2 to MD0, EMLE	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL		–0.3	—	$V_{CC} \times 0.2$	V	
	Ports 3, 5, and 6, Port 8, ports A to J*3, P14 to P17*5, P24 to P24*6		–0.3	—	$V_{CC} \times 0.2$	V	
	Port 4, Port 9		–0.3	—	AV_{CC} +0.2	V	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.3	—	—	V	I_{OH} = –200 µA
			V_{CC} – 0.5	—	—	V	I_{OH} = –1 mA
			V_{CC} – 0.8	—	—	V	I_{OH} = –2 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 4.0 mA
	P26 to P27*4, P32 to P35*4, P50 to P51*4		—	—	0.4	V	I_{OL} = 8.0 mA
Input leakage current	$\overline{RES}$	$\|I_{in}\|$	—	—	10.0	µA	V_{in} = 0.5 to V_{CC} –0.5 V
	$\overline{STBY}$, NMI, MD2 to MD0		—	—	1.0	µA	
	Port 4, Port 9		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} –0.5 V

Notes: Port 2, P32 to P35, P50, P51, PJ0 to PJ2 are 5-V-tolerant pins.

1. When the A/D and D/A converters are not used, the AV_{CC}, V_{ref}, and AV_{SS} pins should not be open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.
2. When used as $\overline{IRQ}$, TIOC, TCLK, TMRI, SCL, or SDA.
3. When used as other than $\overline{IRQ}$, TIOC, TCLK, TMRI, SCL, or SDA.
4. When used as SCL or SDA.
5. When used as SSO, SSI, SSCK, or $\overline{SCS}$.
6. When used as RxD, $\overline{WAIT}$, or $\overline{ADTRG1}$.

Table 27.3 DC Characteristics (2)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
$V_{SS} = AV_{SS}$ = 0 V*[1], T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Three-state leakage current (off state)	Ports 1 to 3, P50 to P53, ports 6 and 8, ports A to I	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} –0.5 V
Input pull-up MOS current	Ports A to E	$-I_p$	10	—	300	µA	V_{CC} = 3.0 to 3.6 V V_{in} = 0 V
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V
	NMI		—	—	30	pF	f = 1 MHz
	All input pins except $\overline{RES}$ and NMI		—	—	18	pF	T_a = 25°C
Current consumption*[2]	Normal operation	I_{CC}*[4]	—	55 (3.3 V)	75	mA	f = 33 MHz
	Sleep mode		—	35 (3.3 V)	45	mA	f = 33 MHz
	Standby mode*[3]		—	20	—	µA	$T_a \leq$ 50°C
			—	80	—	µA	50°C < T_a
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	1.0 (3.3 V)	2.0	mA	When channel 1 is in use
	Idle		—	0.01	5.0	µA	When channel 1 is in use
Reference power supply current	During A/D and D/A conversion	AI_{CC}	—	0.5 (3.3 V)	1.0	mA	
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.5	—	—	V	
V_{CC} start voltage*[5]		$V_{CC\ start}$	—	—	0.8	V	
V_{CC} rising slope *[5]		SV_{CC}	—	—	20	ms/V	

Notes: 1. When the A/D and D/A converters are not used, the AV_{CC}, V_{ref}, and AV_{SS} pins should not be open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

2. Current consumption values are for V_{IH}min = V_{CC} –0.2 V and V_{IL}max = 0.2 V with all output pins unloaded and all input pull-up MOSs in the off state.
3. The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH}min = $V_{CC} \times 0.9$, and V_{IL}max = 0.3 V.
4. I_{CC} depends on V_{CC} and f as follows:
   I_{CC}max = 32 (mA) + 1.3 (mA/(MHz)) × f (normal operation)
   I_{CC}max = 18 (mA) + 0.8 (mA/(MHz)) × f (sleep mode)
5. Applied when $\overline{RES}$ is low at power-on.

Table 27.4 Permissible Output Currents

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
$V_{SS} = AV_{SS}$ = 0 V*, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	All output pins except the ICC pins	I_{OL}	—	—	4.0	mA
	ICC output pins	I_{OL}	—	—	8.0	mA
Permissible output low current (total)	Total of all output pins	ΣI_{OL}	—	—	80	mA
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	40	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in table 27.30.

Note: * When the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

27.1.3 AC Characteristics

The following shows the timings of the clock, control signals, bus, DMAC, EXDMAC, and on-chip peripheral functions. For the AC characteristic test conditions, see figure 27.1.

(1) Clock Timing

Table 27.5 Clock Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
Clock cycle time	t_{cyc}	30.3	125	ns	Figure 27.3
Clock pulse high width	t_{CH}	10	—	ns	Figure 27.3
Clock pulse low width	t_{CL}	10	—	ns	
Clock rising time	t_{Cr}	—	5	ns	
Clock falling time	t_{Cf}	—	5	ns	
Reset oscillation settling time (crystal)	t_{OSC1}	15	—	ms	Figure 27.5(1)
Software standby oscillation settling time (crystal)	t_{OSC2}	5	—	ms	Figure 27.5(2)
External clock output delay settling time	t_{DEXT}	15	—	ms	Figure 27.5(1)
Clock phase difference*	t_{cdif}	1/4 × t_{cyc} –3	1/4 × t_{cyc} +3	ns	Figure 27.4
Clock pulse high width (SDRAMϕ)*	t_{SDCH}	9	—	ns	Figure 27.4
Clock pulse low width (SDRAMϕ)*	t_{SDCL}	9	—	ns	Figure 27.4
Clock rising time (SDRAMϕ)*	t_{sdcr}	—	5	ns	Figure 27.4
Clock falling time (SDRAMϕ)*	t_{sdcf}	—	5	ns	Figure 27.4

Note: * Supported only by the H8S/2456R Group.

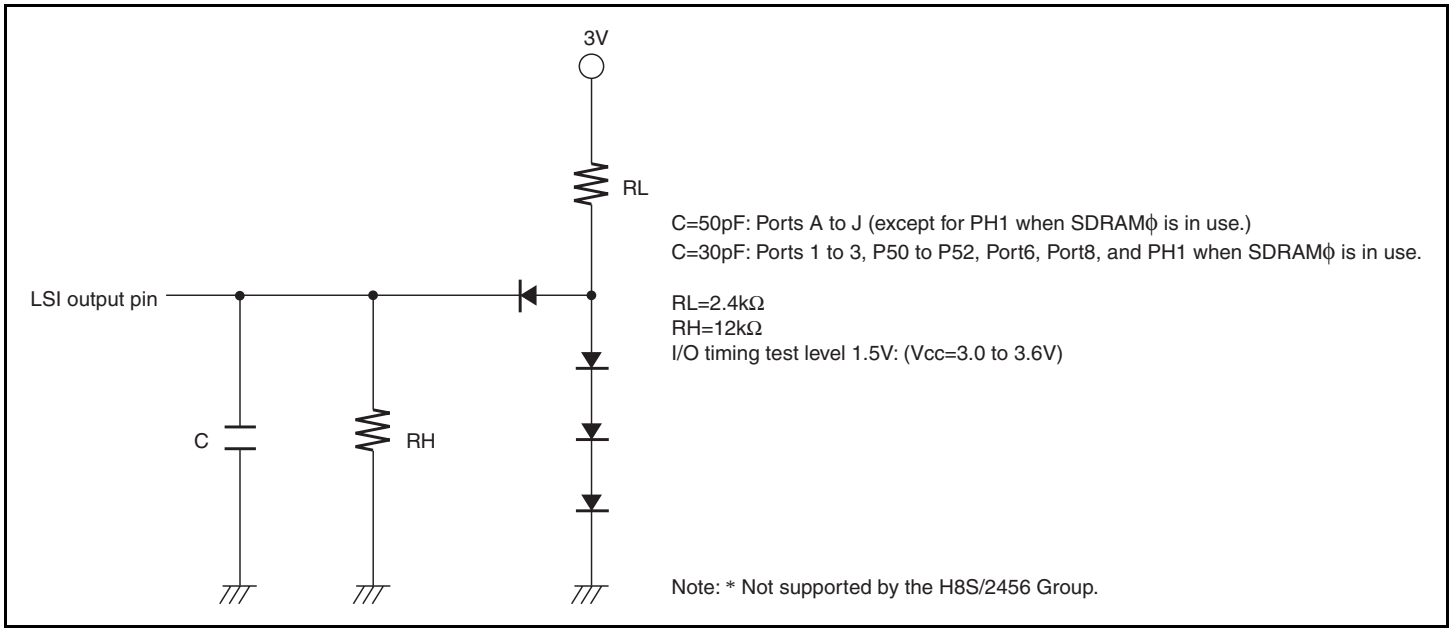


Figure 27.1 Output Load Circuit

(2) Control Signal Timing

Table 27.6 Control Signal Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 27.6
$\overline{RES}$ pulse width	t_{RESW}	2	—	ms	
NMI setup time	t_{NMIS}	150	—	ns	Figure 27.7
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (in recovery from software standby mode)	t_{NMIW}	200	—		
$\overline{IRQ}$ setup time	t_{IRQS}	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—		
$\overline{IRQ}$ pulse width (in recovery from software standby mode)	t_{IRQW}	200	—		

(3) Bus Timing

Table 27.7 Bus Timing (1)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	t_{AD}	—	20	ns	Figures 27.8 to 27.23, Figures 27.34, and 27.35
Address setup time 1	t_{AS1}	$0.5 \times t_{cyc}$ –13	—	ns	
Address setup time 2	t_{AS2}	$1.0 \times t_{cyc}$ –13	—	ns	
Address setup time 3	t_{AS3}	$1.5 \times t_{cyc}$ –13	—	ns	
Address setup time 4	t_{AS4}	$2.0 \times t_{cyc}$ –13	—	ns	
Address hold time 1	t_{AH1}	$0.5 \times t_{cyc}$ –8	—	ns	
Address hold time 2	t_{AH2}	$1.0 \times t_{cyc}$ –8	—	ns	
Address hold time 3	t_{AH3}	$1.5 \times t_{cyc}$ –8	—	ns	
$\overline{CS}$ delay time 1	t_{CSD1}	—	15	ns	
$\overline{CS}$ delay time 2	t_{CSD2}	—	15	ns	
$\overline{CS}$ delay time 3	t_{CSD3}	—	20	ns	
$\overline{AS}$ delay time	t_{ASD}	—	15	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	15	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	15	ns	
Read data setup time 1	t_{RDS1}	15	—	ns	
Read data setup time 2	t_{RDS2}	15	—	ns	
Read data hold time 1	t_{RDH1}	0	—	ns	
Read data hold time 2	t_{RDH2}	0	—	ns	
Read data access time 1	t_{AC1}	—	$1.0 \times t_{cyc}$ – 25	ns	
Read data access time 2	t_{AC2}	—	$1.5 \times t_{cyc}$ – 25	ns	
Read data access time 3	t_{AC3}	—	$2.0 \times t_{cyc}$ – 25	ns	
Read data access time 4	t_{AC4}	—	$2.5 \times t_{cyc}$ – 25	ns	
Read data access time 5	t_{AC5}	—	$1.0 \times t_{cyc}$ – 25	ns	
Read data access time 6	t_{AC6}	—	$2.0 \times t_{cyc}$ – 25	ns	
Read data access time 7	t_{AC7}	—	$4.0 \times t_{cyc}$ – 25	ns	
Read data access time 8	t_{AC8}	—	$3.0 \times t_{cyc}$ – 25	ns	
Counter address read data access time 1	t_{AA1}	—	$1.0 \times t_{cyc}$ – 25	ns	
Counter address read data access time 2	t_{AA2}	—	$1.5 \times t_{cyc}$ – 25	ns	
Counter address read data access time 3	t_{AA3}	—	$2.0 \times t_{cyc}$ – 25	ns	
Counter address read data access time 4	t_{AA4}	—	$2.5 \times t_{cyc}$ – 25	ns	
Counter address read data access time 5	t_{AA5}	—	$3.0 \times t_{cyc}$ – 25	ns	
Counter address read data access time 6	t_{AA6}	—	$4.0 \times t_{cyc}$ – 25	ns	

Item	Symbol	Min.	Max.	Unit	Test Conditions
Multiplexed address delay time	t_{MAD}	—	20	ns	Figures 27.8 to 27.23, Figures 27.34, and 27.35
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	
Multiplexed address hold time	t_{MAH}	$1.0 \times t_{cyc} - 15$	—	ns	
$\overline{AH}$ delay time	t_{AHD}	—	15	ns	

Table 27.8 Bus Timing (2)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{WR}$ delay time 1	t_{WRD1}	—	15	ns	Figures 27.8 to 27.23, Figure 27.34, and figure 27.35
$\overline{WR}$ delay time 2	t_{WRD2}	—	15	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 13$	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 13$	—	ns	
Write data delay time	t_{WDD}	—	23	ns	
Write data setup time 1	t_{WDS1}	$0.5 \times t_{cyc} - 15$	—	ns	
Write data setup time 2	t_{WDS2}	$1.0 \times t_{cyc} - 15$	—	ns	
Write data setup time 3	t_{WDS3}	$1.5 \times t_{cyc} - 15$	—	ns	
Write data hold time 1	t_{WDH1}	$0.5 \times t_{cyc} - 13$	—	ns	
Write data hold time 2	t_{WDH2}	$1.0 \times t_{cyc} - 13$	—	ns	
Write data hold time 3	t_{WDH3}	$1.5 \times t_{cyc} - 13$	—	ns	
Write command setup time 1	t_{WCS1}	$0.5 \times t_{cyc} - 10$	—	ns	
Write command setup time 2	t_{WCS2}	$1.0 \times t_{cyc} - 10$	—	ns	
Write command hold time 1	t_{WCH1}	$0.5 \times t_{cyc} - 10$	—	ns	
Write command hold time 2	t_{WCH2}	$1.0 \times t_{cyc} - 10$	—	ns	
Read command setup time 1	t_{RCS1}	$1.5 \times t_{cyc} - 10$	—	ns	
Read command setup time 2	t_{RCS2}	$2.0 \times t_{cyc} - 10$	—	ns	
Read command hold time	t_{RCH}	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{CAS}$ delay time 1	t_{CASD1}	—	15	ns	
$\overline{CAS}$ delay time 2	t_{CASD2}	—	15	ns	
$\overline{CAS}$ setup time 1	t_{CSR1}	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{CAS}$ setup time 2	t_{CSR2}	$1.5 \times t_{cyc} - 10$	—	ns	
$\overline{CAS}$ pulse width 1	t_{CASW1}	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{CAS}$ pulse width 2	t_{CASW2}	$1.5 \times t_{cyc} - 20$	—	ns	
$\overline{CAS}$ precharge time 1	t_{CPW1}	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{CAS}$ precharge time 2	t_{CPW2}	$1.5 \times t_{cyc} - 20$	—	ns	

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{OE}$ delay time 1*[1]	t_{OED1}	—	15	ns	Figures 27.8 to 27.23, Figures 27.34 and 27.35
	t_{OED1B}	—	19	ns	
$\overline{OE}$ delay time 2*[1]	t_{OED2}	—	15	ns	
	t_{OED2B}	—	19	ns	
Precharge time 1	t_{PCH1}	$1.0 \times t_{cyc} - 20$	—	ns	
Precharge time 2	t_{PCH2}	$1.5 \times t_{cyc} - 20$	—	ns	
Self-refresh precharge time 1	t_{RPS1}	$2.5 \times t_{cyc} - 20$	—	ns	Figures 27.22 and 27.23
Self-refresh precharge time 2	t_{RPS2}	$3.0 \times t_{cyc} - 20$	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	25	—	ns	Figures 27.10, 27.16, and 27.35
$\overline{WAIT}$ hold time	t_{WTH}	1	—	ns	
$\overline{BREQ}$ setup time	t_{BREQS}	30	—	ns	Figure 27.24
$\overline{BACK}$ delay time	t_{BACD}	—	15	ns	
Bus floating time	t_{BZD}	—	40	ns	
$\overline{BREQO}$ delay time	t_{BRQOD}	—	25	ns	Figure 27.25
Address delay time 2*[2]	t_{AD2}	—	16.5	ns	Figure 27.26
$\overline{CS}$ delay time 4*[2]	t_{CSD4}	—	16.5	ns	Figure 27.26
DQM delay time*[2]	t_{DQMD}	—	16.5	ns	Figure 27.26
$\overline{CKE}$ delay time 1*[2]*[3]	t_{CKED}	—	16.5	ns	Figures 27.27 and 27.28
	t_{CKEDB}	—	19	ns	
Read data setup time 3*[2]	t_{RDS3}	15	—	ns	Figure 27.26
Read data hold time 3*[2]	t_{RDH3}	0	—	ns	Figure 27.26
Write data delay time 2*[2]	t_{WDD}	—	31.5	ns	Figure 27.26
Write data hold time 4*[2]	t_{WDH4}	2	—	ns	Figure 27.26

Notes: 1. t_{OED1}, and t_{OED2} correspond to the $\overline{OE}$-A, t_{OED1B}, and t_{OED2B} correspond to the $\overline{OE}$-B.
2. Supported only by the H8S/2456R Group.
3. t_{CKED} corresponds to the $\overline{CKE}$-A, t_{CKEDB} corresponds to the $\overline{CKE}$-B.

(4) DMAC and EXDMAC Timing

Table 27.9 DMAC and EXDMAC Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{\text{DREQ}}$ setup time	t_{DRQS}	25	—	ns	Figure 27.32
$\overline{\text{DREQ}}$ hold time	t_{DRQH}	10	—		
$\overline{\text{TEND}}$ delay time	t_{TED}	—	18	ns	Figure 27.31
$\overline{\text{DACK}}$ delay time 1	t_{DACD1}	—	18		Figures 27.29 and 27.30
$\overline{\text{DACK}}$ delay time 2	t_{DACD2}	—	18		
$\overline{\text{EDREQ}}$ setup time	t_{EDRQS}	25	—	ns	Figure 27.32
$\overline{\text{EDREQ}}$ hold time	t_{EDRQH}	10	—		
$\overline{\text{ETEND}}$ delay time	t_{ETED}	—	18	ns	Figure 27.31
$\overline{\text{EDACK}}$ delay time 1	t_{EDACD1}	—	18	ns	Figures 27.29 and 27.30
$\overline{\text{EDACK}}$ delay time 2	t_{EDACD2}	—	18		
$\overline{\text{EDRAK}}$ delay time	t_{EDRKD}	—	18	ns	Figure 27.33

(5) Timing of On-Chip Peripheral Modules

Table 27.10 Timing of On-Chip Peripheral Modules

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item			Symbol	Min.	Max.	Unit	Test Conditions
I/O ports	Output data delay time		t_{PWD}	—	40	ns	Figure 27.39
	Input data setup time		t_{PRS}	25	—	ns	
	Input data hold time		t_{PRH}	25	—	ns	
PPG	Pulse output delay time		t_{POD}	—	40	ns	Figure 27.40
TPU	Timer output delay time		t_{TOCD}	—	40	ns	Figure 27.41
	Timer input setup time		t_{TICS}	25	—	ns	
	Timer clock input setup time		t_{TCKS}	25	—	ns	Figure 27.42
	Timer clock pulse width	Single-edge specification	t_{TCKWH}	1.5	—	t_{cyc}	
		Both-edge specification	t_{TCKWL}	2.5	—	t_{cyc}	
8-bit timer	Timer output delay time		t_{TMOD}	—	40	ns	Figure 27.43
	Timer reset input setup time		t_{TMRS}	25	—	ns	Figure 27.45
	Timer clock input setup time		t_{TMCS}	25	—	ns	Figure 27.44
	Timer clock pulse width	Single-edge specification	t_{TMCWH}	1.5	—	t_{cyc}	
		Both-edge specification	t_{TMCWL}	2.5	—	t_{cyc}	
WDT	Overflow output delay time		t_{WOVD}	—	40	ns	Figure 27.46
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 27.47
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rising time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock falling time		t_{SCKf}	—	1.5		
	Transmit data delay time		t_{TXD}	—	40	ns	Figure 27.48
	Receive data setup time (synchronous)		t_{RXS}	40	—	ns	
	Receive data hold time (synchronous)		t_{RXH}	40	—	ns	

Item			Symbol	Min.	Max.	Unit	Test Conditions
A/D converter	Trigger input setup time		t_{TRGS}	30	—	ns	Figure 27.49
IIC2	SCL input cycle time		t_{SCL}	12 t_{cyc} +600	—	ns	Figure 27.50
	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	
SSU*	Clock cycle	Master	t_{SUcyc}	4	256	t_{cyc}	Figures 27.51 to 27.54
		Slave		4	256		
	Clock high pulse width	Master	t_{HI}	80	—	ns	
		Slave		80	—		
	Clock low pulse width	Master	t_{LO}	80	—	ns	
		Slave		80	—		
	Clock rising time		t_{RISE}	—	20	ns	
	Clock falling time		t_{FALL}	—	20	ns	
	Data input setup time	Master	t_{SU}	25	—	ns	
		Slave		30	—		
	Data input hold time	Master	t_H	10	—	ns	
		Slave		10	—		
	$\overline{SCS}$ setup time	Master	t_{LEAD}	2.5	—	t_{cyc}	
		Slave		2.5	—		

Item			Symbol	Min.	Max.	Unit	Test Conditions
SSU*	$\overline{SCS}$ hold time	Master	t_{LAG}	2.5	—	t_{cyc}	
		Slave		2.5	—		
	Data output delay time	Master	t_{OD}	—	40	ns	
		Slave		—	40		
	Data output hold time	Master	t_{OH}	0	—	ns	
		Slave		0	—		
	Continuous transmit delay time	Master	t_{TD}	2.5	—	t_{cyc}	
		Slave		2.5	—		
	Slave access time		t_{SA}	—	1	t_{cyc}	Figures 27.53 and 27.54
	Slave out release time		t_{REL}	—	1	t_{cyc}	

Note * SSU: Synchronous serial communication unit

27.1.4 A/D Conversion Characteristics

Table 27.11 A/D Conversion Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Min.	Typ.	Max.	Unit
Resolution	10	10	10	Bit
Conversion time	4.0*	—	—	µs
Analog input capacitance	—	—	15	pF
Permissible signal source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±5.5	LSB
Offset error	—	—	±5.5	LSB
Full-scale error	—	—	±5.5	LSB
Quantization error	—	—	±0.5	LSB
Absolute accuracy	—	—	±6.0	LSB

Note: * For 40 states at ADCLK = 10 MHz.

27.1.5 D/A Conversion Characteristics

Table 27.12 D/A Conversion Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	8	8	8	Bit	
Conversion time	—	—	10	µs	15 pF capacitive load
Absolute accuracy	—	±2.0	±3.0	LSB	2 MΩ resistive load
	—	—	±2.0	LSB	4 MΩ resistive load

27.1.6 USB Characteristics

Table 27.13 USB Characteristics when On-Chip USB Transceiver is Used (USD+, USD- pin characteristics)

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 3.0 V to 3.6 V, V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0V,
CKU = 48MHz T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Max	Unit	Test Conditions	
Input	Input high voltage	V_{IH}	2.0	—	V		Figure 27.36
	Input low voltage	V_{IL}	—	0.8	V		Figure 27.37
	Differential input sensitivity	V_{DI}	0.2	—	V	|(D+)-(D-)|	
	Differential common mode range	V_{CM}	0.8	2.5	V		
Output	Output high voltage	V_{OH}	2.8	—	V	I_{OH} =-200µA	
	Output low voltage	V_{OL}	—	0.3	V	I_{OL} = 2mA	
	Crossover voltage	V_{CRS}	1.3	2.0	V		
	Rising time	t_R	4	20	ns		
	Falling time	t_F	4	20	ns		
	Ratio of rising time to falling time	t_{RFM}	90	111.11	%	(T_R/T_F)	
	Output resistance	Z_{DRV}	28	44	Ω	Including RS = 27Ω	

Table 27.14 USB PLL Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6, V_{ref} = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0V, EXTAL = 8 to 16 MHz,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
PLL for USB: oscillation stabilization time	t_{USOSC}	1	—	ms	Figure 27.38

27.1.7 Flash Memory Characteristics

Table 27.15 Flash Memory Characteristics

Conditions: $V_{CC} = AV_{CC} = 3.0$ to 3.6V, $V_{SS} = AV_{SS} = 0$V, $T_a = 0°C$ to + 75°C

Item	Symbol	Test conditions	Applicable area	Standard value			Unit
				Min.	Typ.	Max.	
Programming and erase count*1			Programming ROM	100*2	—	—	Times
			Data flash area*3	TBD*3	—	—	
Programming time (per 4 bytes)			Programming ROM	—	150	—	µs
			Data flash area*3	—	300*3	—	
Erase time (per 1 block)			Programming ROM	—	300	—	ms
			Data flash area*3	—	300*3	—	
Programming and erase voltage			Programming ROM	3.0	—	3.6	V
			Data flash area*3				
Read voltage			Programming ROM	3.0	—	3.6	V
			Data flash area*3				
Access state			Programming ROM	1	—	—	State
			Data flash area*3	2	—	—	
Programming and erase temperature		-	Programming ROM	0	—	75	°C
			Data flash area*3	0	—	75	

Notes: 1. When programming is to be performed multiple times on a system, reduce the effective number of programming operations by shifting the writing addresses in sequence and so on until the remaining blank area is as small as possible and only then erasing the entire block once. For example, if sets of 16 bytes are being programmed, erasing the block once after programming the maximum number of sets (256) minimizes the effective number of programming operations.

We recommend keeping information on the number of times erasure is performed for each block, and setting up the limit on the number of times.

2. If an erase error occurs during erasure, execute the clear status command and then the erase command at least 3 times until the erase does not recur.

*1. Determination of the number of times for programming /erasure operations.

Number of times programming / erasure is performed in each block.

When the number of times for programming / erasure operations is n (n = 100), data can be erased n times in each block.

For example, if programming of 4 bytes is done 1024 times, each at a different address in a 4-kbyte per block, and the block is then erased, this counts as programming / erasure one time.

However, programming of any location in a block multiple times is not possible (overwriting is prohibited).

*2. This is the number of times for which all electrical characteristics are guaranteed.

*3 Values for the data flush are in planning.

27.2 Electrical Characteristics for H8S/2454 Group

27.2.1 Absolute Maximum Ratings

Table 27.16 lists the absolute maximum ratings.

Table 27.16 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC} $PLLV_{CC}$ D_rV_{CC}	–0.3 to +4.3	V
Input voltage (except ports 4, 9, 2, P32 to P35, P50, P51, P81, and P83)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (ports 2, P32 to P35, P50, P51, P81, and P83)	V_{in}	–0.3 to +0.7	V
Input voltage (ports 4 and 9)	V_{in}	–0.3 to AV_{CC} +0.3	V
Reference power supply voltage	V_{ref}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +4.3	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75*	°C
		Wide-range specifications: –40 to +85*	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

Note: * Ranges of operating temperature when flash memory is programmed/erased:
Regular specifications: 0 to +75°C
Wide-range specifications: 0 to +85°C

27.2.2 DC Characteristics

Table 27.17 DC Characteristics (1)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V*1, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	Port 1, port 2, P32 to P35*2, P50 to P53*2, port 8*2, PA4 to PA7*2, port B*2, port C*2, PF1*2, PF2*2, P81*2, P83*2	VT^-	$V_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.07$	—	—	V	
Input high voltage	$\overline{STBY}$, MD2 to MD0	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} +0.3	V	
	$\overline{RES}$, NMI, EMLE						
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	P14 to P17*5, P24 to P26*6, port 3, P50 to P53*3, port 8*3, ports A to G*3		2.2	—	V_{CC} +0.3	V	
	Port 4, Port 9		2.2	—	AV_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD2 to MD0, EMLE	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL		–0.3	—	$V_{CC} \times 0.2$	V	
	P14 to P17*5 P24 to P26*6 Ports 3, 5, and 6, Port 8, Ports A to J*3,		–0.3	—	$V_{CC} \times 0.2$	V	
	Port 4, Port 9		–0.3	—	$AV_{CC} \times 0.2$	V	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.3	—	—	V	I_{OH} = –200 µA
			V_{CC} – 0.5	—	—	V	I_{OH} = –1 mA
			V_{CC} – 0.8	—	—	V	I_{OH} = –2 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 4.0 mA
	P26 to P27*4, P32 to P35*4, P50 to P51*4		—	—	0.4	V	I_{OL} = 8.0 mA
Input leakage current	$\overline{RES}$	$\|I_{in}\|$	—	—	10.0	µA	V_{in} = 0.5 to V_{CC} –0.5 V
	$\overline{STBY}$, NMI, MD2 to MD0		—	—	1.0	µA	
	Port 4, Port 9		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} –0.5 V

Notes: Port 2, P32 to P35, P50, P51, P81, and P83 are 5 V-tolerant pins.

1. When the A/D and D/A converters are not used, the AV_{CC}, V_{ref}, and AV_{SS} pins should not be open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.
2. When used as $\overline{IRQ}$, TIOC, TCLK, TMRI, SCL, or SDA.
3. When used as other than $\overline{IRQ}$, TIOC, TCLK, TMRI, SCL, or SDA.
4. When used as SCL or SDA.
5. When used as SSO, SSI, SSCK, or $\overline{SCS}$.
6. When used as RxD, $\overline{WAIT}$, or $\overline{ADTRG1}$.

Table 27.18 DC Characteristics (2)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
$V_{SS} = AV_{SS}$ = 0 V*[1], T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Three-state leakage current (off state)	Ports 1 to 3, P50 to P53, ports 6 and 8, ports A to I	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} –0.5 V
Input pull-up MOS current	Ports A to E	$-I_p$	10	—	300	µA	V_{CC} = 3.0 to 3.6 V V_{in} = 0 V
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V
	NMI		—	—	30	pF	f = 1 MHz
	All input pins except $\overline{RES}$ and NMI		—	—	18	pF	T_a = 25°C
Current consumption*[2]	Normal operation	I_{CC}*[4]	—	55 (3.3 V)	75	mA	f = 33 MHz
	Sleep mode		—	35 (3.3 V)	45	mA	f = 33 MHz
	Standby mode*[3]		—	20	—	µA	$T_a \leq$ 50°C
			—	80	—	µA	50°C < T_a
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	1.0 (3.3 V)	2.0	mA	When channel 1 is in use
	Idling		—	0.01	5.0	µA	When channel 1 is in use
Reference power supply current	During A/D and D/A conversion	AI_{CC}	—	0.5 (3.3 V)	6.0	mA	
	Idling		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.5	—	—	V	
V_{CC} start voltage*[5]		$V_{CC\ start}$	—	—	0.8	V	
V_{CC} rising slope *[5]		SV_{CC}	—	—	20	ms/V	

Notes: 1. When the A/D and D/A converters are not used, the AV_{CC}, V_{ref}, and AV_{SS} pins should not be open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

2. Current consumption values are for V_{IH}min = V_{CC} –0.2 V and V_{IL}max = 0.2 V with all output pins unloaded and all input pull-up MOSs in the off state.
3. The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH}min = $V_{CC} \times 0.9$, and V_{IL}max = 0.3 V.
4. I_{CC} depends on V_{CC} and f as follows:
   I_{CC}max = 32 (mA) + 1.3 (mA/(MHz)) × f (normal operation)
   I_{CC}max = 18 (mA) + 0.8 (mA/(MHz)) × f (sleep mode)
5. Applied when $\overline{RES}$ is low at power-on.

Table 27.19 Permissible Output Currents

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
$V_{SS} = AV_{SS}$ = 0 V*, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	All output pins except the ICC pins	I_{OL}	—	—	2.0	mA
	ICC output pins	I_{OL}	—	—	8.0	mA
Permissible output low current (total)	Total of all output pins	ΣI_{OL}	—	—	80	mA
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	40	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in table 27.30.

Note: * When the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

27.2.3 AC Characteristics

The following shows the timings of the clock, control signals, bus, DMAC, and on-chip peripheral functions. For the AC characteristic test conditions, see figure 27.2.

(1) Clock Timing

Table 27.20 Clock Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
Clock cycle time	t_{cyc}	30.3	125	ns	Figure 27.3
Clock pulse high width	t_{CH}	10	—	ns	Figure 27.3
Clock pulse low width	t_{CL}	10	—	ns	
Clock rising time	t_{Cr}	—	5	ns	
Clock falling time	t_{Cf}	—	5	ns	
Reset oscillation settling time (crystal)	t_{OSC1}	15	—	ms	Figure 27.5(1)
Software standby oscillation settling time (crystal)	t_{OSC2}	5	—	ms	Figure 27.5(2)
External clock output delay settling time	t_{DEXT}	15	—	ms	Figure 27.5(1)

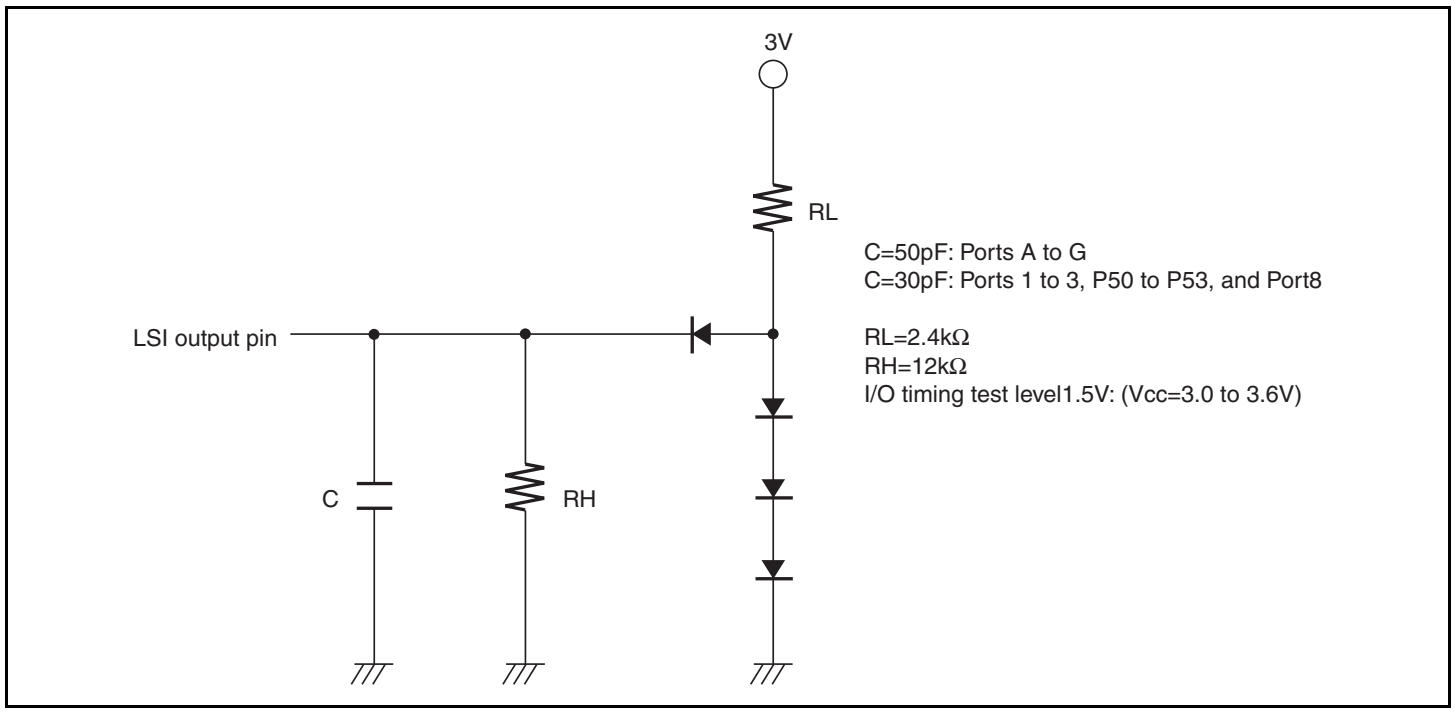


Figure 27.2 Output Load Circuit

(2) Control Signal Timing

Table 27.21 Control Signal Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 27.6
$\overline{RES}$ pulse width	t_{RESW}	2	—	ms	
NMI setup time	t_{NMIS}	150	—	ns	Figure 27.7
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (in recovery from software standby mode)	t_{NMIW}	200	—		
$\overline{IRQ}$ setup time	t_{IRQS}	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—		
$\overline{IRQ}$ pulse width (in recovery from software standby mode)	t_{IRQW}	200	—		

(3) Bus Timing

Table 27.22 Bus Timing (1)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	t_{AD}	—	20	ns	Figures 27.8 to 27.23, Figures 27.34, and 27.35
Address setup time 1	t_{AS1}	$0.5 \times t_{cyc}$ –13	—	ns	
Address setup time 2	t_{AS2}	$1.0 \times t_{cyc}$ –13	—	ns	
Address setup time 3	t_{AS3}	$1.5 \times t_{cyc}$ –13	—	ns	
Address setup time 4	t_{AS4}	$2.0 \times t_{cyc}$ –13	—	ns	
Address hold time 1	t_{AH1}	$0.5 \times t_{cyc}$ –8	—	ns	
Address hold time 2	t_{AH2}	$1.0 \times t_{cyc}$ –8	—	ns	
Address hold time 3	t_{AH3}	$1.5 \times t_{cyc}$ –8	—	ns	
$\overline{CS}$ delay time 1	t_{CSD1}	—	15	ns	
$\overline{CS}$ delay time 2	t_{CSD2}	—	15	ns	
$\overline{CS}$ delay time 3	t_{CSD3}	—	20	ns	
$\overline{AS}$ delay time	t_{ASD}	—	15	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	15	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	15	ns	
Read data setup time 1	t_{RDS1}	15	—	ns	
Read data setup time 2	t_{RDS2}	15	—	ns	
Read data hold time 1	t_{RDH1}	0	—	ns	
Read data hold time 2	t_{RDH2}	0	—	ns	
Read data access time 1	t_{AC1}	—	$1.0 \times t_{cyc}$ – 25	ns	
Read data access time 2	t_{AC2}	—	$1.5 \times t_{cyc}$ – 25	ns	
Read data access time 3	t_{AC3}	—	$2.0 \times t_{cyc}$ – 25	ns	
Read data access time 4	t_{AC4}	—	$2.5 \times t_{cyc}$ – 25	ns	
Read data access time 5	t_{AC5}	—	$1.0 \times t_{cyc}$ – 25	ns	
Read data access time 6	t_{AC6}	—	$2.0 \times t_{cyc}$ – 25	ns	
Read data access time 7	t_{AC7}	—	$4.0 \times t_{cyc}$ – 25	ns	
Read data access time 8	t_{AC8}	—	$3.0 \times t_{cyc}$ – 25	ns	
Counter address read data access time 1	t_{AA1}	—	$1.0 \times t_{cyc}$ – 25	ns	
Counter address read data access time 2	t_{AA2}	—	$1.5 \times t_{cyc}$ – 25	ns	
Counter address read data access time 3	t_{AA3}	—	$2.0 \times t_{cyc}$ – 25	ns	
Counter address read data access time 4	t_{AA4}	—	$2.5 \times t_{cyc}$ – 25	ns	
Counter address read data access time 5	t_{AA5}	—	$3.0 \times t_{cyc}$ – 25	ns	
Counter address read data access time 6	t_{AA6}	—	$4.0 \times t_{cyc}$ – 25	ns	

Item	Symbol	Min.	Max.	Unit	Test Conditions
Multiplex address delay time 6	T_{MAD}	—	20	ns	Figures 27.8 to 27.23, Figures 27.34, and 27.35
Multiplex address setup time 1	T_{MAS1}	$0.5 \times t_{cyc} - 25$	—	ns	
Multiplex address setup time 2	T_{MAS2}	$1.5 \times t_{cyc} - 25$	—	ns	
Multiplex address hold time	T_{MAH}	$1.0 \times t_{cyc} - 25$	—	ns	
$\overline{AH}$ delay time	T_{AHD}	—	15	ns	

Table 27.22 Bus Timing (2)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{WR}$ delay time 1	t_{WRD1}	—	15	ns	Figures 27.8 to 27.23, Figures 27.34 and 27.35
$\overline{WR}$ delay time 2	t_{WRD2}	—	15	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	1.0 × t_{cyc} –13	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	1.5 × t_{cyc} –13	—	ns	
Write data delay time	t_{WDD}	—	23	ns	
Write data setup time 1	t_{WDS1}	0.5 × t_{cyc} –15	—	ns	
Write data setup time 2	t_{WDS2}	1.0 × t_{cyc} –15	—	ns	
Write data setup time 3	t_{WDS3}	1.5 × t_{cyc} –15	—	ns	
Write data hold time 1	t_{WDH1}	0.5 × t_{cyc} –13	—	ns	
Write data hold time 2	t_{WDH2}	1.0 × t_{cyc} –13	—	ns	
Write data hold time 3	t_{WDH3}	1.5 × t_{cyc} –13	—	ns	
Write command setup time 1	t_{WCS1}	0.5 × t_{cyc} –10	—	ns	
Write command setup time 2	t_{WCS2}	1.0 × t_{cyc} –10	—	ns	
Write command hold time 1	t_{WCH1}	0.5 × t_{cyc} –10	—	ns	
Write command hold time 2	t_{WCH2}	1.0 × t_{cyc} –10	—	ns	
Read command setup time 1	t_{RCS1}	1.5 × t_{cyc} –10	—	ns	
Read command setup time 2	t_{RCS2}	2.0 × t_{cyc} –10	—	ns	
Read command hold time	t_{RCH}	0.5 × t_{cyc} –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 × t_{cyc} –10	—	ns	
$\overline{CAS}$ setup time 2	t_{CSR2}	1.5 × t_{cyc} –10	—	ns	
$\overline{CAS}$ pulse width 1	t_{CASW1}	1.0 × t_{cyc} –20	—	ns	
$\overline{CAS}$ pulse width 2	t_{CASW2}	1.5 × t_{cyc} –20	—	ns	
$\overline{CAS}$ precharge time 1	t_{CPW1}	1.0 × t_{cyc} –20	—	ns	
$\overline{CAS}$ precharge time 2	t_{CPW2}	1.5 × t_{cyc} –20	—	ns	

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{OE}$ delay time 1*	t_{OED1}	—	15	ns	Figures 27.8 to 27.23, Figures 27.34 and 27.35
	t_{OED1B}		19	ns	
$\overline{OE}$ delay time 2*	t_{OED2}	—	15	ns	
	t_{OED2B}		19	ns	
Precharge time 1	t_{PCH1}	$1.0 \times t_{cyc} - 20$	—	ns	
Precharge time 2	t_{PCH2}	$1.5 \times t_{cyc} - 20$	—	ns	
Self-refresh precharge time 1	t_{RPS1}	$2.5 \times t_{cyc} - 20$	—	ns	Figures 27.22 and 27.23
Self-refresh precharge time 2	t_{RPS2}	$3.0 \times t_{cyc} - 20$	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	25	—	ns	Figures 27.10, 27.16, and 27.35
$\overline{WAIT}$ hold time	t_{WTH}	1	—	ns	
$\overline{BREQ}$ setup time	t_{BREQS}	30	—	ns	Figure 27.24
$\overline{BACK}$ delay time	t_{BACD}	—	15	ns	
Bus floating time	t_{BZD}	—	40	ns	
$\overline{BREQO}$ delay time	t_{BRQOD}	—	25	ns	Figure 27.25

Note: * t_{OED1} and t_{OED2}.correspond to $\overline{OE}$-A, and t_{OED1B} and t_{OED2B}.correspond to $\overline{OE}$-B.

(4) DMAC Timing

Table 27.23 DMAC Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{DREQ}$ setup time	t_{DRQS}	25	—	ns	Figure 27.32
$\overline{DREQ}$ hold time	t_{DRQH}	10	—		
$\overline{TEND}$ delay time	t_{TED}	—	18		Figure 27.31
$\overline{DACK}$ delay time 1	t_{DACD1}	—	18		Figures 27.29 and 27.30
$\overline{DACK}$ delay time 2	t_{DACD2}	—	18		

(5) Timing of On-Chip Peripheral Modules

Table 27.24 Timing of On-Chip Peripheral Modules

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item			Symbol	Min.	Max.	Unit	Test Conditions
I/O ports	Output data delay time		t_{PWD}	—	40	Ns	Figure 27.39
	Input data setup time		t_{PRS}	25	—	Ns	
	Input data hold time		t_{PRH}	25	—	Ns	
PPG	Pulse output delay time		t_{POD}	—	40	Ns	Figure 27.40
TPU	Timer output delay time		t_{TOCD}	—	40	Ns	Figure 27.41
	Timer input setup time		t_{TICS}	25	—	Ns	
	Timer clock input setup time		t_{TCKS}	25	—	Ns	Figure 27.42
	Timer clock pulse width	Single-edge specification	t_{TCKWH}	1.5	—	t_{cyc}	
		Both-edge specification	t_{TCKWL}	2.5	—	t_{cyc}	
8-bit timer	Timer output delay time		t_{TMOD}	—	40	Ns	Figure 27.43
	Timer reset input setup time		t_{TMRS}	25	—	Ns	Figure 27.45
	Timer clock input setup time		t_{TMCS}	25	—	Ns	Figure 27.44
	Timer clock pulse width	Single-edge specification	t_{TMCWH}	1.5	—	t_{cyc}	
		Both-edge specification	t_{TMCWL}	2.5	—	t_{cyc}	
WDT	Overflow output delay time		t_{WOVD}	—	40	Ns	Figure 27.46
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 27.47
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rising time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock falling time		t_{SCKf}	—	1.5		
	Transmit data delay time		t_{TXD}	—	40	Ns	Figure 27.48
	Receive data setup time (synchronous)		t_{RXS}	40	—	Ns	
	Receive data hold time (synchronous)		t_{RXH}	40	—	Ns	

Item			Symbol	Min.	Max.	Unit	Test Conditions
A/D converter	Trigger input setup time		t_{TRGS}	30	—	Ns	Figure 27.49
IIC2	SCL input cycle time		t_{SCL}	12 t_{cyc} +600	—	Ns	Figure 27.50
	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	
IIC2	SCL, SDA Input spike pulse removal time		t_{SP}	—	1 t_{cyc}	Ns	Figure 27.50
	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	
SSU*	Clock cycle	Master	t_{SUcyc}	4	256	t_{cyc}	Figures 27.51 to 27.54
		Slave		4	256		
	Clock high pulse width	Master	t_{HI}	80	—	Ns	
		Slave		80	—		
	Clock low pulse width	Master	t_{LO}	80	—	Ns	
		Slave		80	—		
	Clock rising time		t_{RISE}	—	20	Ns	
	Clock falling time		t_{FALL}	—	20	Ns	
	Data input setup time	Master	t_{SU}	25	—	Ns	
		Slave		30	—		
	Data input hold time	Master	t_H	10	—	Ns	
		Slave		10	—		
	$\overline{SCS}$ setup time	Master	t_{LEAD}	2.5	—	t_{cyc}	
		Slave		2.5	—		

Item			Symbol	Min.	Max.	Unit	Test Conditions
SSU*	$\overline{SCS}$ hold time	Master	t_{LAG}	2.5	—	t_{cyc}	Figures 27.51 to 27.54
		Slave		2.5	—		
	Data output delay time	Master	t_{OD}	—	40	Ns	
		Slave		—	40		
	Data output hold time	Master	t_{OH}	0	—	Ns	
		Slave		0	—		
	Continuous transmit delay time	Master	t_{TD}	2.5	—	t_{cyc}	
		Slave		2.5	—		
	Slave access time		t_{SA}	—	1	t_{cyc}	Figures 27.53 and 27.54
	Slave out release time		t_{REL}	—	1	t_{cyc}	

Note * SSU: Synchronous serial communication unit

27.2.4 A/D Conversion Characteristics

Table 27.25 A/D Conversion Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Min.	Typ.	Max.	Unit
Resolution	10	10	10	Bit
Conversion time	4.0*	—	—	µs
Analog input capacitance	—	—	15	pF
Permissible signal source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±5.5	LSB
Offset error	—	—	±5.5	LSB
Full-scale error	—	—	±5.5	LSB
Quantization error	—	—	±0.5	LSB
Absolute accuracy	—	—	±6.0	LSB

Note: * For 40 states at ADCLK = 10 MHz.

27.2.5 D/A Conversion Characteristics

Table 27.26 D/A Conversion Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	8	8	8	Bit	
Conversion time	—	—	10	µs	15 pF capacitive load
Absolute accuracy	—	±2.0	±3.0	LSB	2 MΩ resistive load
	—	—	±2.0	LSB	4 MΩ resistive load

27.2.6 USB Characteristics

Table 27.27 USB Characteristics when On-Chip USB Transceiver is Used (USD+, USD- pin characteristics)

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 3.0 V to 3.6 V, V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0V,
CKU = 48MHz T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	min	max	Unit	Test Conditions	
Input	Input high voltage	V_{IH}	2.0	—	V		Figures 27.35
	Input low voltage	V_{IL}	—	0.8	V		Figures 27.36
	Differential input sensitivity	V_{DI}	0.2	—	V	|(D+)-(D-)|	
	Differential common mode range	V_{CM}	0.8	2.5	V		
Output	Output high voltage	V_{OH}	2.8	—	V	I_{OH} =-200µA	
	Output low voltage	V_{OL}	—	0.3	V	I_{OL} = 2mA	
	Crossover voltage	V_{CRS}	1.3	2.0	V		
	Rising time	t_R	4	20	ns		
	Falling time	t_F	4	20	ns		
	Ratio of rising time to falling time	t_{RFM}	90	111.11	%	(T_R/T_F)	
	Output resistance	Z_{DRV}	28	44	Ω	Including RS = 27Ω	

Table 27.28 USB PLL Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6, V_{ref} = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0V, EXTAL = 8 to 16 MHz,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
PLL for USB: oscillation stabilization time	t_{USOSC}	1	—	ms	Figure 27.37

27.2.7 Flash Memory Characteristics

Table 27.29 Flash Memory Characteristics

Conditions: $V_{CC} = AV_{CC} = 3.0$ to 3.6V, $V_{SS} = AV_{SS} = 0$V, $T_a = 0°C$ to + 75°C

Item	Symbol	Test conditions	Applicable area	Standard value			Unit
				Min.	Typ.	Max.	
Programming and erase count*[1]			Programming ROM	100*[2]	—	—	Times
			Data flash area*[3]	TBD*[3]	—	—	
Programming time (per 4 bytes)			Programming ROM	—	150	—	µs
			Data flash area*[3]	—	300*[3]	—	
Erase time (per 1 block)			Programming ROM	—	300	—	ms
			Data flash area*[3]	—	300*[3]	—	
Programming and erase voltage			Programming ROM	3.0	—	3.6	V
			Data flash area*[3]				
Read voltage			Programming ROM	3.0	—	3.6	V
			Data flash area*[3]				
Access state			Programming ROM	1	—	—	State
			Data flash area*[3]	2	—	—	
Programming and erase temperature		-	Programming ROM	0	—	75	°C
			Data flash area*[3]	0	—	75	

Notes: 1. In the system where multiple programming are executed, erase once so as to effectively diminish the programming times after having written with leaving the blank area as least as possible by shifting writing address one by one.

For example, if 16 bytes per 1 set is being programmed, erase once after maximum 256 sets of programming has been done, which diminish the effective programming times.

Keep the information of the times of erasure and set up the limitation times is recommended.

2. If an erase error is occurred, execute the clear status command -> erase command for at least 3 times until no erase error is occurred.

*1. Determination of the number of times the programming /erase operation.

Number of times the programming / erase performed in each block.

When the number of times the programming / erase is n times (n = 100), data can be erased n times in each block.

For example, if 4 bytes programming is done 1024 times, each at a different address in a 4-kbyte per block, and then the block is erased, number of times the programming / erase can be one time.

However, programming cannot be done multiple times in the block (overwriting is prohibited).

*2. Number of times that ensures all the electrical characteristics

*3 As for the data flush is under planning.

27.3 Timing Charts

27.3.1 Clock Timing

The clock timings are shown below.

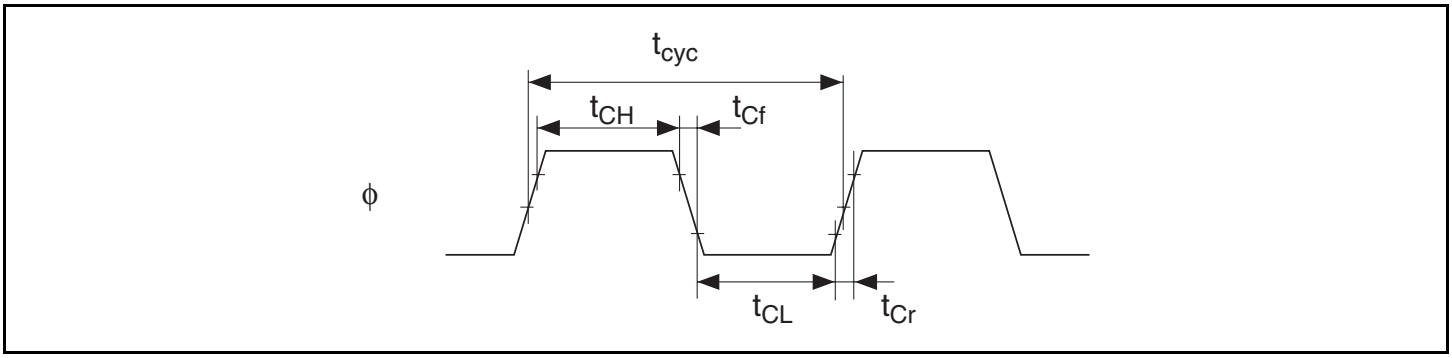


Figure 27.3 System Clock Timing

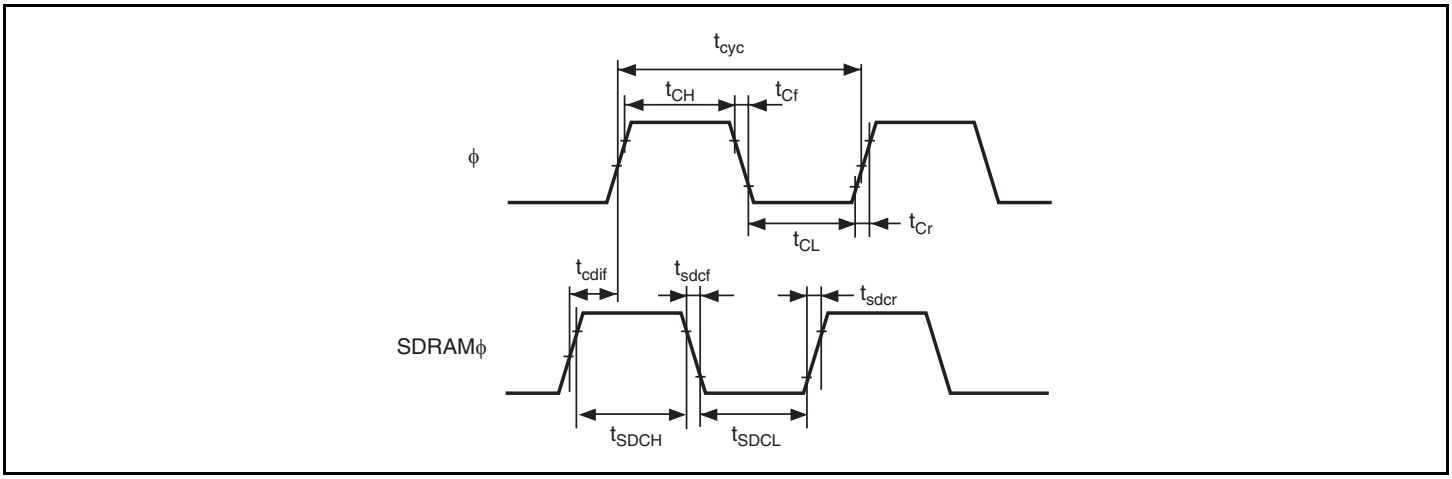


Figure 27.4 SDRAMϕ Timing

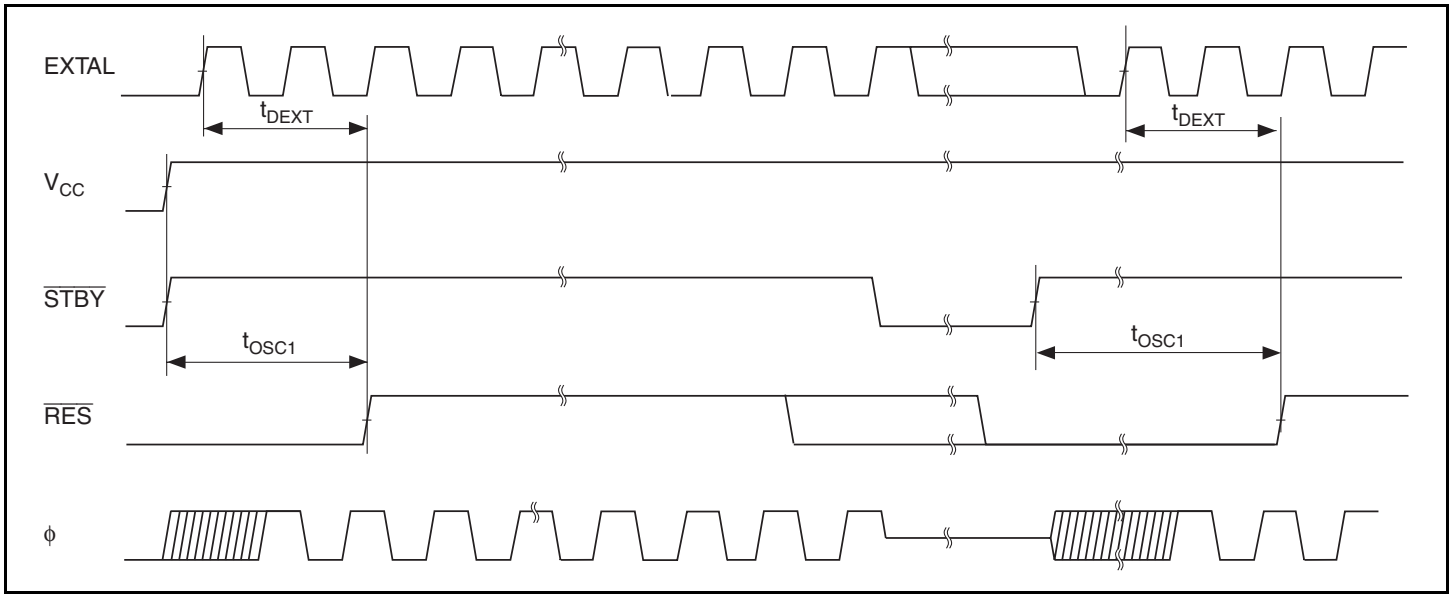


Figure 27.5 (1) Oscillation Settling Timing

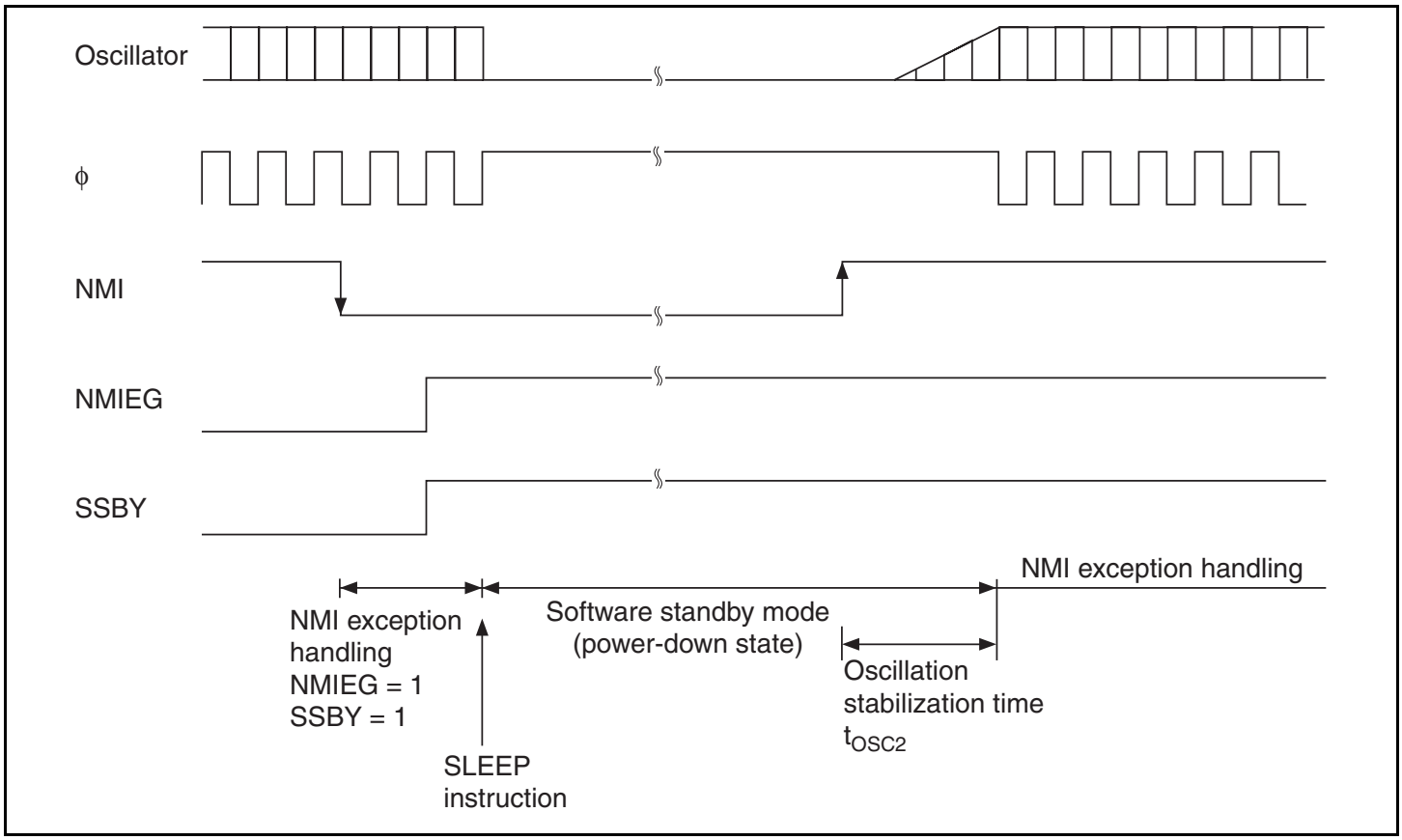


Figure 27.5 (2) Oscillation Settling Timing

27.3.2 Control Signal Timing

The control signal timings are shown below.

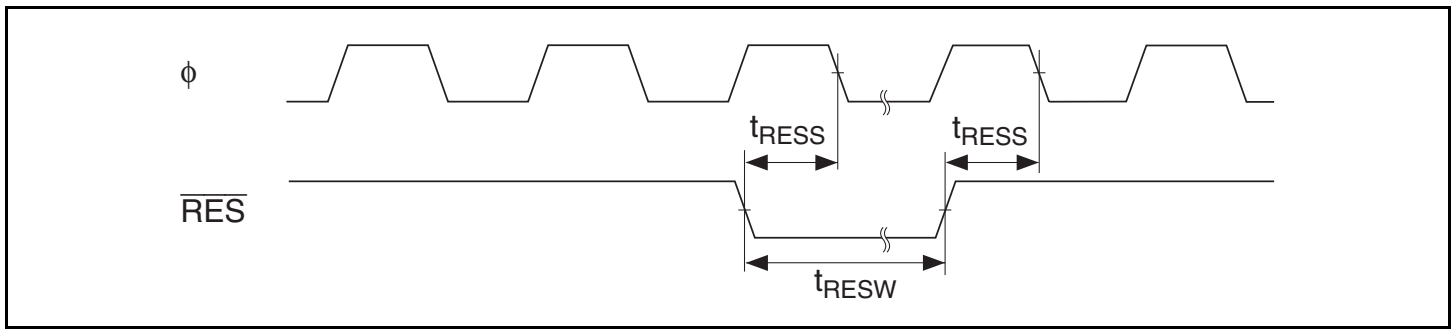


Figure 27.6 Reset Input Timing

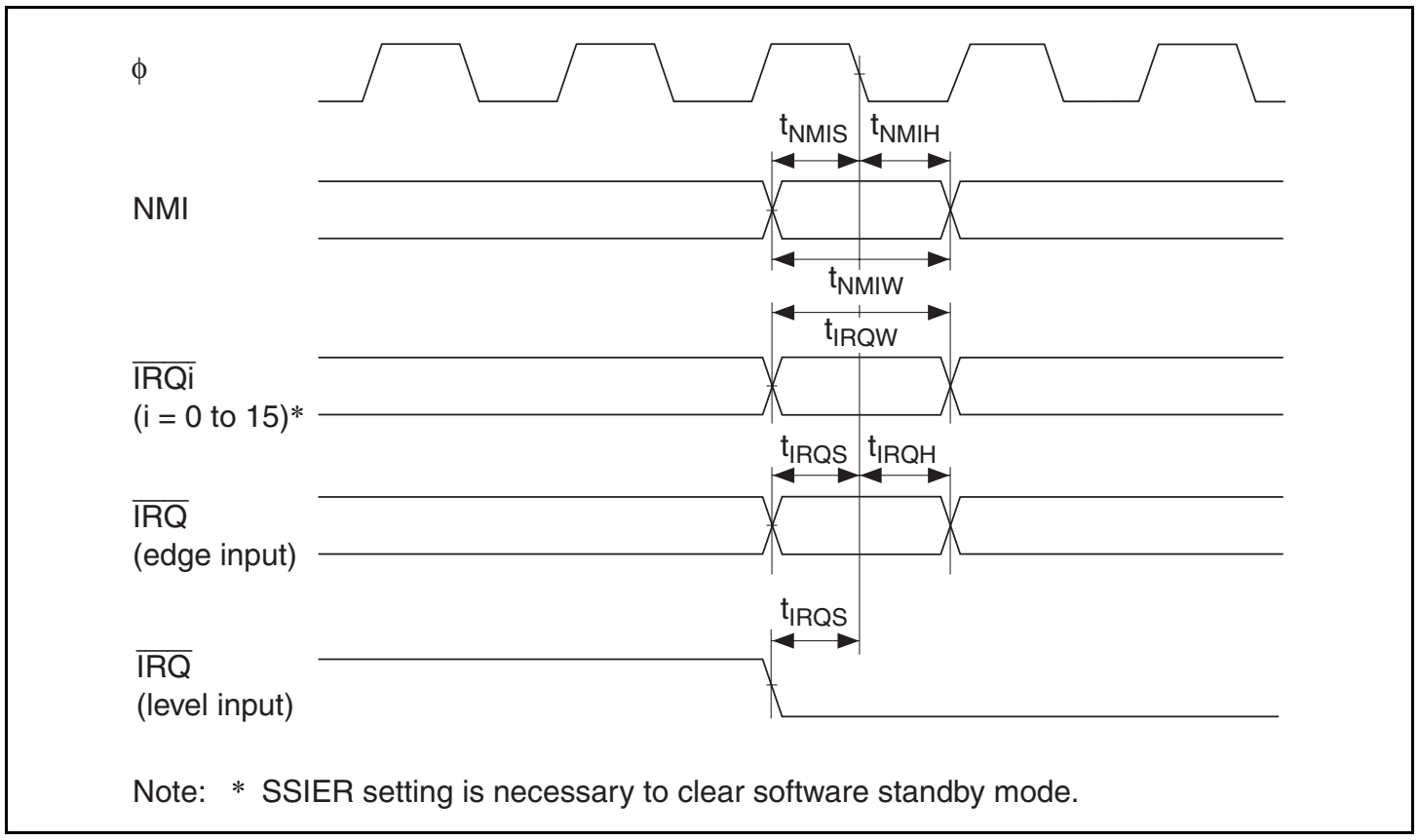


Figure 27.7 Interrupt Input Timing

27.3.3 Bus Timing

The bus timings are shown below.

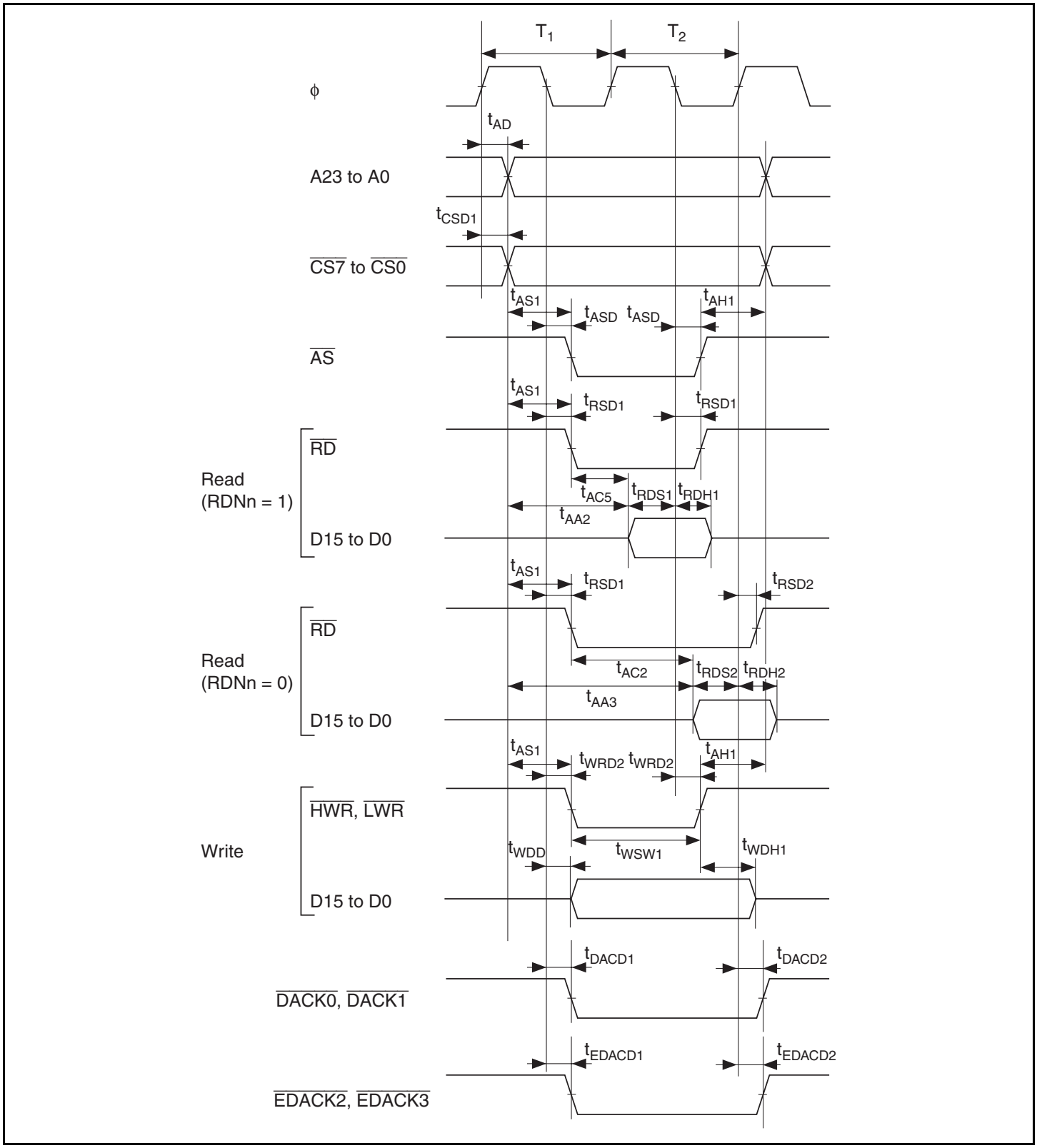


Figure 27.8 Basic Bus Timing: Two-State Access

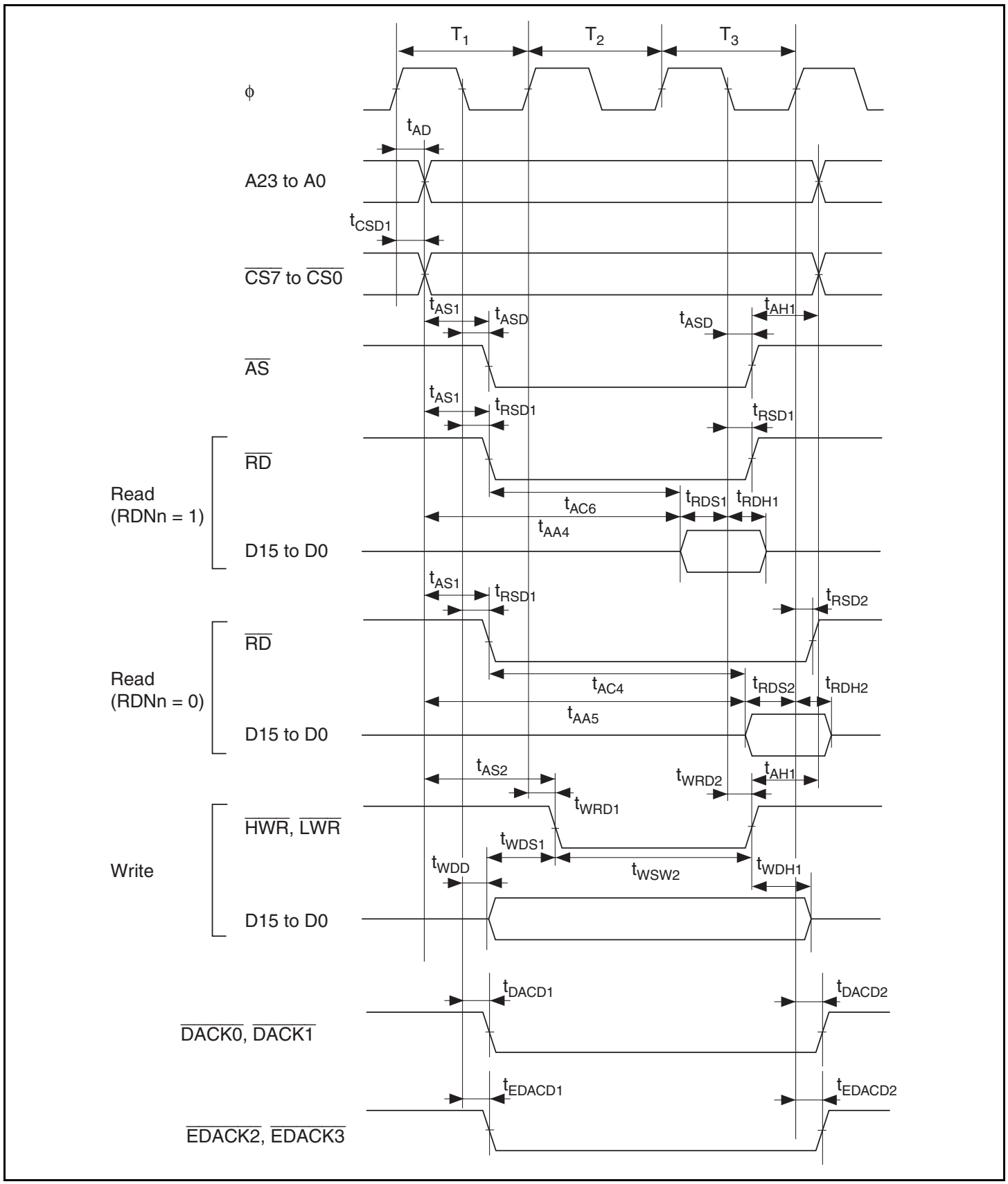


Figure 27.9 Basic Bus Timing: Three-State Access

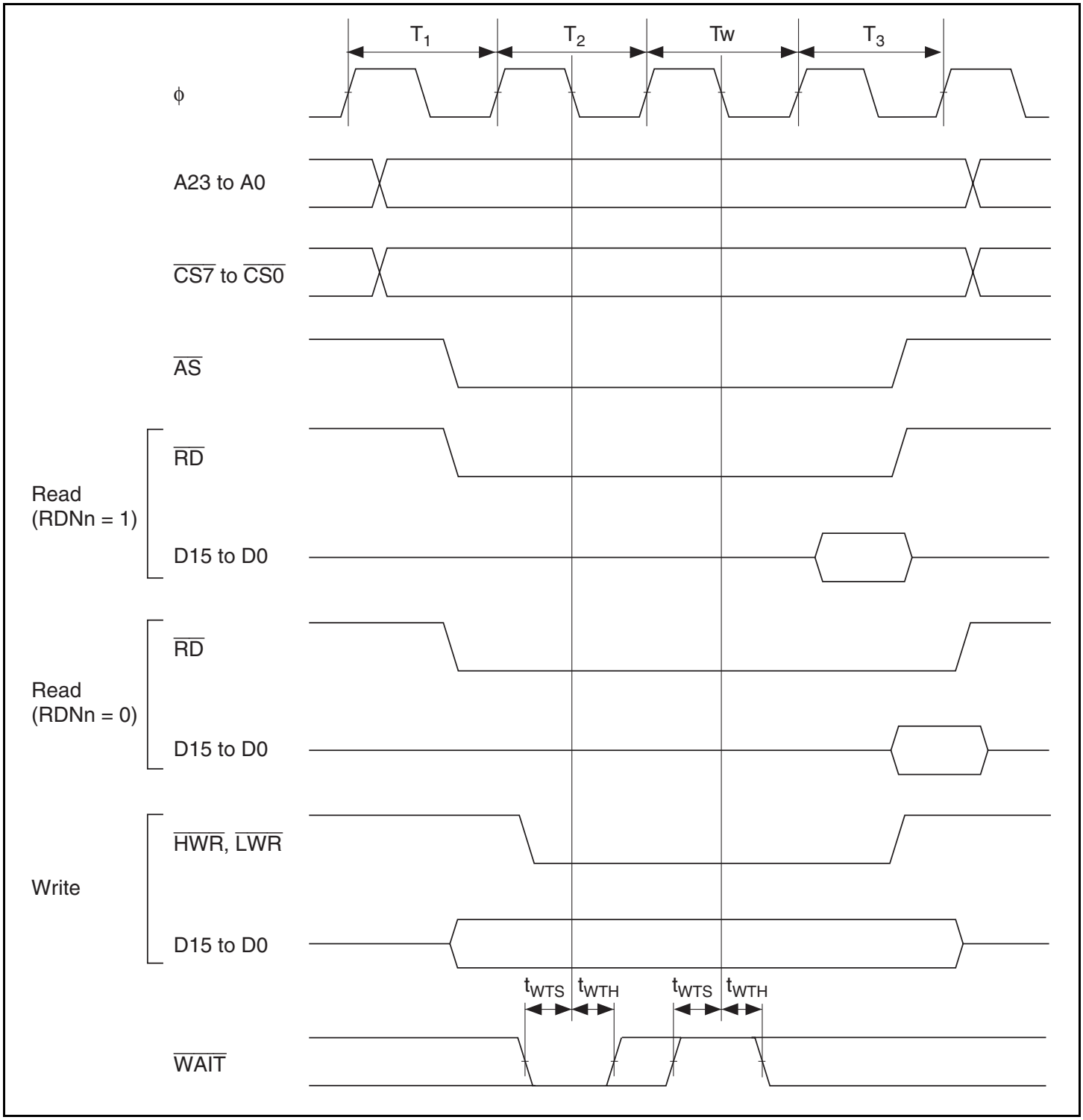


Figure 27.10 Basic Bus Timing: Three-State Access, One Wait

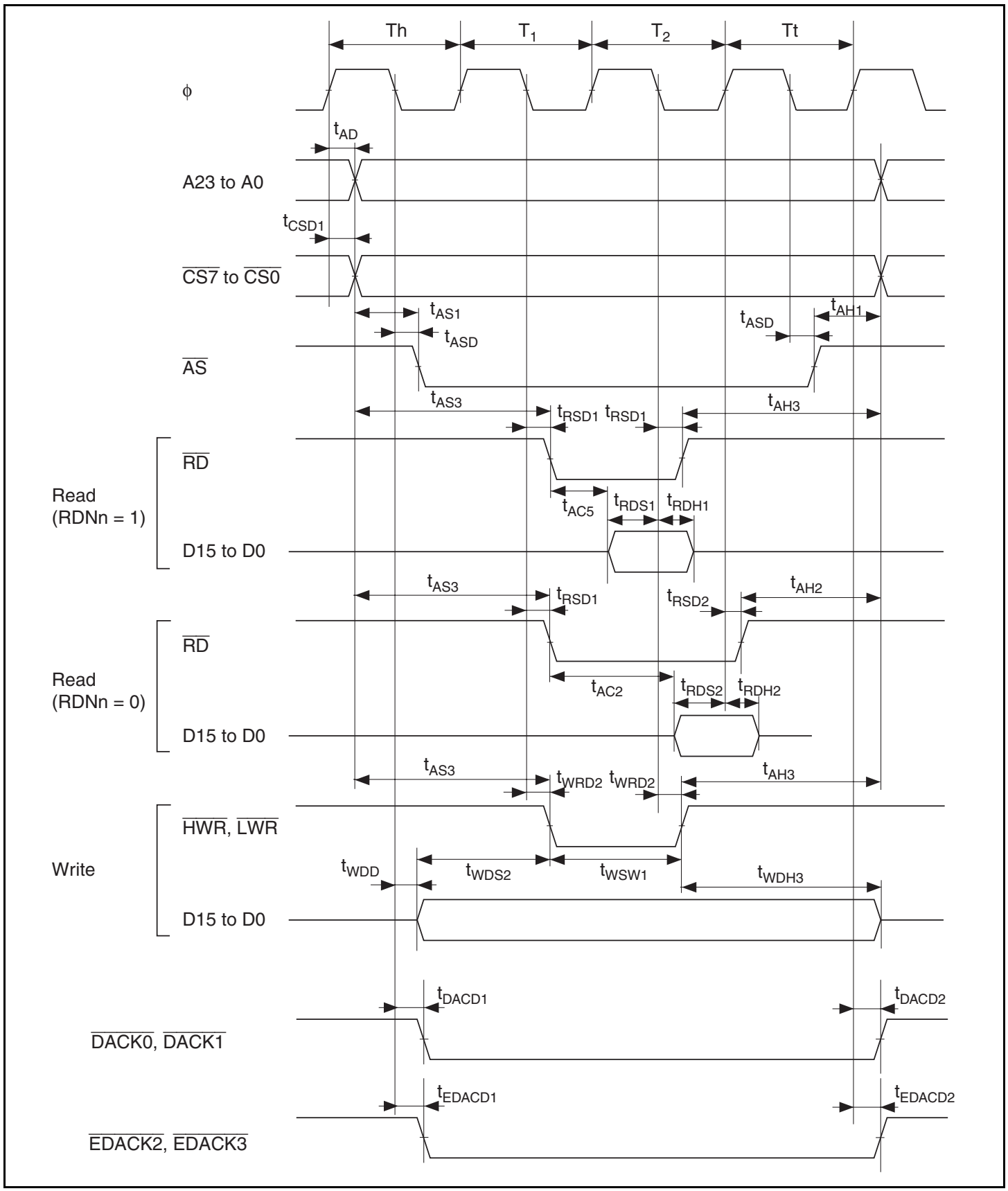


Figure 27.11 Basic Bus Timing: Two-State Access ($\overline{CS}$ Assertion Period Extended)

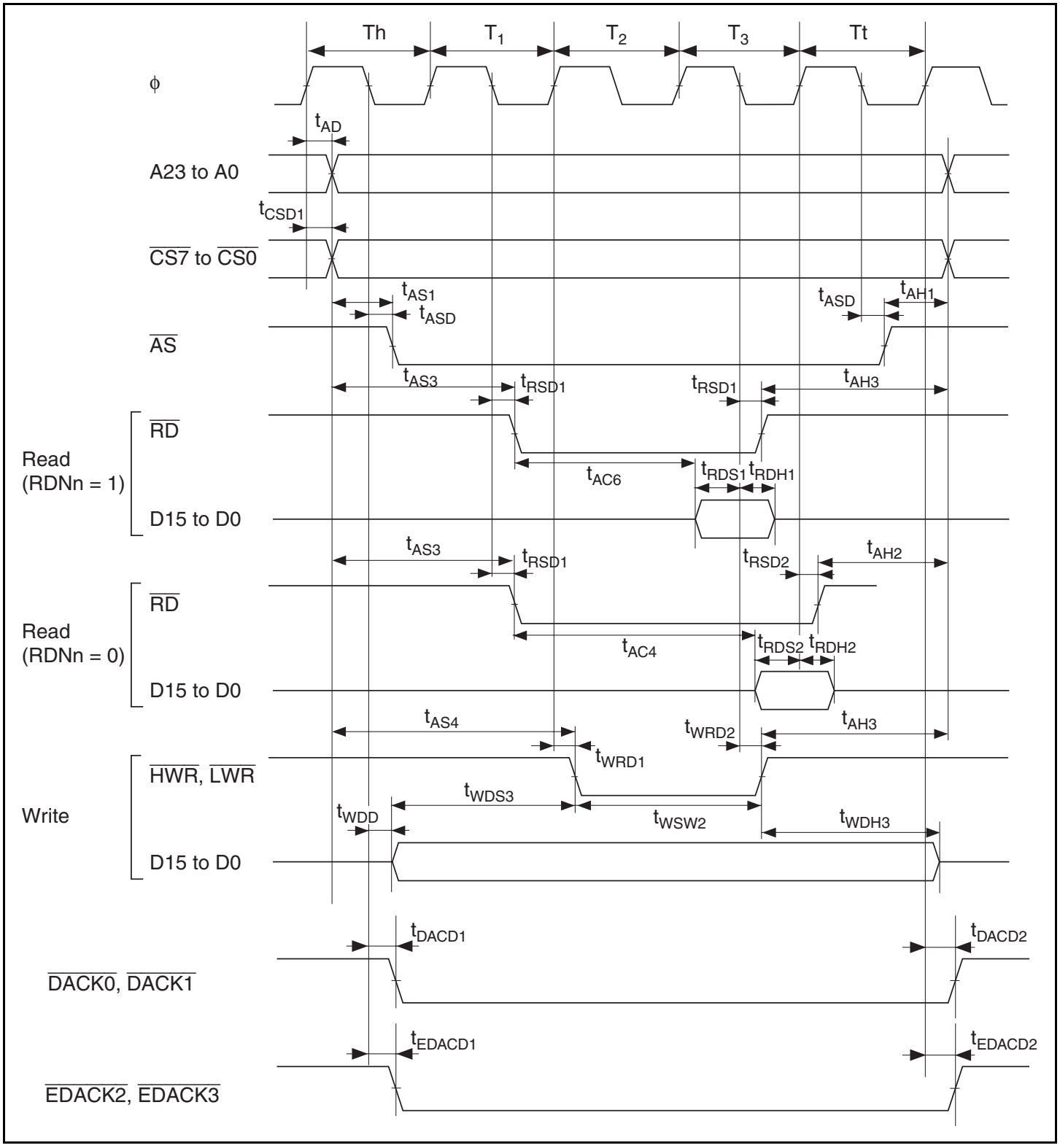


Figure 27.12 Basic Bus Timing: Three-State Access ($\overline{CS}$ Assertion Period Extended)

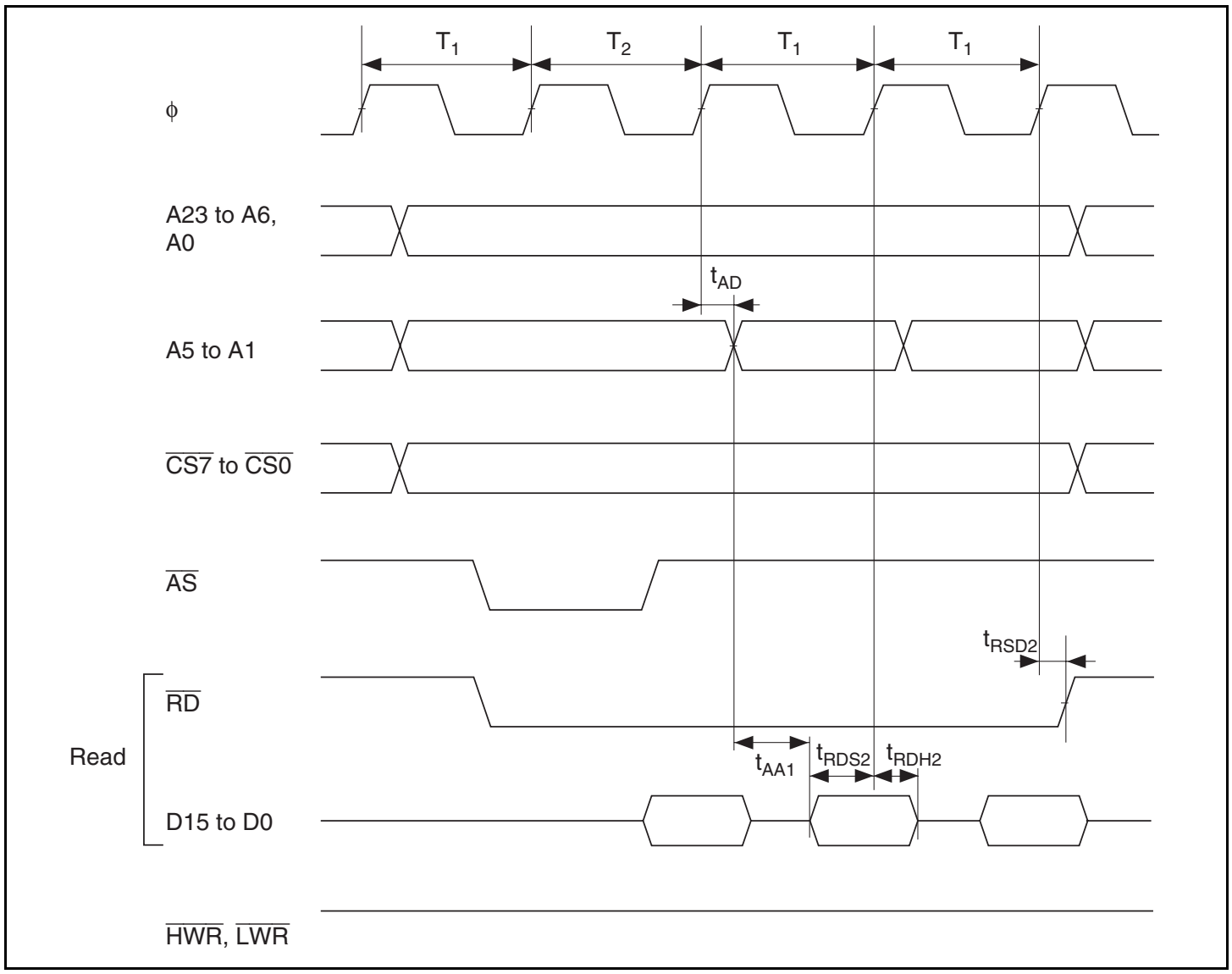


Figure 27.13 Burst ROM Access Timing: One-State Burst Access

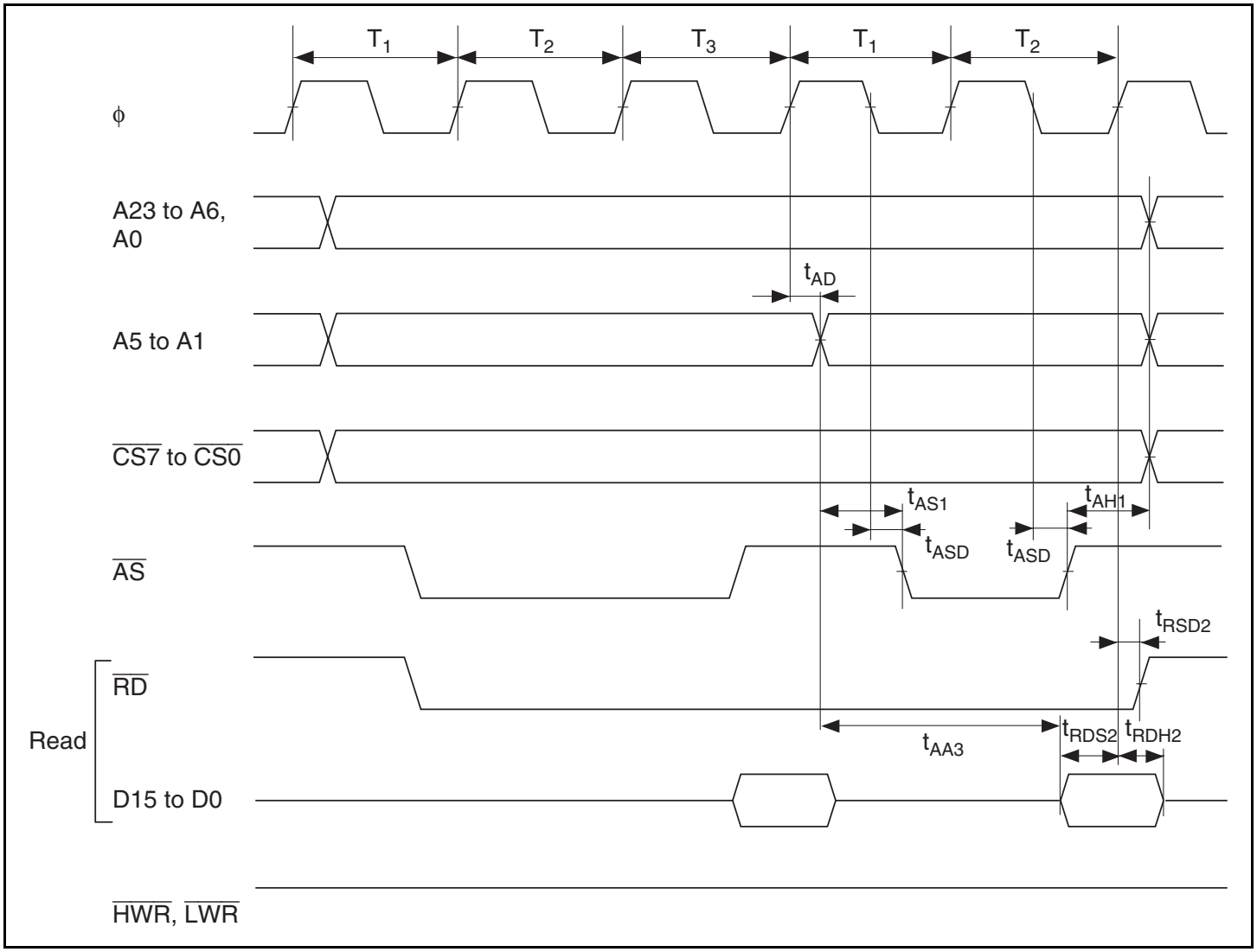
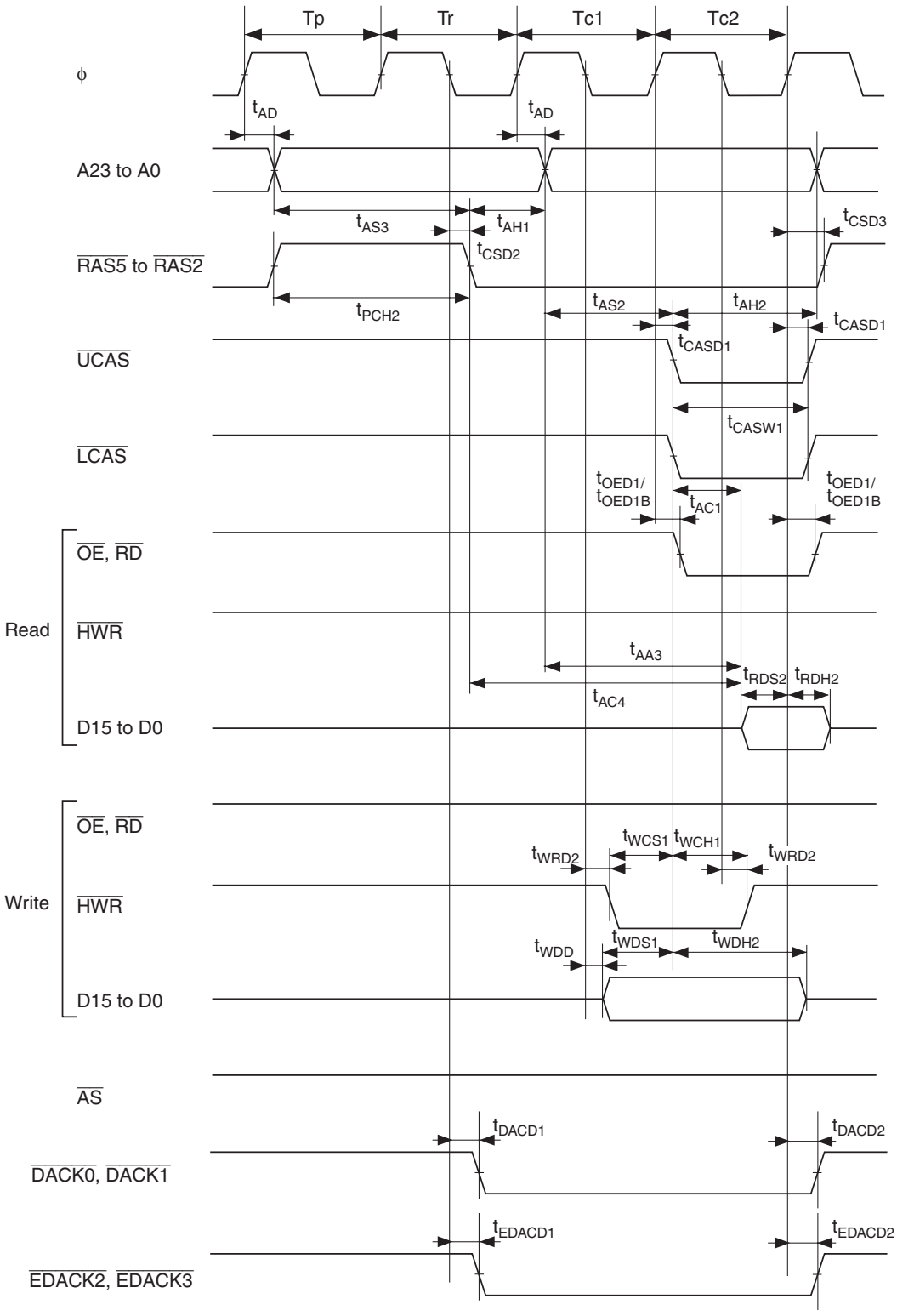
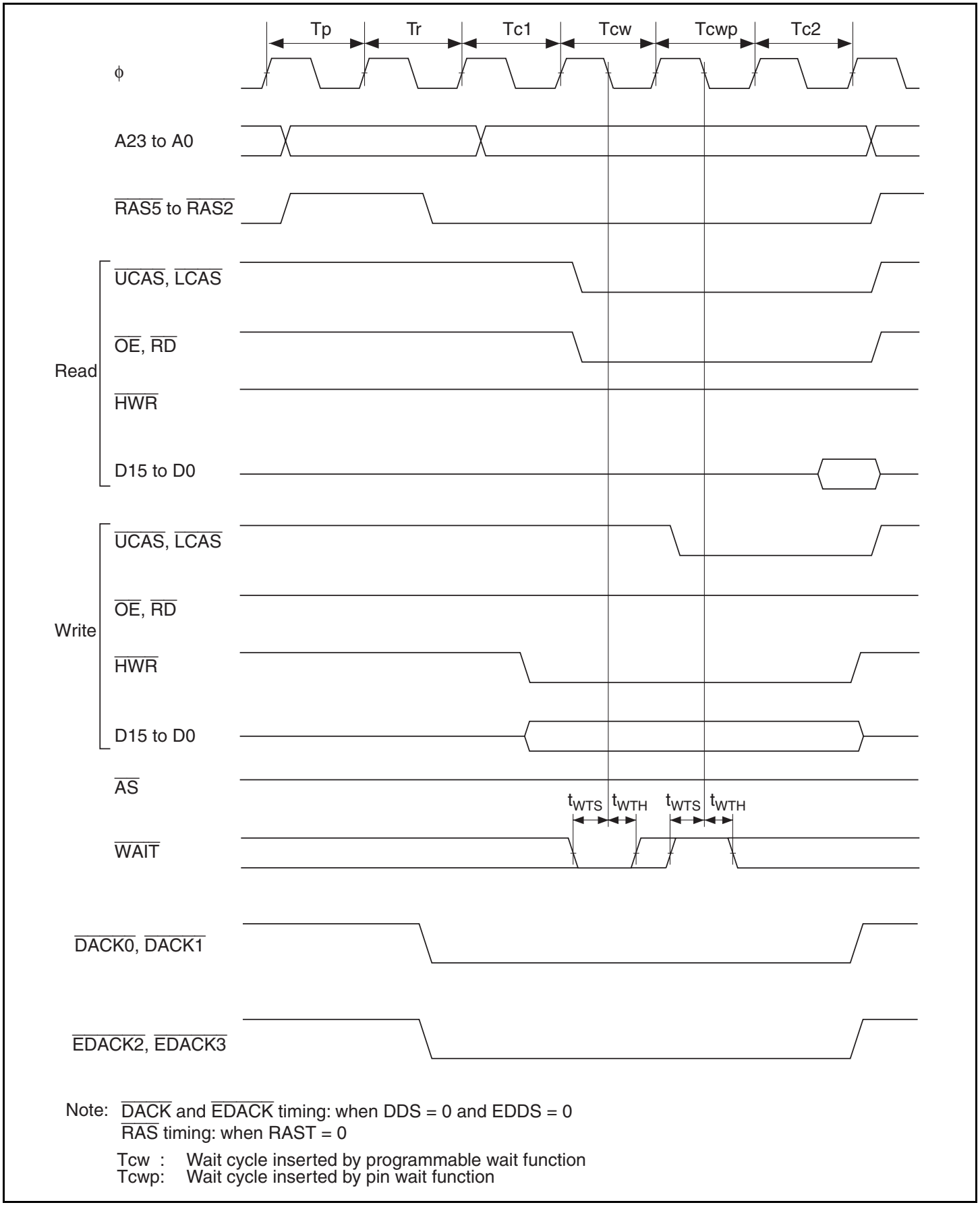


Figure 27.14 Burst ROM Access Timing: Two-State Burst Access



Note: DACK and EDACK timing: when DDS = 0 and EDDS = 0
RAS timing: when RAST = 0

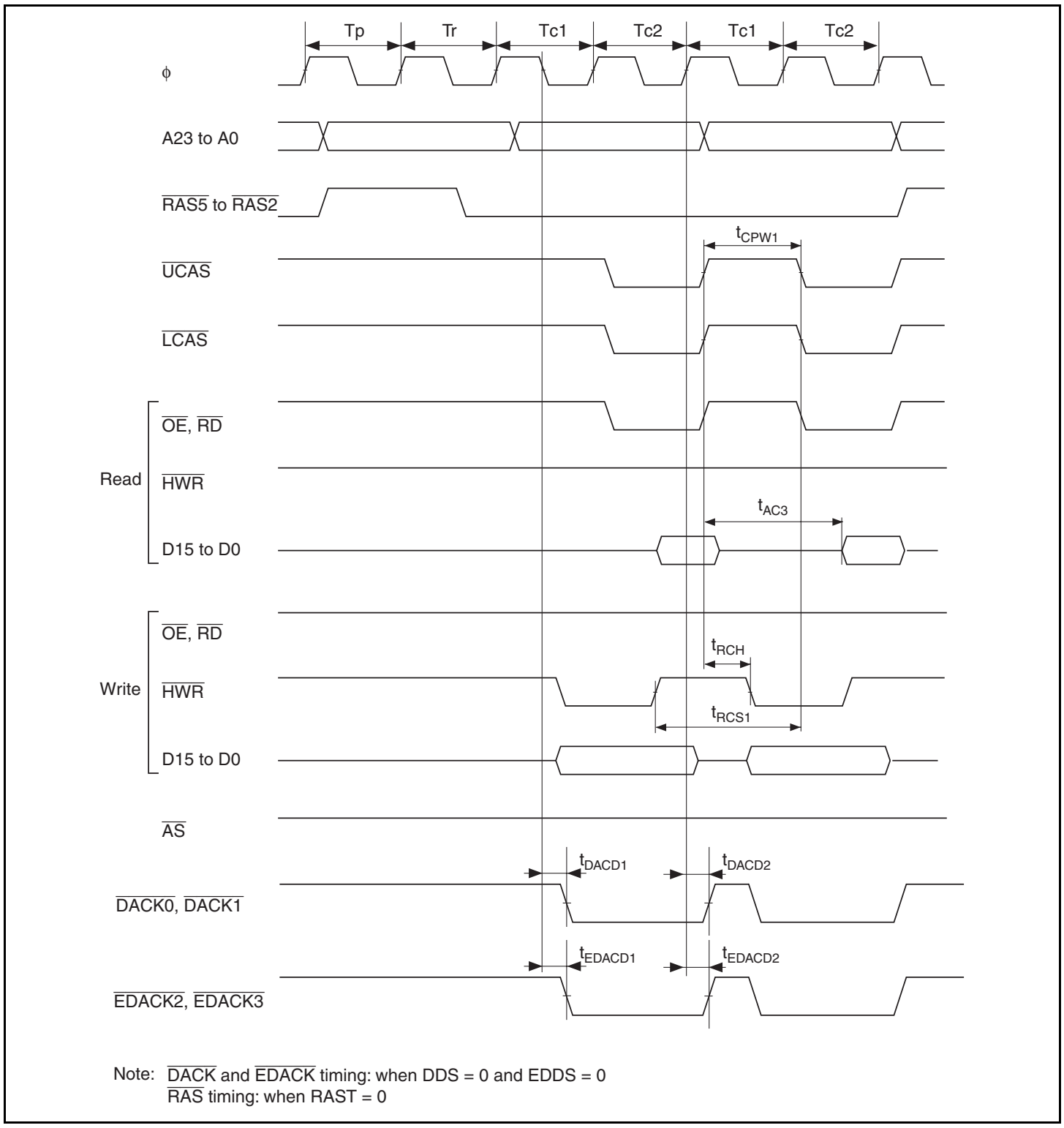
Figure 27.15 DRAM Access Timing: Two-State Access



Note: DACK and EDACK timing: when DDS = 0 and EDDS = 0
RAS timing: when RAST = 0

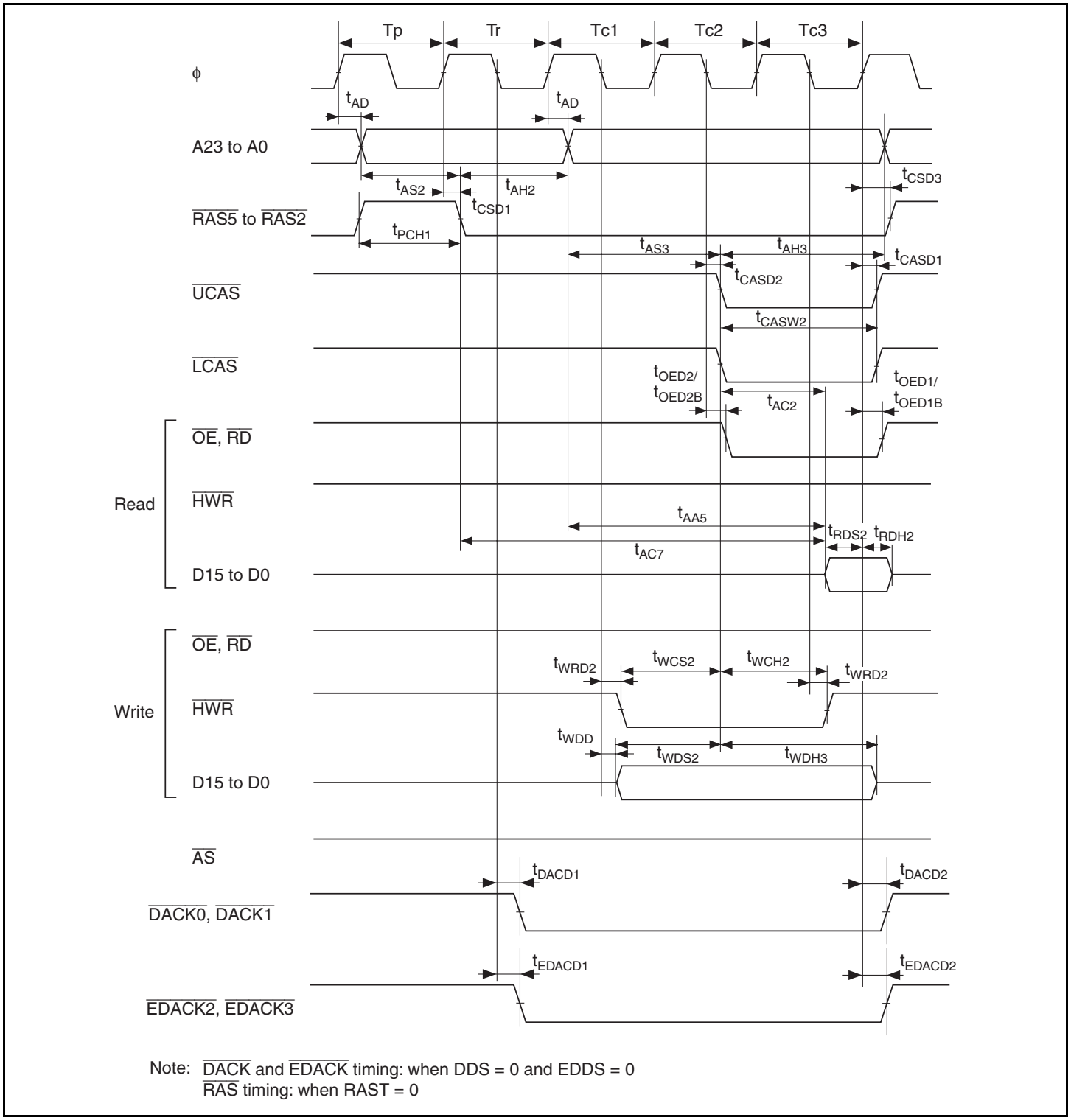
Tcw : Wait cycle inserted by programmable wait function
Tcwp: Wait cycle inserted by pin wait function

Figure 27.16 DRAM Access Timing: Two-State Access, One Wait



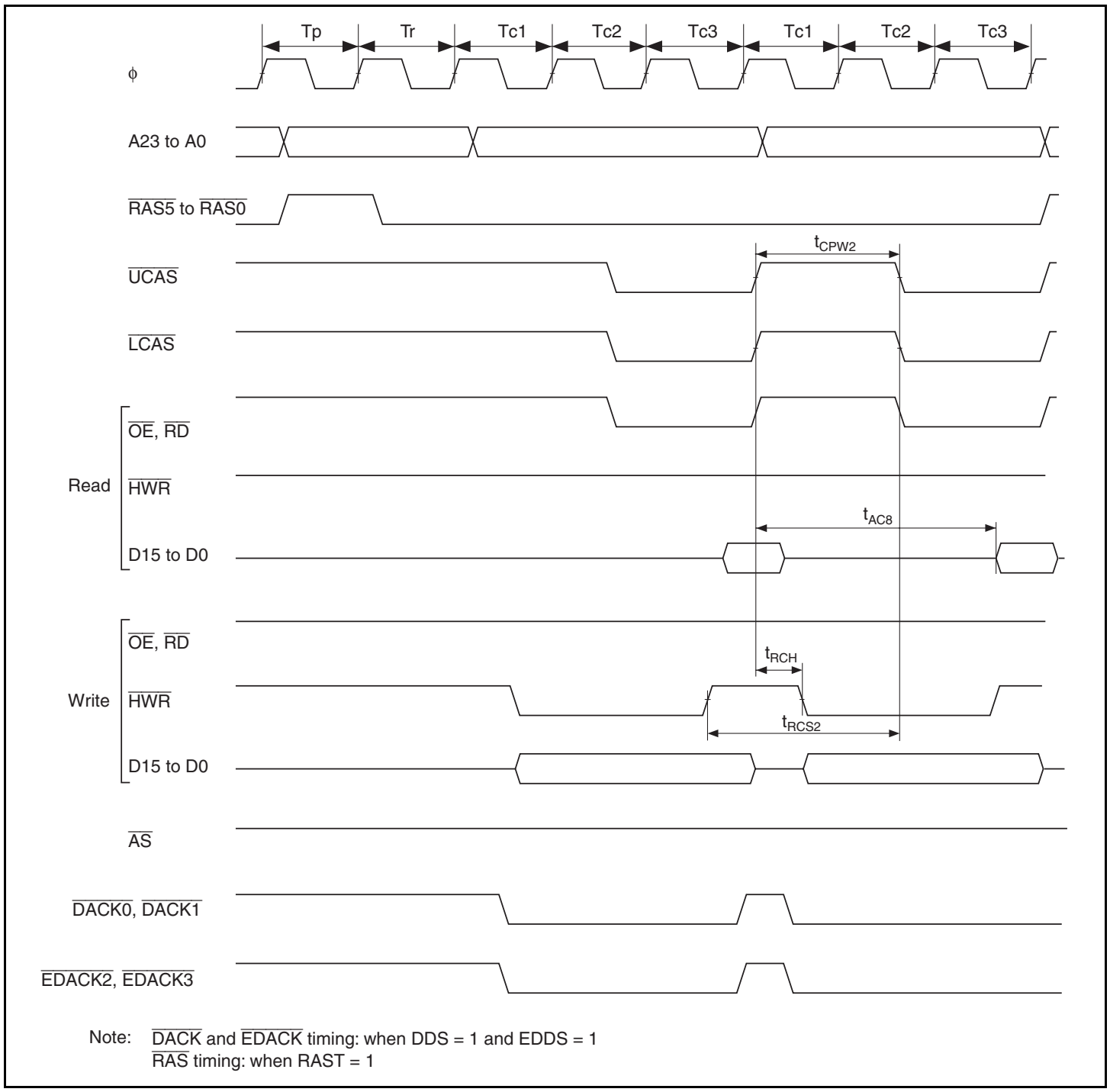
Note: DACK and EDACK timing: when DDS = 0 and EDDS = 0
RAS timing: when RAST = 0

Figure 27.17 DRAM Access Timing: Two-State Burst Access



Note: DACK and EDACK timing: when DDS = 0 and EDDS = 0
RAS timing: when RAST = 0

Figure 27.18 DRAM Access Timing: Three-State Access (RAST = 1)



Note: DACK and EDACK timing: when DDS = 1 and EDDS = 1
RAS timing: when RAST = 1

Figure 27.19 DRAM Access Timing: Three-State Burst Access

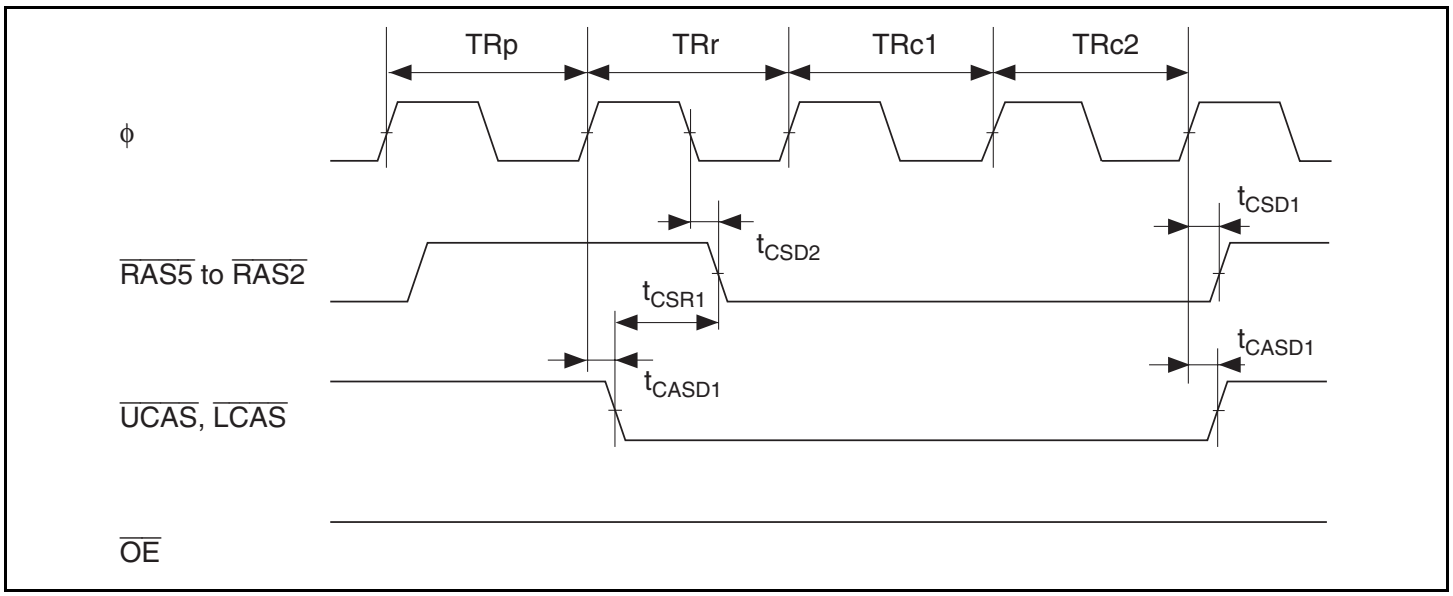


Figure 27.20 CAS-Before-RAS Refresh Timing

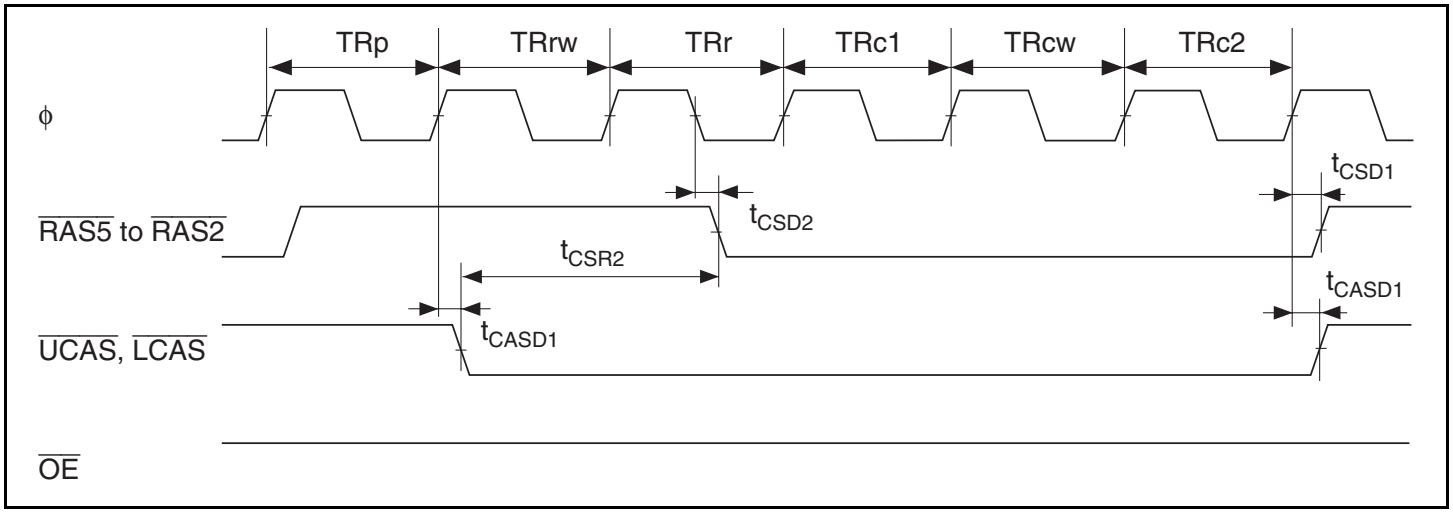


Figure 27.21 CAS-Before-RAS Refresh Timing (with Wait Cycle Insertion)

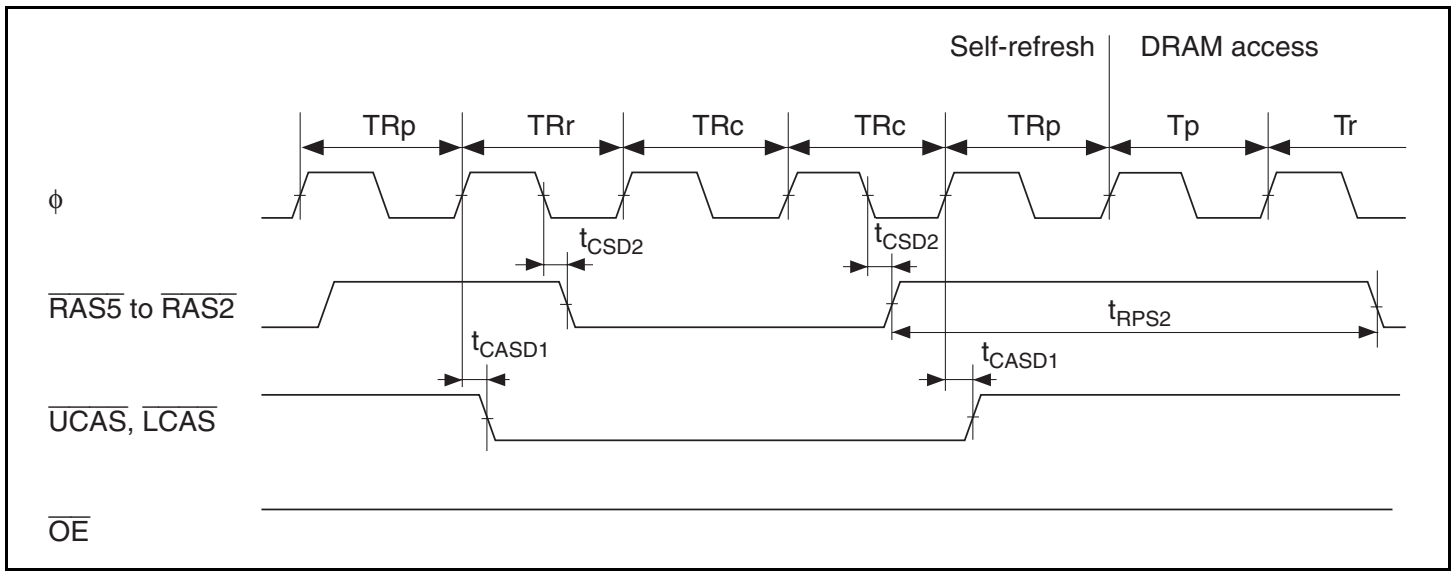


Figure 27.22 Self-Refresh Timing (Return from Software Standby Mode: RAST = 0)

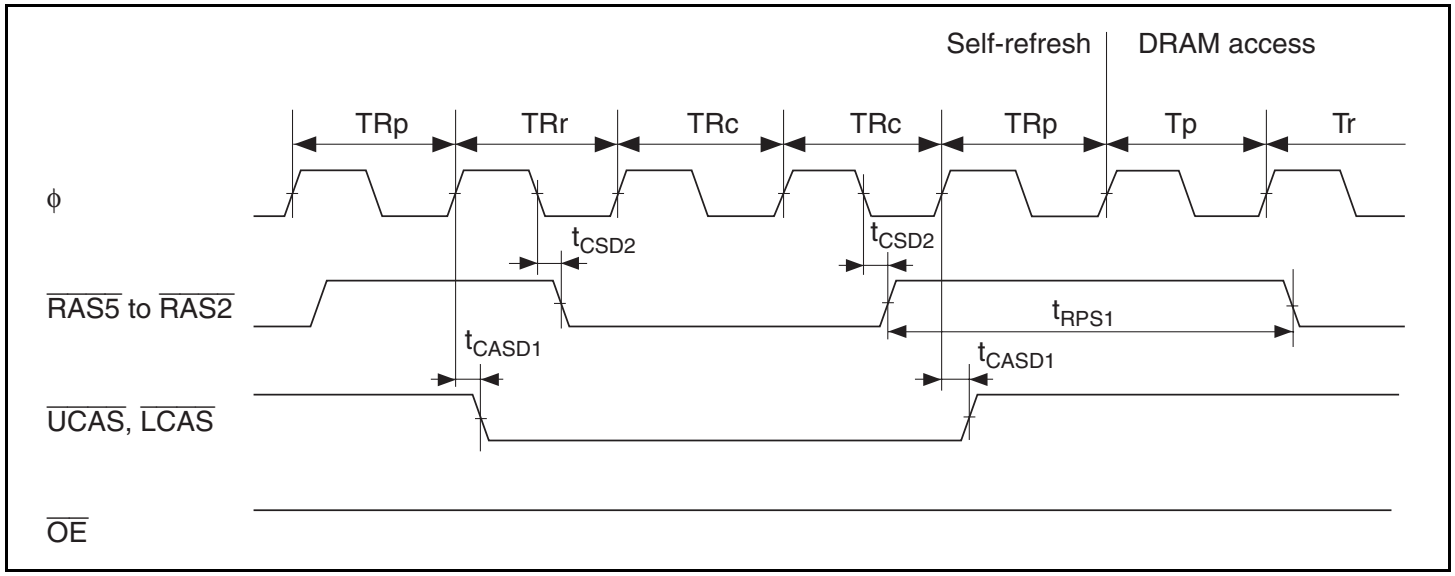


Figure 27.23 Self-Refresh Timing (Return from Software Standby Mode: RAST = 1)

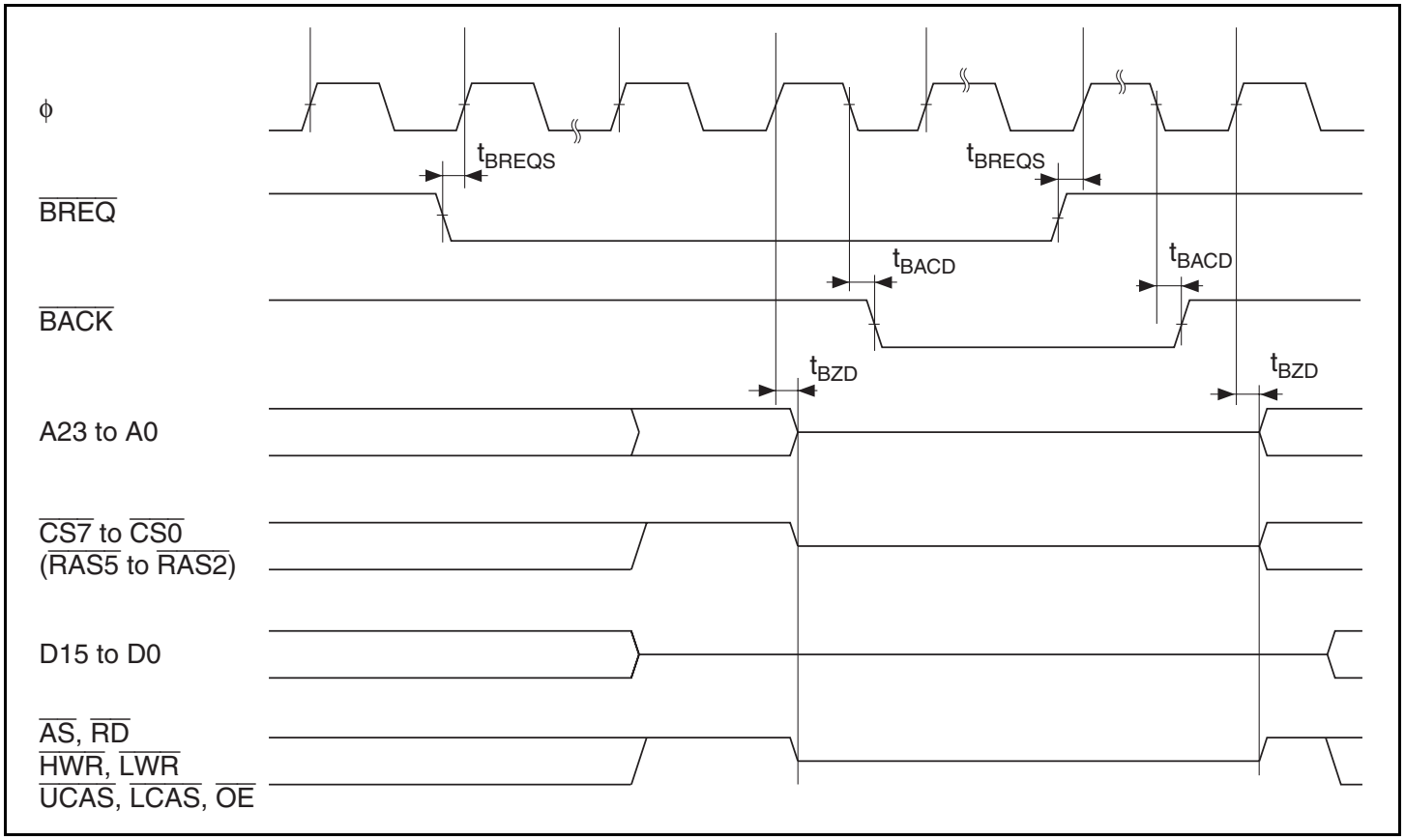


Figure 27.24 External Bus Release Timing

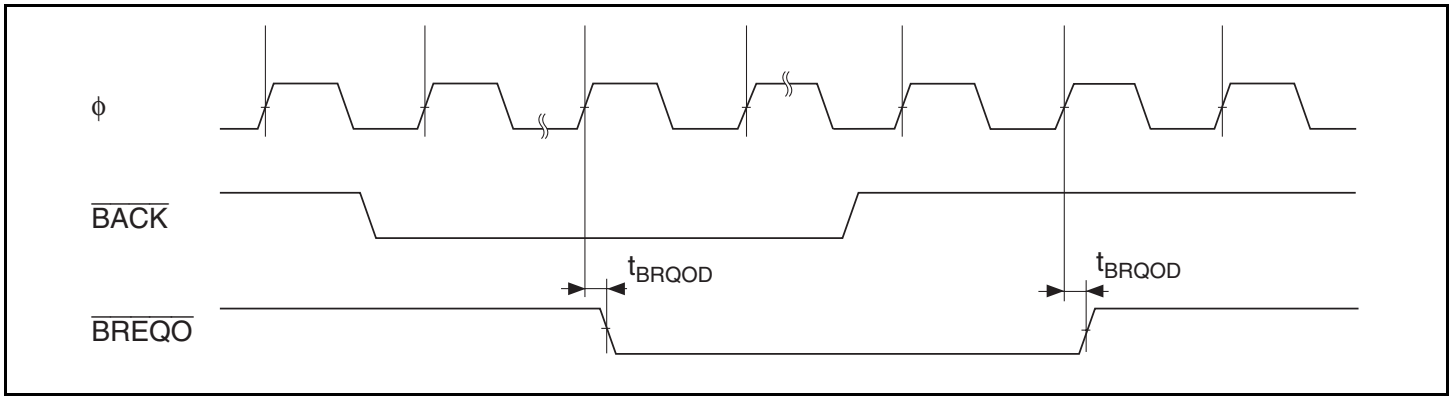


Figure 27.25 External Bus Request Output Timing

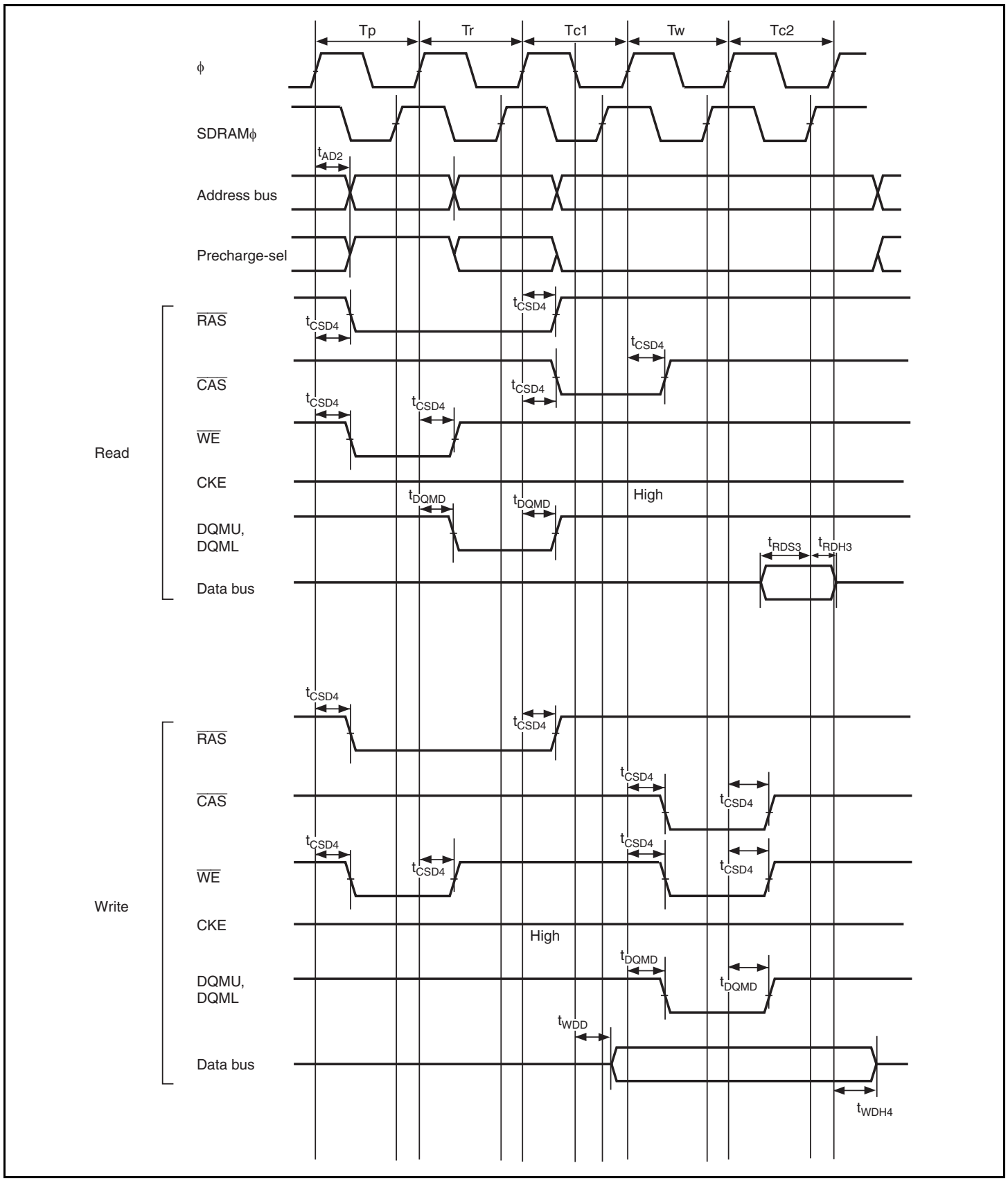


Figure 27.26 Synchronous DRAM Basic Access Timing (CAS Latency 2)

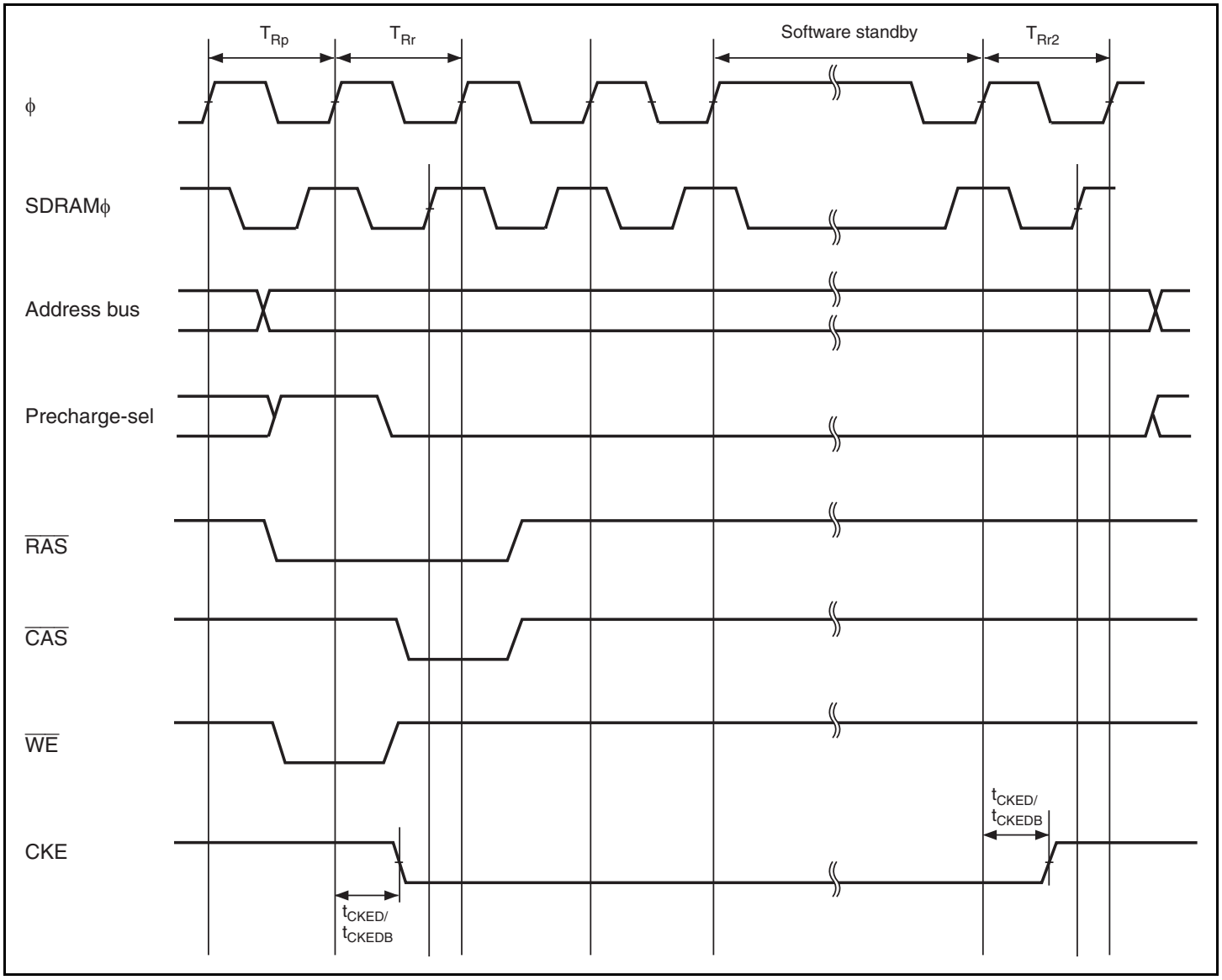


Figure 27.27 Synchronous DRAM Self-Refresh Timing

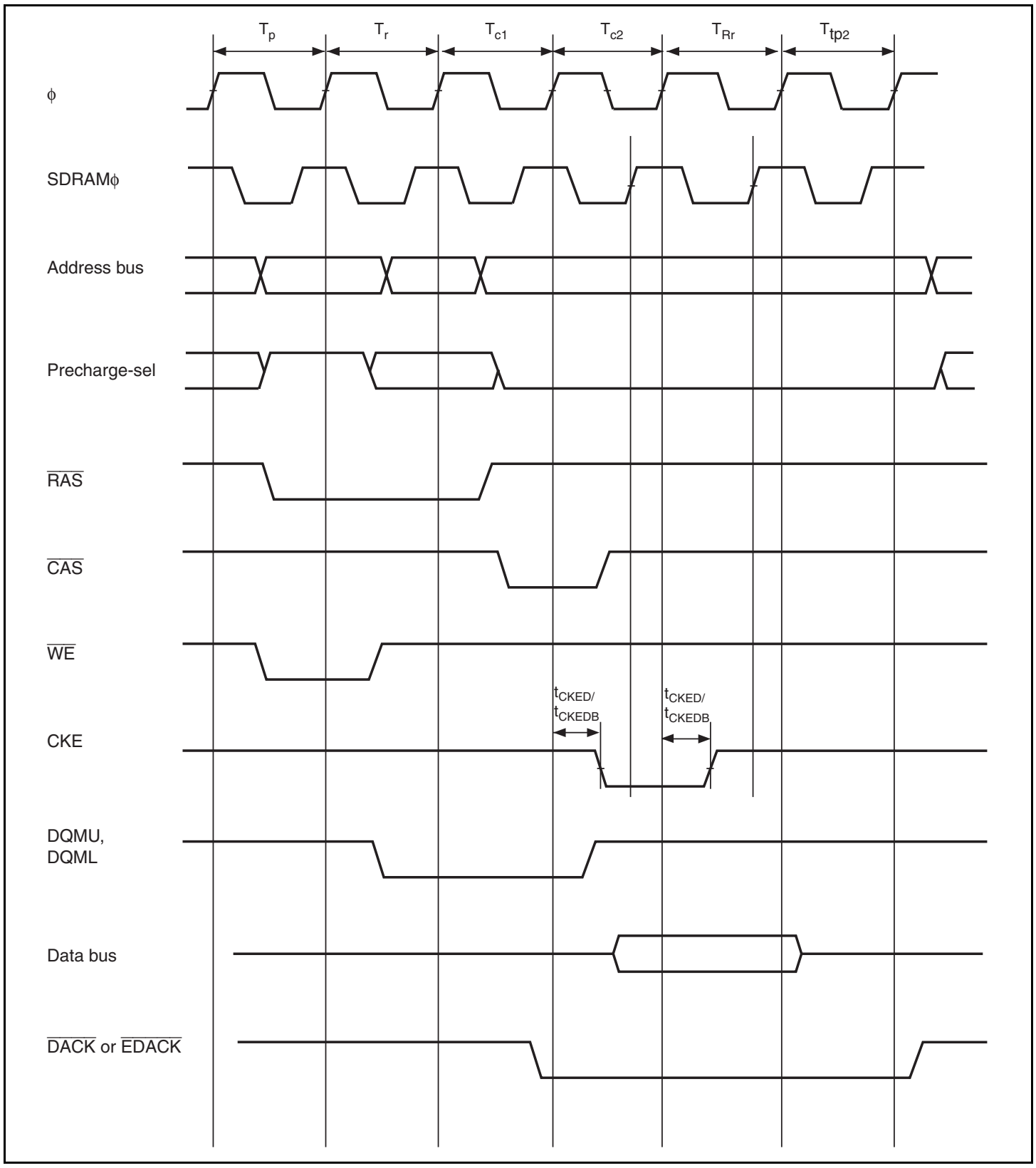


Figure 27.28 Read Data: Two-State Expansion (CAS Latency 2)

27.3.4 DMAC and EXDMAC Timing

The DMAC and EXDMAC timings are shown below.

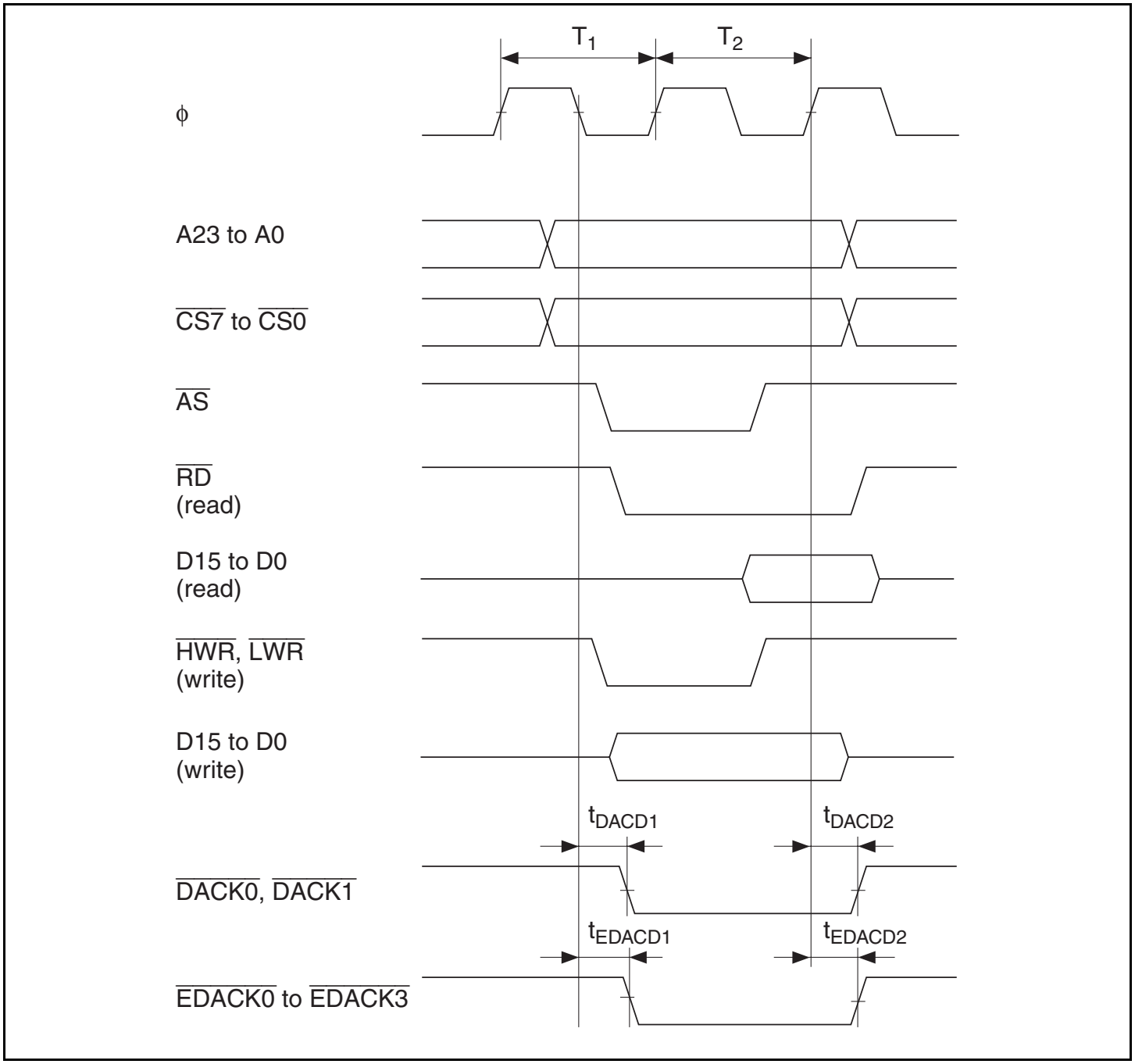


Figure 27.29 DMAC and EXDMAC Single Address Transfer Timing: Two-State Access

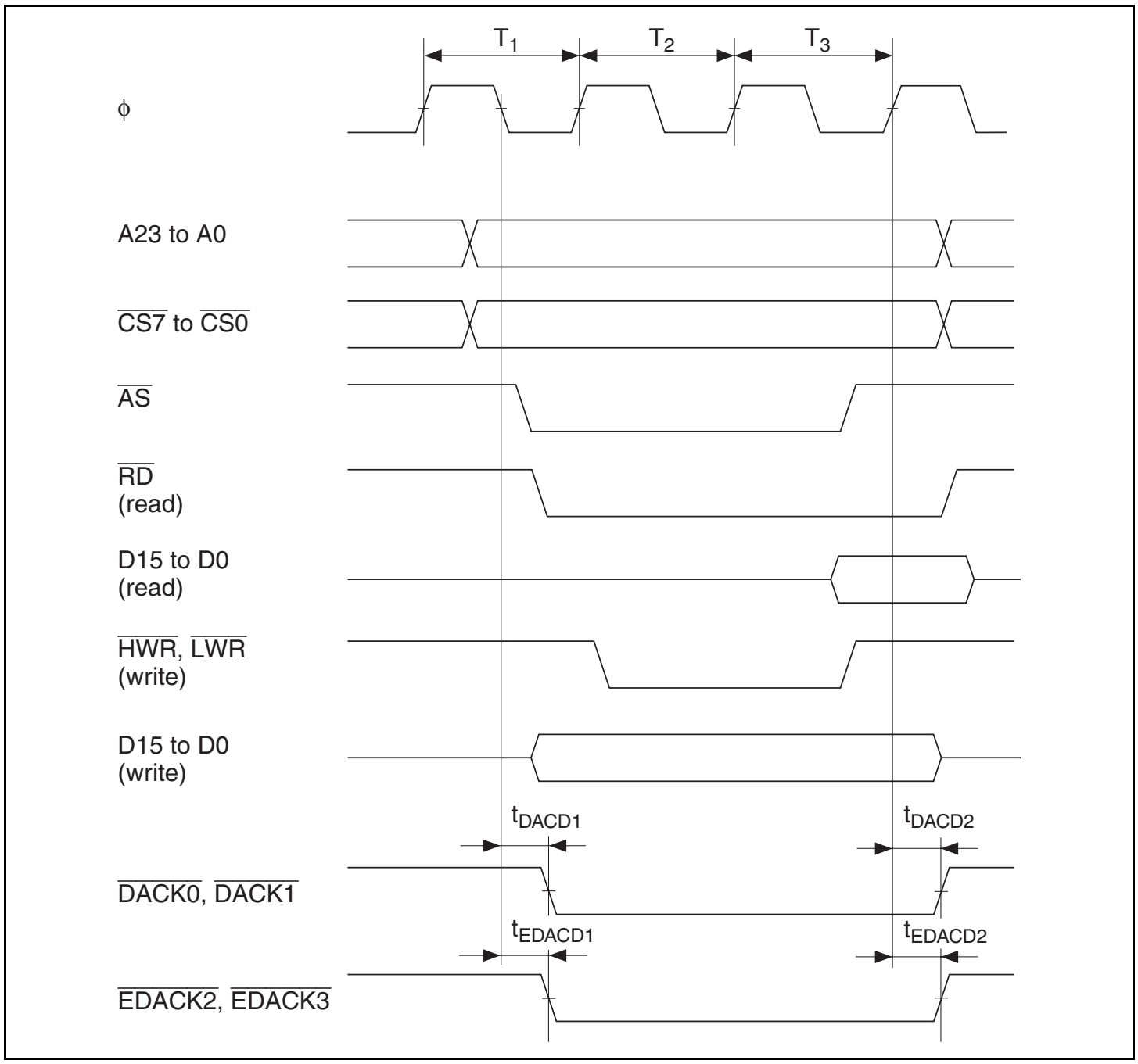


Figure 27.30 DMAC and EXDMAC Single Address Transfer Timing: Three-State Access

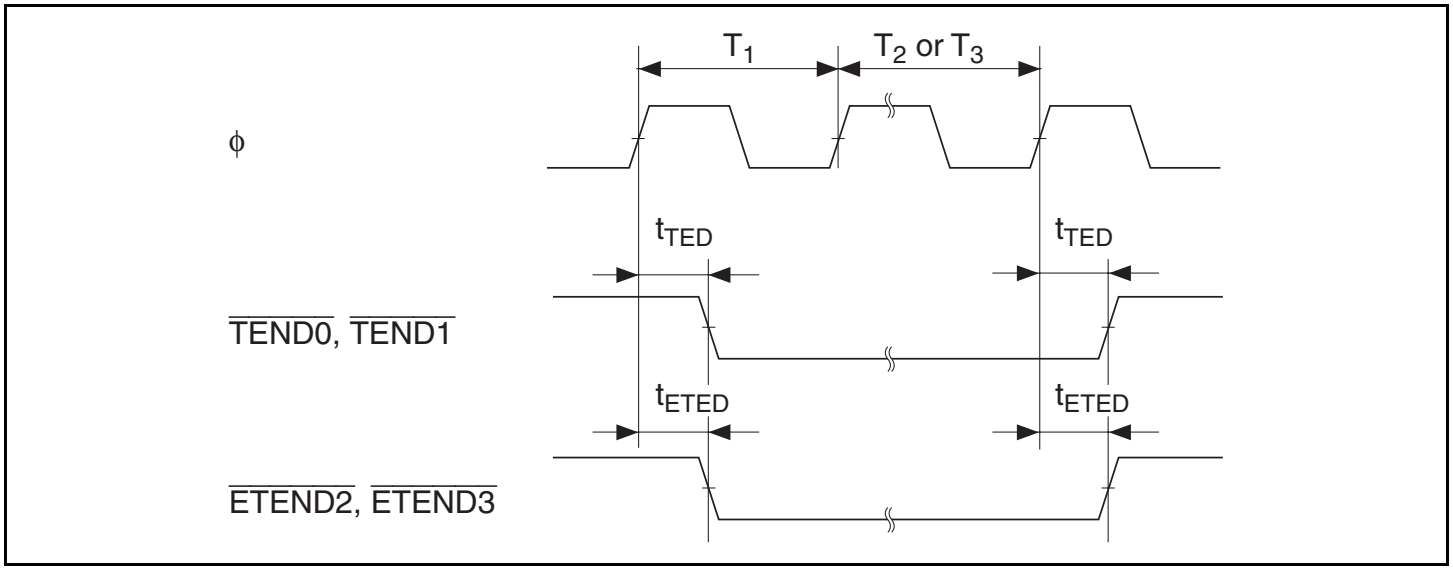


Figure 27.31 DMAC and EXDMAC, $\overline{\text{TEND}}$/$\overline{\text{ETEND}}$ Output Timing

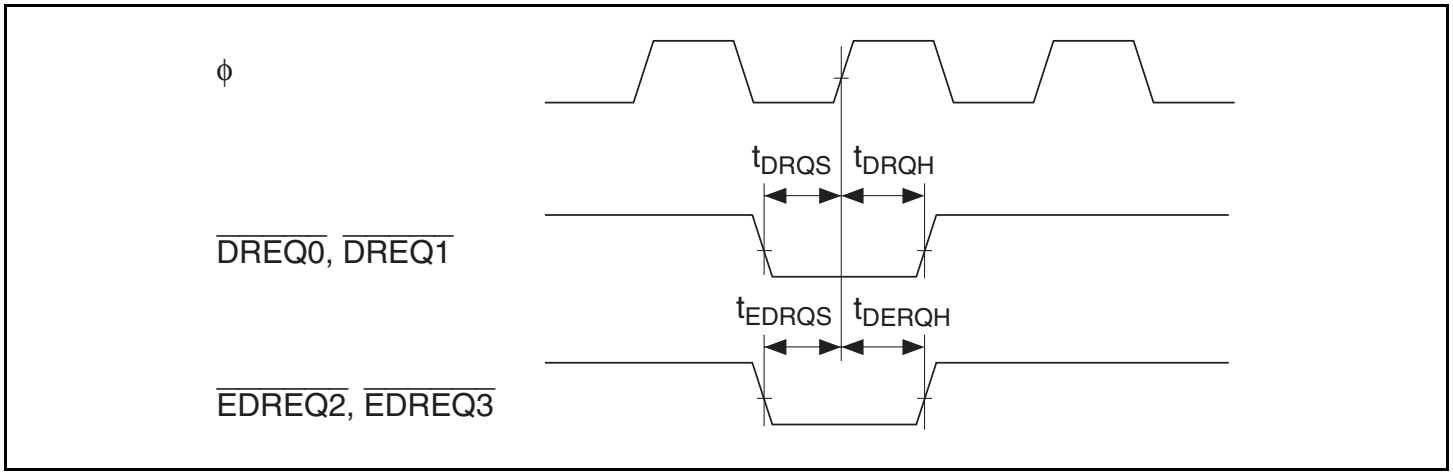


Figure 27.32 DMAC and EXDMAC, $\overline{\text{DREQ}}$/$\overline{\text{EDREQ}}$ Input Timing

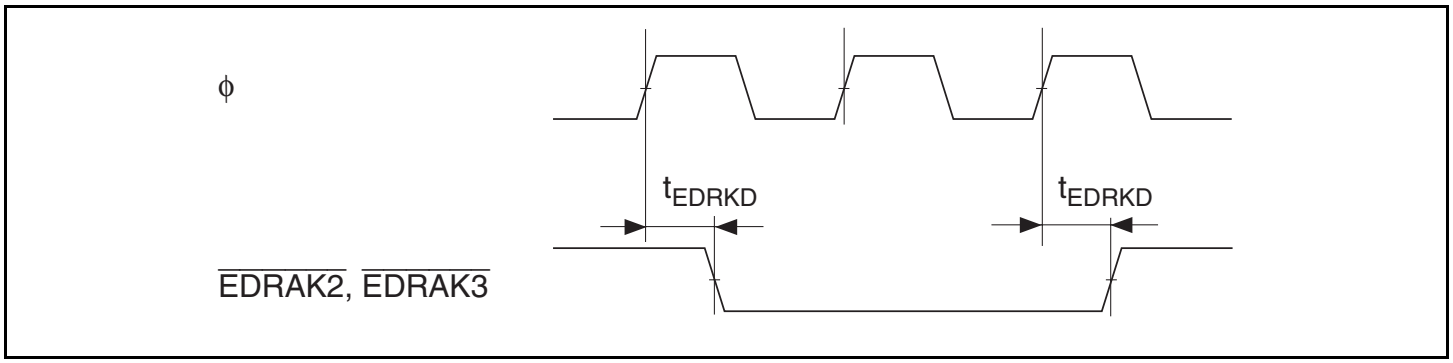


Figure 27.33 EXDMAC, $\overline{\text{EDRAK}}$ Output Timing

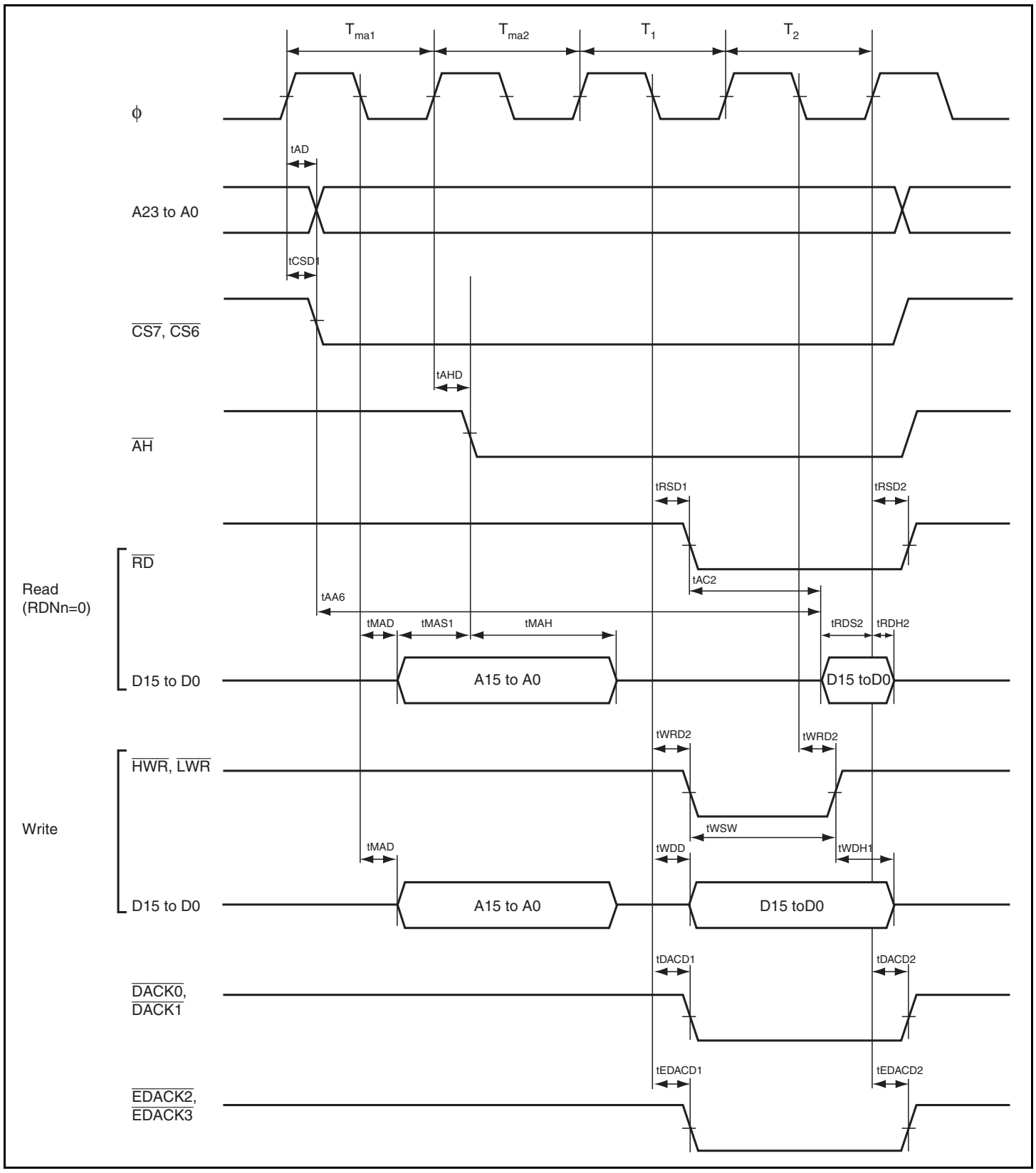


Figure 27.34 Multiplexed Bus Timing: Data Two-State Access

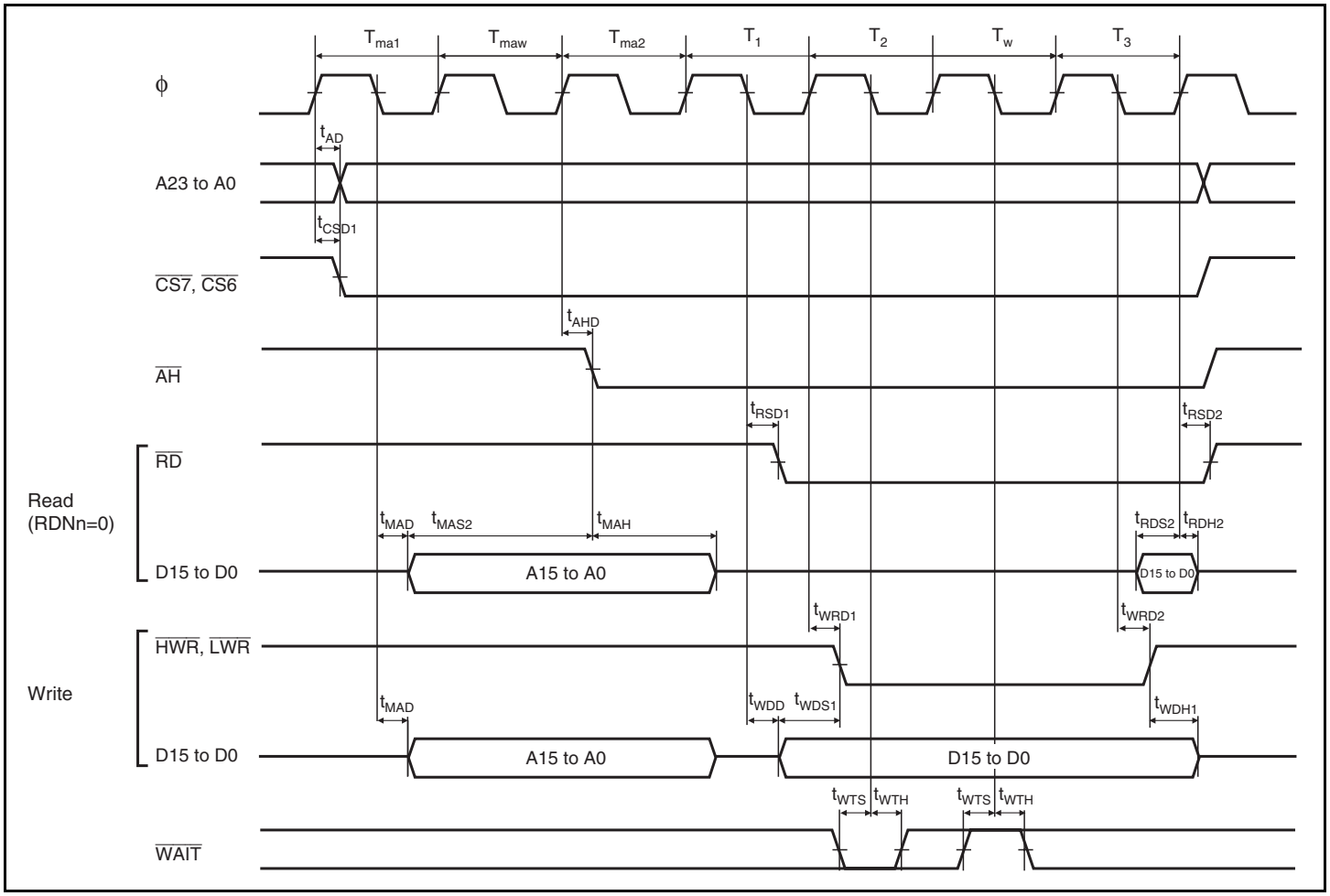


Figure 27.35 Multiplexed Bus Timing: Data Three-State Access, One Wait (with Address Wait: When ADDEX = 1)

27.3.5 USB Characteristics

The following figures describe USB characteristics.

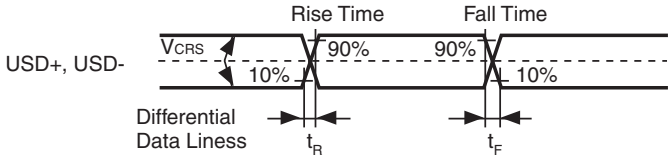


Figure 27.36 Data Signal Timing

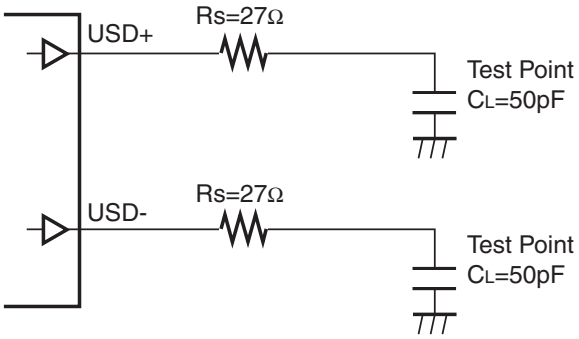


Figure 27.37 Load Conditions

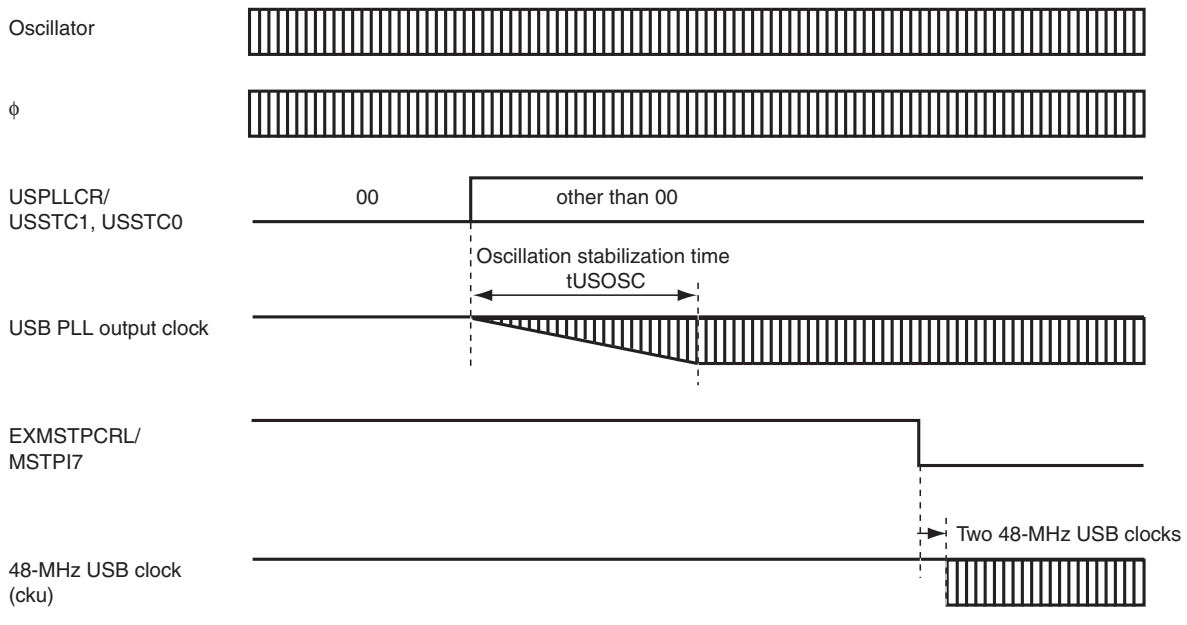


Figure 27.38 Timing of Oscillation Stabilization Time of USB PLL

27.3.6 Timing of On-Chip Peripheral Modules

The on-chip peripheral module timings are shown below.

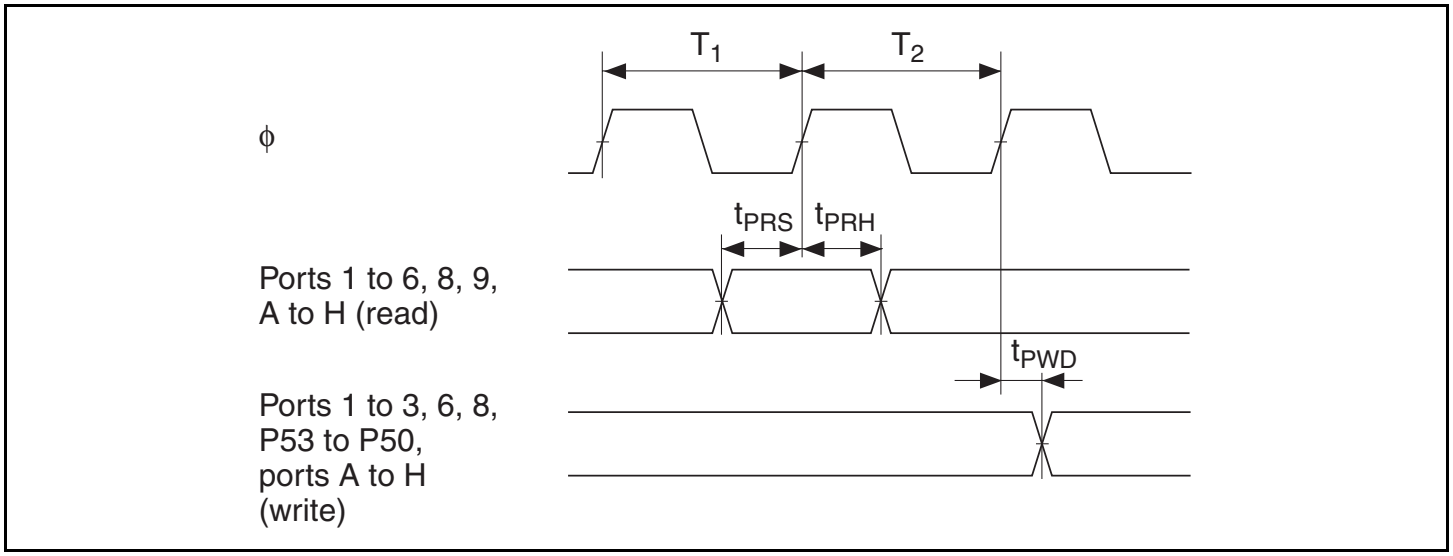


Figure 27.39 I/O Port Input/Output Timing

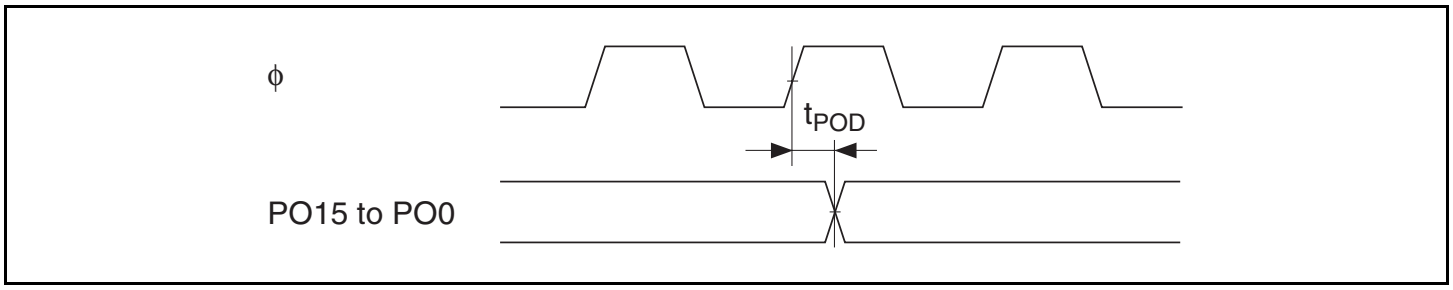
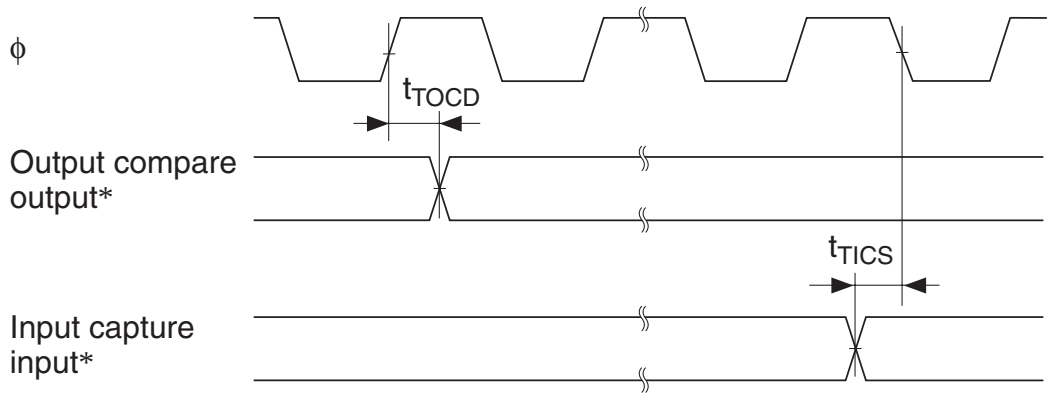


Figure 27.40 PPG Output Timing



Note: * TIOCA0 to TIOCA11, TIOCB0 to TIOCB11, TIOCC0, TIOCC3, TIOCC6, TIOCC9, TIOCD0, TIOCD3, TIOCD6, and TIOCD9

Figure 27.41 TPU Input/Output Timing

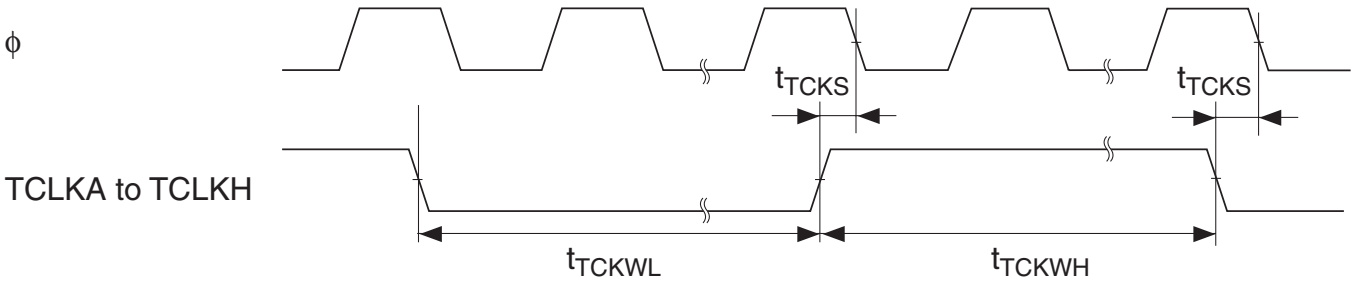


Figure 27.42 TPU Clock Input Timing

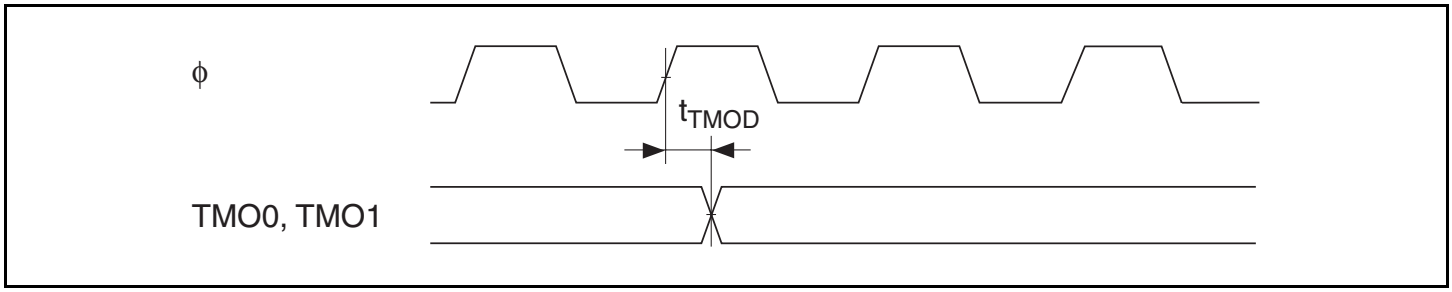


Figure 27.43 8-Bit Timer Output Timing

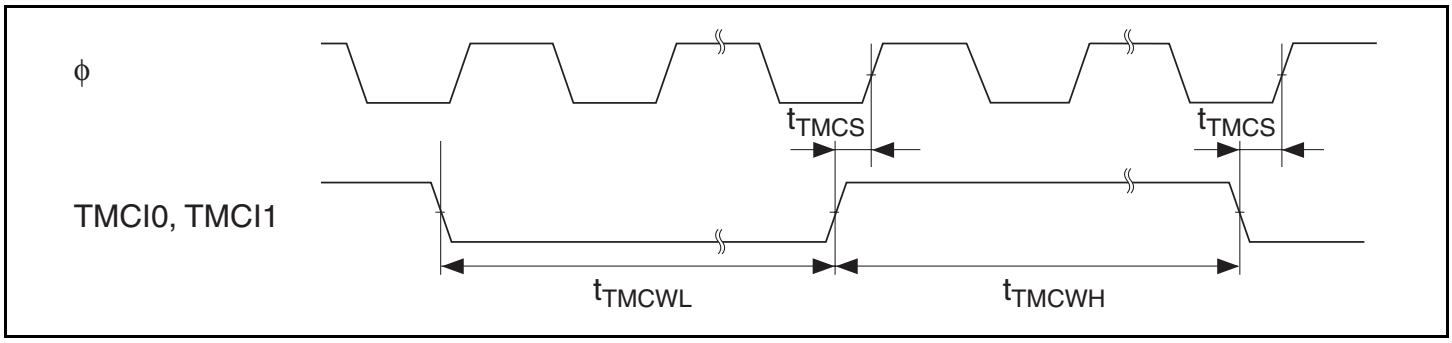


Figure 27.44 8-Bit Timer Clock Input Timing

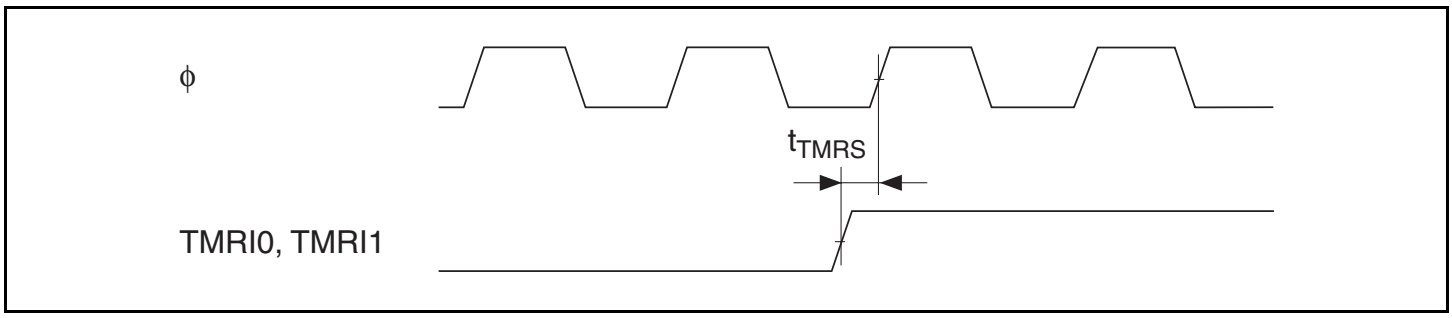


Figure 27.45 8-Bit Timer Reset Input Timing

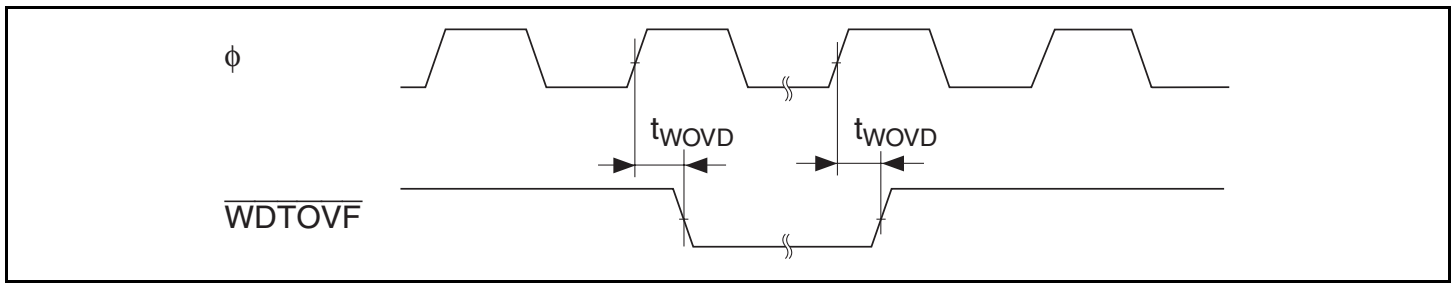


Figure 27.46 WDT Output Timing

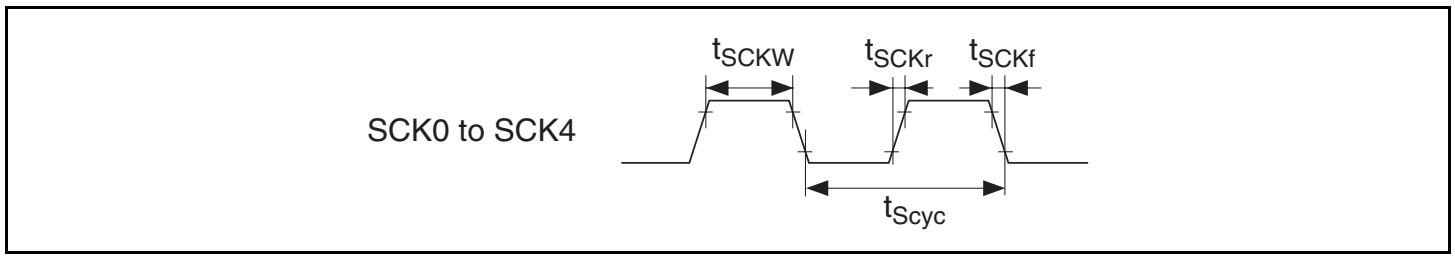


Figure 27.47 SCK Clock Input Timing

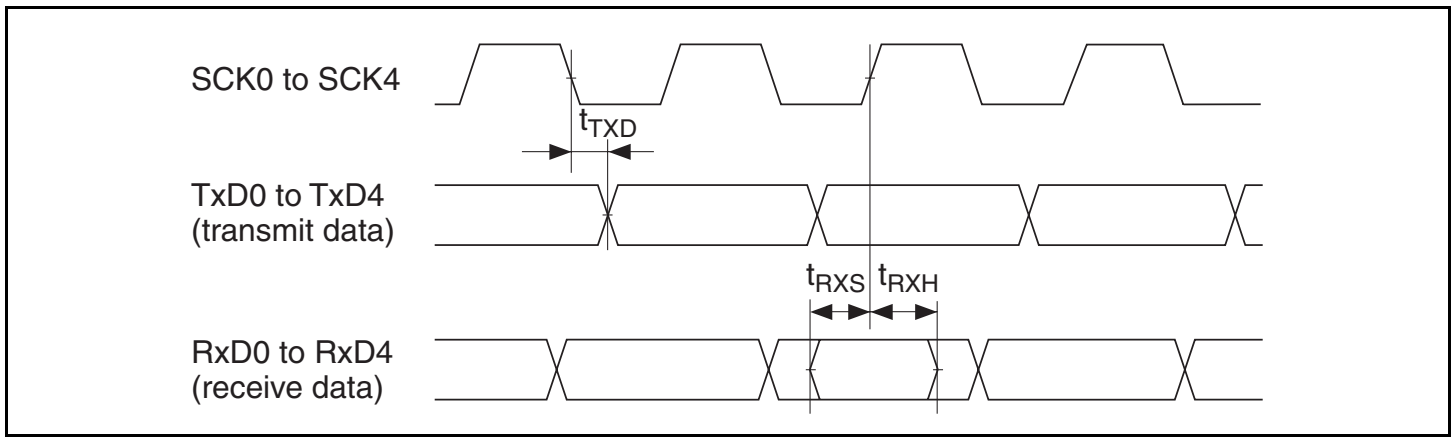


Figure 27.48 SCI Input/Output Timing: Synchronous Mode

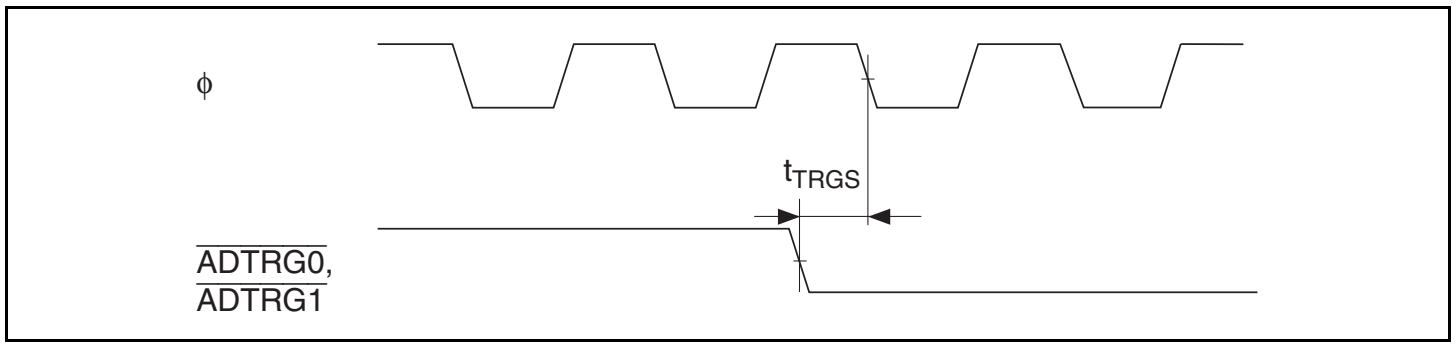
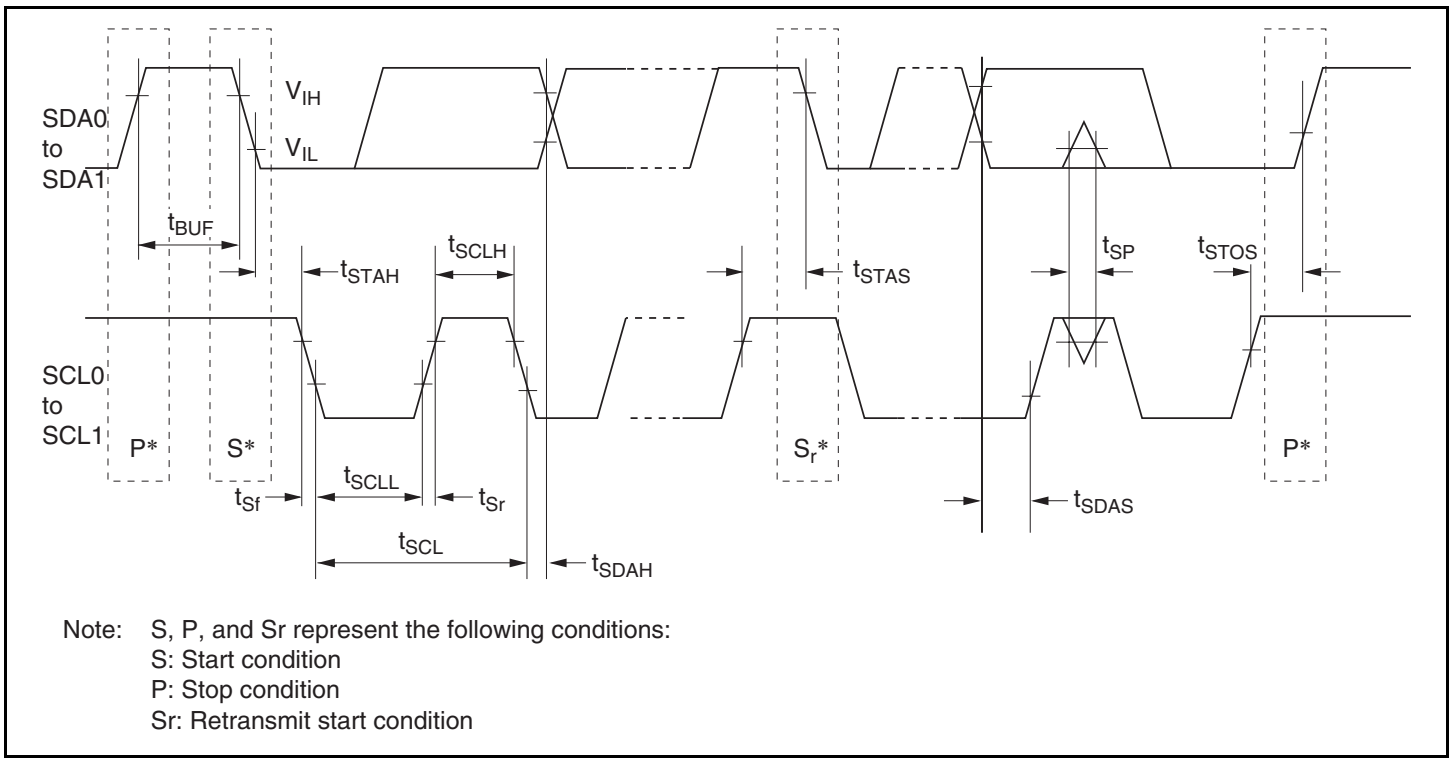


Figure 27.49 A/D Converter External Trigger Input Timing



Note: S, P, and Sr represent the following conditions:
S: Start condition
P: Stop condition
Sr: Retransmit start condition

Figure 27.50 I²C Bus Interface 2 Input/Output Timing (Option)

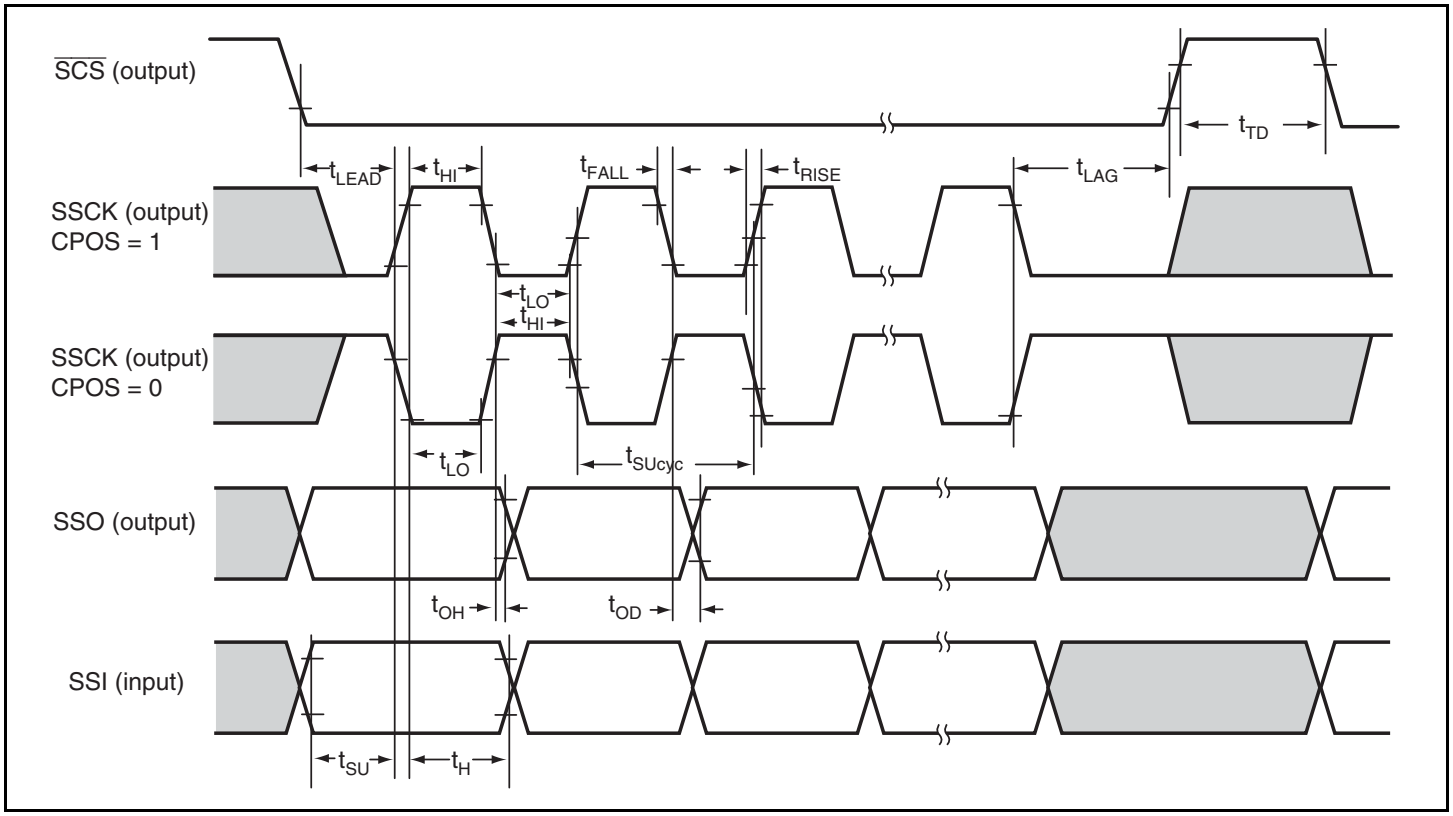


Figure 27.51 SSU Timing (Master, CPHS = 1)

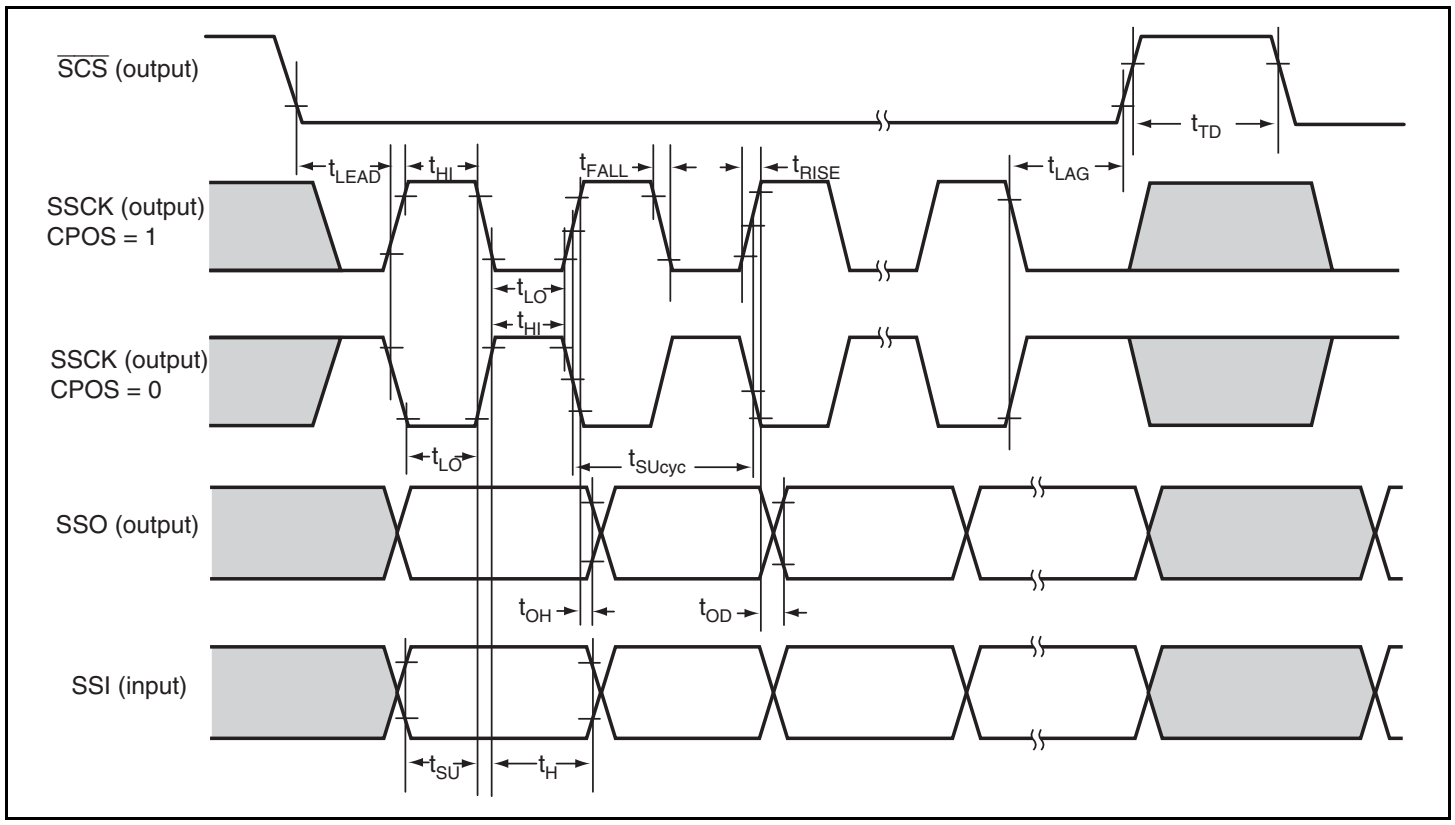


Figure 27.52 SSU Timing (Master, CPHS = 0)

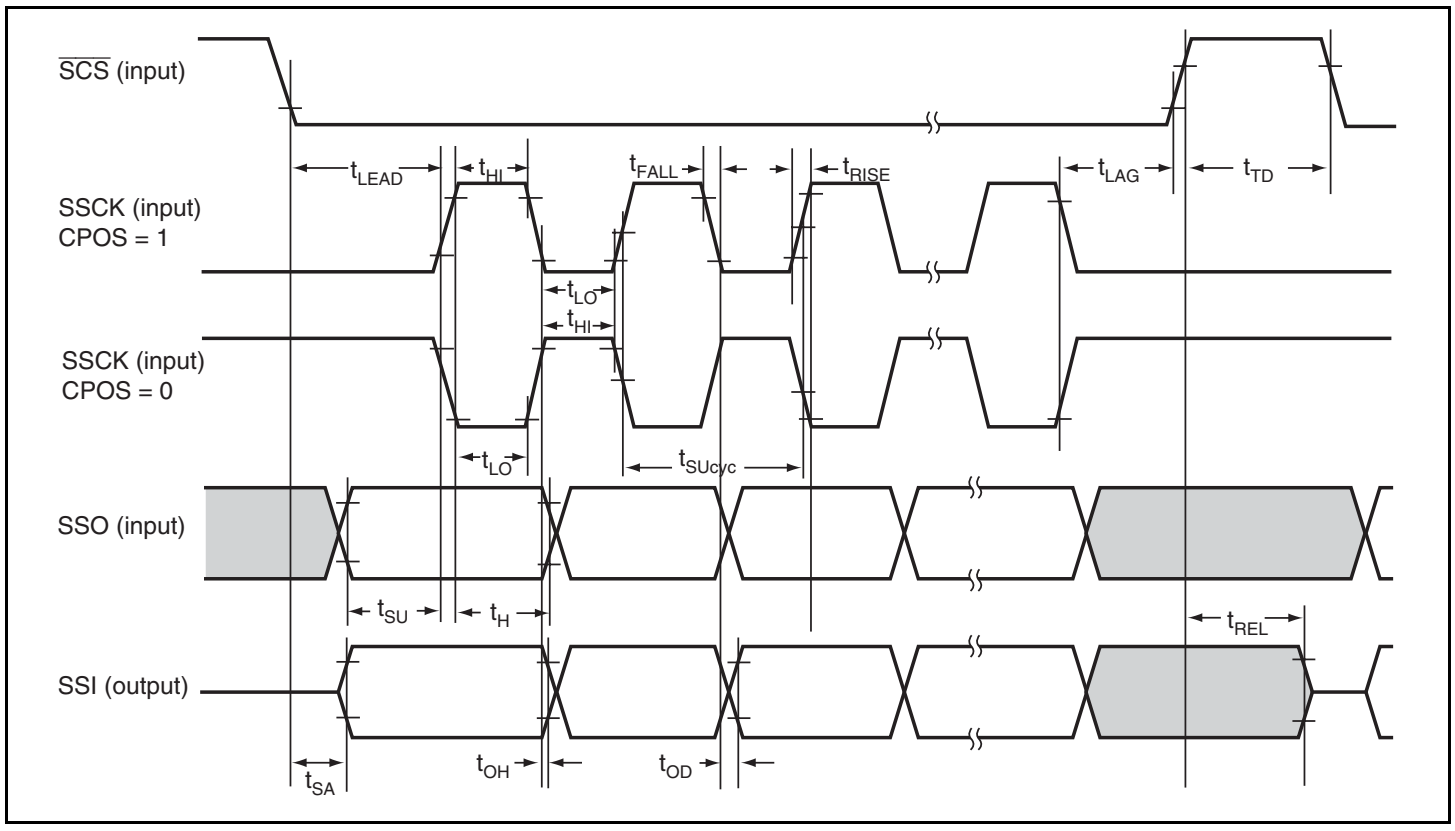


Figure 27.53 SSU Timing (Slave, CPHS = 1)

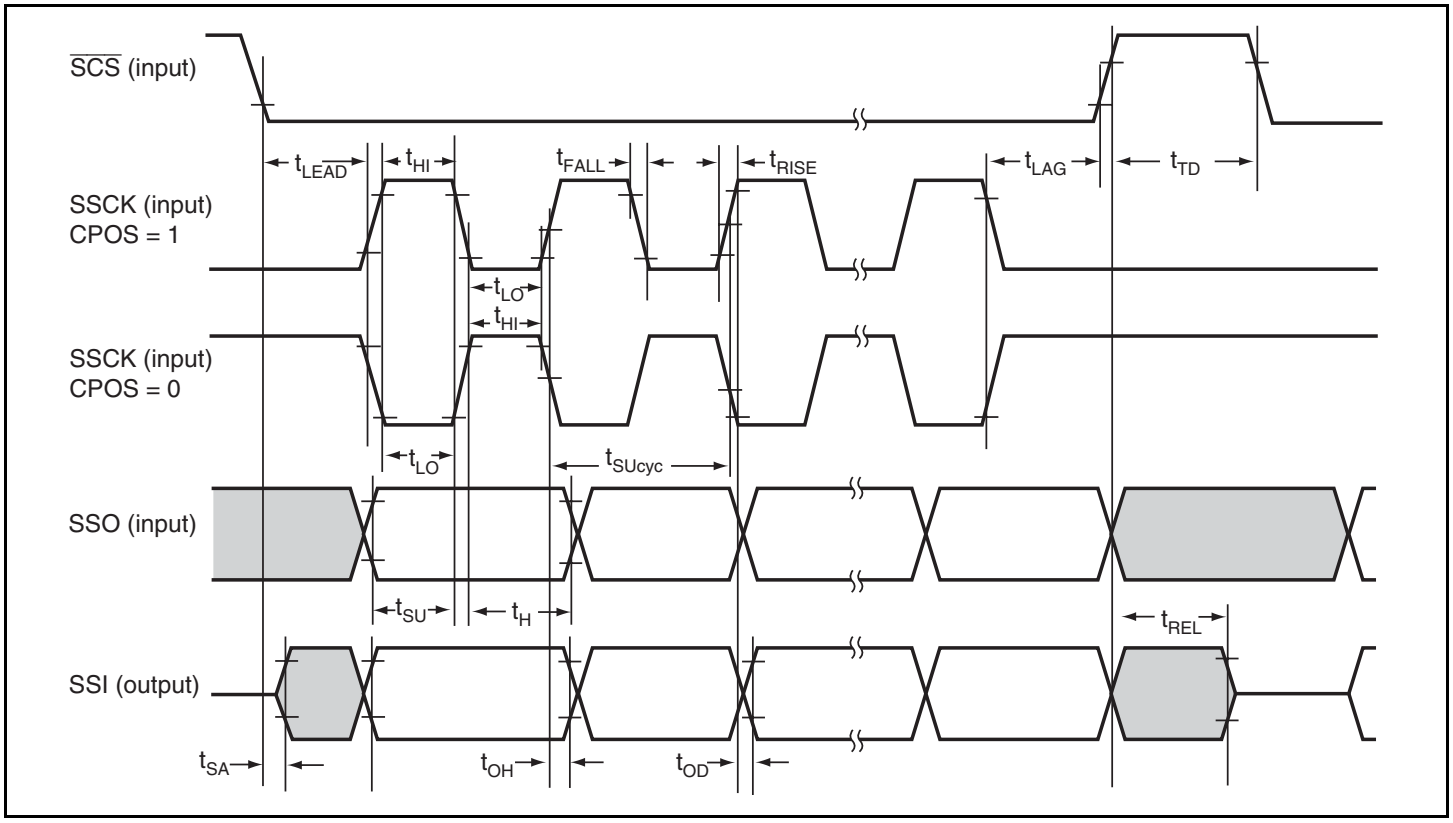


Figure 27.54 SSU Timing (Slave, CPHS = 0)

Appendix

A. Port States in Each Processing State

Table A.1 Port States in Each Processing State (H8S/2456R Group and H8S/2456 Group)

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port 1	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P27 to P26	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P25/$\overline{\text{WAIT}}$	1, 2, 4, 7	T	T	[$\overline{\text{WAIT-B}}$ input] T [Other than the above] Keep	[$\overline{\text{WAIT-B}}$ input] T [Other than the above] Keep	[$\overline{\text{WAIT-B}}$ input] $\overline{\text{WAIT-B}}$ [Other than the above] I/O port
P20	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P34 to P30	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P35/$\overline{\text{OE-B}}$/ CKE-B*[1]	1, 2, 4, 7	T	T	[$\overline{\text{OE-B}}$, CKE-B output, OPE = 0] T [$\overline{\text{OE-B}}$ output, OPE = 1] H [CKE-B output, OPE = 1] L [Other than the above] Keep	[$\overline{\text{OE-B}}$, CKE-B output, OPE = 0] T [Other than the above] Keep	[$\overline{\text{OE-B}}$, CKE-B output, OPE = 0] $\overline{\text{OE-B}}$, CKE-B [Other than the above] I/O port
Port 4	1, 2, 4, 7	T	T	T	T	Input port
P53	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P52/$\overline{\text{BACK-B}}$	1, 2, 4, 7	T	T	[$\overline{\text{BACK-B}}$ output] $\overline{\text{BACK-B}}$ [Other than the above] Keep	[$\overline{\text{BACK-B}}$ output] $\overline{\text{BACK-B}}$ [Other than the above] Keep	[$\overline{\text{BACK-B}}$ output] $\overline{\text{BACK-B}}$ [Other than the above] I/O port

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
P51/$\overline{\text{BREQ-B}}$	1, 2, 4, 7	T	T	[$\overline{\text{BREQ-B}}$ input] T [Other than the above] Keep	[$\overline{\text{BREQ-B}}$ input] $\overline{\text{BREQ-B}}$ [Other than the above] Keep	[$\overline{\text{BREQ-B}}$ input] $\overline{\text{BREQ-B}}$ [Other than the above] I/O port
P50/$\overline{\text{BREQO-B}}$	1, 2, 4, 7	T	T	[$\overline{\text{BREQO-B}}$ output] $\overline{\text{BREQO-B}}$ [Other than the above] Keep	[$\overline{\text{BREQO-B}}$ output] $\overline{\text{BREQO-B}}$ [Other than the above] Keep	[$\overline{\text{BREQO-B}}$ output] $\overline{\text{BREQO-B}}$ [Other than the above] I/O port
Port 6	1, 2, 4, 7	T	T	Keep	Keep	I/O port
Port 8	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P97 to P96	1, 2, 4, 7	T	T	T	T	Input port
P95/DA3	1, 2, 4, 7	T	T	[DAOE3 = 1] Keep [DAOE3 = 0] T	Keep	Input port
P94/DA2	1, 2, 4, 7	T	T	[DAOE2 = 1] Keep [DAOE2 = 0] T	Keep	Input port
P93 to P90	1, 2, 4, 7	T	T	T	T	Input port
PA7/A23 PA6/A22 PA5/A21	1, 2, 4, 7	T	T	[Address output, OPE = 0] T [Address output, OPE = 1] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A23 to A21 [Other than the above] I/O port

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PA4/A20 PA3/A19 PA2/A18 PA1/A17 PA0/A16	1, 2	L	T	[OPE = 0] T [OPE = 1] Keep	T	[Address output] A20 to A16
	3, 4, 7	T	T	[Address output, OPE = 0] T [Address output, OPE = 1] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A20 to A16 [Other than the above] I/O port
Port B	1, 2	L	T	[OPE = 0] T [OPE = 1] Keep	T	[Address output] A15 to A8
	3, 4, 7	T	T	[Address output, OPE = 0] T [Address output, OPE = 1] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A15 to A8 [Other than the above] I/O port

Port Name Pin Name	MCU Operating Mode		Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port C	1, 2		L	T	[OPE = 0] T [OPE = 1] Keep	T	[Address output] A7 to A0
	3, 4, 7		T	T	[Address output, OPE = 0] T [Address output, OPE = 1] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A7 to A0 [Other than the above] I/O port
Port D	1, 2, 4		T	T	T	T	D15 to D8
	3, 5, 7		T	T	[Data bus] T [Other than the above] Keep	[Data bus] T [Other than the above] Keep	[Data bus] D15 to D8 [Other than the above] I/O port
Port E	1, 2, 4	8-bit bus	T	T	Keep	Keep	I/O port
		16-bit bus	T	T	T	T	D7 to D0

Port Name Pin Name	MCU Operating Mode		Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port E	3, 7	8-bit bus	T	T	Keep	Keep	I/O port
		16-bit bus	T	T	[Data bus] T [Other than the above] Keep	[Data bus] T [Other than the above] Keep	[Data bus] D7 to D0 [Other than the above] I/O port
PF7/φ	1, 2, 4		Clock output	T	[Clock output] H	[Clock output] Clock output	[Clock output] Clock output
	3, 7		T		[Other than the above] Keep	[Other than the above] Keep	[Other than the above] Input port
PF6/$\overline{AS}$/$\overline{AH}$	1, 2, 4		H	T	[$\overline{AS}$ output, OPE = 0] T [$\overline{AS}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{AS}$ output] T [Other than the above] Keep	[$\overline{AS}$ output] $\overline{AS}$ [Other than the above] I/O port
	3, 7		T				
PF5/$\overline{RD}$ PF4/$\overline{HWR}$	1, 2, 4		H	T	[OPE = 0] T [OPE = 1] H	T	$\overline{RD}$, $\overline{HWR}$
	3, 7		T		[$\overline{RD}$, $\overline{HWR}$ output, OPE = 0] T [$\overline{RD}$, $\overline{HWR}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{RD}$, $\overline{HWR}$ output] T [Other than the above] Keep	[$\overline{RD}$, $\overline{HWR}$ output] $\overline{RD}$, $\overline{HWR}$ [Other than the above] I/O port

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PF3/$\overline{\text{LWR}}$	1, 2, 4	H	T	[$\overline{\text{LWR}}$ output, OPE = 0] T [$\overline{\text{LWR}}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{\text{LWR}}$ output] T [Other than the above] Keep	[$\overline{\text{LWR}}$ output] $\overline{\text{LWR}}$ [Other than the above] I/O port
	3, 7	T				
PF2/$\overline{\text{LCAS}}$/ DQML*[1]	1, 2, 4, 7	T	T	[$\overline{\text{LCAS}}$, DQML output, OPE = 0] T [$\overline{\text{LCAS}}$, DQML output, OPE = 1] H [Other than the above] Keep	[$\overline{\text{LCAS}}$, DQML output] T [Other than the above] Keep	[$\overline{\text{LCAS}}$, DQML output] $\overline{\text{LCAS}}$, DQML [Other than the above] I/O port
PF1/$\overline{\text{UCAS}}$/ DQMU*[1]	1, 2, 4, 7	T	T	[$\overline{\text{UCAS}}$, DQMU output, OPE = 0] T [$\overline{\text{UCAS}}$, DQMU output, OPE = 1] H [Other than the above] Keep	[$\overline{\text{UCAS}}$, DQMU output] T [Other than the above] Keep	[$\overline{\text{UCAS}}$, DQMU output] $\overline{\text{UCAS}}$ [Other than the above] I/O port
PF0/$\overline{\text{WAIT-A}}$	1, 2, 4, 7	T	T	[$\overline{\text{WAIT-A}}$ input] T [Other than the above] Keep	[$\overline{\text{WAIT-A}}$ input] T [Other than the above] Keep	[$\overline{\text{WAIT-A}}$ input] $\overline{\text{WAIT-A}}$ [Other than the above] I/O port
PG6/$\overline{\text{BREQ-A}}$	1, 2, 4, 7	T	T	[$\overline{\text{BREQ-A}}$ input] T [Other than the above] Keep	[$\overline{\text{BREQ-A}}$ input] $\overline{\text{BREQ-A}}$ [Other than the above] Keep	[$\overline{\text{BREQ-A}}$ input] $\overline{\text{BREQ-A}}$ [Other than the above] I/O port

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PG5/$\overline{\text{BACK-A}}$	1, 2, 4, 7	T	T	[$\overline{\text{BACK-A}}$ output] $\overline{\text{BACK-A}}$ [Other than the above] Keep	[$\overline{\text{BACK-A}}$ output] $\overline{\text{BACK-A}}$ [Other than the above] Keep	[$\overline{\text{BACK-A}}$ output] $\overline{\text{BACK-A}}$ [Other than the above] I/O port
PG4/$\overline{\text{BREQO-A}}$	1, 2, 4, 7	T	T	[$\overline{\text{BREQO-A}}$ output] $\overline{\text{BREQO-A}}$ [Other than the above] Keep	[$\overline{\text{BREQO-A}}$ output] $\overline{\text{BREQO-A}}$ [Other than the above] Keep	[$\overline{\text{BREQO-A}}$ output] $\overline{\text{BREQO-A}}$ [Other than the above] I/O port
PG3/$\overline{\text{CS3}}$ $\overline{\text{RAS3}}$/$\overline{\text{CAS}}$*[1] PG2/$\overline{\text{CS2}}$ $\overline{\text{RAS2}}$/$\overline{\text{RAS}}$*[1] PG1/$\overline{\text{CS1}}$	1, 2, 4, 7	T	T	[$\overline{\text{CS}}$ output, OPE = 0] T [$\overline{\text{CS}}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{\text{CS}}$ output] T [Other than the above] Keep	[$\overline{\text{CS}}$ output] $\overline{\text{CS}}$ [Other than the above] I/O port
PG0/$\overline{\text{CS0}}$	1, 2	H	T	[$\overline{\text{CS}}$ output, OPE = 0] T [$\overline{\text{CS}}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{\text{CS}}$ output] T [Other than the above] Keep	[$\overline{\text{CS}}$ output] $\overline{\text{CS}}$ [Other than the above] I/O port
	3, 4, 7	T				

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PH3/$\overline{\text{OE-A}}$/ CKE-A*[1]/$\overline{\text{CS7}}$	1, 2, 4, 7	T	T	[$\overline{\text{OE-A}}$, $\overline{\text{CS}}$, CKE-A output, OPE = 0] T [$\overline{\text{OE-A}}$ output, OPE = 1] H [$\overline{\text{CS}}$ output, OPE = 1] H [CKE-A output, OPE = 1] L [Other than the above] Keep	[$\overline{\text{OE-A}}$, $\overline{\text{CS}}$, CKE-A output] T [Other than the above] Keep	[$\overline{\text{OE-A}}$, CKE-A output] $\overline{\text{OE-A}}$, CKE-A [$\overline{\text{CS}}$ output] $\overline{\text{CS}}$ [Other than the above] I/O port
PH2/$\overline{\text{CS6}}$	1, 2, 4, 7	T	T	[$\overline{\text{CS}}$ output, OPE = 0] T [$\overline{\text{CS}}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{\text{CS}}$ output] T [Other than the above] Keep	[$\overline{\text{CS}}$ output] $\overline{\text{CS}}$ [Other than the above] I/O port
PH1/$\overline{\text{CS5}}$/ $\overline{\text{RAS5}}$ SDRAMϕ*[1]	1, 2, 4, 7	[SDPSTP = 0] Clock output	[SDPSTP = 0] L [SDPSTP = 1] T	[SDPSTP = 0] L [SDPSTP = 1, $\overline{\text{CS}}$ output, OPE = 0] T [SDPSTP = 1, $\overline{\text{CS}}$ output, OPE = 1] H [Other than the above] Keep	[SDPSTP = 0] Clock output [SDPSTP = 1, $\overline{\text{CS}}$ output] T [Other than the above] Keep	[SDPSTP = 0] Clock output [SDPSTP = 1, $\overline{\text{CS}}$ output] $\overline{\text{CS}}$ [Other than the above] Keep

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PH0/$\overline{CS4}$/ $\overline{RAS4}$/$\overline{WE}$*[1]	1, 2, 4, 7	T	T	[$\overline{CS}$ output, OPE = 0] T [$\overline{CS}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{CS}$ output] T [Other than the above] Keep	[$\overline{CS}$ output] $\overline{CS}$ [Other than the above] I/O port
PJ2	1, 2, 4, 7	T	T	T	T	Input port
PJ1 to PJ0	1, 2, 4, 7	T	T	Keep	Keep	I/O port
WDTOVF	1, 2, 4, 7	H	H	H	H	H*[2]

[Legend]

H: High-level

L: Low-level

Keep: Input ports become high-impedance, and output ports retain their state.

T: High-impedance

DDR: Data direction register

OPE: Output port enable

Notes: 1. Not supported by the H8S/2456 Group.

2. Low output if a watchdog timer overflow occurs when WT/IT is set to 1.

Table A.2 Port States in Each Processing State (H8S/2454 Group)

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port 1	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P27, P26	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P25/$\overline{\text{WAIT-B}}$	1, 2, 4, 7	T	T	[$\overline{\text{WAIT-B}}$ input] T [Other than the above] Keep	[$\overline{\text{WAIT-B}}$ input] T [Other than the above] Keep	[$\overline{\text{WAIT-B}}$ input] $\overline{\text{WAIT-B}}$ [Other than the above] I/O port
P20	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P34 to P30	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P35/$\overline{\text{OE-B}}$	1, 2, 4, 7	T	T	[$\overline{\text{OE-B}}$ output, OPE = 0] T [$\overline{\text{OE-B}}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{\text{OE-B}}$ output] T [Other than the above] Keep	[$\overline{\text{OE-B}}$ output] $\overline{\text{OE}}$ [Other than the above] I/O port
Port 4	1, 2, 4, 7	T	T	T	T	Input port
P53	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P52/$\overline{\text{BACK-B}}$	1, 2, 4, 7	T	T	[$\overline{\text{BACK-B}}$ output] $\overline{\text{BACK-B}}$ [Other than the above] Keep	[$\overline{\text{BACK-B}}$ output] $\overline{\text{BACK-B}}$ [Other than the above] Keep	[$\overline{\text{BACK-B}}$ output] $\overline{\text{BACK-B}}$ [Other than the above] I/O port
P51/$\overline{\text{BREQ-B}}$	1, 2, 4, 7	T	T	[$\overline{\text{BREQ-B}}$ input] T [Other than the above] Keep	[$\overline{\text{BREQ-B}}$ input] $\overline{\text{BREQ-B}}$ [Other than the above] Keep	[$\overline{\text{BREQ-B}}$ input] $\overline{\text{BREQ-B}}$ [Other than the above] I/O port

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
P50/ $\overline{\text{BREQO-B}}$	1, 2, 4, 7	T	T	[$\overline{\text{BREQO-B}}$ output] $\overline{\text{BREQO-B}}$ [Other than the above] Keep	[$\overline{\text{BREQO-B}}$ output] $\overline{\text{BREQO-B}}$ [Other than the above] Keep	[$\overline{\text{BREQO-B}}$ output] $\overline{\text{BREQO-B}}$ [Other than the above] I/O port
Port 8	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P95/DA3	1, 2, 4, 7	T	T	[DAOE3 = 1] Keep [DAOE3 = 0] T	Keep	Input port
P94/DA2	1, 2, 4, 7	T	T	[DAOE2 = 1] Keep [DAOE2 = 0] T	Keep	Input port
PA7/A23/$\overline{\text{CS7}}$	1, 2, 4, 7	T	T	[$\overline{\text{CS}}$ output, OPE = 0] T [$\overline{\text{CS}}$ output, OPE = 1] H [Address output, OPE = 0] T [Address output, OPE = 1] Keep [Other than the above] Keep	[$\overline{\text{CS}}$ output] T [Address output] T [Other than the above] Keep	[$\overline{\text{CS}}$ output] $\overline{\text{CS}}$ [Address output] A23 [Other than the above] I/O port

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PA6/A22 PA5/A21	1, 2, 4, 7	T	T	[Address output, OPE = 0] T [Address output, OPE = 1] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A22 to A21 [Other than the above] I/O port
PA4/A20 PA3/A19 PA2/A18 PA1/A17 PA0/A16	1, 2	L	T	[OPE = 0] T [OPE = 1] Keep	T	[Address output] A20 to A16
	3, 4, 7	T	T	[Address output, OPE = 0] T [Address output, OPE = 1] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A20 to A16 [Other than the above] I/O port

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port B	1, 2	L	T	[OPE = 0] T [OPE = 1] Keep	T	[Address output] A15 to A8
	4	T	T	[Address output, OPE = 0] T [Address output, OPE = 1] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A15 to A8 [Other than the above] I/O port
	3, 7	T	T	[Address output, OPE = 0] T [Address output, OPE = 1] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A15 to A8 [Other than the above] I/O port

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port C	1, 2	L	T	[OPE = 0] T [OPE = 1] Keep	T	[Address output] A7 to A0
	4	T	T	[Address output, OPE = 0] T [Address output, OPE = 1] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A7 to A0 [Other than the above] I/O port
	3, 7	T	T	[Address output, OPE = 0] T [Address output, OPE = 1] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A7 to A0 [Other than the above] I/O port
Port D	1, 2, 4	T	T	T	T	D15 to D8
	3, 7	T	T	[Data bus] T [Other than the above] Keep	[Data bus] T [Other than the above] Keep	[Data bus] D15 to D8 [Other than the above] I/O port

Port Name Pin Name	MCU Operating Mode		Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port E	1, 2, 4	8-bit bus	T	T	Keep	Keep	I/O port
		16-bit bus	T	T	T	T	D7 to D0
	3, 7	8-bit bus	T	T	Keep	Keep	I/O port
		16-bit bus	T	T	[Data bus] T [Other than the above] Keep	[Data bus] T [Other than the above] Keep	[Data bus] D7 to D0 [Other than the above] I/O port
PF7/ϕ	1, 2, 4		Clock output	T	[Clock output] H [Other than the above] Keep	[Clock output] Clock output [Other than the above] Keep	[Clock output] Clock output [Other than the above] Input port
	3, 7		T				
PF6/$\overline{AS}$	1, 2, 4		H	T	[$\overline{AS}$ output, OPE = 0] T [$\overline{AS}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{AS}$ output] T [Other than the above] Keep	[$\overline{AS}$ output] $\overline{AS}$ [Other than the above] I/O port
	3, 7		T				
PF5/$\overline{RD}$ PF4/$\overline{HWR}$	1, 2, 4		H	T	[OPE = 0] T [OPE = 1] H	T	$\overline{RD}$, $\overline{HWR}$
	3, 7		T		[$\overline{RD}$, $\overline{HWR}$ output, OPE = 0] T [$\overline{RD}$, $\overline{HWR}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{RD}$, $\overline{HWR}$ output] T [Other than the above] Keep	[$\overline{RD}$, $\overline{HWR}$ output] $\overline{RD}$, $\overline{HWR}$ [Other than the above] I/O port

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PF3/$\overline{\text{LWR}}$	1, 2, 4	H	T	[$\overline{\text{LWR}}$ output, OPE = 0] T [$\overline{\text{LWR}}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{\text{LWR}}$ output] T [Other than the above] Keep	[$\overline{\text{LWR}}$ output] $\overline{\text{LWR}}$ [Other than the above] I/O port
	3, 7	T				
PF2/$\overline{\text{LCAS}}$/$\overline{\text{CS6}}$	1, 2, 4, 7	T	T	[$\overline{\text{LCAS}}$ output, OPE = 0] T [$\overline{\text{LCAS}}$ output, OPE = 1] H [$\overline{\text{CS}}$ output, OPE = 1] T [$\overline{\text{CS}}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{\text{LCAS}}$ output] T [$\overline{\text{CS}}$ output] T [Other than the above] Keep	[$\overline{\text{LCAS}}$ output] $\overline{\text{LCAS}}$ [$\overline{\text{CS}}$ output] $\overline{\text{CS}}$ [Other than the above] I/O port
PF1/$\overline{\text{UCAS}}$/$\overline{\text{CS5}}$	1, 2, 4, 7	T	T	[$\overline{\text{UCAS}}$ output, OPE = 0] T [$\overline{\text{UCAS}}$ output, OPE = 1] H [$\overline{\text{CS}}$ output, OPE = 1] T [$\overline{\text{CS}}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{\text{UCAS}}$ output] T [$\overline{\text{CS}}$ output] T [Other than the above] Keep	[$\overline{\text{UCAS}}$ output] $\overline{\text{UCAS}}$ [$\overline{\text{CS}}$ output] $\overline{\text{CS}}$ [Other than the above] I/O port

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PF0/$\overline{\text{WAIT-A}}$/$\overline{\text{OE-A}}$	1, 2, 4, 7	T	T	[$\overline{\text{WAIT-A}}$ input] T [$\overline{\text{OE-A}}$ output] T [$\overline{\text{OE-A}}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{\text{WAIT-A}}$ input] T [$\overline{\text{OE-A}}$ output, OPE = 0] T [Other than the above] Keep	[$\overline{\text{WAIT-A}}$ input] $\overline{\text{WAIT-A}}$ [$\overline{\text{OE-A}}$ output, OPE = 0] $\overline{\text{OE-A}}$ [Other than the above] I/O port
PG6/$\overline{\text{BREQ-A}}$	1, 2, 4, 7	T	T	[$\overline{\text{BREQ-A}}$ input] T [Other than the above] Keep	[$\overline{\text{BREQ-A}}$ input] $\overline{\text{BREQ-A}}$ [Other than the above] Keep	[$\overline{\text{BREQ-A}}$ input] $\overline{\text{BREQ-A}}$ [Other than the above] I/O port
PG5/$\overline{\text{BACK-A}}$	1, 2, 4, 7	T	T	[$\overline{\text{BACK-A}}$ output] $\overline{\text{BACK-A}}$ [Other than the above] Keep	[$\overline{\text{BACK-A}}$ output] $\overline{\text{BACK-A}}$ [Other than the above] Keep	[$\overline{\text{BACK-A}}$ output] $\overline{\text{BACK-A}}$ [Other than the above] I/O port
PG4/ $\overline{\text{BREQO-A}}$/ $\overline{\text{CS4}}$	1, 2, 4, 7	T	T	[$\overline{\text{BREQO-A}}$ output] $\overline{\text{BREQO-A}}$ [$\overline{\text{CS4}}$ output, OPE = 0] T [$\overline{\text{CS4}}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{\text{BREQO-A}}$ output] $\overline{\text{BREQO-A}}$ [$\overline{\text{CS4}}$ output] T [Other than the above] Keep	[$\overline{\text{BREQO-A}}$ output] $\overline{\text{BREQO-A}}$ [$\overline{\text{CS4}}$ output] $\overline{\text{CS4}}$ [Other than the above] I/O port

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PG3/$\overline{CS3}$ PG2/$\overline{CS2}$ PG1/$\overline{CS1}$	1, 2, 4, 7	T	T	[$\overline{CS}$ output, OPE = 0] T [$\overline{CS}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{CS}$ output] T [Other than the above] Keep	[$\overline{CS}$ output] $\overline{CS}$ [Other than the above] I/O port
PG0/$\overline{CS0}$	1, 2	H	T	[$\overline{CS}$ output, OPE = 0] T [$\overline{CS}$ output, OPE = 1] H [Other than the above] Keep	[$\overline{CS}$ output] T [Other than the above] Keep	[$\overline{CS}$ output] $\overline{CS}$ [Other than the above] I/O port
	3, 4, 7	T				

[Legend]

H: High-level

L: Low-level

Keep: Input ports become high-impedance, and output ports retain their state.

T: High-impedance

DDR: Data direction register

OPE: Output port enable

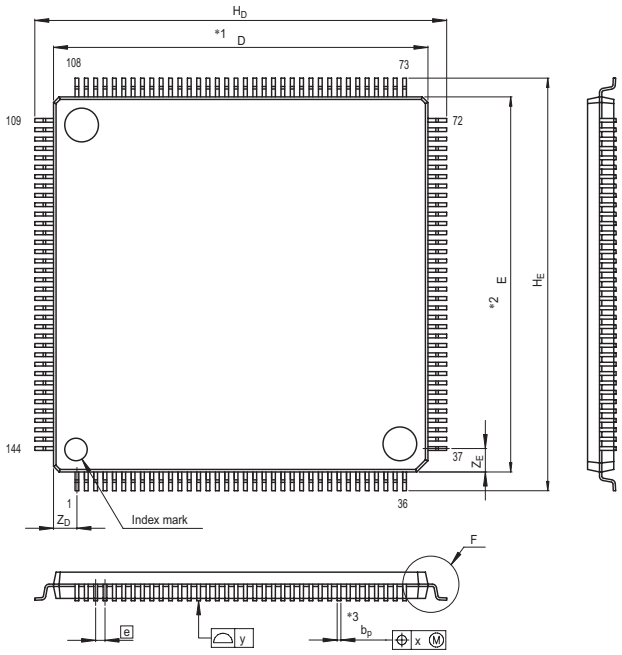
B. Product Code Lineup

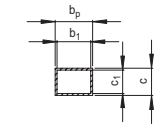
Product Type		Type Code	Mark Code	Package code
H8S/2456R Group	Flash memory version	R4F24569R*	R4F24569VRFQV*	PLQP0144KA-K
		R4F24568R	R4F24568VRFQV	
		R4F24565R	R4F24565VRFQV	
	ROM-less version	R4S24562R*	R4S24562VRFQV*	
		R4S24561R	R4S24561VRFQV	
H8S/2456 Group	Flash memory version	R4F24569*	R4F24569VFQV*	
		R4F24568	R4F24568VFQV	
		R4F24565	R4F24565VFQV	
	ROM-less version	R4S24562*	R4S24562VFQV*	
		R4S24561	R4S24561VFQV	
H8S/2454 Group	Flash memory version	R4F24549*	R4F24549VFQV*	PLQP0120LA-A
		R4F24548	R4F24548VFQV	
		R4F24545	R4F24545VFQV	
	ROM-less version	R4S24542*	R4S24542VFQV*	
		R4S24541	R4S24541VFQV	

Note: * In planning

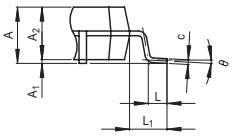
C. Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP144-20x20-0.50	PLQP0144KA-A	144P6Q-A / FP-144L / FP-144LV	1.2g





Terminal cross section



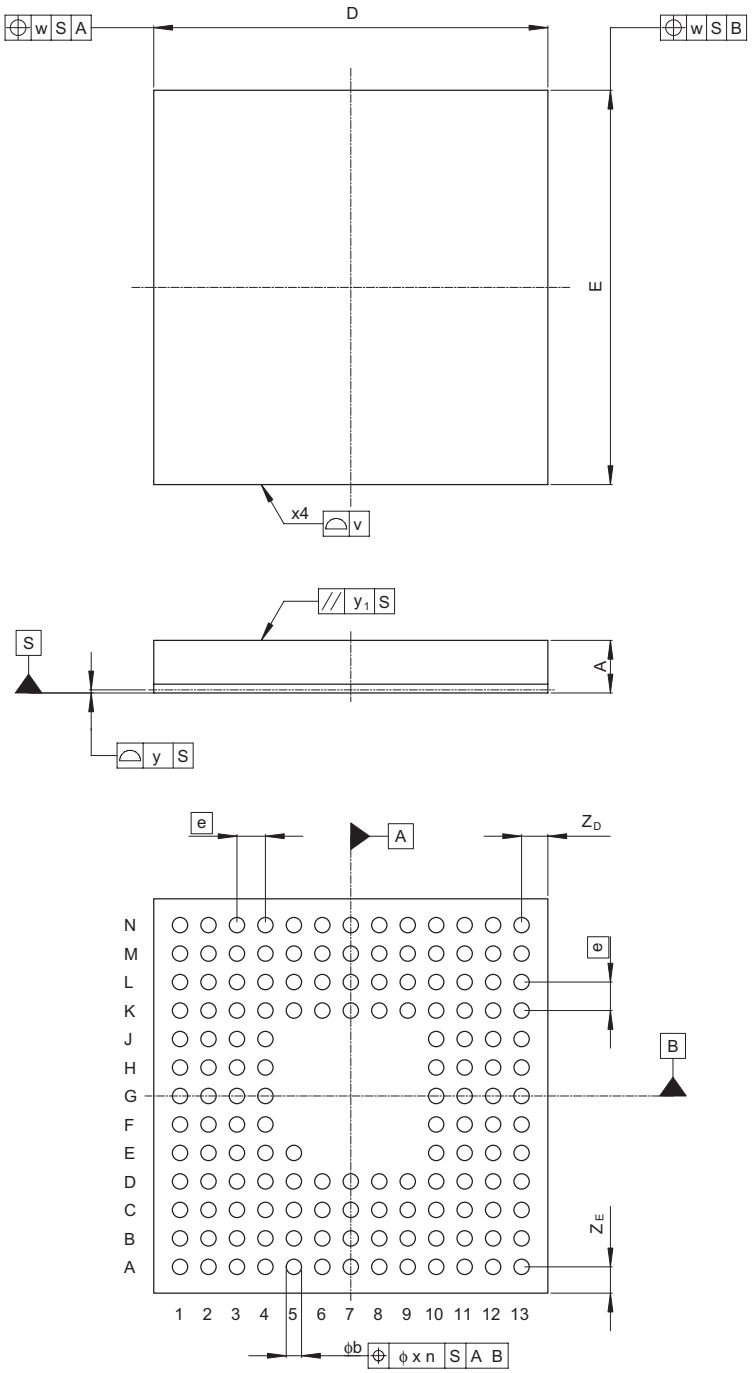
Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	19.9	20.0	20.1
E	19.9	20.0	20.1
A_2	—	1.4	—
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

Figure C.1 Package Dimensions (LQFP2020-144)

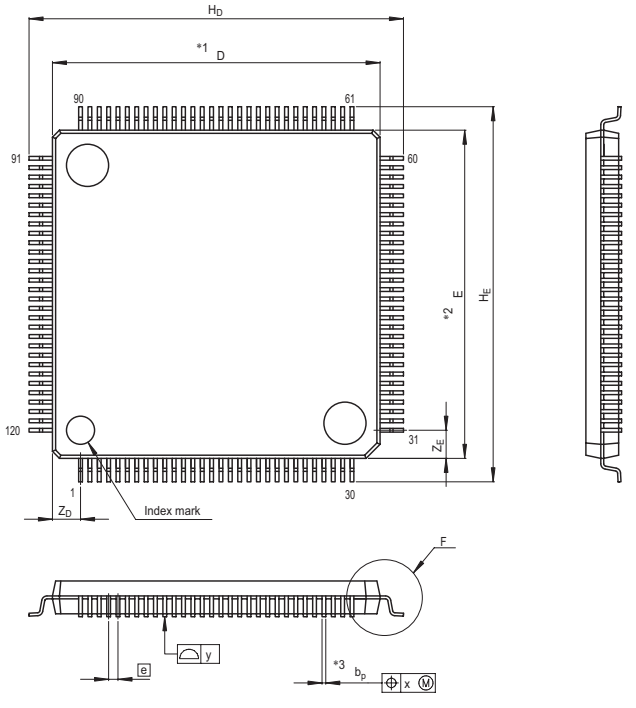
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TFLGA145-9x9-0.65	PTLG0145JB-A	-	0.15g

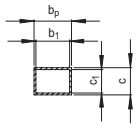


Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	9.0	—
E	—	9.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.2
A_1	—	—	—
e	—	0.65	—
b	0.30	0.35	0.40
x	—	—	0.08
y	—	—	0.1
y_1	—	—	0.20
S_D	—	—	—
S_E	—	—	—
Z_D	—	0.6	—
Z_E	—	0.6	—

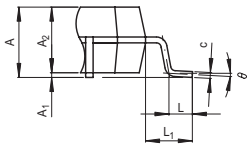
Figure C.2 Package Dimensions (TFLGA-145)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP120-14x14-0.40	PLQP0120LA-A	120P6R-A / FP-120B / FP-120BV	0.7g





Terminal cross section



Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.13	0.18	0.23
b_1	—	0.16	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.4	—
x	—	—	0.07
y	—	—	0.08
Z_D	—	1.2	—
Z_E	—	1.2	—
L	0.35	0.5	0.65
L_1	—	1.0	—

Figure C.3 Package Dimensions (LQFP1414-120)

Index

E

F

G

I

L

M

S

T

U

V

W

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2456, H8S/2456R, H8S/2454 Group

Publication Date:	Rev.1.00, Sep. 19, 2008
Published by:	Sales Strategic Planning Div. Renesas Technology Corp.
Edited by:	Customer Support Department Global Strategic Communication Div. Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

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

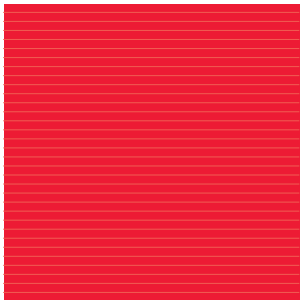
Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

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, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

Colophon 6.2

H8S/2456, H8S/2456R, H8S/2454 Group
Hardware Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan