REJ09B0023-0400

Everywhere you imagine. RENESAS

32

SH7641

Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family / SH7641 Series

SH7641 HD6417641

Hardware Manual

Rev.4.00
Revision Date: Sep. 14, 2005

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

General Precautions on Handling of Product

1. Treatment of NC Pins

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

2. Treatment of Unused Input Pins

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

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

5. Treatment of Power Supply (0 V) Pins

Note: There should be no voltage difference between the system ground pins (0 V power supply), VssQ, Vss, Vss, Vss (PLL1), and Vss (PLL2).

If voltage difference is created between the system ground pins, malfunctions may occur or excessive current flows during standby due to through current. Voltage difference should not be created between the system ground pins, VssQ, Vss, Vss (PLL1), and Vss (PLL2).

Important Notice on the Quality Assurance for this LSI

Although the wafer process and assembly process of this LSI are entrusted to the external silicon foundries, the quality of this LSI is guaranteed for the customers under the quality assurance system of our company.

However, if it is clear that our company is responsible for a defective product, we will only offer, after the agreement of both parties, to exchange it with a new product from stock.

The following shows the robustness (reference values) of the LSI against static-electricity-induced breakdown.

Robustness (Reference Values) of the LSI against Static-electricity-induced Breakdown

Machine Model Method	± 200 V or more
Human Body Model Method	± 1500 V or more
Charged Device Model Method	± 1000 V or more

For the details on the quality assurance of this LSI, contact your nearest Renesas Technology sales representative.

Configuration of This Manual

This manual comprises the following items:

1. General Precautions on Handling of Product
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

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

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

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

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

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

11. Index

Preface

The SH7641 RISC (Reduced Instruction Set Computer) microcomputer includes a Renesas Technology original RISC CPU as its core, and the peripheral functions required to configure a system.

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

Objective: This manual was written to explain the hardware functions and electrical characteristics of this LSI to the above users.
Refer to the SH-3/SH-3E/SH3-DSP Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- Product names
  The following products are covered in this manual.

Product Classifications and Abbreviations

Basic Classification	Product Code
SH7641	HD6417641

- In order to understand the overall functions of the chip

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

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

  Read the SH-3/SH-3E/SH3-DSP Software Manual.

  This product does not support the MMU functions. For example, the LDTLB instruction code is executed as the NOP instruction.

Rules: Register name: The following notation is used for cases when the same or a similar function, e.g. serial communication, is implemented on more than one channel:
XXX_N (XXX is the register name and N is the channel number)

Bit order: The MSB (most significant bit) is on the left and the LSB (least significant bit) is on the right.

Number notation: Binary is B'xxxx, hexadecimal is H'xxxx, decimal is xxxx.

Signal notation: An overbar is added to a low-active signal: $\overline{xxxx}$

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

SH7641manuals:

Document Title	Document No.
SuperH RISC engine SH7641Hardware Manual	This manual
SH-3/SH-3E/SH3-DSP Software Manual	REJ09B0171

Users manuals for development tools:

Document Title	Document No.
SuperH™ RISC engine C/C++ Compiler,Assembler,Optimizing Linkage Editor Compiler Package V.9.00 User's Manual	REJ10B0152
SuperH™ RISC engine High-performance Embedded Workshop 3 Users Manual	REJ10B0025
SuperH RISC engine High-Performance Embedded Workshop 3 Tutorial	REJ10B0023

Application note:

Document Title	Document No.
SuperH RISC engine C/C++ Compiler Package Application Note	REJ05B0463

Abbreviations

ADC	Analog to digital converter
ALU	Arithmetic logic unit
bpp	bits per pixel
bps	bits per second
BSC	Bus state controller
CODEC	Coder-decoder
CPG	Clock pulse generator
CPU	Central processing unit
CRC	Cyclic redundancy check
DMAC	Direct memory access controller
DSP	Digital signal processor
ESD	Electrostatic discharge
ECC	Error checking and correction
etu	Elementary time unit
FIFO	First-in first-out
Hi-Z	High impedance
H-UDI	User debugging interface
INTC	Interrupt controller
LSB	Least significant bit
MSB	Most significant bit
PC	Program counter
PFC	Pin function controller
PLL	Phase locked loop
RAM	Random access memory
RISC	Reduced instruction set computer
ROM	Read only memory
SCIF	Serial communication interface with FIFO
SOF	Start of frame
TAP	Test access port
T.B.D	To be determined
UBC	User break controller

USB	Universal serial bus
WDT	Watch dog timer

Contents

Figures

Section 11 User Break Controller (UBC)

Section 12 Bus State Controller (BSC)

Section 13 Direct Memory Access Controller (DMAC)

Section 14 U Memory

Section 15 User Debugging Interface (H-UDI)

Section 16 I2C Bus Interface 2 (IIC2)

Section 17 Compare Match Timer (CMT)

Section 18 Multi-Function Timer Pulse Unit (MTU)

Section 19 Serial Communication Interface with FIFO (SCIF)

Section 20 USB Function Module

Section 21 A/D Converter

Section 22 Pin Function Controller (PFC)

Section 23 I/O Ports

Section 25 Electrical Characteristics

Appendix

Tables

Section 3 DSP Operation

Section 4 Clock Pulse Generator (CPG)

Section 6 Power-Down Modes

Section 7 Cache

Section 13 Direct Memory Access Controller (DMAC)

Appendix

Section 1 Overview

This LSI is a single-chip RISC microprocessor that integrates a Renesas Technology original 32-bit SuperH RISC engine architecture CPU with a digital signal processing (DSP) extension as its core, with 16-kbyte of cache memory, 16-kbyte of an on-chip X/Y memory, and peripheral functions required for system configuration such as an interrupt controller. This LSI comes in 256-pin package.

High-speed data transfers can be formed by an on-chip direct memory access controller (DMAC), and an external memory access support function enables direct connection to different kinds of memory. This LSI also supports powerful peripheral functions such as USB function and serial communication interface with FIFO.

1.1 Features

The features of this LSI are listed in table 1.1.

Table 1.1 Features

Items	Specification
CPU	• Renesas Technology original SuperH architecture • Compatible with SH-1, SH-2 and SH-3 at object code level • 32-bit internal data bus • Support of an abundant register-set — Sixteen 32-bit general registers (eight 32-bit bank registers) — Eight 32-bit control registers — Four 32-bit system registers • RISC-type instruction set — Instruction length: 16-bit fixed length for improved code efficiency — Load/store architecture — Delayed branch instructions — Instruction set based on C language • Instruction execution time: one instruction/cycle for basic instructions • Logical address space: 4Gbytes • Five-stage pipeline

Items	Specification
DSP	• Mixture of 16-bit and 32-bit instructions • 32-/40-bit internal data paths • Multiplier, ALU, barrel shifter and DSP register • Large DSP data registers — Six 32-bit data registers — Two 40-bit data registers • Extended Harvard Architecture for DSP data bus — Two data buses — One instruction bus • Max. four parallel operations: ALU, multiply, and two load or store • Two addressing units to generate addresses for two memory access • DSP data addressing modes: increment, indexing (with or without modulo addressing) • Zero-overhead repeat loop control • Conditional execution instructions
Clock pulse generator (CPG)	• Clock mode: Input clock can be selected from external input (EXTAL or CKIO) or crystal oscillator • Three types of clocks generated: — CPU clock: maximum 100 MHz — Bus clock: maximum 50 MHz — Peripheral clock: maximum 33 MHz • Power-down modes: — Sleep mode — Standby mode — Module standby mode • Three types of clock modes (selectable PLL2 × 2 / × 4, clock / crystal oscillator)
Watchdog timer	• On-chip one-channel watchdog timer • Select from operation in watchdog-timer or interval-timer mode. • Interrupt generation is supported for the interval-timer mode.

Items	Specification
Cache memory	• 16-kbyte cache, mixed instruction/data • 256 entries, 4-way set associative, 16-byte block length • Write-back, write-through, LRU replacement algorithm • 1-stage write-back buffer • Maximum 2 ways of the cache can be locked
X/Y memory	• Three independent read/write ports — 8-/16-/32-bit access from the CPU — Maximum two 16-bit accesses from the DSP — 8-/16-/32-bit access from the DMAC • Total memory: 16-kbyte (XRAM: 8-kbyte, YRAM: 8-kbyte)
Interrupt controller (INTC)	• Nine external interrupt pins ($\overline{\text{NMI}}$, $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$) • On-chip peripheral interrupts: Priority level set for each module • Supports soft vector mode
User break controller (UBC)	• Addresses, data values, type of access, and data size can all be set as break conditions • Supports a sequential break function • Two break channels

Items	Specification
Bus state controller (BSC)	• Physical address space divided into eight areas, four areas (area 0, areas 2 to 4), each a maximum of 64 Mbytes and other four areas (areas 5A, 5B, areas 6A, 6B), each a maximum of 32 Mbytes • The following features settable for each area independently — Bus size (8, 16, or 32 bits), but different support size by each areas — Number of wait cycles (wait read/write settable independently area exists) — Idle wait cycles (same area/another area) — Specifying the memory to be connected to each area enables direct connection to SRAM, SDRAM, Burst ROM, address/data MPX mode supporting area exists — Outputs chip select signal ($\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$, $\overline{CS5A/B}$, $\overline{CS6A/B}$) for corresponding area (selectable for programming CS assert/negate timing) • SDRAM refresh function — Supports auto-refresh and self-refresh mode • SDRAM burst access function • Area 2/3 enables connection to different SDRAM (size/latency)
Direct memory access controller (DMAC)	• Number of channels: four channels (two channels can accept external requests) • Two types of bus modes — Cycle steal mode and burst mode • Interrupt can be requested to the CPU at completion of data transfer • Supports intermittent mode (16/64 cycles)
User debugging interface (H-UDI)	• E10A emulator support • JTAG-standard pin assignment • Realtime branch trace

Items	Specification
Advanced user debugger (AUD)	• Six output pins • Trace of branch source/destination address • Window data trace function • Full trace function — All trace data can be output by stalling the CPU even when the trace data is not output in time • Real-time trace function — Function to output trace data that can be output at the range not to stall the CPU
Multi-function timer pulse unit (MTU)	• 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 — A maximum 12-phase PWM output is possible in combination with synchronous operation • Buffer operation settable for channels 0,3,and 4 • Phase counting mode settable independently for each of channels 1 and 2 • Cascade connection operation • Fast access via internal 16-bit bus • 23 interrupt sources • Automatic transfer of register data • A/D converter conversion start trigger can be generated • Module standby mode can be set

Items	Specification
Compare match timer (CMT)	• 16-bit counter × 2 channels • Selection of four clocks • Interrupt request or DMA transfer request can be generated by compare-match
Serial communication interface with FIFO (SCIF)	• 3 channels • Asynchronous mode or clock synchronous mode can be selected • Simultaneous transmission/reception (full-duplex) capability • Built-in dedicated baud rate generator • Separate 16-stage FIFO registers for transmission and reception • Dedicated Modem control function (Asynchronous mode)
I/O ports	• Input or output can be selected for each bits
USB function module	• Conforming to the USB standard • Corresponds mode of USB internal transceiver or external transceiver • Supports control (endpoint 0), balk transmission (endpoint 1, 2), interrupt (endpoint 3) • Supports USB standard command and transaction class or vendor command in firmware • FIFO buffer for end point (128-byte/endpoint) • Module input clock: 48MHz. Either self-powered or bus-powered mode can be selected.
I^2C bus interface (IIC2)	• One channel • Conforms to the Phillips I^2C bus interface specification. • Master/slave mode supported • Continuous transmission/reception supported • Either the I^2C bus format or clock synchronous serial format is selectable.
A/D converter	• 10 bits±8 LSB, 8 channels • Input range: 0 to AVcc (max. 3.6V)
U memory	• Three independent read/write ports — 8-/16-/32-bit access from the CPU — 8-/16-/32-bit access from the DSP — 8-/16-bit access from the DMAC • Total memory: 64-kbyte

1.2 Block Diagram

The block diagram of this LSI is shown in figure1.1.

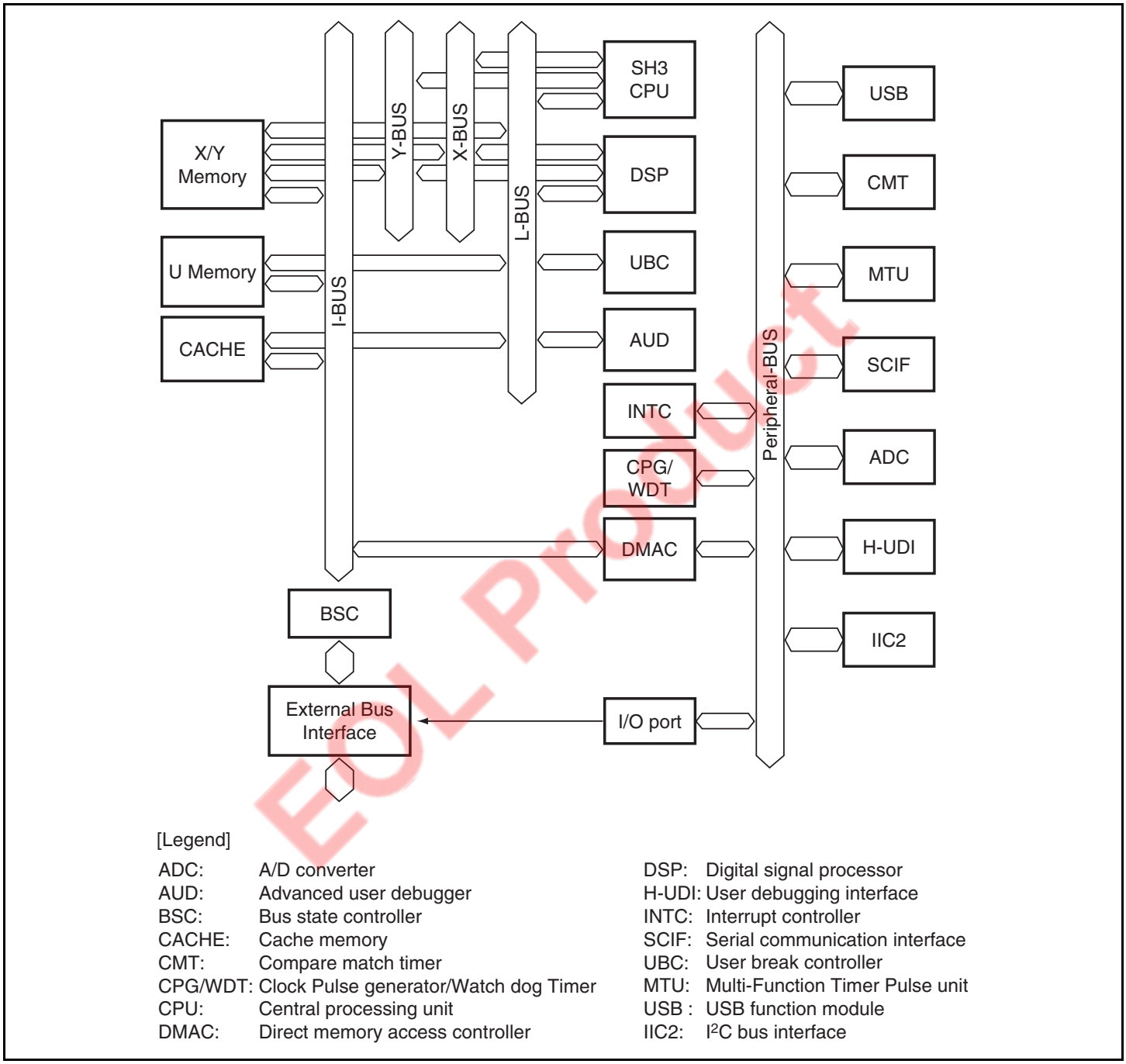


Figure 1.1 Block Diagram

1.3 Pin Assignments

The pin assignments of this LSI is shown in figure 1.2.

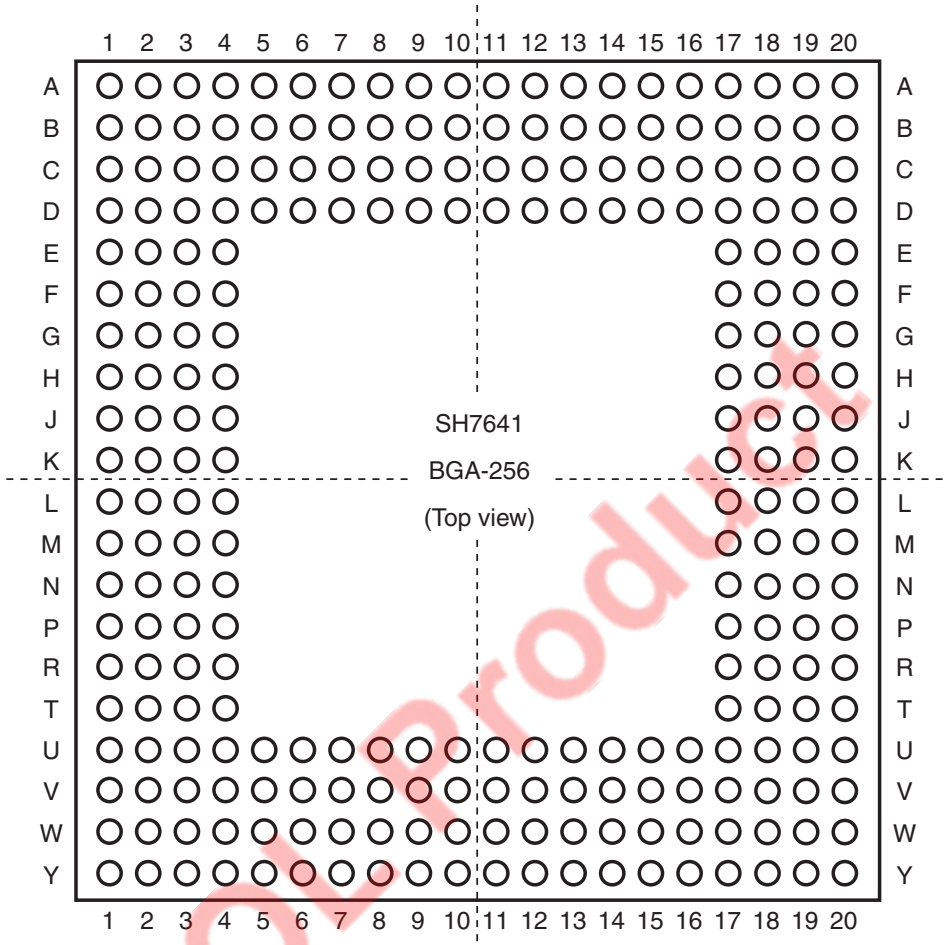


Figure 1.2 Pin Assignments (BGA-256)

1.4 Pin functions

Table 1.2 summarizes the pin functions.

Table 1.2 Pin functions

No. (BGA256)	Pin Name	Description
B2	D7	Data bus
C2	D6	Data bus
D2	D5	Data bus
B1	D4	Data bus
E2	D3	Data bus
E3	D2	Data bus
C1	VssQ	Ground for I/O circuits (0V)
D3	D1	Data bus
D1	VccQ	Power supply for I/O circuits (3.3V)
E4	D0	Data bus
F2	$\overline{CS3}$/PTA[3]	Chip select 3/Port A
F3	Vss	Ground (0V)
E1	$\overline{CS2}$/PTA[2]	Chip select 2/Port A
F4	Vcc	Power supply (1.8V)
G2	UCLK/PTB[0]	USB external input clock/Port B
G3	VBUS/PTB[1]	USB power detection/Port B
F1	SUSPND/PTB[2]	USB suspend/Port B
G4	XVDATA/PTB[3]	Receive data input from USB differential receiver/Port B
H2	TXENL/PTB[4]	USB output enable/Port B
H3	VccQ	Power supply for I/O circuits (3.3V)*[3]
G1	DP	D+
H1	DM	D-
H4	VssQ	Power supply for US I/O circuits (0V)*[3]
J3	TXDMNS/PTB[5]	D- Transmit output for USB transceiver/Port B
J2	TXDPLS/PTB[6]	D+ Transmit output for USB transceiver/Port B
J4	DMNS/PTB[7]	USB D- input from Receiver/Port B

No. (BGA256)	Pin Name	Description
J1	DPLS/PTB[8]	USB D+ input from Receiver/Port B
K3	Vss	Ground (0V)
K2	A18	Address bus
K4	Vcc	Power supply (1.8V)
K1	A19/PTA[8]	Address bus/Port A
L1	A20/PTA[9]	Address bus/Port A
L4	A21/PTA[10]	Address bus/Port A
M1	A22/PTA[11]	Address bus/Port A
L3	A23/PTA[12]	Address bus/Port A
L2	A24/PTA[13]	Address bus/Port A
M4	VssQ	Ground for I/O circuits (0V)
N1	AUDCK	AUD clock
M3	VccQ	Power supply for I/O circuits (3.3V)
M2	A25/PTA[14]	Address bus/Port A
N4	AUDATA[0]/PTJ[8]	AUD data/Port J
P1	AUDATA[1]/PTJ[9]	AUD data/Port J
N3	AUDATA[2]/PTJ[10]	AUD data/Port J
N2	AUDATA[3]/PTJ[11]	AUD data/Port J
P4	$\overline{\text{AUDSYNC}}$/PTJ[12]	AUD synchronized/Port J
R1	TCK	Test clock
P3	TDI	Test data input
T1	TDO	Test data output
R4	TMS	Test mode select
P2	$\overline{\text{TRST}}$	Test reset
R3	$\overline{\text{NMI}}$	Nonmaskable interrupt request
U1	$\overline{\text{IRQ0}}$/PTJ[0]	External interrupt request/Port J
T4	Vcc	Power supply (1.8V)
R2	$\overline{\text{IRQ1}}$/PTJ[1]	External interrupt request/Port J
U4	Vss	Ground (0V)
V1	VssQ	Ground for I/O circuits (0V)
U2	$\overline{\text{IRQ2}}$/PTJ[2]	External interrupt request/Port J

No. (BGA256)	Pin Name	Description
W1	VccQ	Power supply for I/O circuits (3.3V)
V3	$\overline{\text{IRQ3}}$/PTJ[3]	External interrupt request/Port J
T2	$\overline{\text{IRQ4}}$/PTJ[4]	External interrupt request/Port J
T3	$\overline{\text{IRQ5}}$/PTJ[5]	External interrupt request/Port J
U3	$\overline{\text{IRQ6}}$/PTJ[6]	External interrupt request/Port J
V2	$\overline{\text{IRQ7}}$/PTJ[7]	External interrupt request/Port J
Y1	SCK0/PTH[0]	Serial clock 0/Port H
W2	$\overline{\text{CTS0}}$/PTH[1]	Transmit clear 0/Port H
W3	TxD0/PTH[2]	Transmit data 0/Port H
W4	RxD0/PTH[3]	Receive data 0/Port H
Y2	$\overline{\text{RTS0}}$/PTH[4]	Transmit request 0/Port H
W5	SCK1/PTH[5]	Serial clock 1/Port H
V5	$\overline{\text{CTS1}}$/PTH[6]	Transmit clear 1/Port H
Y3	TxD1/PTH[7]	Transmit data 1/Port H
V4	RxD1/PTH[8]	Receive data 1/Port H
Y4	$\overline{\text{RTS1}}$/PTH[9]	Transmit request 1/Port H
U5	SCK2/PTH[10]	Serial clock 2/Port H
W6	$\overline{\text{CTS2}}$/PTH[11]	Transmit clear 2/Port H
V6	Vss	Ground (0V)
Y5	TxD2/PTH[12]	Transmit data 2/Port H
U6	Vcc	Power supply (1.8V)
W7	RxD2/PTH[13]	Receive data 2/Port H
V7	VccQ	Power supply for I/O circuits (3.3V)
Y6	$\overline{\text{RTS2}}$/PTH[14]	Transmit request 2/Port H
U7	VssQ	Ground for I/O circuits (0V)
W8	TIOC4D/PTE[0]	Timer input output 4D/Port E
V8	TIOC4C/PTE[1]	Timer input output 4C/Port E
Y7	TIOC4B/PTE[2]	Timer input output 4B/Port E
U8	TIOC4A/PTE[3]	Timer input output 4A/Port E
Y8	TIOC3D/PTE[4]	Timer input output 3D/Port E
V9	TIOC3B/PTE[6]	Timer input output 3B/Port E

No. (BGA256)	Pin Name	Description
W9	TIOC3C/PTE[5]	Timer input output 3C/Port E
U9	TIOC3A/PTE[7]	Timer input output 3A/Port E
Y9	TIOC2B/PTE[8]	Timer input output 2B/Port E
V10	Vss	Ground (0V)
W10	TIOC2A/PTE[9]	Timer input output 2A/Port E
U10	Vcc	Power supply (1.8V)
Y10	TIOC1B/PTE[10]	Timer input output 1B/Port E
Y11	TIOC1A/PTE[11]	Timer input output 1A/Port E
U11	TIOC0D/PTE[12]	Timer input output 0D/Port E
Y12	TIOC0C/PTE[13]	Timer input output 0C/Port E
V11	TIOC0B/PTE[14]	Timer input output 0B/Port E
W11	TIOC0A/PTE[15]	Timer input output 0A/Port E
U12	VssQ	Ground for I/O circuits (0V)
Y13	TCLKD/PTF[8]	Timer Clock Input D/Port F
V12	VccQ	Power supply for I/O circuits (3.3V)
W12	TCLKC/PTF[9]	Timer Clock Input C/Port F
U13	TCLKB/PTF[10]	Timer Clock Input B/Port F
Y14	TCLKA/PTF[11]	Timer Clock Input A/Port F
V13	$\overline{\text{POE0}}$/PTF[12]	Port output enable input 0/Port F
W13	$\overline{\text{POE1}}$/PTF[13]	Port output enable input 1/Port F
U14	$\overline{\text{POE2}}$/PTF[14]	Port output enable input 2/Port F
Y15	$\overline{\text{POE3}}$/PTF[15]	Port output enable input 3/Port F
V14	PTF[0]	Port F
Y16	PTF[1]	Port F
U15	PTF[2]	Port F
W14	PTF[3]	Port F
V15	PTF[4]	Port F
Y17	PTF[5]	Port F
U16	Vcc	Power supply (1.8V)
W15	PTF[6]	Port F
U17	Vss	Ground (0V)

No. (BGA256)	Pin Name	Description
Y18	VssQ	Ground for I/O circuits (0V)
W17	PTF[7]	Port F
Y19	VccQ	Power supply for I/O circuits (3.3V)
V18	PTG[8]	Port G
W16	SCL/PTG[9]	Serial clock/Port G*[2]
V16	SDA/PTG[10]	Serial data/Port G*[2]
V17	PTG[11]	Port G
W18	PTG[12]	Port G
Y20	PTG[13]	Port G
W19	AVss (AD)	Ground for A/D (0V)
V19	AN[0]/PTG[0]	A/D converter input/Port G*[2]
U19	AN[1]/PTG[1]	A/D converter input/Port G*[2]
W20	AN[2]/PTG[2]	A/D converter input/Port G*[2]
T19	AN[3]/PTG[3]	A/D converter input/Port G*[2]
T18	AN[4]/PTG[4]	A/D converter input/Port G*[2]
V20	AN[5]/PTG[5]	A/D converter input/Port G*[2]
U18	AN[6]/PTG[6]	A/D converter input/Port G*[2]
U20	AVcc (AD)	Power supply for A/D (3.3V)
T17	AN[7]/PTG[7]	A/D converter input/Port G*[2]
R19	VccQ*[1]	Power supply for I/O circuits (3.3V)*[1]
R18	Vss	Ground (0V)
T20	$\overline{\text{DREQ0}}$/PTC[9]	DMA request/Port C
R17	Vcc	Power supply (1.8V)
P19	$\overline{\text{DREQ1}}$/PTC[10]	DMA request/Port C
P18	STATUS0/PTC[14]	Processor status/Port C
R20	STATUS1/PTC[15]	Processor status/Port C
P17	$\overline{\text{BREQ}}$/PTC[6]	Bus request/Port C
N19	$\overline{\text{BACK}}$/PTC[7]	Bus acknowledge/Port C
N18	VccQ*[1]	Power supply for I/O circuits (3.3V)*[1]
P20	VccQ*[1]	Power supply for I/O circuits (3.3V)*[1]
N17	$\overline{\text{ASEBRKAK}}$/PTC[13]	ASE brake acknowledge/Port C

No. (BGA256)	Pin Name	Description
N20	$\overline{\text{RESETP}}$	Power–on Reset request
M18	VccQ	Power supply for I/O circuits (3.3V)
M19	VssQ	Ground for I/O circuits (0V)
M17	XTAL	Clock oscillator pin
M20	EXTAL	External clock/Crystal oscillator pin
L18	Vss	Ground (0V)
L19	$\overline{\text{RESETM}}$	Manual Reset request
L17	Vcc	Power supply (1.8V)
L20	$\overline{\text{ASEMD0}}$	ASE mode
K20	Vss(PLL2)	Ground for PLL 2 (0V)
K17	Vcc(PLL2)	Power supply for PLL 2 (1.8V)
J20	Vcc(PLL1)	Power supply for PLL 1 (1.8V)
K18	Vss(PLL1)	Ground for PLL 1 (0V)
K19	MD3	Bus width set for area 0
J17	MD2	Clock mode set
H20	VccQ*[1]	Power supply for I/O circuits (3.3V)*[1]
J18	MD0	Clock mode set
J19	$\overline{\text{CS6B}}$/PTC[4]	Chip select 6B/Port C
H17	VssQ	Ground for I/O circuits (0V)
G20	$\overline{\text{CS6A}}$/PTC[3]	Chip select 6A/Port C
H18	VccQ	Power supply for I/O circuits (3.3V)
H19	$\overline{\text{CS5B}}$/PTC[2]	Chip select 5B/Port C
G17	$\overline{\text{CS5A}}$/PTC[1]	Chip select 5A/Port C
F20	$\overline{\text{CS4}}$/PTC[0]	Chip select 4/Port C
G18	$\overline{\text{WAIT}}$	Hardware wait request
E20	$\overline{\text{CS0}}$	Chip select 0
F17	$\overline{\text{BS}}$	Bus cycle start
G19	$\overline{\text{TEND}}$/PTC[8]	DMA transfer end/Port C
F18	$\overline{\text{FRAME}}$/PTC[5]	FRAME output/Port C
D20	$\overline{\text{RD}}$	Read strobe
E17	Vcc	Power supply (1.8V)

No. (BGA256)	Pin Name	Description
F19	$\overline{\text{DACK0}}$/PTC[11]	DMA request acknowledge/Port C
D17	Vss	Ground (0V)
C20	VssQ	Ground for I/O circuits (0V)
D19	$\overline{\text{DACK1}}$/PTC[12]	DMA request acknowledge/Port C
B20	VccQ	Power supply for I/O circuits (3.3V)
C18	D31/PTD[15]	Data bus/Port D
E19	D30/PTD[14]	Data bus/Port D
E18	D29/PTD[13]	Data bus/Port D
D18	D28/PTD[12]	Data bus/Port D
C19	D27/PTD[11]	Data bus/Port D
A20	D26/PTD[10]	Data bus/Port D
B19	D25/PTD[9]	Data bus/Port D
B18	D24/PTD[8]	Data bus/Port D
B17	D23/PTD[7]	Data bus/Port D
A19	D22/PTD[6]	Data bus/Port D
B16	D21/PTD[5]	Data bus/Port D
C16	D20/PTD[4]	Data bus/Port D
A18	VssQ	Ground for I/O circuits (0V)
C17	D19/PTD[3]	Data bus/Port D
A17	VccQ	Power supply for I/O circuits (3.3V)
D16	D18/PTD[2]	Data bus/Port D
B15	D17/PTD[1]	Data bus/Port D
C15	Vss	Ground (0V)
A16	D16/PTD[0]	Data bus/Port D
D15	Vcc	Power supply (1.8V)
B14	CKIO2	System clock output
C14	VccQ	Power supply for I/O circuits (3.3V)
A15	CKIO	System clock for I/O circuits
D14	VssQ	Ground for I/O circuits (0V)
B13	RD/$\overline{\text{WR}}$	Read/Write
C13	VccQ	Power supply for I/O circuits (3.3V)

No. (BGA256)	Pin Name	Description
A14	$\overline{WE0}$/DQMLL	D7 to D0 Select signal/DQM (SDRAM)
D13	VssQ	Ground for I/O circuits (0V)
A13	$\overline{WE1}$/DQMLU	D15 to D8 Select signal/DQM (SDRAM)
C12	$\overline{CASU}$/PTA[5]	CAS for Upper-32M-byte address/Port A
B12	$\overline{WE3}$/DQMUU/AH	D31 to D24 Select signal/DQM (SDRAM)/ Address hold (MPX)
D12	$\overline{RASU}$/PTA[7]	RAS for Upper-32M-byte address/Port A
A12	$\overline{WE2}$/DQMUL	D23 to D16 Select signal/DQM (SDRAM)
C11	Vss	Ground (0V)
B11	CKE/PTA[1]	CK enable/Port A
D11	Vcc	Power supply (1.8V)
A11	$\overline{CASL}$/PTA[4]	CAS for Lower-32M-byte address/Port A
A10	$\overline{RASL}$/PTA[6]	RAS for Lower-32M-byte address/Port A
D10	A17	Address bus
A9	A16	Address bus
C10	A15	Address bus
B10	A14	Address bus
D9	A13	Address bus
A8	A12	Address bus
C9	A11	Address bus
B9	A10	Address bus
D8	VssQ	Ground for I/O circuits (0V)
A7	A9	Address bus
C8	VccQ	Power supply for I/O circuits (3.3V)
B8	A8	Address bus
D7	A7	Address bus
A6	A6	Address bus
C7	A5	Address bus
A5	A4	Address bus
D6	A3	Address bus
B7	A2	Address bus

No. (BGA256)	Pin Name	Description
C6	A1	Address bus
A4	A0/PTA[0]	Address bus/Port A
D5	Vcc	Power supply (1.8V)
B6	D15	Data bus
D4	Vss	Ground (0V)
A3	VssQ	Ground for I/O circuits (0V)
B4	D14	Data bus
A2	VccQ	Power supply for I/O circuits (3.3V)
C3	D13	Data bus
B5	D12	Data bus
C5	D11	Data bus
C4	D10	Data bus
B3	D9	Data bus
A1	D8	Data bus

Notes: Treatment of unused pins: All the I/O buffers except PTG10, PTG9, and PTG 7 to PTG 0 (IIC2 and analog pins) have weak keepers. Weak-keeper circuits are provided on input/output pins, and fix the pin inputs to high or low level when the pins are not driven externally. Unused pins that are provided weak-keeper circuits need not to be fixed their input levels. Fix unused pins that are not provided weak-keeper circuits to high or low level.

1. These pins are not real power supply for LSI, but each pin should be supplied each specified voltage for correct action.
2. Weak-keeper circuits are not provided on the I/O buffer pins. Accordingly, pull the pins up or down when they are not in use. Furthermore, do not apply intermediate voltages to these pins when you are using them as port input pins.
3. H3 and H4 are a pair of power-supply pins located in the nearest position to the USB module in this LSI.
   Insert a bypass capacitor to the pair of pins to improve the electrical characteristic for the USB input/output.

Table 1.3 lists the pin functions.

Table 1.3 Pin Functions

Classification	Symbol	I/O	Name	Function
Power supply	Vcc	I	Power supply	Power supply for the internal LSI. Connect all Vcc pins to the system. There will be no operation if any pins are open.
	Vss	I	Ground	Ground pin. Connect all Vss pins to the system power supply (0V). There will be no operation if any pins are open.
	VccQ	I	Power supply	Power supply for I/O pins. Connect all VccQ pins to the system power supply. There will be no operation if any pins are open.
	VssQ	I	Ground	Ground pin. Connect all VssQ pins to the system power supply (0V). There will be no operation if any pins are open.
Clock	Vcc (PLL1)	I	PLL1 power supply	Power supply for the on-chip PLL1 oscillator
	Vss (PLL1)	I	PLL1 ground	Ground pin for the on-chip PLL1 oscillator
	Vcc (PLL2)	I	PLL2 power supply	Power supply for the on-chip PLL2 oscillator
	Vcc (PLL2)	I	PLL2 ground	Ground pin for the on-chip PLL2 oscillator
	EXTAL	I	External clock	Connected to a crystal resonator. An external clock signal may also be input to the EXTAL pin. For examples of the connection of crystal resonator or an external clock signal, see section 4, Clock Pulse Generator (CPG).
	XTAL	O	Crystal	Connected to a crystal resonator. For examples of the connection of crystal resonator or an external clock signal, see section 4, Clock Pulse Generator (CPG).

Classification	Symbol	I/O	Name	Function
Clock	CKIO	O	System clock	Supplies the system clock to external devices.
	CKIO2	O	System clock	Supplies the system clock to external devices.
Operating mode control	MD3, MD2, MD0	I	Mode set	Sets the operating mode. Do not change values on these pins during operation. MD2, MD0 set the clock mode, MD3 set the bus-width mode of area 0.
System control	$\overline{\text{RESETP}}$	I	Power-on reset	When low, this LSI enters the power-on reset state.
	$\overline{\text{RESETM}}$	I	Manual reset	When low, this LSI enters the manual reset state.
	STATUS1, STATUS0	O	Status output	Indicate that this LSI is in software standby, reset, or sleep mode.
	$\overline{\text{BREQ}}$	I	Bus-mastership request	Low when an external device requests the release of the bus mastership.
	$\overline{\text{BACK}}$	O	Bus-mastership request acknowledge	Indicates that the bus mastership has been released to an external device. Reception of the $\overline{\text{BACK}}$ signal informs the device which has output the $\overline{\text{BREQ}}$ signal that it has acquired the bus.
Interrupts	$\overline{\text{NMI}}$	I	Non-maskable interrupt	Non-maskable interrupt request pin. Fix to high level when not in use.
	$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	I	Interrupt requests 7 to 0	Maskable interrupt request pin. Selectable as level input or edge input. The rising edge, falling edge, and both edges are selectable as edges.
Address bus	A25 to A0	O	Address bus	Outputs addresses.
Data bus	D31 to D0	I/O	Data bus	32-bit bidirectional bus.
Bus control	$\overline{\text{CS0}}$, $\overline{\text{CS2}}$ to $\overline{\text{CS4}}$, $\overline{\text{CS5A}}$, $\overline{\text{CS5B}}$, $\overline{\text{CS6A}}$, $\overline{\text{CS6B}}$	O	Chip select 0, 2 to 4, 5A, 5B, 6A, 6B	Chip-select signal for external memory or devices.
	$\overline{\text{RD}}$	O	Read	Indicates reading of data from external devices.

Classification	Symbol	I/O	Name	Function
Bus control	RD/$\overline{WR}$	O	Read/write	Read/write signal
	$\overline{BS}$	O	Bus start	Bus-cycle start
	$\overline{WE3}$/DQMUU/ $\overline{AH}$	O	Byte specification	Indicates that bits 31 to 24 of the data in the external memory or device are being written. Selects D31 to D24 when SDRAM is connected. Address hold signal for address/data multiplexed I/O.
	$\overline{WE2}$/DQMUL	O	Byte specification	Indicates that bits 23 to 16 of the data in the external memory or device are being written. Selects D23 to D16 when SDRAM is connected.
	$\overline{WE1}$/DQMLU	O	Byte specification	Indicates that bits 15 to 8 of the data in the external memory or device are being written. Selects D15 to D8 when SDRAM is connected.
	$\overline{WE0}$/DQMLL	O	Byte specification	Indicates that bits 7 to 0 of the data in the external memory or device are being written. Selects D7 to D0 when SDRAM is connected.
	$\overline{RASU}$, $\overline{RASL}$	O	RAS	Connected to the $\overline{RAS}$ pin when the SDRAM is connected.
	$\overline{CASU}$, $\overline{CASL}$	O	CAS	Connected to the $\overline{CAS}$ pin when the SDRAM is connected.
	CKE	O	CK enable	Connected to the CKE pin when the SDRAM is connected.
	$\overline{FRAME}$	O	FRAME signal	Connects the $\overline{FRAME}$ signal for the burst MPX-IO interface.
	$\overline{WAIT}$	I	Wait	When active, inserts a wait cycle into the bus cycles during access to the external space.

Classification	Symbol	I/O	Name	Function
Direct memory access controller (DMAC)	$\overline{\text{DREQ0}}$, $\overline{\text{DREQ1}}$	I	DMA-transfer request	Input pin for external requests for DMA transfer.
	$\overline{\text{DACK0}}$, $\overline{\text{DACK1}}$	O	DMA-transfer request receive	Output pin for request receive, in response to external requests for DMA transfer.
	$\overline{\text{TEND0}}$	O	DMA-transfer end output	Output pin for DMA transfer end signal
User debugging interface (H-UDI)	TCK	I	Test clock	Test-clock input pin.
	TMS	I	Test mode select	Inputs the test-mode select signal.
	TDI	I	Test data input	Serial input pin for instructions and data.
	TDO	O	Test data output	Serial output pin for instructions and data.
	$\overline{\text{TRST}}$	I	Test reset	Initialization-signal input pin.
Advanced user debugger (AUD)	AUDATA3 to AUDATA0	O	AUD data	Data output pins in AUD-trace mode.
	AUDCK	O	AUD clock	Sync-clock output pin in AUD-trace mode.
	$\overline{\text{AUDSYNC}}$	O	AUD sync signal	Data start-position acknowledge-signal output pin in AIUD-trace mode.
E10A interface	$\overline{\text{ASEBRKAK}}$	O	Break mode acknowledge	Indicates that the E10A emulator has entered its break mode. For the connection with the E10A, see the SH7641 E10A Emulator User's Manual (tentative title).
	$\overline{\text{ASEMD0}}$	I	ASE mode	Sets the ASE mode.
I^2C bus interface 2	SCL	I/O	Serial clock pin	Serial clock input/output pin
	SDA	I/O	Serial data pin	Serial data input/output pin

Classification	Symbol	I/O	Name	Function
Multi function timer-pulse unit (MTU)	TCLKA TCLKB TCLKC TCLKD	I	Clock input	External clock input pins
	TIOC0A TIOC0B TIOC0C TIOC0D	I/O	Input capture/ output compare match	The TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
	TIOC1A TIOC1B	I/O	Input capture/ output compare match	The TGRA_1 to TGRB_1 input capture input/output compare output/PWM output pins.
	TIOC2A TIOC2B	I/O	Input capture/ output compare match	The TGRA_2 to TGRB_2 input capture input/output compare output/PWM output pins.
	TIOC3A TIOC3B TIOC3C TIOC3D	I/O	Input capture/ output compare match	The TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.
	TIOC4A TIOC4B TIOC4C TIOC4D	I/O	Input capture/ output compare	The TGRA_4 to TGRB_4 input capture input/output compare output/PWM output pins
Port output enable (POE)	$\overline{\text{POE3}}$ to $\overline{\text{POE0}}$	I	Port output enable	Request signal input to set the high current pins to the high impedance status
Serial communication interface with FIFO (SCIF)	SCK0 SCK1 SCK2	I/O	Serial clock	Clock input/output pins
	RxD0 RxD1 RxD2	I	Received data	Data input pins
	TxD0 TxD1 TxD2	O	Transmitted data	Data output pins
	$\overline{\text{RTS0}}$ $\overline{\text{RTS1}}$ $\overline{\text{RTS2}}$	I/O	Request to send	Request to send
	$\overline{\text{CTS0}}$ $\overline{\text{CTS1}}$ $\overline{\text{CTS2}}$	I/O	Clear to send	Clear to send

Classification	Symbol	I/O	Name	Function
USB function module	XVDATA	I	Data input	Input pin for receive data from USB differential receiver
	DPLS	I	D+ input	Input pin for D+ signal from USB receiver
	DMNS	I	D- input	Input pin for D- signal from USB receiver
	TXDPLS	O	D+ output	D+ transmit output pin to USB transceiver
	TXDMNS	O	D- output	D- transmit output pin to USB transceiver
	TXENL	O	Output enable	Output enable pin to USB transceiver
	VBUS	I	USB power supply monitor	USB cable connection monitor pin
	SUSPND	O	Suspend	USB transceiver suspend state output pin
	UCLK	I	USB clock	USB clock input pin (48 MHz input)
	DP	I/O	D+ input/output	Input/output pin for D+ signal to/from transceiver
	DM	I/O	D- input/output	Input/output pin for D- signal to/from transceiver
A/D converter	AN7 to AN0	I	Analog input pins	Analog input pins
	AVcc	I	Analog power supply for the A/D converter	Power supply pin for the A/D converter
	AVss	I	Analog ground for the A/D converter	The ground pin for the A/D converter.

Classification	Symbol	I/O	Name	Function
I/O ports	PTA14 to PTA0	I/O	General purpose port	15 bits general purpose input/output pins
	PTB8 to PTB0	I/O	General purpose port	9 bits general purpose input/output pins
	PTC15 to PTC0	I/O	General purpose port	16 bits general purpose input/output pins.
	PTD15 to PTD0	I/O	General purpose port	16 bits general purpose input/output pins
	PTE15 to PTE0	I/O	General purpose port	16 bits general purpose input/output pins
	PTF15 to PTF0	I/O	General purpose port	16 bits general purpose input/output pins
	PTG13 to PTG8	I/O	General purpose port	14 bits general purpose input/output and input pins
	PTG7 to PTG0	I		
	PTH14 to PTG0	I/O	General purpose port	15 bits general purpose input/output pins
	PTJ12 to PTG0	I/O	General purpose port	13 bits general purpose input/output pins

Section 2 CPU

2.1 Registers

This LSI has the same registers as the SH-3. In addition, this LSI also supports the same DSP-related registers as in the SH-DSP. The basic software-accessible registers are divided into four distinct groups:

- General registers
- Control registers
- System registers
- DSP registers

With the exception of some DSP registers, all of these registers are 32-bit width. The general registers are accessible, with R0 to R7 banked to provide access to a separate set of R0 to R7 registers (i.e. R0 to R7_BANK0, and R0 to R7_BANK1) depending on the value of the RB bit. The register bank (RB) bit in the status register (SR) defines which set of banked registers (R0 to R7_BANK0 or R0 to R7_BANK1) are accessed as general registers, and which are accessed only by LDC/STC instructions.

The control registers can be accessed by LDC/STC instructions. Control registers are:

- SR: Status register
- SSR: Saved status register
- SPC: Saved program counter
- GBR: Global base register
- VBR: Vector base register
- RS: Repeat start register (DSP mode only)
- RE: Repeat end register (DSP mode only)
- MOD: Modulo register (DSP mode only)

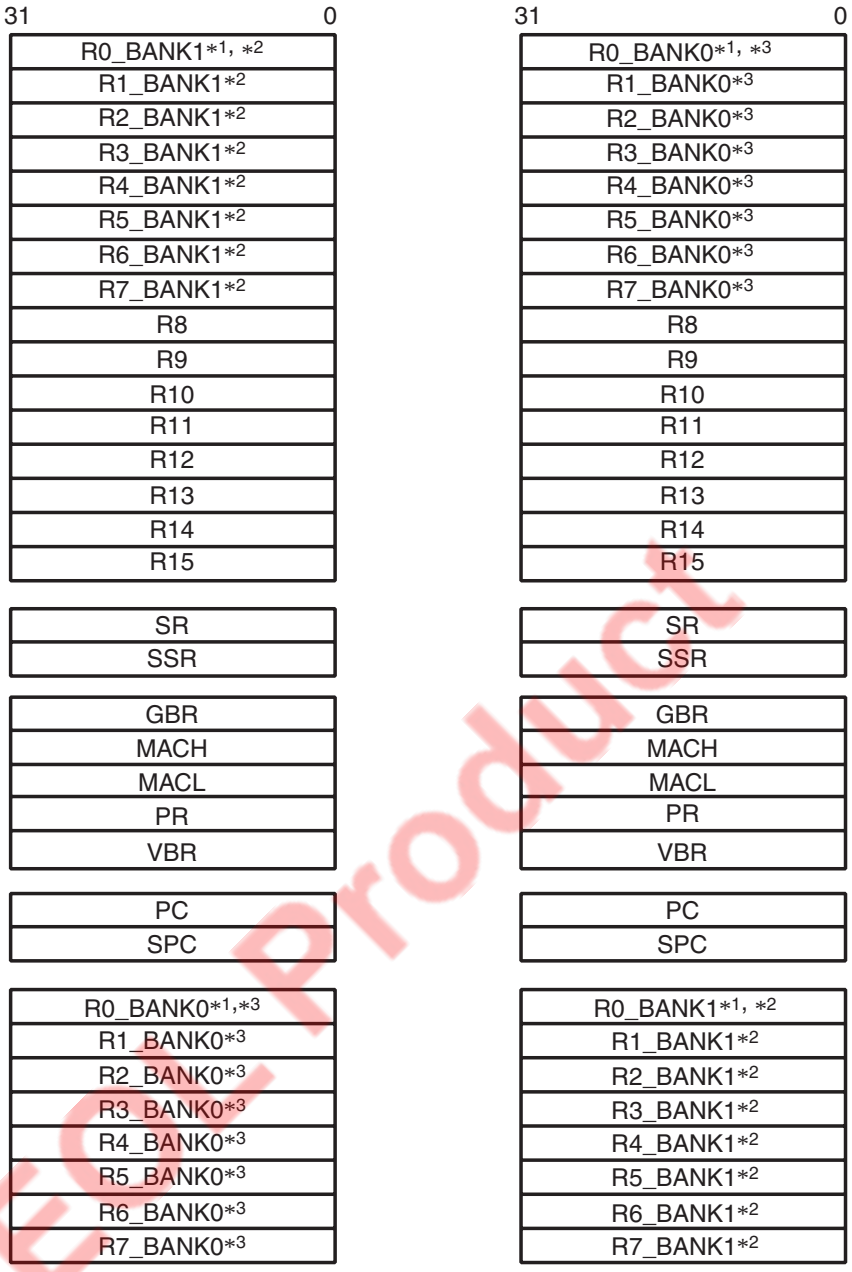
The system registers are accessed by the LDS/STS instructions (the PC is software-accessible, but is included here because its contents are saved in, and restored from, SPC in exception handling). The system registers are:

- MACH: Multiply and accumulate high register
- MACL: Multiply and accumulate low register
- PR: Procedure register
- PC: Program counter

This section explains the usage of these registers in different modes.

Figures 2.1 and 2.2 show the register configuration in each processing mode.

The DSP mode is switched by means of the DSP bit in the status register.

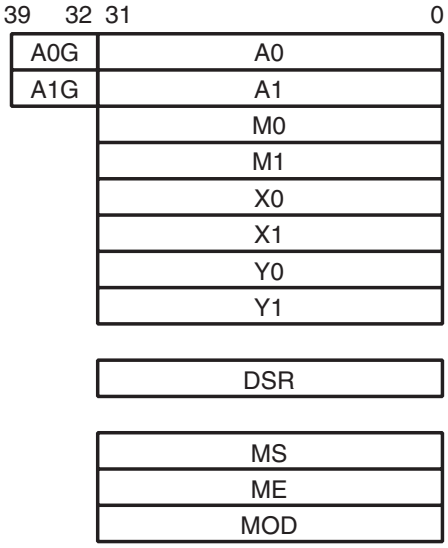


(a) Register configuration for DSP mode and non_DSP mode (RB = 1)

(b) Register configuration for DSP mode and non_DSP mode (RB = 0)

Notes: 1. The R0 register is used as an index register in indexed register indirect addressing mode and indexed GBR indirect addressing mode.
2. Bank register
Accessed as a general register when the RB bit is set to 1 in the SR register.
Accessed only by LDC/STC instructions when the RB bit is cleared to 0.
3. Bank register
Accessed as a general register when the RB bit is cleared to 0 in the SR register.
Accessed only by LDC/STC instructions when the RB bit is set to 1.

Figure 2.1 Register Configuration in Each Processing Mode (1)



(c) DSP mode register configuration (DSP = 1)

Figure 2.2 Register Configuration in Each Processing Mode (2)

Register values after a reset are shown in table 2.1.

Table 2.1 Initial Register Values

Type	Registers	Initial Value*
General registers	R0 to R15	Undefined
Control registers	SR	RB bit = 1, BL bit = 1, I3 to I0 = 1111 (H'F), The reserved bits other than bit 30 are all 0; bit 30 is 1, others undefined
	GBR, SSR, SPC	Undefined
	VBR	H'00000000
	RS, RE	Undefined
	MOD	Undefined
System registers	MACH, MACL, PR	Undefined
	PC	H'A0000000
DSP registers	A0, A0G, A1, A1G, M0, M1, X0, X1, Y0, Y1	Undefined
	DSR	H'00000000

Note: * Initialized by a power-on or manual reset.

2.1.1 General Registers

There are sixteen 32-bit general registers (Rn), designated R0 to R15. The general registers are used for data processing and address calculation.

With SuperH microcomputer type instructions, R0 is used as an index register. With a number of instructions, R0 is the only register that can be used.

With DSP type instructions, eight of the sixteen general registers are used for addressing of X and Y data memory and data memory (single data) that uses the L-bus.

To access X memory, R4 and R5 are used as the X address register [Ax] and R8 is used as the X index register [Ix]. To access Y memory, R6 and R7 are used as the Y address register [Ay] and R9 is used as the Y index register [Iy]. To access single data that uses the L-bus, R2, R3, R4, and R5 are used as the single data address register [As] and R8 is used as the single data index register [Is].

Figure 2.3 shows the general registers, which are identical to those of the SH3, when DSP extension is disabled.

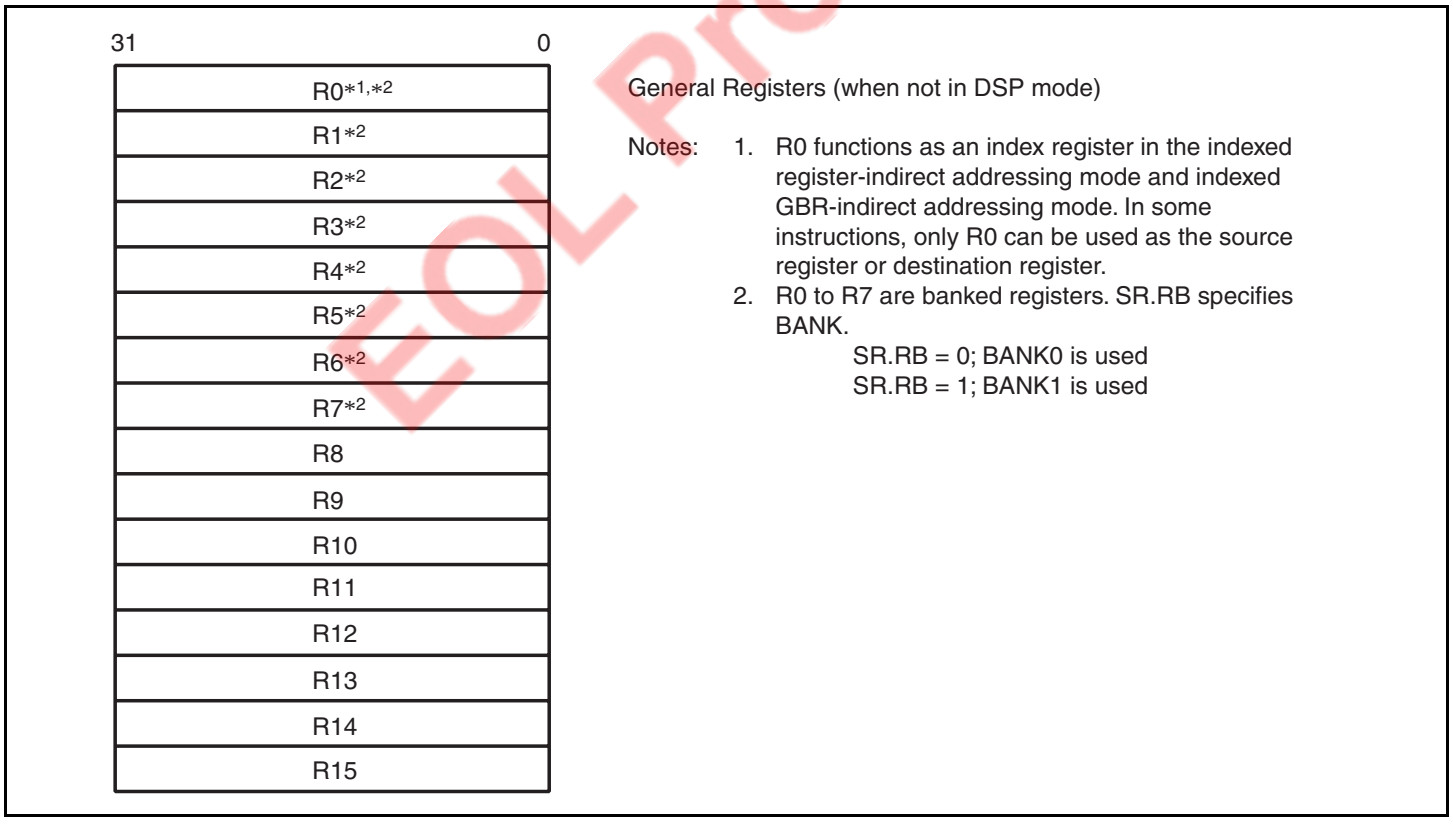


Figure 2.3 General Registers (Not in DSP Mode)

On the other hand, registers R2 to R9 are also used for DSP data address calculation when DSP extension is enabled (see figure 2.4). Other symbols that represent the purpose of the registers in DSP type instructions is shown in [ ].

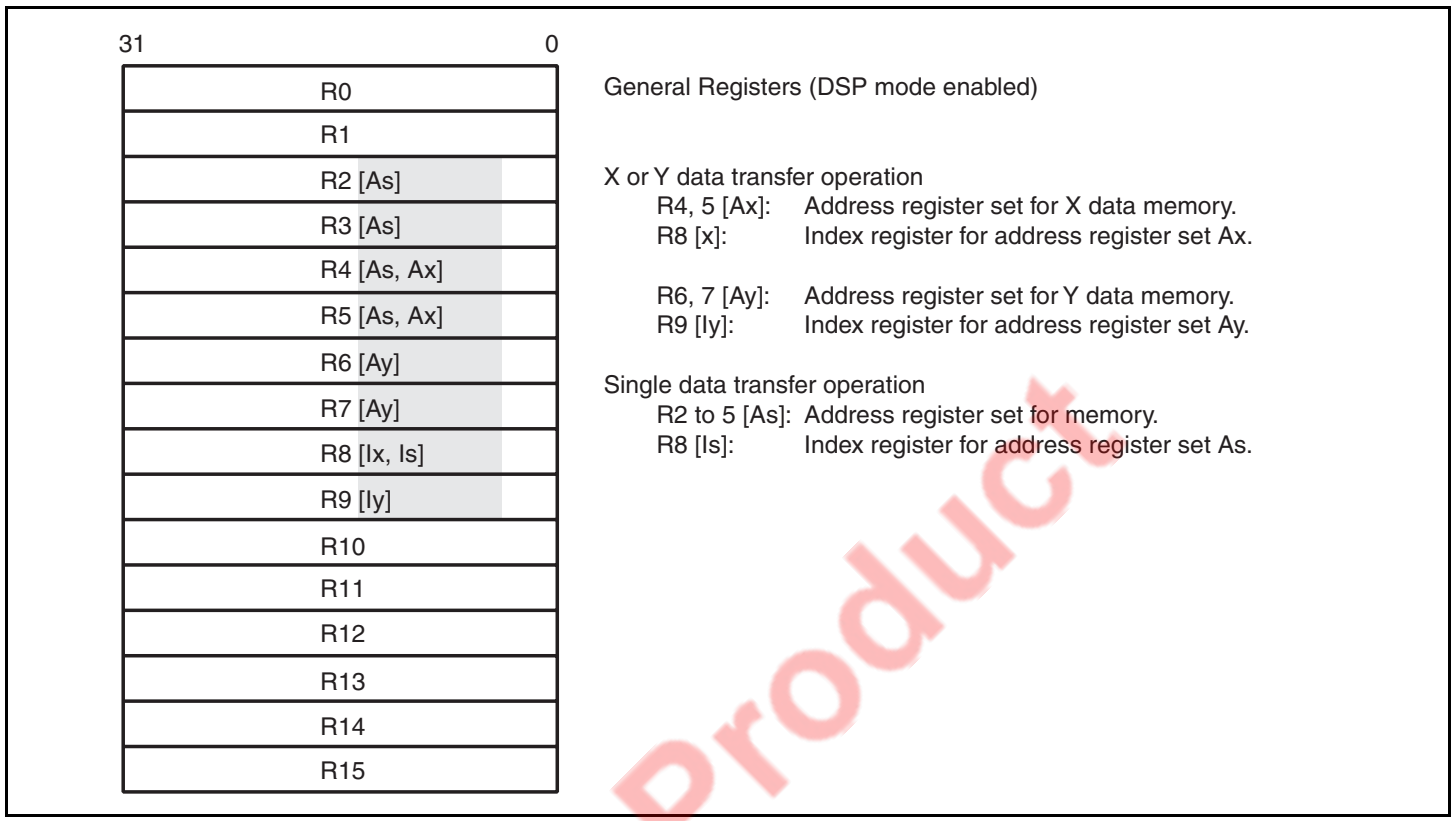


Figure 2.4 General Registers (DSP Mode)

DSP type instructions can access X and Y data memory simultaneously. To specify addresses for X and Y data memory, two address pointer sets are provided. These are:

R8[Ix], R4,5[Ax] for X memory access, and R9[Iy], R6,7[Ay] for Y memory access.

The symbols R2 to R9 are used by the assembler, but users can use other register names (aliases) that indicate the purpose of the register in the DSP instruction. The coding in assembler is as follows.

```
Ix:     .REG    (R8)
```

The name Ix is the alias for R8. Other aliases are as follows.

```
Ax0:    .REG    (R4)
Ax1:    .REG    (R5)
Ix:     .REG    (R8)
Ay0:    .REG    (R6)
```

```
Ay1:    .REG    (R7)
Iy:     .REG    (R9)
As0:    .REG    (R4)    ; This is optional, if another alias is required for single data transfer.
As1:    .REG    (R5)    ; This is optional, if another alias is required for single data transfer.
As2:    .REG    (R2)
As3:    .REG    (R3)
Is:     .REG    (R8)    ; This is optional, if another alias is required for single data transfer.
```

2.1.2 Control Registers

This LSI has 8 control registers: SR, SSR, SPC, GBR, VBR, RS, RE, and MOD (figure 2.5). SSR, SPC, GBR and VBR are the same as the SH-3 registers. The DSP mode is activated only when SR.DSP = 1.

Repeat start register RS, repeat end register RE, and repeat counter RC (12-bit part of SR) and repeat control bits RF0 and RF1 are new registers and control bits which are used for repeat control. Modulo register MOD and modulo control bits DMX and DMY in SR are also new register and control bits.

In SR, there are six additional control bits: RC11 to RC0, RF0, RF1, DMX, DMY and DSP. DMX and DMY are used for modulo addressing control. If DMX is 1, the modulo addressing mode is effective for the X memory address pointer, Ax (R4 or R5). If DMY is 1, the modulo addressing mode is effective for the Y memory address pointer, Ay (R6 or R7). However, both X and Y address pointers cannot be operated in modulo addressing mode even though both DMX and DMY bits are set. The case where DMX = DMY = 1 is reserved for future expansion. If both DMX and DMY are set simultaneously, the hardware will provisionally treat only the Y address pointer as the modulo addressing mode pointer. Modulo addressing is available for X and Y data transfer operations (MOVX and MOVY), but not for a single data transfer operation (MOVS).

RF1 and RF0 hold information on the number of repeat steps, and are set when a SETRC instruction is executed. When RF1 and RF0 = 00, the current repeat module consists of one instruction step. RF1 and RF0 = 01 means two instruction steps, RF1 and RF0 = 11 means three instruction steps, and RF1 and RF0 = 10 means the current repeat module consists of four or more instructions.

Although RC11 to RC0 and RF1 and RF0 can be changed by a store/load to SR, use of the dedicated manipulation instruction SETRC is recommended.

SR also has a 12-bit repeat counter, RC, which is used for efficient loop control. The repeat start register (RS) and repeat end register (RE) are also provided for loop control. They hold the start

and end addresses of a loop (the contents of the RS and RE registers are slightly different from the actual loop start and end addresses).

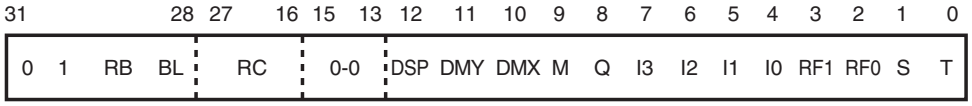
The modulo register, MOD, is provided to implement modulo addressing for circular data buffering. MOD holds the modulo start address (MS) and modulo end address (ME).

In order to access RS, RE and MOD, load/store (control register) instructions for these registers are provided. An example for RS is as follows:

```
LDC Rm,RS;    Rm -> RS
LDC.L @Rm+,RS;   (Rm) -> RS, Rm+4 -> Rm
STC RS,Rn;    RS -> Rn
STC.L RS,@-Rn;   Rn-4 -> Rn, RS -> (Rn)
```

Address set instructions for RS and RE are also provided.

```
LDRS @(disp,PC); disp × 2 + PC -> RS
LDRE @(disp,PC); disp × 2 + PC -> RE
```



RB bit: Register bank bit; used to define the general registers.
RB = 1: R0_BANK1 to R7_BANK1 are used as general registers.
R0_BANK0 to R7_BANK0 accessed by LDC/STC instructions.
RB = 0: R0_BANK0 to R7_BANK0 are used as general registers.
R0_BANK1 to R7_BANK1 accessed by LDC/STC instructions.

BL bit: Block bit; used to mask exception.
BL = 1: Interrupts are masked (not accepted)
BL = 0: Interrupts are accepted

RC [11:0]: 12-bit repeat counter

DSP bit: DSP operation mode
DSP = 1: DSP instructions (LDS Rm, DSR/A0/X0/X1/Y0/Y1,
LDS.L @Rm+, DSR/A0/X0/X1/Y0/Y1, STS DSR/A0/X0/X1/Y0/Y1, Rn,
STS.L DSR/A0/X0/X1/Y0/Y1, @–Rn, LDC Rm, RS/RE/MOD,
LDC.L @Rm+, RS/RE/MOD, STC RS/RE/MOD,Rn, STC.L RS/RE/MOD, @–Rn,
LDRS, LDRE, SETRC, MOVS, MOVX, MOVY, Pxxx) are enabled.
DSP = 0: All DSP instructions are treated as illegal instructions; only SH3 instructions are supported.

DMY bit: Modulo addressing enable for Y side

DMX bit: Modulo addressing enable for X side

Q, M bit: Used by DIV0U/S and DIV1 instructions.

I [3:0]: 4-bit field indicating the interrupt request mask level.

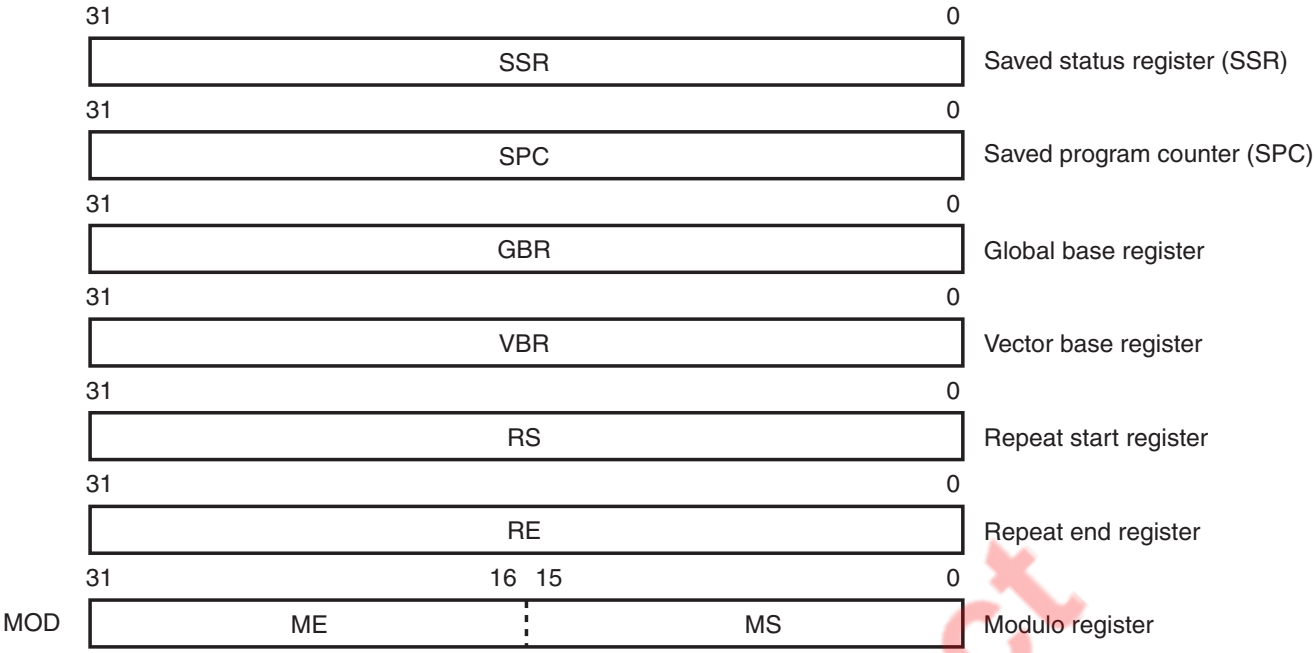
RF [1:0]: Used for repeat control

S bit: Used by the MAC instructions and DSP data.

T bit: The MOVT, CMP/cond, TAS, TST, BT, BF, SETT, CLRT and DT instructions use the T bit to indicate true (logic one) or false (logic zero). The ADDV/C, SUBV/C, DIV0U/S, DIV1, NEGC, SHAR/L, SHLR/L, ROTR/L and ROTCR/L instructions also use the T bit to indicate a carry, borrow, overflow, or underflow.

Reserved bits: A fixed value (either 0 or 1) is read from each of the bits. When writing, write the values shown in the above register. Operation is not guaranteed if a value other than that given above is written to the reserved bits.

Figure 2.5 Control Registers (1)



ME: Modulo end address, MS: Modulo start address

Saved status register (SSR)
Stores current SR value at time of exception to indicate processor status when returning to instruction stream from exception handler.

Saved program counter (SPC)
Stores current PC value at time of exception to indicate return address on completion of exception handling.

Global base register (GBR)
Stores base address of GBR-indirect addressing mode. The GBR-indirect addressing mode is used for data transfer and logical operations on the on-chip peripheral module register area.

Vector base register (VBR)
Stores base address of exception vector area.

Repeat start register (RS)
Used in DSP mode only. Indicates start address of repeat loop.

Repeat end register (RE)
Used in DSP mode only. Indicates address of repeat loop end.

Modulo register (MOD)
Used in DSP mode only.
MD[31:16]: ME: Modulo end address, MD[15:0]: Modulo start address.
In X/Y operand address generation, the CPU compares the address with ME, and if it is the same, loads MS in either the X or Y operand address register (depending on bits DMX and DMY in the SR register).

Figure 2.5 Control Registers (2)

2.1.3 System Registers

This LSI has four system registers, MACL, MACH, PR and PC (figure 2.6).

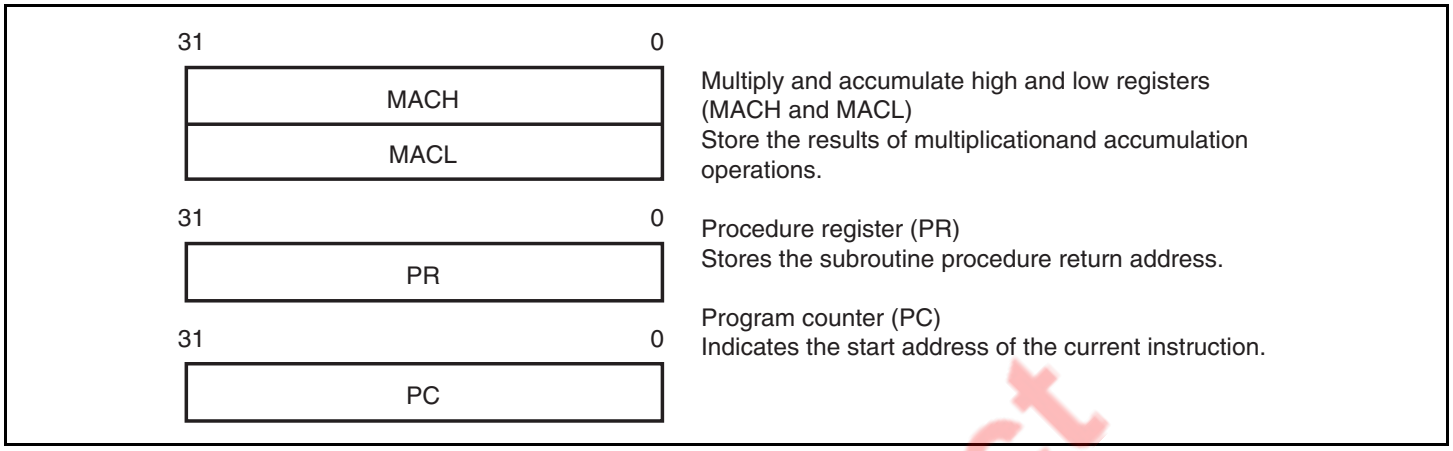


Figure 2.6 System Registers

The DSR, A0, X0, X1, Y0 and Y1 registers are also treated as system registers. Therefore, instructions for data transfer between general registers and system registers are supported for these registers.

2.1.4 DSP Registers

This LSI has eight data registers and one control register as DSP registers (figure 2.7). The data registers are 32-bit width with the exception of registers A0 and A1. Registers A0 and A1 include 8 guard bits (fields A0G and A1G), giving them a total width of 40 bits.

Three kinds of operation access the DSP data registers. The first is DSP data processing. When a DSP fixed-point data operation uses A0 or A1 as the source register, it uses the guard bits (bits 39 to 32). When it uses A0 or A1 as the destination register, guard bits 39 to 32 are valid. When a DSP fixed-point data operation uses a DSP register other than A0 or A1 as the source register, it sign-extends the source value to bits 39 to 32. When it uses one of these registers as the destination register, bits 39 to 32 of the result are discarded.

The second kind of operation is an X or Y data transfer operation, "MOVX.W" or "MOVY.W". This operation accesses the X and Y memories through the 16-bit X and Y data buses (figure 2.8). The register to be loaded or stored by this operation always comprises the upper 16 bits (bits 31 to 16). X0 or X1 can be the destination of an X memory load and Y0 or Y1 can be the destination of a Y memory load, but no other register can be the destination register in this operation.

When data is read into the upper 16 bits of a register (bits 31 to 16), the lower 16 bits of the register (bits 15 to 0) are automatically cleared. A0 and A1 can be stored in the X or Y memory by this operation, but no other registers can be stored.

The third kind of operation is a single-data transfer instruction, "MOVS.W" or "MOVS.L". These instructions access any memory location through the LDB (figure 2.8). All DSP registers connect to the LDB and can be the source or destination register of the data transfer. These instructions have word and longword access modes. In word mode, registers to be loaded or stored by this instruction comprise the upper 16 bits (bits 31 to 16) for DSP registers except A0G and A1G. When data is loaded into a register other than A0G and A1G in word mode, the lower half of the register is cleared. When A0 or A1 is used, the data is sign-extended to bits 39 to 32 and the lower half is cleared. When A0G or A1G is the destination register in word mode, data is loaded into an 8-bit register, but A0 or A1 is not cleared. In longword mode, when the destination register is A0 or A1, it is sign-extended to bits 39 to 32.

Tables 2.2 and 2.3 show the data type of registers used in DSP instructions. Some instructions cannot use some registers shown in the tables because of instruction code limitations. For example, PMULS can use A1 as the source register, but cannot use A0. These tables ignore details of register selectability.

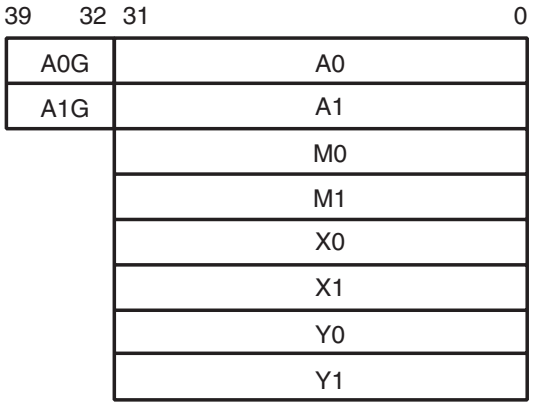
Table 2.2 Destination Register in DSP Instructions

Registers		Instructions	Guard Bits 39 32	Register Bits 31 16	15 0
A0, A1	DSP	Fixed-point, PSHA, PMULS	Sign-extended	40-bit result	
		Integer, PDMSB	Sign-extended	24-bit result	Cleared
		Logical, PSHL	Cleared	16-bit result	Cleared
	Data transfer	MOVS.W	Sign-extended	16-bit data	Cleared
		MOVS.L	Sign-extended	32-bit data	
A0G, A1G	Data transfer	MOVS.W	Data	No update	
		MOVS.L	Data	No update	
X0, X1 Y0, Y1 M0, M1	DSP	Fixed-point, PSHA, PMULS		32-bit result	
		Integer, logical, PDMSB, PSHL		16-bit result	Cleared
	Data transfer	MOVX/Y.W, MOVS.W		16-bit result	Cleared
		MOVS.L		32-bit data	

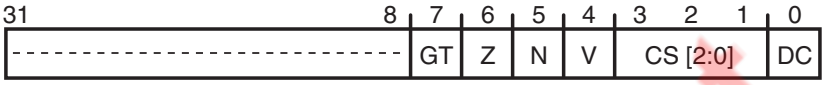
Table 2.3 Source Register in DSP Operations

Registers		Instructions	Guard Bits 39 32	Register Bits 31 16	15 0
A0, A1	DSP	Fixed-point, PDMSB, PSHA		40-bit data	
		Integer		24-bit data	
		Logical, PSHL, PMULS		16-bit data	
	Data transfer	MOVX/Y.W, MOVS.W		16-bit data	
		MOVS.L		32-bit data	
A0G, A1G	Data transfer	MOVS.W	Data		
		MOVS.L	Data		
X0, X1 Y0, Y1 M0, M1	DSP	Fixed-point, PDMSB, PSHA	Sign*	32-bit data	
		Integer	Sign*	16-bit data	
		Logical, PSHL, PMULS		16-bit data	
	Data transfer	MOVS.W		16-bit data	
		MOVS.L		32-bit data	

Note: * The data is sign-extended and input to the ALU.



(a) DSP Data Registers



(b) DSP Status Register (DSR)

Reset status
DSR: All zeros
Others: Undefined

Figure 2.7 DSP Registers

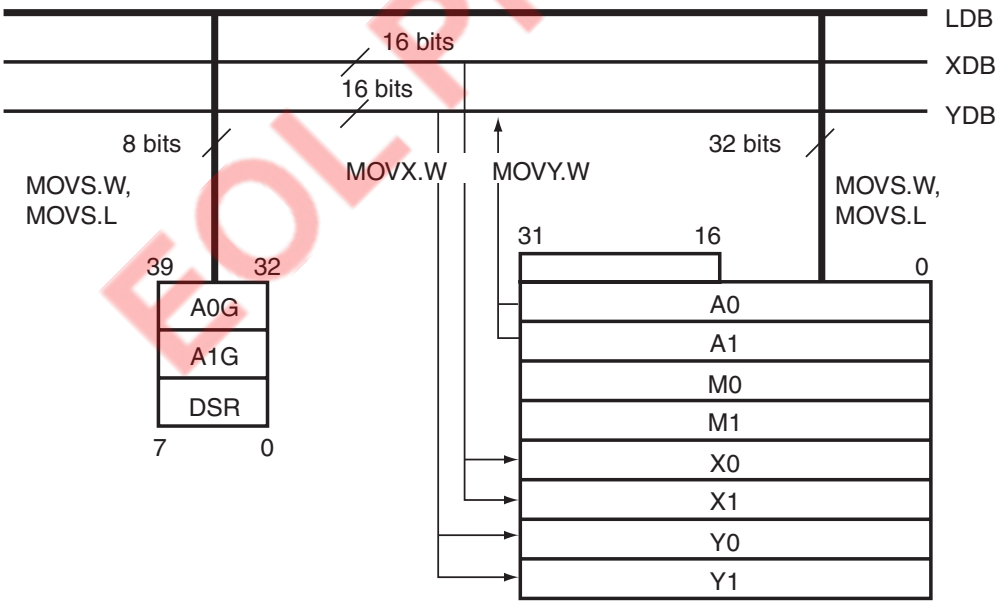


Figure 2.8 Connections of DSP Registers and Buses

The DSP unit has one control register, the DSP status register (DSR). DSR holds the status of DSP data operation results (zero, negative, and so on) and has a DC bit which is similar to the T bit in the CPU. The DC bit indicates one of the status flags. A DSP data processing instruction controls its execution based on the DC bit. This control affects only the operations in the DSP unit; it controls the update of DSP registers only. It cannot control operations in the CPU, such as address register updating and load/store operations. Control bits CS2 to CS0 specify the condition to be reflected in the DC bit.

Unconditional DSP type data operations, except PMULS, MOVX, MOVY and MOVS, update the condition flags and DC bit, but no CPU instructions, including MAC instructions, update the DC bit. Conditional DSP type instructions do NOT update DSR either.

Table 2.4 DSR Register Bits

Bits	Name (Abbreviation)	Function
31 to 8	Reserved bits	0: Always read as 0; always use 0 as the write value
7	Signed Greater Than bit (GT)	Indicates that the operation result is positive (except 0), or that operand 1 is greater than operand 2 1: Operation result is positive, or operand 1 is greater than operand 2
6	Zero bit (Z)	Indicates that the operation result is zero (0), or that operand 1 is equal to operand 2 1: Operation result is zero (0), or operands are equal
5	Negative bit (N)	Indicates that the operation result is negative, or that operand 1 is smaller than operand 2 1: Operation result is negative, or operand 1 is smaller than operand 2
4	Overflow bit (V)	Indicates that the operation result has overflowed 1: Operation result has overflowed
3 to 1	Condition Select bits (CS)	Designate the mode for selecting the operation result status to be set in the DC bit Do not set these bits to 110 or 111 000: Carry/borrow mode 001: Negative value mode 010: Zero mode 011: Overflow mode 100: Signed greater mode 101: Signed greater than or equal to mode
0	DSP Condition bit (DC)	Sets the status of the operation result in the mode designated by the CS bits 0: Designated mode status has not occurred (false) 1: Designated mode status has occurred

Note: After execution of a PADDC/PSUBC instruction, the DC bit sets the status of the operation result in carry/borrow mode regardless of the CS bits.

DSR is assigned as a system register and the following load/store instructions are provided:

```
STS DSR,Rn;
STS.L DSR,@-Rn;
LDS Rn,DSR;
LDS.L @Rn+,DSR;
```

When DSR is read by an STS instruction, the upper bits (bits 31 to 8) are all 0.

2.2 Data Formats

2.2.1 Register Data Format (Non-DSP Type)

Register operands are always longwords (32 bits) (figure 2.9). When the memory operand is only a byte (8 bits) or a word (16 bits), it is sign-extended into a longword when loaded into a register.

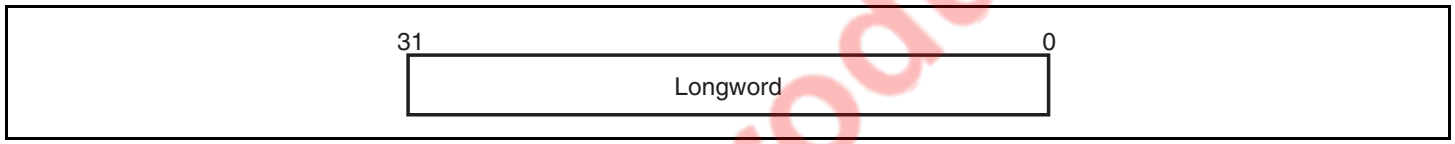


Figure 2.9 Longword Operand

2.2.2 DSP-Type Data Formats

This LSI has several different data formats that depend on the instruction. This section explains the data formats for DSP type instructions.

Figure 2.10 shows three DSP-type data formats with different binary point positions. A CPU-type data format with the binary point to the right of bit 0 is also shown for reference.

The DSP-type fixed point data format has the binary point between bit 31 and bit 30. The DSP-type integer format has the binary point between bit 16 and bit 15. The DSP-type logical format does not have a binary point. The valid data lengths of the data formats depend on the instruction and the DSP register.

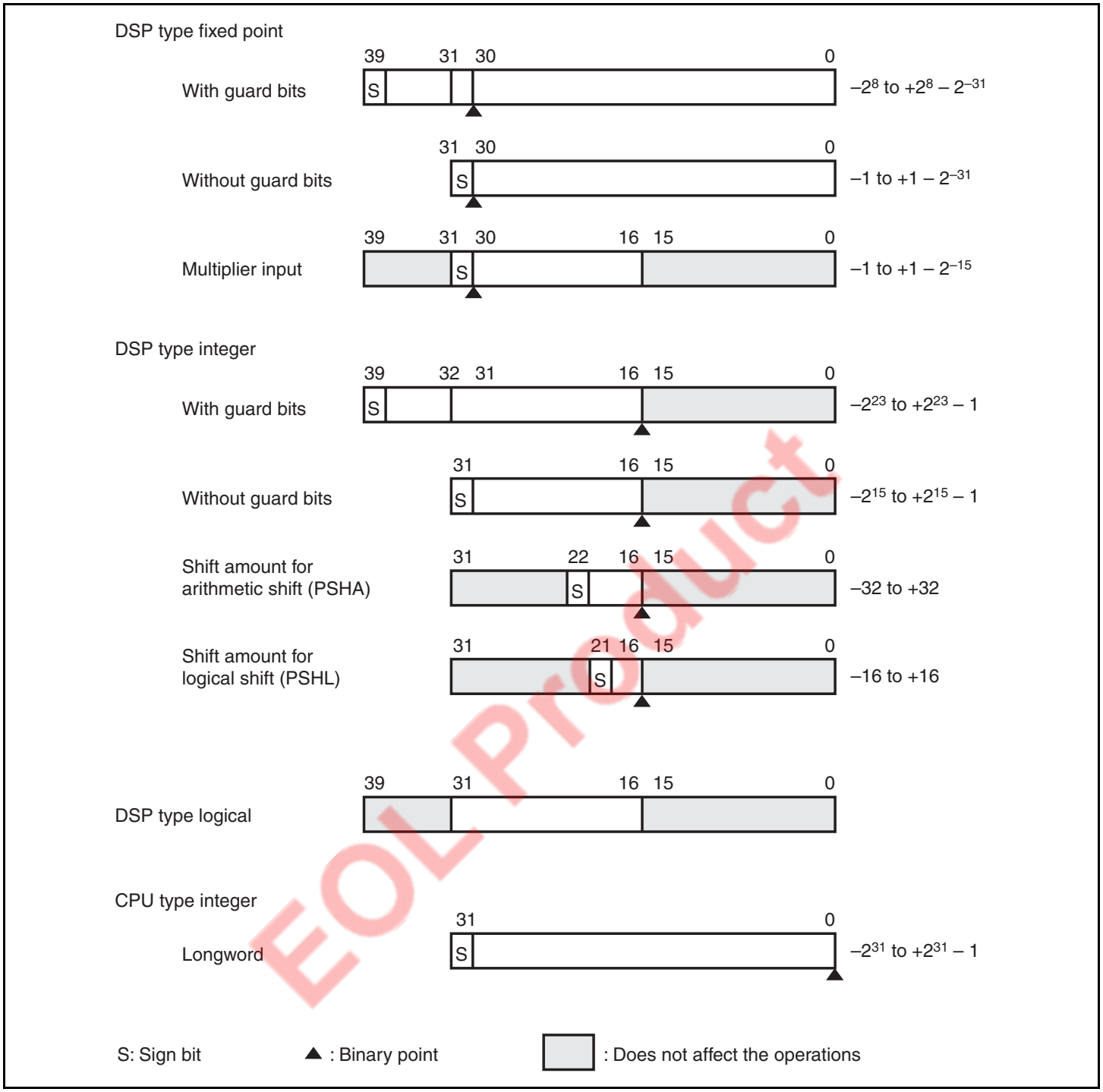


Figure 2.10 Data Formats

The shift amount for the arithmetic shift (PSHA) instruction has a 7-bit field that can represent values from –64 to +63, but –32 to +32 are valid numbers for the instruction. Also the shift amount for a logical shift operation has a 6-bit field, but –16 to +16 are valid numbers for the instruction.

2.2.3 Memory Data Formats

Memory data formats are classified into byte, word, and longword. Byte data can be accessed from any address, but an address error will occur if word data starting from an address other than 2n or longword data starting from an address other than 4n is accessed. In such cases, the data accessed cannot be guaranteed (figure 2.11).

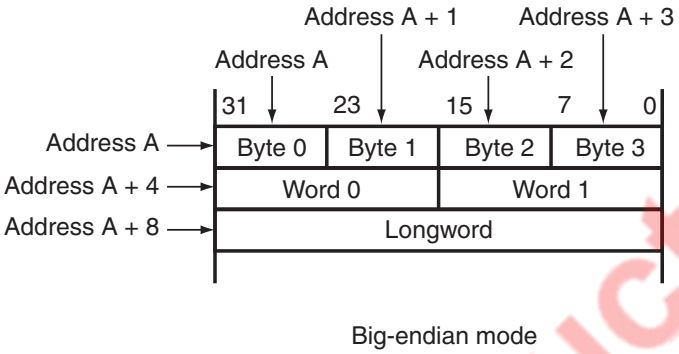


Figure 2.11 Byte, Word, and Longword Alignment

2.3 Features of CPU Core Instructions

The CPU core instructions are RISC-type instructions with the following features:

Fixed 16-Bit Length: All instructions have a fixed length of 16 bits. This improves program code efficiency.

One Instruction per State: Pipelining is used, and basic instructions can be executed in one state.

Data Size: The basic data size for operations is longword. Byte, word, or longword can be selected as the memory access size. Memory byte or word data is sign-extended and operated on as longword data. Immediate data is sign-extended to longword size for arithmetic operations or zero-extended to longword size for logical operations.

Table 2.5 Word Data Sign Extension

This LSI's CPU		Description	Example of Other CPU	
MOV.W	@(disp,PC),R1	Sign-extended to 32 bits, R1 becomes H'00001234, and is then operated on by the ADD instruction.	ADD.W	#H'1234,R0
ADD	R1,R0			
.....				
.DATA.W	H'1234			

Note: Immediate data is referenced by @(disp,PC).

Load/Store Architecture: Basic operations are executed between registers. In operations involving memory, data is first loaded into a register (load/store architecture). However, bit manipulation instructions such as AND are executed directly on memory.

Delayed Branching: Unconditional branch instructions, etc., are executed as delayed branches. With a delayed branch instruction, the branch is made after execution of the instruction (called the slot instruction) immediately following the delayed branch instruction. This minimizes disruption of the pipeline when a branch is made.

With a delayed branch, the actual branch operation occurs after execution of the slot instruction. However, instruction execution for register updating, etc., excluding the branch operation, is performed in delayed branch instruction → delay slot instruction order. For example, even though the contents of the register holding the branch destination address are changed in the delay slot, the branch destination address remains as the register contents prior to the change.

Table 2.6 Delayed Branch Instructions

This LSI's CPU		Description	Example of Other CPU	
BRA	TRGET	ADD is executed before branch to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

Multiply/Multiply-and-Accumulate Operations: A $16 \times 16 \rightarrow 32$ multiply operation is executed in 1 to 2 states, and a $16 \times 16 + 64 \rightarrow 64$ multiply-and-accumulate operation in 2 states. A $32 \times 32 \rightarrow 64$ multiply operation and a $32 \times 32 + 64 \rightarrow 64$ multiply-and-accumulate operation are each executed in 2 to 3 states.

T Bit: The result of a comparison is indicated by the T bit in the status register (SR), and a conditional branch is performed according to whether the result is True or False. Processing speed has been improved by keeping the number of instructions that modify the T bit to a minimum.

Table 2.7 T Bit

This LSI's CPU		Description	Example of Other CPU	
CMP/GE	R1,R0	If R0 ≥ R1, the T bit is set.	CMP.W	R1,R0
BT	TRGET0	A branch is made to TRGET0 if R0 ≥ R1, or to TRGET1 if R0 < R1.	BGE	TRGET0
BF	TRGET1		BLT	TRGET1
ADD	#–1,R0	The T bit is not set by ADD.	SUB.W	#1,R0
CMP/EQ	#0,R0	If R0 = 0, the T bit is set.	BEQ	TRGET
BT	TRGET	A branch is made if R0 = 0.		

Immediate Data: Byte immediate data is placed inside the instruction code. Word and longword immediate data is not placed inside the instruction code, but in a table in memory. The table in memory is referenced with an immediate data transfer instruction (MOV) using PC-relative addressing mode with displacement.

Table 2.8 Immediate Data Referencing

Type	This LSI's CPU		Example of Other CPU	
8-bit immediate	MOV	#H'12,R0	MOV.B	#H'12,R0
16-bit immediate	MOV.W	@(disp,PC),R0	MOV.W	#H'1234,R0
	.....			
	.DATA.W	H'1234		
32-bit immediate	MOV.L	@(disp,PC),R0	MOV.L	#H'12345678,R0
	.....			
	.DATA.L	H'12345678		

Note: Immediate data is referenced by @(disp,PC).

Absolute Addresses: When data is referenced by an absolute address, the absolute address value is placed in a table in memory beforehand. Using the method whereby immediate data is loaded when an instruction is executed, this value is transferred to a register and the data is referenced using register indirect addressing mode.

Table 2.9 Absolute Address Referencing

Type	This LSI's CPU		Example of Other CPU	
Absolute address	MOV.L	@(disp,PC),R1	MOV.B	@H'12345678,R0
	MOV.B	@R1,R0		
		.....		
	.DATA.L	H'12345678		

16-Bit/32-Bit Displacement: When data is referenced with a 16- or 32-bit displacement, the displacement value is placed in a table in memory beforehand. Using the method whereby immediate data is loaded when an instruction is executed, this value is transferred to a register and the data is referenced using indexed register indirect addressing mode.

Table 2.10 Displacement Referencing

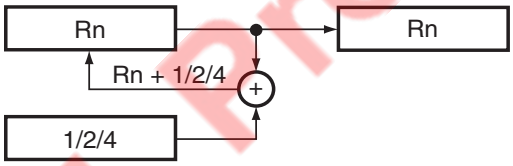
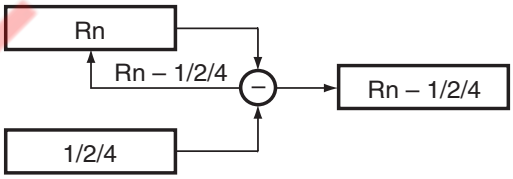
Type	This LSI's CPU		Example of Other CPU	
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W	@(H'1234,R1),R2
	MOV.W	@(R0,R1),R2		
		.....		
	.DATA.W	H'1234		

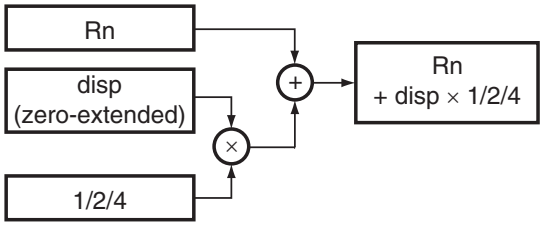
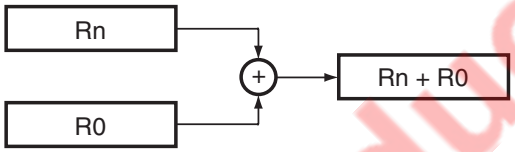
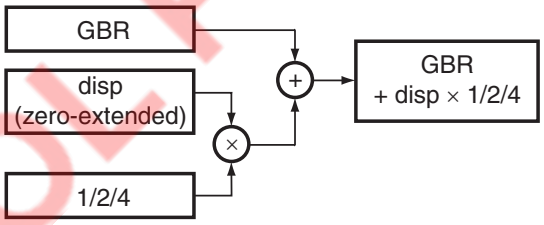
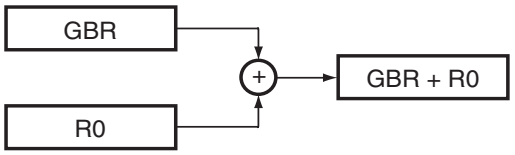
2.4 Instruction Formats

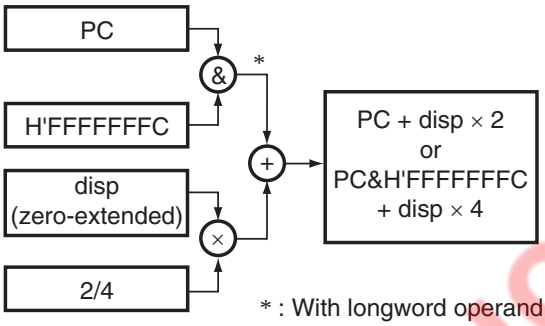
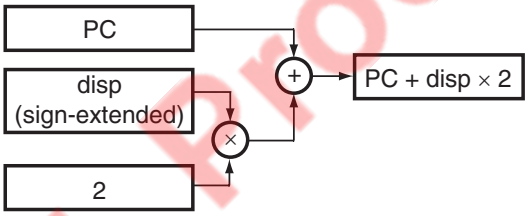
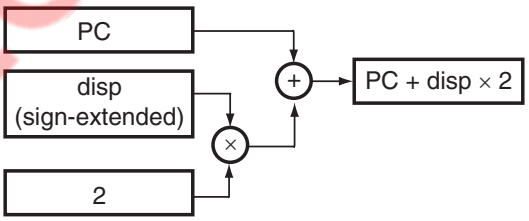
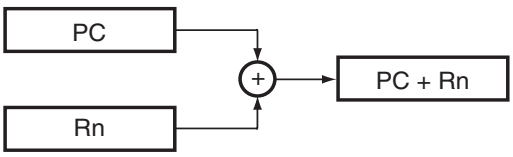
2.4.1 CPU Instruction Addressing Modes

The following table shows addressing modes and effective address calculation methods for instructions executed by the CPU core.

Table 2.11 Addressing Modes and Effective Addresses for CPU Instructions

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register direct	Rn	Effective address is register Rn. (Operand is register Rn contents.)	—
Register indirect	@Rn	Effective address is register Rn contents. 	Rn
Register indirect with post-increment	@Rn+	Effective address is register Rn contents. A constant is added to Rn after instruction execution: 1 for a byte operand, 2 for a word operand, 4 for a longword operand. 	Rn After instruction execution Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn
Register indirect with pre-decrement	@–Rn	Effective address is register Rn contents. It is decremented by a constant beforehand: 1 for a byte operand, 2 for a word operand, 4 for a longword operand. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn (Instruction executed with Rn after calculation)

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

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
PC-relative with displacement	@(disp:8, PC)	Effective address is PC with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 2 (word) or 4 (longword), according to the operand size. With a longword operand, the lower 2 bits of PC are masked.  * : With longword operand	Word: PC + disp × 2 Longword: PC&H'FFFFFFFC + disp × 4
PC-relative	disp:8	Effective address is PC with 8-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + disp × 2
	disp:12	Effective address is PC with 12-bit displacement disp added after being sign-extended and multiplied by 2 	PC + disp × 2
	Rn	Effective address is sum of PC and Rn. 	PC + Rn

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

2.4.2 DSP Data Addressing

Two different memory accesses are made with DSP instructions. The two kinds of instructions are X and Y data transfer instructions (MOVX.W and MOVY.W) and single data transfer instructions (MOVS.W and MOVSL). The data addressing is different for these two kinds of instructions. An overview of the data transfer instructions is given in table 2.12.

Table 2.12 Overview of Data Transfer Instructions

	X/Y Data Transfer Processing (MOVX.W, MOVY.W)	Single Data Transfer Processing (MOVS.W, MOVS.L)
Address register	Ax: R4, R5, Ay: R6, R7	As: R2, R3, R4, R5
Index register	Ix: R8, Iy: R9	Is: R8
Addressing	Nop/Inc (+2)/index addition: post-increment	Nop/Inc (+2, +4)/index addition: post-increment
	—	Dec (–2, –4): pre-decrement
Modulo addressing	Possible	Not possible
Data bus	XDB, YDB	LDB
Data length	16 bits (word)	16/32 bits (word/longword)
Bus contention	No	Yes
Memory	X/Y data memory	Entire memory space
Source register	Dx, Dy: A0, A1	Ds: A0/A1, M0/M1, X0/X1, Y0/Y1, A0G, A1G
Destination register	Dx: X0/X1, Dy: Y0/Y1	Ds: A0/A1, M0/M1, X0/X1, Y0/Y1, A0G, A1G

X/Y Data Addressing: With DSP instructions, the X and Y data memory can be accessed simultaneously using the MOVX.W and MOVY.W instructions. Two address pointers are provided for DSP instructions to enable simultaneous access to X and Y data memory. Only pointer addressing can be used with DSP instructions; immediate addressing is not available. Address registers are divided into two, with register R4 or R5 functioning as the X memory address register (Ax), and register R6 or R7 as the Y memory address register (Ay). The following three kinds of addressing can be used with X and Y data transfer instructions.

1. Non-update address register addressing:

   The Ax and Ay registers are address pointers. They are not updated.
2. Addition index register addressing:

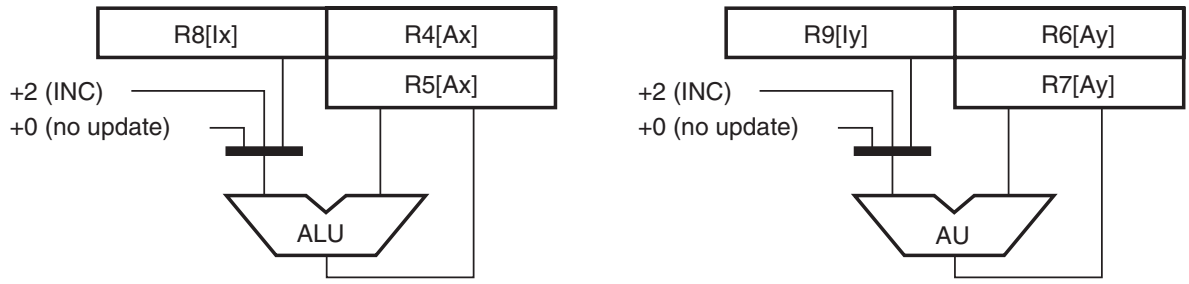
   The Ax and Ay registers are address pointers. After a data transfer, the value of the Ix or Iy register is added to each (post-increment).
3. Increment address register addressing:

   The Ax and Ay registers are address pointers. After a data transfer, they are each incremented by 2 (post- increment).

There is an index register for each address pointer. The R8 register is the index register (Ix) for the X memory address register (Ax), and the R9 register is the index register (Iy) for the Y memory address register (Ay).

The X and Y data transfer instructions perform word-length processing, and use 16-bit access to the X/Y data memory. A value of 2 is therefore added to the address register in the increment processing. To perform decrementing, –2 is set in the index register and addition index register addressing is specified. In X/Y data addressing, only bits 1 to 15 of the address pointer are valid. When using X/Y data addressing, 0 must always be written to bit 0 of the address pointer and index register.

X/Y data transfer addressing is shown in figure 2.12. When accessing X and Y memory using the X and Y buses, the upper word of Ax (R4 or R5) and Ay (R6 or R7) is ignored. The result of @AY+ or @Ay+Iy is stored in the lower word of Ay, while the upper word retains its original value.



[Legend]
AU: Adder provided for DSP addressing

Note: Three address processing methods:
1. Increment
2. Index register addition (Ix/Iy)
3. No increment

Post-updating is used in all cases.
The address pointer can be decremented by setting in the index register.

Figure 2.12 X and Y Data Transfer Addressing

Single Data Addressing: DSP instructions include two single data transfer instructions (MOVS.W and MOVS.L) that load data into, or store data from, a DSP register. With these instructions, one of registers R2 to R5 is used as the single data transfer address register (As).

The following four kinds of addressing can be used with single data transfer instructions.

1. Non-update address register addressing:
   The As register is an address pointer. It is not updated.
2. Addition index register addressing:
   The As register is an address pointer. After a data transfer, the value of the Is register is added to the As register (post-increment).
3. Increment address register addressing:
   The As register is an address pointer. After a data transfer, the As register is incremented by 2 or 4 (post-increment).
4. Decrement address register addressing:
   The As register is an address pointer. Before a data transfer, –2 or –4 is added to the As register (i.e. 2 or 4 is subtracted) (pre-decrement).

The R8 register is the index register (Is) for the address pointer (As). Single data transfer addressing is shown in figure 2.13.

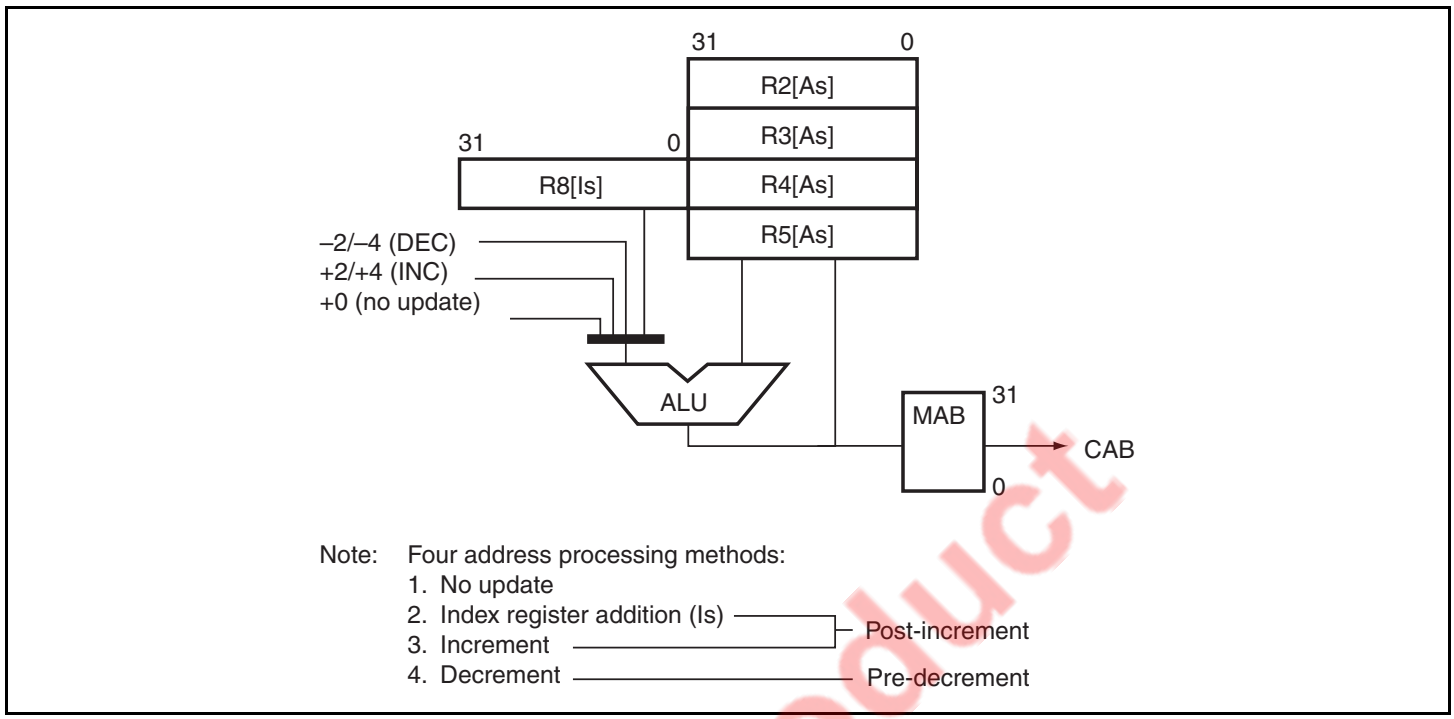


Figure 2.13 Single Data Transfer Addressing

Modulo Addressing: Like other DSPs, this LSI has a modulo addressing mode. Address registers are updated in the same way in this mode. When the address pointer value reaches the preset modulo end address, the address pointer value becomes the modulo start address.

Modulo addressing is only available for the X and Y data transfer instructions (MOVX.W and MOVY.W). Modulo addressing mode is specified for the X address register by setting the DMX bit in the SR register, and for the Y address register by setting the DMY bit. Modulo addressing is valid for either the X or the Y address register, only; it cannot be set for both at the same time. Therefore, DMX and DMY cannot both be set simultaneously. If they are, only the DMY setting will be valid.

The MOD register is provided to set the start and end addresses of the modulo address area. The MOD register contains MS (Modulo Start) and ME (Modulo End). An example of the use of the MOD register (MS and ME fields) is shown below.

```
            MOV.L ModAddr,Rn;     Rn=ModEnd, ModStart
            LDC Rn,MOD;           ME=ModEnd, MS=ModStart
ModAddr:    .DATA.W               mEnd;      ModEnd
            .DATA.W               mStart;    ModStart

ModStart:   .DATA
              :
ModEnd:     .DATA
```

The start and end addresses are specified in MS and ME, then the DMX or DMY bit is set to 1.

When the X/Y data transfer instruction set in DMX/DMY is executed, the address register contents before update are compared with ME*[1]. If they match, modulo start address MS is stored in the address register as the updated value*[2]. If non-update address register addressing is specified for the X/Y data transfer instruction, the address pointer will not return to modulo start address MS even though the address register contents match ME.

Notes: 1. Bits 1 to 15 of the address register are used for comparison. Though ME retains its previous value for bit 0, 0 must always be written to bit 0.
2. The MS value is stored in bits 1 to 15 of the address register. Though MS retains its previous value for bit 0, 0 must always be written to bit 0.

The maximum modulo size is 64-kbytes. This is sufficient to access the X and Y data memory. A block diagram of modulo addressing is shown in figure 2.14.

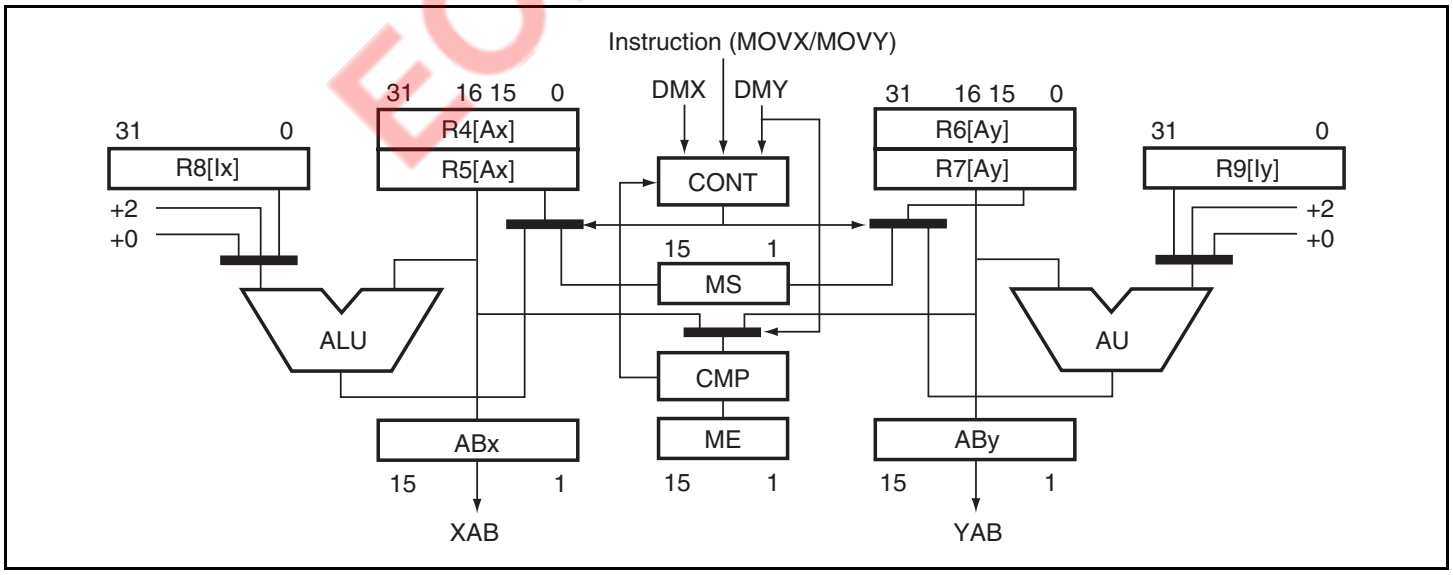


Figure 2.14 Modulo Addressing

An example of modulo addressing is given below.

```
MS = H'7000; ME=H'7004; R4=H'A50070008;
DMX = 1; DMY = 0: (Modulo addressing setting for address register Ax)
```

As a result of the above settings, the R4 register changes as follows.

```
; R4: H'A5007000    (Initial value)
; R4: H'A5007000 -> H'A5007002
; R4: H'A5007002 -> H'A5007004
; R4: H'A5007004 -> H'A5007000 (After reading H'A5007004, MS value is written to
                                address register)
; R4: H'A5007000 -> H'A5007002
```

Place the data so that the upper 16 bits of the modulo start and end addresses are the same. This is because the modulo start address overwrites only the lower 16 bits of the address register.

Note: When addition index is the data addressing type for X and Y data transfer instructions, the address pointer may exceed the ME value without actually reaching it. In this case, the address pointer will not return to the modulo start address. Not only with modulo addressing, but when X and Y data addressing is used, bit 0 is ignored. 0 must always be written to bit 0 of the address pointer, index register, MS, and ME.

DSP Addressing Operations: DSP addressing operations in the pipeline execution stage (EX), including modulo addressing, are shown below.

```
if ( Operation is MOVX.W MOVY.W ) {
   ABx=Ax; ABy=Ay;
   /* memory access cycle uses ABx and ABy. The addresses to be used have not been updated */

   /* Ax is one of R4,5 */
   if ( DMX==0 || DMX==1 && DMY == 1 )} Ax=Ax+(+2 or R8[Ix] or +0);
   /* Inc,Index,Not-Update */
   else if (! not-update) Ax=modulo( Ax, (+2 or R8[Ix]) );

   /* Ay is one of R6,7 */
   if ( DMY==0 ) Ay=Ay+(+2 or R9[Iy] or +0); /* Inc,Index,Not-Update */
   else if (! not-update) Ay=modulo( Ay, (+2 or R9[Iy]) );
}
else if ( Operation is MOVS.W or MOVS.L ) {
   if ( Addressing is Nop, Inc, Add-index-reg ) {
      MAB=As;
      /* memory access cycle uses MAB. The address to be used has not been updated */
      /* As is one of R2 to R5 */
      As=As+(+2 or +4 or R8[Is] or +0); /* Inc,Index,Not-Update */
   else { /* Decrement, Pre-update */
      /* As is one of R2 to R5 */
      As=As+(-2 or -4);
      MAB=As;
      /* memory access cycle uses MAB. The address to be used has been updated */
}

/* The value to be added to the address register depends on addressing operations.
For example, (+2 or R8[Ix] or +0) means that
       +2 : if operation is increment
   R8[Ix] : if operation is add-index-reg
       +0 : if operation is not-update
*/

function modulo ( AddrReg, Index ) {
   if ( AdrReg[15:0]==ME ) AdrReg[15:0]==MS;
   else AdrReg=AdrReg+Index;
   return AddrReg;
}
```

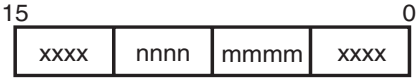
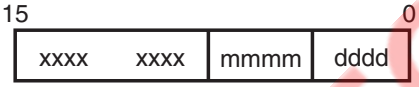
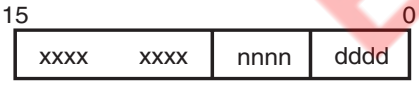
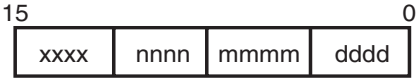
2.4.3 CPU Instruction Formats

Table 2.13 shows the instruction formats, and the meaning of the source and destination operands, for instructions executed by the CPU core. The meaning of the operands depends on the instruction code. The following symbols are used in the table.

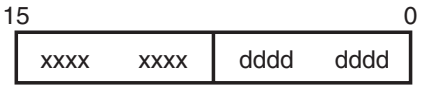
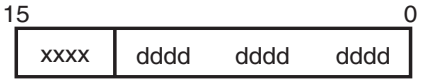
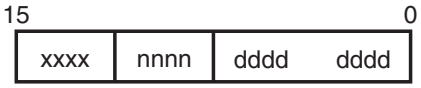
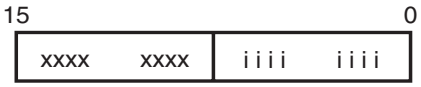
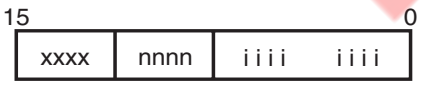
xxxx: Instruction code
mmmm: Source register
nnnn: Destination register
iiii: Immediate data
dddd: Displacement

Table 2.13 CPU Instruction Formats

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

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

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

Instruction Format	Source Operand	Destination Operand	Sample Instruction
d type 	dddddddd: GBR indirect with displacement	R0 (register direct)	MOV.L @(disp,GBR),R0
	R0 (register direct)	dddddddd: GBR indirect with displacement	MOV.L @R0,@(disp,GBR)
	dddddddd: PC-relative with displacement	R0 (register direct)	MOVA @(disp,PC),R0
	dddddddd: PC-relative	—	BF label
d12 type 	dddddddddddd: PC-relative	—	BRA label (label=disp+PC)
nd8 type 	dddddddd: PC-relative with displacement	nnnn: register direct	MOV.L @(disp,PC),Rn
i type 	iiiiiiii: immediate	Indexed GBR indirect	AND.B #imm,@(R0,GBR)
	iiiiiiii: immediate	R0 (register direct)	AND #imm,R0
	iiiiiiii: immediate	—	TRAPA #imm
ni type 	iiiiiiii: immediate	nnnn: register direct	ADD #imm,Rn

2.4.4 DSP Instruction Formats

This LSI includes new instructions for digital signal processing. The new instructions are of the following two kinds.

1. Memory and DSP register double and single data transfer instructions (16-bit length)
2. Parallel processing instructions processed by the DSP unit (32-bit length)

The instruction formats are shown in figure 2.15.

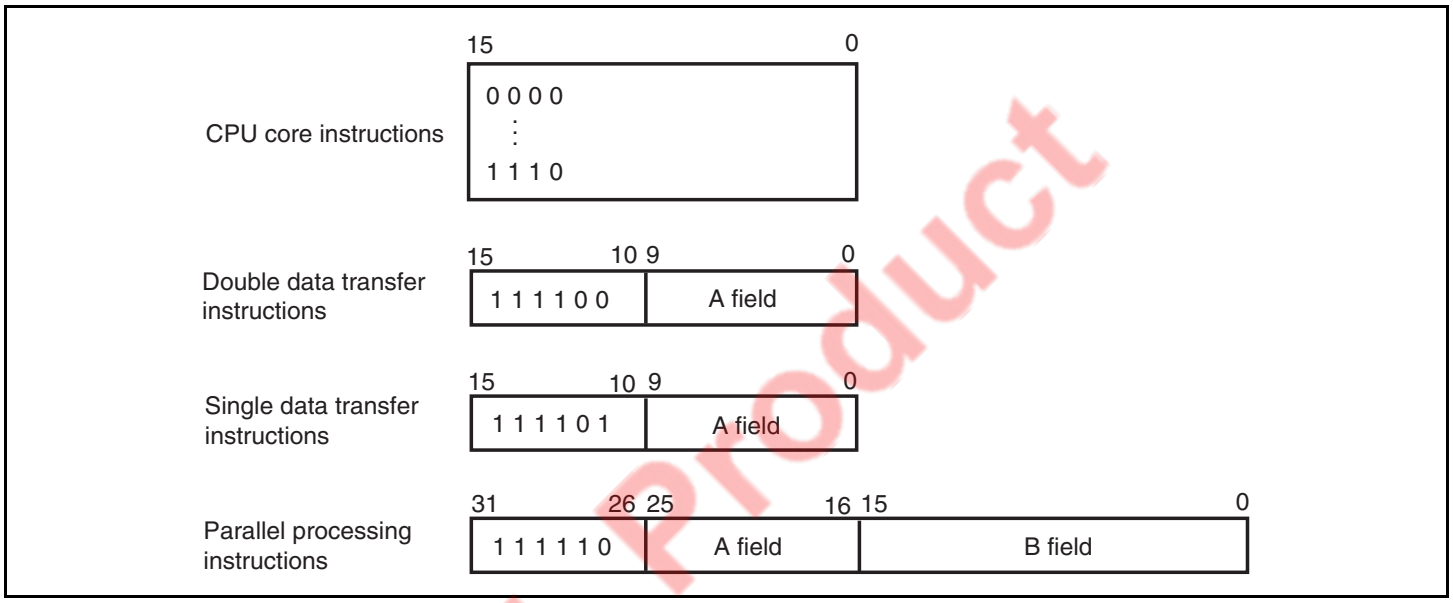


Figure 2.15 DSP Instruction Formats

Double and Single Data Transfer Instructions: The format of double data transfer instructions is shown in table 2.14, and that of single data transfer instructions in table 2.15.

Table 2.14 Double Data Transfer Instruction Formats

Type	Mnemonic	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X memory	NOPX	1	1	1	1	0	0	0		0		0		0	0		
data	MOVX.W @Ax,Dx							Ax		Dx		0		0	1		
transfer	MOVX.W @Ax+,Dx													1	0		
	MOVX.W @Ax+Ix,Dx													1	1		
	MOVX.W Da,@Ax									Da		1		0	1		
	MOVX.W Da,@Ax+													1	0		
	MOVX.W Da,@Ax+Ix													1	1		
Y memory	NOPY	1	1	1	1	0	0		0		0		0			0	0
data	MOVY.W @Ay,Dy								Ay		Dy		0			0	1
transfer	MOVY.W @Ay+,Dy															1	0
	MOVY.W @Ay+Iy,Dy															1	1
	MOVY.W Da,@Ay										Da		1			0	1
	MOVY.W Da,@Ay+															1	0
	MOVY.W Da,@Ay+Iy															1	1

Note: Ax: 0 = R4, 1 = R5
Ay: 0 = R6, 1 = R7
Dx: 0 = X0, 1 = X1
Dy: 0 = Y0, 1 = Y1
Da: 0 = A0, 1 = A1

Table 2.15 Single Data Transfer Instruction Formats

Type	Mnemonic	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Single data transfer	MOVS.W @-As,Ds	1	1	1	1	0	1	As		Ds		0:(*)		0	0	0	0
	MOVS.W @As,Ds							0:R4				1:(*)		0	1		
	MOVS.W @As+,Ds							1:R5				2:(*)		1	0		
	MOVS.W @As+Ix,Ds							2:R2				3:(*)		1	1		
	MOVS.W Ds,@-As							3:R3				4:(*)		0	0	0	1
	MOVS.W Ds,@As											5:A1		0	1		
	MOVS.W Ds,@As+											6:(*)		1	0		
	MOVS.W Ds,@As+Ix											7:A0		1	1		
	MOVS.L @-As,Ds											8:X0		0	0	1	0
	MOVS.L @As,Ds											9:X1		0	1		
	MOVS.L @As+,Ds											A:Y0		1	0		
	MOVS.L @As+Ix,Ds											B:Y1		1	1		
	MOVS.L Ds,@-As											C:M0		0	0	1	1
	MOVS.L Ds,@As											D:A1G		0	1		
	MOVS.L Ds,@As+											E:M1		1	0		
	MOVS.L Ds,@As+Ix											F:A0G		1	1		

Note: * Codes reserved for system use.

Parallel Processing Instructions: Parallel processing instructions are provided for efficient execution of digital signal processing using the DSP unit. They are 32 bits long and allow four simultaneous processes, an ALU operation, multiplication, and two data transfers.

Parallel processing instructions are divided into an A field and a B field. The A field defines data transfer instructions and the B field an ALU operation instruction and multiply instruction. These instructions can be defined independently, and the processing is executed in parallel, independently and simultaneously. A-field parallel data transfer instructions are shown in table 2.16, and B-field ALU operation instructions and multiply instructions in table 2.17.

Table 2.16 A-Field Parallel Data Transfer Instructions

Type	Mnemonic	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X memory data transfer	NOPX	1	1	1	1	1	0	0		0		0		0	0			B field															
	MOVX.W @Ax, Dx							Ax		Dx		0		0	1																		
	MOVX.W @Ax+, Dx													1	0																		
	MOVX.W @Ax+Ix, Dx													1	1																		
	MOVX.W Da, @Ax									Da		1		0	1																		
	MOVX.W Da, @Ax+													1	0																		
	MOVX.W Da, @Ax+Ix													1	1																		
Y memory data transfer	NOPY								0		0		0			0	0	B field															
	MOVY.W @Ay, Dy								Ay		Dy		0			0	1																
	MOVY.W @Ay+, Dy															1	0																
	MOVY.W @Ay+Iy, Dy															1	1																
	MOVY.W Da, @Ay										Da		1			0	1																
	MOVY.W Da, @Ay+															1	0																
	MOVY.W Da, @Ay+Iy															1	1																

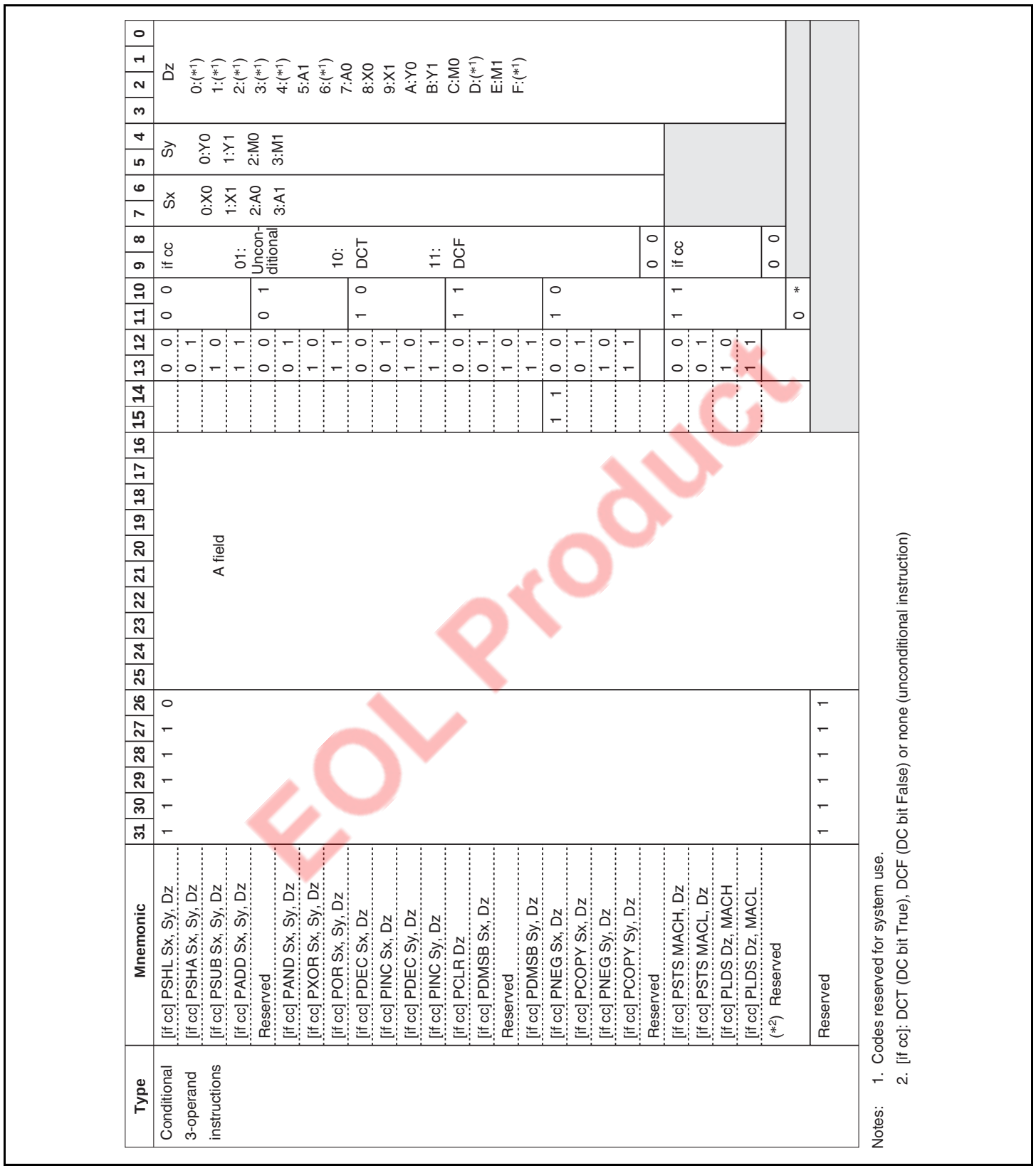
Note: Ax: 0 = R4, 1 = R5
Ay: 0 = R6, 1 = R7
Dx: 0 = X0, 1 = X1
Dy: 0 = Y0, 1 = Y1
Da: 0 = A0, 1 = A1

Table 2.17 B-Field ALU Operation Instructions and Multiply Instructions (1)

Type	Mnemonic	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
imm. shift	PSHL #imm, Dz	1	1	1	1	1	0	A field										0	0	0	0	0	−16 <= imm <= +16							Dz			
	PSHA #imm, Dz																	0	0	0	0	1	−32 <= imm <= +32										
	Reserved																	0	0	0	1	0											
																		0	0	0	1	1											
6-operand parallel instructions	PMULS Se, Sf, Dg																	0	1	0	0		Se	Sf		Sx		Sy		Dg		Du	
	Reserved																	0	1	0	1		0:X0	0:Y0		0:X0		0:Y0		0:M0		0:X0	
																		0	1	1	0		1:X1	1:Y1		1:X1		1:Y1		1:M1		1:Y0	
	PSUB Sx, Sy, Du																	0	1	1	1		2:Y0	2:X0		2:A0		2:M0		2:A0		2:A0	
	PMULS Se, Sf, Dg																						3:A1	3:A1		3:A1		3:M1		3:A1		3:A1	
	PADD Sx, Sy, Du																	0	1	1	1												
	PMULS Se, Sf, Dg																																
3-operand instructions	Reserved																	1	0	0	0	0	0	0	0					Dz			
																				0	1	0	0							0:(*1)			
	PSUBC Sx, Sy, Dz																			1	0	0	1							1:(*1)			
	PADDC Sx, Sy, Dz																			1	1	1	0							2:(*1)			
	PCMP Sx, Sy																			0	0	1	1							3:(*1)			
	Reserved																					0	1							4:(*1)			
	Reserved																					1	0							5:A1			
	Reserved																					1	1							6:(*1)			
	PABS Sx, Dz																					0	0							7:A0			
	PRND Sx, Dz																					0	1							8:X0			
	PABS Sy, Dz																					1	0							9:X1			
	PRND Sy, Dz																					1	1							A:Y0			
	Reserved																					0	0							B:Y1			
																						0	1							C:M0			
																						1	0							D:(*1)			
																						1	1							E:M1			
																														F:(*1)			

Note: 1. Codes reserved for system use.

Table 2.17 B-Field ALU Operation Instructions and Multiply Instructions (2)



Type	Mnemonic	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Conditional 3-operand instructions	[if cc] PSHL Sx, Sy, Dz	1	1	1	1	1	1	0	A field											0	0	0	0	if cc		Sx		Sy		Dz			
	[if cc] PSHA Sx, Sy, Dz																			0	1						0:X0		0:Y0		0:(*1)		
	[if cc] PSUB Sx, Sy, Dz																			1	0						1:X1		1:Y1		1:(*1)		
	[if cc] PADD Sx, Sy, Dz																			1	1						2:A0		2:M0		2:(*1)		
	Reserved																			0	0	0	1	01: Unconditional			3:A1		3:M1		3:(*1)		
	[if cc] PAND Sx, Sy, Dz																			0	1										4:(*1)		
	[if cc] PXOR Sx, Sy, Dz																			1	0										5:A1		
	[if cc] POR Sx, Sy, Dz																			1	1										6:(*1)		
	[if cc] PDEC Sx, Dz																			0	0	1	0	10: DCT							7:A0		
	[if cc] PINC Sx, Dz																			0	1										8:X0		
	[if cc] PDEC Sy, Dz																			1	0										9:X1		
	[if cc] PINC Sy, Dz																			1	1										A:Y0		
	[if cc] PCLR Dz																			0	0	1	1	11: DCF							B:Y1		
	[if cc] PDMSB Sx, Dz																			0	1										C:M0		
	Reserved																			1	0										D:(*1)		
	[if cc] PDMSB Sy, Dz																			1	1										E:M1		
	[if cc] PNEG Sx, Dz																		1	0	0	0	1	0	0						F:(*1)		
	[if cc] PCOPY Sx, Dz																				0	1											
	[if cc] PNEG Sy, Dz																				1	0											
	[if cc] PCOPY Sy, Dz																				1	1											
	Reserved																							0	0								
	[if cc] PSTS MACH, Dz																			0	0	1	1	1	1	if cc							
	[if cc] PSTS MACL, Dz																			0	1												
	[if cc] PLDS Dz, MACH																			1	0												
	[if cc] PLDS Dz, MACL																			1	1												
	(*2) Reserved																							0	0								
	Reserved	1	1	1	1	1	1	1																									

Notes: 1. Codes reserved for system use.

2. [if cc]: DCT (DC bit True), DCF (DC bit False) or none (unconditional instruction)

2.5 Instruction Set

2.5.1 CPU Instruction Set

The SH-1/SH-2/SH-3 compatible instruction set consists of 67 basic instruction types divided into seven functional groups, as shown in table 2.18. Tables 2.19 to 2.24 show the instruction notation, machine code, execution time, and function.

Table 2.18 CPU Instruction Types

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Data transfer instructions	5	MOV	Data transfer Immediate data transfer Peripheral module data transfer Structure data transfer	39
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Upper/lower swap	
		XTRCT	Extraction of middle of linked registers	
Arithmetic operation instructions	21	ADD	Binary addition	34
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Signed division initialization	
		DIV0U	Unsigned division initialization	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate, double-precision multiply-and-accumulate	

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Arithmetic operation instructions	21	MUL	Double-precision multiplication (32 × 32 bits)	34
		MULS	Signed multiplication (16 × 16 bits)	
		MULU	Unsigned multiplication (16 × 16 bits)	
		NEG	Sign inversion	
		NEGC	Sign inversion with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with carry	
		SUBV	Binary subtraction with underflow	
Logic operation instructions	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit setting	
		TST	Logical AND and T bit setting	
		XOR	Exclusive logical OR	
Shift instructions	12	ROTL	1-bit left rotation	16
		ROTR	1-bit right rotation	
		ROTCL	1-bit left rotation with T bit	
		ROTCR	1-bit right rotation with T bit	
		SHAL	Arithmetic 1-bit left shift	
		SHAR	Arithmetic 1-bit right shift	
		SHLL	Logical 1-bit left shift	
		SHLLn	Logical n-bit left shift	
		SHLR	Logical 1-bit right shift	
		SHLRn	Logical n-bit right shift	
		SHAD	Arithmetic dynamic shift	
		SHLD	Logical dynamic shift	

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Branch instructions	9	BF	Conditional branch, delayed conditional branch (T = 0)	11
		BT	Conditional branch, delayed conditional branch (T = 1)	
		BRA	Unconditional branch	
		BRAF	Unconditional branch	
		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	
System control instructions	14	CLRT	T bit clear	74
		CLRMAC	MAC register clear	
		CLRS	S bit clear	
		LDC	Load into control register	
		LDS	Load into system register	
		NOP	No operation	
		PREF	Data prefetch to cache	
		RTE	Return from exception handling	
		SETS	S bit setting	
		SETT	T bit setting	
		SLEEP	Transition to power-down mode	
		STC	Store from control register	
		STS	Store from system register	
		TRAPA	Trap exception handling	
Total:	67			188

The instruction code, operation, and number of execution states of the CPU instructions are shown in the following tables, classified by instruction type, using the format shown below.

Instruction	Instruction Code	Operation	Execution States	T Bit
Indicated by mnemonic.	Indicated in MSB ↔ LSB order.	Indicates summary of operation.	Value when no wait states are inserted*[1]	Value of T bit after instruction is executed
Explanation of Symbols OP.Sz SRC, DEST OP: Operation code Sz: Size SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement	Explanation of Symbols mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ..... 1111: R15 iiii: Immediate data dddd: Displacement*[2]	Explanation of Symbols →, ←: Transfer direction (xx): Memory operand M/Q/T: Flag bits in SR &: Logical AND of each bit |: Logical OR of each bit ^: Exclusive logical OR of each bit ~: Logical NOT of each bit <<n: n-bit left shift >>n: n-bit right shift		Explanation of Symbols —: No change

Notes: 1. The table shows the minimum number of execution states. In practice, the number of instruction execution states will be increased in cases such as the following:
(1) When there is contention between an instruction fetch and a data access
(2) When the destination register of a load instruction (memory → register) is also used by the following instruction
2. Scaled (×1, ×2, or ×4) according to the instruction operand size, etc.

Data Transfer Instructions

Table 2.19 Data Transfer Instructions

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

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

Arithmetic Operation Instructions

Table 2.20 Arithmetic Operation Instructions

Instruction		Instruction Code	Operation	Execution States	T Bit
ADD	Rm,Rn	0011nnnnmmmm1100	Rn + Rm → Rn	1	—
ADD	#imm,Rn	0111nnnniiiiiiii	Rn + imm → Rn	1	—
ADDC	Rm,Rn	0011nnnnmmmm1110	Rn + Rm + T → Rn, Carry → T	1	Carry
ADDV	Rm,Rn	0011nnnnmmmm1111	Rn + Rm → Rn, Overflow → T	1	Overflow
CMP/EQ	#imm,R0	10001000iiiiiiii	If R0 = imm, 1 → T	1	Comparison result
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	If Rn = Rm, 1 → T	1	Comparison result
CMP/HS	Rm,Rn	0011nnnnmmmm0010	If Rn ≥ Rm with unsigned data, 1 → T	1	Comparison result
CMP/GE	Rm,Rn	0011nnnnmmmm0011	If Rn ≥ Rm with signed data, 1 → T	1	Comparison result
CMP/HI	Rm,Rn	0011nnnnmmmm0110	If Rn > Rm with unsigned data, 1 → T	1	Comparison result
CMP/GT	Rm,Rn	0011nnnnmmmm0111	If Rn > Rm with signed data, 1 → T	1	Comparison result
CMP/PL	Rn	0100nnnn00010101	If Rn > 0, 1 → T	1	Comparison result
CMP/PZ	Rn	0100nnnn00010001	If Rn ≥ 0, 1 → T	1	Comparison result
CMP/STR	Rm,Rn	0010nnnnmmmm1100	If Rn and Rm have an equivalent byte, 1 → T	1	Comparison result
DIV1	Rm,Rn	0011nnnnmmmm0100	Single-step division (Rn/Rm)	1	Calculation result
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	1	Calculation result
DIV0U		0000000000011001	0 → M/Q/T	1	0
DMULS.L	Rm,Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2(5) *1	—

Instruction		Instruction Code	Operation	Execution States	T Bit
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 4 bits	2(5) *[1]	—
DT	Rn	0100nnnn00010000	Rn – 1 → Rn, if Rn = 0, 1 → T, else 0 → T	1	Comparison result
EXTS.B	Rm,Rn	0110nnnnmmmm1110	A byte in Rm is sign-extended → Rn	1	—
EXTS.W	Rm,Rn	0110nnnnmmmm1111	A word in Rm is sign-extended → Rn	1	—
EXTU.B	Rm,Rn	0110nnnnmmmm1100	A byte in Rm is zero-extended → Rn	1	—
EXTU.W	Rm,Rn	0110nnnnmmmm1101	A word in Rm is zero-extended → Rn	1	—
MAC.L	@Rm+,@Rn+	0000nnnnmmmm1111	Signed operation of (Rn) × (Rm) → MAC → MAC, Rn + 4 → Rn, Rm + 4 → Rm 32 × 32 + 64 → 64 bits	2(5)*[1]	—
MAC.W	@Rm+,@Rn+	0100nnnnmmmm1111	Signed operation of (Rn) × (Rm) → MAC → MAC, Rn + 2 → Rn, Rm + 2 → Rm 16 × 16 + 64 → 64 bits	2(5)*[1]	—
MUL.L	Rm,Rn	0000nnnnmmmm0111	Rn × Rm → MACL 32 × 32 → 32 bits	2(5)*[1]	—
MULS.W	Rm,Rn	0010nnnnmmmm1111	Signed operation of Rn × Rm → MAC 16 × 16 → 32 bits	1(3)*[2]	—
MULU.W	Rm,Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MAC 16 × 16 → 32 bits	1(3)*[2]	—
NEG	Rm,Rn	0110nnnnmmmm1011	0–Rm → Rn	1	—
NEGC	Rm,Rn	0110nnnnmmmm1010	0–Rm–T → Rn, Borrow → T	1	Borrow
SUB	Rm,Rn	0011nnnnmmmm1000	Rn–Rm → Rn	1	—
SUBC	Rm,Rn	0011nnnnmmmm1010	Rn–Rm–T → Rn, Borrow → T	1	Borrow

Instruction		Instruction Code	Operation	Execution States	T Bit
SUBV	Rm,Rn	0011nnnnmmmm1011	Rn–Rm → Rn, Underflow → T	1	Underflow

Notes: 1. The normal minimum number of execution cycles is two, but five cycles are required when the operation result is read from the MAC register immediately after the instruction.

2. The normal minimum number of execution cycles is one, but three cycles are required when the operation result is read from the MAC register immediately after the MUL instruction.

Logic Operation Instructions

Table 2.21 Logic Operation Instructions

Instruction		Instruction Code	Operation	Execution States	T Bit
AND	Rm,Rn	0010nnnnmmmm1001	Rn & Rm → Rn	1	—
AND	#imm,R0	11001001iiiiiiii	R0 & imm → R0	1	—
AND.B	#imm,@(R0,GBR)	11001101iiiiiiii	(R0 + GBR) & imm → (R0 + GBR)	3	—
NOT	Rm,Rn	0110nnnnmmmm0111	~Rm → Rn	1	—
OR	Rm,Rn	0010nnnnmmmm1011	Rn | Rm → Rn	1	—
OR	#imm,R0	11001011iiiiiiii	R0 | imm → R0	1	—
OR.B	#imm,@(R0,GBR)	11001111iiiiiiii	(R0 + GBR) | imm → (R0 + GBR)	3	—
TAS.B	@Rn	0100nnnn00011011	If (Rn) is 0, 1 → T; 1 → MSB of (Rn)	4	Test result
TST	Rm,Rn	0010nnnnmmmm1000	Rn & Rm; if the result is 0, 1 → T	1	Test result
TST	#imm,R0	11001000iiiiiiii	R0 & imm; if the result is 0, 1 → T	1	Test result
TST.B	#imm,@(R0,GBR)	11001100iiiiiiii	(R0 + GBR) & imm; if the result is 0, 1 → T	3	Test result
XOR	Rm,Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	1	—
XOR	#imm,R0	11001010iiiiiiii	R0 ^ imm → R0	1	—
XOR.B	#imm,@(R0,GBR)	11001110iiiiiiii	(R0 + GBR) ^ imm → (R0 + GBR)	3	—

Shift Instructions

Table 2.22 Shift Instructions

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

Branch Instructions

Table 2.23 Branch Instructions

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

Note: * One state when the branch is not executed.

System Control Instructions

Table 2.24 System Control Instructions

Instruction	Instruction Code	Operation	Execution States	T Bit
CLRMAC	0000000000101000	0 → MACH, MACL	1	—
CLRS	0000000001001000	0 → S	1	—
CLRT	0000000000001000	0 → T	1	0
LDC Rm,SR	0100mmmm00001110	Rm → SR	6	LSB
LDC Rm,GBR	0100mmmm00011110	Rm → GBR	4	—
LDC Rm,VBR	0100mmmm00101110	Rm → VBR	4	—
LDC Rm,SSR	0100mmmm00111110	Rm → SSR	4	—
LDC Rm,SPC	0100mmmm01001110	Rm → SPC	4	—
LDC Rm,R0_BANK	0100mmmm10001110	Rm → R0_BANK	4	—
LDC Rm,R1_BANK	0100mmmm10011110	Rm → R1_BANK	4	—
LDC Rm,R2_BANK	0100mmmm10101110	Rm → R2_BANK	4	—
LDC Rm,R3_BANK	0100mmmm10111110	Rm → R3_BANK	4	—
LDC Rm,R4_BANK	0100mmmm11001110	Rm → R4_BANK	4	—
LDC Rm,R5_BANK	0100mmmm11011110	Rm → R5_BANK	1	—
LDC Rm,R6_BANK	0100mmmm11101110	Rm → R6_BANK	4	—
LDC Rm,R7_BANK	0100mmmm11111110	Rm → R7_BANK	4	—
LDC.L @Rm+,SR	0100mmmm00000111	(Rm) → SR, Rm + 4 → Rm	8	LSB
LDC.L @Rm+,GBR	0100mmmm00010111	(Rm) → GBR, Rm + 4 → Rm	4	—
LDC.L @Rm+,VBR	0100mmmm00100111	(Rm) → VBR, Rm + 4 → Rm	4	—
LDC.L @Rm+,SSR	0100mmmm00110111	(Rm) → SSR, Rm + 4 → Rm	4	—
LDC.L @Rm+,SPC	0100mmmm01000111	(Rm) → SPC, Rm + 4 → Rm	4	—
LDC.L @Rm+, R0_BANK	0100mmmm10000111	(Rm) → R0_BANK, Rm + 4 → Rm	4	—
LDC.L @Rm+, R1_BANK	0100mmmm10010111	(Rm) → R1_BANK, Rm + 4 → Rm	4	—
LDC.L @Rm+, R2_BANK	0100mmmm10100111	(Rm) → R2_BANK, Rm + 4 → Rm	4	—
LDC.L @Rm+, R3_BANK	0100mmmm10110111	(Rm) → R3_BANK, Rm + 4 → Rm	4	—

Instruction		Instruction Code	Operation	Execution States	T Bit
LDC.L	@Rm+, R4_BANK	0100mmmm11000111	(Rm) → R4_BANK, Rm + 4 → Rm	4	—
LDC.L	@Rm+, R5_BANK	0100mmmm11010111	(Rm) → R5_BANK, Rm + 4 → Rm	4	—
LDC.L	@Rm+, R6_BANK	0100mmmm11100111	(Rm) → R6_BANK, Rm + 4 → Rm	4	—
LDC.L	@Rm+, R7_BANK	0100mmmm11110111	(Rm) → R7_BANK, Rm + 4 → Rm	4	—
LDS	Rm,MACH	0100mmmm00001010	Rm → MACH	1	—
LDS	Rm,MACL	0100mmmm00011010	Rm → MACL	1	—
LDS	Rm,PR	0100mmmm00101010	Rm → PR	1	—
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm) → MACH, Rm + 4 → Rm	1	—
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm) → MACL, Rm + 4 → Rm	1	—
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm) → PR, Rm + 4 → Rm	1	—
NOP		0000000000001001	No operation	1	—
PREF	@Rm	0000mmmm10000011	(Rm) → cache	1	—
RTE		0000000000101011	Delayed branch, SSR/SPC → SR/PC	5	—
SETS		0000000001011000	1 → S	1	—
SETT		0000000000011000	1 → T	1	1
SLEEP		0000000000011011	Sleep	4*	—
STC	SR,Rn	0000nnnn00000010	SR → Rn	1	—
STC	GBR,Rn	0000nnnn00010010	GBR → Rn	1	—
STC	VBR,Rn	0000nnnn00100010	VBR → Rn	1	—
STC	SSR,Rn	0000nnnn00110010	SSR → Rn	1	—
STC	SPC,Rn	0000nnnn01000010	SPC → Rn	1	—
STC	R0_BANK,Rn	0000nnnn10000010	R0_BANK→ Rn	1	—
STC	R1_BANK,Rn	0000nnnn10010010	R1_BANK→ Rn	1	—
STC	R2_BANK,Rn	0000nnnn10100010	R2_BANK→ Rn	1	—
STC	R3_BANK,Rn	0000nnnn10110010	R3_BANK→ Rn	1	—
STC	R4_BANK,Rn	0000nnnn11000010	R4_BANK→ Rn	1	—
STC	R5_BANK,Rn	0000nnnn11010010	R5_BANK→ Rn	1	—
STC	R6_BANK,Rn	0000nnnn11100010	R6_BANK→ Rn	1	—

Instruction		Instruction Code	Operation	Execution States	T Bit
STC	R7_BANK,Rn	0000nnnn11110010	R7_BANK→ Rn	1	—
STC.L	SR,@–Rn	0100nnnn00000011	Rn–4 → Rn, SR → (Rn)	1	—
STC.L	GBR,@–Rn	0100nnnn00010011	Rn–4 → Rn, GBR → (Rn)	1	—
STC.L	VBR,@–Rn	0100nnnn00100011	Rn–4 → Rn, VBR → (Rn)	1	—
STC.L	SSR,@–Rn	0100nnnn00110011	Rn–4 → Rn, SSR → (Rn)	1	—
STC.L	SPC,@–Rn	0100nnnn01000011	Rn–4 → Rn, SPC → (Rn)	1	—
STC.L	R0_BANK, @–Rn	0100nnnn10000011	Rn–4 → Rn, R0_BANK → (Rn)	1	—
STC.L	R1_BANK, @–Rn	0100nnnn10010011	Rn–4 → Rn, R1_BANK → (Rn)	1	—
STC.L	R2_BANK, @–Rn	0100nnnn10100011	Rn–4 → Rn, R2_BANK → (Rn)	1	—
STC.L	R3_BANK, @–Rn	0100nnnn10110011	Rn–4 → Rn, R3_BANK → (Rn)	1	—
STC.L	R4_BANK, @–Rn	0100nnnn11000011	Rn–4 → Rn, R4_BANK → (Rn)	1	—
STC.L	R5_BANK, @–Rn	0100nnnn11010011	Rn–4 → Rn, R5_BANK → (Rn)	1	—
STC.L	R6_BANK, @–Rn	0100nnnn11100011	Rn–4 → Rn, R6_BANK → (Rn)	1	—
STC.L	R7_BANK, @–Rn	0100nnnn11110011	Rn–4 → Rn, R7_BANK → (Rn)	1	—
STS	MACH,Rn	0000nnnn00001010	MACH → Rn	1	—
STS	MACL,Rn	0000nnnn00011010	MACL → Rn	1	—
STS	PR,Rn	0000nnnn00101010	PR → Rn	1	—
STS.L	MACH,@–Rn	0100nnnn00000010	Rn–4 → Rn, MACH → (Rn)	1	—
STS.L	MACL,@–Rn	0100nnnn00010010	Rn–4 → Rn, MACL → (Rn)	1	—
STS.L	PR,@–Rn	0100nnnn00100010	Rn–4 → Rn, PR → (Rn)	1	—
TRAPA	#imm	11000011iiiiiiii	PC → SPC, SR → SSR, imm << 2 → TRA, VBR + H'0100 → PC	8	—

Note: * Number of states before the chip enters the sleep state.

The table shows the minimum number of clocks required for execution. In practice, the number of execution cycles will be increased if there is contention between an instruction fetch and a data access, or if the destination register of a load instruction (memory → register) is also used by the following instruction.

2.6 DSP Extended-Function Instructions

2.6.1 Introduction

The newly added instructions are classified into the following three groups:

1. Additional system control instructions for the CPU unit
2. DSP unit memory-register single and double data transfer
3. DSP unit parallel processing

Group 1 instructions are provided to support loop control and data transfer between CPU core registers or memory and new control registers added to the CPU core. DSP operations employ a multi-level nested-loop structure. With a single-level loop, use of the decrement and test, DTRn, and conditional delayed branch BF/S instructions supported by the SH-3 is adequate. However, with nested loops, DSP performance can be improved by means of a zero-overhead loop control function.

The RS, RE, and MOD registers have been added to support loop control and modulo addressing functions. Instructions are supported for data transfer between these new control registers and general registers or memory. In addition, the LDRS and LDRE address calculation registers have been added to reduce the code size for the initial settings for zero-overhead loop control.

An independent control register, DSR, is provided for the DSP engine. This register is treated as a system register such as MACL and MACH. The A0, X0, X1, Y0, and Y1 registers are treated as system registers from the CPU side, and LDS/STS instructions are supported for the same purpose. Table 2.25 shows the instruction code map for the new system control instructions for the CPU core.

Group 2 instructions are provided to reduce DSP operation program code size. Data transfer instructions that perform no data processing are frequently executed by the DSP engine. In this case, a 32-bit instruction code is unnecessarily long, and wastes space in the program memory area. All instructions in this class have a 16-bit code length, the same as conventional SH core instructions. Single data transfer instructions have greater flexibility in terms of operands than the double data transfer instruction or parallel instruction class.

Group 3 instructions are provided for fast execution of digital signal processing operations using the DSP unit. These instructions have a 32-bit instruction code, so that a maximum of four instructions—an ALU operation, multiplication, and two data transfer instructions—can be executed in parallel.

2.6.2 Added CPU System Control Instructions

The new instructions in this class are treated as part of the CPU core functions, and therefore all the added instructions have a 16-bit code length. All the additional instructions belong to the system control instruction group. Table 2.25 summarizes the added system instructions. New control registers—RS, RE, and MOD—have been added to the CPU core to support loop control and modulo addressing functions, and LDC and STS type instructions have been provided for these registers.

The DSP engine's DSR, A0, X0, X1, Y0, and Y1 registers are treated as system registers such as MACH and MACL, and therefore STS and LDS instructions are supported for these registers. As digital signal processing operations usually employ a multi-level nested-loop structure, DSP performance can be improved by means of a zero-overhead loop control function. SETRC type instructions are provided to set the repeat count in the RC field in SR[27:16]. When an immediate operand type SETRC instruction is executed, the 8-bit immediate operand data is set in SR[23:16], and 0 is set in the remaining bits, SR[27:24]. When a register operand type SETRC instruction is executed, Rn[11:0] is set in SR[27:16]. The start address and end address of the repeat loop are set in the RS register and RE register. There are two ways of setting the addresses: by using an LDC type instruction, or by using the LDRS and LDRE instructions.

Table 2.25 Added CPU System Control Instructions

Instruction		Instruction Code	Operation	Execution States	T Bit
SETRC	#imm	10000010iiiiiiii	imm → RC (of SR)	1	—
SETRC	Rn	0100nnnn00010100	Rn[11:0] → R C (of SR)	1	—
LDRS	@(disp,PC)	10001100dddddddd	(disp × 2 + PC) → RS	1	—
LDRE	@(disp,PC)	10001110dddddddd	(disp × 2 + PC) → RE	1	—
STC	MOD,Rn	0000nnnn01010010	MOD → Rn	1	—
STC	RS,Rn	0000nnnn01100010	RS → Rn	1	—
STC	RE,Rn	0000nnnn01110010	RE → Rn	1	—
STS	DSR,Rn	0000nnnn01101010	DSR → Rn	1	—
STS	A0,Rn	0000nnnn01111010	A0 → Rn	1	—
STS	X0,Rn	0000nnnn10001010	X0 → Rn	1	—
STS	X1,Rn	0000nnnn10011010	X1 → Rn	1	—
STS	Y0,Rn	0000nnnn10101010	Y0 → Rn	1	—
STS	Y1,Rn	0000nnnn10111010	Y1 → Rn	1	—

Instruction	Instruction Code	Operation	Execution States	T Bit
STS.L DSR,@-Rn	0100nnnn01100010	Rn – 4 → Rn, DSR → (Rn)	1	—
STS.L A0,@-Rn	0100nnnn01110010	Rn – 4 → Rn, A0 → (Rn)	1	—
STS.L X0,@-Rn	0100nnnn10000010	Rn – 4 → Rn, X0 → (Rn)	1	—
STS.L X1,@-Rn	0100nnnn10010010	Rn – 4 → Rn, X1 → (Rn)	1	—
STS.L Y0,@-Rn	0100nnnn10100010	Rn – 4 → Rn, Y0 → (Rn)	1	—
STS.L Y1,@-Rn	0100nnnn10110010	Rn – 4 → Rn, Y1 → (Rn)	1	—
STC.L MOD,@-Rn	0100nnnn01010011	Rn – 4 → Rn, MOD → (Rn)	1	—
STC.L RS,@-Rn	0100nnnn01100011	Rn – 4 → Rn, RS → (Rn)	1	—
STC.L RE,@-Rn	0100nnnn01110011	Rn – 4 → Rn, RE → (Rn)	1	—
LDS.L @Rn+,DSR	0100nnnn01100110	(Rn) → DSR, Rn + 4 → Rn	1	—
LDS.L @Rn+,A0	0100nnnn01110110	(Rn) → A0, Rn + 4 → Rn	1	—
LDS.L @Rn+,X0	0100nnnn10000110	(Rn) → X0, Rn + 4 → Rn	1	—
LDS.L @Rn+,X1	0100nnnn10010110	(Rn) → X1, Rn + 4 → Rn	1	—
LDS.L @Rn+,Y0	0100nnnn10100110	(Rn) → Y0, Rn + 4 → Rn	1	—
LDS.L @Rn+,Y1	0100nnnn10110110	(Rn) → Y1, Rn + 4 → Rn	1	—
LDC.L @Rn+,MOD	0100nnnn01010111	(Rn) → MOD, Rn + 4 → Rn	4	—
LDC.L @Rn+,RS	0100nnnn01100111	(Rn) → RS, Rn + 4 → Rn	4	—
LDC.L @Rn+,RE	0100nnnn01110111	(Rn) → RE, Rn + 4 → Rn	4	—
LDS Rn,DSR	0100nnnn01101010	Rn → DSR	1	—
LDS Rn,A0	0100nnnn01111010	Rn → A0	1	—
LDS Rn,X0	0100nnnn10001010	Rn → X0	1	—
LDS Rn,X1	0100nnnn10011010	Rn → X1	1	—
LDS Rn,Y0	0100nnnn10101010	Rn → Y0	1	—
LDS Rn,Y1	0100nnnn10111010	Rn → Y1	1	—
LDC Rn,MOD	0100nnnn01011110	Rn → MOD	4	—
LDC Rn,RS	0100nnnn01101110	Rn → RS	4	—
LDC Rn,RE	0100nnnn01111110	Rn → RE	4	—

2.6.3 Single and Double Data Transfer for DSP Data Instructions

The new instructions in this class are provided to reduce the program code size for DSP operations. All the new instructions in this class have a 16-bit code length. Instructions in this class are divided into two groups: single data transfer instructions and double data transfer instructions. The double data-transfer instructions provide the same flexibility in operand specification as is provided by the A fields of the data-transfer instruction fields of parallel-processing instructions. This is described in section 2.4.4, DSP Instruction Formats. Conditional load instructions cannot be used with these 16-bit instructions. In single transfer, the Ax pointer and two other pointers are used as the As pointer, but the Ay pointer is not used. Tables 2.26 and 2.27 list the single and double data transfer instructions.

With double data transfer group instructions, X memory and Y memory can be accessed in parallel. The Ax pointer can only be used by X memory access instructions, and the Ay pointer only by Y memory access instructions. Double data transfer instructions can only access the on-chip X and Y memory areas. Single data transfer instructions use a 16-bit instruction code, and can access any memory address space.

Rn (n = 2 to 7) registers are normally used as the Ax, Ay, and As pointers. The pointer names themselves can be changed with the assembler rename function. The following renaming scheme is recommended.

R2:As2, R3:As3, R4:Ax0 (As0), R5:Ax1 (As1), R6:Ay0, R7:Ay1, R8:Ix, R9:Iy

Table 2.26 Double Data Transfer Instructions

Instruction		Instruction Code	Operation	Execution States	DC
X memory data transfer	NOPX	1111000*0*0*00**	X memory no access	1	—
	MOVX.W @Ax,Dx	111100A*D*0*01**	(Ax) → MSW of Dx, 0 → LSW of Dx	1	—
	MOVX.W @Ax+,Dx	111100A*D*0*10**	(Ax) → MSW of Dx, 0 → LSW of Dx, Ax + 2 → Ax	1	—
	MOVX.W @Ax+Ix,Dx	111100A*D*0*11**	(Ax) → MSW of Dx, 0 → LSW of Dx, Ax + Ix → Ax	1	—
	MOVX.W Da,@Ax	111100A*D*1*01**	MSW of Da → (Ax)	1	—
	MOVX.W Da,@Ax+	111100A*D*1*10**	MSW of Da → (Ax), Ax + 2 → Ax	1	—
	MOVX.W Da,@Ax+Ix	111100A*D*1*11**	MSW of Da → (Ax), Ax + Ix → Ax	1	—
Y memory data transfer	NOPY	111100*0*0*0**00	Y memory no access	1	—
	MOVY.W @Ay,Dy	111100*A*D*0**01	(Ay) → MSW of Dy, 0 → LSW of Dy	1	—
	MOVY.W @Ay+,Dy	111100*A*D*0**10	(Ay) → MSW of Dy, 0 → LSW of Dy, Ay + 2 → Ay	1	—
	MOVY.W @Ay+Iy,Dy	111100*A*D*0**11	(Ay) → MSW of Dy, 0 → LSW of Dy, Ay + Iy → Ay	1	—
	MOVY.W Da,@Ay	111100*A*D*1**01	MSW of Da → (Ay)	1	—
	MOVY.W Da,@Ay+	111100*A*D*1**10	MSW of Da → (Ay), Ay + 2 → Ay	1	—
	MOVY.W Da,@Ay+Iy	111100*A*D*1**11	MSW of Da → (Ay), Ay + Iy → Ay	1	—

Table 2.27 Single Data Transfer Instructions

Instruction	Instruction Code	Operation	Execution States	DC
MOVS.W @-As,Ds	111101AADDDD0000	As – 2 → As, (As) → MSW of Ds, 0 → LSW of Ds	1	—
MOVS.W @As,Ds	111101AADDDD0100	(As) → MSW of Ds, 0 → LSW of Ds	1	—
MOVS.W @As+,Ds	111101AADDDD1000	(As) → MSW of Ds, 0 → LSW of Ds, As + 2 → As	1	—
MOVS.W @As+Is,Ds	111101AADDDD1100	(Asc) → MSW of Ds, 0 → LSW of Ds, As + Is → As	1	—
MOVS.W Ds,@-As*	111101AADDDD0001	As – 2 → As, MSW of Ds → (As)	1	—
MOVS.W Ds,@As*	111101AADDDD0101	MSW of Ds → (As)	1	—
MOVS.W Ds,@As+*	111101AADDDD1001	MSW of Ds → (As), As + 2 → As	1	—
MOVS.W Ds,@As+Is*	111101AADDDD1101	MSW of Ds → (As), As + Is → As	1	—
MOVS.L @-As,Ds	111101AADDDD0010	As – 4 → As, (As) → Ds	1	—
MOVS.L @As,Ds	111101AADDDD0110	(As) → Ds	1	—
MOVS.L @As+,Ds	111101AADDDD1010	(As) → Ds, As + 4 → As	1	—
MOVS.L @As+Is,Ds	111101AADDDD1110	(As) → Ds, As + Is → As	1	—
MOVS.L Ds,@-As	111101AADDDD0011	As – 4 → As, Ds → (As)	1	—
MOVS.L Ds,@As	111101AADDDD0111	Ds → (As)	1	—
MOVS.L Ds,@As+	111101AADDDD1011	Ds → (As), As + 4 → As	1	—
MOVS.L Ds,@As+Is	111101AADDDD1111	Ds → (As), As + Is → As	1	—

Note: * If guard bit registers A0G and A1G are specified in source operand Ds, the data is output to the LDB[7:0] bus and the sign bit is copied into the upper bits, [31:8].

The correspondence between DSP data transfer operands and registers is shown in table 2.28. CPU core registers are used as a pointer address that indicates a memory address.

Table 2.28 Correspondence between DSP Data Transfer Operands and Registers

Register		Ax	Ix	Dx	Ay	Iy	Dy	Da	As	Ds
CPU registers	R0	—	—	—	—	—	—	—	—	—
	R1	—	—	—	—	—	—	—	—	—
	R2 (As2)	—	—	—	—	—	—	—	Yes	—
	R3 (As3)	—	—	—	—	—	—	—	Yes	—
	R4 (Ax0)	Yes	—	—	—	—	—	—	Yes	—
	R5 (Ax1)	Yes	—	—	—	—	—	—	Yes	—
	R6 (Ay0)	—	—	—	Yes	—	—	—	—	—
	R7 (Ay1)	—	—	—	Yes	—	—	—	—	—
	R8 (Ix)	—	Yes	—	—	—	—	—	—	—
	R9 (Iy)	—	—	—	—	Yes	—	—	—	—
DSP registers	A0	—	—	—	—	—	—	Yes	—	Yes
	A1	—	—	—	—	—	—	Yes	—	Yes
	M0	—	—	—	—	—	—	—	—	Yes
	M1	—	—	—	—	—	—	—	—	Yes
	X0	—	—	Yes	—	—	—	—	—	Yes
	X1	—	—	Yes	—	—	—	—	—	Yes
	Y0	—	—	—	—	—	Yes	—	—	Yes
	Y1	—	—	—	—	—	Yes	—	—	Yes
	A0G	—	—	—	—	—	—	—	—	Yes
	A1G	—	—	—	—	—	—	—	—	Yes

2.6.4 DSP Operation Instruction Set

DSP operation instructions are instructions for digital signal processing performed by the DSP unit. These instructions have a 32-bit instruction code, and multiple instructions can be executed in parallel. The instruction code is divided into an A field and B field; a parallel data transfer instruction is specified in the A field, and a single or double data operation instruction in the B field. Instructions can be specified independently, and are also executed independently. The parallel data transfer instruction specified in the A field is exactly the same as a double data transfer instruction. The function of the A field—that is, the data transfer instruction field—is basically the same as in the double data transfer instructions described in section 2.6.3, Single and Double Data Transfer for DSP Data Instructions, but has a special function in load instructions.

B-field data operation instructions are of three kinds: double data operation instructions, conditional single data operation instructions, and unconditional single data operation instructions. The formats of the DSP operation instructions are shown in table 2.29. The respective operands are selected independently from the DSP registers. The correspondence between DSP operation instruction operands and registers is shown in table 2.30.

Table 2.29 DSP Operation Instruction Formats

Type	Instruction Formats	
Double data operation instructions		ALUop. Sx, Sy, Du
		MLTop. Se, Df, Dg
Conditional single data operation instructions		ALUop. Sx, Sy, Dz
	DCT	ALUop. Sx, Sy, Dz
	DCF	ALUop. Sx, Sy, Dz
		ALUop. Sx, Dz
	DCT	ALUop. Sx, Dz
	DCF	ALUop. Sx, Dz
		ALUop. Sy, Dz
	DCT	ALUop. Sy, Dz
	DCF	ALUop. Sy, Dz
Unconditional single data operation instructions		ALUop. Sx, Sy, Dz
		ALUop. Sx, Dz
		ALUop. Sy, Dz
		MLTop. Se, Sf, Dg

Table 2.30 Correspondence between DSP Instruction Operands and Registers

	ALU/BPU Operations				Multiply Operations		
Register	**Sx**	**Sy**	**Dz**	**Du**	**Se**	**Sf**	**Dg**
A0	Yes	—	Yes	Yes	—	—	Yes
A1	Yes	—	Yes	Yes	Yes	Yes	Yes
M0	—	Yes	Yes	—	—	—	Yes
M1	—	Yes	Yes	—	—	—	Yes
X0	Yes	—	Yes	Yes	Yes	Yes	—
X1	Yes	—	Yes	—	Yes	—	—
Y0	—	Yes	Yes	Yes	Yes	Yes	—
Y1	—	Yes	Yes	—	—	Yes	—

When writing parallel instructions, the B-field instruction is written first, followed by the A-field instruction. A sample parallel processing program is shown in figure 2.16.

```
        PADD A0, M0, A0   PMULS X0, Y0, M0  MOVX.W @R4+, X0     MOVY.W @R6+, Y0 [;]
DCF     PINC X1, A1                         MOVX.W A0, @R5+R8   MOVY.W @R7+, Y0 [;]
        PCMP X1, M0                         MOVX.W @R4           [NOPY] [;]
```

Figure 2.16 Sample Parallel Instruction Program

Square brackets mean that the contents can be omitted.

The no operation instructions NOPX and NOPY can be omitted. Table 2.31 gives an overview of the B field in parallel operation instructions.

A semicolon is the instruction line delimiter, but this can also be omitted. If the semicolon delimiter is used, the area to the right of the semicolon can be used as a comment field. This has the same function as with conventional SH tools.

The DSR register condition code bit (DC) is always updated on the basis of the result of an unconditional ALU or shift operation instruction. Conditional instructions do not update the DC bit. Multiply instructions, also, do not update the DC bit. DC bit updating is performed by means of bits CS0 to CS2 in the DSR register. The DC bit update rules are shown in table 2.32.

Table 2.31 DSP Operation Instructions

Instruction		Instruction Code	Operation	Execution States	DC
	PMULS Se,Sf,Dg	111110********** 0100eeff0000gg00	Se * Sf → Dg (signed)	1	—
	PADD Sx,Sy,Du PMULS Se,Sf,Dg	111110********** 0111eeffxxyygguu	Sx + Sy → Du Se * Sf → Dg (signed)	1	*
	PSUB Sx,Sy,Du PMULS Se,Sf,Dg	111110********** 0110eeffxxyygguu	Sy – Sy → Du Se * Sf → Dg (signed)	1	*
	PADD Sx,Sy,Dz	111110********** 10110001xxyyzzzz	Sx + Sy → Dz	1	*
DCT	PADD Sx,Sy,Dz	111110********** 10110010xxyyzzzz	If DC = 1, Sx + Sy → Dz If DC = 0, nop	1	—
DCF	PADD Sx,Sy,Dz	111110********** 10110011xxyyzzzz	If DC = 0, Sx + Sy → Dz If DC = 1, nop	1	—
	PSUB Sx,Sy,Dz	111110********** 10100001xxyyzzzz	Sx – Sy → Dz	1	*
DCT	PSUB Sx,Sy,Dz	111110********** 10100010xxyyzzzz	If DC = 1, Sx – Sy → Dz If DC = 0, nop	1	—
DCF	PSUB Sx,Sy,Dz	111110********** 10100011xxyyzzzz	If DC = 0, Sx – Sy → Dz If DC = 1, nop	1	—
	PSHA Sx,Sy,Dz	111110********** 10010001xxyyzzzz	If Sy > = 0, Sx << Sy → Dz (arithmetic shift) If Sy<0, Sx>>Sy → Dz	1	*
DCT	PSHA Sx,Sy,Dz	111110********** 10010010xxyyzzzz	If DC = 1 & Sy > = 0, Sx << Sy → Dz (arithmetic shift) If DC = 1 & Sy < 0, Sx >> Sy → Dz If DC = 0, nop	1	—

Instruction		Instruction Code	Operation	Execution States	DC
DCF	PSHA Sx,Sy,Dz	111110********** 10010011xxyyzzzz	If DC = 0 & Sy > = 0, Sx << Sy → Dz (arithmetic shift) If DC = 0 & Sy < 0, Sx >> Sy → Dz If DC = 1, nop	1	—
	PSHL Sx,Sy,Dz	111110********** 10000001xxyyzzzz	If Sy > = 0, Sx << Sy → Dz (logical shift) If Sy < 0, Sx >> Sy → Dz	1	*
DCT	PSHL Sx,Sy,Dz	111110********** 10000010xxyyzzzz	If DC = 1 & Sy > = 0, Sx << Sy → Dz (logical shift) If DC = 1 & Sy < 0, Sx >> Sy → Dz If DC = 0, nop	1	—
DCF	PSHL Sx,Sy,Dz	111110********** 10000011xxyyzzzz	If DC = 0 & Sy > = 0, Sx << Sy → Dz (logical shift) If DC = 0 & Sy < 0, Sx >> Sy → Dz If DC = 1, nop	1	—
	PCOPY Sx,Dz	111110********** 11011001xx00zzzz	Sx → Dz	1	*
	PCOPY Sy,Dz	111110********** 1111100100yyzzzz	Sy → Dz	1	*
DCT	PCOPY Sx,Dz	111110********** 11011010xx00zzzz	If DC = 1, Sx → Dz If DC = 0, nop	1	—
DCT	PCOPY Sy,Dz	111110********** 1111101000yyzzzz	If DC = 1, Sy → Dz If DC = 0, nop	1	—
DCF	PCOPY Sx,Dz	111110********** 11011011xx00zzzz	If DC = 0, Sx → Dz If DC = 1, nop	1	—
DCF	PCOPY Sy,Dz	111110********** 1111101100yyzzzz	If DC = 0, Sy → Dz If DC = 1, nop	1	—

Instruction		Instruction Code	Operation	Execution States	DC
	PDMSB Sx,Dz	111110********** 10011101xx00zzzz	Sx → Dz normalization count shift value	1	*
	PDMSB Sy,Dz	111110********** 1011110100yyzzzz	Sx → Dz normalization count shift value	1	*
DCT	PDMSB Sx,Dz	111110********** 10011110xx00zzzz	If DC = 1, normalization count shift value Sx → Dz If DC = 0, nop	1	—
DCT	PDMSB Sy,Dz	111110********** 1011111000yyzzzz	If DC = 1, normalization count shift value Sy → Dz If DC = 0, nop	1	—
DCF	PDMSB Sx,Dz	111110********** 10011111xx00zzzz	If DC = 0, normalization count shift value Sx → Dz If DC = 1, nop	1	—
DCF	PDMSB Sy,Dz	111110********** 1011111100yyzzzz	If DC = 0, normalization count shift value Sy → Dz If DC = 1, nop	1	—
	PINC Sx,Dz	111110********** 10011001xx00zzzz	MSW of Sx → Dz	1	*
	PINC Sy,Dz	111110********** 1011100100yyzzzz	MSW of Sy → Dz	1	*
DCT	PINC Sx,Dz	111110********** 10011010xx00zzzz	If DC = 1, MSW of Sx + 1 → Dz If DC = 0, nop	1	—
DCT	PINC Sy,Dz	111110********** 1011101000yyzzzz	If DC = 1, MSW of Sy + 1 → Dz If DC = 0, nop	1	—
DCF	PINC Sx,Dz	111110********** 10011011xx00zzzz	If DC = 0, MSW of Sx + 1 → Dz If DC = 1, nop	1	—
DCF	PINC Sy,Dz	111110********** 1011101100yyzzzz	If DC = 0, MSW of Sy + 1 → Dz If DC = 1, nop	1	—
	PNEG Sx,Dz	111110********** 11001001xx00zzzz	0 – Sx → Dz	1	*

Instruction		Instruction Code	Operation	Execution States	DC
	PNEG Sy,Dz	111110**********	0 – Sy → Dz	1	*
		1110100100yyzzzz			
DCT	PNEG Sx,Dz	111110**********	If DC = 1, 0 – Sx → Dz	1	—
		11001010xx00zzzz	If DC = 0, nop		
DCT	PNEG Sy,Dz	111110**********	If DC = 1, 0 – Sy → Dz	1	—
		1110101000yyzzzz	If DC = 0, nop		
DCF	PNEG Sx,Dz	111110**********	If DC = 0, 0 – Sx → Dz	1	—
		11001011xx00zzzz	If DC = 1, nop		
DCF	PNEG Sy,Dz	111110**********	If DC = 0, 0 – Sy → Dz	1	—
		1110101100yyzzzz	If DC = 1, nop		
	POR Sx,Sy,Dz	111110**********	Sx | Sy → Dz	1	*
		10110101xxyyzzzz			
DCT	POR Sx,Sy,Dz	111110**********	If DC = 1, Sx | Sy → Dz	1	—
		10110110xxyyzzzz	If DC = 0, nop		
DCF	POR Sx,Sy,Dz	111110**********	If DC = 0, Sx | Sy → Dz	1	—
		10110111xxyyzzzz	If DC = 1, nop		
	PAND Sx,Sy,Dz	111110**********	Sx & Sy → Dz	1	*
		10010101xxyyzzzz			
DCT	PAND Sx,Sy,Dz	111110**********	If DC = 1, Sx & Sy → Dz	1	—
		10010110xxyyzzzz	If DC = 0, nop		
DCF	PAND Sx,Sy,Dz	111110**********	If DC = 0, Sx & Sy → Dz	1	—
		10010111xxyyzzzz	If DC = 1, nop		
	PXOR Sx,Sy,Dz	111110**********	Sx ^ Sy → Dz	1	*
		10100101xxyyzzzz			
DCT	PXOR Sx,Sy,Dz	111110**********	If DC = 1, Sx ^ Sy → Dz	1	—
		10100110xxyyzzzz	If DC = 0, nop		
DCF	PXOR Sx,Sy,Dz	111110**********	If DC = 1, Sx ^ Sy → Dz	1	—
		10100111xxyyzzzz	If DC = 0, nop		
	PDEC Sx,Dz	111110**********	Sx [39:16] – 1 → Dz	1	*
		10001001xx00zzzz			

Instruction		Instruction Code	Operation	Execution States	DC
	PDEC Sy,Dz	111110********** 1010100100yyzzzz	Sy [31:16] – 1 → Dz	1	*
DCT	PDEC Sx,Dz	111110********** 10001010xx00zzzz	If DC = 1, Sx [39:16] – 1 → Dz If DC = 0, nop	1	—
DCT	PDEC Sy,Dz	111110********** 1010101000yyzzzz	If DC = 1, Sy [31:16] – 1 → Dz If DC = 0, nop	1	—
DCF	PDEC Sx,Dz	111110********** 10001011xx00zzzz	If DC = 0, Sx [39:16] – 1 → Dz If DC = 1, nop	1	—
DCF	PDEC Sy,Dz	111110********** 1010101100yyzzzz	If DC = 0, Sy [31:16] – 1 → Dz If DC = 1, nop	1	—
	PCLR Dz	111110********** 1000110100000zzzz	h'00000000 → Dz	1	*
DCT	PCLR Dz	111110********** 1000111000000zzzz	If DC = 1, h'00000000 → Dz If DC = 0, nop	1	—
DCF	PCLR Dz	111110********** 1000111100000zzzz	If DC = 0, h'00000000 → Dz If DC = 1, nop	1	—
	PSHA #imm,Dz	111110********** 00010iiiiiiizzzz	If imm > = 0, Dz << imm → Dz (arithmetic shift) If imm<0, Dz>>imm → Dz	1	*
	PSHL #imm,Dz	111110********** 00000iiiiiiizzzz	If imm > = 0, Dz << imm → Dz (logical shift) If imm < 0, Dz >> imm → Dz	1	*
	PSTS MACH,Dz	111110********** 1100110100000zzzz	MACH → Dz	1	—
DCT	PSTS MACH,Dz	111110********** 1100111000000zzzz	If DC = 1, MACH → Dz	1	—
DCF	PSTS MACH,Dz	111110********** 1100111100000zzzz	If DC = 0, MACH → Dz	1	—
	PSTS MACL,Dz	111110********** 1101110100000zzzz	MACL → Dz	1	—

Instruction		Instruction Code	Operation	Execution States	DC
DCT	PSTS MACL,Dz	111110**********	If DC = 1, MACL → Dz	1	—
		1101111100000zzzz			
DCF	PSTS MACL,Dz	111110**********	If DC = 0, MACL → Dz	1	—
		1101111110000zzzz			
	PLDS Dz,MACH	111110**********	Dz → MACH	1	—
		1110111010000zzzz			
DCT	PLDS Dz,MACH	111110**********	If DC = 1, Dz → MACH	1	—
		1110111100000zzzz			
DCF	PLDS Dz,MACH	111110**********	If DC = 0, Dz → MACH	1	—
		1110111110000zzzz			
	PLDS Dz,MACL	111110**********	Dz → MACL	1	—
		1111111010000zzzz			
DCT	PLDS Dz,MACL	111110**********	If DC = 1, Dz → MACL	1	—
		1111111100000zzzz			
DCF	PLDS Dz,MACL	111110**********	If DC = 0, Dz → MACL	1	—
		1111111110000zzzz			
	PADDC Sx,Sy,Dz	111110**********	Sx + Sy + DC → Dz	1	Carry
		10110000xxyyzzzz	Carry → DC		
	PSUBC Sx,Sy,Dz	111110**********	Sx – Sy – DC → Dz	1	Borrow
		10100000xxyyzzzz	Borrow → DC		
	PCMP Sx,Sy	111110**********	Sx – Sy → DC update*	1	*
		10000100xxyy0000			
	PABS Sx,Dz	111110**********	If Sx < 0, 0 – Sx → Dz	1	*
		10001000xx00zzzz	If Sx > = 0, nop		
	PABS Sy,Dz	111110**********	If Sy < 0, 0 – Sy → Dz	1	*
		1010100000yyzzzz	If Sx > = 0, nop		
	PRND Sx,Dz	111110**********	Sx + h'00008000 → Dz	1	*
		10011000xx00zzzz	LSW of Dz → h'0000		
	PRND Sy,Dz	111110**********	Sy + h'00008000 → Dz	1	*
		1011100000yyzzzz	LSW of Dz → h'0000		

Note: * See table 2.32.

Table 2.32 DC Bit Update Definitions

CS [2:0]			Condition Mode	Description
0	0	0	Carry or borrow mode	The DC bit is set if an ALU arithmetic operation generates a carry or borrow, and is cleared otherwise. When a PSHA or PSHL shift instruction is executed, the last bit data shifted out is copied into the DC bit. When an ALU logical operation is executed, the DC bit is always cleared.
0	0	1	Negative value mode	When an ALU or shift (PSHA) arithmetic operation is executed, the MSB of the result, including the guard bits, is copied into the DC bit. When an ALU or shift (PSHL) logical operation is executed, the MSB of the result, excluding the guard bits, is copied into the DC bit.
0	1	0	Zero value mode	The DC bit is set if the result of an ALU or shift operation is all-zeros, and is cleared otherwise.
0	1	1	Overflow mode	The DC bit is set if the result of an ALU or shift (PSHA) arithmetic operation exceeds the destination register range, excluding the guard bits, and is cleared otherwise. When an ALU or shift (PSHL) logical operation is executed, the DC bit is always cleared.
1	0	0	Signed greater-than mode	This mode is similar to signed greater-or-equal mode, but DC is cleared if the result is all-zeros. DC = ~{(negative value ^ over-range) | zero value}; In case of arithmetic operation DC = 0; In case of logical operation
1	0	1	Signed greater-or-equal mode	If the result of an ALU or shift (PSHA) arithmetic operation exceeds the destination register range, including the guard bits ("over-range"), the definition is the same as in negative value mode. If the result is not over-range, the definition is the opposite of that in negative value mode. When an ALU or shift (PSHL) logical operation is executed, the DC bit is always cleared. DC = ~(negative value ^ over-range); In case of arithmetic operation DC = 0 ; In case of logical operation
1	1	0	Reserved	
1	1	1	Reserved	

Conditional Operations and Data Transfer: Some instructions belonging to this class can be executed conditionally, as described earlier. The specified condition is valid only for the B field of the instruction, and is not valid for data transfer instructions for which a parallel specification is made. Examples are shown in figure 2.17.

```
DCT PADD X0,Y0,A0  MOVX.W @R4+,X0  MOVY.W A0,@R6+R9 ;

When condition is True

Before execution: X0=H'33333333, Y0=H'55555555, A0=H'123456789A,
                  R4=H'00008000, R6=H'00008233, R9=H'00000004
                  (R4)=H'1111, (R6)=H'2222
After execution:  X0=H'11110000, Y0=H'55555555, A0=H'0088888888,
                  R4=H'00008002, R6=H'00008237, R9=H'00000004
                  (R4)=H'1111, (R6)=H'3456

When condition is False

Before execution: X0=H'33333333, Y0=H'55555555, A0=H'123456789A,
                  R4=H'00008000, R6=H'00008233, R9=H'00000004
                  (R4)=H'1111, (R6)=H'2222
After execution:  X0=H'11110000, Y0=H'55555555, A0=H'123456789A,
                  R4=H'00008002, R6=H'00008237, R9=H'00000004
                  (R4)=H'1111, (R6)=H'3456
```

Figure 2.17 Examples of Conditional Operations and Data Transfer Instructions

Assignment of NOPX and NOPY Instruction Codes: When there is no data transfer instruction to be parallel-processed simultaneously with a DSP operation instruction, an NOPX or NOPY instruction can be written as the data transfer instruction, or the instruction can be omitted. The instruction code is the same whether an NOPX or NOPY instruction is written or the instruction is omitted. Examples of NOPX and NOPY instruction codes are shown in table 2.33.

Table 2.33 Examples of NOPX and NOPY Instruction Codes

Instruction			Code
PADD X0,Y0,A0	MOVX.W @R4+,X0	MOVY.W @R6+R9,Y0	1111100000001011
			1011000100000111
PADD X0,Y0,A0	NOPX	MOVY.W @R6+R9,Y0	1111100000000011
			1011000100000111
PADD X0,Y0,A0	NOPX	NOPY	1111100000000000
			1011000100000111
PADD X0,Y0,A0	NOPX		1111100000000000
			1011000100000111
PADD X0,Y0,A0			1111100000000000
			1011000100000111
	MOVX.W @R4+,X0	MOVY.W @R6+R9,Y0	1111000000001011
	MOVX.W @R4+,X0	NOPY	1111000000001000
	MOVS.W @R4+,X0		1111010010001000
	NOPX	MOVY.W @R6+R9,Y0	1111000000000011
		MOVY.W @R6+R9,Y0	1111000000000011
	NOPX	NOPY	1111000000000000
NOP			0000000000001001

Section 3 DSP Operation

3.1 Data Operations of DSP Unit

3.1.1 ALU Fixed-Point Operations

Figure 3.1 shows the ALU arithmetic operation flow. Table 3.1 shows the variation of this type of operation and table 3.2 shows the correspondence between each operand and registers.

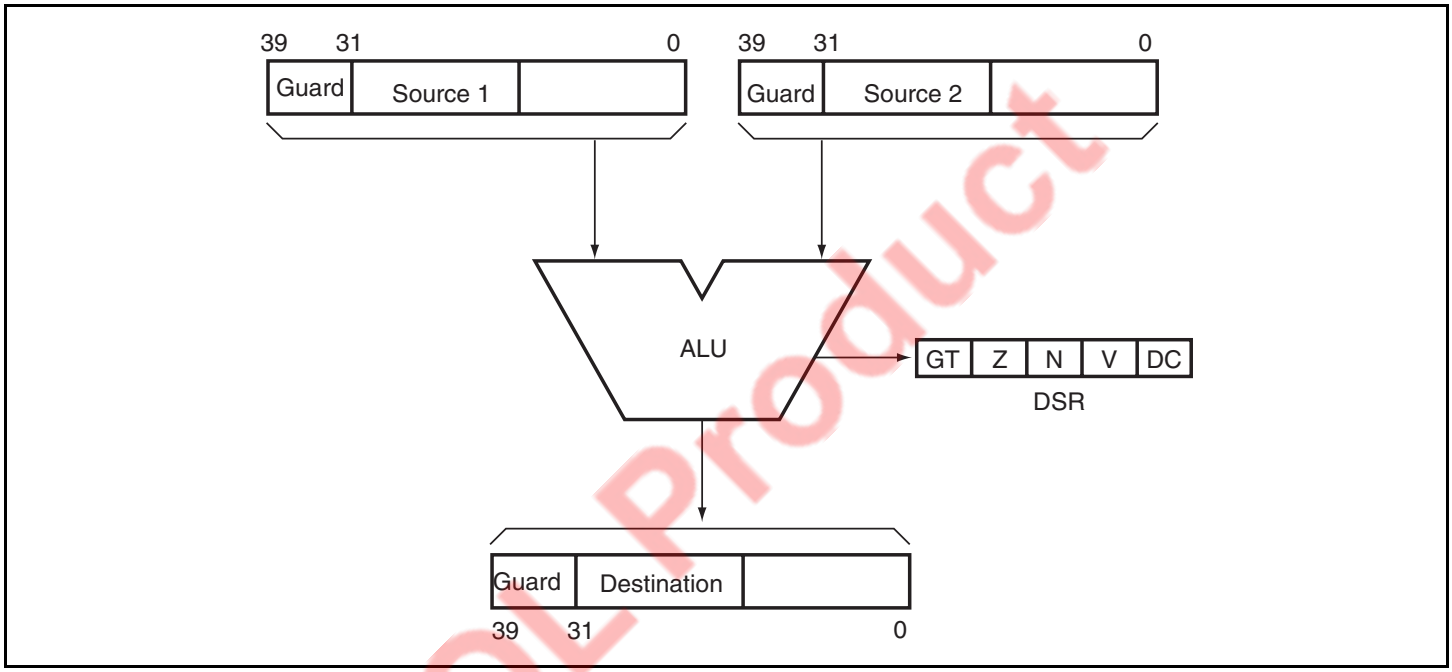


Figure 3.1 ALU Fixed-Point Arithmetic Operation Flow

Note: The ALU fixed-point arithmetic operations are basically 40-bit operation; 32 bits of the base precision and 8 bits of the guard-bit parts. So the signed bit is copied to the guard-bit parts when a register not providing the guard-bit parts is specified as the source operand. When a register not providing the guard-bit parts is specified as a destination operand, the lower 32 bits of the operation result are input into the destination register.

ALU fixed-point operations are executed between registers. Each source and destination operand are selected independently from one of the DSP registers. When a register providing guard bits is specified as an operand, the guard bits are activated for this type of operation. These operations are executed in the DSP stage, as shown in figure 3.2. The DSP stage is the same stage as the MA stage in which memory access is performed.

Table 3.1 Variation of ALU Fixed-Point Operations

Mnemonic	Function	Source 1	Source 2	Destination
PADD	Addition	Sx	Sy	Dz (Du)
PSUB	Subtraction	Sx	Sy	Dz (Du)
PADDC	Addition with carry	Sx	Sy	Dz
PSUBC	Subtraction with borrow	Sx	Sy	Dz
PCMP	Comparison	Sx	Sy	—
PCOPY	Data copy	Sx	All 0	Dz
		All 0	Sy	Dz
PABS	Absolute	Sx	All 0	Dz
		All 0	Sy	Dz
PNEG	Negation	Sx	All 0	Dz
		All 0	Sy	Dz
PCLR	Clear	All 0	All 0	Dz

Table 3.2 Correspondence between Operands and Registers

Register	Sx	Sy	Dz	Du
A0	Yes	—	Yes	Yes
A1	Yes	—	Yes	Yes
M0	—	Yes	Yes	—
M1	—	Yes	Yes	—
X0	Yes	—	Yes	Yes
X1	Yes	—	Yes	—
Y0	—	Yes	Yes	Yes
Y1	—	Yes	Yes	—

As shown in figure 3.2, data loaded from the memory at the MA stage, which is programmed at the same line as the ALU operation, is not used as a source operand for this operation, even though the destination operand of the data load operation is identical to the source operand of the ALU operation. In this case, previous operation results are used as the source operands for the ALU operation, and then updated as the destination operand of the data load operation.

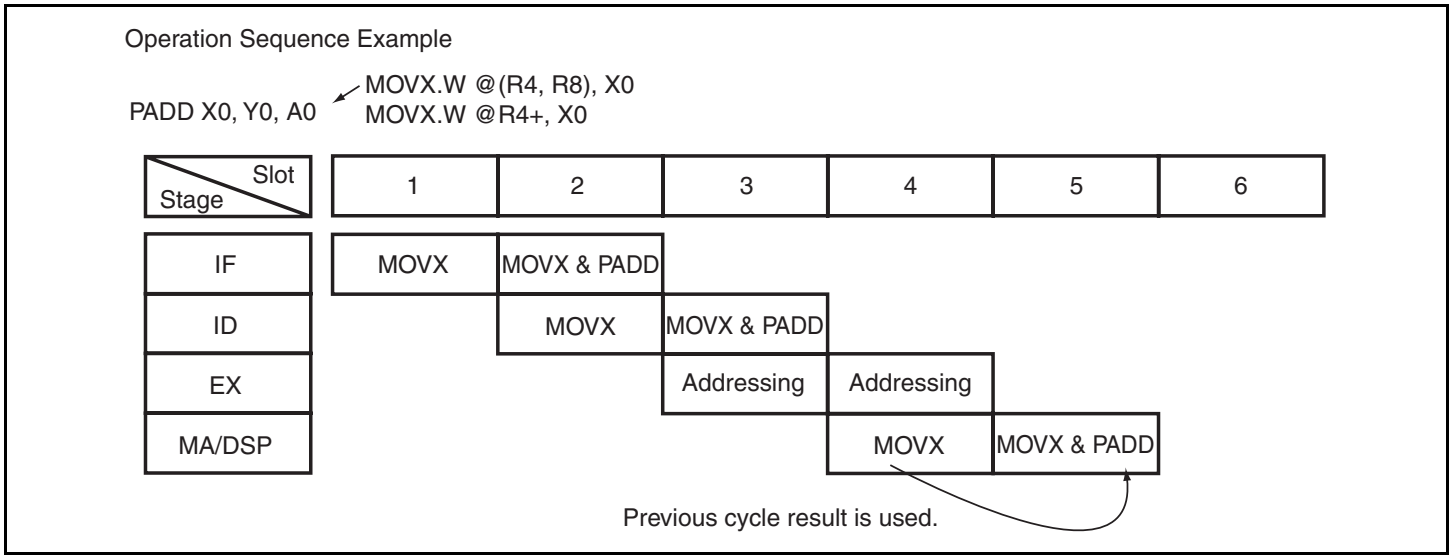


Figure 3.2 Operation Sequence Example

Every time an ALU arithmetic operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. However, in case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. The definition of a DC bit is selected by CS0 to CS2 (condition selection) bits in DSR. The DC bit result is as follows:

Carry or Borrow Mode: CS[2:0] = 000: The DC bit indicates that carry or borrow is generated from the most significant bit of the operation result, except the guard-bit parts. Some examples are shown in figure 3.3. This mode is the default condition. When the input data is negative in a PABS or PNEG instruction, carry is generated to add 1 to the LSB.

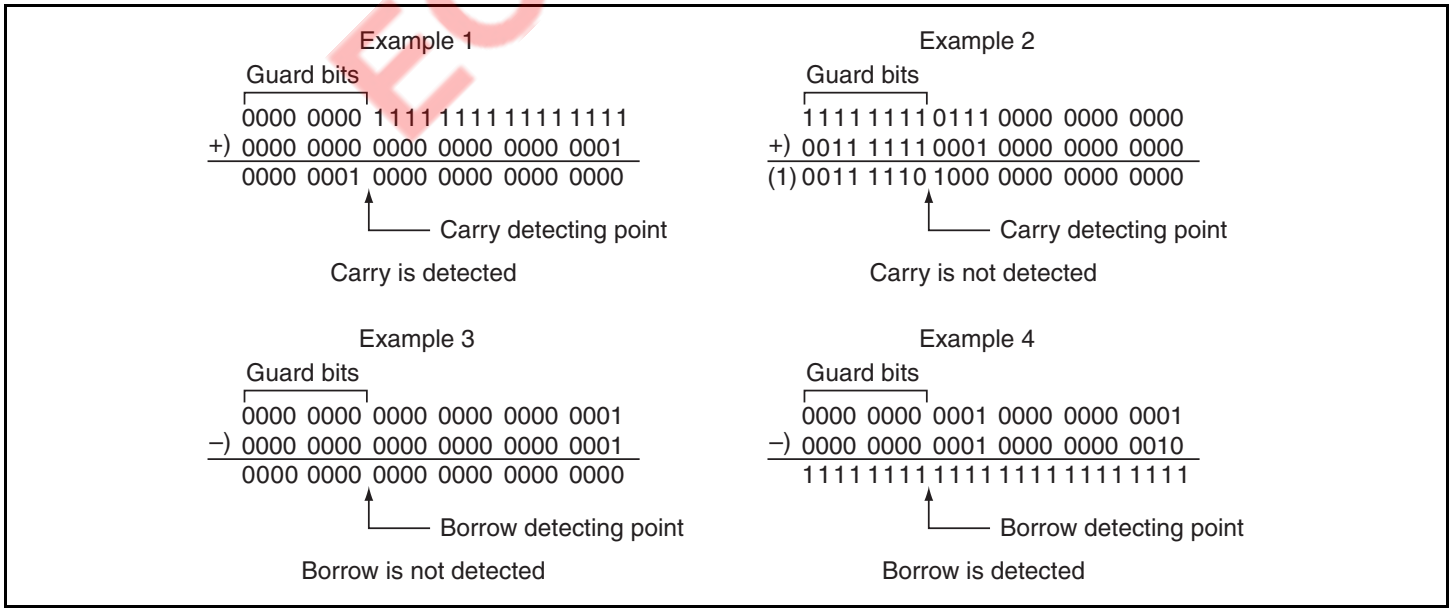


Figure 3.3 DC Bit Generation Examples in Carry or Borrow Mode

Negative Value Mode: CS[2:0] = 001: The DC flag indicates the same state as the MSB of the operation result. When the result is a negative number, the DC bit shows 1. When it is a positive number, the DC bit shows 0. The ALU always executes 40-bit arithmetic operation, so the sign bit to detect whether positive or negative is always got from the MSB of the operation result regardless of the destination operand. Some examples are shown in figure 3.4.

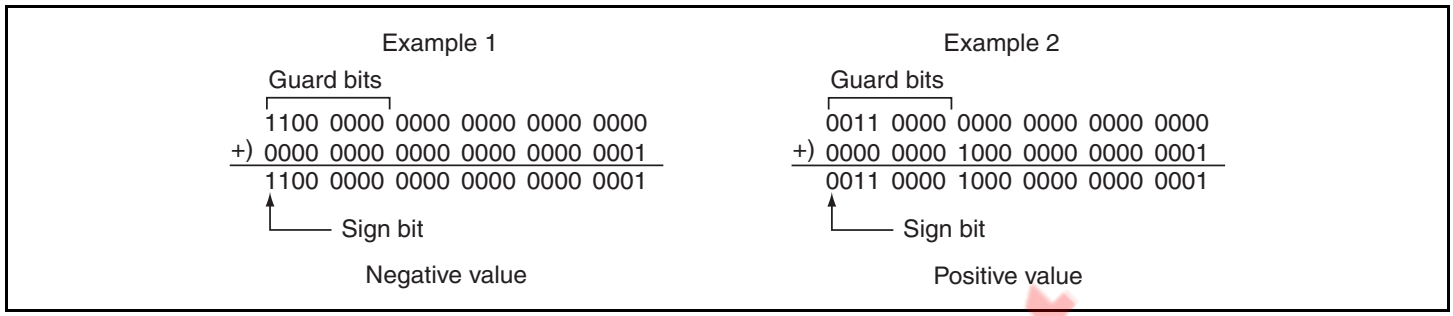


Figure 3.4 DC Bit Generation Examples in Negative Value Mode

Zero Value Mode: CS[2:0] = 010: The DC flag indicates whether the operation result is 0 or not. When the result is 0, the DC bit shows 1. When it is not 0, the DC bit shows 0.

Overflow Mode: CS[2:0] = 011: The DC bit indicates whether or not overflow occurs in the result. When an operation yields a result beyond the range of the destination register, except the guard-bit parts, the DC bit is set. Even though guard bits are provided, the DC bit always indicates the result of when no guard bits are provided. So, the DC bit is always set if the guard-bit parts are used for large number representation. Some DC bit generation examples in overflow mode are shown in figure 3.5.

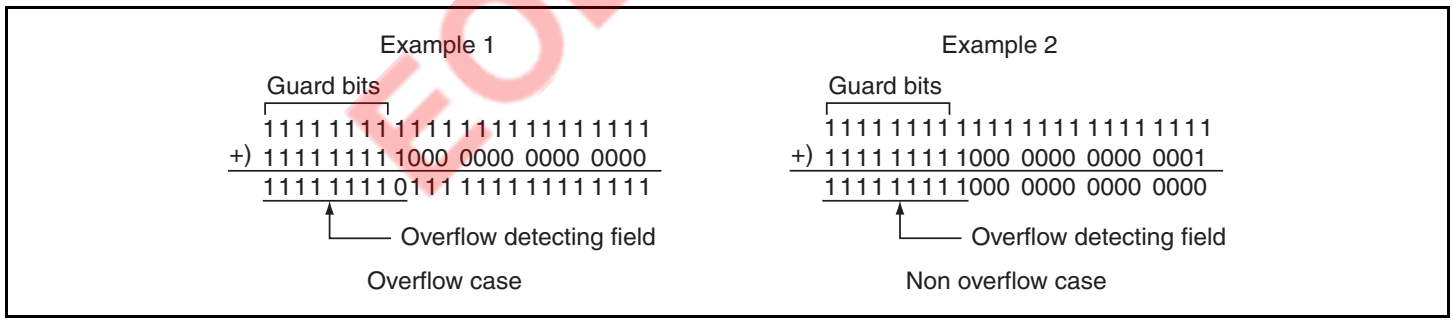


Figure 3.5 DC Bit Generation Examples in Overflow Mode

Signed Greater Than Mode: CS[2:0] = 100: The DC bit indicates whether or not the source 1 data (signed) is greater than the source 2 data (signed) as the result of compare operation PCMP. So, a PCMP operation should be executed in advance when a conditional operation is executed under this condition mode. This mode is similar to the Negative Value Mode described before, because the result of a compare operation is usually a positive value if the source 1 data is greater than the source 2 data. However, the signed bit of the result shows a negative value if the compare operation yields a result beyond the range of the destination operand, including the guard-bit parts (called "Over-range"), even though the source 1 data is greater than the source 2 data. The DC bit is updated concerning this type of special case in this condition mode. The equation below shows the definition of getting this condition:

DC = ~ {(Negative ^ Over-range) | Zero}

When the PCMP operation is executed under this condition mode, the result of the DC bit is the same as the T bit's result of the CMP/GT operation of the SH core instruction.

Signed Greater Than or Equal Mode: CS[2:0] = 101: The DC bit indicates whether the source 1 data (signed) is greater than or equal to the source 2 data (signed) as the result of compare operation PCMP. So, a PCMP operation should be executed in advance when a conditional operation is executed under this condition mode. This mode is similar to the Signed Greater Than Mode described before but the equal case is also included in this mode. The equation below shows the definition of getting this condition:

DC = ~ (Negative ^ Over-range)

When the PCMP operation is executed under this condition mode, the result of the DC bit is the same as the T bit's result of a CMP/GE operation of the SH core instruction.

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in overflow mode by the CS[2:0] bits. See the overflow mode part above. The GT bit always indicates the same state as the DC bit set in signed greater than mode by the CS[2:0] bits. See the signed greater than mode part above.

Note: The DC bit is always updated as the carry flag for "PADDC" and is always updated as the borrow flag for "PSUBC" regardless of the CS[2:0] state.

Overflow Protection: The S bit in SR is effective for any ALU fixed-point arithmetic operations in the DSP unit. See section 3.1.8, Overflow Protection, for details.

3.1.2 ALU Integer Operations

Figure 3.6 shows the ALU integer arithmetic operation flow. Table 3.3 shows the variation of this type of operation. The correspondence between each operand and registers is the same as ALU fixed-point operations as shown in table 3.2.

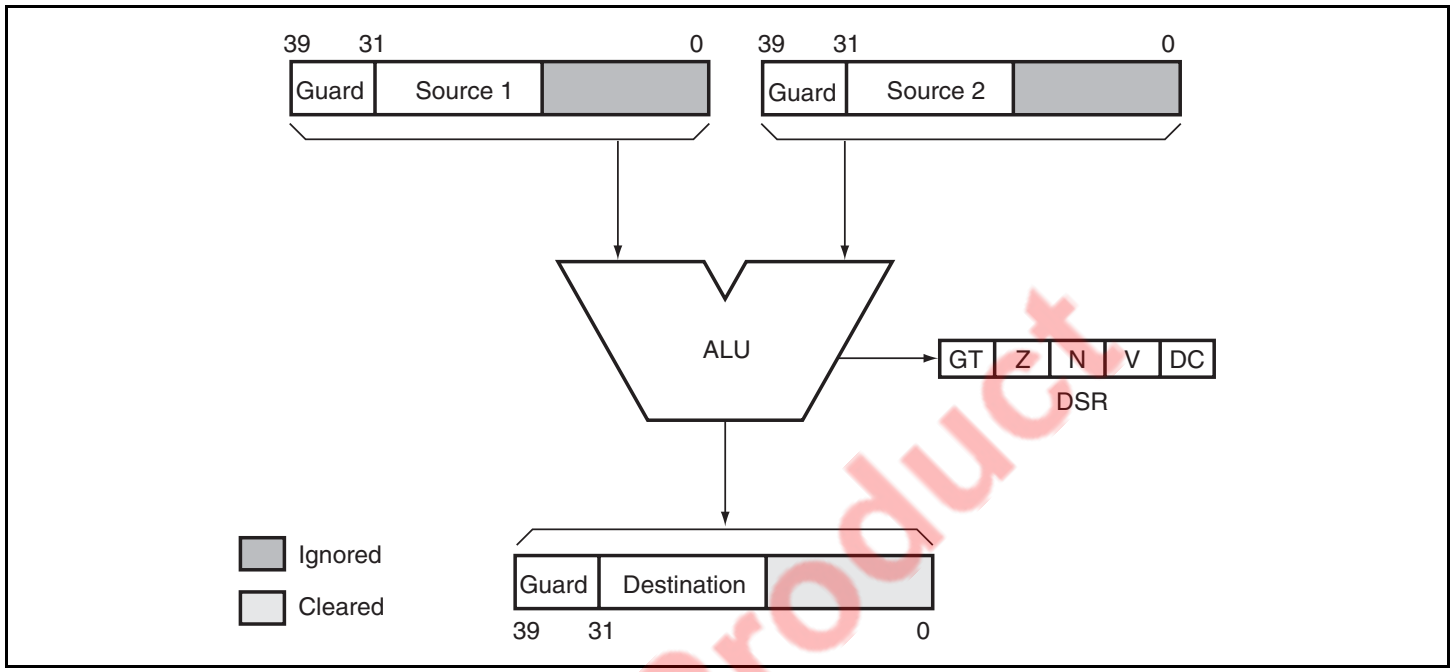


Figure 3.6 ALU Integer Arithmetic Operation Flow

Table 3.3 Variation of ALU Integer Operations

Mnemonic	Function	Source 1	Source 2	Destination
PINC	Increment by 1	Sx	+1	Dz
		+1	Sy	Dz
PDEC	Decrement by 1	Sx	–1	Dz
		–1	Sy	Dz

Note: The ALU integer operations are basically 24-bit operation, the upper 16 bits of the base precision and 8 bits of the guard bits parts. So the signed bit is copied to the guard-bit parts when a register not providing the guard-bit parts is specified as the source operand. When a register not providing the guard-bit parts is specified as a destination operand, the upper word excluding the guard bits of the operation result are input into the destination register.

In ALU integer arithmetic operations, the lower word of the source operand is ignored and the lower word of the destination operand is automatically cleared. The guard-bit parts are effective in integer arithmetic operations if they are supported. Others are basically the same operation as ALU fixed-point arithmetic operations. As shown in table 3.3, however, this type of operation provides two kinds of instructions only, so that the second operand is actually either +1 or –1. When a word data is loaded into one of the DSP unit's registers, it is input as an upper word data. When a register providing guard bits is specified as an operand, the guard bits are also activated. These operations, as well as fixed-point operations, are executed in the DSP stage, as shown in figure 3.2. The DSP stage is the same stage as the MA stage in which memory access is performed.

Every time an ALU arithmetic operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. This is the same as fixed-point operations but the lower word of each source and destination operand is not used in order to generate them. See section 3.1.1, ALU Fixed-Point Operations, for details.

In case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. See section 3.1.1, ALU Fixed-Point Operations, for details.

Overflow Protection: The S bit in SR is effective for any ALU integer arithmetic operations in DSP unit. See section 3.1.8, Overflow Protection, for details.

3.1.3 ALU Logical Operations

Figure 3.7 shows the ALU logical operation flow. Table 3.4 shows the variation of this type of operation. The correspondence between each operand and registers is the same as the ALU fixed-point operations as shown in table 3.2.

Logical operations are also executed between registers. Each source and destination operand are selected independently from one of the DSP registers. As shown in figure 3.7, this type of operation uses only the upper word of each operand. The lower word and guard-bit parts are ignored for the source operand and those of the destination operand are automatically cleared. These operations are also executed in the DSP stage, as shown in figure 3.2. The DSP stage is the same stage as the MA stage in which memory access is performed.

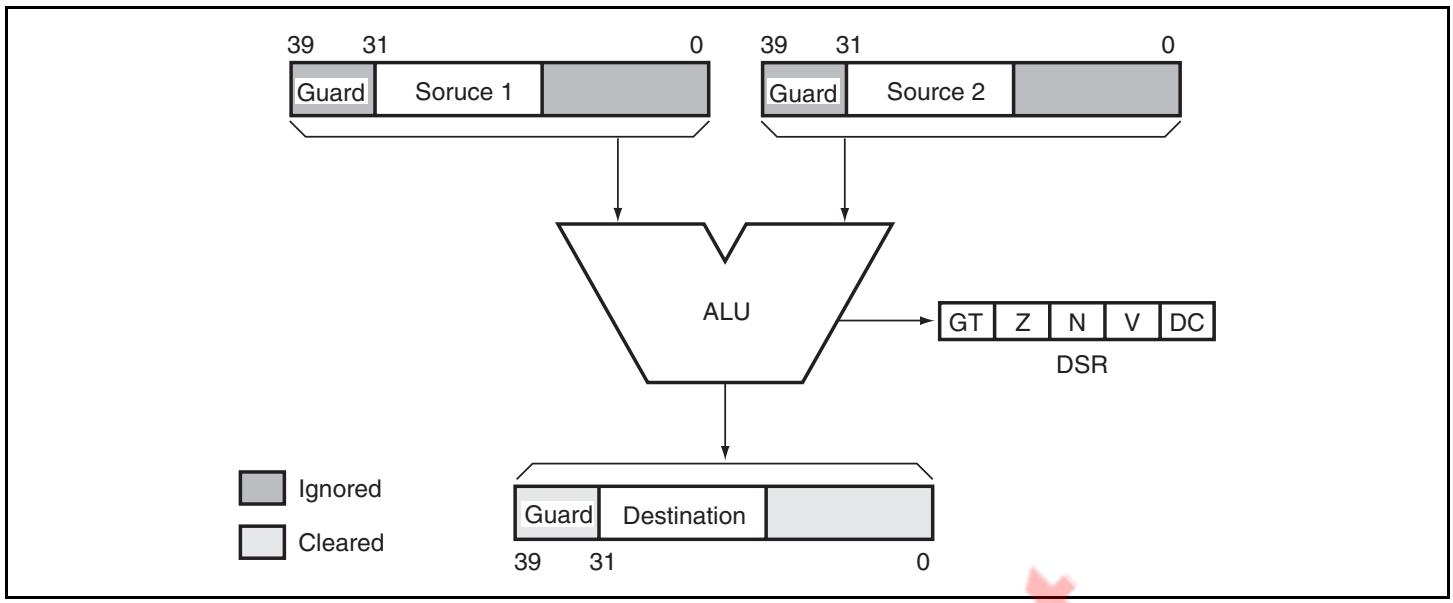


Figure 3.7 ALU Logical Operation Flow

Table 3.4 Variation of ALU Logical Operations

Mnemonic	Function	Source 1	Source 2	Destination
PAND	Logical AND	Sx	Sy	Dz
POR	Logical OR	Sx	Sy	Dz
PXOR	Logical exclusive OR	Sx	Sy	Dz

Every time an ALU logical operation is executed, the DC, N, Z, V, and GT bits in the DSR register are basically updated in accordance with the operation result. In case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. The definition of the DC bit is selected by the CS0 to CS2 (condition selection) bits in DSR. The DC bit result is:

1. Carry or Borrow Mode: CS[2:0] = 000
   The DC bit is always cleared.
2. Negative Value Mode: CS[2:0] = 001
   Bit 31 of the operation result is loaded into the DC bit.
3. Zero Value Mode: CS[2:0] = 010
   The DC bit is set when the operation result is zero; otherwise it is cleared.
4. Overflow Mode: CS[2:0] = 011
   The DC bit is always cleared.

5. Signed Greater Than Mode: CS[2:0] = 100
   The DC bit is always cleared.
6. Signed Greater Than or Equal Mode: CS[2:0] = 101
   The DC bit is always cleared.

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in overflow mode by the CS[2:0] bits. See the overflow mode part above. The GT bit always indicates the same state as the DC bit set in signed greater than mode by the CS[2:0] bits. See the signed greater than mode part above.

3.1.4 Fixed-Point Multiply Operation

Figure 3.8 shows the multiply operation flow. Table 3.5 shows the variation of this type of operation and table 3.6 shows the correspondence between each operand and registers. The multiply operation of the DSP unit is single-word signed single-precision multiplication. These operations are executed in the DSP stage, as shown in figure 3.2. The DSP stage is the same stage as the MA stage in which memory access is performed.

If a double-precision multiply operation is needed, the SH-3's standard double-word multiply instructions can be made of use.

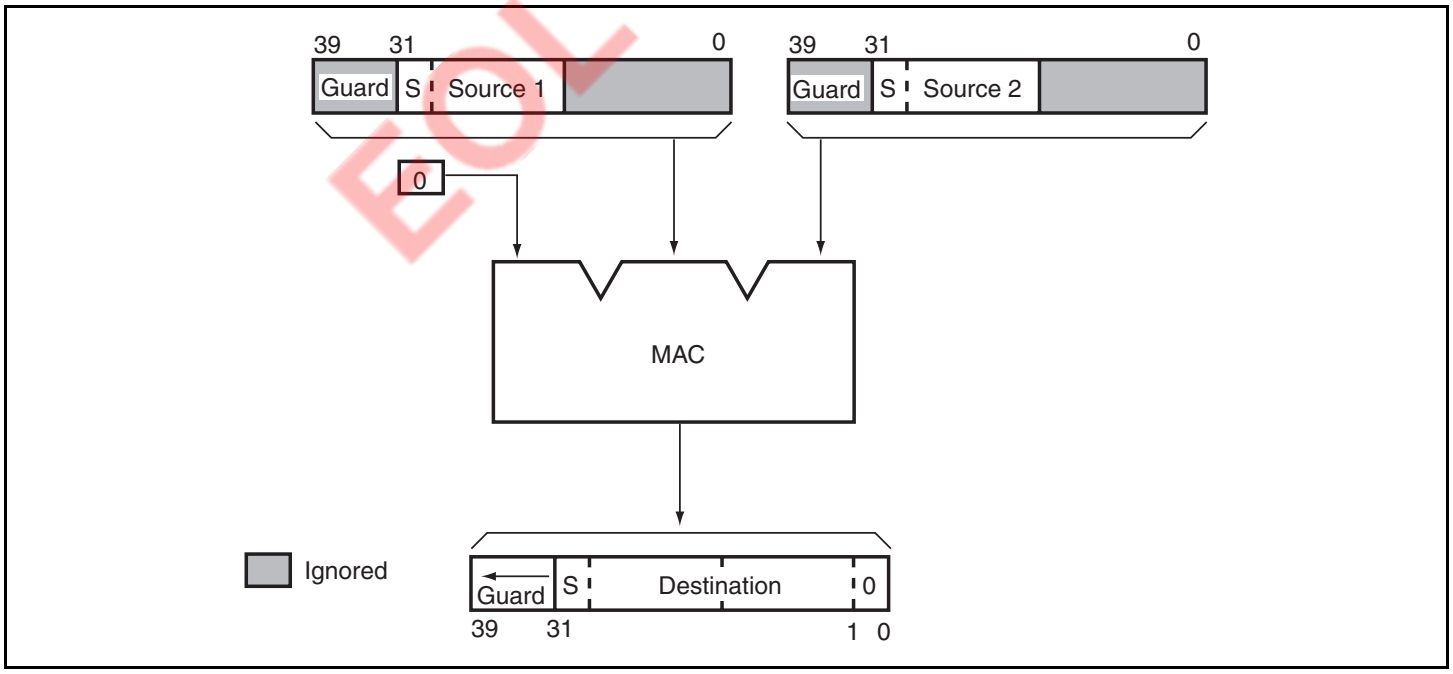


Figure 3.8 Fixed-Point Multiply Operation Flow

Table 3.5 Variation of Fixed-Point Multiply Operation

Mnemonic	Function	Source 1	Source 2	Destination
PMULS	Signed multiplication	Se	Sf	Dg

Table 3.6 Correspondence between Operands and Registers

Register	Se	Sf	Dg
A0	—	—	Yes
A1	Yes	Yes	Yes
M0	—	—	Yes
M1	—	—	Yes
X0	Yes	Yes	—
X1	Yes	—	—
Y0	Yes	Yes	—
Y1	—	Yes	—

Note: The multiply operations basically generate 32-bit operation results. So when a register providing the guard-bit parts are specified as a destination operand, the guard-bit parts will copy bit 31 of the operation result.

The multiply operation of the DSP unit side is not integer but fixed-point arithmetic. So, the upper words of each multiplier and multiplicand are input into a MAC unit as shown in figure 3.8. In the SH's standard multiply operations, the lower words of both source operands are input into a MAC unit. The operation result is also different from the SH's case. The SH's multiply operation result is aligned to the LSB of the destination, but the fixed-point multiply operation result is aligned to the MSB, so that the LSB of the fixed-point multiply operation result is always 0.

This fixed-point multiply operation is executed in one cycle Multiply operation is always unconditional, but does not affect any condition code bits, DC, N, Z, V and GT, in DSR.

Overflow Protection: The S bit in SR is effective for this multiply operation in the DSP unit. See section 3.1.8, Overflow Protection, for details.

If the S bit is 0, overflow occurs only when `H'8000*H'8000 ((-1.0)*(-1.0))` operation is executed as signed fixed-point multiply. The result is `H'00 8000 0000` but it does not mean (`+1.0`). If the S bit is 1, overflow is prevented and the result is `H'00 7FFF FFFF`.

3.1.5 Shift Operations

Shift operations can use either register or immediate value as the shift amount operand. Other source and destination operands are specified by the register. There are two kinds of shift operations. Table 3.7 shows the variation of this type of operation. The correspondence between each operand and registers, except for immediate operands, is the same as the ALU fixed-point operations as shown in table 3.2.

Table 3.7 Variation of Shift Operations

Mnemonic	Function	Source 1	Source 2	Destination
PSHA Sx, Sy, Dz	Arithmetic shift	Sx	Sy	Dz
PSHL Sx, Sy, Dz	Logical shift	Sx	Sy	Dz
PSHA #Imm1, Dz	Arithmetic shift with immediate.	Dz	Imm1	Dz
PSHL #Imm2, Dz	Logical shift with immediate.	Dz	Imm2	Dz

Note: –32 <= Imm1 <= +32, –16 <= Imm2 <= +16

Arithmetic Shift: Figure 3.9 shows the arithmetic shift operation flow.

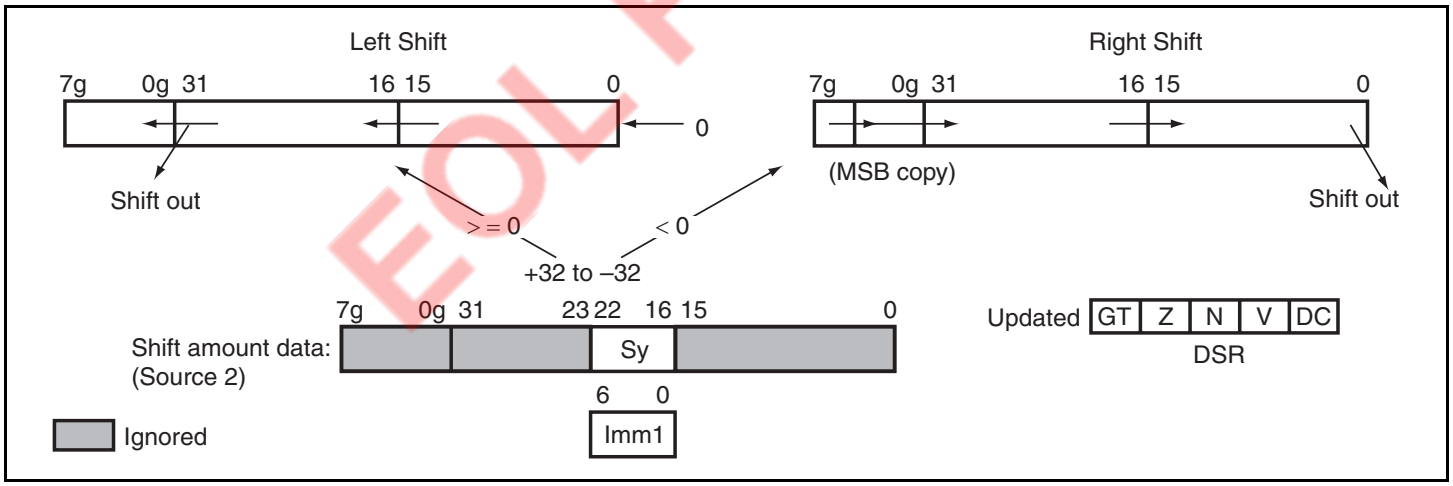


Figure 3.9 Arithmetic Shift Operation Flow

Note: The arithmetic shift operations are basically 40-bit operation, that is, the 32 bits of the base precision and 8 bits of the guard-bit parts. So the signed bit is copied to the guard-bit parts when a register not providing the guard-bit parts is specified as the source operand. When a register not providing the guard-bit parts is specified as a destination operand, the lower 32 bits of the operation result are input into the destination register.

In this arithmetic shift operation, all bits of the source 1 and destination operands are activated. The shift amount is specified by the source 2 operand as an integer data. The source 2 operand can be specified by either a register or immediate operand. The available shift range is from –32 to +32. Here, a negative value means the right shift, and a positive value means the left shift. It is possible for any source 2 operand to specify from –64 to +63 but the result is unknown if an invalid shift value is specified. In case of a shift with an immediate operand instruction, the source 1 operand must be the same register as the destination's. This operation is executed in the DSP stage, as shown in figure 3.2 as well as in fixed-point operations. The DSP stage is the same stage as the MA stage in which memory access is performed.

Every time an arithmetic shift operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. In case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. The definition of the DC bit is selected by the CS[2:0] (condition selection) bits in DSR. The DC bit result is:

1. Carry or Borrow Mode: CS[2:0] = 000
   The DC bit indicates the last shifted out data as the operation result.
2. Negative Value Mode: CS[2:0] = 001
   The DC bit is set when the operation result is a negative value, and cleared when the operation result is zero or a positive value.
3. Zero Value Mode: CS[2:0] = 010
   The DC bit is set when the operation result is zero; otherwise it is cleared.
4. Overflow Mode: CS[2:0] = 011
   The DC bit is always cleared.
5. Signed Greater Than Mode: CS[2:0] = 100
   The DC bit is always cleared.
6. Signed Greater Than or Equal Mode: CS[2:0] = 101
   The DC bit is always cleared.

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in overflow mode by the CS[2:0] bits. See the overflow mode part above. The GT bit always indicates the same state as the DC bit set in signed greater than mode by the CS[2:0] bits. See the signed greater than mode part above.

Overflow Protection: The S bit in SR is also effective for arithmetic shift operation in the DSP unit. See section 3.1.8, Overflow Protection, for details.

Logical Shift: Figure 3.10 shows the logical shift operation flow.

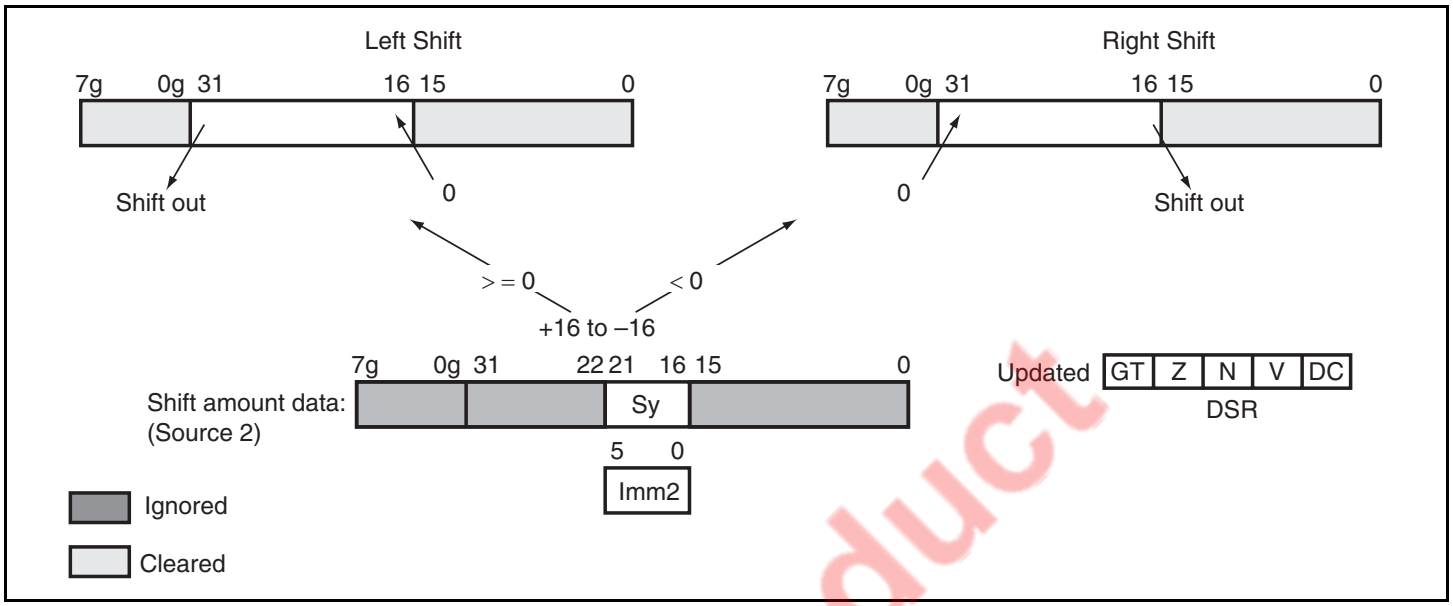


Figure 3.10 Logical Shift Operation Flow

As shown in figure 3.10, the logical shift operation uses the upper word of the source 1 operand and the destination operand. The lower word and guard-bit parts are ignored for the source operand and those of the destination operand are automatically cleared as in the ALU logical operations. The shift amount is specified by the source 2 operand as an integer data. The source 2 operand can be specified by either the register or immediate operand. The available shift range is from –16 to +16. Here, a negative value means the right shift, and a positive value means the left shift. It is possible for any source 2 operand to specify from –32 to +31, but the result is unknown if an invalid shift value is specified. In case of a shift with an immediate operand instruction, the source 1 operand must be the same register as the destination's. These operations are executed in the DSP stage, as shown in figure 3.2. The DSP stage is the same stage as the MA stage in which memory access is performed.

Every time a logical shift operation is executed, the DC, N, Z, V and GT bits in DSR are basically updated in accordance with the operation result. In case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. The definition of the DC bit is selected by the CS0 to CS2 (condition selection) bits in DSR. The DC bit result is:

1. Carry or Borrow Mode: CS[2:0] = 000
   The DC bit indicates the last shifted out data as the operation result.
2. Negative Value Mode: CS[2:0] = 001
   Bit 31 of the operation result is loaded into the DC bit.
3. Zero Value Mode: CS[2:0] = 010
   The DC bit is set when the operation result is zero; otherwise it is cleared.
4. Overflow Mode: CS[2:0] = 011
   The DC bit is always cleared.
5. Signed Greater Than Mode: CS[2:0] = 100
   The DC bit is always cleared.
6. Signed Greater Than or Equal Mode: CS[2:0] = 101
   The DC bit is always cleared.

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in overflow mode by the CS[2:0] bits, but it is always cleared in this operation. So is the GT bit.

3.1.6 Most Significant Bit Detection Operation

The PDMSB, most significant bit detection operation, is used to calculate the shift amount for normalization. Figure 3.11 shows the PDMSB operation flow and table 3.8 shows the operation definition. Table 3.9 shows the possible variations of this type of operation. The correspondence between each operand and registers is the same as for ALU fixed-point operations, as shown in table 3.2.

Note: The result of the MSB detection operation is basically 24 bits as well as ALU integer operation, the upper 16 bits of the base precision and 8 bits of the guard-bit parts. When a register not providing the guard-bit parts is specified as a destination operand, the upper word of the operation result is input into the destination register.

As shown in figure 3.11, the PDMSB operation uses all bits as a source operand, but the destination operand is treated as an integer operation result because shift amount data for normalization should be integer data as described in section 3.1.5 Shift Operations, Arithmetic Shift. These operations are executed in the DSP stage, as shown in figure 3.2. The DSP stage is the same stage as the MA stage in which memory access is performed.

Every time a PDMSB operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. In case of a conditional operation, they are not updated, even though the specified condition is true, and the operation is executed. In case of an unconditional operation, they are always updated with the operation result.

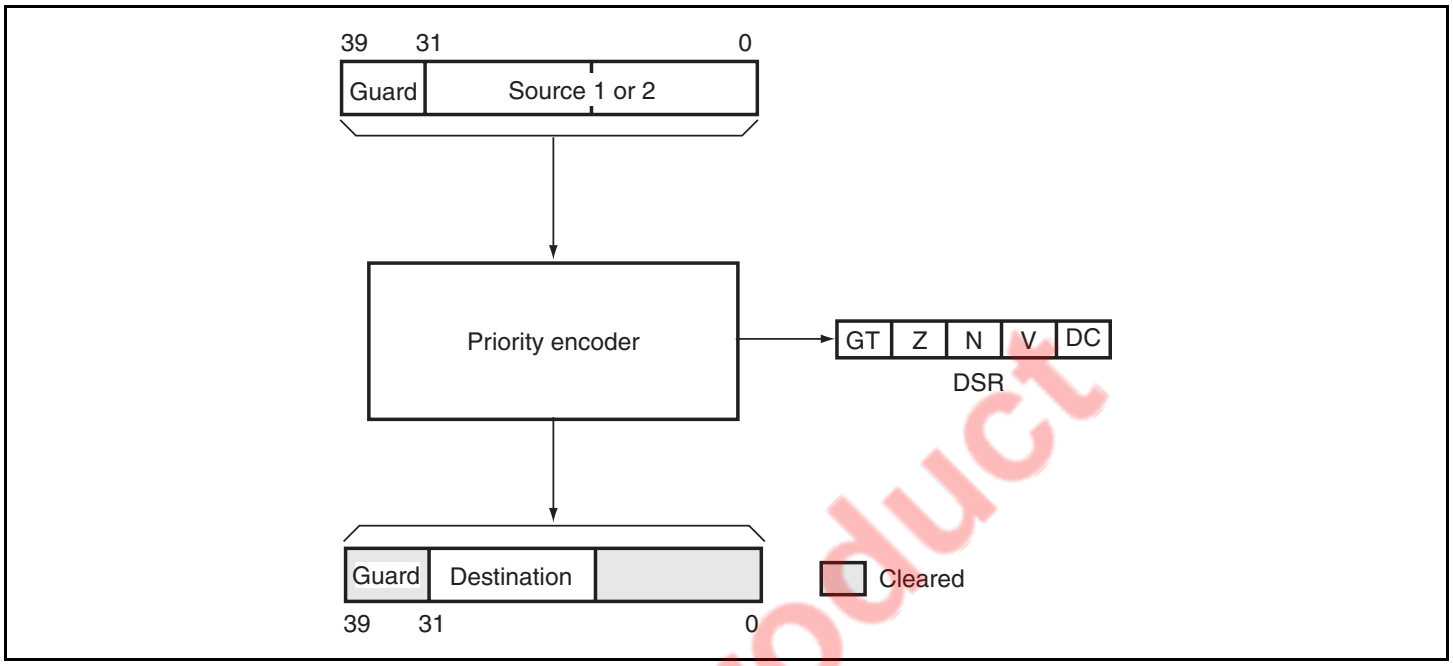


Figure 3.11 PDMSB Operation Flow

The definition of the DC bit is selected by the CS0 to CS2 (condition selection) bits in DSR. The DC bit result is:

1. Carry or Borrow Mode: CS[2:0] = 000
   The DC bit is always cleared.
2. Negative Value Mode: CS[2:0] = 001
   The DC bit is set when the operation result is a negative value, and cleared when the operation result is zero or a positive value.
3. Zero Value Mode: CS[2:0] = 010
   The DC bit is set when the operation result is zero; otherwise it is cleared.
4. Overflow Mode: CS[2:0] = 011
   The DC bit is always cleared.
5. Signed Greater Than Mode: CS[2:0] = 100
   The DC bit is set when the operation result is a positive value; otherwise it is cleared.
6. Signed Greater Than or Equal Mode: CS[2:0] = 101
   The DC bit is set when the operation result is zero or a positive value; otherwise it is cleared.

Table 3.8 Operation Definition of PDMSB

Source Data														Result for DST							
Guard Bit					Upper Word					Lower Word				Guard Bit	Upper Word						
7g	6g	…	1g	0g	31	30	29	28	…	3	2	1	0	7g to 0g	31 to 22	21	20	19	18	17	16	Decimal
0	0	…	0	0	0	0	0	0	…	0	0	0	0	All 0	All 0	0	1	1	1	1	1	+31
0	0	…	0	0	0	0	0	0	…	0	0	0	1	All 0	All 0	0	1	1	1	1	0	+30
0	0	…	0	0	0	0	0	0	…	0	0	1	*	All 0	All 0	0	1	1	1	0	1	+29
0	0	…	0	0	0	0	0	0	…	0	1	*	*	All 0	All 0	0	1	1	1	0	0	+28
		:							:									:				
0	0	…	0	0	0	0	0	1	…	*	*	*	*	All 0	All 0	0	0	0	0	1	0	+2
0	0	…	0	0	0	0	1	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0	1	+1
0	0	…	0	0	0	1	*	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0	0	0
0	0	…	0	0	1	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1	1	–1
0	0	…	0	1	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1	0	–2
		:							:									:				
0	1	…	*	*	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	0	0	0	–8
1	0	…	*	*	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	0	0	0	–8
																		:				
1	1	…	1	0	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1	0	–2
1	1	…	1	1	0	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1	1	–1
1	1	…	1	1	1	0	*	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0	0	0
1	1	…	1	1	1	1	0	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0	1	+1
1	1	…	1	1	1	1	1	0	…	*	*	*	*	All 0	All 0	0	0	0	0	1	0	+2
		:							:									:				
1	1	…	1	1	1	1	1	1	…	1	0	*	*	All 0	All 0	0	1	1	1	0	0	+28
1	1	…	1	1	1	1	1	1	…	1	1	0	*	All 0	All 0	0	1	1	1	0	1	+29
1	1	…	1	1	1	1	1	1	…	1	1	1	0	All 0	All 0	0	1	1	1	1	0	+30
1	1	…	1	1	1	1	1	1	…	1	1	1	1	All 0	All 0	0	1	1	1	1	1	+31

Note: * means Don't care.

Table 3.9 Variation of PDMSB Operation

Mnemonic	Function	Source	Source 2	Destination
PDMSB	MSB detection	Sx	—	Dz
		—	Sy	Dz

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in overflow mode by the CS[2:0] bits. See the overflow mode part above. The GT bit always indicates the same state as the DC bit set in signed greater than mode by the CS[2:0] bits. See the signed greater than mode part above.

3.1.7 Rounding Operation

The DSP unit provides the rounding function that rounds from 32 bits to 16 bits. In case of providing guard-bit parts, it rounds from 40 bits to 24 bits. When a round instruction is executed, H'00008000 is added to the source operand data and then, the lower word is cleared. Figure 3.12 shows the rounding operation flow and figure 3.13 shows the operation definition. Table 3.10 shows the variation of this type of operation. The correspondence between each operand and registers is the same as ALU fixed-point operations as shown in table 3.2.

As shown in figure 3.12, the rounding operation uses full-size data for both source and destination operands. These operations are executed in the DSP stage as shown in figure 3.2. The DSP stage is the same stage as the MA stage in which memory access is performed.

The rounding operation is always executed unconditionally, so that the DC, N, Z, V, and GT bits in DSR are always updated in accordance with the operation result. The definition of the DC bit is selected by the CS[2:0] (condition selection) bits in DSR. The result of these condition code bits is the same as the ALU-fixed point arithmetic operations.

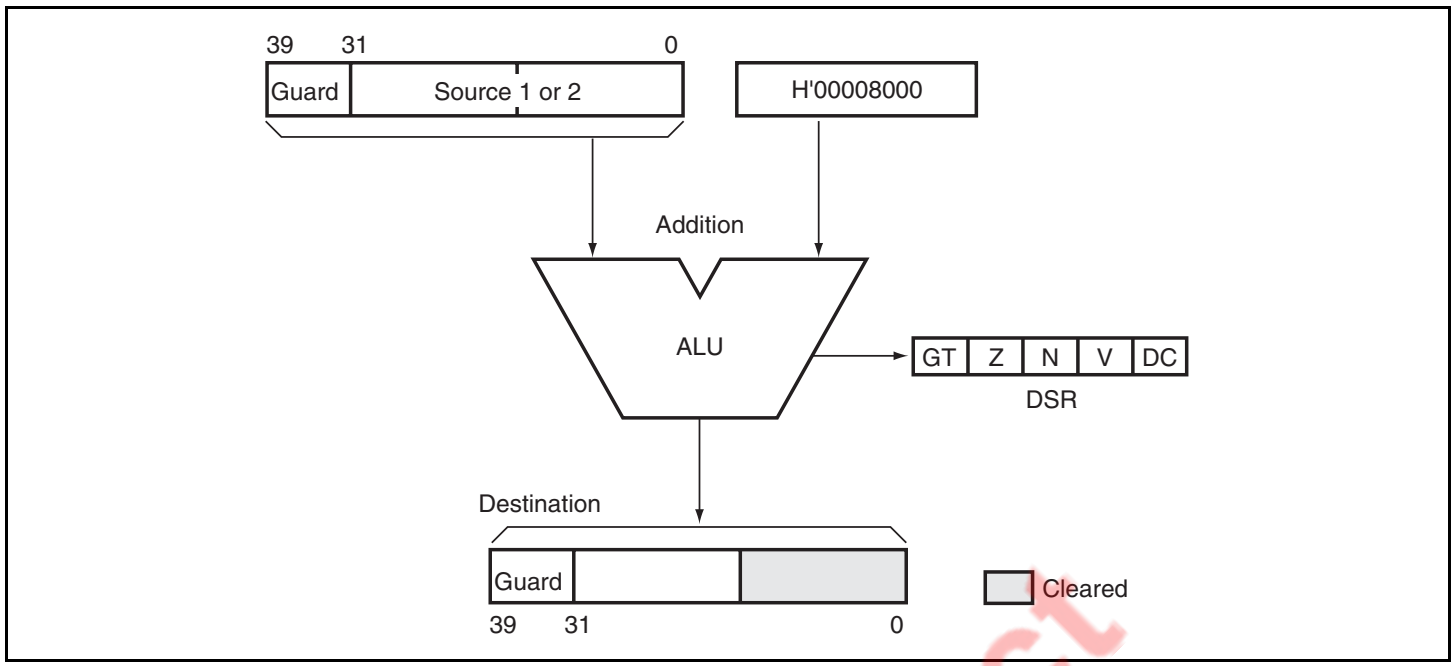


Figure 3.12 Rounding Operation Flow

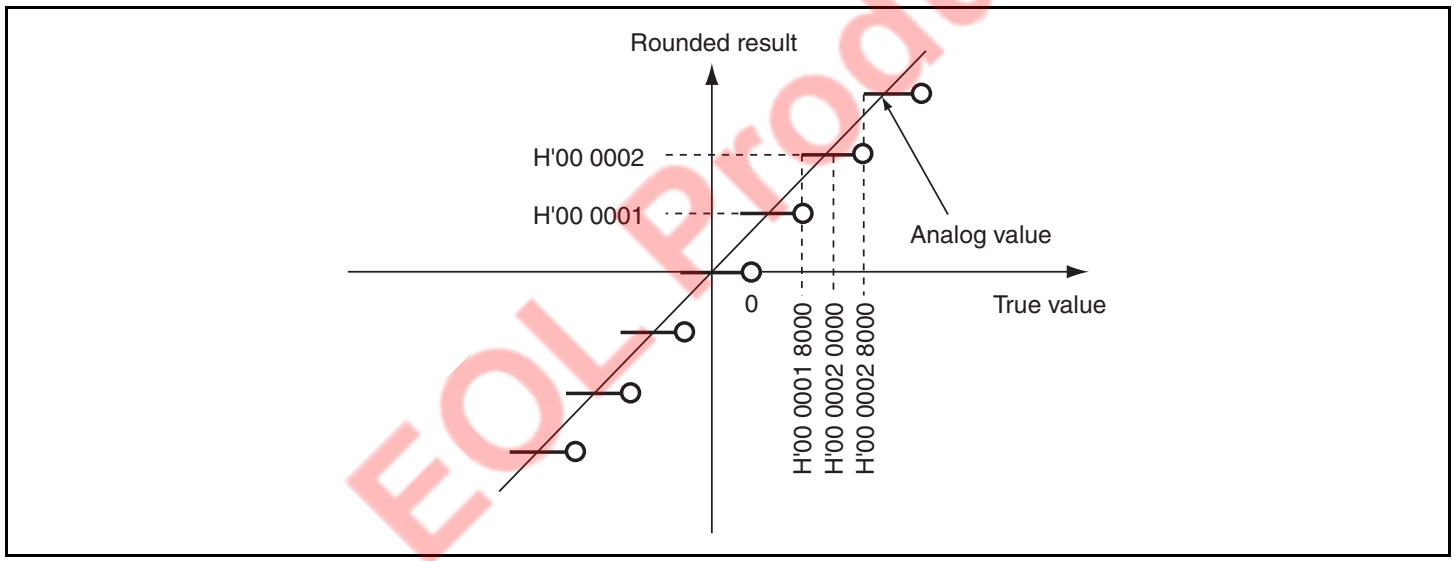


Figure 3.13 Definition of Rounding Operation

Table 3.10 Variation of Rounding Operation

Mnemonic	Function	Source 1	Source 2	Destination
PRND	Rounding	Sx	—	Dz
		—	Sy	Dz

Overflow Protection: The S bit in SR is effective for any rounding operations in the DSP unit. See section 3.1.8, Overflow Protection, for details.

3.1.8 Overflow Protection

The S bit in SR is effective for any arithmetic operations executed in the DSP unit, including the SH's standard multiply and MAC operations. The S bit in SR, in SH's CPU core, is used as the overflow protection enable bit. The arithmetic operation overflows when the operation result exceeds the range of two's complement representation without guard-bit parts. Table 3.11 shows the definition of overflow protection for fixed-point arithmetic operations, including fixed-point signed by signed multiplication described in section 3.1.4, Fixed-Point Multiply Operation. Table 3.12 shows the definition of overflow protection for integer arithmetic operations. When an SH's standard multiply or MAC operation is executed, the S bit function is completely the same as the current SH CPU's definition.

When the overflow protection is effective, overflow never occurs. So, the V bit is cleared, and the DC bit is also cleared when the overflow mode is selected by the CS[2:0] bits.

Table 3.11 Definition of Overflow Protection for Fixed-Point Arithmetic Operations

Sign	Overflow Condition	Fixed Value	Hex Representation
Positive	Result > $1 - 2^{-31}$	$1 - 2^{-31}$	00 7FFF FFFF
Negative	Result < –1	–1	FF 8000 0000

Table 3.12 Definition of Overflow Protection for Integer Arithmetic Operations

Sign	Overflow Condition	Fixed Value	Hex Representation
Positive	Result > $2^{15} - 1$	$2^{15} - 1$	00 7FFF ****
Negative	Result < -2^{15}	-2^{15}	FF 8000 ****

Note: * means Don't care.

3.1.9 Data Transfer Operation

This LSI can execute a maximum of two data transfer operations between the DSP register and the on-chip data memory in parallel for the DSP unit. Three types of data transfer instructions are provided for the DSP unit.

1. Parallel operation type (using XDB and YDB buses)
2. Double data transfer type (using XDB or YDB buses)
3. Single data transfer type (using LDB bus)

The type 1 instructions execute both DSP data processing and data transfer operations in parallel. The 32-bit instruction code is used for this type of instruction. Basically, two data transfer operations can be specified by this type of instruction, but they don't always have to be specified.

One data transfer is for X memory and another is for Y memory. Both of these data transfer operations cannot be executed for one memory. A load instruction for X memory can specify either the X0 or X1 register as a destination operand and for a load instruction for Y memory can specify either the Y0 or Y1 register as a destination operand. Both store operations for X and Y memories can specify either the A0 or A1 register as a source operand. This type of operation treats only word data (16 bits). When a word data transfer operation is executed, the upper word of the register operand is used. In case of word data load, the data is loaded into the upper word of the destination register, and then the lower side of the destination is automatically cleared.

When a conditional operation is specified as a data processing operation, the specified condition does not affect any data transfer operations. Figure 3.14 shows this type of data transfer operation flow.

This type of data transfer operation can access X or Y memory only. Any other memory space cannot be accessed.

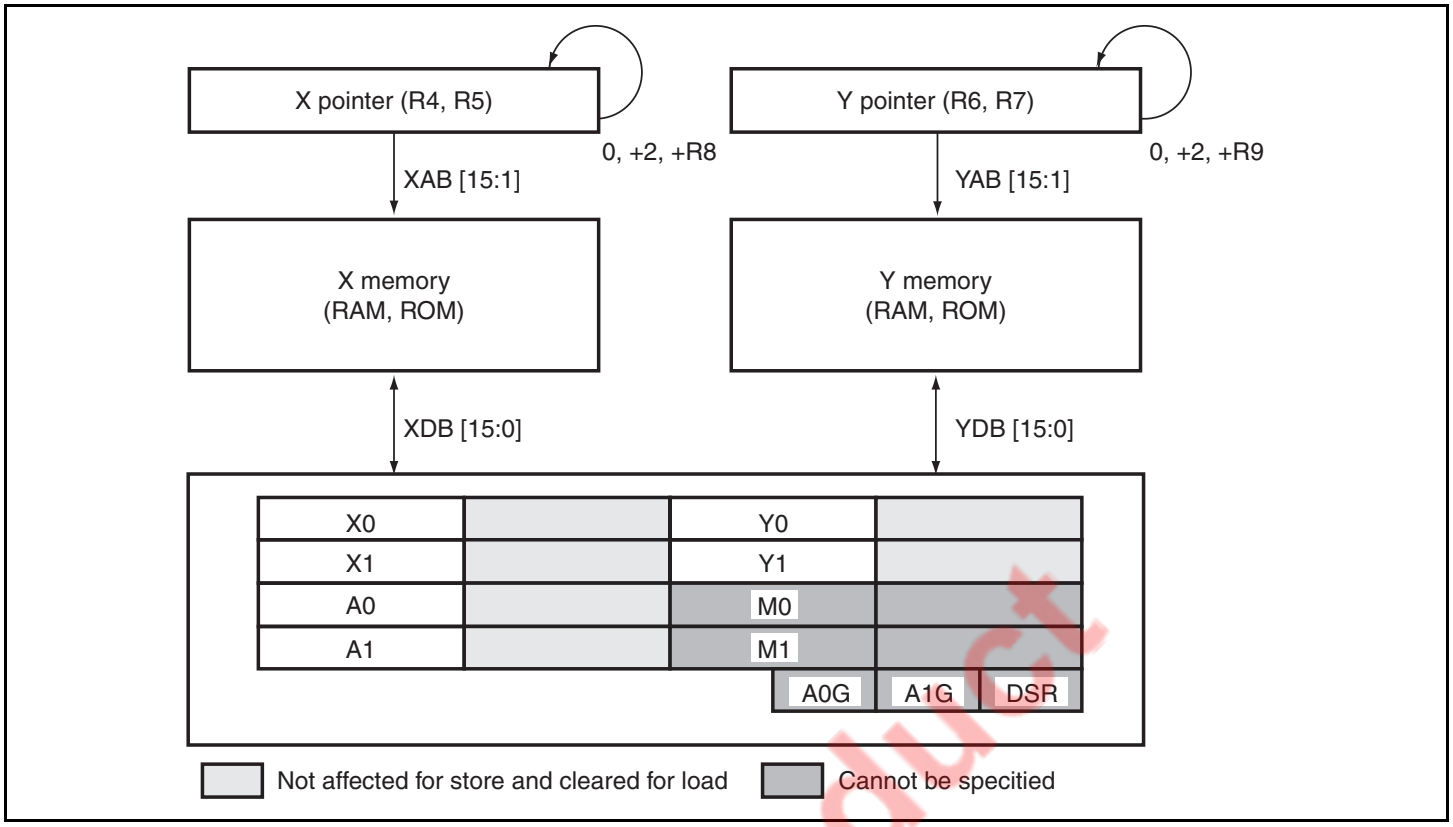


Figure 3.14 Data Transfer Operation Flow

Type 2 instructions execute just two data transfer operations. The 16-bit instruction code is used for this type of instructions. Basically, operation and operand flexibility are the same as in type 1 but conditional operation is not supported. This type of data transfer operation can access X or Y memory only. Any other memory space cannot be accessed.

Type 3 instructions execute single data transfer operations only. The 16-bit instruction code is used for this type of instructions. X pointers and other two extra pointers are available for this type of operation, but Y pointers are not available. This type of operation can access any memory address space, and all registers in the DSP unit, except for DSR, can be specified for both source and destination operands. The guard-bit registers, A0G and A1G, can also be specified as independent registers.

This type of operation can treat both single-word data and longword data. When a word-data transfer operation is executed, the upper word of the register operand is activated. In case of word data load, the data is loaded into the upper word of the destination register, the lower side of the destination register is automatically cleared, and the signed bit is copied into the guard-bit parts, if supported. In case of longword data load, the data is loaded into the upper word and lower word of the destination register and the signed bit is sign-extended and copied into the guard-bit parts, if supported. In case of the guard register store, the signed bit is sign-extended and copied on the upper 24 bits of LDB. Figures 3.15 and 3.16 show this type of data transfer operation flows.

Note: Data transfer by an LDS or STS instruction is possible since DSR is defined as a system register.

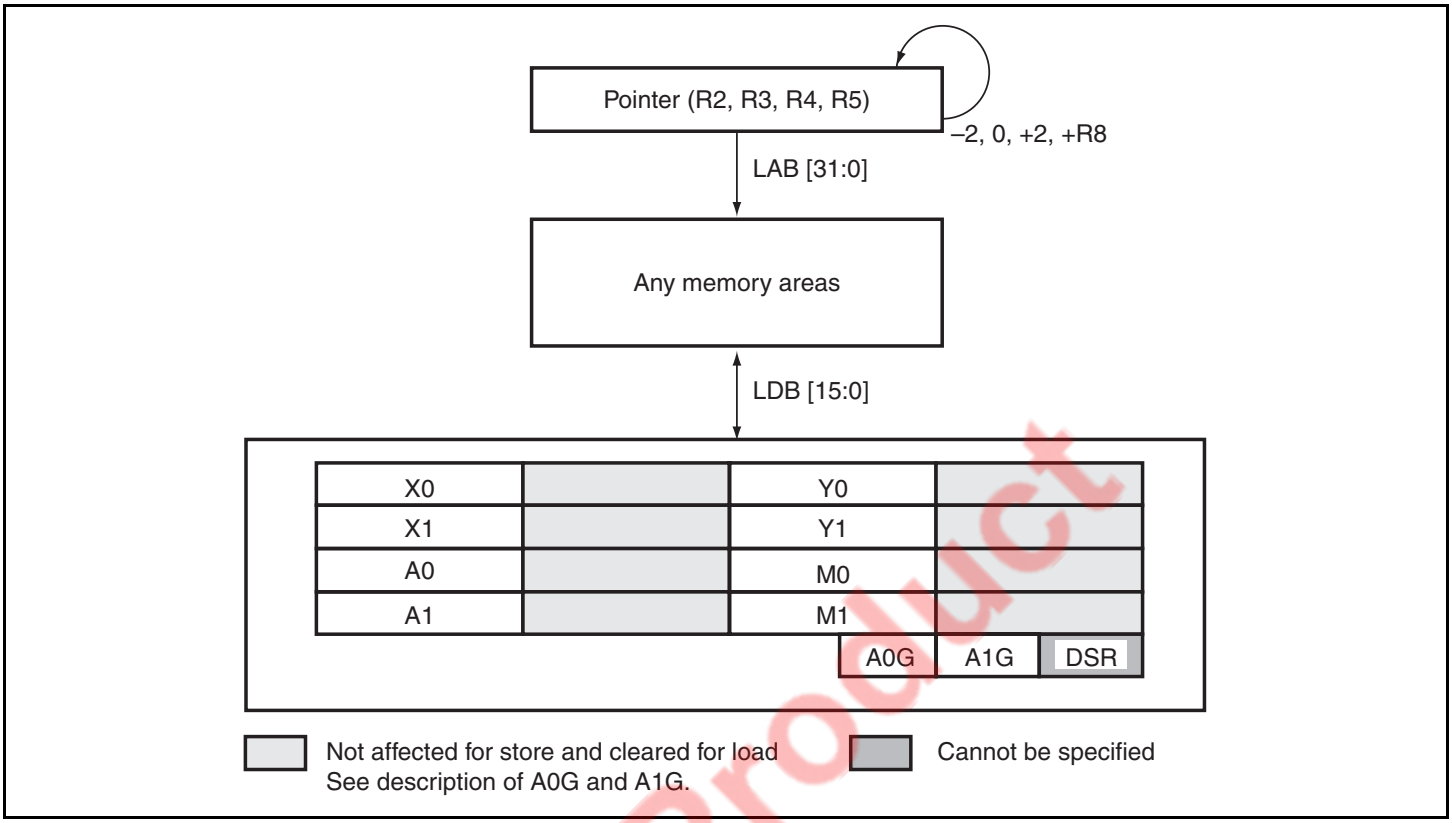


Figure 3.15 Single Data-Transfer Operation Flow (Word)

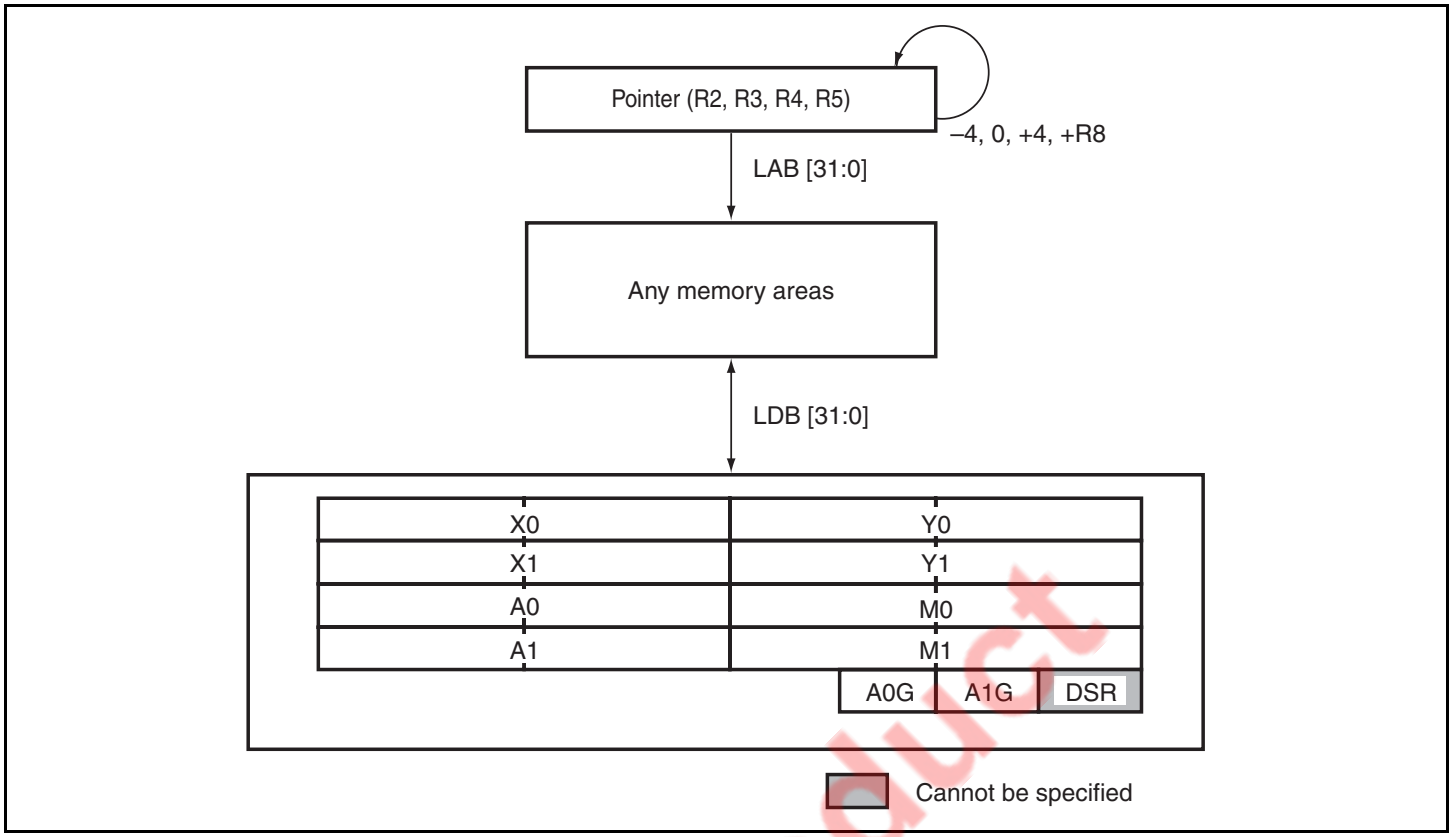


Figure 3.16 Single Data-Transfer Operation Flow (Longword)

All data transfer operations are executed in the MA stage of the pipeline.

All data transfer operations do not update any condition code bits in DSR.

3.1.10 Local Data Move Instruction

The DSP unit of this LSI provides additional two independent registers, MACL and MACH, in order to support SH's standard multiply/MAC operations. They can be also used as temporary storage registers by local data move instructions between MACH/L and other DSP registers Figure 3.17 shows the flow of seven local data move instructions. Table 3.13 shows the variation of this type of instruction.

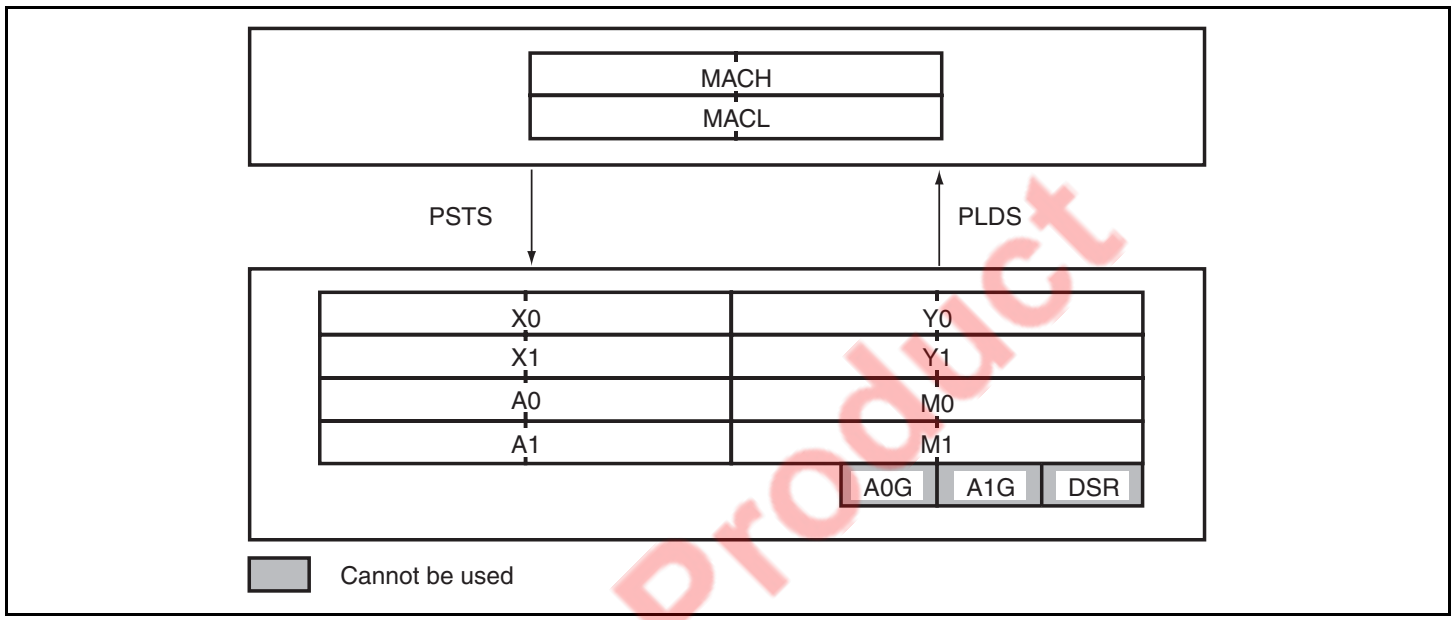


Figure 3.17 Local Data Move Instruction Flow

Table 3.13 Variation of Local Data Move Operations

Mnemonic	Function	Operand
PLDS	Data move from DSP register to MACL/MACH	Dz
PSTS	Data move from MACL/MACH to DSP register	Dz

This instruction is very similar to other transfer instructions. If either the A0 or A1 register is specified as the destination operand of PSTS, the signed bit is sign-extended and copied into the corresponding guard-bit parts, A0G or A1G. The DC bit in DSR and other condition code bits are not updated regardless of the instruction result. This instruction can operate with MOVX and MOVY in parallel.

3.1.11 Operand Conflict

When an identical destination operand is specified with multiple parallel instructions, data conflict occurs. Table 3.14 shows the correspondence between each operand and registers.

Table 3.14 Correspondence between Operands and Registers

		X-Memory Load			Y-Memory Load			6-Instruction ALU			3-Instruction Multiply			3-Instruction ALU		
		Ax	Ix	Dx	Ay	Iy	Dy	Sx	Sy	Du	Se	Sf	Dg	Sx	Sy	Dz
DSP Registers	A0							*1		*2			*2	*1		*1
	A1							*1		*2	*1	*1	*2	*1		*1
	M0								*1				*1		*1	*1
	M1								*1				*1		*1	*1
	X0			*2				*1		*2	*1	*1		*1		*2
	X1			*2				*1			*1			*1		*2
	Y0						*2		*1	*2	*1	*1			*1	*2
	Y1						*2		*1			*1			*1	*2

Notes: 1. Registers available for operands
2. Registers available for operands (when there is operand conflict)

There are three cases of operand conflict problems.

1. When ALU and multiply instructions specify the same destination operand (Du and Dg)
2. When X-memory load and ALU instructions specify the same destination operand (Dx, Du, and Dz)
3. When Y-memory load and ALU instructions specify the same destination operand (Dy, Du, and Dz)

In these cases above, the result is not guaranteed.

3.2 DSP Addressing

3.2.1 DSP Repeat Control

This LSI prepares a special control mechanism for efficient repeat loop control. An instruction SETRC sets the repeat times into the repeat counter RC (12 bits), and an execution mode in which a program loop executes repetitively until RC is equal to 1. After completion of the repeat instructions, the RC value becomes 0.

Repeat start address register RS keeps the start address of a repeat loop. Repeat end register RE keeps the repeat end address. (There are some exceptions. See note, Actual Implementation Options.) Repeat counter RC keeps the number of repeat times. In order to perform loop control, the following steps are required.

Step 1) Set loop start address into RS
Step 2) Set loop end address into RE
Step 3) Set repeat counter into RC
Step 4) Start repeat control

To do steps 1 and 2, use the following instructions:

```
LDRS @(disp,PC)
LDRE @(disp,PC)
```

For steps 3 and 4, use the SETRC instruction. An operand of SETRC is an immediate value or one of the general-purpose registers that will specify the repeat times.

```
SETRC #imm;   #imm->RC, enable repeat control
SETRC Rm;     Rm->RC, enable repeat control
```

#imm is 8 bits while RC is 12 bits. Therefore, to set more than 256 into RC, use Rm. A sample program is shown below.

```
                LDRS    RptStart;
                LDRE    RptEnd3+4;
                SETRC   #imm;       RC = #imm
                instr0;
  ; instr1-5  executes repeatedly
  RptStart:     instr1;
  RptEnd3:      instr2;
                instr3;
                instr4;
  RptEnd:       instr5;
                instr6;
```

In this implementation, there are some restrictions to use this repeat control function as follows:

1. There must be at least one instruction between SETRC and the first instruction in a repeat loop.
2. LDRS and LDRE must be executed before SETRC.
3. In a case that the repeat loop has four or more instructions in it, stall cycles are necessary according to the pipeline state at execution.
4. If a repeat loop has less than four instructions in it, it cannot have any branch instructions (BRA, BSR, BT, BF, BT/S, BF/S, BSRF, RTS, BRAF, RTE, JSR and JMP), repeat control instructions (SETRC, LDRS and LDRE), load instructions for SR, RS, RE, and a TRAPA instruction in it. If these instructions are executed, a general invalid instruction exception handling starts, and a certain address value shown in table 3.15 is stored into SPC.

Table 3.15 Address Value to be Stored into SPC (1)

Condition	Location	Address to be Pushed
RC ≥ 2	Any	RptStart
RC = 1	Any	Address of the illegal instruction

5. If a repeat loop has four or more instructions in it, any branch instructions (BRA, BSR, BT, BF, BT/S, BF/S, BSRF, RTS, BRAF, RTE, JSR and JMP), repeat control instructions (SETRC, LDRS and LDRE), load instructions for SR, RS, RE, and the TRAPA instruction must not be written within the last three instructions from the bottom of a repeat loop. If written, a general invalid instruction exception handling starts, and a certain address value shown in table 3.16 is stored into SPC. In cases of repeat control instructions (SETRC, LDRS and LDRE) and load instructions for SR, RS, and RE, and the TRAPA instruction they cannot be placed in any other location of the repeat loop, either. If they are, the operation is not guaranteed.

Table 3.16 Address Value to be Stored into SPC (2)

Condition	Location	Address to be Pushed
RC ≥ 2	instr3	Address of the illegal instruction
	instr4	RptStart – 4
	instr5	RptStart – 2
RC = 1	Any	Address of the illegal instruction

6. If a repeat loop has less than four instructions in it, any PC relative instructions (MOVA @(disp, PC),R0, etc.) don't work properly except for the instruction at the repeat top (instr1).
7. If a repeat loop has four or more instructions in it, any PC relative instructions (MOVA @(disp, PC),R0, etc.) don't work properly at two instructions from the bottom of the repeat loop.
8. The CPU has no repeat enable flag, however it uses the condition RC = 0 to disable repeat control. Whenever the RC is not 0 and PC matches RE, the repeat control is alive. When 0 is set in the RC, the repeat control is disabled but the repeat loop is executed once and does not return to the repeat start as well as in the RC = 1 case. When RC = 1, the repeat loop is executed once and does not return to the repeat start but the RC becomes 0 after completing the execution of the repeat loop.
9. If a repeat loop has four or more instructions in it, any branch instructions, including subroutine call and return instructions, cannot specify the instruction from inst3 to inst5 in the previous example as the branch target address. If executed, the repeat control doesn't work, so the program goes to the following instruction and the RC is not updated. When a repeat loop has less than four instructions in it, the repeat control doesn't work properly and the RC value in SR is not updated if the branch target is RptStart or a subsequent address.
10. Exception acceptance is restricted during repeat loop processing. See figure 3.18 for details on restrictions.

In figure 3.18, exceptions generated by instructions marked as B and C are handled as follows:

- Interrupt and DMA address errors

  An exception is accepted at neither instruction B or C, and the request is not even saved. A request is detected for the first time and accepted when the next instruction A is executed. Interrupts and DMA address errors are not accepted during a repeat loop with four or less instructions, as shown in 1 to 4 in figure 3.18.
- User break before execution

  An exception is accepted at instruction B, and the address of instruction B is stored into SPC. An exception is not accepted at instruction C, but the request is saved, and is accepted just before the next instruction A or B is to be executed. The address of this next instruction A or B is stored into SPC.
- User break after execution

  An exception is accepted at neither instruction B nor C, but the request is saved, and is accepted just before the next instruction A or B is to be executed. The address of this next instruction A or B is stored into SPC.
- CPU address error

  When a CPU address error occurs by execution of instruction B or C, the exception is accepted, but the value stored into SPC is not the address of the instruction at where the exception occurred. Therefore, return from the exception handler routine cannot be performed correctly. In this case, H'070 is set in EXPEVT as the exception code (also see section 9, Exception Handling). To finish the repeat loop correctly, a CPU error must not be generated at instruction B or C.

Exception Type	Instruction B	Instruction C
Interrupt	Not accepted	Not accepted
DMA address error	Not accepted	Not accepted
UDI break	Not accepted	Not accepted
User break before execution	Not accepted	Not accepted
User break after execution	Not accepted	Not accepted
CPU address error	Accepted as exception code H'070	Accepted as exception code H'070

A: Acceptable for any interrupts
B and C: Acceptable for some interrupts

RC > 1 :

1. 1 repeated step

```
                instr – 1   ; A
                instr0      ; B
Start(End):     instr1      ; C
                instr2      ; A
```

2. 2 repeated steps

```
                instr – 1   ; A
                instr0      ; B
Start:          instr1      ; C
End:            instr2      ; C
                instr3      ; A
```

3. 3 repeated steps

```
                instr – 1   ; A
                instr0      ; B
Start:          instr1      ; C
                instr2      ; C
End:            instr3      ; C
                instr4      ; A
```

4. 4 repeated steps

```
                instr – 1   ; A
                instr0      ; A
Start:          instr1      ; B
                instr2      ; C
                instr3      ; C
End:            instr4      ; C
                instr4      ; A
```

5. 5 or more repeated steps

```
                instr – 1       ; A
                instr0          ; A
Start:          instr1          ; A
                  :
                  :
                instr n – 3     ; B
                instr n – 2     ; C
                instr n – 1     ; C
End:            instr n         ; C
                instr n + 1     ; A
```

RC = 0 :
Acceptable for any interrupts

Figure 3.18 Restriction of Interrupt Acceptance in Repeat Loop

Note 1: Actual Implementation

Repeat start and repeat end registers, RS and RE, specify the repeat start instruction and repeat end instruction. The actual addresses that are kept in these registers depend on the number of instructions in the repeat loop. The rule is as follows:

Repeat_Start: An address of the instruction at the repeat top
Repeat_Start0: An address of the instruction before one instruction at the repeat top
Repeat_End3: An address of the instruction before three instructions at the repeat bottom

Table 3.17 RS and RE Setting Rule

	Number of Instructions in Repeat Loop			
	1	**2**	**3**	**≥4**
RS	Repeat_start0 + 8	Repeat_start0 + 6	Repeat_start0 + 4	Repeat_start
RE	Repeat_start0 + 4	Repeat_start0 + 4	Repeat_start0 + 4	Repeat_End3 + 4

Based on this table, the actual repeat programming for various cases should be described as in the following examples:

CASE 1: 1 Repeated Instruction

```
            LDRS                RptStart0+8;
            LDRE                RptStart0+4;
            SETRC               RptCount;
               - - - -
RptStart0: instr0;
RptStart:  instr1;              Repeated instruction
            instr2;
```

CASE 2: 2 Repeated Instructions

```
            LDRS                RptStart0+6;
            LDRE                RptStart0+4;
            SETRC               RptCount;
               - - - -
RptStart0: instr0;
RptStart:  instr1;              Repeated instruction 1
RptEnd:    instr2;              Repeated instruction 2
            instr3;
```

CASE 3: 3 Repeated Instructions

```
            LDRS                RptStart0+4;
            LDRE                RptStart0+4;
            SETRC               RptCount;
               - - - -
RptStart0: instr0;
RptStart:  instr1;              Repeated instruction 1
            instr2;             Repeated instruction 2
RptEnd:    instr3;              Repeated instruction 3
            instr4;
```

CASE 4: 4 or More Repeated Instructions

```
            LDRS                RptStart;
            LDRE                RptEnd3+4;
            SETRC               RptCount;
               - - - -
RptStart0: instr0;
RptStart:  instr1;              Repeated instruction 1
           instr2;              Repeated instruction 2
           instr3;              Repeated instruction 3
-------------------------------------------------------
RptEnd3:   instrN-3;            Repeated instruction N-3
           instrN-2;            Repeated instruction N-2
           instrN-1;            Repeated instruction N-1
RptEnd:    instrN;              Repeated instruction N
           instrN+1;
```

The examples above can be used as a template to program this repeat loop sequences. However, for easy programming, an extended instruction REPEAT is provided to handle these complex labeling and offset issues. Details will be described in the following note 2.

Note 2: Extended Instruction REPEAT

This REPEAT extended instruction will handle all the delicate labeling and offset processing described in table 3.17 and note 1. The labels used here are:

Rptart: An address of the instruction at the top of the repeat loop

RptEnd: An address of the instruction at the bottom of the repeat loop

RptCount: Repeat count immediate number

This instruction can be used in the following way:

Here the repeat count can be specified as an immediate value #Imm or a register indirect value Rn.

CASE 1: 1 Repeated Instruction

```
REPEAT RptStart, RptStart, RptCount;
              - - - -
           instr0;
RptStart:  instr1;            Repeated instruction
           instr2;
```

CASE 2: 2 Repeated Instructions

```
REPEAT RptStart, RptEnd, RptCount;
              - - - -
           instr0;
RptStart:  instr1;            Repeated instruction 1
RptEnd:    instr2;            Repeated instruction 2
```

CASE 3: 3 Repeated Instructions

```
REPEAT RptStart, RptEnd, RptCount;
              - - - -
           instr0;
RptStart:  instr1;            Repeated instruction 1
           instr2;            Repeated instruction 2
RptEnd:    instr3;            Repeated instruction 3
```

CASE 4: 4 or More Repeated Instructions

```
REPEAT RptStart, RptEnd, RptCount;
              - - - -
           instr0;
RptStart   instr1;            Repeated instruction 1
           instr2;            Repeated instruction 2
           instr3;            Repeated instruction 3
-------------------------------------------------------
           instrN-3;          Repeated instruction N-3
           instrN-2;          Repeated instruction N-2
           instrN-1;          Repeated instruction N-1
RptEnd     instrN;            Repeated instruction N
           instrN+1;
```

The expanded results of each case corresponds to the same case numbers in note 1.

3.2.2 DSP Data Addressing

This LSI has two types of memory access instructions: one type is X and Y data transfer instructions (MOVX.W and MOVY.W), and the other is single data transfer instructions (MOVS.W and MOVS.L). Data addressing of these two types of instruction are different. Table 3.18 shows a summary of DSP data transfer instructions.

Table 3.18 Summary of DSP Data Transfer Instructions

	X and Y Data Transfer Operation (MOVX.W and MOVY.W)	Single Data Transfer Operation (MOVS.W and MOVS.L)
Address registers	Ax: R4 and R5, Ay: R6 and R7	As: R2, R3, R4 and R5
Index register(s)	Ix: R8, Iy: R9	Is: R8
Addressing operations	Not update/Increment (+2)/ Add-index-register Post-update	Not update/Increment (+2)/ Add-index-register Post-update Decrement (–2, –4): Pre-update
Modulo addressing	Yes	No
Data bus	XDB and YDB	LDB
Data length	16 bits (word)	16 bits/32 bits (word/longword)
Bus conflict	No	Possible (same as the SH)
Memory	X and Y data memories	All memory spaces
Source registers	Dx, Dy: A0 and A1	DS: A0/A1, M0/M1, X0/X1, Y0/Y1, A0G, A1G
Destination registers	Dx: X0/X1, Dy: Y0/Y1	Ds: A0/A1, M0/M1, X0/X1, Y0/Y1, A0G, A1G

Addressing for MOVX.W and MOV.W: This LSI can access X and Y data memories simultaneously (MOVX.W and MOVY.W). The DSP instructions have two address pointers that simultaneously access X and Y data memories. The DSP instruction has only pointer-addressing (it does not have immediate-addressing). Address registers are divided into two sets, R4 and R5 (Ax: Address register for X memory) and R6 and R7 (Ay: Address register for Y memory). There are three data addressing types for X and Y data transfer instructions.

1. Not-update address register
2. Add-index register
3. Increment address register

Each address pointer set has an index register, R8[Ix] for set Ax, and R9[Iy] for set Ay. Address instructions for set Ax use ALU in the CPU, and address instructions for set Ay use a different address unit (figure 3.19).

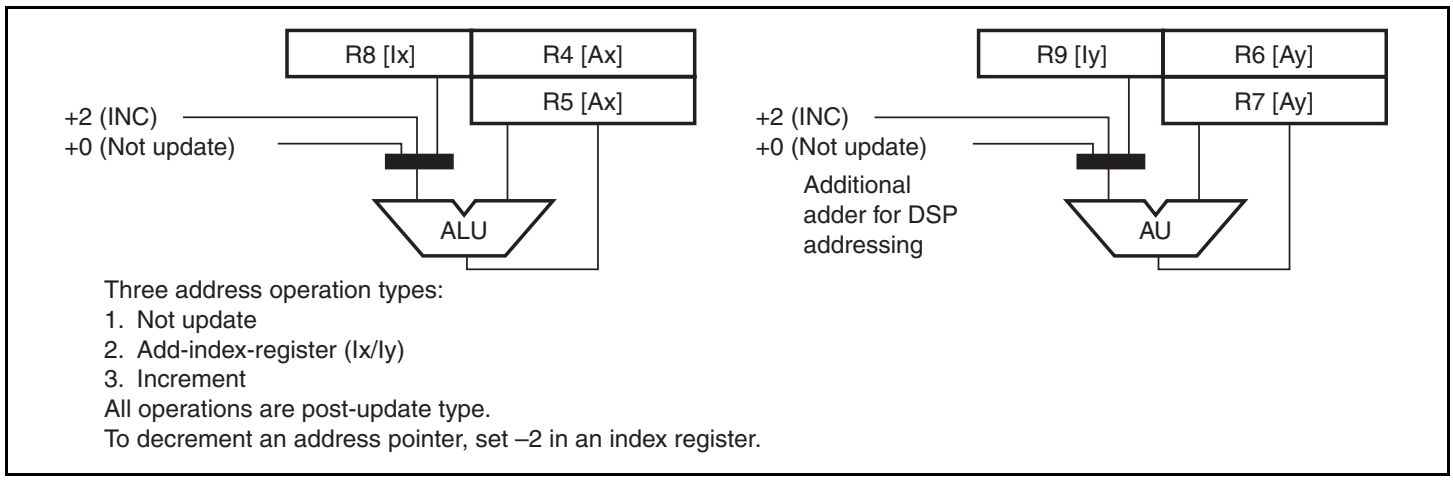


Figure 3.19 DSP Addressing Instructions for MOVX.W and MOVY.W

Addressing in X and Y data transfer operation is always word mode; that is access to X and Y data memories are 16-bit data width. Therefore, the increment operation adds 2 to an address register. To realize decrement, set –2 in an index register and use add-index-register operation.

Addressing for MOVS: This LSI has single-data transfer instructions (MOVS.W and MOVS.L) to load/store DSP data registers. In these instructions, R2 to R5 (As: Address register for single-data transfer) are used for the address pointer.

There are four data addressing types for single-data transfer operation.

1. Not-update address register
2. Add-index register (post-update)
3. Increment address register (post-update)
4. Decrement address register (pre-update)

The address pointer set As has an index register R8[Is] (figure 3.20)

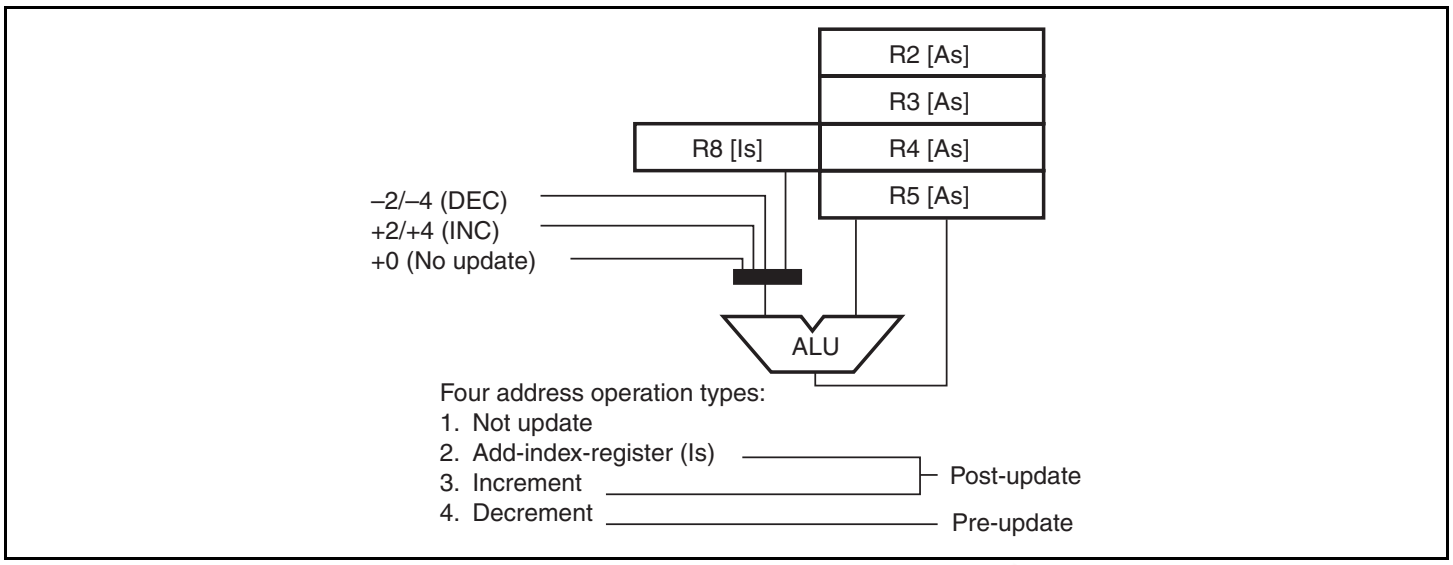


Figure 3.20 DSP Addressing Instructions for MOVS

Modulo Addressing: This LSI provides modulo addressing mode, which is common in DSPs. In modulo addressing mode, the address register is updated as explained above. When the address pointer reaches the pre-defined address (modulo-end address), it goes to the modulo start address.

Modulo addressing is available for X and Y data transfer instructions (MOVX and MOVY), but not for the single-data transfer instruction (MOVS). DMX and DMY in SR are used for the modulo addressing control. If DMX is 1, the modulo addressing mode is effective for the X memory address pointer Ax (R4 or R5). If the DMY is 1, it is effective for the Y memory address pointer Ay (R6 or R7). Modulo addressing is available for one of X and Y address registers at one time. A DMX = DMY = 1 case is reserved for future expansion. When both DMX and DMY are set simultaneously, the hardware will preliminary assume that the modulo addressing mode is effective for the Y address pointer only.

To specify the start and end addresses of the modulo address area, the MOD register, which includes MS (modulo start) and ME (modulo end) is prepared. The following example shows a way to set the MOD (MS and ME) register.

```
            MOV.L ModAddr,Rn;     Rn=ModEnd, ModStart
            LDC Rn,MOD;           ME=ModEnd, MS=ModStart
ModAddr:    .DATA.W     mEnd;     Lower 16 bits of ModEnd
            .DATA.W     mStart;   Lower 16 bits of ModStart

ModStart:   .DATA
              :
ModEnd:     .DATA
```

MS and ME are set to specify the start and end addresses, and then later to set the DMX or DMY bit to 1.

When the X/Y data transfer instruction set in DMX/DMY is executed, the address register contents before update are compared with ME*[1]. If they match, modulo start address MS is stored in the address register as the updated value*[2]. If non-update address register addressing is specified for the X/Y data transfer instruction, the address pointer will not return to modulo start address MS even though the address register contents match ME.

Notes: 1. Bits 1 to 15 of the address register are used for comparison. Though ME retains its previous value for bit 0, 0 must always be written to bit 0.
2. The MS value is stored in bits 1 to 15 of the address register. Though MS retains its previous value for bit 0, 0 must always be written to bit 0.

The maximum modulo size is 64-kbytes. This is sufficient for accessing the X or Y data memory.

Figure 3.21 shows a block diagram of modulo addressing.

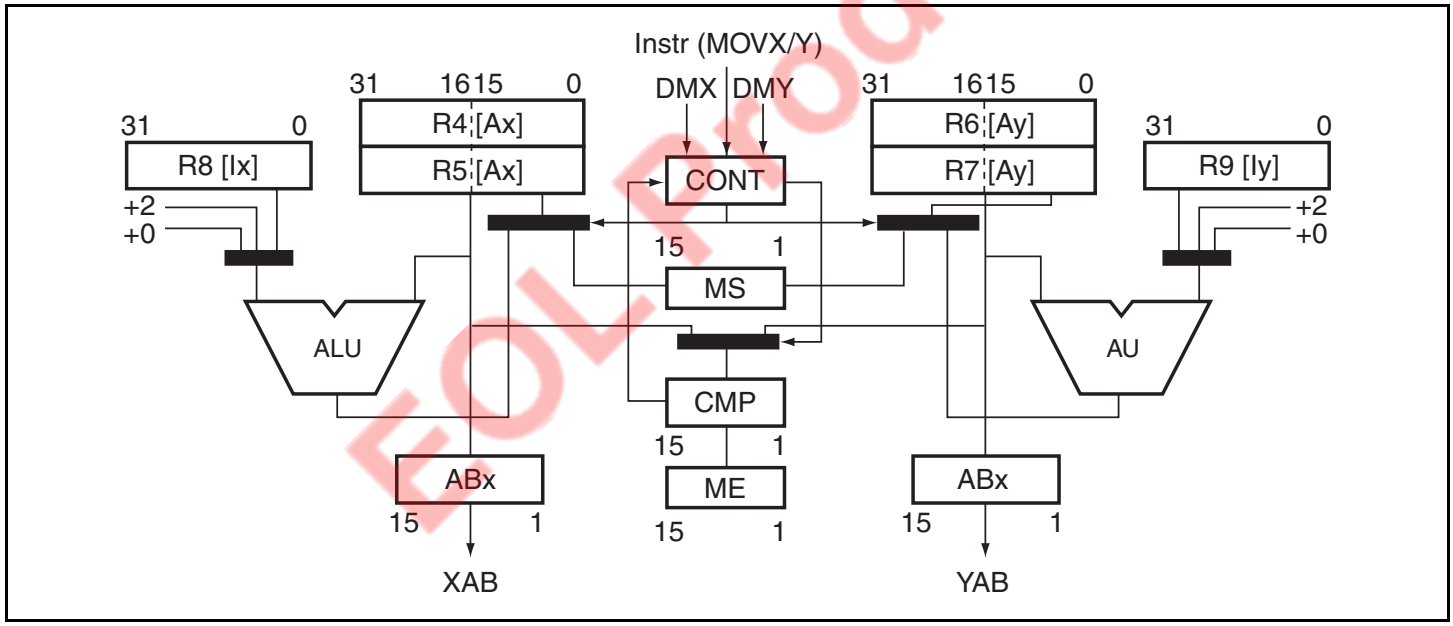


Figure 3.21 Modulo Addressing

An example is shown below.

```
MS=H'7000; ME=H'7004; R4=H'A5007000;
DMX=1; DMY=0 (modulo addressing for address register Ax)
```

As a result of the above settings, the R4 register changes as follows.

```
                    ; R4: H'A5007000 (Initial value)
MOVX.W @R4+,Dx      ; R4: H'A5007000 → H'A5007002
MOVX.W @R4+,Dx      ; R4: H'A5007002 → H'A5007004
MOVX.W @R4+,Dx      ; R4: H'A5007004 → H'A5007000 (After reading
                      H'A5007004, MS value is written to address
                      register)
MOVX.W @R4+,Dx      ; R4: H'A5007000 → H'A5007002
```

Place the data so that the upper 16 bits of the modulo start and end addresses are the same. This is because the modulo start address overwrites only the lower 16 bits of the address register.

Note: When addition index is the data addressing type for X and Y data transfer instructions, the address pointer may exceed the ME value without actually reaching it. In this case, the address pointer will not return to the modulo start address. Not only with modulo addressing, but when X and Y data addressing is used, bit 0 is ignored. 0 must always be written to bit 0 of the address pointer, index register, MS, and ME.

Addressing Instructions in Execution Stage: Address instructions, including modulo addressing, are executed in the execution stage of the pipeline. Behavior of the DSP data addressing in the execution stage is shown below.

```
if ( Operation is MOVX.W MOVY.W ) {
ABx=Ax; ABy=Ay;
/* Memory access cycle uses ABx and ABy. The addresses to be used have
not been updated. */

/* Ax is one of R4,5 */
if { DMX==0 || ( DMX==1 && DMY==1 )} Ax = Ax+(+2 or R8[Ix] or +0);
/* Inc,Index,Not-Update */
else if (! not-update) Ax=modulo( Ax, (+2 or R8[Ix]) );

/* Ay is one of R6,7 */
if ( DMY==0 ) Ay=Ay+(+2 or R9[Iy] or +0); /* Inc,Index,Not-Update */
else if (! not-update) Ay=modulo( Ay, (+2 or R9[Iy]) );
}
else if ( Operation is MOVS.W or MOVS.L ) {
   if ( Addressing is Nop, Inc, Add-index-reg ) {
MAB=As;
/* Memory access cycle uses MAB. The address to be used has not been
updated.*/

/* As is one of R2 to R5 */
As = As+(+2 or +4 or R8[Is] or +0); /* Inc,Index,Not-Update */
   else { /* Decrement, Pre-update */
/* As is one of R2 to R5 */
As=As+(-2 or -4);
MAB=As;
/* Memory access cycle uses MAB. The address to be used has been
updated. */
}
```

```
/* The value to be added to the address register depends on addressing
instructions.

For example, (+2 or R8[Ix] or +0) means that
+2:     if instruction is increment
R8[Ix]: if instruction is add-index-register
+0:     if instruction is not-update
*/

function modulo ( AddrReg, Index ) {
if ( AddrReg[15:1]==ME[15:1] ) AddrReg[15:1]==MS[15:1];
else AddrReg=AddrReg+Index;
return AddrReg;
}
```

X and Y Data Transfer Instructions (MOVX.W and MOVY.W): This type of instruction uses the XDB and the YDB to access X and Y data memories (they cannot access other memory spaces). These two buses are separate from the instruction bus, therefore, there is no access conflict between data memory access and instruction memory access.

Figure 3.22 shows load/store control for X and Y data transfer instructions. All memory accesses are word mode accesses.

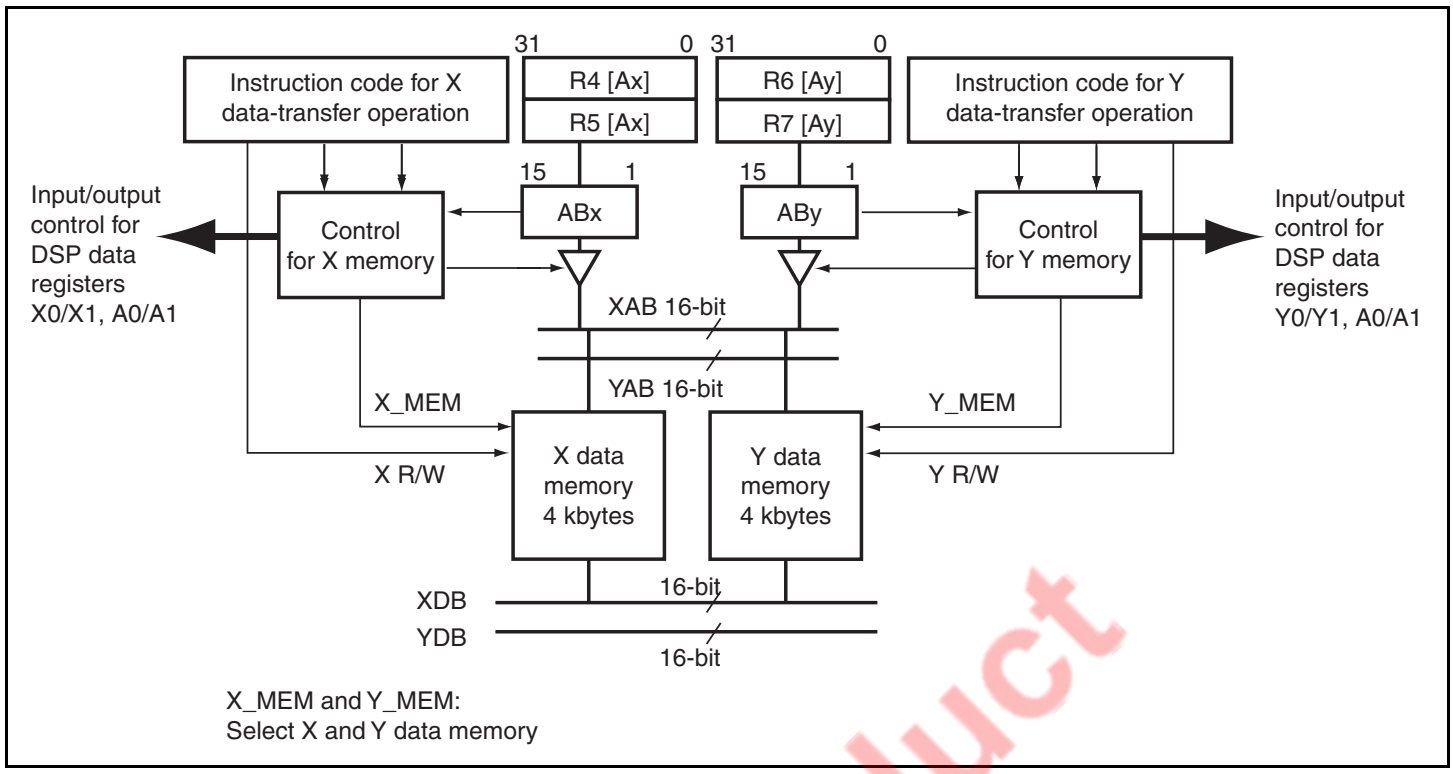


Figure 3.22 Load/Store Control for X and Y Data-Transfer Instructions

Control for X Memory:

```
if ( !Nop ) {
      X_MEM=1; XAB=ABx;
      if ( load operation ) {
            Dx[31:16]=XDB;
            Dx[15:0]=0x0000;     /* Dx is X0 or X1 */
      }
      else XDB = Dx[31:16];      /* Dx is A0 or A1 */
}
else { X_MEM=0; XAB=0x000; }
```

The conditional execution based on the DC flag in DSR cannot control any MOVX/MOVY instructions.

Single-Data Transfer Instructions (MOVS.W and MOVS.L): This LSI has single load/store instructions for the DSP registers. It is similar to a load/store instruction for a system register. It transfers data between memory and DSP data registers using LAB and LDB buses. There may be access conflict between data access and instruction fetch.

The single-data transfer instruction has word and longword access modes. Figure 3.23 shows a block diagram of single-data transfer. The existing CPU core's hardware resource is used for control of the memory address buffer (MAB) and memory selection.

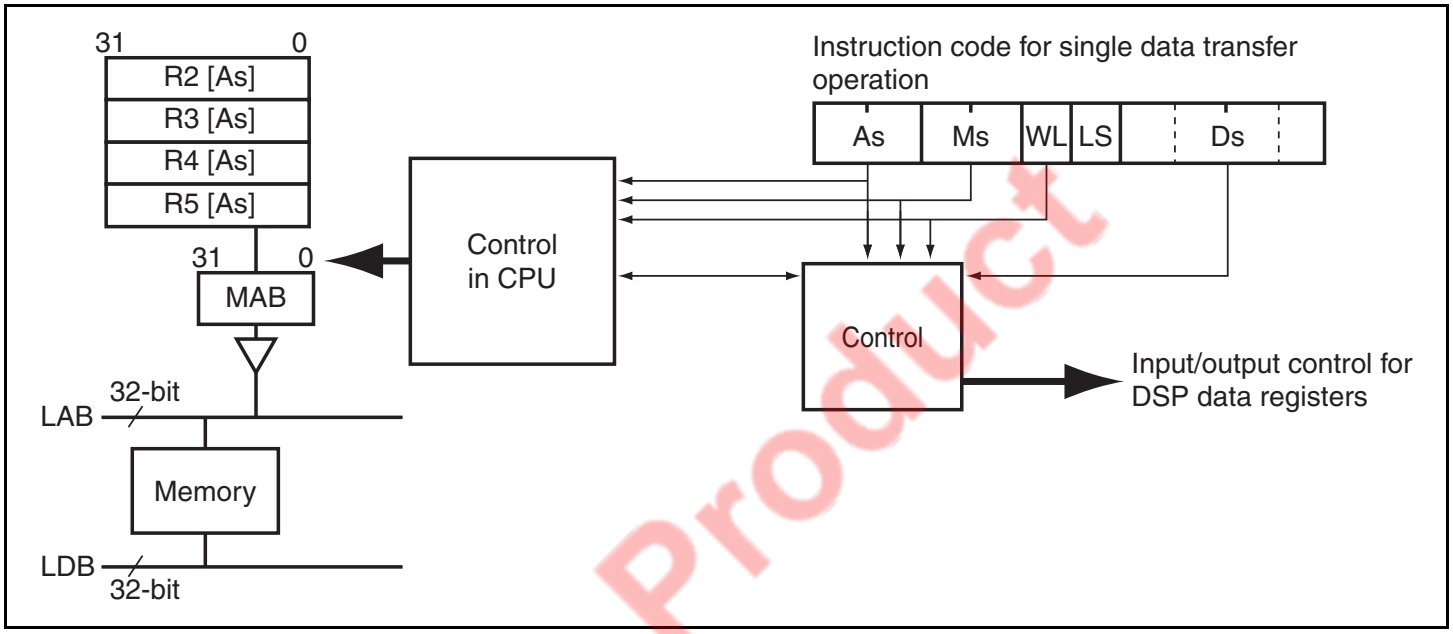


Figure 3.23 Load/Store Control for Single-Data Transfer Instruction

Control

```
LAB=MAB;
if ( Ms!=NLS && W/L is word access ) { /* MOVS.W */
            if (LS==load) {
                        if (Ds!=A0G && Ds!=A1G) {
                              Ds[31:16] = LDB[15:0]; Ds[15:0] = 0x0000;
                              if (Ds==A0) A0G[7:0] = sign-extension of LDB;
                              if (Ds==A1) A1G[7:0] = sign-extension of LDB;
                        }
                        else Ds[7:0] = LDB[7:0];      /* Ds is A0G or A1G */
            }
            else { /* Store */
                        if (Ds!=A0G && Ds!=A1G) LDB[15:0] = Ds[31:16];
                        /* Ds is A0G or A1G */
                        else LDB[15:0] = Ds[7:0] with 8bit sign-extension;
            }
}
else if ( MA!=NLS && W/L is long-word access ) { /* MOVS.L */
            if (LS==load) {
                        if (Ds!=A0G && Ds!=A1G) {
                                    Ds[31:0] = LDB[31:0];
                                    if (Ds==A0) A0G[7:0] = sign-extension of LDB;
                                    if (Ds==A1) A1G[7:0] = sign-extension of LDB;
                        }
                        else Ds[7:0] = LDB[7:0]; /* Ds is A0G or A1G */
            }
            else { /* Store */
                        if (Ds!=A0G && Ds!=A1G) LDB[31:0] = Ds[31:0];
                        /* Ds is A0G or A1G */
                        else LDB[31:0] = Ds[7:0] with 24bit sign-extension;
            }
}
```

Section 4 Clock Pulse Generator (CPG)

This LSI has a clock pulse generator (CPG) that generates an internal clock (Iϕ), a peripheral clock (Pϕ), and a bus clock (Bϕ). The CPG consists of an oscillator, PLL circuit, and divider circuit.

4.1 Features

The CPG has the following features.

- Three clock modes

  The mode is selected from among the three clock modes by the selection of the following three conditions: the frequency-divisor in use, whether the PLL is on or off, and whether the internal crystal resonator or the input on the external clock-signal line is used.
- Three clocks generated independently

  An internal clock (Iϕ) for the CPU and cache; a peripheral clock (Pϕ) for the on-chip peripheral modules; a bus clock (Bϕ = CKIO) for the external bus interface.
- Frequency change function

  Internal and peripheral clock frequencies can be changed independently using the PLL (phase locked loop) circuit and divider circuit within the CPG. Frequencies are changed by software using frequency control register (FRQCR) settings.
- Power-down mode control

  The clock can be stopped for sleep mode, and standby mode and specific modules can be stopped using the module standby function. For details on clock control in the low-power consumption modes, see section 6, Power-Down Modes.

A block diagram of the CPG is given in figure 4.1.

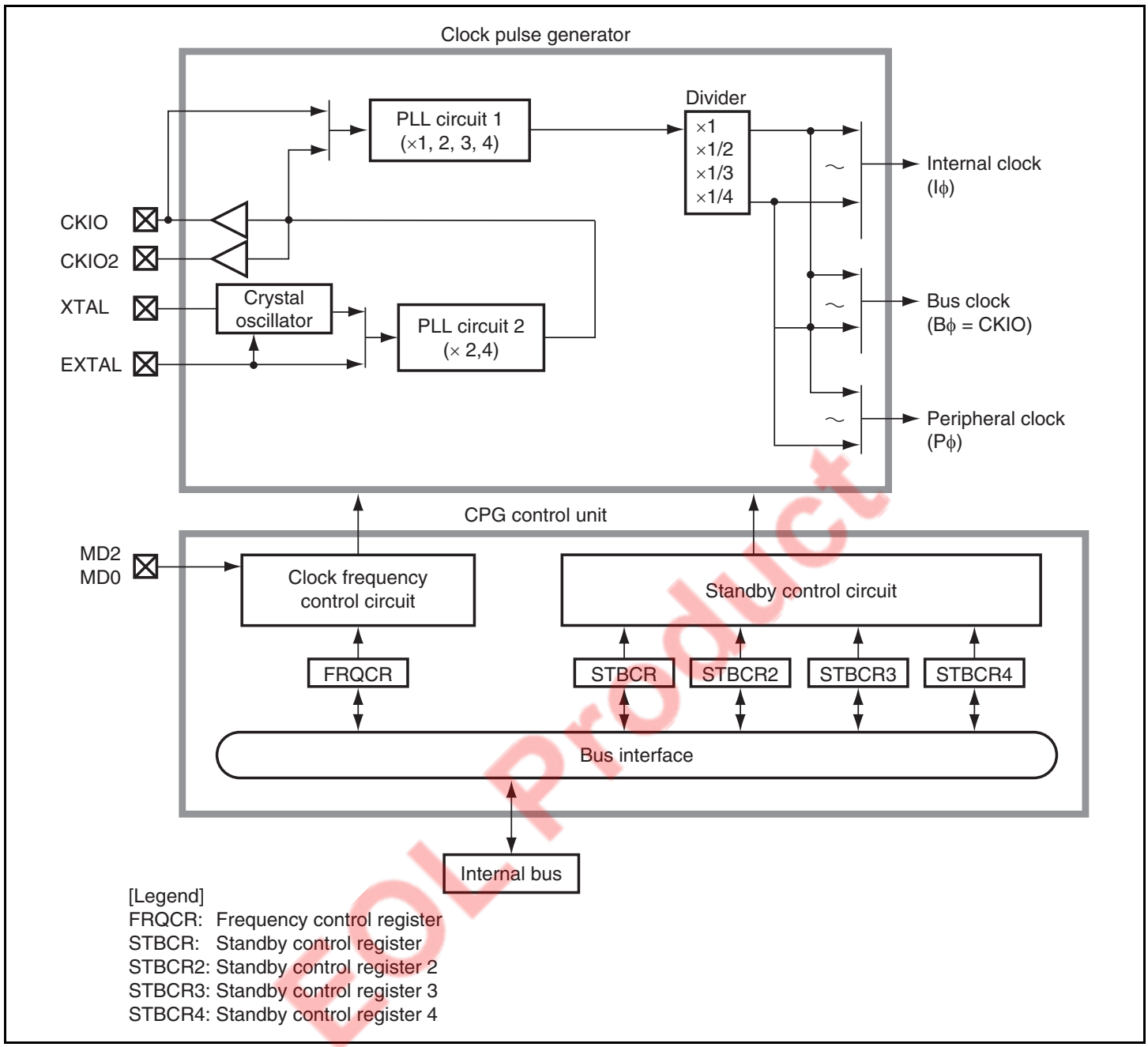


Figure 4.1 Block Diagram of Clock Pulse Generator

The clock pulse generator blocks function as follows:

PLL Circuit 1: PLL circuit 1 doubles, triples, or quadruples, the input clock frequency from the CKIO pin. The multiplication rate is set by the frequency control register. When this is done, the phase of the rising edge of the internal clock is controlled so that it will agree with the phase of the rising edge of the CKIO pin.

PLL Circuit 2: PLL circuit 2 doubles, or quadruples the input clock frequency from the crystal oscillator or EXTAL pin. The multiplication rate is fixed according to the clock operating mode. The clock operating mode is specified by the MD0, and MD2 pins. For details on clock operating mode, see table 4.2.

Crystal Oscillator: The crystal oscillator is an oscillator circuit in which a crystal resonator is connected to the XTAL pin or EXTAL pin. This can be used according to the clock operating mode.

Divider: The divider generates a clock signal at the operating frequency used by the internal or peripheral clock. The operating frequency can be 1, 1/2, 1/3 or 1/4 times the output frequency of PLL circuit 1, as long as it stays at or above the clock frequency of the CKIO pin. The division ratio is set in the frequency control register.

Clock Frequency Control Circuit: The clock frequency control circuit controls the clock frequency using the MD0, and MD2 pins and the frequency control register.

Standby Control Circuit: The standby control circuit controls the states of the clock pulse generator and other modules during clock switching or sleep, or standby modes.

Frequency Control Register: The frequency control register has control bits assigned for the following functions: clock output/non-output from the CKIO pin during standby modes, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the internal clock and the peripheral clock.

Standby Control Register: The standby control register has bits for controlling the power-down modes. See section 6, Power-Down Modes, for more information.

4.2 Input/Output Pins

Table 4.1 lists the CPG pins and their functions.

Table 4.1 Pin Configuration and Functions of the Clock Pulse Generator

Pin Name	Symbol	I/O	Function (clock operating modes 2 and 6)	Function (clock operating mode 7)
Mode control pins	MD0	Input	Set the clock operating mode.	
	MD2	Input	Set the clock operating mode.	
Crystal input/output pins (Clock input pins)	XTAL	Output	Connected to the crystal resonator (leave this pin open-circuit when the crystal resonator is not in use).	
	EXTAL	Input	Connected to the crystal resonator or used to input an external clock.	
Clock input/output pin	CKIO	I/O	Clock output pin. The pin can also be placed in the high-impedance state.	Input for the external clock pulse.
Clock-output pin	CKIO2	Output	Low-level output or clock output pin. The selection is described in the description of the common control registers in section 12, Bus State Controller (BSC).	High impedance

4.3 Clock Operating Modes

Table 4.2 shows the relationship between the mode control pins (MD2 and MD0) combinations and the clock operating modes. Table 4.3 shows the usable frequency ranges in the clock operating modes.

Table 4.2 Clock Operating Modes

	Pin Values		Clock I/O				
Mode	MD2	MD0	Source	Output	PLL2 On/Off	PLL1 On/Off	CKIO Frequency
2	0	0	EXTAL or Crystal resonator	CKIO	ON (×4)	ON (×1, 2)	(EXTAL or Crystal resonator) ×4
6	1	0	EXTAL or Crystal resonator	CKIO	ON (×2)	ON (×1, 2, 3, 4)	(EXTAL or Crystal resonator) ×2
7	1	1	CKIO	—	OFF	ON (×1, 2, 3, 4)	(CKIO)

Mode 2: The frequency of the signal received from the EXTAL pin or crystal resonator LSI is quadrupled by the PLL circuit 2 before it is supplied as the clock signal. This lowers the frequency required of the externally generated clock. Either a crystal resonator with a frequency in the range from 10 to 12.5 MHz or an external signal in the same frequency range input on the EXTAL pin may be used. The frequency range of CKIO is from 40 to 50 MHz.

Mode 6: The frequency of the signal received from the EXTAL pin or crystal resonator LSI is doubled by the PLL circuit 2 before it is supplied as the clock signal. This lowers the frequency required of the crystal resonator. A crystal resonator or an external signal with a frequency in the range from 10 to 25 MHz may be used. The frequency range of CKIO is from 20 to 50 MHz.

Mode 7: In this mode, the CKIO pin functions as an input pin. An external clock signal is supplied to this pin; after this signal is received, the PLL circuit 1 shapes its waveform and multiplies its frequency. The resulting clock signal is then supplied within the LSI. For reduced current and hence power consumption, pull up the EXTAL pin and open the XTAL pin when the LSI is used in mode 7.

Table 4.3 Relationship between Clock Mode and Frequency Range

Clock operating mode	FRQCR register setting	PLL frequency multiplier		Ratio of internal clock frequencies	Selectable frequency ranges (MHz)				
		PLL Circuit 1	PLL Circuit 2	(I:B:P)	Input clock	Output clock (CKIO pin)	Internal clock	Bus clock	Peripheral clock
2	H'1001	ON (×1)	ON (×4)	4:4:2	10 to 12.5	40 to 50	40 to 50	40 to 50	20 to 25
	H'1002	ON (×1)	ON (×4)	4:4:4/3	10 to 12.5	40 to 50	40 to 50	40 to 50	13.33 to 16.66
	H'1003	ON (×1)	ON (×4)	4:4:1	10 to 12.5	40 to 50	40 to 50	40 to 50	10 to 12.5
	H'1103	ON (×2)	ON (×4)	8:4:2	10 to 12.5	40 to 50	80 to 100	40 to 50	20 to 25
	H'1113	ON (×2)	ON (×4)	4:4:2	10 to 12.5	40 to 50	40 to 50	40 to 50	20 to 25
6	H'1000	ON (×1)	ON (×2)	2:2:2	10 to 16.66	20 to 33.33	20 to 33.33	20 to 33.33	20 to 33.33
	H'1001	ON (×1)	ON (×2)	2:2:1	10 to 25	20 to 50	20 to 50	20 to 50	10 to 25
	H'1002	ON (×1)	ON (×2)	2:2:2/3	10 to 25	20 to 50	20 to 50	20 to 50	6.66 to 16.66
	H'1003	ON (×1)	ON (×2)	2:2:1/2	10 to 25	20 to 50	20 to 50	20 to 50	5 to 12.5
	H'1101	ON (×2)	ON (×2)	4:2:2	10 to 16.66	20 to 33.33	40 to 66.66	20 to 33.33	20 to 33.33
	H'1103	ON (×2)	ON (×2)	4:2:1	10 to 25	20 to 50	40 to 100	20 to 50	10 to 25
	H'1111	ON (×2)	ON (×2)	2:2:2	10 to 16.66	20 to 33.33	20 to 33.33	20 to 33.33	20 to 33.33
	H'1113	ON (×2)	ON (×2)	2:2:1	10 to 25	20 to 50	20 to 50	20 to 50	10 to 25
	H'1202	ON (×3)	ON (×2)	6:2:2	13.33 to 16.66	26.66 to 33.33	80 to 100	26.66 to 33.33	26.66 to 33.33
	H'1222	ON (×3)	ON (×2)	2:2:2	13.33 to 16.66	26.66 to 33.33	26.66 to 33.33	26.66 to 33.33	26.66 to 33.33

Clock operating mode	FRQCR register setting	PLL frequency multiplier: PLL Circuit 1	PLL frequency multiplier: PLL Circuit 2	Ratio of internal clock frequencies (I:B:P)	Selectable frequency ranges (MHz): Input clock	Output clock (CKIO pin)	Internal clock	Bus clock	Peripheral clock
6	H'1303	ON (×4)	ON (×2)	8:2:2	10 to 12.5	20 to 25	80 to 100	20 to 25	20 to 25
	H'1313	ON (×4)	ON (×2)	4:2:2	10 to 16.66	20 to 33.33	40 to 66.66	20 to 33.33	20 to 33.33
	H'1333	ON (×4)	ON (×2)	2:2:2	10 to 16.66	20 to 33.33	20 to 33.33	20 to 33.33	20 to 33.33
7	H'1000	ON (×1)	OFF	1:1:1	20 to 33.33	20 to 33.33	20 to 33.33	20 to 33.33	20 to 33.33
	H'1001	ON (×1)	OFF	1:1:1/2	20 to 50	20 to 50	20 to 50	20 to 50	10 to 25
	H'1002	ON (×1)	OFF	1:1:1/3	20 to 50	20 to 50	20 to 50	20 to 50	6.66 to 16.66
	H'1003	ON (×1)	OFF	1:1:1/4	20 to 50	20 to 50	20 to 50	20 to 50	5 to 12.5
	H'1101	ON (×2)	OFF	2:1:1	20 to 33.33	20 to 33.33	40 to 66.66	20 to 33.33	20 to 33.33
	H'1103	ON (×2)	OFF	2:1:1/2	20 to 50	20 to 50	40 to 100	20 to 50	10 to 25
	H'1111	ON (×2)	OFF	1:1:1	20 to 33.33	20 to 33.33	20 to 33.33	20 to 33.33	20 to 33.33
	H'1113	ON (×2)	OFF	1:1:1/2	20 to 50	20 to 50	20 to 50	20 to 50	10 to 25
	H'1202	ON (×3)	OFF	3:1:1	26.66 to 33.33	26.66 to 33.33	80 to 100	26.66 to 33.33	26.66 to 33.33
	H'1222	ON (×3)	OFF	1:1:1	26.66 to 33.33	26.66 to 33.33	26.66 to 33.33	26.66 to 33.33	26.66 to 33.33
	H'1303	ON (×4)	OFF	4:1:1	20 to 25	20 to 25	80 to 100	20 to 25	20 to 25
	H'1313	ON (×4)	OFF	2:1:1	20 to 33.33	20 to 33.33	40 to 66.66	20 to 33.33	20 to 33.33
	H'1333	ON (×4)	OFF	1:1:1	20 to 33.33	20 to 33.33	20 to 33.33	20 to 33.33	20 to 33.33

Notes:
1. The ratio of clock frequencies, where the input clock frequency is assumed to be 1.
2. In modes 2 and 6, the frequency of the clock input from the EXTAL pin or the frequency of the crystal resonator. In mode 7, the frequency of the clock input from the CKIO pin.

Caution:
1. The frequency of the internal clock is the frequency of the signal input to the CKIO pin after multiplication by the frequency-multiplier of PLL circuit 1 and division by the divider's divisor. Do not set a frequency for the internal clock below the frequency of the signal on the CKIO pin.
2. The frequency of the peripheral clock is the frequency of the signal input to the CKIO pin after multiplication by the frequency-multiplier of PLL circuit 1 and division by the divider's divisor. Set the frequency of the peripheral clock to 33.33 MHz or below. In addition, do not set a higher frequency for the internal clock than the frequency on the CKIO pin.
3. The frequency multiplier of the PLL circuit can be selected as x1, x2, x3 or x4. The divisor of the divider can be selected as x1, x1/2, x1/3 or x1/4. The settings are made in the respective frequency-control registers.
4. The signal output by PLL circuit 1 is the signal on the CKIO pin multiplied by the frequency multiplier of PLL circuit 1. Ensure that the frequency of the signal from PLL circuit 1 is no more than 100 MHz.

4.4 Register Descriptions

The CPG's control register is called the frequency control register (FRQCR). Refer the section 24, List of Registers, for the addresses of the registers and the state of each register in each processor state.

4.4.1 Frequency Control Register (FRQCR)

The frequency control register (FRQCR) is a 16-bit readable/writable register used to specify whether a clock is output from the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the internal clock and the peripheral clock.

Only word access can be used on the FRQCR register.

This register is initialized (to H'1003) only in the case of a power-on reset. This register retains its previous value after a manual reset or period in standby mode. The previous value is also retained when an internal reset is triggered by an overflow of the WDT.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	CKOEN	1	R/W	Clock Output Enable CKOEN specifies whether a clock is output from the CKIO and CKIO2 pins, or whether the CKIO and CKIO2 pins is placed in the level-fixed state during release of the standby mode (until the state enters STATUS1 = L and STATUS0 = L from an interrupt). If CKOEN is cleared to 0, the CKIO and CKIO2 pins are fixed at low during STATUS1 = L and STATUS0 = H. Therefore, the malfunction of an external circuit because of an unstable CKIO clock during release of the standby mode can be prevented. In clock operating mode 7, the CKIO pin functions as an input regardless of this bit value. 0: The CKIO pin is fixed to the low level in the standby mode and while the system is leaving standby mode. 1: Clock is output from CKIO pin (placed in the high-impedance state during periods in standby mode).

Bit	Bit Name	Initial Value	R/W	Description
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	STC1	0	R/W	Frequency multiplication ratio of PLL circuit 1
8	STC0	0	R/W	00: × 1 time 01: × 2 times 10: × 3 times 11: × 4 times
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	IFC1	0	R/W	Internal Clock Frequency Division Ratio
4	IFC0	0	R/W	These bits specify the frequency division ratio of the internal clock with respect to the output frequency of PLL circuit 1. 00: × 1 time 01: × 1/2 time 10: × 1/3 time 11: × 1/4 time
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PFC1	1	R/W	Peripheral Clock Frequency Division Ratio
0	PFC0	1	R/W	These bits specify the division ratio of the peripheral clock frequency with respect to the output frequency of PLL circuit 1. 00: × 1 time 01: × 1/2 time 10: × 1/3 time 11: × 1/4 time

4.5 Changing the Frequency

The frequency of the internal clock and peripheral clock can be changed either by changing the multiplication rate of PLL circuit 1 or by changing the division rates of divider. All of these are controlled by software through the frequency control register. The methods are described below.

4.5.1 Changing the Multiplication Rate

A PLL settling time is required when the multiplication rate of PLL circuit 1 is changed. The on-chip WDT counts the settling time.

1. In the initial state, the multiplication rate of PLL circuit 1 is 1.
2. Set a value that will become the specified oscillation settling time in the WDT and stop the WDT. The following must be set:

   WTCSR register TME bit = 0: WDT stops

   WTCSR register CKS2 to CKS0 bits: Division ratio of WDT count clock

   WTCNT counter: Initial counter value
3. Set the desired value in the STC1 and STC0 bits. The division ratio can also be set in the IFC[1:0] and PFC[1:0] bits.
4. The processor pauses temporarily and the WDT starts incrementing. The internal and peripheral clocks both stop and the WDT is supplied with the clock. The clock will continue to be output at the CKIO pin. This state is the same as the standby state. Whether or not registers are initialized depends on the module. For details, see table 6.3, Register States in Standby Mode in section 6, Power-Down Modes.
5. Supply of the clock that has been set begins at WDT count overflow, and the processor begins operating again. The WDT stops after it overflows.

4.5.2 Changing the Division Ratio

Counting by the WDT does not proceed if the frequency divisor is changed but the multiplier is not.

1. In the initial state, IFC[1:0] = B'00 and PFC[1:0] = B'11
2. Set the desired value in the IFC[1:0] and PFC[1:0] bits. The values that can be set are limited by the clock mode and the multiplication rate of PLL circuit 1. Note that if the wrong value is set, the processor will malfunction.
3. The clock is immediately supplied at the new division ratio.

4.6 Notes on Board Design

Note on Using an External Crystal Resonator: Place the crystal resonator, capacitors CL1 and CL2, and feedback resistor R1 as close to the XTAL and EXTAL pins as possible. In addition, to minimize induction and thus obtain oscillation at the correct frequency, the capacitors to be attached to the resonator must be grounded to the same ground. Do not bring wiring patterns close to these components.

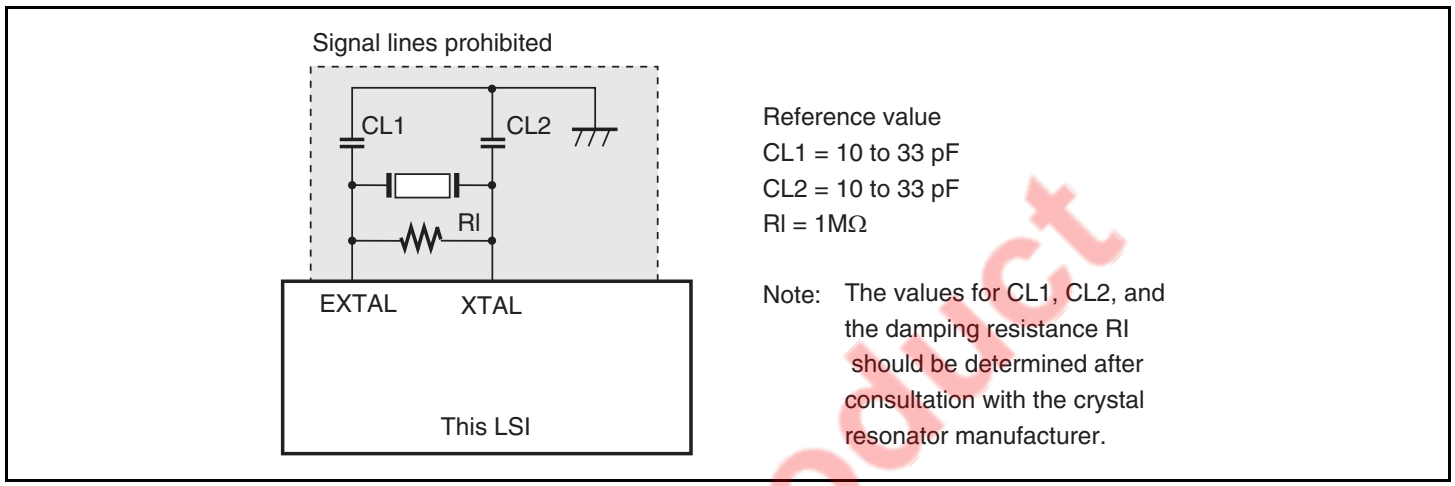


Figure 4.2 Note on Using a Crystal Resonator

Notes on Using External Clocks: When external clocks are input from the EXTAL pin, leave the XTAL pin open. In order to prevent a malfunction due to the reflection noise caused in a signal line which connected to XTAL pin, cut this signal line as short as possible.

Notes on Bypass Capacitor: A multilayer ceramic capacitor must be inserted for each pair of Vss and Vcc as a bypass capacitor. The bypass capacitor must be inserted as close as possible to the power supply pins of the LSI. Note that the capacitance and frequency characteristics of the bypass capacitor must be appropriate for the operating frequency of the LSI.

- A pair of Vss and VCC for the input/output power supply
  C1 to D1, M4 to M3, V1 to W1, U7 to V7, U12 to V12, Y18 to Y19, M19 to M18, H17 to H18, C20 to B20, A18 to A17, D14 to C14, D13 to C13, D8 to C8, A3 to A2
- A pair of Vss and Vcc for the digital modules
  F3 to F4, K3 to K4, U4 to T4, V6 to U6, V10 to U10, U17 to U16, R18 to R17, L18 to L17, D17 to E17, C15 to D15, C11 to D11, D4 to D5
- A pair of Vss and Vcc for the on-chip oscillator
  K20 to K17, K18 to J20

- A pair of Vss and Vcc for the input/output power supply nearest the USB module
  H3 to H4
- A pair of Vss and Vcc for the A/D converter.
  W19 to U20

Notes on Using a PLL Oscillator Circuit: In the Vcc and Vss connection pattern for the PLL, signal lines from the board power supply pins must be as short as possible and pattern width must be as wide as possible to reduce inductive interference.

In clock operating mode 7, the EXTAL pin is pulled up and the XTAL pin is left open.

Since the analog power supply pins of the PLL are sensitive to the noise, the system may malfunction due to inductive interference at the other power supply pins. To prevent such malfunction, the analog power supply pin Vcc and digital power supply pin VccQ should not supply the same resources on the board if at all possible.

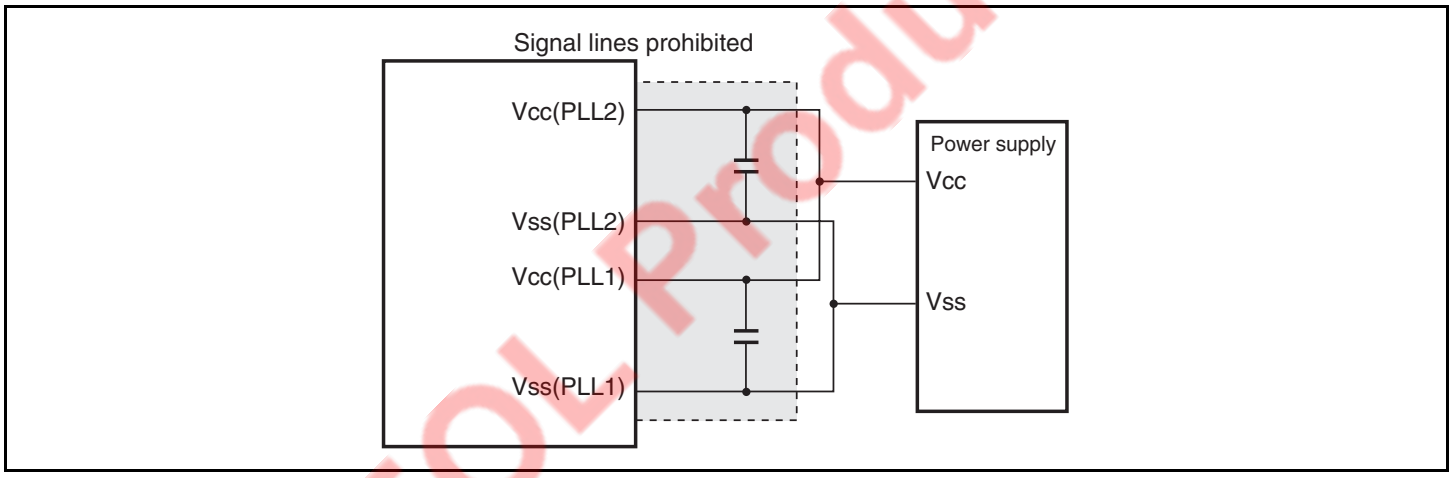


Figure 4.3 Note on Using a PLL Oscillator Circuit

Section 5 Watchdog Timer (WDT)

This LSI includes the watchdog timer (WDT), which enables reset the LSI on overflow of the counter when the value of the counter has not been updated because of a system malfunction.

The WDT is a single channel timer that counts up the clock-settling period when the system leaves standby mode or the temporary periods on standby that occur when the clock frequency is changed. It can also be used as a watchdog timer or interval timer.

5.1 Features

The WDT has the following features:

- Can be used to ensure the clock settling time: The WDT is used in leaving standby mode or the temporary periods on standby that occur when the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Generates internal resets in watchdog timer mode: Internal resets occur after counter overflow. Power-on reset or manual reset can be selected as a reset.
- Interrupt generation in interval timer mode
  An interval timer interrupt is generated when the counter overflows.
- Choice of eight counter input clocks
  Eight clocks (×1 to ×1/4096) that are obtained by dividing the peripheral clock can be selected.

Figure 5.1 shows a block diagram of the WDT.

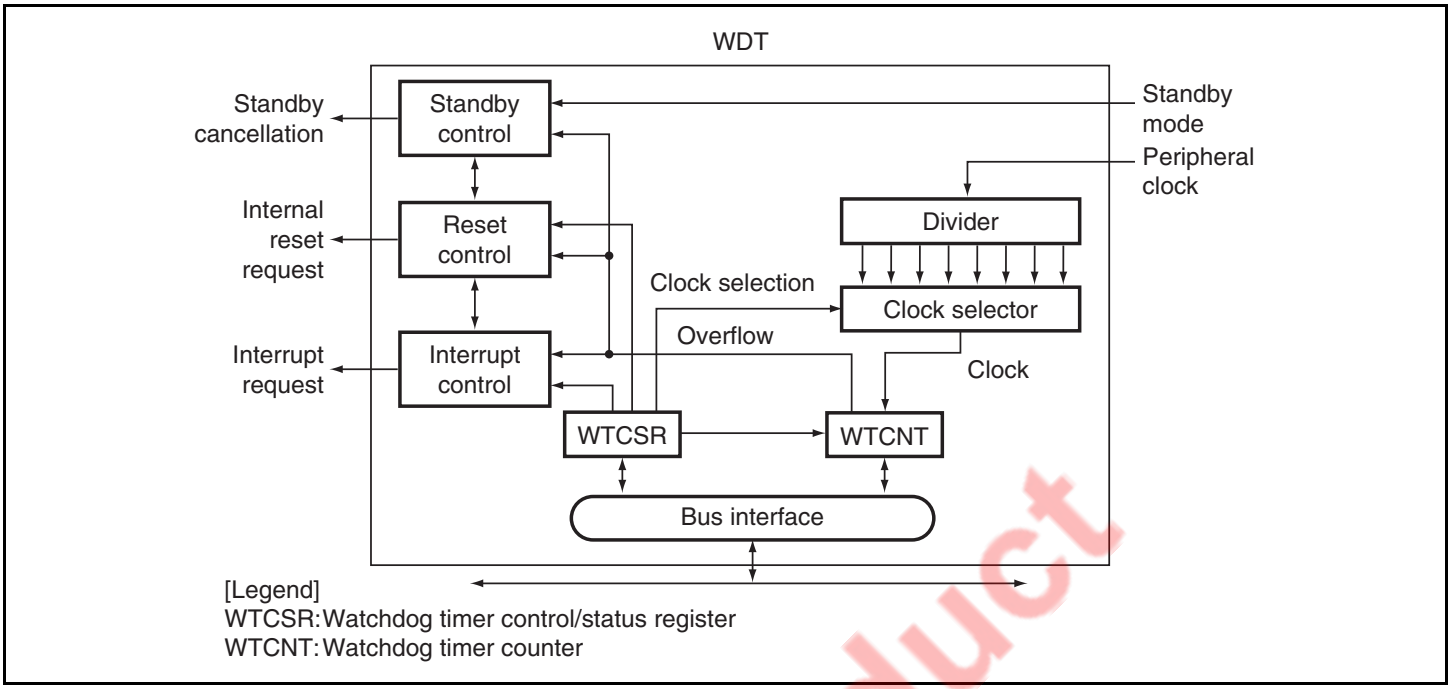


Figure 5.1 Block Diagram of the WDT

5.2 Register Descriptions

The WDT has the following two registers. See section 24, List of Registers, for the addresses and access sizes of these registers.

- Watchdog timer counter (WTCNT)
- Watchdog timer control/status register (WTCSR)

5.2.1 Watchdog Timer Counter (WTCNT)

The watchdog timer counter (WTCNT) is an 8-bit readable/writable register that is incremented by cycles of the selected clock signal. When an overflow occurs, it generates a reset in watchdog timer mode and an interrupt in interval timer mode. The WTCNT counter is initialized to H'00 only by a power-on reset caused by the $\overline{\text{RESETP}}$ pin. Use a word access to write to the WTCNT counter, writing H'5A in the upper byte. Use a byte access to read the WTCNT.

Note: The WTCNT differs from other registers in the prevention of erroneous writes. See section 5.2.3, Notes on Register Access, for details.

5.2.2 Watchdog Timer Control/Status Register (WTCSR)

The watchdog timer control/status register (WTCSR) is an 8-bit readable/writable register composed of bits to select the clock used for the count, overflow flags, and timer enable bit. The WTCSR register holds its value in an internal reset due to WDT overflow. The WTCSR register is initialized to H'00 only by a power-on reset caused by the $\overline{\text{RESETP}}$ pin.

When used to count the clock settling time for canceling a standby, it retains its value after counter overflow. Use a word access to write to the WTCSR counter, writing H'A5 in the upper byte. Use a byte access to read the WTCSR.

Note: The WTCNT differs from other registers in the prevention of erroneous writes. See section 5.2.3, Notes on Register Access, for details.

Bit	Bit Name	Initial Value	R/W	Description
7	TME	0	R/W	Timer Enable Starts and stops timer operation. Clear this bit to 0 when using the WDT in standby mode or when changing the clock frequency. 0: Timer disabled: Count-up stops and WTCNT value is retained 1: Timer enabled
6	WT/$\overline{\text{IT}}$	0	R/W	Timer Mode Select Selects whether to use the WDT as a watchdog timer or an interval timer. 0: Use as interval timer 1: Use as watchdog timer Note: If WT/$\overline{\text{IT}}$ is modified when the WDT is running, the up-count may not be performed correctly.
5	RSTS	0	R/W	Reset Select Selects the type of reset when the WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored. 0: Power-on reset 1: Manual reset

Bit	Bit Name	Initial Value	R/W	Description
4	WOVF	0	R/W	Watchdog Timer Overflow Indicates that the WTCNT has overflowed in watchdog timer mode. This bit is not set in interval timer mode. 0: No overflow 1: WTCNT has overflowed in watchdog timer mode
3	IOVF	0	R/W	Interval Timer Overflow Indicates that the WTCNT has overflowed in interval timer mode. This bit is not set in watchdog timer mode. 0: No overflow 1: WTCNT has overflowed in interval timer mode
2	CKS2	0	R/W	Clock Select
1	CKS1	0	R/W	These bits select the clock to be used for the WTCNT count from the eight types obtainable by dividing the peripheral clock (Pϕ). The overflow period that is shown inside the parenthesis in the table is the value when the peripheral clock (Pϕ) is 15 MHz.
0	CKS0	0	R/W	

Bits 2 to 0	Clock Ratio	Overflow Cycle
000:	1	17 us
001:	1/4	68 us
010:	1/16	273 us
011:	1/32	546 us
100:	1/64	1.09 ms
101:	1/256	4.36 ms
110:	1/1024	17.48 ms
111:	1/4096	69.91 ms

Note: If bits CKS2 to CKS0 are modified when the WDT is running, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not running. In addition, the timing of the first overflow includes deviation. See section 5.4, Precautions to Take when Using the WDT.

5.2.3 Notes on Register Access

The watchdog timer counter (WTCNT) and watchdog timer control/status register (WTCSR) are more difficult to write to than other registers. The procedures for reading or writing to these registers are given below.

Writing to WTCNT and WTCSR: These registers must be written by a word transfer instruction. They cannot be written by a byte or longword transfer instruction. When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as shown in figure 5.2. When writing to WTCSR, set the upper byte to H'A5 and transfer the lower byte as the write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

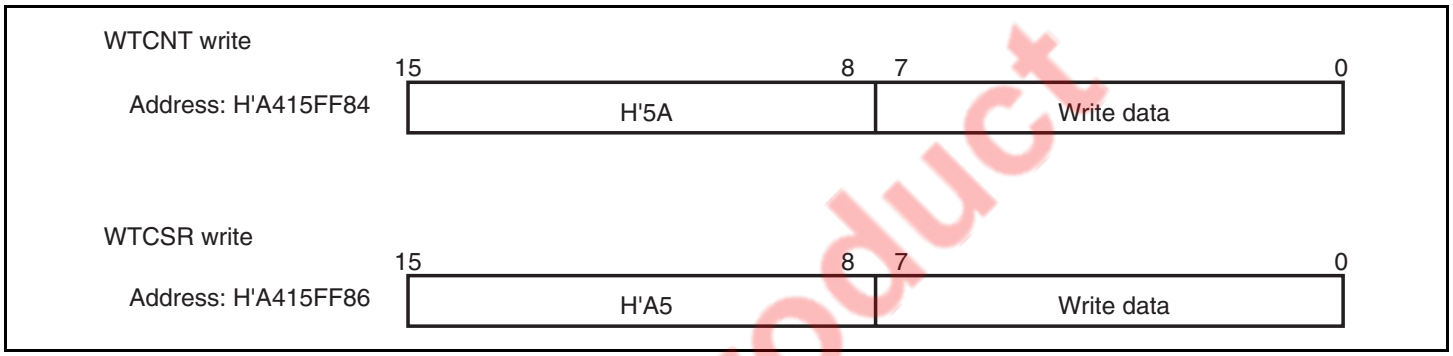


Figure 5.2 Writing to WTCNT and WTCSR

5.3 Use of the WDT

5.3.1 Canceling Standbys

The WDT can be used to cancel standby mode with an interrupt such as an NMI interrupt. The procedure is described below. (The WDT does not operate when resets are used for canceling, so keep the $\overline{\text{RESETP}}$ or $\overline{\text{RESETM}}$ pin low until the clock stabilizes.)

1. Before transitioning to standby mode, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2 to CKS0 bits in WTCSR and the initial values for the counter in the WTCNT. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. The execution of a SLEEP instruction after the STBY bit of the standby control register (STBCR: see section 6, Power-Down Modes) puts the system in standby mode and clock operation then stops.
4. The WDT starts counting by detecting the edge change of the NMI signal.

5. When the WDT count overflows, the CPG starts supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set when this happens.
6. Since the WDT continues counting from H'00, set the STBY bit in the STBCR register to 0 in the interrupt processing program and this will stop the WDT. When the STBY bit remains 1, the LSI again enters the standby mode when the WDT has counted up to H'80. This standby mode can be canceled by power-on resets.

5.3.2 Changing the Frequency

To change the frequency used by the PLL, use the WDT. When changing the frequency only by switching the divider, do not use the WDT.

1. Before changing the frequency, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2 to CKS0 bits of WTCSR and the initial values for the counter in the WTCNT counter. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. When the frequency control register (FRQCR) is written, the processor stops temporarily. The WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and the processor resumes operation. The WOVF in WTCSR is not set when this happens.
5. The counter stops at the values H'00.
6. Before changing the WTCNT after the execution of the frequency change instruction, always confirm that the value of the WTCNT is H'00 by reading the WTCNT.

5.3.3 Using Watchdog Timer Mode

1. Set the WT/$\overline{\text{IT}}$ bit in the WTCSR to 1, set the reset type in the RSTS bit, set the type of count clock in the CKS2 to CKS0 bits, and set the initial value of the counter in the WTCNT.
2. Set the TME bit in WTCSR to 1 to start the count in watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 to prevent the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1 and generates the type of reset specified by the RSTS bit. The counter then resumes counting.

5.3.4 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every overflow of the counter. This enables interrupts to be generated at set periods.

1. Clear the WT/$\overline{IT}$ bit in the WTCSR to 0, set the type of count clock in the CKS2 to CKS0 bits, and set the initial value of the counter in the WTCNT.
2. Set the TME bit in WTCSR to 1 to start the count in interval timer mode.
3. When the counter overflows, the WDT sets the IOVF in WTCSR to 1 and an interval timer interrupt request is sent to INTC. The counter then resumes counting.

5.4 Precautions to Take when Using the WDT

Pay attention to the following points when using the WDT in either the interval timer or watchdog timer mode.

1. Timer tolerance

   After timer operation has started, the period from the power-on reset point to the first count up timing of the WTCNT varies depending on the time period that is set by the TME bit of the WTCSR register. The shortest such time period is thus one cycle of the peripheral clock, pϕ, while the longest is the result of frequency division according to the value in CKS2 to CKS0. The timing of subsequent incrementation is in accord with the selected frequency divisor. This time difference is referred to as timer variation. This also applies to the timing of the first incrementation after the WTCNT register has been written to during timer operation.
2. Do not set WTCNT to H'FF

   When the value in WTCNT reaches H'FF, the WDT assumes that an overflow has occurred. Accordingly, when H'FF is placed in WTCNT, an interval timer interrupt or WDT reset will occur immediately, regardless of the current clock selection by bits CKS2 to CKS0.

Section 6 Power-Down Modes

In the low power-consumption modes, operation of some of the internal peripheral modules and of the CPU stops. This leads to reduced power consumption. These modes are canceled by a reset or interrupt.

6.1 Features

6.1.1 Power-Down Modes

This LSI has the following power-down modes and function:

1. Sleep mode
2. Standby mode
3. Module standby function

Table 6.1 shows the transition conditions for entering the modes from the program execution state, as well as the CPU and peripheral module states in each mode and the procedures for canceling each mode.

Table 6.1 States of Power-Down Modes

		State*						
Mode	Transition Conditions	CPG	CPU	CPU Register	On-Chip Memory	On-Chip Peripheral Modules	External Memory	Canceling Procedure
Sleep mode	Execute SLEEP instruction with STBY bit cleared to 0 in STBCR	Runs	Halts	Held	Halts (The contents are retained.)	The UBC stops. Other modules continue to run.	Refreshed automati-cally	1. Interrupt 2. Reset
Standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR	Halts	Halts	Held	Halts (The contents are retained.)	Halt	Self-refreshed	1. Interrupt 2. Reset
Module standby function	Set the MSTP bits in STBCR, STBCR2, STBCR3, and STBCR4 to 1 (with the exception of the MSTP bits for the USB module; clear these bits).	Runs	Runs	Held	The specified module stops (the contents are retained).	Specified module halts	Refreshed automati-cally	1. Clear MSTP bit to 0. (with the exception of the MSTP bits for the USB module; set these bits). 2. Power-on reset

Note: * The pin state is retained or set to high impedance. For details, see Appendix A, Pin States.

6.1.2 Reset

A reset is used at power-on or to re-execute from the initial state. This LSI supports two types of reset: power-on reset and manual reset. In power-on reset, any processing to be currently executed is terminated and any events not executed are canceled to execute reset processing immediately. In manual reset, processing required to maintain external memory contents is continued. The following shows the conditions in which power-on reset or manual reset occurs.

- Power-on reset
  1. A low level signal is input to the $\overline{\text{RESETP}}$ pin.
  2. The WDT counter overflows if WDT starts counting while the WT/$\overline{\text{IT}}$ and RSTS bits of the WTCSR are set to 1 and cleared to 0, respectively.
  3. An H-UDI reset occurs. (For details on H-UDI reset, refer to section 15, User Debugging Interface (H-UDI).)

- Manual-on reset
  1. A low signal is input to the $\overline{\text{RESETM}}$ pin.
  2. The WDT counter overflows if WDT starts counting while the WT/$\overline{\text{IT}}$ and RSTS bits of the WTCSR are set to 1.

6.1.3 Input/Output Pins

Table 6.2 lists the pins used for the power-down modes.

Table 6.2 Pin Configuration

Pin Name	Symbol	I/O	Description
Processing state 1	STATUS1	Output	Indicates the operational state of this LSI.
Processing state 0	STATUS0		HH: Manual reset HL: Sleep mode LH: Standby mode LL: Normal operation
Power-on reset	RESETP	Input	Inputting low level signal to this pin cause a transition to power-on reset processing.
Manual reset	RESETM	Input	Inputting low level signal to this pin cause a transition to manual reset processing.

Note: H and L indicate high and low levels, respectively. STATUS1 and STATUS0 indicate the pin status in this order. To use this pin as the STATUS pin, a PFC setting is required. For details on this, see section 22, Pin Function Controller (PFC).

6.2 Register Descriptions

The following registers are used in the low power-consumption modes. For the addresses and access sizes of these registers, see section 24, List of Registers.

- Standby control register (STBCR)
- Standby control register 2 (STBCR2)
- Standby control register 3 (STBCR3)
- Standby control register 4 (STBCR4)

6.2.1 Standby Control Register (STBCR)

The standby control register (STBCR) is an 8-bit readable/writable register that specifies the state of the power-down mode. This register is initialized (to H'00) by a power-on reset but retains its previous value after a manual reset or a period in the standby mode. Only byte access is valid.

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Software Standby Specifies transition to software standby mode. 0: Executing SLEEP instruction puts chip into sleep mode. 1: Executing SLEEP instruction puts chip into software standby mode.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

6.2.2 Standby Control Register 2 (STBCR2)

The standby control register 2 (STBCR2) is a readable/writable 8-bit register that controls the operation of modules in the power-down mode. STBCR2 is initialized (to H'00) by a power-on reset but retains its previous value after a manual reset or a period in the standby mode. Only byte access is valid.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP10	0	R/W	Module Stop 10 When the MSTP10 bit is set to 1, the supply of the clock to the H-UDI is halted. 0: H-UDI runs. 1: Clock supply to H-UDI halted.
6	MSTP9	0	R/W	Module Stop 9 When the MSTP9 bit is set to 1, the supply of the clock to the UBC is halted. 0: UBC runs. 1: Clock supply to UBC halted.
5	MSTP8	0	R/W	Module Stop 8 When the MSTP8 bit is set to 1, the supply of the clock to the DMAC is halted. 0: DMAC runs. 1: Clock supply to DMAC halted.
4	MSTP7	0	R/W	Module Stop 7 When the MSTP7 bit is set to 1, the supply of the clock to the DSP is halted. 0: DSP runs. 1: Clock supply to DSP halted.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	MSTP5	0	R/W	Module Stop 5 When the MSTP5 bit is set to 1, the supply of the clock to the cache memory is halted. 0: The cache memory runs. 1: Clock supply to the cache memory halted.
1	MSTP4	0	R/W	Module Stop 4 When the MSTP4 bit is set to 1, the supply of the clock to the U memory is halted. 0: The U memory runs. 1: Clock supply to the U memory halted.
0	MSTP3	0	R/W	Module Stop 3 When the MSTP3 bit is set to 1, the supply of the clock to the X/Y memory is halted. 0: The X/Y memory runs. 1: Clock supply to the X/Y memory halted.

6.2.3 Standby Control Register 3 (STBCR3)

STBCR3 is a readable/writable 8-bit register used to select whether or not individual modules operate in power-down mode. STBCR3 is initialized (to H'00) by a power-on reset, but retains its previous value after a manual reset or a period in the standby mode. Only byte access is valid.

Bit	Bit Name	Initial Value	R/W	Description
7	HIZ	0	R/W	Port High Impedance This bit selects whether the state of a specified pin is retained or the pin is placed in the high-impedance state. See Appendix A, Pin States to determine the pin to which this control is applied. Do not set this bit when the TME bit of WTSCR of the WDT is 1. When setting the output pin to the high-impedance state, set the HIZ bit with the TME bit being 0. 0: The pin state is held in standby mode. 1: The pin state is set to the high-impedance state in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
6	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
5	MSTP35	0	R/W	Module Stop 35 When the MSTP35 bit is set to 1, supply of the clock to the CMT0 stops. 0: The CMT0 runs. 1: Supply of the clock to the GMT0 stops.
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	MSTP33	0	R/W	Module Stop 33 When the MSTP33 bit is set to 1, supply of the clock to the ADC stops. 0: The ADC runs. 1: Supply of the clock to the ADC stops.
2	MSTP32	0	R/W	Module Stop 32 When the MSTP32 bit is set to 1, supply of the clock to the SCIF2 stops. 0: The SCIF2 runs. 1: Supply of the clock to the SCIF2 stops.
1	MSTP31	0	R/W	Module Stop 31 When the MSTP31 bit is set to 1, supply of the clock to the SCIF1 stops. 0: The SCIF1 runs. 1: Supply of the clock to the SCIF1 stops.
0	MSTP30	0	R/W	Module Stop 30 When the MSTP30 bit is set to 1, supply of the clock to the SCIF0 stops. 0: The SCIF0 runs. 1: Supply of the clock to the SCIF0 stops.

6.2.4 Standby Control Register 4 (STBCR4)

STBCR4 is a readable/writable 8-bit register used to select whether or not individual modules operate in power-down mode. STBCR4 is initialized (to H'00) by a power-on reset, but retains its previous value after a manual reset or a period in the standby mode. Only byte access is valid.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	MSTP46	0	R/W	Module Stop 46 0: The USB module stops. 1: Supply of the clock to the USB is started.
5	MSTP45	0	R/W	Module Stop 45 When the MSTP45 bit is set to 1, supply of the clock to the MTU stops. 0: The MTU runs. 1: Supply of the clock to the MTU stops.
4	MSTP44	0	R/W	Module Stop 44 When the MSTP44 bit is set to 1, supply of the clock to the POE stops. 0: The POE runs. 1: Supply of the clock to the POE stops.
3	MSTP43	0	R/W	Module Stop 43 When the MSTP43 bit is set to 1, supply of the clock to the CMT1 stops. 0: The CMT1 runs. 1: Supply of the clock to the CMT1 stops.
2	MSTP42	0	R/W	Module Stop 42 When the MSTP42 bit is set to 1, supply of the clock to the IIC2 stops. 0: The IIC2 runs. 1: Supply of the clock to the IIC2 stops.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

6.3 Operation

6.3.1 Sleep Mode

1. Transition to Sleep Mode

   Executing the SLEEP instruction when the STBY bit in STBCR is 0 causes a transition from the program execution state to sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip modules continue to run in sleep mode, but the on-chip memory is not accessible. If the on-chip memory is accessed by, for example, the DMAC, the access is ignored and the value read is not defined. Clock pulses continue to be output on the CKIO and CKIO2 pins. In sleep mode, a high signal and low signal are output from the STATUS1 and STATUS0 pins, respectively.

2. Canceling Sleep Mode

   Sleep mode is canceled by an interrupt ($\overline{\text{NMI}}$, $\overline{\text{IRQ}}$, and on-chip peripheral module) or reset. Interrupts are accepted in sleep mode even when the BL bit in the SR register is 1. If necessary, save SPC and SSR to the stack before executing the SLEEP instruction.

- Canceling with an Interrupt

  When an $\overline{\text{NMI}}$, $\overline{\text{IRQ}}$, or on-chip peripheral module interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. A code indicating the interrupt source is set in the INTEVT2 registers.

- Canceling with a Reset

  Sleep mode is canceled by a power-on reset or a manual reset.

6.3.2 Standby Mode

1. Transition to Standby Mode

   The LSI switches from a program execution state to a standby mode by executing the SLEEP instruction when the STBY bit is 1 in STBCR register. In standby mode, not only the CPU but also the clock and on-chip peripheral modules halt. The clock outputs from the CKIO and CKIO2 pins also halt.

   The contents of the CPU and cache registers remain unchanged. Some registers of on-chip peripheral modules are, however, initialized. Table 6.3 lists the states of on-chip peripheral modules registers in standby mode.

Table 6.3 Register States in Standby Mode

Module	Registers Initialized	Registers Retaining Data
Interrupt controller (INTC)	—	All registers
On-chip clock pulse generator (CPG)	—	All registers
User break controller (UBC)	—	All registers
Bus state controller (BSC)	—	All registers
A/D converter (ADC)	All registers	—
I/O port	—	All registers
H-UDI	—	All registers
SCIF	—	All registers
USB	—	All registers
MTU	All registers	—
POE	—	All registers
DMAC	—	All registers
CMT	—	All registers
IIC2	—	All registers

The procedure for switching to standby mode is as follows:

A. Clear the TME bit in the WDT's timer control register (WTCSR) to 0 to stop the WDT.
B. Set the WDT's timer counter (WTCNT) to 0 and the CKS2 to CKS0 bits in the WTCSR register to appropriate values to secure the specified oscillation settling time.
C. After the STBY bit in the STBCR register is set to 1, a SLEEP instruction is executed.
D. Standby mode is entered and the clocks within the chip are halted. The STATUS1 and STATUS0 pins output low and high, respectively.

2. Canceling Standby Mode

   Standby mode is canceled by interrupts ($\overline{\text{NMI}}$, $\overline{\text{IRQ}}$) or a reset.

- Canceling with an Interrupt

  The on-chip WDT can be used for hot starts. When an interrupt request is detected at the rising or falling edge of $\overline{\text{NMI}}$ or $\overline{\text{IRQ}}$, the clock will be supplied to the entire chip and standby mode canceled after the time set in the WDT's timer control/status register has elapsed. The STATUS1 and STATUS0 pins go low. Interrupt handling then begins and a code indicating the interrupt source is set in the INTEVT2 registers. After the branch to the interrupt handling routine, clear the STBY bit in the STBCR register. WDT stops automatically. If the STBY bit is not cleared, WDT continues operation and a transition is made to standby mode* when the WTCNT reaches H'80. A manual reset will not be accepted while the STBY bit is set. Interrupts are accepted in standby mode even when the BL bit in the SR register is 1. If necessary, save SPC and SSR to the stack before executing the SLEEP instruction.

  Immediately after an interrupt is detected and until the system is taken out of standby mode, the phase of the clock outputs from the CKIO and CKIO2 pins may be unstable.

Notes: * This standby mode can be canceled only by a power-on reset.

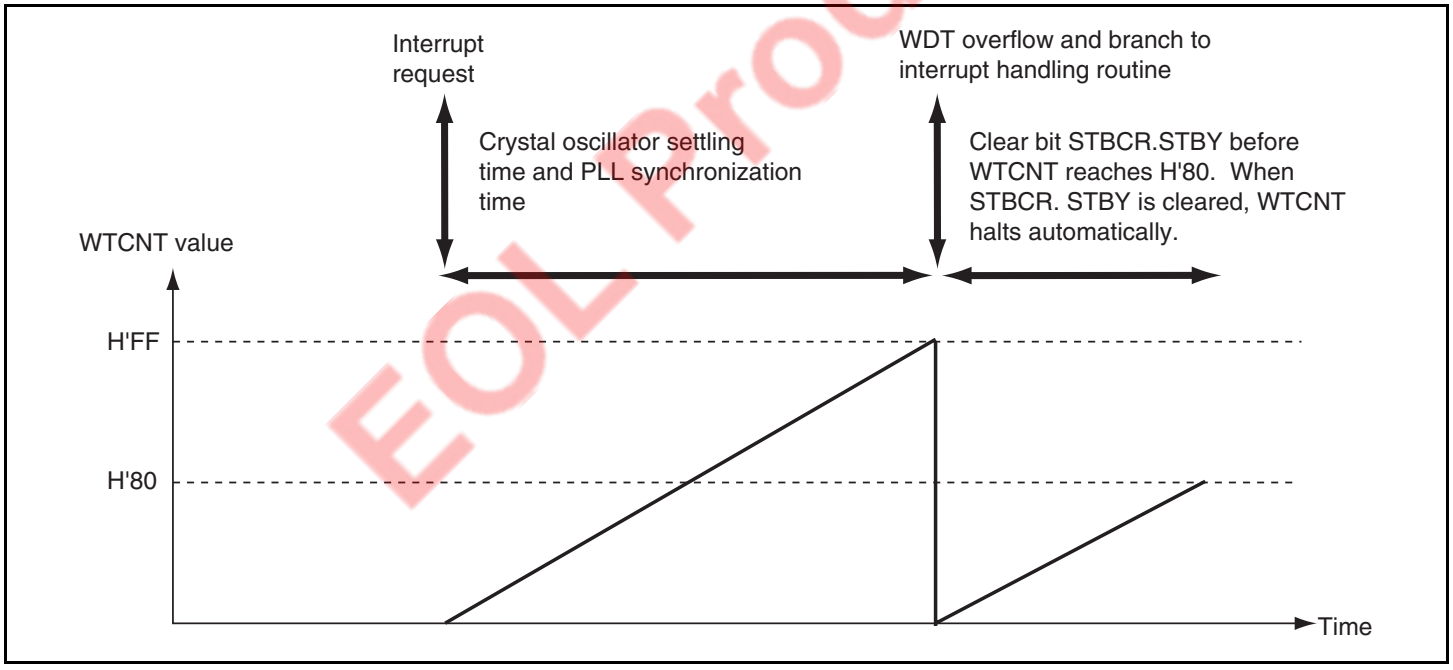


Figure 6.1 Canceling Standby Mode with STBCR.STBY

- Canceling with a Reset

  Standby mode is canceled by a reset using the $\overline{\text{RESETP}}$ or $\overline{\text{RESETM}}$ pin. Keep the $\overline{\text{RESETP}}$ or $\overline{\text{RESETM}}$ pin low until the clock oscillation settles. The internal clock will continue to be output to the CKIO pin.

6.3.3 Module Standby Function

1. Transition to Module Standby Function

   Setting the standby control register MSTP bits to 1 halts the supply of clocks to the corresponding on-chip peripheral modules (however, the initial state of the USB stops). This function can be used to reduce the power consumption in normal mode and sleep mode.

   Disable a module before placing it in the module standby mode. In addition, do not access the module's registers while it is in the module-standby state.

   In module standby state, the functions of the external pins of the on-chip peripheral modules change depending on the on-chip peripheral module. For details on this, see Appendix A, Pin States. The states of the registers are the same as in the standby mode. See table 6.3.

2. Canceling Module Standby Function

   The module standby function except USB can be canceled by clearing the MSTP bits to 0, or by a power-on reset. In the case of the USB module, setting the corresponding MSTP bit to 1 cancels the module standby state. When taking a module out of the module standby state by clearing the corresponding bit to 0 (or setting it to 1 in the case of the USB module), read the bit to confirm that it has been cleared to 0 (or set to 1 in the USB case).

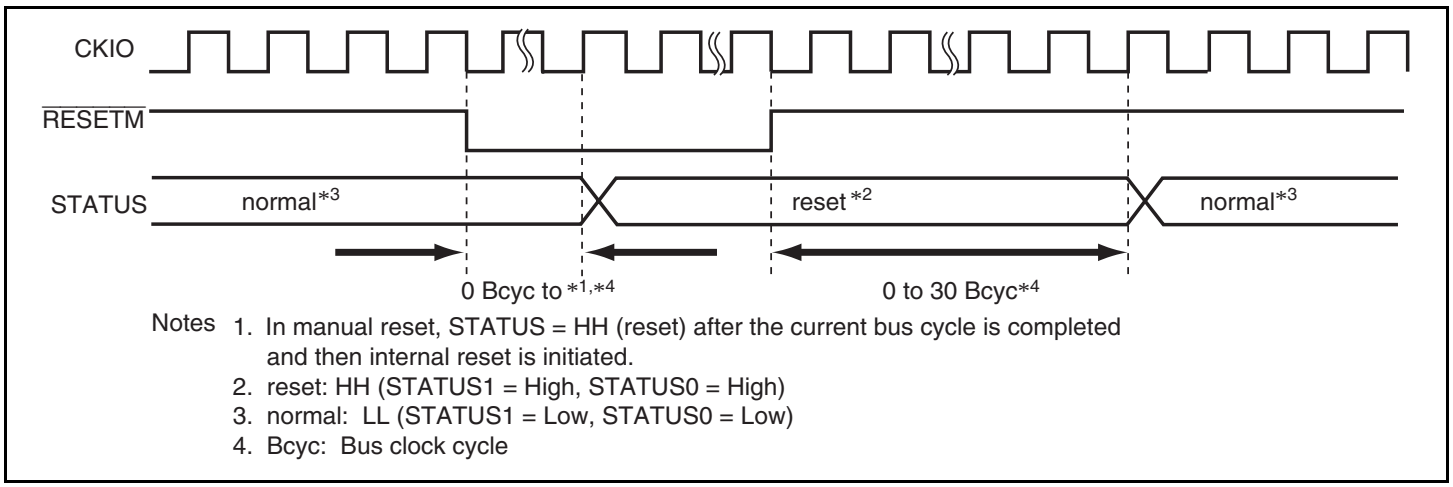
6.3.4 STATUS Pin Change Timings

To use these pins as the STATUS1 and STATUS0 pins, the corresponding setting must be made in the PFC. For details on setting of the PFC, see section 22, Pin Function Controller (PFC). A power-on reset initializes the PFC setting; the default value selects operation as PTC15 and PTC14 port-input pins. Accordingly, when you wish to use these pins for the STATUS function immediately after a power-on reset, take the following steps. This also applies to power-on resets from the WDT and resets from the H-UDI.

1. Pull up the STATUS1 and STATUS0 pins.
2. Change the PFC setting made by power-on reset processing so that the STATUS function is selected for these pins.

Both STATUS1 and STATUS0 become high during a power-on reset and are low on completion of power-on reset processing. The state of the LSI is thus indicated. The timing of the level changes of the STATUS1 and STATUS0 pins is shown below.

1. Manual Reset

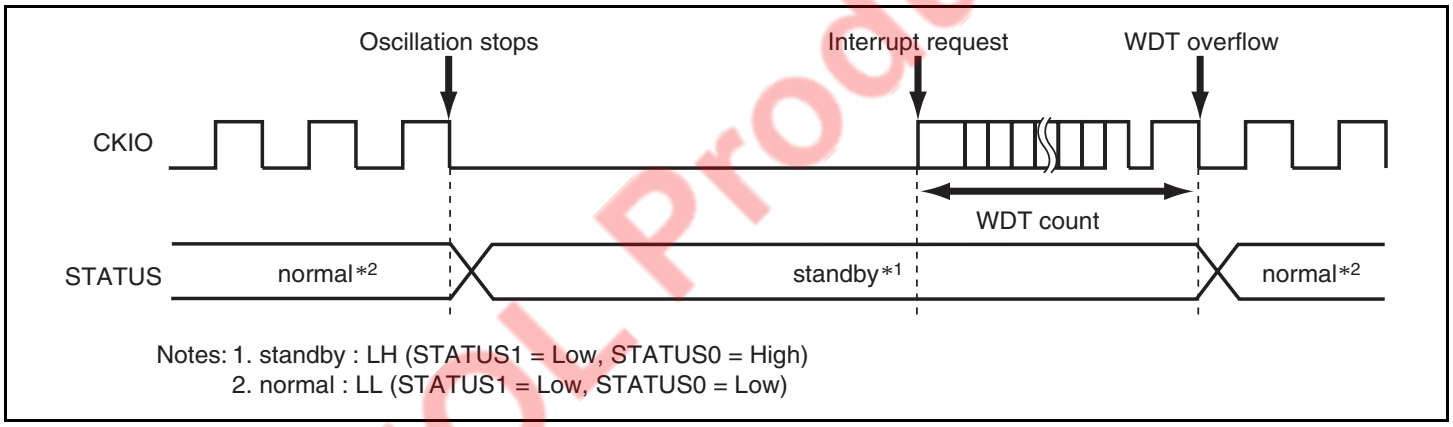


Notes 1. In manual reset, STATUS = HH (reset) after the current bus cycle is completed and then internal reset is initiated.
2. reset: HH (STATUS1 = High, STATUS0 = High)
3. normal: LL (STATUS1 = Low, STATUS0 = Low)
4. Bcyc: Bus clock cycle

Figure 6.2 STATUS Output at Manual Reset

2. Standby Mode

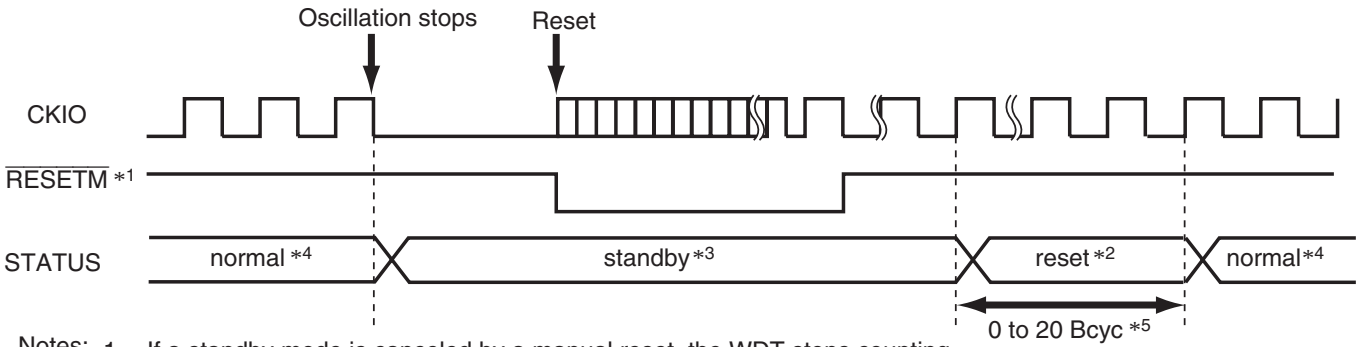
A Standby mode is canceled by an interrupt



Notes: 1. standby : LH (STATUS1 = Low, STATUS0 = High)
2. normal : LL (STATUS1 = Low, STATUS0 = Low)

Figure 6.3 STATUS Output when Standby Mode is Canceled by an Interrupt

B Standby mode is canceled by a manual reset

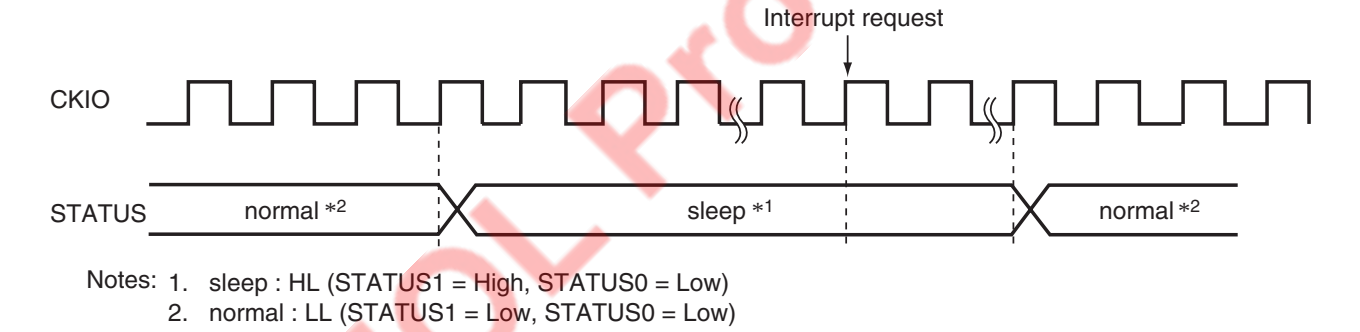


Notes:
1. If a standby mode is canceled by a manual reset, the WDT stops counting. RESETM must be kept low for the PLL oscillation stabilization time.
2. reset : HH (STATUS1 = High, STATUS0 = High)
3. standby : LH (STATUS1 = Low, STATUS0 = High)
4. normal : LL (STATUS1 = Low, STATUS0 = Low)
5. Bcyc : Bus clock cycle

Figure 6.4 STATUS Output When Software Standby Mode is Canceled by a Manual Reset

3. Sleep Mode

A Sleep mode is canceled by an interrupt



Notes:
1. sleep : HL (STATUS1 = High, STATUS0 = Low)
2. normal : LL (STATUS1 = Low, STATUS0 = Low)

Figure 6.5 STATUS Output when Sleep Mode is Canceled by an Interrupt

B Sleep standby mode is canceled by a manual reset

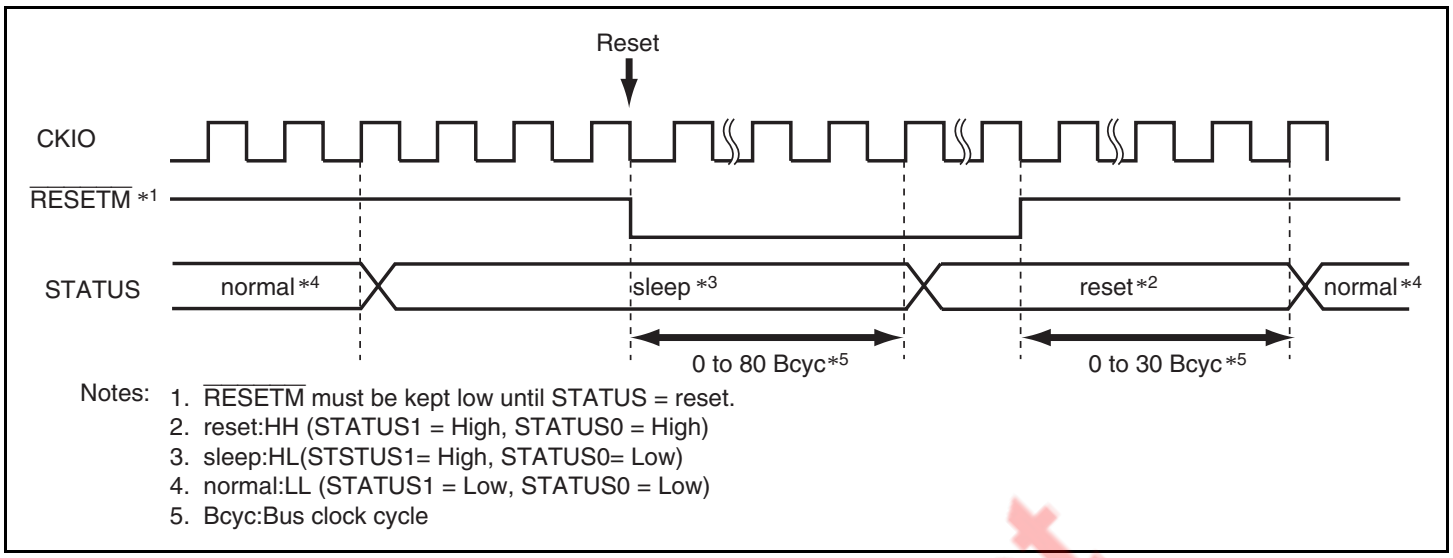


Figure 6.6 STATUS Output When Sleep Mode is Canceled by a Manual Reset

Section 7 Cache

7.1 Features

The cache specifications are listed in table 7.1.

Table 7.1 Cache Specifications

Parameter	Specification
Capacity	16 kbytes
Structure	Instructions/data mixed, 4-way set associative
Locking	Way 2 and way 3 are lockable
Line size	16 bytes
Number of entries	256 entries/way
Write system	P0, P1, P3: Write-back/write-through selectable
Replacement method	Least-recently-used (LRU) algorithm

In this LSI, the address space is partitioned into five subdivisions, and the cache access method is determined by the address. Table 7.2 shows the kind of cache access available in each address space subdivision.

Table 7.2 Address Space Subdivisions and Cache Operation

Address Bits A31 to 29	Address Space Subdivision	Cache Operation
0xx	P0	Write-back/write-through selectable
100	P1	Write-back/write-through selectable
101	P2	Non-cacheable
110	P3	Write-back/write-through selectable
111	P4	I/O area, non-cacheable

Note that area P4 is an I/O area, to which the addresses of on-chip registers, etc., are allocated.

To ensure data consistency, the cache stores 32-bit addresses with the upper 3 bits masked to 0.

7.1.1 Cache Structure

The cache mixes data and instructions and uses a 4-way set associative system. It is composed of four ways (banks), each of which is divided into an address section and a data section. Each of the address and data sections is divided into 256 entries. The data section of the entry is called a line. Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 4 kbytes (16 bytes × 256 entries), with a total of 16 kbytes in the cache as a whole (4 ways). Figure 7.1 shows the cache structure.

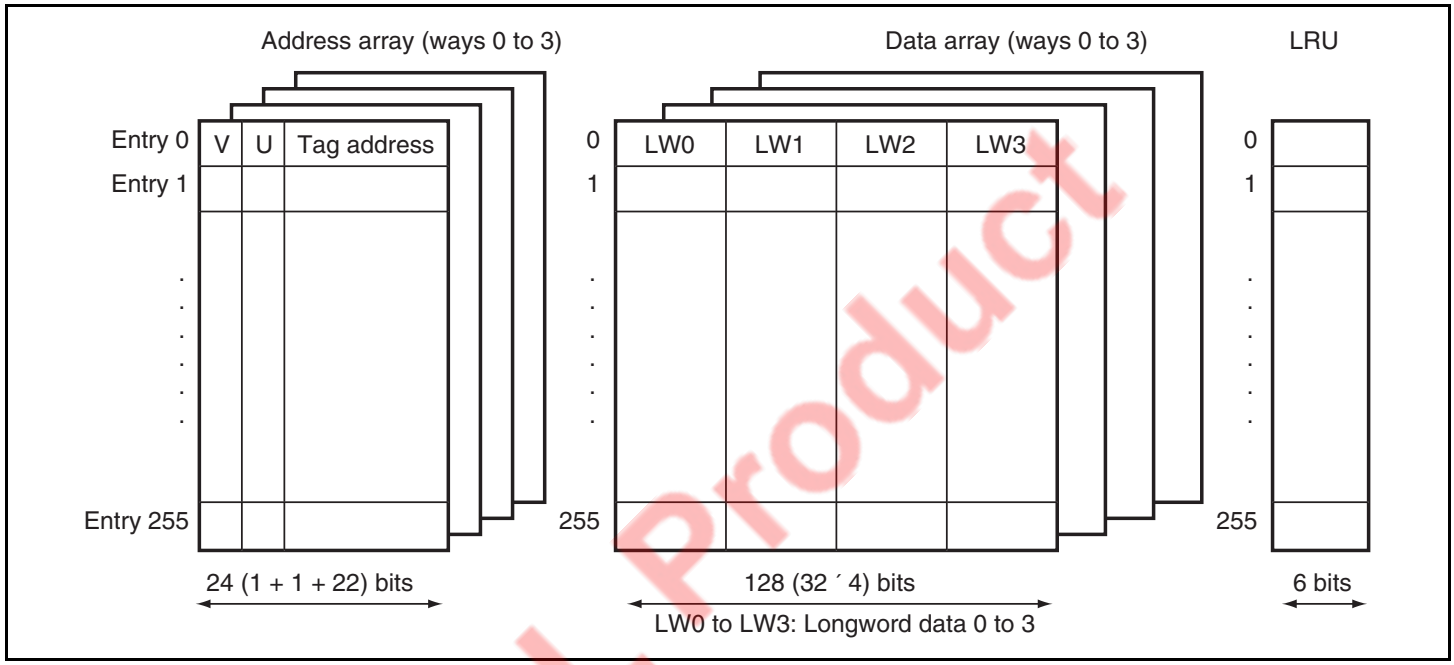


Figure 7.1 Cache Structure

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

In this LSI, the top three of 32 physical address bits are used as shadow bits (see section 12, Bus State Controller (BSC)), and therefore the top three bits of the tag address are cleared to 0.

The V and U bits are initialized to 0 by a power-on reset and retain the previous value by a manual reset, standby mode, module standby mode, and sleep mode. The tag address is not initialized by a power-on or manual reset, standby mode, module standby mode, and sleep mode.

Data Array: Holds a 16-byte instruction or data. Entries are registered in the cache in line units (16 bytes). The data array is not initialized by a power-on or manual reset, standby mode, module standby mode, and sleep mode.

LRU: With the 4-way set associative system, up to four instructions or data with the same entry address (address bits 11 to 4) can be registered in the cache. When an entry is registered, LRU shows which of the four ways it is recorded in. There are six LRU bits, controlled by hardware. A least-recently-used (LRU) algorithm is used to select the way that has been least recently accessed.

Six LRU bits indicate the way to be replaced in case of a cache miss.

The relationship between LRU and way replacement is shown is table 7.3 when the cache lock function is not used 1 concerning the case where the cache lock function is used, see section 7.2.2, Cache Control Register 2 (CCR2). If a bit pattern other than those listed in table 7.3 is set in the LRU bits by software, the cache will not function correctly. When modifying the LRU bits by software, set one of the patterns listed in table 7.3.

The LRU bits are initialized to 000000 by a power-on reset and retaining the previous value by a manual reset, standby mode, module standby mode, and sleep mode.

Table 7.3 LRU and Way Replacement

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

7.2 Register Descriptions

The cache has the following registers.

- Cache control register 1 (CCR1)
- Cache control register 2 (CCR2)

7.2.1 Cache Control Register 1 (CCR1)

The cache is enabled or disabled using the CE bit in CCR1. CCR1 also has the CF bit (which invalidates all cache entries), and the WT and WB bits (which select either write-through mode or write-back mode). Programs that change the contents of CCR1 should be placed in an address space that is not cached. When updating the contents of CCR1, bits 31 to 4 must always be cleared to 0.

CCR1 is initialized to H'00000000 by a power-on or manual reset and retain the previous value by standby mode, module standby mode, and sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CF	0	R/W	Cache Flush Writing 1 flushes all cache entries (clears the V, U, and LRU bits of all cache entries to 0). Always reads 0. Write-back to external memory is not performed when the cache is flushed.
2	WB	0	R/W	Write Back Switches write-back/write-through the cache's operating mode for area P1. 0: Write-through mode 1: Write-back mode
1	WT	0	R/W	Write Through Indicates the cache's operating mode for areas P0 and P3. 0: Write-back mode 1: Write-through mode

Bit	Bit Name	Initial Value	R/W	Description
0	CE	0	R/W	Cache Enable Indicates whether the cache function is used. 0: Cache not used 1: Cache used

7.2.2 Cache Control Register 2 (CCR2)

CCR2 is used to enable or disable the cache locking function and is valid in cache locking mode only. In cache locking mode, the DSP bit (bit 12) in the status register (SR) of the CPU is set to 1. Alternatively, the lock enable bit (bit 16) in CCR2 is set to 1. In the non-cache-locking mode, the cache locking function is invalid.

When a cache miss occurs in cache locking mode by executing the prefetch instruction (PREF @Rn), the line of data pointed to by Rn is loaded into the cache according to bits 9 and 8 (the W3LOAD and W3LOCK bits) and bits 1 and 0 (the W2LOAD and W2LOCK bits) in CCR2. The relationship between the setting of each bit and a way, to be replaced when the prefetch instruction is executed, are listed in table 7.4. On the other hand, when the prefetch instruction is executed and a cache hit occurs, new data is not fetched and the entry which is already enabled is held. For example, when the prefetch instruction is executed with W3LOAD = 1 and W3LOCK = 1 specified in cache locking mode while one-line data already exists in way 0 which is specified by Rn, a cache hit occurs and data is not fetched to way 3.

In the cache access other than the prefetch instruction in cache locking mode, ways to be replaced by bits W3LOCK and W2LOCK are restricted. The relationship between the setting of each bit in CCR2 and ways to be replaced are listed in table 7.5.

The program that change the contents of CCR2 should be placed in an address space that is not cached.

CCR2 is initialized to H'00000000 by a power-on or manual reset and retain the previous value by standby mode, module standby mode, and sleep mode.

Bit	Bit Name	Initial value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	LE	0	R/W	Lock Enable This bit enables or disables the cache locking function. 0: Cache locking mode is entered when SR.DSP=1 1: Cache locking mode is entered regardless of the value of SR.DSP
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	W3LOAD	0	R/W	Way 3 Load
8	W3LOCK	0	R/W	Way 3 Lock When a cache miss occurs by the prefetch instruction while W3LOAD = 1 and W3LOCK = 1 in cache locking mode, the data is always loaded into way 3. Under any other condition, the prefetched data is loaded into the way to which LRU points.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	W2LOAD	0	R/W	Way 2 Load
0	W2LOCK	0	R/W	Way 2 Lock When a cache miss occurs by the prefetch instruction while W2LOAD = 1 and W2LOCK in cache locking mode, the data is always loaded into way 2. Under any other condition, the prefetched data is loaded into the way to which LRU points.

Note: The W2LOAD and W3LOAD bits should not be set to 1 at the same time.

Table 7.4 Way to be Replaced when a Cache Miss Occurs in PREF Instruction

Cache Locking Mode Bit	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Way to be Replaced
0	*	*	*	*	Decided by LRU (table 7.3)
1	*	0	*	0	Decided by LRU (table 7.3)
1	*	0	0	1	Decided by LRU (table 7.6)
1	0	1	*	0	Decided by LRU (table 7.7)
1	0	1	0	1	Decided by LRU (table 7.8)
1	0	*	1	1	Way 2
1	1	1	0	*	Way 3

[Legend]

* : Don't care

Note: The W2LOAD and W3LOAD bits should not be set to 1 at the same time.

Table 7.5 Way to be Replaced when a Cache Miss Occurs in Other than PREF Instruction

Cache Locking Mode Bit	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Way to be Replaced
0	*	*	*	*	Decided by LRU (table 7.3)
1	*	0	*	0	Decided by LRU (table 7.3)
1	*	0	*	1	Decided by LRU (table 7.6)
1	*	1	*	0	Decided by LRU (table 7.7)
1	*	1	*	1	Decided by LRU (table 7.8)

[Legend]

* : Don't care

Note: The W2LOAD and W3LOAD bits should not be set to 1 at the same time.

Table 7.6 LRU and Way Replacement (when W2LOCK=1 and W3LOCK=0)

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

Table 7.7 LRU and Way Replacement (when W2LOCK=0 and W3LOCK=1)

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

Table 7.8 LRU and Way Replacement (when W2LOCK=1 and W3LOCK=1)

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

7.3 Cache Operation

7.3.1 Searching Cache

If the cache is enabled (CE bit in CCR register is 1), whenever instructions or data in spaces of P0, P1, and P3 are accessed the cache will be searched to see if the desired instruction or data is in the cache. Figure 7.2 illustrates the method by which the cache is searched. The cache is a physical cache of which tag address hold an address.

Entries are selected using bits 11 to 4 of the address used to access memory (virtual address) and the tag address of that entry is read. The physical address (bits 31 to 12) after translation and the physical address read from the address section are compared. The address comparison uses all four ways. When the comparison shows a match and the selected entry is valid (V = 1), a cache hit occurs. When the comparison does not show a match or the selected entry is not valid (V = 0), a cache miss occurs. Figure 7.2 shows a hit on way 1.

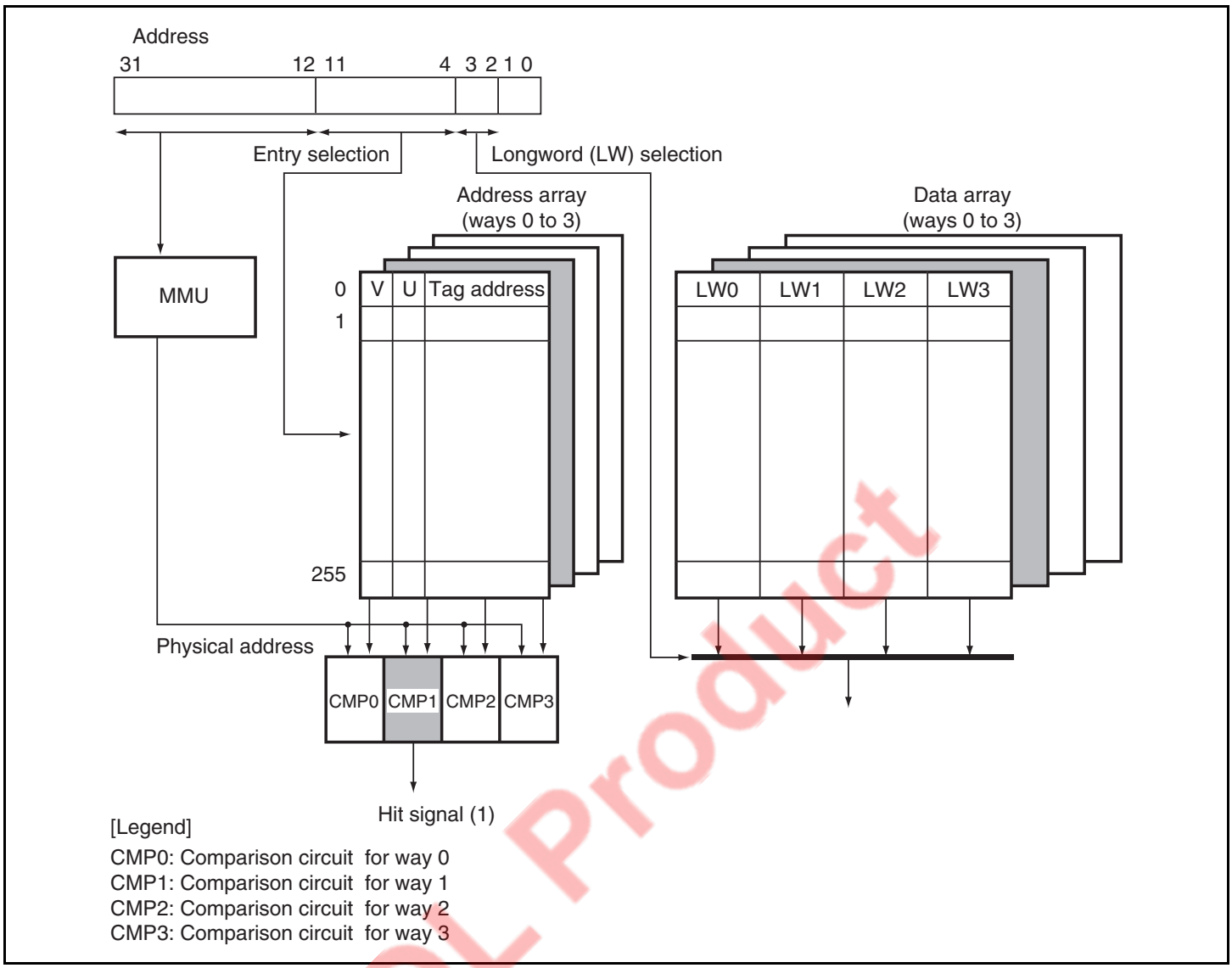


Figure 7.2 Cache Search Scheme

7.3.2 Read Access

Read Hit: In a read access, instructions and data are transferred from the cache to the CPU. LRU is updated so that the hit way is the latest.

Read Miss: An external bus cycle starts and the entry is updated. The way replaced follows table 7.5. Entries are updated in 16-byte units. When the desired instruction or data that caused the miss is loaded from external memory to the cache, the instruction or data is transferred to the CPU in parallel with being loaded to the cache. When it is loaded in the cache, the U bit is cleared to 0 and the V bit is set to 1. LRU is updated so that the replaced way becomes the latest. When the U bit of the entry to be replaced by updating the entry in write-back mode is 1, the cache update cycle starts after the entry is transferred to the write-back buffer. After the cache completes its update cycle, the write-back buffer writes the entry back to the memory. The write-back unit is 16 bytes.

7.3.3 Prefetch Operation

Prefetch Hit: LRU is updated so that the hit way becomes the latest. The contents in other caches are not modified. No instructions or data is transferred to the CPU.

Prefetch Miss: No instructions or data is transferred to the CPU. The way to be replaced follows table 7.4. Other operations are the same in case of read miss.

7.3.4 Write Access

Write Hit: In a write access in write-back mode, the data is written to the cache and no external memory write cycle is issued. The U bit of the entry written is set to 1 and LRU is updated so that the hit way becomes the latest. In write-through mode, the data is written to the cache and an external memory write cycle is issued. The U bit of the written entry is not updated and LRU is updated so that the replaced way becomes the latest.

Write Miss: In write-back mode, an external bus cycle starts when a write miss occurs, and the entry is updated. The way to be replaced follows table 7.5. When the U bit of the entry to be replaced is 1, the cache update cycle starts after the entry is transferred to the write-back buffer. The write-back unit is 16 bytes. Data is written to the cache and the U bit is set to 1. V bit is set to 1. LRU is updated so that the replaced way becomes the latest. After the cache completes its update cycle, the write-back buffer writes the entry back to the memory. In write-through mode, no write to cache occurs in a write miss; the write is only to the external memory.

7.3.5 Write-Back Buffer

When the U bit of the entry to be replaced in the write-back mode is 1, it must be written back to the external memory. To increase performance, the entry to be replaced is first transferred to the write-back buffer and fetching of new entries to the cache takes priority over writing back to the external memory. After the cache completes to fetch the new entry, the write-back buffer writes the entry back to external memory. During the write-back cycles, the cache can be accessed. The write-back buffer can hold one line of cache data (16 bytes) and its physical address. Figure 7.3 shows the configuration of the write-back buffer.

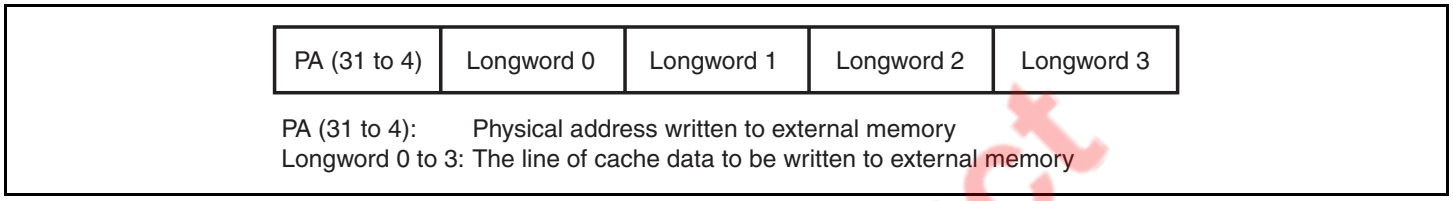


Figure 7.3 Write-Back Buffer Configuration

7.3.6 Coherency of Cache and External Memory

Use software to ensure coherency between the cache and the external memory. When memory shared by this LSI and another device is mapped in the address space to be cached, operate the memory mapped cache to invalidate and write back as required.

7.4 Memory-Mapped Cache

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

7.4.1 Address Array

The address array is mapped onto H'F0000000 to H'F0FFFFFF. To access an address array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the address, V bit, U bit, and LRU bits to be written to the address array (figure 7.4 (1)).

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

7.4.2 Data Array

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

Specify the entry address for selecting the entry (bits 11 to 4), L indicating the longword position within the (16-byte) line, W for selecting the way (bits 13 and 12), and H'F1 to indicate data array access (bits 31 to 24).

In L (bits 3 and 2), 00 is longword 0, 01 is longword 1, 10 is longword 2, and 11 is longword 3. In W (bits 13 and 12), 00 is way 0, 01 is way 1, 10 is way 2, and 11 is way 3. The access size of the data array is fixed at longword, so specify 00 for bits 1 and 0.

Following two operations are possible for the data array. Information in the address array is not modified by this operation.

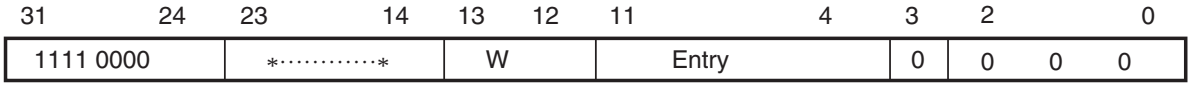
Data Array Read: The data specified by L (bits 3 and 2) in the address is read from the entry address specified by the address and the entry corresponding to the way.

Data Array Write: The longword data specified by the data is written to the position specified by L (bits 3 and 2) in the address from the entry address specified by the address and the entry corresponding to the way.

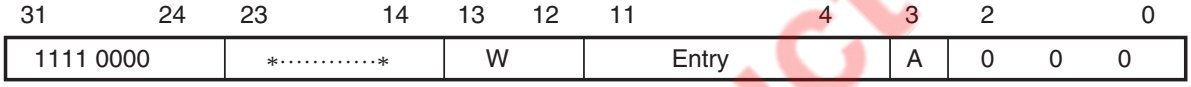
1. Address array access

(a) Address specification

Read access



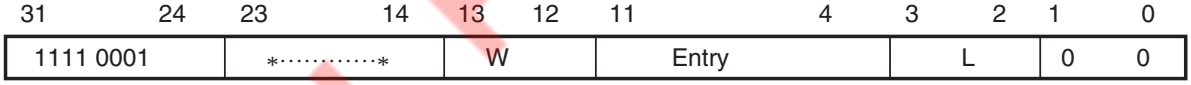
Write access



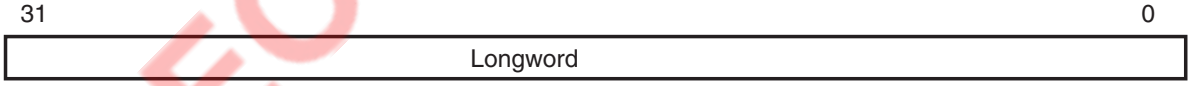
(b) Data specification (both read and write accesses)

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

(a) Address specification



(b) Data specification



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

Figure 7.4 Specifying Address and Data for Memory-Mapped Cache Access

7.4.3 Usage Examples

Invalidating Specific Entries

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

An example when a write data is specified in R0 and an address is specified in R1 is shown below.

```
; R0=H'0110 0000; tag address=B'0000 0001 0001 0000 0000 00, U=0, V=0
; R1=H'F000 0088; address array access, entry=B'00001000, A=1
;
 MOV.L R0,@R1
```

Reading the Data of a Specific Entry

The data section of a specific cache entry can be read by the memory mapping cache access. The longword indicated in the data field of the data array in figure 7.4 is read into the register.

An Example when an address is specified in R0 and data is read in R1.

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

Section 8 X/Y Memory

This LSI has on-chip X-RAM and Y-RAM. It can be used by CPU, DSP and DMAC to store instructions or data.

8.1 Features

The X/Y Memory features are listed in table 8.1.

Table 8.1 X/Y Memory Specifications

Parameter	Features
Addressing method	Mapping is possible in space P0 or P2
Ports	3 independent read/write ports • 8-/16-/32-bit access by the CPU (via L bus or I bus) • Maximum of two simultaneous 16-bit accesses (via X and Y buses), or 16/32-bit accesses, by the DSP (via L bus) • 8-/16-/32-bit access by the DMAC (via I bus)
Size	8-kbyte RAM for X and Y memory each

The X memory resides in addresses H'05007000 to H'05008FFF in space P0 or addresses H'A5007000 to H'A5008FFF (8 kbytes) in space P2. The X RAM is divided into page 0 and page 1 according to the addresses. The X memory can be accessed from the L bus, X bus, and I bus.

The Y memory resides in addresses H'05017000 to H'05018FFF in space P0 or addresses H'A5017000 to H'A5018FFF (8 kbytes) in space P2. The X RAM is divided into page 0 and page 1 according to the addresses. The Y memory can be accessed from the L bus, Y bus, and I bus.

In the event of simultaneous accesses to the same page from different buses, the priority order is: I bus > X bus > L bus in the X memory and I bus > Y bus > L bus in the Y memory. Since this kind of contention tends to lower X/Y memory accessibility, it is advisable to provide software measures to prevent such contention as far as possible. For example, contention will not arise if different memory or different pages are accessed by each bus.

X/Y memory is accessed by the CPU or DSP from space P0 via the I bus, a contention with the DMAC may occur on the I bus. Since this kind of contention also tends to lower X/Y memory accessibility, it is advisable to provide software measures to prevent such contention as far as possible. For example, contention on the I bus can be prevented by using space P2 when the X/Y memory is accessed by the CPU or DSP.

8.2 X/Y Memory Access from CPU

The X/Y memory can be accessed by the CPU from spaces P0 and P2. Access from space P0 uses the I bus, and access from space P2 use the L bus. To use the L bus, one cycle access is performed unless page conflict occurs. Using the I bus takes more than one cycle access. Figure 8.1 shows X/Y memory address mapping.

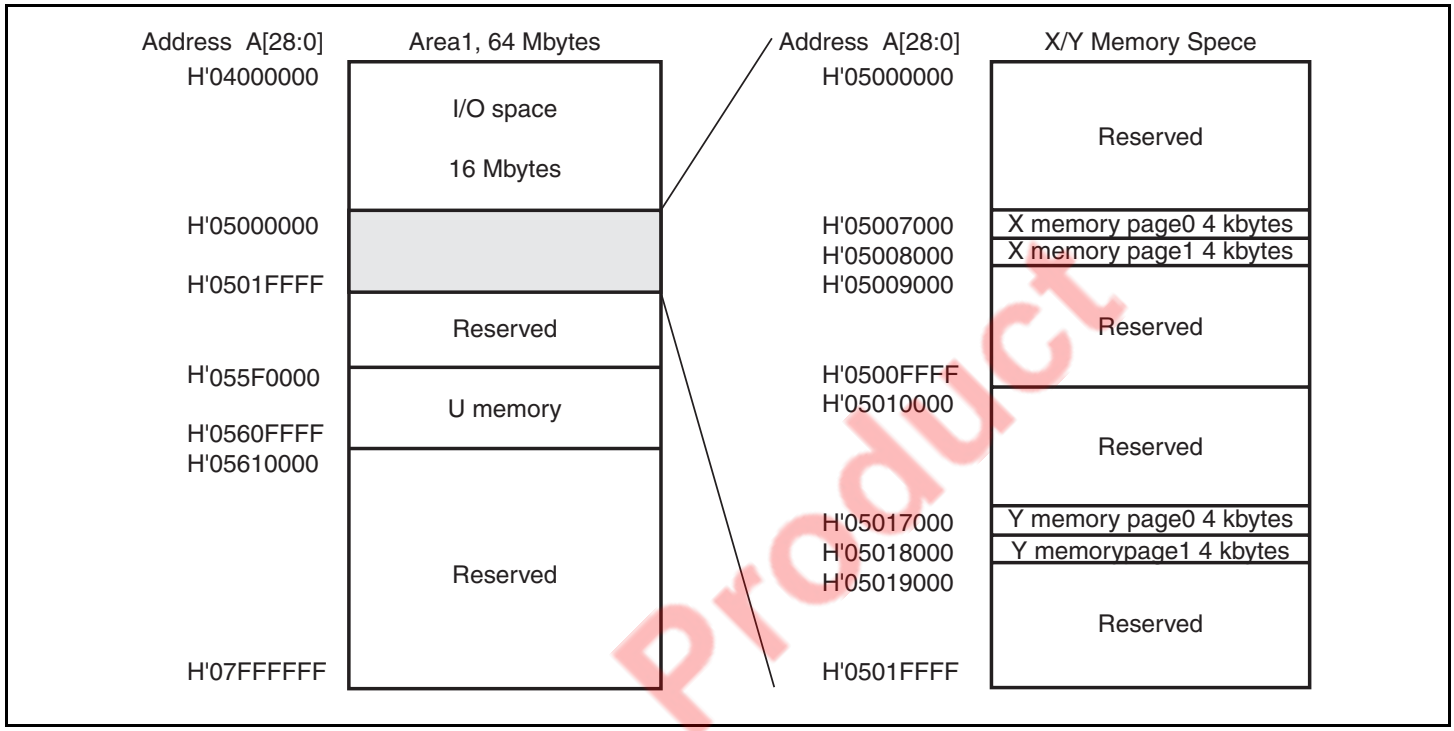


Figure 8.1 X/Y Memory Address Mapping

8.3 X/Y Memory Access from DSP

The X/Y memory can be accessed by the DSP from spaces P0 and P2. Methods for accessing differ according to instructions. Accesses via the X bus/Y bus are always 16-bit, while accesses via the L bus are either 16-bit or 32-bit. To use the L bus, one cycle access is performed unless page conflict occurs. Using the I bus takes more than one cycle access.

With X data transfer instructions and Y data transfer instructions, the X/Y memory is accessed via the X bus or Y bus. These accesses are always 16-bit. In the case of a single data transfer instruction, the X/Y memory is accessed via the L bus. In this case the access is either 16-bit or 32-bit.

Accesses via the X bus and Y bus can be specified simultaneously.

8.4 X/Y Memory Access from DMAC

The X/Y memory can be accessed by the DMAC via the I bus. Use the addresses between H'05007000 and H'05008FFF or H'05017000 and H'05018FFF.

8.5 Usage Note

When accessing the X/Y memory from the CPU and DSP, if the cache is on, access must be performed from space P2 (non-cacheable space). Operation during access from space P0 cannot be guaranteed. When the cache is off, spaces P0 and P2 can both be used. Specify the P2 area for parallel operation and double data transfer. (See section 3.1.9, Data Transfer Operation.)

8.6 Sleep Mode

In sleep mode, the X/Y memory is not accessed from the I bus master module such as DMAC.

8.7 Address Error

When an address error in write access to the X/Y memory occur, the contents of the X/Y memory may be corrupted.

Section 9 Exception Handling

Exception handling is separate from normal program processing, and is performed by a routine separate from the normal program. For example, if an attempt is made to execute an undefined instruction code or an instruction protected by the CPU processing mode, a control function may be required to return to the source program by executing the appropriate operation or to report an abnormality and carry out end processing. In addition, a function to control processing requested by LSI on-chip modules or an LSI external module to the CPU may also be required.

Transferring control to a user-defined exception processing routine and executing the process to support the above functions are called exception handling. This LSI has two types of exceptions: general exceptions and interrupts. The user can execute the required processing by assigning exception handling routines corresponding to the required exception processing and then return to the source program.

A reset input can terminate the normal program execution and pass control to the reset vector after register initialization. This reset operation can also be regarded as an exception handling. This section describes an overview of the exception handling operation. Here, general exceptions and interrupts are referred to as exception handling. For interrupts, this section describes only the process executed for interrupt requests. For details on how to generate an interrupt request, refer to section 10, Interrupt Controller (INTC).

9.1 Register Descriptions

There are three registers for exception handling. A register with an undefined initial value should be initialized by the software.

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

Figure 9.1 shows the bit configuration of each register.

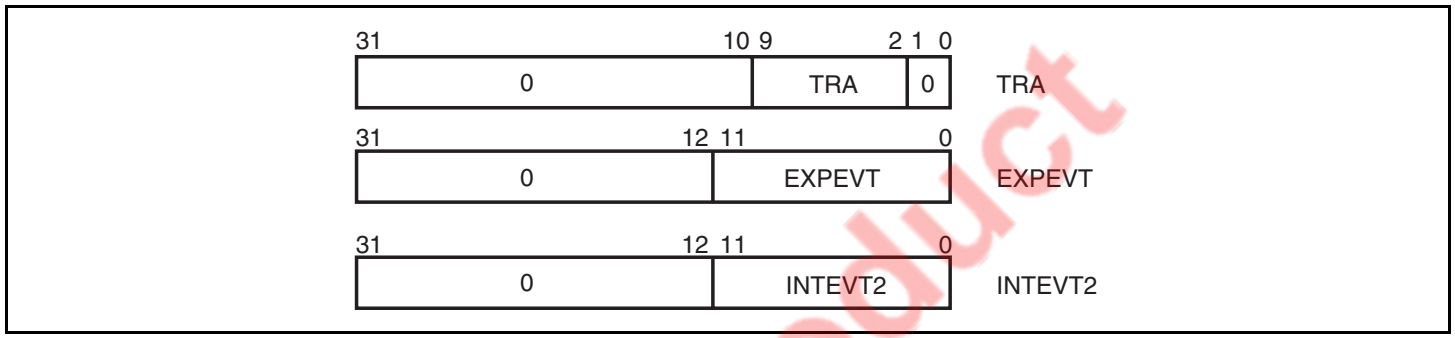


Figure 9.1 Register Bit Configuration

9.1.1 TRAPA Exception Register (TRA)

TRA is assigned to address H'FFFFFFD0 and consists of the 8-bit immediate data (imm) of the TRAPA instruction. TRA is automatically specified by the hardware when the TRAPA instruction is executed. Only bits 9 to 2 of the TRA can be re-written using the software.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 2	TRA	—	R/W	8-Bit Immediate Data
1, 0	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.

9.1.2 Exception Event Register (EXPEVT)

EXPEVT is assigned to address H'FFFFFFD4 and consists of a 12-bit exception code. Exception codes to be specified in EXPEVT are those for resets and general exceptions. These exception codes are automatically specified the hardware when an exception occurs. Only bits 11 to 0 of EXPEVT can be re-written using the software.

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

Note: * Initialized to H'000 at power-on reset and H'020 at manual reset.

9.1.3 Interrupt Event Register 2 (INTEVT2)

INTEVT2 is assigned to address H'A4000000 and consists of a 12-bit exception code. Exception codes to be specified in INTEVT2 are those for interrupt requests. These exception codes are automatically specified by the hardware when an exception occurs. INTEVT2 cannot be modified using the software.

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	INTEVT2	—	R/W	12-Bit Exception Code

Note: Initialized to H'000 at power-on reset and H'020 at manual reset.

9.2 Exception Handling Function

9.2.1 Exception Handling Flow

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

A basic exception handling sequence consists of the following operations. If an exception occurs and the CPU accepts it, operations 1 to 8 are executed.

1. The contents of PC is saved in SPC.
2. The contents of SR is saved in SSR.
3. The block (BL) bit in SR is set to 1, masking any subsequent exceptions.
4. The register bank (RB) bit in SR is set to 1.
5. An exception code identifying the exception event is written to bits 11 to 0 of the exception event (EXPEVT) or interrupt event (INTEVT2) register.
6. If a TRAPA instruction is executed, an 8-bit immediate data specified by the TRAPA instruction is set to TRA.
7. Instruction execution jumps to the designated exception vector address to invoke the handler routine.

The above operations from 1 to 7 are executed in sequence. During these operations, no other exceptions may be accepted unless multiple exception acceptance is enabled.

In an exception handling routine for a general exception, the appropriate exception handling must be executed based on an exception source determined by the EXPEVP. In an interrupt exception handling routine, the appropriate exception handling must be executed based on an exception source determined by the INTEVT2. After the exception handling routine has been completed, program execution can be resumed by executing an RTE instruction. The RTE instruction causes the following operations to be executed.

1. The contents of the SSR are restored into the SR to return to the processing state in effect before the exception handling took place.
2. A delay slot instruction of the RTE instruction is executed.
3. Control is passed to the address stored in the SPC.

The above operations from 1 to 3 are executed in sequence. During these operations, no other exceptions may be accepted. By changing the SPT and SSR before executing the RTE instruction, a status different from that in effect before the exception handling can also be specified.

Note: For details on the CPU processing mode in which RTE delay slot instructions are executed, please refer to section 9.6, Usage Notes.

9.2.2 Exception Vector Addresses

A vector address for general exceptions is determined by adding a vector offset to a vector base address. The vector offset for general exceptions other than the TLB error exception is H'00000100. The vector offset for interrupts is H'00000600. The vector base address is loaded into the vector base register (VBR) using the software.

9.2.3 Exception Codes

The exception codes are written to bits 11 to 0 of the EXPEVT register (for reset or general exceptions) or the INTEVT2 register (for interrupt requests) to identify each specific exception event. See section 10, Interrupt Controller (INTC), for details of the exception codes for interrupt requests. Table 9.1 lists exception codes for resets and general exceptions.

9.2.4 Exception Request and BL Bit (Multiple Exception Prevention)

The BL bit in SR is set to 1 when a reset or exception is accepted. While the BL bit is set to 1, acceptance of general exceptions is restricted as described below, making it possible to effectively prevent multiple exceptions from being accepted.

If the BL bit is set to 1, an interrupt request is not accepted and is retained. The interrupt request is accepted when the BL bit is cleared to 0. If the CPU is in low power consumption mode, an interrupt is accepted even if the BL bit is set to 1 and the CPU returns from the low power consumption mode.

A DMA error is not accepted and is retained if the BL bit is set to 1 and accepted when the BL bit is cleared to 0. User break requests generated while the BL bit is set are ignored and are not retained. Accordingly, user breaks are not accepted even if the BL bit is cleared to 0.

If a general exception other than a DMA address error or user break occurs while the BL bit is set to 1, the CPU enters a state similar to that in effect immediately after a reset, and passes control to the reset vector (H'A0000000) (multiple exception). In this case, unlike a normal reset, modules

other than the CPU are not initialized, the contents of EXPEVT, SPC, and SSR are undefined, and this status is not detected by an external device.

To enable acceptance of multiple exceptions, the contents of SPC and SSR must be saved while the BL bit is set to 1 after an exception has been accepted, and then the BL bit must be cleared to 0. Before restoring the SPC and SSR, the BL bit must be set to 1.

9.2.5 Exception Source Acceptance Timing and Priority

Exception Request of Instruction Synchronous Type and Instruction Asynchronous Type: Resets and interrupts are requested asynchronously regardless of the program flow. In general exceptions, a DMA address error and a user break under the specific condition are also requested asynchronously. The user cannot expect on which instruction an exception is requested. For general exceptions other than a DMA address error and a user break under a specific condition, each general exception corresponds to a specific instruction.

Re-execution Type and Processing-completion Type Exceptions: All exceptions are classified into two types: a re-execution type and a processing-completion type. If a re-execution type exception is accepted, the current instruction executed when the exception is accepted is terminated and the instruction address is saved to the SPC. After returning from the exception processing, program execution resumes from the instruction where the exception was accepted. In a processing-completion type exception, the current instruction executed when the exception is accepted is completed, the next instruction address is saved to the SPC, and then the exception processing is executed.

During a delayed branch instruction and delay slot, the following operations are executed. A re-execution type exception detected in a delay slot is accepted before executing the delayed branch instruction. A processing-completion type exception detected in a delayed branch instruction or a delay slot is accepted when the delayed branch instruction has been executed. In this case, the acceptance of delayed branch instruction or a delay slot precedes the execution of the branch destination instruction. In the above description, a delay slot indicates an instruction following an unconditional delayed branch instruction or an instruction following a conditional delayed branch instruction whose branch condition is satisfied. If a branch does not occur in a conditional delayed branch, the normal processing is executed.

Acceptance Priority and Test Priority: Acceptance priorities are determined for all exception requests. The priority of resets, general exceptions, and interrupts are determined in this order: a reset is always accepted regardless of the CPU status. Interrupts are accepted only when resets or general exceptions are not requested.

If multiple general exceptions occur simultaneously in the same instruction, the priority is determined as follows.

1. A processing-completion type exception generated at the previous instruction*
2. A user break before instruction execution (re-execution type)
3. An exception related to an instruction fetch (CPU address error: re-execution type)
4. An exception caused by an instruction decode (General illegal instruction exceptions and slot illegal instruction exceptions: re-execution type, unconditional trap: processing-completion type)
5. An exception related to data access (CPU address error: re-execution type)
6. Unconditional trap (processing-completion type)
7. A user break other than one before instruction execution (processing-completion type)
8. DMA address error

Note: If a processing-completion type exception is accepted at an instruction, exception processing starts before the next instruction is executed. This exception processing executed before an exception generated at the next instruction is detected.

Only one exception is accepted at a time. Accepting multiple exceptions sequentially results in all exception requests being processed.

Table 9.1 Exception Event Vectors

Exception Type	Current Instruction	Exception Event	Priority*[1]	Exception Order	Process at BL=1	Vector Code	Vector Offset
Reset	Aborted	Power-on reset	1	1	Reset	H'A00	—
		Manual reset	1	2	Reset	H'020	—
		H-UDI reset	1	1	Reset	H'000	—
General exception events	Re-executed	User break (before instruction execution)	2	0	Ignored	H'1E0	H'00000100
		CPU address error (instruction access) *[4]	2	1	Reset	H'0E0	H'00000100
		Illegal general instruction exception	2	2	Reset	H'180	H'00000100
		Illegal slot instruction exception	2	2	Reset	H'1A0	H'00000100
		CPU address error (data access)*[4]	2	3	Reset	H'0E0/ H'100	H'00000100
	Completed	Unconditional trap (TRAPA instruction)	2	4	Reset	H'160	H'00000100
		User breakpoint (After instruction execution, address)	2	5	Ignored	H'1E0	H'00000100
General exception events	Completed	User breakpoint (Data break, I-BUS break)	2	5	Ignored	H'1E0	H'00000100
		DMA address error	2	6	Retained	H'5C0	H'00000100
General interrupt requests	Completed	Interrupt requests	3	—*[2]	Retained	—*[3]	H'00000600

Notes: 1. Priorities are indicated from high to low, 1 being the highest and 3 the lowest.
A reset has the highest priority. An interrupt is accepted only when general exceptions are not requested.

2. For details on priorities in multiple interrupt sources, refer to section 10, Interrupt Controller (INTC).

3. If an interrupt is accepted, the exception event register (EXPEVT) is not changed. The interrupt source code is specified in interrupt source register 2 (INTEVT2). For details, refer to section 10, Interrupt Controller (INTC).

4. If one of these exceptions occurs in a specific part of the repeat loop, a specific code and vector offset are specified.

9.3 Individual Exception Operations

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

9.3.1 Resets

Power-On Reset:

- Conditions

  Power-on reset is request
- Operations

  Set EXPEVT to H'000, initialize the CPU and on-chip peripheral modules, and branch to the reset vector H'A0000000. For details, refer to the register descriptions in the relevant sections.

Manual Reset:

- Conditions

  Manual reset is request
- Operations

  Set EXPEVT to H'020, initialize the CPU and on-chip peripheral modules, and branch to the reset vector H'A0000000. For details, refer to the register descriptions in the relevant sections.

H-UDI Reset:

- Conditions

  The H-UDI reset command is entered (See section 15.4.4, H-UDI Reset.)
- Operations

  Set EXPEVT to H'000, initialize the VBR and SR, and branch to the PC H'A0000000.

  The VBR register is set to H'00000000 by initialization. For the SR, the BL and RB bits are set to 1 and the interrupt mask bits (I3 to I0) are set to 1111.

  Initialize the CPU and on-chip peripheral modules. For details, refer to the register descriptions in the relevant sections.

Table 9.2 Type of Reset

		Internal state	
Type	**Condition to reset**	**CPU**	**On-chip peripheral module**
Power-on reset	$\overline{\text{RESETP}}$ = Low level	Initialization	Refer to the register configurations in the relevant sections.
Manual reset	$\overline{\text{RESETM}}$ = Low level		
H-UDI reset	H-UDI reset command entry		

9.3.2 General Exceptions

CPU address error:

- Conditions
  - Instruction is fetched from odd address (4n + 1, 4n + 3)
  - Word data is accessed from addresses other than word boundaries (4n + 1, 4n + 3)
  - Long word is accessed from addresses other than longword boundaries (4n + 1, 4n + 2, 4n + 3)
  - The area ranging from H'80000000 to H'FFFFFFFF in logical space is accessed in user mode
- Types

  Instruction synchronous, re-execution type
- Save address

  Instruction fetch: An instruction address to be fetched when an exception occurred

  Data access: An instruction address where an exception occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)
- Exception code

  An exception occurred during read: H'0E0

  An exception occurred during write: H'1E0
- Remarks

  None

Illegal general instruction exception:

- Conditions
  - When undefined code not in a delay slot is decoded

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

Note: For details on undefined code, refer to SH-3/SH-3E/SH-3DSP Software Manual. When an undefined code other than H'FC00 to H'FFFF is decoded, operation cannot be guaranteed.

- Types

  Instruction synchronous, re-execution type
- Save address

  An instruction address where an exception occurs
- Exception code

  H'180
- Remarks

  None

Illegal slot instruction:

- Conditions
  - When undefined code in a delay slot is decoded

    Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
  - When an instruction that rewrites PC in a delay slot is decoded

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

  Instruction synchronous, re-execution type
- Save address

  A delayed branch instruction address
- Exception code

  H'1A0
- Remarks

  None

Unconditional trap:

- Conditions
  TRAPA instruction executed
- Types
  Instruction synchronous, processing-completion type
- Save address
  An address of an instruction following TRAPA
- Exception code
  H'160
- Remarks
  The exception is a processing-completion type, so PC of the instruction after the TRAPA instruction is saved to SPC. The 8-bit immediate value in the TRAPA instruction is quadrupled and set in TRA9 to TRA0.

User break point trap:

- Conditions
  When a break condition set in the user break controller is satisfied
- Types
  Break (L bus) before instruction execution: Instruction synchronous, re-execution type
  Operand break (L bus): Instruction synchronous, processing-completion type
  Data break (L bus): Instruction asynchronous, processing-completion type
  I bus break: Instruction asynchronous, processing-completion type
- Save address
  Re-execution type: An address of the instruction where a break occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)
  Operand break (L bus): An address of the instruction following the instruction where a break occurs (a delayed branch instruction destination address if an instruction is assigned to a delay slot)
  Data break (L bus): Instruction asynchronous, processing-completion type
- Exception code
  H'1E0
- Remarks
  For details on user break controller, refer to section 11, User Break Controller (UBC).

DMA address error:

- Conditions
  - Word data accessed from addresses other than word boundaries (4n + 1, 4n + 3)
  - Longword accessed from addresses other than longword boundaries (4n + 1, 4n + 2, 4n + 3)
- Types

  Instruction synchronous, processing-completion type
- Save address

  An address of the instruction following the instruction where a break occurs (a delayed branch instruction destination address if an instruction is assigned to a delay slot)
- Exception code

  H'5C0
- Remarks

  An exception occurs when a DMA transfer is executed while an illegal instruction address described above is specified in the DMAC. Since the DMA transfer is performed asynchronously with the CPU instruction operation, an exception is also requested asynchronously with the instruction execution. For details on DMAC, refer to section 13, Direct Memory Access Controller (DMAC).

9.4 Exception Processing While DSP Extension Function is Valid

When the DSP extension function is valid (the DSP bit of SR is set to 1), some exception processing acceptance conditions or exception processing may be changed.

9.4.1 Illegal Instruction Exception and Slot Illegal Instruction Exception

In the DSP mode, a DSP extension instruction can be executed. If a DSP extension instruction is executed when the DSP bit of SR is cleared to 0 (in a mode other than the DSP mode), an illegal instruction exception occurs.

9.4.2 Exception in Repeat Control Period

If an exception is requested or an exception is accepted during repeat control, the exception may not be accepted correctly or a program execution may not be returned correctly from exception processing that is different from the normal state. These restrictions may occur from repeat detection instruction to repeat end instruction while the repeat counter is 1 or more. In this section, this period is called the repeat control period.

The following shows program examples where the number of instructions in the repeat loop are 4 or more, 3, 2, and 1, respectively. In this section, a repeat detection instruction and its instruction address are described as RPTDTCT. The first, second, and third instructions following the repeat detection instruction are described as RptDtct1, RptDtct2, and RptDtct2. In addition, [A], [B], [C1], and [C2] in the following examples indicate instructions where a restriction occurs. Table 9.3 summarizes the instruction positions and restriction types.

Table 9.3 Instruction Positions and Restriction Types

Instruction Position	SPC*[1]	Illegal Instruction*[2]	Interrupt, Break*[3]	CPU Address Error*[4]
[A]				
[B]			Retained	
[C1]		Added	Retained	Instruction/data
[C2]	Illegal	Added	Retained	Instruction/data

Notes: 1. A specific address is specified in the SPC if an exception occurs while SR.RC[11:0] ≥2.
2. There are a greater number of instructions that can be illegal instructions while SR.RC[11:0] ≥1.
3. An interrupt break or DMA address error request is retained while SR.RC[11:0] ≥1.
4. A specific exception code is specified while SR.RC[11:0] ≥1.

- Example 1: Repeat loop consisting of four instructions

```
            LDRS RptStart     ; [A]
            LDRS RptDtct + 4  ; [A]
            SETRCT #4         ; [A]
            instr0            ; [A]
RptStart:   instr1            ; [A]
            ……                ; [A]
            ……                ; [A]
RptDtct:    RptDtct           ; [B] A repeat detection instruction is an
                                    instruction three instructions before
                                    a repeat end instruction
            RptDtct1          ; [C1]
            RptDtct2          ; [C2]
RptEnd:     RptDtct3          ; [C2][Repeat end instruction]
            InstrNext         ; [A]
```

- Example 2: Repeat loop consisting of three instructions

```
LDRS        RptDtct + 4       ; [A]
LDRS        RptDtct + 4       ; [A]
            SETRCT #4         ; [A]
RptDtct:    RptDtct           ; [B] A repeat detection instruction is an
                                    instruction prior to a repeat start
                                    instruction
RptStart:   RptDtct1          ; [C1][Repeat start instruction]
            RptDtct2          ; [C2]
RptEnd:     RptDtct3          ; [C2][Repeat end instruction]
            InstrNext         ; [A]
```

- Example 3: Repeat loop consisting of two instructions

```
LDRS      RptDtct + 6      ; [A]
LDRS      RptDtct + 4      ; [A]
          SETRCT #4        ; [A]
RptDtct: RptDtct           ; [B] A repeat detection instruction is an
                                 instruction prior to a repeat start
                                 instruction
RptStart:   RptDtct1       ; [C1][Repeat start instruction]
RptEnd:  RptDtct3          ; [C2][Repeat end instruction]
          InstrNext        ; [A]
```

- Example 4: Repeat loop consisting of one instruction

```
LDRS      RptDtct + 8      ; [A]
LDRS      RptDtct + 4      ; [A]
          SETRCT #4        ; [A]
RptDtct: RptDtct           ; [B] A repeat detection instruction is an
                                 instruction prior to a repeat start
                                 instruction
RptStart:
RptEnd:  RptDtct1          ; [C1][Repeat start instruction]== [Repeat end
                                  instruction]
               InstrNext   ; [A]
```

SPC Saved by an Exception in Repeat Control Period: If an exception is accepted in the repeat control period while the repeat counter (RC11 to RC0) in the SR register is two or greater, the program counter to be saved may not indicate the value to be returned correctly. To execute the repeat control after returning from an exception processing, the return address must indicate an instruction prior to a repeat detection instruction. Accordingly, if an exception is accepted in repeat control period, an exception other than re-execution type exception by a repeat detection instruction cannot return to the repeat control correctly.

Table 9.4 SPC Value When a Re-Execution Type Exception Occurs in Repeat Control

Instruction Where an Exception Occurs	Number of Instructions in a Repeat Loop			
	1	2	3	4 or Greater
RptDtct	RptDtct	RptDtct	RptDtct	RptDtct
RptDtct1	RptDtct1	RptDtct1	RptDtct1	RptDtct1
RptDtct2	—	RptDtct1	RptDtct1	RS-4
RptDtct3	—	—	RptDtct1	RS-2

Note: The following labels are used here.
RptDtct: Repeat detection instruction address
RptDtct1: An instruction address one instruction following the repeat detection instruction
RptDtct2: An instruction address two instruction following the repeat detection instruction
RptDtct3: An instruction address three instruction following the repeat detection instruction
RS: Repeat start instruction address
If a re-execution type exception is accepted at an instruction in the hatched areas above, a return address to be saved in the SPC is incorrect. If SR.RC[11:0] is 1 or 0, a correct return address is saved in the SPC.

Illegal Instruction Exception in Repeat Control Period: If one of the following instructions is executed at the address following RptDtct1, a general illegal instruction exception occurs. For details on an address to be saved in the SPC, refer to SPC Saved by an Exception in Repeat Control Period description in section 9.4.2, Exception in Repeat Control Period.

- Branch instructions
  BRA, BSR, BT, BF, BT/S, BF/S, BSRF, RTS, BRAF, RTE, JSR, JMP, TRAPA
- Repeat control instructions
  SETRC, LDRS, LDRE
- Load instructions for SR, RS, and RE
  LDC Rn,SR, LDC @Rn+,SR, LDC Rn,RE, LDC @Rn+,RE, LDC Rn,RS, LDC @Rn+, Rs

Note: In a repeat loop consisting of one to three instructions, some restrictions apply to repeat detection instructions and all the remaining instructions. In a repeat loop consisting of four or more instructions, restrictions apply to only the three instructions that include a repeat end instruction.

An Exception Retained in Repeat Control Period: In the repeat control period, an interrupt or some exception will be retained to prevent an exception acceptance at an instruction where returning from the exception cannot be performed correctly. For details, refer to repeat loop program example 1 to 4. In the examples, exceptions generated at instructions indicated as [B], [C], [C1], or [C2], the following processing is executed.

- Interrupt, DMA address error

  An exception request is not accepted and retained at instructions [B] and [C]. If an instruction indicates as [A] is executed the next time, an exception request is accepted.* As shown in example 1 to 4, any interrupt or DMA address error cannot be accepted in a repeat loop consisting of four instructions or less.

Note: * An interrupt request or a DMA address error exception request is retained in the interrupt controller (INTC) and the direct memory access controller (DMAC) until the CPU can accept a request.

- User break before instruction execution

  A user break before instruction execution is accepted at instruction [B], and an address of instruction [B] is saved in the SPC. This exception cannot be accepted at instruction [C] but the exception request is retained until an instruction [A] or [B] is executed the next time. Then, the exception request is accepted before an instruction [A] or [B] is executed. In this case, an address of instruction [A] or [B] is saved in the SPC.

- User break after instruction execution

  A user break after instruction execution cannot be accepted at instructions [B] and [C] but the exception request is retained until an instruction [A] or [B] is executed the next time. Then, the exception request is accepted before an instruction [A] or [B] is executed. In this case, an address of instruction [A] or [B] is saved in the SPC.

Table 9.5 Exception Acceptance in the Repeat Loop

Exception Type	Instruction [B]	Instruction [C]
Interrupt	Not accepted	Not accepted
DMA address error	Not accepted	Not accepted
User break before instruction execution	Accepted	Not accepted
User break after instruction execution	Not accepted	Not accepted

CPU Address Error in Repeat Control Period: If a CPU address error occurs in the repeat control period, the exception is accepted but an exception code (H'070) indicating the repeat loop period is specified in the EXPEVT. If a CPU address error occurs in instructions following a repeat detection instruction to repeat end instruction, an exception code for instruction access or data access is specified in the EXPEVT.

The SPC is saved according to the SPC Saved by an Exception in Repeat Control Period description in section 9.4.2, Exception in Repeat Control Period.

After the CPU address error exception processing, the repeat control cannot be returned correctly. To execute a repeat loop correctly, care must be taken not to generate a CPU address error in the repeat control period.

Note: In a repeat loop consisting of one to three instructions, some restrictions apply to repeat detection instructions and all the remaining instructions. In a repeat loop consisting of four or more instructions, restrictions apply to only the three instructions that include a repeat end instruction. The restriction occurs when SR.RC[11:0] ≥ 1.

Table 9.6 Instruction Where a Specific Exception Occurs When a Memory Access Exception Occurs in Repeat Control

Instruction Where an Exception Occurs	Number of Instructions in a Repeat Loop			
	1	2	3	4 or Greater
RptDtct				
RptDtct1	Instruction/data access	Instruction/data access	Instruction/data access	Instruction/data access
RptDtct2	—	Instruction/data access	Instruction/data access	Instruction/data access
RptDtct3	—	—	Instruction/data access	Instruction/data access

Note: The following labels are used here.
RptDtct: Repeat detection instruction address
RptDtct1: An instruction address one instruction following the repeat detection instruction
RptDtct2: An instruction address two instruction following the repeat detection instruction
RptDtct3: An instruction address three instruction following the repeat detection instruction

9.5 Note on Initializing this LSI

This LSI needs to be initialized by a software reset before the power is turned on. Execute the following program immediately after a power-on reset.

Note that the following program overwrites contents of CPU general registers. Save contents of registers which should not be overwritten before executing the following program.

```
;-------------------------------------------------------------
; Intialization of sh7641 for power-on reset
;-------------------------------------------------------------
; ATTENTION:
;  1. Please execute below instructions on power-on reset.
;  2. This routine would overwrite the general registers on the CPU.
;  3. Do not modify these codes.
;-------------------------------------------------------------
          MOV.L     #H'A5007000,R4;
          MOV.L     #H'A5008000,R5;
          MOV.L     #H'A5017000,R6;
          MOV.L     #H'A5018000,R7;
          MOV.L     @R4,R0;
          MOV.L     @R5,R0;
          MOV.L     @R6,R0;
          MOV.L     @R7,R0;
;
          MOV.W     #H'FF40,R10;
          MOV.L     #H'A4FC0000,R8;
          MOV       #H'10,R9;
          MOV.B     R10,@R10;
          MOV.B     R10,@R10;
          MOV.B     R10,@R10;
          MOV.L     R9,@R8;
;
          MOV.L     #H'FC000000,R1;
          MOV.W     @R1,R0;
;
          MOV       #H'00,R9;
          MOV.B     R10,@R10;
          MOV.B     R10,@R10;
          MOV.B     R10,@R10;
          MOV.L     R9,@R8;
;-------------------------------------------------------------
```

9.6 Usage Notes

1. An instruction assigned at a delay slot of the RTE instruction is executed after the contents of the SSR is restored into the SR. An acceptance of an exception related to instruction access is determined according to the SR before restore. An acceptance of other exceptions is determined by the SR after restore, processing mode, and BL bit value. A processing-completion type exception is accepted before an instruction at the RTE branch destination address is executed. However, note that the correct operation cannot be guaranteed if a re-execution type exception occurs.
2. In an instruction assigned at a delay slot of the RTE instruction, a user break cannot be accepted.
3. If the BL bit of the SR register is changed by the LDC instruction, an exception is accepted according to the changed SR value from the next instruction.* A processing-completion type exception is accepted before the next instruction is executed. An interrupt and DMA address error in re-execution type exceptions are accepted before the next instruction is executed.

Note: * If an LDC instruction is executed for the SR, the following instructions are re-fetched and an instruction fetch exception is accepted according to the modified SR value.

Section 10 Interrupt Controller (INTC)

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The INTC registers set the order of priority of each interrupt, allowing the user to process interrupt requests according to the user-set priority.

10.1 Features

The INTC has the following features:

- 16 levels of interrupt priority can be set

  By setting the ten interrupt-priority registers, the priorities of on-chip peripheral modules, and IRQ interrupts can be selected from 16 levels for individual request sources.
- NMI noise canceler function

  An NMI input-level bit indicates the NMI pin state. By reading this bit in the interrupt exception service routine, the pin state can be checked, enabling it to be used as a noise canceler.
- IRQ interrupts can be set

  Detection of low level, rising edge, falling edge, or high level
- Interrupts can be enabled or disabled

  Interrupts can be enabled or disabled individually for each interrupt source with the interrupt mask registers (IMR) and interrupt mask clear registers (IMCR).

Figure 10.1 shows a block diagram of the INTC.

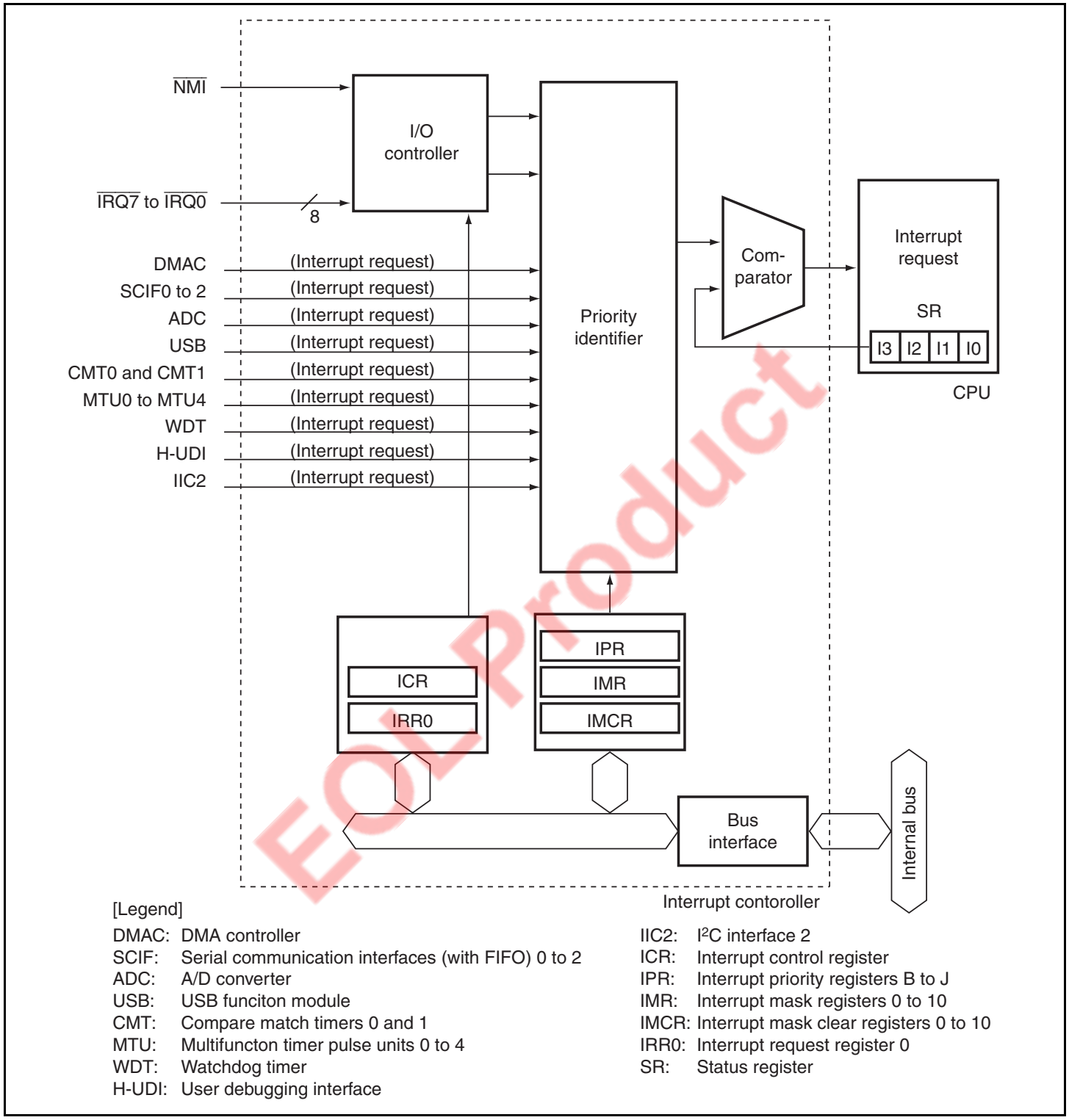


Figure 10.1 Block Diagram of INTC

10.2 Input/Output Pins

Table 10.1 shows the INTC pin configuration.

Table 10.1 Pin Configuration

Name	Abbreviation	I/O	Description
Nonmaskable interrupt input pin	$\overline{\text{NMI}}$	Input	Input of interrupt request signal, not maskable by the interrupt mask bits in SR
Interrupt input pins	$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	Input	Input of interrupt request signals, maskable by the interrupt mask bits in SR

10.3 Register Descriptions

The INTC has the following registers. For details on register addresses and register states during each processing, refer to section 24, List of Registers.

- Interrupt control register 0 (ICR0)
- Interrupt control register 1 (ICR1)
- Interrupt control register 3 (ICR3)
- 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 request register 0 (IRR0)
- Interrupt mask register 0 (IMR0)
- Interrupt mask register 1 (IMR1)
- Interrupt mask register 2 (IMR2)
- Interrupt mask register 3 (IMR3)
- Interrupt mask register 4 (IMR4)
- Interrupt mask register 5 (IMR5)

- Interrupt mask register 6 (IMR6)
- Interrupt mask register 7 (IMR7)
- Interrupt mask register 8 (IMR8)
- Interrupt mask register 9 (IMR9)
- Interrupt mask register 10 (IMR10)
- Interrupt mask clear register 0 (IMCR0)
- Interrupt mask clear register 1 (IMCR1)
- Interrupt mask clear register 2 (IMCR2)
- Interrupt mask clear register 3 (IMCR3)
- Interrupt mask clear register 4 (IMCR4)
- Interrupt mask clear register 5 (IMCR5)
- Interrupt mask clear register 6 (IMCR6)
- Interrupt mask clear register 7 (IMCR7)
- Interrupt mask clear register 8 (IMCR8)
- Interrupt mask clear register 9 (IMCR9)
- Interrupt mask clear register 10 (IMCR10)

10.3.1 Interrupt Priority Registers B to J (IPRB to IPRJ)

IPRB to IPRJ are 16-bit readable/writable registers in which priority levels from 0 to 15 are set for on-chip peripheral module and IRQ interrupts. These registers are initialized to H'0000 by a power-on reset or manual reset, but are not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	IPR15	0	R/W	These bits set the priority level for each interrupt source in 4-bit units. For details, see table 10.2, Interrupt Sources and IPRB to IPRJ.
14	IPR14	0	R/W	
13	IPR13	0	R/W	
12	IPR12	0	R/W	
11	IPR11	0	R/W	
10	IPR10	0	R/W	
9	IPR9	0	R/W	
8	IPR8	0	R/W	
7	IPR7	0	R/W	
6	IPR6	0	R/W	
5	IPR5	0	R/W	
4	IPR4	0	R/W	
3	IPR3	0	R/W	
2	IPR2	0	R/W	
1	IPR1	0	R/W	
0	IPR0	0	R/W	

Table 10.2 Interrupt Sources and IPRB to IPRJ

Register	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
IPRB	WDT	Reserved*	Reserved*	Reserved*
IPRC	IRQ3	IRQ2	IRQ1	IRQ0
IPRD	IRQ7	IRQ6	IRQ5	IRQ4
IPRE	Reserved*	SCIF0	SCIF1	ADC0
IPRF	ADC1	SCIF2	USB	CMT
IPRG	MTU0 (A/B/C/D)	MTU0 (V)	MTU1 (A/B)	MTU1 (V/U)
IPRH	MTU2 (A/B)	MTU2 (V/U)	MTU3 (A/B/C/D)	MTU3 (V)
IPRI	MTU4 (A/B/C/D)	MTU4 (V)	POE	IIC2
IPRJ	DMAC0	DMAC1	DMAC2	DMAC3

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

As shown in table 10.2, on-chip peripheral module or IRQ interrupts are assigned to four 4-bit groups in each register. These 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, and bits 3 to 0) are set with values from H'0 (0000) to H'F (1111). Setting of H'0 means priority level 0 (masking is requested); H'F means priority level 15 (the highest level).

10.3.2 Interrupt Control Register 0 (ICR0)

ICR0 is a register that sets the input signal detection mode of external interrupt input pin $\overline{\text{NMI}}$, and indicates the input signal level at the $\overline{\text{NMI}}$ pin. This register is initialized to H'0000 or H'8000 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	0/1*	R	NMI Input Level Sets the level of the signal input at the NMI pin. This bit can be read from to determine the NMI pin level. This bit cannot be modified. 0: NMI input level is low 1: NMI input level is high
14 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	NMIE	0	R/W	NMI Edge Select Selects whether the falling or rising edge of the interrupt request signal on the $\overline{\text{NMI}}$ pin is detected. 0: Interrupt request is detected on falling edge of $\overline{\text{NMI}}$ input 1: Interrupt request is detected on rising edge of $\overline{\text{NMI}}$ input
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * 1 when $\overline{\text{NMI}}$ input is high, 0 when $\overline{\text{NMI}}$ input is low.

10.3.3 Interrupt Control Register 1 (ICR1)

ICR1 is a 16-bit register that specifies the detection mode for external interrupt input pins $\overline{IRQ5}$ to $\overline{IRQ0}$ individually: rising edge, falling edge, high level, or low level. This register is initialized to H'4000 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	IRQE*	1	R/W	Interrupt Request Enable Enables or disables the use of pins $\overline{IRQ7}$ to $\overline{IRQ0}$ as eight independent interrupt pins. 0: Use of pins $\overline{IRQ7}$ to $\overline{IRQ0}$ as eight independent interrupt pins enabled* 1: Use of pins $\overline{IRQ7}$ to $\overline{IRQ0}$ as interrupt pins disabled
13, 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	IRQ51S	0	R/W	IRQn Sense Select These bits select whether interrupt request signals corresponding to pins $\overline{IRQ5}$ to $\overline{IRQ0}$ are detected by a rising edge, falling edge, high level, or low level. Bit 2n+1 IRQn1S, Bit 2n IRQn0S 0 0 : Interrupt request is detected on falling edge of $\overline{IRQn}$ input 0 1 : Interrupt request is detected on rising edge of $\overline{IRQn}$ input 1 0 : Interrupt request is detected on low level of $\overline{IRQn}$ input 1 1 : Interrupt request is detected on high level of $\overline{IRQn}$ input n = 0 to 5
10	IRQ50S	0	R/W	
9	IRQ41S	0	R/W	
8	IRQ40S	0	R/W	
7	IRQ31S	0	R/W	
6	IRQ30S	0	R/W	
5	IRQ21S	0	R/W	
4	IRQ20S	0	R/W	
3	IRQ11S	0	R/W	
2	IRQ10S	0	R/W	
1	IRQ01S	0	R/W	
0	IRQ00S	0	R/W	

Note: * The IRQE bit must be cleared to 0 in the initialization routine after a reset, and must then not be changed.

10.3.4 Interrupt Control Register 3 (ICR3)

ICR3 is a 16-bit register that specifies the detection mode for external interrupt input pins $\overline{\text{IRQ7}}$ and $\overline{\text{IRQ6}}$ individually: rising edge, falling edge, high level, or low level. This register is initialized to H'0000 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	IRQ71S	0	R/W	IRQn Sense Select These bits select whether interrupt request signals corresponding to pins $\overline{\text{IRQ7}}$ and $\overline{\text{IRQ6}}$ are detected by a rising edge, falling edge, high level, or low level. Bit 2n+1 Bit 2n IRQn1S IRQn0S 0 0 : Interrupt request is detected at the falling edge of $\overline{\text{IRQn}}$ input 0 1 : Interrupt request is detected at the rising edge of $\overline{\text{IRQn}}$ input 1 0 : Interrupt request is detected on low level of $\overline{\text{IRQn}}$ input 1 1 : Interrupt request is detected on high level of $\overline{\text{IRQn}}$ input n = 6 and 7
2	IRQ70S	0	R/W	
1	IRQ61S	0	R/W	
0	IRQ60S	0	R/W	

10.3.5 Interrupt Request Register 0 (IRR0)

IRR0 is an 8-bit register that indicates interrupt requests from external input pins $\overline{IRQ7}$ to $\overline{IRQ0}$. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7R	0	R/W	IRQn Interrupt Request
6	IRQ6R	0	R/W	Indicates whether there is interrupt request input to the $\overline{IRQn}$ pin. When edge-detection mode is set for $\overline{IRQn}$, an interrupt request is cleared by writing 0 to the IRQnR bit after reading IRQnR = 1.
5	IRQ5R	0	R/W	
4	IRQ4R	0	R/W	
3	IRQ3R	0	R/W	When level-detection mode is set for $\overline{IRQn}$, an interrupt request is set/cleared by only 1/0 input to the IRQn pin.
2	IRQ2R	0	R/W	
1	IRQ1R	0	R/W	
0	IRQ0R	0	R/W	IRQnR 0: No interrupt request input to $\overline{IRQn}$ pin 1: Interrupt request input to $\overline{IRQn}$ pin n = 0 to 7

10.3.6 Interrupt Mask Registers 0 to 10 (IMR0 to IMR10)

IMR0 to IMR10 are 8-bit readable/writable registers that mask the $\overline{\text{IRQ}}$ and on-chip peripheral module interrupts. When an interrupt source is masked, interrupt requests may be mistakenly detected, depending on the operation state of the IRQ pins and on-chip peripheral modules. To prevent this, set IMR0 to IMR9 while no interrupts are set to be generated, and then read the new settings from these registers.

Table 10.3 shows the relationship between IMR and each interrupt source.

Bit	Bit Name	Initial Value	R/W	Description
7	IM7	0	R/W	Interrupt Mask
6	IM6	0	R/W	Table 10.3 lists the correspondence between the interrupt sources and interrupt mask registers.
5	IM5	0	R/W	
4	IM4	0	R/W	IMn
3	IM3	0	R/W	1: Interrupt source of the corresponding bit is masked.
2	IM2	0	R/W	0: When reading, Interrupt source of the corresponding bit is not masked. When writing, No processing.
1	IM1	0	R/W	n = 7 to 0
0	IM0	0	R/W	

Table 10.3 Correspondence between Interrupt Sources and IMR0 to IMR10

Register Name	Bit Name (Function Name)							
	7	6	5	4	3	2	1	0
IMR0	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
	(IRQ)	(IRQ)	(IRQ)	(IRQ)	(IRQ)	(IRQ)	(IRQ)	(IRQ)
IMR1	TxI0	BRI0	RxI0	ERI0	DEI3	DEI2	DEI1	DEI0
	(SCIF0)	(SCIF0)	(SCIF0)	(SCIF0)	(DMAC)	(DMAC)	(DMAC)	(DMAC)
IMR2	—	—	—	ADI0	TxI1	BRI1	RxI1	ERI1
	(ADC0)	(ADC0)	(ADC0)	(ADC0)	(SCIF1)	(SCIF1)	(SCIF1)	(SCIF1)
IMR4	—	—	—	—	ITI	—	—	—
	—	—	—	—	WDT	WDT	WDT	WDT
IMR5	TxI2	BRI2	RxI2	ERI2	ADI1	USIHP	USI1	USI0
	(SCIF2)	(SCIF2)	(SCIF2)	(SCIF2)	(ADC1)	(USB)	(USB)	(USB)
IMR6	TCI2U	TCI2V	TGI2B	TGI2A	TCI1U	TCI1V	TGI1B	TGI1A
	(MTU2)	(MTU2)	(MTU2)	(MTU2)	(MTU1)	(MTU1)	(MTU1)	(MTU1)
IMR7	—	—	—	TCI0V	TGI0D	TGI0C	TGI0B	TGI0A
	(MTU0)	(MTU0)	(MTU0)	(MTU0)	(MTU0)	(MTU0)	(MTU0)	(MTU0)
IMR8	—	—	—	TCI3V	TGI3D	TGI3C	TGI3B	TGI3A
	(MTU3)	(MTU3)	(MTU3)	(MTU3)	(MTU3)	(MTU3)	(MTU3)	(MTU3)
IMR9	—	—	—	TCI4V	TGI4D	TGI4C	TGI4B	TGI4A
	(MTU4)	(MTU4)	(MTU4)	(MTU4)	(MTU4)	(MTU4)	(MTU4)	(MTU4)
IMR10	—	—	CMI1	CMI0	IIC2I	—	—	OEI
	(CMT)	(CMT)	(CMT)	(CMT)	(IIC2)	(IIC2)	(POE)	(POE)

Note: —: Reserved: The read value is not guaranteed.

10.3.7 Interrupt Mask Clear Registers 0 to 10 (IMCR0 to IMCR10)

IMCR0 to IMCR10 are 8-bit writable registers that clear the mask settings for the $\overline{\text{IRQ}}$ and on-chip peripheral module interrupts. Table 10.4 shows the relationship between IMCR and each interrupt source.

Bit	Bit Name	Initial Value	R/W	Description
7	IMC7	—	W	Interrupt Mask Clear
6	IMC6	—	W	Table 10.4 lists the correspondence between the interrupt sources and interrupt mask clear registers.
5	IMC5	—	W	
4	IMC4	—	W	IMCn (Write)
3	IMC3	—	W	1: The corresponding bit in interrupt mask register IMCn is cleared
2	IMC2	—	W	0: No processing
1	IMC1	—	W	n = 7 to 0
0	IMC0	—	W	

Table 10.4 Correspondence between Interrupt Sources and IMCR0 to IMCR10

Register Name	Bit Name (Function Name)							
	7	6	5	4	3	2	1	0
IMCR0	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
	(IRQ)	(IRQ)	(IRQ)	(IRQ)	(IRQ)	(IRQ)	(IRQ)	(IRQ)
IMCR1	TxI0	BRI0	RxI0	ERI0	DEI3	DEI2	DEI1	DEI0
	(SCIF0)	(SCIF0)	(SCIF0)	(SCIF0)	(DMAC)	(DMAC)	(DMAC)	(DMAC)
IMCR2	—	—	—	ADI0	TxI1	BRI1	RxI1	ERI1
	(ADC0)	(ADC0)	(ADC0)	(ADC0)	(SCIF1)	(SCIF1)	(SCIF1)	(SCIF1)
IMCR4	—	—	—	—	ITI	—	—	—
	—	—	—	—	(WDT)	(WDT)	(WDT)	(WDT)
IMCR5	TxI2	BRI2	RxI2	ERI2	ADI1	USIHP	USI1	USI0
	(SCIF2)	(SCIF2)	(SCIF2)	(SCIF2)	(ADC1)	(USB)	(USB)	(USB)
IMCR6	TCI2U	TCI2V	TGI2B	TGI2A	TCI1U	TCI1V	TGI1B	TGI1A
	(MTU2)	(MTU2)	(MTU2)	(MTU2)	(MTU1)	(MTU1)	(MTU1)	(MTU1)
IMCR7	—	—	—	TCI0V	TGI0D	TGI0C	TGI0B	TGI0A
	(MTU0)	(MTU0)	(MTU0)	(MTU0)	(MTU0)	(MTU0)	(MTU0)	(MTU0)
IMCR8	—	—	—	TCI3V	TGI3D	TGI3C	TGI3B	TGI3A
	(MTU3)	(MTU3)	(MTU3)	(MTU3)	(MTU3)	(MTU3)	(MTU3)	(MTU3)
IMCR9	—	—	—	TCI4V	TGI4D	TGI4C	TGI4B	TGI4A
	(MTU4)	(MTU4)	(MTU4)	(MTU4)	(MTU4)	(MTU4)	(MTU4)	(MTU4)
IMCR10	—	—	CMI1	CMI0	IIC2I	—	—	OEI
	(CMT)	(CMT)	(CMT)	(CMT)	(IIC2)	(IIC2)	(POE)	(POE)

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

10.4 Interrupt Sources

There are four types of interrupt sources: NMI, H-UDI, IRQ, and on-chip peripheral modules. Each interrupt has a priority level (0 to 16), with 1 the lowest and 16 the highest. Priority level 0 masks an interrupt, so the interrupt request is ignored.

10.4.1 NMI Interrupt

The NMI interrupt has the highest priority level of 16. When the BL bit in the status register (SR) is 0, NMI interrupts are accepted. NMI interrupts are edge-detected. In sleep or standby mode, the interrupt is accepted regardless of the BL setting. The NMI edge select bit (NMIE) in the interrupt control register 0 (ICR0) is used to select either rising or falling edge detection.

When using edge-input detection for NMI interrupts, a pulse width of at least two Pϕ cycles (peripheral clock) is necessary. NMI interrupt exception handling does not affect the interrupt mask level bits (I3 to I0) in the status register (SR).

It is possible to wake the chip up from sleep mode or standby mode with an NMI interrupt.

10.4.2 H-UDI Interrupt

The H-UDI interrupt is accepted between one instruction and another when the H-UDI interrupt command (section 15.4.5, H-UDI Interrupt.) is entered, the SR interrupt mask bit is set to the value smaller than 15, and the BL bit in SR is set to 0.

The H-UDI interrupt allows the PC to be saved to the SPC immediately after accepting the interrupt instruction. The SR at the time of the interrupt acceptation is saved to the SSR. The INTEVT2 is set to H'5E0. The BL and RB bits in SR are set to 1 and branched to VBR + H'0600.

10.4.3 IRQ Interrupts

IRQ interrupts are input by level or edge from pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$. The priority can be set by interrupt priority registers C and D (IPRC and IPRD) in a range from 0 to 15.

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

When ICR1 and ICR3 are overwritten, IRQ interrupts may be mistakenly detected, depending on the $\overline{\text{IRQ}}$ pin level. To prevent this, overwrite the register while interrupts are masked, then release the mask after clearing the illegal interrupt by writing 0 to interrupt request register 0 (IRR0).

Edge input interrupt detection requires input of a pulse width of more than two cycles on a P clock basis.

When using level-sensing for IRQ interrupts, the pin levels must be retained until the CPU samples the pins. Therefore, the interrupt source must be cleared by the interrupt handler.

The interrupt mask bits (I3 to I0) in the status register (SR) are not affected by IRQ interrupt handling.

10.4.4 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are generated by the following 9 modules:

- DMA controller (DMAC)
- Serial communication interfaces (SCIF0 to SCIF2)
- A/D converters (ADC0 and ADC1)
- Compare match timers (CMT0 and CMT1)
- USB function module (USB)
- Multifunction timer pulse units (MTU0 to MTU4)
- Watchdog timer (WDT)
- User debugging interface (H-UDI)
- I^2C bus interface 2 (IIC2)

Not every interrupt source is assigned a different interrupt vector. Sources are reflected in the interrupt event register (INTEVT2). It is easy to identify sources by using the value of the INTEVT2 register as a branch offset.

A priority level (from 0 to 15) can be set for each module except H-UDI by writing to interrupt priority registers B to J (IPRB to IPRJ). The priority level of the H-UDI interrupt is 15 (fixed).

The interrupt mask bits (I3 to I0) in the status register are not affected by on-chip peripheral module interrupt handling.

10.4.5 Interrupt Exception Handling and Priority

There are three types of interrupt sources: NMI, IRQ, and on-chip peripheral modules. The priority of each interrupt source is set within level 0 to level 16; level 16 is the highest and level 1 is the lowest. When the priority is set to level 0, that interrupt is masked and the interrupt request is ignored.

Table 10.5 lists the codes for the interrupt event register (INTEVT2) and the order of interrupt priority.

Each interrupt source is assigned a unique code by INTEVT2. The start address of the interrupt service routine is common for each interrupt source. This is why, for instance, the value of INTEVT2 is used as an offset at the start of the interrupt service routine and branched to in order to identify the interrupt source.

IRQ interrupt and on-chip peripheral module interrupt priorities can be set freely between 0 and 15 for each module by setting interrupt priority registers A to J (IPRA to IPRJ). A reset assigns priority level 0 to IRQ and on-chip peripheral module interrupts.

If the same priority level is assigned to two or more interrupt sources and interrupts from those sources occur simultaneously, their priority order is the default priority order indicated at the right in table 10.5.

Table 10.5 Interrupt Exception Handling Sources and Priority

Interrupt Source		Exception Code	Interrupt Priory (Initial Value)	IPR (Bit Number)	Priority within IPR Setting Unit	Default Priority
NMI		H'1C0	16	—	—	High
H-UDI interrupt		H'5E0	15	—	—	
IRQ	IRQ0	H'600	0 to 15 (0)	IPRC (3 to 0)	—	
	IRQ1	H'620	0 to 15 (0)	IPRC (7 to 4)	—	
	IRQ2	H'640	0 to 15 (0)	IPRC (11 to 8)	—	
	IRQ3	H'660	0 to 15 (0)	IPRC (15 to 12)	—	
	IRQ4	H'680	0 to 15 (0)	IPRD (3 to 0)	—	
	IRQ5	H'6A0	0 to 15 (0)	IPRD (7 to 4)	—	
	IRQ6	H'6C0	0 to 15 (0)	IPRD (11 to 8)	—	
	IRQ7	H'6E0	0 to 15 (0)	IPRD (15 to 12)	—	
DMAC0	DEI0	H'800	0 to 15 (0)	IPRJ (15 to 12)	—	
DMAC1	DEI1	H'820	0 to 15 (0)	IPRJ (11 to 8)	—	
DMAC2	DEI2	H'840	0 to 15 (0)	IPRJ (7 to 4)	—	
DMAC3	DEI3	H'860	0 to 15 (0)	IPRJ (3 to 0)	—	
SCIF0	ERI0	H'880	0 to 15 (0)	IPRE (11 to 8)	High	
	RxI0	H'8A0				
	BRI0	H'8C0				
	TxI0	H'8E0			Low	
SCIF1	ERI1	H'900	0 to 15 (0)	IPRE (7 to 4)	High	
	RxI1	H'920				
	BRI1	H'940				
	TxI1	H'960			Low	
ADC	ADI0	H'980	0 to 15 (0)	IPRE (3 to 0)	High	
	ADI1	H'9A0		IPRF (15 to 12)	Low	
USB	USI0	H'A00	0 to 15 (0)	IPRF (7 to 4)	High	
	USI1	H'A20				
	USIHP	H'A40			Low	Low

Interrupt Source		Exception Code	Interrupt Priory (Initial Value)	IPR (Bit Number)	Priority within IPR Setting Unit	Default Priority
MTU0	TGI0A	H'A80	0 to 15 (0)	IPRG (15 to 12)	High	High
	TGI0B	H'AA0				
	TGI0C	H'AC0				
	TGI0D	H'AE0			Low	
	TCI0V	H'B00		IPRG (11 to 8)	—	
MTU1	TGI1A	H'C00	0 to 15 (0)	IPRG (7 to 4)	High	
	TGI1B	H'C20			Low	
	TCI1V	H'C40		IPRG (3 to 0)	High	
	TCI1U	H'C60			Low	
MTU2	TGI2A	H'C80	0 to 15 (0)	IPRH (15 to 12)	High	
	TGI2B	H'CA0			Low	
	TCI2V	H'CC0		IPRH (11 to 8)	High	
	TCI2U	H'CE0			Low	
MTU3	TGI3A	H'D00	0 to 15 (0)	IPRH (7 to 4)	High	
	TGI3B	H'D20				
	TGI3C	H'D40				
	TGI3D	H'D60			Low	
	TCI3V	H'D80		IPRH (3 to 0)	—	
MTU4	TGI4A	H'E00	0 to 15 (0)	IPRI (15 to 12)	High	
	TGI4B	H'E20				
	TGI4C	H'E40				
	TGI4D	H'E60			Low	
	TCI4V	H'E80		IPRI (11 to 8)	—	
CMT	CMI0	H'F00	0 to 15 (0)	IPRF (3 to 0)	High	
	CMI1	H'F20			Low	
SCIF2	ERI2	H'400	0 to 15 (0)	IPRF (11 to 8)	High	
	RxI2	H'420				
	BRI2	H'440				
	TxI2	H'460			Low	
POE	OEI	H'480	0 to 15 (0)	IPRI (7 to 4)	—	
IIC2	IIC2I	H'F40	0 to 15 (0)	IPRI (3 to 0)	—	
WDT	ITI	H'560	0 to 15 (0)	IPRB (15 to 12)	—	Low

10.5 INTC Operation

10.5.1 Interrupt Sequence

The sequence of interrupt operations is described below. Figure 10.2 is a flowchart of the operations.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest-priority interrupt from the interrupt requests sent, following the priority levels set in interrupt priority registers B to J (IPRB to IPRJ). Lower priority interrupts are held pending. If two of these interrupts have the same priority level or if multiple interrupts occur within a single module, the interrupt with the highest priority is selected, according to table 10.5.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt mask bits (I3 to I0) in the status register (SR) of the CPU. If the request priority level is higher than the level in bits I3 to I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. Detection timing: The INTC operates, and notifies the CPU of interrupt requests, in synchronization with the peripheral clock (Pϕ). The CPU receives an interrupt at a break in instructions.
5. The interrupt source code is set in the interrupt event register (INTEVT2).
6. The status register (SR) and program counter (PC) are saved to SSR and SPC, respectively.
7. The block bit (BL) and register bank bit (RB) in SR are set to 1.
8. The CPU jumps to the start address of the interrupt handler (the sum of the value set in the vector base register (VBR) and H'00000600). This jump is not a delayed branch. The interrupt handler may branch with the INTEVT2 register value as its offset in order to identify the interrupt source. This enables it to branch to the handling routine for the individual interrupt source.

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

2. The interrupt source flag should be cleared in the interrupt handler. To ensure that an interrupt request that should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, and then execute an RTE instruction.

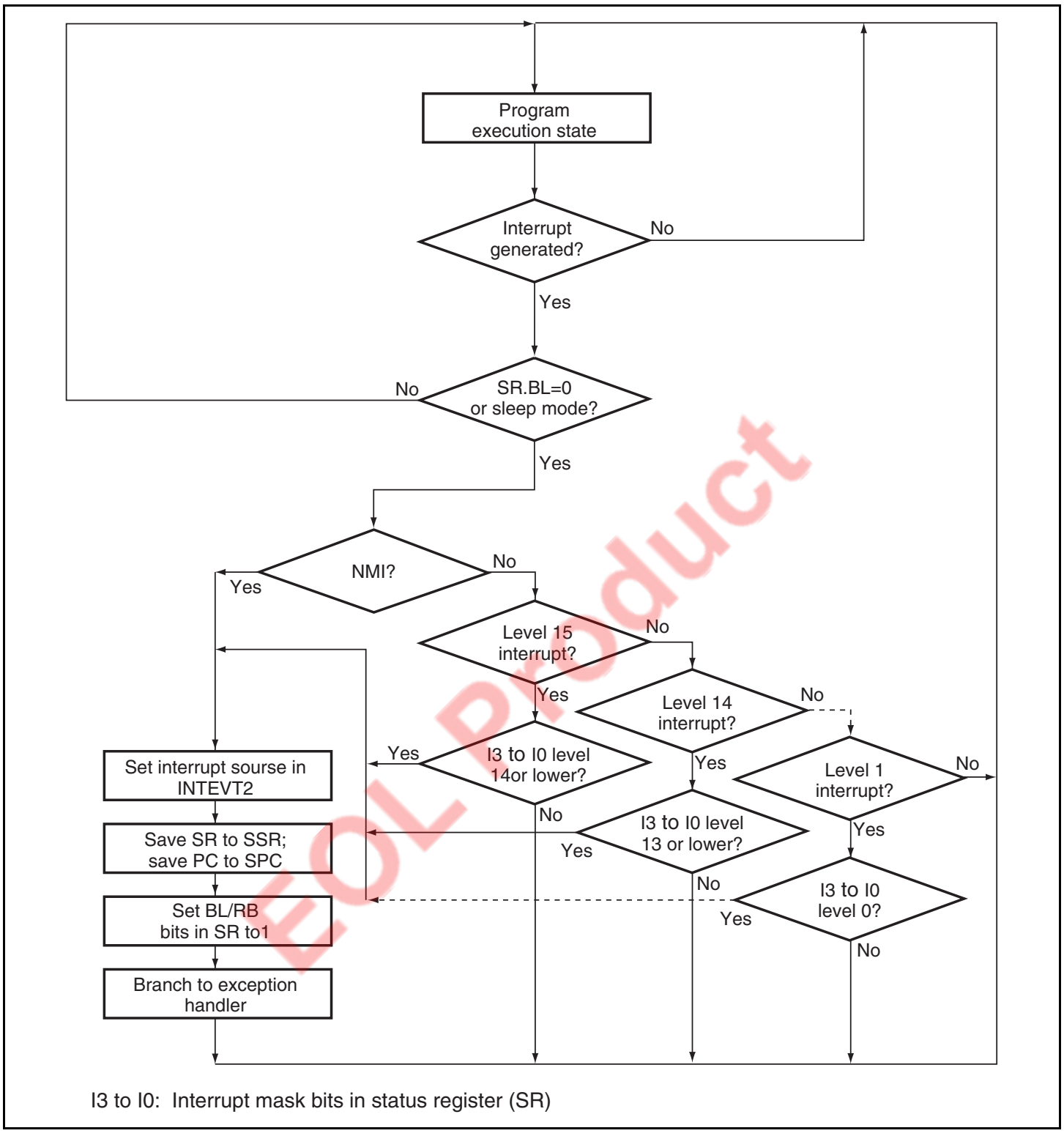


Figure 10.2 Interrupt Operation Flowchart

10.5.2 Multiple Interrupts

When handling multiple interrupts, an interrupt handler should include the following procedures:

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

When these procedures are followed in order, an interrupt of higher priority than the one being handled can be accepted after clearing BL in step 4. Figure 10.2 shows a sample interrupt operation flowchart.

10.6 Notes on Use

10.6.1 Notes on USB Bus Power Control

Use $\overline{\text{IRQ0}}$/$\overline{\text{IRQ1}}$ carefully. The USB bus power control uses the interrupt control logic block for $\overline{\text{IRQ0}}$/$\overline{\text{IRQ1}}$.

For the details about the USB bus power control, refer to section 20, USB Function Module.

10.6.2 Timing to Clear an Interrupt Source

As described in section 10.5.1, Interrupt Sequence, clear the interrupt source flags in the interrupt handler.

To avoid accepting an interrupt source flag that has been cleared, read the flag and then, execute the RTE instruction.

Section 11 User Break Controller (UBC)

The user break controller (UBC) provides functions that simplify program debugging. These functions make it easy to design an effective self-monitoring debugger, enabling the chip to debug programs without using an in-circuit emulator. Break conditions that can be set in the UBC are instruction fetch or data read/write access, data size, data contents, address value, and stop timing in the case of instruction fetch.

11.1 Features

The UBC has the following features:

1. The following break comparison conditions can be set.
   Number of break channels: two channels (channels A and B)
   User break can be requested as either the independent or sequential condition on channels A and B (sequential break setting: channel A and then channel B match with break conditions, but not in the same bus cycle).

- Address
  Comparison bits are maskable in 1-bit units.
  One of the four address buses (logic address bus (LAB), internal address bus (IAB), X-memory address bus (XAB), and Y-memory address bus (YAB)) can be selected.
- Data
  Only on channel B, 32-bit maskable.
  One of the four data buses (L-bus data (LDB), I-bus data (IDB), X-memory data bus (XDB) and Y-memory data bus (YDB)) can be selected.
- Bus cycle
  Instruction fetch or data access
- Read/write
- Operand size
  Byte, word, and longword

2. A user-designed user-break condition exception processing routine can be run.
3. In an instruction fetch cycle, it can be selected that a break is set before or after an instruction is executed.
4. Maximum repeat times for the break condition (only for channel B): $2^{12} - 1$ times.
5. Eight pairs of branch source/destination buffers.

Figure 11.1 shows a block diagram of the UBC.

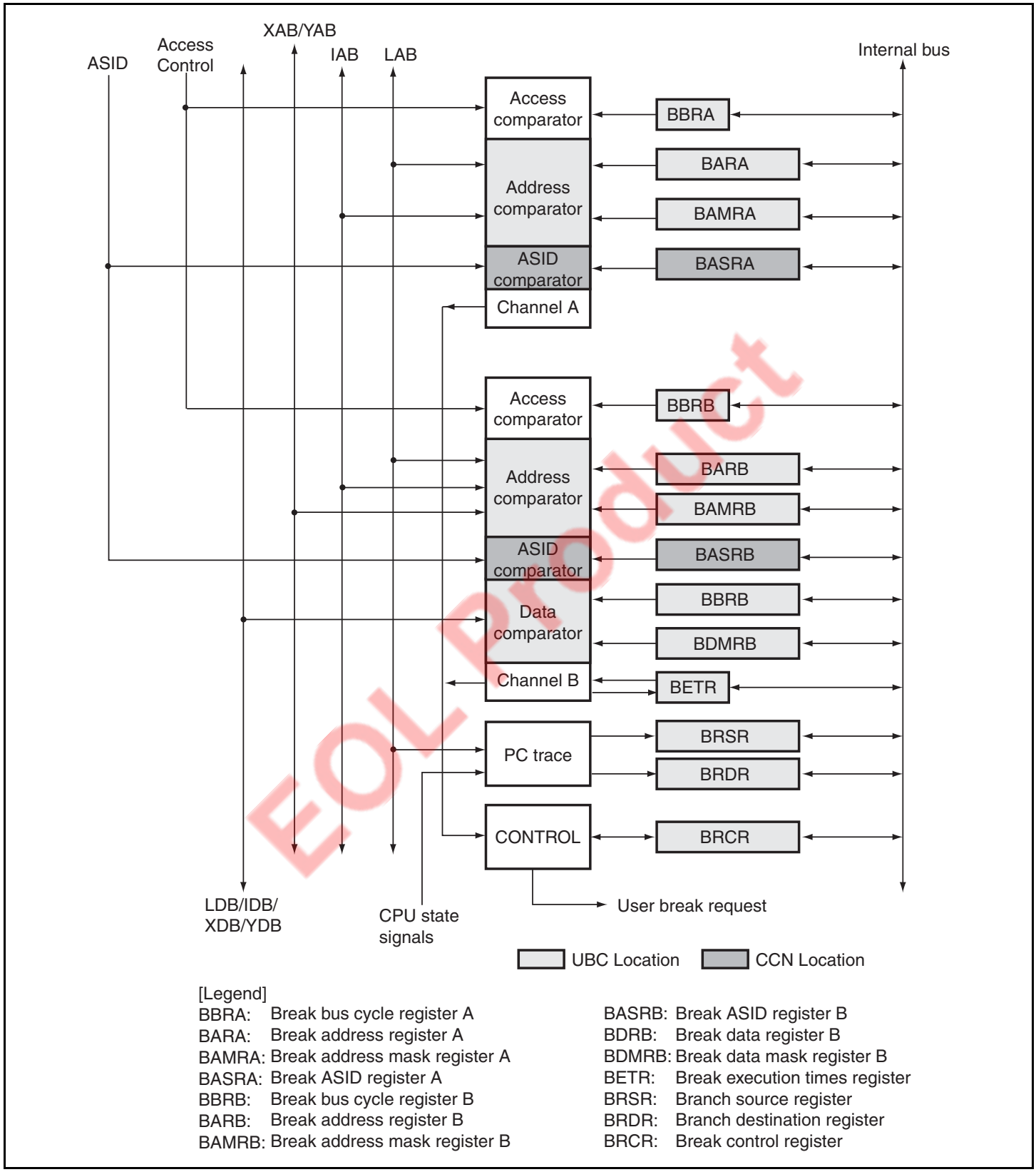


Figure 11.1 Block Diagram of User Break Controller

11.2 Register Descriptions

The user break controller has the following registers. For details on register addresses and access sizes, refer to section 24, List of Registers.

- Break address register A (BARA)
- Break address mask register A (BAMRA)
- Break bus cycle register A (BBRA)
- Break address register B (BARB)
- Break address mask register B (BAMRB)
- Break bus cycle register B (BBRB)
- Break data register B (BDRB)
- Break data mask register B (BDMRB)
- Break control register (BRCR)
- Execution times break register (BETR)
- Branch source register (BRSR)
- Branch destination register (BRDR)

11.2.1 Break Address Register A (BARA)

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

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

11.2.2 Break Address Mask Register A (BAMRA)

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

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

11.2.3 Break Bus Cycle Register A (BBRA)

Break bus cycle register A (BBRA) is a 16-bit readable/writable register, which specifies (1) L bus cycle or I bus cycle, (2) instruction fetch or data access, (3) read or write, and (4) operand size in the break conditions of channel A.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	CDA1	0	R/W	L Bus Cycle/I Bus Cycle Select A
6	CDA0	0	R/W	Select the L bus cycle or I bus cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the L bus cycle 10: The break condition is the I bus cycle 11: The break condition is the L bus cycle

Bit	Bit Name	Initial Value	R/W	Description
5	IDA1	0	R/W	Instruction Fetch/Data Access Select A
4	IDA0	0	R/W	Select the instruction fetch cycle or data access cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the instruction fetch cycle 10: The break condition is the data access cycle 11: The break condition is the instruction fetch cycle or data access cycle
3	RWA1	0	R/W	Read/Write Select A
2	RWA0	0	R/W	Select the read cycle or write cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the read cycle 10: The break condition is the write cycle 11: The break condition is the read cycle or write cycle
1	SZA1	0	R/W	Operand Size Select A
0	SZA0	0	R/W	Select the operand size of the bus cycle for the channel A break condition. 00: The break condition does not include operand size 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

11.2.4 Break Address Register B (BARB)

BARB is a 32-bit readable/writable register. BARB specifies the address used as a break condition in channel B. Control bits CDB1, CDB0, XYE, and XYS in BBRB select one of the four address buses for break condition B.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAB31 to BAB0	All 0	R/W	Break Address B Store an address which specifies a break condition in channel B. If the I bus or L bus is selected in BBRB, an IAB or LAB address is set in BAB31 to BAB0. If the X memory is selected in BBRB, the values in bits 15 to 1 in XAB are set in BAB31 to BAB17. In this case, the values in BAB16 to BAB0 are arbitrary. If the Y memory is selected in BBRB, the values in bits 15 to 1 in YAB are set in BAB15 to BAB1. In this case, the values in BAB31 to BAB16 are arbitrary.

Table 11.1 Specifying Break Address Register

Bus Selection in BBRB	BAB31 to BAB17	BAB16	BAB15 to BAB1	BAB0
L bus	LAB31 to LAB0			
I bus	IAB31 to IAB0			
X bus	XAB15 to XAB1	Don't care	Don't care	Don't care
Y bus	Don't care	Don't care	YAB15 to YAB1	Don't care

11.2.5 Break Address Mask Register B (BAMRB)

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

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

11.2.6 Break Data Register B (BDRB)

BDRB is a 32-bit readable/writable register. The control bits CDB1, CDB0, XYE, and XYS in BBRB select one of the four data buses for break condition B.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDB31 to BDB0	All 0	R/W	Break Data Bit B Store data which specifies a break condition in channel B. If the I bus is selected in BBRB, the break data on IDB is set in BDB31 to BDB0. If the L bus is selected in BBRB, the break data on LDB is set in BDB31 to BDB0. If the X memory is selected in BBRB, the break data in bits 15 to 0 in XDB is set in BDB31 to BDB16. In this case, the values in BDB15 to BDB0 are arbitrary. If the Y memory is selected in BBRB, the break data in bits 15 to 0 in YDB are set in BDB15 to BDB0. In this case, the values in BDB31 to BDB16 are arbitrary.

Table 11.2 Specifying Break Data Register

Bus Selection in BBRB	BDB31 to BDB16	BDB15 to BDB0
L bus	LDB31 or LDB0	
I bus	IDB31 to IDB0	
X bus	XDB15 to XDB0	Don't care
Y bus	Don't care	YDB15 to YDB0

Notes: 1. Specify an operand size when including the value of the data bus in the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 15 to 8 and 7 to 0 in BDRB as the break data.
3. Set the data in bits 31 to 16 when including the value of the data bus as an L-bus break condition for the MOVS.W @-As,Ds, MOVS.W @As,Ds, MOVS.W @As+,Ds, or MOVS.W @As+Ix,Ds instruction.

11.2.7 Break Data Mask Register B (BDMRB)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDMB31 to BDMB0	All 0	R/W	Break Data Mask B Specify bits masked in the break data of channel B specified by BDRB (BDB31 to BDB0). 0: Break data BDBn of channel B is included in the break condition 1: Break data BDBn of channel B is masked and is not included in the break condition Note: n = 31 to 0

Notes: 1. Specify an operand size when including the value of the data bus in the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 15 to 8 and 7 to 0 in BDRB as the break mask data in BDMRB.
3. Set the mask data in bits 31 to 16 when including the value of the data bus as an L-bus break condition for the MOVS.W @-As,Ds, MOVS.W @As,Ds, MOVS.W @As+,Ds, or MOVS.W @As+Ix,Ds instruction.

11.2.8 Break Bus Cycle Register B (BBRB)

Break bus cycle register B (BBRB) is a 16-bit readable/writable register, which specifies (1) X bus or Y bus, (2) L bus cycle or I bus cycle, (3) instruction fetch or data access, (4) read or write, and (5) operand size in the break conditions of channel B.

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	XYE	0	R/W	Selects the X memory bus or Y memory bus as the channel B break condition. Note that this bit setting is enabled only when the L bus is selected with the CDB1 and CDB0 bits. Selection between the X memory bus and Y memory bus is done by the XYS bit. 0: Selects L bus for the channel B break condition unconditionally 1: Selects X/Y memory bus for the channel B break condition
8	XYS	0	R/W	Selects the X bus or the Y bus as the bus of the channel B break condition. 0: Selects the X bus for the channel B break condition 1: Selects the Y bus for the channel B break condition
7	CDB1	0	R/W	L Bus Cycle/I Bus Cycle Select B
6	CDB0	0	R/W	Select the L bus cycle or I bus cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the L bus cycle 10: The break condition is the I bus cycle 11: The break condition is the L bus cycle

Bit	Bit Name	Initial Value	R/W	Description
5	IDB1	0	R/W	Instruction Fetch/Data Access Select B
4	IDB0	0	R/W	Select the instruction fetch cycle or data access cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the instruction fetch cycle 10: The break condition is the data access cycle 11: The break condition is the instruction fetch cycle or data access cycle
3	RWB1	0	R/W	Read/Write Select B
2	RWB0	0	R/W	Select the read cycle or write cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the read cycle 10: The break condition is the write cycle 11: The break condition is the read cycle or write cycle
1	SZB1	0	R/W	Operand Size Select B
0	SZB0	0	R/W	Select the operand size of the bus cycle for the channel B break condition. 00: The break condition does not include operand size 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

11.2.9 Break Control Register (BRCR)

BRCR sets the following conditions:

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

The break control register (BRCR) is a 32-bit readable/writable register that has break conditions match flags and bits for setting a variety of break conditions.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	SCMFCA	0	R/W	L Bus Cycle Condition Match Flag A When the L bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The L bus cycle condition for channel A does not match 1: The L bus cycle condition for channel A matches
14	SCMFCB	0	R/W	L Bus Cycle Condition Match Flag B When the L bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The L bus cycle condition for channel B does not match 1: The L bus cycle condition for channel B matches

Bit	Bit Name	Initial Value	R/W	Description
13	SCMFDA	0	R/W	I Bus Cycle Condition Match Flag A When the I bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The I bus cycle condition for channel A does not match 1: The I bus cycle condition for channel A matches
12	SCMFDB	0	R/W	I Bus Cycle Condition Match Flag B When the I bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The I bus cycle condition for channel B does not match 1: The I bus cycle condition for channel B matches
11	PCTE	0	R/W	PC Trace Enable 0: Disables PC trace 1: Enables PC trace
10	PCBA	0	R/W	PC Break Select A Selects the break timing of the instruction fetch cycle for channel A as before or after instruction execution. 0: PC break of channel A is set before instruction execution 1: PC break of channel A is set after instruction execution
9, 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	DBEB	0	R/W	Data Break Enable B Selects whether or not the data bus condition is included in the break condition of channel B. 0: No data bus condition is included in the condition of channel B 1: The data bus condition is included in the condition of channel B

Bit	Bit Name	Initial Value	R/W	Description
6	PCBB	0	R/W	PC Break Select B Selects the break timing of the instruction fetch cycle for channel B as before or after instruction execution. 0: PC break of channel B is set before instruction execution 1: PC break of channel B is set after instruction execution
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SEQ	0	R/W	Sequence Condition Select Selects two conditions of channels A and B as independent or sequential conditions. 0: Channels A and B are compared under independent conditions 1: Channels A and B are compared under sequential conditions (channel A, then channel B)
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ETBE	0	R/W	Number of Execution Times Break Enable Enables the execution-times break condition only on channel B. If this bit is 1 (break enable), a user break is issued when the number of break conditions matches with the number of execution times that is specified by BETR. 0: The execution-times break condition is disabled on channel B 1: The execution-times break condition is enabled on channel B

11.2.10 Execution Times Break Register (BETR)

BETR is a 16-bit readable/writable register. When the execution-times break condition of channel B is enabled, this register specifies the number of execution times to make the break. The maximum number is $2^{12} - 1$ times. When a break condition is satisfied, it decreases BETR. A break is issued when the break condition is satisfied after BETR becomes H'0001.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	BET11 to BET0	All 0	R/W	Number of Execution Times

11.2.11 Branch Source Register (BRSR)

BRSR is a 32-bit read-only register. BRSR stores bits 27 to 0 in the address of the branch source instruction. BRSR has the flag bit that is set to 1 when a branch occurs. This flag bit is cleared to 0 when BRSR is read, the setting to enable PC trace is made, or BRSR is initialized by a power-on reset. Other bits are not initialized by a power-on reset. The eight BRSR registers have a queue structure and a stored register is shifted at every branch.

Bit	Bit Name	Initial Value	R/W	Description
31	SVF	0	R	BRSR Valid Flag Indicates whether the branch source address is stored. When a branch source address is fetched, this flag is set to 1. This flag is cleared to 0 by reading from BRSR. 0: The value of BRSR register is invalid 1: The value of BRSR register is valid
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 0	BSA27 to BSA0	—	R	Branch Source Address Store bits 27 to 0 of the branch source address.

11.2.12 Branch Destination Register (BRDR)

BRDR is a 32-bit read-only register. BRDR stores bits 27 to 0 in the address of the branch destination instruction. BRDR has the flag bit that is set to 1 when a branch occurs. This flag bit is cleared to 0 when BRDR is read, the setting to enable PC trace is made, or BRDR is initialized by a power-on reset. Other bits are not initialized by a power-on reset. The eight BRDR registers have a queue structure and a stored register is shifted at every branch.

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

11.3 Operation

11.3.1 Flow of the User Break Operation

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

1. The break addresses is set in the break address registers (BARA or BARB). The masked addresses are set in the break address mask registers (BAMRA or BAMRB). The break data is set in the break data register (BDRB). The masked data is set in the break data mask register (BDMRB). The bus break conditions are set in the break bus cycle registers (BBRA or BBRB). Three groups of BBRA or BBRB (L bus cycle/I bus cycle select, instruction fetch/data access select, and read/write select) are each set. No user break will be generated if even one of these groups is set with 00. The respective conditions are set in the bits of the break control register (BRCR). Make sure to set all registers related to breaks before setting BBRA or BBRB.
2. When the break conditions are satisfied, the UBC sends a user break request to the CPU and sets the L bus condition match flag (SCMFCA or SCMFCB) and the I bus condition match flag (SCMFDA or SCMFDB) for the appropriate channel. When the X/Y memory bus is specified for channel B, SCMFCB is used for the condition match flag.
3. The appropriate condition match flags (SCMFCA, SCMFDA, SCMFCB, and SCMFDB) can be used to check if the set conditions match or not. The matching of the conditions sets flags, but they are not reset. 0 must first be written to them before they can be used again.
4. There is a chance that the break set in channel A and the break set in channel B occur around the same time. In this case, there will be only one break request to the CPU, but these two break channel match flags could be both set.
5. When selecting the I bus as the break condition, note the following:
   - Several bus masters, including the CPU and DMAC, are connected to the I bus. The UBC monitors bus cycles generated by all bus masters, and determines the condition match.
   - Physical addresses are used for the I bus. Set a physical address in break address registers (BARA and BARB). The upper three bits of logical addresses in the P0 to P3 area issued by the CPU on the L bus are masked (to 0) before they are placed on the I bus. The upper three bits of the source and destination addresses set in the DMAC are masked in the same way. However, logical addresses in the P4 area are output unchanged on the I bus.
   - For data access cycles issued on the L bus by the CPU, if their logical addresses are not to be cached, they are issued with the data size specified on the L bus.
   - For instruction fetch cycles issued on the L bus by the CPU, even though their logical addresses are not to be cached, they are issued in longwords and their addresses are rounded to match longword boundaries.

— If a logical address issued on the L bus by the CPU is an address to be cached and a cache miss occurs, its bus cycle is issued as a cache fill cycle on the I bus. In this case, it is issued in longwords and its address is rounded to match longword boundaries. However note that cache fill is not performed for a write miss in write through mode. In this case, the bus cycle is issued with the data size specified on the L bus and its address is not rounded. In write back mode, a write back cycle may be issued in addition to a read fill cycle. It is a longword bus cycle whose address is rounded to match longword boundaries.

— I bus cycles (including read fill cycles) resulting from instruction fetches on the L bus by the CPU are defined as instruction fetch cycles on the I bus, while other bus cycles are defined as data access cycles.

— The DMAC only issues data access cycles for I bus cycles.

— If a break condition is specified for the I bus, even when the condition matches in an I bus cycle resulting from an instruction executed by the CPU, at which instruction the break is to be accepted cannot be clearly defined.

6. While the block bit (BL) in the CPU status register (SR) is set to 1, no breaks can be accepted. However, condition determination will be carried out, and if the condition matches, the corresponding condition match flag is set to 1.

11.3.2 Break on Instruction Fetch Cycle

1. When L bus/instruction fetch/read/word or longword is set in the break bus cycle register (BBRA or BBRB), the break condition becomes the L bus instruction fetch cycle. Whether it breaks before or after the execution of the instruction can then be selected with the PCBA or PCBB bit of the break control register (BRCR) for the appropriate channel. If an instruction fetch cycle is set as a break condition, clear LSB in the break address register (BARA or BARB) to 0. A break cannot be generated as long as this bit is set to 1.
2. An instruction set for a break before execution breaks when it is confirmed that the instruction has been fetched and will be executed. This means this feature cannot be used on instructions fetched by overrun (instructions fetched at a branch or during an interrupt transition, but not to be executed). When this kind of break is set for the delay slot of a delayed branch instruction, the break is generated prior to execution of the delayed branch instruction.

   Note: If a branch does not occur at a delay condition branch instruction, the subsequent instruction is not recognized as a delay slot.

3. When the condition is specified to be occurred after execution, the instruction set with the break condition is executed and then the break is generated prior to the execution of the next instruction. As with pre-execution breaks, this cannot be used with overrun fetch instructions. When this kind of break is set for a delayed branch instruction and its delay slot, a break is not generated until the first instruction at the branch destination.

4. When an instruction fetch cycle is set for channel B, the break data register B (BDRB) is ignored. Therefore, break data cannot be set for the break of the instruction fetch cycle.
5. If the I bus is set for a break of an instruction fetch cycle, the condition is determined for the instruction fetch cycles on the I bus. For details, see 5 in section 11.3.1, Flow of the User Break Operation.

11.3.3 Break on Data Access Cycle

1. If the L bus is specified as a break condition for data access break, condition comparison is performed for the logical addresses (and data) accessed by the executed instructions, and a break occurs if the condition is satisfied. If the I bus is specified as a break condition, condition comparison is performed for the physical addresses (and data) of the data access cycles that are issued on the I bus by all bus masters including the CPU, and a break occurs if the condition is satisfied. For details on the CPU bus cycles issued on the I bus, see 5 in section 11.3.1, Flow of the User Break Operation.
2. The relationship between the data access cycle address and the comparison condition for each operand size is listed in table 11.3.

Table 11.3 Data Access Cycle Addresses and Operand Size Comparison Conditions

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

This means that when address H'00001003 is set in the break address register (BARA or BARB), for example, the bus cycle in which the break condition is satisfied is as follows (where other conditions are met).

Longword access at H'00001000

Word access at H'00001002

Byte access at H'00001003

3. When the data value is included in the break conditions on channel B:

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

word data in bits 31 to 16 in BDRB and BDMRB when including the value of the data bus as a break condition for the MOVS.W @-As,Ds, MOVS.W @As,Ds, MOVS.W @As+,Ds, or MOVS.W @As+Ix,Ds instruction (bits 15 to 0 are ignored).

4. Access by a PREF instruction is handled as read access in longword units without access data. Therefore, if including the value of the data bus when a PREF instruction is specified as a break condition, a break will not occur.
5. If the L bus is selected, a break occurs on ending execution of the instruction that matches the break condition, and immediately before the next instruction is executed. However, when data is also specified as the break condition, the break may occur on ending execution of the instruction following the instruction that matches the break condition. If the I bus is selected, the instruction at which the break will occur cannot be determined. When this kind of break occurs at a delayed branch instruction or its delay slot, the break may not actually take place until the first instruction at the branch destination.

11.3.4 Break on X/Y-Memory Bus Cycle

1. The break condition on an X/Y-memory bus cycle is specified only in channel B. If the XYE bit in BBRB is set to 1, the break address and break data on X/Y-memory bus are selected. At this time, select the X-memory bus or Y-memory bus by specifying the XYS bit in BBRB. The break condition cannot include both X-memory and Y-memory at the same time. The break condition is applied to an X/Y-memory bus cycle by specifying L bus/data access/read or write/word or no specified operand size in bits 7 to 0 in the break bus cycle register B (BBRB).
2. When an X-memory address is selected as the break condition, specify an X-memory address in the upper 16 bits in BARB and BAMRB. When a Y-memory address is selected, specify a Y-memory address in the lower 16 bits. Specification of X/Y-memory data is the same for BDRB and BDMRB.
3. The timing of a data access break for the X memory or Y memory bus to occur is the same as a data access break of the L bus. For details, see 5 in section 11.3.3, Break on Data Access Cycle.

11.3.5 Sequential Break

1. By setting the SEQ bit in BRCR to 1, the sequential break is issued when a channel B break condition matches after a channel A break condition matches. A user break is not generated even if a channel B break condition matches before a channel A break condition matches. When channels A and B conditions match at the same time, the sequential break is not issued. To clear the channel A condition match when a channel A condition match has occurred but a channel B condition match has not yet occurred in a sequential break specification, clear the SEQ bit in BRCR to 0.
2. In sequential break specification, the L/I/X/Y bus can be selected and the execution times break condition can be also specified. For example, when the execution times break condition is specified, the break condition is satisfied when a channel B condition matches with BETR = H'0001 after a channel A condition has matched.

11.3.6 Value of Saved Program Counter

When a break occurs, the address of the instruction from where execution is to be resumed is saved in the SPC, and the exception handling state is entered. If the L bus is specified as a break condition, the instruction at which the break should occur can be clearly determined (except for when data is included in the break condition). If the I bus is specified as a break condition, the instruction at which the break should occur cannot be clearly determined.

1. When instruction fetch (before instruction execution) is specified as a break condition:
   The address of the instruction that matched the break condition is saved in the SPC. The instruction that matched the condition is not executed, and the break occurs before it. However when a delay slot instruction matches the condition, the address of the delayed branch instruction is saved in the SPC.
2. When instruction fetch (after instruction execution) is specified as a break condition:
   The address of the instruction following the instruction that matched the break condition is saved in the SPC. The instruction that matches the condition is executed, and the break occurs before the next instruction is executed. However when a delayed branch instruction or delay slot matches the condition, these instructions are executed, and the branch destination address is saved in the SPC.
3. When data access (address only) is specified as a break condition:
   The address of the instruction immediately after the instruction that matched the break condition is saved in the SPC. The instruction that matches the condition is executed, and the break occurs before the next instruction is executed. However when a delay slot instruction matches the condition, the branch destination address is saved in the SPC.

4. When data access (address + data) is specified as a break condition:

   When a data value is added to the break conditions, the address of an instruction that is within two instructions of the instruction that matched the break condition is saved in the SPC. At which instruction the break occurs cannot be determined accurately.

   When a delay slot instruction matches the condition, the branch destination address is saved in the SPC. If the instruction following the instruction that matches the break condition is a branch instruction, the break may occur after the branch instruction or delay slot has finished. In this case, the branch destination address is saved in the SPC.

11.3.7 PC Trace

1. Setting PCTE in BRCR to 1 enables PC traces. When branch (branch instruction, and interrupt exception) is generated, the branch source address and branch destination address are stored in BRSR and BRDR, respectively.
2. The values stored in BRSR and BRDR are as given below due to the kind of branch.
   - If a branch occurs due to a branch instruction, the address of the branch instruction is saved in BRSR and the address of the branch destination instruction is saved in BRDR.
   - If a branch occurs due to an interrupt or exception, the value saved in SPC due to exception occurrence is saved in BRSR and the start address of the exception handling routine is saved in BRDR.

   When a repeat loop of the DSP extended function is used, control being transferred from the repeat end instruction to the repeat start instruction is not recognized as a branch, and the values are not stored in BRSR and BRDR.
3. BRSR and BRDR have eight pairs of queue structures. The top of queues is read first when the address stored in the PC trace register is read. BRSR and BRDR share the read pointer. Read BRSR and BRDR in order, the queue only shifts after BRDR is read. After switching the PCTE bit (in BRCR) off and on, the values in the queues are invalid.

11.3.8 Usage Examples

Break Condition Specified for L Bus Instruction Fetch Cycle:

(Example 1-1)

- Register specifications

  BARA = H'00000404, BAMRA = H'00000000, BBRA = H'0054, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000400

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

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

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

  <Channel B>

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

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

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

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

(Example 1-2)

- Register specifications

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

  Specified conditions: Channel A/channel B sequential mode

  <Channel A>

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

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

  <Channel B>

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

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

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

After an instruction with and address H'00037226 is executed, a user break occurs before an instruction with and address H'0003722E is executed.

(Example 1-3)

- Register specifications

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

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

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

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

  <Channel B>

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

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

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

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

(Example 1-4)

- Register specifications

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

  Specified conditions: Channel A/channel B sequential mode

  <Channel A>

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

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

  <Channel B>

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

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

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

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

(Example 1-5)

- Register specifications

  BARA = H'00000500, BAMRA = H'00000000, BBRA = H'0057, BARB = H'00001000, BAMRB = H'00000000, BBRB = H'0057, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000001, BETR = H'0005

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

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

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

  <Channel B>

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

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

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

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

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

(Example 1-6)

- Register specifications

  BARA = H'00008404, BAMRA = H'00000FFF, BBRA = H'0054, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000400

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00008404, Address mask: H'00000FFF

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

  <Channel B>

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

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

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

  A user break occurs after an instruction with addresses H'00008000 to H'00008FFE is executed or before an instruction with addresses H'00008010 to H'00008016 are executed.

Break Condition Specified for L Bus Data Access Cycle:

(Example 2-1)

- Register specifications

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

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

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

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

  <Channel B>

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

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

  Bus cycle: L bus/data access/write/word

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

(Example 2-2)

- Register specifications

  BARA = H'01000000, BAMRA = H'00000000, BBRA = H'0066, BARB = H'0000F000, BAMRB = H'FFFF0000, BBRB = H'036A, BDRB = H'00004567, BDMRB = H'00000000, BRCR = H'00000080

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'01000000, Address mask: H'00000000

  Bus cycle: L bus/data access/read/word

  <Channel B>

  Y Address: H'0000F000, Address mask: H'FFFF0000

  Data: H'00004567, Data mask: H'00000000

  Bus cycle: Y bus/data access/write/word

  On channel A, a user break occurs during word read from address H'01000000 in the memory space. On channel B, a user break occurs when word data H'4567 is written in address H'0000F000 in the Y memory space. The X/Y-memory space is changed by a mode setting.

Break Condition Specified for I Bus Data Access Cycle:

(Example 3-1)

- Register specifications

  BARA = H'00314156, BAMRA = H'00000000, BBRA = H'0094, BARB = H'00055555, BAMRB = H'00000000, BBRB = H'00A9, BDRB = H'00007878, BDMRB = H'00000F0F, BRCR = H'00000080

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00314156, Address mask: H'00000000

  Bus cycle: I bus/instruction fetch/read (operand size is not included in the condition)

  <Channel B>

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

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

  Bus cycle: I bus/data access/write/byte

  On channel A, a user break occurs when instruction fetch is performed for address H'00314156 in the memory space.

  On channel B, a user break occurs when the I bus writes byte data H'7* in address H'00055555.

11.4 Usage Notes

1. The CPU can read from or write to the UBC registers via the I bus. Accordingly, during the period from executing an instruction to rewrite the UBC register till the new value is actually rewritten, the desired break may not occur. In order to know the timing when the UBC register is changed, read from the last written register. Instructions after then are valid for the newly written register value.
2. UBC cannot monitor access to the L bus and I bus in the same channel.
3. Note on specification of sequential break:

   A condition match occurs when a B-channel match occurs in a bus cycle after an A-channel match occurs in another bus cycle in sequential break setting. Therefore, no break occurs even if a bus cycle, in which an A-channel match and a channel B match occur simultaneously, is set.

4. When a user break and another exception occur at the same instruction, which has higher priority is determined according to the priority levels defined in table 9.1 in section 9, Exception Handling. If an exception with higher priority occurs, the user break is not generated.
   - Pre-execution break has the highest priority.
   - When a post-execution break or data access break occurs simultaneously with a re-execution-type exception (including pre-execution break) that has higher priority, the re-execution-type exception is accepted, and the condition match flag is not set (see the exception in the following note). The break will occur and the condition match flag will be set only after the exception source of the re-execution-type exception has been cleared by the exception handling routine and re-execution of the same instruction has ended.
   - When a post-execution break or data access break occurs simultaneously with a completion-type exception (TRAPA) that has higher priority, though a break does not occur, the condition match flag is set.
5. Note the following exception for the above note.

   If a post-execution break or data access break is satisfied by an instruction that generates a CPU address error by data access, the CPU address error is given priority to the break. Note that the UBC condition match flag is set in this case.
6. Note the following when a break occurs in a delay slot.

   If a pre-execution break is set at the delay slot instruction of the RTE instruction, the break does not occur until the branch destination of the RTE instruction.
7. User breaks are disabled during USB module standby mode. Do not read from or write to the UBC registers during USB module standby mode; the values are not guaranteed.
8. When the repeat loop of the DSP extended function is used, even though a break condition is satisfied during execution of the entire repeat loop or several instructions in the repeat loop, the break may be held. For details, see section 9, Exception Handling.

Section 12 Bus State Controller (BSC)

The bus state controller (BSC) outputs control signals for various types of memory that is connected to the external address space and external devices. BSC functions enable this LSI to connect directly with SRAM, SDRAM, and other memory storage devices, and external devices.

12.1 Features

The BSC has the following features:

1. Physical address space is divided into eight areas
   - A maximum 32 or 64 Mbytes for each of the eight areas, CS0, CS2 to CS4, CS5A, CS5B, CS6A and CS6B, totally 384 Mbytes.
   - A maximum 64 Mbytes for each of the six areas, CS0, CS2 to CS4, CS5, and CS6, totally a total of 384 Mbytes.
   - Can specify the normal space interface, SRAM interface with byte selection, burst ROM (clock synchronous or asynchronous), MPX-I/O, burst MPX-I/O, and SDRAM for each address space.
   - Can select the data bus width (8, 16, or 32 bits) for each address space.
   - Controls the insertion of the wait state for each address space.
   - Controls the insertion of the wait state for each read access and write access.
   - Can set the independent idling cycle in the continuous access for five cases: read-write (in same space/different space), read-read (in same space/different space), the first cycle is a write access.
2. Normal space interface
   - Supports the interface that can directly connect to the SRAM.
3. Burst ROM interface (clock asynchronous)
   - High-speed access to the ROM that has the page mode function.
4. MPX-I/O interface
   - Can directly connect to a peripheral LSI that needs an address/data multiplexing.
5. SDRAM interface
   - Can set the SDRAM up to 2 areas.
   - Multiplex output for row address/column address.
   - Efficient access by single read/single write.
   - High-speed access by the bank-active mode.
   - Supports an auto-refresh and self-refresh.

— Supports low-frequency and power-down modes.
— Issues MRS and EMRS commands.

6. Byte-selection SRAM interface
   — Can connect directly to a byte-selection SRAM
7. Burst MPX-IO interface
   — Can connect directly to a peripheral LSI that needs an address/data multiplexing.
   — Supports burst transfer
8. Burst ROM interface (clock synchronous)
   — Can connect directly to a ROM of the clock synchronous type
9. Bus arbitration
   — Shares all of the resources with other CPU and outputs the bus enable after receiving the bus request from external devices.
10. Refresh function
   — Supports the auto-refresh and self-refresh functions.
   — Specifies the refresh interval using the refresh counter and clock selection
   — Can execute concentrated refresh by specifying the refresh counts (1, 2, 4, 6, or 8)

BSC functional block diagram is shown in figure 12.1.

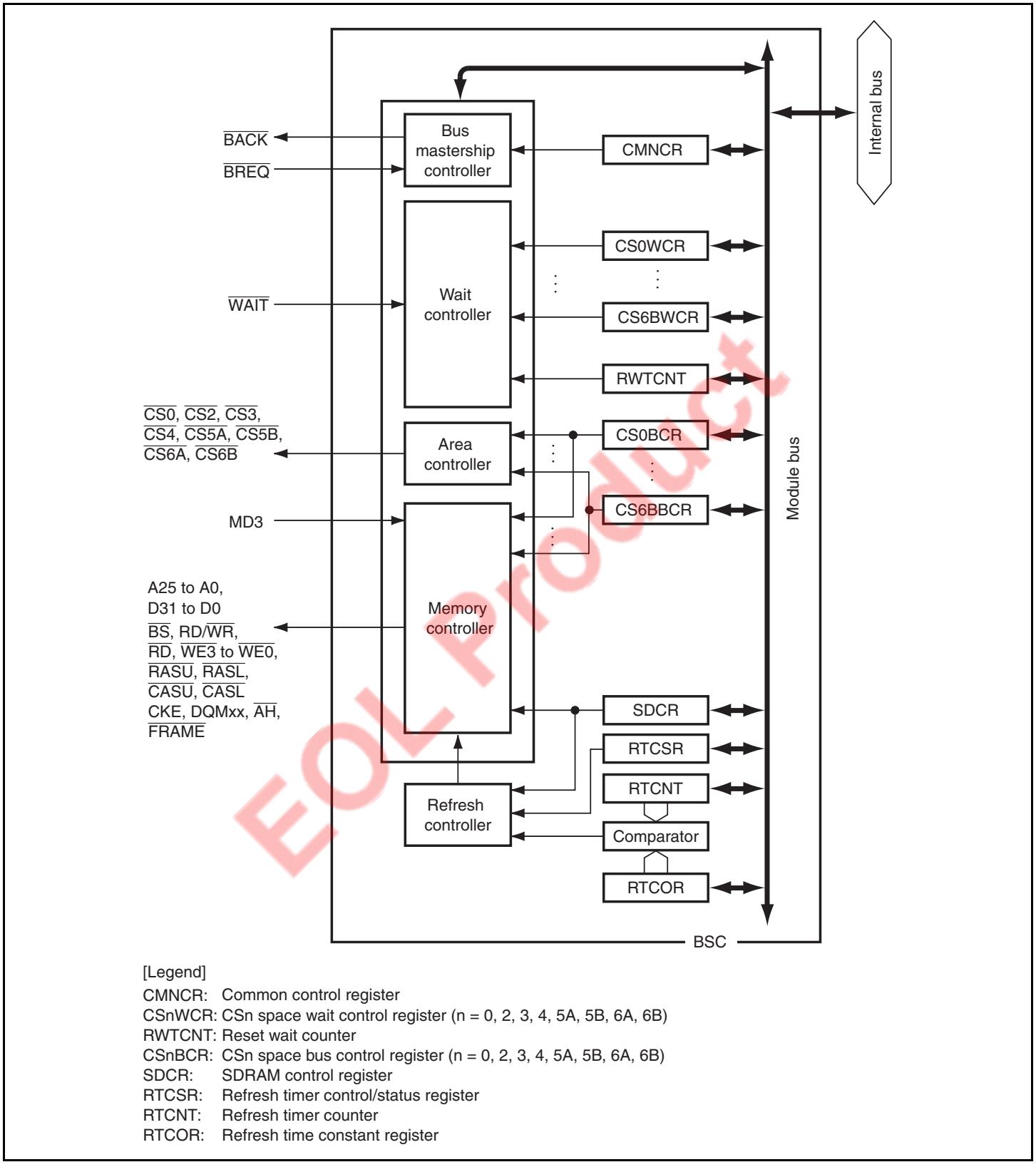


Figure 12.1 BSC Functional Block Diagram

12.2 Input/Output Pins

Table 12.1 shows pin configuration of the BSC.

Table 12.1 Pin Configuration

Name	I/O	Function
A25 to A0	Output	Address bus
D31 to D0	I/O	Data bus
$\overline{BS}$	Output	Bus cycle start
$\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$	Output	Chip select
$\overline{CS5A}$	Output	Chip select Active only for address map 1
RD/$\overline{WR}$	Output	Read/write Connects to $\overline{WE}$ pins when SDRAM or byte-selection SRAM is connected.
$\overline{RD}$	Output	Read pulse signal (read data output enable signal)
$\overline{WE3}$/$\overline{ICIOWR}$/$\overline{AH}$	Output	Indicates that D31 to D24 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Functions as the address hold signal when the MPX-IO is used. Functions as the selection signals for D31 to D24 when SDRAM is connected.
$\overline{WE2}$/$\overline{ICIRD}$	Output	Indicates that D23 to D16 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Functions as the selection signals for D23 to D16 when SDRAM is connected.
$\overline{WE1}$/$\overline{WE}$	Output	Indicates that D15 to D8 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Functions as the selection signals for D15 to D8 when SDRAM is connected.

Name	I/O	Function
$\overline{WE0}$	Output	Indicates that D7 to D0 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Functions as the selection signals for D7 to D0 when SDRAM is connected.
$\overline{RASU}$ $\overline{RASL}$	Output	Connects to $\overline{RAS}$ pin when SDRAM is connected.
$\overline{CASU}$ $\overline{CASL}$	Output	Connects to $\overline{CAS}$ pin when SDRAM is connected.
CKE	Output	Clock enable for SDRAM
$\overline{FRAME}$	Output	Functions as FRAME signal when connected to burst MPX-IO interface
$\overline{WAIT}$	Input	External wait input
$\overline{BREQ}$	Input	Bus request input
$\overline{BACK}$	Output	Bus enable input
MD3	Input	MD3: Select area 0 bus width (16/32 bits)

12.3 Area Overview

12.3.1 Area Division

In the architecture of this LSI, both logical spaces and physical spaces have 32-bit address spaces. The cache access method is shown by the upper three bits. For details see section 7, Cache. The remaining 29 bits are used for division of the space into ten areas (address map 1) or eight areas (address map 2) according to the MAP bit in the CMNCR register setting. The BSC performs control for this 29-bit space.

As listed in tables 12.2 and 12.3, this LSI can connect various memories to eight areas or six areas, and it outputs chip select signals ($\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$, $\overline{CS5A}$, $\overline{CS5B}$, $\overline{CS6A}$, and $\overline{CS6B}$) for each of them. $\overline{CS0}$ is asserted during area 0 access; $\overline{CS5A}$ is asserted during area 5A access when address map 1 is selected; and $\overline{CS5B}$ is asserted when address map 2 is selected. Also $\overline{CS6A}$ is asserted during area 6A access when address map 1 is selected; and $\overline{CS6B}$ is asserted when address map 2 is selected.

12.3.2 Shadow Area

Areas 0, 2 to 4, 5A, 5B, 6A, and 6B are decoded by addresses A28 to A26, which correspond to areas 000 to 110. Address bits 31 to 29 are ignored. This means that the range of area 0 addresses, for example, is H'00000000 to H'03FFFFFF, and its corresponding shadow space is the address space between P0 and P3 obtained by adding to it H'20000000 × n (n = 1 to 6). The address range for area 7 is H'1C000000 to H'1FFFFFFF. The address space H'1C000000 + H'20000000 × n– H'1FFFFFFF + H'20000000 × n (n = 0 to 7) corresponding to the area 7 shadow space is reserved, so do not use it.

Area P4 (H'E0000000 to H'EFFFFFFF) is an I/O area and is assigned for internal register addresses.

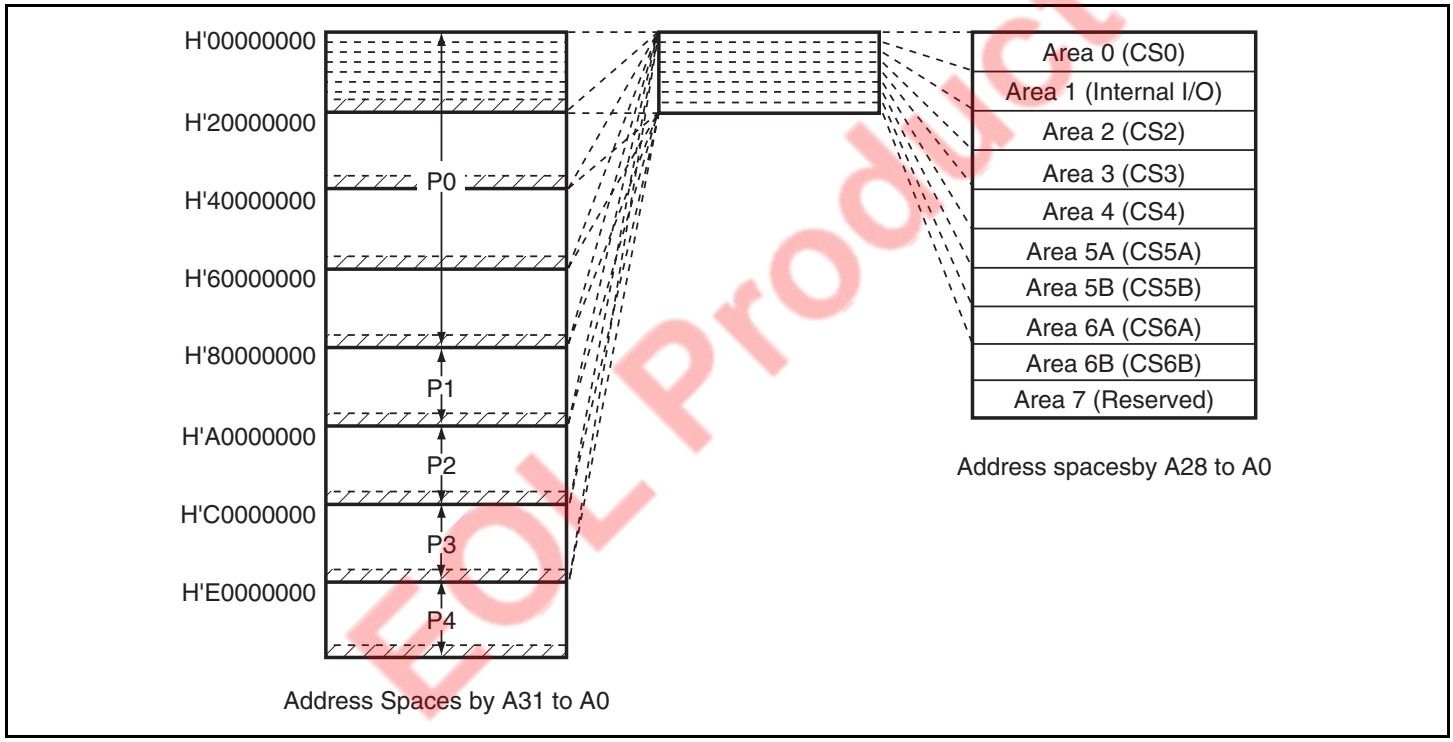


Figure 12.2 Address Space

12.3.3 Address Map

The external address space has a capacity of 384 Mbytes and is used by dividing 8 partial spaces. The kind of memory to be connected and the data bus width are specified in each partial space. The address map for the external address space is listed below.

Table 12.2 Address Space Map 1 (CMNCR.MAP = 0)

Physical Address	Area	Memory to be Connected	Capacity
H'00000000 to H'03FFFFFF	Area 0	Normal memory Burst ROM (asynchronous) Burst ROM (synchronous)	64 Mbytes
H'04000000 to H'07FFFFFF	Area 1	Internal I/O register area*[2]	64 Mbytes
H'08000000 to H'0BFFFFFF	Area 2	Normal memory Byte-selection SRAM SDRAM	64 Mbytes
H'0C000000 to H'0FFFFFFF	Area 3	Normal memory Byte-selection SRAM SDRAM	64 Mbytes
H'10000000 to H'13FFFFFF	Area 4	Normal memory Byte-selection SRAM Burst ROM (asynchronous)	64 Mbytes
H'14000000 to H'15FFFFFF	Area 5A	Normal memory	32 Mbytes
H'16000000 to H'17FFFFFF	Area 5B	Normal memory Byte-selection SRAM MPX-I/O	32 Mbytes
H'18000000 to H'19FFFFFF	Area 6A	Normal memory	32 Mbytes
H'1A000000 to H'1BFFFFFF	Area 6B	Normal memory Byte-selection SRAM MPX-I/O	32 Mbytes
H'1C000000 to H'1FFFFFFF	Area 7	Reserved*[1]	64 Mbytes

Notes: 1. Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed.

2. Access the address indicated in section 24, List of Registers, for the on-chip I/O register in area 1. Do not access area 1 addresses which are not described in the register map. Otherwise, the correct operation cannot be guaranteed.

Table 12.3 Address Space Map 2 (CMNCR.MAP = 1)

Physical Address	Area	Memory to be Connected	Capacity
H'00000000 to H'03FFFFFF	Area 0	Normal memory Burst ROM (Asynchronous) Burst ROM (Synchronous)	64 Mbytes
H'04000000 to H'07FFFFFF	Area 1	Internal I/O register area*3	64 Mbytes
H'08000000 to H'0BFFFFFF	Area 2	Normal memory Byte-selection SRAM SDRAM	64 Mbytes
H'0C000000 to H'0FFFFFFF	Area 3	Normal memory Byte-selection SRAM SDRAM	64 Mbytes
H'10000000 to H'13FFFFFF	Area 4	Normal memory Byte-selection SRAM Burst ROM (Asynchronous)	64 Mbytes
H'14000000 to H'17FFFFFF	Area 5*2	Normal memory Byte-selection SRAM MPX-I/O	64 Mbytes
H'18000000 to H'1BFFFFFF	Area 6*2	Normal memory Byte-selection SRAM Burst MPX-I/O	64 Mbytes
H'1C000000 to H'1FFFFFFF	Area 7	Reserved*1	64 Mbytes

Notes: 1. Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed.

2. For area 5, the CS5BBCR and CS5BWCR registers and the $\overline{\text{CS5B}}$ signal are valid.
   For area 6, the CS6BBCR and CS6BWCR registers and the $\overline{\text{CS6B}}$ signal are valid.

3. Access the address indicated in section 24, List of Registers, for the on-chip I/O register in area 1. Do not access area 1 addresses which are not described in the register map. Otherwise, the correct operation cannot be guaranteed.

12.3.4 Area 0 Memory Type and Memory Bus Width

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

Table 12.4 Correspondence between External Pin MD3 and Bus Width of Area 0

MD3	Bus Width of Area 0
0	16 bits
1	32 bits

12.4 Register Descriptions

The BSC has the following registers. For the addresses and access sizes of these registers, see section 24, List of Registers.

Do not access spaces other than CS0 until the termination of the setting the memory interface.

- Common control register (CMNCR)
- Bus control register for area 0 (CS0BCR)
- Bus control register for area 2 (CS2BCR)
- Bus control register for area 3 (CS3BCR)
- Bus control register for area 4 (CS4BCR)
- Bus control register for area 5A (CS5ABCR)
- Bus control register for area 5B (CS5BBCR)
- Bus control register for area 6A (CS6ABCR)
- Bus control register for area 6B (CS6BBCR)
- Wait control register for area 0 (CS0WCR)
- Wait control register for area 2 (CS2WCR)
- Wait control register for area 3 (CS3WCR)
- Wait control register for area 4 (CS4WCR)
- Wait control register for area 5A (CS5AWCR)
- Wait control register for area 5B (CS5BWCR)
- Wait control register for area 6A (CS6AWCR)
- Wait control register for area 6B (CS6BWCR)
- SDRAM control register (SDCR)

- Refresh timer control/status register (RTCSR)
- Refresh timer counter (RTCNT)
- Refresh time constant register (RTCOR)
- Reset wait counter (RWTCNT)

12.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the common items for each area. This register is only initialized by a power-on reset, and it is not initialized by a manual reset and in the standby mode. Do not access external memory other than area 0 until the CMNCR register initialization is complete.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	WAITSEL	0	R/W	WAIT Signal Sampling Timing Specification Specifies the external WAIT signal sampling timing. 0: Samples the WAIT signal at the falling edge of the CKIO. In this case, the WAIT signal can be input asynchronously. 1: Samples the WAIT signal at the rising edge of the CKIO. In this case, the WAIT signal must be input synchronously.
14, 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	MAP	0	R/W	Space Specification Selects the address map for the external address space. The address maps to be selected are shown in tables 12.2 and 12.3. 0: Selects address map 1. 1: Selects address map 2.

Bit	Bit Name	Initial Value	R/W	Description
11	BLOCK	0	R/W	Bus Clock Specifies whether or not the BREQ signal is received. 0: Receives BREQ. 1: Does not receive BREQ.
10	DPRTY1	0	R/W	DMA Burst Transfer Priority
9	DPRTY0	0	R/W	Specify the priority for a refresh request/bus mastership request during DMA burst transfer. 00: Accepts a refresh request and bus mastership request during DMA burst transfer. 01: Accepts a refresh request but does not accept a bus mastership request during DMA burst transfer. 10: Accepts neither a refresh request nor a bus mastership request during DMA burst transfer. 11: Reserved (setting prohibited)
8	DMAIW2	0	R/W	Wait states between access cycles when DMA single address transfer is performed.
7	DMAIW1	0	R/W	
6	DMAIW0	0	R/W	Specify the number of idle cycles to be inserted after an access to an external device with $\overline{\text{DACK}}$ when DMA single address transfer is performed. The method of inserting idle cycles depends on the contents of DMAIWA. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycled inserted 100: 6 idle cycled inserted 101: 8 idle cycle inserted 110: 10 idle cycles inserted 111: 12 idle cycled inserted

Bit	Bit Name	Initial Value	R/W	Description
5	DMAIWA	0	R/W	Method of inserting wait states between access cycles when DMA single address transfer is performed. Specifies the method of inserting the idle cycles specified by the DMAIW[2:0] bit. Clearing this bit will make this LSI insert the idle cycles when another device, which includes this LSI, drives the data bus after an external device with $\overline{\text{DACK}}$ drove it. However, when the external device with $\overline{\text{DACK}}$ drives the data bus continuously, idle cycles are not inserted. Setting this bit will make this LSI insert the idle cycles after an access to an external device with $\overline{\text{DACK}}$, even when the continuous accesses to an external device with $\overline{\text{DACK}}$ are performed. 0: Idle cycles inserted when another device drives the data bus after an external device with $\overline{\text{DACK}}$ drove it. 1: Idle cycles always inserted after an access to an external device with $\overline{\text{DACK}}$
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	CKD2RDV	0	R	CKIO2 Drive Specifies whether the CKIO2 pin outputs a low level signal or clock (Bϕ). In clock mode 7 (CKIO pin input), the CKIO2 pin has high impedance. The CK2DRV bit setting is enabled in the 2 or 6 clock mode. 0: Outputs a low level signal 1: Outputs a clock (Bϕ)
1	HIZMEM	0	R/W	High-Z Memory Control Specifies the pin state in software standby mode for A25 to A0, $\overline{\text{BS}}$, $\overline{\text{CS}}$, RD/$\overline{\text{WR}}$, $\overline{\text{WE}}$n/DQNxx, $\overline{\text{RD}}$, and $\overline{\text{FRAME}}$. 0: High impedance in standby mode. 1: Driven in standby mode

Bit	Bit Name	Initial Value	R/W	Description
0	HIZCNT	0	R/W	High-Z Control Specifies the state in software standby mode and bus released for CKIO2, $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, and $\overline{\text{CASL}}$. 0: High impedance in software standby mode and bus released for CKIO2, $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, and $\overline{\text{CASL}}$. 1: Driven in standby mode and bus released for CKIO2, $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, and $\overline{\text{CASL}}$.

12.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0, 2, 3, 4, 5A, 5B, 6A, 6B)

CSnBCR is a 32-bit readable/writable register that specifies the function of each area, the number of idle cycles between bus cycles, and the bus-width. This register is initialized to H'36DB0600 by a power-on reset, and it is not initialized by a manual reset and in the standby mode.

Do not access external memory other than area 0 until CSnBCR register initialization is completed.

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30	IWW2	1	R/W	Idle Cycles between Write-Read Cycles and Write-Write Cycles These bits specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycles are the write-read cycle and write-write cycle. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
29	IWW1	1	R/W	
28	IWW0	1	R/W	

Bit	Bit Name	Initial Value	R/W	Description
27	IWRWD2	1	R/W	Idle Cycles for Another Space Read-Write
26	IWRWD1	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycle is a read-write one in which continuous accesses switch between different spaces. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
25	IWRWD0	1	R/W	
24	IWRWS2	1	R/W	Idle Cycles for Read-Write in the Same Space
23	IWRWS1	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-write cycle of which continuous accesses are for the same space. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
22	IWRWS0	1	R/W	

Bit	Bit Name	Initial Value	R/W	Description
21	IWRRD2	1	R/W	Idle Cycles for Read-Read in Another Space
20	WRRD1	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous accesses switch between different spaces.
19	IWRRD0	1	R/W	000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
18	IWRRS2	1	R/W	Idle Cycles for Read-Read in the Same Space
17	IWRRS1	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous accesses are for the same space.
16	IWRRS0	1	R/W	000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14	TYPE2	0	R/W	Specify the type of memory connected to a space.
13	TYPE1	0	R/W	0000: Normal space
12	TYPE0	0	R/W	0001: Burst ROM (clock synchronous)
				0010: MPX-I/O
				0011: Byte-selection SRAM
				0100: SDRAM
				0101: Reserved (Setting prohibited)
				0110: Burst MPX-I/O
				0111: Burst ROM (Clock synchronous)
				For details for memory type in each area, refer to tables 12.2 and 12.3.
11	—	0	R	Reserved
				This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10	BSZ1	1*	R/W	Data Bus Size
9	BSZ0	1*	R/W	Specify the data bus sizes of spaces. The data bus sizes of areas 2, 3, 4 and 5A are shown below. 00: Reserved (setting prohibited) 01: 8-bit size 10: 16-bit size 11: 32-bit size For MPX-I/O, selects bus width by address Notes: 1. If area 5B is specified as MPX-I/O, the bus width can be specified as 8 bits or 16 bits by the address according to the SZSEL bit in CS5BWCR by specifying these bits to 11. 2. The data bus width for area 0 is specified by the external pin. The BSZ1 and BSZ0 bit settings in the CS0BCR register are ignored. 3. If area 6 is specified as burst MPX-I/O, the bus width can be specified as 32 bits only. 4. If area 2 or area 3 is specified as SDRAM space, the bus width can be specified as either 16 bits or 32 bits.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * The CS0CR samples the external pins (MD3) that specify the bus width at power-on reset.

12.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0, 2, 3, 4, 5A, 5B, 6A, 6B)

This register specifies various wait cycles for memory accesses. The bit configuration of this register varies as shown below according to the memory type (TYPE2 to TYPE0) specified by the CSn space bus control register (CSnBCR). Specify the CSnWCR register before accessing the target area. Specify CSnBCR register first, then specify the CSnWCR register.

CSnWCR is initialized to H'00000500 by a power-on reset, and it is not initialized by a manual reset and in the standby mode.

Normal Space, Byte-Selection SRAM, MPX-I/O:

- CS0WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—*	All 0	R/W	Reserved When the normal space interface and SRAM interface with byte selection are specified, these bits should be set to 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WEn}$ Assertion
11	SW0	0	R/W	Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WEn}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of cycles that are necessary for read/write access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	HW1	0	R/W	Delay Cycles from RD, $\overline{WEn}$ Negation to Address, $\overline{CSn}$ Negation Specify the number of delay cycles from RD and $\overline{WEn}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

Note: * To connect the burst ROM to the CS0 area and use the burst ROM interface after the BSC is activated, enables the burst access through bit 20, specifies the number of burst wait cycles through bits 17 and 16, and then set the bits TYPE[2:0] in CS0BCR. Reserved bits other than above should not be set to 1.
For details on the burst ROM interface, see Burst ROM (Clock Asynchronous).

- CS2WCR, CS3WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	Byte-Selection SRAM Byte Access Selection Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{WEn}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of cycles that are necessary for read/write access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- CS4WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	Byte-Selection SRAM Byte Access Selection Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{WEn}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18	WW2	0	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
17	WW1	0	R/W	
16	WW0	0	R/W	
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WE}$ Assertion
11	SW0	0	R/W	Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WE}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10	WR3	1	R/W	Number of Read Access Wait Cycles
9	WR2	0	R/W	Specify the number of cycles that are necessary for read access.
8	WR1	1	R/W	0000: No cycle
7	WR0	0	R/W	0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored

Bit	Bit Name	Initial Value	R/W	Description
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HW1	0	R/W	Delay Cycles from RD, $\overline{WEn}$ Negation to Address, $\overline{CSn}$ Negation Specify the number of delay cycles from RD and $\overline{WEn}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

- CS5AWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18	WW2	0	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
17	WW1	0	R/W	
16	WW0	0	R/W	
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WE}$ Assertion
11	SW0	0	R/W	Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WE}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10	WR3	1	R/W	Number of Read Access Wait Cycles
9	WR2	0	R/W	Specify the number of cycles that are necessary for read access.
8	WR1	1	R/W	0000: No cycle
7	WR0	0	R/W	0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specify whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored

Bit	Bit Name	Initial Value	R/W	Description
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HW1	0	R/W	Delay Cycles from RD, $\overline{WEn}$ Negation to Address, $\overline{CSn}$ Negation Specify the number of delay cycles from RD and $\overline{WEn}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

- CS5BWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	SZSEI	0	R/W	MPX-IO Interface Bus Width Specification Specifies an address to select the bus width when the BSZ[1:0] of CS5BBCR are specified as 11. This bit is valid only when area 5B is specified as MPX-I/O. 0: Selects the bus width by address A14 1: Selects the bus width by address A21 The relationship between the SZSEL bit and bus width selected by A14 or A21 are summarized below.

SZSEL	A14	A21	Bus Width
0	0	Not affected	8 bits
0	0	Not affected	16 bits
1	Not affected	0	8 bits
1	Not affected	1	16 bits

Bit	Bit Name	Initial Value	R/W	Description
20	MPX	0	R/W	MPX-IO Interface Address Wait Specifies the address cycle insertion wait for MPX-IO interface. This bit setting is valid only when area 5B is specified as MPX-I/O. 0: Inserts no wait cycle 1: Inserts 1 wait cycle
	BAS	0	R/W	Byte-Selection SRAM Byte Access Selection This bit setting is valid only when area 5B is specified as byte-selection SRAM. Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{WEn}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18	WW2	0	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
17	WW1	0	R/W	
16	WW0	0	R/W	
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WE}$ Assertion
11	SW0	0	R/W	Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WE}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10	WR3	1	R/W	Number of Read Access Wait Cycles
9	WR2	0	R/W	Specify the number of cycles that are necessary for read access.
8	WR1	1	R/W	0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
7	WR0	0	R/W	
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored

Bit	Bit Name	Initial Value	R/W	Description
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HW1	0	R/W	Delay Cycles from RD, $\overline{WEn}$ Negation to Address, $\overline{CSn}$ Negation Specify the number of delay cycles from RD and $\overline{WEn}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

- CS6AWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WE}$ Assertion Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WE}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
11	SW0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of cycles that are necessary for read/write access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	HW1	0	R/W	Delay Cycles from RD, $\overline{WEn}$ Negation to Address, $\overline{CSn}$ Negation Specify the number of delay cycles from RD and $\overline{WEn}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

- CS6BWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	Byte-Selection SRAM Byte Access Selection Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{WEn}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read/write access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WEn}$ Assertion Specify the number of delay cycles from address, $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WEn}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
11	SW0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of cycles that are necessary for read/write access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
6	WN	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification of this bit is valid even when the number of access wait cycles is 0. 0: The external wait input is valid. 1: The external wait input is ignored.
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CSn}$ Negation Specify the number of delay cycles from $\overline{RD}$, $\overline{WEn}$ negation to address, and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

Burst ROM (Clock Asynchronous):

- CS0WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BEN	0	R/W	Burst Enable Specification Enables or disables 8-burst access for a 16-bit bus width or 16-burst access for an 8-bit bus width during 16-byte access. If this bit is set to 0, 2-burst access is performed four times when the bus width is 16 bits and 4-burst access is performed four times when the bus width is 8 bits. To use a device that does not support 8-burst access or 16-burst access, set this bit to 1. 0: Enables 8-burst access for a 16-bit bus width and 16-burst access for an 8-bit bus width. 1: Disables 8-burst access for a 16-bit bus width and 16-burst access for an 8-bit bus width.
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
17	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted between the second or later access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	W3	1	R/W	Number of Access Wait Cycles
9	W2	0	R/W	Specify the number of wait cycles to be inserted in the first access cycle.
8	W1	1	R/W	0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
7	W0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- CS4WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BEN	0	R/W	Burst Enable Specification Enables or disables 8-burst access for a 16-bit bus width or 16- burst access for an 8-bit bus width during 16-byte access. If this bit is set to 0, 2-burst access is performed four times when the bus width is 16 bits and 4-burst access is performed four times when the bus width is 8 bits. To use a device that does not support 8-burst access or 16-burst access, set this bit to 1. 0: Enables 8-burst access for a 16-bit bus width and 16-burst access for an 8-bit bus width. 1: Disables 8-burst access for a 16-bit bus width and 16-burst access for an 8-bit bus width.
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
17	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted between the second or later access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WE}$ Assertion Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WE}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
11	SW0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
10	W3	1	R/W	Number of Access Wait Cycles
9	W2	0	R/W	Specify the number of wait cycles to be inserted in the first access cycle.
8	W1	1	R/W	
7	W0	0	R/W	0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	HW1	0	R/W	Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CSn}$ Negation
0	HW0	0	R/W	Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

SDRAM*:

- CS2WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	A2CL1	1	R/W	CAS Latency for Area 2
7	A2CL0	0	R/W	Specify the CAS latency for area 2. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- CS3WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14	WTRP1*	0	R/W	Number of Auto-Precharge Completion Wait Cycles
13	WTRP0	0	R/W	Specify the number of minimum precharge completion wait cycles during the periods shown below. • From the start of auto-precharge to issuing of the ACTV command for the same bank • From issuing of the PRE/PALL command to issuing of the ACTV command for the same bank • Until entering the power-down mode or deep power-down mode • From issuing of the PALL command to issuing of the REF command in auto refreshing • From issuing of the PALL command to issuing of the SELF command in self refreshing The setting for areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11	WTRCD1	0	R/W	Number of Wait Cycles between ACTV Command and READ(A)/WRIT(A) Command
10	WTRCD0	1	R/W	Specify the minimum number of wait cycles from issuing the ACTV command to issuing the READ(A)/WRIT(A) command. The setting for areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

Bit	Bit Name	Initial Value	R/W	Description
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	A3CL1	1	R/W	CAS Latency for Area 3
7	A3CL0	0	R/W	Specify the CAS latency for area 3. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TRWL1*	0	R/W	Number of Auto-Precharge Startup Wait Cycles
3	TRWL0	0	R/W	Specify the number of minimum precharge startup wait cycles during the periods shown below. • From issuing of the WRITA command by this LSI to starting of auto-precharge in SDRAM The number of cycles from issuing the WRITA command to issuing the ACTV command for the same bank. See the SDRAM data sheets to confirm the number of cycles precede issuing of auto-precharge after the SDRAM has received the WRITA command. Set these bits so that the confirmed cycles should be equal to or less than the cycles specified by these bits. • From issuing of the WRIT command by this LSI to issuing of the PRE command When different row addresses are accessed from the same bank address in bank-active mode The setting for areas 2 and 3 is common. 00: No cycle (Initial value) 01: 1 cycle 10: 2 cycles 11: 3 cycles

Bit	Bit Name	Initial Value	R/W	Description
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	WTRC1*	0	R/W	Number of Idle Cycles from REF Command/Self-Refresh Release to ACTV/REF/MRS Command Specify the number of minimum idle cycles during the periods shown below. • From issuing of the REF command to issuing of the ACTV/REF/MRS command • From releasing self-refresh to issuing of the ACTV/REF/MRS command The setting for areas 2 and 3 is common. 00: 2 cycles (Initial value) 01: 3 cycles 10: 5 cycles 11: 8 cycles
0	WTRC0	0	R/W	

Note: * If both areas 2 and 3 are specified as SDRAM, WTRP[1:0], WTRCD[1:0], TRWL[1:0], and WTRC[1:0] bit settings are common. If only one area is connected to the SDRAM, specify area 3. In this case, specify area 2 as normal space or byte-selection SRAM.

Burst MPX-IO:

- CS6BWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	MPXAW1	0	R/W	Number of Address Cycle Waits Specify the number of waits to be inserted in the address cycle. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
20	MPXAW0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
19	MPXMD	0	R/W	Burst MPX-IO Interface Mode Specification Specify the access mode in 16-byte access 0: One 4-burst access by 16-byte transfer 1: Two 2-bursts accesses by quad word (8-byte) transfer Transfer size when MPXMD = 0 D31 D30 D29 : Transfer Size 0 0 0 : Byte (1 byte) 0 0 1 : Word (2 byte) 0 1 0 : Longword (4 bytes) 0 1 1 : Reserved (quad word) (8 bytes) 1 0 0 : 16 bytes 1 0 1 : Reserved (32 bytes) 1 1 0 : Reserved (64 bytes) Transfer size when MPXMD = 1 D31 D30 D29 : Transfer Size 0 0 0 : Byte (1 byte) 0 0 1 : Word (2 byte) 0 1 0 : Longword (4 bytes) 0 1 1 : Quad word (8 bytes) 1 0 0 : Reserved (32 bytes)
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
17	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	1	R/W	Specify the number of wait cycles to be inserted at the 2nd and the subsequent access cycles in burst access 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	W3	1	R/W	Number of Access Wait Cycles
9	W2	0	R/W	Specify the number of wait cycles to be inserted in the first access cycle.
8	W1	1	R/W	
7	W0	0	R/W	0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Burst ROM (Clock Synchronous):

- CS0WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted between the second or later access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10	W3	1	R/W	Number of Access Wait Cycles
9	W2	0	R/W	Specify the number of wait cycles to be inserted in the first access cycle.
8	W1	1	R/W	
7	W0	0	R/W	0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

12.4.4 SDRAM Control Register (SDCR)

SDCR specifies the method to refresh and access SDRAM, and the types of SDRAMs to be connected.

This register is initialized to H'00000000 by a power-on reset, and it is not initialized by a manual reset and in the standby mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	A2ROW1	0	R/W	Number of Bits of Row Address for Area 2
19	A2ROW0	0	R/W	Specify the number of bits of row address for area 2. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (Setting prohibited)
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
17	A2COL1	0	R/W	Number of Bits of Column Address for Area 2
16	A2COL0	0	R/W	Specify the number of bits of column address for area 2. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (Setting prohibited)
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
13	DEEP	0	R/W	Deep Power-Down Mode This bit is valid for low-power SDRAM. If the RFSH or RMODE bit is set to 1 while this bit is set to 1, the deep power-down entry command is issued and the low-power SDRAM enters the deep power-down mode. 0: Self-refresh mode 1: Deep power-down mode
12	SLOW	0	R/W	Low-Frequency Mode Specifies the output timing of command, address, and write data for SDRAM and the latch timing of read data from SDRAM. Setting this bit makes the hold time for command, address, write and read data extended for half cycle (output or read at the falling edge of CKIO). This mode is suitable for SDRAM with low-frequency clock. 0: Command, address, and write data for SDRAM is output at the rising edge of CKIO. Read data from SDRAM is latched at the rising edge of CKIO. 1: Command, address, and write data for SDRAM is output at the falling edge of CKIO. Read data from SDRAM is latched at the falling edge of CKIO.
11	RFSH	0	R/W	Refresh Control Specifies whether or not the refresh operation of the SDRAM is performed. 0: No refresh 1: Refresh
10	RMODE	0	R/W	Refresh Control Specifies whether to perform auto-refresh or self-refresh when the RFSH bit is 1. When the RFSH bit is 1 and this bit is 1, self-refresh starts immediately. When the RFSH bit is 1 and this bit is 0, auto-refresh starts according to the contents that are set in registers RTCSR, RTCNT, and RTCOR. 0: Auto-refresh is performed 1: Self-refresh is performed

Bit	Bit Name	Initial Value	R/W	Description
9	PDOWN	0	R	Power-Down Mode Specifies whether the SDRAM will enter the power-down mode or not after the access to the external memory other than the SDRAM or to the internal I/O resister. With this bit being set to 1, the access to the external memory other than the SDRAM or to the internal I/O register drives the CKE signal low and causes the SDRAM to enter the power-down mode. 0: The SDRAM does not enter the power-down mode. 1: The SDRAM enters the power-down mode after the access to the external memory other than the SDRAM or to the internal I/O resister.
8	BACTV	0	R/W	Bank Active Mode Specifies to access whether in auto-precharge mode (using READA and WRITA commands) or in bank active mode (using READ and WRIT commands). 0: Auto-precharge mode (using READA and WRITA commands) 1: Bank active mode (using READ and WRIT commands) Note: Bank active mode can be used only when either the upper or lower bits of the CS3 space are used. When both the CS2 and CS3 spaces are set to SDRAM, specify the auto-precharge mode.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	A3ROW1	0	R/W	Number of Bits of Row Address for Area 3
3	A3ROW0	0	R/W	Specify the number of bits of the row address for area 3. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (Setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	A3COL1	0	R/W	Number of Bits of Column Address for Area 3
0	A3COL0	0	R/W	Specify the number of bits of the column address for area 3. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (Setting prohibited)

12.4.5 Refresh Timer Control/Status Register (RTCSR)

RTCSR specifies various items about refresh for SDRAM. This register is initialized to H'00000000 by a power-on reset, and it is not initialized by a manual reset and in the standby mode. When the RTCSR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

The clock which counts up the refresh timer counter (RTCNT) is adjusted its phase only by a power-on reset. Thus, when CKS[2:0] are set to other than B'000 and a timer is in operation, an error is found until the first compare match flag is set.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7	CMF	0	R/W	Compare Match Flag Indicates that a compare match occurs between the refresh timer counter (RTCNT) and refresh time constant register (RTCOR). This bit is set or cleared in the following conditions. 0: Clearing condition: When 0 is written in CMF after reading out RTCSR during CMF = 1. 1: Setting condition: When the condition RTCNT = RTCOR is satisfied.

Bit	Bit Name	Initial Value	R/W	Description
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	CKS2	0	R/W	Clock Select
4	CKS1	0	R/W	Select the clock input to count-up the refresh timer counter (RTCNT).
3	CKS0	0	R/W	000: Stop the counting-up 001: Bϕ/4 010: Bϕ/16 011: Bϕ/64 100: Bϕ/256 101: Bϕ/1024 110: Bϕ/2048 111: Bϕ/4096
2	RRC2	0	R/W	Refresh Count
1	RRC1	0	R/W	Specify the number of continuous refresh cycles, when the refresh request occurs after the coincidence of the values of the refresh timer counter (RTCNT) and the refresh time constant register (RTCOR). These bits can make the period of occurrence of refresh long.
0	RRC0	0	R/W	000: Once 001: Twice 010: 4 times 011: 6 times 100: 8 times 101: Reserved (Setting prohibited) 110: Reserved (Setting prohibited) 111: Reserved (Setting prohibited)

12.4.6 Refresh Timer Counter (RTCNT)

RTCNT is an 8-bit counter that increments using the clock selected by bits CKS2 to CKS0 in RTCSR. When RTCNT matches RTCOR, RTCNT is cleared to 0. The value in RTCNT returns to 0 after counting up to 255. When the RTCNT is written, the upper 16 bits of the write data must be H'A55A to cancel write protection. This counter is initialized to H'00000000 by a power-on reset, and it is not initialized by a manual reset and in the standby mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0	—	All 0	R/W	8-Bit Counter

12.4.7 Refresh Time Constant Register (RTCOR)

RTCOR is an 8-bit register. When RTCOR matches RTCNT, the CMF bit in RTCSR is set to 1 and RTCNT is cleared to 0.

When the RFSH bit in SDCR is 1, a memory refresh request is issued by this matching signal. This request is maintained until the refresh operation is performed. If the request is not processed when the next matching occurs, the previous request is ignored.

When the RTCOR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection. This register is initialized to H'00000000 by a power-on reset, and it is not initialized by a manual reset and in the standby mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0	—	All 0	R/W	8-Bit Counter

12.4.8 Reset Wait Counter (RWTCNT)

RWTCNT is a 7-bit counter. This counter starts to increment by synchronizing the CKIO after a power-on reset is released, and stops when the value reaches H'007F. External bus access is suspended while the counter is operating. This counter is provided to minimize the time from releasing a reset for flash memory to the first access.

If a value is written to the lower seven bits of this register, the counter starts to increment from the specified value and the external bus access is suspended until the incrementing has been completed. When the RWTCNT is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved These bits are always read as 0.
6 to 0	—	All 0	R/W	7-Bit Counter

12.5 Operating Description

12.5.1 Endian/Access Size and Data Alignment

This LSI supports big endian, in which the 0 address is the most significant byte (MSByte) in the byte data.

Three data bus widths (8 bits, 16 bits, and 32 bits) are available for normal memory and byte-selection SRAM. Two data bus width (16 bits and 32 bits) are available for SDRAM. Data bus width for MPX-IO is fixed to 32 bits. Data alignment is performed in accordance with the data bus width of the device. This also means that when longword data is read from a byte-width device, the read operation must be done four times. In this LSI, data alignment and conversion of data length is performed automatically between the respective interfaces.

Table 12.5 through 12.7 show the relationship between device data width and access unit.

Table 12.5 32-Bit External Device Access and Data Alignment

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

Table 12.6 16-Bit External Device Access and Data Alignment

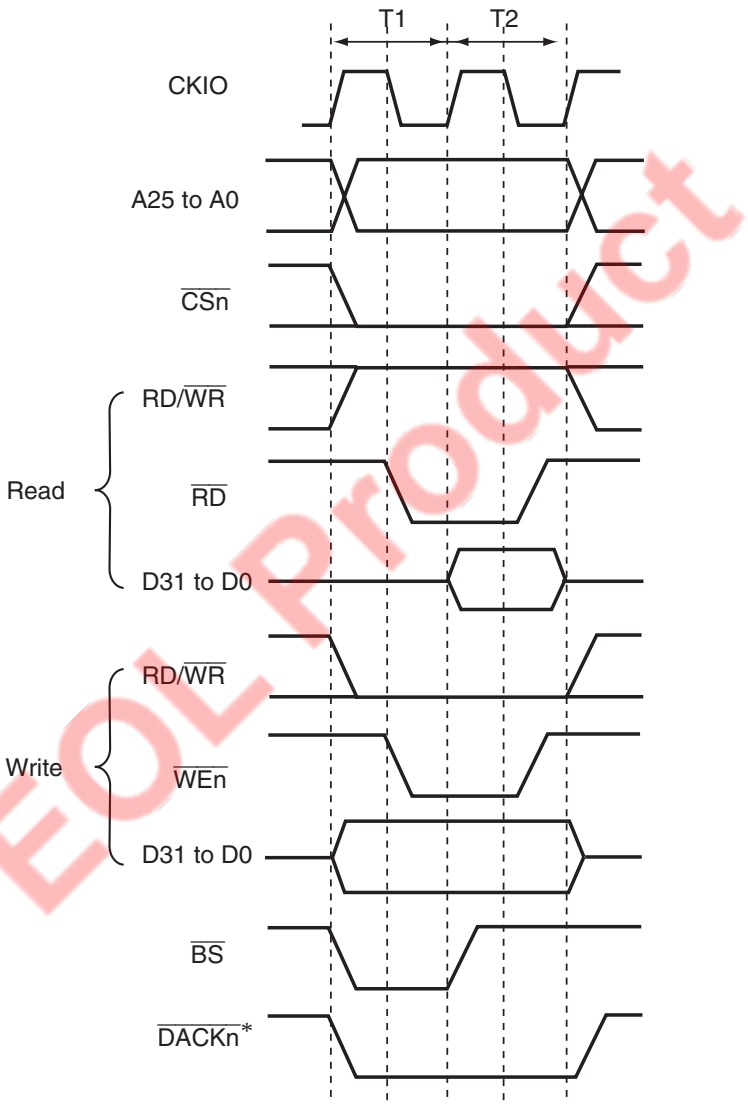
Operation		Data Bus				Strobe Signals			
		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$, DQMUU	$\overline{WE2}$, DQMUL	$\overline{WE1}$, DQMLU	$\overline{WE0}$, DQMLL
Byte access at 0		—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 1		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 2		—	—	Data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

Table 12.7 8-Bit External Device Access and Data Alignment

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

12.5.2 Normal Space Interface

Basic Timing: For access to a normal space, this LSI uses strobe signal output in consideration of the fact that mainly static RAM will be directly connected. When using SRAM with a byte-selection pin, see section 12.5.8, Byte-Selection SRAM Interface. Figure 12.3 shows the basic timings of normal space access. A no-wait normal access is completed in two cycles. The $\overline{BS}$ signal is asserted for one cycle to indicate the start of a bus cycle.



Note: * The waveform for $\overline{DACKn}$ is when active low is specified.

Figure 12.3 Normal Space Basic Access Timing (Access Wait 0)

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 32 bits are always read in case of a 32-bit device, and 16 bits in case of a 16-bit device. When writing, only the $\overline{WEn}$ signal for the byte to be written is asserted.

It is necessary to output the data that has been read using $\overline{RD}$ when a buffer is established in the data bus. The RD/$\overline{WR}$ signal is in a read state (high output) when no access has been carried out. Therefore, care must be taken when controlling the external data buffer, to avoid collision.

Figures 12.4 and 12.5 show the basic timings of normal space accesses. If the WM bit in CSnWCR is cleared to 0, a Tnop cycle is inserted after the CSn space access to evaluate the external wait (figure 12.4). If the WM bit in CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (figure 12.5).

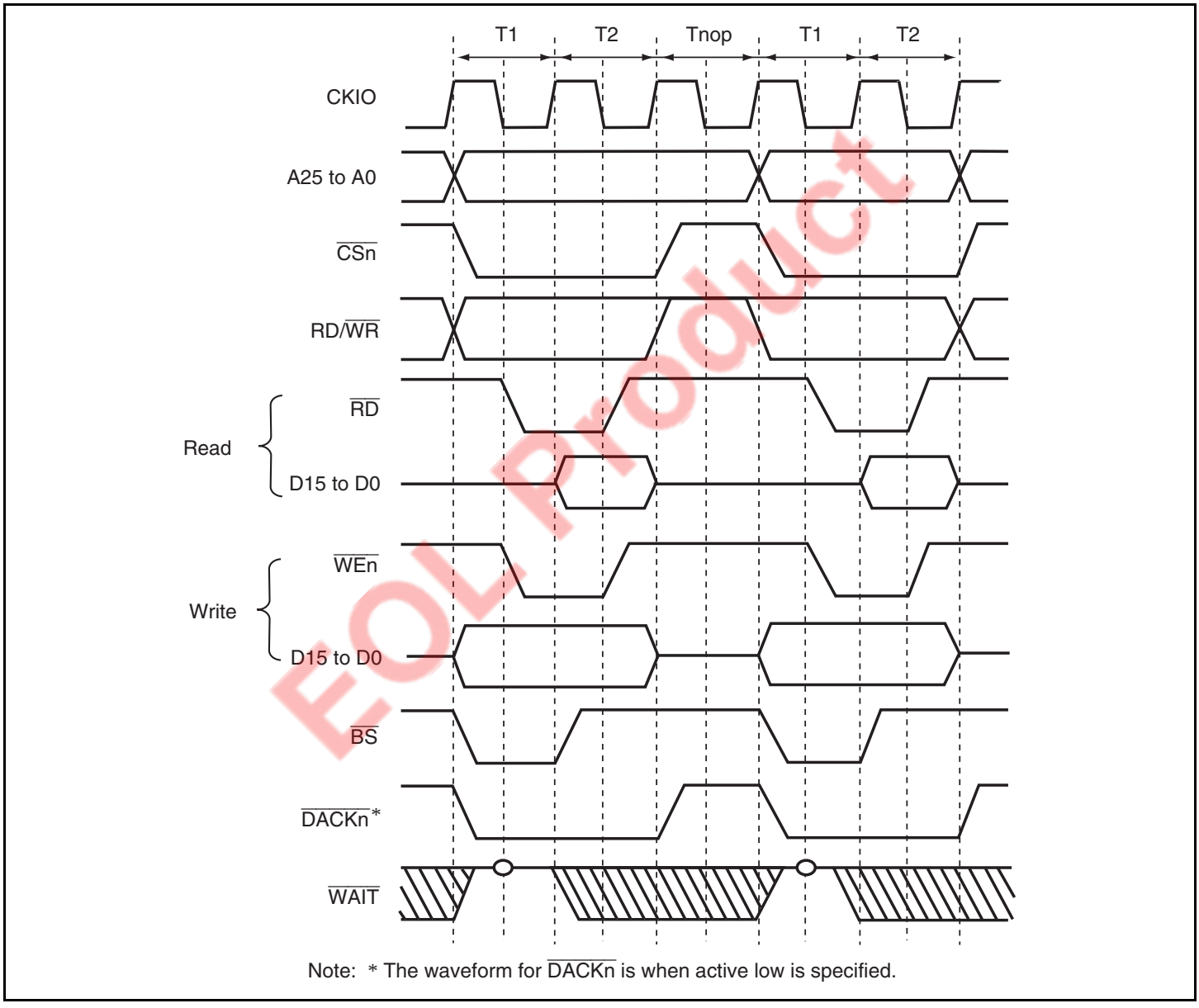


Figure 12.4 Continuous Access for Normal Space 1
Bus Width = 16 Bits, Longword Access, CSnWCR.WN Bit = 0
(Access Wait = 0, Cycle Wait = 0)

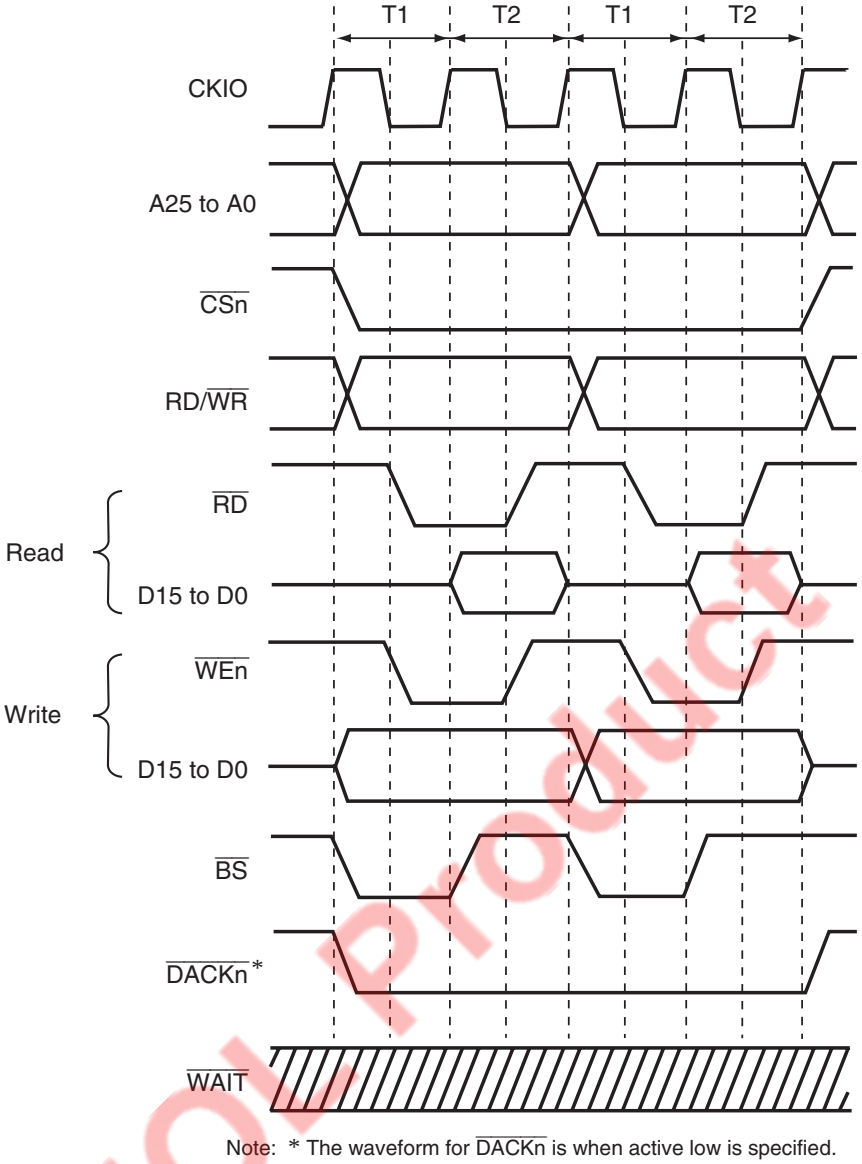


Figure 12.5 Continuous Access for Normal Space 2
Bus Width = 16 Bits, Longword Access, CSnWCR.WN Bit = 1
(Access Wait = 0, Cycle Wait = 0)

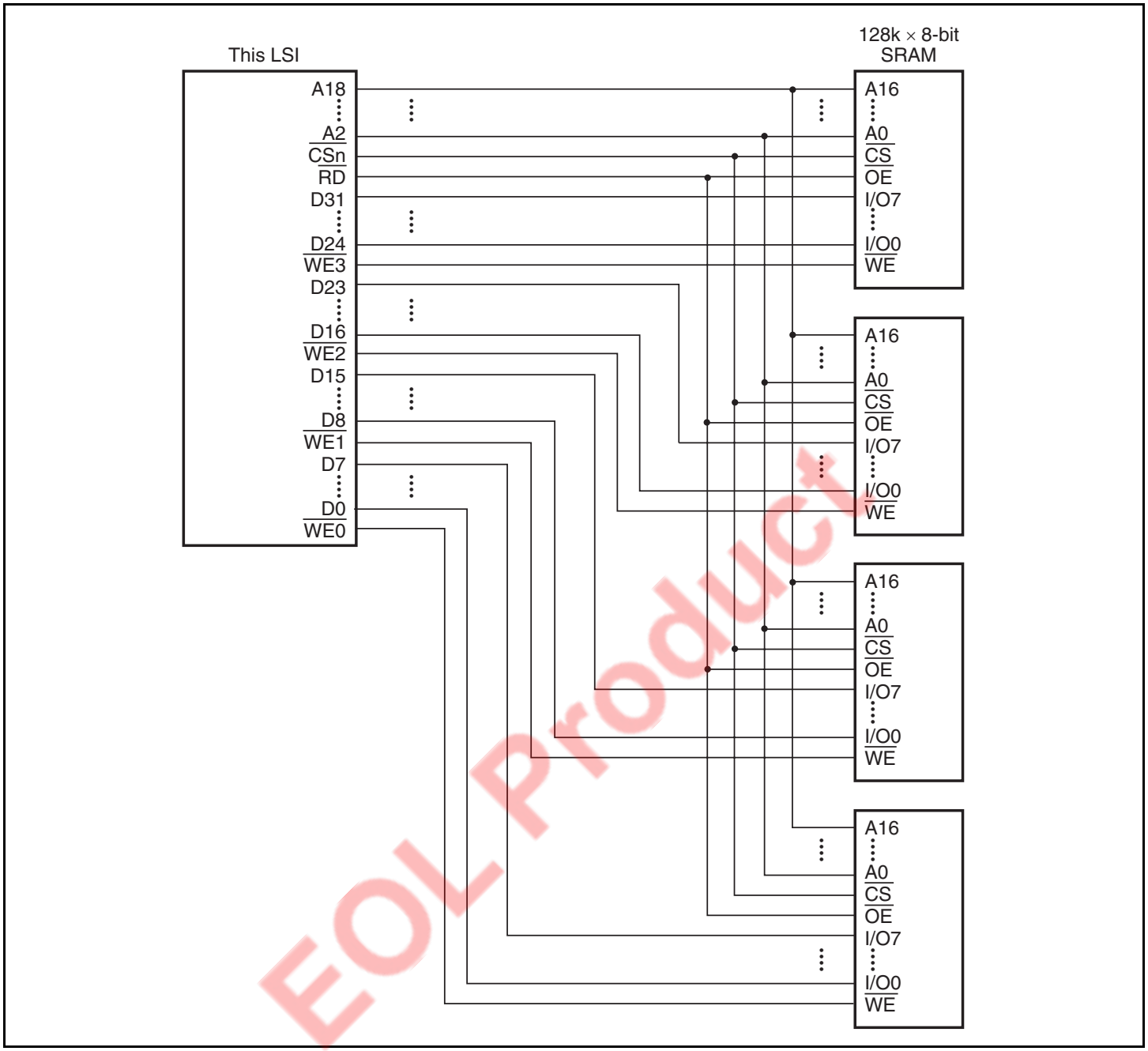


Figure 12.6 Example of 32-Bit Data-Width SRAM Connection

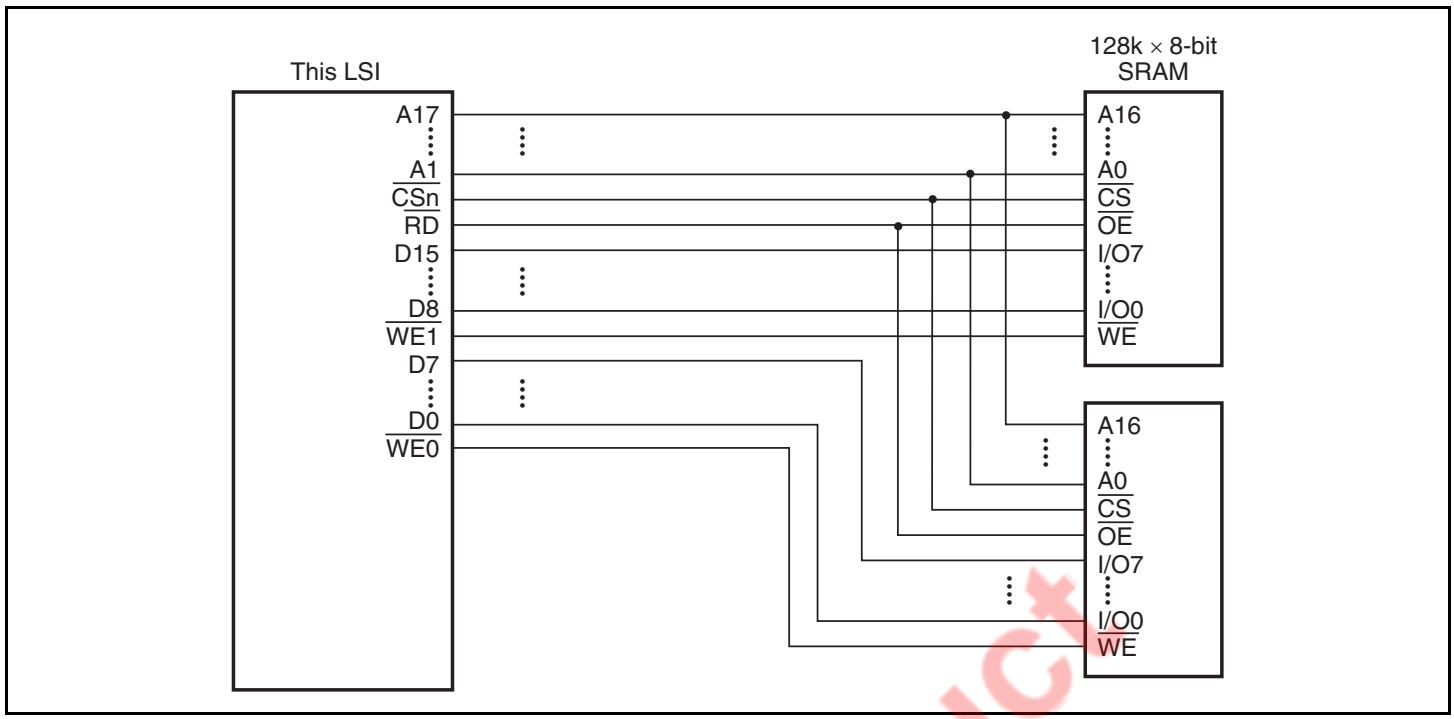


Figure 12.7 Example of 16-Bit Data-Width SRAM Connection

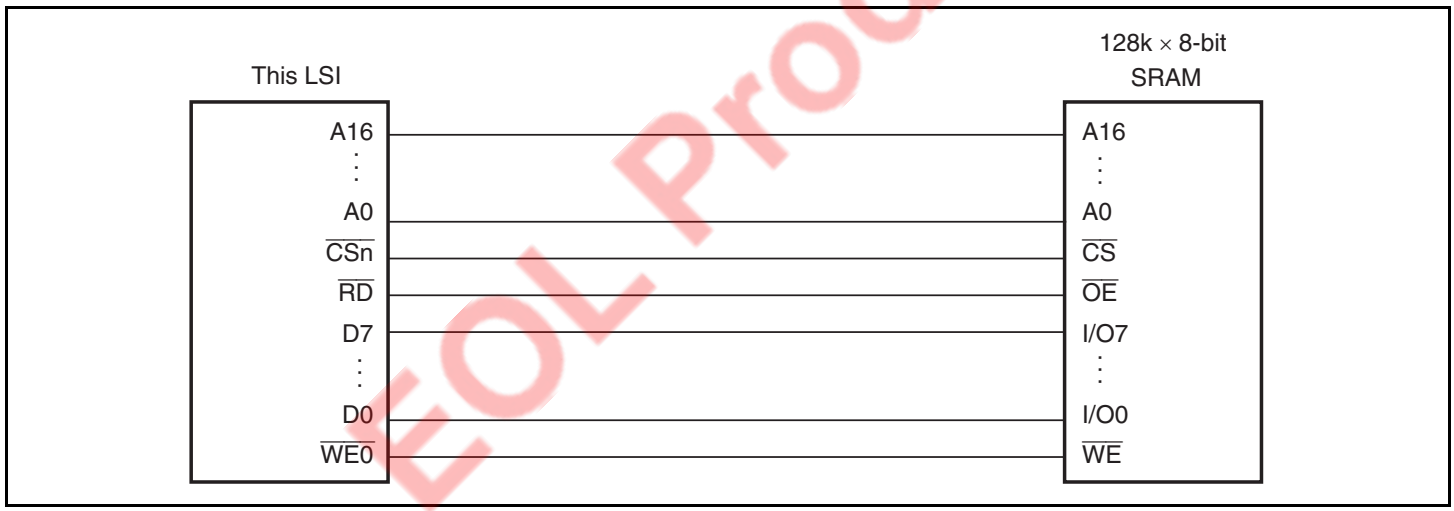
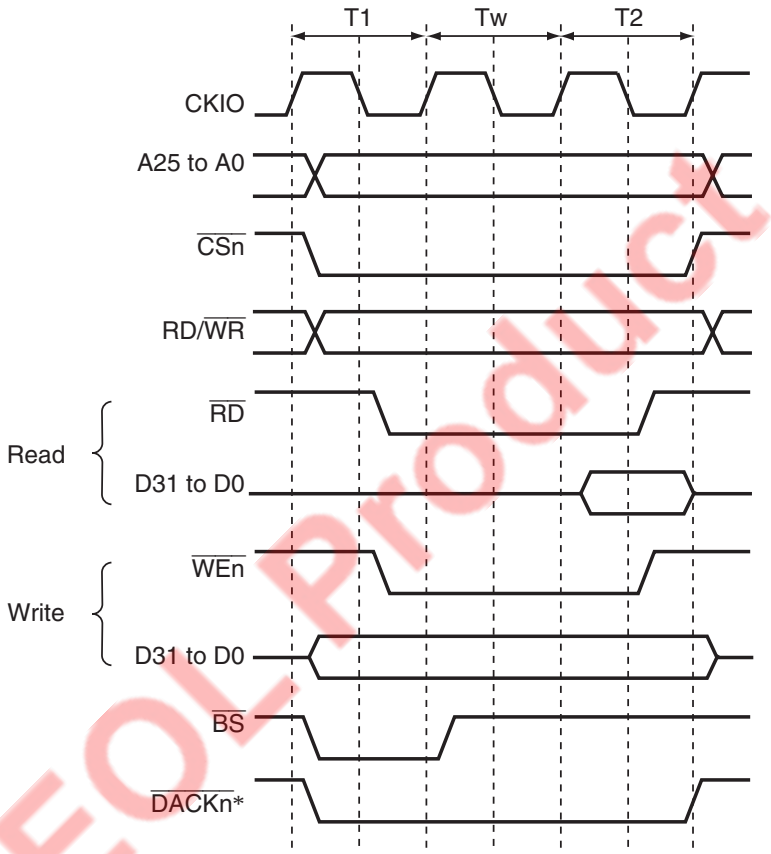


Figure 12.8 Example of 8-Bit Data-Width SRAM Connection

12.5.3 Access Wait Control

Wait cycle insertion on a normal space access can be controlled by the settings of bits WR3 to WR0 in CSnWCR. It is possible for areas 4, 5A, and 5B to insert wait cycles independently in read access and in write access. The areas other than 4, 5A, and 5B have common access wait for read cycle and write cycle. The specified number of Tw cycles are inserted as wait cycles in a normal space access shown in figure 12.9.



Note: * The waveform for DACKn is when active low is specified.

Figure 12.9 Wait Timing for Normal Space Access (Software Wait Only)

When the WM bit in CSnWCR is cleared to 0, the external wait input $\overline{\text{WAIT}}$ signal is also sampled. $\overline{\text{WAIT}}$ pin sampling is shown in figure 12.10. A 2-cycle wait is specified as a software wait. The $\overline{\text{WAIT}}$ signal is sampled on the falling edge of CKIO at the transition from the T1 or Tw cycle to the T2 cycle.

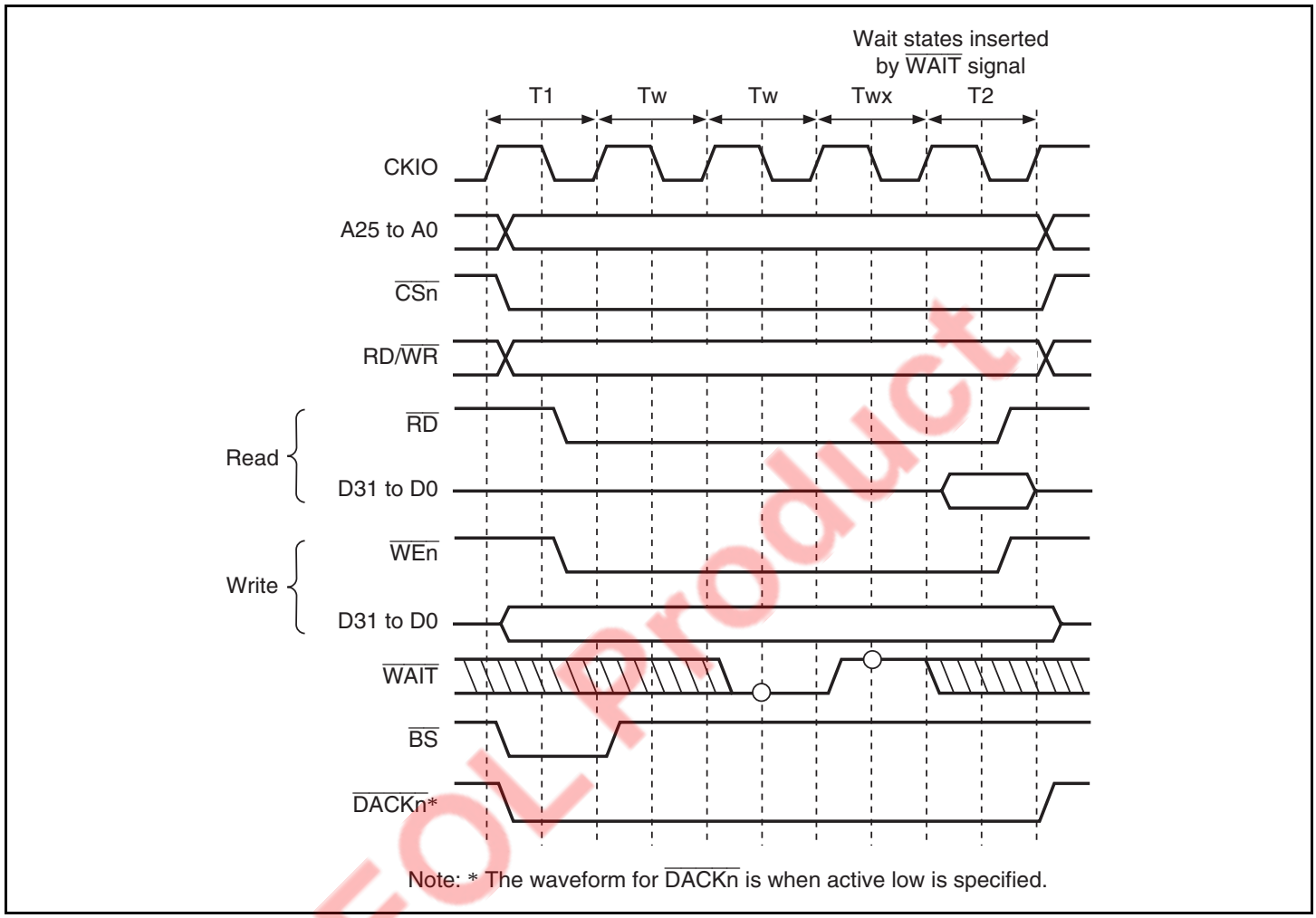
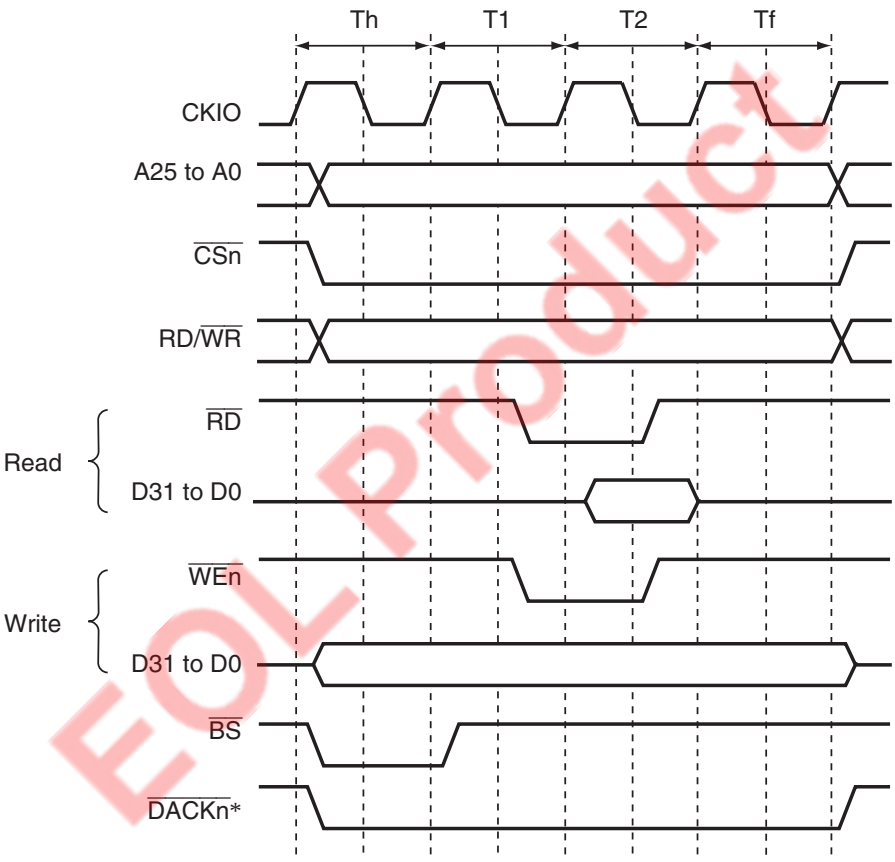


Figure 12.10 Wait State Timing for Normal Space Access (Wait State Insertion Using $\overline{\text{WAIT}}$ Signal)

12.5.4 $\overline{\text{CSn}}$ Assert Period Expansion

The number of cycles from $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from $\overline{\text{RD}}$, $\overline{\text{WEn}}$ negation to $\overline{\text{CSn}}$ negation can be specified by setting bits HW1 and HW0. Therefore, a flexible interface to an external device can be obtained. Figure 12.11 shows an example. A Th cycle and a Tf cycle are added before and after an ordinary cycle, respectively. In these cycles, $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ are not asserted, while other signals are asserted. The data output is prolonged to the Tf cycle, and this prolongation is useful for devices with slow writing operations.



Note: * The waveform for $\overline{\text{DACKn}}$ is when active low is specified.

Figure 12.11 $\overline{\text{CSn}}$ Assert Period Expansion

12.5.5 MPX-I/O Interface

Access timing for the MPX space is shown below. In the MPX space, $\overline{CS5B}$, $\overline{AH}$, $\overline{RD}$, and $\overline{WEn}$ signals control the accessing. The basic access for the MPX space consists of 2 cycles of address output followed by an access to a normal space. The bus width for the address output cycle or the data input/output cycle is fixed to 8 bits or 16 bits. Alternatively, it can be 8 bits or 16 bits depending on the address to be accessed.

Output of the addresses D15 to D0 or D7 to D0 is performed from cycle Ta2 to cycle Ta3. Because cycle Ta1 has a high-impedance state, collisions of addresses and data can be avoided without inserting idle cycles, even in continuous accesses. Address output is increased to 3 cycles by setting the MPXW bit in the CS5BWCR register to 1. The RD/$\overline{WR}$ signal is output at the same time as the $\overline{CS5B}$ signal; it is high in the read cycle and low in the write cycle.

The data cycle is the same as that in a normal space access.

Timing charts are shown in figures 12.12 to 12.14.

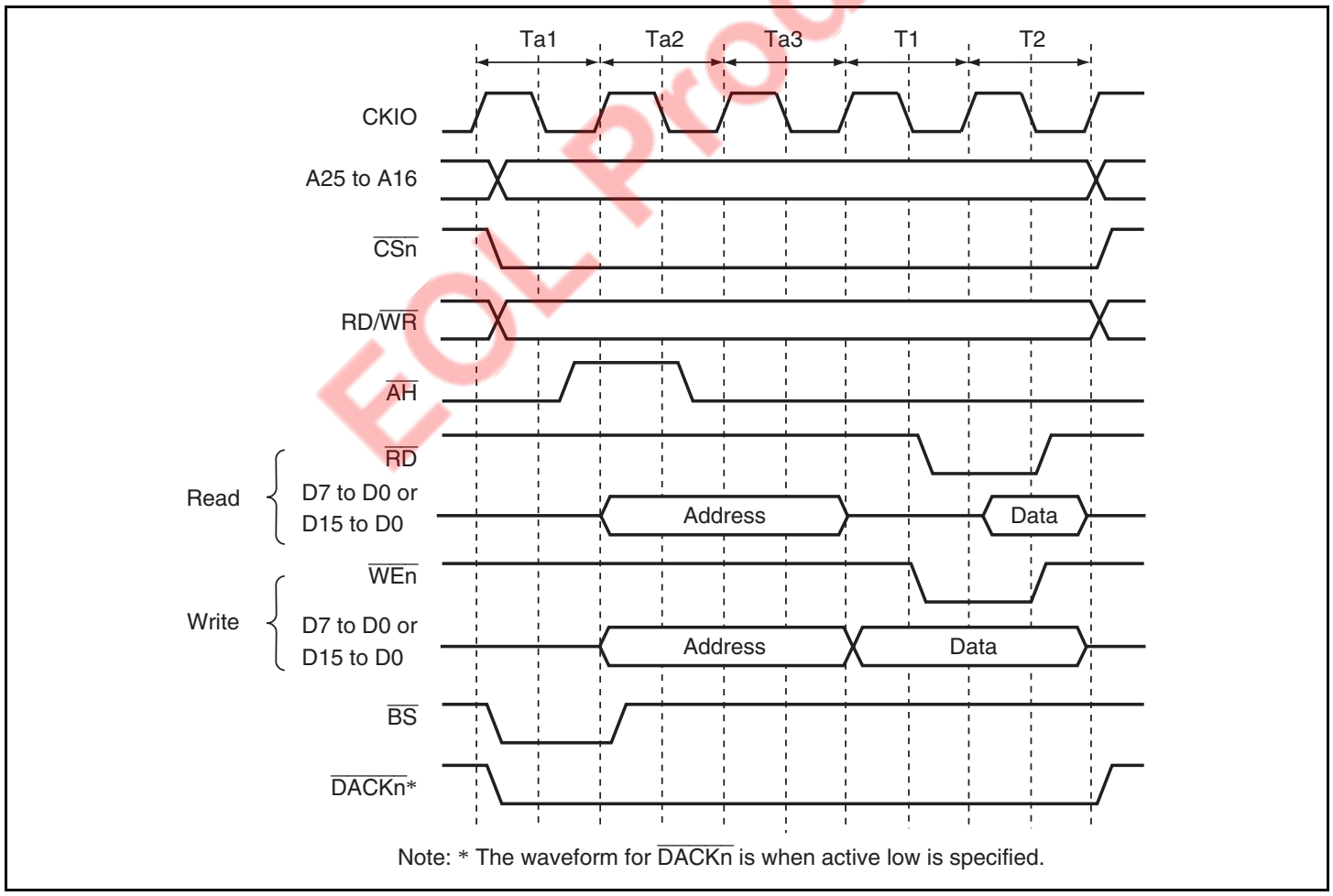


Figure 12.12 Access Timing for MPX Space (Address Cycle No Wait, Data Cycle No Wait)

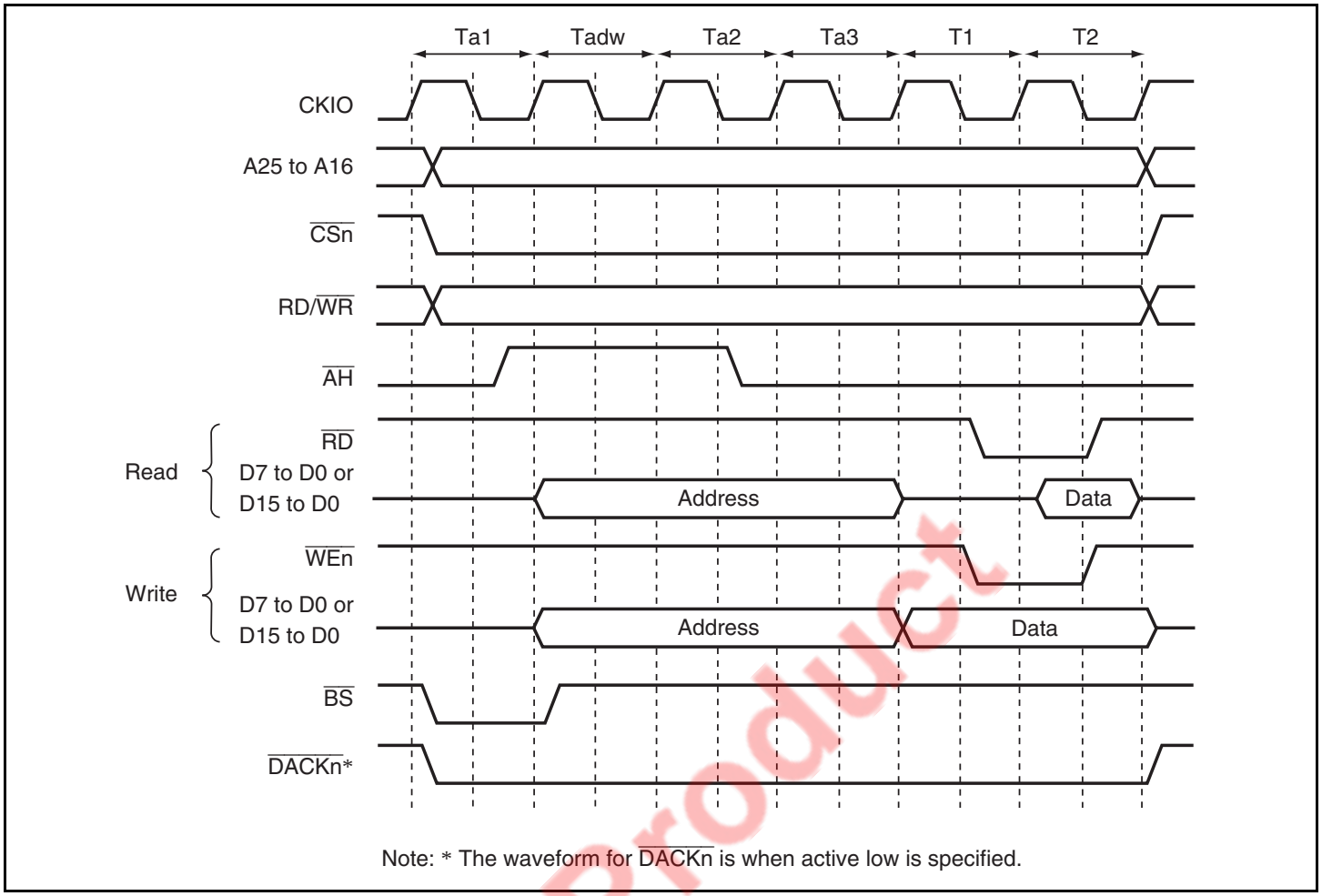


Figure 12.13 Access Timing for MPX Space (Address Cycle Wait 1, Data Cycle No Wait)

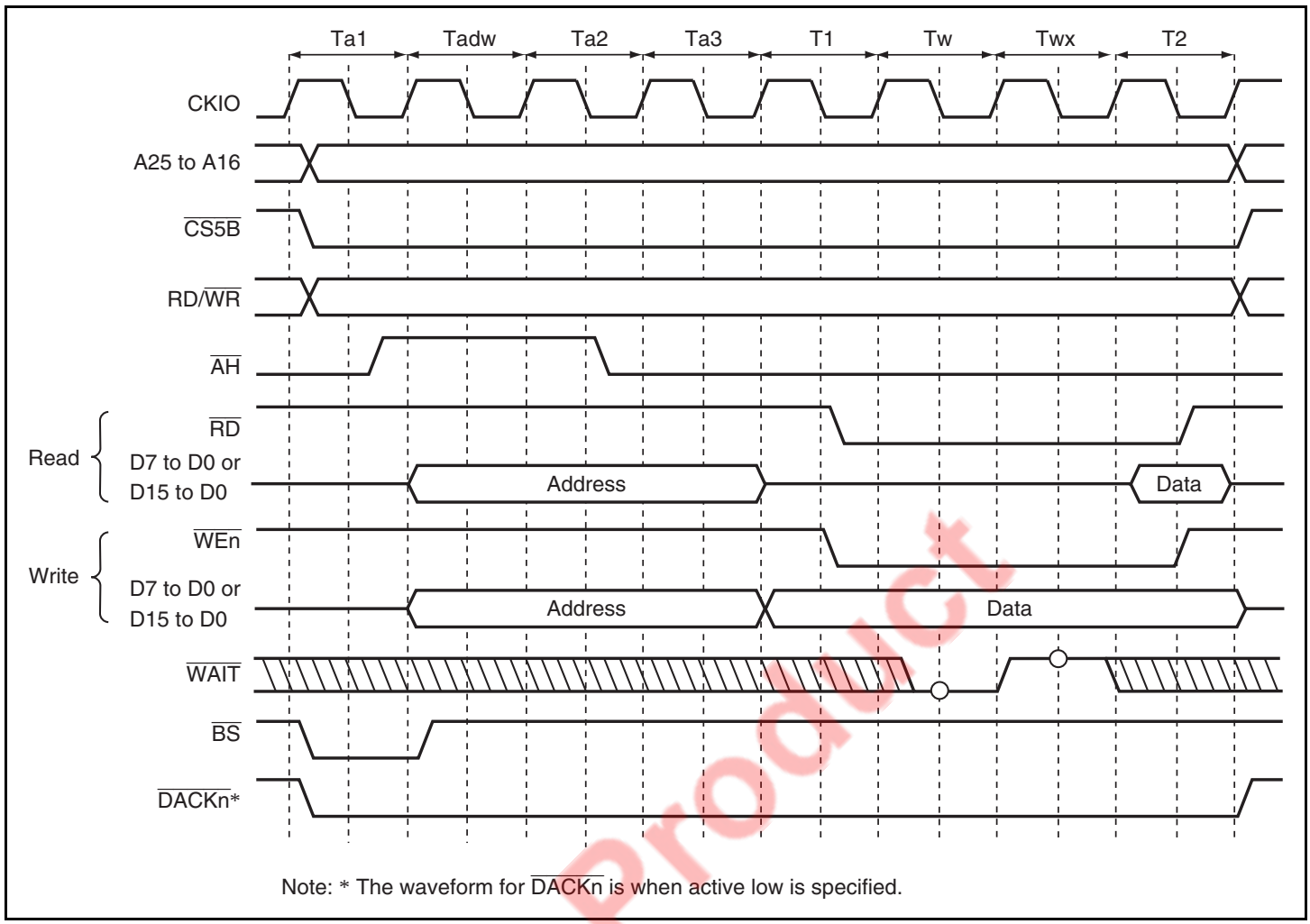


Figure 12.14 Access Timing for MPX Space (Address Cycle Access Wait 1, Data Cycle Wait 1, External Wait 1)

12.5.6 SDRAM Interface

SDRAM Direct Connection: The SDRAM that can be connected to this LSI is a product that has 11/12/13 bits of row address, 8/9/10 bits of column address, 4 or less banks, and uses the A10 pin for setting precharge mode in read and write command cycles. The control signals for direct connection of SDRAM are $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$, RD/$\overline{\text{WR}}$, DQMUU, DQMUL, DQMLU, DQMLL, CKE, $\overline{\text{CS2}}$, and $\overline{\text{CS3}}$. All the signals other than $\overline{\text{CS2}}$ and $\overline{\text{CS3}}$ are common to all areas, and signals other than CKE are valid when $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$ is asserted. SDRAM can be connected to up to 2 spaces. The data bus width of the area that is connected to SDRAM can be set to 32 or 16 bits.

Burst read/single write (burst length 1) and burst read/burst write (burst length 1) are supported as the SDRAM operating mode.

Commands for SDRAM can be specified by $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$, RD/$\overline{\text{WR}}$, and specific address signals. These commands supports:

- NOP
- Auto-refresh (REF)
- Self-refresh (SELF)
- All banks pre-charge (PALL)
- Specified bank pre-charge (PRE)
- Bank active (ACTV)
- Read (READ)
- Read with pre-charge (READA)
- Write (WRIT)
- Write with pre-charge (WRITA)
- Write mode register (MRS)
- EMRS

The byte to be accessed is specified by DQMUU, DQMUL, DQMLU, and DQMLL. Reading or writing is performed for a byte whose corresponding DQMxx is low. For details on the relationship between DQMxx and the byte to be accessed, refer to section 12.5.1, Endian/Access Size and Data Alignment.

Figures 12.15 to 12.17 show examples of the connection of the SDRAM with the LSI.

As shown in figure 12.17, two sets of SDRAMs of 32 Mbytes or smaller can be connected to the same CS space by using $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, and $\overline{\text{CASL}}$. In this case, a total of 8 banks are assigned to the same CS space: 4 banks specified by $\overline{\text{RASL}}$ and $\overline{\text{CASL}}$, and 4 banks specified by $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$. When accessing the address with A25 = 0, $\overline{\text{RASL}}$ and $\overline{\text{CASL}}$ are asserted. When accessing the address with A25 = 1, $\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are asserted.

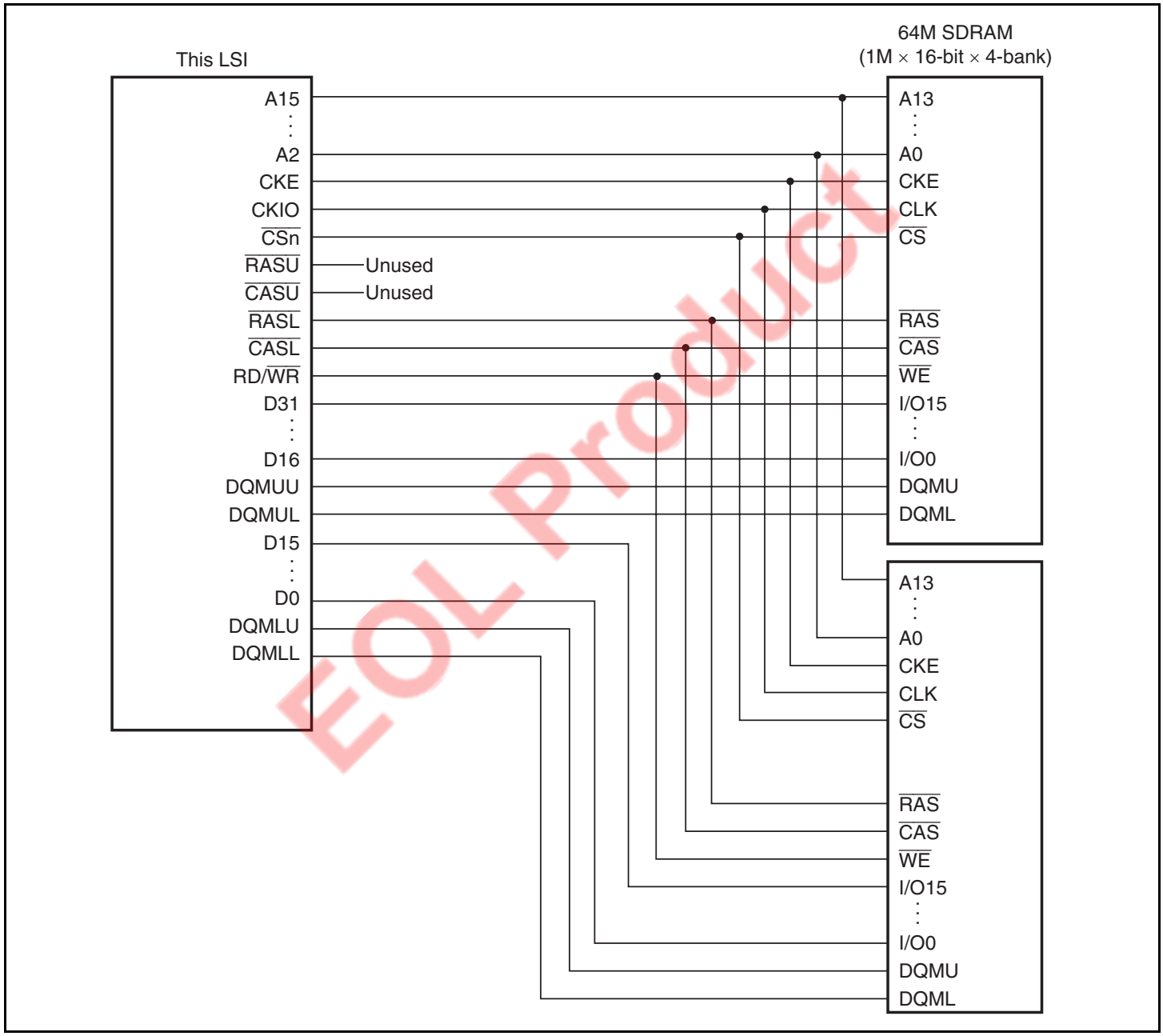


Figure 12.15 Example of 32-Bit Data Width SDRAM Connection ($\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are Not Used)

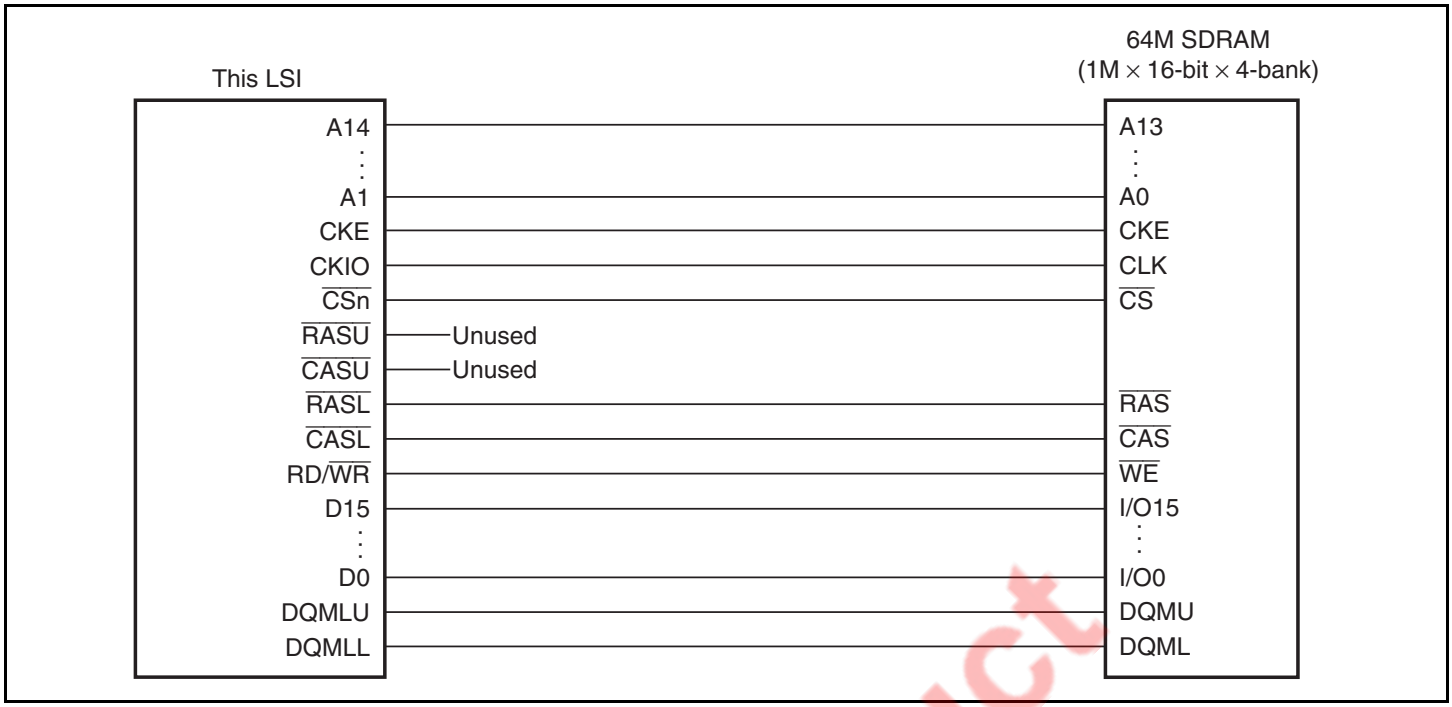


Figure 12.16 Example of 16-Bit Data Width SDRAM Connection ($\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are Not Used)

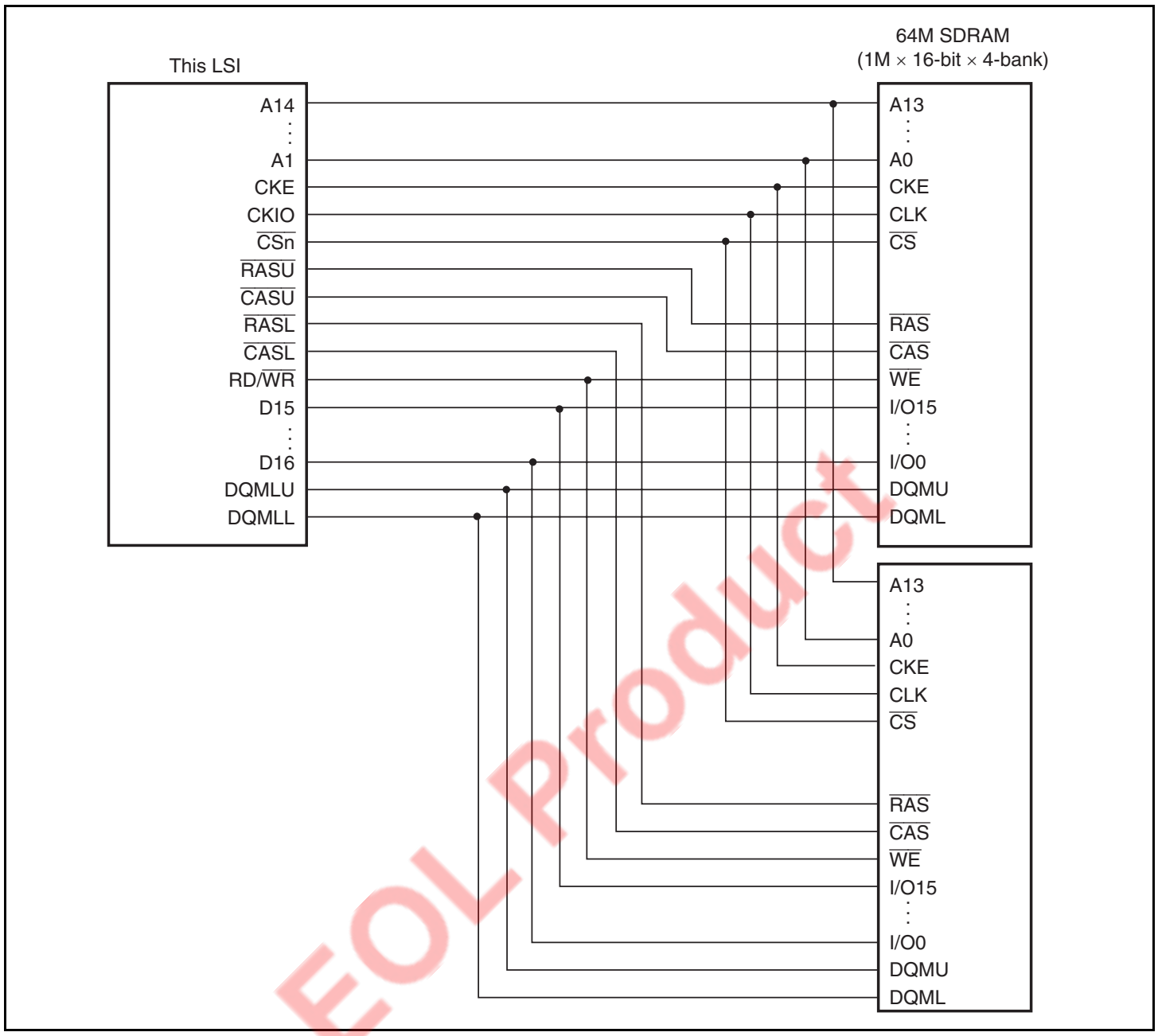


Figure 12.17 Example of 16-Bit Data Width SDRAM Connection ($\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are Used)

Address Multiplexing: An address multiplexing is specified so that SDRAM can be connected without external multiplexing circuitry according to the setting of bits BSZ1 and BSZ0 in CSnBCR, AxROW[1:0] and AxCOL[1:0] in SDCR. Tables 12.8 to 12.13 show the relationship between the settings of bits BSZ1 and BSZ0, AxROW[1:0], and AxCOL[1:0] and the bits output at the address pins. Do not specify those bits in the manner other than this table, otherwise the operation of this LSI is not guaranteed. A25 to A18 are not multiplexed and the original values of address are always output at these pins.

When the data bus width is 16 bits (BSZ1 and BSZ0 = B'10), A0 of SDRAM specifies a word address. Therefore, connect this A0 pin of SDRAM to the A1 pin of the LSI; the A1 pin of SDRAM to the A2 pin of the LSI, and so on. When the data bus width is 32 bits (BSZ1 and BSZ0 = B'11), the A0 pin of SDRAM specifies a longword address. Therefore, connect this A0 pin of SDRAM to the A2 pin of the LSI; the A1 pin of SDRAM to the A3 pin of the LSI, and so on.

Table 12.8 Relationship between BSZ1, 0, A2/3ROW1, 0, and Address Multiplex Output (1)-1

Setting				
BSZ 1, 0	**A2/3 ROW 1, 0**	**A2/3 COL 1, 0**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*[2]	A22*[2]	A12 (BA1)	Specifies bank
A13	A21*[2]	A21*[2]	A11 (BA0)	
A12	A20*[2]	L/H*[1]	A10/AP	Specifies address/precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unused
A0	A8	A0		
Example of connected memory				
64-Mbit product (512 kwords × 32 bits × 4 banks, column 8 bits product): 1				
16-Mbit product (512 kwords × 16 bits × 2 banks, column 8 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
2. Bank address specification

Table 12.8 Relationship between BSZ1, 0, A2/3ROW1, 0, and Address Multiplex Output (1)-2

Setting				
BSZ 1, 0	**A2/3 ROW 1, 0**	**A2/3 COL 1, 0**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23*[2]	A23*[2]		
A14	A22*[2]	A22*[2]	A13 (BA1)	Specifies bank
A13	A21	A13	A12 (BA0)	
A12	A20*[2]	L/H*[1]	A10/AP	Specifies address/precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unused
A0	A8	A0		
Example of connected memory				
128-Mbit product (1 Mword × 32 bits × 4 banks, column 8 bits product): 1				
64-Mbit product (1 Mword × 16 bits × 4 banks, column 8 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
2. Bank address specification

Table 12.9 Relationship between BSZ1, 0, A2/3ROW1, 0, and Address Multiplex Output (2)-1

Setting				
BSZ 1, 0	**A2/3 ROW 1, 0**	**A2/3 COL 1, 0**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*[2]	A24*[2]	A13 (BA1)	Specifies bank
A14	A23*[2]	A23*[2]	A12 (BA0)	
A13	A22	A13	A11	Address
A12	A21	L/H*[1]	A10/AP	Specifies address/precharge
A11	A20*[2]	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
256-Mbit product (2 Mwords × 32 bits × 4 banks, column 9 bits product): 1				
128-Mbit product (2 Mwords × 16 bits × 4 banks, column 9 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A25 pin specified the bank address. $\overline{\text{RASU}}$ is not asserted.

Table 12.9 Relationship between BSZ1, 0, A2/3ROW1, 0, and Address Multiplex Output (2)-2

Setting				
BSZ 1, 0	**A2/3 ROW 1, 0**	**A2/3 COL 1, 0**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*[2]	A25*[2]*[3]	A13 (BA1)	Specifies bank
A14	A24*[2]	A24*[2]	A12 (BA0)	
A13	A23	A13	A11	Address
A12	A22	L/H*[1]	A10/AP	Specifies address/precharge
A11	A21	A11	A9	Address
A10	A20*[2]	A10	A8	
A9	A19	A9	A7	
A8	A18	A8	A6	
A7	A17	A7	A5	
A6	A16	A6	A4	
A5	A15	A5	A3	
A4	A14	A4	A2	
A3	A13	A3	A1	
A2	A12	A2	A0	
A1	A11	A1		Unused
A0	A10	A0		
Example of connected memory				
512-Mbit product (4 Mwords × 32 bits × 4 banks, column 10 bits product): 1				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 10 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A25 pin specified the bank address. $\overline{\text{RASU}}$ is not asserted.

Table 12.10 Relationship between BSZ1, 0, A2/3ROW1, 0, and Address Multiplex Output (3)

Setting				
BSZ 1, 0	**A2/3 ROW 1, 0**	**A2/3 COL 1, 0**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25*[2]*[3]	A25*[2]	A14 (BA1)	Specifies bank
A15	A24*[2]	A24*[2]	A13 (BA0)	
A14	A23	A14	A12	Address
A13	A22	A13	A11	
A12	A21	L/H*[1]	A10/AP	Specifies address/precharge
A11	A20*[2]	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
512-Mbit product (4 Mwords × 32 bits × 4 banks, column 9 bits product): 1				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 9 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A 25 pin specified the bank address. $\overline{\text{RASU}}$ is not asserted.

Table 12.11 Relationship between BSZ1, 0, A2/3ROW1, 0, and Address Multiplex Output (4)-1

Setting				
BSZ 1, 0	**A2/3 ROW 1, 0**	**A2/3 COL 1, 0**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21*[2]	A21*[2]	A12 (BA1)	Specifies bank
A12	A20*[2]	A20*[2]	A11 (BA0)	
A11	A19	L/H*[1]	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
16-Mbit product (512 kwords × 16 bits × 2 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
2. Bank address specification

Table 12.11 Relationship between BSZ1, 0, A2/3ROW1, 0, and Address Multiplex Output (4)-2

Setting				
BSZ 1, 0	**A2/3 ROW 1, 0**	**A2/3 COL 1, 0**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*[2]	A22*[2]	A13 (BA1)	Specifies bank
A13	A21*[2]	A21*[2]	A12 (BA0)	Address
A12	A20	A20	A11	
A11	A19	L/H*[1]	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
64-Mbit product (1 Mword × 16 bits × 4 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
2. Bank address specification

Table 12.12 Relationship between BSZ1, 0, A2/3ROW1, 0, and Address Multiplex Output (5)-1

Setting				
BSZ 1, 0	**A2/3 ROW 1, 0**	**A2/3 COL 1, 0**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24	A15		
A14	A23	A23*[2]	A13 (BA1)	Specifies bank
A13	A22*[2]	A22*[2]	A12 (BA0)	
A12	A21*[2]	A12	A11	Address
A11	A20	L/H*[1]	A10/AP	Specifies address/precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
128-Mbit product (2 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
2. Bank address specification

Table 12.12 Relationship between BSZ1, 0, A2/3ROW1, 0, and Address Multiplex Output (5)-2

Setting				
BSZ 1, 0	**A2/3 ROW 1, 0**	**A2/3 COL 1, 0**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25	A15		
A14	A24*[2]	A24*[2]	A13 (BA1)	Specifies bank
A13	A23*[2]	A23*[2]	A12 (BA0)	
A12	A22	A12	A11	Address
A11	A21	L/H*[1]	A10/AP	Specifies address/precharge
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
2. Bank address specification

Table 12.13 Relationship between BSZ1, 0, A2/3ROW1, 0, and Address Multiplex Output (6)-1

Setting				
BSZ 1, 0	**A2/3 ROW 1, 0**	**A2/3 COL 1, 0**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*[2]	A24*[2]	A14 (BA1)	Specifies bank
A14	A23*[2]	A23*[2]	A13 (BA0)	
A13	A22	A13	A12	Address
A12	A21	A12	A11	
A11	A20*[2]	L/H*[1]	A10/AP	Specifies address/precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A25 pin specified the bank address. $\overline{\text{RASU}}$ is not asserted.

Table 12.13 Relationship between BSZ1, 0, A2/3ROW1, 0, and Address Multiplex Output (6)-2

Setting				
BSZ 1, 0	**A2/3 ROW 1, 0**	**A2/3 COL 1, 0**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*[2]	A25*[2]*[3]	A14 (BA1)	Specifies bank
A14	A24*[2]	A24*[2]	A13 (BA0)	
A13	A23	A13	A12	Address
A12	A22	A12	A11	
A11	A21	L/H*[1]	A10/AP	Specifies address/precharge
A10	A20*[2]	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
512-Mbit product (8 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A25 pin specified the bank address. $\overline{\text{RASU}}$ is not asserted.

Burst Read: A burst read occurs in the following cases with this LSI.

- Access size in reading is larger than data bus width.
- 16-byte transfer in cache error.
- 16-byte transfer in DMAC

This LSI always accesses the SDRAM with burst length 1. For example, read access of burst length 1 is performed consecutively 4 times to read 16-byte continuous data from the SDRAM that is connected to a 32-bit data bus.

Table 12.14 shows the relationship between the access size and the number of bursts.

Table 12.14 Relationship between Access Size and Number of Bursts

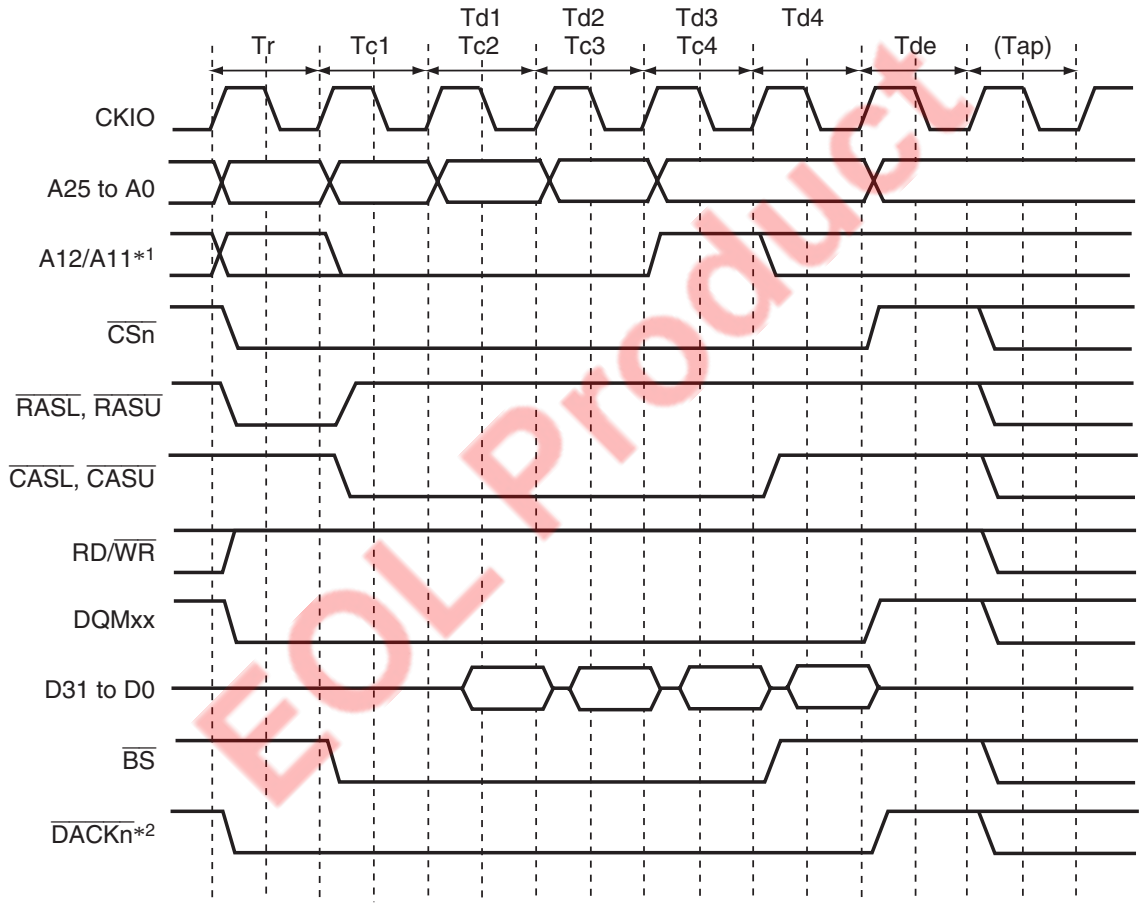
Bus Width	Access Size	Number of Bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	16 bits	8
32 bits	8 bits	1
	16 bits	1
	32 bits	1
	16 bits	4

Figures 12.18 and 12.19 show a timing chart in burst read. In burst read, an ACTV command is output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, the READA command is issued in the Tc4 cycle, and the read data is received at the rising edge of the external clock (CKIO) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion of an auto-precharge induced by the READA command in the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Tap cycles is specified by the WTRP1 and WTRP0 bits in the CS3WCR register.

In this LSI, wait cycles can be inserted by specifying each bit in the CS3WCR register to connect the SDRAM in variable frequencies. Figure 12.19 shows an example in which wait cycles are inserted. The number of cycles from the Tr cycle where the ACTV command is output to the Tc1 cycle where the READ command is output can be specified using the WTRCD1 and WTRCD0 bits in the CS3WCR register. If the WTRCD1 and WTRCD0 bits specify one cycles or more, a Trw cycle where the NOT command is issued is inserted between the Tr cycle and Tc1 cycle. The

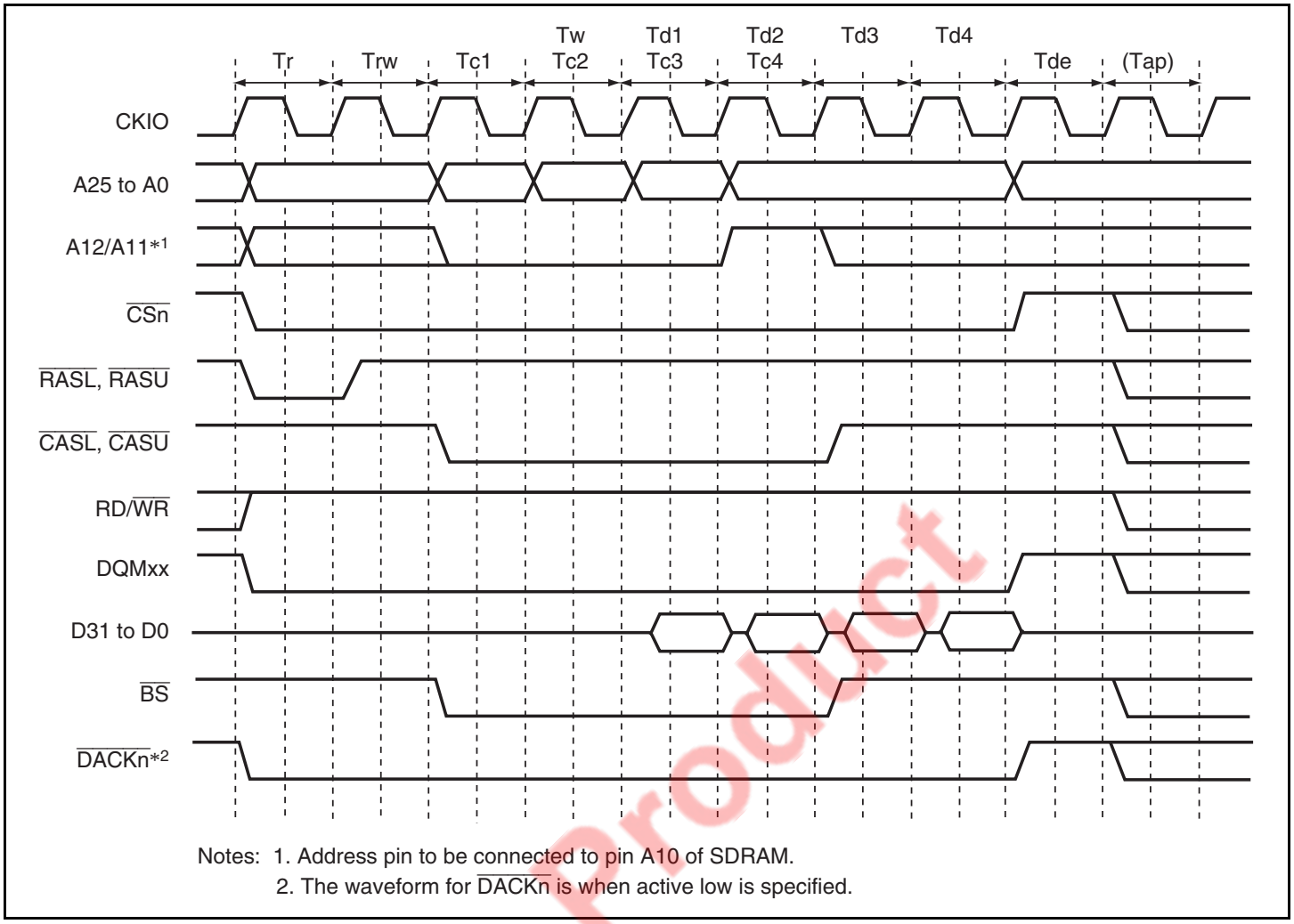
number of cycles from the Tc1 cycle where the READ command is output to the Td1 cycle where the read data is latched can be specified for the CS2 and CS3 spaces independently, using the A2CL1 and A2CL0 bits in the CS2WCR register or the A3CL1 and A3CL0 bits in the CS3WCR register and WTRCD0 bit in the CS3WCR register. The number of cycles from Tc1 to Td1 corresponds to the SDRAM CAS latency. The CAS latency for the SDRAM is normally defined as up to three cycles. However, the CAS latency in this LSI can be specified as 1 to 4 cycles. This CAS latency can be achieved by connecting a latch circuit between this LSI and the SDRAM.

A Tde cycle is an idle cycle required to transfer the read data into this LSI and occurs once for every burst read or every single read.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.18 Burst Read Basic Timing (CAS Latency 1, Auto Pre-Charge)

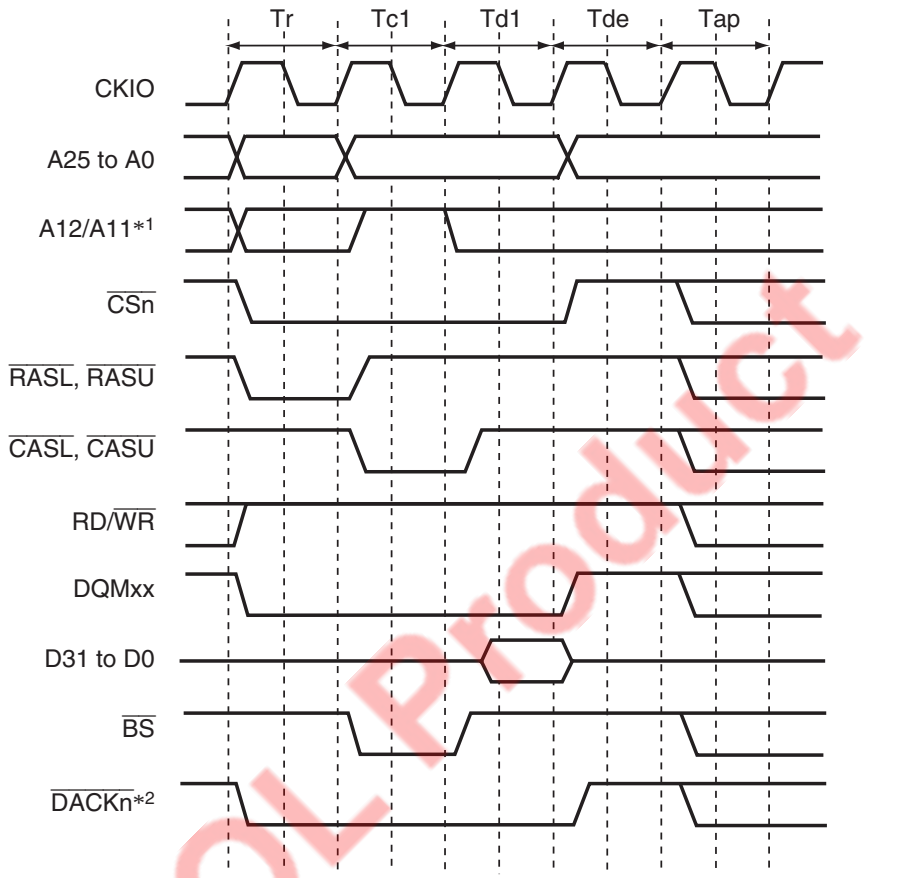


Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.19 Burst Read Wait Specification Timing (CAS Latency 2, WTRCD1 and WTRCD0 = 1 Cycle, Auto Pre-Charge)

Single Read: A read access ends in one cycle when data exists in non-cacheable region and the data bus width is larger than or equal to access size. As the burst length is set to 1 in synchronous DRAM burst read/single write mode, only the required data is output.

Figure 12.20 shows the single read basic timing.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.20 Basic Timing for Single Read (CAS Latency 1, Auto Pre-Charge)

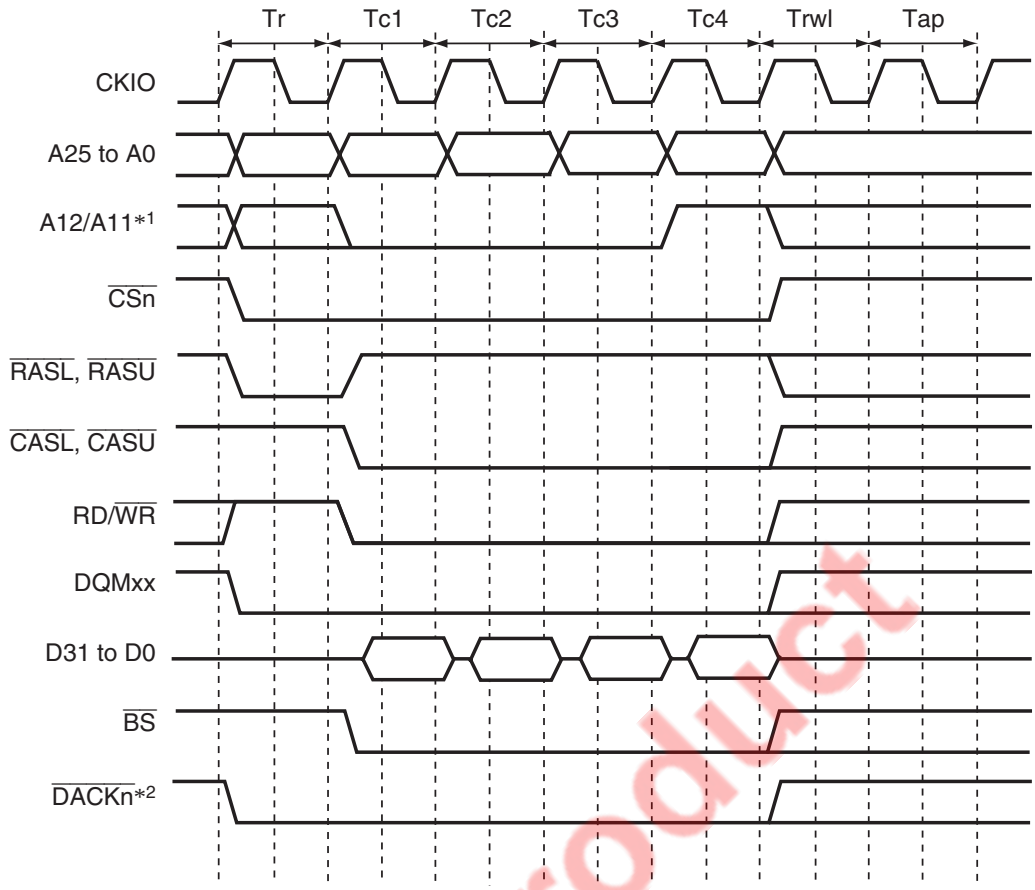
Burst Write: A burst write occurs in the following cases in this LSI.

- Access size in writing is larger than data bus width.
- Write-back of the cache
- 16-byte transfer in DMAC

This LSI always accesses SDRAM with burst length 1. For example, write access of burst length 1 is performed continuously 4 times to write 16-byte continuous data to the SDRAM that is connected to a 32-bit data bus.

The relationship between the access size and the number of bursts is shown in table 12.14.

Figure 12.21 shows a timing chart for burst writes. In burst write, an ACTV command is output in the Tr cycle, the WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and the WRITA command is issued to execute an auto-precharge in the Tc4 cycle. In the write cycle, the write data is output simultaneously with the write command. After the write command with the auto-precharge is output, the Trw1 cycle that waits for the auto-precharge initiation is followed by the Tap cycle that waits for completion of the auto-precharge induced by the WRITA command in the SDRAM. Between the Trwl and the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Trw1 cycles is specified by the TRWL1 and TRWL0 bits in the CS3WCR register. The number of Tap cycles is specified by the WTRP1 and WTRP0 bits in the CS3WCR register.

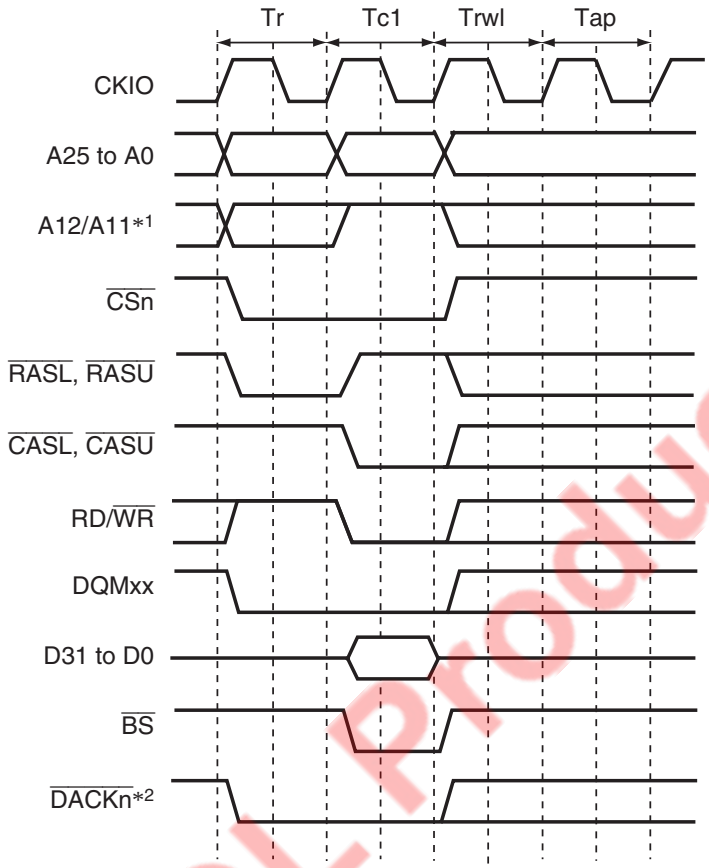


Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.21 Basic Timing for Burst Write (Auto Pre-Charge)

Single Write: A write access ends in one cycle when data is written in non-cacheable region and the data bus width is larger than or equal to access size.

Figure 12.22 shows the single write basic timing.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.22 Single Write Basic Timing (Auto-Precharge)

Bank Active: The synchronous DRAM bank function is used to support high-speed accesses to the same row address. When the BACTV bit in SDCR is 1, accesses are performed using commands without auto-precharge (READ or WRIT). This function is called bank-active function. This function is valid only for either the upper or lower bits of area 3. When area 3 is set to bank-active mode, area 2 should be set to normal space. When areas 2 and 3 are both set to SDRAM or both the upper and lower bits of area 3 are connected to SDRAM, auto pre-charge mode must be set. In this case, precharging is not performed when the access ends. When accessing the same row address in the same bank, it is possible to issue the READ or WRIT command immediately, without issuing an ACTV command. As synchronous DRAM is internally divided into several banks, it is possible to activate one row address in each bank. If the next access is to a different row address, a PRE command is first issued to precharge the relevant bank, then when precharging is completed, the access is performed by issuing an ACTV command followed by a READ or WRIT command. If this is followed by an access to a different row address, the access time will be longer because of the precharging performed after the access request is issued. The number of cycles between issuance of the PRE command and the ACTV command is determined by the WTRP1 and WTPR0 bits in CS3WCR.

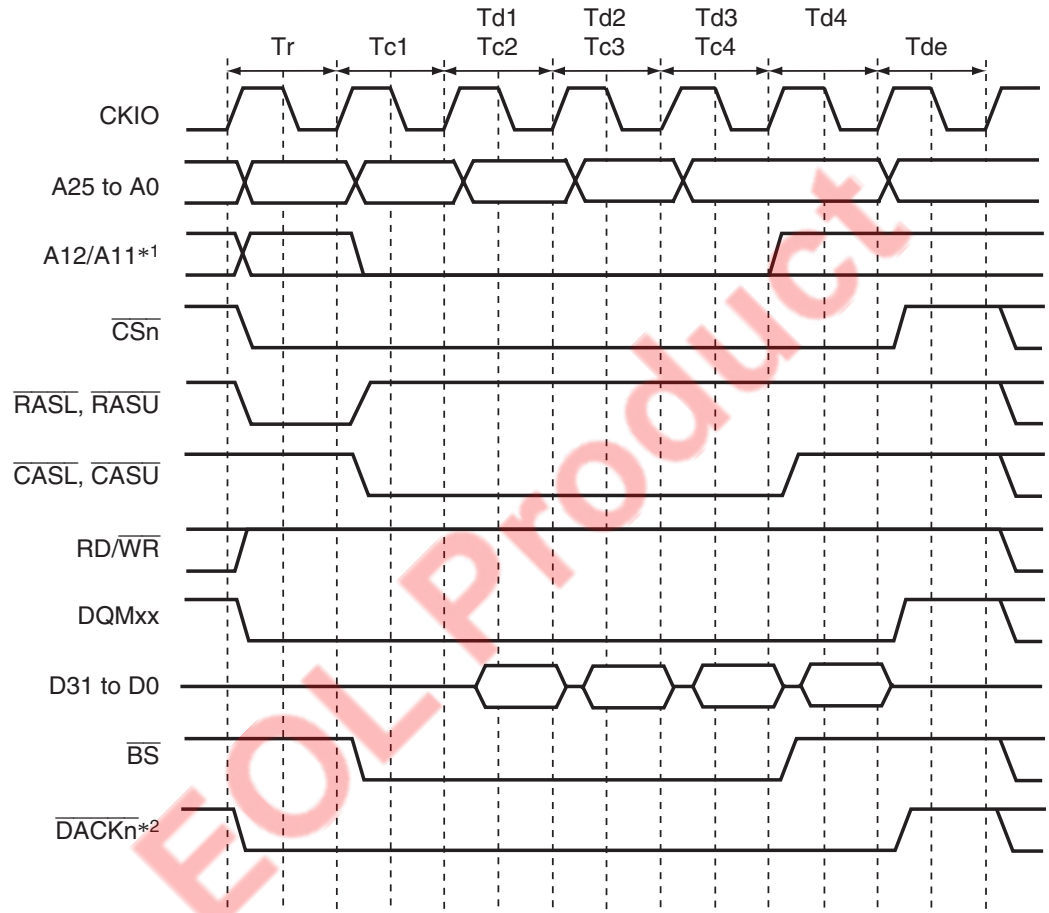
In a write, when an auto-precharge is performed, a command cannot be issued to the same bank for a period of Trwl + Tap cycles after issuance of the WRITA command. When bank active mode is used, READ or WRIT commands can be issued successively if the row address is the same. The number of cycles can thus be reduced by Trwl + Tap cycles for each write.

There is a limit on tRAS, the time for placing each bank in the active state. If there is no guarantee that there will not be a cache hit and another row address will be accessed within the period in which this value is maintained by program execution, it is necessary to set auto-refresh and set the refresh cycle to no more than the maximum value of tRAS.

A burst read cycle without auto-precharge is shown in figure 12.23, a burst read cycle for the same row address in figure 12.24, and a burst read cycle for different row addresses in figure 12.25. Similarly, a burst write cycle without auto-precharge is shown in figure 12.26, a burst write cycle for the same row address in figure 12.27, and a burst write cycle for different row addresses in figure 12.28.

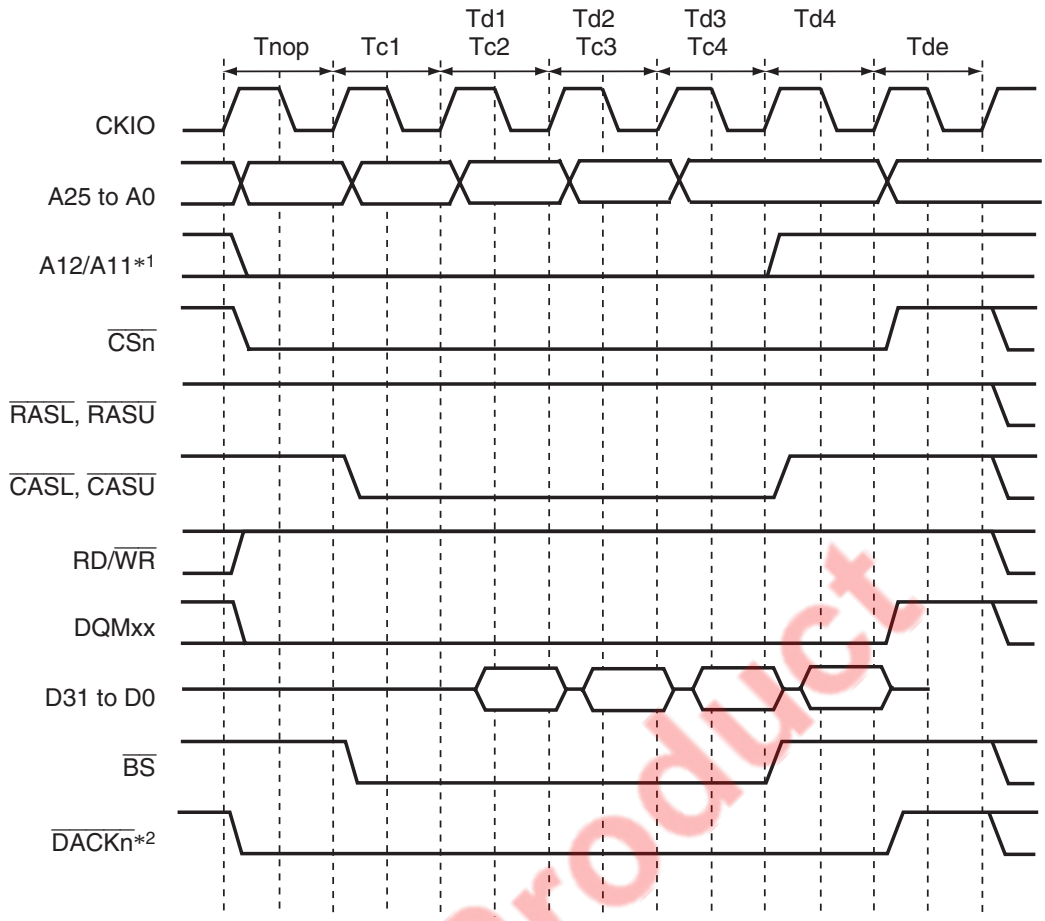
In figure 12.24, a Tnop cycle in which no operation is performed is inserted before the Tc cycle that issues the READ command. The Tnop cycle is inserted to acquire two cycles of CAS latency for the DQMxx signal that specifies the read byte in the data read from the SDRAM. If the CAS latency is specified as two cycles or more, the Tnop cycle is not inserted because the two cycles of latency can be acquired even if the DQMxx signal is asserted after the Tc cycle.

When bank active mode is set, if only accesses to the respective banks in the area 3 space are considered, as long as accesses to the same row address continue, the operation starts with the cycle in figure 12.23 or 12.26, followed by repetition of the cycle in figure 12.24 or 12.27. An access to a different area during this time has no effect. If there is an access to a different row address in the bank active state, after this is detected the bus cycle in figure 12.24 or 12.27 is executed instead of that in figure 12.25 or 12.28. In bank active mode, too, all banks become inactive after a refresh cycle or after the bus is released as the result of bus arbitration.



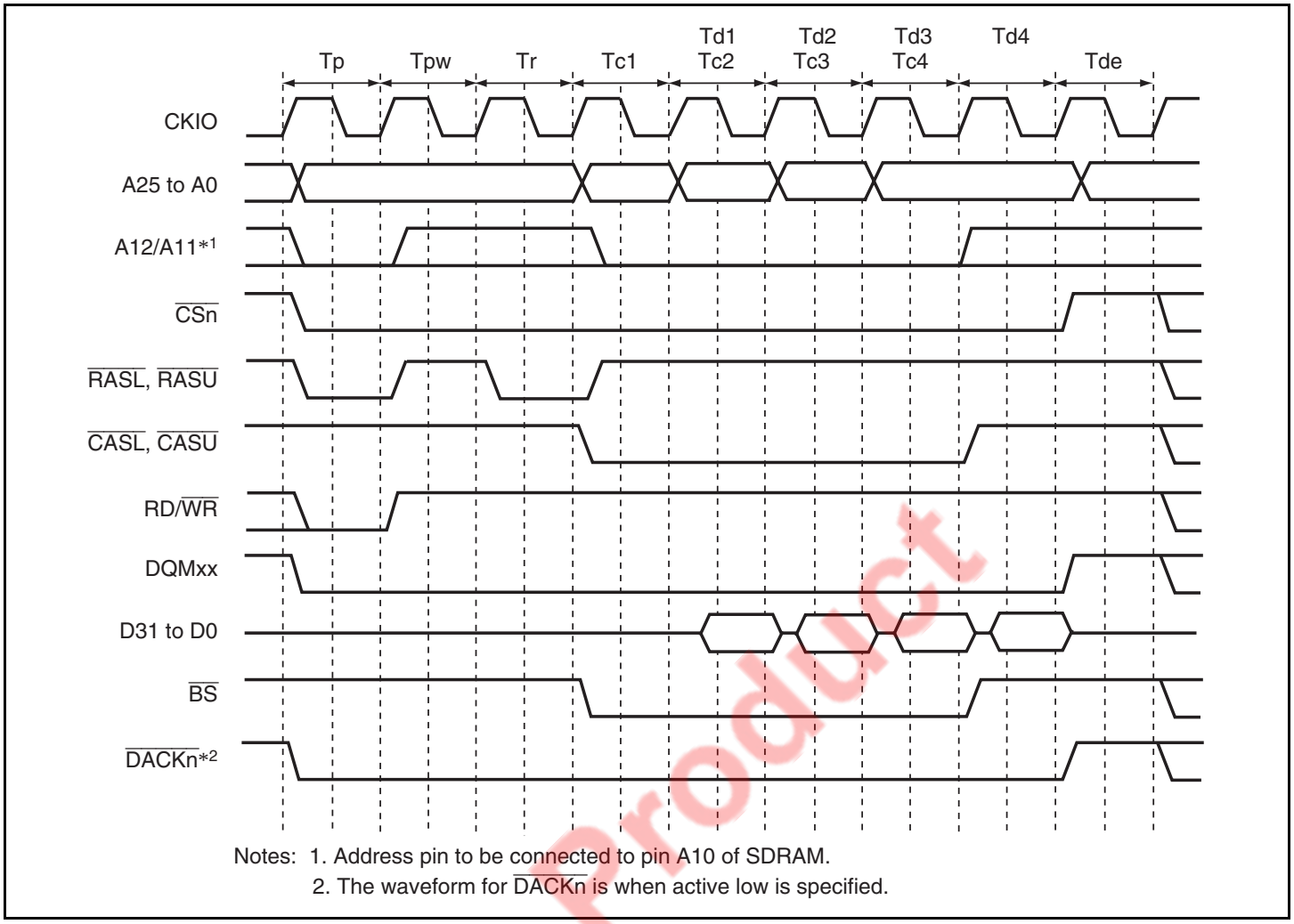
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.23 Burst Read Timing (Bank Active, Different Bank, CAS Latency 1)



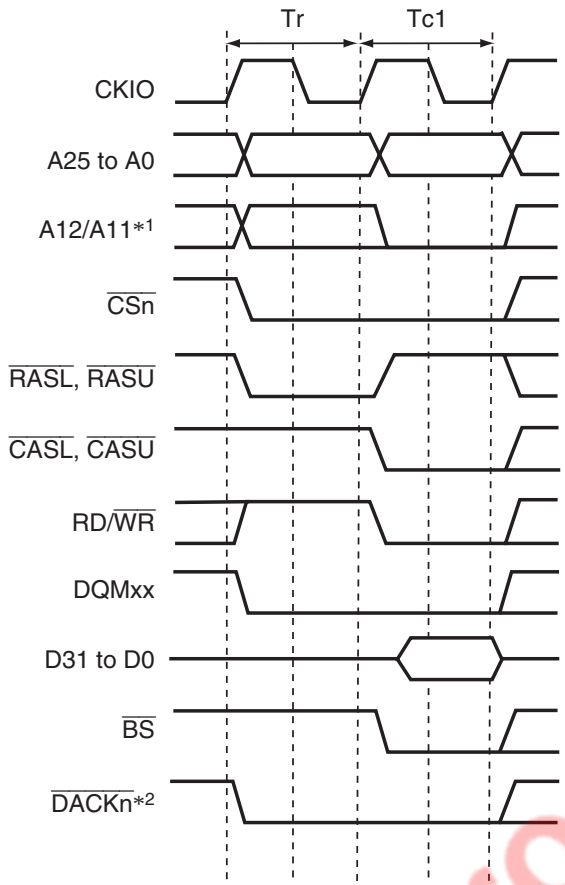
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.24 Burst Read Timing
(Bank Active, Same Row Addresses in the Same Bank, CAS Latency 1)



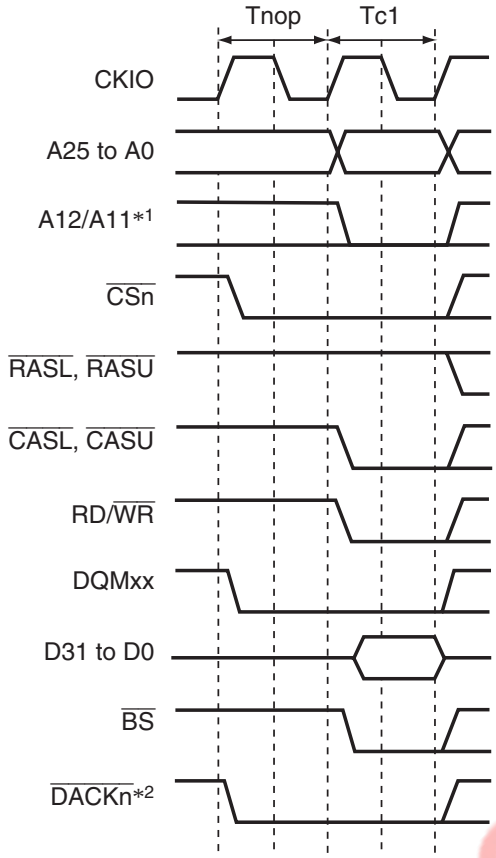
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.25 Burst Read Timing
(Bank Active, Different Row Addresses in the Same Bank, CAS Latency 1)



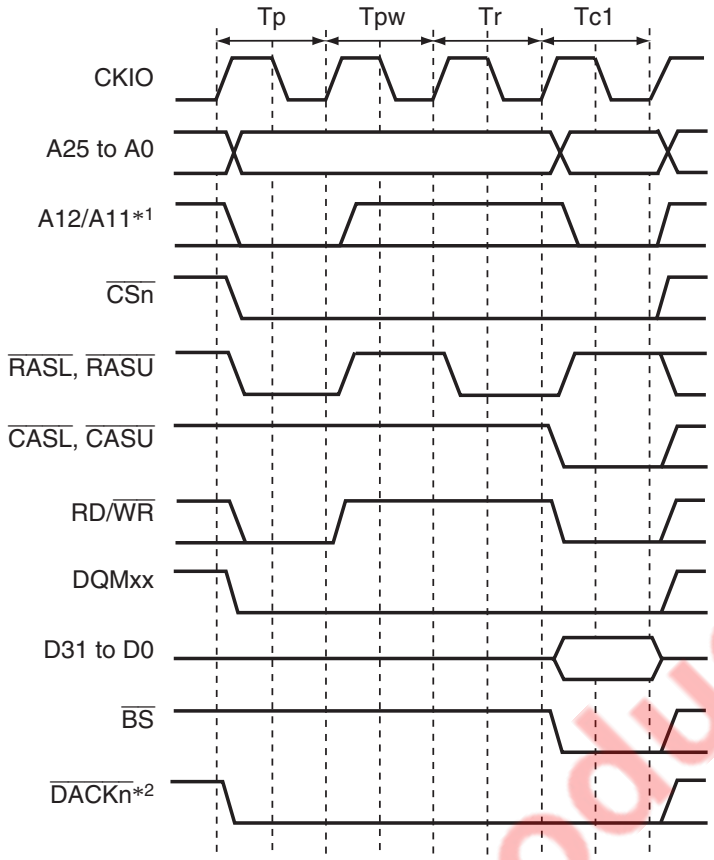
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.26 Single Write Timing (Bank Active, Different Bank)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.27 Single Write Timing (Bank Active, Same Row Addresses in the Same Bank)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.28 Single Write Timing
(Bank Active, Different Row Addresses in the Same Bank)

Refreshing: This LSI has a function for controlling synchronous DRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in SDCR. A continuous refreshing can be performed by setting the RRC2 to RRC0 bits in RTCSR. If synchronous DRAM is not accessed for a long period, self-refresh mode, in which the power consumption for data retention is low, can be activated by setting both the RMODE bit and the RFSH bit to 1.

1. Auto-refreshing

   Refreshing is performed at intervals determined by the input clock selected by bits CKS2 to CKS0 in RTCSR, and the value set by in RTCOR. The value of bits CKS2 to CKS0 in RTCOR should be set so as to satisfy the refresh interval stipulation for the synchronous DRAM used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in SDCR, then make the CKS2 to CKS0 and RRC2 to RRC0 settings. When the clock is selected by bits CKS2 to CKS0, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed for the number of times specified by the RRC2 to RRC0. At the same time, RTCNT is cleared to zero and the count-up is restarted. Figure 12.29 shows the auto-refresh cycle timing.

   After starting, the auto refreshing, PALL command is issued in the Tp cycle to make all the banks to pre-charged state from active state when some bank is being pre-charged. Then REF command is issued in the Trr cycle after inserting idle cycles of which number is specified by the WTRP1 and WTRP0 bits in CS3WCR. A new command is not issued for the duration of the number of cycles specified by the WTRC1 and WTRC0 bits in CS3WCR after the Trr cycle. The WTRC1 and WTRC0 bits must be set so as to satisfy the SDRAM refreshing cycle time stipulation (tRC). An idle cycle is inserted between the Tp cycle and Trr cycle when the setting value of the WTRP1 and WTRP0 bits in CS3WCR is longer than or equal to 1 cycle.

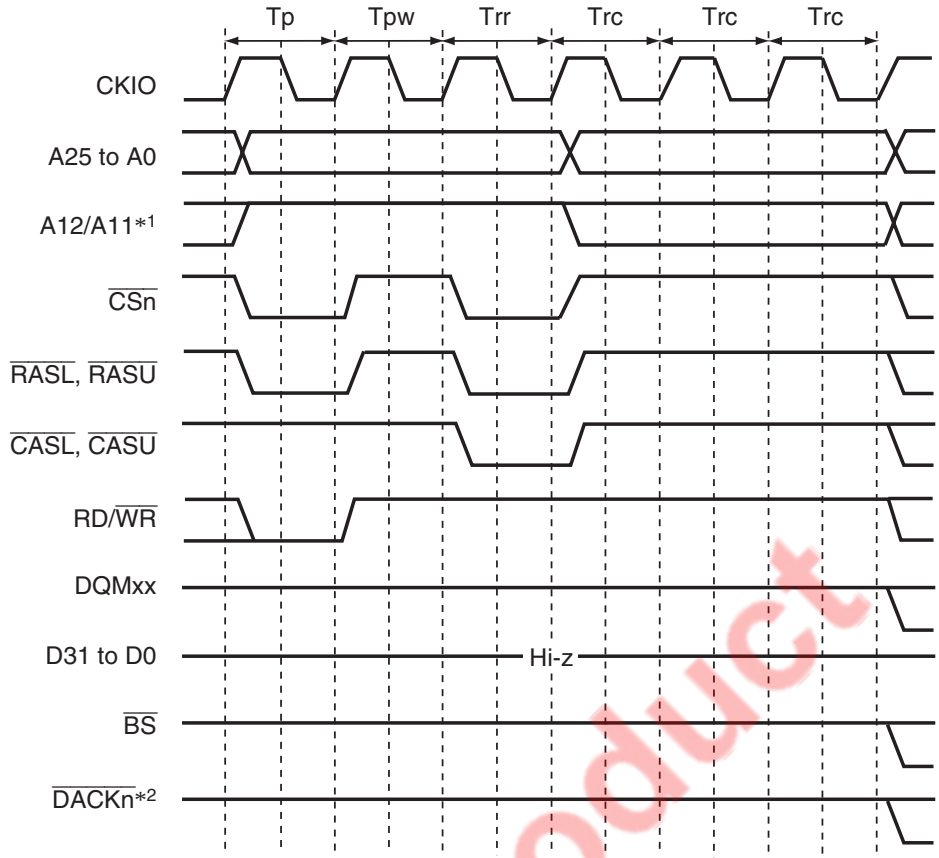


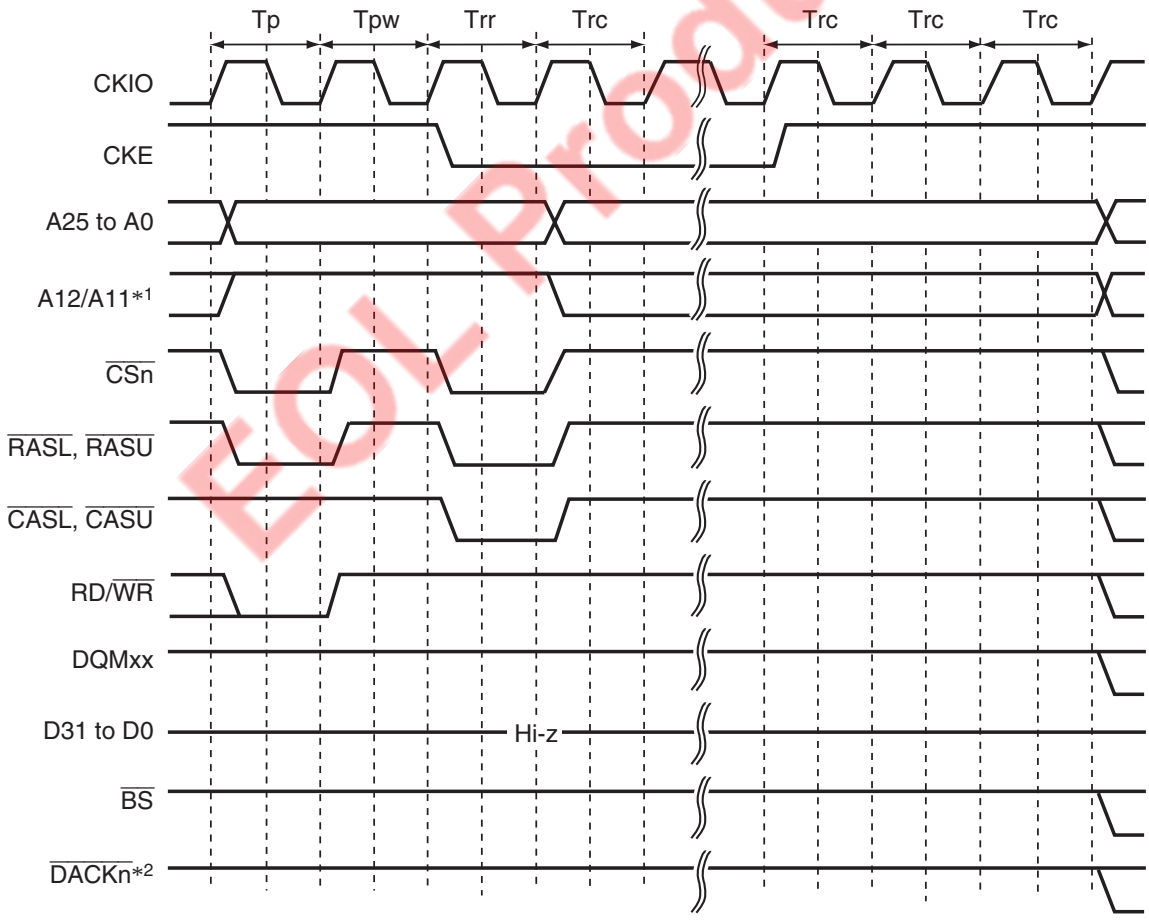
Figure 12.29 Auto-Refresh Timing

2. Self-refreshing

Self-refresh mode in which the refresh timing and refresh addresses are generated within the synchronous DRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit in SDCR to 1. After starting the self-refreshing, PALL command is issued in Tp cycle after the completion of the pre-charging bank. A SELF command is then issued after inserting idle cycles of which number is specified by the WTRP1 and WTRP0 bits in CS3WSR. Synchronous DRAM cannot be accessed while in the self-refresh state. Self-refresh mode is cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared, command issuance is disabled for the number of cycles specified by the WTRC1 and WTRC0 bits in CS3WCR.

Self-refresh timing is shown in figure 12.30. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, or when exiting standby mode other than through a power-on reset, auto-refreshing is restarted if the RFSH bit is set to 1 and the RMODE bit is cleared to 0 when self-refresh mode is cleared. If the transition from clearing of self-refresh mode to the start of auto-refreshing takes time, this time should be taken into consideration when setting the initial value of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable refreshing to be started immediately.

After self-refreshing has been set, the self-refresh state continues even if the chip standby state is entered using the LSI standby function, and is maintained even after recovery from standby mode. Note that the HIZCNT bit in the CMNCR register needs to be set to 1 and pins such as CKE are driven in standby mode. The self-refresh state cannot be cleared through a manual reset. In case of a power-on reset, the bus state controller's registers are initialized, and therefore the self-refresh state is cleared.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.30 Self-Refresh Timing

Relationship between Refresh Requests and Bus Cycles: If a refresh request occurs during bus cycle execution, the refresh cycle must wait for the bus cycle to be completed. If a refresh request occurs while the bus is released by the bus arbitration function, the refresh will not be executed until the bus mastership is acquired.

If a new refresh request occurs while waiting for the previous refresh request, the previous refresh request is deleted. To refresh correctly, a bus cycle longer than the refresh interval or the bus mastership occupation must be prevented from occurring. If a bus mastership is requested during self-refresh, the bus will not be released until the refresh is completed.

Low-Frequency Mode: When the SLOW bit in SDCR is set to 1, output of commands, addresses, and write data, and fetch of read data are performed at a timing suitable for operating SDRAM at a low frequency.

Figure 12.31 shows the access timing in low-frequency mode. In this mode, commands, addresses, and write data are output in synchronization with the falling edge of CKIO, which is half a cycle delayed than the normal timing. Read data is fetched at the rising edge of CKIO, which is half a cycle faster than the normal timing. This timing allows the hold time of commands, addresses, write data, and read data to be extended.

If SDRAM is operated at a high frequency with the SLOW bit set to 1, the setup time of commands, addresses, write data, and read data are not guaranteed. Take the operating frequency and timing design into consideration when making the SLOW bit setting.

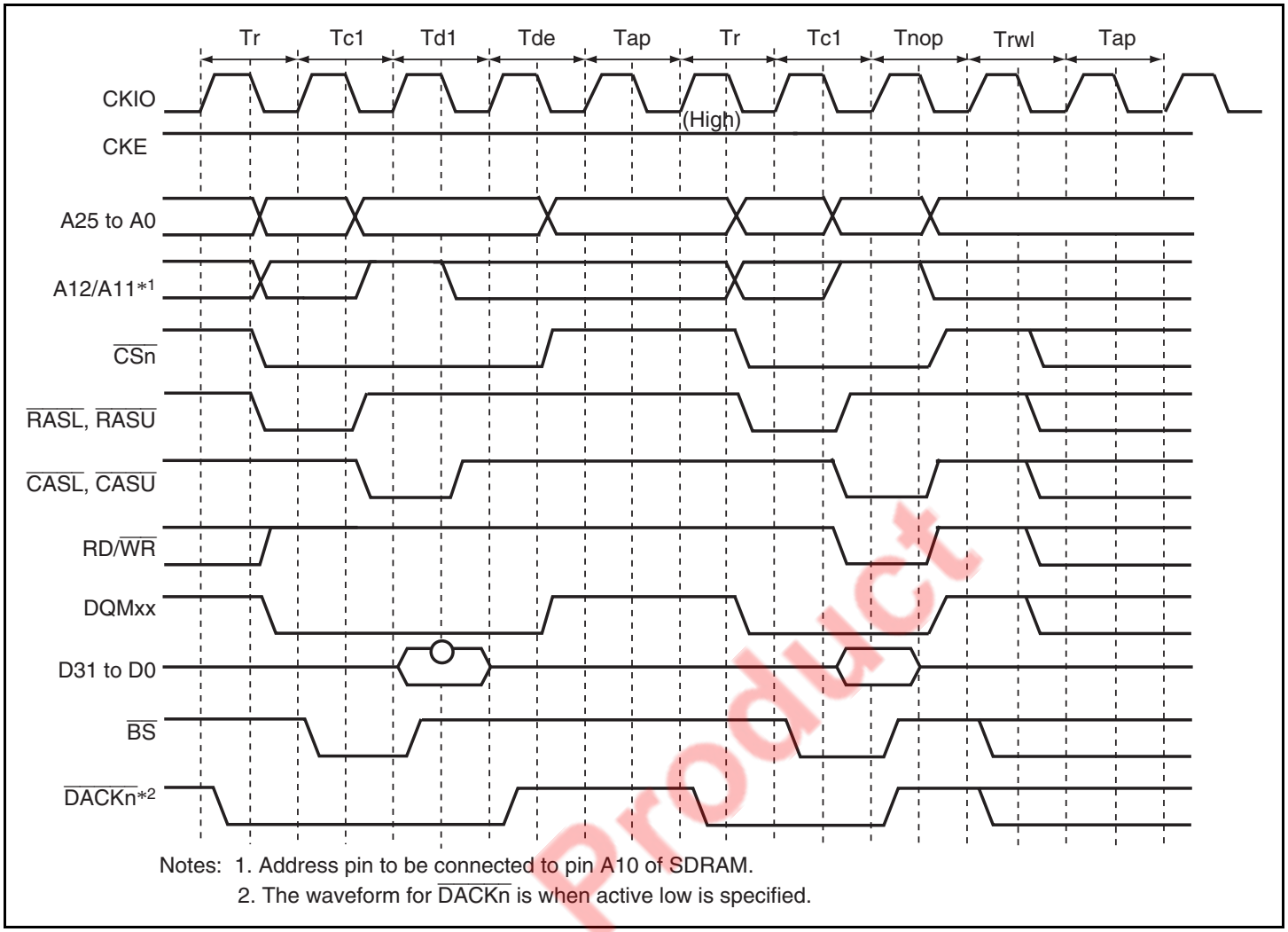


Figure 12.31 Low-Frequency Mode Access Timing

Power-Down Mode: If the PDOWN bit in the SDCR register is set to 1, the SDRAM is placed in the power-down mode by bringing the CKE signal to the low level in the non-access cycle. This power-down mode can effectively lower the power consumption in the non-access cycle. However, please note that if an access occurs in the power-down mode, a cycle of overhead occurs because a cycle is needed to assert the CKE in order to cancel the power-down mode.

Figure 12.32 shows the access timing in the power-down mode.

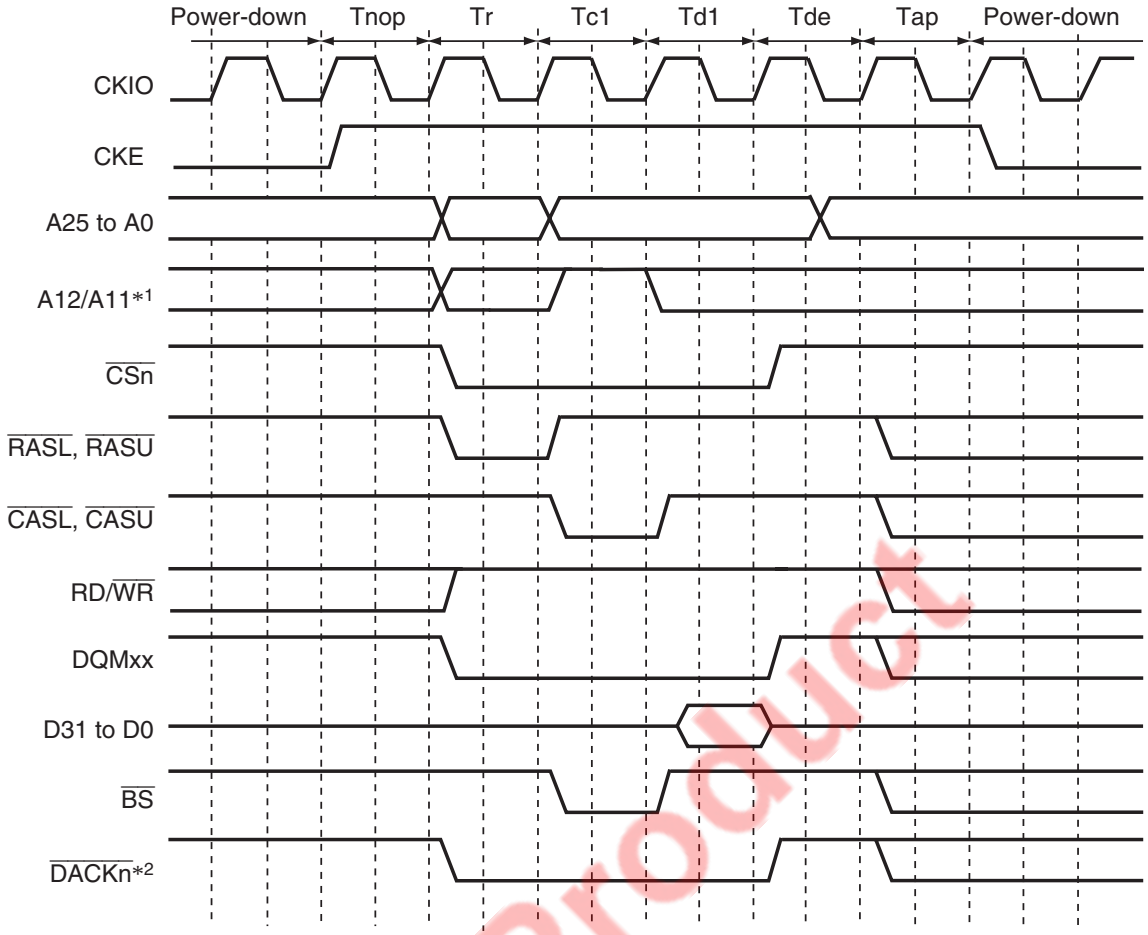


Figure 12.32 Power-Down Mode Access Timing

The conditions to shift to the power-down mode are as follows.

- Write or read access (including instruction fetch) occurs to the memory other than the SDRAM, which is to be set to the power-down mode.
- Read or write access occurs to the control register with the address H'Axxx xxxx or to the peripheral I/O register.

Power-On Sequence: In order to use synchronous DRAM, mode setting must first be performed after powering on. To perform synchronous DRAM initialization correctly, the bus state controller registers must first be set, followed by a write to the synchronous DRAM mode register. In synchronous DRAM mode register setting, the address signal value at that time is latched by a combination of the $\overline{CSn}$, $\overline{RASU}$, $\overline{RASL}$, $\overline{CASU}$, $\overline{CASL}$, and RD/$\overline{WR}$ signals. If the value to be set is X, the bus state controller provides for value X to be written to the synchronous DRAM mode register by performing a write to address H'A4FD4000 + X for area 2 synchronous DRAM, and to address H'A4FD5000 + X for area 3 synchronous DRAM. In this operation the data is ignored, but the mode write is performed as a byte-size access. To set burst read/single write, CAS latency 2 to 3, wrap type = sequential, and burst length 1 supported by the LSI, arbitrary data is written in a byte-size access to the addresses shown in table 12.15. In this time 0 is output at the external address pins of A12 or later.

Table 12.15 Access Address in SDRAM Mode Register Write

- Setting for Area 2

  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'A4FD4440	H'0000440
	3	H'A4FD4460	H'0000460
32 bits	2	H'A4FD4880	H'0000880
	3	H'A4FD48C0	H'00008C0

Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'A4FD4040	H'0000040
	3	H'A4FD4060	H'0000060
32 bits	2	H'A4FD4080	H'0000080
	3	H'A4FD40C0	H'00000C0

- Setting for Area 3

  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'A4FD5440	H'0000440
	3	H'A4FD5460	H'0000460
32 bits	2	H'A4FD5880	H'0000880
	3	H'A4FD58C0	H'00008C0

  Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'A4FD5040	H'0000040
	3	H'A4FD5060	H'0000060
32 bits	2	H'A4FD5080	H'0000080
	3	H'A4FD50C0	H'00000C0

Mode register setting timing is shown in figure 12.33. A PALL command (all bank pre-charge command) is firstly issued. A REF command (auto refresh command) is then issued 8 times. An MRS command (mode register write command) is finally issued. Idle cycles, of which number is specified by the WTRP1 and WTRP0 bits in CS3WCR, are inserted between the PALL and the first REF. Idle cycles, of which number is specified by the WTRC1 and WTRC0 bits in CS3WCR, are inserted between REF and REF, and between the 8th REF and MRS. Idle cycles, of which number is one or more, are inserted between the MRS and a command to be issued next.

It is necessary to keep idle time of certain cycles for SDRAM before issuing PALL command after power-on. Refer the manual of the SDRAM for the idle time to be needed. When the pulse width of the reset signal is longer then the idle time, mode register setting can be started immediately after the reset, but care should be taken when the pulse width of the reset signal is shorter than the idle time.

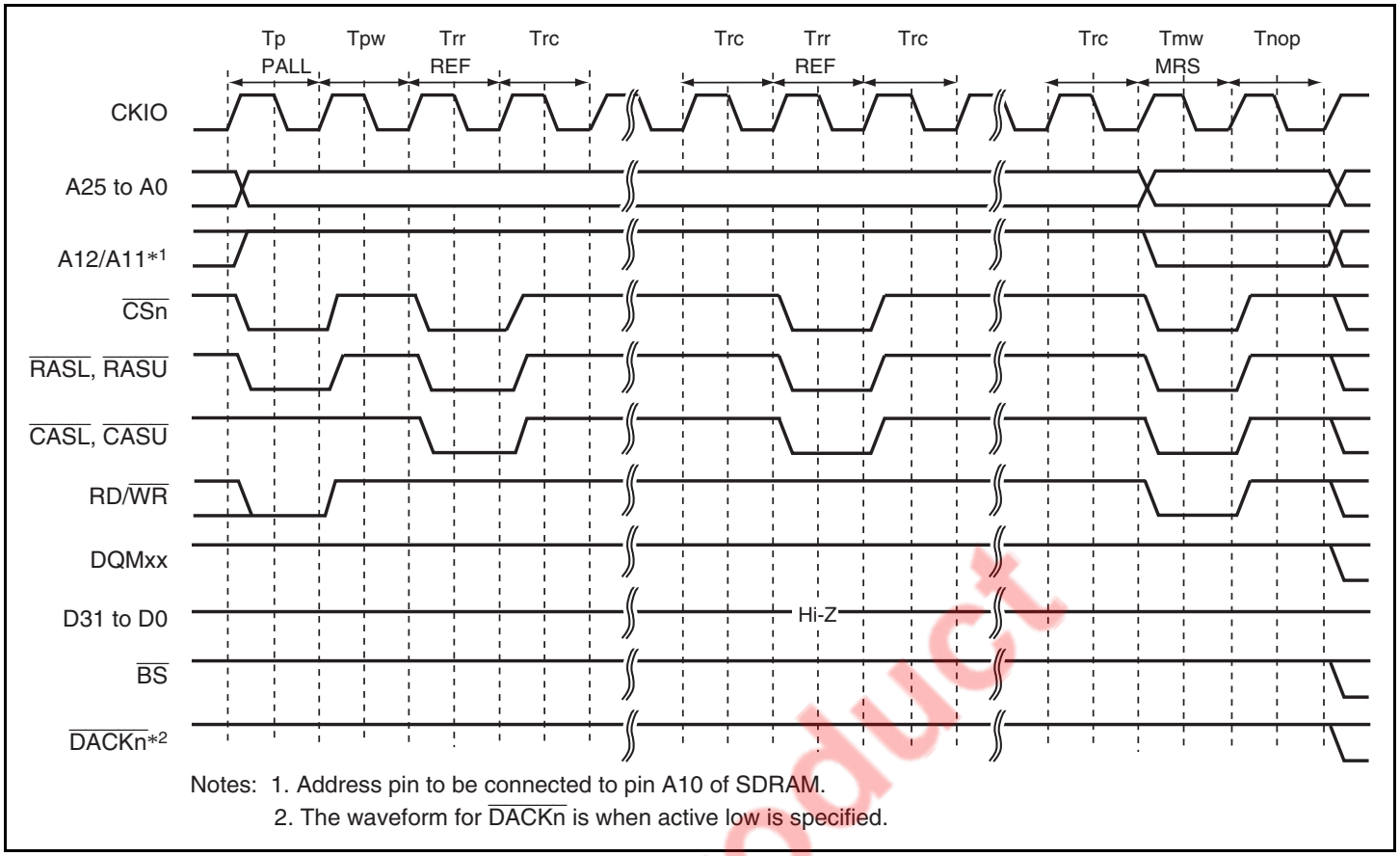


Figure 12.33 Synchronous DRAM Mode Write Timing (Based on JEDEC)

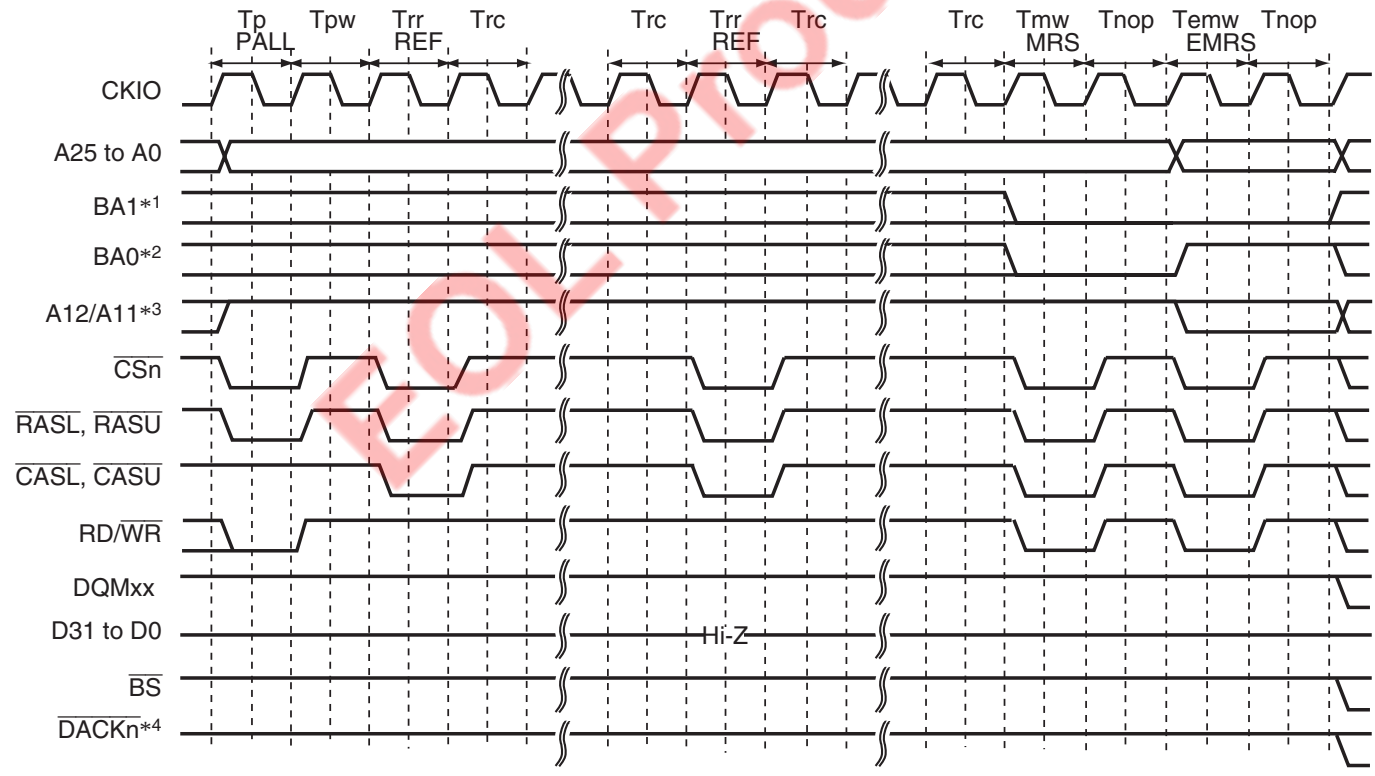
Low-Power SDRAM: The low-power SDRAM can be accessed using the same protocol as the normal SDRAM. The differences between the low-power SDRAM and normal SDRAM are that partial refresh takes place that puts only a part of the SDRAM in the self-refresh state during the self-refresh function, and that power consumption is low during refresh under user conditions such as the operating temperature. The partial refresh is effective in systems in which there is data in a work area other than the specific area can be lost without severe repercussions.

The low-power SDRAM supports the extension mode register (EMRS) in addition to the mode registers as the normal SDRAM. This LSI supports issuing of the EMRS command.

The EMRS command is issued according to the conditions specified in table 12.21. For example, if data H'0YYYYYYY is written to address H'A4FD5XX0 in longword, the commands are issued to the CS3 space in the following sequence: PALL -> REF × 8 -> MRS -> EMRS. In this case, the MRS and EMRS issue addresses are H'0000XX0 and H'YYYYYYY, respectively. If data H'1YYYYYYY is written to address H'A4FD5XX0 in longword, the commands are issued to the CS3 space in the following sequence: PALL -> MRS -> EMRS.

Table 12.16 Output Addresses when EMRS Command Is Issued

Command to be Issued	Access Address	Access Data	Write Access Size	MRS Command Issue Address	EMRS Command Issue Address
CS2 MRS	H'A4FD4XX0	H'********	16 bits	H'0000XX0	—
CS3 MRS	H'A4FD5XX0	H'********	16 bits	H'0000XX0	—
CS2 MRS + EMRS (with refresh)	H'A4FD4XX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS3 MRS + EMRS (with refresh)	H'A4FD5XX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS2 MRS + EMRS (without refresh)	H'A4FD4XX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS3 MRS + EMRS (without refresh)	H'A4FD5XX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY



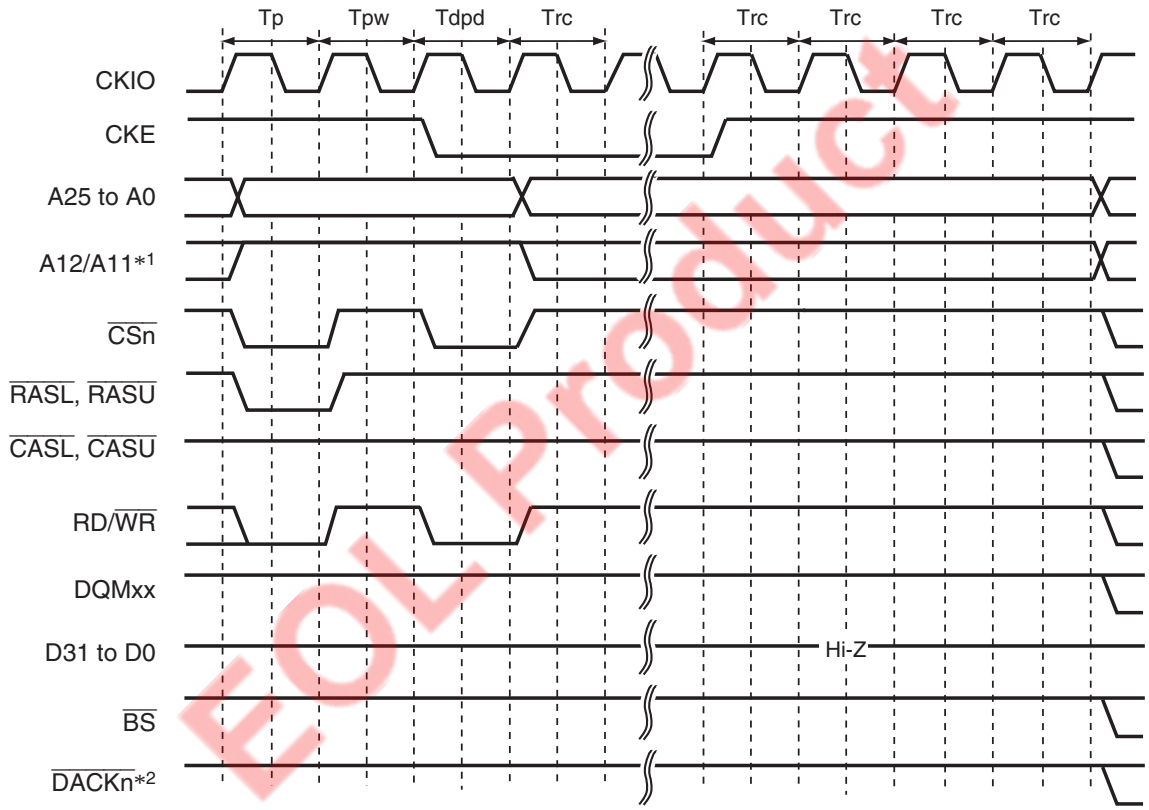
Notes: 1. Address pin to be connected to pin BA1 of SDRAM.
2. Address pin to be connected to pin BA0 of SDRAM.
3. Address pin to be connected to pin A10 of SDRAM.
4. The waveform for DACKn is when active low is specified.

Figure 12.34 EMRS Command Issue Timing

- Deep power-down mode

  The low-power SDRAM supports the deep power-down mode as a low-power consumption mode. In the partial self-refresh function, self-refresh is performed on a specific area. In the deep power-down mode, self-refresh will not be performed on any memory area. This mode is effective in systems where all of the system memory areas are used as work areas.

If the RMODE bit in the SDCR is set to 1 while the DEEP and RFSH bits in the SDCR are set to 1, the low-power SDRAM enters the deep power-down mode. If the RMODE bit is cleared to 0, the CKE signal is pulled high to cancel the deep power-down mode. Before executing an access after returning from the deep power-down mode, the power-up sequence must be re-executed.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.35 Deep Power-Down Mode Transition Timing

12.5.7 Burst ROM (Clock Asynchronous) Interface

The burst ROM (clock asynchronous) interface is used to access a memory with a high-speed read function using a method of address switching called the burst mode or page mode. In a burst ROM (clock asynchronous) interface, basically the same access as the normal space is performed, but the 2nd and subsequent accesses are performed only by changing the address, without negating the $\overline{RD}$ signal at the end of the 1st cycle. In the 2nd and subsequent accesses, addresses are changed at the falling edge of the CKIO.

For the 1st access cycle, the number of wait cycles specified by the W3 to W0 bits in the CSnWCR register is inserted. For the 2nd and subsequent access cycles, the number of wait cycles specified by the W1 to W0 bits in the CSnWCR register is inserted.

In the access to the burst ROM (clock asynchronous), the $\overline{BS}$ signal is asserted only to the first access cycle. An external wait input is valid only to the first access cycle. In the single access or write access that do not perform the burst operation in the page flash ROM interface, access timing is same as a normal space. Table 12.17 lists a relationship between bus width, access size, and the number of bursts. Figure 12.36 shows a timing chart.

Table 12.17 Relationship between Bus Width, Access Size, and Number of Bursts

Bus Width	CSnWCR. BEN Bit	Access Size	Number of Bursts	Number of Accesses
8 bits	Not affected	8 bits	1	1
	Not affected	16 bits	2	1
	Not affected	32 bits	4	1
	0	16 bytes	16	1
	1		4	4
16 bits	Not affected	8 bits	1	1
	Not affected	16 bits	1	1
	Not affected	32 bits	2	1
	0	16 bytes	8	1
	1		2	4
32 bits	Not affected	8 bits	1	1
	Not affected	16 bits	1	1
	Not affected	32 bits	1	1
	Not affected	16 bytes	4	1

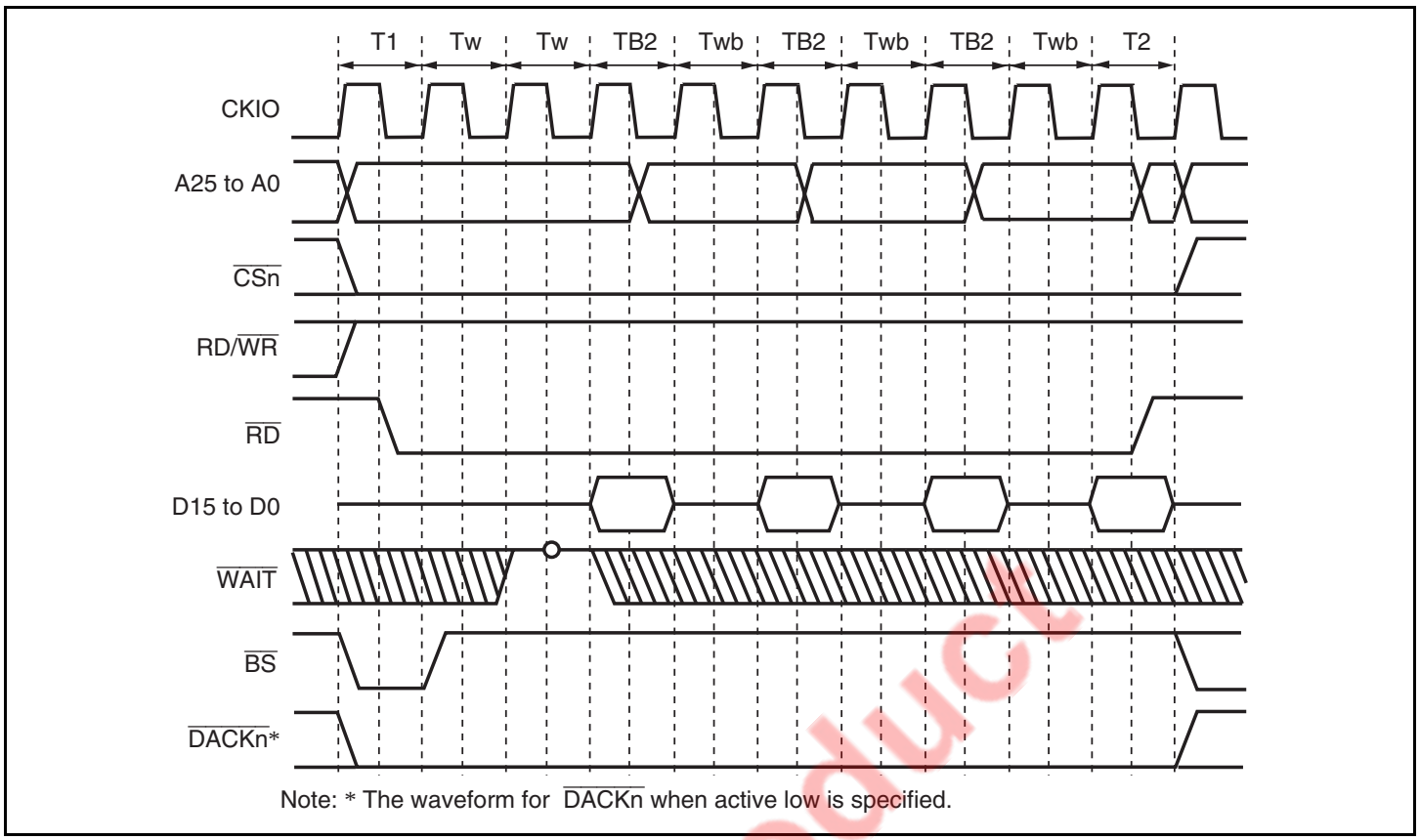


Figure 12.36 Burst ROM Access Timing (Clock Asynchronous)
(Bus Width = 32 Bits, 16-Byte Transfer (Number of Burst 4), Wait Cycles Inserted in First Access = 2, Wait Cycles Inserted in Second and Subsequent Accesses = 1)

12.5.8 Byte-Selection SRAM Interface

The byte-selection SRAM interface is for access to an SRAM which has a byte-selection pin (WEn). This interface has 16-bit data pins and accesses SRAMs having upper and lower byte selection pins, such as UB and LB.

When the BAS bit in the CSnWCR register is cleared to 0 (initial value), the write access timing of the byte-selection SRAM interface is the same as that for the normal space interface. While in read access of a byte-selection SRAM interface, the byte-selection signal is output from the WEn pin, which is different from that for the normal space interface. The basic access timing is shown in figure 12.37. In write access, data is written to the memory according to the timing of the byte-selection pin (WEn). For details, please refer to the Data Sheet for the corresponding memory.

If the BAS bit in the CSnWCR register is set to 1, the WEn pin and RD/WR pin timings change. Figure 12.38 shows the basic access timing. In write access, data is written to the memory according to the timing of the write enable pin (RD/WR). The data hold timing from RD/WR negation to data write must be acquired by setting the HW1 and HW0 bits in the CSnWCR register. Figure 12.39 shows the access timing when a software wait is specified.

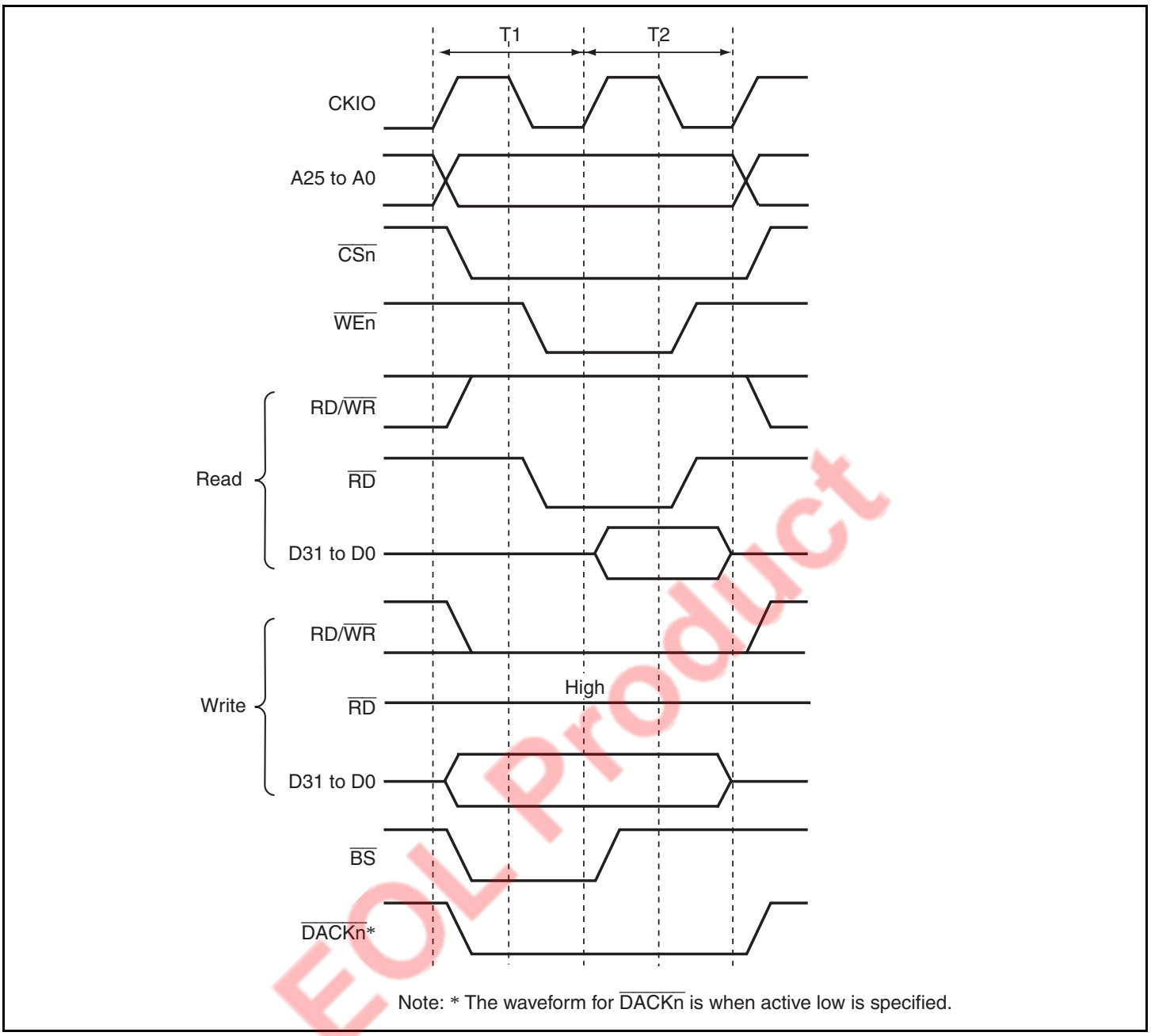
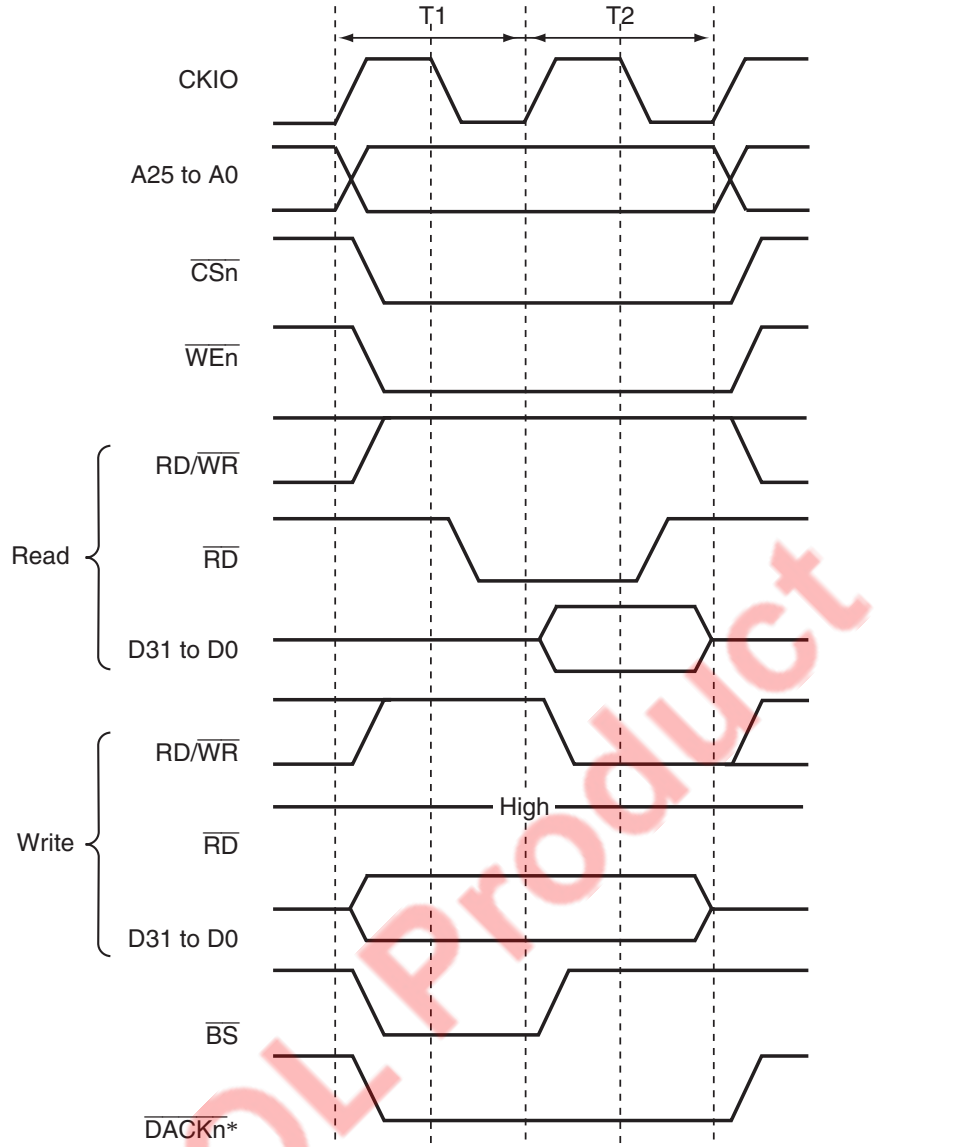


Figure 12.37 Byte-Selection RAM Basic Access Timing (BAS = 0)



Note: * The waveform for DACKn is when active low is specified.

Figure 12.38 Byte-Selection RAM Basic Access Timing (BAS = 1)

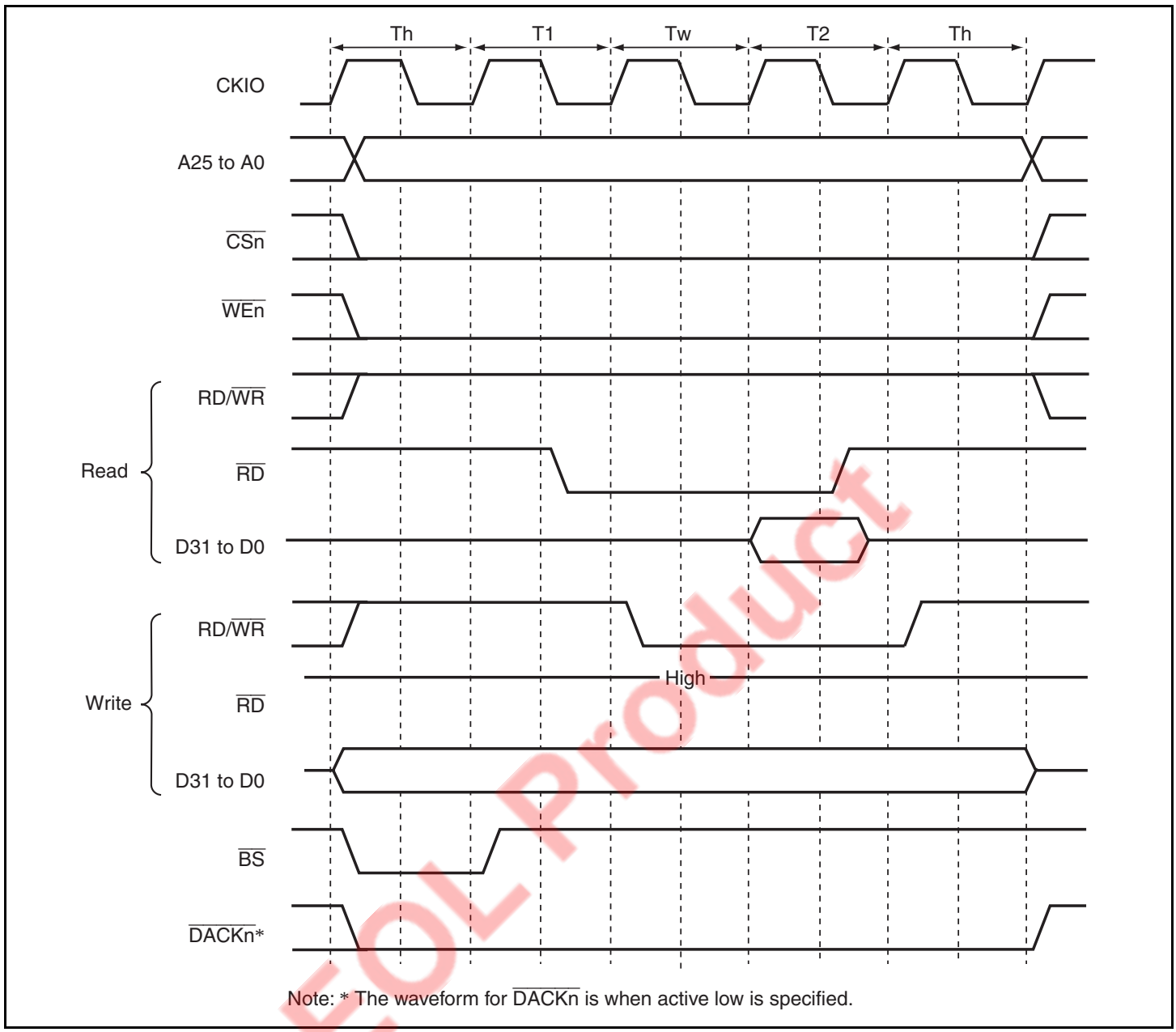


Figure 12.39 Byte-Selection SRAM Wait Timing (BAS = 1)
(SW[1:0] = 01, WR[3:0] = 0001, HW[1:0] = 01)

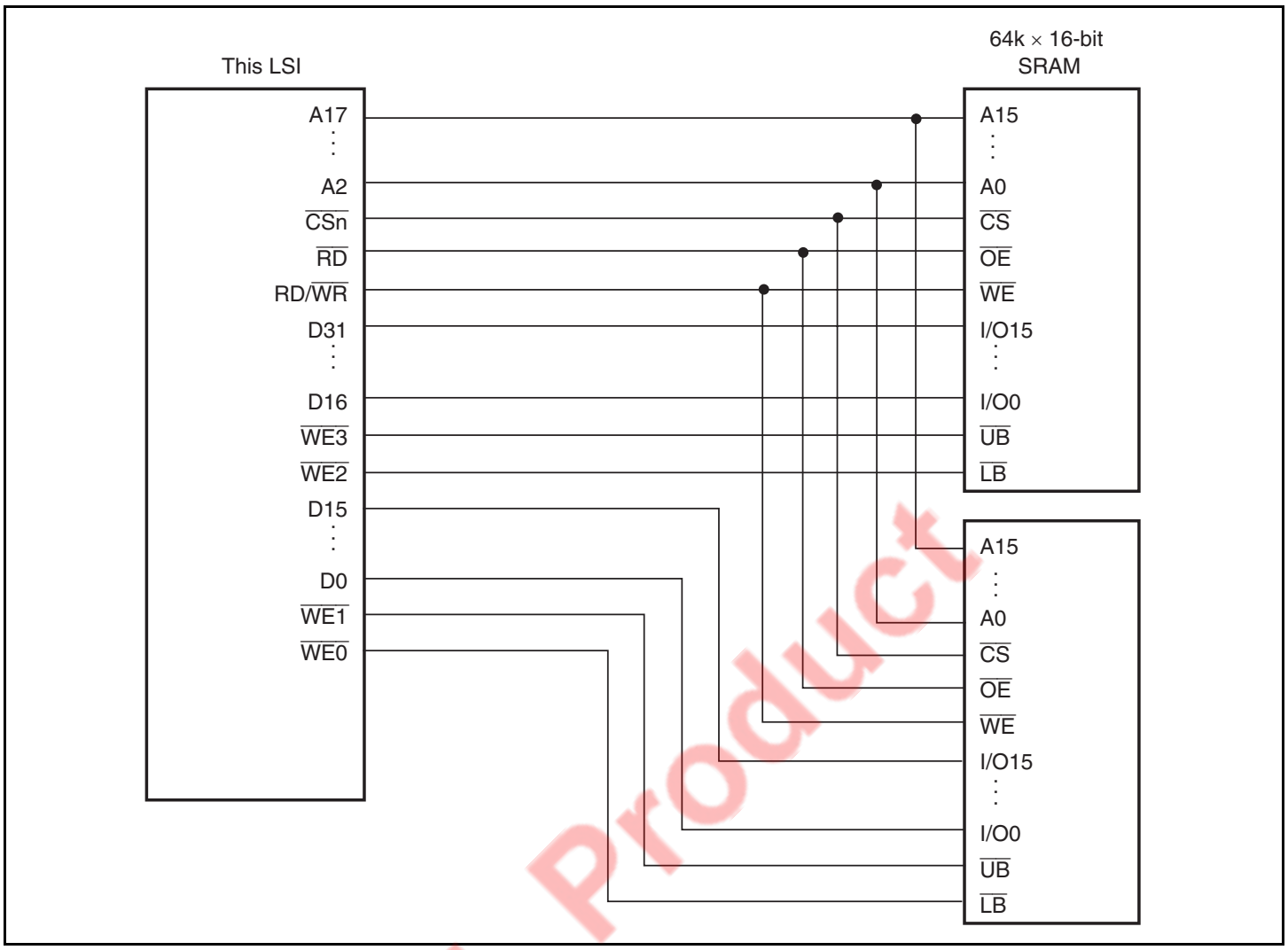


Figure 12.40 Example of Connection with 32-Bit Data-Width Byte-Selection SRAM

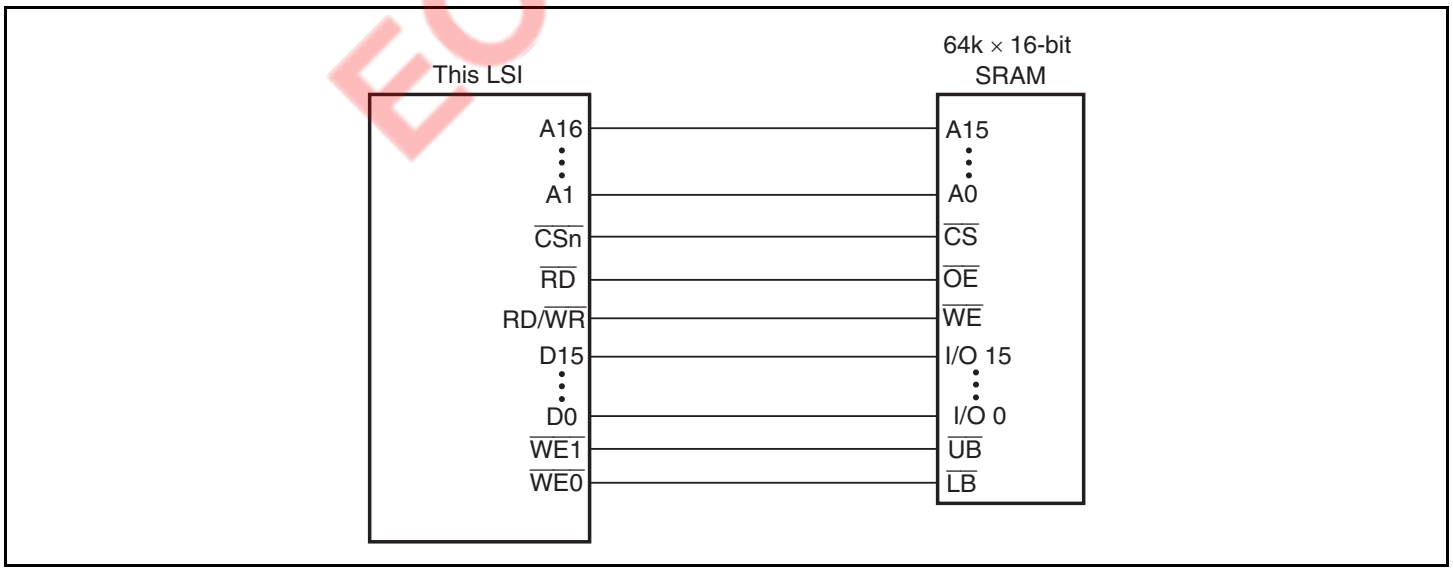


Figure 12.41 Example of Connection with 16-Bit Data-Width Byte-Selection SRAM

12.5.9 Burst MPX-I/O Interface

Figure 12.42 shows an example of a connection between the LSI and an MPX device. Figures 12.43 to 12.46 show the burst MPX space access timings.

Area 6 can be specified as the address/data multiplex I/O (MPX-I/O) interface using the TYPE2 to TYPE0 bits in the CS6BCR register. This MPX-I/O interface enables the LSI to be easily connected to an external memory controller chip that uses an address/data multiplexed 32-bit single bus. In this case, the address and the access size for the MPX-I/O interface are output to D25 to D0 and D31 to D29, respectively, in address cycles. For the access sizes of D31 to D29, see the description of the CS6BWCR register (burst MPX-I/O). Address pins A25 to A0 are used to output normal addresses.

In the burst MPX-I/O interface, the bus size is fixed at 32 bits. The BSZ1 and BSZ0 bits in CS6BBCR must be specified as 32 bits.

In the MPX-I/O interface, a software wait or hardware wait can be inserted using the $\overline{\text{WAIT}}$ pin. In read cycles, a wait cycle is inserted automatically following the address output even if the software wait insertion is specified as 0.

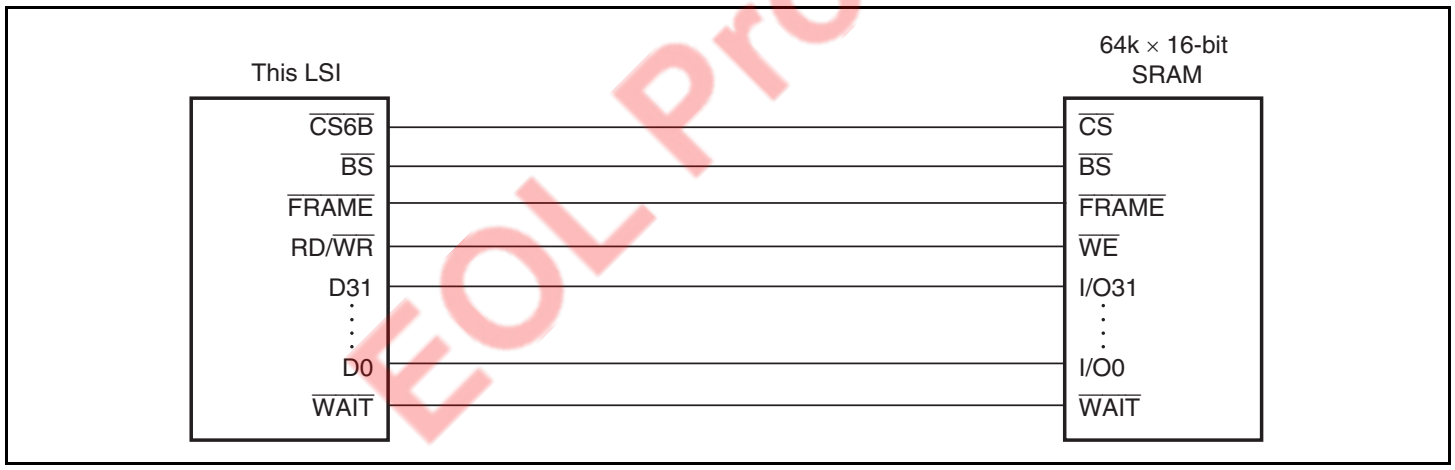
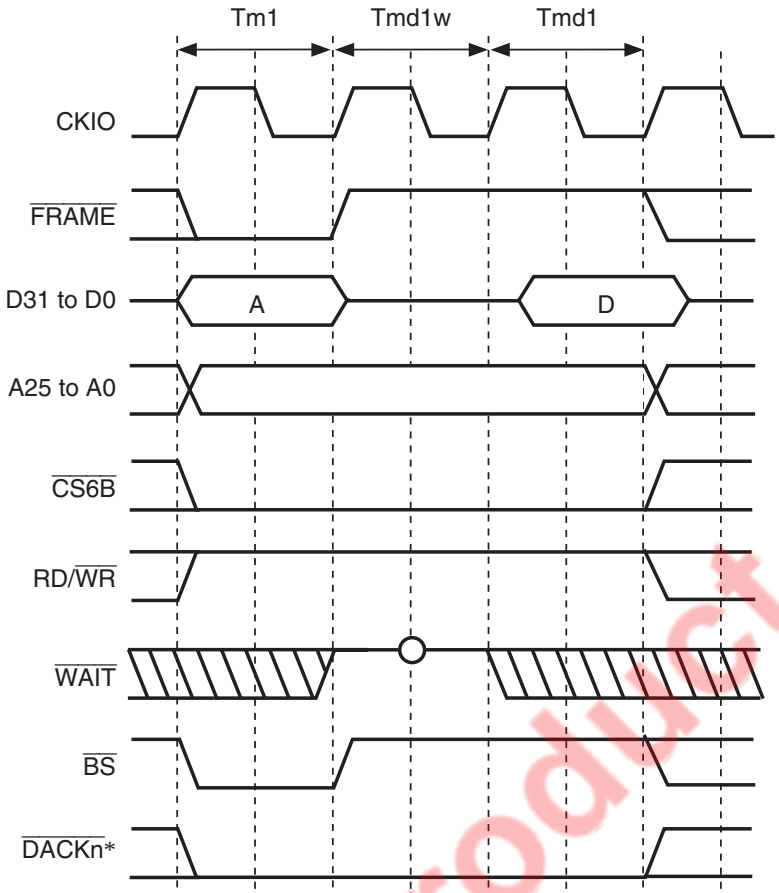
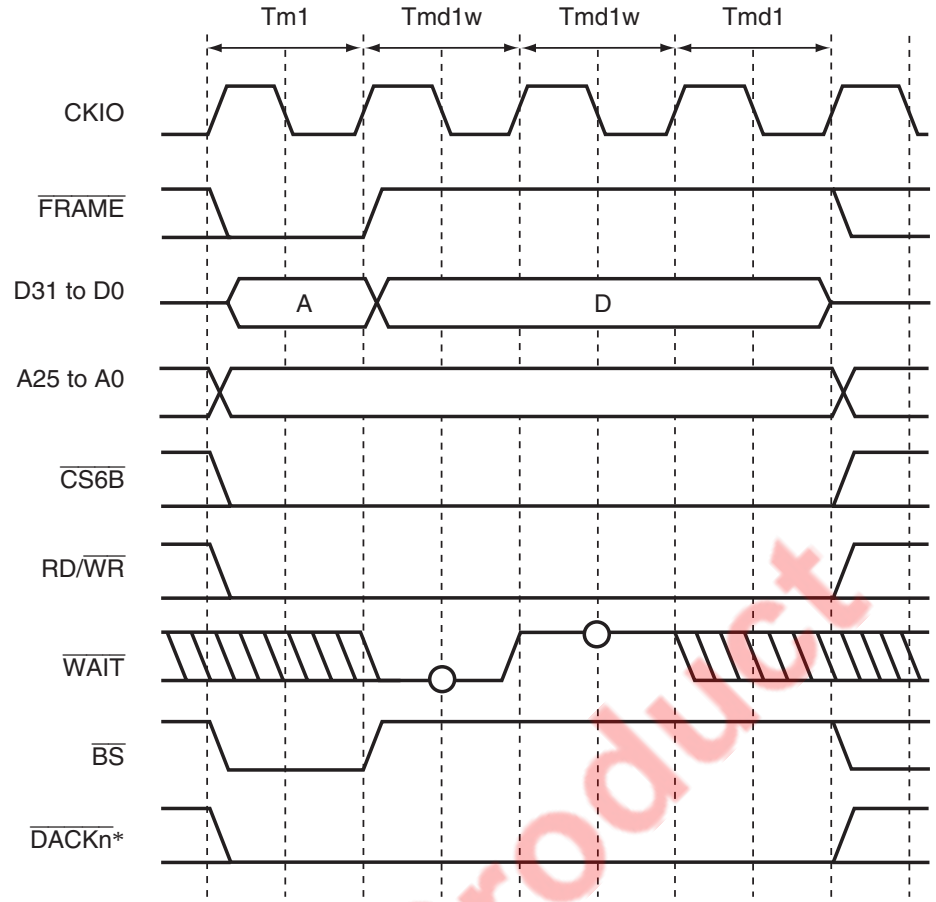


Figure 12.42 Burst MPX Device Connection Example



Note: * The waveform for DACKn is when active low is specified.

Figure 12.43 Burst MPX Space Access Timing (Single Read, No Wait, or Software Wait 1)



Note: * The waveform for DACKn is when active low is specified.

Figure 12.44 Burst MPX Space Access Timing (Single Write, Software Wait 1, Hardware Wait 1)

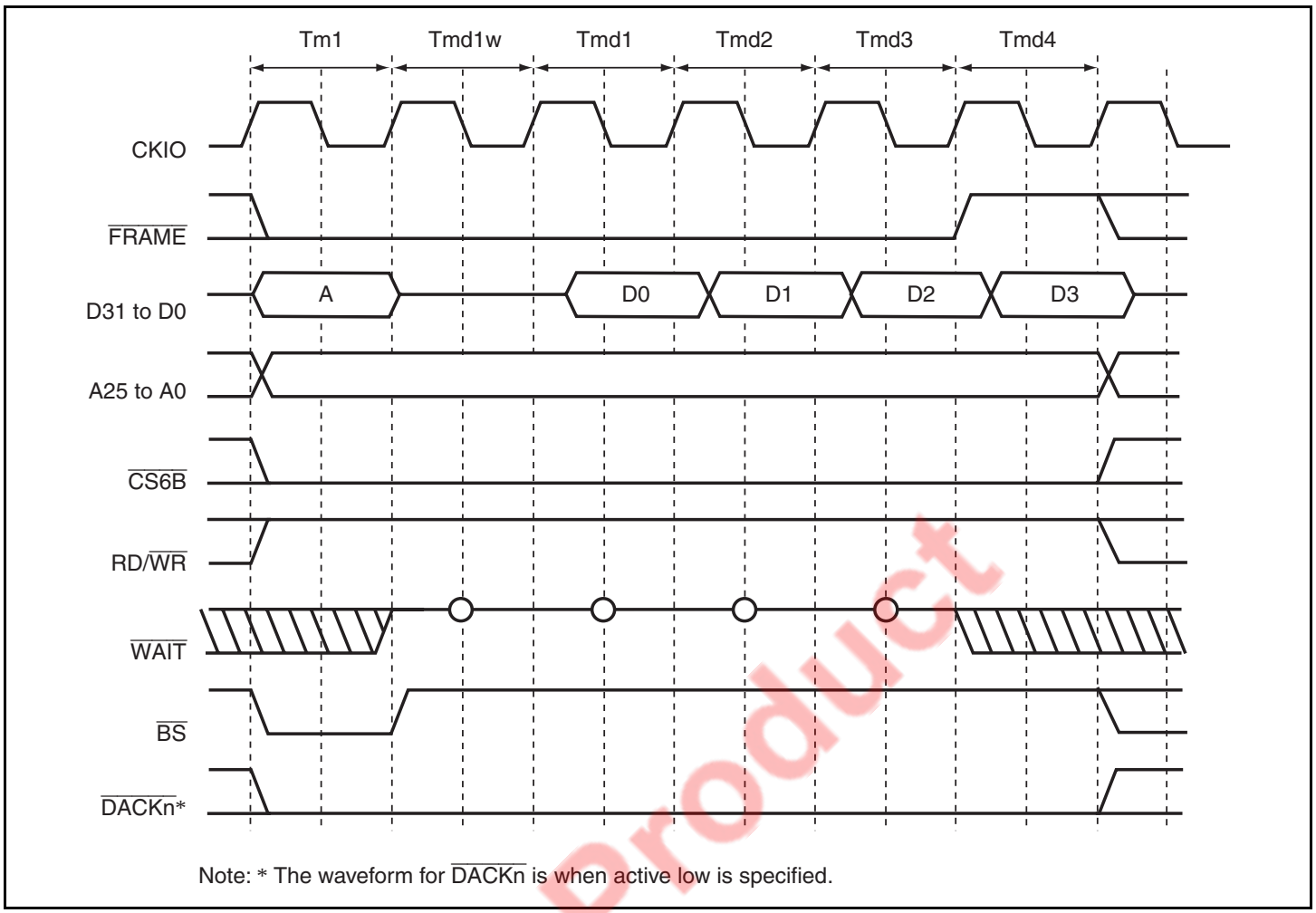
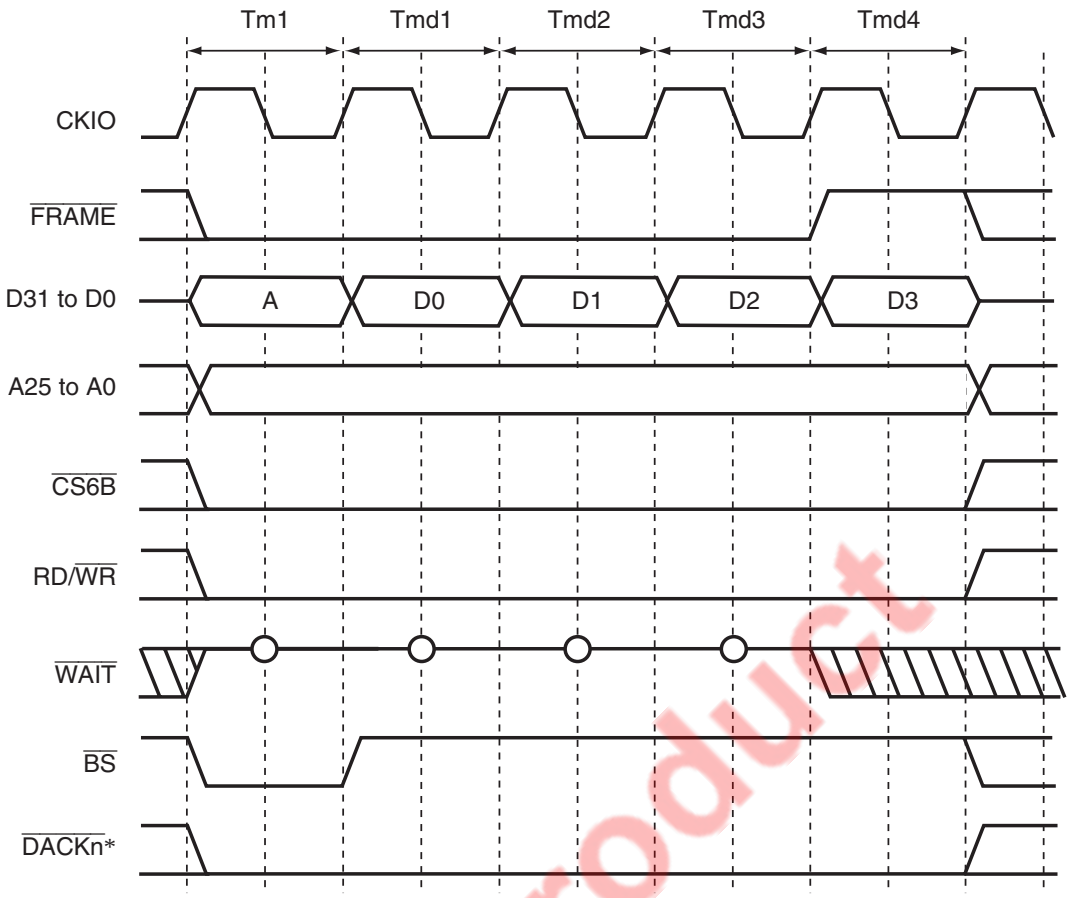


Figure 12.45 Burst MPX Space Access Timing
(Burst Read, No Wait, or Software Wait 1, CS6BWCR.MPXMD = 0)



Note: * The waveform for DACKn is when active low is specified.

Figure 12.46 Burst MPX Space Access Timing (Burst Write, No Wait, CS6BWCR.MPXMD = 0)

12.5.10 Burst ROM Interface (Clock Synchronous)

The burst ROM (clock synchronous) interface is supported to access a ROM with a synchronous burst function at high speed. The burst ROM interface accesses the burst ROM in the same way as a normal space. This interface is valid only for area 0.

In the first access cycle, wait cycles are inserted. In this case, the number of wait cycles to be inserted is specified by the W3 to W0 bits in CS0WCR. In the second and subsequent cycles, the number of wait cycles to be inserted is specified by the BW1 and BW0 bits in CS0WCR.

While the burst ROM is accessed (clock synchronous), the $\overline{BS}$ signal is asserted only for the first access cycle and an external wait input is also valid for the first access cycle.

If the bus width is 16 bits, the burst length must be specified as 8. If the bus width is 32 bits, the burst length must be specified as 4. The burst ROM interface does not support the 8-bit bus width for the burst ROM.

The burst ROM interface performs burst operations for all read accesses. For example, in a longword access over a 16-bit bus, valid 16-bit data is read two times and invalid 16-bit data is read six times. These invalid data read cycles increase the memory access time and degrade the program execution speed and DMA transfer speed. To prevent this problem, it is recommend using a 16-byte read by cache fill or 16-byte read by the DMAC. Thus, the burst ROM (clock synchronous) should be accessed with the cache having been set on. The burst ROM interface performs write accesses in the same way as normal space access.

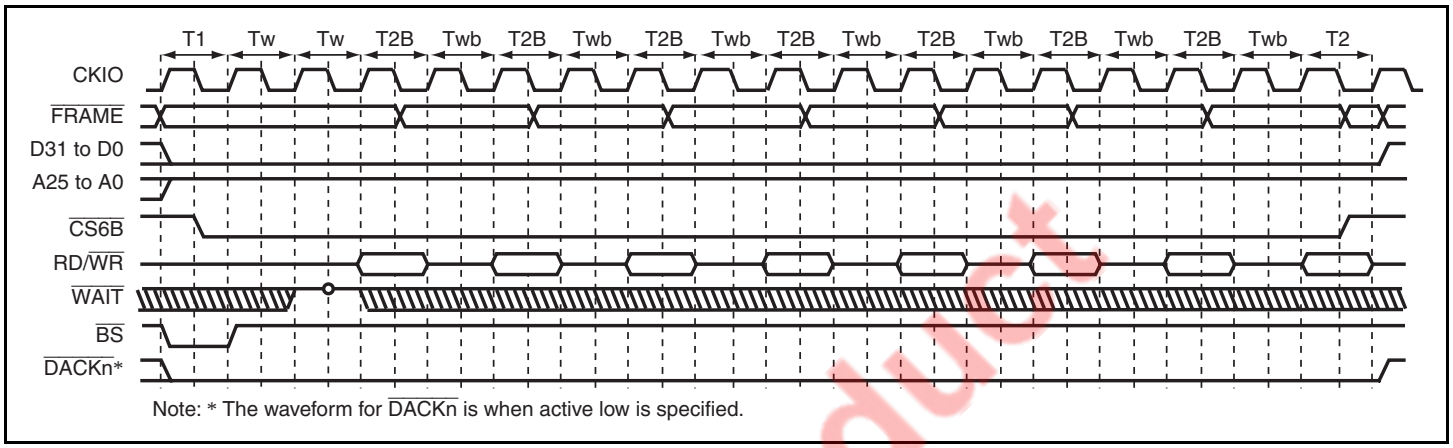


Figure 12.47 Burst ROM Access Timing (Clock Synchronous)
(Burst Length = 8, Wait Cycles Inserted in First Access = 2,
Wait Cycles Inserted in Second and Subsequent Accesses = 1)

12.5.11 Wait between Access Cycles

As the operating frequency of LSIs becomes higher, the off-operation of the data buffer often collides with the next data access when the read operation from devices with slow access speed is completed. As a result of these collisions, the reliability of the device is low and malfunctions may occur. A function that avoids data collisions by inserting wait cycles between continuous access cycles has been newly added.

The number of wait cycles between access cycles can be set by bits IWW2 to IWW0, IWRWD2 to IWRWD0, IWRWS2 to IWRWS0, IWRRD2 to IWRRD0, and IWRRS2 to IWRRS 0 in the CSnBCR register , and bit DMAIW2 to DMAIW0 and DMAIWA in CMNCR. The conditions for setting the wait cycles between access cycles (idle cycles) are shown below.

1. Continuous accesses are write-read or write-write
2. Continuous accesses are read-write for different spaces
3. Continuous accesses are read-write for the same space
4. Continuous accesses are read-read for different spaces
5. Continuous accesses are read-read for the same space

6. Data output from an external device caused by DMA single address transfer is followed by data output from another device that includes this LSI (DMAIWA = 0)
   For details, see the description of the DMAIWA bit in the CMNCR register.
7. Data output from an external device caused by DMA single address transfer is followed by any type of access (DMAIWA = 1)

Besides the wait cycles between access cycles (idle cycles) described above, idle cycles must be inserted to reserve the minimum pulse width for an interface with an internal bus and a multiplexed pin (WEn).

8. Idle cycle of the external bus for the interface with the internal bus
   A. Insert one idle cycle immediately before a write access cycle after an external bus idle cycle or a read cycle.
   B. Insert one idle cycle to transfer the read data to the internal bus when a read cycle of the external bus terminates.
      Insert two to three idle cycles including the idle cycle in A. for the write cycle immediately after a read cycle.
9. Idle cycle of the external bus for accessing different memory
   For accessing different memory, insert idle cycles as follows. The byte-selection SRAM interface with the BAS bit = 1 specified is handled as an SDRAM interface because the WEn change timing is identical.
   A. Insert one idle cycle to access the interface other than the SDRAM interface after the write access cycle is performed in the SDRAM interface.
   B. Insert one idle cycle to access the SDRAM interface after the normal space interface with the external wait invalidated or the byte-selection SRAM interface with the BAS bit = 0 specified is accessed.
   C. Insert one idle cycle to access the SDRAM interface after the MPX-IO interface is accessed.
   D. Insert one idle cycle to access the MPX-IO interface from the external bus that is in the idle status.
   E. Insert one idle cycle to access the MPX-IO interface after a read cycle is performed in the normal space interface, byte-selection SRAM interface with the BAS bit = 0 specified or the SDRAM interface.
   F. Insert two idle cycles to access the MPX-IO interface after a write cycle is performed in the SDRAM interface.
   G. Insert one idle cycle to access the SDRAM interface which is not in the low frequency mode after the interface in the SDRAM low frequency mode (SDCR.SLOW = 1) is accessed.

Tables 12.18 to 12.22 lists the minimum number of idle cycles to be inserted for the normal space interface and the SDRAM interface. The CSnBCR Idle Setting column in the tables describes the number of idle cycles to be set for IWW, IWRWD, IWRWS, IWRRD, and IWRRS.

Table 12.18 Minimum Number of Idle Cycles between CPU Access Cycles for the Normal Space Interface

BSC Register Setting		When Access Size is Less than Bus Width				When Access Size Exceeds Bus Width					
CSnWCR. WM Setting	CSnBCR Idle Setting	Read to Read	Write to Write	Read to Write	Write to Read	Continuous Read*[1]	Continuous Write*[1]	Read to Read*[2]	Write to Write*[2]	Read to Write*[2]	Write to Read*[2]
1	0	1/1/1/2	1/1/2/3	3/3/4/5	0/0/0/0	0/0/0/0	0/0/0/0	1/1/1/2	0/0/0/1	3/3/4/5	0/0/0/0
0	0	1/1/1/2	1/1/2/3	3/3/4/5	1/1/1/1	1/1/1/1	1/1/1/1	1/1/1/2	1/1/1/1	3/3/4/5	1/1/1/1
1	1	1/1/1/2	1/1/2/3	3/3/4/5	1/1/1/1	1/1/1/1	1/1/1/1	1/1/1/2	1/1/1/1	3/3/4/5	1/1/1/1
0	1	1/1/1/2	1/1/2/3	3/3/4/5	1/1/1/1	1/1/1/1	1/1/1/1	1/1/1/2	1/1/1/1	3/3/4/5	1/1/1/1
1	2	2/2/2/2	2/2/2/3	3/3/4/5	2/2/2/2	2/2/2/2	2/2/2/2	2/2/2/2	2/2/2/2	3/3/4/5	2/2/2/2
0	2	2/2/2/2	2/2/2/3	3/3/4/5	2/2/2/2	2/2/2/2	2/2/2/2	2/2/2/2	2/2/2/2	3/3/4/5	2/2/2/2
1	4	4/4/4/4	4/4/4/4	4/4/4/5	4/4/4/4	4/4/4/4	4/4/4/4	4/4/4/4	4/4/4/4	4/4/4/5	4/4/4/4
0	4	4/4/4/4	4/4/4/4	4/4/4/5	4/4/4/4	4/4/4/4	4/4/4/4	4/4/4/4	4/4/4/4	4/4/4/5	4/4/4/4
1	6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6
0	6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6	6/6/6/6
0, 1	n (n>=8)	n/n/n/n	n/n/n/n	n/n/n/n	n/n/n/n	n/n/n/n	n/n/n/n	n/n/n/n	n/n/n/n	n/n/n/n	n/n/n/n

Notes: The minimum number of idle cycles is described sequentially for Iϕ: Bϕ (4:1/3:1/2:1/1:1).

1. Minimum number of idle cycles between the upper and lower 16-bit access cycles in the 32-bit access cycle when the bus width is 16 bits, and the minimum number of idle cycles between continuous access cycles during 16-byte transfer
2. Minimum number of idle cycles for other than the above cases

Table 12.19 Minimum Number of Idle Cycles between Access Cycles during DMAC Dual Address Mode Transfer for the Normal Space Interface

BSC Register Setting		When Access Size is Less than Bus Width		When Access Size Exceeds Bus Width			
CSnWCR. WM Setting	CSnBCR Idle Setting	Read to Write	Write to Read	Continuous Read*[1]	Read to Write*[2]	Continuous Write*[1]	Write to Read*[2]
1	0	2	0	0	2	0	0
0	0	2	1	1	2	1	1
1	1	2	1	1	2	1	1
0	1	2	1	1	2	1	1
1	2	2	2	2	2	2	2
0	2	2	2	2	2	2	2
1	4	4	4	4	4	4	4
0	4	4	4	4	4	4	4
0, 1	n (n≥6)	n	n	n	n	n	n

Notes: DMAC is operated by Bϕ. The minimum number of idle cycles is not affected by changing a clock ratio.

1. Minimum number of idle cycles between the upper and lower 16-bit access cycles in the 32-bit access cycle when the bus width is 16 bits, and the minimum number of idle cycles between continuous access cycles during 16-byte transfer
2. Minimum number of idle cycles for other than the above cases.

Table 12.20 Minimum Number of Idle Cycles during DMAC Single Address Mode Transfer to the Normal Space Interface from the External Device with $\overline{\text{DACK}}$

(1) Transfer from the external device with $\overline{\text{DACK}}$ to the normal space interface

BSC Register Setting*[3]			When Access Size is Less than Bus Width	
CSnWCR.WM Setting	CMNCR.DMAIWA Setting	CMNCR.DMAIW Idle Setting	Continuous Transfer*[1]	Non-Continuous Transfer*[2]
1	0	—	0	2
0	0	—	1	2
1	1	0	0	2
0	1	0	1	2
1	1	1	1	2
0	1	1	1	2
1	1	2	2	2
0	1	2	2	2
1	1	4	4	4
0	1	4	4	4
0, 1	1	n (n≥6)	n	n

(2) Transfer from the normal space interface to the external device with $\overline{\text{DACK}}$

BSC Register Setting*4		When Access Size is Less than Bus Width	
CSnWCR.WM Setting	CSnBCR Idle Setting	Continuous Transfer*1	Non-Continuous Transfer*2
1	0	0	3
0	0	1	3
1	1	1	3
0	1	1	3
1	2	2	3
0	2	2	3
1	4	4	4
0	4	4	4
0, 1	n (n≥6)	n	n

Notes: DMAC is operated by Bϕ. The minimum number of idle cycles is not affected by changing a clock ratio.

1. Minimum number of idle cycles between the upper and lower 16-bit access cycles in the 32-bit access cycle when the bus width is 16 bits, and the minimum number of idle cycles between continuous access cycles during 16-byte transfer
2. Other than the above cases.
3. For single transfer from the external device with $\overline{\text{DACK}}$ to the normal space interface, the minimum number of idle cycles is not affected by the IWW, IWRWD, IWRWS, IWRRD, and IWRRS bits in CSnBCR.
4. For single transfer from the normal space interface to the external device with $\overline{\text{DACK}}$, the minimum number of idle cycles is not affected by the DMAIWA and DMAIW bits in CMNCR.

Table 12.21 Minimum Number of Idle Cycles between Access Cycles of CPU and the DMAC Dual Address Mode for the SDRAM Interface

BSC Register Setting			CPU Access				DMAC Access	
CSnBCR Idle Setting	CS3WCR. WTRP Setting	CS3WCR. TRWL Setting	Read to Read	Write to Write	Read to Write	Write to Read	Read to Write	Write to Read
0	0	0	1/1/1/2	1/1/2/3	3/3/4/5	0/0/0/0	2	0
0	0	1	1/1/1/2	1/1/2/3	3/3/4/5	1/1/1/1	2	1
0	0	2	1/1/1/2	2/2/2/3	3/3/4/5	2/2/2/2	2	2
0	0	3	1/1/1/2	3/3/3/3	3/3/4/5	3/3/3/3	2	3
0	1	0	2/2/2/2	1/1/2/3	3/3/4/5	1/1/1/1	2	1
0	1	1	2/2/2/2	2/2/2/3	3/3/4/5	2/2/2/2	2	2
0	1	2	2/2/2/2	3/3/3/3	3/3/4/5	3/3/3/3	2	3
0	1	3	2/2/2/2	4/4/4/4	3/3/4/5	4/4/4/4	2	4
0	2	0	3/3/3/3	2/2/2/3	3/3/4/5	2/2/2/2	3	2
0	2	1	3/3/3/3	3/3/3/3	3/3/4/5	3/3/3/3	3	3
0	2	2	3/3/3/3	4/4/4/4	3/3/4/5	4/4/4/4	3	4
0	2	3	3/3/3/3	5/5/5/5	3/3/4/5	5/5/5/5	3	5
0	3	0	4/4/4/4	3/3/3/3	4/4/4/5	3/3/3/3	4	3
0	3	1	4/4/4/4	4/4/4/4	4/4/4/5	4/4/4/4	4	4
0	3	2	4/4/4/4	5/5/5/5	4/4/4/5	5/5/5/5	4	5
0	3	3	4/4/4/4	6/6/6/6	4/4/4/5	6/6/6/6	4	6
1	0	0	2/2/2/2	1/1/2/3	3/3/4/5	1/1/1/1	2	1
1	0	1	2/2/2/2	1/1/2/3	3/3/4/5	1/1/1/1	2	1
1	0	2	2/2/2/2	2/2/2/3	3/3/4/5	2/2/2/2	2	2
1	0	3	2/2/2/2	3/3/3/3	3/3/4/5	3/3/3/3	2	3
1	1	0	2/2/2/2	1/1/2/3	3/3/4/5	1/1/1/1	2	1
1	1	1	2/2/2/2	2/2/2/3	3/3/4/5	2/2/2/2	2	2
1	1	2	2/2/2/2	3/3/3/3	3/3/4/5	3/3/3/3	2	3
1	1	3	2/2/2/2	4/4/4/4	3/3/4/5	4/4/4/4	2	4
1	2	0	3/3/3/3	2/2/2/3	3/3/4/5	2/2/2/2	3	2
1	2	1	3/3/3/3	3/3/3/3	3/3/4/5	3/3/3/3	3	3
1	2	2	3/3/3/3	4/4/4/4	3/3/4/5	4/4/4/4	3	4

BSC Register Setting			CPU Access				DMAC Access	
CSnBCR Idle Setting	CS3WCR. WTRP Setting	CS3WCR. TRWL Setting	Read to Read	Write to Write	Read to Write	Write to Read	Read to Write	Write to Read
1	2	3	3/3/3/3	5/5/5/5	3/3/4/5	5/5/5/5	3	5
1	3	0	4/4/4/4	3/3/3/3	4/4/4/5	3/3/3/3	4	3
1	3	1	4/4/4/4	4/4/4/4	4/4/4/5	4/4/4/4	4	4
1	3	2	4/4/4/4	5/5/5/5	4/4/4/5	5/5/5/5	4	5
1	3	3	4/4/4/4	6/6/6/6	4/4/4/5	6/6/6/6	4	6
2	0	0	3/3/3/3	2/2/2/3	3/3/4/5	2/2/2/2	3	2
2	0	1	3/3/3/3	2/2/2/3	3/3/4/5	2/2/2/2	3	2
2	0	2	3/3/3/3	2/2/2/3	3/3/4/5	2/2/2/2	3	2
2	0	3	3/3/3/3	3/3/3/3	3/3/4/5	3/3/3/3	3	3
2	1	0	3/3/3/3	2/2/2/2	3/3/4/5	2/2/2/2	3	2
2	1	1	3/3/3/3	2/2/2/2	3/3/4/5	2/2/2/2	3	2
2	1	2	3/3/3/3	3/3/3/3	3/3/4/5	3/3/3/3	3	3
2	1	3	3/3/3/3	4/4/4/4	3/3/4/5	4/4/4/4	3	4
2	2	0	3/3/3/3	2/2/2/3	3/3/4/5	2/2/2/2	3	2
2	2	1	3/3/3/3	3/3/3/3	3/3/4/5	3/3/3/3	3	3
2	2	2	3/3/3/3	4/4/4/4	3/3/4/5	4/4/4/4	3	4
2	2	3	3/3/3/3	5/5/5/5	3/3/4/5	5/5/5/5	3	5
2	3	0	4/4/4/4	3/3/3/3	4/4/4/5	3/3/3/3	4	3
2	3	1	4/4/4/4	4/4/4/4	4/4/4/5	4/4/4/4	4	4
2	3	2	4/4/4/4	5/5/5/5	4/4/4/5	5/5/5/5	4	5
2	3	3	4/4/4/4	6/6/6/6	4/4/4/5	6/6/6/6	4	6
4	0	0	5/5/5/5	4/4/4/4	5/5/5/5	4/4/4/4	5	4
4	0	1	5/5/5/5	4/4/4/4	5/5/5/5	4/4/4/4	5	4
4	0	2	5/5/5/5	4/4/4/4	5/5/5/5	4/4/4/4	5	4
4	0	3	5/5/5/5	4/4/4/4	5/5/5/5	4/4/4/4	5	4
4	1	0	5/5/5/5	4/4/4/4	5/5/5/5	4/4/4/4	5	4
4	1	1	5/5/5/5	4/4/4/4	5/5/5/5	4/4/4/4	5	4
4	1	2	5/5/5/5	4/4/4/4	5/5/5/5	4/4/4/4	5	4
4	1	3	5/5/5/5	4/4/4/4	5/5/5/5	4/4/4/4	5	4

BSC Register Setting			CPU Access				DMAC Access	
CSnBCR Idle Setting	CS3WCR. WTRP Setting	CS3WCR. TRWL Setting	Read to Read	Write to Write	Read to Write	Write to Read	Read to Write	Write to Read
4	2	0	5/5/5/5	4/4/4/4	5/5/5/5	4/4/4/4	5	4
4	2	1	5/5/5/5	4/4/4/4	5/5/5/5	4/4/4/4	5	4
4	2	2	5/5/5/5	4/4/4/4	5/5/5/5	4/4/4/4	5	4
4	2	3	5/5/5/5	5/5/5/5	5/5/5/5	5/5/5/5	5	5
4	3	0	5/5/5/5	4/4/4/4	5/5/5/5	4/4/4/4	5	4
4	3	1	5/5/5/5	4/4/4/4	5/5/5/5	4/4/4/4	5	4
4	3	2	5/5/5/5	5/5/5/5	5/5/5/5	5/5/5/5	5	5
4	3	3	5/5/5/5	6/6/6/6	5/5/5/5	6/6/6/6	5	6
n (n>=6)	—	—	All n+1	n/n/n/n	All n+1	n/n/n/n	n+1	n

Notes: The minimum number of idle cycles in CPU Access is described sequentially for Iϕ:Bϕ (4:1/3:1/2:1/1:1).

1. DMAC is operated by Bϕ. The minimum number of idle cycles is not affected by changing a clock ratio.

Table 12.22 Minimum Number of Idle Cycles between Access Cycles of the DMAC Single Address Mode for the SDRAM Interface

(1) Transfer from the external device with $\overline{\text{DACK}}$ to the SDRAM interface

BSC Register Setting*[2]			
CMNCR.DMAIW Setting	CS3WCR.WTRP Setting	CS3WCR.TRWL Setting	Minimum Number of Idle Cycles
0	0	0	3
0	0	1	3
0	0	2	3
0	0	3	3
0	1	0	3
0	1	1	3
0	1	2	3
0	1	3	4
0	2	0	3
0	2	1	3
0	2	2	4
0	2	3	5
0	3	0	3
0	3	1	4
0	3	2	5
0	3	3	6
1	0	0	3
1	0	1	3
1	0	2	3
1	0	3	3
1	1	0	3
1	1	1	3
1	1	2	3
1	1	3	4
1	2	0	3
1	2	1	3
1	2	2	4
1	2	3	5
1	3	0	3
1	3	1	4

BSC Register Setting*[2]			
CMNCR.DMAIW Setting	CS3WCR.WTRP Setting	CS3WCR.TRWL Setting	Minimum Number of Idle Cycles
1	3	2	5
1	3	3	6
2	0	1	3
2	0	2	3
2	0	3	3
2	1	0	3
2	1	1	3
2	1	2	3
2	1	3	4
2	2	0	3
2	2	1	3
2	2	2	4
2	2	3	5
2	3	0	3
2	3	1	4
2	3	2	5
2	3	3	6
4	0	0	4
4	0	1	4
4	0	2	4
4	0	3	4
4	1	0	4
4	1	1	4
4	1	2	4
4	1	3	4
4	2	0	4
4	2	1	4
4	2	2	4
4	2	3	5
4	3	0	4
4	3	1	4
4	3	2	5
4	3	3	6
n (n>=6)	—	—	n

(2) Transfer from the SDRAM interface to the external device with $\overline{\text{DACK}}$

BSC Register Setting*2		Minimum Number of Idle Cycles
CS3BCR Idle Setting	CS3WCR.WTRP Setting	
0	0	3
0	1	3
0	2	3
0	3	4
1	0	3
1	1	3
1	2	3
1	3	4
2	0	3
2	1	3
2	2	3
2	3	4
4	0	5
4	1	5
4	2	5
4	3	5
n (n>=6)	—	n+1

Notes: DMAC is operated by Bϕ. The minimum number of idle cycles is not affected by changing a clock ratio.

1. For single transfer from the external device with $\overline{\text{DACK}}$ to the SDRAM interface, the minimum number of idle cycles is not affected by the IWW, IWRWD, IWRWS, IWRRD, and IWRRS bits in CSnBCR.

   For CMNCR.DMIWA = 0, the setting is identical to CMNCR.DMAIW[1:0] in (1) in the above table.
2. Minimum number of idle cycles for other than the above cases.

12.5.12 Bus Arbitration

The bus arbitration of this LSI has the bus mastership in the normal state and releases the bus mastership after receiving a bus request from another device.

Bus mastership is transferred at the boundary of bus cycles. Namely, bus mastership is released immediately after receiving a bus request when a bus cycle is not being performed. The release of bus mastership is delayed until the bus cycle is complete when a bus cycle is in progress. Even when from outside the LSI it looks like a bus cycle is not being performed, a bus cycle may be performing internally, started by inserting wait cycles between access cycles. Therefore, it cannot be immediately determined whether or not bus mastership has been released by looking at the $\overline{\text{CSn}}$ signal or other bus control signals. The states that do not allow bus mastership release are shown below.

1. 16-byte transfer because of a cache miss
2. During write-back operation for the cache
3. Between the read and write cycles of a TAS instruction
4. Multiple bus cycles generated when the data bus width is smaller than the access size (for example, between bus cycles when longword access is made to a memory with a data bus width of 8 bits)
5. 16-byte transfer by the DMAC
6. Setting the BLOCK bit in the CMNCR register to 1

The LSI has the bus mastership until a bus request is received from another device. Upon acknowledging the assertion (low level) of the external bus request signal $\overline{\text{BREQ}}$, the LSI releases the bus at the completion of the current bus cycle and asserts the $\overline{\text{BACK}}$ signal. After the LSI acknowledges the negation (high level) of the $\overline{\text{BREQ}}$ signal that indicates the external device has released the bus, it negates the $\overline{\text{BACK}}$ signal and resumes the bus usage.

The SDRAM interface issues all bank pre-charge commands (PALLs) when active banks exist and releases the bus after completion of a PALL command.

The bus sequence is as follows. The address bus and data bus are placed in a high-impedance state synchronized with the rising edge of CKIO. The bus mastership enable signal is asserted 0.5 cycles after the above timing, synchronized with the falling edge of CKIO. The bus control signals ($\overline{\text{BS}}$, $\overline{\text{CSn}}$, $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$, CKE, DQMxx, $\overline{\text{WEn}}$, $\overline{\text{RD}}$, and RD/$\overline{\text{WR}}$) are placed in the high-impedance state at subsequent rising edges of CKIO. Bus request signals are sampled at the falling edge of CKIO. Even when the bus is released, signals CKE, $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, and $\overline{\text{CASL}}$ can be driven with previous values according to the setting of the HIZCNT bit in CMNCR.

The sequence for reclaiming the bus mastership from an external device is described below. 1.5 cycles after the negation of $\overline{\text{BREQ}}$ is detected at the falling edge of CKIO, the bus control signals are driven high. The bus enable signal is negated at the next falling edge of the clock. The fastest timing at which actual bus cycles can be resumed after bus control signal assertion is at the rising edge of the CKIO where address and data signals are driven. Figure 12.48 shows the bus arbitration timing.

While releasing the bus mastership, the SLEEP instruction (to enter the sleep mode or the standby mode), as well as a manual reset, cannot be executed until the LSI obtains the bus mastership. The $\overline{\text{BREQ}}$ input signal is ignored in the standby mode and the $\overline{\text{BACK}}$ output signal are placed in the high impedance state. If the bus mastership request is required in this state, the bus mastership must be released by pulling down the $\overline{\text{BACK}}$ pin to enter the standby mode. The bus mastership release ($\overline{\text{BREQ}}$ signal for high level negation) after the bus mastership request ($\overline{\text{BREQ}}$ signal for low level assertion) must be performed after the bus usage permission ($\overline{\text{BACK}}$ signal for low level assertion). If the $\overline{\text{BREQ}}$ signal is negated before the $\overline{\text{BACK}}$ signal is asserted, only one cycle of the $\overline{\text{BACK}}$ signal is asserted depending on the timing of the $\overline{\text{BREQ}}$ signal to be negated and this may cause a bus contention between the external device and the LSI.

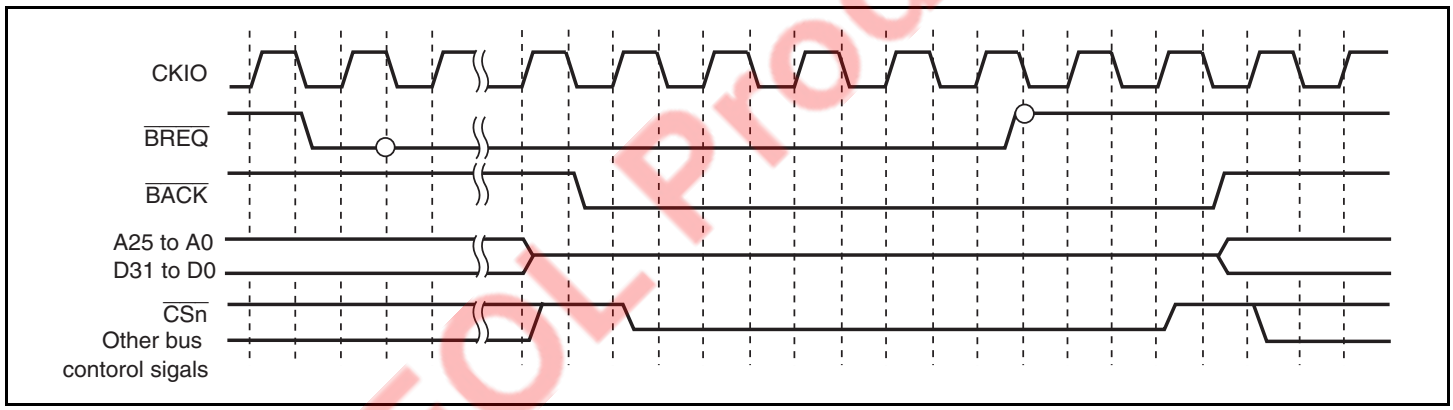


Figure 12.48 Bus Arbitration Timing (Clock Mode 7 or CMNCR.HIZCNT = 1)

12.5.13 Others

Reset: The bus state controller (BSC) can be initialized completely only at power-on reset. When a power-on reset occurs, internal clocks are synchronized by the reset, then all signals are negated and output buffers are turned off regardless of the bus cycle state. All control registers are initialized.

In standby, sleep, and manual reset, control registers of the bus state controller are not initialized. At manual reset, the current bus cycle being executed is completed and then the access wait state is entered. If a 16-byte transfer is performed by a cache or if another LSI on-chip bus master module is executed when a manual reset occurs, the current access is cancelled in longword units because the access request is cancelled by the bus master at manual reset. If a manual reset is requested during cache fill operations, the contents of the cache cannot be guaranteed. Since the RTCNT continues counting up during manual reset signal assertion, a refresh request occurs to initiate the refresh cycle. However, a bus arbitration request by the $\overline{\text{BREQ}}$ signal cannot be accepted during manual reset signal assertion.

Some flash memories may specify a minimum time from reset release to the first access. To ensure this minimum time, the bus state controller supports a 7-bit counter (RWTCNT). At power-on reset, the RWTCNT is cleared to 0. After power-on reset, RWTCNT is counted up synchronously together with CKIO and an external access will not be generated until RWTCNT is counted up to H'007F. At manual reset, RWTCNT is not cleared.

Access from the Site of the LSI Internal Bus Master: There are three types of LSI internal buses: a cache bus, internal bus, and peripheral bus. The CPU and cache memory are connected to the cache bus. Internal bus masters other than the CPU and bus state controller are connected to the internal bus. Low-speed peripheral modules are connected to the peripheral bus. Internal memories other than the cache memory are connected bidirectionally to the cache bus and internal bus. Access from the cache bus to the internal bus is enabled but access from the internal bus to the cache bus is disabled. This gives rise to the following problems.

On-chip bus masters such as DMAC other than the CPU can access internal memory other than the cache memory but cannot access the cache memory. If an on-chip bus master other than the CPU writes data to an external memory other than the cache, the contents of the external memory may differ from that of the cache memory. To prevent this problem, if the external memory whose contents is cached is written by an on-chip bus master other than the CPU, the corresponding cache memory should be purged by software.

If the CPU initiates read access for the cache, the cache is searched. If the cache stores data, the CPU latches the data and completes the read access. If the cache does not store data, the CPU performs four contiguous longword read cycles to perform cache fill operations via the internal bus. If a cache miss occurs in byte or word operand access or at a branch to an odd word boundary (4n + 2), the CPU performs four contiguous longword accesses to perform a cache fill operation on the external interface. For a cache-through area, the CPU performs access according to the actual access addresses. For an instruction fetch to an even word boundary (4n), the CPU performs longword access. For an instruction fetch to an odd word boundary (4n + 2), the CPU performs word access.

For a read cycle of a non-cache area or an on-chip peripheral module, the read cycle is first accepted and then read cycle is initiated. The read data is sent to the CPU via the cache bus.

In a write cycle for the cache area, the write cycle operation differs according to the cache write methods.

In write-back mode, the cache is first searched. If data is detected at the address corresponding to the cache, the data is then re-written to the cache. In the actual memory, data will not be re-written until data in the corresponding address is re-written. If data is not detected at the address corresponding to the cache, the cache is modified. In this case, data to be modified is first saved to the internal buffer, 16-byte data including the data corresponding to the address is then read, and data in the corresponding access of the cache is finally modified. Following these operations, a write-back cycle for the saved 16-byte data is executed.

In write-through mode, the cache is first searched. If data is detected at the address corresponding to the cache, the data is re-written to the cache simultaneously with the actual write via the internal bus. If data is not detected at the address corresponding to the cache, the cache is not modified but an actual write is performed via the internal bus.

Since the bus state controller (BSC) incorporates a one-stage write buffer, the BSC can execute an access via the internal bus before the previous external bus cycle is completed in a write cycle. If the on-chip module is read or written after the external low-speed memory is written, the on-chip module can be accessed before the completion of the external low-speed memory write cycle. In read cycles, the CPU is placed in the wait state until read operation has been completed. To continue the process after the data write to the device has been completed, perform a dummy read to the same address to check for completion of the write before the next process to be executed.

The write buffer of the BSC functions in the same way for an access by a bus master other than the CPU such as the DMAC. Accordingly, to perform dual address DMA transfers, the next read cycle is initiated before the previous write cycle is completed. Note, however, that if both the

DMA source and destination addresses exist in external memory space, the next write cycle will not be initiated until the previous write cycle is completed.

If BSC registers are modified while the write buffer is functioning, correct access cannot be performed. Thus, do not modify BSC registers immediately after the writing has finished. If BSC registers need to be modified, modify the registers after dummy reading the write data.

On-Chip Peripheral Module Access: To access an on-chip module register, two or more peripheral module clock (Pϕ) cycles are required. Care must be taken in system design.

Section 13 Direct Memory Access Controller (DMAC)

This LSI includes the direct memory access controller (DMAC).

The DMAC can be used in place of the CPU to perform high-speed transfers between external devices that have $\overline{\text{DACK}}$ (transfer request acknowledge signal), external memory, on-chip memory, memory-mapped external devices, and on-chip peripheral modules.

Figure 13.1 shows a block diagram of the DMAC.

13.1 Features

- Four channels (Two channels can receive an external request)
- 4-Gbyte physical address space
- Data transfer unit is selectable: Byte, word (two bytes), longword (four bytes), and 16 bytes (longword × 4)
- Maximum transfer count: 16,777,216 transfers (24 bits)
- Address mode: Dual address mode and single address mode are supported.
- Transfer requests
  - External request
  - On-chip peripheral module request
  - Auto request

  The following modules can issue an on-chip peripheral module request.
  - SCIF0, SCIF1, SCIF2, MTU0, MTU1, MTU2, MTU3, MTU4, USB, CMT0, CMT1, A/D converter 0, A/D converter 1
- Selectable bus modes
  - Cycle steal mode (normal mode and intermittent mode)
  - Burst mode
- Selectable channel priority levels: The channel priority levels are selectable between fixed mode and round-robin mode.
- Interrupt request: An interrupt request can be generated to the CPU at the end of the specified counts of data transfer.
- External request detection: There are following four types of $\overline{\text{DREQ}}$ input detection.
  - Low level detection
  - High level detection
  - Rising edge level detection
  - Falling edge level detection

- Transfer request acknowledge and transfer end signals: Active levels for $\overline{\text{DACK}}$ and $\overline{\text{TEND}}$ can be set independently.

Figure 13.1 shows the block diagram of the DMAC.

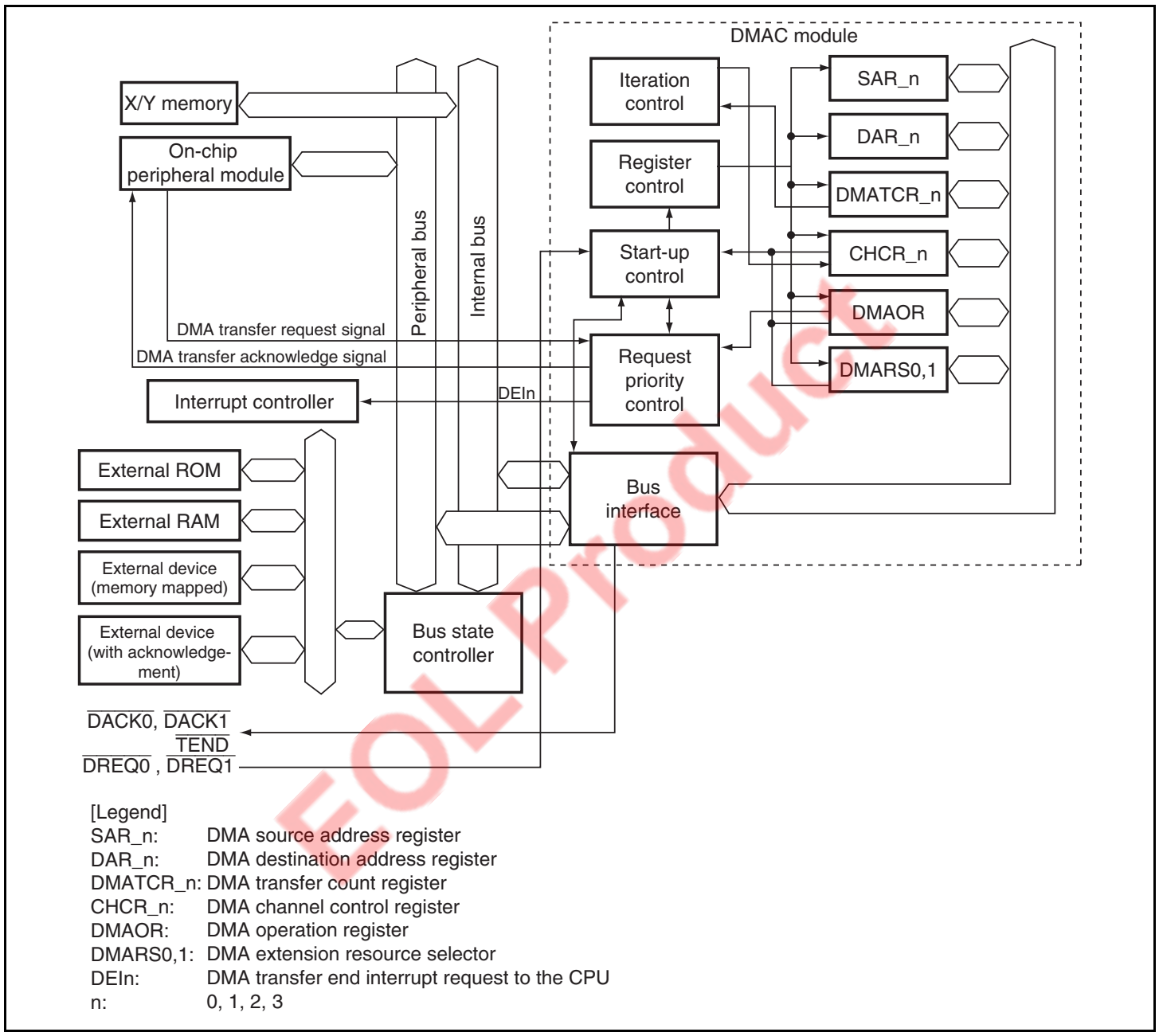


Figure 13.1 Block Diagram of the DMAC

13.2 Input/Output Pins

The external pins for DMAC are described below. Table 13.1 lists the configuration of the pins that are connected to external bus. DMAC has pins for 2 channels (channels 0 and 1) for external bus use.

Table 13.1 Pin Configuration

Channel	Name	Symbol	I/O	Function
0	DMA transfer request	$\overline{\text{DREQ0}}$	I	DMA transfer request input from external device to channel 0
	DMA transfer request acknowledge	$\overline{\text{DACK0}}$	O	Strobe output to an external I/O at DMA transfer request from external device to channel 0
	DMA transfer end	$\overline{\text{TEND}}$	O	DMA transfer end output for channel 0
1	DMA transfer request	$\overline{\text{DREQ1}}$	I	DMA transfer request input from external device to channel 1
	DMA transfer request acknowledge	$\overline{\text{DACK1}}$	O	Strobe output to an external I/O at DMA transfer request from external device to channel 1

13.3 Register Descriptions

Register configuration is described below. See section 24, List of Registers, for the addresses of these registers and the state of them in each processing status.

Channel 0:

- DMA source address register_0 (SAR_0)
- DMA destination address register_0 (DAR_0)
- DMA transfer count register_0 (DMATCR_0)
- DMA channel control register_0 (CHCR_0)

Channel 1:

- DMA source address register_1 (SAR_1)
- DMA destination address register_1 (DAR_1)
- DMA transfer count register_1 (DMATCR_1)
- DMA channel control register _1 (CHCR_1)

Channel 2:

- DMA source address register_2 (SAR_2)
- DMA destination address register_2 (DAR_2)
- DMA transfer count register_2 (DMATCR_2)
- DMA channel control register_2 (CHCR_2)

Channel 3:

- DMA source address register_3 (SAR_3)
- DMA destination address register_3 (DAR_3)
- DMA transfer count register_3 (DMATCR_3)
- DMA channel control register_3 (CHCR_3)

Common:

- DMA operation register (DMAOR)
- DMA extension resource selector 0 (DMARS0)
- DMA extension resource selector 1 (DMARS1)

13.3.1 DMA Source Address Registers (SAR)

DMA source address registers (SAR) are 32-bit read/write registers that specify the source address of a DMA transfer. During a DMA transfer, these registers indicate the next source address. When the data of an external device with $\overline{\text{DACK}}$ is transferred in the single address mode, the SAR is ignored.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit address boundary. When transferring data in 16-byte units, a 16-byte boundary (address 16n) must be set for the source address value. The SAR is undefined at reset and retains the current value in standby or module standby mode.

13.3.2 DMA Destination Address Registers (DAR)

DMA destination address registers (DAR) are 32-bit read/write registers that specify the destination address of a DMA transfer. These registers include count functions, and during a DMA transfer, these registers indicate the next destination address. When the data of an external device with $\overline{\text{DACK}}$ is transferred in the single address mode, the DAR is ignored.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit address boundary. When transferring data in 16-byte units, a 16-byte boundary (address 16n) must be set for the source address value. The DAR is undefined at reset and retains the current value in standby or module standby mode.

13.3.3 DMA Transfer Count Registers (DMATCR)

DMA transfer count registers (DMATCR) are 32-bit read/write registers that specify the DMA transfer count (bytes, words, or longwords). The number of transfers is 1 when the setting is H'000001, 16777215 when H'00FFFFFF is set, and 16777216 (the maximum) when H'000000 is set. During a DMA transfer, these registers indicate the remaining transfer count.

The upper eight bits of DMATCR will return 0 if read, and should only be written with 0. To transfer data in 16 bytes, one 16-byte transfer (128 bits) counts one. The DMATCR is undefined at reset and retains the current value in standby or module standby mode.

13.3.4 DMA Channel Control Registers (CHCR)

DMA channel control registers (CHCR) are 32-bit read/write registers that control the DMA transfer mode. The CHCR is initialized to H'00000000 at reset and retains the current value in the standby or module standby mode.

Bit	Bit Name	Initial Value	R/W	Descriptions
31	TC	0	R/W	Transfer Count Mode This bit selects whether it transmits once by one transfer request or transmits the number of setting times of DMATCR by one transfer request. This bit is effective only when transfer request original is MTU0 to MTU4, and CMT0 and CMT1 at an On-chip peripheral module request. Other than this, please specify 0 to be this bit then. 0: It transmits once by one transfer request. 1: It transmits the number of setting times of DMATCR by one transfer request.
30 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	DO	0	R/W	DMA Overrun This bit selects whether $\overline{\text{DREQ}}$ is detected by overrun 0 or by overrun 1. This bit is valid only in CHCR_0 and CHCR_1.This bit is always read as 0 in CHCR_1 and CHCR_3. The write value should always be 0. 0: Detects $\overline{\text{DREQ}}$ by overrun 0 1: Detects $\overline{\text{DREQ}}$ by overrun 1
22	TL	0	R/W	Transfer End Level This bit specifies the $\overline{\text{TEND}}$ signal output is high active or low active. This bit is valid only in CHCR_0.This bit is always read as 0 in CHCR_1 and CHCR_3. The write value should always be 0. 0: Low-active output of $\overline{\text{TEND}}$ 1: High-active output of $\overline{\text{TEND}}$

Bit	Bit Name	Initial Value	R/W	Descriptions
21 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	AM	0	R/W	Acknowledge Mode AM specifies whether $\overline{\text{DACK}}$ is output in data read cycle or in data write cycle in dual address mode. In single address mode, $\overline{\text{DACK}}$ is always output regardless of the specification by this bit. This bit is valid only in CHCR_0 and CHCR_1.This bit is always read as 0 in CHCR_2 and CHCR_3. The write value should always be 0. 0: $\overline{\text{DACK}}$ output in read cycle (Dual address mode) 1: $\overline{\text{DACK}}$ output in write cycle (Dual address mode)
16	AL	0	R/W	Acknowledge Level AL specifies the $\overline{\text{DACK}}$ (acknowledge) signal output is high active or low active. This bit is valid only in CHCR_0 and CHCR_1.This bit is always read as 0 in CHCR_2 and CHCR_3. The write value should always be 0. 0: Low-active output of $\overline{\text{DACK}}$ 1: High-active output of $\overline{\text{DACK}}$
15	DM1	0	R/W	Destination Address Mode
14	DM0	0	R/W	DM1 and DM0 select whether the DMA destination address is incremented, decremented, or left fixed. (In single address mode, DM1 and DM0 bits are ignored when data is transferred to an external device with $\overline{\text{DACK}}$.) 00: Fixed destination address (Setting prohibited in 16-byte transfer) 01: Destination address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +16 in 16-byte transfer) 10: Destination address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer; illegal setting in 16-byte transfer) 11: Reserved (Setting prohibited)

Bit	Bit Name	Initial Value	R/W	Descriptions
13	SM1	0	R/W	Source Address Mode
12	SM0	0	R/W	SM1 and SM0 select whether the DMA source address is incremented, decremented, or left fixed. (In single address mode, SM1 and SM0 bits are ignored when data is transferred from an external device with $\overline{\text{DACK}}$.) 00: Fixed source address (Setting prohibited in 16-byte transfer) 01: Source address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +16 in 16-byte transfer) 10: Source address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer; illegal setting in 16-byte transfer) 11: Reserved (Setting prohibited)
11	RS3	0	R/W	Resource Select
10	RS2	0	R/W	RS3 to RS0 specify which transfer requests will be sent to the DMAC. The changing of transfer request source should be done in the state that DMA enable bit (DE) is set to 0.
9	RS1	0	R/W	
8	RS0	0	R/W	

0	0	0	0	External request, dual address mode
0	0	0	1	Reserved (Setting prohibited)
0	0	1	0	External request/Single address mode External address space → external device with $\overline{\text{DACK}}$
0	0	1	1	External request/Single address mode External device with $\overline{\text{DACK}}$ → external address space
0	1	0	0	Auto request
0	1	0	1	Reserved (Setting prohibited)
0	1	1	0	Reserved (Setting prohibited)
0	1	1	1	Reserved (Setting prohibited)
1	0	0	0	DMA expansion request module selection specification
1	0	0	1	Reserved (Setting prohibited)
1	0	1	0	Reserved (Setting prohibited)
1	0	1	1	Reserved (Setting prohibited)
1	1	0	0	Reserved (Setting prohibited)
1	1	0	1	Reserved (Setting prohibited)
1	1	1	0	A/D converter 0
1	1	1	1	CMT0

Note: External request specification is valid only in CHCR_0 and CHCR_1. None of the request sources can be selected in channels CHCR_2 and CHCR_3.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	DL	0	R/W	$\overline{\text{DREQ}}$ Level and $\overline{\text{DREQ}}$ Edge Select
6	DS	0	R/W	These bits specify the sampling method of the $\overline{\text{DREQ}}$ pin input and the sampling level. These bits are valid only in CHCR_0 and CHCR_1. These bits are always read as 0 in CHCR_2 and CHCR_3. The write value should always be 0. In channels 0 and 1, also, if the transfer request source is specified as an on-chip peripheral module or if an auto-request is specified, the specification by this bit is ignored. 00: $\overline{\text{DREQ}}$ detected in low level 01: $\overline{\text{DREQ}}$ detected at falling edge 10: $\overline{\text{DREQ}}$ detected in high level 11: $\overline{\text{DREQ}}$ detected at rising edge
5	TB	0	R/W	Transfer Bus Mode This bit specifies the bus mode when DMA transfers data. 0: Cycle steal mode (Initial Value) 1: Burst mode Set this bit to 0 when the on-chip peripheral module is requesting DMA transfer, the setting for the transfer count mode bit is 0, and the source of the transfer request is the MTU.
4	TS1	0	R/W	Transmit Size
3	TS0	0	R/W	TS1 and TS0 specify the size of data to be transferred. Select the size of data to be transferred when the source or destination is an on-chip peripheral module register of which transfer size is specified. 00: Byte size 01: Word size (two bytes) 10: Longword size (four bytes) 11: 16-byte unit (four longword transfers)

Bit	Bit Name	Initial Value	R/W	Descriptions
2	IE	0	R/W	Interrupt Enable This bit specifies whether or not an interrupt request is generated to the CPU at the end of the DMA transfer. Setting this bit to 1 generates an interrupt request (DEI) to the CPU when TE bit is set to 1. 0: Interrupt request is not generated 1: Interrupt request is generated
1	TE	0	R/W*	Transfer End Flag This bit shows that DMA transfer ends. TE is set to 1 when data transfer ends when DMATCR becomes to 0. The TE bit is not set to 1 in the following cases. • DMA transfer ends due to a NMI interrupt or DMA address error before DMATCR becomes to 0. • DMA transfer is ended by clearing the DE bit and DME bit in DMA operation register (DMAOR). Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled. 0: During the DMA transfer or DMA transfer has been interrupted 1: Data transfer ends by the specified count (DMACTR = 0) [Clearing condition] Writing 0 after TE = 1 read

Bit	Bit Name	Initial Value	R/W	Descriptions
0	DE	0	R/W	DMA Enable This bit enabler or disables the DMA transfer. In an auto request mode, DMA transfer starts by setting the DE bit and DME bit in DMAOR to 1. In this time, all of the bits TE, NMIF in DMAOR, and AE must be 0's. In an external request or peripheral module request, DMA transfer starts if DMA transfer request is generated by the devices or peripheral modules after setting the bits DE and DME to 1. In this case, however, all of the bits TE, NMIF, and AE must be 0's an in the case of auto request mode. Clearing the DE bit to 0 can terminate the DMA transfer. 0: DMA transfer disabled 1: DMA transfer enabled

Note: * Writing 0 is possible to clear the flag.

13.3.5 DMA Operation Register (DMAOR)

The DMA operation register (DMAOR) is a 32-bit read/write register that specifies the priority level of channels at the DMA transfer. This register shows the DMA transfer status. The DMAOR is initialized to H'00000000 at reset and retains the current value in the standby or module standby mode.

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
29	CMS1	0	R/W	Cycle Steal Mode Select 1, 0
28	CMS0	0	R/W	These bits select either normal mode or intermittent mode in cycle steal mode. It is necessary that the bus modes of all channels be set to cycle steal mode to make the intermittent mode valid. 00: Normal mode 01: Reserved (Setting prohibited) 10: Intermittent mode 16 Executes one DMA transfer in each of 16 clocks of an external bus clock. 11: Intermittent mode 64 Executes one DMA transfer in each of 64 clocks of an external bus clock.
27, 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
25	PR1	0	R/W	Priority Mode 1, 0
24	PR0	0	R/W	PR1 and PR0 select the priority level between channels when there are transfer requests for multiple channels simultaneously. 00: Fixed mode 1: CH0 > CH1 > CH2 > CH3 01: Fixed mode 2: CH0 > CH2 > CH3 > CH1 10: The status of the channel select round-robin mode: RCn bit is reflected to the priority. 11: All channel round-robin mode
23 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18	AE	0	R/(W)*	Address Error Flag AE indicates that an address error occurred during DMA transfer. If this bit is set during data transfer, transfers on all channels are suspended. The CPU cannot write 1 to this bit. This bit can only be cleared by writing 0 after reading 1. 0: No DMAC address error 1: DMAC address error [Clear condition] Writing AE = 0 after AE = 1 read
17	NMIF	0	R/(W)*	NMI Flag NMIF indicates that a NMI interrupt occurred. This bit is set regardless of whether DMAC is in operating or halt state. The CPU cannot write 1 to this bit. Only 0 can be written to clear this bit after 1 is read. When the NMI is input, the DMA transfer in progress can be done in one transfer unit. When the DMAC is not in operational, the NMIF bit is set to 1 even if the NMI interrupt was input. 0: No NMI input 1: NMI interrupt occurs [Clearing condition] Writing NMIF = 0 after NMIF = 1 read

Bit	Bit Name	Initial Value	R/W	Description
16	DME	0	R/W	DMA Master Enable DME enables or disables DMA transfers on all channels. If the DME bit and the DE bit corresponding to each channel in CHCR are set to 1s, transfer is enabled in the corresponding channel. If this bit is cleared during transfer, transfers in all the channels can be terminated. Even if the DME bit is set, transfer is not enabled if the TE bit is 1 or the DE bit is 0 in CHCR, or the NMIF bit is 1 in DMAOR. 0: Disable DMA transfers on all channels 1: Enable DMA transfers on all channels
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	RC0	0	R/W	Round Robin Cannel Select RC3, RC2, RC1, and RC0 select the priority level between channels when there are transfer requests for multiple channels simultaneously. 0: The priority level of the CHn (n: 0 to 3) is fixed. When all RC bits is 0, the priority level is: CH0 > CH1 > CH2 > CH3, equals with fixed mode (mdoe7). 1: The priority level of the CHn (n: 0 to 3) is determined by the round-robin. When all RC bits are 1, the priority level between channels equals with round-robin mode (mode 5).
4	RC1	0	R/W	
3	RC2	0	R/W	
2	RC3	0	R/W	
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * Writing 0 is possible to clear the flag.

If DMA transfers are requested to multiple channels simultaneously, the DMAC performs transfers according to the specified channel priority. The channel priority is determined by the round-robin select bits (RC0, RC1, RC2, RC3) and priority mode bits (PR1 and PR0) of the DMAOR register.

If (PR1 and PR0) = (B'10) is specified, the channel priority is determined according to the settings of the round-robin select bits. In this case, the channel priority is changed between channels whose corresponding round-robin select bit is set to 1. If (PR1 and PR0) = (B'01) is specified, the channel priority is specified as fixed mode 2 (CH0 > CH2 > CH3 > CH1). If (PR1 and PR0) = (B'11) is specified, the channel priority is specified as the all-channel round-robin mode. If (PR1 and PR0) = (B'00) is specified, the channel priority is specified as fixed mode 1 (CH0 > CH1 > CH2 > CH3). Note that the round-robin select bit values are ignored except when (PR1 and PR0) = (B'10) is specified.

If the round-robin select bit or the priority mode bit is modified after a DMA transfer, the channel priority is initialized to be changed. If fixed mode 2 is specified, the channel priority is specified as CH0 > CH2 > CH3 > CH1. If fixed mode 1 is specified, the channel priority is specified as CH0 > CH1 > CH2 > CH3. If a mode including round-robin mode is specified again, the transfer end channel is reset.

Table 13.2 summarizes the relationship among the round-robin select bits, priority bits, channel priority, and priority modes (mode 0 to mode 7). Each priority mode includes up to five kinds of channel priority according to the transfer end channel.

For example, if the round-robin select bits are specified as (RC0 to RC3) = (B'1110) to select mode 3 and if the transfer end channel is channel 1, the priority of the channel to accept the next transfer request is specified as CH0 > CH1 > CH2 >CH3. When the channel on which the transfer was just finished is CH3, CH3 is not intended for round-robin. Therefore the priority level is not changed.

Table 13.2 Combination of the Round-Robin Select Bits and Priority Mode Bits

	Round-robin Select bit				Transfer End	Priority bit		Priority Level High ◀——▶ Low			
Mode No.	**RC0**	**RC1**	**RC2**	**RC3**	**CH No.**	**PR1**	**PR0**	**0**	**1**	**2**	**3**
0	0	0	1	1	CH2	1	0	CH0	CH1	CH3	CH2
1	0	1	1	1	CH1	1	0	CH0	CH2	CH3	CH1
	0	1	1	1	CH2	1	0	CH0	CH3	CH1	CH2
2	1	1	0	0	CH0	1	0	CH1	CH0	CH2	CH3
3	1	1	1	0	CH0	1	0	CH1	CH2	CH0	CH3
	1	1	1	0	CH1	1	0	CH2	CH0	CH1	CH3
4	1	1	1	1	CH0	1	0	CH1	CH2	CH3	CH0
	1	1	1	1	CH1	1	0	CH2	CH3	CH0	CH1
	1	1	1	1	CH2	1	0	CH3	CH0	CH1	CH2
—	Other than the above setting prohibited				—	1	0	—	—	—	—
5	*	*	*	*	CH0	1	1	CH1	CH2	CH3	CH0
(All-channel	*	*	*	*	CH1	1	1	CH2	CH3	CH0	CH1
round-robin)	*	*	*	*	CH2	1	1	CH3	CH0	CH1	CH2
6 (Fixed mode 2)	*	*	*	*	*	0	1	CH0	CH2	CH3	CH1
7 (Fixed mode 1)	*	*	*	*	*	0	0	CH0	CH1	CH2	CH3

Note: * Any

13.3.6 DMA Extension Resource Selector 0 and 1 (DMARS0, DMARS1)

DMARS is a 16-bit read/write register that specifies the DMA transfer sources from peripheral modules in each channel. DMARS0 specifies for channels 0 and 1, DMARS1 specifies for channels 2 and 3. This register can set the transfer request of SCIF0, SCIF1, SCIF2, MTU0, MTU1, MTU2, MTU3, MTU4, MTU, USB, A/D converter 1, and CMT1.

This register is initialized to H'0000 by power-on manual reset. The previous value is held in standby mode or module standby mode.

- DMARS0

Bit	Bit Name	Initial Value	R/W	Description
15	C1MID5	0	R/W	Transfer request module ID5 for DMA channel 1 (MID). See table 13.3.
14	C1MID4	0	R/W	
13	C1MID3	0	R/W	
12	C1MID2	0	R/W	
11	C1MID1	0	R/W	
10	C1MID0	0	R/W	
9	C1RID1	0	R/W	Transfer request register ID for DMA channel 1 (RID). See table 13.3.
8	C1RID0	0	R/W	
7	C0MID5	0	R/W	Transfer request module ID for DMA channel 0 (MID). See table 13.3
6	C0MID4	0	R/W	
5	C0MID3	0	R/W	
4	C0MID2	0	R/W	
3	C0MID1	0	R/W	
2	C0MID0	0	R/W	
1	C0RID1	0	R/W	Transfer request register ID for DMA channel 0 (RID). See table 13.3.
0	C0RID0	0	R/W	

- DMARS1

Bit	Bit Name	Initial Value	R/W	Description
15	C3MID5	0	R/W	Transfer request module ID for DMA channel 3 (MID). See table 13.3.
14	C3MID4	0	R/W	
13	C3MID3	0	R/W	
12	C3MID2	0	R/W	
11	C3MID1	0	R/W	
10	C3MID0	0	R/W	
9	C3RID1	0	R/W	Transfer request module ID for DMA channel 3 (RID). See table 13.3.
8	C3RID0	0	R/W	
7	C2MID5	0	R/W	Transfer request module ID for DMA channel 2 (MID). See table 13.3.
6	C2MID4	0	R/W	
5	C2MID3	0	R/W	
4	C2MID2	0	R/W	
3	C2MID1	0	R/W	
2	C2MID0	0	R/W	
1	C2RID1	0	R/W	Transfer request module ID for DMA channel 2 (RID). See table 13.3.
0	C2RID0	0	R/W	

Transfer requests from the various modules are specified by the MID and RID as shown in table 13.3.

Table 13.3 Transfer Request Module/Register ID

Peripheral Module	Setting Value for One Channel (MID + RID)	MID	RID	Function
SCIF0	H'88	B'100010	B'00	Transmit
	H'89		B'01	Receive
SCIF1	H'90	B'100100	B'00	Transmit
	H'91		B'01	Receive
SCIF2	H'40	B'010000	B'00	Transmit
	H'41		B'01	Receive
MTU0	H'A8	B'101010	B'00	TGI0A
MTU1	H'C0	B'110000	B'00	TGI1A
MTU2	H'C8	B'110010	B'00	TGI2A
MTU3	H'D0	B'110100	B'00	TGI3A
MTU4	H'E8	B'111010	B'00	TGI4A
USB	H'A0	B'101000	B'00	Transmit
	H'A1		B'01	Receive
A/D converter 1	H'B0	B'101100	B'00	—
CMT1	H'F0	B'111100	B'00	—

When MID/RID other than the values listed in table 13.3 is set, the operation of this LSI is not guaranteed. The transfer request from the DMARS register is valid only when the resource select bits (RS3 to RS0) have been set to B'1000 for CHCR0 to CHCR3 registers. Otherwise, even if the DMARS has been set, transfer request source is not accepted.

13.4 Operation

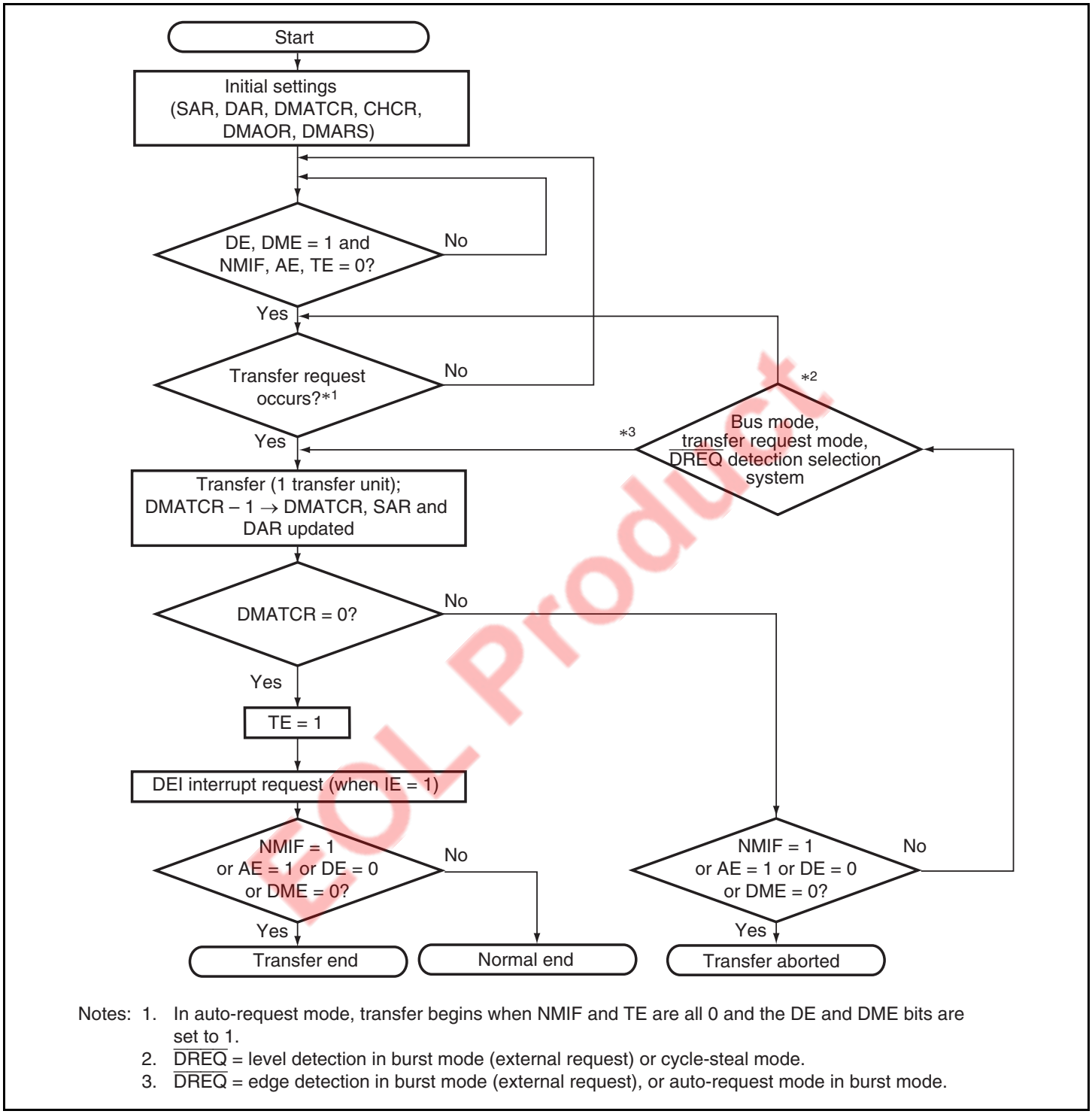
When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip module request. The dual address mode has direct address transfer mode and indirect address transfer mode. In the bus mode, the burst mode or the cycle steal mode can be selected.

13.4.1 DMA Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA operation register (DMAOR), and DMA extension resource selector (DMARS) are set, the DMAC transfers data according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0)
2. When a transfer request comes and transfer is enabled, the DMAC transfers 1 transfer unit of data (depending on the TS0 and TS1 settings). For an auto request, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decrement for each transfer. The actual transfer flows vary by address mode and bus mode.
3. When the specified number of transfer have been completed (when DMATCR reaches 0), the transfer ends normally. If the IE bit of the CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU.
4. When a NMI interrupt is generated, the transfer is aborted. Transfers are also aborted when the DE bit of the CHCR or the DME bit of the DMAOR are changed to 0.

Figure 13.2 is a flowchart of this procedure.



Notes: 1. In auto-request mode, transfer begins when NMIF and TE are all 0 and the DE and DME bits are set to 1.
2. $\overline{\text{DREQ}}$ = level detection in burst mode (external request) or cycle-steal mode.
3. $\overline{\text{DREQ}}$ = edge detection in burst mode (external request), or auto-request mode in burst mode.

Figure 13.2 DMA Transfer Flowchart

13.4.2 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated by devices and on-chip peripheral modules that are neither the source nor the destination. Transfers can be requested in three modes: auto request, external request, and on-chip module request. The request mode is selected in the RS3 to RS0 bits of the DMA channel control registers 0 to 3 (CHCR_0 to CHCR_3), and the DMA extension resource selectors 0 and 1 (DMARS0, DMARS1).

Auto-Request Mode: When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, the auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits of CHCR_0 to CHCR_3 and the DME bit of the DMAOR are set to 1, the transfer begins so long as the TE bits of CHCR_0 to CHCR_3 and the NMIF bit of DMAOR are all 0.

External Request Mode: In this mode a transfer is performed at the request signals ($\overline{\text{DREQ0}}$ to $\overline{\text{DREQ1}}$) of an external device. This is valid for DMA channels 0 to 1. Choose one of the modes shown in table 13.4 according to the application system. When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), a transfer is performed upon a request at the $\overline{\text{DREQ}}$ input.

Table 13.4 Selecting External Request Modes with the RS Bits

RS3	RS2	RS1	RS0	Address Mode	Source	Destination
0	0	0	0	Dual address mode	Any	Any
0	0	1	0	Single address mode	External memory, memory-mapped external device	External device with $\overline{\text{DACK}}$
			1		External device with $\overline{\text{DACK}}$	External memory, memory-mapped external device

Choose to detect $\overline{\text{DREQ}}$ by either the falling edge or low level of the signal input with the $\overline{\text{DREQ}}$ level (DL) bit and DS bit of CHCR_0 and CHCR_1 as shown in table 13.5. The source of the transfer request does not have to be the data transfer source or destination.

Table 13.5 Selecting External Request Detection with Dl, DS Bits

CHCR		
DL	**DS**	**Detection of External Request**
0	0	Low level detection
	1	Falling edge detection
1	0	High level detection
	1	Rising edge detection

When $\overline{\text{DREQ}}$ is accepted, the $\overline{\text{DREQ}}$ pin becomes request accept disabled state (non-sensitive period). After issuing acknowledge signal $\overline{\text{DACK}}$ for the accepted $\overline{\text{DREQ}}$, the $\overline{\text{DREQ}}$ pin again becomes request accept enabled state.

When $\overline{\text{DREQ}}$ is used by level detection, there are following two cases by the timing to detect the next $\overline{\text{DREQ}}$ after outputting $\overline{\text{DACK}}$.

Overrun0: Transfer is aborted after the same number of transfer has been performed as requests.

Overrun1: Transfer is aborted after transfers have been performed for (the number of requests plus 1) times.

The DO bit in CHCR selects this overrun 0 or overrun 1.

Table 13.6 Selecting External Request Detection with DO Bit

CHCR_0 or CHCR_1	
DO	**External Request**
0	Overrun 0
1	Overrun 1

On-Chip Peripheral Module Request: In this mode, the transfer is performed in response to the DMA transfer request signal of an on-chip peripheral module. Signals that request DMA transfer include A/D conversion-completed transfer requests from A/D converter 0, compare-match transfer requests from the CMT0 timer, transmit-data empty transfer requests and receive-data full transfer requests from the SCIF0 to SCIF2 that are set by DMARS0 and 1, compare-match and input-capture interrupts from the MTU0 to MTU4 timers, transmit-data-empty transfer requests and receive-data-full transfer requests from the USB module, A/D conversion-completed transfer requests from A/D converter 1, and compare-match transfer requests from the CMT1 timer.

When the transfer request is a transmit-data-empty transfer request, set the transfer destination as the corresponding SCIF transmit-data register. Likewise, when the transfer request is a receive-data full transfer request, set the transfer destination as the corresponding SCIF receive-data register. Requests from the USB are handled in an analogous way. If a transfer is requested from the A/D converter 0 and A/D converter 1, the transfer source must be the A/D data register (ADDR). Any address can be specified for data source and destination, when transfer request is generated by CMT0, CMT1, and MTU0 to MTU4.

Table 13.7 Selecting On-Chip Peripheral Module Request Modes with the RS3 to RS0 Bits

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
RS[3:0]	MID	RID					
1110	Any	Any	A/D converter 0	ADI (A/D conversion end interrupt)	ADDR	Any	Cycle steal
1111	Any	Any	CMT0	Compare-match transfer request	Any	Any	Burst/ cycle steal
1000	100010	00	SCIF0 transmitter	TXI (transmit data FIFO empty interrupt)	Any	SCFTDR0	Cycle steal
		01	SCIF0 receiver	RXI (receive data FIFO full interrupt)	SCFRDR0	Any	Cycle steal
	100100	00	SCIF1 transmitter	TXI (transmit data FIFO empty interrupt)	Any	SCFTDR1	Cycle steal
		01	SCIF1 receiver	RXI (receive data FIFO full interrupt)	SCFRDR1	Any	Cycle steal
	010000	00	SCIF2 transmitter	TXI (transmit data FIFO empty interrupt)	Any	SCFTDR2	Cycle steal
		01	SCIF2 receiver	RXI (receive data FIFO full interrupt)	SCFRDR2	Any	Cycle steal

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
RS[3:0]	MID	RID					
1000	101010	00	MTU0	TGI0A (input capture interrupt/ compare match interrupt)	Any	Any	Burst/ cycle steal
	110000	00	MTU1	TGI1A (input capture interrupt/ compare match interrupt)	Any	Any	Burst/ cycle steal
	110010	00	MTU2	TGI2A (input capture interrupt/ compare match interrupt)	Any	Any	Burst/ cycle steal
	110100	00	MTU3	TGI3A (input capture interrupt/ compare match interrupt)	Any	Any	Burst/ cycle steal
	111010	00	MTU4	TGI4A (input capture interrupt/ compare match interrupt)	Any	Any	Burst/ cycle steal
	101000	00	USB transmitter	EP2FIFO empty transfer request	Any	USBEPDR2	Cycle steal
		01	USB receiver	EP1FIFO full transfer request	USBEPDR1	Any	Cycle steal
	101100	00	A/D converter 1	ADI (A/D conversion end interrupt)	ADDR1	Any	Cycle steal
	111100	00	CMT1	Compare-match transfer request	Any	Any	Burst/ cycle steal

13.4.3 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority order. The four modes (fixed mode 1, fixed mode 2, channel selective round-robin mode, and all-channel round-robin mode) are selected using the priority bits PR0, PR1, and RC0 to RC3 in the DMA operation register (DMAOR).

Fixed Mode: In these modes, the priority levels among the channels remain fixed. There are two kinds of fixed modes as follows:

Fixed mode 1: CH0 > CH1 > CH2 > CH3

Fixed mode 2: CH0 > CH2 > CH3 > CH1

These are selected by the PR1 and the PR0 bits in the DMA operation register (DMAOR).

Round-Robin Mode: Each time one word, byte, or longword is transferred on one channel, the priority order is rotated. The channel on which the transfer was just finished rotates to the bottom of the priority order. The round-robin mode operation is shown in figure 13.3. The priority of the round-robin mode is CH0 > CH1 > CH2 > CH3 immediately after reset.

When the round-robin mode has been specified, do not concurrently specify cycle steal mode and burst mode as the bus modes of any two channels.

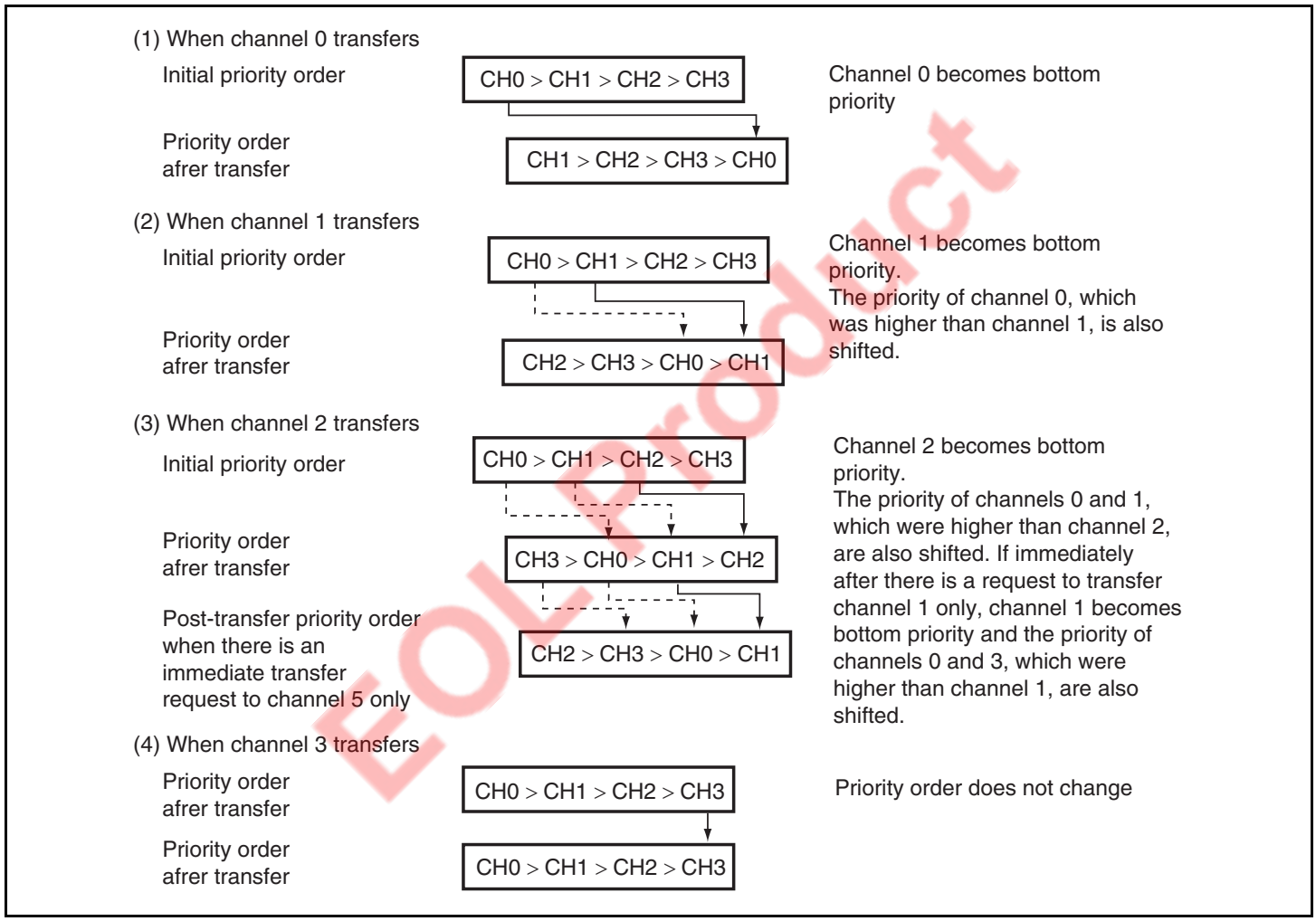


Figure 13.3 Round-Robin Mode

Figure 13.4 shows how the priority order changes when channel 0 and channel 3 transfers are requested simultaneously and a channel 1 transfer is requested during the channel 0 transfer. The DMAC operates as follows:

1. Transfer requests are generated simultaneously to channels 0 and 3.
2. Channel 0 has a higher priority, so the channel 0 transfer begins first (channel 3 waits for transfer).
3. A channel 1 transfer request occurs during the channel 0 transfer (channels 1 and 3 are both waiting)
4. When the channel 0 transfer ends, channel 0 becomes lowest priority.
5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfer begins (channel 3 waits for transfer).
6. When the channel 1 transfer ends, channel 1 becomes lowest priority.
7. The channel 3 transfer begins.
8. When the channel 3 transfer ends, channels 3 and 2 shift downward in priority so that channel 3 becomes the lowest priority.

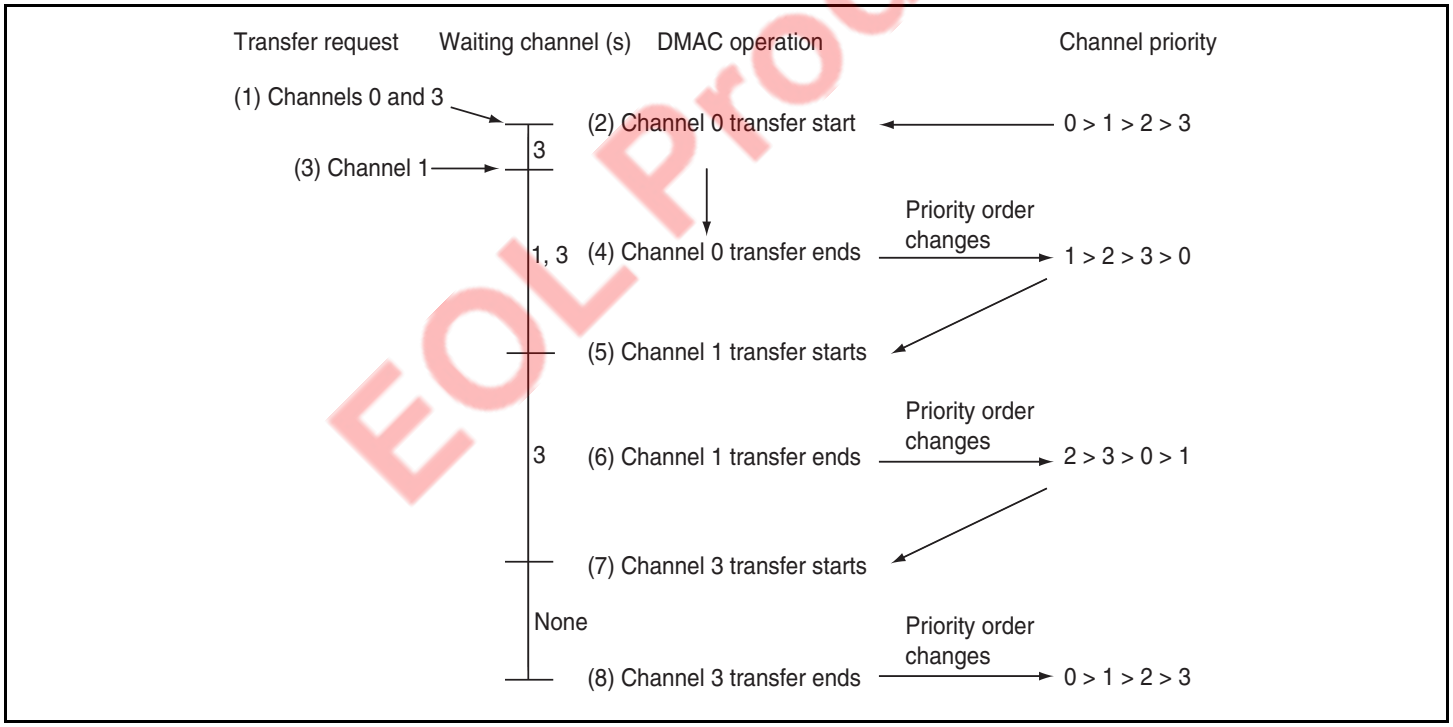


Figure 13.4 Changes in Channel Priority in Round-Robin Mode

13.4.4 DMA Transfer Types

DMA transfer has two types; single address mode transfer and dual address mode transfer. They depend on the number of bus cycles of access to source and destination. A data transfer timing depends on the bus mode, which has cycle steal mode and burst mode. The DMAC supports the transfers shown in table 13.8.

Table 13.8 Supported DMA Transfers

	Destination				
Source	**External Device with $\overline{\text{DACK}}$**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Peripheral Module**	**X/Y Memory U Memory**
External device with $\overline{\text{DACK}}$	Not available	Dual, single	Dual, single	Not available	Not available
External memory	Dual, single	Dual	Dual	Dual	Dual
Memory-mapped external device	Dual, single	Dual	Dual	Dual	Dual
On-chip peripheral module	Not available	Dual	Dual	Dual	Dual
X/Y memory, U memory	Not available	Dual	Dual	Dual	Dual

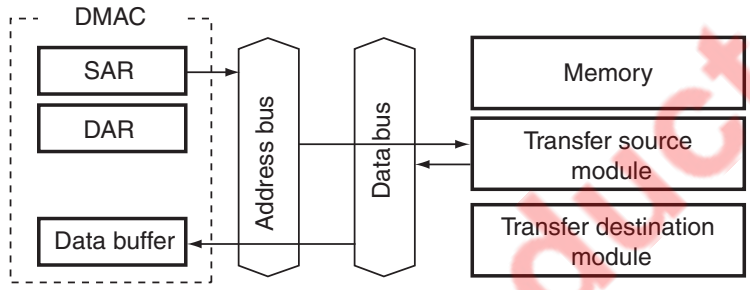
Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. 16-byte transfer is not available for on-chip peripheral modules.

Address Modes:

1. Dual Address Mode

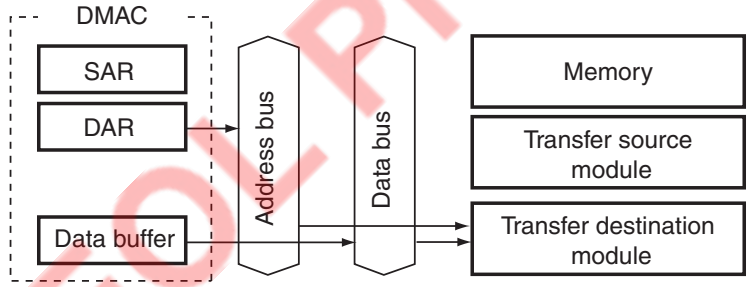
   In the dual address mode, both the transfer source and destination are accessed (selected) by an address. The source and destination can be located externally or internally.

   DMA transfer requires two bus cycles because data is read from the transfer source in a data read cycle and written to the transfer destination in a data write cycle. At this time, transfer data is temporarily stored in the DMAC. In the transfer between external memories as shown in figure 13.5, data is read to the DMAC from one external memory in a data read cycle, and then that data is written to the other external memory in a write cycle.



The SAR value is an address, data is read from the transfer source module, and the data is tempolarily stored in the DMAC.

First bus cycle



The DAR value is an address and the value stored in the data buffer in the DMAC is written to the transfer destination module.

Second bus cycle

Figure 13.5 Data Flow of Dual Address Mode

Auto request, external request, and on-chip peripheral module request are available for the transfer request. DACK can be output in read cycle or write cycle in dual address mode. The AM bit of the channel control register (CHCR) can specify whether the $\overline{\text{DACK}}$ is output in read cycle or write cycle.

Figure 13.6 shows an example of DMA transfer timing in dual address mode.

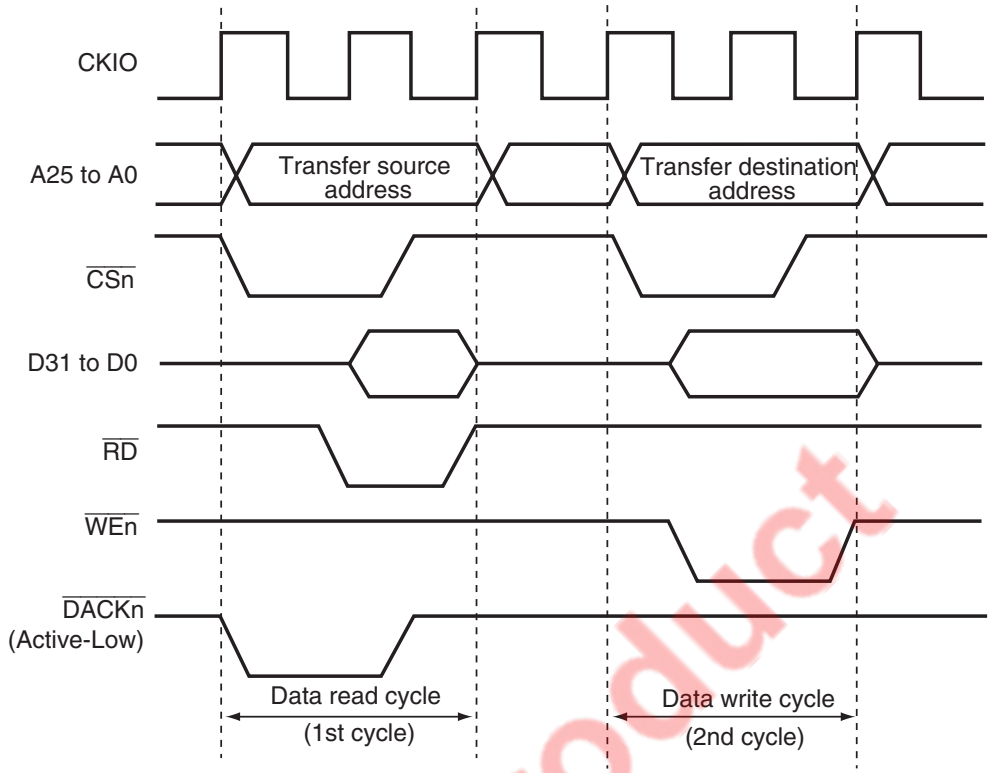


Figure 13.6 Example of DMA Transfer Timing in Dual Mode (Source: Ordinary Memory, Destination: Ordinary Memory)

2. Single Address Mode

   In single address mode, either the transfer source or transfer destination peripheral device is accessed (selected) by means of the DACK signal, and the other device is accessed by address. In this mode, the DMAC performs one DMA transfer in one bus cycle, accessing one of the external devices by outputting the DACK transfer request acknowledge signal to it, and at the same time outputting an address to the other device involved in the transfer. For example, in the case of transfer between external memory and an external device with DACK shown in figure 13.7, when the external device outputs data to the data bus, that data is written to the external memory in the same bus cycle.

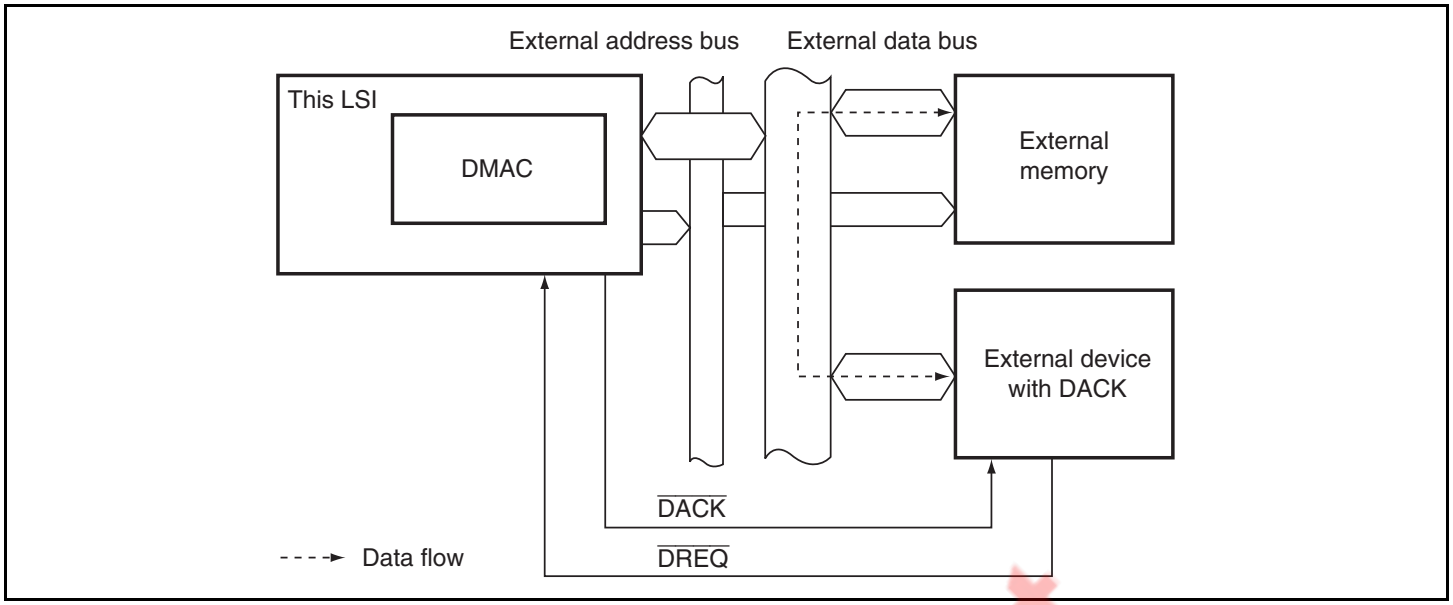


Figure 13.7 Data Flow in Single Address Mode

Two kinds of transfer are possible in single address mode: (1) transfer between an external device with $\overline{\text{DACK}}$ and a memory-mapped external device, and (2) transfer between an external device with $\overline{\text{DACK}}$ and external memory. In both cases, only the external request signal ($\overline{\text{DREQ}}$) is used for transfer requests.

Figure 13.8 shows example of DMA transfer timing in single address mode.

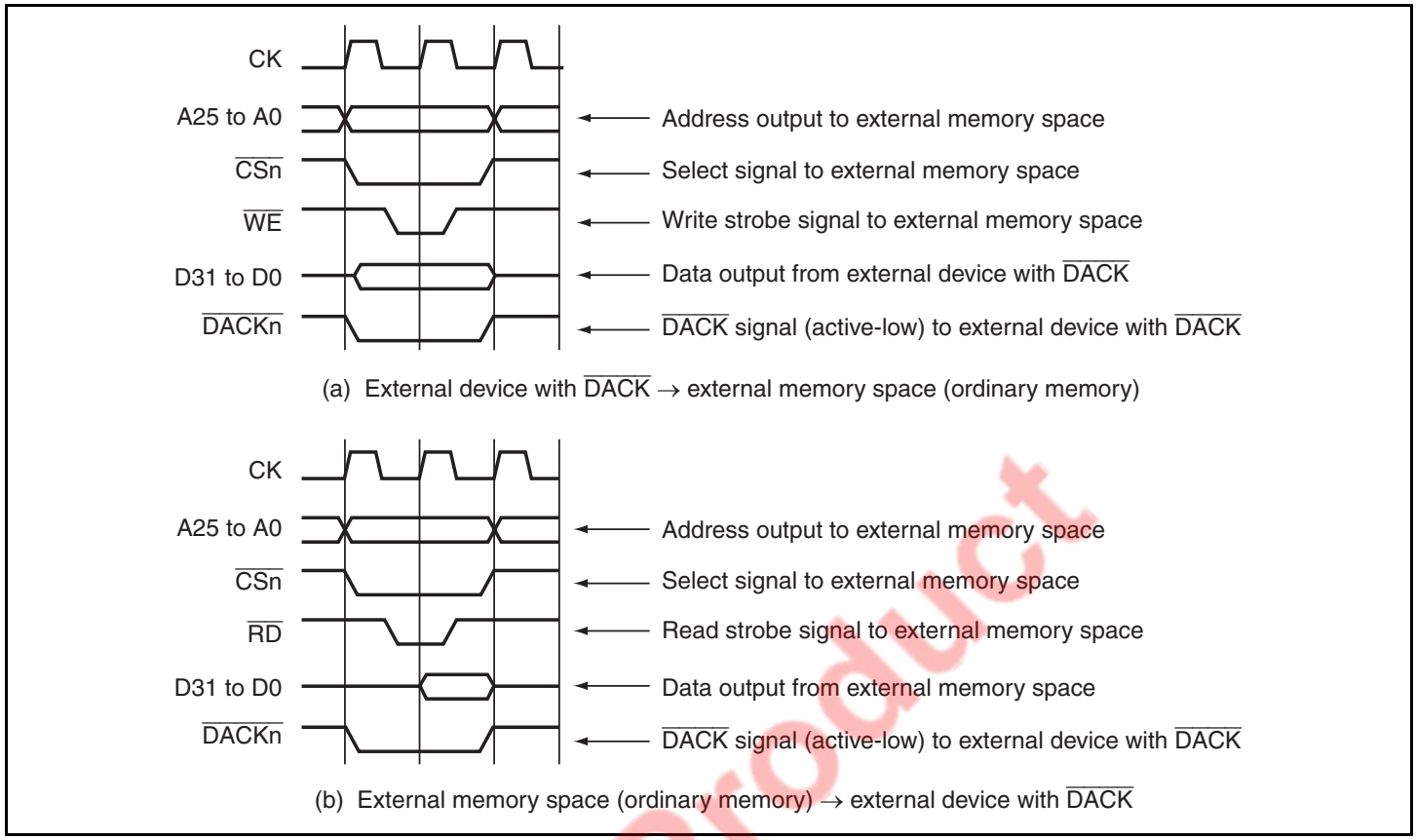


Figure 13.8 Example of DMA Transfer Timing in Single Address Mode

Bus Modes: There are two bus modes: cycle steal and burst. Select the mode in the TB bits of the channel control register (CHCR).

1. Cycle-Steal Mode

- Normal mode

  In the normal mode of cycle-steal, the bus mastership is given to another bus master after a one-transfer-unit (byte, word, longword, or 16 bytes unit) DMA transfer. When another transfer request occurs, the bus masterships are obtained from the other bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus mastership is passed to the other bus master. This is repeated until the transfer end conditions are satisfied.

  In the cycle-steal mode, transfer areas are not affected regardless of settings of the transfer request source, transfer source, and transfer destination.

Figure 13.9 shows an example of DMA transfer timing in the cycle steal mode. Transfer conditions shown in the figure are:

1. Dual address mode
2. $\overline{\text{DREQ}}$ low level detection

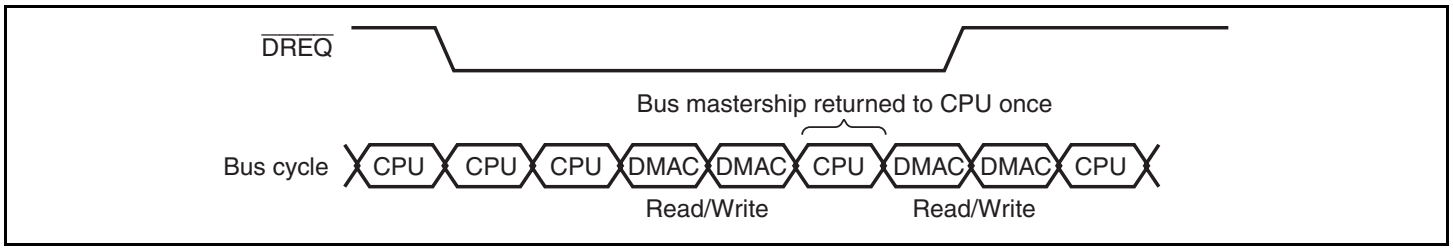


Figure 13.9 DMA Transfer Example in the Cycle-Steal Normal Mode (Dual Address, $\overline{\text{DREQ}}$ Low Level Detection)

- Intermittent Mode 16 and Intermittent Mode 64

  In the intermittent mode of cycle steal, DMAC returns the bus mastership to other bus master whenever a unit of transfer (byte, word, longword, or 16 bytes) is complete. If the next transfer request occurs after that, DMAC gets the bus mastership from other bus master after waiting for 16 or 64 clocks in Bϕ count. DMAC then transfers data of one unit and returns the bus mastership to other bus master. These operations are repeated until the transfer end condition is satisfied. It is thus possible to make lower the ratio of bus occupation by DMA transfer than the normal mode of cycle steal.

  When DMAC gets again the bus mastership, DMA transfer can be postponed in case of entry updating due to cache miss.

  This intermittent mode can be used for all transfer section; transfer requester, source, and destination. The bus modes, however, must be cycle steal mode in all channels.

Figure 13.10 shows an example of DMA transfer timing in cycle steal intermittent mode. Transfer conditions shown in the figure are:

1. Dual address mode
2. $\overline{\text{DREQ}}$ low level detection

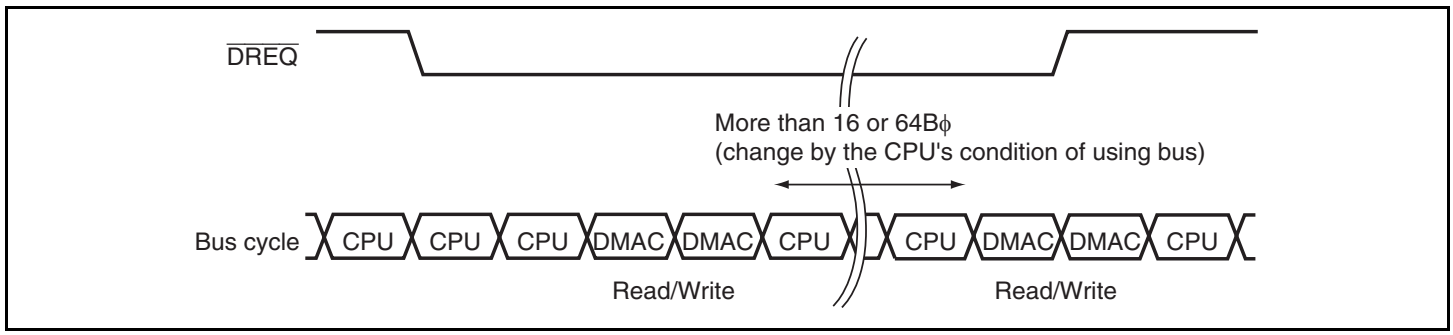


Figure 13.10 Example of DMA Transfer in Cycle Steal Intermittent Mode (Dual Address, $\overline{\text{DREQ}}$ Low Level Detection)

2. Burst Mode

   Once the bus mastership is obtained, the transfer is performed continuously until the transfer end condition is satisfied. In the external request mode with low level detection of the $\overline{\text{DREQ}}$ pin, however, when the $\overline{\text{DREQ}}$ pin is driven high, the bus passes to the other bus master after the DMAC transfer request that has already been accepted ends, even if the transfer end conditions have not been satisfied.

   The burst mode cannot be used for other than CMT0, CMT1, and MTU0 to MTU4 when the on-chip peripheral module is the transfer request source. Figure 13.11 shows DMA transfer timing in the burst mode.

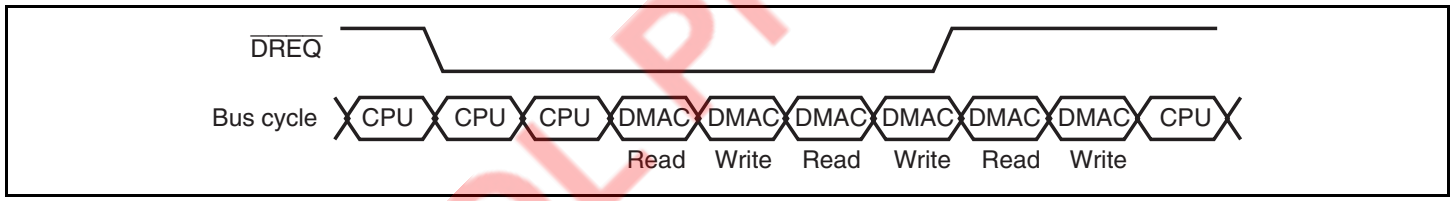


Figure 13.11 DMA Transfer Example in the Burst Mode (Dual Address, $\overline{\text{DREQ}}$ Low Level Detection)

Relationship between Request Modes and Bus Modes by DMA Transfer Category: Table 13.9 shows the relationship between request modes and bus modes by DMA transfer category.

Table 13.9 Relationship of Request Modes and Bus Modes by DMA Transfer Category

Address Mode	Transfer Category	Request Mode	Bus Mode	Transfer Size (Bits)	Usable Channels
Dual	External device with $\overline{\text{DACK}}$ and external memory	External	B/C	8/16/32/128	0, 1
	External device with $\overline{\text{DACK}}$ and memory-mapped external device	External	B/C	8/16/32/128	0, 1
	External memory and external memory	All*[1]	B/C	8/16/32/128	0 to 3*[5]
	External memory and memory-mapped external device	All*[1]	B/C	8/16/32/128	0 to 3*[5]
	Memory-mapped external device and memory-mapped external device	All*[1]	B/C	8/16/32/128	0 to 3*[5]
	External memory and on-chip peripheral module	All*[2]	B/C*[3]	8/16/32/128*[4]	0 to 3*[5]
	Memory-mapped external device and on-chip peripheral module	All*[2]	B/C*[3]	8/16/32/128*[4]	0 to 3*[5]
	On-chip peripheral module and on-chip peripheral module	All*[2]	B/C*[3]	8/16/32/128*[4]	0 to 3*[5]
	X/Y memory, U memory and X/Y memory, U memory	All*[1]	B/C	8/16/32/128	0 to 3*[5]
	X/Y memory, U memory and memory-mapped external device	All*[1]	B/C	8/16/32/128	0 to 3*[5]
	X/Y memory, U memory and on-chip peripheral module	All*[2]	B/C*[3]	8/16/32/128*[4]	0 to 3*[5]
	X/Y memory, U memory and external memory	All*[1]	B/C	8/16/32/128	0 to 3*[5]
Single	External device with $\overline{\text{DACK}}$ and external memory	External	B/C	8/16/32/128	0, 1
	External device with $\overline{\text{DACK}}$ and memory-mapped external device	External	B/C	8/16/32/128	0, 1

[Legend]

B: Burst

C: Cycle steal

Notes: 1. External requests, auto requests, and on-chip peripheral module requests are all available. In the case of on-chip peripheral module requests, however, CMT0, CMT1, and MTU0 to MTU4 are only available.

2. External requests, auto requests, and on-chip peripheral module requests are all available. However, for on-chip peripheral module requests, the module must be designated as the transfer request source or the transfer destination except CMT0, CMT1, and MTU0 to MTU4.
3. For on-chip peripheral module requests, the transfer source is in cycle steal mode except CMT0, CMT1, and MTU0 to MTU4.
4. Access size permitted for the on-chip peripheral module register functioning as the transfer source or transfer destination.
5. If the transfer request is an external request, channels 0 to 1 are only available.

Bus Mode and Channel Priority Order: When channel 1 is transferring data in burst mode and a request arrives for transfer on channel 0, which has higher-priority, the data transfer on channel 0 will begin immediately. In this case, if the transfer on channel 0 is also in burst mode, the transfer on channel 1 will only resume on completion of the transfer on channel 0.

When channel 0 is in cycle steal mode, one transfer-unit of the data on this channel, which has the higher priority, is transferred. Data is then transferred continuously on channel 1 without releasing the internal bus. The bus will then switch between the two in this order: channel 1, channel 0, channel 1, channel 0, etc. That is, the bus state changes so that CPU cycles are for burst-mode transfer after the data transfer in cycle steal mode has been completed. An example of this is shown in figure 13.12. When multiple channels are in the burst mode, data transfer on the channel that has the highest priority is given precedence. When DMA transfer is being performed on multiple channels, bus mastership is not released to another bus-master device until all of the competing burst-mode transfers have been completed.

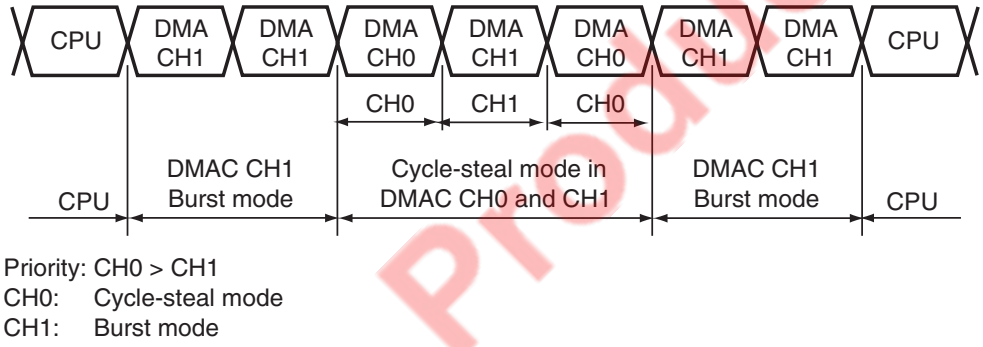


Figure 13.12 Bus State when Multiple Channels Are Operating

In the round-robin mode, do not mix channels in the cycle steal and burst modes. In this case, although the transfer operation on each channel will be performed correctly, switching between channels might not correctly follow the priority order.

13.4.5 Number of Bus Cycle States and $\overline{\text{DREQ}}$ Pin Sampling Timing

Number of Bus Cycle States: When the DMAC is the bus master, the number of bus cycle states is controlled by the bus state controller (BSC) in the same way as when the CPU is the bus master. For details, see section 12, Bus State Controller (BSC).

$\overline{\text{DREQ}}$ Pin Sampling Timing: Figures 13.13 to 13.16 show the $\overline{\text{DREQ}}$ input sampling timings in each bus mode.

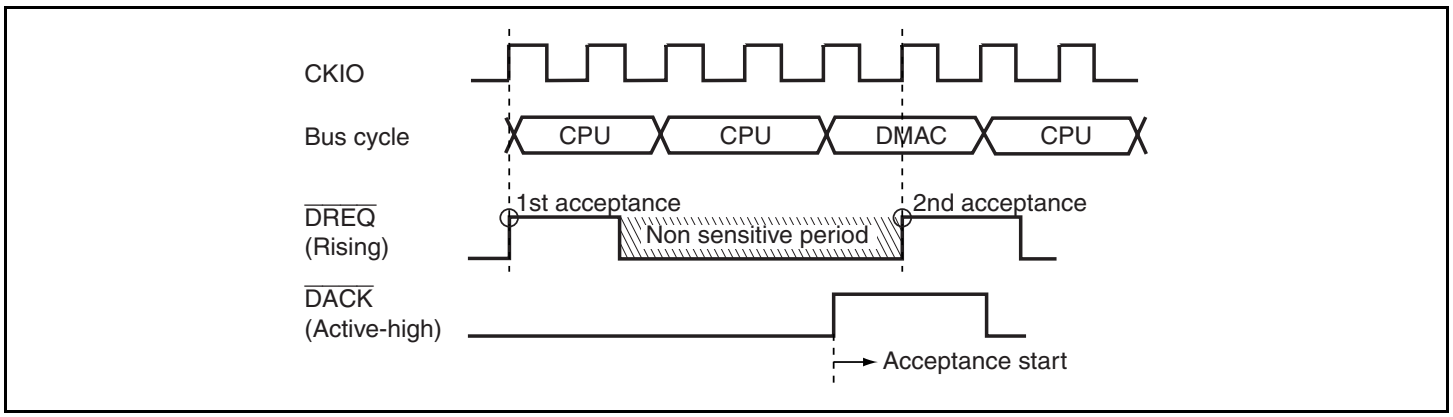


Figure 13.13 Example of $\overline{\text{DREQ}}$ Input Detection in Cycle Steal Mode Edge Detection

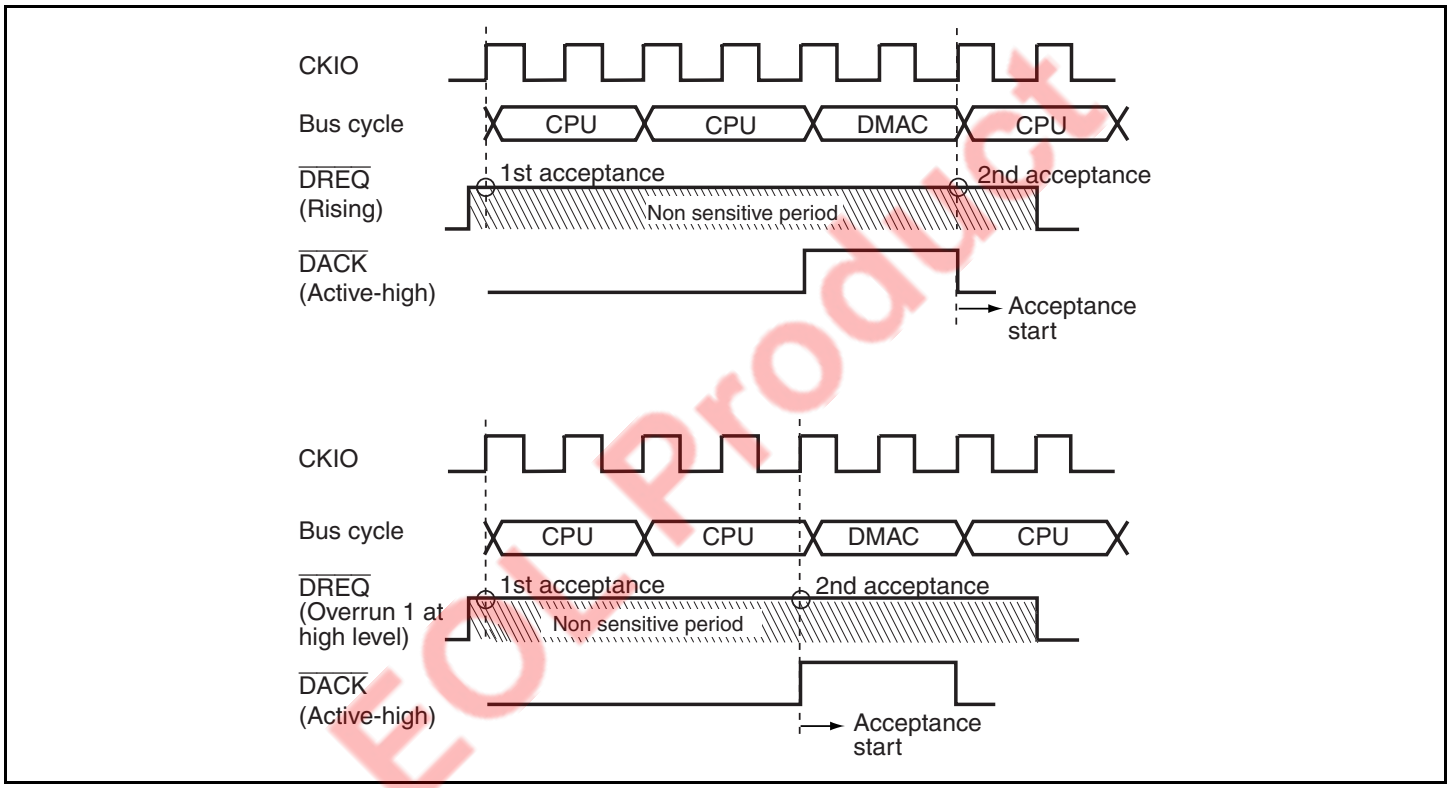


Figure 13.14 Example of $\overline{\text{DREQ}}$ Input Detection in Cycle Steal Mode Level Detection

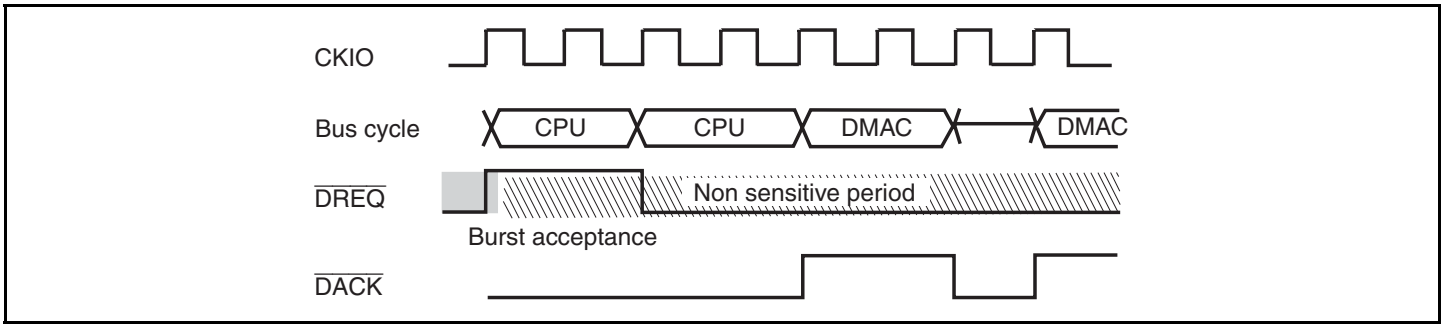


Figure 13.15 Example of $\overline{\text{DREQ}}$ Input Detection in Burst Mode Edge Detection

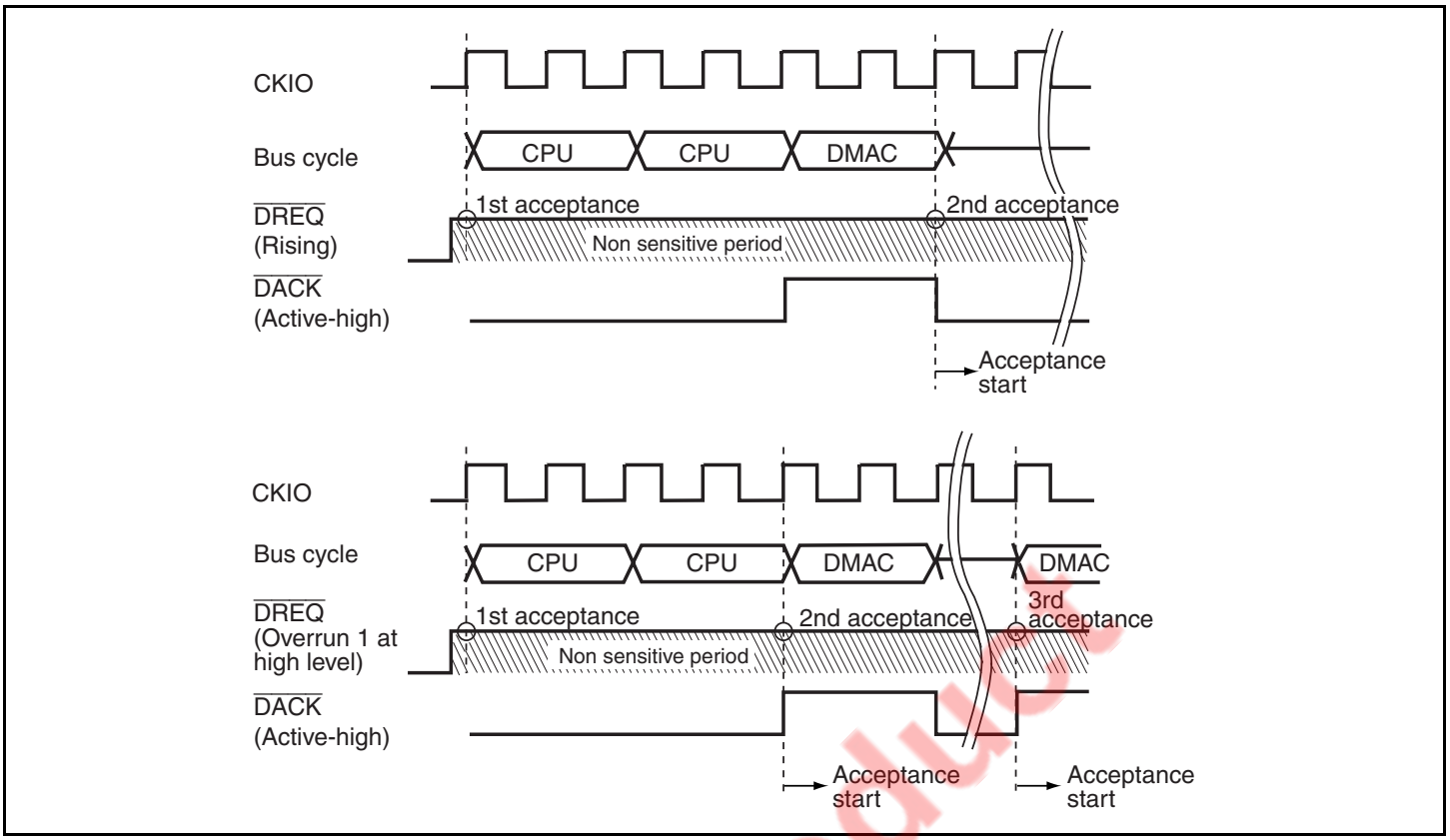


Figure 13.16 Example of $\overline{\text{DREQ}}$ Input Detection in Burst Mode Level Detection

Figure 13.17 shows the TEND output timing.

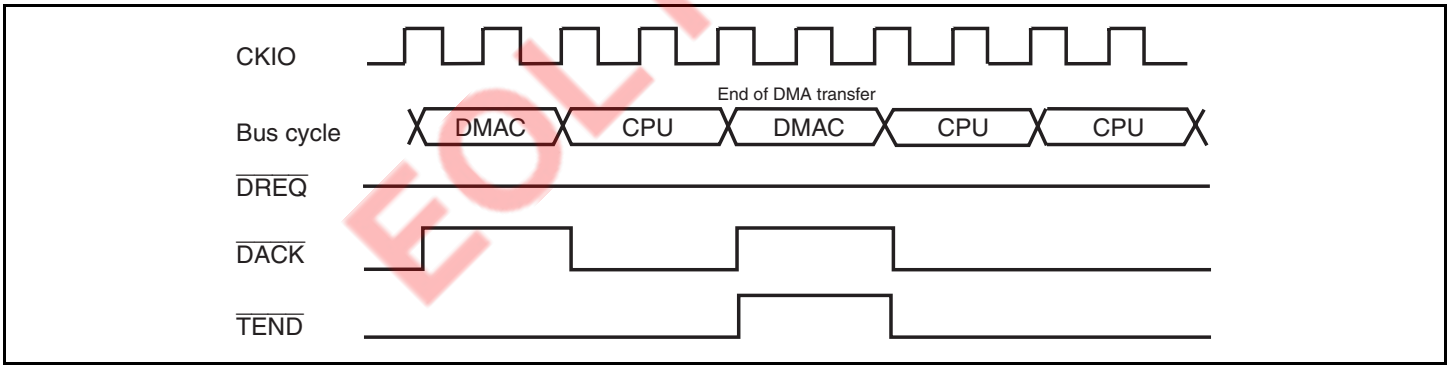
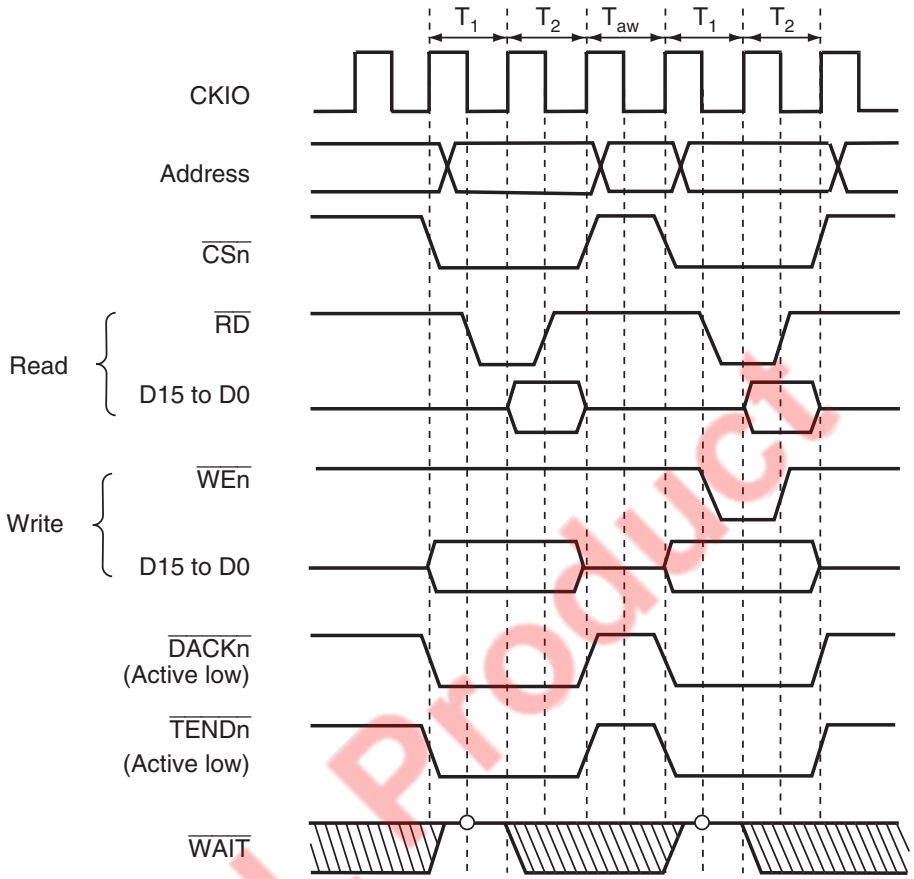


Figure 13.17 Example of $\overline{\text{DREQ}}$ Input Detection in Burst Mode Level Detection

To execute a longword access to an 8-bit or 16-bit external device or to execute a word access to an 8-bit external device, the $\overline{\text{DACK}}$ and $\overline{\text{TEND}}$ outputs are divided for data alignment as shown in figure 13.18.



Note: $\overline{\text{TEND}}$ is asserted for the last transfer unit of DMA transfers.
If a transfer unit is divided into multiple bus cycles and
if $\overline{\text{CSn}}$ is negated during the bus cycle, $\overline{\text{TEND}}$ is also divided.

Figure 13.18 BSC Ordinary Memory Access
(No Wait, Idle Cycle 1, Longword Access to 16-Bit Device)

13.4.6 Completion of DMA Transfer

The conditions for the completion of DMA transfer differ according to whether we are considering completion of transfer on individual channels or simultaneous completion of transfer on all channels.

1. Conditions for the completion of transfer on individual channels

   Either of the following events indicates the completion of transfer on the corresponding channel.

   — The value in the DMA transfer count register (TCR) becomes 0.

   — The DMA enable bit (DE) in the DMA channel control register (CHCR) becomes 0.

   A. Completion of transfer indicated by TCR = 0

      The TCR value becomes 0 when the DMA transfer on the corresponding channel has been completed, and the transfer-end bit flag (TE) is set to indicate this. In this case, if the interrupt enable bit (IE) has been set, a DMAC interrupt (DEI) request is sent to the CPU. When transferring data in 16-byte units, specify a number of transfers, as for transfers with other transfer-units.

   B. Completion of transfer indicated by DE = 0 in CHCR

      Clearing of the DMA enable bit (DE) of CHCR halts DMA transfer on the corresponding channel. In this case, the TE bit is not set.

2. Concurrent completion of transfer on all channels:

   Either of the following events indicates the concurrent completion of transfer on all channels.

   — The NMI flag bit (NMIF) or address error flag bit (AE) of the DMA operation register becomes 1.

   — The DMA master enable bit (DME) of DMAOR becomes 0.

   A. Completion of transfer indicated by NMIF = 1 or AE = 1 in DMAOR

      When an NMI interrupt is generated or the DMAC generates an address error and the NMIF bit or AE bit of DMAOR is set to 1, DMA transfer on all channels is suspended. The contents of the DMA source address register (SAR), the DMA destination register (DAR), and the DMA transfer count register (TCR) are updated (including that of the channel on which the address error occurred) by the transfer immediately before the suspension. If the transfer is the final transfer, TE becomes 1 and the transfer is then completed. If an address error is generated during transfer in the dual address mode, pay attention on the following points.

- When an address error occurs during a read cycle:

  Neither read cycles nor write cycles are generated; only the transfer request is cleared. However, when the transfer-request source was an on-chip peripheral module (MTU), use whichever of the following methods is appropriate to clear the transfer request.

a. When the TC bit of CHCR is 1: Clear the corresponding flag to resume a transfer after address-error exception processing. In this case, the transfer on the corresponding channel resumes when the DE bit is set to 1. If you do not want transfer to resume on a channel, perform a dummy transfer on that channel to clear the transfer request; do this by setting 1 in TCR and dummy addresses in the SAR and DAR.

b. When the TC bit of CHCR is 0: Use software to clear the transfer-request flag of the MTU.

- When an address error occurs during a write cycle:

  Only read cycles are generated and the transfer request is cleared. However, when the transfer-request source is the on-chip peripheral module (MTU) and the TC bit of CHCRn is set to 1, clear the transfer request by software, in the same way as when an address error occurs during a read cycle, described above.

B. Completion of transfer by clearing DME of DMAOR to 0

When the DME bit of DMAOR is cleared to 0, DMA transfer on all channels is forcibly suspended after the current transfer has been completed. If the suspended transfer was the final transfer, TE is set to 1 and the transfer is then completed.

13.4.7 Notes on Usage

1. Clear the DE bit for the corresponding channel before changing the value in the channel control register (CHCR) for that channel of the DMAC.
2. Do not place the system in software standby mode during DMA transfer and do not select module standby mode by setting the module standby bit of the DMAC. Clear the DE bits of all channels before any transition to the software standby mode or module standby mode.
3. Ensure that the system is in the normal operating state for the execution of any DMA transfer where locations in the U memory or X/Y memory are selected as the sources or destinations of the data. While the DMAC can operate in sleep mode, the U memory and X/Y memory are in the operation-stopped state. Accordingly, access from the DMAC is not possible.
4. The same internal request cannot be set for multiple channels.
5. The transfer request should be implemented after the settings of registers in DMAC have been completed.

6. Note the followings when the DMA transfer request is sent from the SCIF.

   Even when the DMAC has completed the TCR times of transfers (the TE bit in CHCR = 1), the DMAC accepts and keeps the transfer request from the SCIF (max. one time of transfer) if all the conditions shown below are satisfied. The DMA transfer, however, is not executed because the TE bit is set to 1. Clearing the TE bit in this condition can immediately restart the transfer.

   Conditions that make the DMA transfer request acceptable:

   — The DME bit in the DMA operation register (DMAOR) is set to 1.

   — The DE bit in the DMA channel control register (CHCR) is set to 1.

   — The peripheral module SCIF is set to the DMA extension resource selector (DMARS).

   Take special care when the SCIF transfer is executed by the DMAC. If the transfer restart is not desired, prevent the transfer from restarting by implementing one of the measures shown below.

   Preventive measures:

   — Clear the DE bit of CHCR in the end interrupt routine of the DMAC. (The DMA transfer request from the SCIF is not accepted.) In this case, set the end interrupt of the DMAC to have the highest priority.

   — Set 1 to TCR, and dummy addresses to SAR and DAR, respectively. Then perform the dummy transfer to clear the transfer request in the DMAC.

13.4.8 Notes On DREQ Sampling When DACK is Divided in External Access

(1) Error Phenomenon

When the DACK output is divided in an external access, DREQ may be sampled twice at maximum in the external access.

(2) Error Conditions and Phenomenon

Conditions: The DACK output is divided in an external access when:

- 16-byte access,
- 32-bit access to the 8-bit space,
- 16-bit access to the 8-bit space, or
- 32-bit access to the 16-bit space

is performed with either of the following idle cycle settings made:

- Idle cycles between write-write cycles (IWW = 01 or more)

- Idle cycles between read-read cycles in the same spaces (IWRRS = 01 or more)
- External wait mask specification (WM = 0).

In addition to the above conditions, the following conditions are included depending on the detection method of DREQ.

- For DREQ level detection: only write access
- For DREQ edge detection: both write access and read access

Phenomenon: The detection timings of the DREQ pin in the above access are shown in figures 13.19 to 13.22.

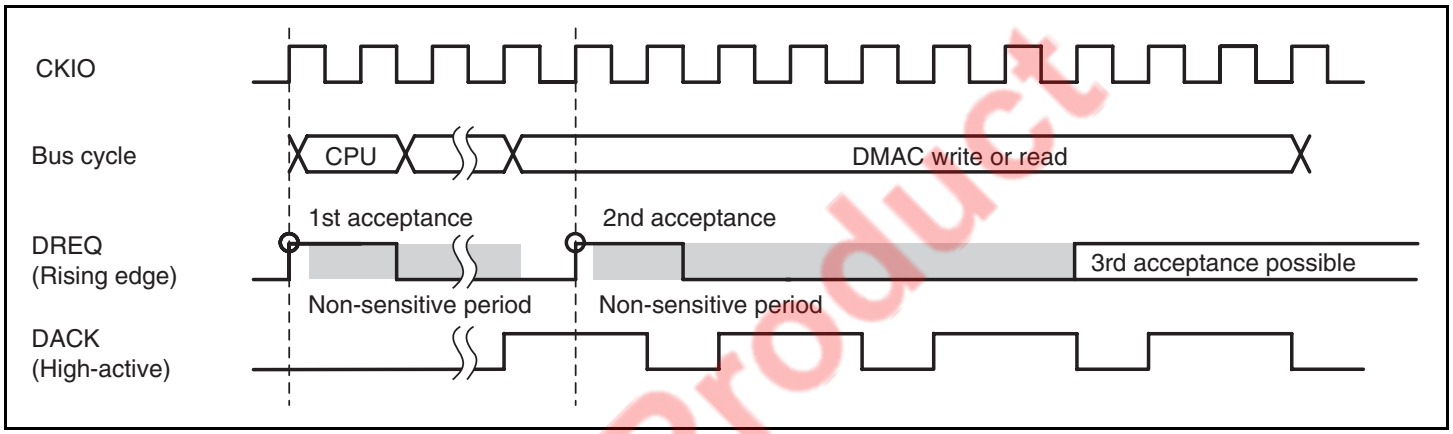


Figure 13.19 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection When DACK is Divided to 4 by Idle Cycles

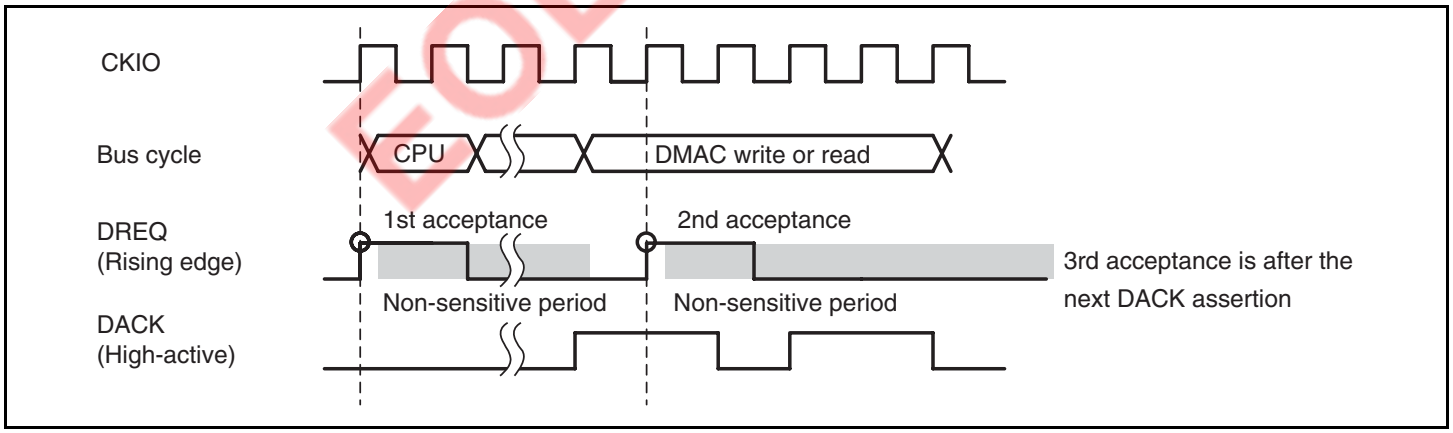


Figure 13.20 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection When DACK is Divided to 2 by Idle Cycles

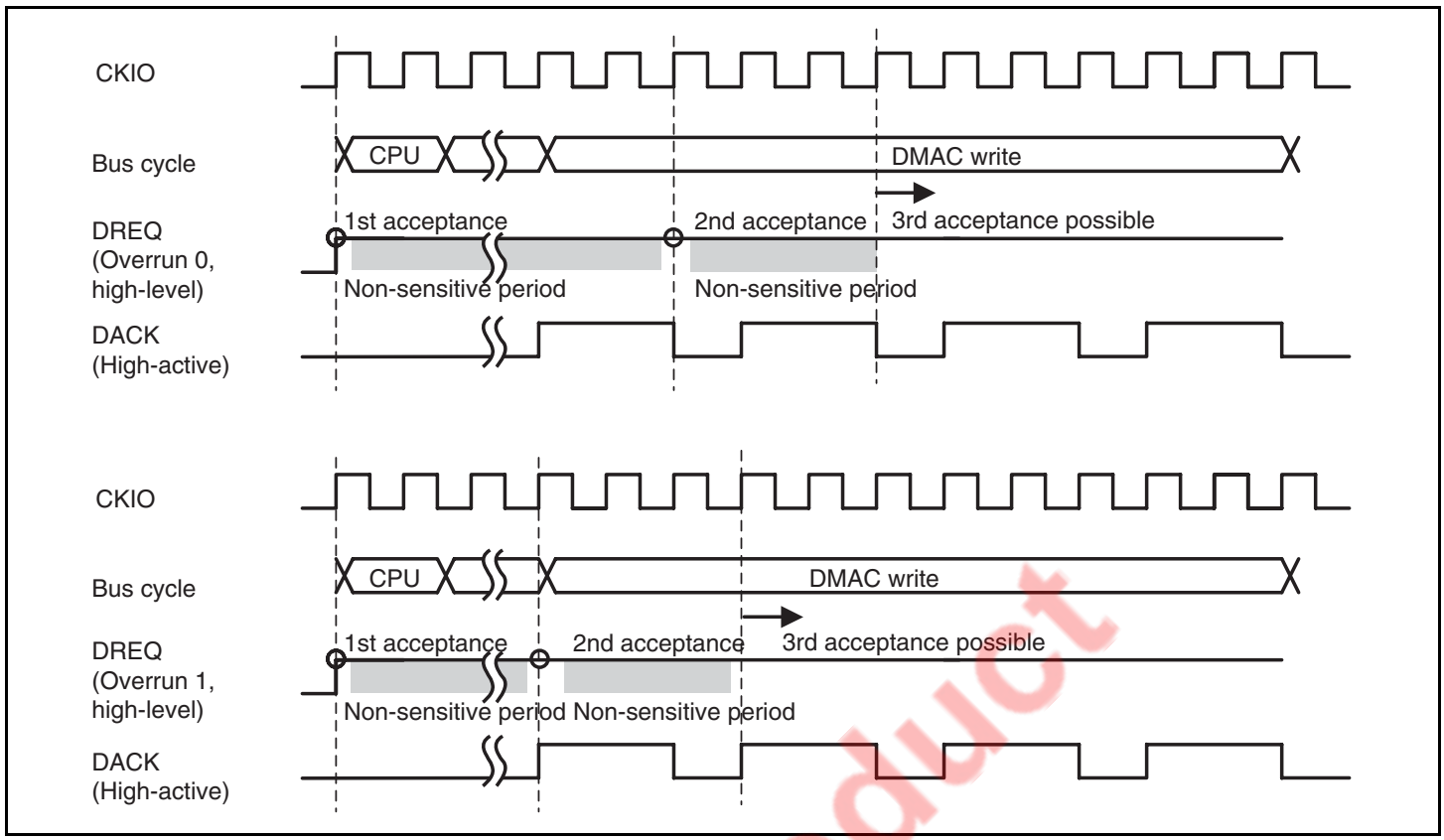


Figure 13.21 Example of DREQ Input Detection in Cycle Steal Mode Level Detection When DACK is Divided to 4 by Idle Cycles

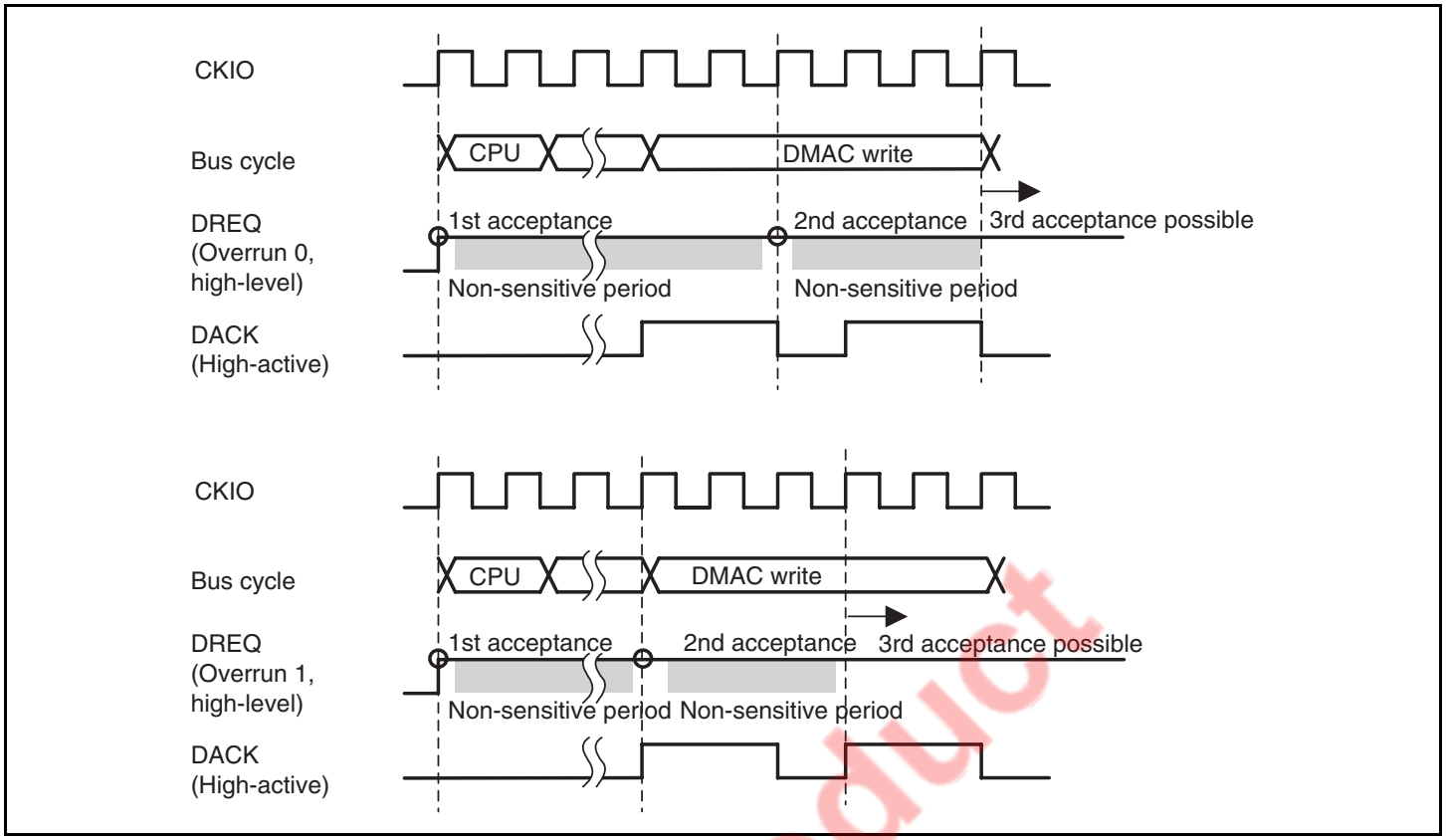


Figure 13.22 Example of DREQ Input Detection in Cycle Steal Mode Level Detection When DACK is Divided to 2 by Idle Cycles

(3) Notes

For the external access described in (2) above, note the following.

1. When the DREQ edge is detected, input one DREQ edge at maximum in the bus cycle.
2. When the DREQ level is detected in overrun 0, negate the DREQ input in the bus cycle after the detection of the first DACK output negation and before the second DACK output negation.
3. When the DREQ level is detected in overrun 1, negate DREQ input after the detection of the first DACK output assertion and before the second DACK output assertion.

Section 14 U Memory

This LSI has on-chip U memory. It can be used by the CPU, DSP, and DMAC to store instructions or data.

14.1 Features

The U memory features are listed in table 14.1.

Table 14.1 U Memory Specifications

Parameter	Features
Addressing method	Mapping is possible in space P0 or P2
Ports	2 independent read/write ports • 8-/16-/32-bit access from the CPU (via L bus or I bus) • 16-/32-bit access from the DSP (via L bus or I bus) • 8-/16-32-bit access from the CPU (via I bus)
Size	128 kbytes

The U memory resides in addresses H'055F0000 to H'0560FFFF in space P0 or addresses H'A55F0000 to H'A560FFFF (128 kbytes) in space P2. The U memory is divided into page 0 and page 1 according to the addresses. The U memory can be accessed from the L bus and I bus.

In the event of simultaneous accesses to the same address from different buses, the priority order is : I bus > L bus. Since this kind of conflict tends to lower U memory accessibility, it is advisable to provide software measures to prevent such conflict as far as possible. For example, conflict will not arise if different memory or different pages are accessed by each bus.

U memory is accessed by the CPU or DSP from space P0 via the I bus, a conflict with the DMAC may occur on the I bus. Since this kind of conflict also tends to lower U memory accessibility, it is advisable to provide software measures to prevent such conflict as far as possible. For example, conflict on the I bus can be prevented by using space P2 when the U memory is accessed by the CPU or DSP.

14.2 U Memory Access from CPU

The U memory can be accessed by the CPU from spaces P0 and P2. Access from the CPU is via the I bus when U memory is space P0, and via the L bus when space P2. To use the L bus, one cycle access is performed unless page conflict occurs. Using the I bus takes more than one cycle.

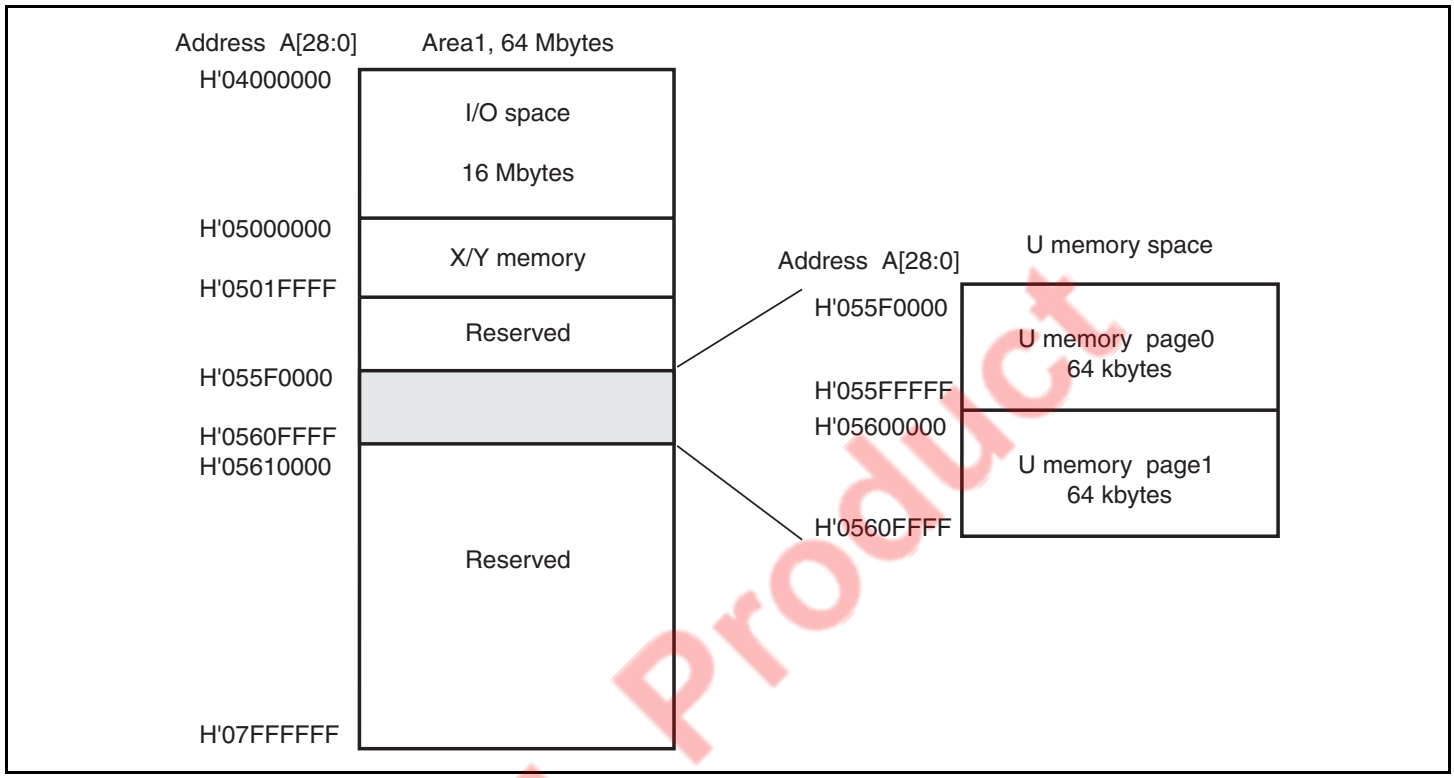


Figure 14.1 U Memory Address Mapping

14.3 U Memory Access from DSP

The DSP can access the U memory through spaces P0 and P2 using a single data transfer instruction. Access from the DSP is via the I bus when the address is space P0, and via the L bus when the address is space P2. To use the L bus, one cycle access is performed unless page conflict occurs. Using the I bus takes more than one cycle.

14.4 U Memory Access from DMAC

The U memory also exists on the I bus and can be accessed by the DMAC. Use addresses H'55F0000 to H'560FFFF.

14.5 Usage Note

When accessing the U memory by the CPU or the DSP, if the cache is on, access must be performed from space P2 (non-cacheable space). Operation during access from space P0 cannot be guaranteed. When the cache is off, spaces P0 and P2 can both be used.

14.6 Sleep Mode

In sleep mode, the U memory cannot be accessed by the I bus master module such as DMAC.

14.7 Address Error

When an address error in write access to the U memory occur, the contents of the U memory may be corrupted.

Section 15 User Debugging Interface (H-UDI)

This LSI incorporates a user debugging interface (H-UDI) and advanced user debugger (AUD) for a boundary scan function and emulator support.

This section describes the H-UDI. The AUD is a function exclusively for use by an emulator. Refer to the User's Manual for the relevant emulator for details of the AUD.

15.1 Features

The user debugging interface (H-UDI) is a serial I/O interface which conforms to JTAG (Joint Test Action Group, IEEE Standard 1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture) specifications.

The H-UDI in this LSI supports a boundary scan mode, and is also used for emulator connection.

When using an emulator, H-UDI functions should not be used. Refer to the emulator manual for the method of connecting the emulator.

Figure 15.1 shows a block diagram of the H-UDI.

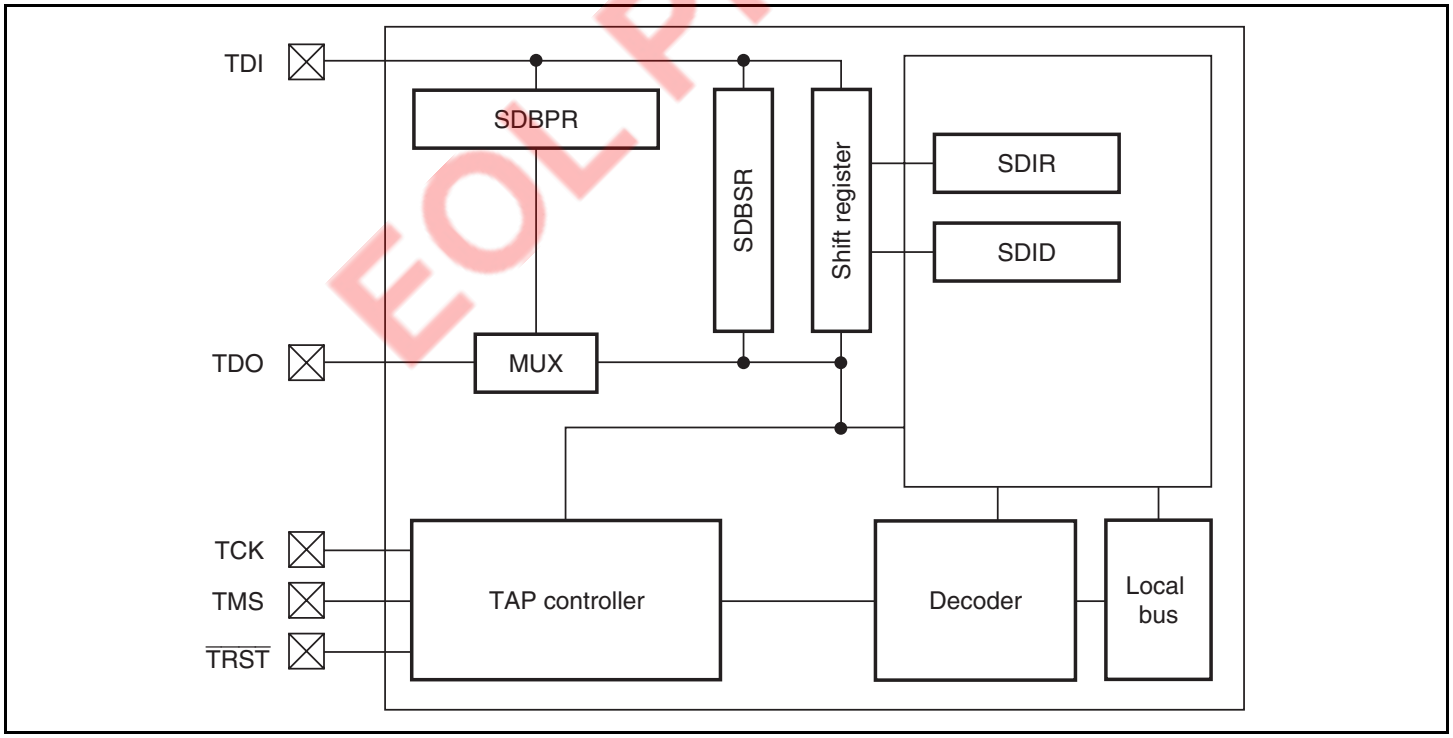


Figure 15.1 Block Diagram of H-UDI

15.2 Input/Output Pins

Table 15.1 shows the pin configuration of the H-UDI.

Table 15.1 Pin Configuration

Pin Name	Input/Output	Description
TCK	Input	Serial data input/output clock pin Data is serially supplied to the H-UDI from the data input pin (TDI), and output from the data output pin (TDO), in synchronization with this clock.
TMS	Input	Mode select input pin The state of the TAP control circuit is determined by changing this signal in synchronization with TCK. The protocol conforms to the JTAG standard (IEEE Std.1149.1).
$\overline{\text{TRST}}$	Input	Reset input pin Input is accepted asynchronously with respect to TCK, and when low, the H-UDI is reset. $\overline{\text{TRST}}$ must be low for a constant period when power is turned on regardless of using the H-UDI function. This is different from the JTAG standard. See section 15.4.2, Reset Configuration, for more information.
TDI	Input	Serial data input pin Data transfer to the H-UDI is executed by changing this signal in synchronization with TCK.
TDO	Output	Serial data output pin Data read from the H-UDI is executed by reading this pin in synchronization with TCK. The data output timing depends on the command type set in the SDIR. See section 15.3.2 Instruction Register (SDIR), for more information.
$\overline{\text{ASEMD0}}$*	Input	ASE mode select pin If a low level is input at the $\overline{\text{ASEMD0}}$ pin while the $\overline{\text{RESETP}}$ pin is asserted, ASE mode is entered; if a high level is input, normal mode is entered. In ASE mode, dedicated emulator function can be used. The input level at the $\overline{\text{ASEMD0}}$ pin should be held for at least one cycle after $\overline{\text{RESETP}}$ negation.
$\overline{\text{ASEBRKAK}}$, $\overline{\text{AUDSYNC}}$, AUDATA3 to AUDATA 0, AUDCK	Output	Dedicated emulator pin

Note: * When the emulator is not in use, fix this pin to the high level.

15.3 Register Descriptions

The H-UDI has the following registers. Refer the section 24, List of Registers, for the addresses and access size for these registers.

- Bypass register (SDBPR)
- Instruction register (SDIR)
- Boundary scan register (SDBSR)
- ID register (SDID)

15.3.1 Bypass Register (SDBPR)

SDBPR is a 1-bit register that cannot be accessed by the CPU. When SDIR is set to the bypass mode, SDBPR is connected between H-UDI pins TDI and TDO. The initial value is undefined but is initialized to 0 if the TAP is in Capture-DR state.

15.3.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register. The register is in JTAG IDCODE in its initial state. It is initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state, and can be written to by the H-UDI irrespective of the CPU mode. Operation is not guaranteed if a reserved command is set in this register.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	TI7 to TI5	All 1	R	Test Instruction 7 to 0
12	TI4	0	R	The H-UDI instruction is transferred to SDIR by a serial input from TDI. For commands, see table 15.2.
11 to 8	TI3 to TI0	All 1	R	
7 to 2	—	All 1	R	Reserved These bits are always read as 1.
1	—	0	R	Reserved This bit is always read as 0.
0	—	1	R	Reserved This bit is always read as 1.

Table 15.2 H-UDI Commands

Bits 15 to 8								
TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0	Description
0	0	0	0	—	—	—	—	JTAG EXTEST
0	0	1	0	—	—	—	—	JTAG CLAMP
0	0	1	1	—	—	—	—	JTAG HIGHZ
0	1	0	0	—	—	—	—	JTAG SAMPLE/PRELOAD
0	1	1	0	—	—	—	—	H-UDI reset negate
0	1	1	1	—	—	—	—	H-UDI reset assert
1	0	1	—	—	—	—	—	H-UDI interrupt
1	1	1	0	—	—	—	—	JTAG IDCODE (Initial value)
1	1	1	1	—	—	—	—	JTAG BYPASS
Other than the above								Reserved

15.3.3 Boundary Scan Register (SDBSR)

SDBSR is a 469-bit shift register, located on the PAD, for controlling the input/output pins of this LSI.

Using the EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ commands, a boundary scan test conforming to the JTAG standard can be carried out. Table 15.3 shows the correspondence between this LSI's pins and boundary scan register bits.

Table 15.3 This LSI Pins and Boundary Scan Register Bits

Bit	Pin Name	I/O	Bit	Pin Name	I/O
	From TDI		454	AUDATA3/PTJ11	IN
483	D7	IN	453	AUDSYNC/PTJ12	IN
482	D6	IN	452	$\overline{\text{NMI}}$	IN
481	D5	IN	451	$\overline{\text{IRQ0}}$/PTJ0	IN
480	D4	IN	450	$\overline{\text{IRQ1}}$/PTJ1	IN
479	D3	IN	449	$\overline{\text{IRQ2}}$/PTJ2	IN
478	D2	IN	448	$\overline{\text{IRQ3}}$/PTJ3	IN
477	D1	IN	447	$\overline{\text{IRQ4}}$/PTJ4	IN
476	D0	IN	446	$\overline{\text{IRQ5}}$/PTJ5	IN
475	$\overline{\text{CS3}}$/PTA3	IN	445	$\overline{\text{IRQ6}}$/PTJ6	IN
474	$\overline{\text{CS2}}$/PTA2	IN	444	$\overline{\text{IRQ7}}$/PTJ7	IN
473	UCLK/PTB0	IN	443	SCK0/PTH0	IN
472	VBUS/PTB1	IN	442	D7	OUT
471	SUSPND/PTB2	IN	441	D6	OUT
470	XVDATA/PTB3	IN	440	D5	OUT
469	TXENL/PTB4	IN	439	D4	OUT
468	TXDMNS/PTB5	IN	438	D3	OUT
467	TXDPLS/PTB6	IN	437	D2	OUT
466	DMNS/PTB7	IN	436	D1	OUT
465	DPLS/PTB8	IN	435	D0	OUT
464	A19/PTA8	IN	434	$\overline{\text{CS3}}$/PTA3	OUT
463	A20/PTA9	IN	433	$\overline{\text{CS2}}$/PTA2	OUT
462	A21/PTA10	IN	432	UCLK/PTB0	OUT
461	A22/PTA11	IN	431	VBUS/PTB1	OUT
460	A23/PTA12	IN	430	SUSPND/PTB2	OUT
459	A24/PTA13	IN	429	XVDATA/PTB3	OUT
458	A25/PTA14	IN	428	TXENL/PTB4	OUT
457	AUDATA0/PTJ8	IN	427	TXDMNS/PTB5	OUT
456	AUDATA1/PTJ9	IN	426	TXDPLS/PTB6	OUT
455	AUDATA2/PTJ10	IN	425	DMNS/PTB7	OUT

Bit	Pin Name	I/O	Bit	Pin Name	I/O
424	DPLS/PTB8	OUT	392	$\overline{CS3}$/PTA3	Control
423	A18	OUT	391	$\overline{CS2}$/PTA2	Control
422	A19/PTA8	OUT	390	UCLK/PTB0	Control
421	A20/PTA9	OUT	389	VBUS/PTB1	Control
420	A21/PTA10	OUT	388	SUSPND/PTB2	Control
419	A22/PTA11	OUT	387	XVDATA/PTB3	Control
418	A23/PTA12	OUT	386	TXENL/PTB4	Control
417	A24/PTA13	OUT	385	TXDMNS/PTB5	Control
416	AUDCK	OUT	384	TXDPLS/PTB6	Control
415	A25/PTA14	OUT	383	DMNS/PTB7	Control
414	AUDATA0/PTJ8	OUT	382	DPLS/PTB8	Control
413	AUDATA1/PTJ9	OUT	381	A18	Control
412	AUDATA2/PTJ10	OUT	380	A19/PTA8	Control
411	AUDATA3/PTJ11	OUT	379	A20/PTA9	Control
410	$\overline{AUDSYNC}$/PTJ12	OUT	378	A21/PTA10	Control
409	$\overline{IRQ0}$/PTJ0	OUT	377	A22/PTA11	Control
408	$\overline{IRQ1}$/PTJ1	OUT	376	A23/PTA12	Control
407	$\overline{IRQ2}$/PTJ2	OUT	375	A24/PTA13	Control
406	$\overline{IRQ3}$/PTJ3	OUT	374	AUDCK	Control
405	$\overline{IRQ4}$/PTJ4	OUT	373	A25/PTA14	Control
404	$\overline{IRQ5}$/PTJ5	OUT	372	AUDATA0/PTJ8	Control
403	$\overline{IRQ6}$/PTJ6	OUT	371	AUDATA1/PTJ9	Control
402	$\overline{IRQ7}$/PTJ7	OUT	370	AUDATA2/PTJ10	Control
401	SCK0/PTH0	OUT	369	AUDATA3/PTJ11	Control
400	D7	Control	368	AUDSYNC/PTJ12	Control
399	D6	Control	367	$\overline{IRQ0}$/PTJ0	Control
398	D5	Control	366	$\overline{IRQ1}$/PTJ1	Control
397	D4	Control	365	$\overline{IRQ2}$/PTJ2	Control
396	D3	Control	364	$\overline{IRQ3}$/PTJ3	Control
395	D2	Control	363	$\overline{IRQ4}$/PTJ4	Control
394	D1	Control	362	$\overline{IRQ5}$/PTJ5	Control
393	D0	Control	361	$\overline{IRQ6}$/PTJ6	Control

Bit	Pin Name	I/O	Bit	Pin Name	I/O
360	$\overline{\text{IRQ7}}$/PTJ7	Control	328	TCLKD/PTF8	IN
359	SCK0/PTH0	Control	327	TCLKC/PTF9	IN
358	$\overline{\text{CTS0}}$/PTH1	IN	326	TCLKB/PTF10	IN
357	TxD0/PTH2	IN	325	TCLKA/PTF11	IN
356	RxD0/PTH3	IN	324	$\overline{\text{POE0}}$/PTF12	IN
355	$\overline{\text{RTS0}}$/PTH4	IN	323	$\overline{\text{POE1}}$/PTF13	IN
354	SCK1/PTH5	IN	322	$\overline{\text{POE2}}$/PTF14	IN
353	$\overline{\text{CTS1}}$/PTH6	IN	321	$\overline{\text{POE3}}$/PTF15	IN
352	TxD1/PTH7	IN	320	PTF0	IN
351	RxD1/PTH8	IN	319	PTF1	IN
350	$\overline{\text{RTS1}}$/PTH9	IN	318	PTF2	IN
349	SCK2/PTH10	IN	317	PTF3	IN
348	$\overline{\text{CTS2}}$/PTH11	IN	316	PTF4	IN
347	TxD2/PTH12	IN	315	PTF5	IN
346	RxD2/PTH13	IN	314	PTF6	IN
345	$\overline{\text{RTS2}}$/PTH14	IN	313	PTF7	IN
344	TIOC4D/PTE0	IN	312	PTG8	IN
343	TIOC4C/PTE1	IN	311	PTG9/SCL	IN
342	TIOC4B/PTE2	IN	310	PTG10/SDA	IN
341	TIOC4A/PTE3	IN	309	PTG11	IN
340	TIOC3D/PTE4	IN	308	PTG12	IN
339	TIOC3B/PTE6	IN	307	PTG13	IN
338	TIOC3C/PTE5	IN	306	$\overline{\text{CTS0}}$/PTH1	OUT
337	TIOC3A/PTE7	IN	305	TxD0/PTH2	OUT
336	TIOC2B/PTE8	IN	304	RxD0/PTH3	OUT
335	TIOC2A/PTE9	IN	303	$\overline{\text{RTS0}}$/PTH4	OUT
334	TIOC1B/PTE10	IN	302	SCK1/PTH5	OUT
333	TIOC1A/PTE11	IN	301	$\overline{\text{CTS1}}$/PTH6	OUT
332	TIOC0D/PTE12	IN	300	TxD1/PTH7	OUT
331	TIOC0C/PTE13	IN	299	RxD1/PTH8	OUT
330	TIOC0B/PTE14	IN	298	$\overline{\text{RTS1}}$/PTH9	OUT
329	TIOC0A/PTE15	IN	297	SCK2/PTH10	OUT

Bit	Pin Name	I/O	Bit	Pin Name	I/O
296	$\overline{\text{CTS2}}$/PTH11	OUT	264	PTF4	OUT
295	TxD2/PTH12	OUT	263	PTF5	OUT
294	RxD2/PTH13	OUT	262	PTF6	OUT
293	$\overline{\text{RTS2}}$/PTH14	OUT	261	PTF7	OUT
292	TIOC4D/PTE0	OUT	260	PTG8	OUT
291	TIOC4C/PTE1	OUT	259	PTG9/SCL	OUT
290	TIOC4B/PTE2	OUT	258	PTG10/SDA	OUT
289	TIOC4A/PTE3	OUT	257	PTG11	OUT
288	TIOC3D/PTE4	OUT	256	PTG12	OUT
287	TIOC3B/PTE6	OUT	255	PTG13	OUT
286	TIOC3C/PTE5	OUT	254	$\overline{\text{CTS0}}$/PTH1	Control
285	TIOC3A/PTE7	OUT	253	TxD0/PTH2	Control
284	TIOC2B/PTE8	OUT	252	RxD0/PTH3	Control
283	TIOC2A/PTE9	OUT	251	$\overline{\text{RTS0}}$/PTH4	Control
282	TIOC1B/PTE10	OUT	250	SCK1/PTH5	Control
281	TIOC1A/PTE11	OUT	249	$\overline{\text{CTS1}}$/PTH6	Control
280	TIOC0D/PTE12	OUT	248	TxD1/PTH7	Control
279	TIOC0C/PTE13	OUT	247	RxD1/PTH8	Control
278	TIOC0B/PTE14	OUT	246	$\overline{\text{RTS1}}$/PTH9	Control
277	TIOC0A/PTE15	OUT	245	SCK2/PTH10	Control
276	TCLKD/PTF8	OUT	244	$\overline{\text{CTS2}}$/PTH11	Control
275	TCLKC/PTF9	OUT	243	TxD2/PTH12	Control
274	TCLKB/PTF10	OUT	242	RxD2/PTH13	Control
273	TCLKA/PTF11	OUT	241	$\overline{\text{RTS2}}$/PTH14	Control
272	$\overline{\text{POE0}}$/PTF12	OUT	240	TIOC4D/PTE0	Control
271	$\overline{\text{POE1}}$/PTF13	OUT	239	TIOC4C/PTE1	Control
270	$\overline{\text{POE2}}$/PTF14	OUT	238	TIOC4B/PTE2	Control
269	$\overline{\text{POE3}}$/PTF15	OUT	237	TIOC4A/PTE3	Control
268	PTF0	OUT	236	TIOC3D/PTE4	Control
267	PTF1	OUT	235	TIOC3B/PTE6	Control
266	PTF2	OUT	234	TIOC3C/PTE5	Control
265	PTF3	OUT	233	TIOC3A/PTE7	Control

Bit	Pin Name	I/O	Bit	Pin Name	I/O
232	TIOC2B/PTE8	Control	200	AN2/PTG2	IN
231	TIOC2A/PTE9	Control	199	AN3/PTG3	IN
230	TIOC1B/PTE10	Control	198	AN4/PTG4	IN
229	TIOC1A/PTE11	Control	197	AN5/PTG5	IN
228	TIOC0D/PTE12	Control	196	AN6/PTG6	IN
227	TIOC0C/PTE13	Control	195	AN7/PTG7	IN
226	TIOC0B/PTE14	Control	194	$\overline{\text{DREQ0}}$/PTC9	IN
225	TIOC0A/PTE15	Control	193	$\overline{\text{DREQ1}}$/PTC10	IN
224	TCLKD/PTF8	Control	192	STATUS0/PTC14	IN
223	TCLKC/PTF9	Control	191	STATUS1/PTC15	IN
222	TCLKB/PTF10	Control	190	$\overline{\text{BREQ}}$/PTC6	IN
221	TCLKA/PTF11	Control	189	$\overline{\text{BACK}}$/PTC7	IN
220	$\overline{\text{POE0}}$/PTF12	Control	188	*VccQ	IN
219	$\overline{\text{POE1}}$/PTF13	Control	187	*VccQ	IN
218	$\overline{\text{POE2}}$/PTF14	Control	186	$\overline{\text{ASEBRKAK}}$/PTC13	IN
217	$\overline{\text{POE3}}$/PTF15	Control	185	MD3	IN
216	PTF0	Control	184	MD2	IN
215	PTF1	Control	183	*VccQ	IN
214	PTF2	Control	182	MD0	IN
213	PTF3	Control	181	$\overline{\text{CS6B}}$/PTC4	IN
212	PTF4	Control	180	$\overline{\text{CS6A}}$/PTC3	IN
211	PTF5	Control	179	$\overline{\text{CS5B}}$/PTC2	IN
210	PTF6	Control	178	$\overline{\text{CS5A}}$/PTC1	IN
209	PTF7	Control	177	$\overline{\text{CS4}}$/PTC0	IN
208	PTG8	Control	176	$\overline{\text{WAIT}}$	IN
207	PTG9/SCL	Control	175	$\overline{\text{TEND}}$/PTC8	IN
206	PTG10/SDA	Control	174	$\overline{\text{FRAME}}$/PTC5	IN
205	PTG11	Control	173	$\overline{\text{DACK0}}$/PTC11	IN
204	PTG12	Control	172	$\overline{\text{DACK1}}$/PTC12	IN
203	PTG13	Control	171	D31/PTD15	IN
202	AN0/PTG0	IN	170	D30/PTD14	IN
201	AN1/PTG1	IN	169	D29/PTD13	IN

Bit	Pin Name	I/O	Bit	Pin Name	I/O
168	D28/PTD12	IN	136	$\overline{\text{BREQ}}$/PTC6	Control
167	D27/PTD11	IN	135	$\overline{\text{BACK}}$/PTC7	Control
166	D26/PTD10	IN	134	$\overline{\text{ASEBRKAK}}$/PTC13	Control
165	$\overline{\text{DREQ0}}$/PTC9	OUT	133	$\overline{\text{CS6B}}$/PTC4	Control
164	$\overline{\text{DREQ1}}$/PTC10	OUT	132	$\overline{\text{CS6A}}$/PTC3	Control
163	STATUS0/PTC14	OUT	131	$\overline{\text{CS5B}}$/PTC2	Control
162	STATUS1/PTC15	OUT	130	$\overline{\text{CS5A}}$/PTC1	Control
161	$\overline{\text{BREQ}}$/PTC6	OUT	129	$\overline{\text{CS4}}$/PTC0	Control
160	$\overline{\text{BACK}}$/PTC7	OUT	128	$\overline{\text{CS0}}$	Control
159	$\overline{\text{ASEBRKAK}}$/PTC13	OUT	127	$\overline{\text{BS}}$	Control
158	$\overline{\text{CS6B}}$/PTC4	OUT	126	$\overline{\text{TEND}}$/PTC8	Control
157	$\overline{\text{CS6A}}$/PTC3	OUT	125	$\overline{\text{FRAME}}$/PTC5	Control
156	$\overline{\text{CS5B}}$/PTC2	OUT	124	$\overline{\text{RD}}$	Control
155	$\overline{\text{CS5A}}$/PTC1	OUT	123	$\overline{\text{DACK0}}$/PTC11	Control
154	$\overline{\text{CS4}}$/PTC0	OUT	122	$\overline{\text{DACK1}}$/PTC12	Control
153	$\overline{\text{CS0}}$	OUT	121	D31/PTD15	Control
152	$\overline{\text{BS}}$	OUT	120	D30/PTD14	Control
151	$\overline{\text{TEND}}$/PTC8	OUT	119	D29/PTD13	Control
150	$\overline{\text{FRAME}}$/PTC5	OUT	118	D28/PTD12	Control
149	$\overline{\text{RD}}$	OUT	117	D27/PTD11	Control
148	$\overline{\text{DACK0}}$/PTC11	OUT	116	D26/PTD10	Control
147	$\overline{\text{DACK1}}$/PTC12	OUT	115	D25/PTD9	IN
146	D31/PTD15	OUT	114	D24/PTD8	IN
145	D30/PTD14	OUT	113	D23/PTD7	IN
144	D29/PTD13	OUT	112	D22/PTD6	IN
143	D28/PTD12	OUT	111	D21/PTD5	IN
142	D27/PTD11	OUT	110	D20/PTD4	IN
141	D26/PTD10	OUT	109	D19/PTD3	IN
140	$\overline{\text{DREQ0}}$/PTC9	Control	108	D18/PTD2	IN
139	$\overline{\text{DREQ1}}$/PTC10	Control	107	D17/PTD1	IN
138	STATUS0/PTC14	Control	106	D16/PTD0	IN
137	STATUS1/PTC15	Control	105	$\overline{\text{CASU}}$/PTA5	IN

Bit	Pin Name	I/O	Bit	Pin Name	I/O
104	$\overline{\text{RASU}}$/PTA7	IN	72	$\overline{\text{RASL}}$/PTA6	OUT
103	CKE/PTA1	IN	71	A17	OUT
102	$\overline{\text{CASL}}$/PTA4	IN	70	A16	OUT
101	$\overline{\text{RASL}}$/PTA6	IN	69	A15	OUT
100	A0/PTA0	IN	68	A14	OUT
99	D15	IN	67	A13	OUT
98	D14	IN	66	A12	OUT
97	D13	IN	65	A11	OUT
96	D12	IN	64	A10	OUT
95	D11	IN	63	A9	OUT
94	D10	IN	62	A8	OUT
93	D9	IN	61	A7	OUT
92	D8	IN	60	A6	OUT
91	D25/PTD9	OUT	59	A5	OUT
90	D24/PTD8	OUT	58	A4	OUT
89	D23/PTD7	OUT	57	A3	OUT
88	D22/PTD6	OUT	56	A2	OUT
87	D21/PTD5	OUT	55	A1	OUT
86	D20/PTD4	OUT	54	A0/PTA0	OUT
85	D19/PTD3	OUT	53	D15	OUT
84	D18/PTD2	OUT	52	D14	OUT
83	D17/PTD1	OUT	51	D13	OUT
82	D16/PTD0	OUT	50	D12	OUT
81	RD$\overline{\text{WR}}$	OUT	49	D11	OUT
80	$\overline{\text{WE0}}$/DQMLL	OUT	48	D10	OUT
79	$\overline{\text{WE1}}$/DQMLU	OUT	47	D9	OUT
78	$\overline{\text{CASU}}$/PTA5	OUT	46	D8	OUT
77	$\overline{\text{WE3}}$/DQMUU/$\overline{\text{AH}}$	OUT	45	D25/PTD9	Control
76	$\overline{\text{RASU}}$/PTA7	OUT	44	D24/PTD8	Control
75	$\overline{\text{WE2}}$/DQMUL	OUT	43	D23/PTD7	Control
74	CKE/PTA1	OUT	42	D22/PTD6	Control
73	$\overline{\text{CASL}}$/PTA4	OUT	41	D21/PTD5	Control

Bit	Pin Name	I/O	Bit	Pin Name	I/O
40	D20/PTD4	Control	19	A11	Control
39	D19/PTD3	Control	18	A10	Control
38	D18/PTD2	Control	17	A9	Control
37	D17/PTD1	Control	16	A8	Control
36	D16/PTD0	Control	15	A7	Control
35	RD/$\overline{WR}$	Control	14	A6	Control
34	$\overline{WE0}$/DQMLL	Control	13	A5	Control
33	$\overline{WE1}$/DQMLU	Control	12	A4	Control
32	$\overline{CASU}$/PTA5	Control	11	A3	Control
31	$\overline{WE3}$/DQMUU/$\overline{AH}$	Control	10	A2	Control
30	$\overline{RASU}$/PTA7	Control	9	A1	Control
29	$\overline{WE2}$/DQMUL	Control	8	A0/PTA0	Control
28	CKE/PTA1	Control	7	D15	Control
27	$\overline{CASL}$/PTA4	Control	6	D14	Control
26	$\overline{RASL}$/PTA6	Control	5	D13	Control
25	A17	Control	4	D12	Control
24	A16	Control	3	D11	Control
23	A15	Control	2	D10	Control
22	A14	Control	1	D9	Control
21	A13	Control	0	D8	Control
20	A12	Control	to TDO		

Notes: 1. Control is an active-high signal.
2. When Control is driven high, the corresponding pin is driven by the value of OUT.
3. *VccQ is not the power supply for the LSI, but is still necessary for operation of the user functions. Accordingly, pull this pin up in the way described in the specifications. These pins must be pulled-up based on the specifications.

15.3.4 ID Register (SDID)

The ID register (SDID) is a 32-bit read-only register in which SDIDH and SDIDL are connected. Each register is a 16-bit that can be read by CPU.

The IDCODE command is set from the H-UDI pin. This register can be read from the TDO when the TAP state is Shift-DR. Writing is disabled.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DID31 to DID0	H'0027200F	R	Device ID Device ID register that is stipulated by JTAG. H'0027200F is initial value specific to the LSI. Upper four bits may be changed by the chip version. SDIDH corresponds to bits 31 to 16. SDIDL corresponds to bits 15 to 0.

15.4 Operation

15.4.1 TAP Controller

Figure 15.2 shows the internal states of the TAP controller. State transitions basically conform with the JTAG standard.

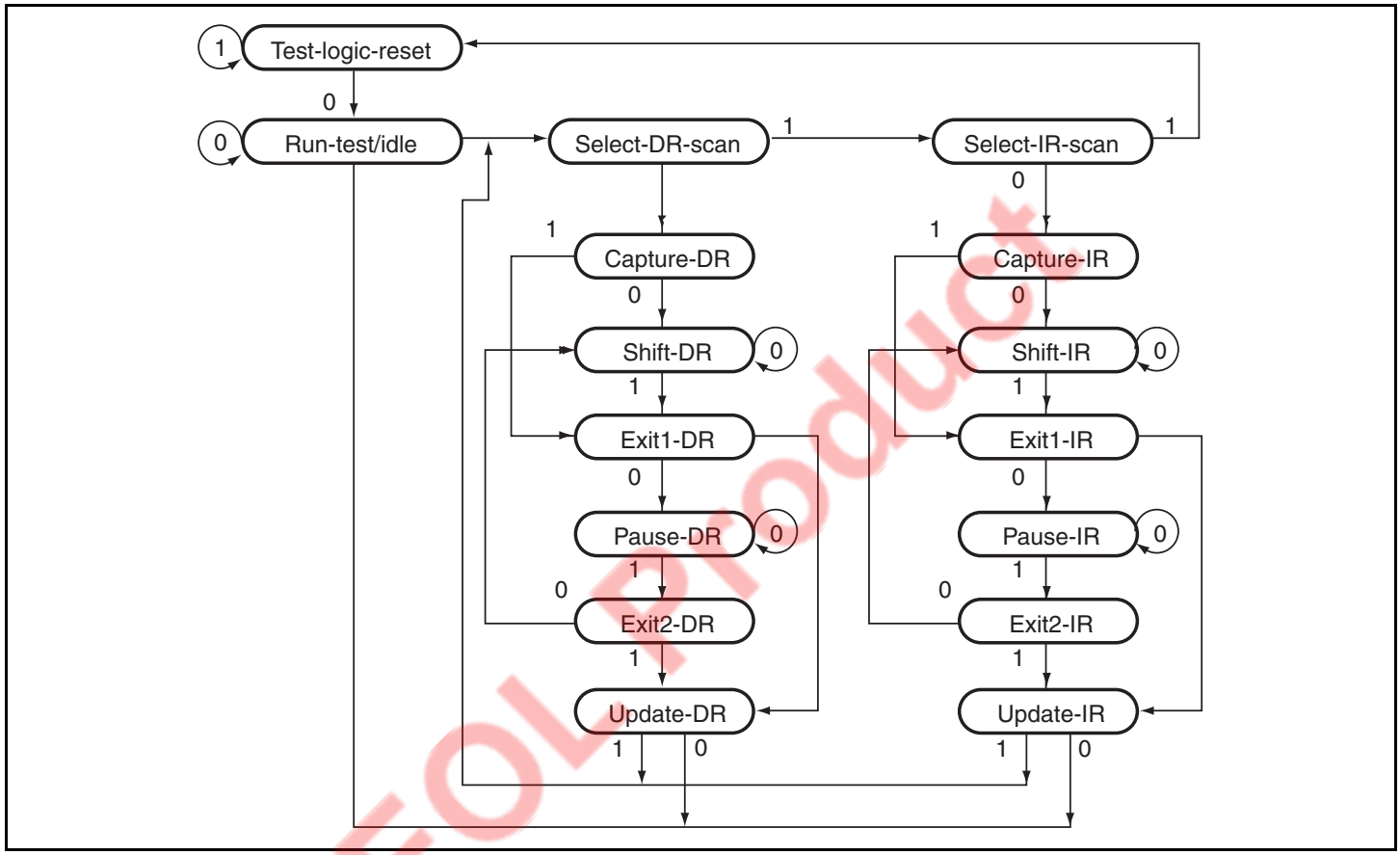


Figure 15.2 TAP Controller State Transitions

Note: The transition condition is the TMS value at the rising edge of TCK. The TDI value is sampled at the rising edge of TCK; shifting occurs at the falling edge of TCK. For details on change timing of the TDO value, see section 15.4.3, TDO Output Timing. The TDO is at high impedance, except with shift-DR and shift-IR states. During the change to $\overline{\text{TRST}}$ = 0, there is a transition to test-logic-reset asynchronously with TCK.

15.4.2 Reset Configuration

Table 15.4 Reset Configuration

$\overline{\text{ASEMD0}}$*1	$\overline{\text{RESETP}}$	$\overline{\text{TRST}}$	Chip State
H	L	L	Normal reset and H-UDI reset
		H	Normal reset
	H	L	H-UDI reset only
		H	Normal operation
L	L	L	Reset hold*2
		H	Normal reset
	H	L	H-UDI reset only
		H	Normal operation

Notes: 1. Performs normal mode and ASE mode settings
$\overline{\text{ASEMD0}}$ = H, normal mode
$\overline{\text{ASEMD0}}$ = L, ASE mode
2. In ASE mode, reset hold is entered if the $\overline{\text{TRST}}$ pin is driven low while the $\overline{\text{RESETP}}$ pin is negated. In this state, the CPU does not start up. When $\overline{\text{TRST}}$ is driven high, H-UDI operation is enabled, but the CPU does not start up. The reset hold state is cancelled by the following:
Another $\overline{\text{RESETP}}$ assert (power-on reset)

15.4.3 TDO Output Timing

The timing of data output from the TDO is switched by the command type set in the SDIR. The timing changes at the TCK falling edge when JTAG commands (EXTEST, CLAMP, HIGHZ, SAMPLE/PRELOAD, IDCODE, and BYPASS) are set. This is a timing of the JTAG standard. When the H-UDI commands (H-UDI reset negate, H-UDI reset assert, and H-UDI interrupt) are set, TDO is output at the TCK rising edge earlier than the JTAG standard by a half cycle.

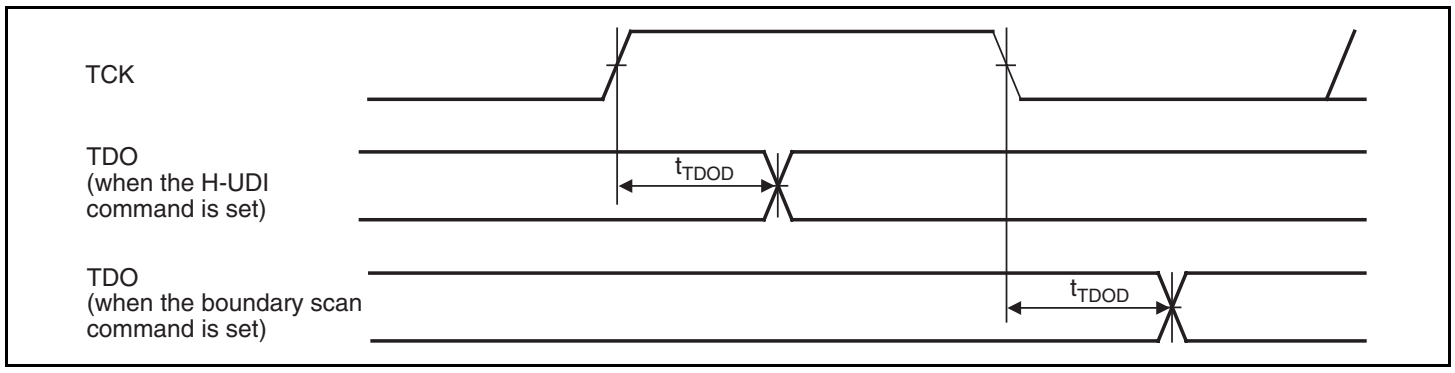


Figure 15.3 H-UDI Data Transfer Timing

15.4.4 H-UDI Reset

An H-UDI reset is executed by inputting an H-UDI reset assert command in SDIR. An H-UDI reset is of the same kind as a power-on reset. An H-UDI reset is released by inputting an H-UDI reset negate command. The required time between the H-UDI reset assert command and H-UDI reset negate command is the same as time for keeping the $\overline{\text{RESETP}}$ pin low to apply a power-on reset.

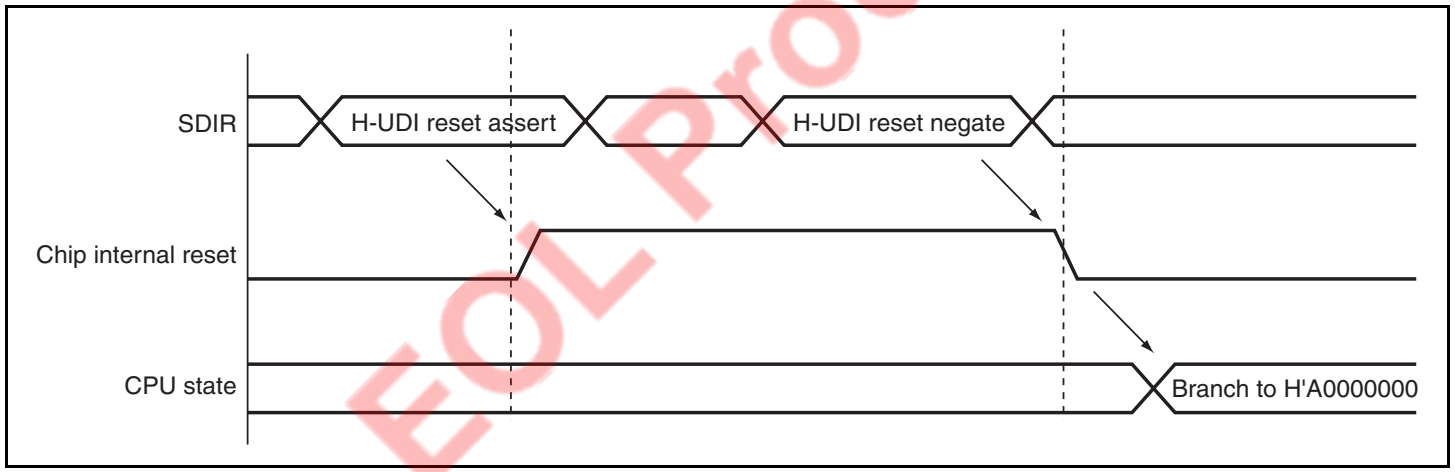


Figure 15.4 H-UDI Reset

15.4.5 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting a command from the H-UDI in the SDIR. An H-UDI interrupt is a general exception/interrupt operation, resulting in a branch to an address based on the VBR value plus offset, and with return by the RTE instruction. This interrupt request has a fixed priority level of 15.

H-UDI interrupts are accepted in sleep mode.

15.5 Boundary Scan

A command can be set in SDIR by the H-UDI to place the H-UDI pins in the boundary scan mode stipulated by JTAG.

15.5.1 Supported Instructions

This LSI supports the three essential instructions defined in the JTAG standard (BYPASS, SAMPLE/PRELOAD, and EXTEST) and three option instructions (IDCODE, CLAMP, and HIGHZ).

BYPASS: The BYPASS instruction is an essential standard instruction that operates the bypass register. This instruction shortens the shift path to speed up serial data transfer involving other chips on the printed circuit board. While this instruction is executing, the test circuit has no effect on the system circuits. The upper four bits of the instruction code are B'1111.

SAMPLE/PRELOAD: The SAMPLE/PRELOAD instruction inputs values from this LSI's internal circuitry to the boundary scan register, outputs values from the scan path, and loads data onto the scan path. When this instruction is executing, this LSI's input pin signals are transmitted directly to the internal circuitry, and internal circuit values are directly output externally from the output pins. This LSI's system circuits are not affected by execution of this instruction. The upper four bits of the instruction code are B'0100.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to the internal circuitry, or a value to be transferred from the internal circuitry to an output pin, is latched into the boundary scan register and read from the scan path. Snapshot latching is performed in synchronization with the rise of TCK in the Capture-DR state. Snapshot latching does not affect normal operation of this LSI.

In a PRELOAD operation, an initial value is set in the parallel output latch of the boundary scan register from the scan path prior to the EXTEST instruction. Without a PRELOAD operation, when the EXTEST instruction was executed an undefined value would be output from the output pin until completion of the initial scan sequence (transfer to the output latch) (with the EXTEST instruction, the parallel output latch value is constantly output to the output pin).

EXTEST: This instruction is provided to test external circuitry when the this LSI is mounted on a printed circuit board. When this instruction is executed, output pins are used to output test data (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed circuit board, and input pins are used to latch test results into the boundary scan register from the printed circuit board. If testing is carried out by using the EXTEST instruction N times, the Nth test data is scanned-in when test data (N-1) is scanned out.

Data loaded into the output pin boundary scan register in the Capture-DR state is not used for external circuit testing (it is replaced by a shift operation).

The upper four bits of the instruction code are B'0000.

IDCODE: A command can be set in SDIR by the H-UDI pins to place the H-UDI pins in the IDCODE mode stipulated by JTAG. When the H-UDI is initialized ($\overline{\text{TRST}}$ is asserted or TAP is in the Test-Logic-Reset state), the IDCODE mode is entered.

CLAMP, HIGHZ: A command can be set in SDIR by the H-UDI pins to place the H-UDI pins in the CLAMP or HIGHZ mode stipulated by JTAG.

15.5.2 Points for Attention

1. Boundary scan mode does not cover clock-related signals (EXTAL, XTAL, CKIO, CKIO2).
2. Boundary scan mode does not cover reset-related signals ($\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$).
3. Boundary scan mode does not cover H-UDI-related signals (TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$).
4. The USB-transceiver-related signals (DP, DM) are beyond the scope of the boundary scan.
5. When the EXTEST, CLAMP, and HIGHZ commands are set, fix the $\overline{\text{RESETP}}$ pin low.
6. When a boundary scan test for other than BYPASS and IDCODE is carried out, fix the $\overline{\text{ASEMD0}}$ pin high.

15.6 Usage Notes

1. An H-UDI command, once set, will not be modified as long as another command is not re-issued from the H-UDI. If the same command is given continuously, the command must be set after a command (BYPASS, etc.) that does not affect chip operations is once set.
2. H-UDI commands are not accepted in standby mode, since operation of the LSI is suspended. To retain the TAP status before and after standby mode, keep TCK high before entering standby mode.
3. The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be used when using an emulator.

Section 16 I²C Bus Interface 2 (IIC2)

The I²C bus interface 2 conforms to and provides a subset of the Philips I²C (Inter-IC) bus interface functions. However, the configuration of the registers that control the I²C bus differs partly from the Philips register configuration.

Figure 16.1 shows a block diagram of the I²C bus interface 2. Figure 16.2 shows an example of I/O pin connections to external circuits.

16.1 Features

- Selection of I²C format or clocked synchronous serial format
- 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.

I²C bus format:

- 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 when the bus drive function is selected.

Clocked synchronous format:

- Four interrupt sources

  Transmit-data-empty, transmit-end, receive-data-full, and overrun error

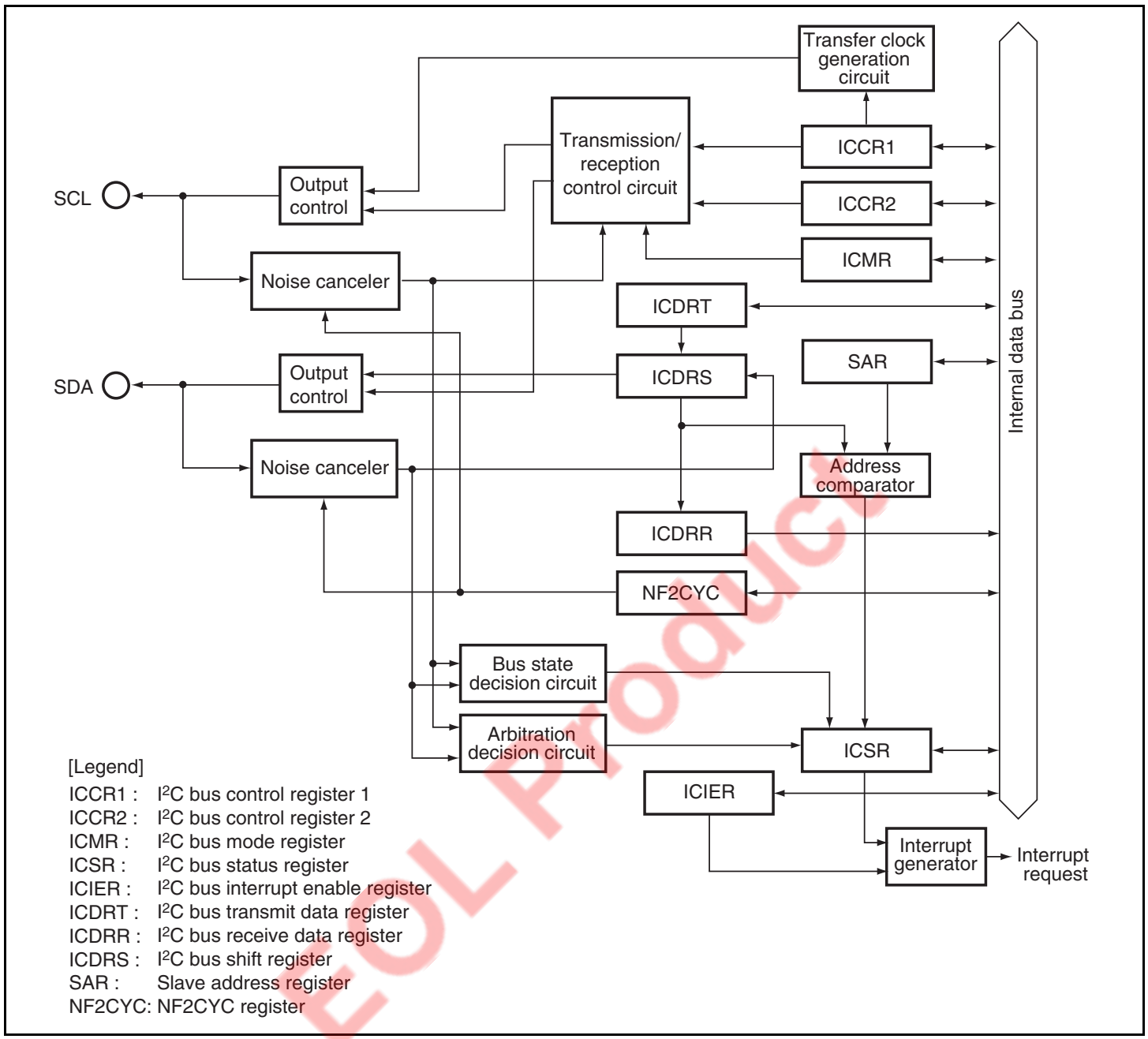
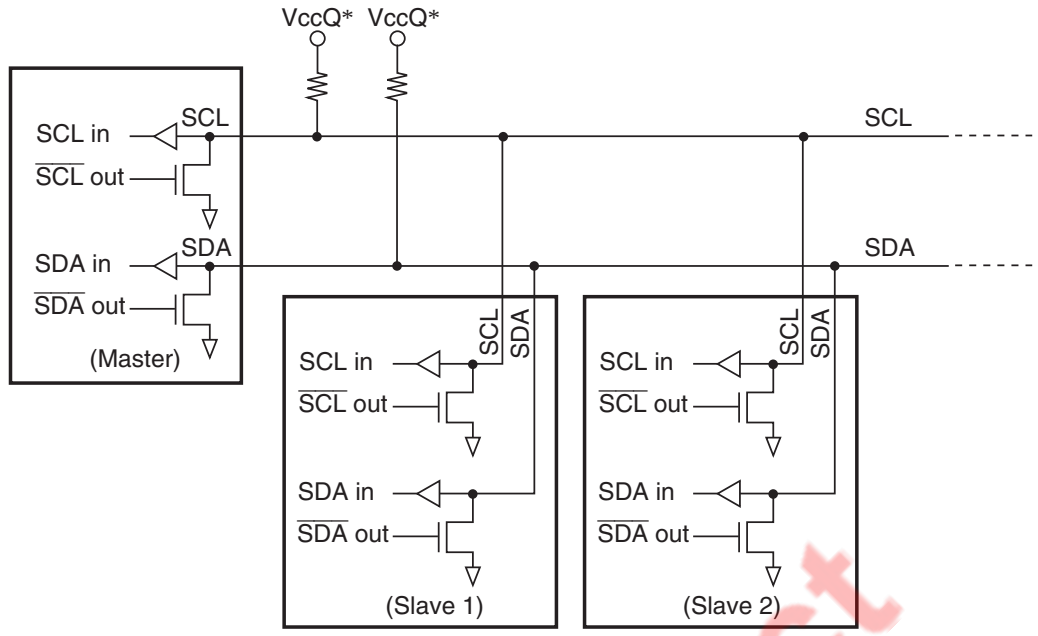


Figure 16.1 Block Diagram of I²C Bus Interface 2



Note: * The I²C bus power supply and this LSI's power supply (VccQ) must be switched ON or OFF simultaneously.

Figure 16.2 External Circuit Connections of I/O Pins

16.2 Input/Output Pins

Table 16.1 shows the pin configuration for the I²C bus interface 2.

Table 16.1 I²C Bus Interface Pin Configuration

Name	Abbreviation	I/O	Function
Serial clock	SCL	I/O	IIC serial clock input/output
Serial data	SDA	I/O	IIC serial data input/output

16.3 Register Descriptions

The I²C bus interface 2 has the following registers:

- I²C bus control register 1 (ICCR1)
- I²C bus control register 2 (ICCR2)
- I²C bus mode register (ICMR)
- I²C bus interrupt enable register (ICIER)
- I²C bus status register (ICSR)
- I²C bus slave address register (SAR)
- I²C bus transmit data register (ICDRT)
- I²C bus receive data register (ICDRR)
- I²C bus shift register (ICDRS)
- NF2CYC register (NF2CYC)

16.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 is an 8-bit readable/writable register that enables or disables the I²C bus interface 2, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

ICCR1 is initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface Enable 0: This module is halted. (SCL and SDA pins are set to port function.) 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

Bit	Bit Name	Initial Value	R/W	Description
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select In master mode with the I²C bus format, when arbitration is lost, 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. When seven bits after the start condition is issued in slave receive mode match the slave address set to SAR and the eighth bit is set to 1, TRS is automatically set to 1. If an overrun error occurs in master receive mode with the clocked synchronous serial format, MST is cleared and the mode changes to slave receive mode. Operating modes are described below according to MST and TRS combination. When clocked synchronous serial format is selected and MST 1, clock is output. 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 These bits are valid only in master mode. These bits should be set according to the necessary transfer rate. For details of transfer rate, see table 16.2.
2	CKS2	0	R/W	
1	CKS1	0	R/W	
0	CKS0	0	R/W	

Table 16.2 Transfer Rate

Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate				
CKS3	CKS2	CKS1	CKS0	Clock	φ=5 MHz	φ=10 MHz	φ=16.5 MHz	φ=30 MHz	φ=33 MHz
0	0	0	0	φ/28	179 kHz	357 kHz	589kHz	1071kHz	1179kHz
			1	φ/40	125 kHz	250 kHz	413kHz	750kHz	825kHz
		1	0	φ/48	104 kHz	208 kHz	344kHz	625kHz	688kHz
			1	φ/64	78.1 kHz	156 kHz	258kHz	469KHz	516kHz
	1	0	0	φ/80	62.5 kHz	125 kHz	206kHz	375kHz	413kHz
			1	φ/100	50.0 kHz	100 kHz	165kHz	300kHz	330kHz
		1	0	φ/112	44.6 kHz	89.3 kHz	147kHz	268kHz	295kHz
			1	φ/128	39.1 kHz	78.1 kHz	129kHz	234kHz	258kHz
1	0	0	0	φ/56	89.3 kHz	179 kHz	295kHz	536kHz	589kHz
			1	φ/80	62.5 kHz	125 kHz	206kHz	375kHz	413kHz
		1	0	φ/96	52.1 kHz	104 kHz	172kHz	313kHz	344kHz
			1	φ/128	39.1 kHz	78.1 kHz	129kHz	234kHz	258kHz
	1	0	0	φ/160	31.3 kHz	62.5 kHz	103kHz	188kHz	206kHz
			1	φ/200	25.0 kHz	50.0 kHz	82.5kHz	150kHz	165kHz
		1	0	φ/224	22.3 kHz	44.6 kHz	73.7kHz	134kHz	147kHz
			1	φ/256	19.5 kHz	39.1 kHz	64.5kHz	117kHz	129kHz

Note: Set the value that satisfies the external specifications.

16.3.2 I²C Bus Control Register 2 (ICCR2)

ICCR2 is an 8-bit readable/writable register that issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls reset in the control part of the I²C bus interface 2.

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/stop conditions in master mode. With the clocked synchronous serial format, this bit has no meaning. With the I²C bus format, 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 in BBSY and 0 in SCP to issue a stop condition.
6	SCP	1	W	Start/Stop Issue Condition Disable 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.
5	SDAO	1	R/W	SDA Output Value Control This bit is used with SDAOP when modifying output level of SDA. This bit should not be manipulated during transfer. 0: When reading, SDA pin outputs low. When writing, SDA pin is changed to output low. 1: When reading, SDA pin outputs high. When writing, SDA pin is changed to output Hi-Z (outputs high by external pull-up resistance).

Bit	Bit Name	Initial Value	R/W	Description
4	SDAOP	1	R/W	SDAO Write Protect This bit controls change of output level of the SDA pin by modifying the SDAO bit. To change the output level, clear SDAO and SDAOP to 0 or set SDAO to 1 and clear SDAOP to 0. This bit is always read as 1.
3	SCLO	1	R	This bit monitors SCL output level. When SCLO is 1, SCL pin outputs high. When SCLO is 0, SCL pin outputs low.
2	—	1	—	Reserved This bit is always read as 1, and cannot be modified.
1	IICRST	0	R/W	IIC Control Part Reset This bit resets the control part except for I²C registers. If this bit is set to 1 when hang-up occurs 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, and cannot be modified.

16.3.3 I²C Bus Mode Register (ICMR)

ICMR is an 8-bit readable/writable register that selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the transfer bit count.

ICMR is initialized to H'38 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I²C bus format is used.
6	—	0	—	Reserved The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
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. In clock synchronous serial mode, BC should not be modified. 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.
2	BC2	0	R/W	Bit Counter 2 to 0 These bits specify the number of bits to be transferred next. When read, the remaining number of transfer bits is indicated. With the I²C bus format, the data is transferred with one addition acknowledge bit. should be made between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL pin is low. The value returns to 000 at the end of a data transfer, including the acknowledge bit. These bits are cleared by a power-on reset and in standby mode. These bits are also cleared by setting IICRST of ICCR2 to 1. With the clock synchronous serial format, these bits should not be modified. I²C Bus Format / Clock Synchronous Serial Format 000: 9 bits / 000: 8 bits 001: 2 bits / 001: 1 bit 010: 3 bits / 010: 2 bits 011: 4 bits / 011: 3 bits 100: 5 bits / 100: 4 bits 101: 6 bits / 101: 5 bits 110: 7 bits / 110: 6 bits 111: 8 bits / 111: 7 bits
1	BC1	0	R/W	
0	BC0	0	R/W	

16.3.4 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 received. ICIER is initialized to H'00 by a power-on reset.

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 receive 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) are disabled. 1: Receive data full interrupt request (RXI) are enabled.
4	NAKIE	0	R/W	NACK Receive Interrupt Enable This bit enables or disables the NACK detection interrupt request (NAKI) when the NACKF or AL/OVE bit in ICSR is set. NAKI can be canceled by clearing the NACKF, AL/OVE, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is disabled. 1: NACK receive interrupt request (NAKI) is enabled.

Bit	Bit Name	Initial Value	R/W	Description
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable This bit enables or disables the stop condition (STPI) when the STOP bit in ICSR is set . 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 Judgment Select 0: The value of the receive acknowledge bit is ignored, and continuous transfer is performed. 1: If the receive acknowledge bit is 1, continuous transfer is halted.
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. This bit can be canceled by clearing the BBSY bit in ICCR2 to 1. 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.

16.3.5 I²C Bus Status Register (ICSR)

ICSR is an 8-bit readable/writable register that confirms interrupt request flags and their status. ICSR is initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Setting conditions] • When data is transferred from ICDRT to ICDRS and ICDRT becomes empty • When TRS is set • When the start condition (including retransmission) is issued • When slave mode is changed from receive mode to transmit mode [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When data is written to ICDRT
6	TEND	0	R/W	Transmit End [Setting conditions] • When the ninth clock of SCL rises with the I²C bus format while the TDRE flag is 1 • When the final bit of transmit frame is sent with the clock synchronous serial format [Clearing conditions] • When 0 is written in TEND after reading TEND = 1 • When data is written to ICDRT with an instruction
5	RDRF	0	R/W	Receive Data Register Full [Setting condition] • When a receive data is transferred from ICDRS to ICDRR [Clearing conditions] • When 0 is written in RDRF after reading RDRF = 1 • When ICDRR is read with an instruction

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] • When a stop condition is detected after frame transfer [Clearing condition] • When 0 is written in STOP after reading STOP = 1
2	AL/OVE	0	R/W	Arbitration Lost Flag/Overrun Error Flag This flag indicates that arbitration was lost in master mode with the I²C bus format and that the final bit has been received while RDRF = 1 with the clocked synchronous format. 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 • When the final bit is received with the clocked synchronous format while RDRF = 1 [Clearing condition] • When 0 is written in AL/OVE after reading AL/OVE = 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 conditions] • 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 I²C bus format slave receive mode. [Setting condition] • When the general call address is detected in slave receive mode [Clearing condition] • When 0 is written in ADZ after reading ADZ=1

16.3.6 Slave Address Register (SAR)

SAR is an 8-bit readable/writable register that selects the communications format and sets the slave address. In slave mode with the I²C bus format, if the upper seven bits of SAR match the upper seven bits of the first frame received after a start condition, this module operates as the slave device. SAR is initialized to H'00 by a power-on reset.

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 form the addresses of other slave devices connected to the I²C bus.
0	FS	0	R/W	Format Select 0: I²C bus format is selected 1: Clocked synchronous serial format is selected

16.3.7 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 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.

16.3.8 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 receive data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot write to this register.

16.3.9 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 directly from the CPU.

16.3.10 NF2CYC Register (NF2CYC)

NF2CYC is an 8-bit readable/writable register that selects the range of the noise filtering for the SCL and SDA pins. For details of the noise filter, see section 16.4.7, Noise Filter.

NF2CYC is initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0.
0	NF2CYC	0	R/W	Noise Filtering Range Select 0: The noise less than one cycle of the peripheral clock can be filtered out 1: The noise less than two cycles of the peripheral clock can be filtered out

16.4 Operation

The I²C bus interface can communicate either in I²C bus mode or clocked synchronous serial mode by setting FS in SAR.

16.4.1 I²C Bus Format

Figure 16.3 shows the I²C bus formats. Figure 16.4 shows the I²C bus timing. The first frame following a start condition always consists of eight bits.

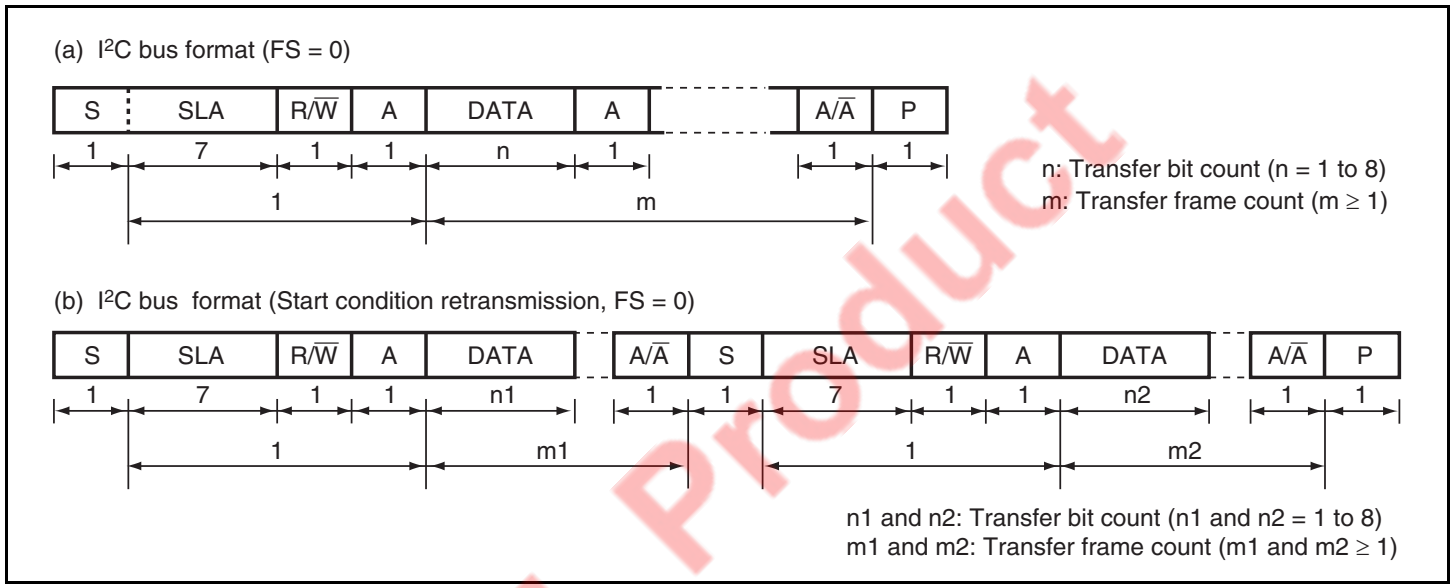


Figure 16.3 I²C Bus Formats

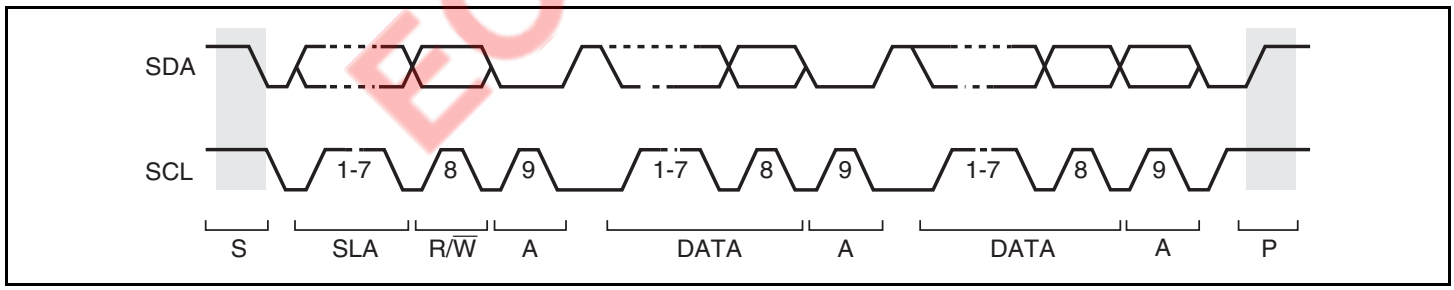


Figure 16.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/W: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives SDA to low.

DATA: Transfer data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

16.4.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For master transmit mode operation timing, refer to figures 16.5 and 16.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS3 to CKS0 in ICCR1 to 1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is free. Set the MST and TRS bits in ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP. (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/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cleared to 0, and 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 ninth 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. 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. 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.
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.

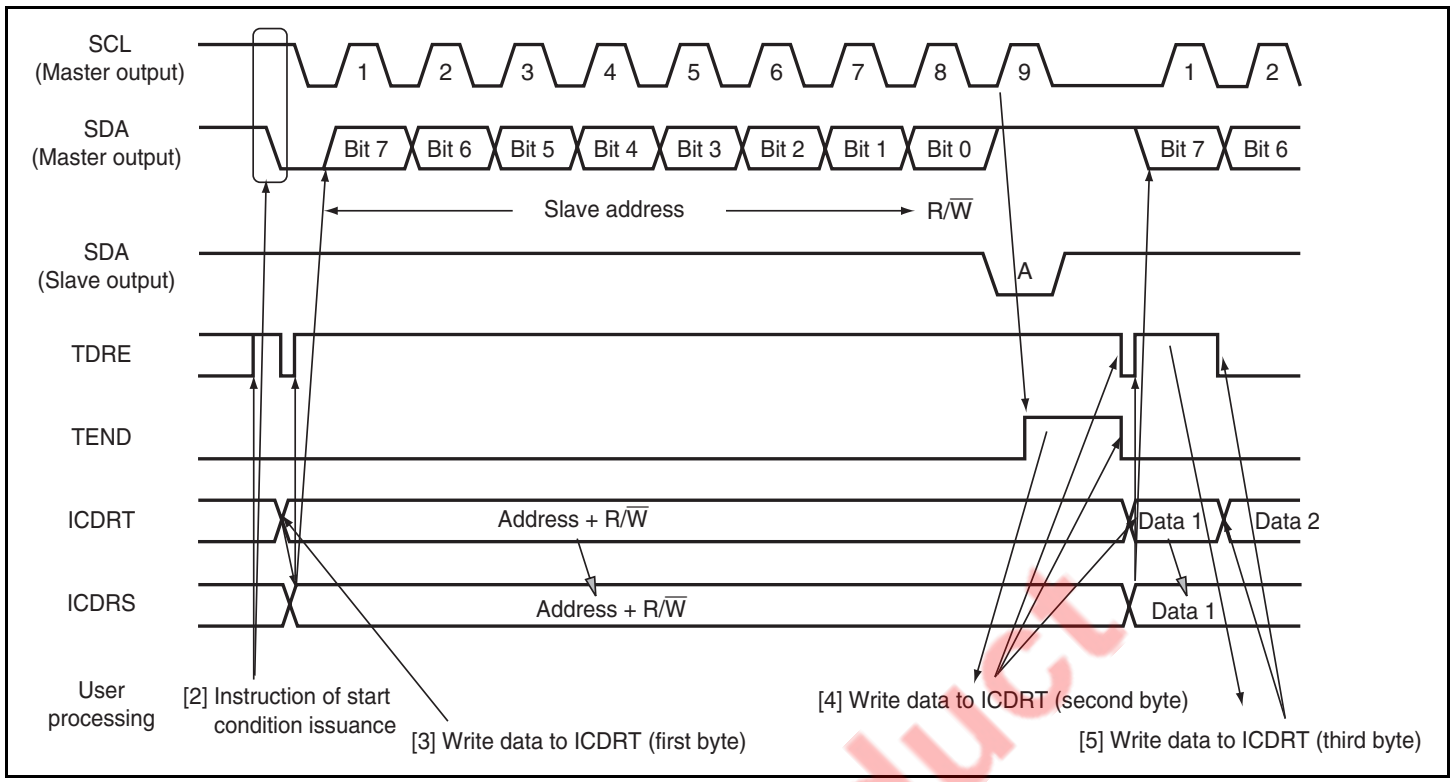


Figure 16.5 Master Transmit Mode Operation Timing (1)

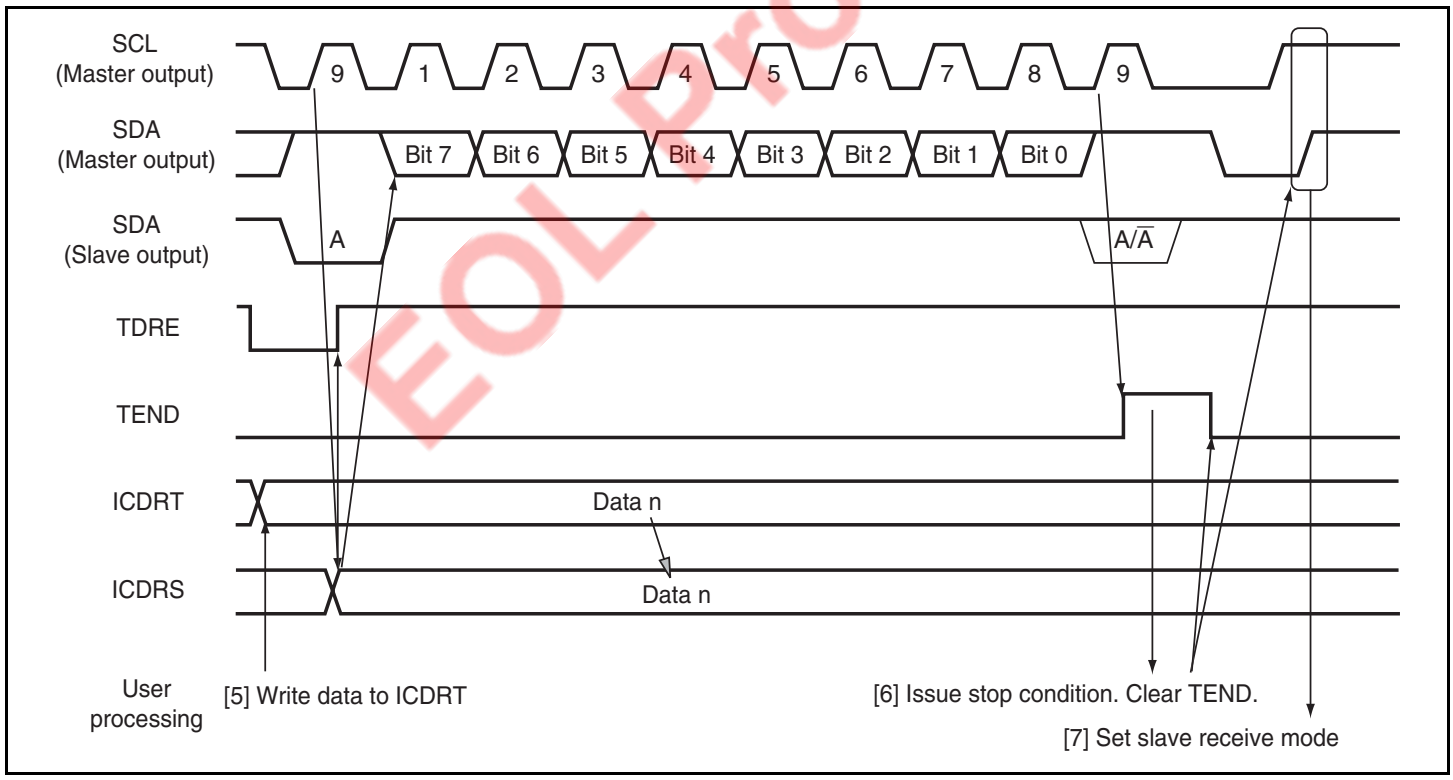


Figure 16.6 Master Transmit Mode Operation Timing (2)

16.4.3 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. For master receive mode operation timing, refer to figures 16.7 and 16.8. 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 ICCR1 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 ninth 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 ninth receive clock pulse. At this time, the receive data is read by reading ICDRR, and RDRF is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. If eighth 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 ninth receive clock pulse, issue the stage condition.
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to the slave receive mode.

Note: Set the RCVD bit in ICCR1 to dummy-read ICDRR to receive only one byte.

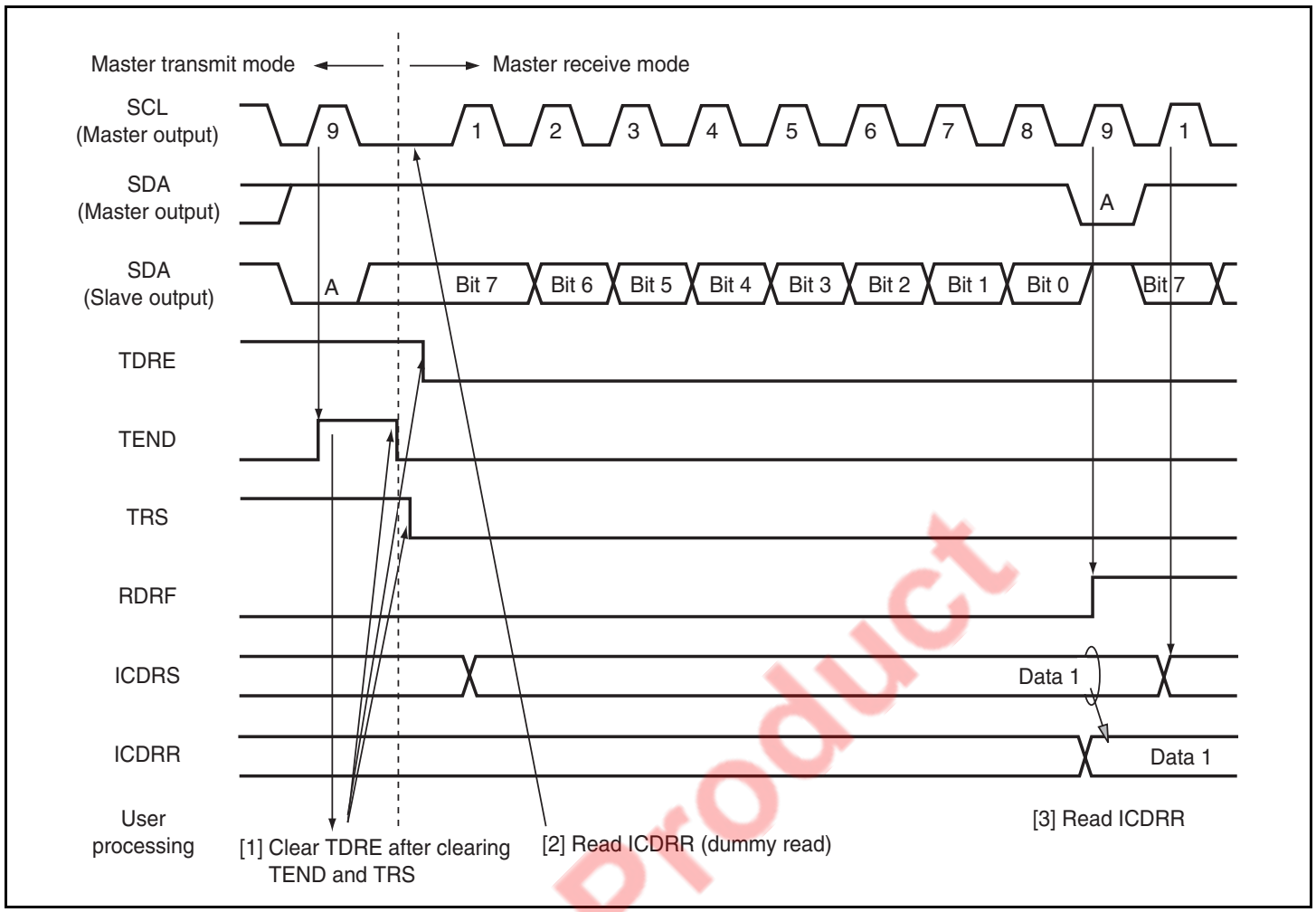


Figure 16.7 Master Receive Mode Operation Timing (1)

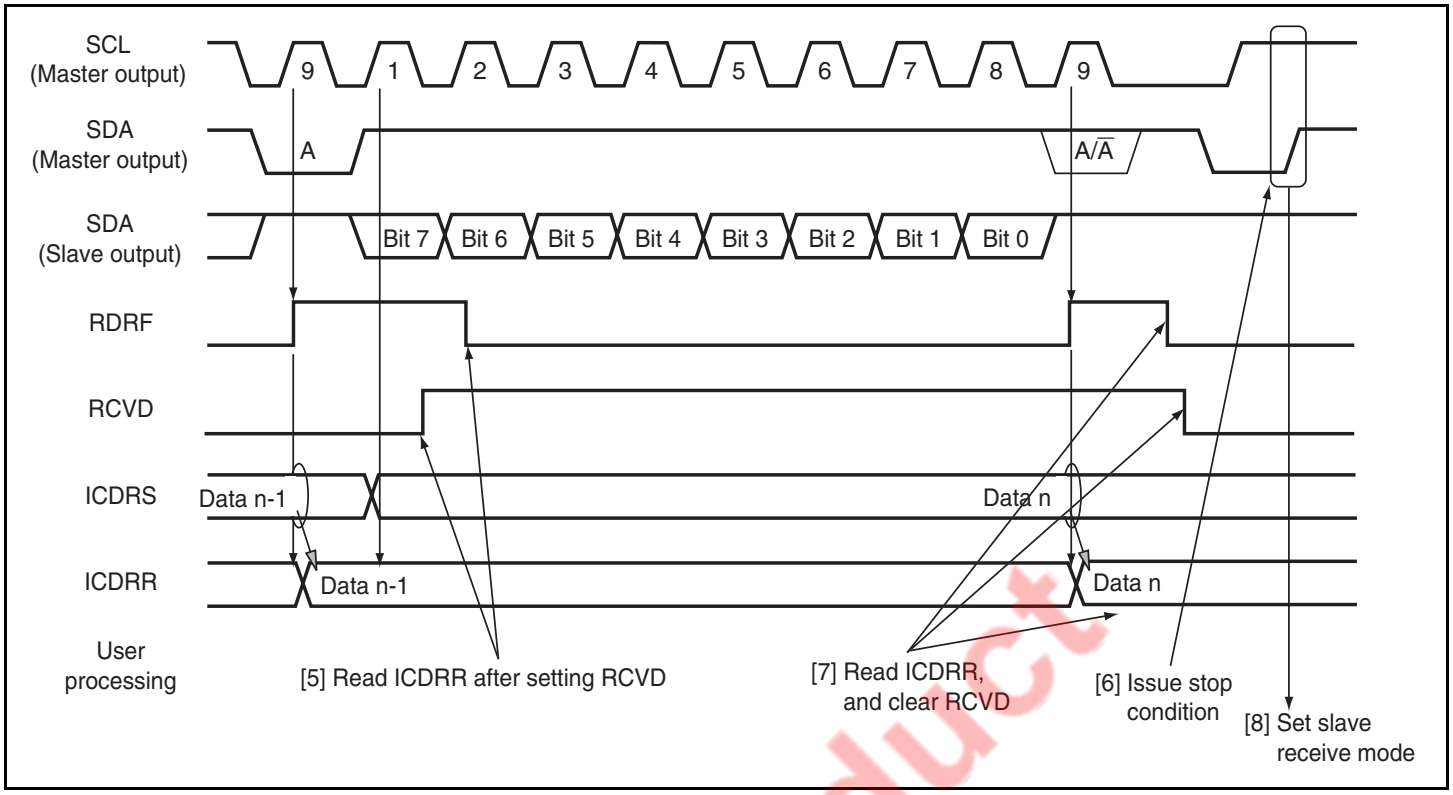


Figure 16.8 Master Receive Mode Operation Timing (2)

16.4.4 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. For slave transmit mode operation timing, refer to figures 16.9 and 16.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS3 to CKS0 in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 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 ninth clock pulse. At this time, if the eighth bit data (R/$\overline{W}$) is 1, the TRS and ICSR bits in ICCR1 are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by 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.

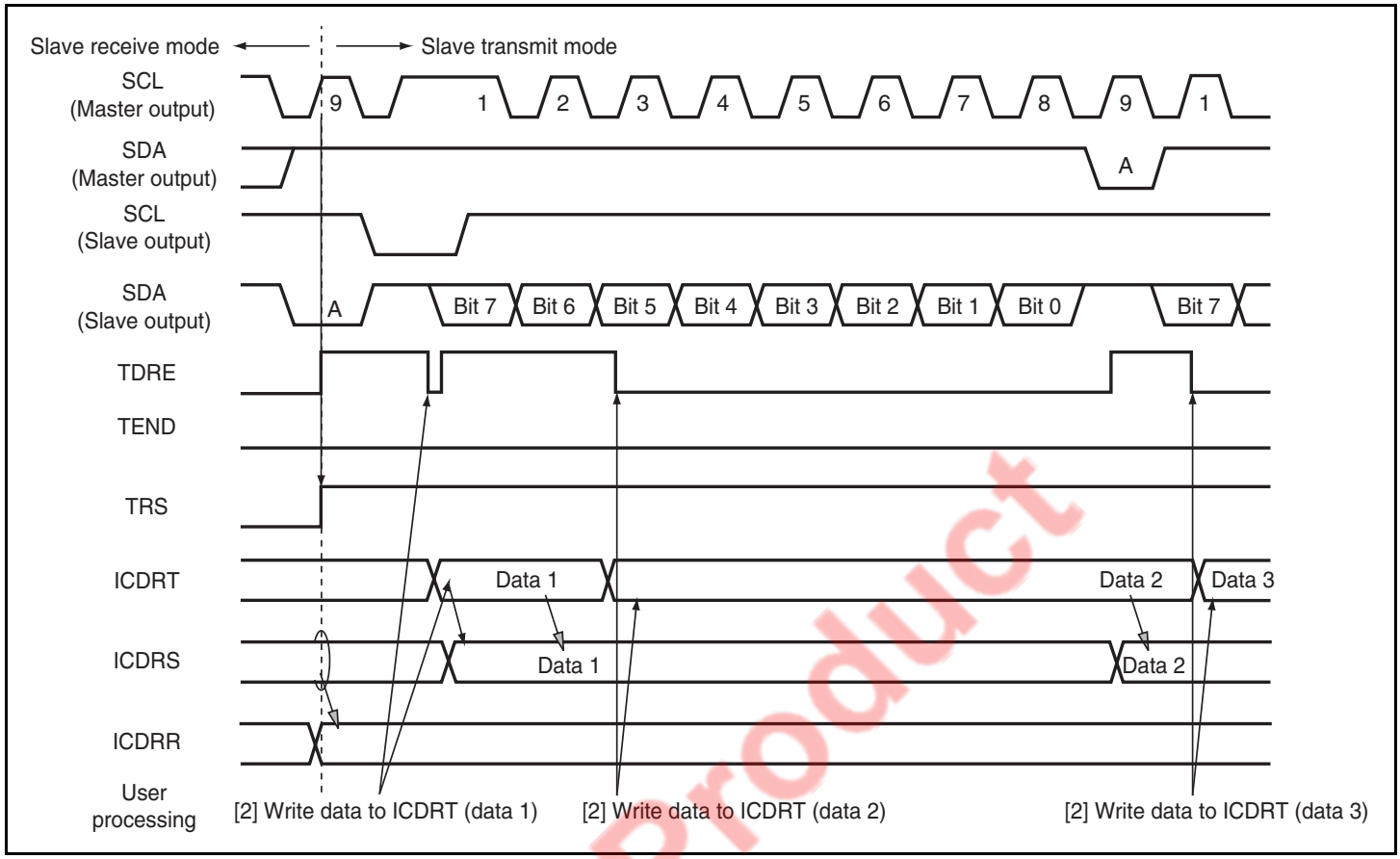


Figure 16.9 Slave Transmit Mode Operation Timing (1)

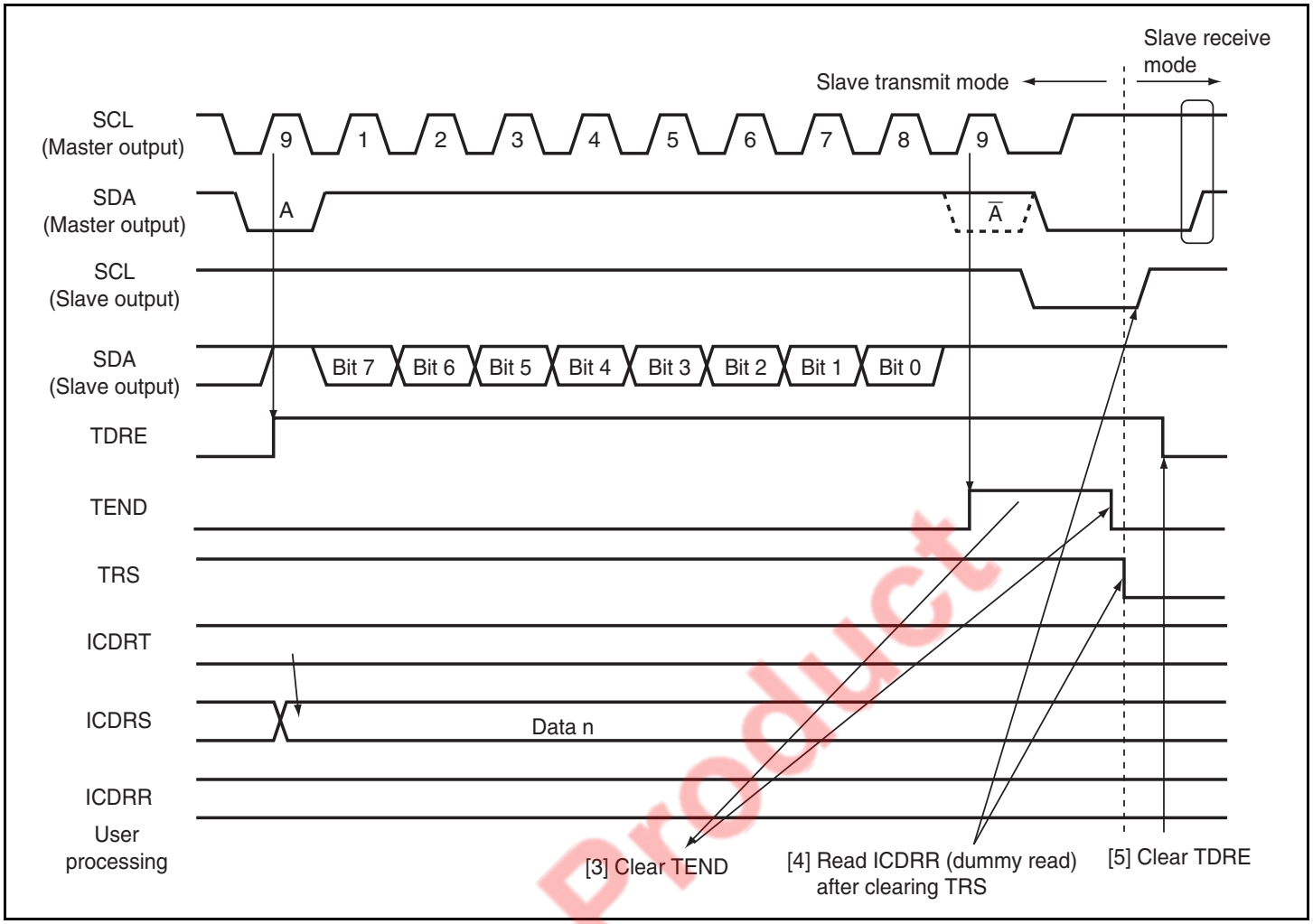


Figure 16.10 Slave Transmit Mode Operation Timing (2)

16.4.5 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. For slave receive mode operation timing, refer to figures 16.11 and 16.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and bits CKS3 to CKS0 in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 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 ninth clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (Since the read data show the slave address and R/$\overline{W}$, it is not used.)
3. Read ICDRR every time RDRF is set. If eighth 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.

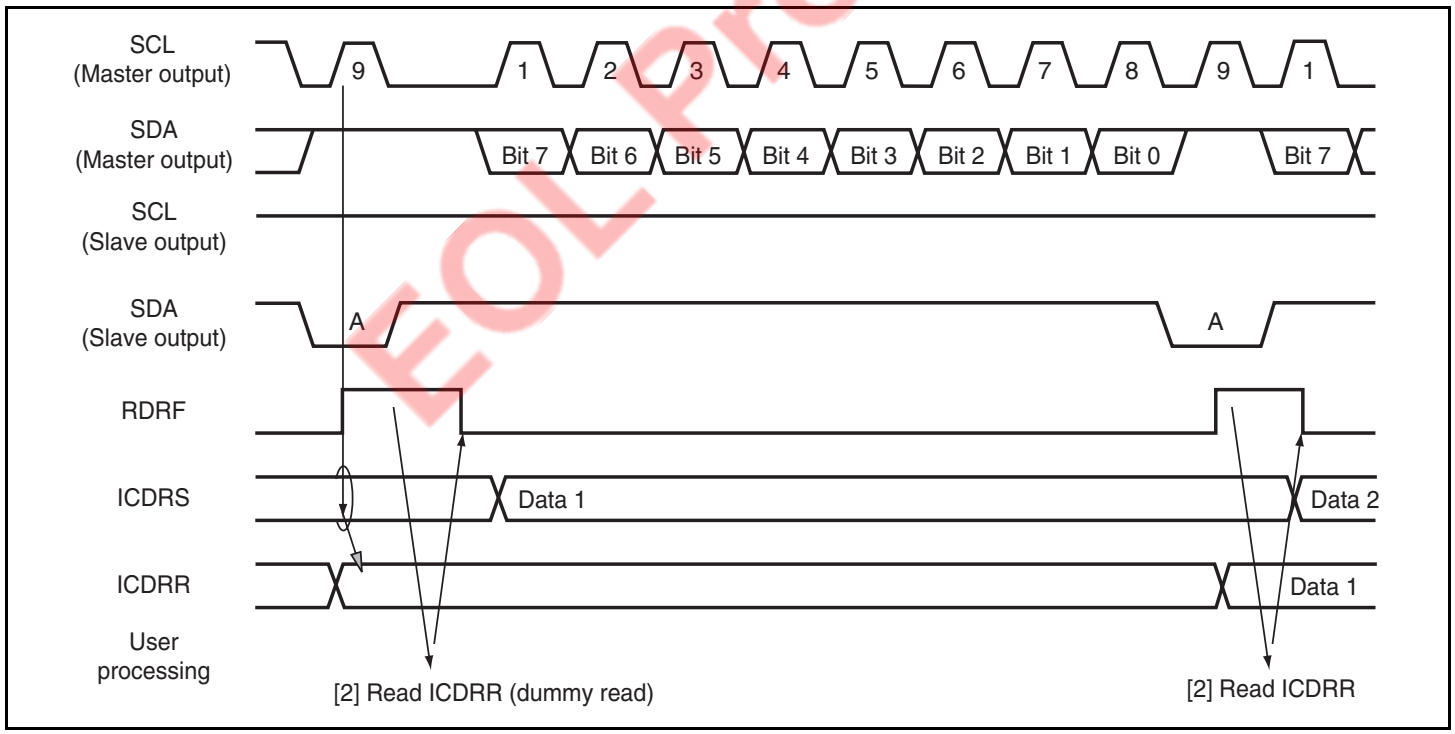


Figure 16.11 Slave Receive Mode Operation Timing (1)

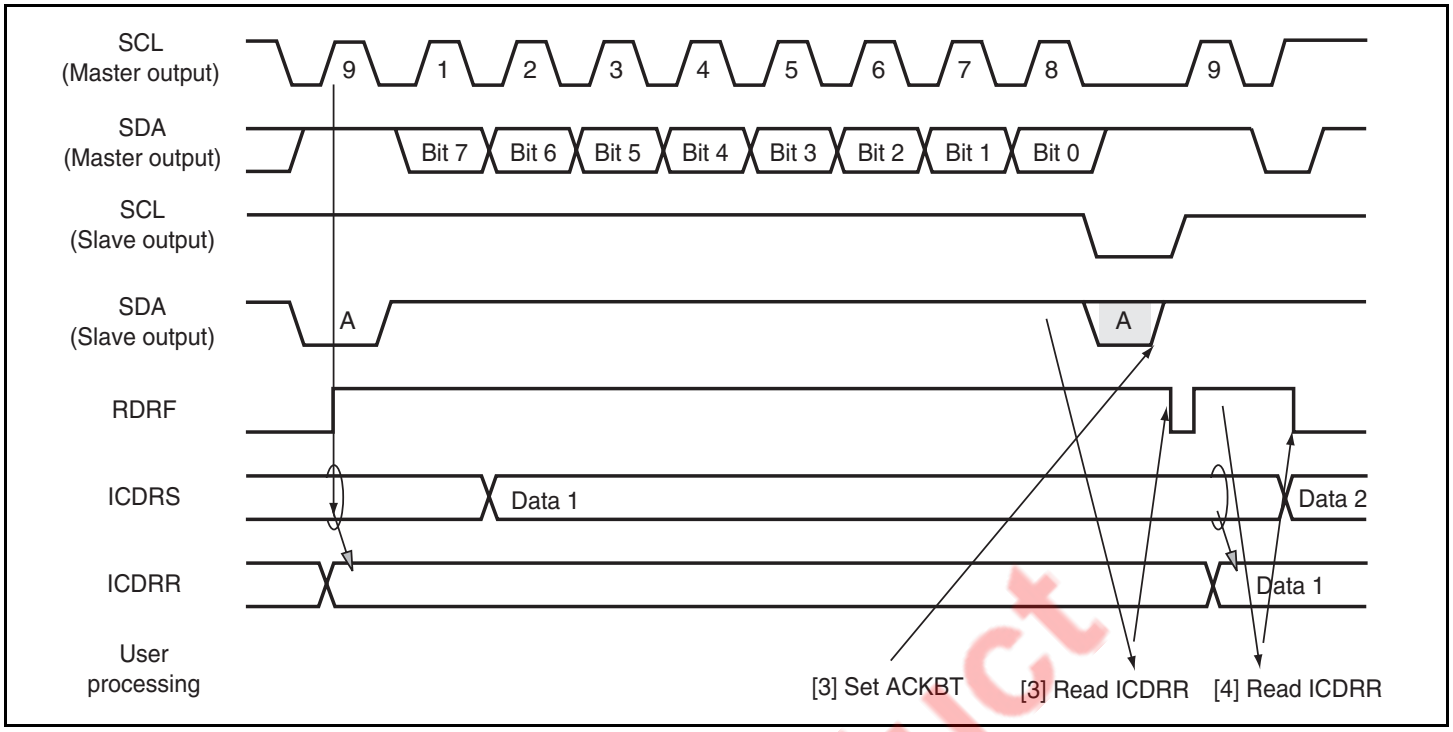


Figure 16.12 Slave Receive Mode Operation Timing (2)

16.4.6 Clocked Synchronous Serial Format

This module can be operated with the clocked synchronous serial format, by setting the FS bit in SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selected. When MST is 0, the external clock input is selected.

Data Transfer Format:

Figure 16.13 shows the clocked synchronous serial transfer format.

The transfer data is output from the rise to the fall of the SCL clock, and the data at the rising edge of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in either the MSB first or LSB first. The output level of SDA can be changed during the transfer wait, by the SDAO bit in ICCR2.

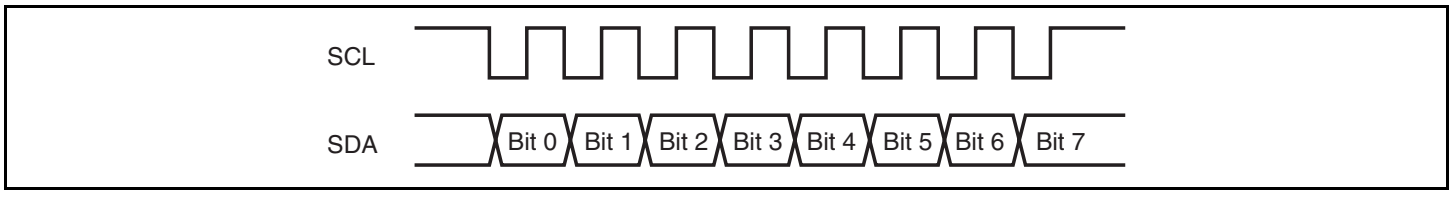


Figure 16.13 Clocked Synchronous Serial Transfer Format

Transmit Operation:

In transmit mode, transmit data is output from SDA, in synchronization with the fall of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For transmit mode operation timing, refer to figure 16.14. The transmission procedure and operations in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. Set the TRS bit in ICCR1 to select the transmit mode. Then, TDRE in ICSR is set.
3. Confirm that TDRE has been set. Then, write the transmit data to ICDRT. The data is transferred from ICDRT to ICDRS, and TDRE is set automatically. The continuous transmission is performed by writing data to ICDRT every time TDRE is set. When changing from transmit mode to receive mode, clear TRS while TDRE is 1.

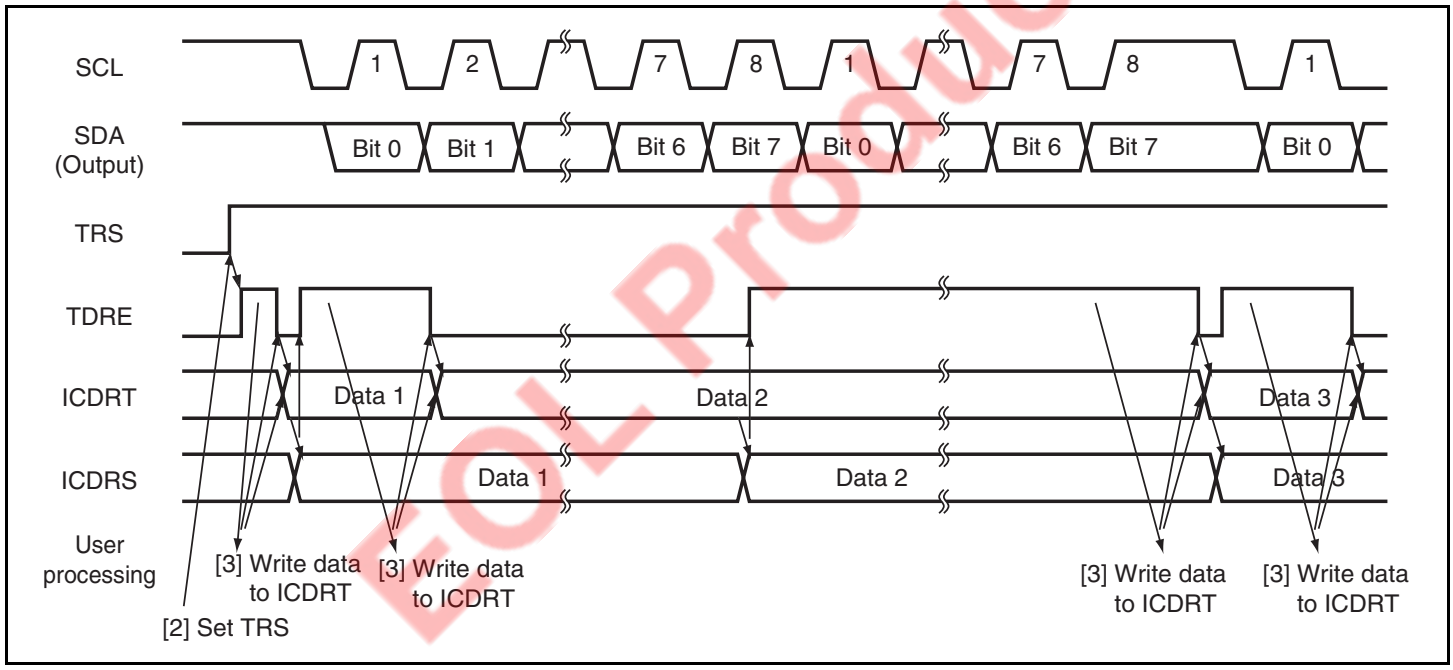


Figure 16.14 Transmit Mode Operation Timing

Receive Operation:

In receive mode, data is latched at the rise of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For receive mode operation timing, refer to figure 16.15. The reception procedure and operations in receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. When the transfer clock is output, set MST to 1 to start outputting the receive clock.
3. When the receive operation is completed, data is transferred from ICDRS to ICDRR and RDRF in ICSR is set. When MST = 1, the next byte can be received, so the clock is continually output. The continuous reception is performed by reading ICDRR every time RDRF is set. When the eighth clock is raised while RDRF is 1, the overrun is detected and AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDRR.
4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then, SCL is fixed high after receiving the next byte data.

Notes: Follow the steps below to receive only one byte with MST=1 specified. See figure 16.16 for the operation timing.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS3 to CKS0 in ICCR1. (Initial setting)
2. Set MST=1 while the RCVD bit in ICCR1 is 0. This causes the receive clock to be output.
3. Check if the BC2 bit in ICMR is set to 1 and then set the RCVD bit in ICCR1 to 1. This causes the SCL to be fixed to the high level after outputting one byte of the receive clock.

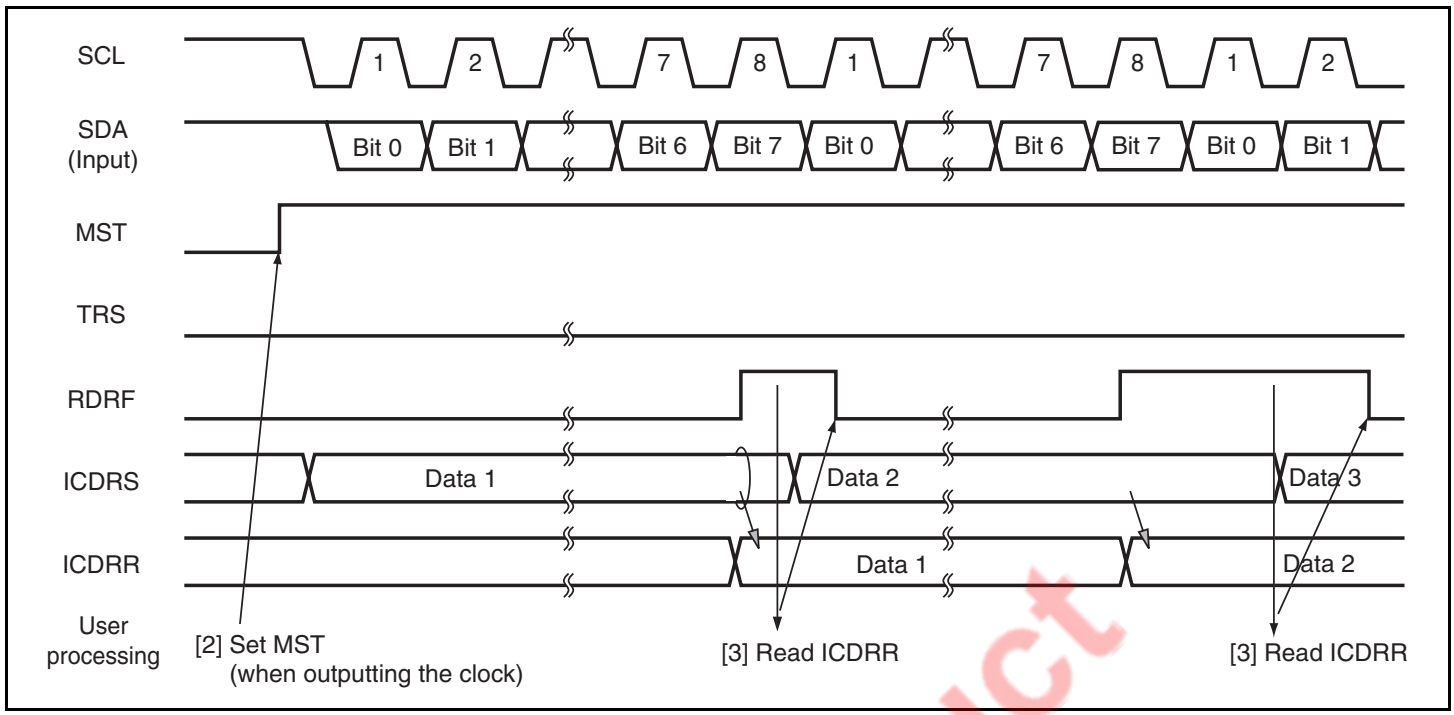


Figure 16.15 Receive Mode Operation Timing

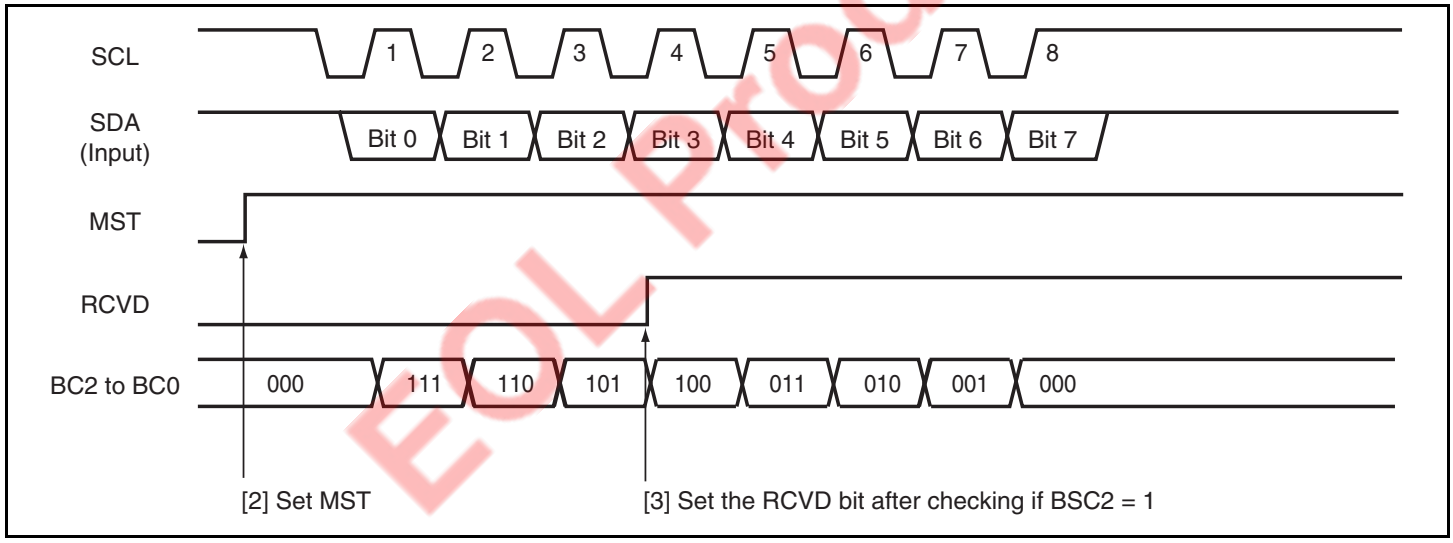


Figure 16.16 Operation Timing For Receiving One Byte

16.4.7 Noise Filter

The logic levels at the SCL and SDA pins are routed through noise filters before being latched internally. Figure 16.17 shows a block diagram of the noise filter circuit.

The noise filter consists of three cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the system clock. When NF2CYC is set to 0, this signal is not passed forward to the next circuit unless the outputs of both latches agree. When NF2CYC is set to 1, this signal is not passed forward to the next circuit unless the outputs of three latches agree. If they do not agree, the previous value is held.

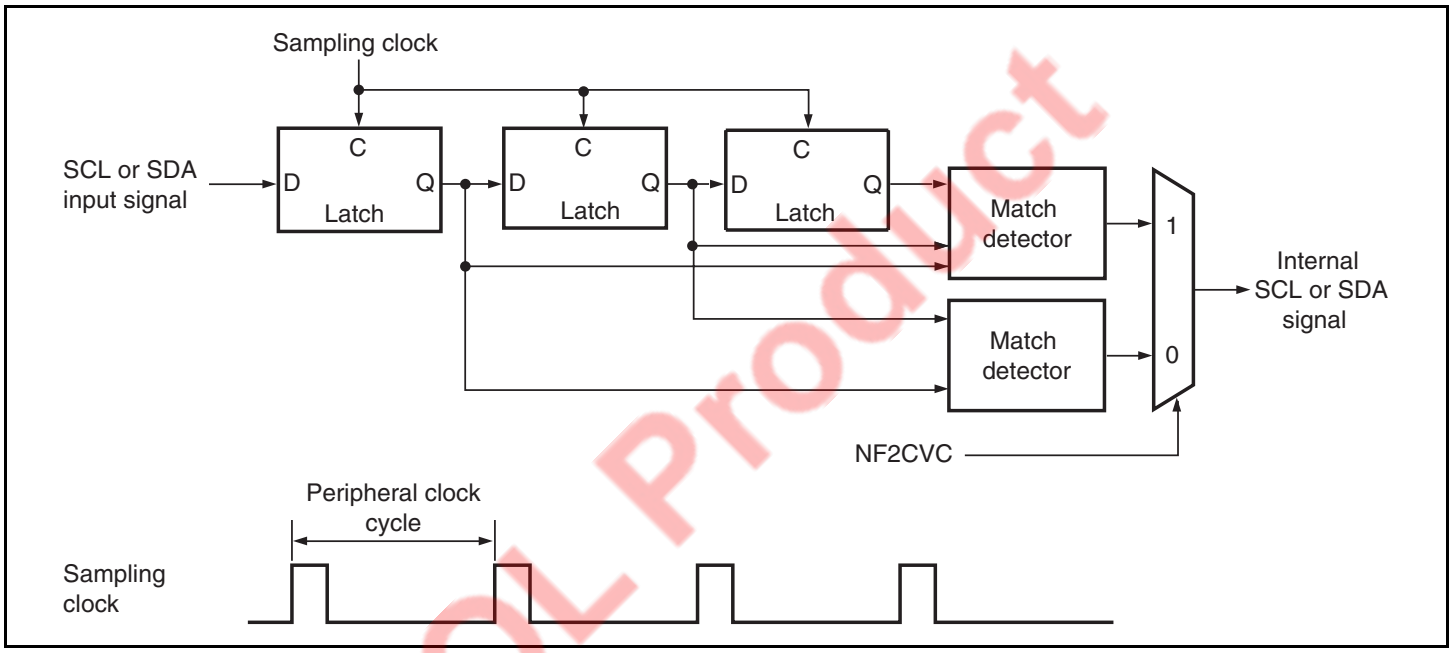


Figure 16.17 Block Diagram of Noise Filter

16.4.8 Example of Use

Flowcharts in respective modes that use the I²C bus interface are shown in figures 16.18 to 16.21.

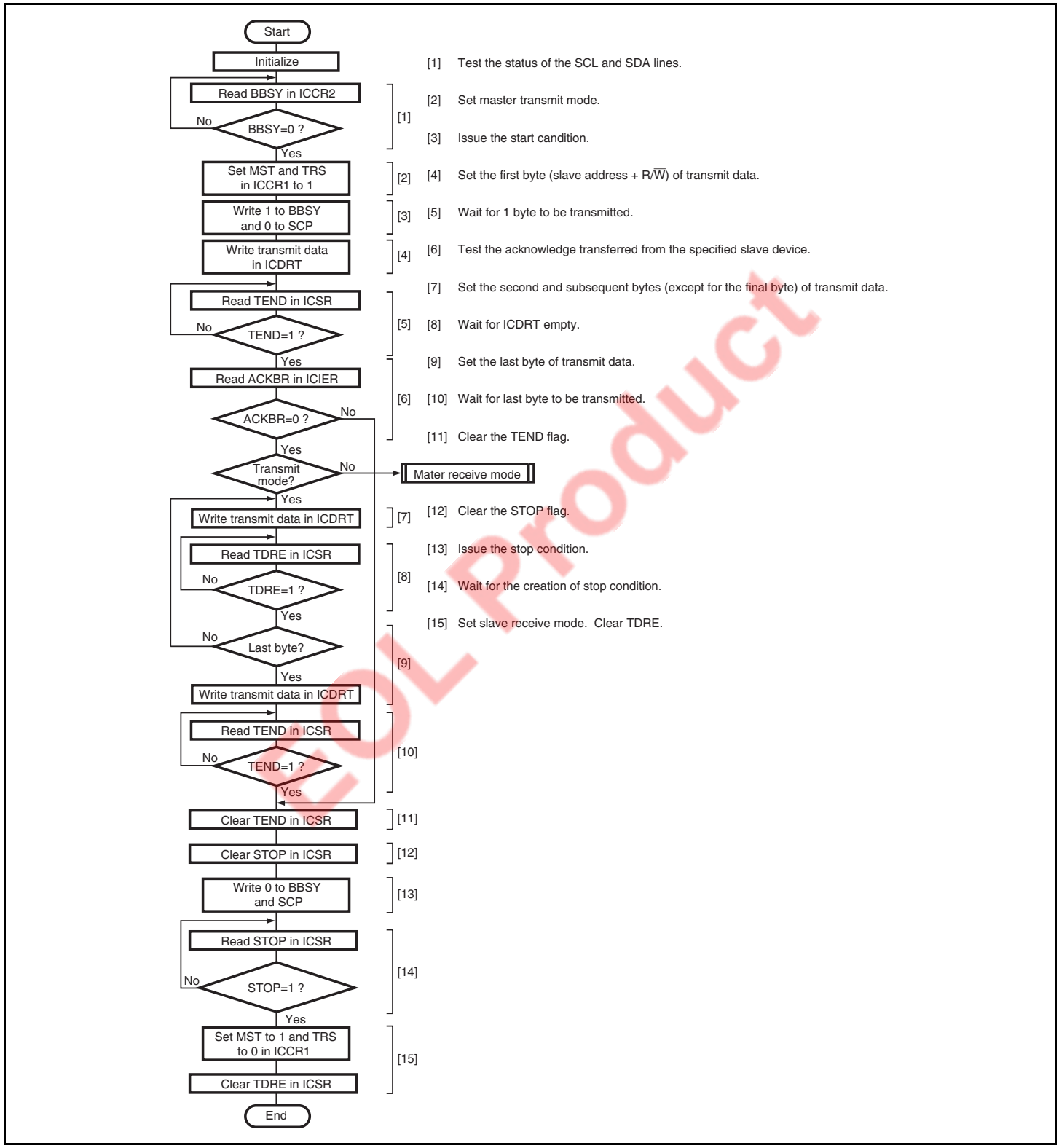
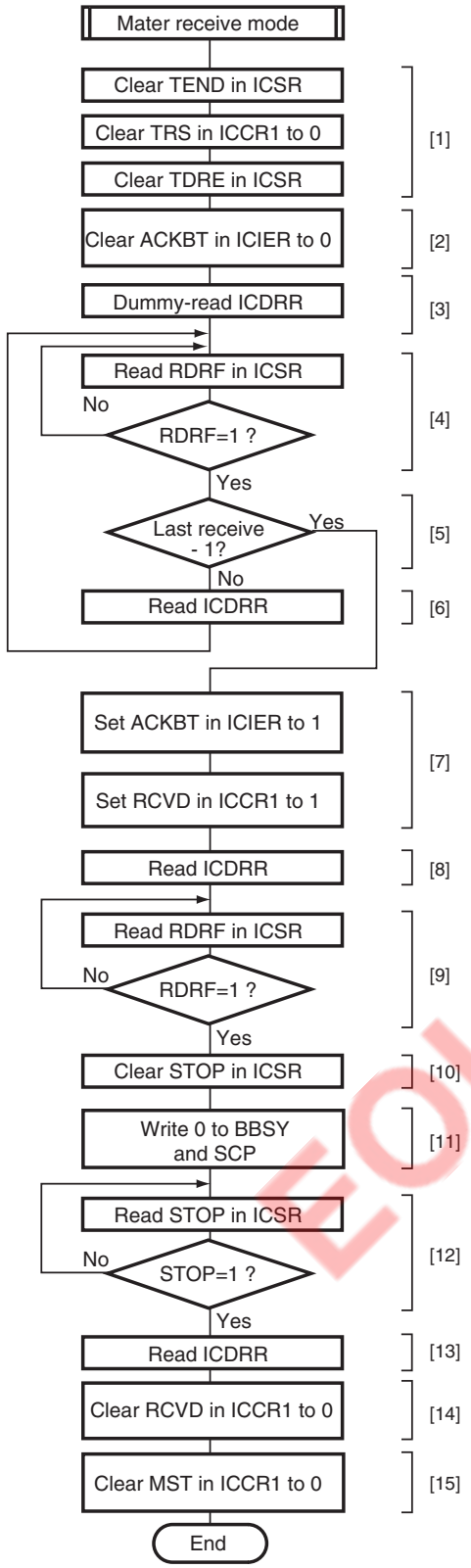


Figure 16.18 Sample Flowchart for Master Transmit Mode



[1] Clear TEND, select master receive mode, and then clear TDRE.

[2] Set acknowledge to the transmit device.

[3] Dummy-read ICDDR.

[4] Wait for 1 byte to be received

[5] Check whether it is the (last receive - 1).

[6] Read the receive data last.

[7] Set acknowledge of the final byte. Disable continuous reception (RCVD = 1).

[8] Read the (final byte - 1) of received data.

[9] Wait for the last byte to be receive.

[10] Clear the STOP flag.

[11] Issue the stop condition.

[12] Wait for the creation of stop condition.

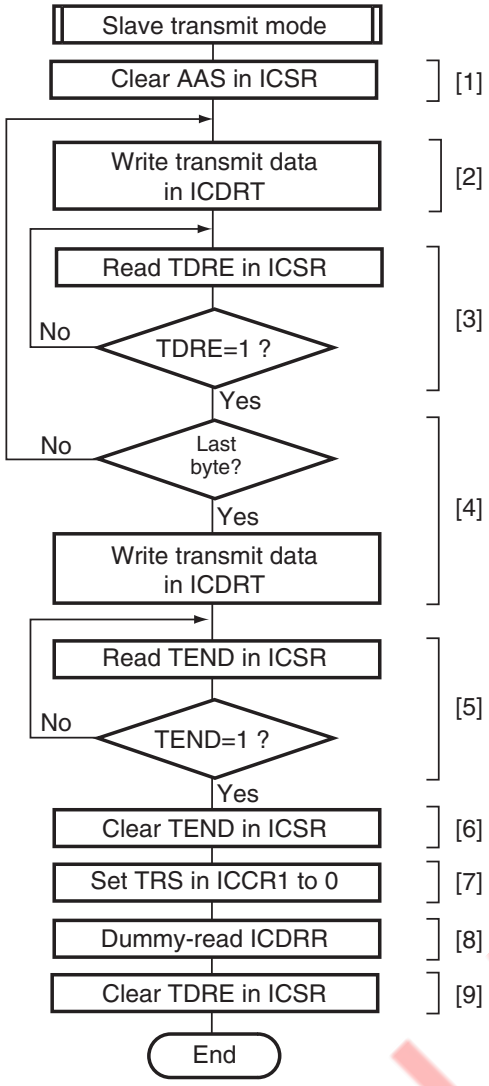
[13] Read the last byte of receive data.

[14] Clear RCVD.

[15] Set slave receive mode.

Note: When the size of receive data is only one byte in reception, steps [2] to [6] are skipped after step [1], before jumping to step [7]. The step [8] is dummy-read in ICDRR.
However, when the size of receive data is two bytes and more, steps [2] to [6] are not skipped after step [1].

Figure 16.19 Sample Flowchart for Master Receive Mode



[1] Clear the AAS flag.

[2] Set transmit data for ICDRT (except for the last byte).

[3] Wait for ICDRT empty.

[4] Set the last byte of transmit data.

[5] Wait for the last byte to be transmitted.

[6] Clear the TEND flag .

[7] Set slave receive mode.

[8] Dummy-read ICDRR to release the SCL line.

[9] Clear the TDRE flag.

Figure 16.20 Sample Flowchart for Slave Transmit Mode

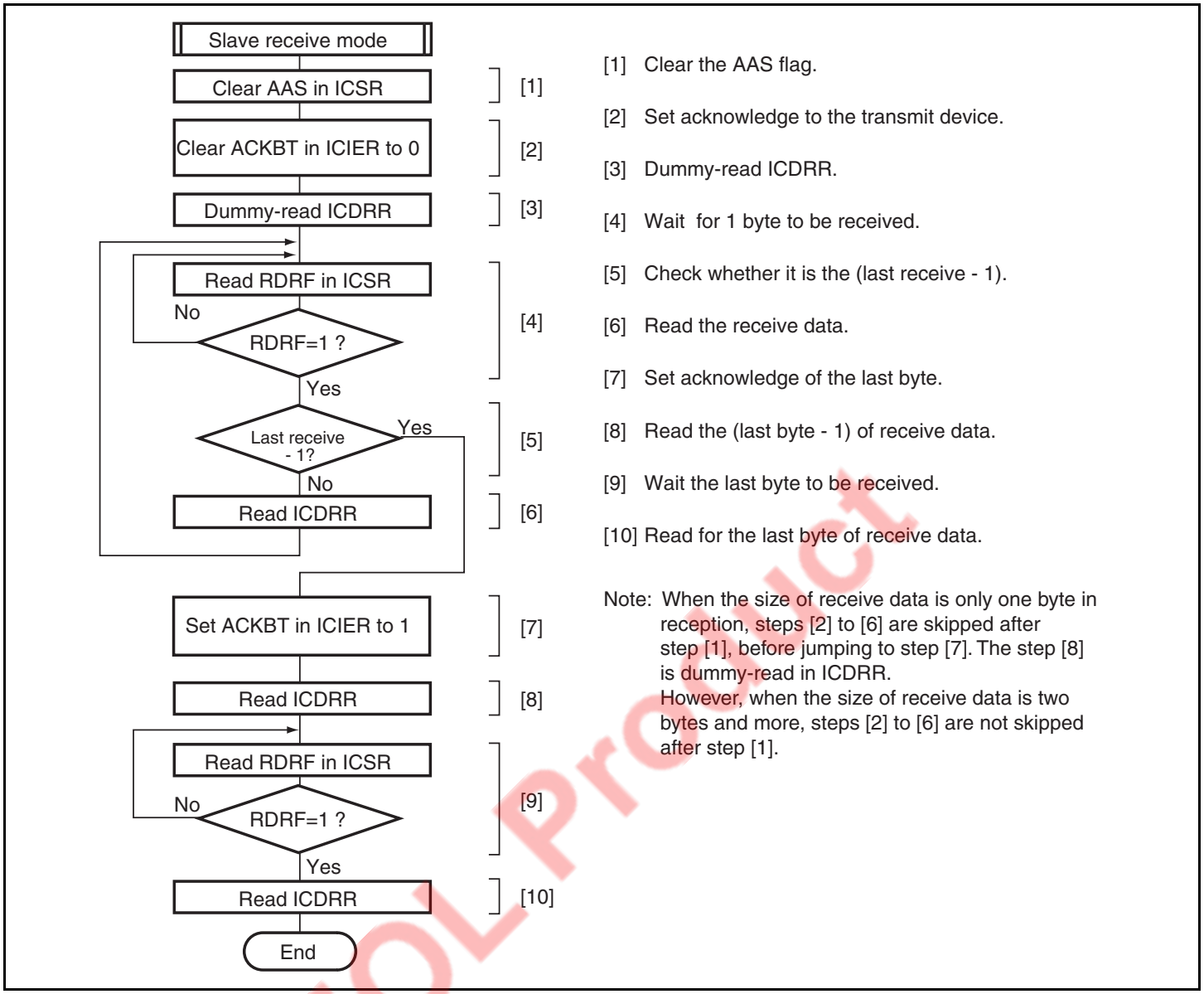


Figure 16.21 Sample Flowchart for Slave Receive Mode

16.5 Interrupt Request

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK receive, STOP recognition, and arbitration lost/overrun error. Table 16.3 shows the contents of each interrupt request.

Table 16.3 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition	I²C Mode	Clocked Synchronous Mode
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 receive	NAKI	{(NACKF=1)+(AL=1)} • (NAKIE=1)	○	×
Arbitration lost/ overrun error			○	○

When the interrupt condition described in table 16.3 is 1, the CPU executes an interrupt exception handling. Interrupt sources should be cleared in the exception handling. The TDRE and TEND bits are automatically cleared to 0 by writing the transmit data to ICDRT. The RDRF bit is automatically cleared to 0 by reading ICDRR. The TDRE bit is set to 1 again at the same time when the transmit data is written to ICDRT. When the TDRE bit is cleared to 0, then an excessive data of one byte may be transmitted.

16.6 Bit Synchronous Circuit

In master mode, this module has a possibility that high level period may be short in the two states described below.

- When SCL is driven to low by the slave device
- When the rising speed of SCL is lowered by the load of the SCL line (load capacitance or pull-up resistance)

Therefore, it monitors SCL and communicates by bit with synchronization.

Figure 16.22 shows the timing of the bit synchronous circuit and table 16.4 shows the time when SCL output changes from low to Hi-Z then SCL is monitored.

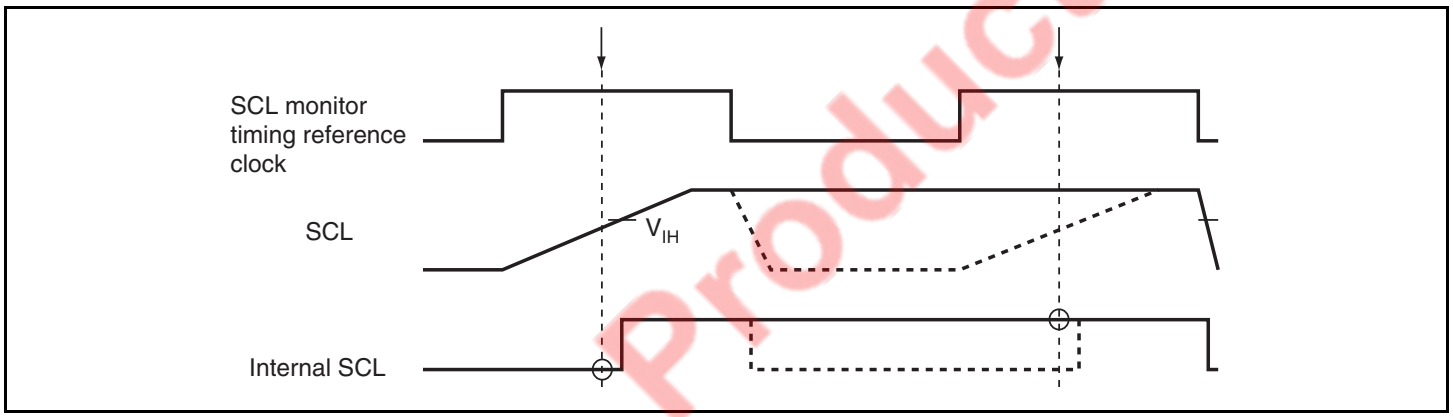


Figure 16.22 The Timing of the Bit Synchronous Circuit

Table 16.4 Time for Monitoring SCL

CKS3	CKS2	CKS2CYC	Time for Monitoring SCL
0	0	0	6.5 pcyc
		1	5.5 pcyc
	1	0	18.5 pcyc
		1	17.5 pcyc
1	0	0	16.5 pcyc
		1	15.5 pcyc
	1	0	40.5 pcyc
		1	39.5 pcyc

Note: The pcyc indicates the peripheral clock cycle.

16.7 Usage Note

Start (retransmission) and stop conditions should be generated after the fall of the ninth clock pulse has been detected. To detect the fall of the ninth clock pulse, read the SCLO bit in the I²C Bus Control Register 2 (ICCR2).

When the start (retransmission) or stop condition is attempt to be generated at the specific timing under the following two conditions, the start or stop condition may not be generated normally. Under conditions other than following two, generation is performed normally.

- When the load of the SCL bus (load capacitance or pull-up resistance) makes the rising speed of SCL slower than speeds shown in section 16.6, Bit Synchronous Circuit
- When the low level period between the eighth and ninth clock pulses is extended and bit synchronous circuit starts operation

Section 17 Compare Match Timer (CMT)

This LSI has an on-chip compare match timer (CMT) consisting of a two-channel 16-bit timer. The CMT has a16-bit counter, and can generate interrupts at set intervals.

17.1 Features

CMT has the following features.

- Selection of four counter input clocks
  - — Any of four internal clocks (Pϕ/4, Pϕ/8, Pϕ/16, and Pϕ/64) can be selected independently for each channel.
- Selection of DMA transfer request or interrupt request generation on compare match
- When not in use, CMT can be stopped by halting its clock supply to reduce power consumption.

Figure 17.1 shows a block diagram of CMT.

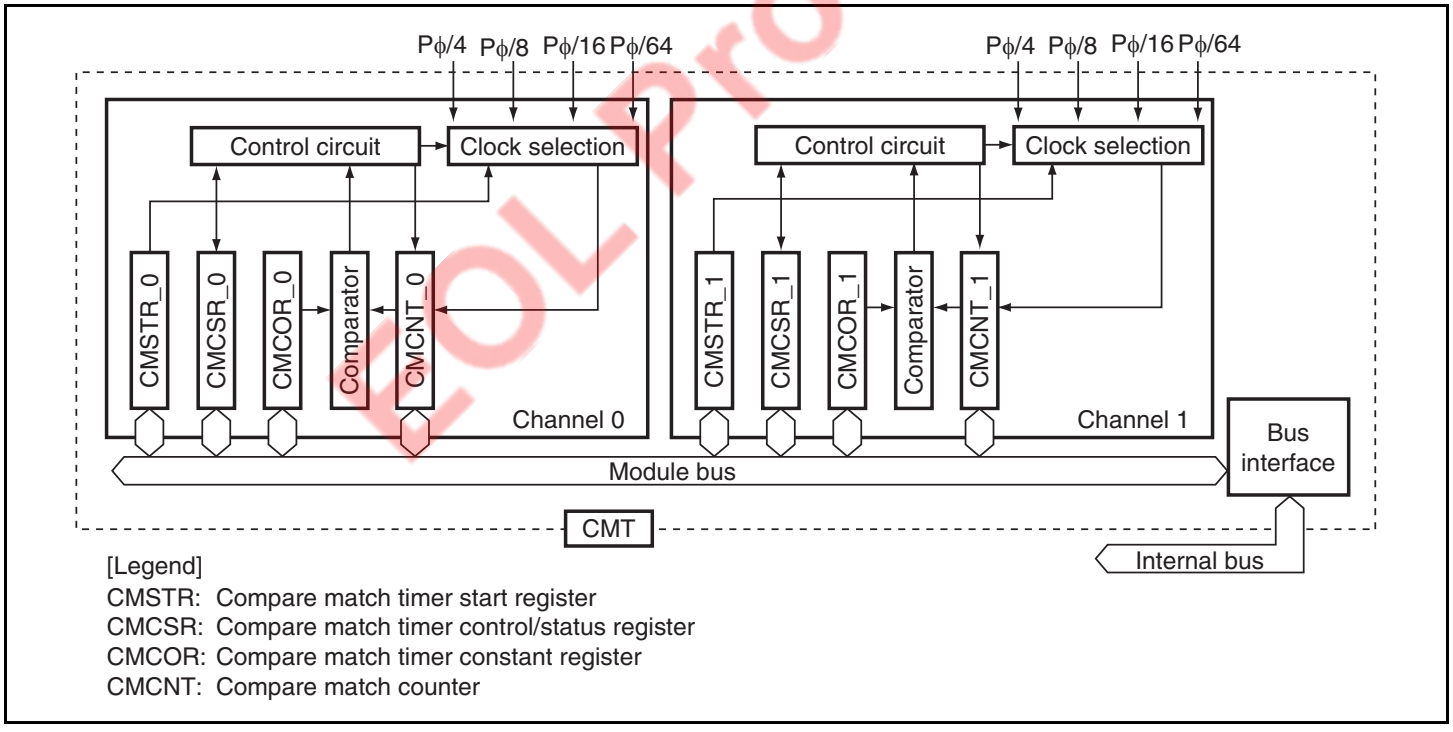


Figure 17.1 Block Diagram of Compare Match Timer

17.2 Register Descriptions

The CMT has the following registers. Refer the section 24, List of Registers and access size for these registers.

- Compare match timer start register_0 (CMSTR_0)
- Compare match timer control/status register_0 (CMCSR_0)
- Compare match counter_0 (CMCNT_0)
- Compare match timer constant register_0 (CMCOR_0)
- Compare match timer start register_1 (CMSTR_1)
- Compare match timer control/status register_1 (CMCSR_1)
- Compare match counter_1 (CMCNT_1)
- Compare match timer constant register_1 (CMCOR_1)

17.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether compare match counter (CMCNT) operates or is stopped.

CMSTR is initialized to H'0000 by a power on reset, but is not initialized in standby mode.

Bit	Bit Name	Initial value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	STR	0	R/W	Count Start Specifies whether compare match counter operates or is stopped. 0: CMCNT count is stopped 1: CMCNT count is started

17.2.2 Compare Match Timer Control/Status Register (CMCSR)

CMCSR is a 16-bit register that indicates compare match generation, enables interrupts or DMA transfer requests, and selects the counter input clock.

CMCSR is initialized to H'0000 by a power on reset, but is not initialized in standby mode.

Bit	Bit Name	Initial value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	CMF	0	R/(W)*	Compare Match Flag Indicates whether or not the values of CMCNT and CMCOR match. 0: CMCNT and CMCOR values do not match [Clearing condition] When 0 is written to CMF after reading CMF = 1 1: CMCNT and CMCOR values match
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	CMR1	0	R/W	Compare Match Request
4	CMR0	0	R/W	These bits enable or disable DMA transfer request or interrupt request generation when a compare match occurs. 00: DMA transfer request/interrupt request disabled 01: DMA transfer request enabled 10: Interrupt request enabled 11: Reserved (Setting prohibited)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial value	R/W	Description
1	CKS1	0	R/W	Clock Select
0	CKS0	0	R/W	These bits select the clock to be input to CMCNT from four internal clocks obtained by dividing the peripheral operating clock (Pϕ). When the STR bit in CMSTR is set to 1, CMCNT starts counting on the clock selected with bits CKS1 and CKS0. 00: Pϕ/4 01: Pϕ/8 10: Pϕ/16 11: Pϕ/64

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

17.2.3 Compare Match Counter (CMCNT)

CMCNT is a 16-bit register used as an up-counter. When the counter input clock is selected with bits CKS1 and CKS0 in CMCSR, and the STR bit in CMSTR is set to 1, CMCNT starts counting using the selected clock.

When the value in CMCNT and the value in compare match constant register (CMCOR) match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1.

CMCNT is initialized to H'0000 by a power on reset, but is not initialized in standby mode.

17.2.4 Compare Match Constant Register (CMCOR)

CMCOR is a 16-bit register that sets the interval up to a compare match with CMCNT.

CMCOR is initialized to H'FFFF by a power on reset, but is not initialized in standby mode.

17.3 Operation

17.3.1 Interval Count Operation

When an internal clock is selected with the CKS1 and CKS0 bits in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts incrementing using the selected clock. When the values in CMCNT and CMCOR match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1. CMCNT then starts counting up again from H'0000.

Figure 17.2 shows the operation of the compare match counter.

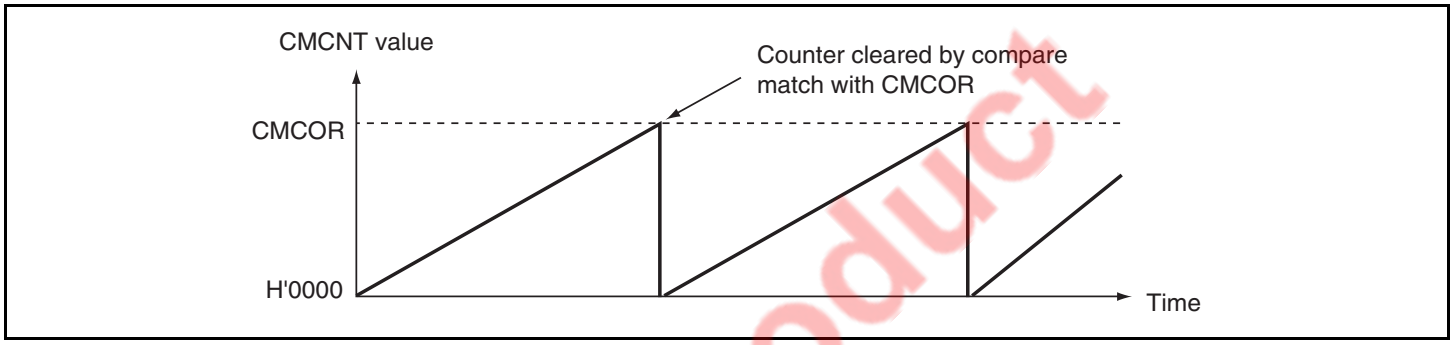


Figure 17.2 Counter Operation

17.3.2 CMCNT Count Timing

One of four internal clocks (Pϕ/4, Pϕ/8, Pϕ/16, and Pϕ/64) obtained by dividing the Pϕ clock can be selected with bits CKS1 and CKS0 in CMCSR. Figure 17.3 shows the timing.

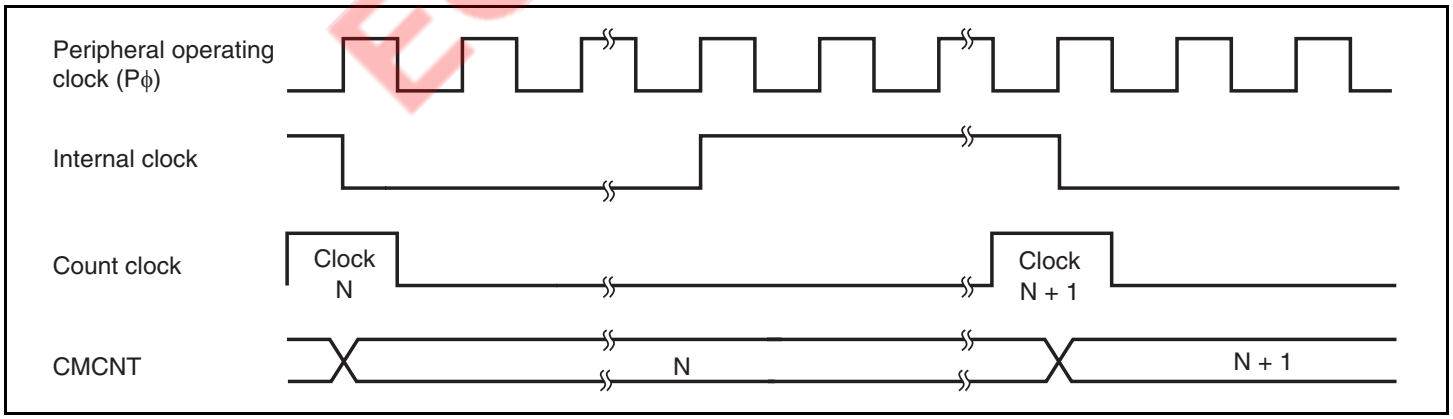


Figure 17.3 Count Timing

17.4 Compare Matches

17.4.1 Timing of Compare Match Flag Setting

When CMCOR and CMCNT match, a compare match signal is generated and the CMF bit in CMCSR is set to 1. The compare match signal is generated in the last state in which the values match (when the CMCNT value is updated to H'0000). That is, after a match between CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT counter clock input. Figure 17.4 shows the timing of CMF bit setting.

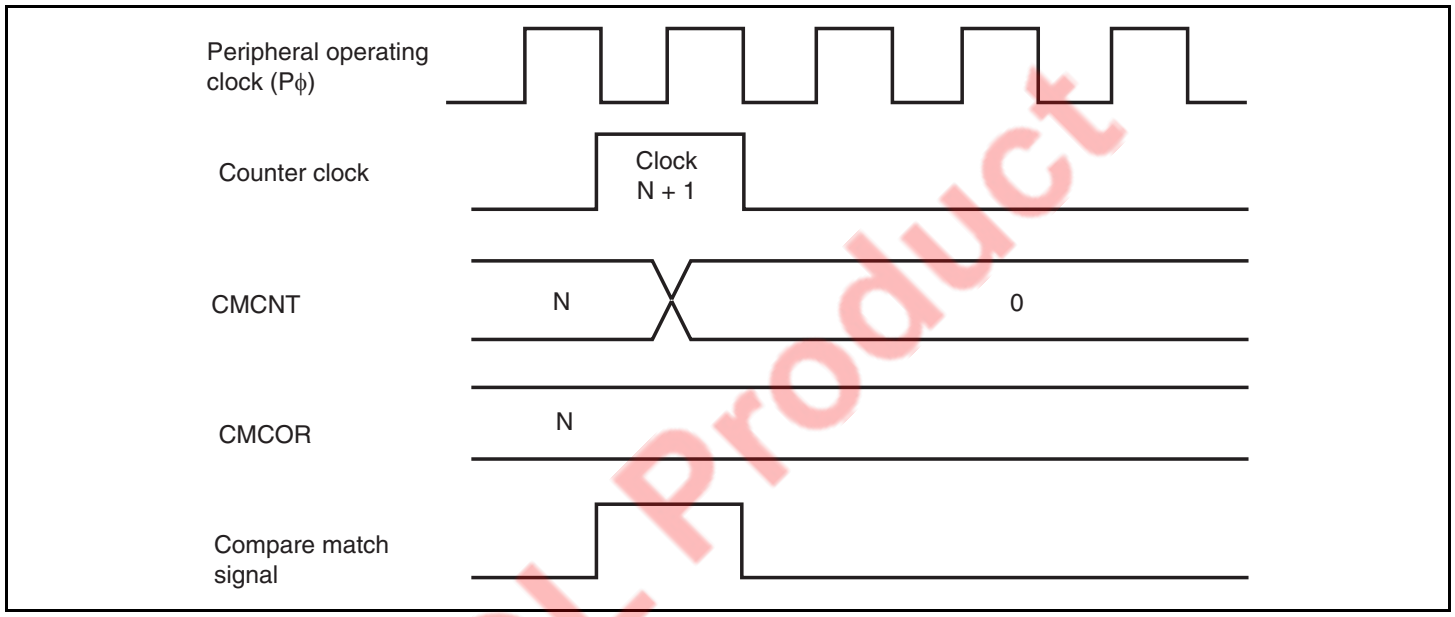


Figure 17.4 Timing of CMF Setting

17.4.2 DMA Transfer Requests and Interrupt Requests

Generation of a DMA transfer request or an interrupt request when a compare match occurs can be selected with bits CMR1 and CMR0 in CMCSR.

With a DMA transfer request, the request signal is cleared automatically when the DMAC accepts the request. However, the CMF bit in CMCSR is not cleared to 0.

An interrupt request is cleared by writing 0 to the CMF bit in CMCSR. Therefore, an operation to set CMF = 0 must be performed by the user in the exception handling routine. If this operation is not carried out, another interrupt will be generated.

17.4.3 Timing of Compare Match Flag Clearing

The CMF bit in CMCSR is cleared by first, reading as 1 then writing to 0.

Section 18 Multi-Function Timer Pulse Unit (MTU)

This LSI has an on-chip multi-function timer pulse unit (MTU) that comprises five 16-bit timer channels.

The block diagram is shown in figure 18.1.

18.1 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

    Multiple timer counters (TCNT) can be written to simultaneously

    Simultaneous clearing by compare match and input capture is possible

    Register simultaneous input/output is possible by synchronous counter operation
  - A maximum 12-phase PWM output is possible in combination with synchronous operation
- Buffer operation settable for channels 0, 3, and 4
- Phase counting mode settable independently for each of channels 1 and 2
- Cascade connection operation
- Fast access via internal 16-bit bus
- 23 interrupt sources
- Automatic transfer of register data
- A/D converter conversion start trigger can be generated
- Module standby mode can be set

Table 18.1 MTU Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
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 TCLKA TCLKB	φ/1 φ/4 φ/16 φ/64 φ/256 φ/1024 TCLKA TCLKB
General registers		TGRA_0 TGRB_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4
General registers/ buffer registers		TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	TGRC_4 TGRD_4
I/O pins		TIOC0A TIOC0B TIOC0C TIOC0D	TIOC1A TIOC1B	TIOC2A TIOC2B	TIOC3A TIOC3B TIOC3C TIOC3D	TIOC4A TIOC4B TIOC4C TIOC4D
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
Compare match output	0 output	O	O	O	O	O
	1 output	O	O	O	O	O
	Toggle output	O	O	O	O	O
Input capture function		O	O	O	O	O
Synchronous operation		O	O	O	O	O
PWM mode 1		O	O	O	O	O
PWM mode 2		O	O	O	—	—
Phase counting mode		—	O	O	—	—
Buffer operation		O	—	—	O	O

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
DMA 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
A/D converter start 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
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	5 sources • Compare match or input capture 4A • Compare match or input capture 4B • Compare match or input capture 4C • Compare match or input capture 4D • Overflow

[Legend]

O: Possible

—: Not possible

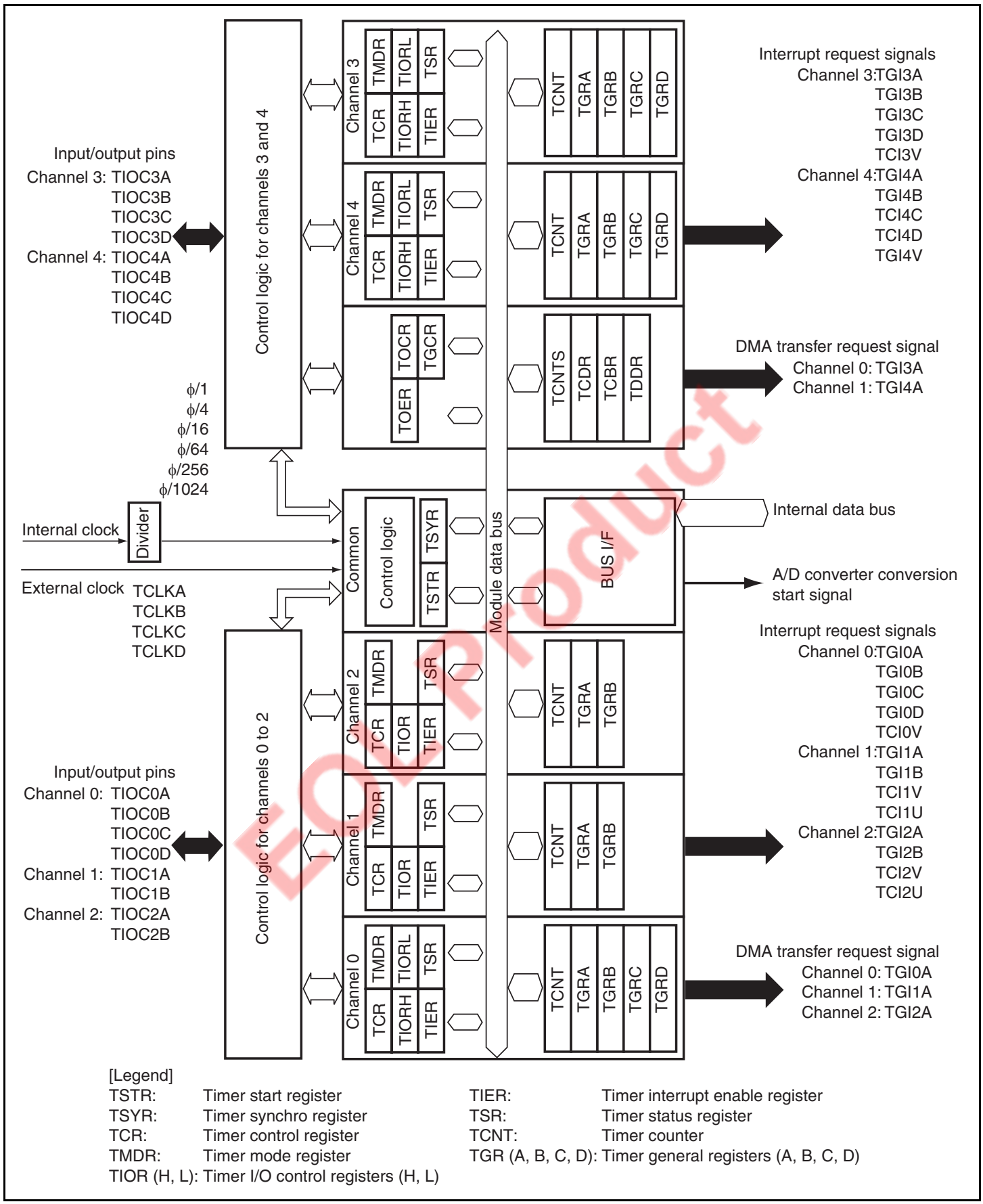


Figure 18.1 Block Diagram of MTU

18.2 Input/Output Pins

Table 18.2 MTU Pin Configuration

Channel	Symbol	I/O	Function
All	TCLKA	Input	External clock A input pin (Channel 1 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channel 1 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 phase counting mode B phase input)
0	TIOC0A	I/O	TGRA_0 input capture input/output compare output/PWM output pin
	TIOC0B	I/O	TGRB_0 input capture input/output compare output/PWM output pin
	TIOC0C	I/O	TGRC_0 input capture input/output compare output/PWM output pin
	TIOC0D	I/O	TGRD_0 input capture input/output compare output/PWM output pin
1	TIOC1A	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOC1B	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOC2A	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOC2B	I/O	TGRB_2 input capture input/output compare output/PWM output pin
3	TIOC3A	I/O	TGRA_3 input capture input/output compare output/PWM output pin
	TIOC3B	I/O	TGRB_3 input capture input/output compare output/PWM output pin
	TIOC3C	I/O	TGRC_3 input capture input/output compare output/PWM output pin
	TIOC3D	I/O	TGRD_3 input capture input/output compare output/PWM output pin
4	TIOC4A	I/O	TGRA_4 input capture input/output compare output/PWM output pin
	TIOC4B	I/O	TGRB_4 input capture input/output compare output/PWM output pin
	TIOC4C	I/O	TGRC_4 input capture input/output compare output/PWM output pin
	TIOC4D	I/O	TGRD_4 input capture input/output compare output/PWM output pin

18.3 Register Descriptions

The MTU has the following registers. To distinguish registers in each channel, TCR for channel 0 is expressed as TCR_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)
- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register _1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)
- Timer control register_2 (TCR_2)
- Timer mode register_2 (TMDR_2)
- Timer I/O control register_2 (TIOR_2)
- Timer interrupt enable register_2 (TIER_2)
- Timer status register_2 (TSR_2)
- Timer counter_2 (TCNT_2)
- Timer general register A_2 (TGRA_2)
- Timer general register B_2 (TGRB_2)
- Timer control register_3 (TCR_3)
- Timer mode register_3 (TMDR_3)
- Timer I/O control register H_3 (TIORH_3)

- Timer I/O control register L_3 (TIORL_3)
- Timer interrupt enable register_3 (TIER_3)
- Timer status register_3 (TSR_3)
- Timer counter_3 (TCNT_3)
- Timer general register A_3 (TGRA_3)
- Timer general register B_3 (TGRB_3)
- Timer general register C_3 (TGRC_3)
- Timer general register D_3 (TGRD_3)
- Timer control register_4 (TCR_4)
- Timer mode register_4 (TMDR_4)
- Timer I/O control register H_4 (TIORH_4)
- Timer I/O control register L_4 (TIORL_4)
- Timer interrupt enable register_4 (TIER_4)
- Timer status register_4 (TSR_4)
- Timer counter_4 (TCNT_4)
- Timer general register A_4 (TGRA_4)
- Timer general register B_4 (TGRB_4)
- Timer general register C_4 (TGRC_4)
- Timer general register D_4 (TGRD_4)

Common registers:

- Timer start register (TSTR)
- Timer synchro register (TSYR)

Common registers for timers 3 and 4:

- Timer output master enable register (TOER)
- Timer output control register (TOCR)
- Timer gate control register (TGCR)
- Timer cycle data register (TCDR)
- Timer dead time data register (TDDR)
- Timer subcounter (TCNTS)
- Timer cycle buffer register (TCBR)

18.3.1 Timer Control Register (TCR)

The TCR registers are 8-bit readable/writable registers that control the TCNT operation for each channel. The MTU has a total of five TCR registers, one for each channel (channel 0 to 4). TCR register settings should be conducted 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 18.3 and 18.4 for details.
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. Pϕ/4 both edges = ϕ/2 rising edge). If phase counting mode is used on channels 1 and 2, 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. When ϕ/1, or the overflow/underflow of another channel is selected for the input clock, although values can be written, counter operation compiles with the initial value. 00: Count at rising edge 01: Count at falling edge 1X: Count at both edges [Legend] X: Don't care
2	TPSC2	0	R/W	Time Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 18.5 to 18.8 for details.
0	TPSC0	0	R/W	

Table 18.3 CCLR0 to CCLR2 (Channels 0, 3, and 4)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3, 4	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 is set 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 18.4 CCLR0 to CCLR2 (Channels 1 and 2)

Channel	Bit 7 Reserved*[2]	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2	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 is selected by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1 and 2. This bit is always read as 0 and cannot be modified.

Table 18.5 TPSC0 to TPSC2 (Channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/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 18.6 TPSC0 to TPSC2 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/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 Pϕ/256
			1	Counts on TCNT_2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

Table 18.7 TPSC0 to TPSC2 (Channel 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/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 Pϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 18.8 TPSC0 to TPSC2 (Channels 3 and 4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3, 4	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	Internal clock: counts on Pϕ/256
			1	Internal clock: counts on Pϕ/1024
		1	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input

18.3.2 Timer Mode Register (TMDR)

The TMDR registers are 8-bit readable/writable registers that are used to set the operating mode of each channel. The MTU has five TMDR registers, one for each channel. TMDR register settings should be changed only when TCNT operation is stopped.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1. The write value should always be 1.
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 and 2, which have no TGRD, bit 5 is reserved. It is always read as 0, and should only be written with 0. 0: TGRB and TGRD operate 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 and 2, which have no TGRC, bit 4 is reserved. It is always read as 0, and should only be written with 0. 0: TGRA and TGRD operate normally 1: TGRA and TGRC used together for buffer operation
3	MD3	0	R/W	Modes 3 to 0
2	MD2	0	R/W	These bits are used to set the timer operating mode.
1	MD1	0	R/W	See table 18.9 for details.
0	MD0	0	R/W	

Table 18.9 MD0 to MD3

Bit 3 MD3	Bit 2 MD2	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
			1	Reserved (do not set)
		1	0	PWM mode 1
			1	PWM mode 2*[1]
	1	0	0	Phase counting mode 1*[2]
			1	Phase counting mode 2*[2]
		1	0	Phase counting mode 3*[2]
			1	Phase counting mode 4*[2]
1	0	0	0	Reset synchronous PWM mode*[3]
			1	Reserved (do not set)
		1	X	Reserved (do not set)
	1	0	0	Reserved (do not set)
			1	Complementary PWM mode 1 (transmit at peak)*[3]
		1	0	Complementary PWM mode 2 (transmit at valley)*[3]
			1	Complementary PWM mode 2 (transmit at peak and valley)*[3]

[Legend]

X: Don't care

Notes: 1. PWM mode 2 cannot be set for channels 3, 4.

2. Phase counting mode cannot be set for channels 0, 3, 4.

3. Reset synchronous PWM mode, complementary PWM mode can only be set for channel 3. When channel 3 is set to reset synchronous PWM mode or complementary PWM mode, the channel 4 settings become ineffective and automatically conform to the channel 3 settings. However, do not set channel 4 to reset synchronous PWM mode or complementary PWM mode. Reset synchronous PWM mode and complementary PWM mode cannot be set for channels 0, 1, 2.

18.3.3 Timer I/O Control Register (TIOR)

The TIOR registers are 8-bit readable/writable registers that control the TGR registers. The MTU has eight TIOR registers, two each for channels 0, 3, and 4, and one each for channels 1 and 2.

Care is required as 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, TIORH_4

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	See the following tables.
4	IOB0	0	R/W	TIORH_0: Table 18.10 TIOR_1: Table 18.12 TIOR_2: Table 18.13 TIORH_3: Table 18.14 TIORH_4: Table 18.16
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	See the following tables.
0	IOA0	0	R/W	TIORH_0: Table 18.18 TIOR_1: Table 18.20 TIOR_2: Table 18.21 TIORH_3: Table 18.22 TIORH_4: Table 18.24

- TIORL_0, TIORL_3, TIORL_4

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	When TGRD is used as the buffer register of TGRB, this setting is disabled, and input capture/output compare does not occur.
4	IOD0	0	R/W	See the following tables. TIORL_0: Table 18.11 TIORL_3: Table 18.15 TIORL_4: Table 18.17
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	When TGRC is used as the buffer register of TGRA, this setting is disabled, and input capture/output compare does not occur.
0	IOC0	0	R/W	See the following tables. TIORL_0: Table 18.19 TIORL_3: Table 18.23 TIORL_4: Table 18.25

Table 18.10 TIORH_0 (Channel 0)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_0 Function**	**TIOC0B Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count- up/count-down

[Legend]

X: Don't care

Note: * 0 is output until TIOR contents is specified after a power-on reset and entering standby mode.

Table 18.11 TIORL_0 (Channel 0)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_0 Function**	**TIOC0D Pin Function**
0	0	0	0	Output compare register*[2]	Output hold*[1]
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*[2]	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down*[2]

[Legend]

X: Don't care

Notes: 1. The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

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 18.12 TIOR_1 (Channel 1)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_1 Function**	**TIOC1B Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Input capture at generation of TGRC_0 compare match/input capture

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

Table 18.13 TIOR_2 (Channel 2)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_2 Function**	**TIOC2B Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

Table 18.14 TIORH_3 (Channel 3)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_3 Function**	**TIOC3B Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

Table 18.15 TIORL_3 (Channel 3)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_3 Function**	**TIOC3D Pin Function**
0	0	0	0	Output compare register*[2]	Output hold*[1]
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*[1]
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*[2]	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

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 18.16 TIORH_4 (Channel 4)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_4 Function**	**TIOC4B Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

Table 18.17 TIORL_4 (Channel 4)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRB_4 Function**	**TIOC4B Pin Function**
0	0	0	0	Output compare register*[2]	Output hold*[1]
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*[1]
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*[2]	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

2. When the BFB bit in TMDR_4 is set to 1 and TGRD_4 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 18.18 TIORH_0 (Channel 0)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_0 Function**	**TIOC0A Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

Table 18.19 TIORL_0 (Channel 0)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_0 Function**	**TIOC0C Pin Function**
0	0	0	0	Output compare register*2	Output hold*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*1
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Notes: 1. The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

2. When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 18.20 TIOR_1 (Channel 1)

				Description	
Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	TGRA_1 Function	TIOC1A Pin Function
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Input capture at generation of channel 0/TGRA_0 compare match/input capture

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

Table 18.21 TIOR_2 (Channel 2)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_2 Function**	**TIOC2A Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

Table 18.22 TIORH_3 (Channel 3)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_3 Function**	**TIOC3A Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

Table 18.23 TIORL_3 (Channel 3)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_3 Function**	**TIOC3C Pin Function**
0	0	0	0	Output compare register*2	Output hold*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*1
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

2. 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 18.24 TIORH_4 (Channel 4)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_4 Function**	**TIOC4A Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

Table 18.25 TIORL_4 (Channel 4)

				Description	
Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	TGRA_4 Function	TIOC4C Pin Function
0	0	0	0	Output compare register*2	Output hold*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold*1
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. The low level output is retained until TIOR contents is specified after a power-on reset and entering standby mode.

2. When the BFA bit in TMDR_4 is set to 1 and TGRC_4 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

18.3.4 Timer Interrupt Enable Register (TIER)

The TIER registers are 8-bit readable/writable registers that control enabling or disabling of interrupt requests for each channel. The MTU has five 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	TGFASEL	0	R/W	TGFA Interrupt/DMA Transfer Select Selects the TGFA interrupt request or DMA transfer request when the TGFA flag in TGRA is set to 1. 0: Interrupt request 1: DMA transfer request
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 and 2. In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0, and should only be written with 0. 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

Bit	Bit Name	Initial value	R/W	Description
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, 3, and 4. In channels 1 and 2, bit 3 is reserved. It is always read as 0, and should only be written with 0. 0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled
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, 3, and 4. In channels 1 and 2, bit 2 is reserved. It is always read as 0, and should only be written with 0. 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 and DMA transfer when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit and DMA transfer disabled 1: Interrupt requests (TGIA) by TGFA bit and DMA transfer enabled

Note: Do not change the setting of the timer interrupt enable register (TIER) during DMA transfer.

18.3.5 Timer Status Register (TSR)

The TSR registers are 8-bit readable/writable registers that indicate the status of each channel. The MTU has five 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, 3, and 4. In channel 0, bit 7 is reserved. It is always read as 1, and should only be written with 1. 0: TCNT counts down 1: TCNT counts up
6	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
5	TCFU	0	R/(W)	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1 and 2 are set to phase counting mode. Only 0 can be written, for flag clearing. In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0, and should only be written with 0. [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

Bit	Bit Name	Initial value	R/W	Description
4	TCFV	0	R/(W)	Overflow Flag Status flag that indicates that TCNT overflow has occurred. Only 0 can be written, for flag clearing. [Setting conditions] • When the TCNT value overflows (changes from H'FFFF to H'0000) • In channel 4, when TCNT_4 is underflowed (H'0001 → H'0000) in complementary PWM mode. [Clearing condition] • When 0 is written to TCFV after reading TCFV = 1
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, 3, and 4. Only 0 can be written, for flag clearing. In channels 1 and 2, bit 3 is reserved. It is always read as 0, and should only be written with 0. [Setting conditions] • When TCNT = TGRD and TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal and TGRD is functioning as input capture register [Clearing condition] • When 0 is written to TGFD after reading TGFD = 1

Bit	Bit Name	Initial value	R/W	Description
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, 3, and 4. Only 0 can be written, for flag clearing. In channels 1 and 2, bit 2 is reserved. It is always read as 0, and should only be written with 0. [Setting conditions] • When TCNT = TGRC and TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal and TGRC is functioning as input capture register [Clearing condition] • When 0 is written to TGFC after reading TGFC = 1
1	TGFB	0	R/(W)	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGRB input capture or compare match. Only 0 can be written, for flag clearing. [Setting conditions] • When TCNT = TGRB and TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal and TGRB is functioning as input capture register [Clearing condition] • When 0 is written to TGFB after reading TGFB = 1

Bit	Bit Name	Initial value	R/W	Description
0	TGFA	0	R/(W)	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA input capture or compare match. Only 0 can be written, for flag clearing. [Setting conditions] • When TCNT = TGRA and TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal and TGRA is functioning as input capture register [Clearing condition] • When 0 is written to TGFA after reading TGFA = 1 For DMA, 0 must not be written to after reading TGFA = 1. This flag is cleared only by hardware.*

Note: Write 0 after reading TGFA=1 only when a DMA address error occurs during a DMA read cycle.

18.3.6 Timer Counter (TCNT)

The TCNT registers are 16-bit readable/writable counters. The MTU has five TCNT counters, one for each channel.

The TCNT counters are initialized to H'0000 by a power-on reset and in standby mode.

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

18.3.7 Timer General Register (TGR)

The TGR registers are dual function 16-bit readable/writable registers, functioning as either output compare or input capture registers. The MTU has 16 TGR registers, four each for channels 0, 3, and 4 and two each for channels 1 and 2. TGRC and TGRD for channels 0, 3, and 4 can also be designated for operation as buffer registers. The TGR registers cannot be accessed in 8-bit units; they must always be accessed in 16-bit units. TGR buffer register combinations are TGRA to TGRC and TGRB to TGRD.

18.3.8 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage for channels 0 to 4. 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	CST4	0	R/W	Counter Start 4 and 3
6	CST3	0	R/W	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_4 and TCNT_3 count operation is stopped 1: TCNT_4 and TCNT_3 performs count operation
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CST2	0	R/W	Counter Start 2 to 0
1	CST1	0	R/W	These bits select operation or stoppage for TCNT.
0	CST0	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. 0: TCNT_2 and TCNT_0 count operation is stopped 1: TCNT_2 and TCNT_0 performs count operation

18.3.9 Timer Synchro Register (TSYR)

TSYR is an 8-bit readable/writable register that selects independent operation or synchronous operation for the channel 0 to 4 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

Bit	Bit Name	Initial value	R/W	Description
7	SYNC4	0	R/W	Timer Synchro 4 and 3
6	SYNC3	0	R/W	These bits are used to select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by 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 CCLR0 to CCLR2 in TCR. 0: TCNT_4 and TCNT_3 operate independently (TCNT presetting/clearing is unrelated to other channels) 1: TCNT_4 and TCNT_3 performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	SYNC2	0	R/W	Timer Synchro 2 to 0
1	SYNC1	0	R/W	These bits are used to select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by 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 CCLR0 to CCLR2 in TCR. 0: TCNT_2 to TCNT_0 operates independently (TCNT presetting /clearing is unrelated to other channels). 1: TCNT_2 to TCNT_0 performs synchronous operation. TCNT synchronous presetting/synchronous clearing is possible.
0	SYNC0	0	R/W	

18.3.10 Timer Output Master Enable Register (TOER)

TOER is an 8-bit readable/writable register that enables/disables output settings for output pins TIOC4D, TIOC4C, TIOC3D, TIOC4B, TIOC4A, and TIOC3B. These pins do not output correctly if the TOER bits have not been set. Set TOER of CH3 and CH4 prior to setting TIOR of CH3 and CH4.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
5	OE4D	0	R/W	Master Enable TIOC4D This bit enables/disables the TIOC4D pin MTU output. 0: MTU output is disabled 1: MTU output is enabled
4	OE4C	0	R/W	Master Enable TIOC4C This bit enables/disables the TIOC4C pin MTU output. 0: MTU output is disabled 1: MTU output is enabled
3	OE3D	0	R/W	Master Enable TIOC3D This bit enables/disables the TIOC3D pin MTU output. 0: MTU output is disabled 1: MTU output is enabled
2	OE4B	0	R/W	Master Enable TIOC4B This bit enables/disables the TIOC4B pin MTU output. 0: MTU output is disabled 1: MTU output is enabled
1	OE4A	0	R/W	Master Enable TIOC4A This bit enables/disables the TIOC4A pin MTU output. 0: MTU output is disabled 1: MTU output is enabled
0	OE3B	0	R/W	Master Enable TIOC3B This bit enables/disables the TIOC3B pin MTU output. 0: MTU output is disabled 1: MTU output is enabled

18.3.11 Timer Output Control Register (TOCR)

TOCR is an 8-bit readable/writable register that enables/disables PWM synchronized toggle output in complementary PWM mode/reset synchronized PWM mode, and controls output level inversion of PWM output.

Bit	Bit Name	Initial value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	PSYE	0	R/W	PWM Synchronous Output Enable This bit selects the enable/disable of toggle output synchronized with the PWM period. 0: Toggle output is disabled 1: Toggle output is enabled
5 to 2	—	All 0	R	Reserved These bits are always read as 0. Only 0 should be written to this bit.
1	OLSN	0	R/W	Output Level Select N This bit selects the reverse phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 18.26
0	OLSP	0	R/W	Output Level Select P This bit selects the positive phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 18.27.

Table 18.26 Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLSN	Initial Output	Active Level	Increment Count	Decrement Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The reverse phase waveform initial output value changes to active level after elapse of the dead time after count start.

Table 18.27 Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLSP	Initial Output	Active Level	Increment Count	Decrement Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Figure 18.2 shows an example of complementary PWM mode output (one phase) when OLSN = 1, OLSP = 1.

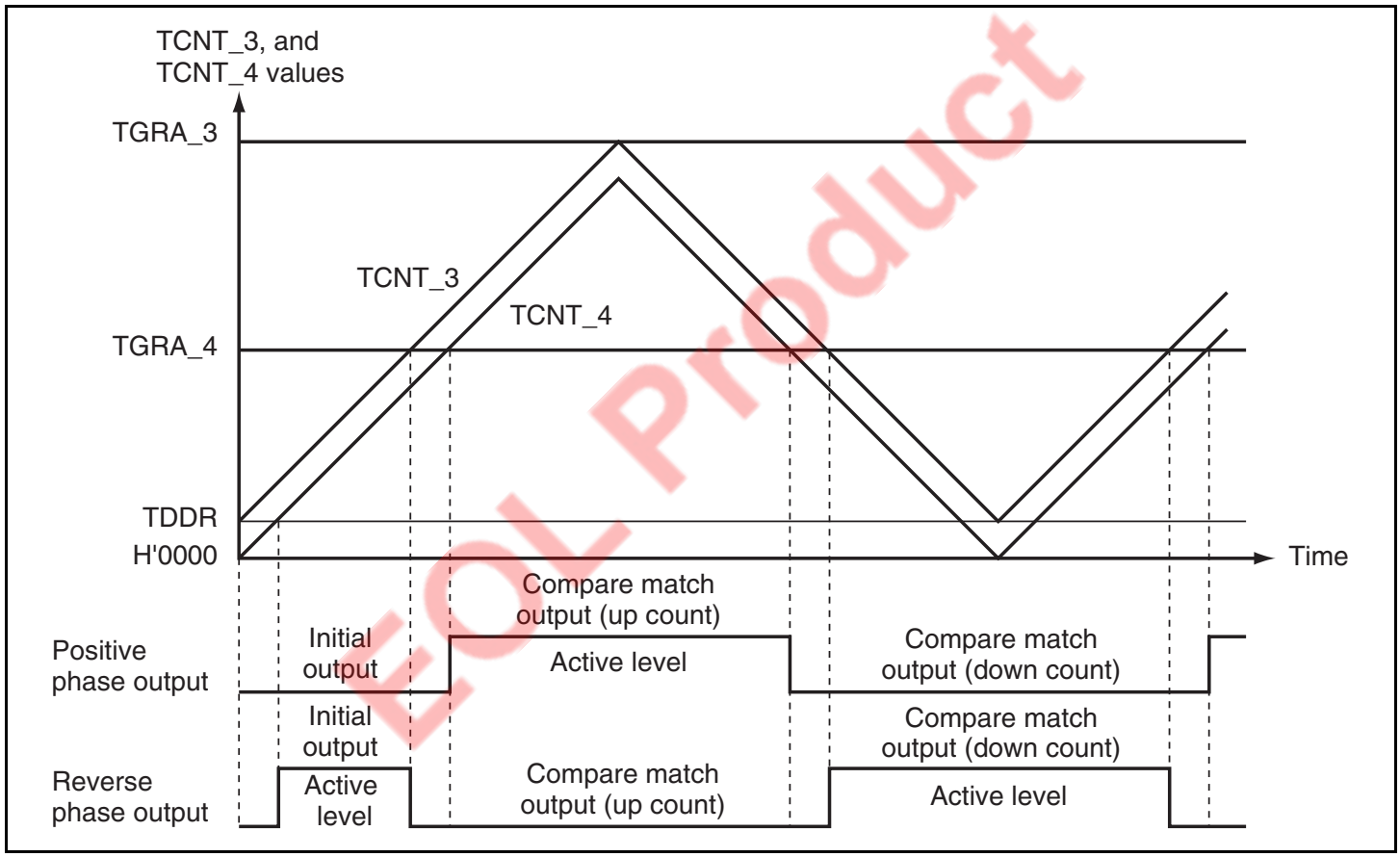


Figure 18.2 Complementary PWM Mode Output Level Example

18.3.12 Timer Gate Control Register (TGCR)

TGCR is an 8-bit readable/writable register that controls the waveform output necessary for brushless DC motor control in reset-synchronized PWM mode/complementary PWM mode. These register settings are ineffective for anything other than complementary PWM mode/reset-synchronized PWM mode.

Bit	Bit Name	Initial value	R/W	Description
7	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
6	BDC	0	R/W	Brushless DC Motor This bit selects whether to make the functions of this register (TGCR) effective or ineffective. 0: Ordinary output 1: Functions of this register are made effective
5	N	0	R/W	Reverse Phase Output (N) Control This bit selects whether the level output or the reset-synchronized PWM/complementary PWM output while the reverse pins (TIOC3D, TIOC4C, and TIOC4D) are on-output. 0: Level output 1: Reset synchronized PWM/complementary PWM output
4	P	0	R/W	Positive Phase Output (P) Control This bit selects whether the level output or the reset-synchronized PWM/complementary PWM output while the positive pin (TIOC3B, TIOC4A, and TIOC4B) are on-output. 0: Level output 1: Reset synchronized PWM/complementary PWM output

Bit	Bit Name	Initial value	R/W	Description
3	FB	0	R/W	External Feedback Signal Enable This bit selects whether the switching of the output of the positive/reverse phase is carried out automatically with the MTU/channel 0 TGRA, TGRB, TGRC input capture signals or by writing 0 or 1 to bits 2 to 0 in TGCR. 0: Output switching is carried out by external input (Input sources are channel 0 TGRA, TGRB, TGRC input capture signal) 1: Output switching is carried out by software (TGCR's UF, VF, WF settings).
2	WF	0	R/W	Output Phase Switch 2 to 0 These bits set the positive phase/negative phase output phase on or off state. The setting of these bits is valid only when the FB bit in this register is set to 1. In this case, the setting of bits 2 to 0 is a substitute for external input. See table 18.28.
1	VF	0	R/W	
0	UF	0	R/W	

Table 18.28 Output level Select Function

			Function					
Bit 2	Bit 1	Bit 0	TIOC3B	TIOC4A	TIOC4B	TIOC3D	TIOC4C	TIOC4D
WF	VF	UF	U Phase	V Phase	W Phase	U Phase	V Phase	W Phase
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
		1	ON	OFF	OFF	OFF	OFF	ON
	1	0	OFF	ON	OFF	ON	OFF	OFF
		1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
		1	ON	OFF	OFF	OFF	ON	OFF
	1	0	OFF	OFF	ON	ON	OFF	OFF
		1	OFF	OFF	OFF	OFF	OFF	OFF

18.3.13 Timer Subcounter (TCNTS)

TCNTS is a 16-bit read-only counter that is used only in complementary PWM mode.

Note: Accessing TCNTS in 8-bit units is prohibited. Always access in 16-bit units.

18.3.14 Timer Dead Time Data Register (TDDR)

TDDR is a 16-bit register, used only in complementary PWM mode, that specifies the TCNT_3 and TCNT_4 counter offset values. In complementary PWM mode, when the TCNT_3 and TCNT_4 counters are cleared and then restarted, the TDDR value is loaded into the TCNT_3 counter and the count operation starts.

Note: Accessing TDDR in 8-bit units is prohibited. Always access in 16-bit units.

18.3.15 Timer Period Data Register (TCDR)

TCDR is a 16-bit register used only in complementary PWM mode. Set half the PWM carrier sync value as the TCDR value. This register is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs, the TCNTS counter switches direction (decrement to increment).

Note: Accessing TCDR in 8-bit units is prohibited. Always access in 16-bit units.

18.3.16 Timer Period Buffer Register (TCBR)

TCBR is a 16-bit register used only in complementary PWM mode. It functions as a buffer register for TCDR. The TCBR values are transferred to TCDR with the transfer timing set in TMDR.

Note: Accessing TCBR in 8-bit units is prohibited. Always access in 16-bit units.

18.3.17 Bus Master Interface

The timer counters (TCNT), general registers (TGR), timer subcounter (TCNTS), timer period buffer register (TCBR), and timer dead time data register (TDDR), and timer period data register (TCDR) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/writes. 8-bit read/write is not possible. Always access in 16-bit units.

All registers other than the above registers are 8-bit registers. These are connected to the CPU by a 16-bit data bus, so 16-bit read/writes and 8-bit read/writes are both possible.

18.4 Operation

18.4.1 Basic Functions

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Always set the MTU external pins function using the pin function controller (PFC).

- Counter Operation

When one of bits CST0 to CST4 is set to 1 in TSTR, the TCNT counter for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

Example of Count Operation Setting Procedure: Figure 18.3 shows an example of the count operation setting procedure.

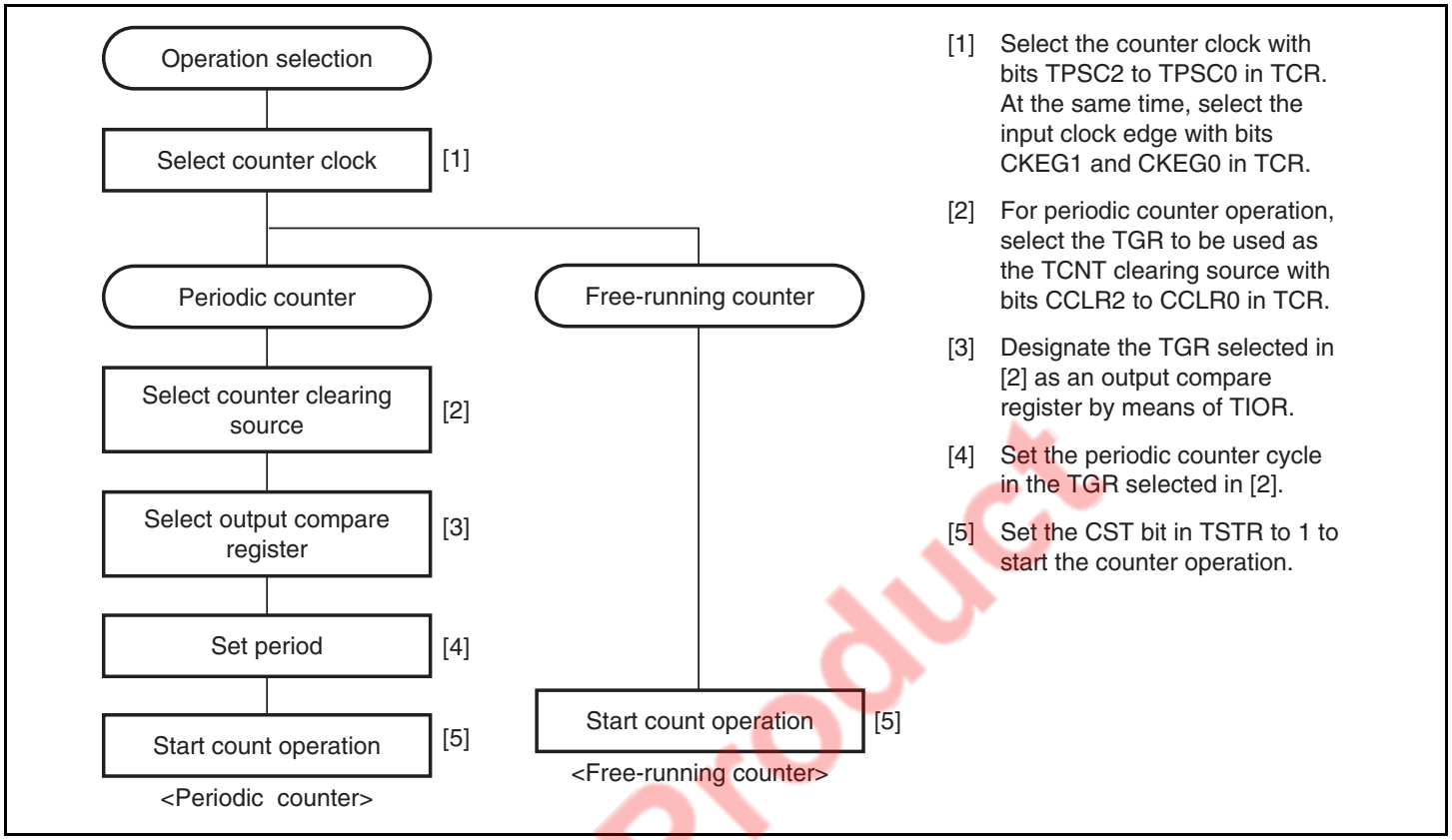


Figure 18.3 Example of Counter Operation Setting Procedure

Free-Running Count Operation and Periodic Count Operation: Immediately after a reset, the MTU'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 (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 MTU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

Figure 18.4 illustrates free-running counter operation.

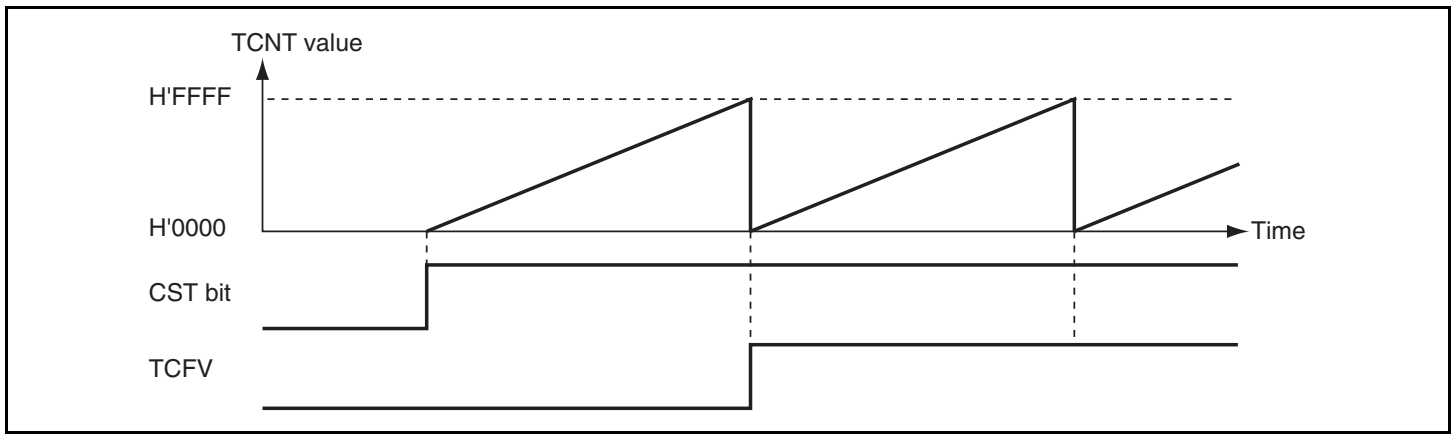


Figure 18.4 Free-Running Counter Operation

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 CCLR0 to CCLR2 in TCR. After the settings have been made, TCNT starts up-count 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 18.5 illustrates periodic counter operation.

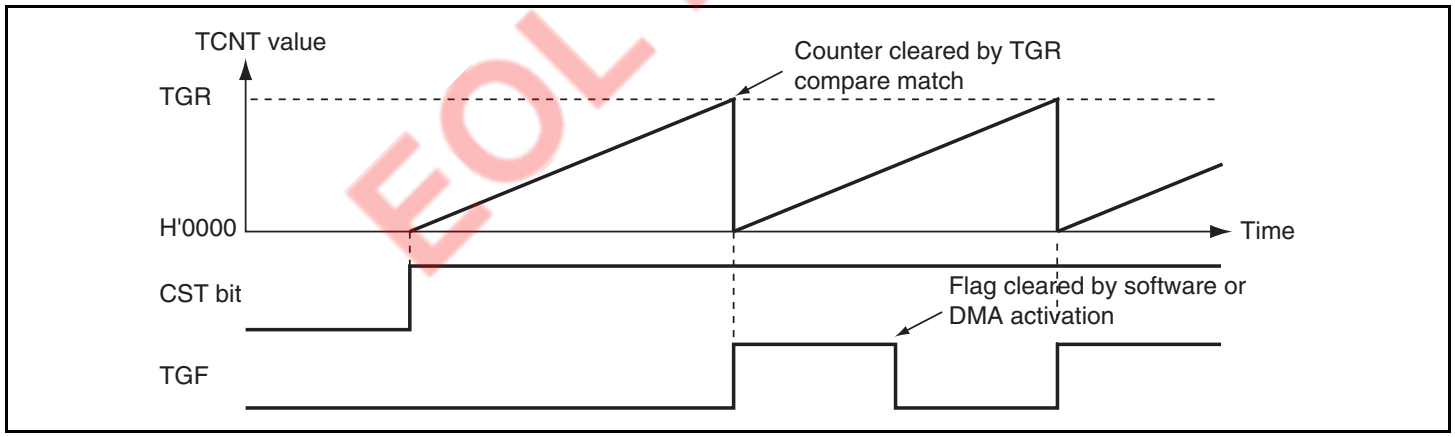


Figure 18.5 Periodic Counter Operation

- Waveform Output by Compare Match

The MTU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

Example of Setting Procedure for Waveform Output by Compare Match: Figure 18.6 shows an example of the setting procedure for waveform output by compare match

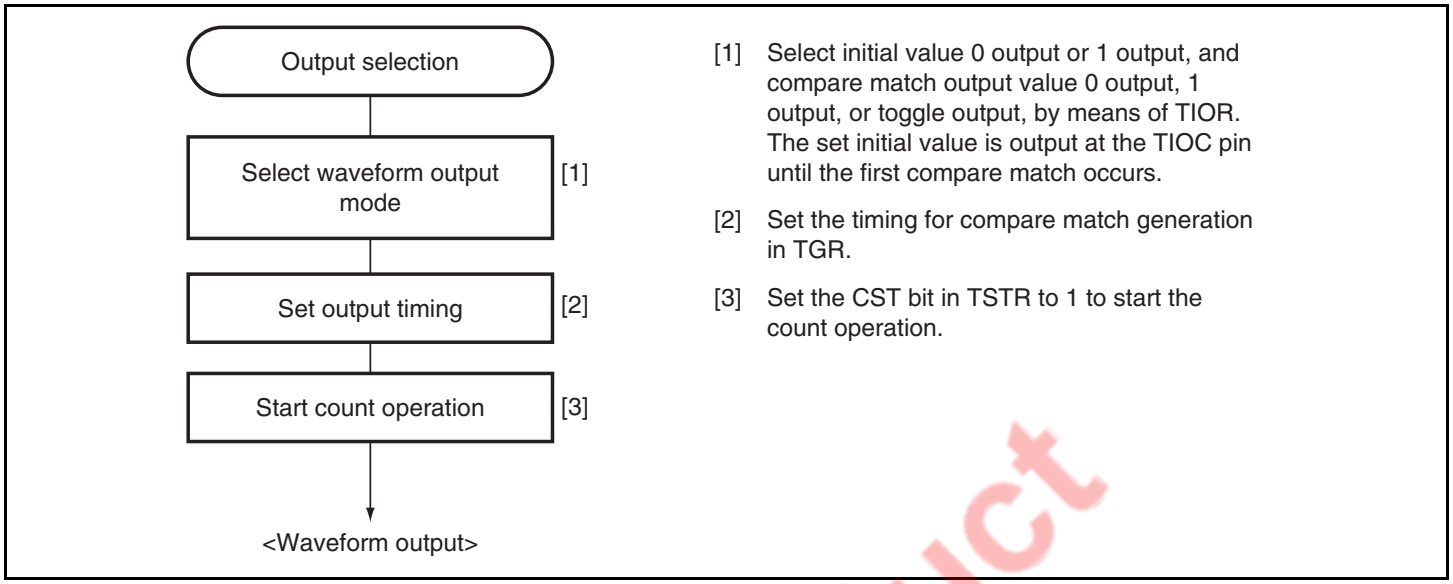


Figure 18.6 Example of Setting Procedure for Waveform Output by Compare Match

Examples of Waveform Output Operation: Figure 18.7 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 such that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

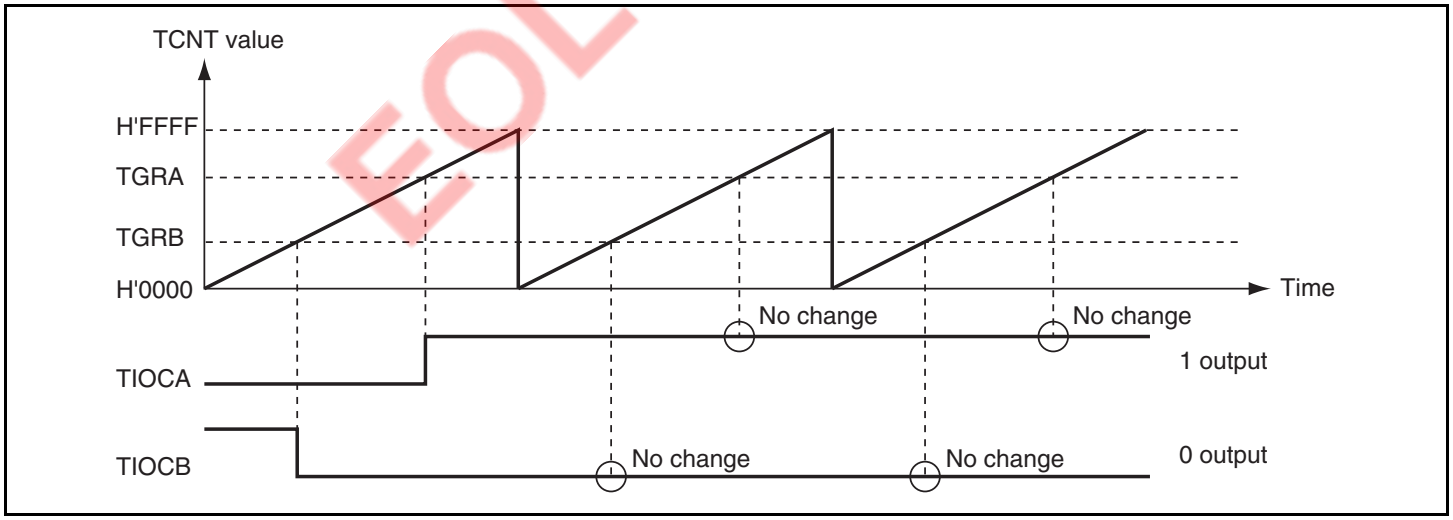


Figure 18.7 Example of 0 Output/1 Output Operation

Figure 18.8 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made such that the output is toggled by both compare match A and compare match B.

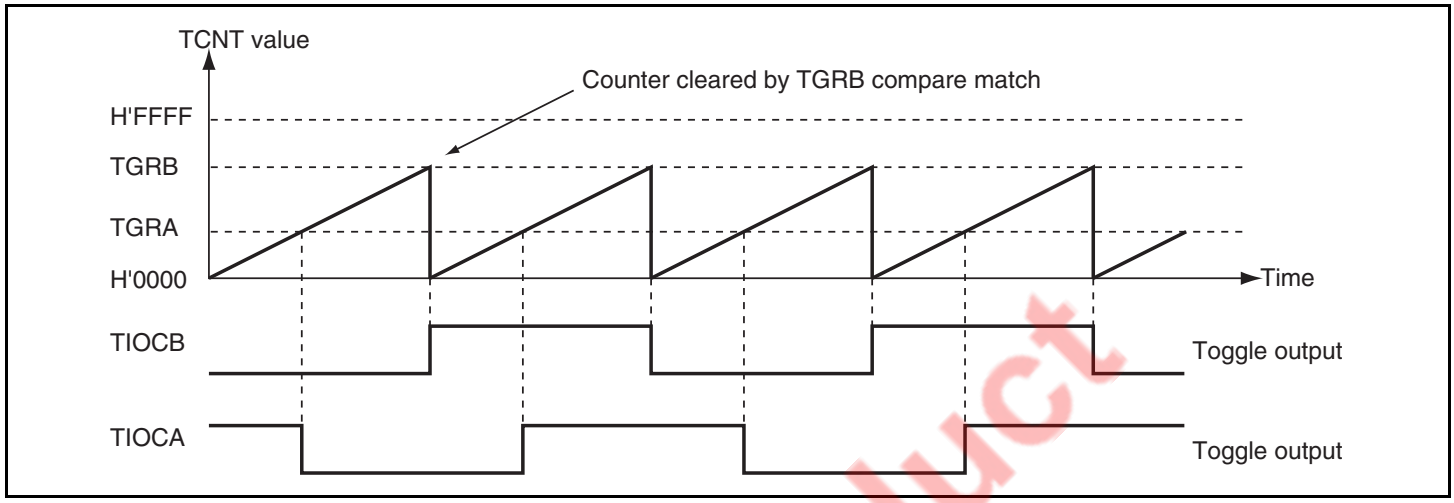


Figure 18.8 Example of Toggle Output Operation

- Input Capture Function

The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge. For channels 0 and 1, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 1, 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.

Example of Input Capture Operation Setting Procedure: Figure 18.9 shows an example of the input capture operation setting procedure.

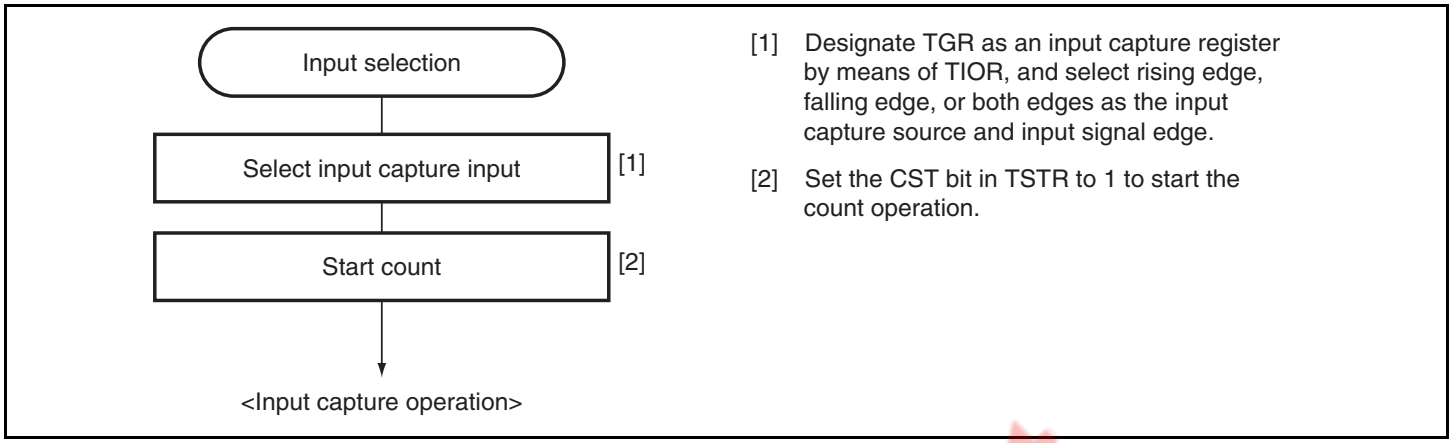


Figure 18.9 Example of Input Capture Operation Setting Procedure

Example of Input Capture Operation: Figure 18.10 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, the 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.

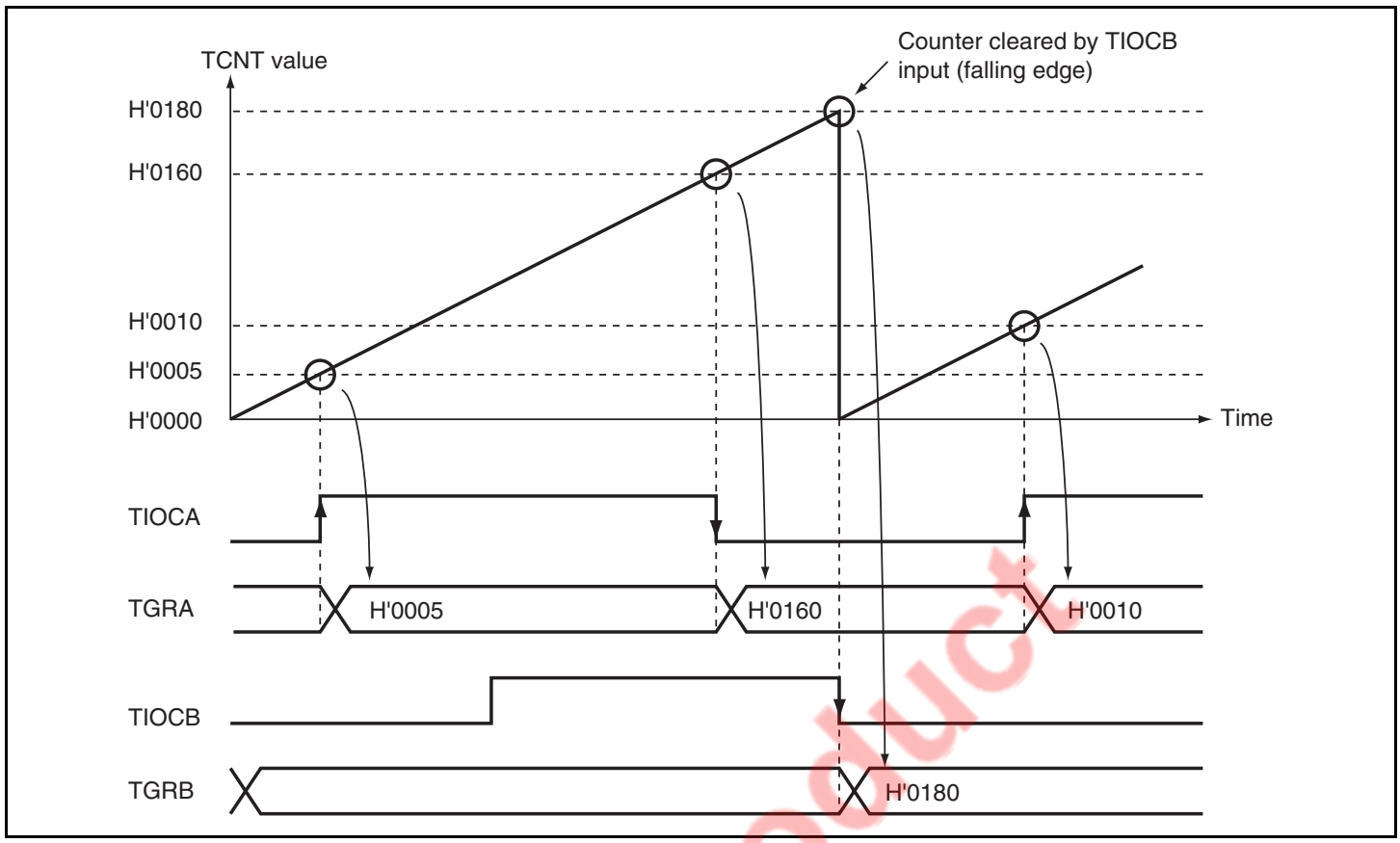


Figure 18.10 Example of Input Capture Operation

18.4.2 Synchronous Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 4 can all be designated for synchronous operation.

Example of Synchronous Operation Setting Procedure: Figure 18.11 shows an example of the synchronous operation setting procedure.

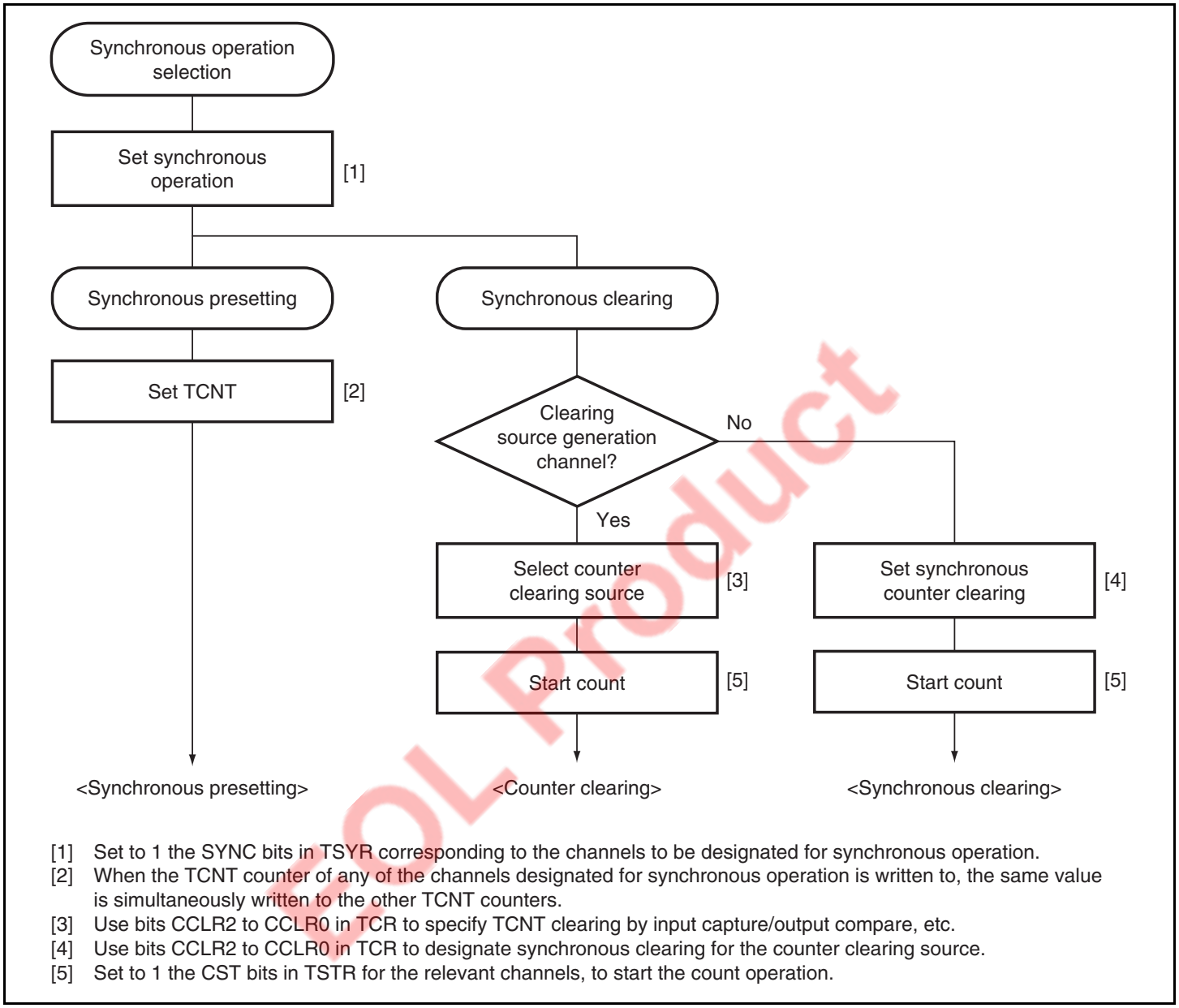


Figure 18.11 Example of Synchronous Operation Setting Procedure

Example of Synchronous Operation: Figure 18.12 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 TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGRB_0 compare match, are performed for channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the PWM cycle.

For details of PWM modes, see section 18.4.5, PWM Modes.

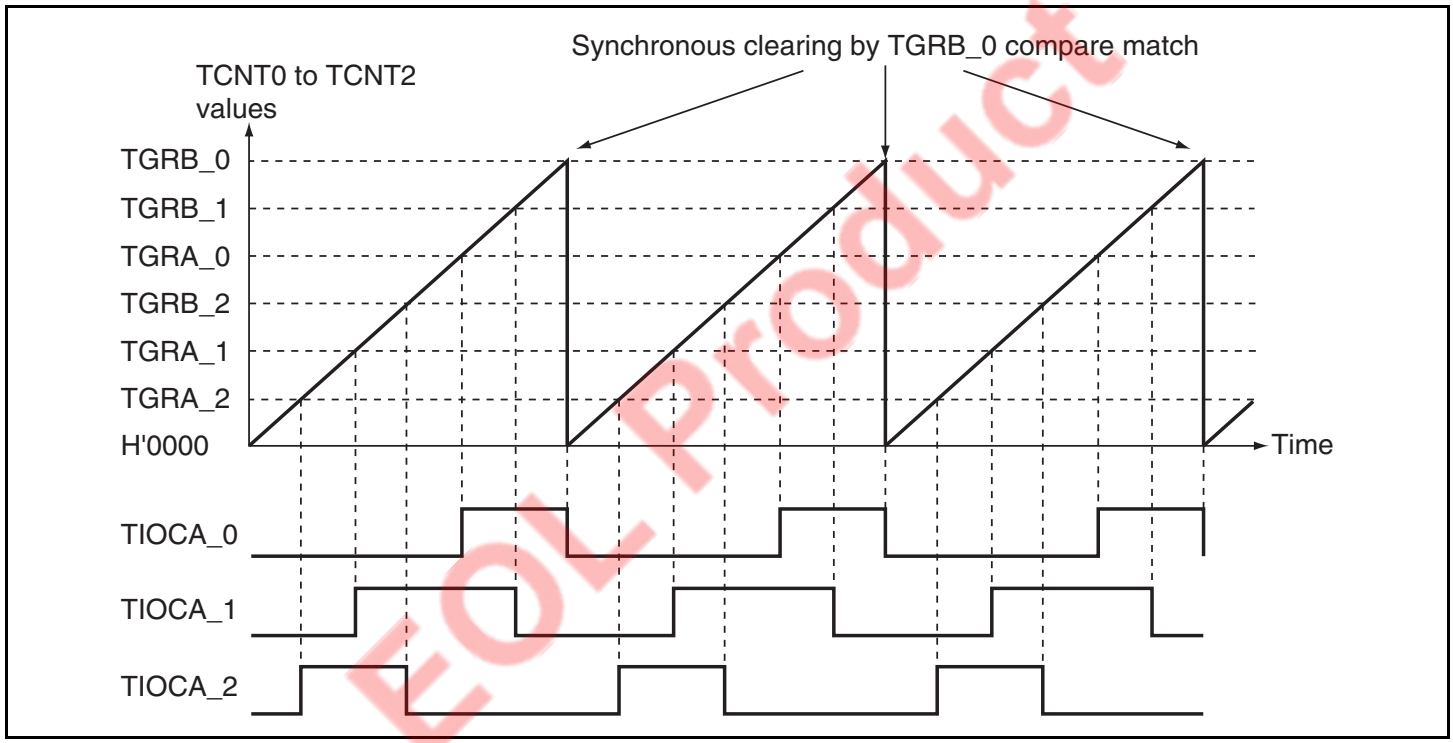


Figure 18.12 Example of Synchronous Operation

18.4.3 Buffer Operation

Buffer operation, provided for channels 0, 3, and 4, 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 as a compare match register.

Table 18.29 shows the register combinations used in buffer operation.

Table 18.29 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3
4	TGRA_4	TGRC_4
	TGRB_4	TGRD_4

- 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 18.13.

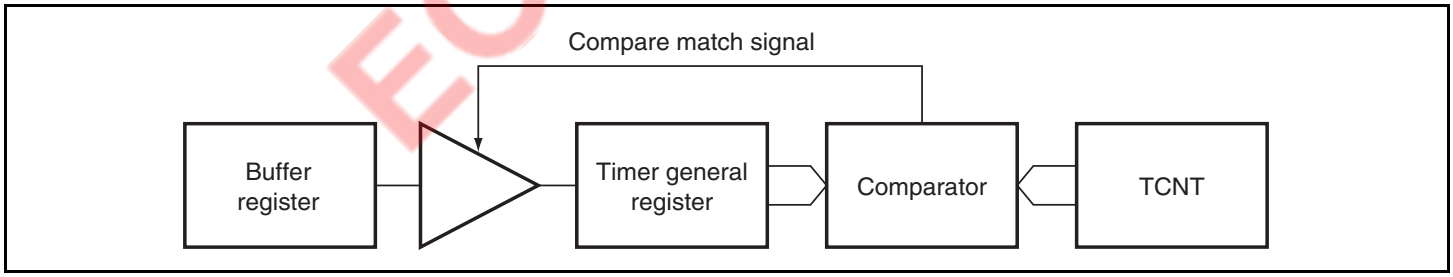


Figure 18.13 Compare Match Buffer Operation

- 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 18.14.

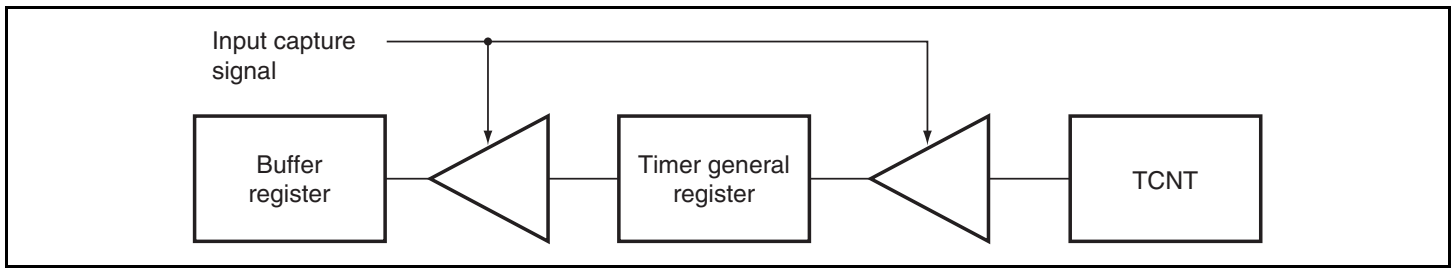


Figure 18.14 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 18.15 shows an example of the buffer operation setting procedure.

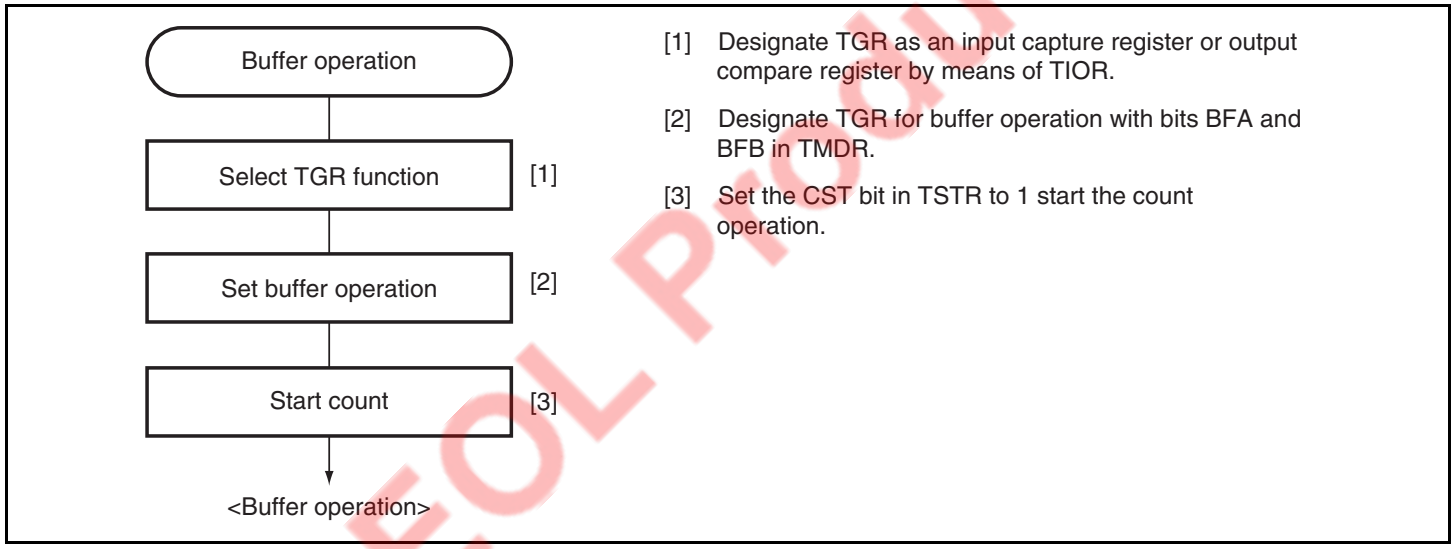


Figure 18.15 Example of Buffer Operation Setting Procedure

Examples of Buffer Operation:

- When TGR is an output compare register

  Figure 18.16 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 that compare match A occurs.

  For details of PWM modes, see section 18.4.5, PWM Modes.

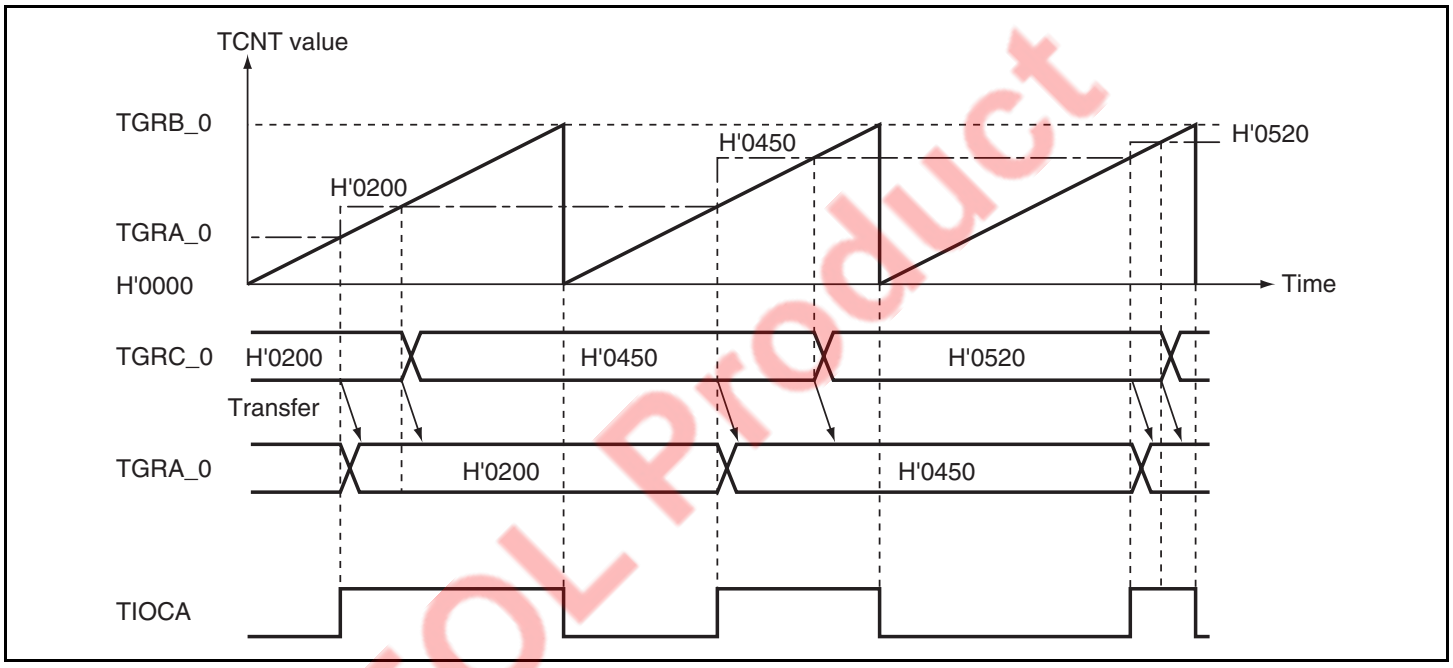


Figure 18.16 Example of Buffer Operation (1)

- When TGR is an input capture register

  Figure 18.17 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 the occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

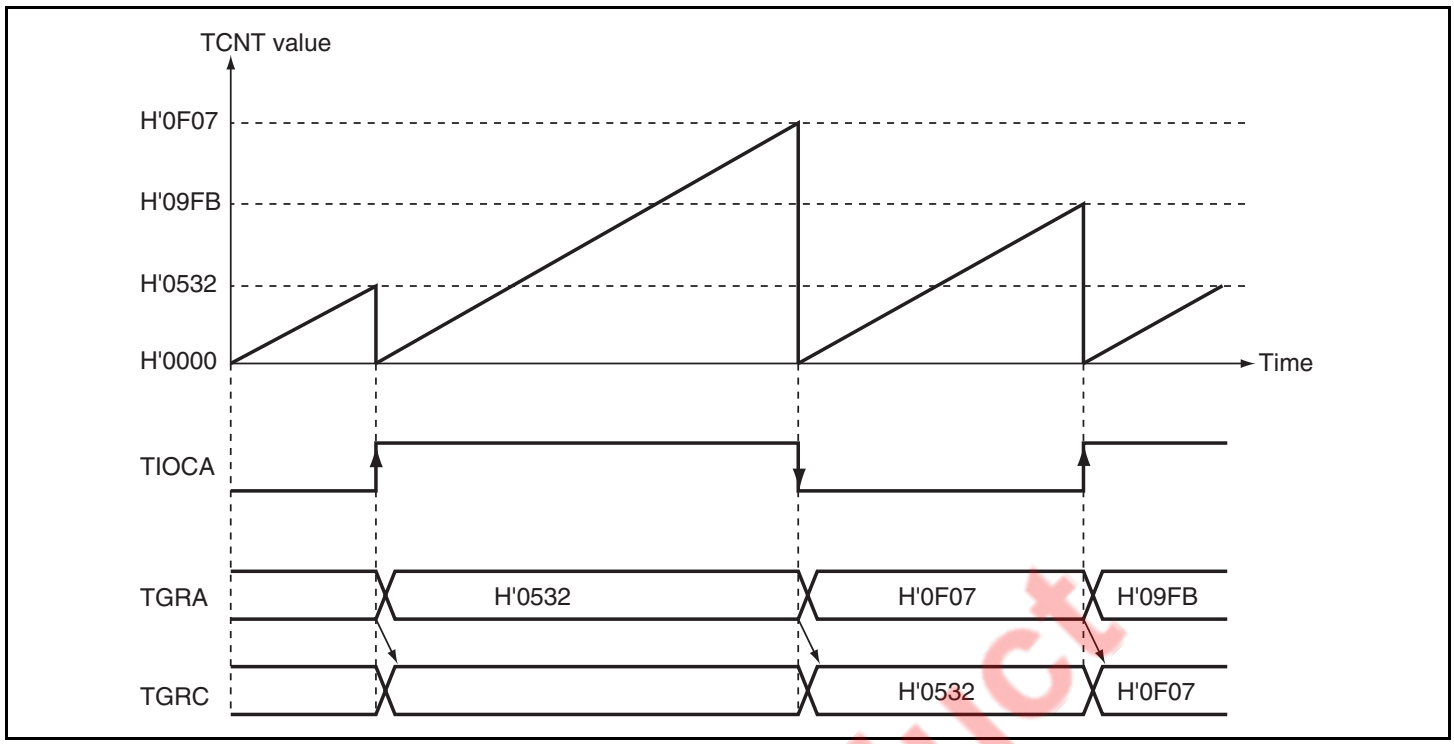


Figure 18.17 Example of Buffer Operation (2)

18.4.4 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 counter clock upon overflow/underflow of TCNT_2 as set in bits TPSC0 to TPSC2 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase-counting mode.

Table 18.30 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 counters operates independently in phase counting mode.

Table 18.30 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2

Example of Cascaded Operation Setting Procedure: Figure 18.18 shows an example of the setting procedure for cascaded operation.

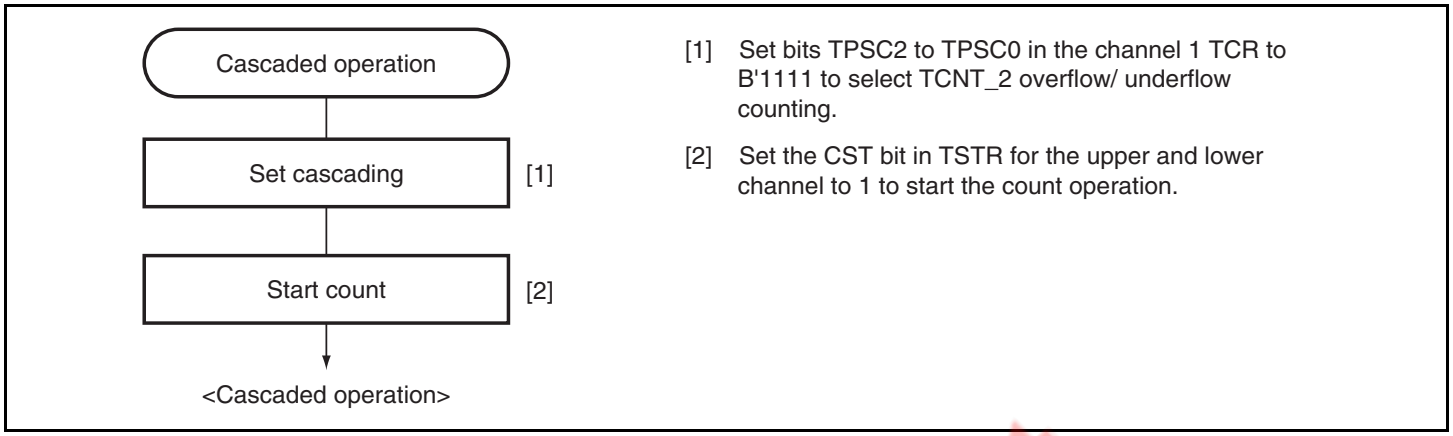


Figure 18.18 Cascaded Operation Setting Procedure

Examples of Cascaded Operation: Figure 18.19 illustrates the operation when TCNT_2 overflow/underflow counting 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.

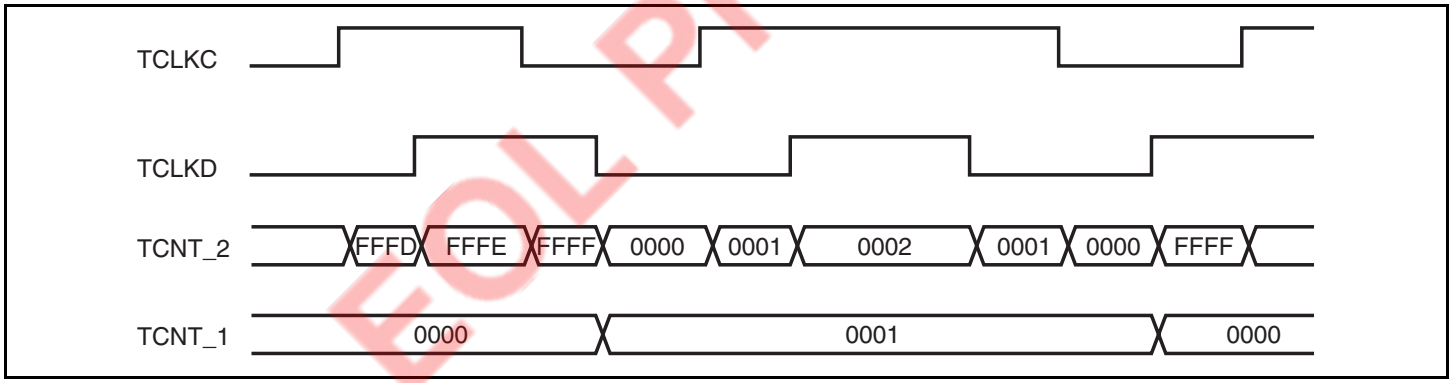


Figure 18.19 Example of Cascaded Operation

18.4.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. The output level can be selected as 0, 1, or toggle output in response to a compare match of each TGR.

TGR registers settings can be used to output a PWM waveform in the range of 0% to 100% duty.

Designating TGR compare match as the counter clearing source enables the period 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 output specified by bits IOA0 to IOA3 and IOC0 to IOC3 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by bits IOB0 to IOB3 and IOD0 to IOD3 in TIOR is output at compare matches B and D. 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 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 registers are identical, the output value does not change when a compare match occurs.

  In PWM mode 2, a maximum 8-phase PWM output is possible in combination use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 18.31.

Table 18.31 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGRA_0	TIOC0A	TIOC0A
	TGRB_0		TIOC0B
	TGRC_0	TIOC0C	TIOC0C
	TGRD_0		TIOC0D
1	TGRA_1	TIOC1A	TIOC1A
	TGRB_1		TIOC1B
2	TGRA_2	TIOC2A	TIOC2A
	TGRB_2		TIOC2B
3	TGRA_3	TIOC3A	Setting prohibited
	TGRB_3		Setting prohibited
	TGRC_3	TIOC3C	Setting prohibited
	TGRD_3		Setting prohibited
4	TGRA_4	TIOC4A	Setting prohibited
	TGRB_4		Setting prohibited
	TGRC_4	TIOC4C	Setting prohibited
	TGRD_4		Setting prohibited

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

Example of PWM Mode Setting Procedure: Figure 18.20 shows an example of the PWM mode setting procedure.

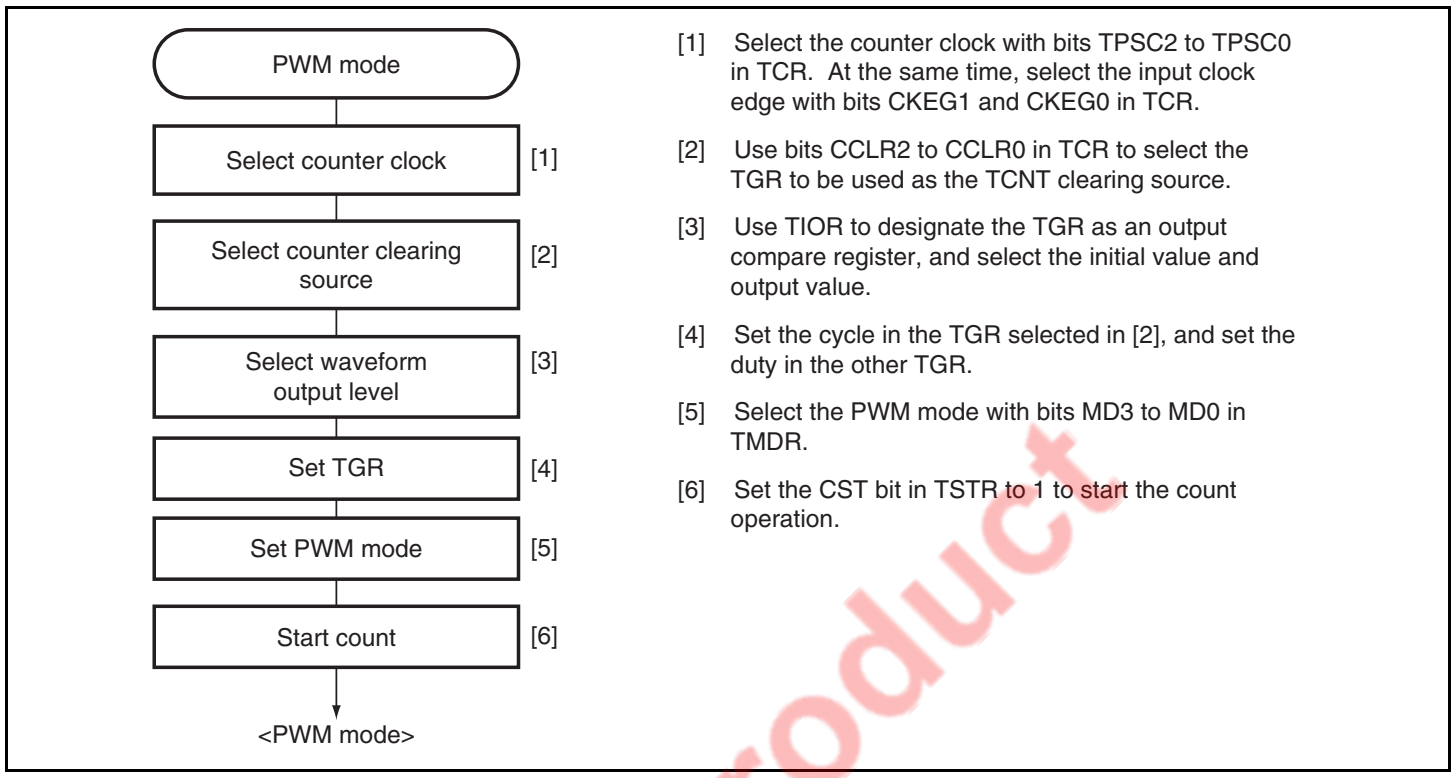


Figure 18.20 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation: Figure 18.21 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in the TGRB registers are used as the duty cycle.

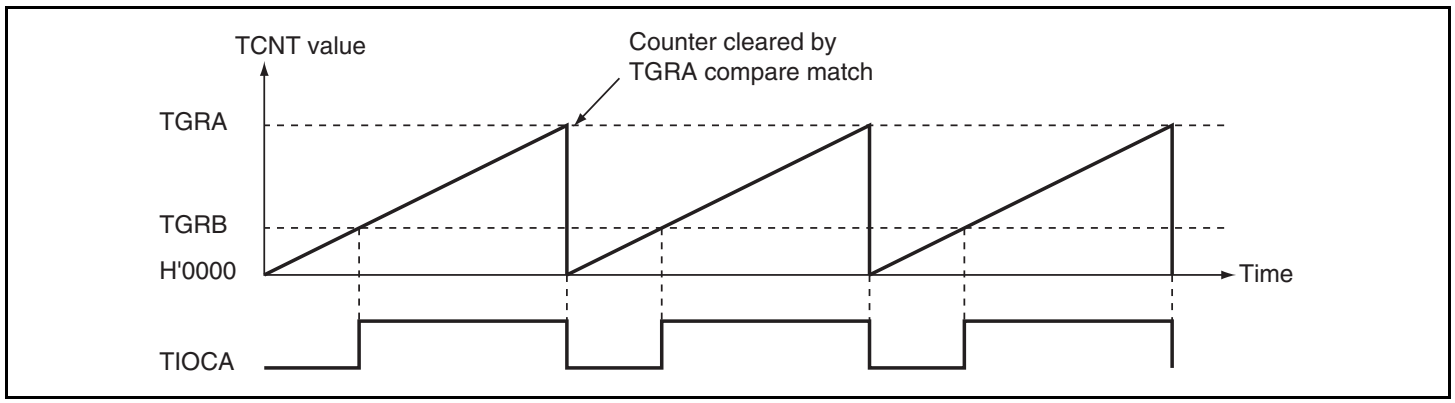


Figure 18.21 Example of PWM Mode Operation (1)

Figure 18.22 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), outputting 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 are used as the duty levels.

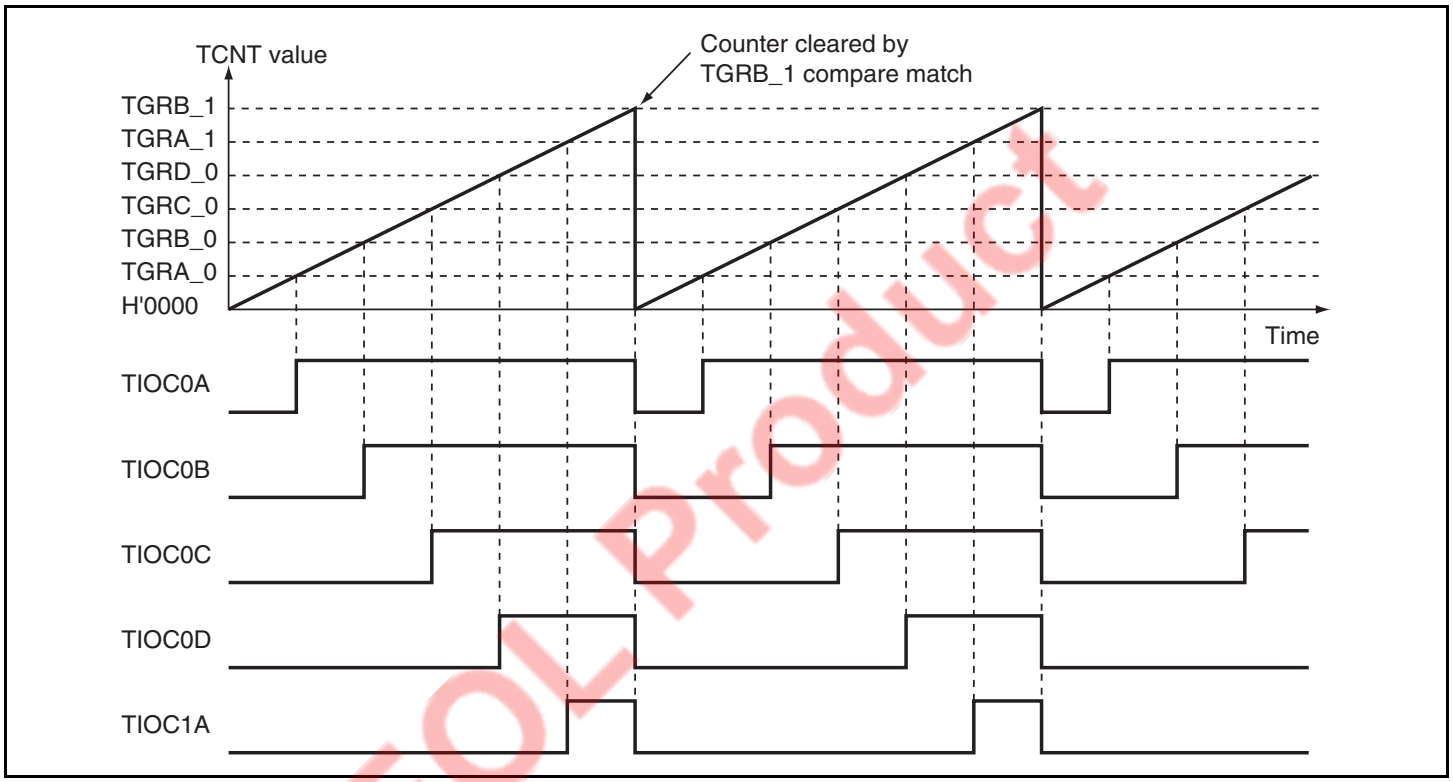


Figure 18.22 Example of PWM Mode Operation (2)

Figure 18.23 shows examples of PWM waveform output with 0% duty cycle and 100% duty cycle in PWM mode.

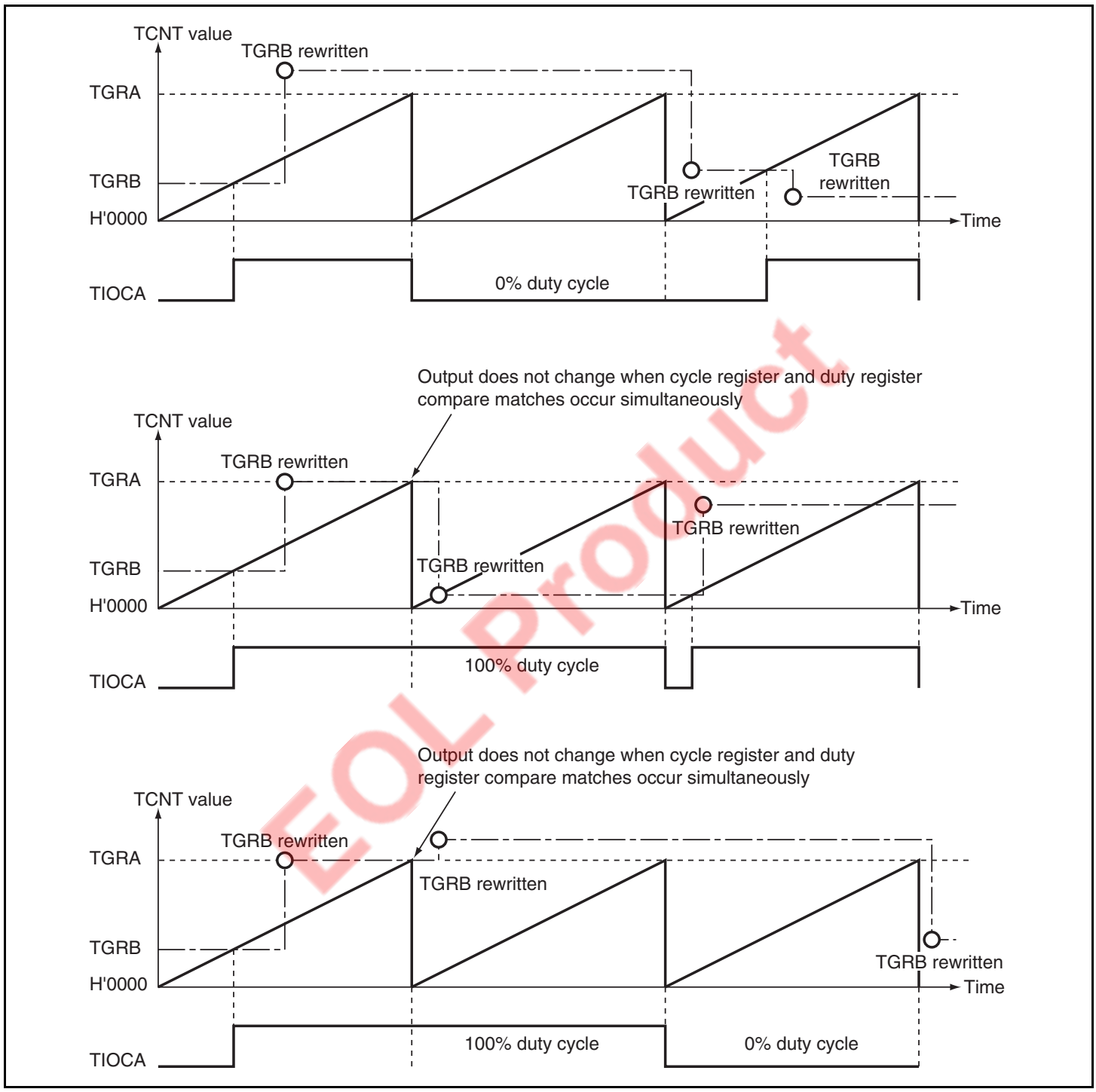


Figure 18.23 Example of PWM Mode Operation (3)

18.4.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT counts up or down accordingly. This mode can be set for channels 1 and 2.

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 TPSC0 to TPSC2 and bits CKEG0 and CKEG1 in TCR. However, the functions of bits CCLR0 and CCLR1 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.

If overflow occurs when TCNT is counting up, the TCFV flag in TSR is set; if underflow occurs when TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag reveals whether TCNT is counting up or down.

Table 18.32 shows the correspondence between external clock pins and channels.

Table 18.32 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 is set to phase counting mode	TCLKA	TCLKB
When channel 2 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure: Figure 18.24 shows an example of the phase counting mode setting procedure.

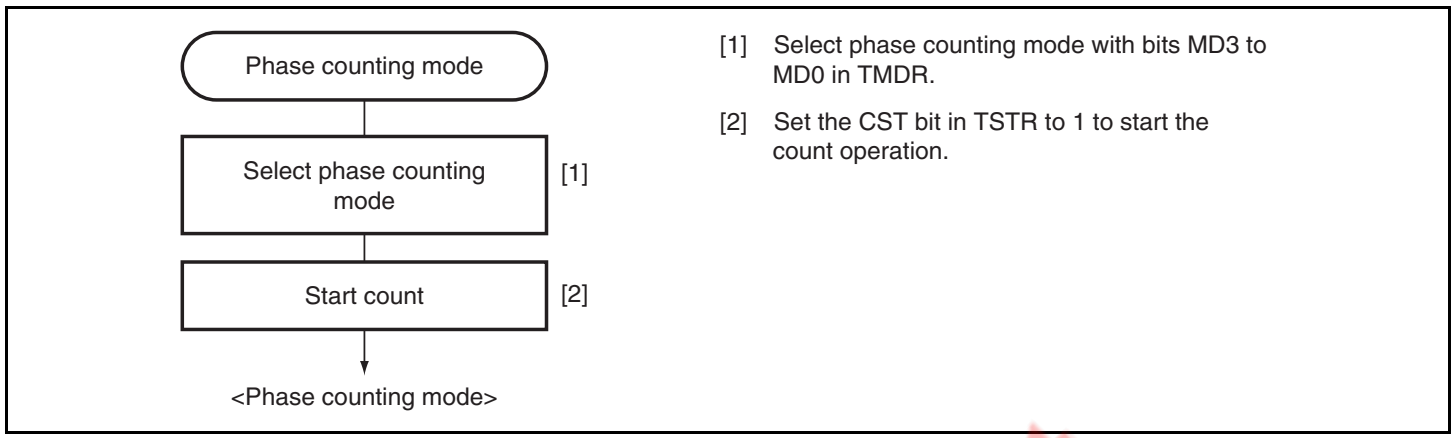


Figure 18.24 Example of Phase Counting Mode Setting Procedure

Examples of Phase Counting Mode Operation: In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

- Phase counting mode 1

  Figure 18.25 shows an example of phase counting mode 1 operation, and table 18.33 summarizes the TCNT up/down-count conditions.

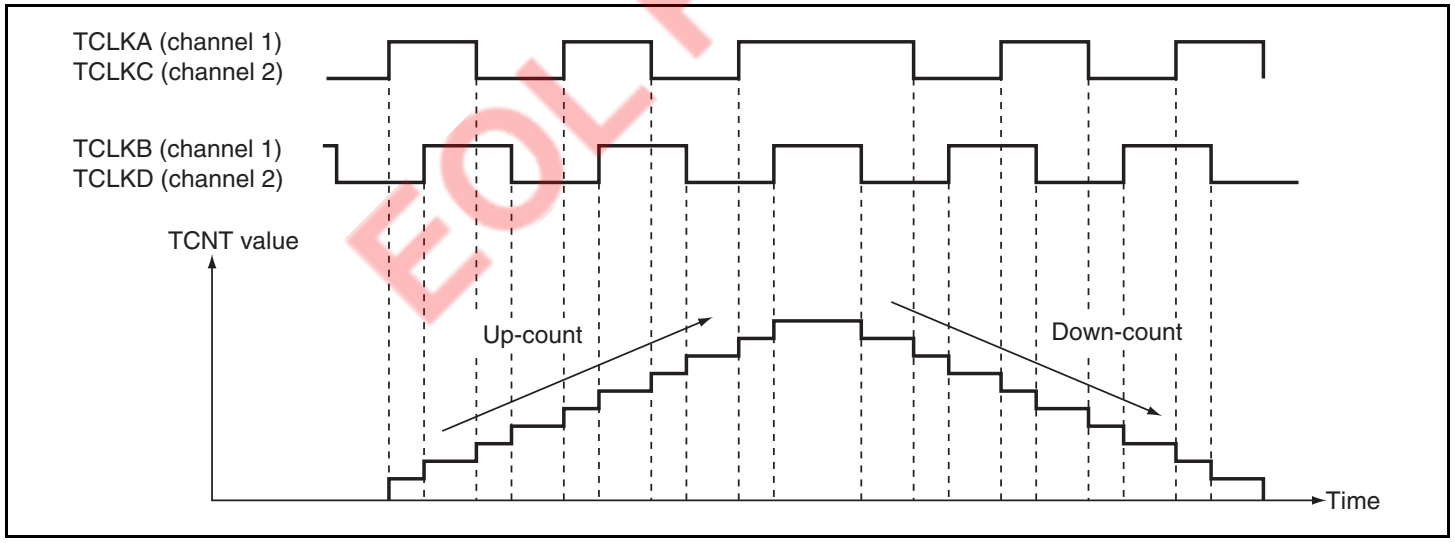


Figure 18.25 Example of Phase Counting Mode 1 Operation

Table 18.33 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	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

- Phase counting mode 2

  Figure 18.26 shows an example of phase counting mode 2 operation, and table 18.34 summarizes the TCNT up/down-count conditions.

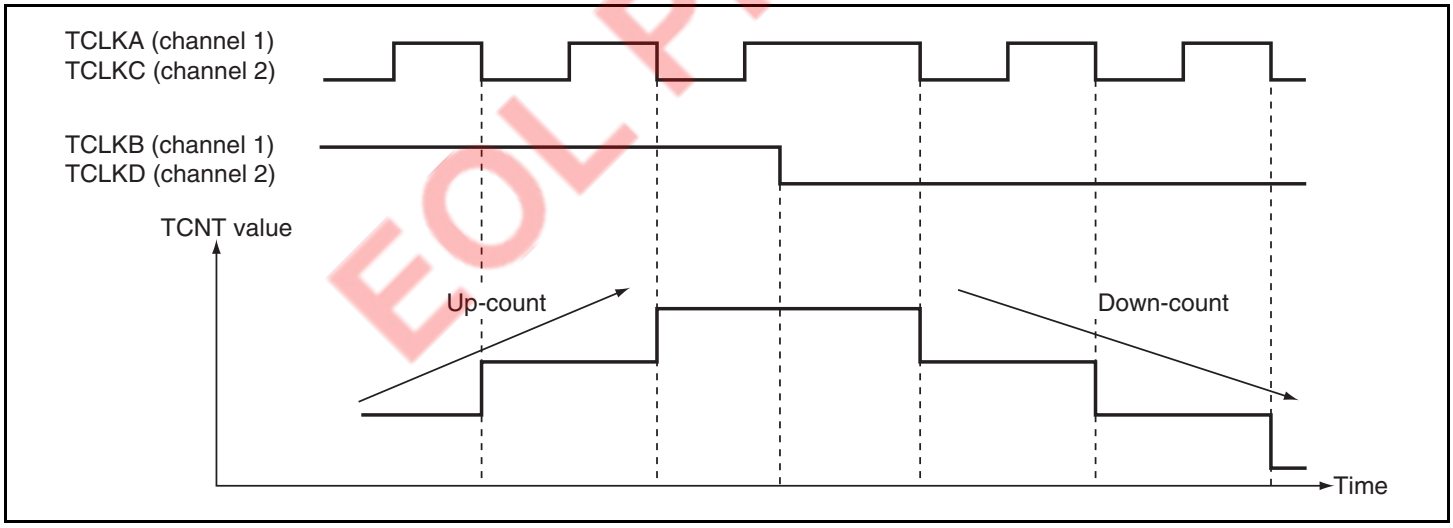


Figure 18.26 Example of Phase Counting Mode 2 Operation

Table 18.34 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	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

- Phase counting mode 3

  Figure 18.27 shows an example of phase counting mode 3 operation, and table 18.35 summarizes the TCNT up/down-count conditions.

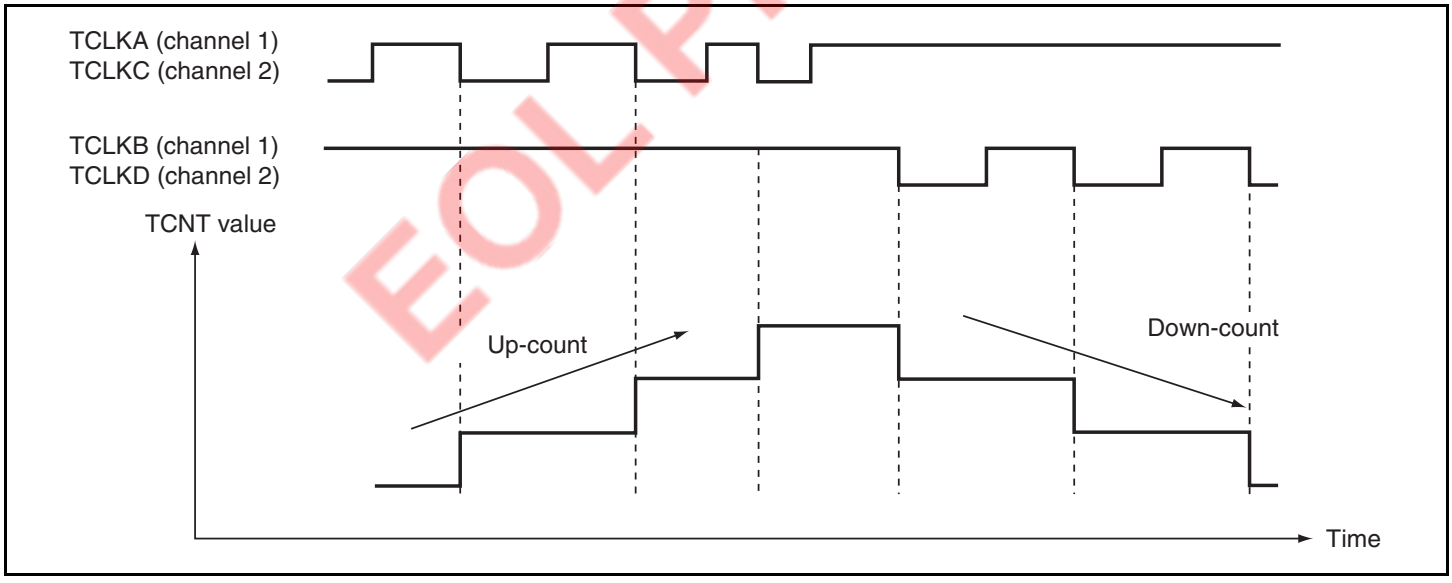


Figure 18.27 Example of Phase Counting Mode 3 Operation

Table 18.35 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	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

- Phase counting mode 4

  Figure 18.28 shows an example of phase counting mode 4 operation, and table 18.36 summarizes the TCNT up/down-count conditions.

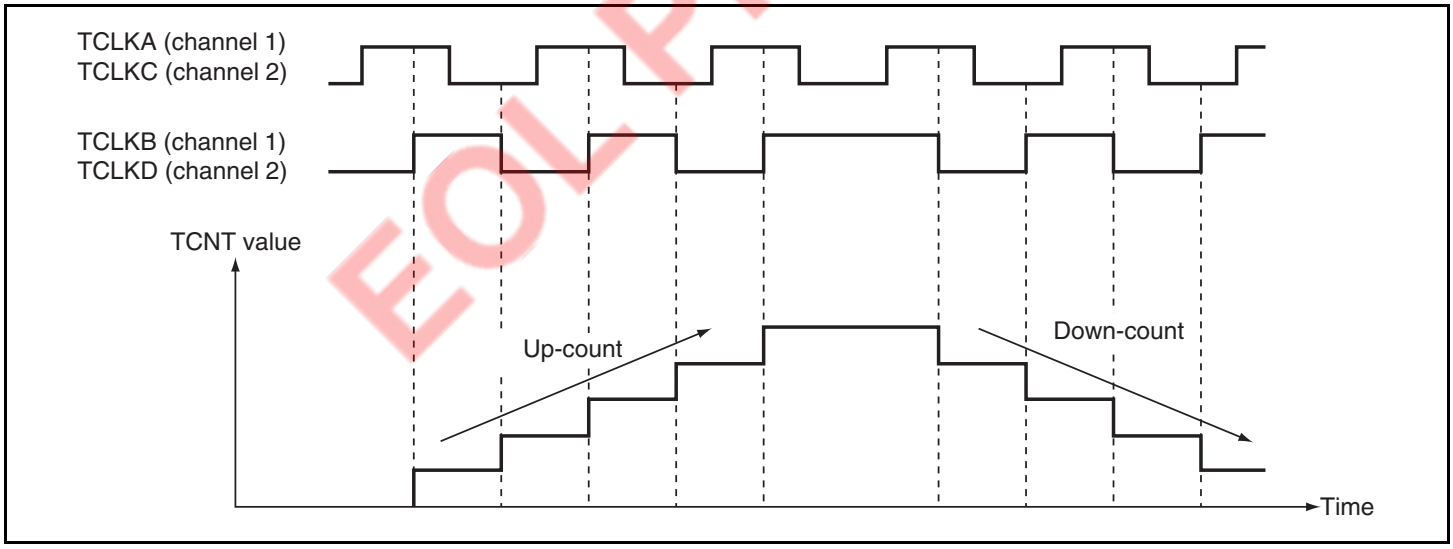


Figure 18.28 Example of Phase Counting Mode 4 Operation

Table 18.36 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	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

Phase Counting Mode Application Example: Figure 18.29 shows an example in which channel 1 is in phase counting mode, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect 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 period and position control period. 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 the pulse widths of 2-phase encoder 4-multiplication pulses are detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, and channel 0 TGRA_0 and TGRC_0 compare matches are selected as the input capture source and store the up/down-counter values for the control periods.

This procedure enables the accurate detection of position and speed.

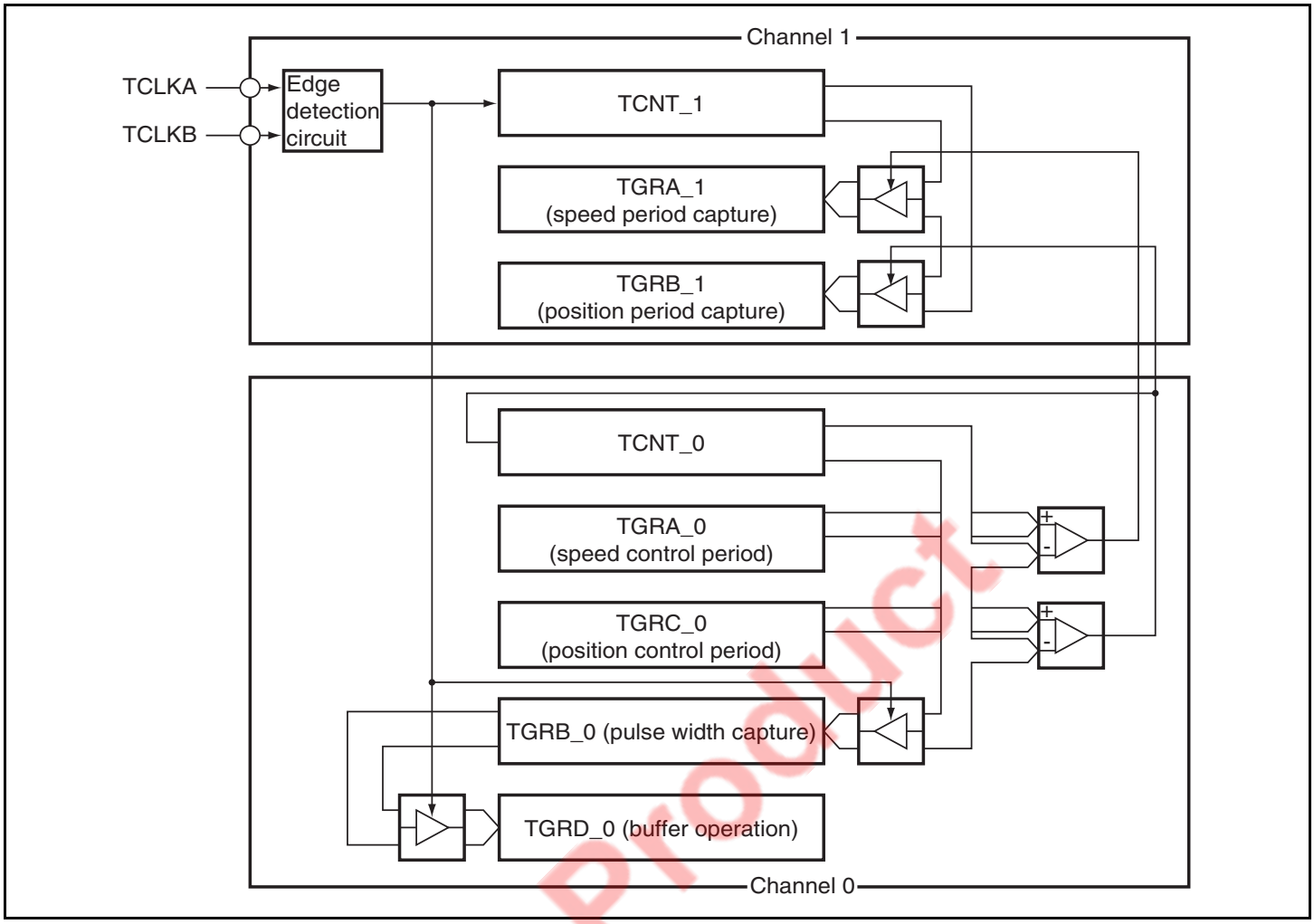


Figure 18.29 Phase Counting Mode Application Example

18.4.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, three-phase output of positive and negative PWM waveforms that share a common wave transition point can be obtained by combining channels 3 and 4.

When set for reset-synchronized PWM mode, the TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, and TIOC4D pins function as PWM output pins and TCNT3 functions as an upcounter.

Table 18.37 shows the PWM output pins used. Table 18.38 shows the settings of the registers.

Table 18.37 Output Pins for Reset-Synchronized PWM Mode

Channel	Output Pin	Description
3	TIOC3B	PWM output pin 1
	TIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
4	TIOC4A	PWM output pin 2
	TIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	TIOC4B	PWM output pin 3
	TIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)

Table 18.38 Register Settings for Reset-Synchronized PWM Mode

Register	Description of Setting
TCNT_3	Initial setting of H'0000
TCNT_4	Initial setting of H'0000
TGRA_3	Set count cycle for TCNT_3
TGRB_3	Sets the turning point for PWM waveform output by the TIOC3B and TIOC3D pins
TGRA_4	Sets the turning point for PWM waveform output by the TIOC4A and TIOC4C pins
TGRB_4	Sets the turning point for PWM waveform output by the TIOC4B and TIOC4D pins

Procedure for Selecting the Reset-Synchronized PWM Mode: Figure 18.30 shows an example of procedure for selecting the reset synchronized PWM mode.

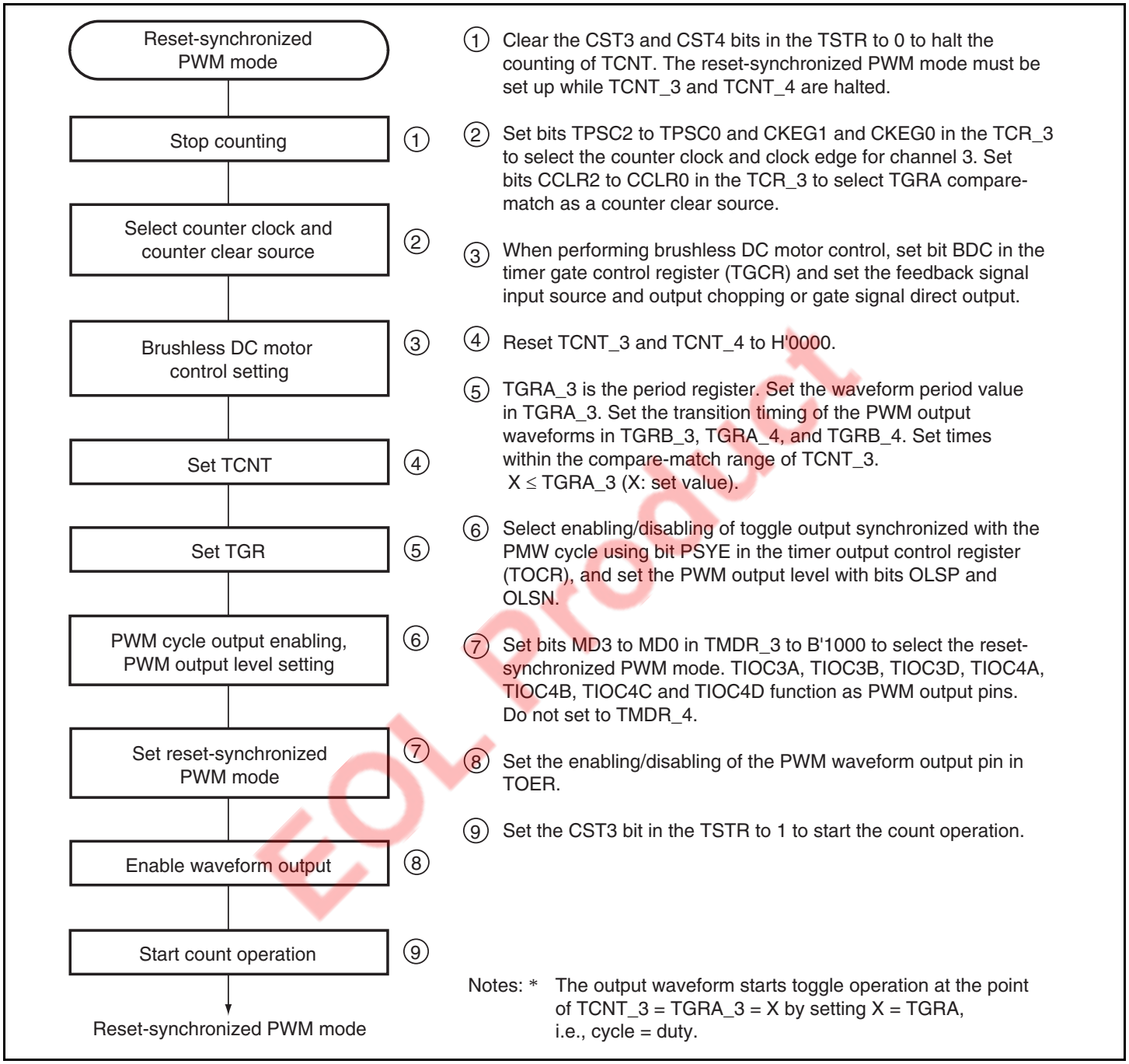


Figure 18.30 Procedure for Selecting the Reset-Synchronized PWM Mode

Reset-Synchronized PWM Mode Operation: Figure 18.31 shows an example of operation in the reset-synchronized PWM mode. TCNT_3 and TCNT_4 operate as upcounters. The counter is cleared when a TCNT_3 and TGRA_3 compare-match occurs, and then begins counting up from H'0000. The PWM output pin output toggles with each occurrence of a TGRB_3, TGRA_4, TGRB_4 compare-match, and upon counter clears.

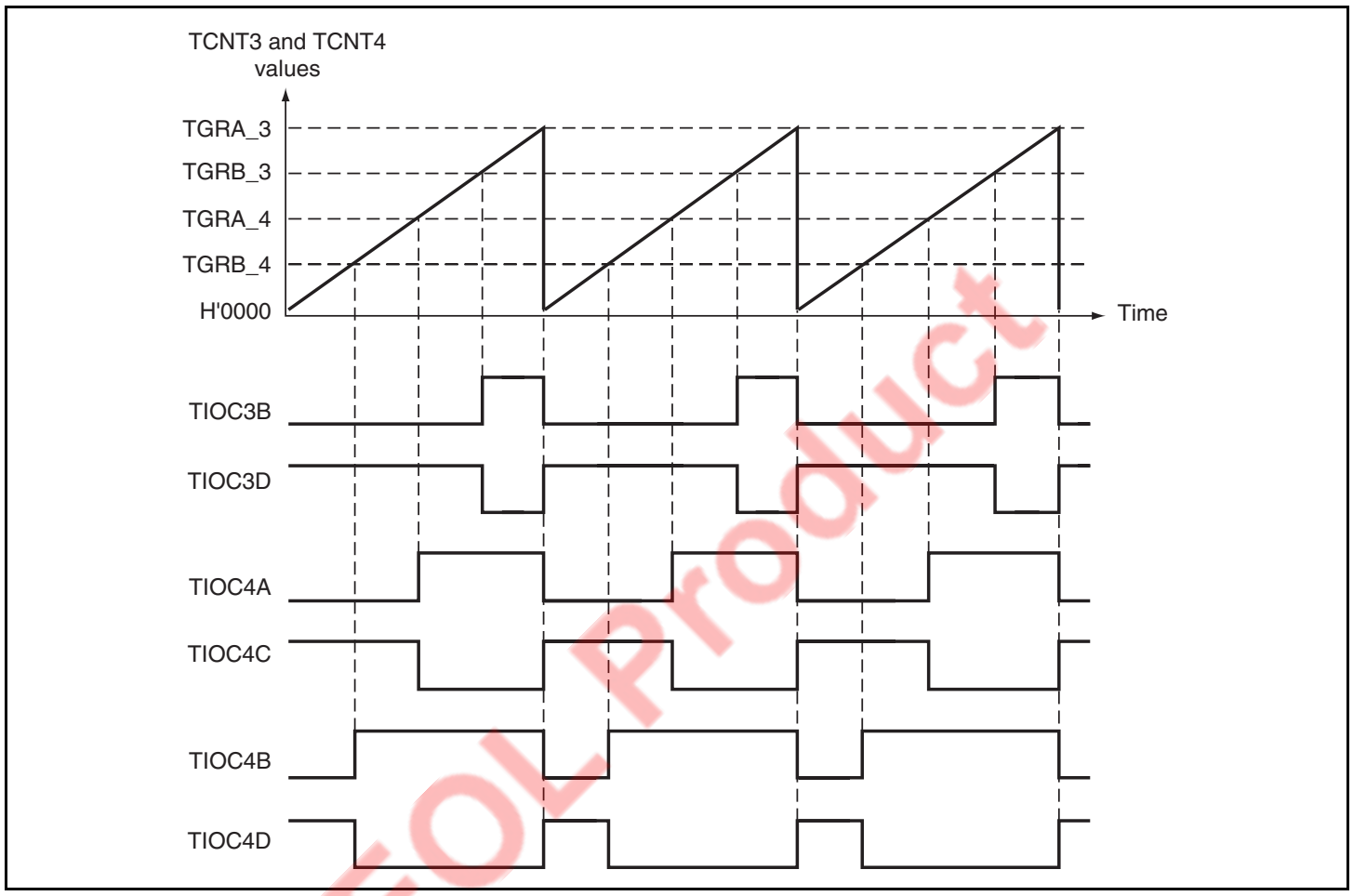


Figure 18.31 Reset-Synchronized PWM Mode Operation Example (When the TOCR's OLSN = 1 and OLSP = 1)

18.4.8 Complementary PWM Mode

In the complementary PWM mode, three-phase output of non-overlapping positive and negative PWM waveforms can be obtained by combining channels 3 and 4.

In complementary PWM mode, TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D pins function as PWM output pins, the TIOC3A pin can be set for toggle output synchronized with the PWM period. TCNT_3 and TCNT_4 function as increment/decrement counters.

Table 18.39 shows the PWM output pins used. Table 18.40 shows the settings of the registers used.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 18.39 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
3	TIOC3B	PWM output pin 1
	TIOC3D	PWM output pin 1' (non-overlapping negative-phase waveform of PWM output 1)
4	TIOC4A	PWM output pin 2
	TIOC4B	PWM output pin 3
	TIOC4C	PWM output pin 2' (non-overlapping negative-phase waveform of PWM output 2)
	TIOC4D	PWM output pin 3' (non-overlapping negative-phase waveform of PWM output 3)

Table 18.40 Register Settings for Complementary PWM Mode

Channel	Counter/Register	Description	Read/Write from CPU
3	TCNT_3	Start of up-count from value set in dead time register	Maskable by PTE/PEMTURWE setting*
	TGRA_3	Set TCNT_3 upper limit value (1/2 carrier cycle + dead time)	Maskable by PTE/PEMTURWE setting*
	TGRB_3	PWM output 1 compare register	Maskable by PTE/PEMTURWE setting*
	TGRC_3	TGRA_3 buffer register	Always readable/writable
	TGRD_3	PWM output 1/TGRB_3 buffer register	Always readable/writable
4	TCNT_4	Up-count start, initialized to H'0000	Maskable by PTE/PEMTURWE setting*
	TGRA_4	PWM output 2 compare register	Maskable by PTE/PEMTURWE setting*
	TGRB_4	PWM output 3 compare register	Maskable by PTE/PEMTURWE setting*
	TGRC_4	PWM output 2/TGRA_4 buffer register	Always readable/writable
	TGRD_4	PWM output 3/TGRB_4 buffer register	Always readable/writable
Timer dead time data register (TDDR)		Set TCNT_4 and TCNT_3 offset value (dead time value)	Maskable by PTE/PEMTURWE setting*
Timer cycle data register (TCDR)		Set TCNT_4 upper limit value (1/2 carrier cycle)	Maskable by PTE/PEMTURWE setting*
Timer cycle buffer register (TCBR)		TCDR buffer register	Always readable/writable
Subcounter (TCNTS)		Subcounter for dead time generation	Read-only
Temporary register 1 (TEMP1)		PWM output 1/TGRB_3 temporary register	Not readable/writable
Temporary register 2 (TEMP2)		PWM output 2/TGRA_4 temporary register	Not readable/writable
Temporary register 3 (TEMP3)		PWM output 3/TGRB_4 temporary register	Not readable/writable

Note: * Access can be enabled or disabled according to the setting of bit 0 (MTURWE) in PTE/PEMTURWE (port E/port E MTU R/W enable register).

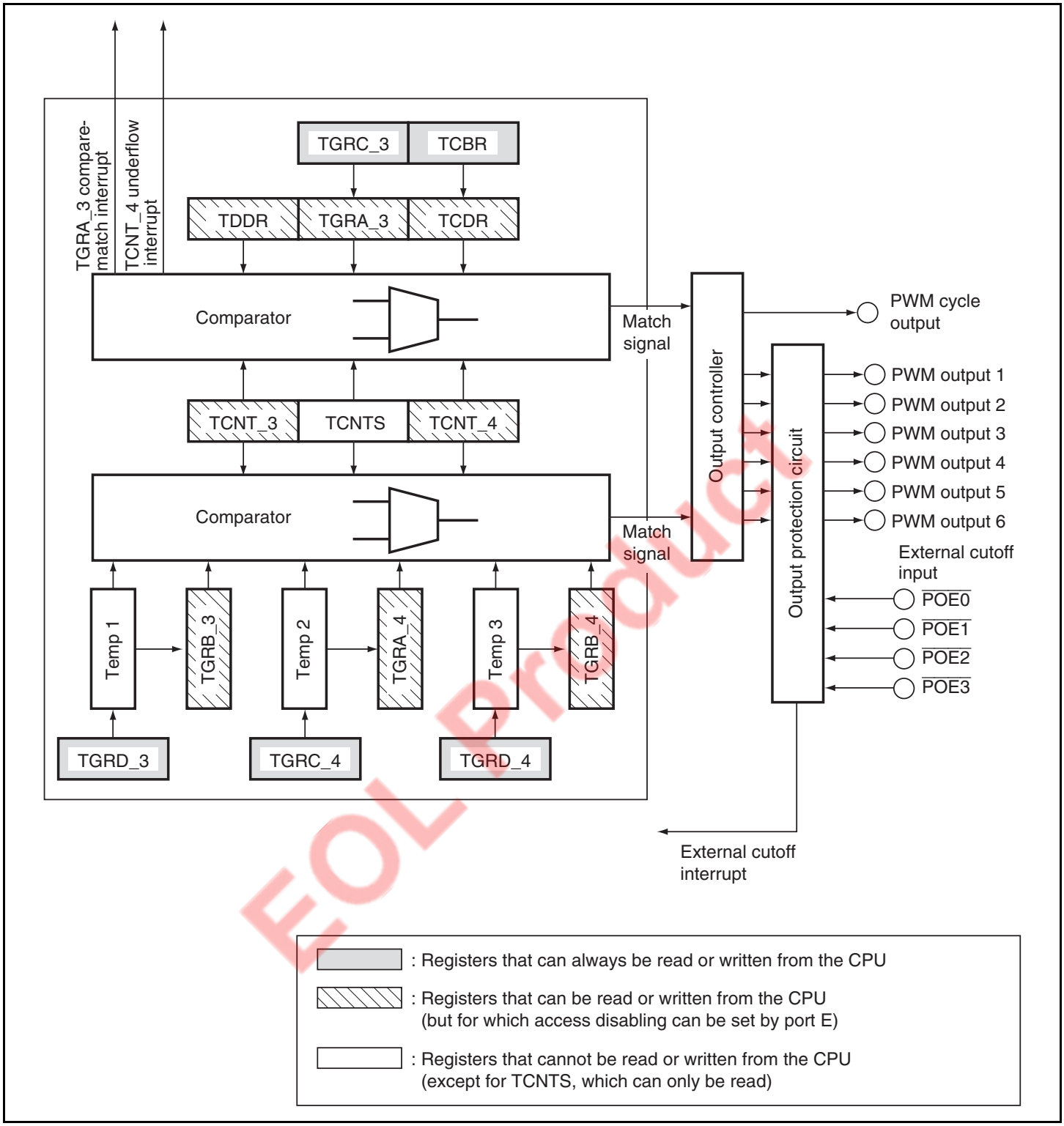


Figure 18.32 Block Diagram of Channels 3 and 4 in Complementary PWM Mode

Example of Complementary PWM Mode Setting Procedure

An example of the complementary PWM mode setting procedure is shown in figure 18.33.

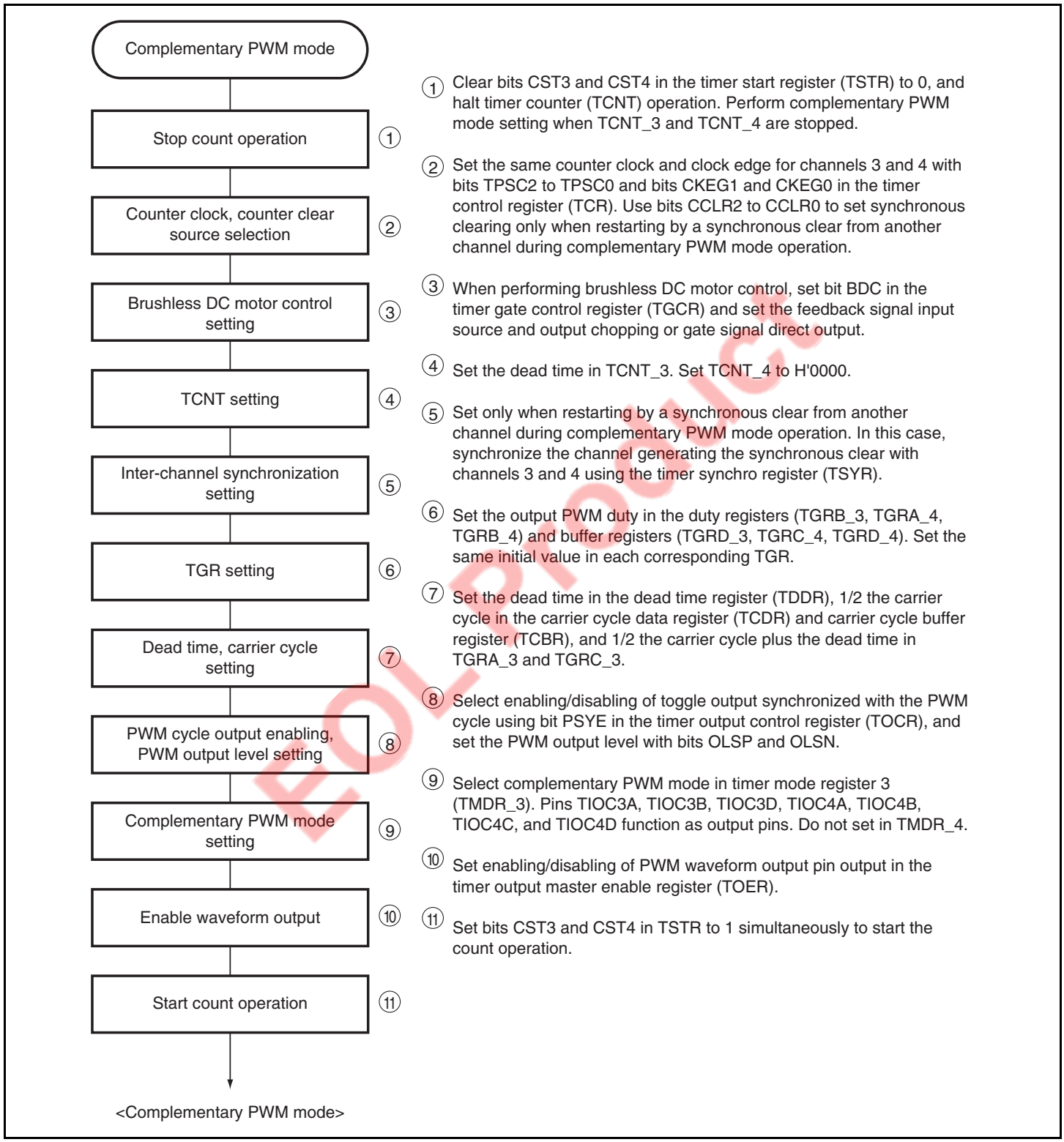


Figure 18.33 Example of Complementary PWM Mode Setting Procedure

Outline of Complementary PWM Mode Operation

In complementary PWM mode, 6-phase PWM output is possible. Figure 18.34 illustrates counter operation in complementary PWM mode, and figure 18.35 shows an example of complementary PWM mode operation.

Counter Operation: In complementary PWM mode, three counters−TCNT_3, TCNT_4, and TCNTS−perform up/down-count operations.

TCNT_3 is automatically initialized to the value set in TDDR when complementary PWM mode is selected and the CST bit in TSTR is 0.

When the CST bit is set to 1, TCNT_3 counts up to the value set in TGRA_3, then switches to down-counting when it matches TGRA_3. When the TCNT3 value matches TDDR, the counter switches to up-counting, and the operation is repeated in this way.

TCNT_4 is initialized to H'0000.

When the CST bit is set to 1, TCNT4 counts up in synchronization with TCNT_3, and switches to down-counting when it matches TCDR. On reaching H'0000, TCNT4 switches to up-counting, and the operation is repeated in this way.

TCNTS is a read-only counter. It need not be initialized.

When TCNT_3 matches TCDR during TCNT_3 and TCNT_4 up/down-counting, down-counting is started, and when TCNTS matches TCDR, the operation switches to up-counting. When TCNTS matches TGRA_3, it is cleared to H'0000.

When TCNT_4 matches TDDR during TCNT_3 and TCNT_4 down-counting, up-counting is started, and when TCNTS matches TDDR, the operation switches to down-counting. When TCNTS reaches H'0000, it is set with the value in TGRA_3.

TCNTS is compared with the compare register and temporary register in which the PWM duty is set during the count operation only.

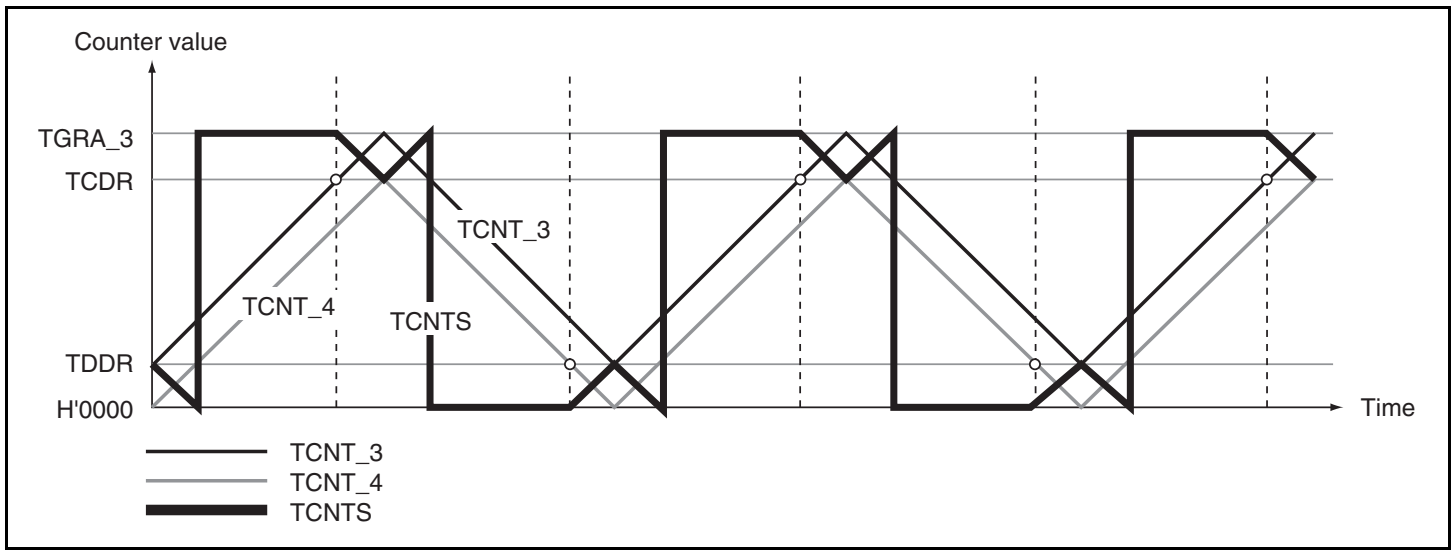


Figure 18.34 Complementary PWM Mode Counter Operation

Register Operation: In complementary PWM mode, nine registers are used, comprising compare registers, buffer registers, and temporary registers. Figure 18.35 shows an example of complementary PWM mode operation.

The registers which are constantly compared with the counters to perform PWM output are TGRB_3, TGRA_4, and TGRB_4. When these registers match the counter, the value set in bits OLSN and OLSP in the timer output control register (TOCR) is output.

The buffer registers for these compare registers are TGRD_3, TGRC_4, and TGRD_4.

Between a buffer register and compare register there is a temporary register. The temporary registers cannot be accessed by the CPU.

Data in a compare register is changed by writing the new data to the corresponding buffer register. The buffer registers can be read or written at any time.

The data written to a buffer register is constantly transferred to the temporary register in the Ta interval. Data is not transferred to the temporary register in the Tb interval. Data written to a buffer register in this interval is transferred to the temporary register at the end of the Tb interval.

The value transferred to a temporary register is transferred to the compare register when TCNTS for which the Tb interval ends matches TGRA_3 when counting up, or H'0000 when counting down. The timing for transfer from the temporary register to the compare register can be selected with bits MD3 to MD0 in the timer mode register (TMDR). Figure 18.35 shows an example in which the mode is selected in which the change is made in the trough.

In the Tb interval (Tb1 in figure 18.35) in which data transfer to the temporary register is not performed, the temporary register has the same function as the compare register, and is compared

with the counter. In this interval, therefore, there are two compare match registers for one-phase output, with the compare register containing the pre-change data, and the temporary register containing the new data. In this interval, the three counters–TCNT_3, TCNT_4, and TCNTS–and two registers–compare register and temporary register–are compared, and PWM output controlled accordingly.

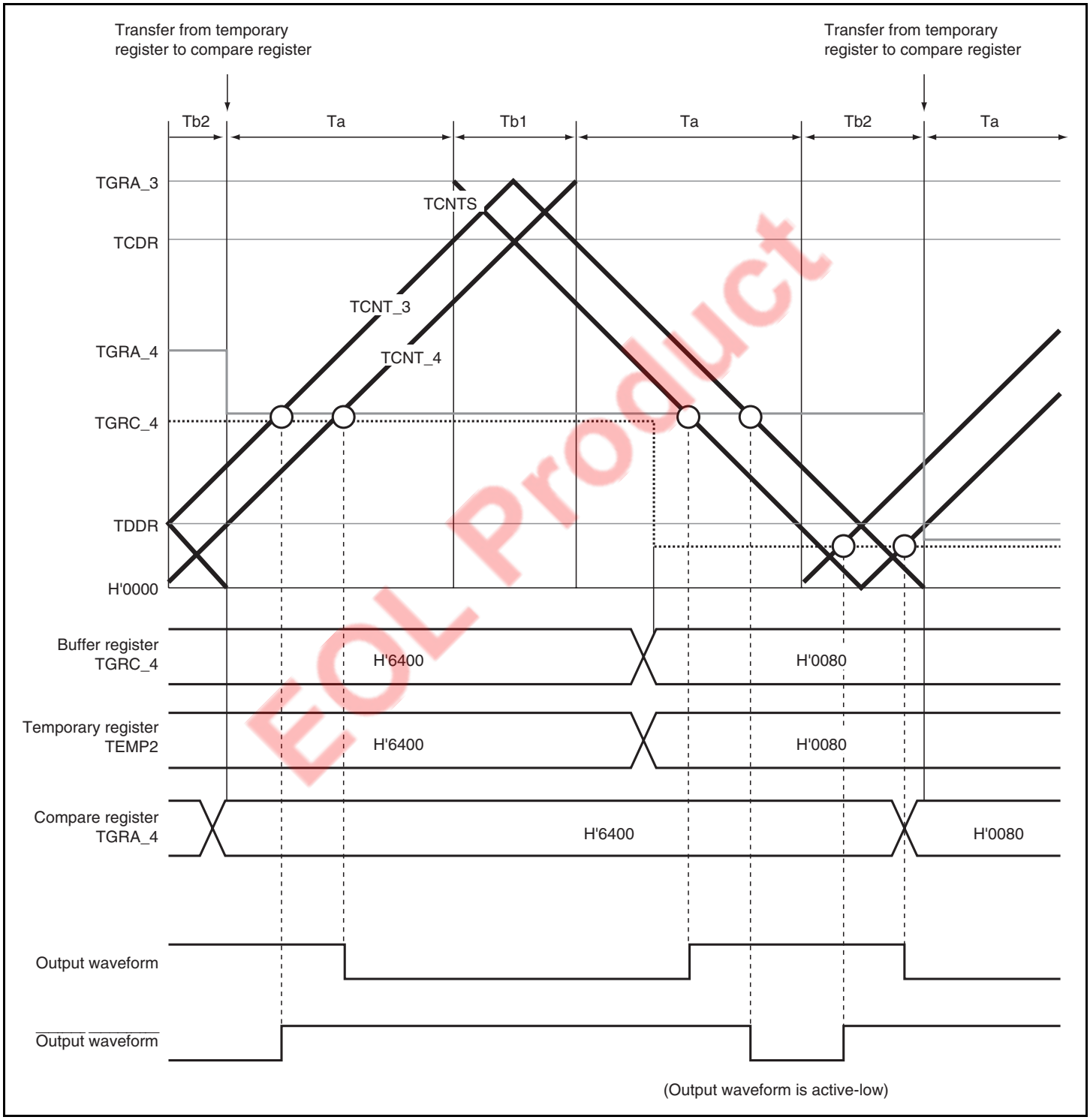


Figure 18.35 Example of Complementary PWM Mode Operation

Initialization: In complementary PWM mode, there are six registers that must be initialized.

Before setting complementary PWM mode with bits MD3 to MD0 in the timer mode register (TMDR), the following initial register values must be set.

TGRC_3 operates as the buffer register for TGRA_3, and should be set with 1/2 the PWM carrier cycle + dead time Td. The timer cycle buffer register (TCBR) operates as the buffer register for the timer cycle data register (TCDR), and should be set with 1/2 the PWM carrier cycle. Set dead time Td in the timer dead time data register (TDDR).

Set the respective initial PWM duty values in buffer registers TGRD_3, TGRC_4, and TGRD_4.

The values set in the five buffer registers excluding TDDR are transferred simultaneously to the corresponding compare registers when complementary PWM mode is set.

Set TCNT_4 to H'0000 before setting complementary PWM mode.

Table 18.41 Registers and Counters Requiring Initialization

Register/Counter	Set Value
TGRC_3	1/2 PWM carrier cycle + dead time Td
TDDR	Dead time Td
TCBR	1/2 PWM carrier cycle
TGRD_3, TGRC_4, TGRD_4	Initial PWM duty value for each phase
TCNT_4	H'0000

Note: The TGRC_3 set value must be the sum of 1/2 the PWM carrier cycle set in TCBR and dead time Td set in TDDR.

PWM Output Level Setting: In complementary PWM mode, the PWM pulse output level is set with bits OLSN and OLSP in the timer output control register (TOCR).

The output level can be set for each of the three positive phases and three negative phases of 6-phase output.

Complementary PWM mode should be cleared before setting or changing output levels.

Dead Time Setting: In complementary PWM mode, PWM pulses are output with a non-overlapping relationship between the positive and negative phases. This non-overlap time is called the dead time.

The non-overlap time is set in the timer dead time data register (TDDR). The value set in TDDR is used as the TCNT_3 counter start value, and creates non-overlap between TCNT_3 and TCNT_4. Complementary PWM mode should be cleared before changing the contents of TDDR.

PWM Cycle Setting: In complementary PWM mode, the PWM pulse cycle is set in two registers—TGRA_3, in which the TCNT_3 upper limit value is set, and TCDR, in which the TCNT_4 upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

TGRA_3 set value = TCDR set value + TDDR set value

The TGRA_3 and TCDR settings are made by setting the values in buffer registers TGRC_3 and TCBR. The values set in TGRC_3 and TCBR are transferred simultaneously to TGRA_3 and TCDR in accordance with the transfer timing selected with bits MD3 to MD0 in the timer mode register (TMDR).

The updated PWM cycle is reflected from the next cycle when the data update is performed at the crest, and from the current cycle when performed in the trough. Figure 18.36 illustrates the operation when the PWM cycle is updated at the crest.

See the following part, Register Data Updating, for the method of updating the data in each buffer register.

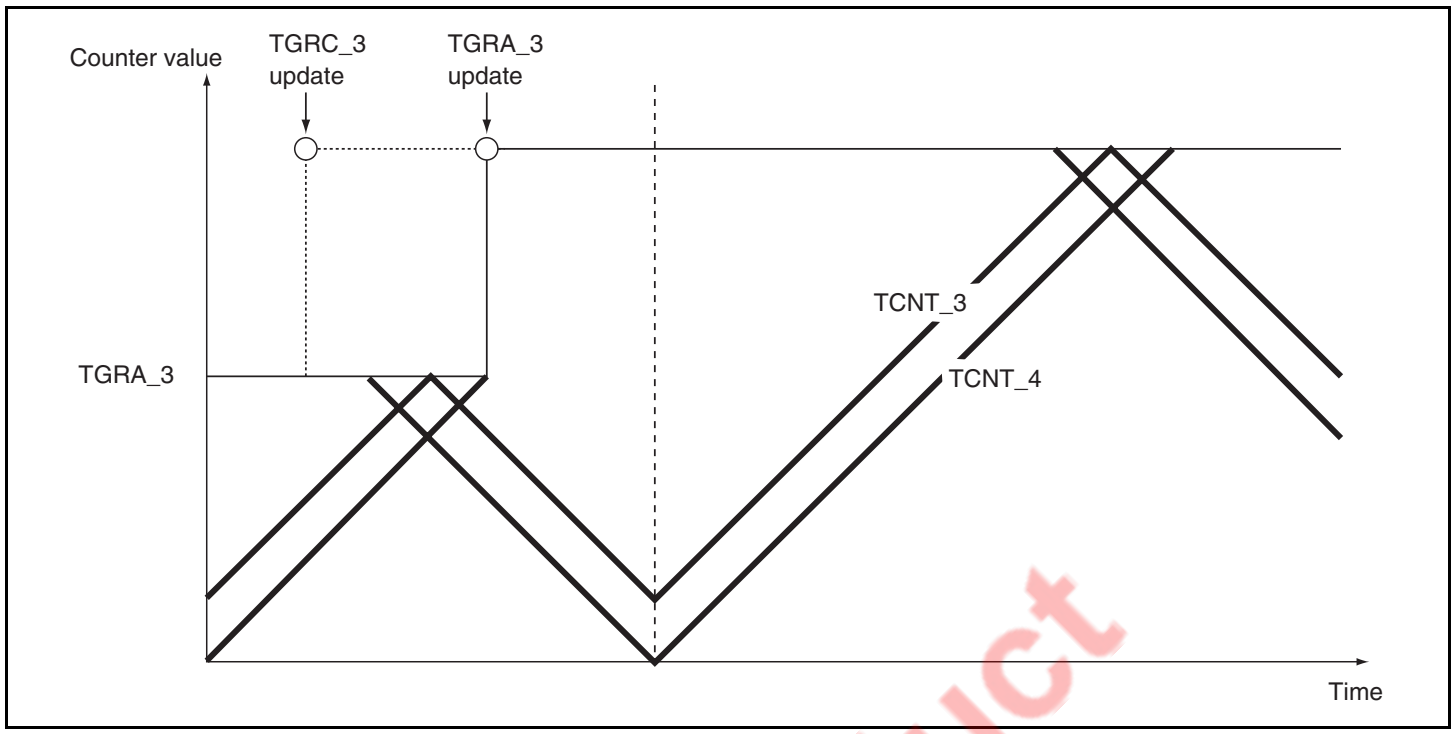


Figure 18.36 Example of PWM Cycle Updating

Register Data Updating: In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five PWM duty and carrier cycle registers that have buffer registers and can be updated during operation.

There is a temporary register between each of these registers and its buffer register. When subcounter TCNTS is not counting, if buffer register data is updated, the temporary register value is also rewritten. Transfer is not performed from buffer registers to temporary registers when TCNTS is counting; in this case, the value written to a buffer register is transferred after TCNTS halts.

The temporary register value is transferred to the compare register at the data update timing set with bits MD3 to MD0 in the timer mode register (TMDR). Figure 18.37 shows an example of data updating in complementary PWM mode. This example shows the mode in which data updating is performed at both the counter crest and trough.

When rewriting buffer register data, a write to TGRD_4 must be performed at the end of the update. Data transfer from the buffer registers to the temporary registers is performed simultaneously for all five registers after the write to TGRD_4.

A write to TGRD_4 must be performed after writing data to the registers to be updated, even when not updating all five registers, or when updating the TGRD_4 data. In this case, the data written to TGRD_4 should be the same as the data prior to the write operation.

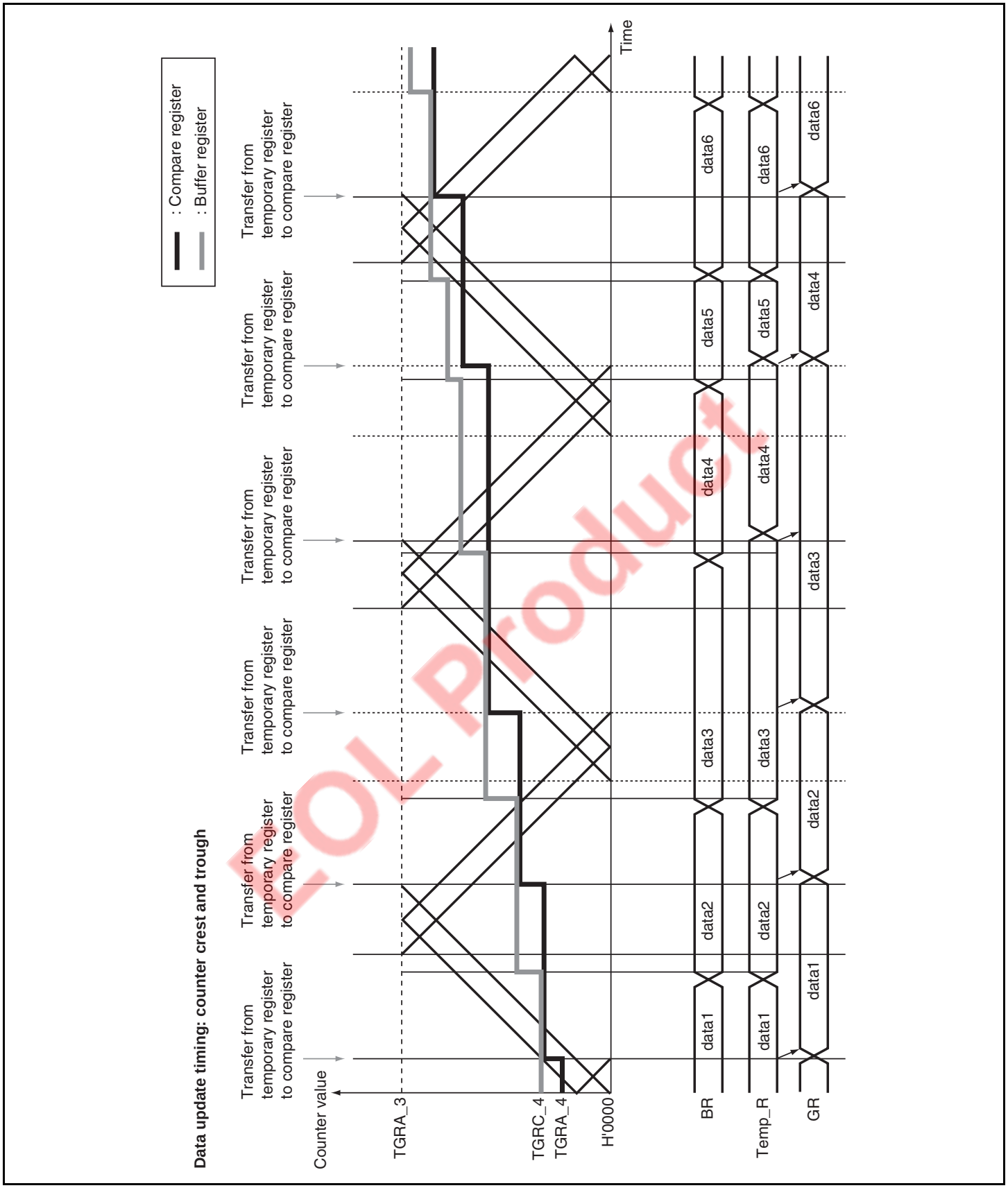


Figure 18.37 Example of Data Update in Complementary PWM Mode

Initial Output in Complementary PWM Mode: In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in the timer output control register (TOCR).

This initial output is the PWM pulse non-active level, and is output from when complementary PWM mode is set with the timer mode register (TMDR) until TCNT_4 exceeds the value set in the dead time register (TDDR). Figure 18.38 shows an example of the initial output in complementary PWM mode.

An example of the waveform when the initial PWM duty value is smaller than the TDDR value is shown in figure 18.39.

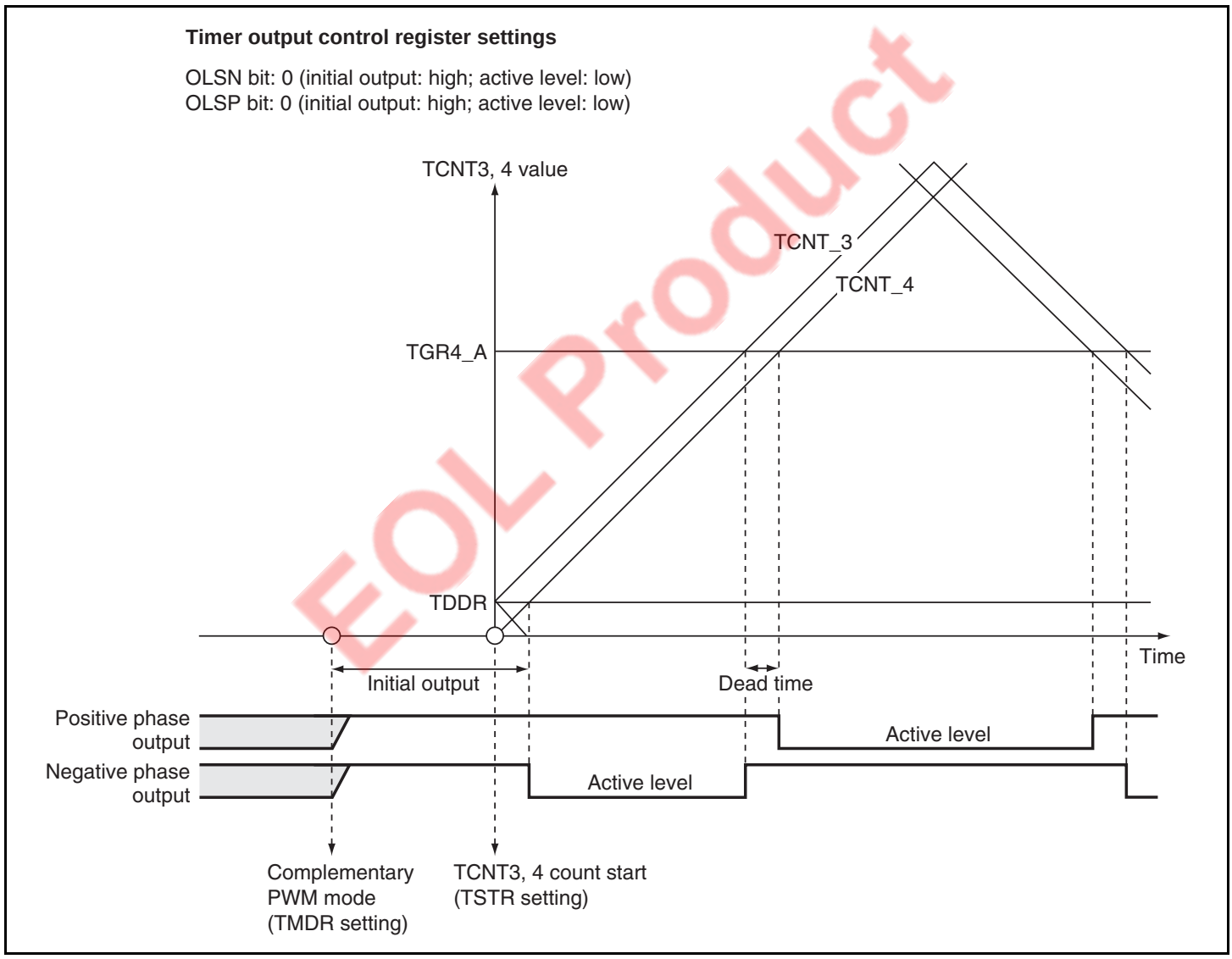


Figure 18.38 Example of Initial Output in Complementary PWM Mode (1)

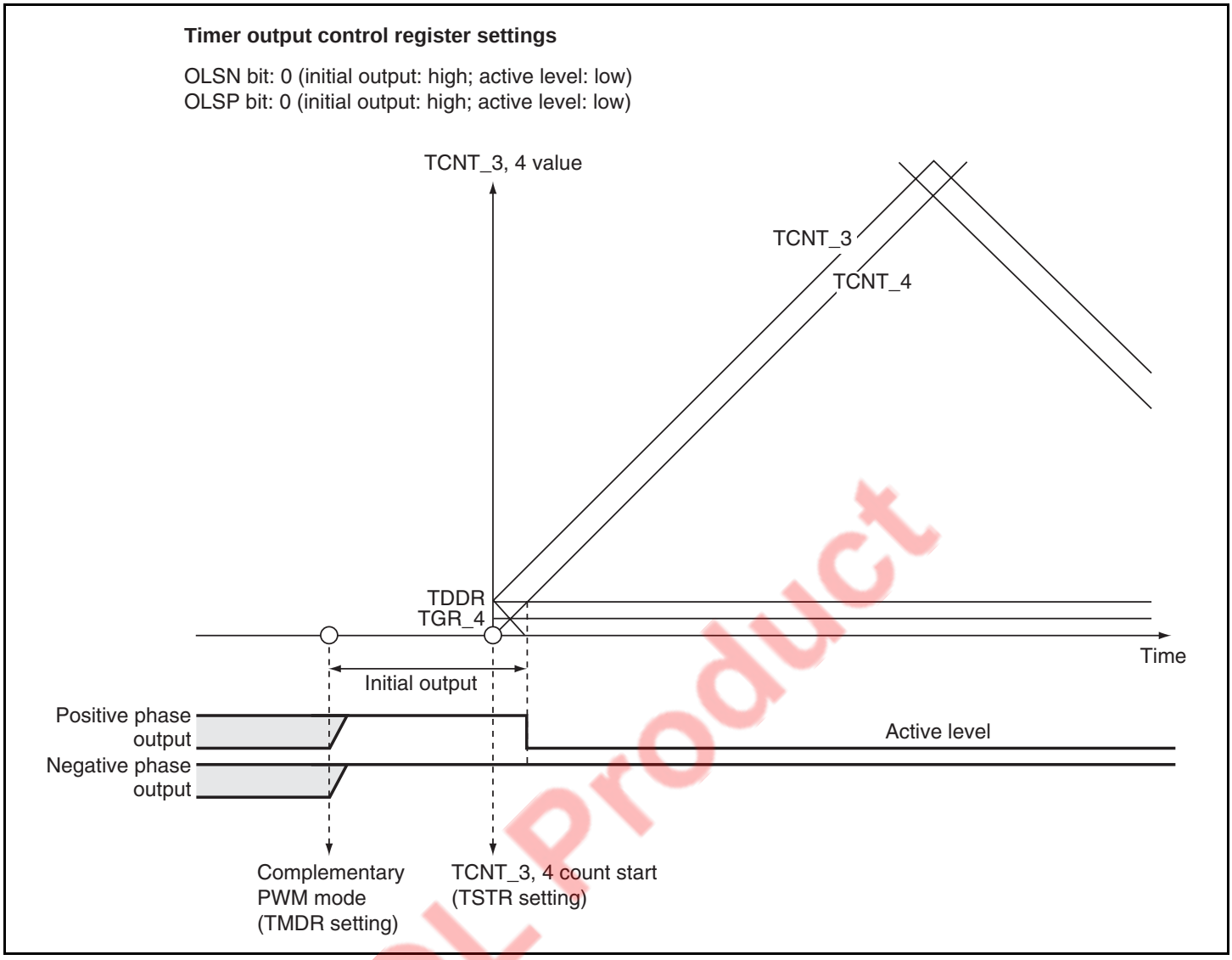


Figure 18.39 Example of Initial Output in Complementary PWM Mode (2)

Complementary PWM Mode PWM Output Generation Method: In complementary PWM mode, 3-phase output is performed of PWM waveforms with a non-overlap time between the positive and negative phases. This non-overlap time is called the dead time.

A PWM waveform is generated by output of the output level selected in the timer output control register in the event of a compare-match between a counter and data register. While TCNTS is counting, data register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0 to 100%. The relative timing of on and off compare-match occurrence may vary, but the compare-match that turns off each phase takes precedence to secure the dead time and ensure that the positive phase and negative phase on times do not overlap. Figures 18.40 to 18.42 show examples of waveform generation in complementary PWM mode.

The positive phase/negative phase off timing is generated by a compare-match with the solid-line counter, and the on timing by a compare-match with the dotted-line counter operating with a delay of the dead time behind the solid-line counter. In the T1 period, compare-match **a** that turns off the negative phase has the highest priority, and compare-matches occurring prior to **a** are ignored. In the T2 period, compare-match **c** that turns off the positive phase has the highest priority, and compare-matches occurring prior to **c** are ignored.

In normal cases, compare-matches occur in the order **a** → **b** → **c** → **d** (or **c** → **d** → **a**' → **b**'), as shown in figure 18.40.

If compare-matches deviate from the **a** → **b** → **c** → **d** order, since the time for which the negative phase is off is less than twice the dead time, the figure shows the positive phase is not being turned on. If compare-matches deviate from the **c** → **d** → **a**' → **b**' order, since the time for which the positive phase is off is less than twice the dead time, the figure shows the negative phase is not being turned on.

If compare-match **c** occurs first following compare-match **a**, as shown in figure 18.41, compare-match **b** is ignored, and the negative phase is turned off by compare-match **d**. This is because turning off of the positive phase has priority due to the occurrence of compare-match **c** (positive phase off timing) before compare-match **b** (positive phase on timing) (consequently, the waveform does not change since the positive phase goes from off to off).

Similarly, in the example in figure 18.42, compare-match **a**' with the new data in the temporary register occurs before compare-match **c**, but other compare-matches occurring up to **c**, which turns off the positive phase, are ignored. As a result, the positive phase is not turned on.

Thus, in complementary PWM mode, compare-matches at turn-off timings take precedence, and turn-on timing compare-matches that occur before a turn-off timing compare-match are ignored.

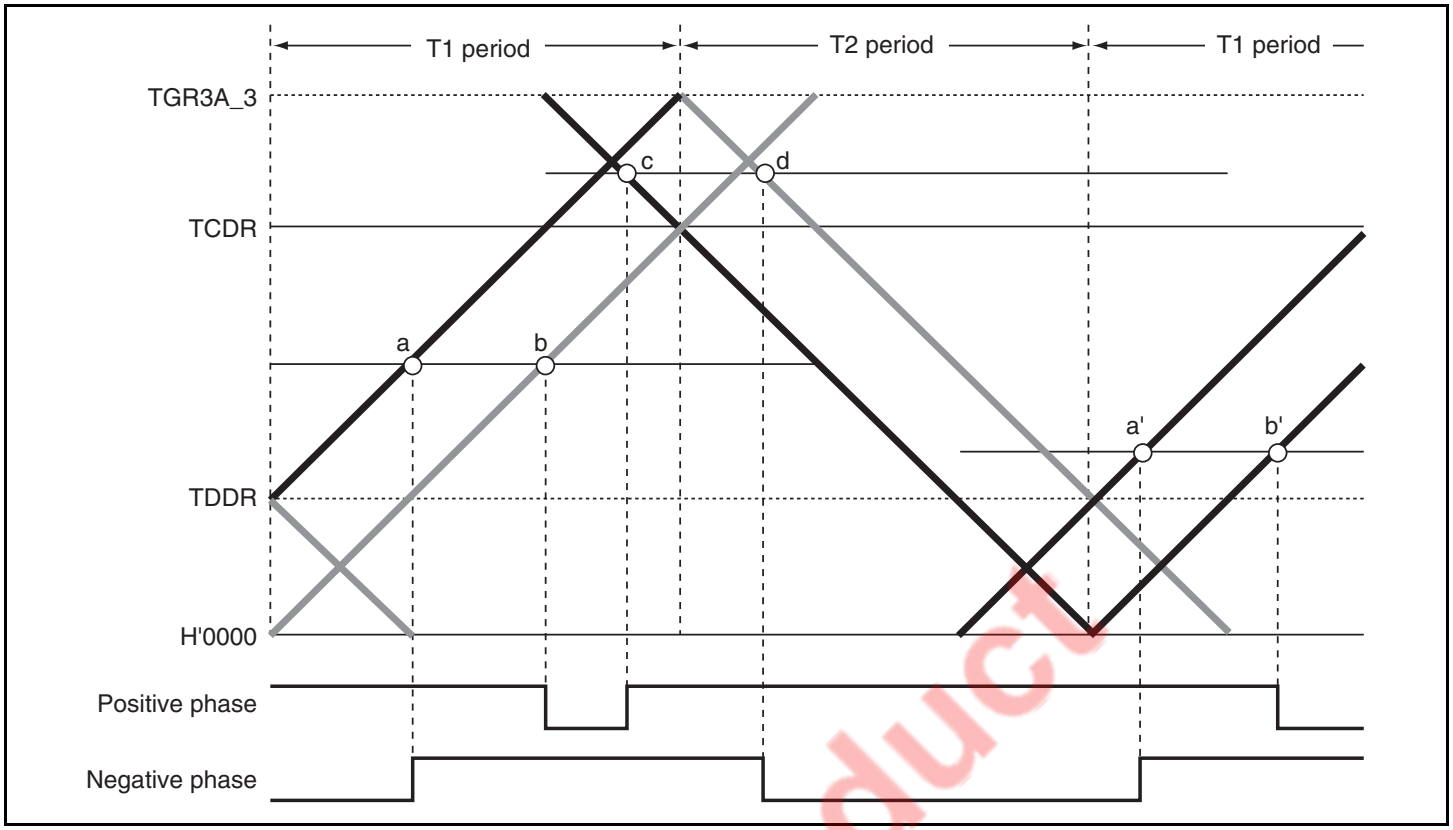


Figure 18.40 Example of Complementary PWM Mode Waveform Output (1)

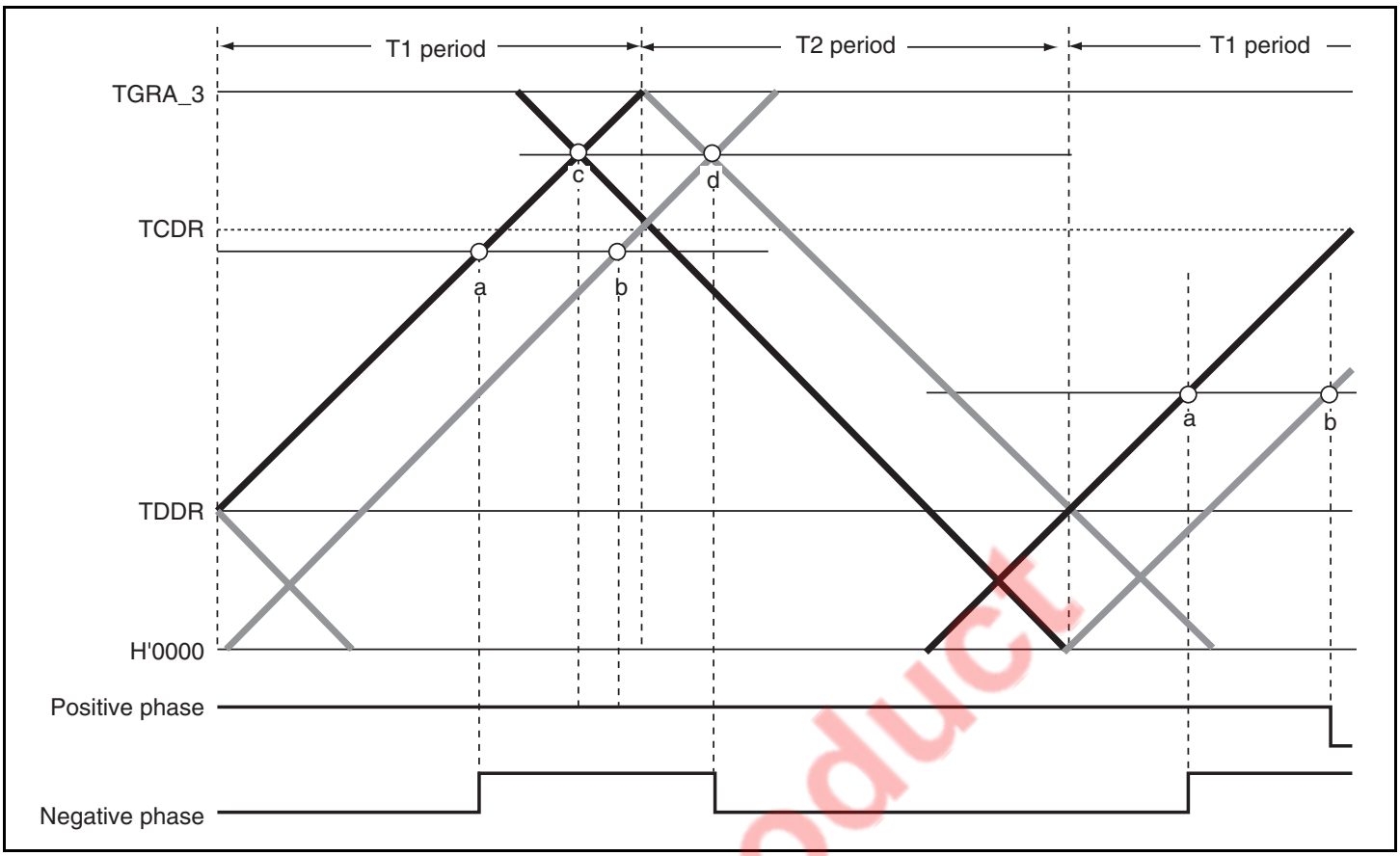


Figure 18.41 Example of Complementary PWM Mode Waveform Output (2)

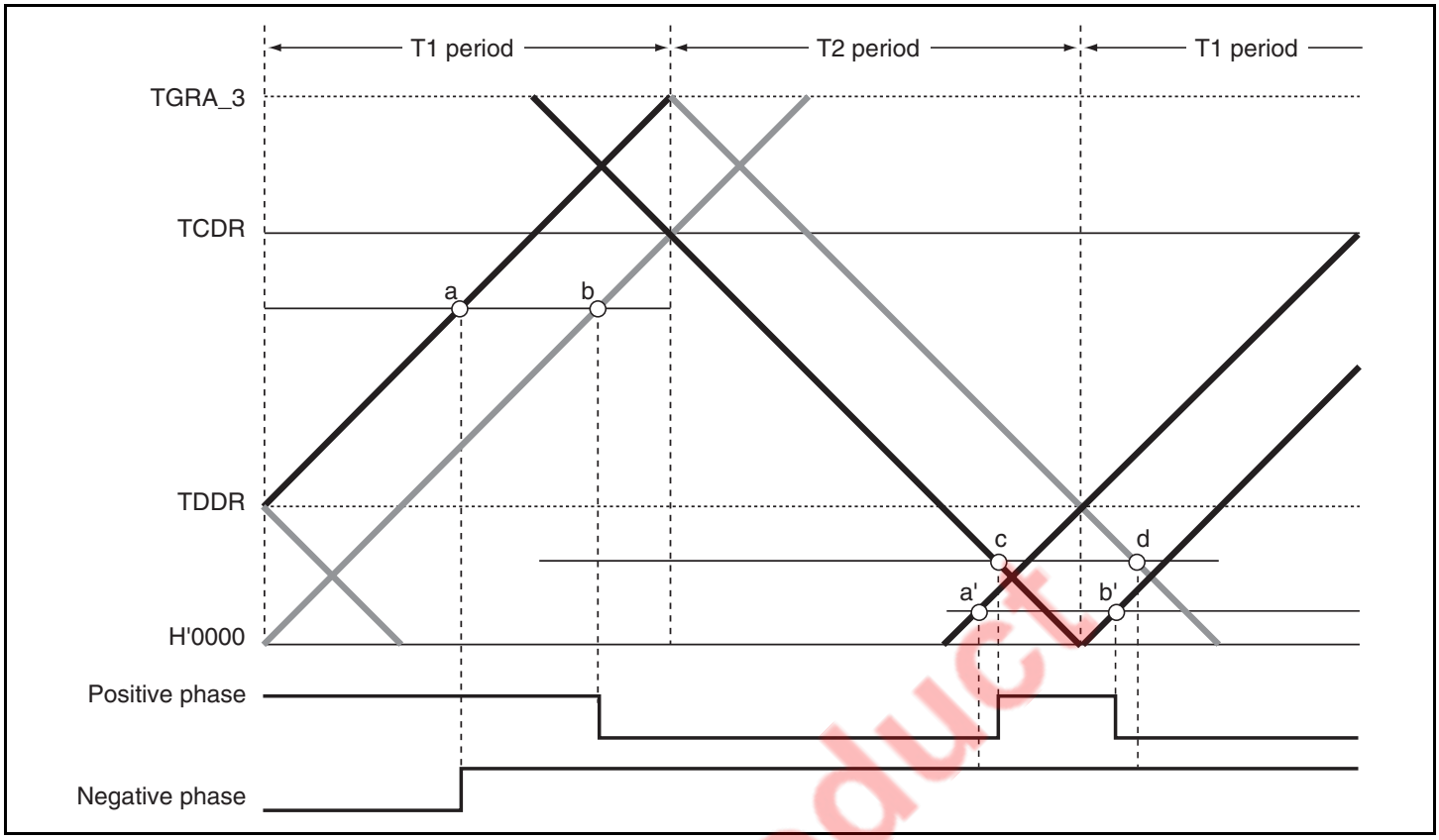


Figure 18.42 Example of Complementary PWM Mode Waveform Output (3)

Complementary PWM Mode 0% and 100% Duty Output: In complementary PWM mode, 0% and 100% duty cycles can be output as required. Figures 18.43 to 18.47 show output examples.

100% duty output is performed when the data register value is set to H'0000. The waveform in this case has a positive phase with a 100% on-state. 0% duty output is performed when the data register value is set to the same value as TGRA_3. The waveform in this case has a positive phase with a 100% off-state.

On and off compare-matches occur simultaneously, but if a turn-on compare-match and turn-off compare-match for the same phase occur simultaneously, both compare-matches are ignored and the waveform does not change.

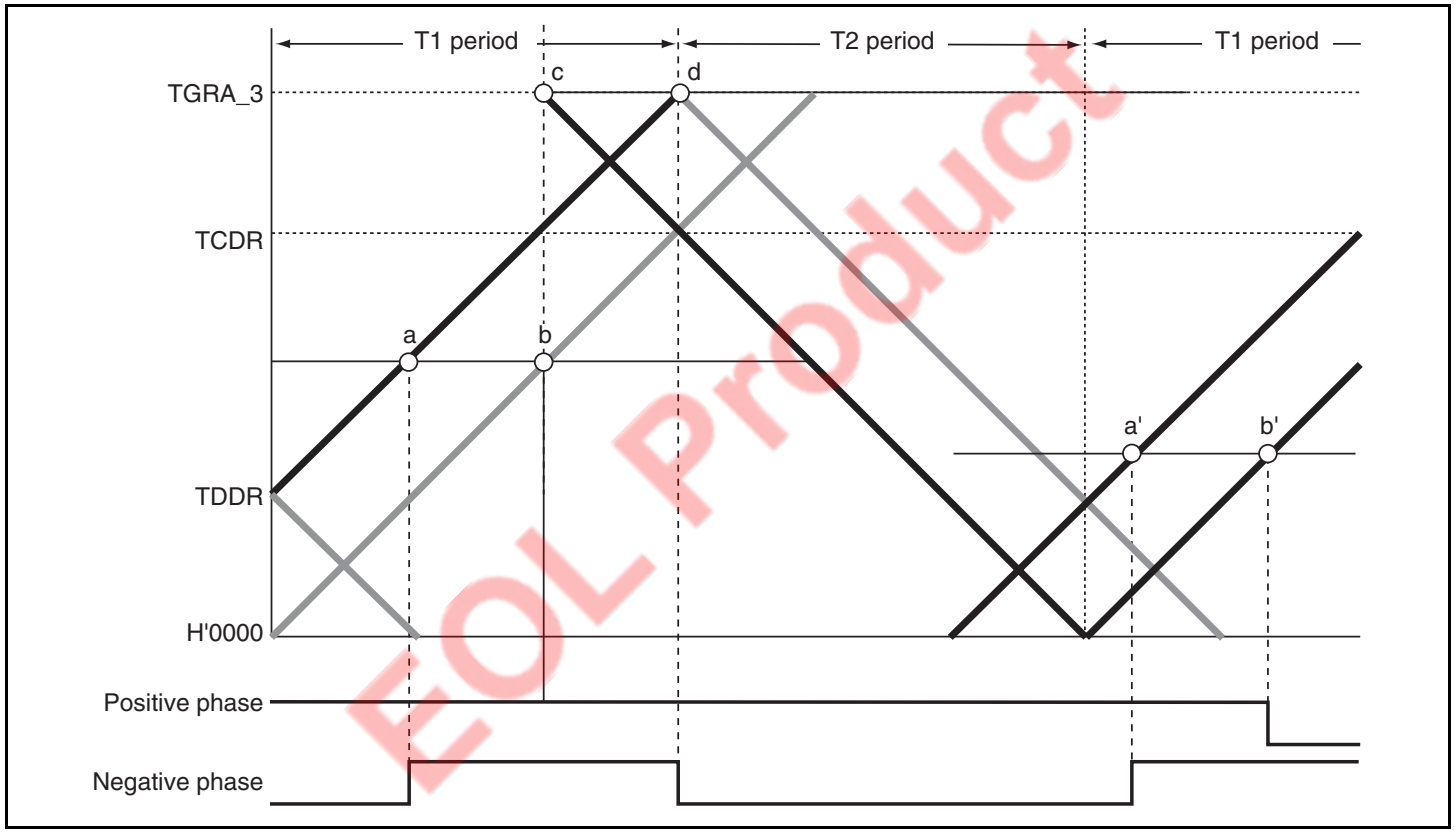


Figure 18.43 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)

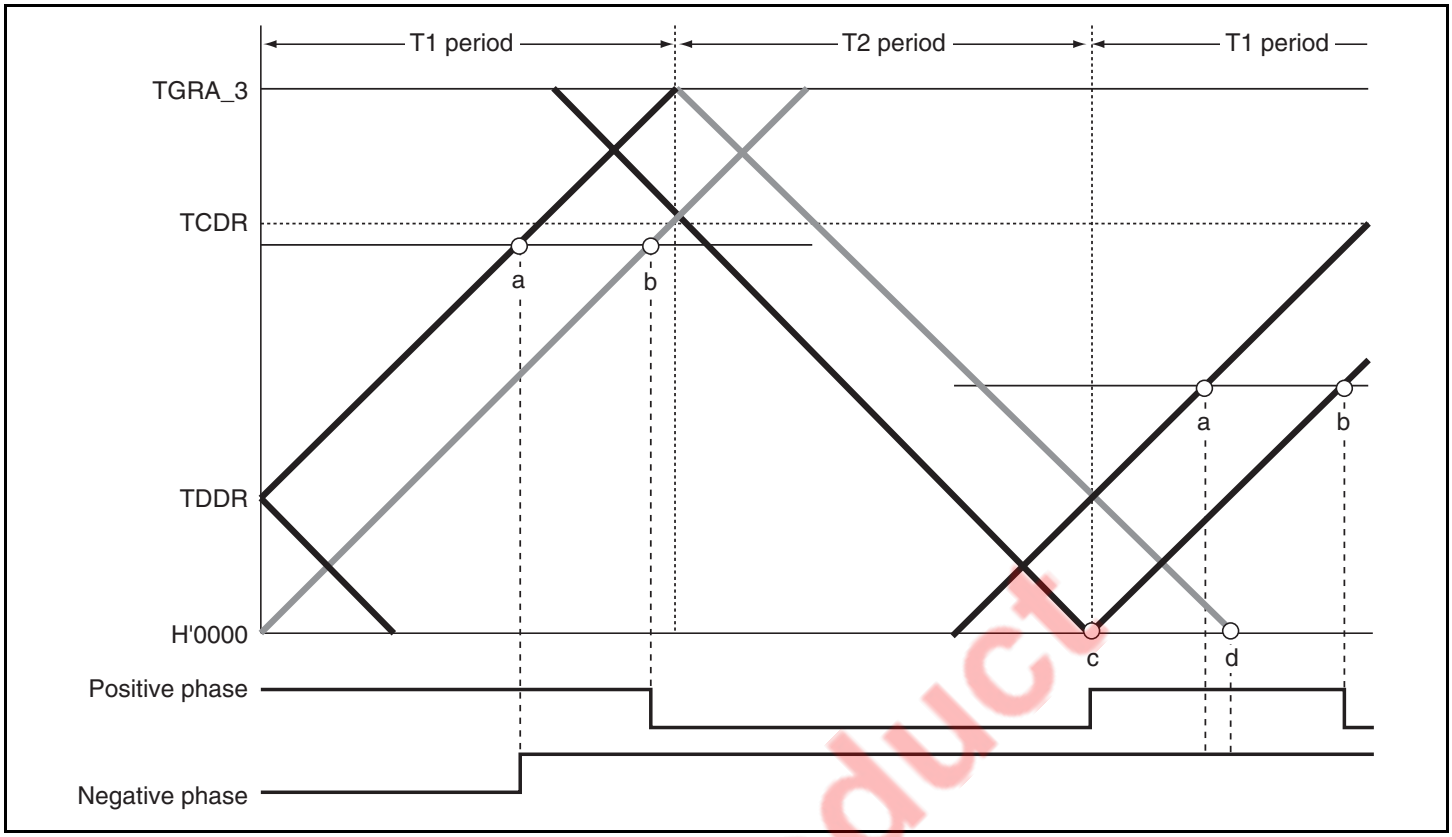


Figure 18.44 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)

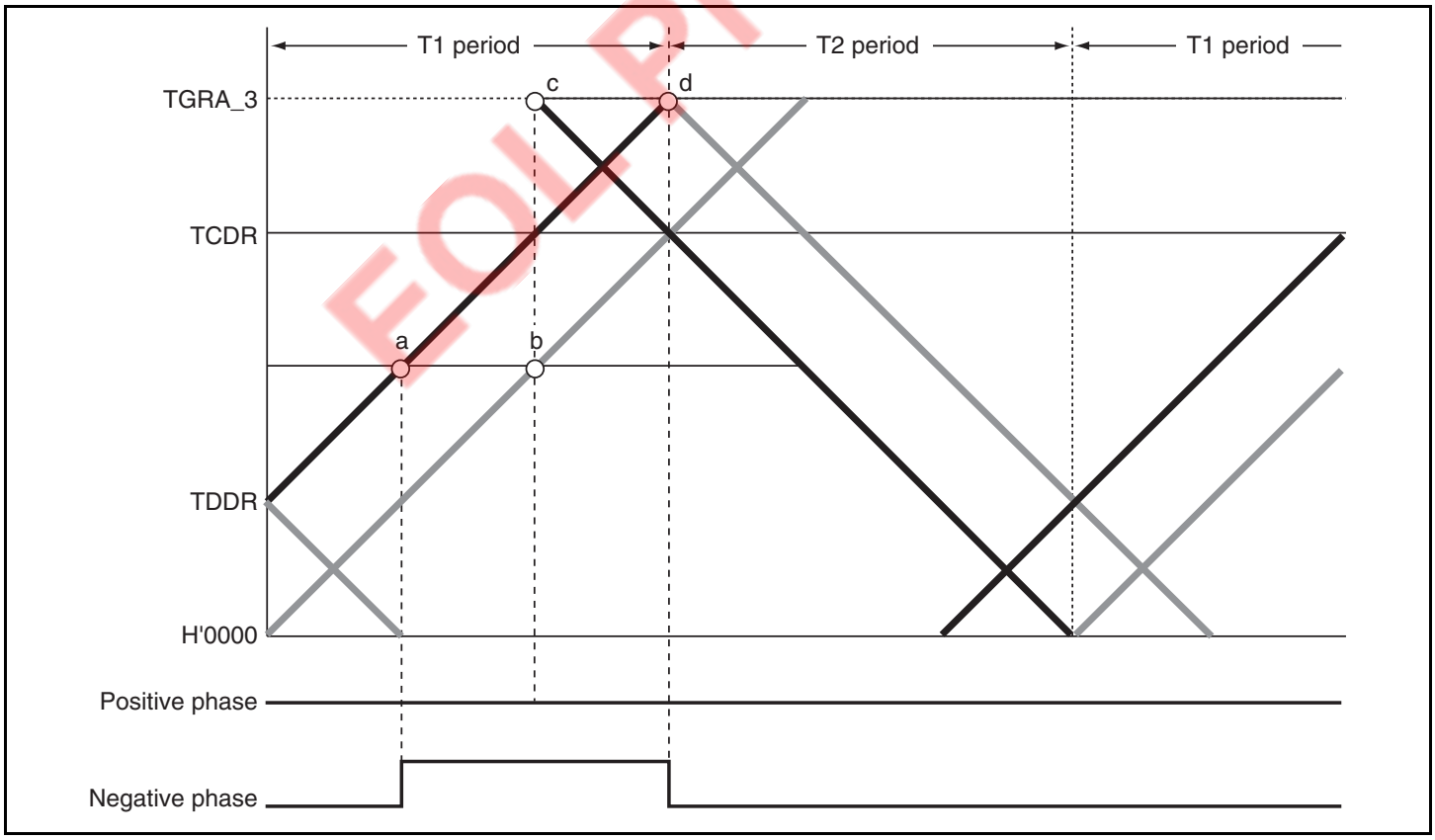


Figure 18.45 Example of Complementary PWM Mode 0% and 100% Waveform Output (3)

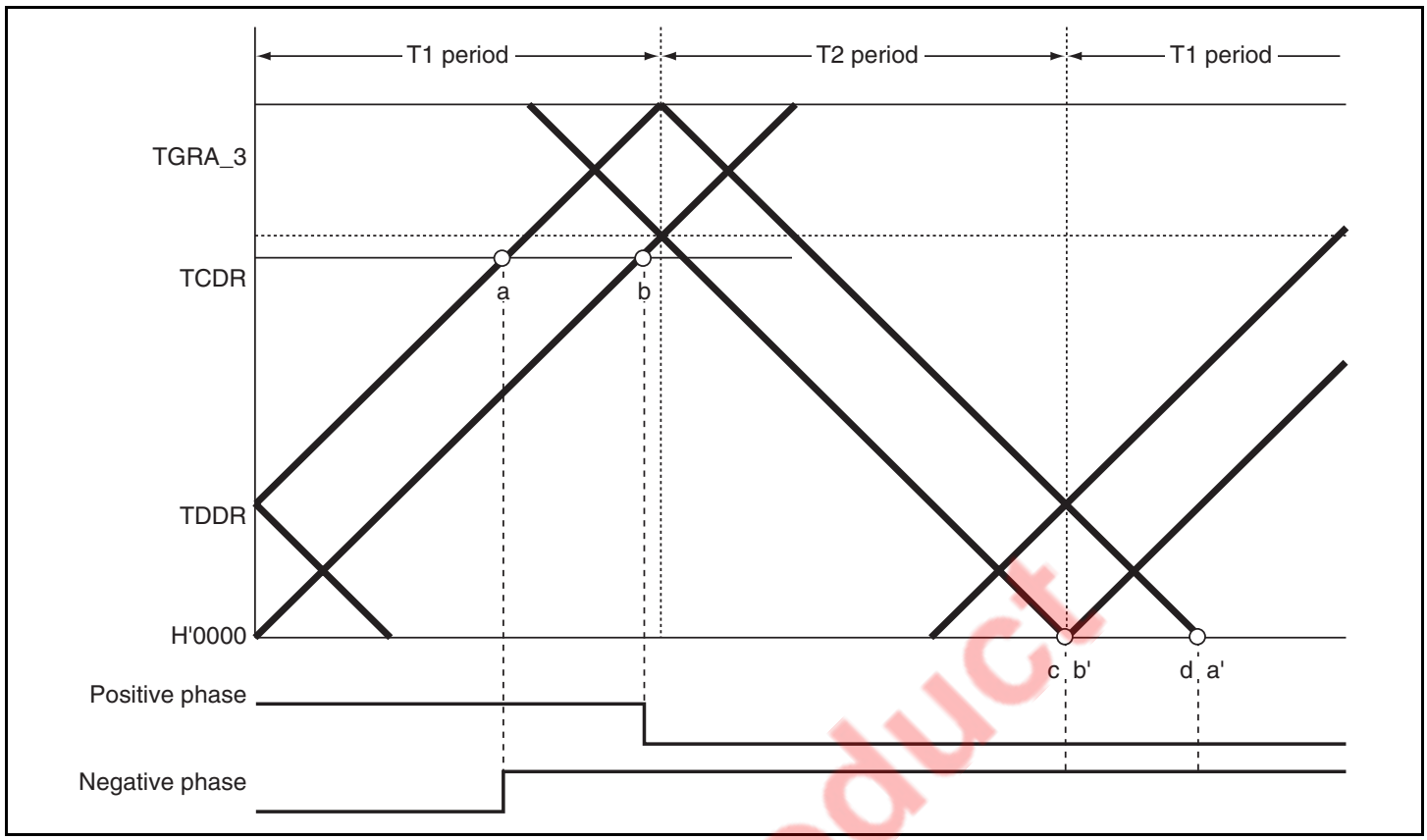


Figure 18.46 Example of Complementary PWM Mode 0% and 100% Waveform Output (4)

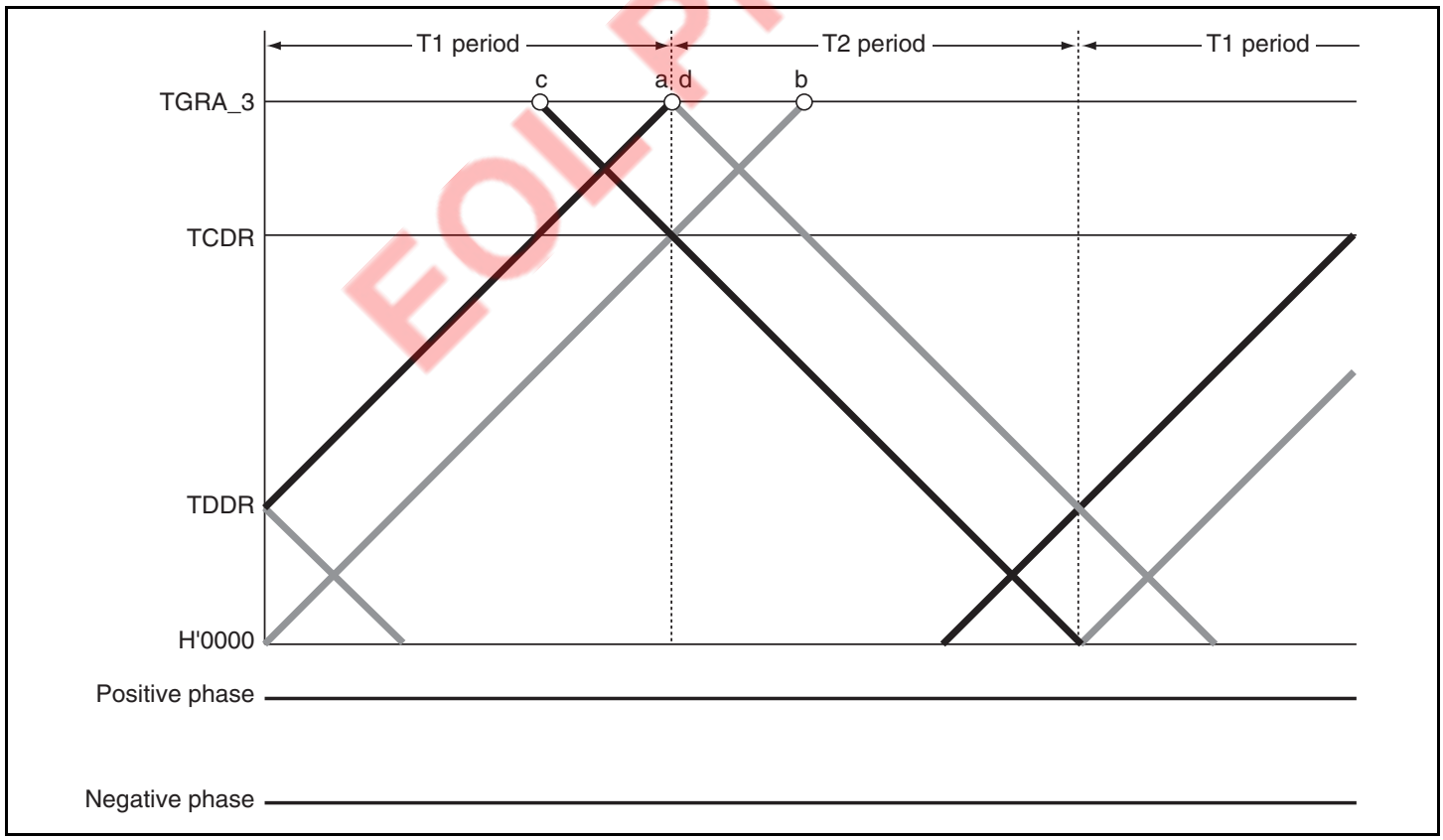


Figure 18.47 Example of Complementary PWM Mode 0% and 100% Waveform Output (5)

Toggle Output Synchronized with PWM Cycle: In complementary PWM mode, toggle output can be performed in synchronization with the PWM carrier cycle by setting the PSYE bit to 1 in the timer output control register (TOCR). An example of a toggle output waveform is shown in figure 18.48.

This output is toggled by a compare-match between TCNT_3 and TGRA_3 and a compare-match between TCNT4 and H'0000.

The output pin for this toggle output is the TIOC3A pin. The initial output is 1.

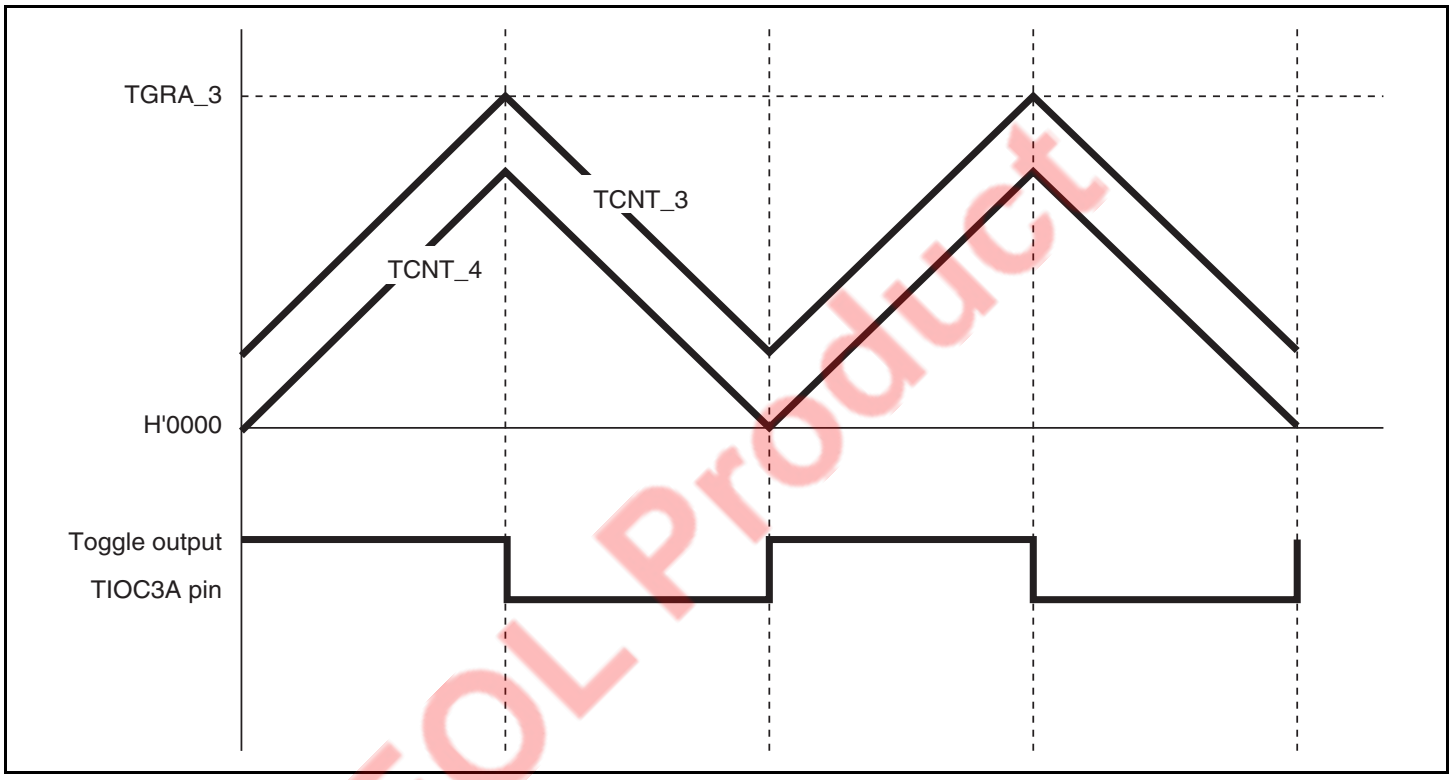


Figure 18.48 Example of Toggle Output Waveform Synchronized with PWM Output

Counter Clearing by another Channel: In complementary PWM mode, by setting a mode for synchronization with another channel by means of the timer synchro register (TSYR), and selecting synchronous clearing with bits CCLR2 to CCLR0 in the timer control register (TCR), it is possible to have TCNT_3, TCNT_4, and TCNTS cleared by another channel.

Figure 18.49 illustrates the operation.

Use of this function enables counter clearing and restarting to be performed by means of an external signal.

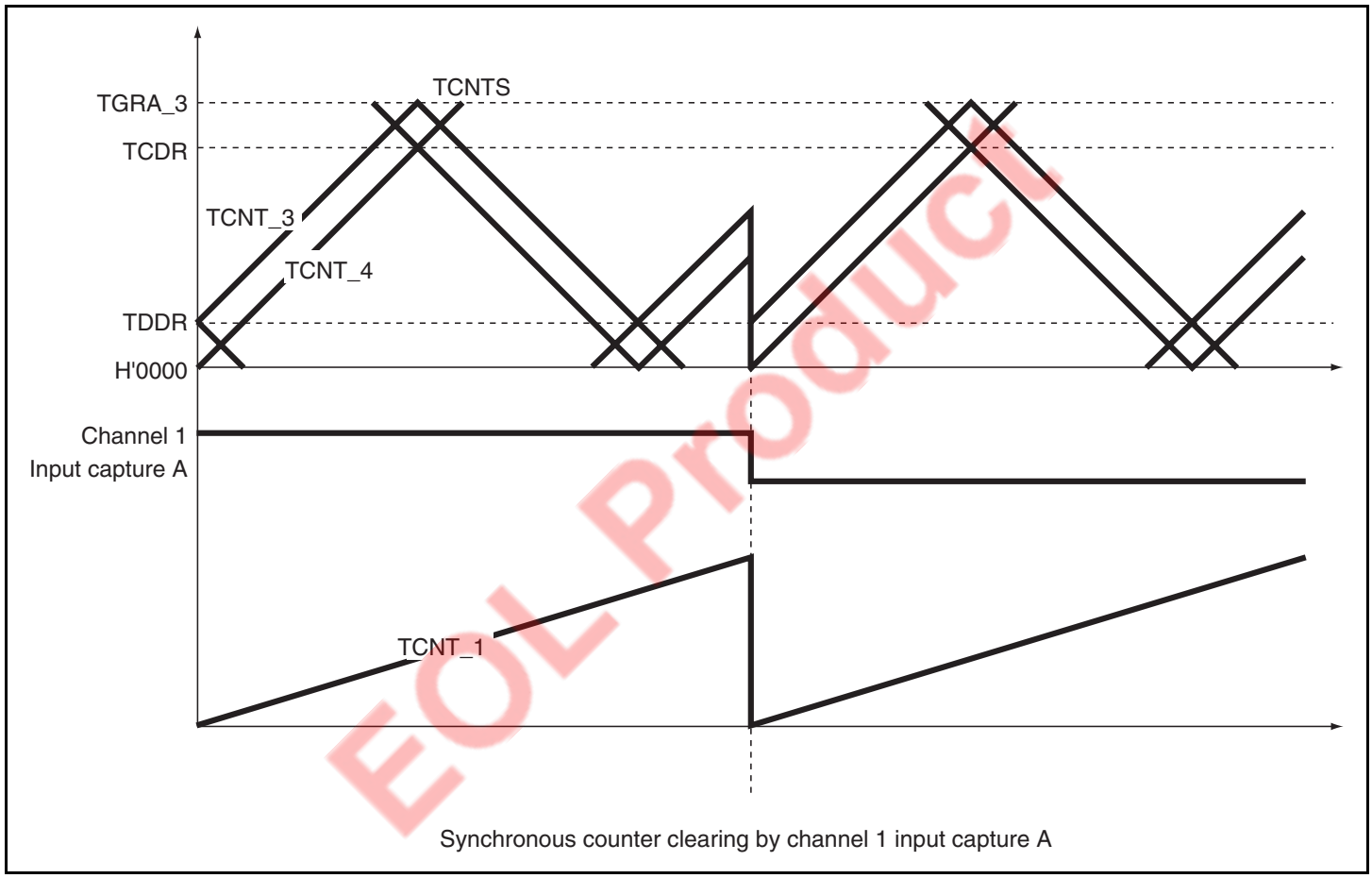


Figure 18.49 Counter Clearing Synchronized with Another Channel

Example of AC Synchronous Motor (Brushless DC Motor) Drive Waveform Output: In complementary PWM mode, a brushless DC motor can easily be controlled using the timer gate control register (TGCR). Figures 18.50 to 18.53 show examples of brushless DC motor drive waveforms created using TGCR.

When output phase switching for a 3-phase brushless DC motor is performed by means of external signals detected with a Hall element, etc., clear the FB bit in TGCR to 0. In this case, the external signals indicating the polarity position are input to channel 0 timer input pins TIOC0A, TIOC0B, and TIOC0C (set with PFC). When an edge is detected at pin TIOC0A, TIOC0B, or TIOC0C, the output on/off state is switched automatically.

When the FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCR is cleared to 0 or set to 1.

The drive waveforms are output from the complementary PWM mode 6-phase output pins. With this 6-phase output, in the case of on output, it is possible to use complementary PWM mode output and perform chopping output by setting the N bit or P bit to 1. When the N bit or P bit is 0, level output is selected.

The 6-phase output active level (on output level) can be set with the OLSN and OLSP bits in the timer output control register (TOCR) regardless of the setting of the N and P bits.

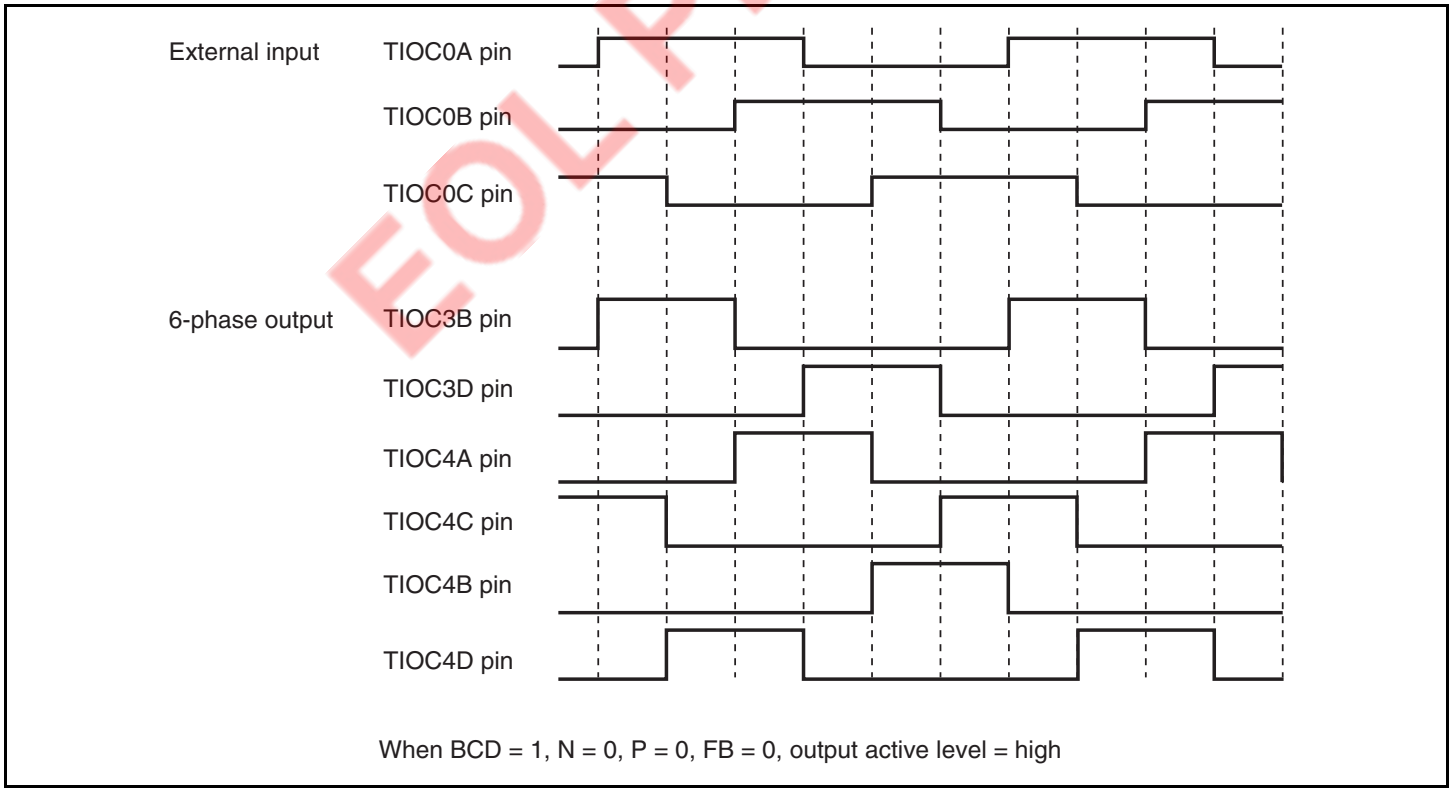


Figure 18.50 Example of Output Phase Switching by External Input (1)

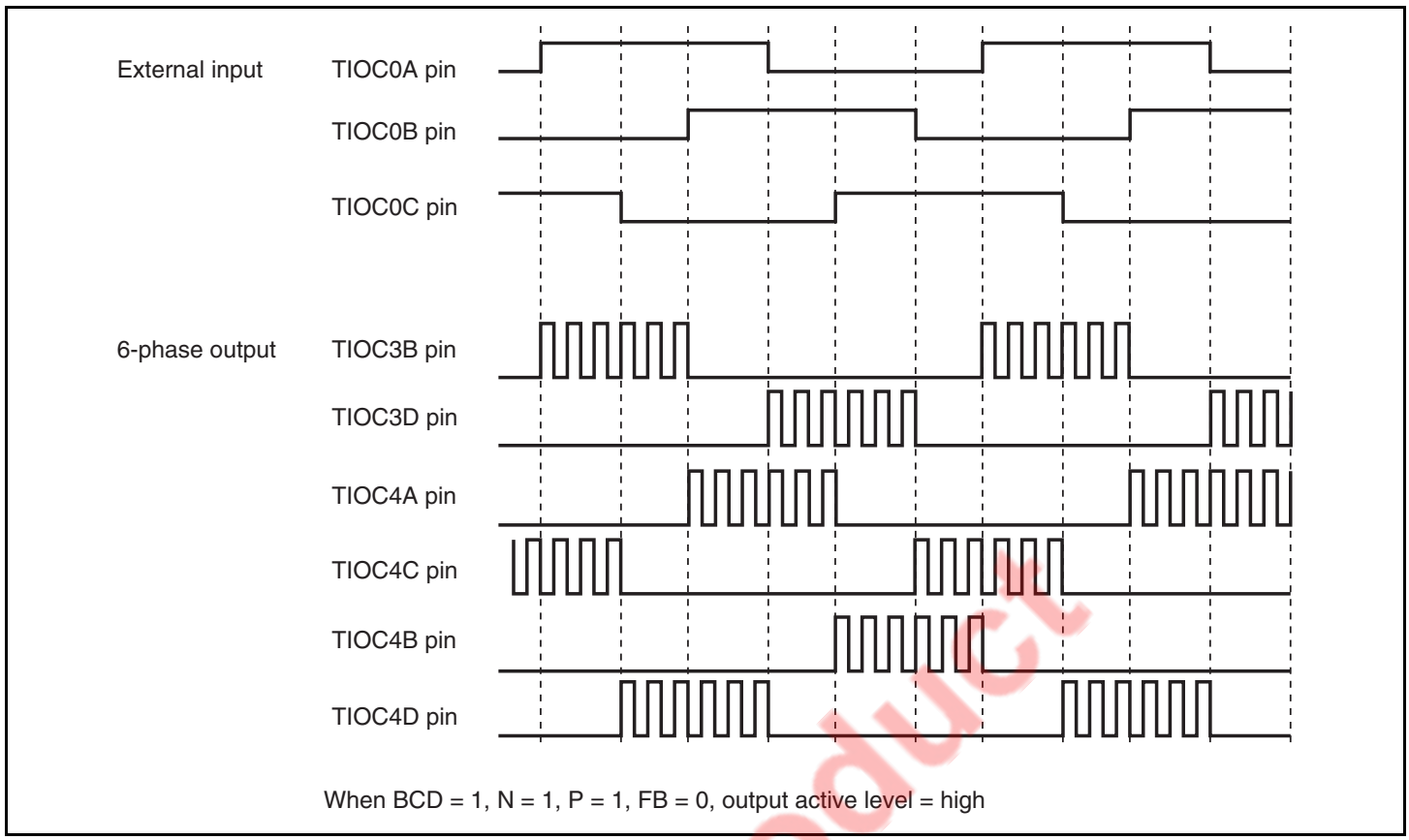


Figure 18.51 Example of Output Phase Switching by External Input (2)

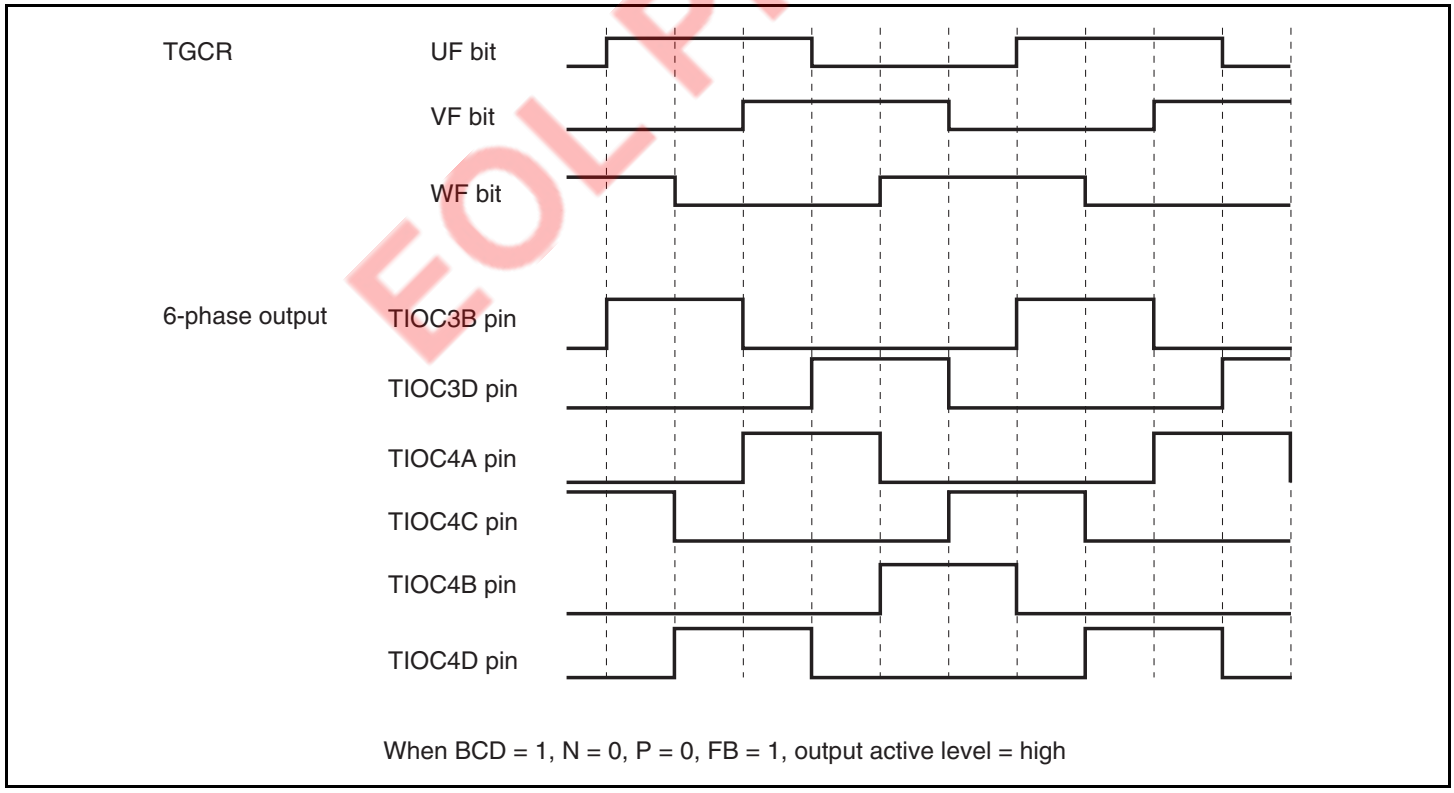


Figure 18.52 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1)

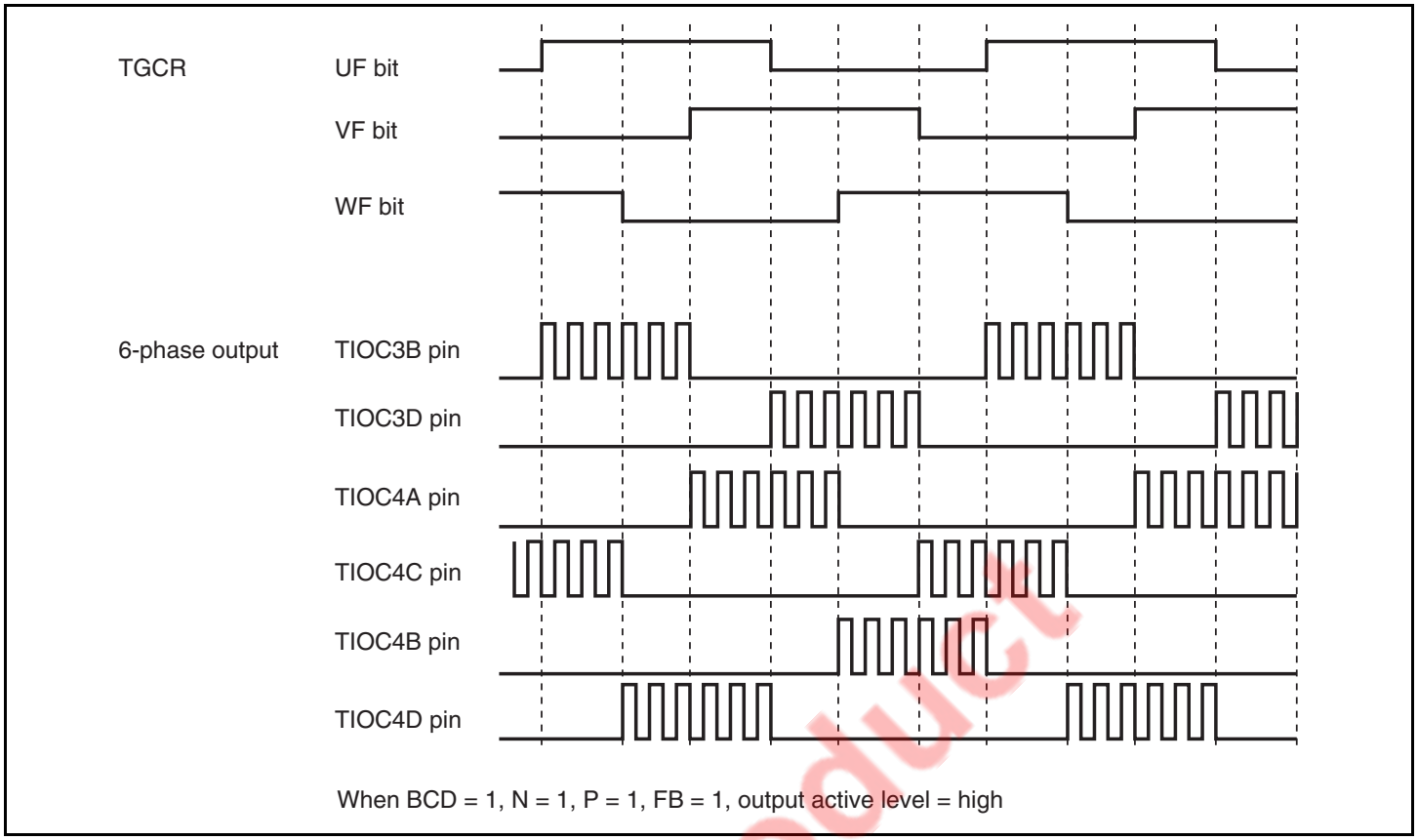


Figure 18.53 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2)

A/D Conversion Start Request Setting: In complementary PWM mode, an A/D conversion start request can be set using a TGRA_3 compare-match or a compare-match on a channel other than channels 3 and 4.

When start requests using a TGRA_3 compare-match are set, A/D conversion can be started at the center of the PWM pulse.

A/D conversion start requests can be set by setting the TTGE bit to 1 in the timer interrupt enable register (TIER).

Complementary PWM Mode Output Protection Function

Complementary PWM mode output has the following protection functions.

Register and Counter Miswrite Prevention Function: With the exception of the buffer registers, which can be rewritten at any time, access by the CPU can be enabled or disabled for the mode registers, control registers, compare registers, and counters used in complementary PWM mode by means of bit 0 (MTURWE) in PEMTURWER of the port E (port E MTU R/W enable register).

Some registers in channels 3 and 4 concerned are listed below: total 21 registers of TCR_3 and TCR_4; TMDR_3 and TMDR_4; TIORH_3 and TIORH_4; TIORL_3 and TIORL_4; TIER_3 and TIER_4; TCNT_3 and TCNT_4; TGRA_3 and TGRA_4; TGRB_3 and TGRB_4; TOER; TOCR; TGCR; TCDR; and TDDR.

This function enables the CPU to prevent miswriting due to the CPU runaway by disabling CPU access to the mode registers, control register, and counters. In access disabled state, an undefined value is read from the registers concerned, and cannot be modified.

Halting of PWM Output by External Signal: The 6-phase PWM output pins can be set automatically to the high-impedance state by inputting specified external signals. There are four external signal input pins.

See section 18.9, Port Output Enable (POE), for details.

18.5 Interrupts

18.5.1 Interrupts and Priority

There are three kinds of MTU interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disabled bit, allowing the 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 priority can be changed by the interrupt controller, however the priority within a channel is fixed.

Table 18.42 lists the MTU interrupt sources.

Table 18.42 MTU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DMA Activation	Priority
0	TGI0A	TGRA_0 input capture/compare match	TGFA_0	Possible	High
	TGI0B	TGRB_0 input capture/compare match	TGFB_0	Not possible	
	TGI0C	TGRC_0 input capture/compare match	TGFC_0	Not possible	
	TGI0D	TGRD_0 input capture/compare match	TGFD_0	Not possible	
	TCI0V	TCNT_0 overflow	TCFV_0	Not possible	
1	TGI1A	TGRA_1 input capture/compare match	TGFA_1	Possible	
	TGI1B	TGRB_1 input capture/compare match	TGFB_1	Not possible	
	TCI1V	TCNT_1 overflow	TCFV_1	Not possible	
	TCI1U	TCNT_1 underflow	TCFU_1	Not possible	
2	TGI2A	TGRA_2 input capture/compare match	TGFA_2	Possible	
	TGI2B	TGRB_2 input capture/compare match	TGFB_2	Not possible	
	TCI2V	TCNT_2 overflow	TCFV_2	Not possible	
	TCI2U	TCNT_2 underflow	TCFU_2	Not possible	
3	TGI3A	TGRA_3 input capture/compare match	TGFA_3	Possible	
	TGI3B	TGRB_3 input capture/compare match	TGFB_3	Not possible	
	TGI3C	TGRC_3 input capture/compare match	TGFC_3	Not possible	
	TGI3D	TGRD_3 input capture/compare match	TGFD_3	Not possible	
	TCI3V	TCNT_3 overflow	TCFV_3	Not possible	
4	TGI4A	TGRA_4 input capture/compare match	TGFA_4	Possible	
	TGI4B	TGRB_4 input capture/compare match	TGFB_4	Not possible	
	TGI4C	TGRC_4 input capture/compare match	TGFC_4	Not possible	
	TGI4D	TGRD_4 input capture/compare match	TGFD_4	Not possible	
	TCI4V	TCNT_4 overflow/underflow	TCFV_4	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priority can be changed by the interrupt controller.

Input Capture/Compare Match Interrupt: An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The MTU has 16 input capture/compare match interrupts, four each for channels 0, 3, and 4, and two each for channels 1 and 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 MTU has five overflow interrupts, one for each channel.

Underflow Interrupt: An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The MTU has four underflow interrupts, one each for channels 1 and 2.

18.5.2 DMA Activation

The DMA can be activated by a TGRA input capture/compare match interrupt in each channel.

If the TGFASEL bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 due to the TGRA input capture/compare match in each channel, the DMA transfer is requested to DMA. For details, see section 13, Direct Memory Access Controller (DMAC).

A total of five MTU input capture/compare match interrupts can be used as DMA activation sources for each channel

When using DMA, do not clear the flag by writing 0 after reading TGFA = 1. The flag can be automatically cleared by DMA hardware. However, if a DMA address error occurs during a DMA read cycle, the flag must be cleared using software by writing 0 after reading TGFA = 1.

18.5.3 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match in each 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 MTU conversion start trigger has been selected on the A/D converter at this time, A/D conversion starts.

In the MTU, a total of five TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

18.6 Operation Timing

18.6.1 Input/Output Timing

TCNT Count Timing: Figure 18.54 shows TCNT count timing in internal clock operation, and figure 18.55 shows TCNT count timing in external clock operation (normal mode), and figure 18.56 shows TCNT count timing in external clock operation (phase counting mode).

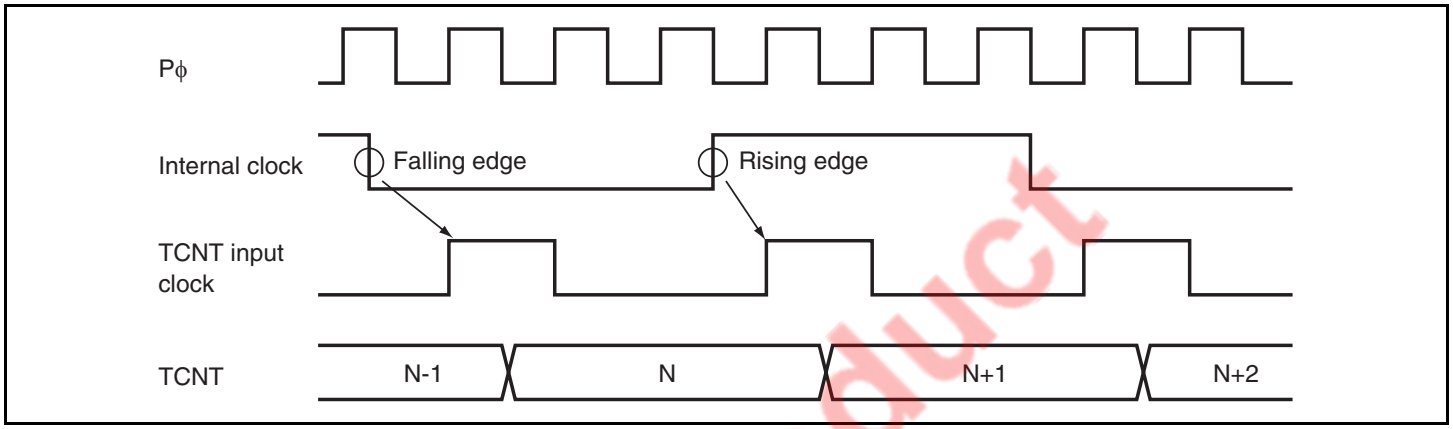


Figure 18.54 Count Timing in Internal Clock Operation

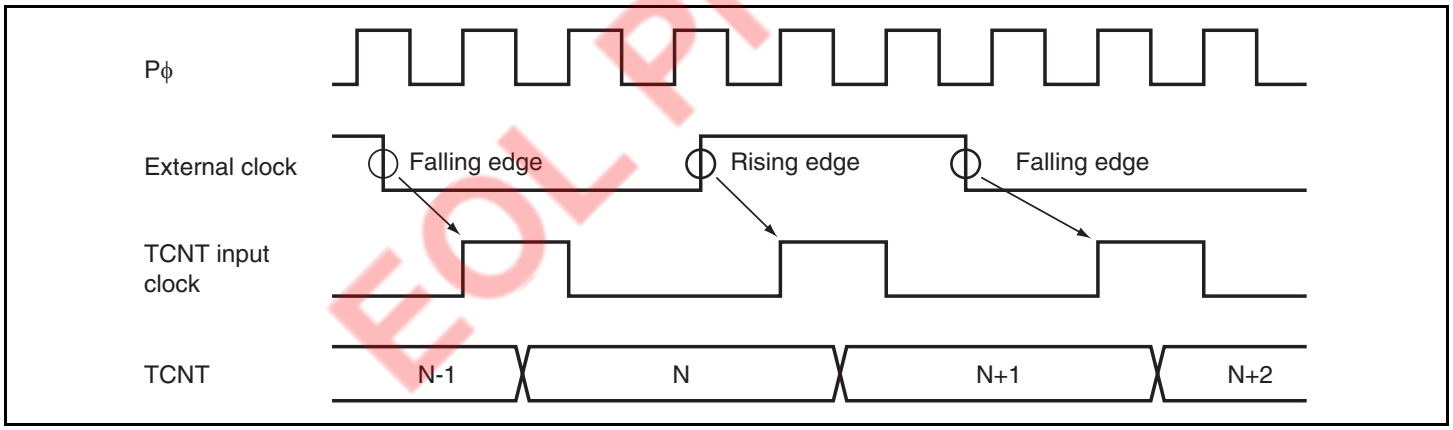


Figure 18.55 Count Timing in External Clock Operation

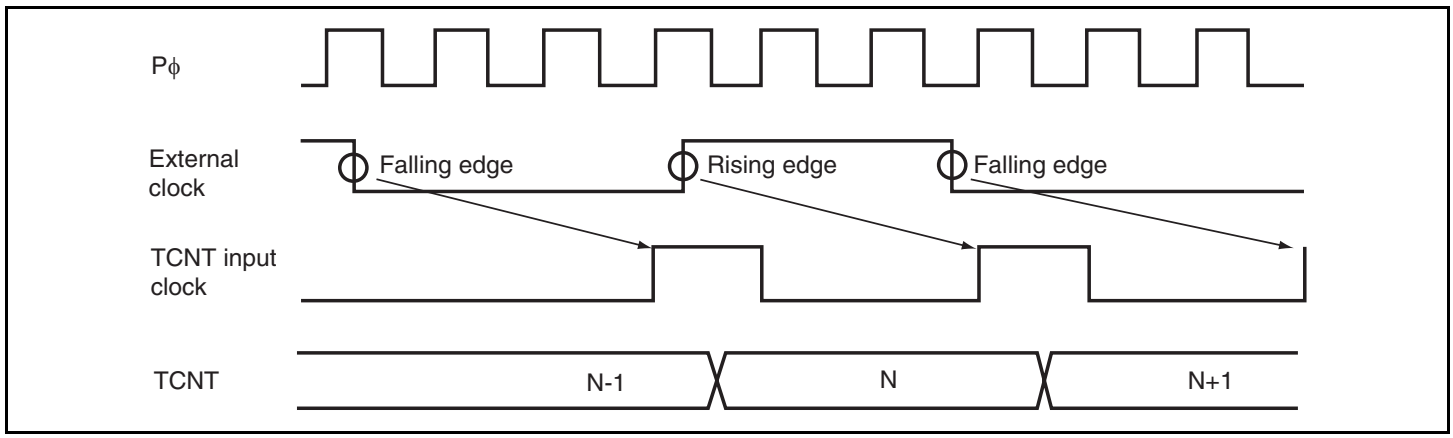


Figure 18.56 Count Timing in External Clock Operation (Phase Counting Mode)

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 (TIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 18.57 shows output compare output timing (normal mode and PWM mode) and figure 18.58 shows output compare output timing (complementary PWM mode and reset synchronous PWM mode).

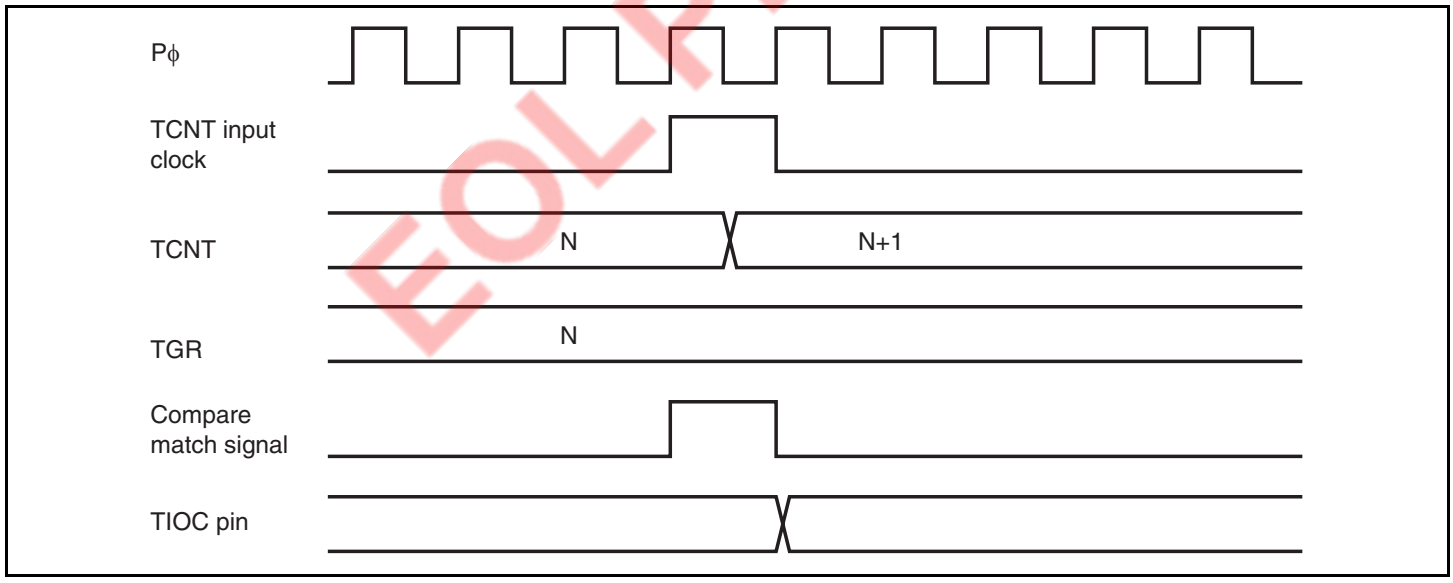


Figure 18.57 Output Compare Output Timing (Normal Mode/PWM Mode)

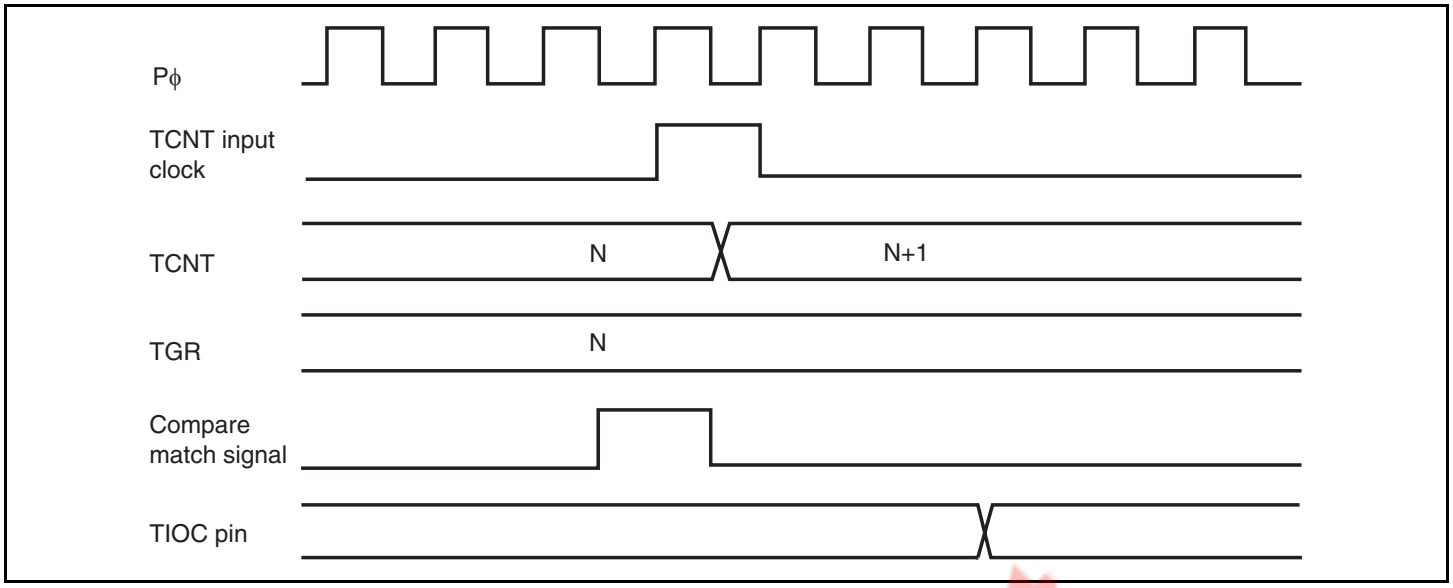


Figure 18.58 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)

Input Capture Signal Timing: Figure 18.59 shows input capture signal timing.

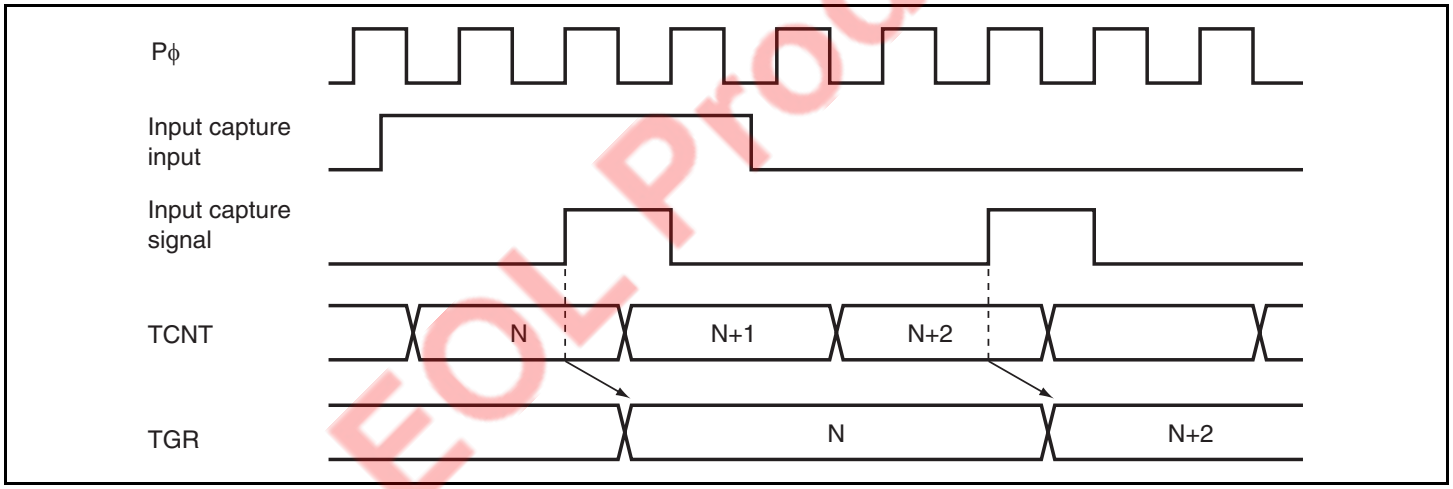


Figure 18.59 Input Capture Input Signal Timing

Timing for Counter Clearing by Compare Match/Input Capture: Figure 18.60 shows the timing when counter clearing on compare match is specified, and figure 18.61 shows the timing when counter clearing on input capture is specified.

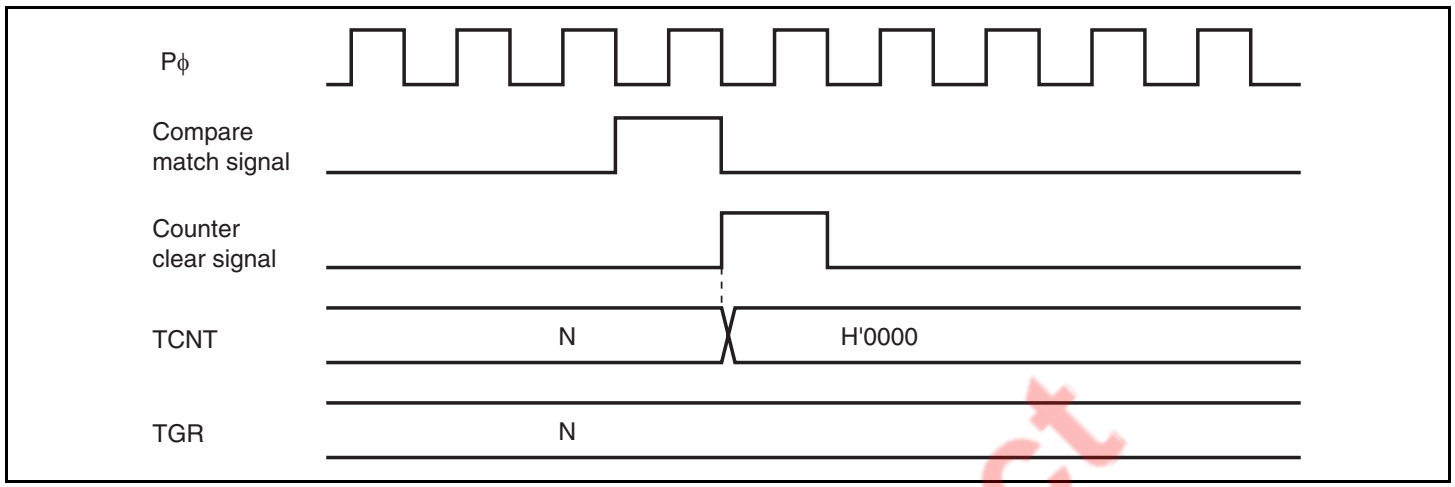


Figure 18.60 Counter Clear Timing (Compare Match)

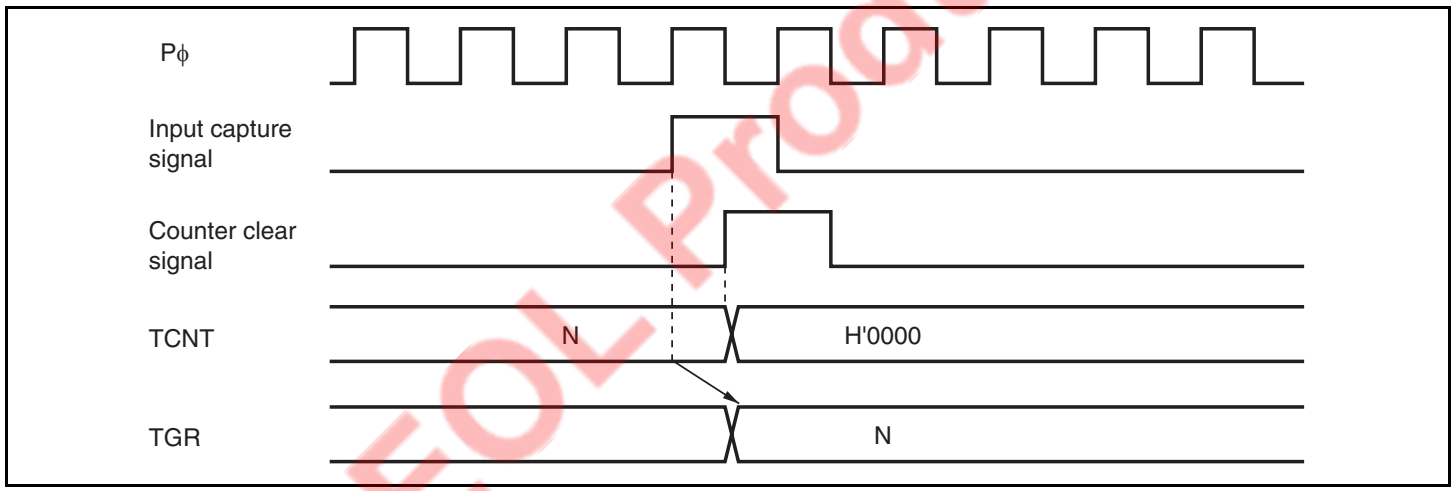


Figure 18.61 Counter Clear Timing (Input Capture)

Buffer Operation Timing: Figures 18.62 and 18.63 show the timing in buffer operation.

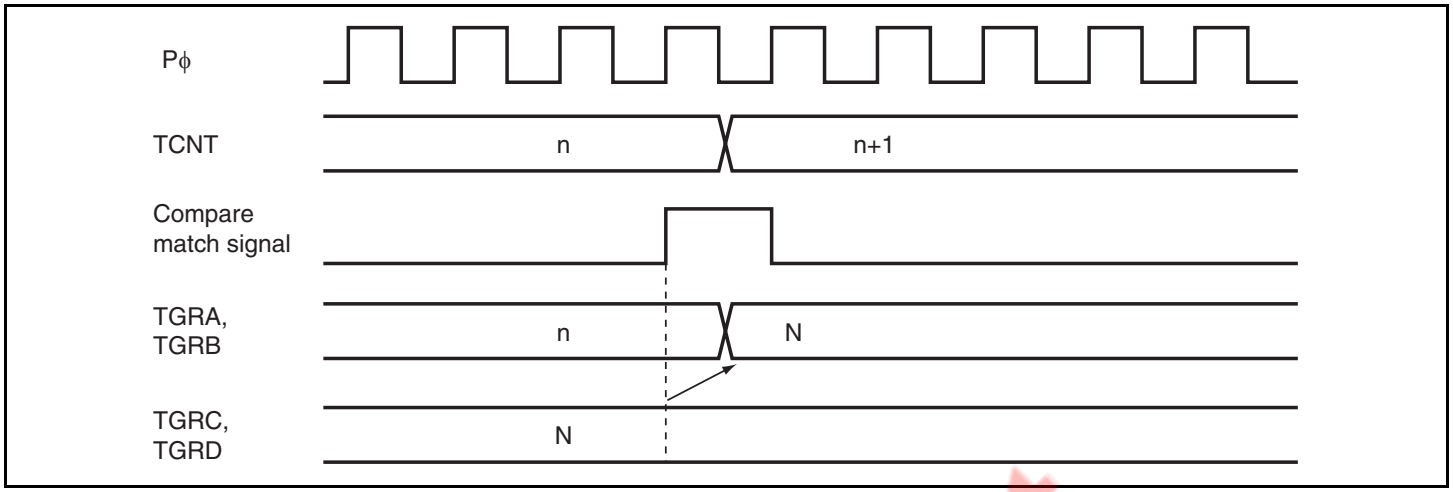


Figure 18.62 Buffer Operation Timing (Compare Match)

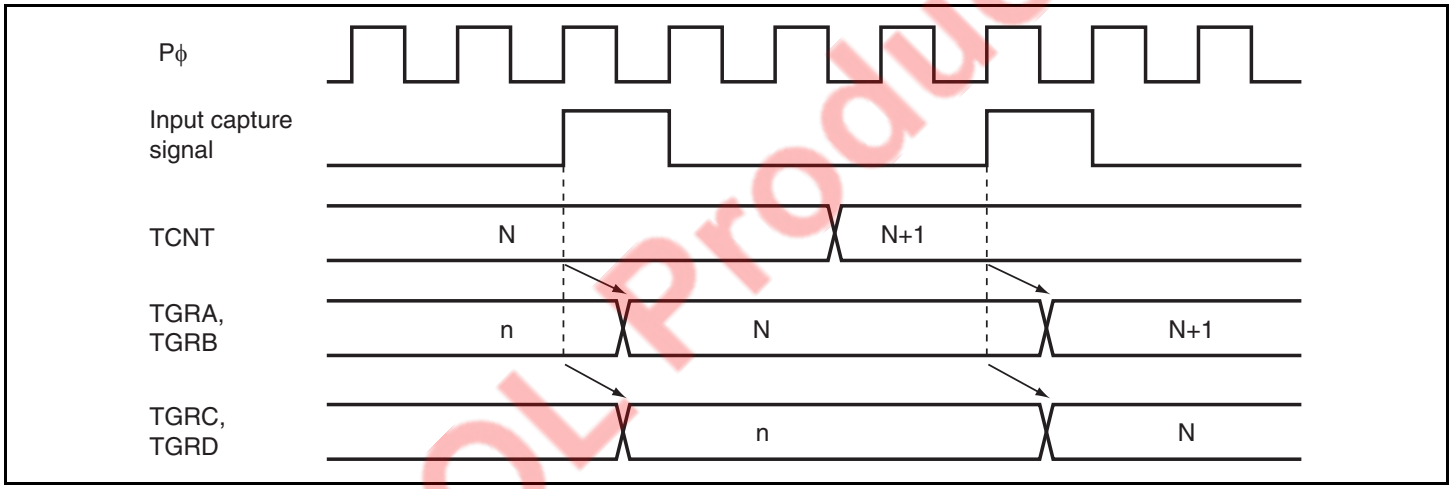


Figure 18.63 Buffer Operation Timing (Input Capture)

18.6.2 Interrupt Signal Timing

TGF Flag Setting Timing in Case of Compare Match: Figure 18.64 shows the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

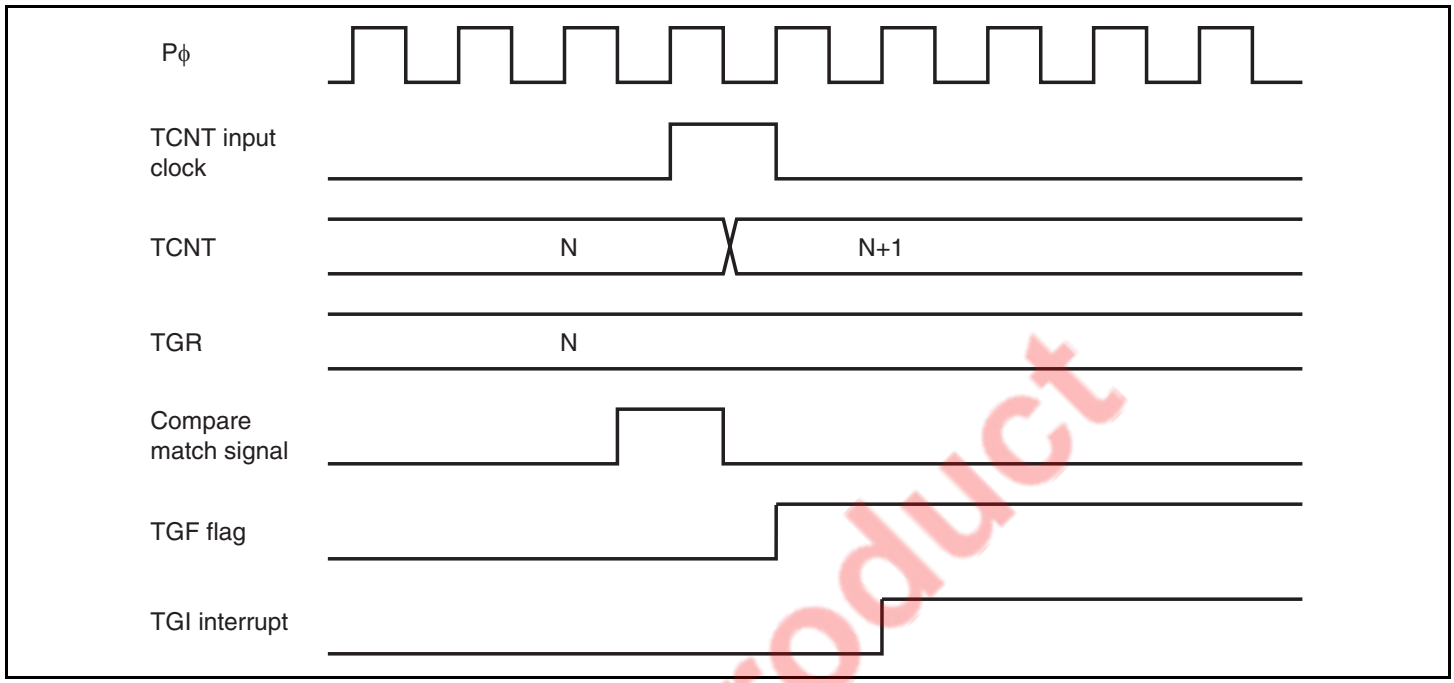


Figure 18.64 TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture: Figure 18.65 shows the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

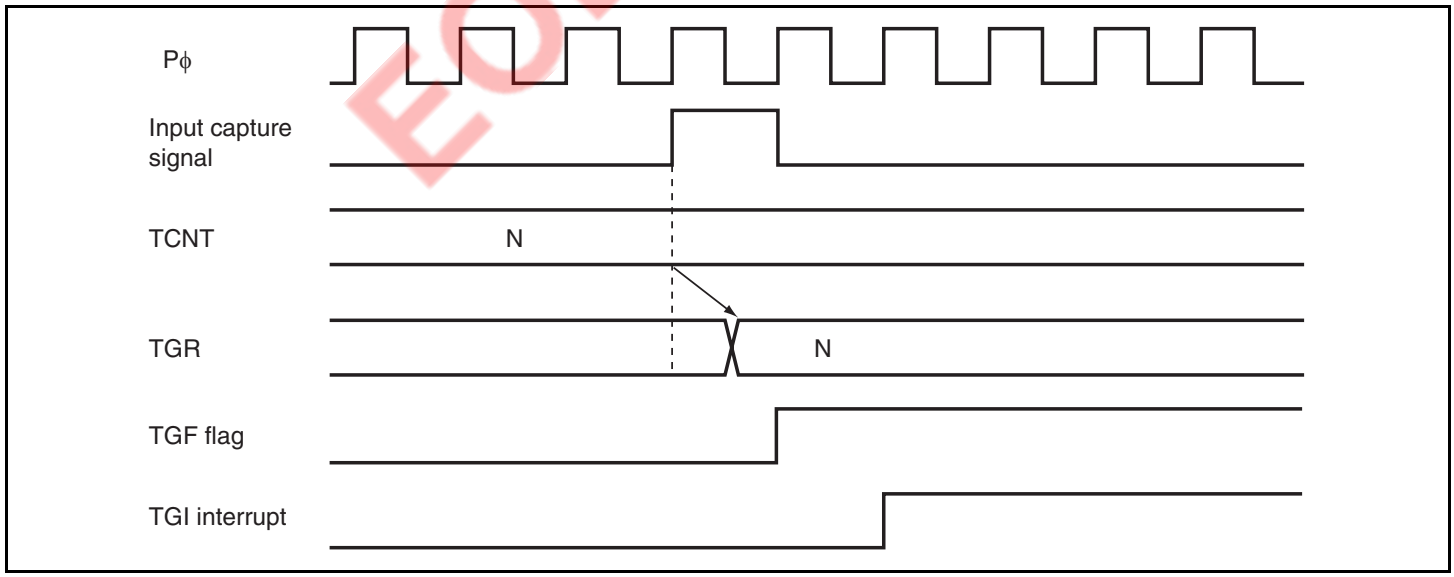


Figure 18.65 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing: Figure 18.66 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 18.67 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

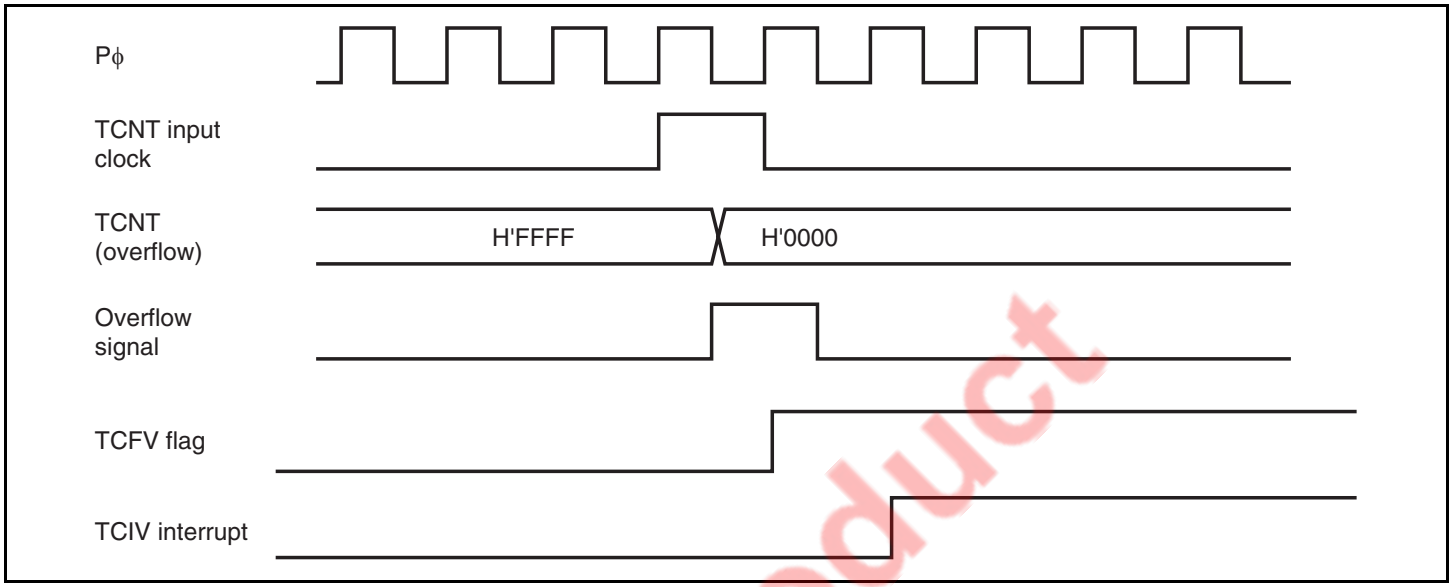


Figure 18.66 TCIV Interrupt Setting Timing

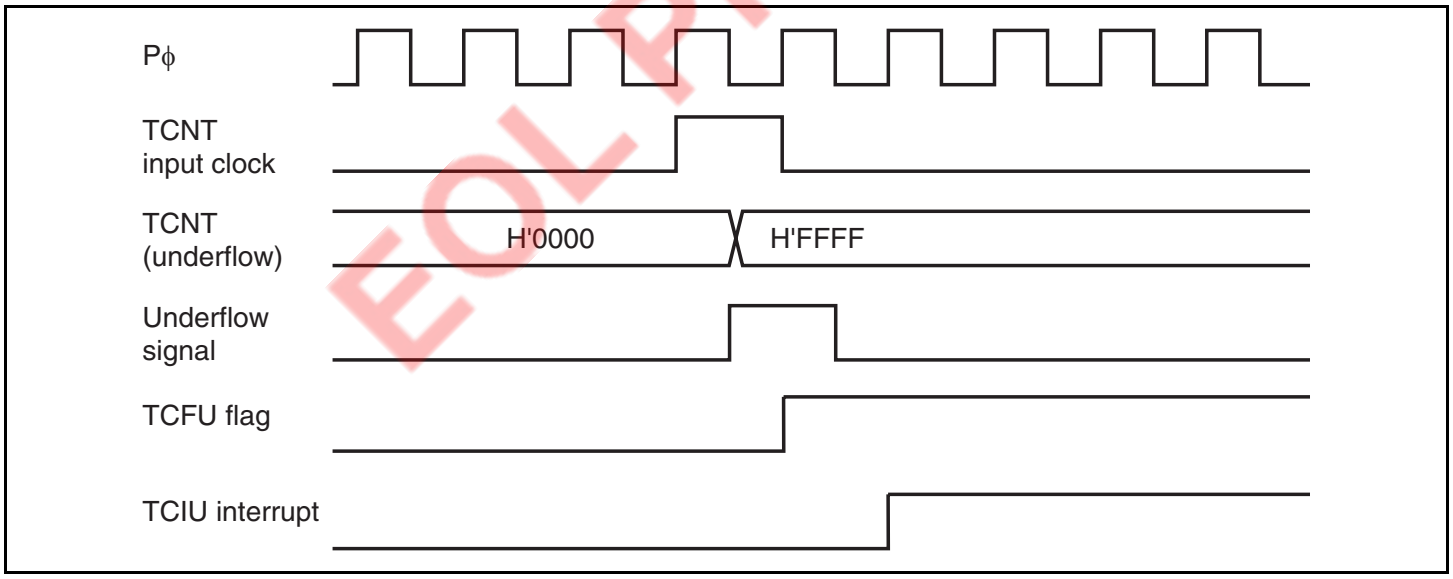


Figure 18.67 TCIU Interrupt Setting Timing

Status Flag Clearing Timing: After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DMA is activated, the flag is cleared automatically. Figure 18.68 shows the timing for status flag clearing by the CPU, and figure 18.69 shows the timing for status flag clearing by the DMA.

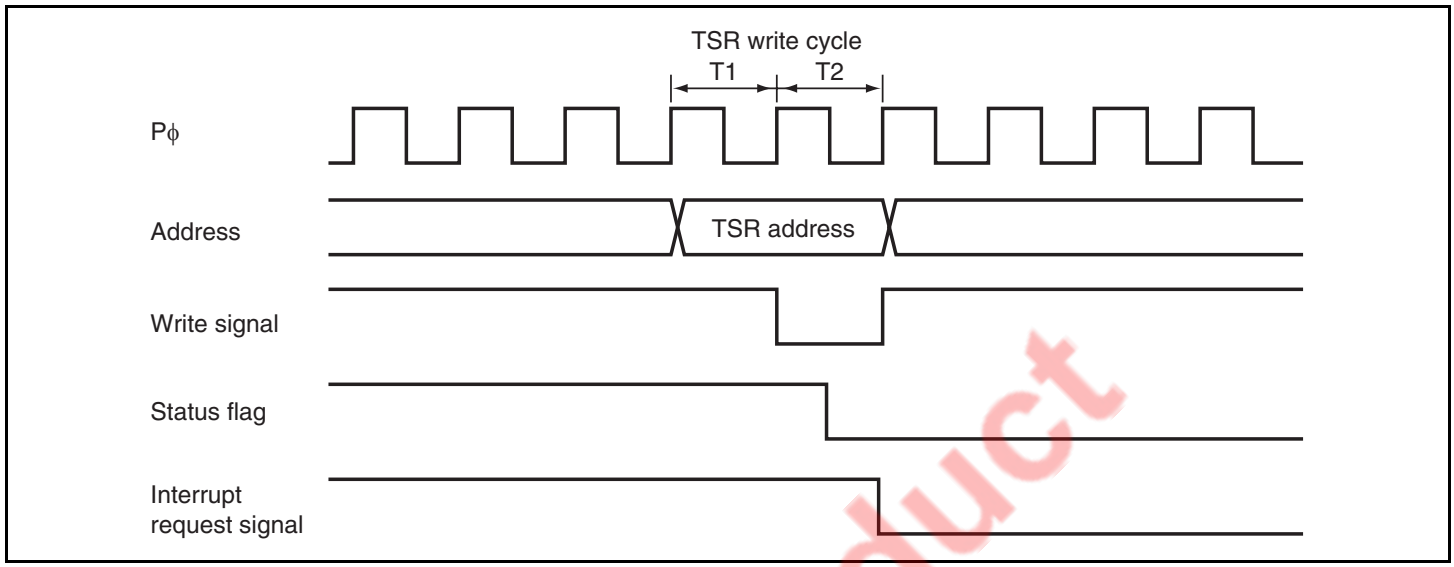


Figure 18.68 Timing for Status Flag Clearing by the CPU

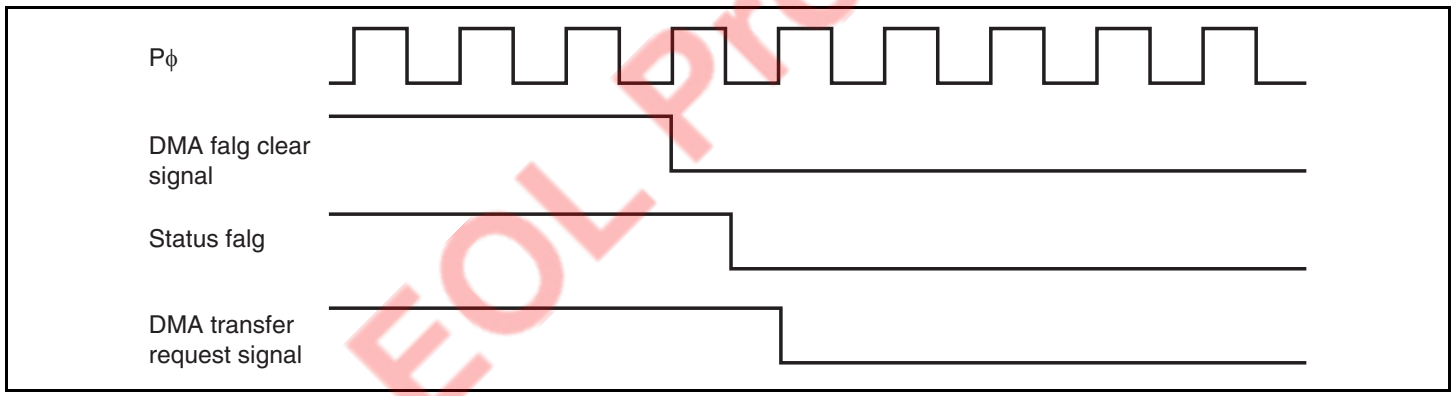


Figure 18.69 Timing for Status Flag Clearing by DMA Activation

18.7 Usage Notes

18.7.1 Module Standby Mode Setting

MTU operation can be disabled or enabled using the module standby register.

18.7.2 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 at narrower pulse widths.

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 18.70 shows the input clock conditions in phase counting mode.

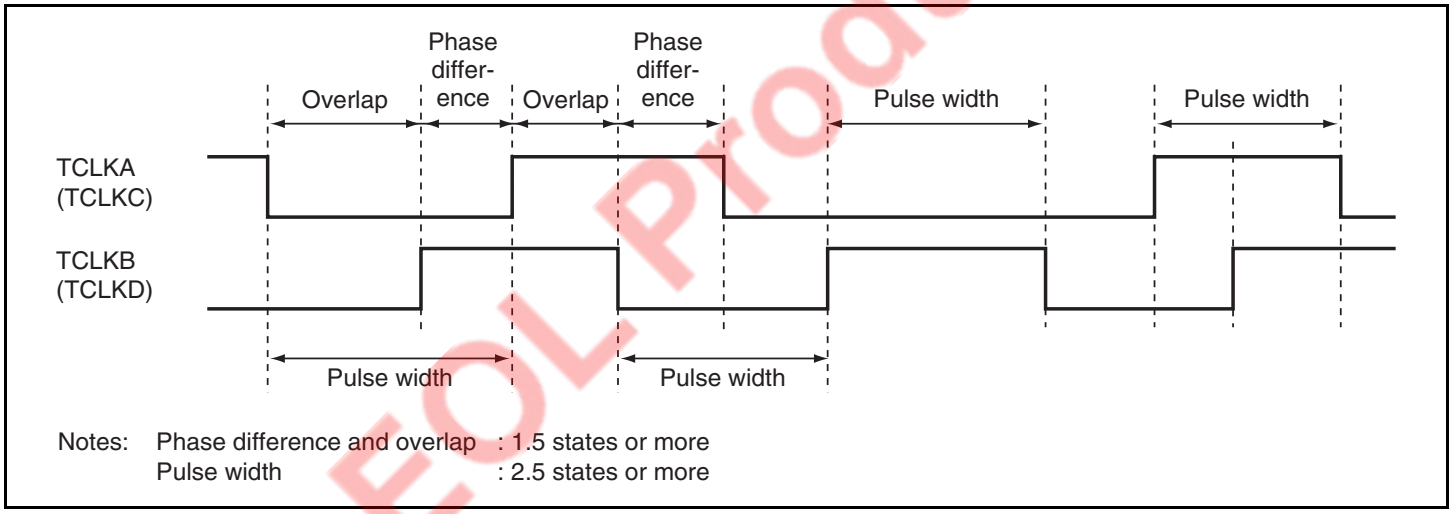


Figure 18.70 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

18.7.3 Caution on Period Setting

When counter clearing on 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:

$$f = \frac{P\phi}{(N + 1)}$$

Where f : Counter frequency
Pϕ : Peripheral clock operating frequency
N : TGR set value

18.7.4 Conflict between TCNT Write and Clear Operations

If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed.

Figure 18.71 shows the timing in this case.

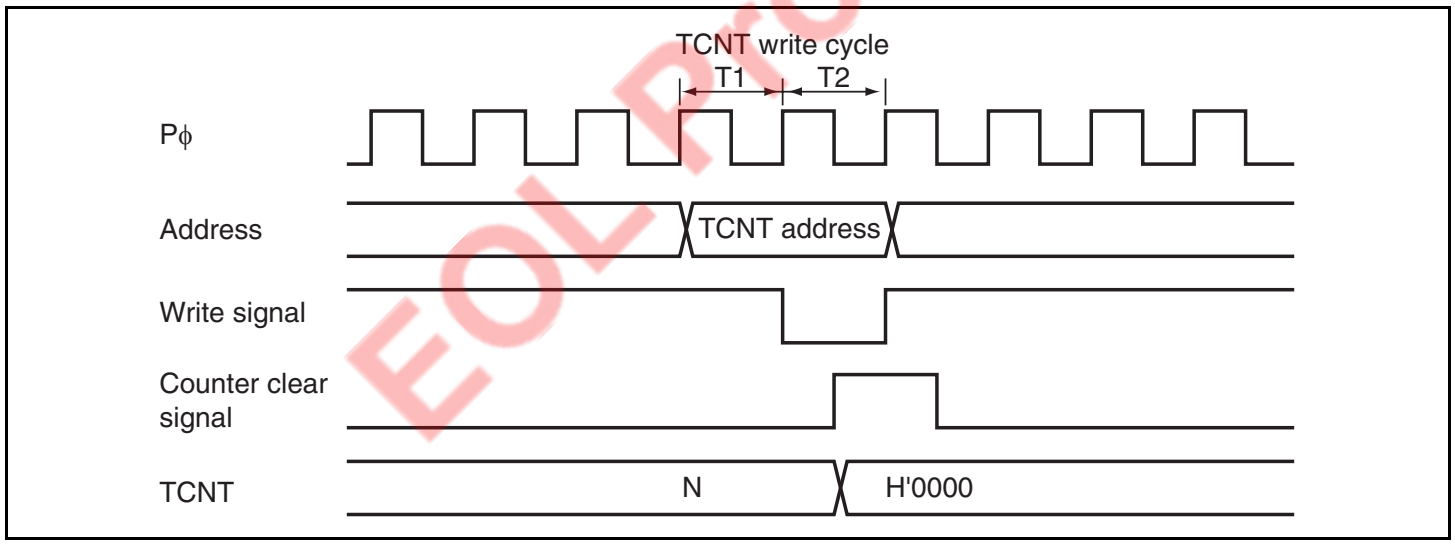


Figure 18.71 Conflict between TCNT Write and Clear Operations

18.7.5 Conflict between TCNT Write and Increment Operations

If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 18.72 shows the timing in this case.

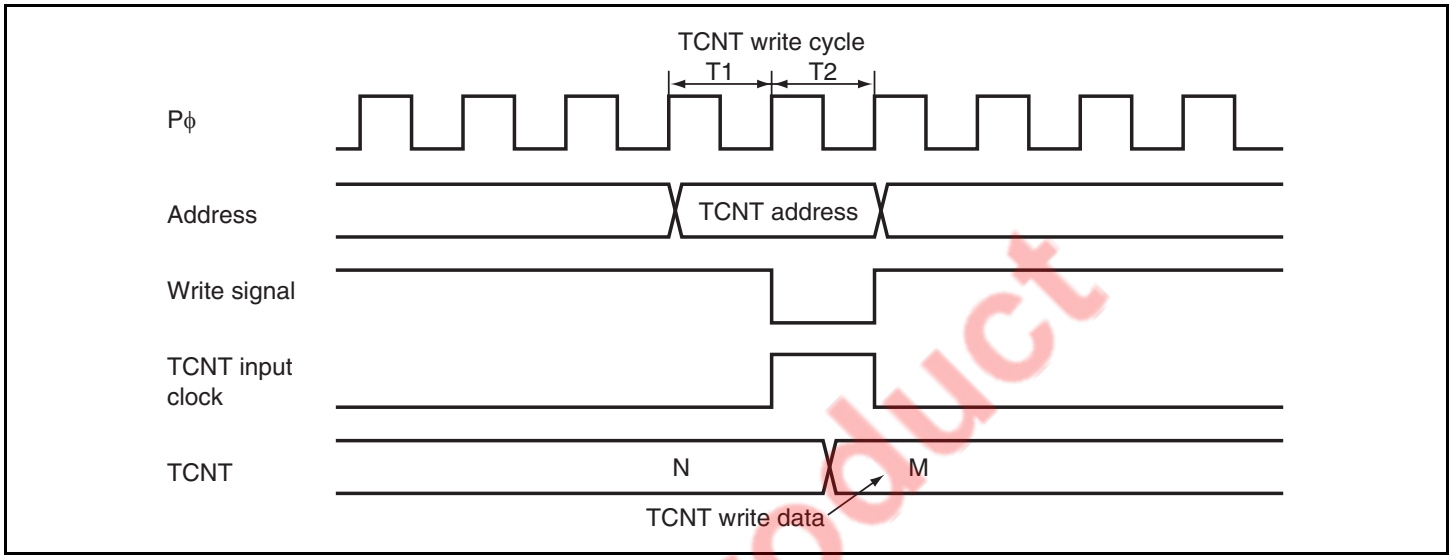


Figure 18.72 Conflict between TCNT Write and Increment Operations

18.7.6 Conflict between TGR Write and Compare Match

When a compare match occurs in the T2 state of a TGR write cycle, the TGR write is executed and the compare match signal is generated.

Figure 18.73 shows the timing in this case.

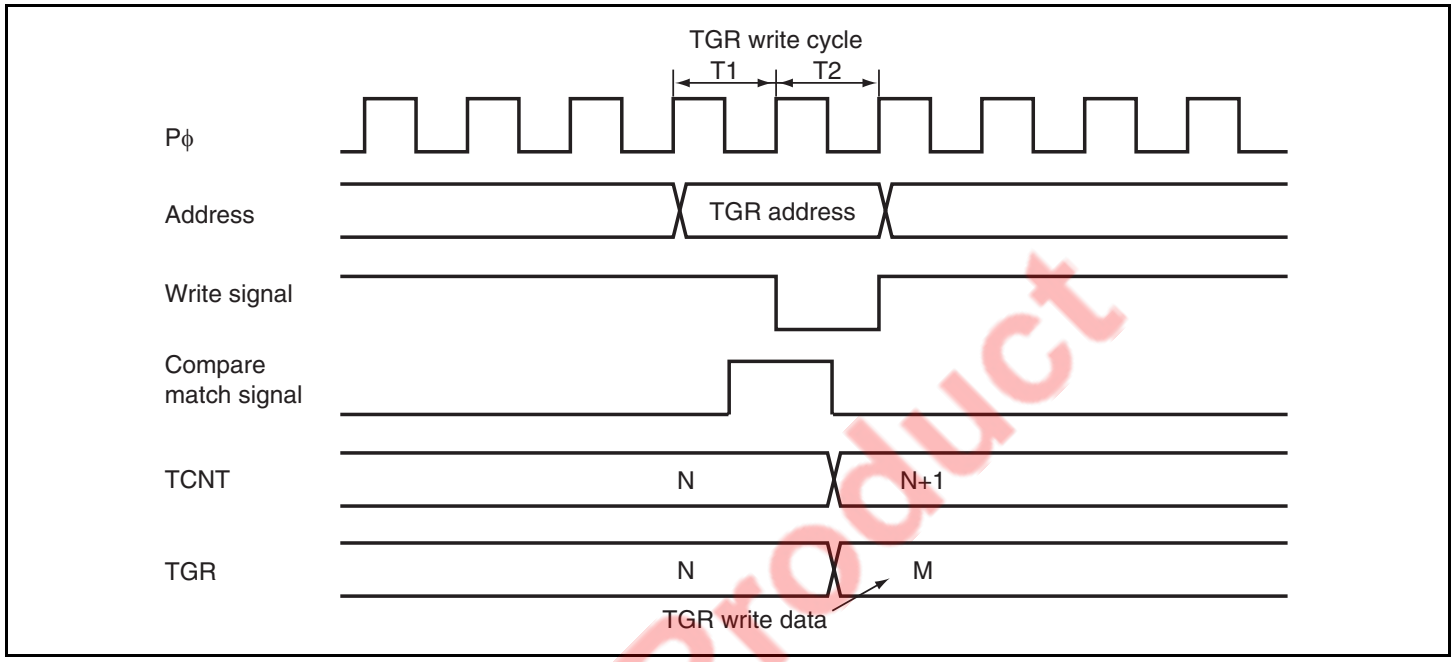


Figure 18.73 Conflict between TGR Write and Compare Match

18.7.7 Conflict between Buffer Register Write and Compare Match

If a compare match occurs in the T1 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation differs depending on channel 0 and channels 3 and 4: data on channel 0 is that after write, and on channels 3 and 4, before write.

Figures 18.74 and 18.75 show the timing in this case.

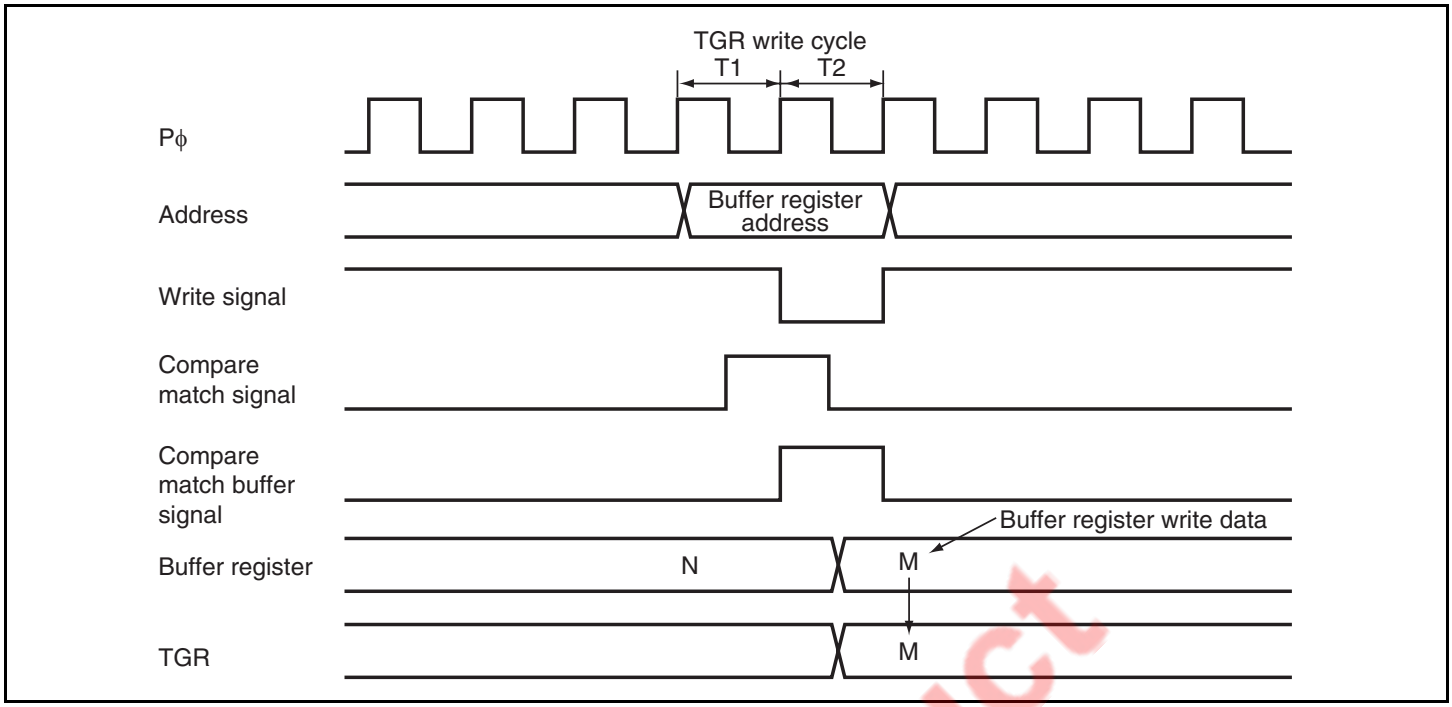


Figure 18.74 Conflict between Buffer Register Write and Compare Match (Channel 0)

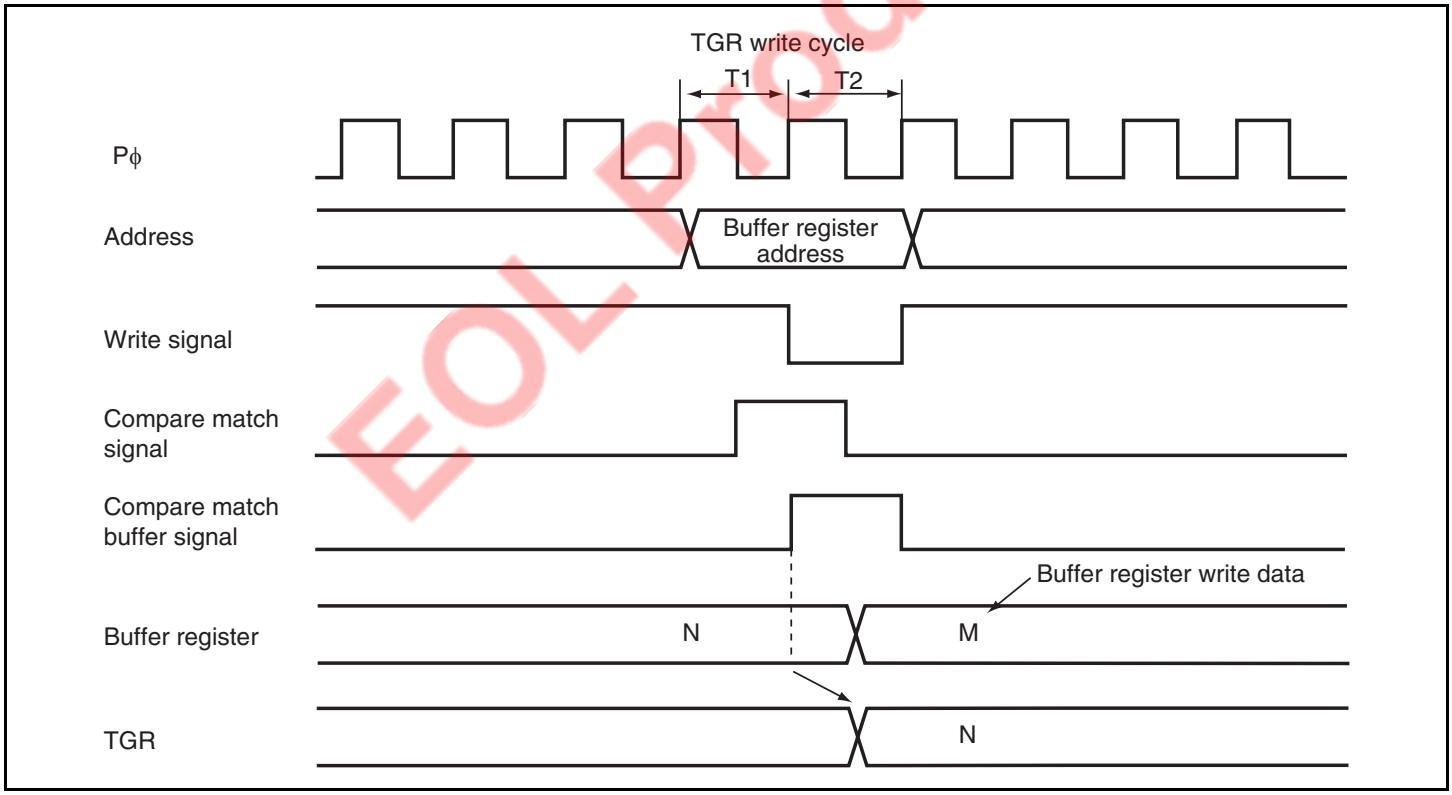


Figure 18.75 Conflict between Buffer Register Write and Compare Match (Channels 3 and 4)

18.7.8 Conflict between TGR Read and Input Capture

If an input capture signal is generated in the T2 state of a TGR read cycle, the data that is read will be that in the buffer after input capture transfer.

Figure 18.76 shows the timing in this case.

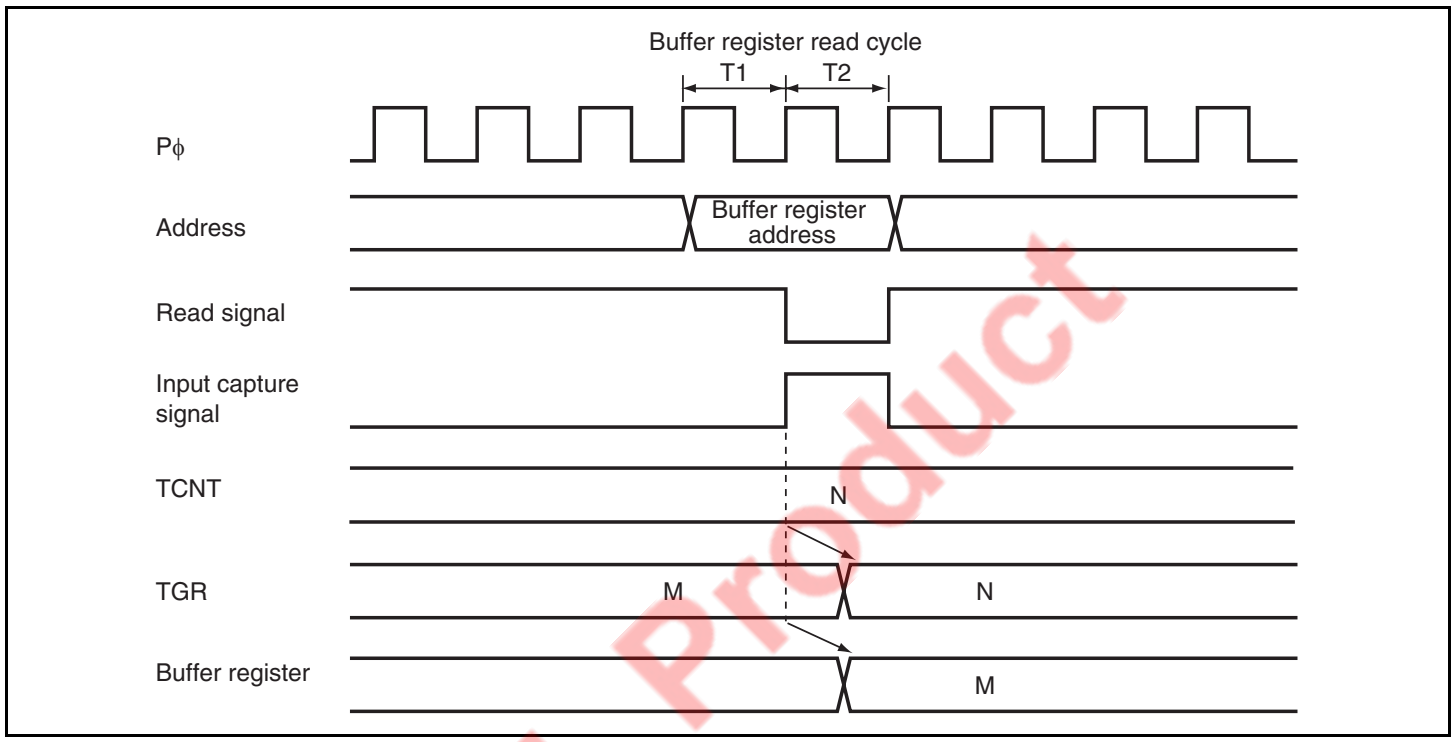


Figure 18.76 Conflict between TGR Read and Input Capture

18.7.9 Conflict between TGR Write and Input Capture

If an input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed.

Figure 18.77 shows the timing in this case.

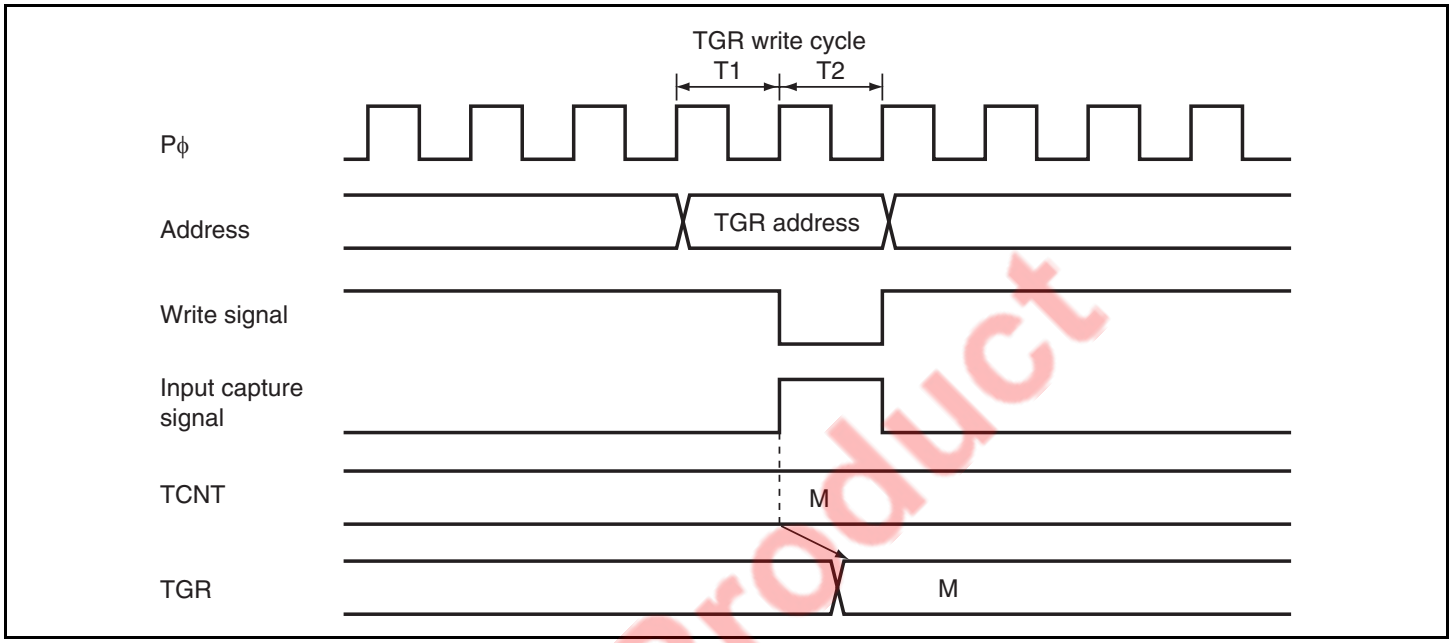


Figure 18.77 Conflict between TGR Write and Input Capture

18.7.10 Conflict between Buffer Register Write and Input Capture

If an input capture signal is generated in the T2 state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 18.78 shows the timing in this case.

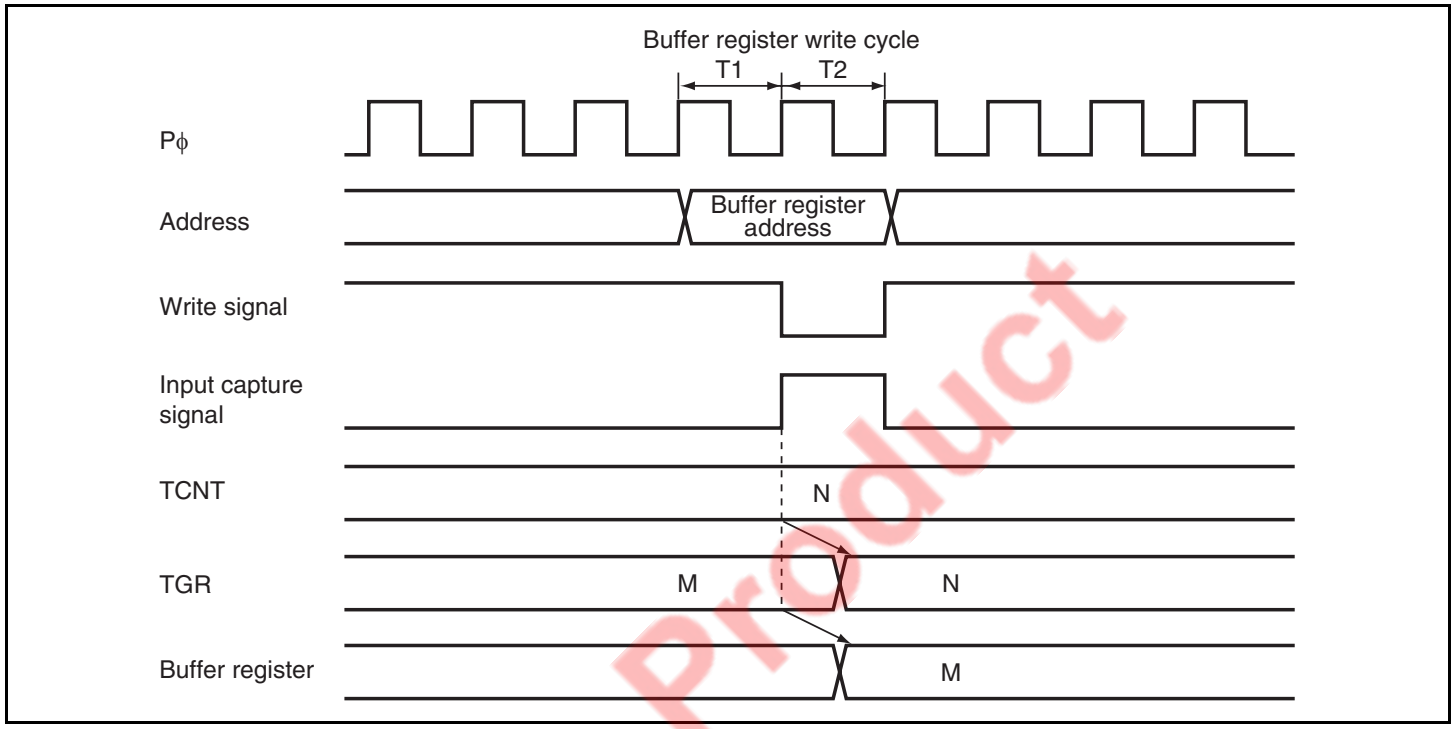


Figure 18.78 Conflict between Buffer Register Write and Input Capture

18.7.11 TCNT2 Write and Overflow/Underflow Conflict in Cascade Connection

With timer counters TCNT1 and TCNT2 in a cascade connection, when a conflict occurs during TCNT_1 count (during a TCNT_2 overflow/underflow) in the T_2 state of the TCNT_2 write cycle, the write to TCNT_2 is conducted, and the TCNT_1 count signal is disabled. At this point, if there is match with TGRA_1 and the TCNT_1 value, a compare signal is issued. Furthermore, when the TCNT_1 count clock is selected as the input capture source of channel 0, TGRA_0 to D_0 carry out the input capture operation. In addition, when the compare match/input capture is selected as the input capture source of TGRB_1, TGRB_1 carries out input capture operation. The timing is shown in figure 18.79.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setting TCNT clearing.

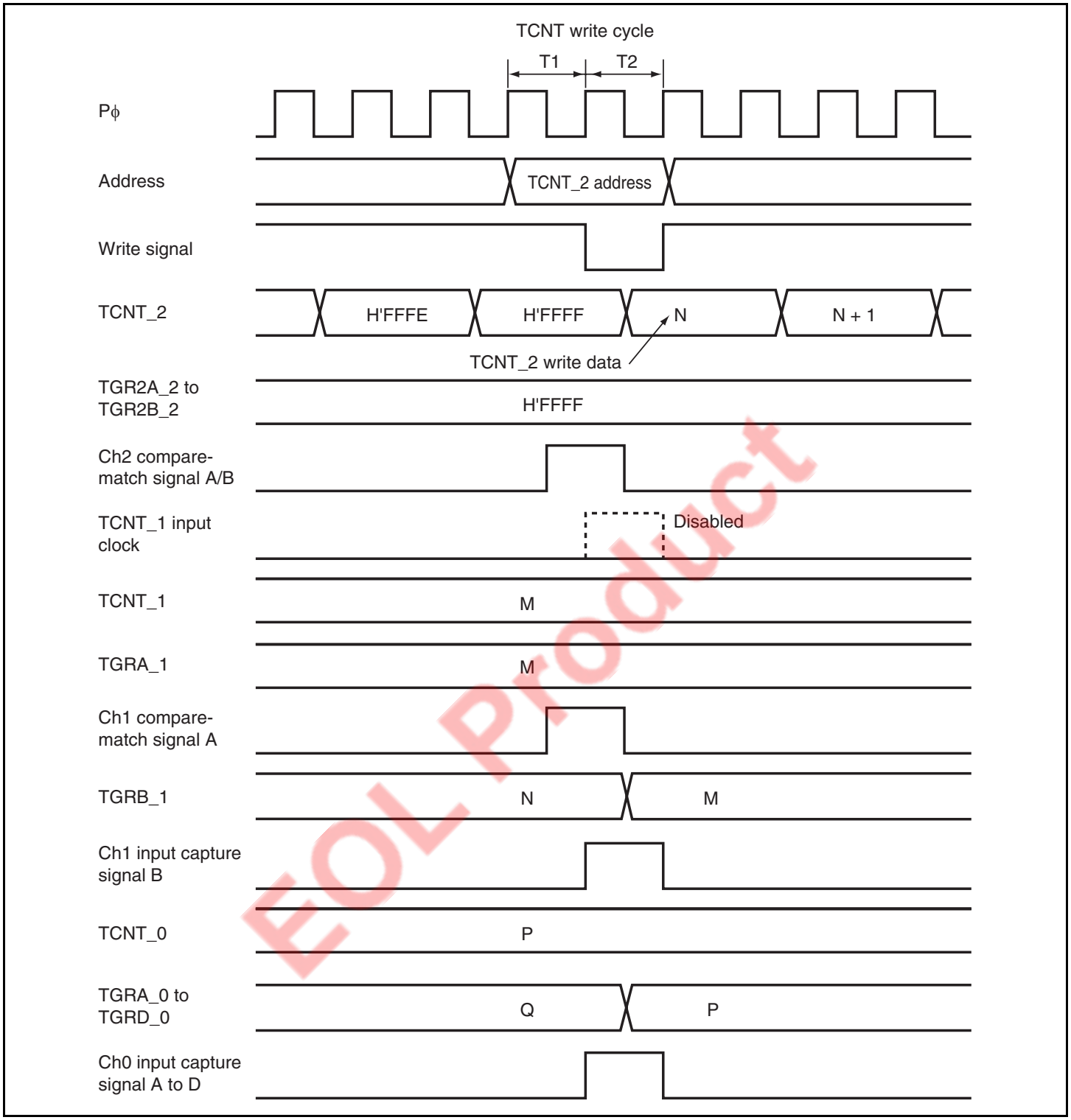


Figure 18.79 TCNT_2 Write and Overflow/Underflow Conflict with Cascade Connection

18.7.12 Counter Value during Complementary PWM Mode Stop

When counting operation is stopped with TCNT_3 and TCNT_4 in complementary PWM mode, TCNT_3 has the timer dead time register (TDDR) value, and TCNT_4 is set to H'0000.

When restarting complementary PWM mode, counting begins automatically from the initialized state. This explanatory diagram is shown in figure 18.80.

When counting begins in another operating mode, be sure that TCNT_3 and TCNT_4 are set to the initial values.

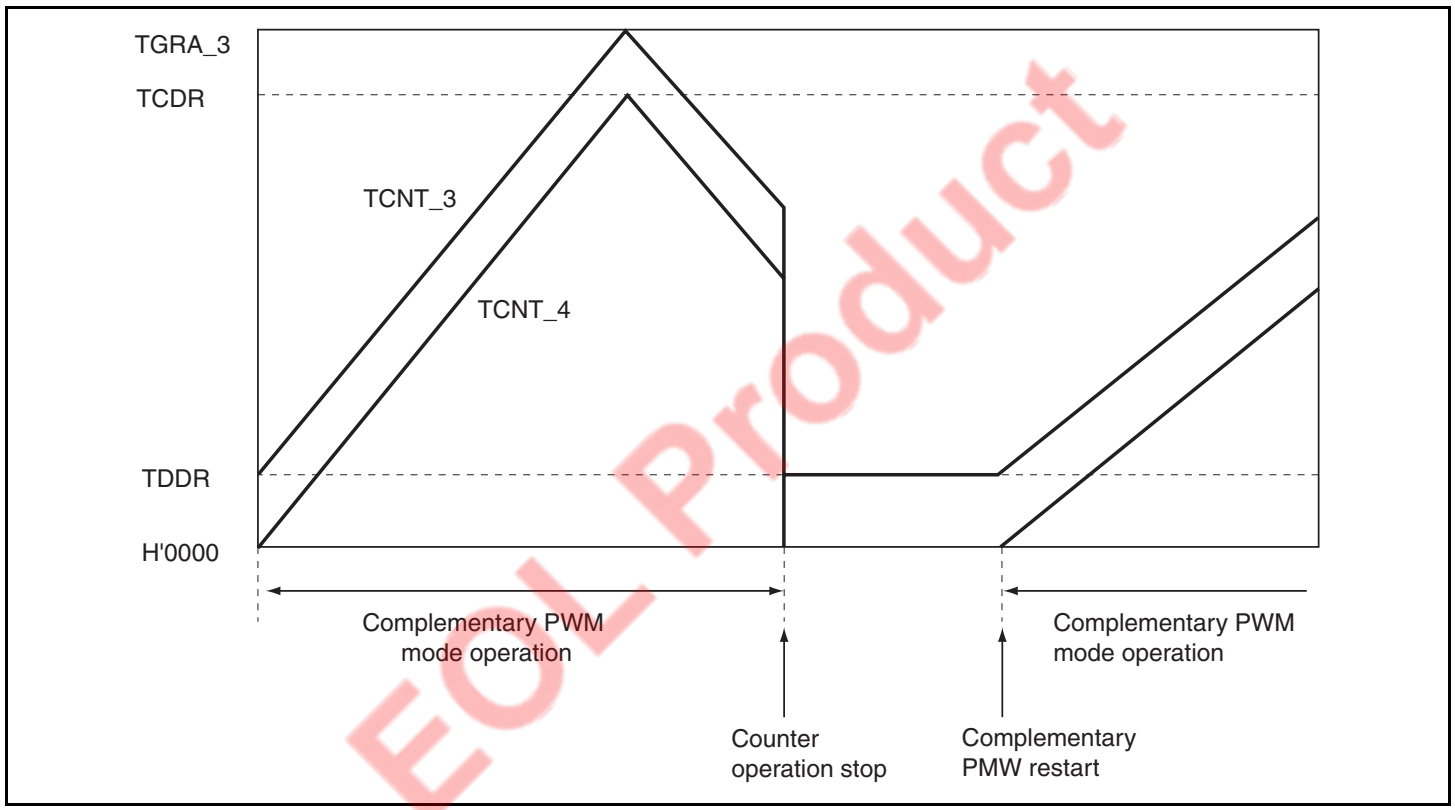


Figure 18.80 Counter Value during Complementary PWM Mode Stop

18.7.13 Buffer Operation Setting in Complementary PWM Mode

In complementary PWM mode, conduct rewrites by buffer operation for the PWM cycle setting register (TGRA_3), timer cycle data register (TCDR), and duty setting registers (TGRB_3, TRGA_4, and TGRB_4).

In complementary PWM mode, channel 3 and channel 4 buffers operate in accordance with bit settings BFA and BFB of TMDR_3. When TMDR_3's BFA bit is set to 1, TGRC_3 functions as a

buffer register for TGRA_3. At the same time, TGRC_4 functions as the buffer register for TRGA_4, while the TCBR functions as the TCDR's buffer register.

18.7.14 Reset Sync PWM Mode Buffer Operation and Compare Match Flag

When setting buffer operation for reset sync PWM mode, set the BFA and BFB bits of TMDR_4 to 0. The TIOC4C pin will be unable to produce its waveform output if the BFA bit of TMDR_4 is set to 1.

In reset sync PWM mode, the channel 3 and channel 4 buffers operate in accordance with the BFA and BFB bit settings of TMDR_3. For example, if the BFA bit of TMDR_3 is set to 1, TGRC_3 functions as the buffer register for TGRA_3. At the same time, TGRC_4 functions as the buffer register for TRGA_4.

The TGFC bit and TGFD bit of TSR_3 and TSR_4 are not set when TGRC_3 and TGRD_3 are operating as buffer registers.

Figure 18.81 shows an example of operations for TGR_3, TGR_4, TIOC3, and TIOC4, with TMDR_3's BFA and BFB bits set to 1, and TMDR_4's BFA and BFB bits set to 0.

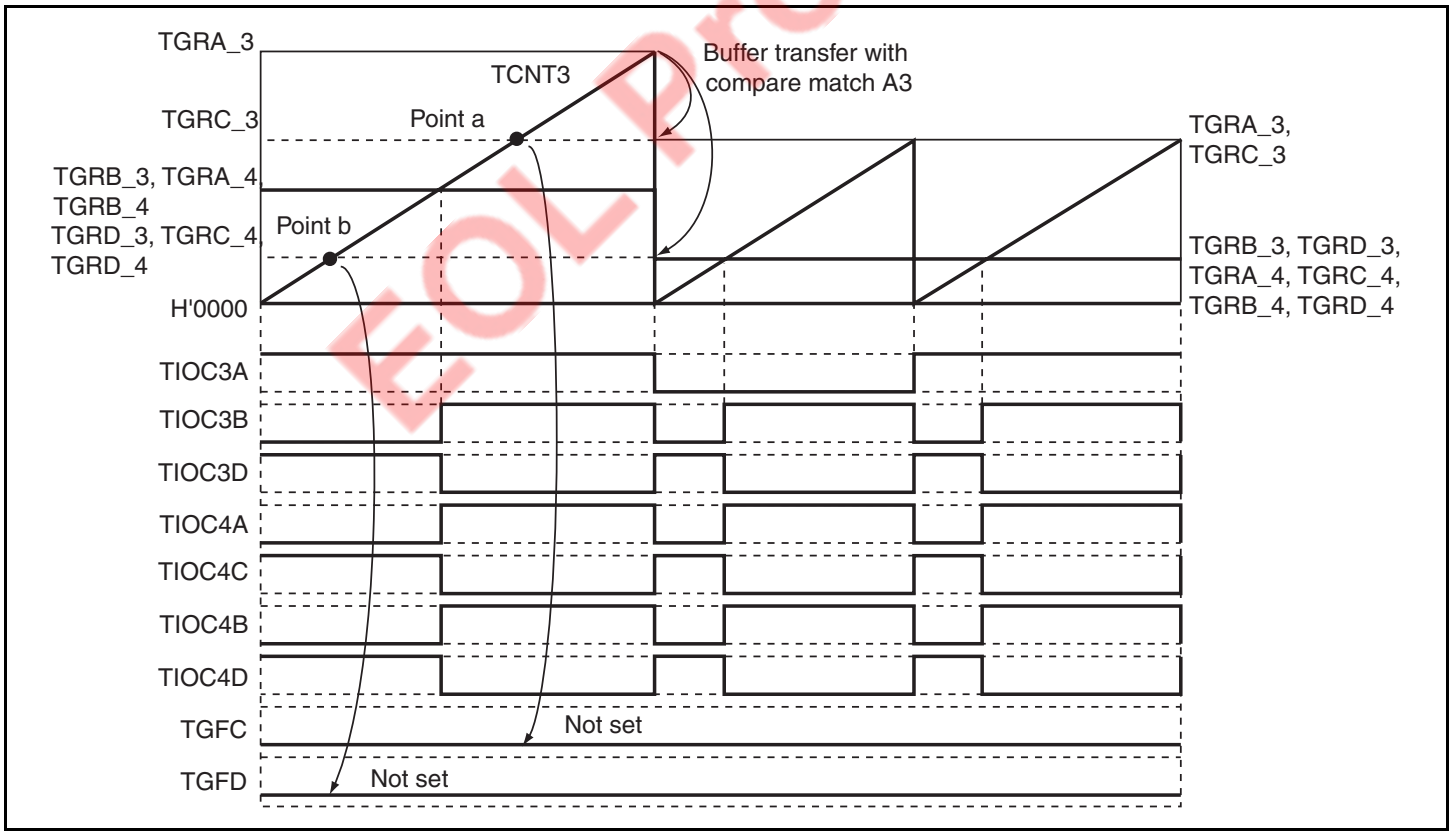


Figure 18.81 Buffer Operation and Compare-Match Flags in Reset Sync PWM Mode

18.7.15 Overflow Flags in Reset Sync PWM Mode

When set to reset sync PWM mode, TCNT_3 and TCNT_4 start counting when the CST3 bit of TSTR is set to 1. At this point, TCNT_4's count clock source and count edge obey the TCR_3 setting.

In reset sync PWM mode, with cycle register TGRA_3's set value at H'FFFF, when specifying TGR3A compare-match for the counter clear source, TCNT_3 and TCNT_4 count up to H'FFFF, then a compare-match occurs with TGRA_3, and TCNT_3 and TCNT_4 are both cleared. At this point, TSR's overflow flag TCFV bit is not set.

Figure 18.82 shows a TCFV bit operation example in reset sync PWM mode with a set value for cycle register TGRA_3 of H'FFFF, when a TGRA_3 compare-match has been specified without synchronous setting for the counter clear source.

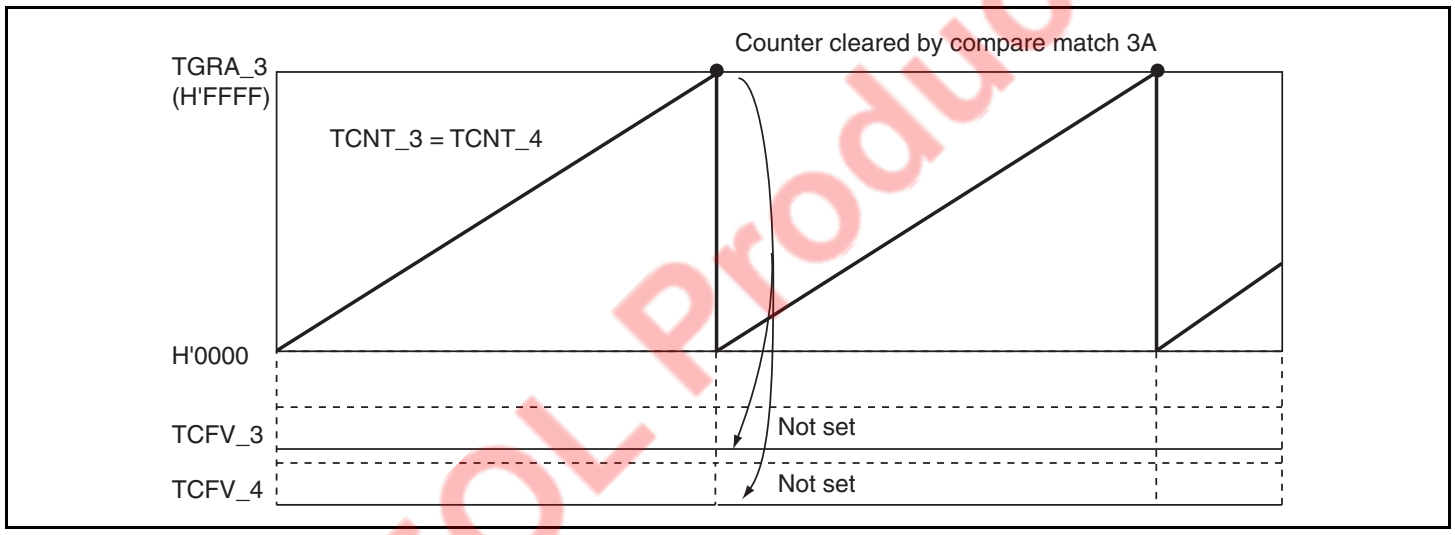


Figure 18.82 Reset Sync PWM Mode Overflow Flag

18.7.16 Conflict 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 18.83 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

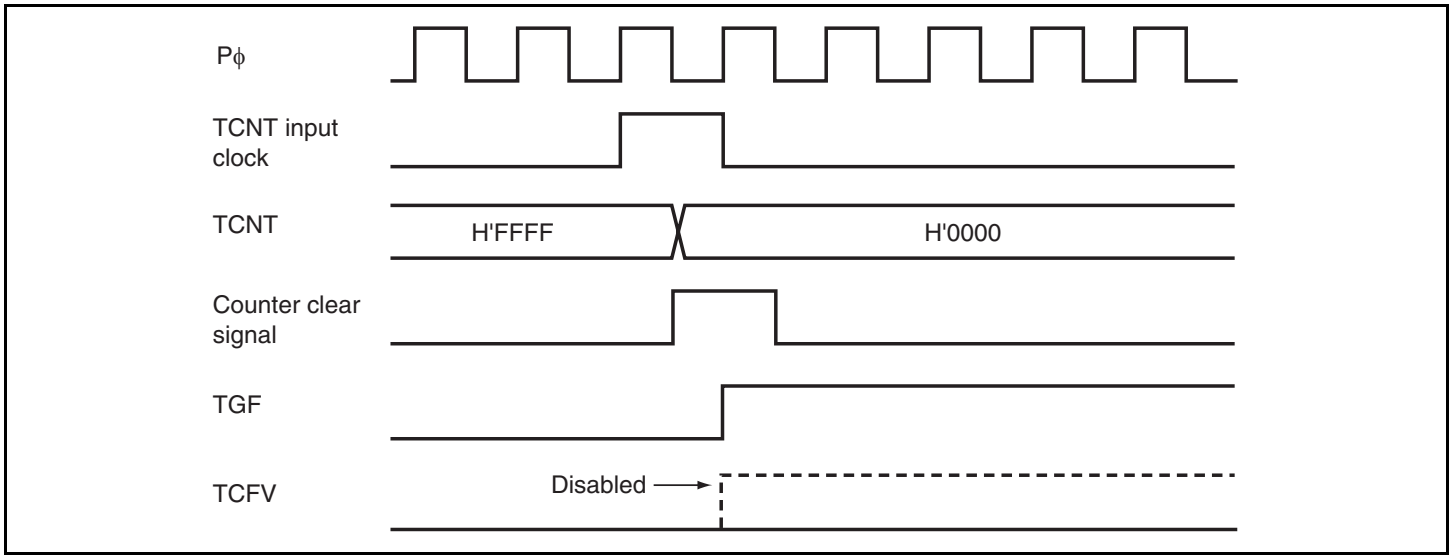


Figure 18.83 Conflict between Overflow and Counter Clearing

18.7.17 Conflict between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 18.84 shows the operation timing when there is conflict between TCNT write and overflow.

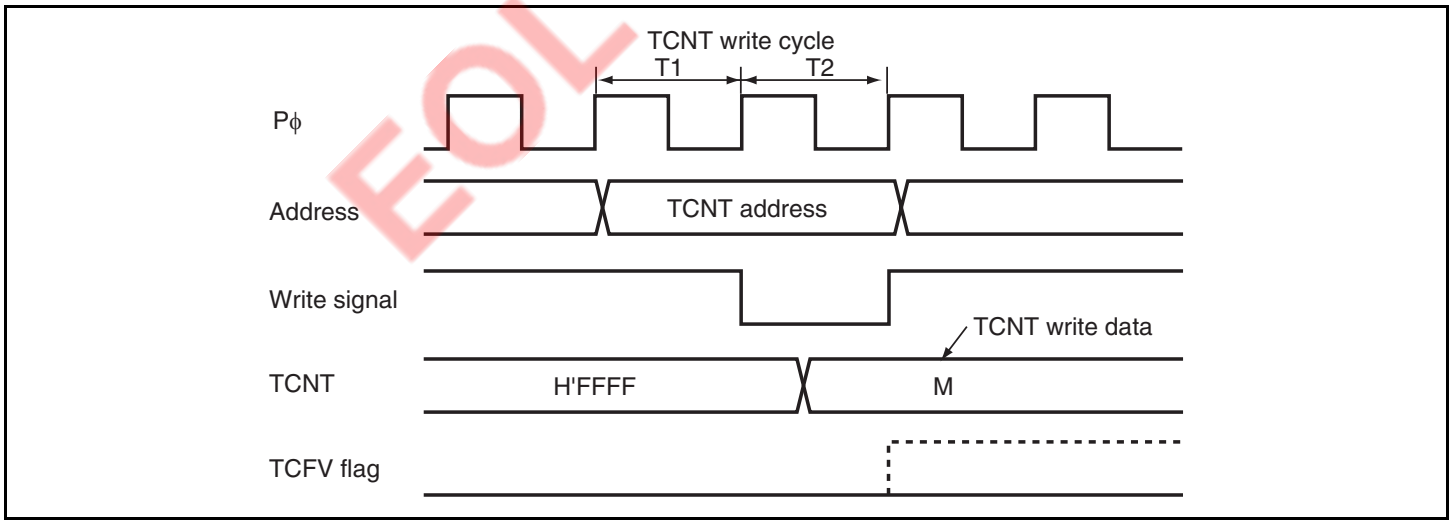


Figure 18.84 Conflict between TCNT Write and Overflow

18.7.18 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronous PWM Mode

When making a transition from channel 3 or 4 normal operation or PWM mode 1 to reset-synchronous PWM mode, if the counter is halted with the output pins (TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, TIOC4D) in the high-impedance state, followed by the transition to reset-synchronous PWM mode and operation in that mode, the initial pin output will not be correct.

When making a transition from normal operation to reset-synchronous PWM mode, write H'11 to registers TIORH_3, TIORL_3, TIORH_4, and TIORL_4 to initialize the output pins to low level output, then set an initial register value of H'00 before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronous PWM mode, first switch to normal operation, then initialize the output pins to low level output and set an initial register value of H'00 before making the transition to reset-synchronous PWM mode.

18.7.19 Output Level in Complementary PWM Mode and Reset-Synchronous PWM Mode

When channels 3 and 4 are in complementary PWM mode or reset-synchronous PWM mode, the PWM waveform output level is set with the OLSP and OLSN bits in the timer output control register (TOCR). In the case of complementary PWM mode or reset-synchronous PWM mode, TIOR should be set to H'00.

18.7.20 Interrupts in Module Standby Mode

If module standby mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DMA activation source. Interrupts should therefore be disabled before entering module standby mode.

18.7.21 Simultaneous Input Capture of TCNT_1 and TCNT_2 in Cascade Connection

When cascade-connected timer counters (TCNT_1 and TCNT_2) are operated, cascade values cannot be captured even if input capture is executed simultaneously with TIOC1A or TIOC2A and TIOC1B or TIOC2B.

18.8 MTU Output Pin Initialization

18.8.1 Operating Modes

The MTU has the following six operating modes. Waveform output is possible in all of these modes.

- Normal mode (channels 0 to 4)
- PWM mode 1 (channels 0 to 4)
- PWM mode 2 (channels 0 to 2)
- Phase counting modes 1 to 4 (channels 1 and 2)
- Complementary PWM mode (channels 3 and 4)
- Reset-synchronous PWM mode (channels 3 and 4)

The MTU output pin initialization method for each of these modes is described in this section.

18.8.2 Reset Start Operation

The MTU output pins (TIOC*) are initialized to low by a power-on reset or in standby mode. Since MTU pin function selection is performed by the pin function controller (PFC), when the PFC is set, the MTU pin states at that point are output to the ports. When MTU output is selected by the PFC immediately after a power-on reset, the MTU output initial level, low, is output directly at the port. When the active level is low, the system will operate at this point, and therefore the PFC setting should be made after initialization of the MTU output pins is completed.

Note: Channel number and port notation are substituted for *.

18.8.3 Operation in Case of Re-Setting Due to Error During Operation, etc.

If an error occurs during MTU operation, MTU output should be cut by the system. Cutoff is performed by switching the pin output to port output with the PFC and outputting the inverse of the active level. For large-current pins, output can also be cut by hardware, using port output enable (POE). The pin initialization procedures for re-setting due to an error during operation, etc., and the procedures for restarting in a different mode after re-setting, are shown below.

The MTU has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Possible mode transition combinations are shown in table 18.43.

Table 18.43 Mode Transition Combinations

	After					
Before	**Normal**	**PWM1**	**PWM2**	**PCM**	**CPWM**	**RPWM**
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	None	None
PCM	(17)	(18)	(19)	(20)	None	None
CPWM	(21)	(22)	None	None	(23) (24)	(25)
RPWM	(26)	(27)	None	None	(28)	(29)

[Legend]
Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1 to 4
CPWM: Complementary PWM mode
RPWM: Reset-synchronous PWM mode

The above abbreviations are used in some places in following descriptions.

18.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, Etc.

- When making a transition to a mode (Normal, PWM1, PWM2, PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of a TIOR setting.
- In PWM mode 1, since a waveform is not output to the TIOC*B (TIOC *D) pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 1.
- In PWM mode 2, since a waveform is not output to the cycle register pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 2.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, setting TIOR will not initialize the buffer register pins. If initialization is required, clear buffer mode, carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIOR will not initialize the TGRC pin. To initialize the TGRC pin, clear buffer mode, carry out initialization, then set buffer mode again.
- When making a transition to a mode (CPWM, RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, switch to normal mode and perform initialization with TIOR, then restore TIOR to its initial value, and temporarily disable channel 3 and 4 output with the timer output master enable register (TOER). Then operate the unit in accordance with the mode setting procedure (TOCR setting, TMDR setting, TOER setting).

Pin initialization procedures are described below for the numbered combinations in table 18.43. The active level is assumed to be low.

Note: Channel number is substituted for * indicated in this article.

(1) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Normal Mode

Figure 18.85 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

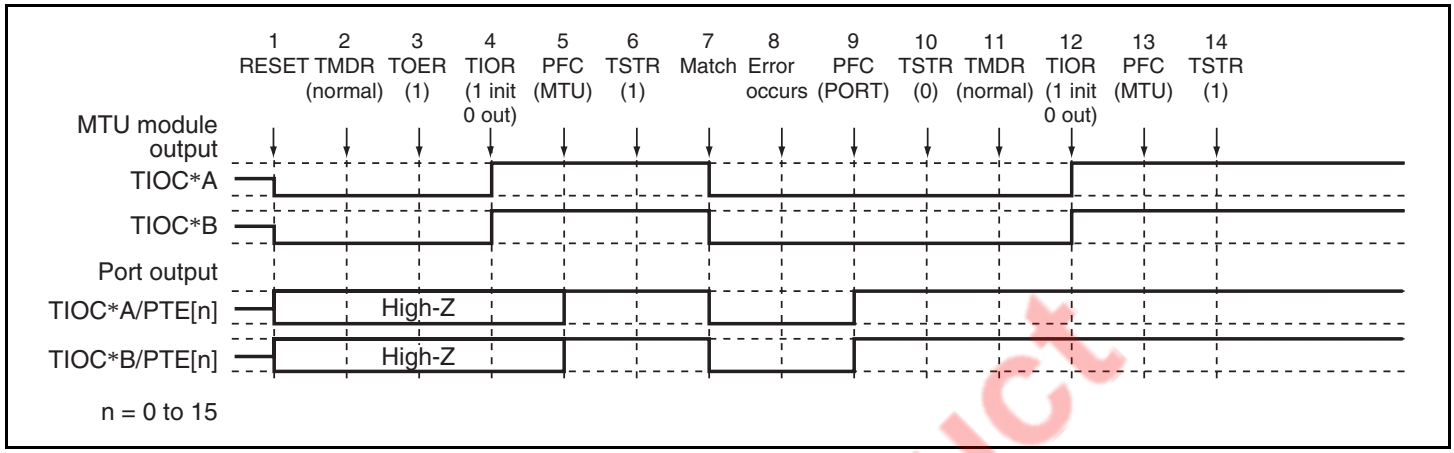


Figure 18.85 Error Occurrence in Normal Mode, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. After a reset, the TMDR setting is for normal mode.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
5. Set MTU output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Not necessary when restarting in normal mode.
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(2) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 18.86 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

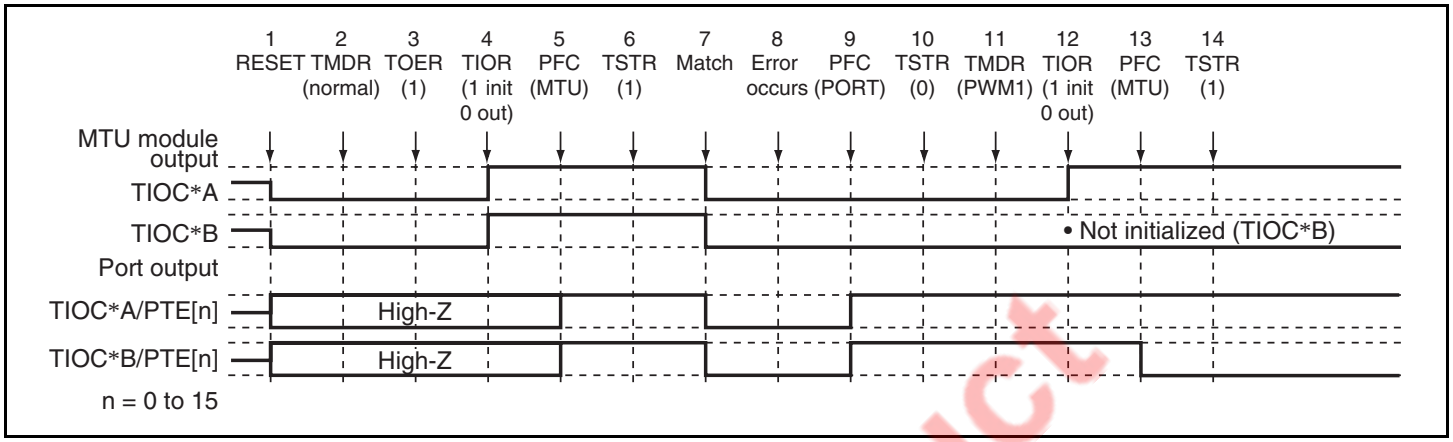


Figure 18.86 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 18.85.

11. Set PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 1.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(3) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 2

Figure 18.87 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

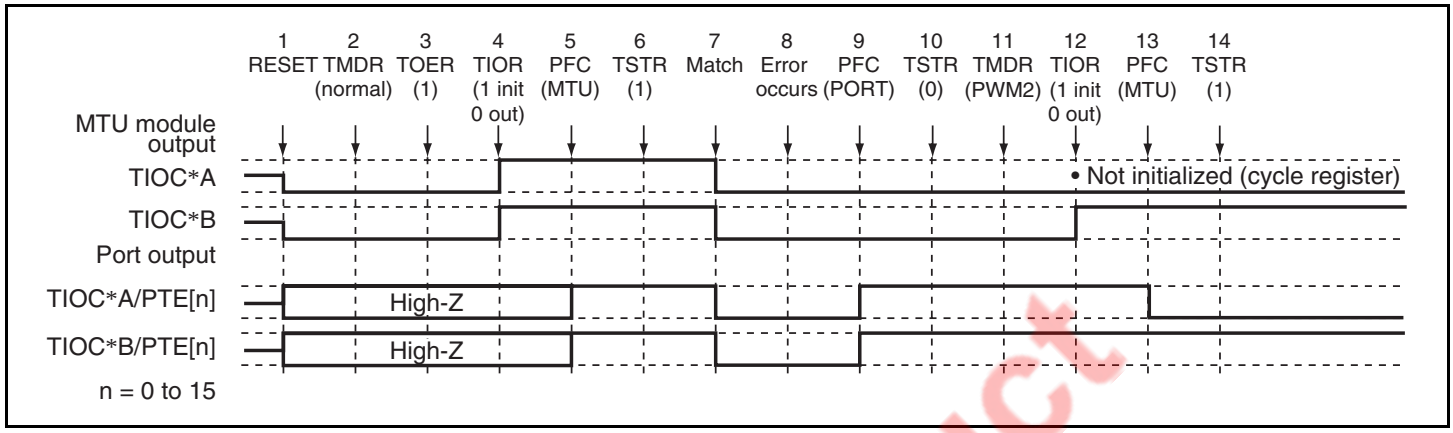


Figure 18.87 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

1 to 10 are the same as in figure 18.85.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 2.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0 to 2, and therefore TOER setting is not necessary.

(4) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Phase Counting Mode

Figure 18.88 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

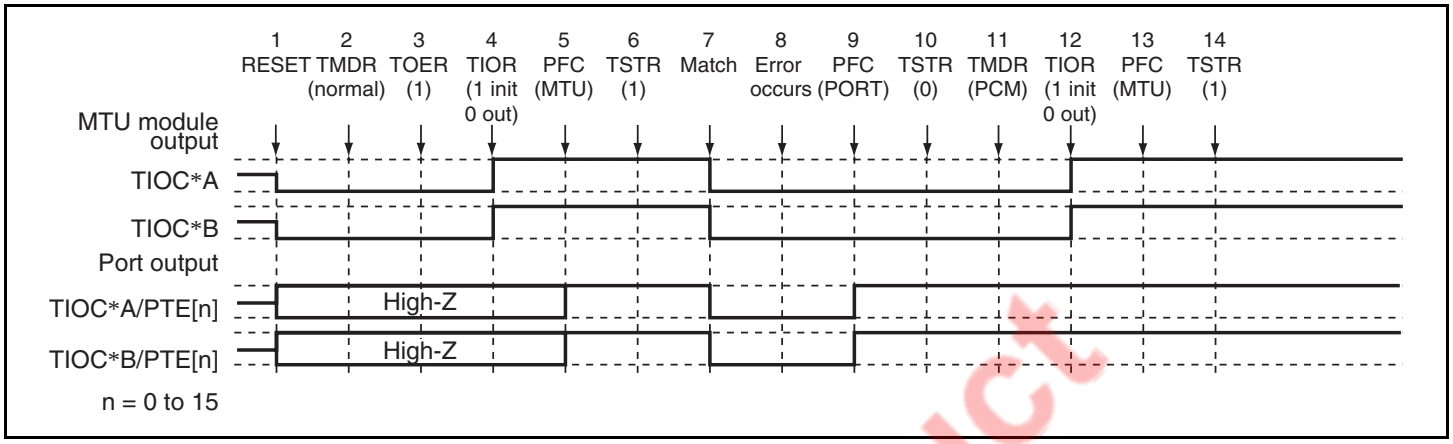


Figure 18.88 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 18.85.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER setting is not necessary.

(5) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 18.89 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

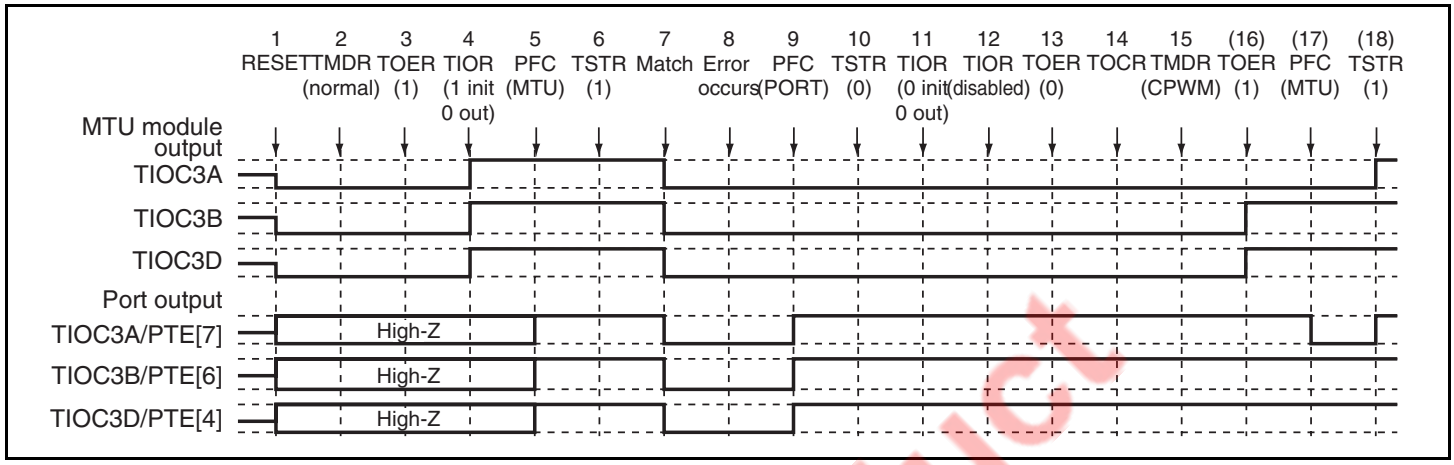


Figure 18.89 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 18.85.

11. Initialize the normal mode waveform generation section with TIOR.
12. Disable operation of the normal mode waveform generation section with TIOR.
13. Disable channel 3 and 4 output with TOER.
14. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
15. Set complementary PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set MTU output with the PFC.
18. Operation is restarted by TSTR.

(6) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Reset-Synchronous PWM Mode

Figure 18.90 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in reset-synchronous PWM mode after re-setting.

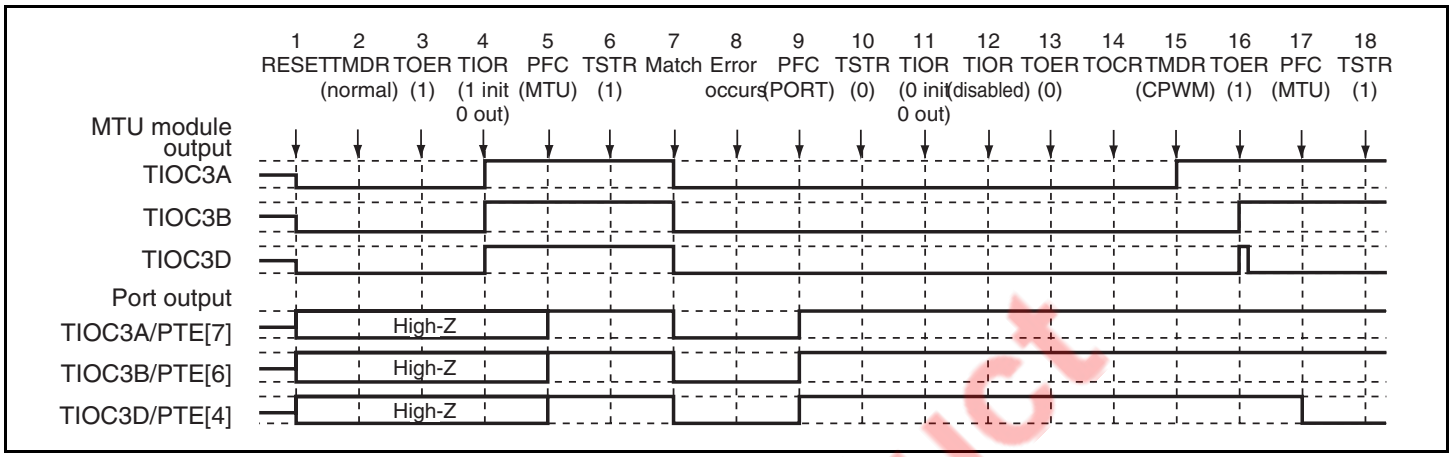


Figure 18.90 Error Occurrence in Normal Mode, Recovery in Reset-Synchronous PWM Mode

1 to 13 are the same as in figure 18.89.

14. Select the reset-synchronous PWM output level and cyclic output enabling/disabling with TOCR.
15. Set reset-synchronous PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set MTU output with the PFC.
18. Operation is restarted by TSTR.

(7) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Normal Mode

Figure 18.91 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

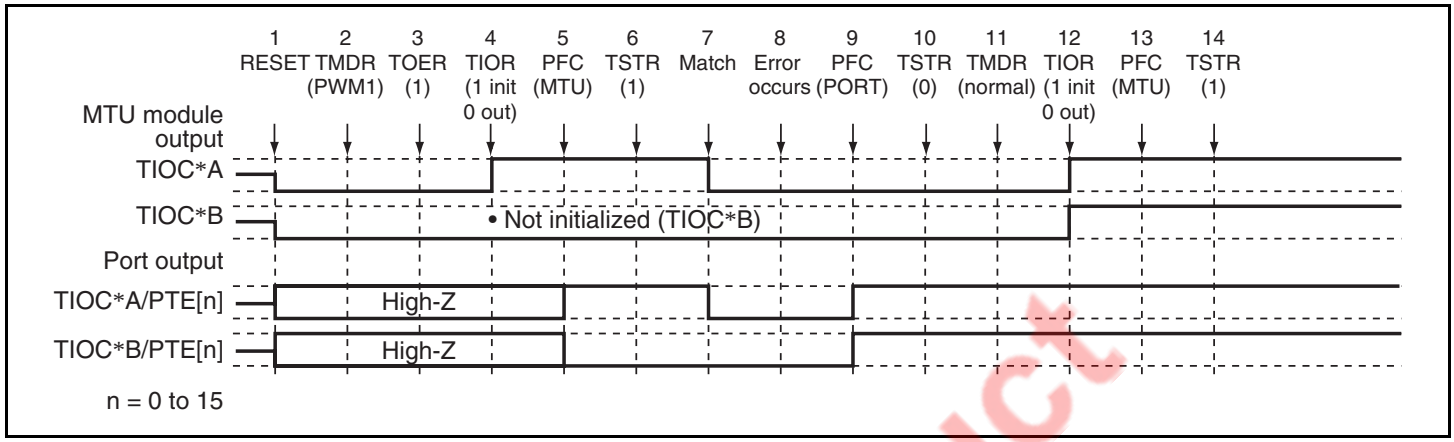


Figure 18.91 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Set PWM mode 1.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 1, the TIOC*B side is not initialized.)
5. Set MTU output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Set normal mode.
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(8) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 1

Figure 18.92 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

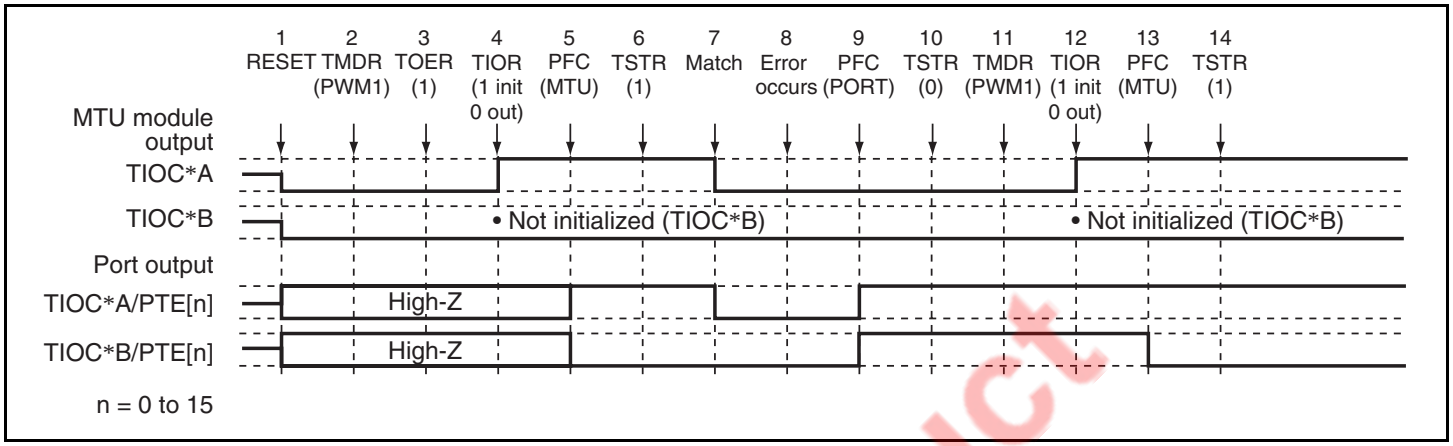


Figure 18.92 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

1 to 10 are the same as in figure 18.91.

11. Not necessary when restarting in PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(9) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 2

Figure 18.93 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

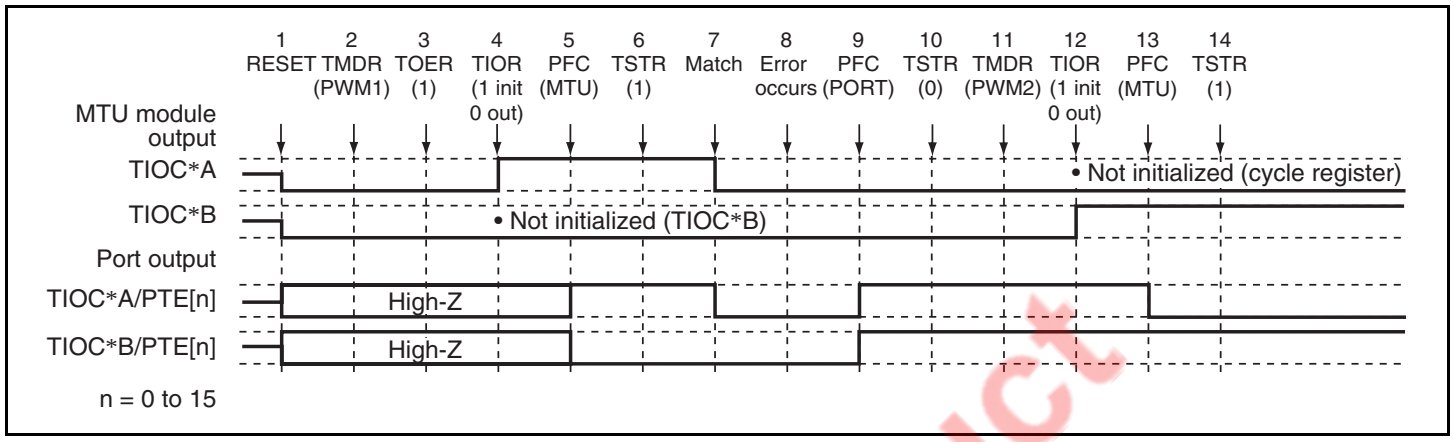


Figure 18.93 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

1 to 10 are the same as in figure 18.91.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0 to 2, and therefore TOER setting is not necessary.

(10) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Phase Counting Mode

Figure 18.94 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

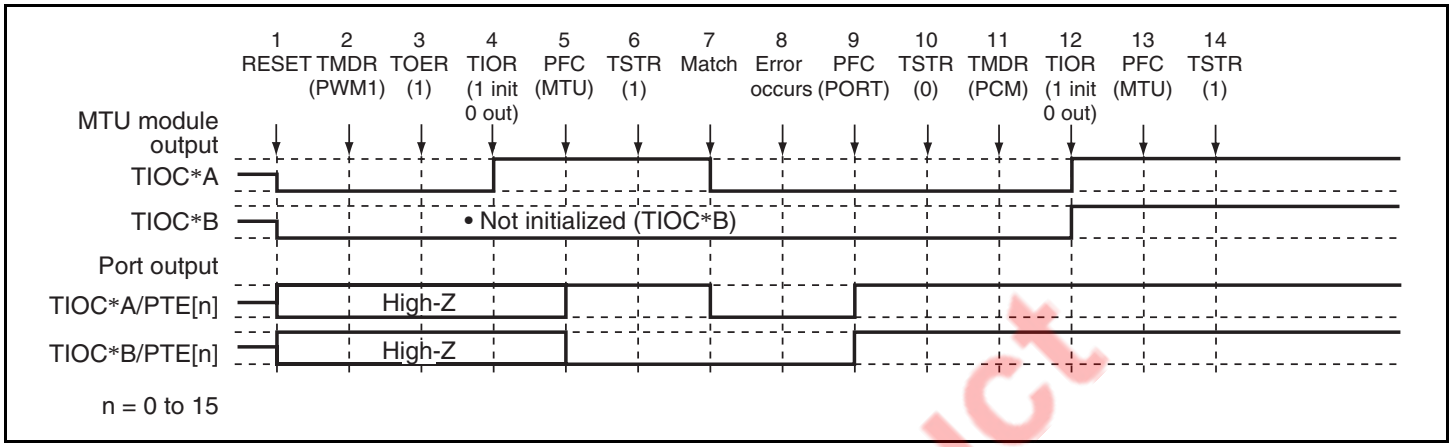


Figure 18.94 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 18.91.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER setting is not necessary.

(11) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Complementary PWM Mode

Figure 18.95 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

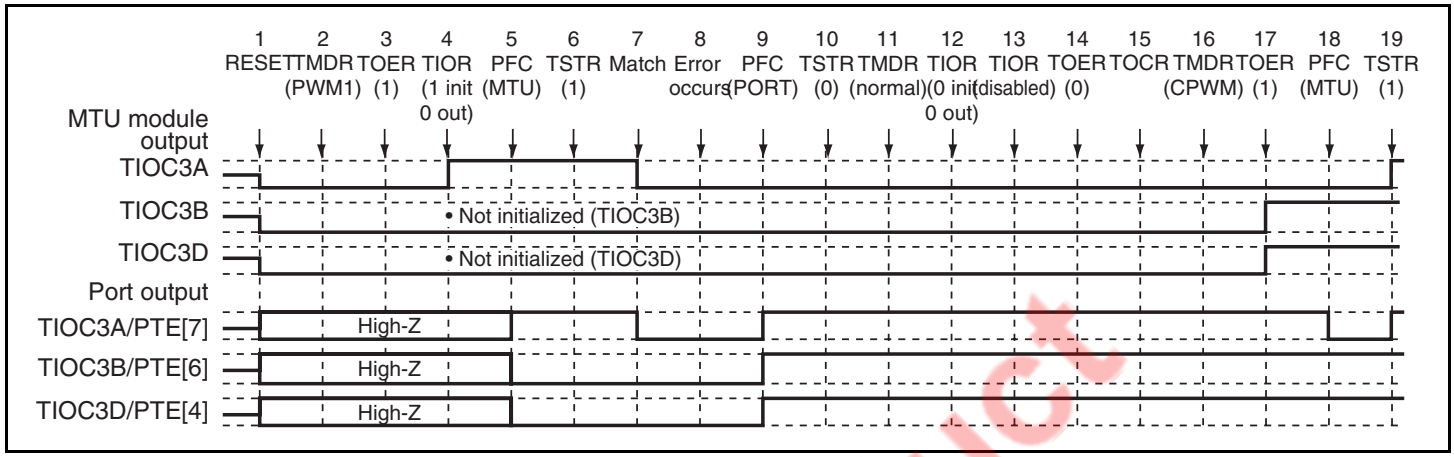


Figure 18.95 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 18.91.

11. Set normal mode for initialization of the normal mode waveform generation section.
12. Initialize the PWM mode 1 waveform generation section with TIOR.
13. Disable operation of the PWM mode 1 waveform generation section with TIOR.
14. Disable channel 3 and 4 output with TOER.
15. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
16. Set complementary PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set MTU output with the PFC.
19. Operation is restarted by TSTR.

(12) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Reset-Synchronous PWM Mode

Figure 18.96 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in reset-synchronous PWM mode after re-setting.

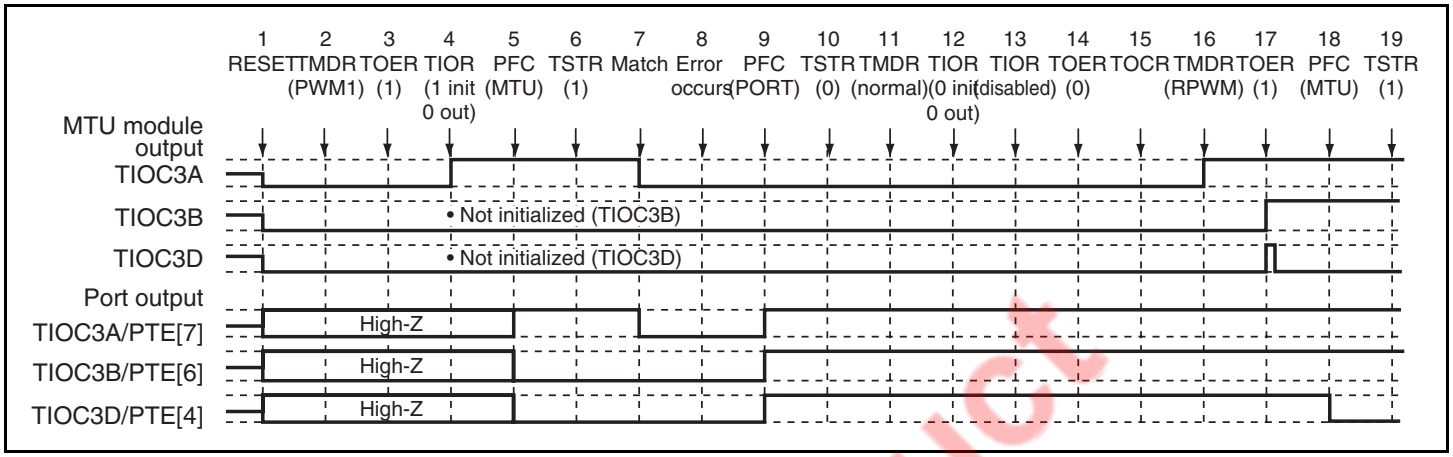


Figure 18.96 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronous PWM Mode

1 to 14 are the same as in figure 18.95.

15. Select the reset-synchronous PWM output level and cyclic output enabling/disabling with TOCR.
16. Set reset-synchronous PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set MTU output with the PFC.
19. Operation is restarted by TSTR.

(13) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Normal Mode

Figure 18.97 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

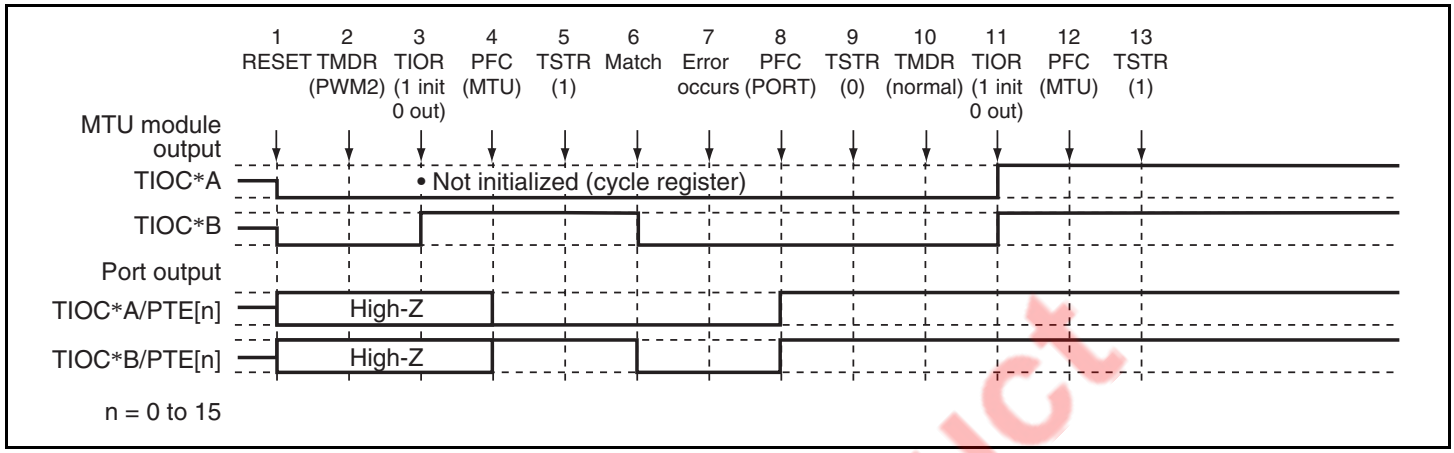


Figure 18.97 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Set PWM mode 2.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, TIOC *A is the cycle register.)
4. Set MTU output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set normal mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(14) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 1

Figure 18.98 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

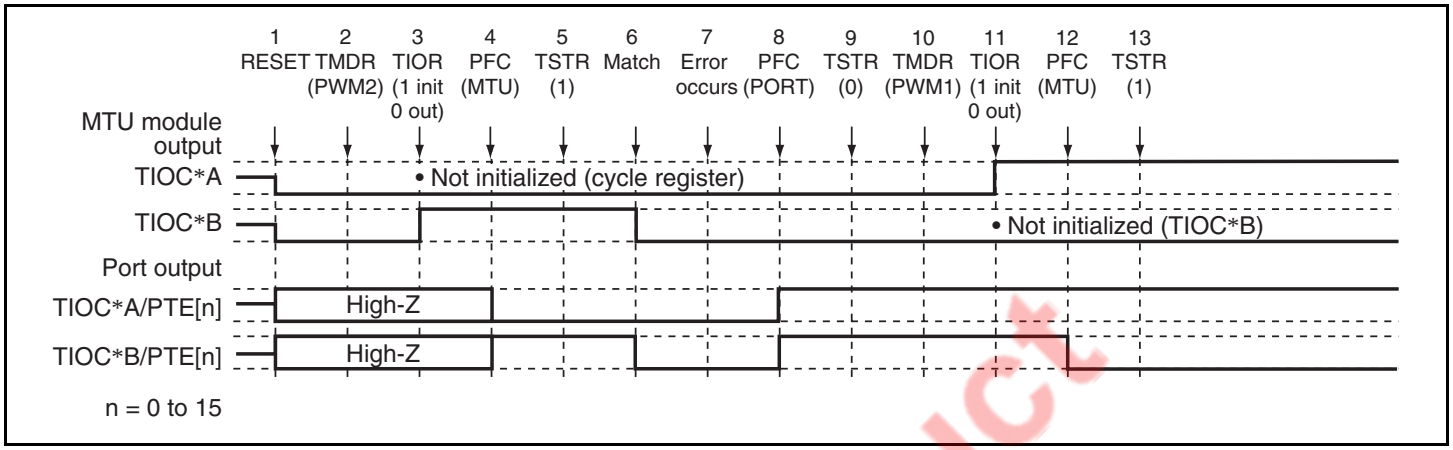


Figure 18.98 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

1 to 9 are the same as in figure 18.97.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(15) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 2

Figure 18.99 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

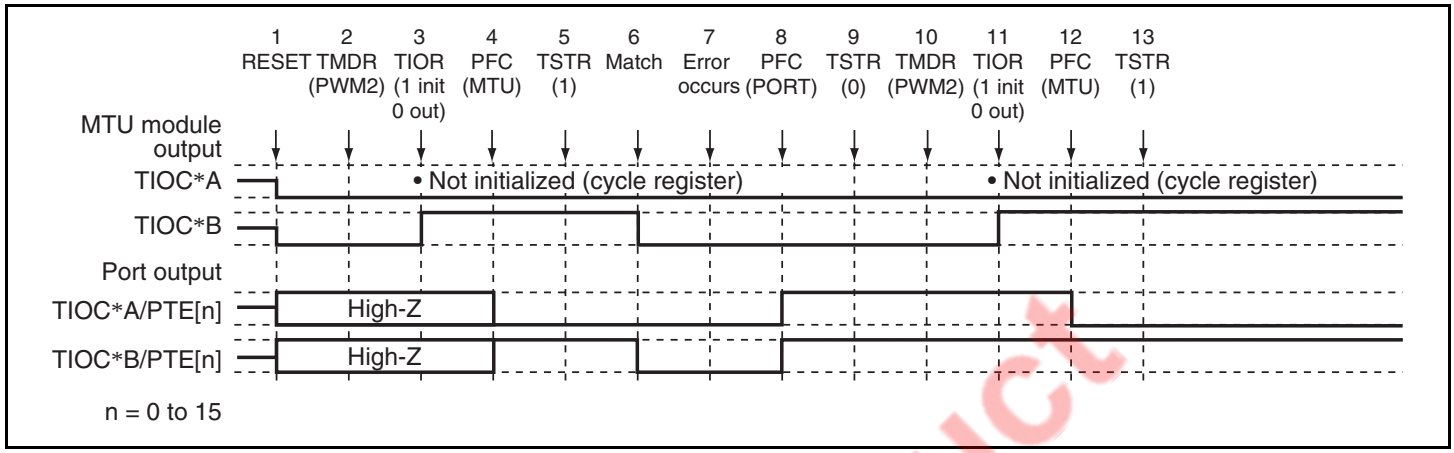


Figure 18.99 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

1 to 9 are the same as in figure 18.97.

10. Not necessary when restarting in PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(16) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Phase Counting Mode

Figure 18.100 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

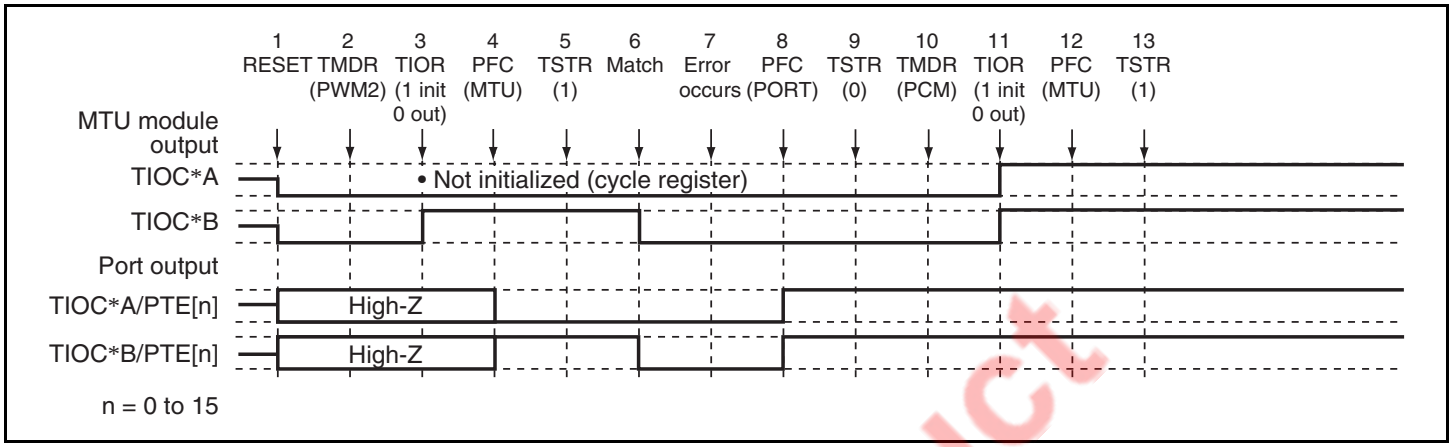


Figure 18.100 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 18.97.

10. Set phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(17) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Normal Mode

Figure 18.101 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

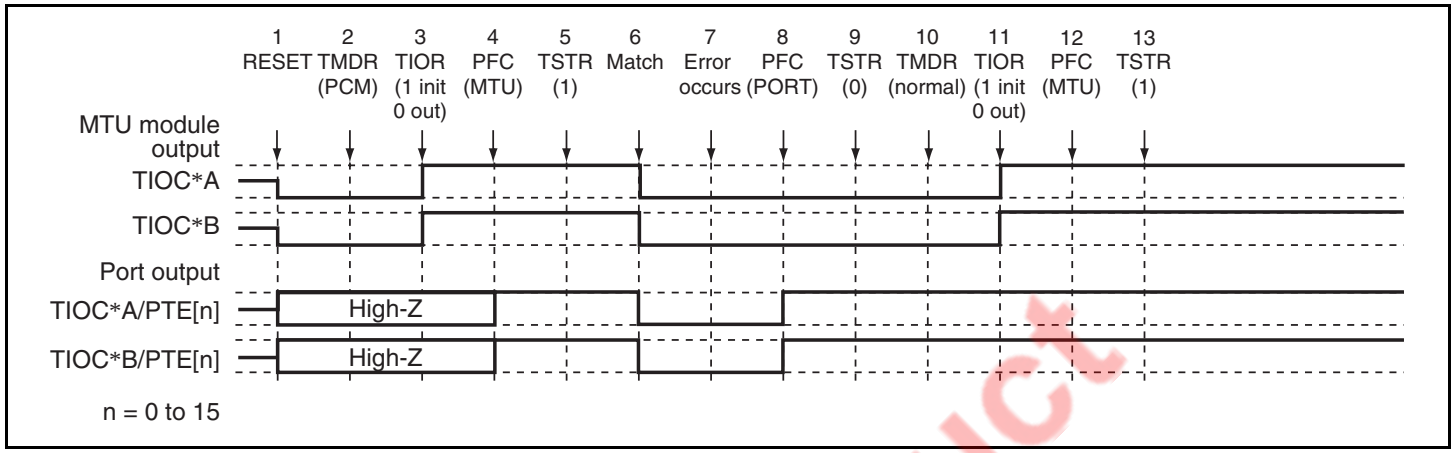


Figure 18.101 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Set phase counting mode.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
4. Set MTU output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set in normal mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(18) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 18.102 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

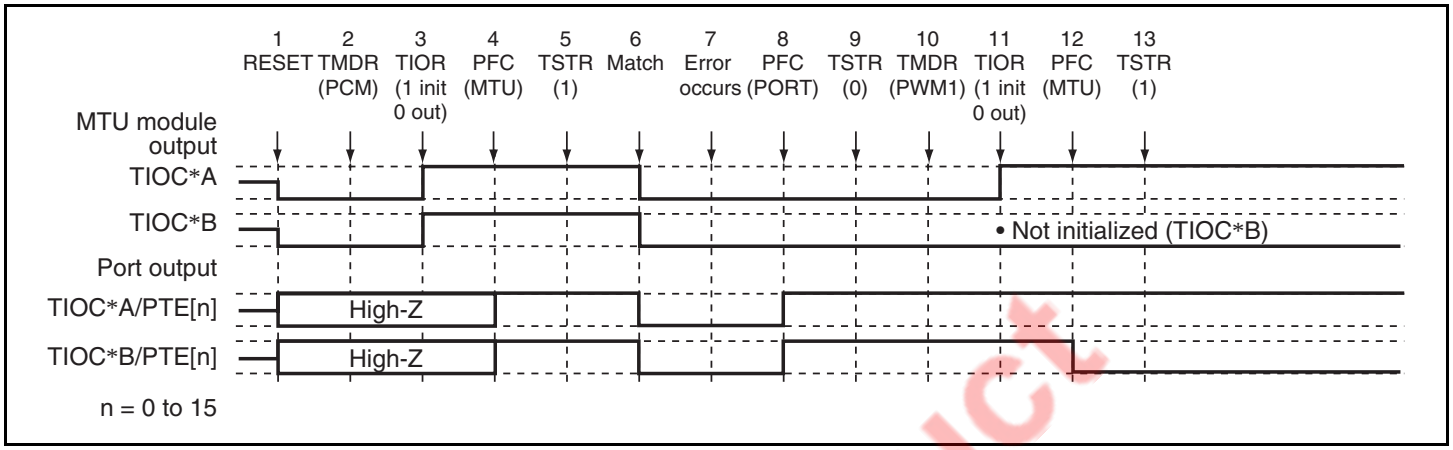


Figure 18.102 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

1 to 9 are the same as in figure 18.101.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(19) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 2

Figure 18.103 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

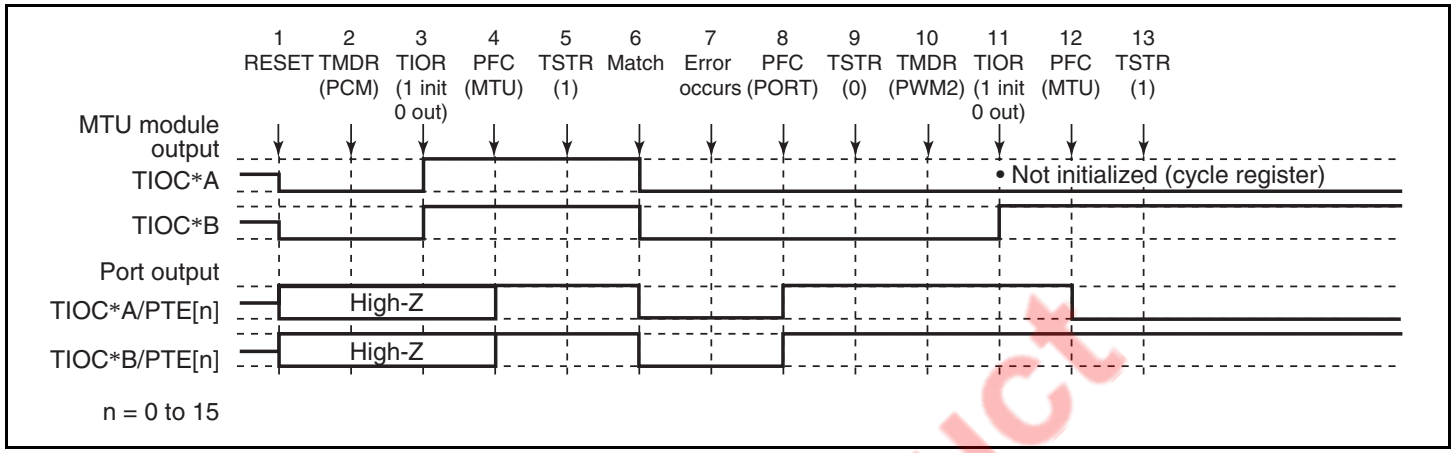


Figure 18.103 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

1 to 9 are the same as in figure 18.101.

10. Set PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(20) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Phase Counting Mode

Figure 18.104 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

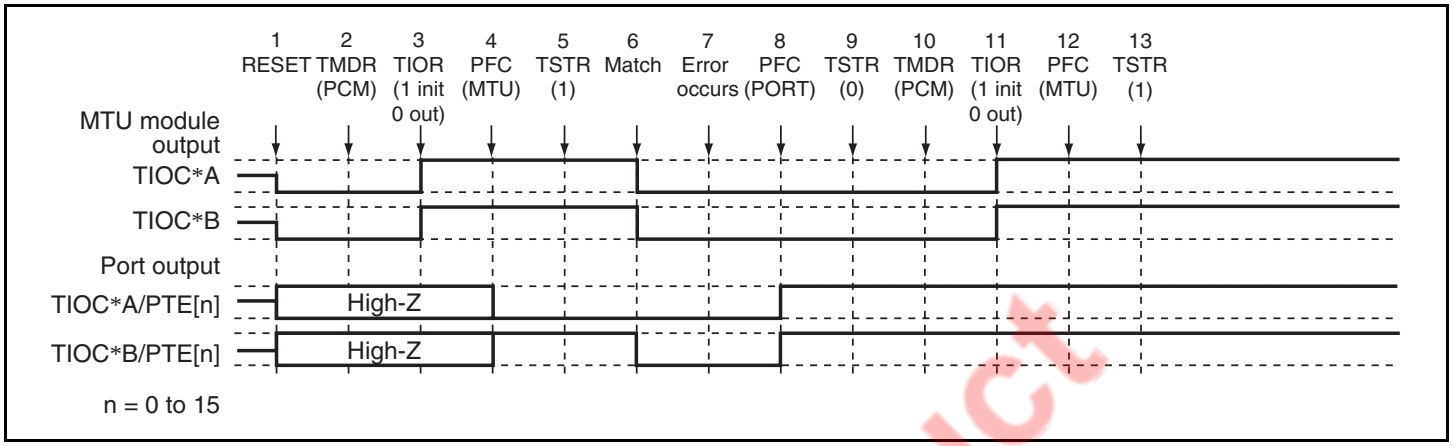


Figure 18.104 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 18.101.

10. Not necessary when restarting in phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(21) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Normal Mode

Figure 18.105 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

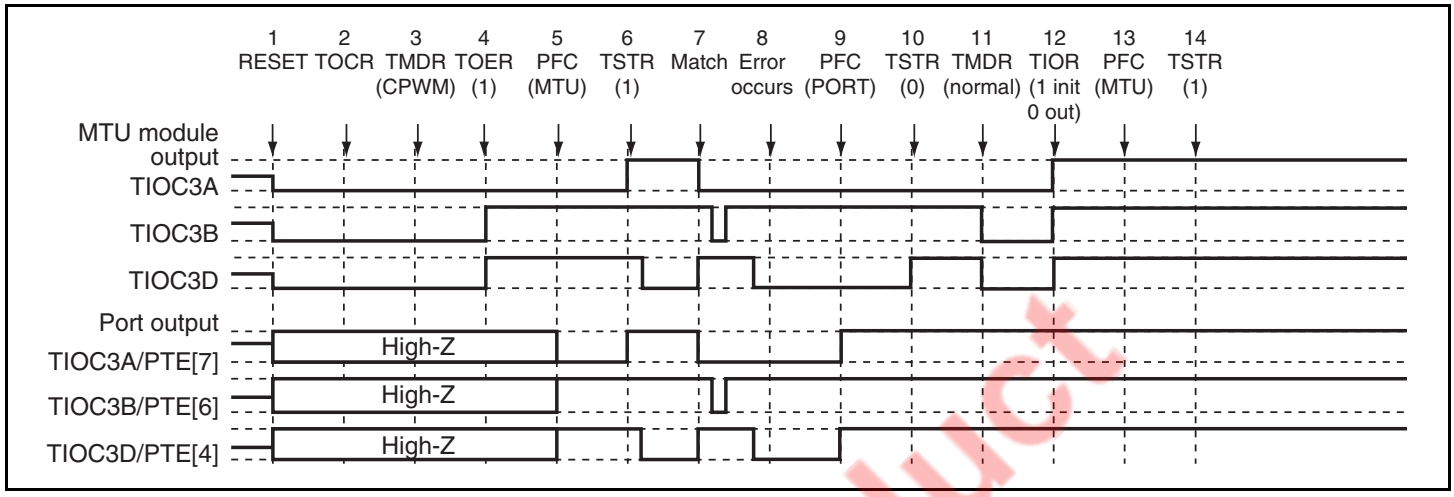


Figure 18.105 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
3. Set complementary PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU output with the PFC.
6. The count operation is started by TSTR.
7. The complementary PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU output becomes the complementary PWM output initial value.)
11. Set normal mode. (MTU output goes low.)
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(22) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 18.106 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

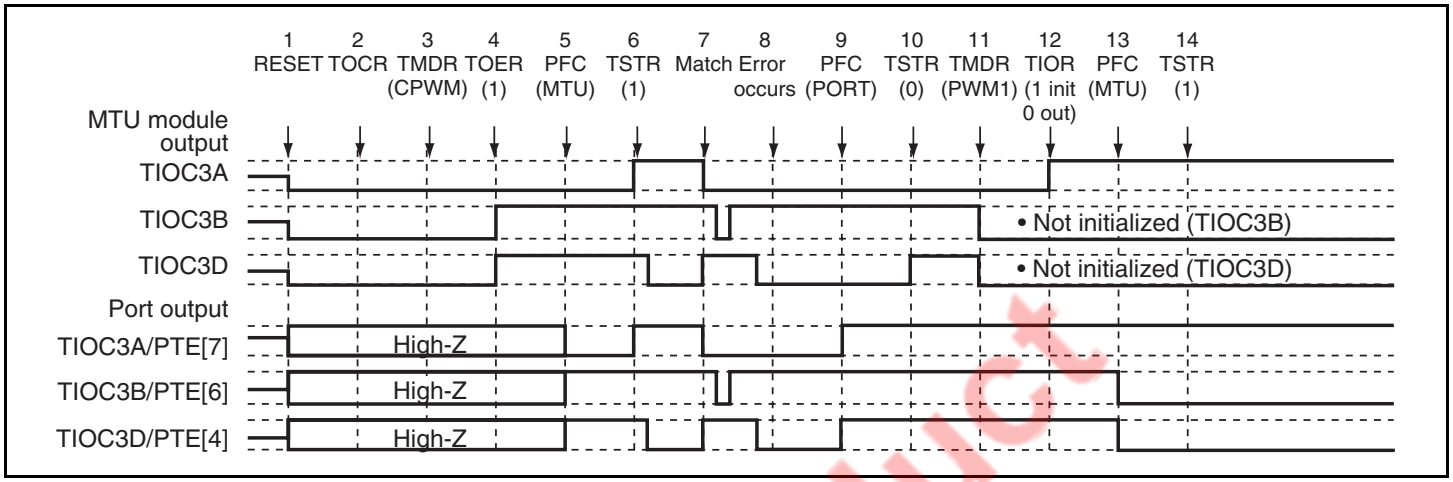


Figure 18.106 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 18.105.

11. Set PWM mode 1. (MTU output goes low.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(23) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 18.107 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the cycle and duty settings at the time the counter was stopped).

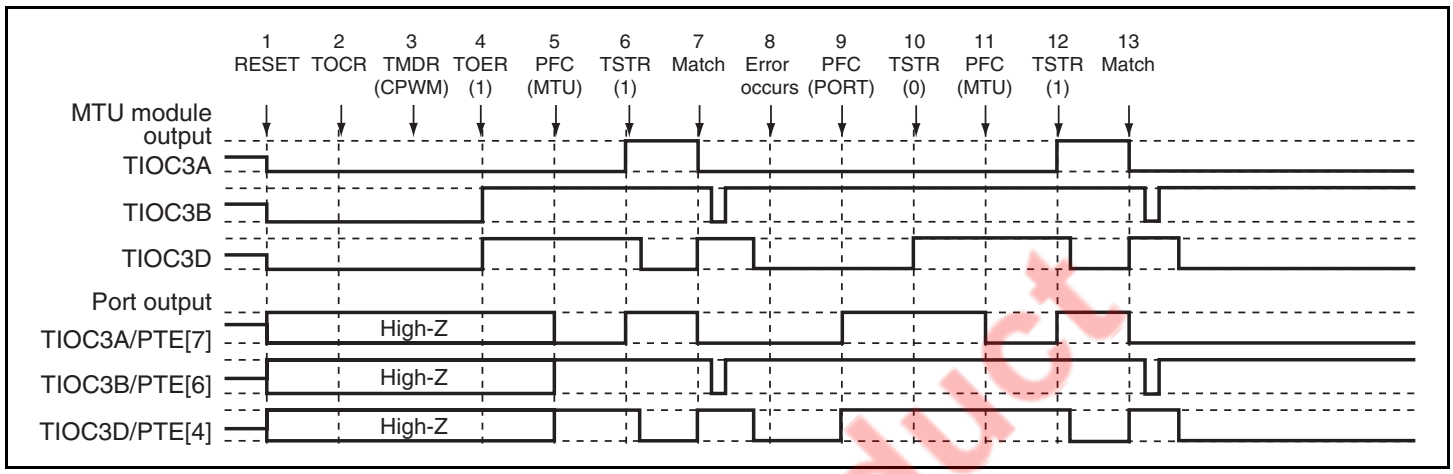


Figure 18.107 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 18.105.

11. Set MTU output with the PFC.
12. Operation is restarted by TSTR.
13. The complementary PWM waveform is output on compare-match occurrence.

(24) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 18.108 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using completely new cycle and duty settings).

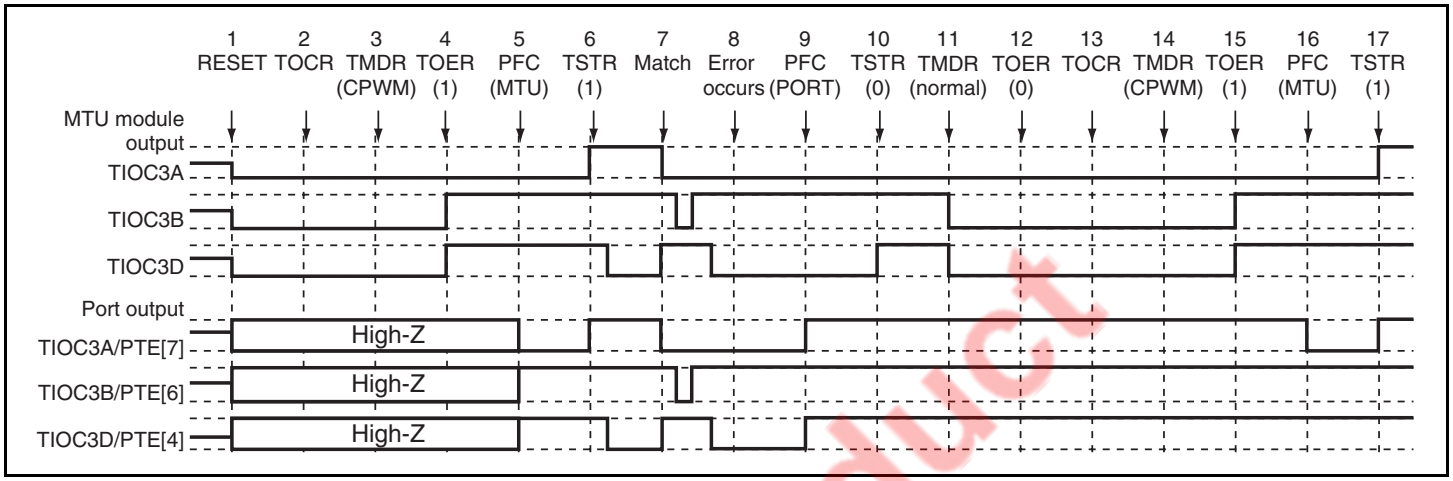


Figure 18.108 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 18.105.

11. Set normal mode and make new settings. (MTU output goes low.)
12. Disable channel 3 and 4 output with TOER.
13. Select the complementary PWM mode output level and cyclic output enabling/disabling with TOCR.
14. Set complementary PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set MTU output with the PFC.
17. Operation is restarted by TSTR.

(25) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Reset-Synchronous PWM Mode

Figure 18.109 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in reset-synchronous PWM mode.

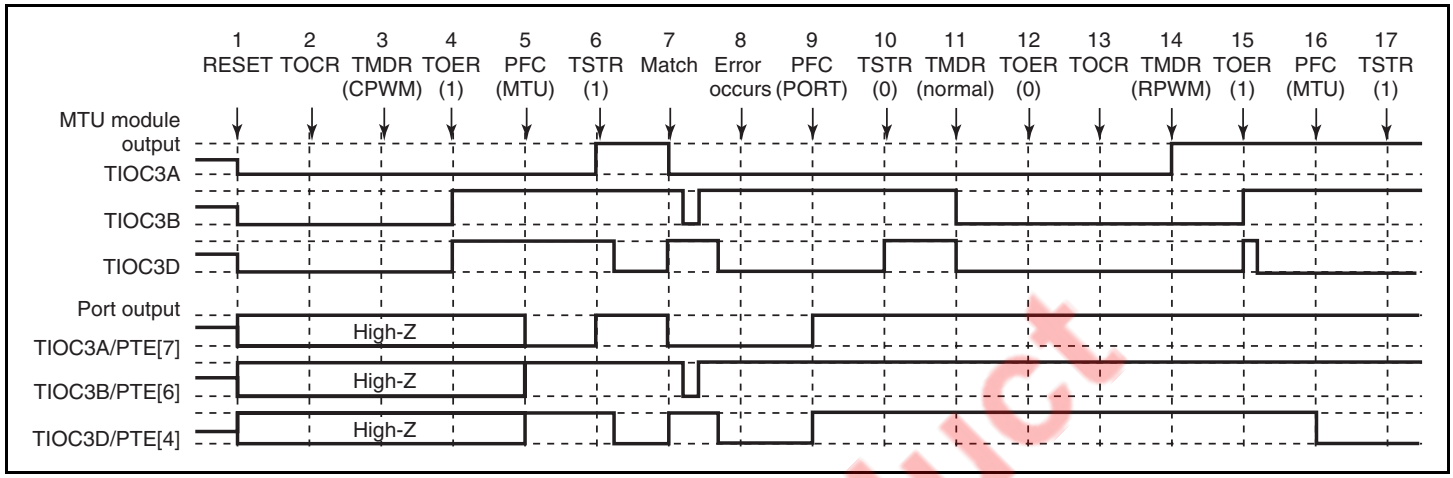


Figure 18.109 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronous PWM Mode

1 to 10 are the same as in figure 18.105.

11. Set normal mode. (MTU output goes low.)
12. Disable channel 3 and 4 output with TOER.
13. Select the reset-synchronous PWM mode output level and cyclic output enabling/disabling with TOCR.
14. Set reset-synchronous PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set MTU output with the PFC.
17. Operation is restarted by TSTR.

(26) Operation when Error Occurs during Reset-Synchronous PWM Mode Operation, and Operation is Restarted in Normal Mode

Figure 18.110 shows an explanatory diagram of the case where an error occurs in reset-synchronous PWM mode and operation is restarted in normal mode after re-setting.

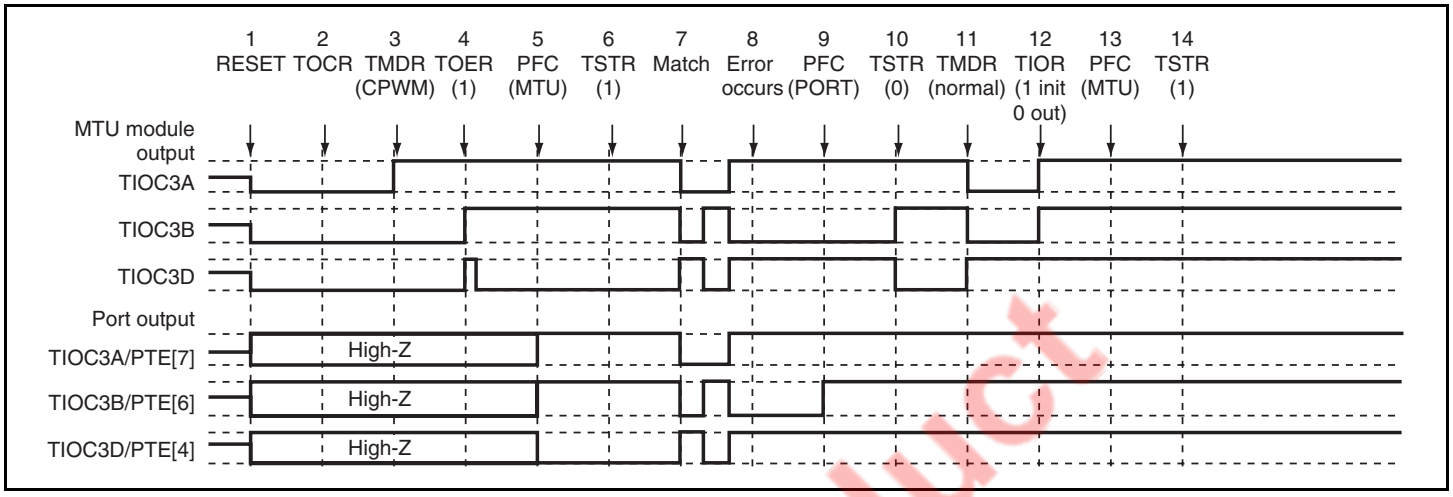


Figure 18.110 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Select the reset-synchronous PWM output level and cyclic output enabling/disabling with TOCR.
3. Set reset-synchronous PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU output with the PFC.
6. The count operation is started by TSTR.
7. The reset-synchronous PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU output becomes the reset-synchronous PWM output initial value.)
11. Set normal mode. (MTU positive phase output is low, and negative phase output is high.)
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(27) Operation when Error Occurs during Reset-Synchronous PWM Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 18.111 shows an explanatory diagram of the case where an error occurs in reset-synchronous PWM mode and operation is restarted in PWM mode 1 after re-setting.

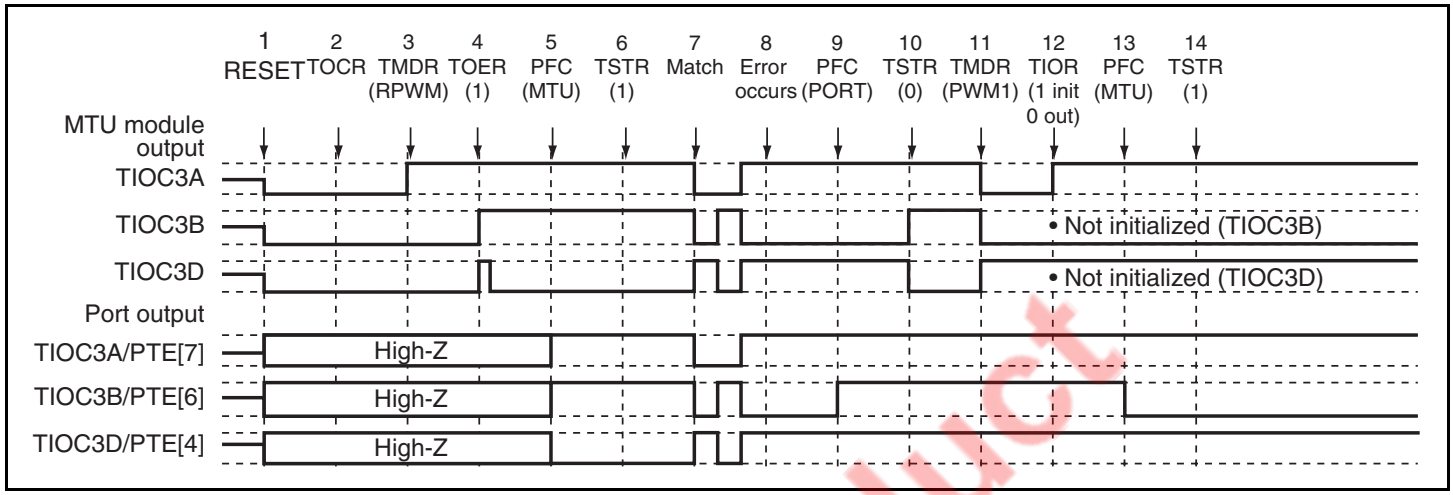


Figure 18.111 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 18.110.

11. Set PWM mode 1. (MTU positive phase output is low, and negative phase output is high.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(28) Operation when Error Occurs during Reset-Synchronous PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 18.112 shows an explanatory diagram of the case where an error occurs in reset-synchronous PWM mode and operation is restarted in complementary PWM mode after re-setting.

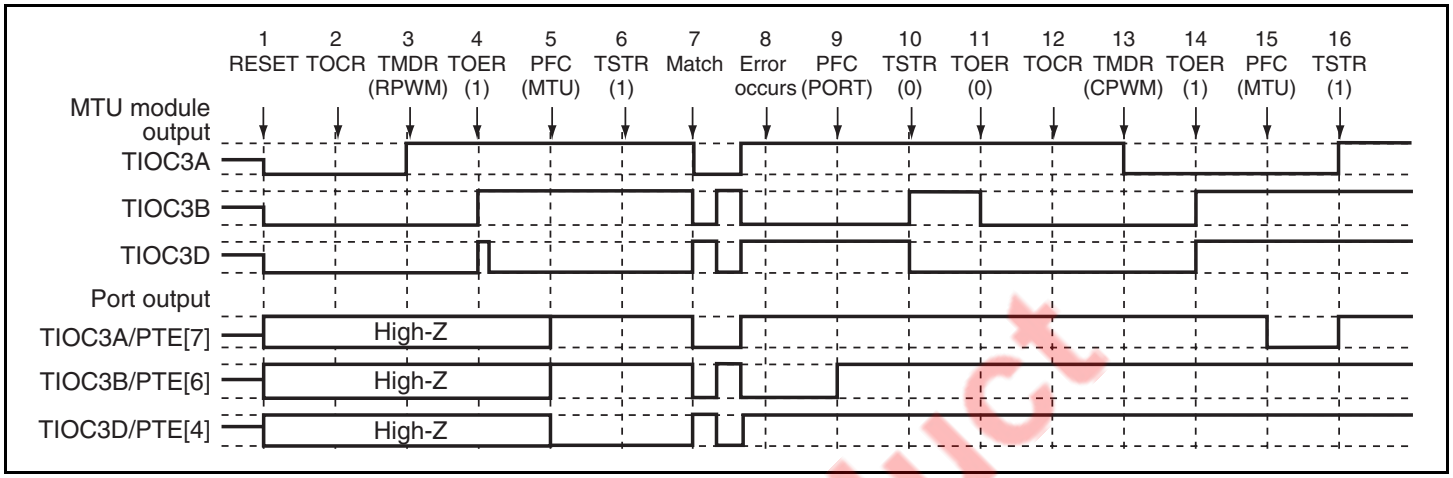


Figure 18.112 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 18.110.

11. Disable channel 3 and 4 output with TOER.
12. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
13. Set complementary PWM. (The MTU cyclic output pin goes low.)
14. Enable channel 3 and 4 output with TOER.
15. Set MTU output with the PFC.
16. Operation is restarted by TSTR.

(29) Operation when Error Occurs during Reset-Synchronous PWM Mode Operation, and Operation is Restarted in Reset-Synchronous PWM Mode

Figure 18.113 shows an explanatory diagram of the case where an error occurs in reset-synchronous PWM mode and operation is restarted in reset-synchronous PWM mode after re-setting.

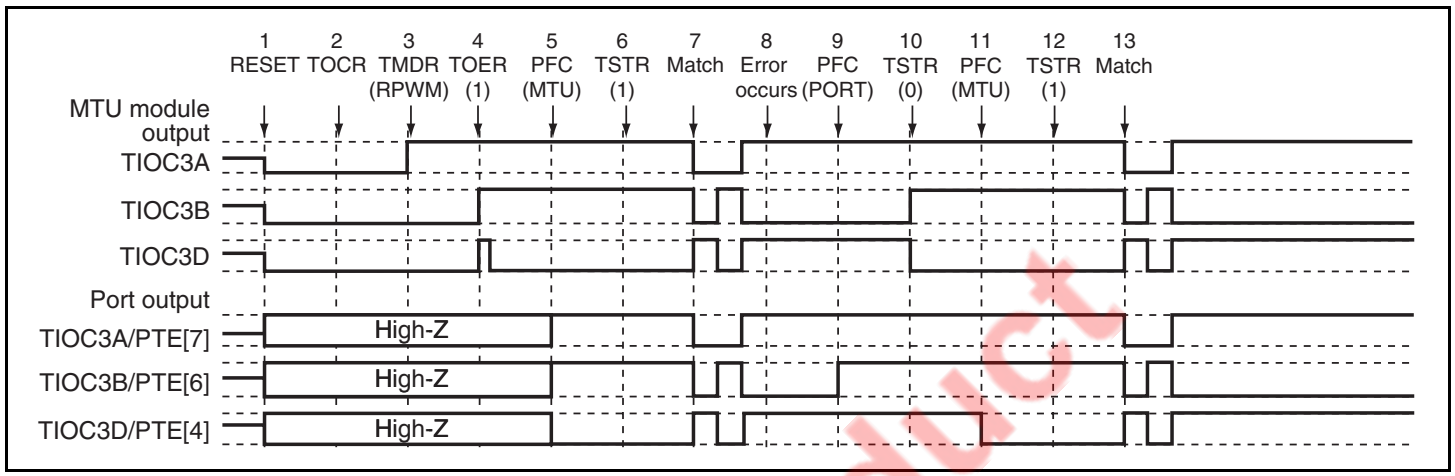


Figure 18.113 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Reset-Synchronous PWM Mode

1 to 10 are the same as in figure 18.110.

11. Set MTU output with the PFC.
12. Operation is restarted by TSTR.
13. The reset-synchronous PWM waveform is output on compare-match occurrence.

18.9 Port Output Enable (POE)

The port output enable (POE) can be used to establish a high-impedance state for high-current pins, by changing the $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$ pin input, depending on the output status of the high-current pins (TIOC3B/PTE[6], TIOC3D/PTE[4], TIOC4A/PTE[3], TIOC4B/PTE[2], TIOC4C/PTE[1], TIOC4D/PTE[0]). It can also simultaneously generate interrupt requests.

18.9.1 Features

- Each of the $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$ input pins can be set for falling edge, Pϕ/8 × 16, Pϕ/16 × 16, or Pϕ/128 × 16 low-level sampling.
- High-current pins can be set to high-impedance state by $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$ pin falling-edge or low-level sampling.
- High-current pins can be set to high-impedance state when the high-current pin output levels are compared and simultaneous low-level output continues for one cycle or more.
- Interrupts can be generated by input-level sampling or output-level comparison results.

The POE has input-level detection circuitry and output-level detection circuitry, as shown in the block diagram of figure 18.114.

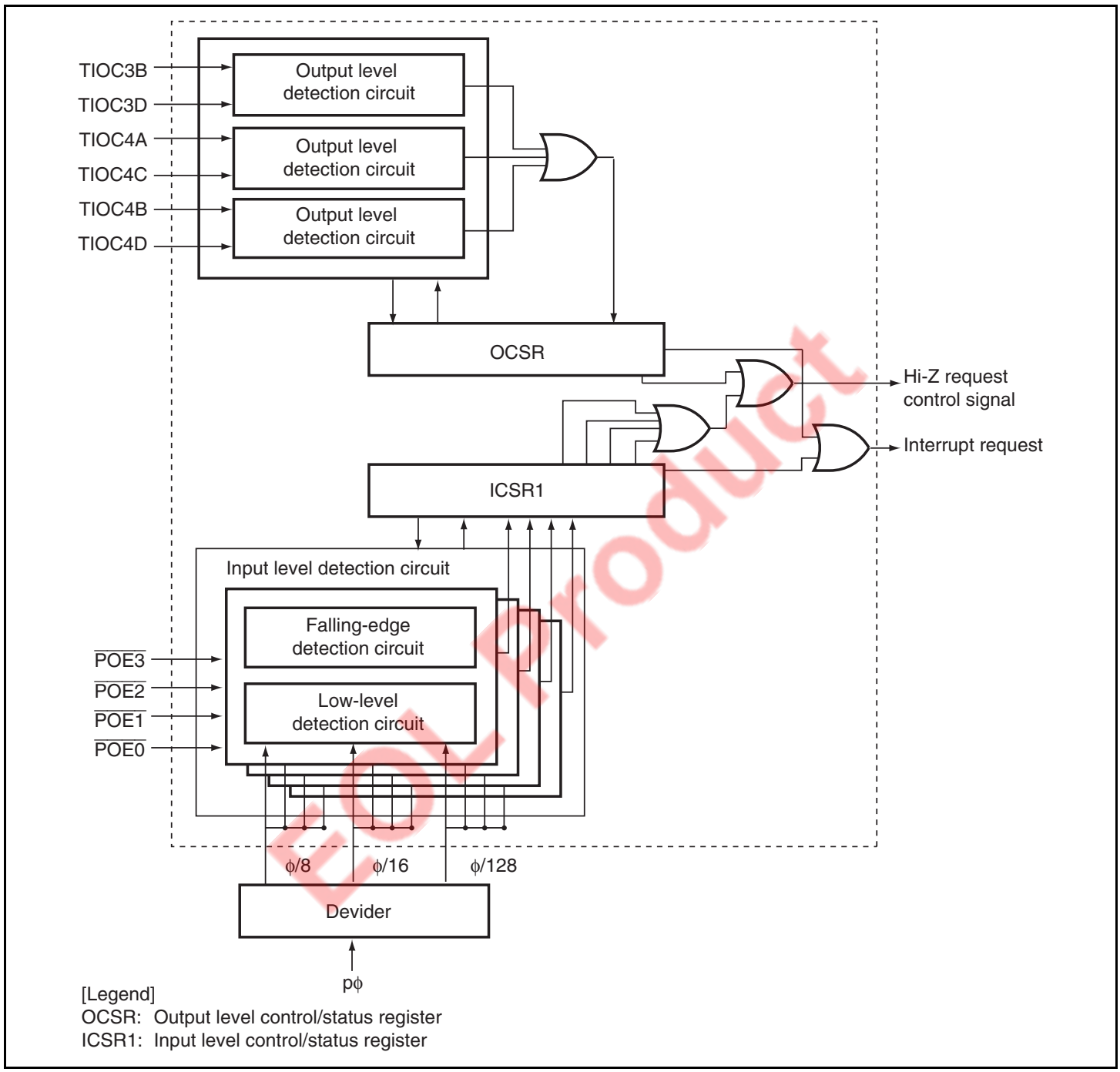


Figure 18.114 POE Block Diagram

18.9.2 Pin Configuration

Table 18.44 Pin Configuration

Name	Abbreviation	I/O	Description
Port output enable input pins	$\overline{\text{POE0}}$ to $\overline{\text{POE3}}$	Input	Input request signals to make high-current pins high-impedance state

Table 18.45 shows output-level comparisons with pin combinations.

Table 18.45 Pin Combinations

Pin Combination	I/O	Description
TIOC3B/PTE[6] and TIOC3D/PTE[4]	Output	All high-current pins are made high-impedance state when the pins simultaneously output low-level for longer than 1 cycle.
TIOC4A/PTE[3] and TIOC4C/PTE[1]	Output	All high-current pins are made high-impedance state when the pins simultaneously output low-level for longer than 1 cycle.
TIOC4B/PTE[2] and TIOC4D/PTE[0]	Output	All high-current pins are made high-impedance state when the pins simultaneously output low-level for longer than 1 cycle.

18.9.3 Register Configuration

The POE has the two registers. The input level control/status register 1 (ICSR1) controls both $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$ pin input signal detection and interrupts. The output level control/status register (OCSR) controls both the enable/disable of output comparison and interrupts.

Input Level Control/Status Register 1 (ICSR1): ICSR1 is a 16-bit readable/writable register that selects the $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$ pin input modes, controls the enable/disable of interrupts, and indicates status.

Bit	Bit Name	Initial value	R/W	Description
15	POE3F	0	R/(W)*	POE3 Flag This flag indicates that a high impedance request has been input to the $\overline{\text{POE3}}$ pin [Clear condition] • By writing 0 to POE3F after reading a POE3F = 1 [Set condition] • When the input set by ICSR1 bits 7 and 6 occurs at the POE3 pin
14	POE2F	0	R/(W)*	POE2 Flag This flag indicates that a high impedance request has been input to the $\overline{\text{POE2}}$ pin [Clear condition] • By writing 0 to POE2F after reading a POE2F = 1 [Set condition] • When the input set by ICSR1 bits 5 and 4 occurs at the POE2 pin
13	POE1F	0	R/(W)*	POE1 Flag This flag indicates that a high impedance request has been input to the $\overline{\text{POE1}}$ pin [Clear condition] • By writing 0 to POE1F after reading a POE1F = 1 [Set condition] • When the input set by ICSR1 bits 3 and 2 occurs at the $\overline{\text{POE1}}$ pin
12	POE0F	0	R/(W)*	POE0 Flag This flag indicates that a high impedance request has been input to the $\overline{\text{POE0}}$ pin [Clear condition] • By writing 0 to POE0F after reading a POE0F = 1 [Set condition] • When the input set by ICSR1 bits 1 and 0 occurs at the $\overline{\text{POE0}}$ pin

Bit	Bit Name	Initial value	R/W	Description
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PIE	0	R/W	Port Interrupt Enable This bit enables/disables interrupt requests when any of the POE0F to POE3F bits of the ICSR1 are set to 1 0: Interrupt requests disabled 1: Interrupt requests enabled
7	POE3M1	0	R/W	POE3 mode 1, 0
6	POE3M0	0	R/W	These bits select the input mode of the $\overline{\text{POE3}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE3}}$ input 01: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/8 clock pulses, and all are low level. 10: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/16 clock pulses, and all are low level. 11: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/128 clock pulses, and all are low level.
5	POE2M1	0	R/W	POE2 mode 1, 0
4	POE2M0	0	R/W	These bits select the input mode of the $\overline{\text{POE2}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE2}}$ input 01: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 Pϕ/8 clock pulses, and all are low level. 10: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 Pϕ/16 clock pulses, and all are low level. 11: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 Pϕ/128 clock pulses, and all are low level.

Bit	Bit Name	Initial value	R/W	Description
3	POE1M1	0	R/W	POE1 mode 1, 0
2	POE1M0	0	R/W	These bits select the input mode of the $\overline{\text{POE1}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE1}}$ input 01: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/8 clock pulses, and all are low level. 10: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/16 clock pulses, and all are low level. 11: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/128 clock pulses, and all are low level.
1	POE0M1	0	R/W	POE0 mode 1, 0
0	POE0M0	0	R/W	These bits select the input mode of the $\overline{\text{POE0}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE0}}$ input 01: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/8 clock pulses, and all are low level. 10: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/16 clock pulses, and all are low level. 11: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/128 clock pulses, and all are low level.

Note: * The write value should always be 0.

Output Level Control/Status Register (OCSR): OCSR is a 16-bit readable/writable register that controls the enable/disable of both output level comparison and interrupts, and indicates status. If the OSF bit is set to 1, the high current pins become high impedance.

Bit	Bit Name	Initial value	R/W	Description
15	OSF	0	R/(W)*	Output Short Flag This flag indicates that any one pair of the three pairs of 2 phase outputs compared have simultaneously become low level outputs. [Clear condition] • By writing 0 to OSF after reading an OSF = 1 [Set condition] • When any one pair of the three 2-phase outputs simultaneously become low level
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial value	R/W	Description
9	OCE	0	R/W	Output Level Compare Enable This bit enables the start of output level comparisons. When setting this bit to 1, pay attention to the output pin combinations shown in table 18.43, Mode Transition Combinations. When 0 is output on both pins, the OSF bit is set to 1 at the same time when this bit is set, and output goes to high impedance. Accordingly, bit 6 and bits 4 to 0 in the port E data register (PEDR) are set to 1. For the MTU output comparison, set the bit to 1 after setting the MTU's output pins with the PFC. Set this bit only when using pins as outputs. When the OCE bit is set to 1, if OIE = 0 a high-impedance request will not be issued even if OSF is set to 1. Therefore, in order to have a high-impedance request issued according to the result of the output level comparison, the OIE bit must be set to 1. When OCE = 1 and OIE = 1, an interrupt request will be generated at the same time as the high-impedance request: however, this interrupt can be masked by means of an interrupt controller (INTC) setting. 0: Output level compare disabled 1: Output level compare enabled; makes an output high impedance request when OSF = 1.
8	OIE	0	R/W	Output Short Interrupt Enable This bit makes interrupt requests when the OSF bit of the OCSR is set. 0: Interrupt requests disabled 1: Interrupt request enabled
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

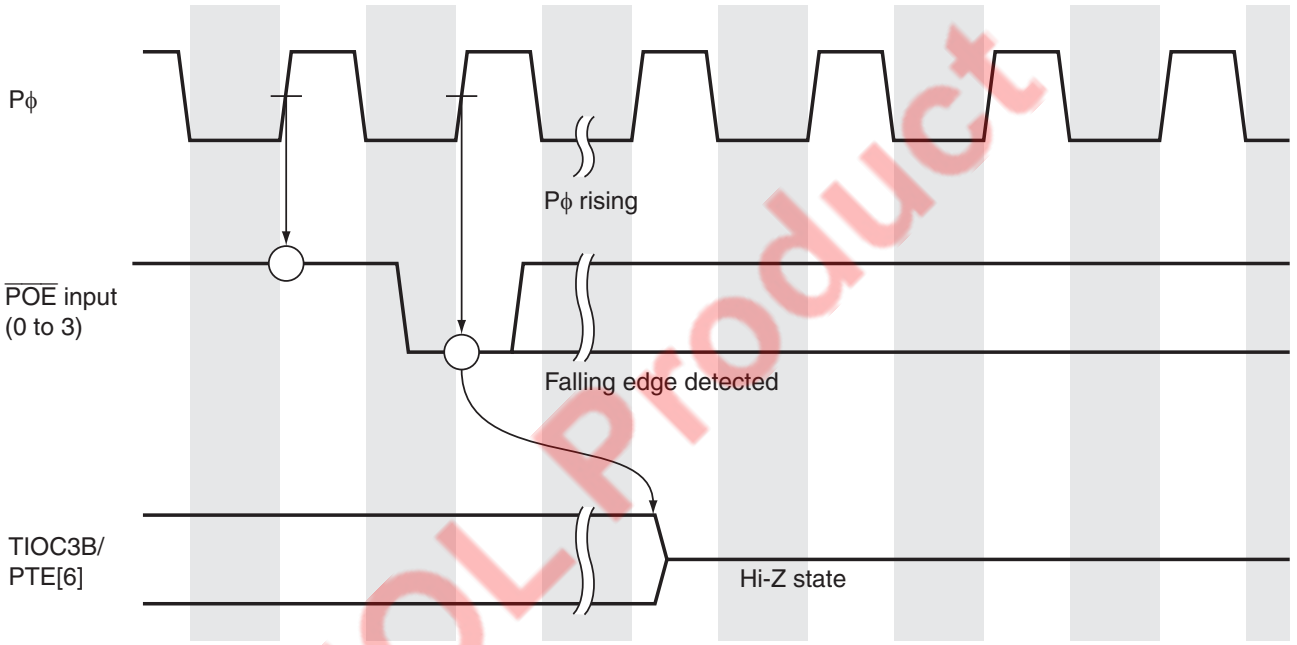
Note: * The write value should always be 0.

18.9.4 Operation

Input Level Detection Operation: If the input conditions set by the ICSR1 occur on any of the $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$ pins, all high-current pins become high-impedance state. However, only when the general input/output function or MTU function is selected, the large-current pin is in the high-impedance state.

1. Falling Edge Detection:

   When a change from high to low level is input to the $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$ pins, all high-current pins become high-impedance state. Figure 18.115 shows the timing example for the $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$ pins which enters the high-impedance state through input of a change from high to low level.



Note: Other high-current pins (TIOC3D/PTE[4], TIOC4A/PTE[3], TIOC4B/PTE[2], TIOC4C/PTE[1], TIOC4D/PTE[0]) also become the Hi-Z state at the same timing.

Figure 18.115 Falling Edge Detection Operation

2. Low-Level Detection

Figure 18.116 shows the low-level detection operation. Sixteen continuous low levels are sampled with the sampling clock established by the ICSR1. If even one high level is detected during this interval, the low level is not accepted.

Furthermore, the timing when the large-current pins enter the high-impedance state from the sampling clock is the same in both falling-edge detection and in low-level detection.

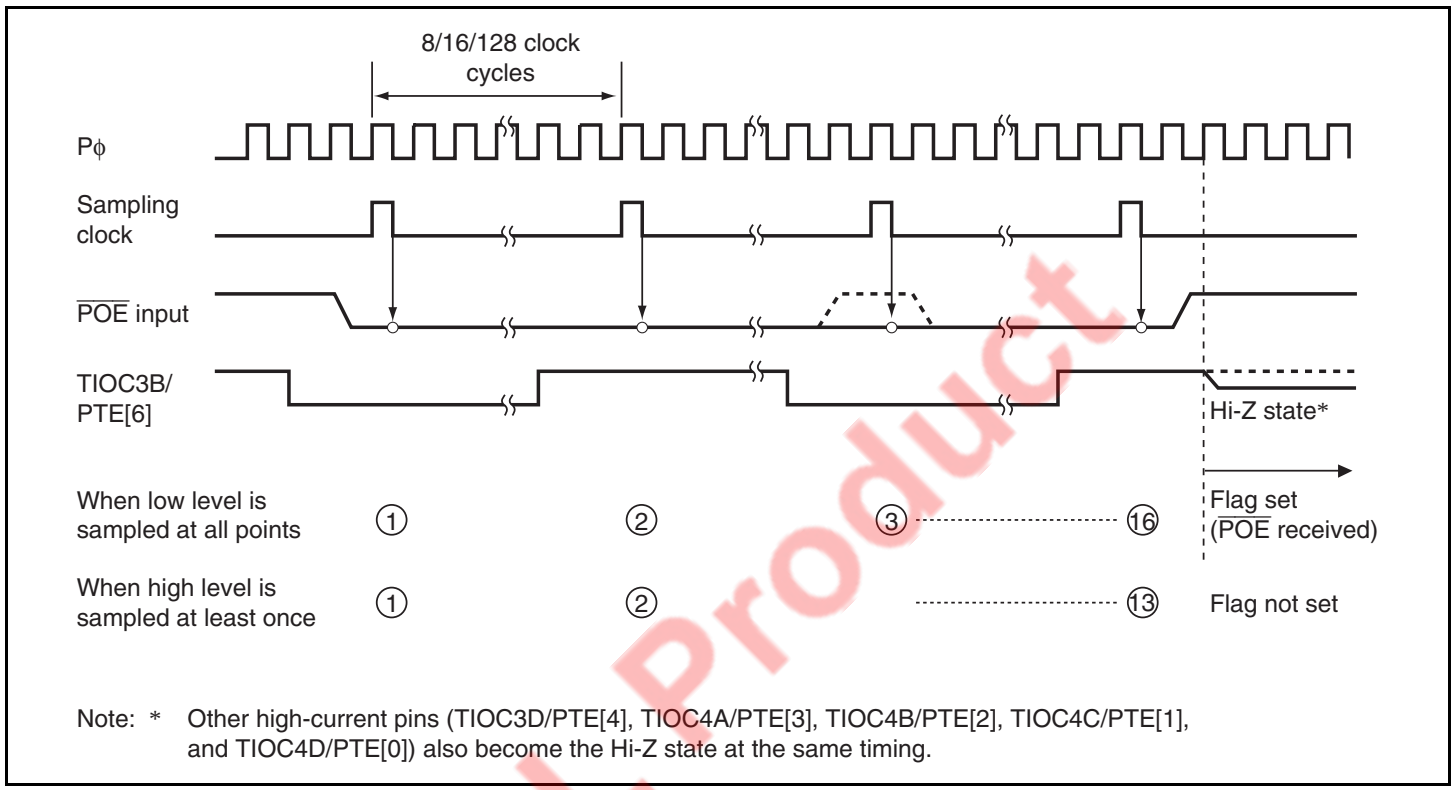


Figure 18.116 Low-Level Detection Operation

Output-Level Compare Operation: Figure 18.117 shows an example of the output-level compare operation for the combination of TIOC3B/PTE[6] and TIOC3D/PTE[4]. The operation is the same for the other pin combinations.

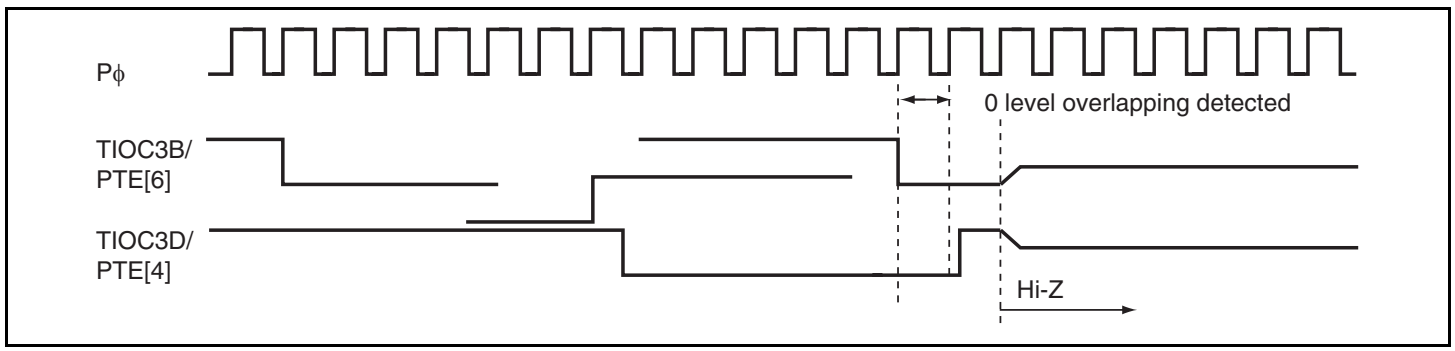


Figure 18.117 Output-Level Detection Operation

Release from High-Impedance State: High-current pins that have entered high-impedance state due to input-level detection can be released either by returning them to their initial state with a power-on reset, or by clearing all of the bit 12 to 15 (POE0F to POE3F) flags of the ICSR1. High-current pins that have become high-impedance due to output-level detection can be released either by returning them to their initial state with a power-on reset, or by first clearing bit 9 (OCE) of the OCSR to disable output-level compares, then clearing the bit 15 (OSF) flag. However, when returning from high-impedance state by clearing the OSF flag, always do so only after outputting a high level from the high-current pins (TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D). High-level outputs can be achieved by setting the MTU internal registers.

Section 19 Serial Communication Interface with FIFO (SCIF)

19.1 Overview

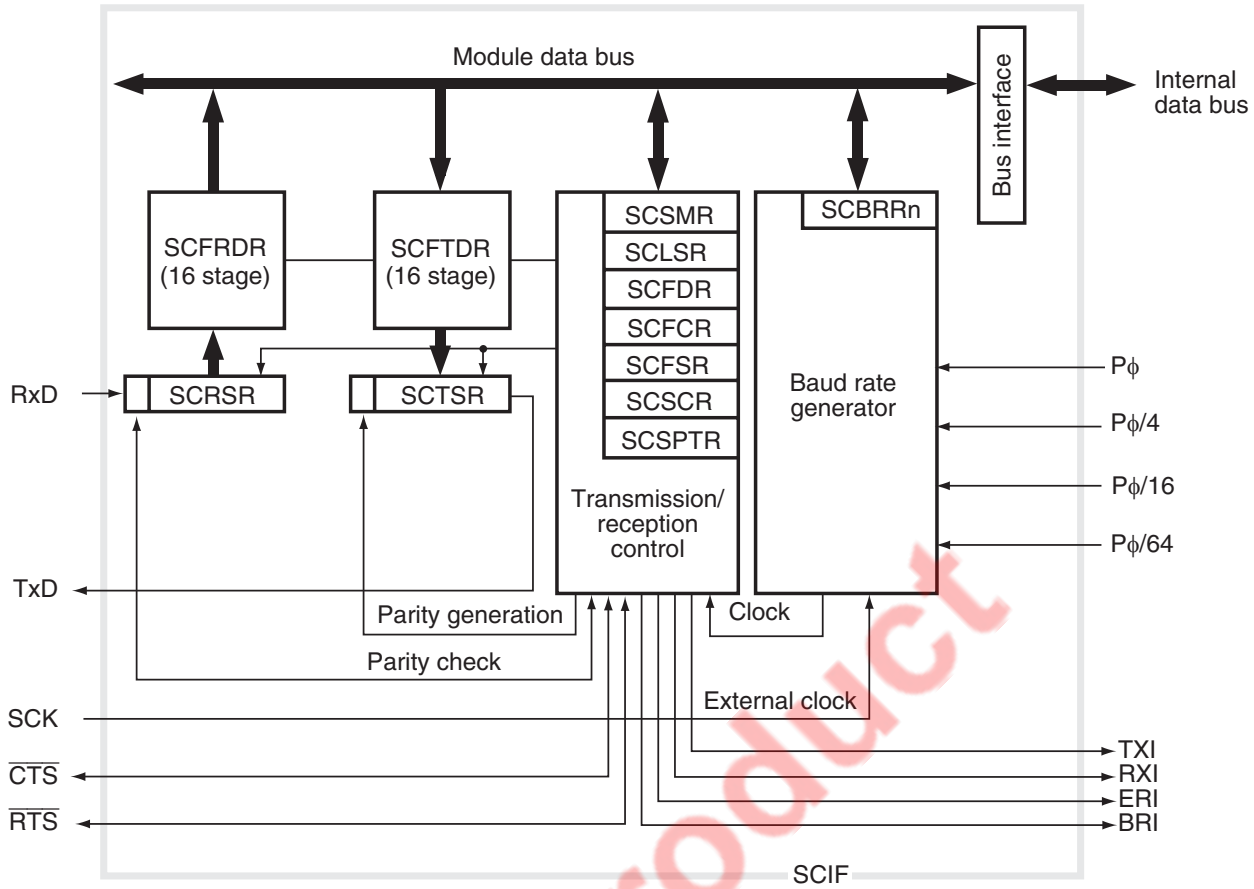
This LSI has a three-channel serial communication interface with FIFO (SCIF) that supports both asynchronous and clock synchronous serial communication. It also has 16-stage FIFO registers for both transmission and reception independently for each channel that enable this LSI to perform efficient high-speed continuous communication.

19.1.1 Features

- Asynchronous serial communication:
  - Serial data communication is performed by start-stop in character units. The SCIF can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity, framing, and overrun errors
  - Break detection: Break is detected when a framing error is followed by at least one frame at the space 0 level (low level). It is also detected by reading the RxD level directly from the port data register when a framing error occurs.
- Synchronous mode:
  - Serial data communication is synchronized with a clock signal. The SCIF can communicate with other chips having a synchronous communication function. There is one serial data communication format.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCIF can transmit and receive simultaneously. Both sections use 16-stage FIFO buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates

- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK pin (external)
- Four types of interrupts: Transmit-FIFO-data-empty, break, receive-FIFO-data-full, and receive-error interrupts are requested independently. The direct memory access controller (DMAC) can be activated to execute a data transfer by a transmit-FIFO-data-empty or receive-FIFO-data-full interrupt.
- When the SCIF is not in use, it can be stopped by halting the clock supplied to it, saving power.
- In asynchronous, on-chip modem control functions ($\overline{\text{RTS}}$ and $\overline{\text{CTS}}$).
- The quantity of data in the transmit and receive FIFO registers and the number of receive errors of the receive data in the receive FIFO register can be ascertained.
- A time-out error (DR) can be detected when receiving in asynchronous mode.

Figure 19.1 shows a block diagram of the SCIF for each channel.



[Legend]

SCRSR:	Receive shift register	SCFSR:	Serial status register
SCFRDR:	Receive FIFO data register	SCBRR:	Bit rate register
SCTSR:	Transmit shift register	SCSPTR:	Serial port register
SCFTDR:	Transmit FIFO data register	SCFCR:	FIFO control register
SCSMR:	Serial mode register	SCFDR:	FIFO data count register
SCSCR:	Serial control register	SCLSR:	Line status register

Figure 19.1 Block Diagram of SCIF

19.2 Pin Configuration

The SCIF has the serial pins summarized in table 19.1.

Table 19.1 SCIF Pins

Channel	Pin Name	Abbreviation	I/O	Function
0	Serial clock pin	SCK0	I/O	Clock I/O
	Receive data pin	RxD0	Input	Receive data input
	Transmit data pin	TxD0	Output	Transmit data output
	Request to send pin	$\overline{\text{RTS0}}$	I/O	Request to send
	Clear to send pin	$\overline{\text{CTS0}}$	I/O	Clear to send
1	Serial clock pin	SCK1	I/O	Clock I/O
	Receive data pin	RxD1	Input	Receive data input
	Transmit data pin	TxD1	Output	Transmit data output
	Request to send pin	$\overline{\text{RTS1}}$	I/O	Request to send
	Clear to send pin	$\overline{\text{CTS1}}$	I/O	Clear to send
2	Serial clock pin	SCK2	I/O	Clock I/O
	Receive data pin	RxD2	Input	Receive data input
	Transmit data pin	TxD2	Output	Transmit data output
	Request to send pin	$\overline{\text{RTS2}}$	I/O	Request to send
	Clear to send pin	$\overline{\text{CTS2}}$	I/O	Clear to send

19.3 Register Description

The SCIF has the following registers. These registers specify the data format and bit rate, and control the transmitter and receiver sections.

- Receive FIFO data register_0 (SCFRDR_0)
- Transmit FIFO data register_0 (SCFTDR_0)
- Serial mode register_0 (SCSMR_0)
- Serial control register_0 (SCSCR_0)
- Serial status register_0 (SCFSR_0)
- Bit rate register_0 (SCBRR_0)
- FIFO control register_0 (SCFCR_0)
- FIFO data count register_0 (SCFDR_0)
- Serial port register_0 (SCSPTR_0)
- Line status register_0 (SCLSR_0)
- Receive FIFO data register_1 (SCFRDR_1)
- Transmit FIFO data register_1 (SCFTDR_1)
- Serial mode register_1 (SCSMR_1)
- Serial control register_1 (SCSCR_1)
- Serial status register_1 (SCFSR_1)
- Bit rate register_1 (SCBRR_1)
- FIFO control register_1 (SCFCR_1)
- FIFO data count register_1 (SCFDR_1)
- Serial port register_1 (SCSPTR_1)
- Line status register_1 (SCLSR_1)
- Receive FIFO data register_2 (SCFRDR_2)
- Transmit FIFO data register_2 (SCFTDR_2)
- Serial mode register_2 (SCSMR_2)
- Serial control register_2 (SCSCR_2)
- Serial status register_2 (SCFSR_2)
- Bit rate register_2 (SCBRR_2)
- FIFO control register_2 (SCFCR_2)
- FIFO data count register_2 (SCFDR_2)
- Serial port register_2 (SCSPTR_2)
- Line status register_2 (SCLSR_2)

19.3.1 Receive Shift Register (SCRSR)

The receive shift register (SCRSR) receives serial data. Data input at the RxD pin is loaded into SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to SCFRDR, the receive FIFO data register. The CPU cannot read or write to SCRSR directly.

19.3.2 Receive FIFO Data Register (SCFRDR)

The 16-byte receive FIFO data register (SCFRDR) stores serial receive data. The SCIF completes the reception of one byte of serial data by moving the received data from the receive shift register (SCRSR) into SCFRDR for storage. Continuous reception is possible until 16 bytes are stored.

The CPU can read but not write to SCFRDR. If data is read when there is no receive data in the SCFRDR, the value is undefined. When this register is full of receive data, subsequent serial data is lost.

SCFRDR is initialized to undefined value by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
7 to 0	—	Undefined	R	FIFO for transmits serial data

19.3.3 Transmit Shift Register (SCTSR)

The transmit shift register (SCTSR) transmits serial data. The SCIF loads transmit data from the transmit FIFO data register (SCFTDR) into SCTSR, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCIF automatically loads the next transmit data from SCFTDR into SCTSR and starts transmitting again. The CPU cannot read or write to SCTSR directly.

19.3.4 Transmit FIFO Data Register (SCFTDR)

The transmit FIFO data register (SCFTDR) is a 16-byte FIFO register that stores data for serial transmission. When the SCIF detects that the transmit shift register (SCTSR) is empty, it moves transmit data written in the SCFTDR into SCTSR and starts serial transmission. Continuous serial transmission is performed until there is no transmit data left in SCFTDR.

When SCFTDR is full of transmit data (16 bytes), no more data can be written. If writing of new data is attempted, the data is ignored.

SCFTDR is initialized to undefined value by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
7 to 0	—	Undefined	W	FIFO for transmits serial data

19.3.5 Serial Mode Register (SCSMR)

The serial mode register (SCSMR) specifies the SCIF serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to SCSMR. SCSMR is initialized to H'0000 by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	C/$\overline{A}$	0	R/W	Communication Mode Selects whether the SCIF operates in asynchronous or synchronous mode. 0: Asynchronous mode 1: Synchronous mode

Bit	Bit Name	Initial value	R/W	Description
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data in asynchronous mode. In the synchronous mode, the data length is always eight bits, regardless of the CHR setting. 0: 8-bit data 1: 7-bit data* Note: * When 7-bit data is selected, the MSB (bit 7) of the transmit FIFO data register is not transmitted.
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting. 0: Parity bit not added or checked 1: Parity bit added and checked* Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/$\overline{E}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$) mode setting.

Bit	Bit Name	Initial value	R/W	Description
4	O/$\overline{E}$	0	R/W	Parity mode Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The O/$\overline{E}$ setting is ignored in synchronous mode, or in asynchronous mode when parity addition and checking is disabled. 0: Even parity*[1] 1: Odd parity*[2] Notes: 1. If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined. 2. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit length in asynchronous mode. This setting is used only in asynchronous mode. It is ignored in synchronous mode because no stop bits are added. When receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character. 0: One stop bit When transmitting, a single 1-bit is added at the end of each transmitted character. 1: Two stop bits When transmitting, two 1 bits are added at the end of each transmitted character.

Bit	Bit Name	Initial value	R/W	Description
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	CKS1	0	R/W	Clock Select 1, 0
0	CKS0	0	R/W	Select the internal clock source of the on-chip baud rate generator. Four clock sources are available. Pϕ, Pϕ/4, Pϕ/16 and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 19.3.8, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

19.3.6 Serial Control Register (SCSCR)

The serial control register (SCSCR) operates the SCIF transmitter/receiver, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCSCR. SCSCR is initialized to H'0000 by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the transmit-FIFO-data-empty interrupt (TXI) requested when the serial transmit data is transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), when the quantity of data in the transmit FIFO register becomes less than the specified number of transmission triggers, and when the TDFE flag in the serial status register (SCFSR) is set to1. 0: Transmit-FIFO-data-empty interrupt request (TXI) is disabled 1: Transmit-FIFO-data-empty interrupt request (TXI) is enabled* Note: * The TXI interrupt request can be cleared by writing a greater quantity of transmit data than the specified transmission trigger number to SCFTDR and by clearing TDFE to 0 after reading 1 from TDFE, or can be cleared by clearing TIE to 0.

Bit	Bit Name	Initial value	R/W	Description
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive-data-full (RXI) interrupts requested when the RDF flag or DR flag in serial status register (SCFSR) is set to1, receive-error (ERI) interrupts requested when the ER flag in SCFSR is set to1, and break (BRI) interrupts requested when the BRK flag in SCFSR or the ORER flag in line status register (SCLSR) is set to1. 0: Receive-data-full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are disabled 1: Receive-data-full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are enabled* Note: * RXI interrupt requests can be cleared by reading the DR or RDF flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0. ERI or BRI interrupt requests can be cleared by reading the ER, BR or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0.
5	TE	0	R/W	Transmit Enable Enables or disables the SCIF serial transmitter. 0: Transmitter disabled 1: Transmitter enabled* Note: * Serial transmission starts after writing of transmit data into SCFTDR. Select the transmit format in SCSMR and SCFCR and reset the transmit FIFO before setting TE to 1.

Bit	Bit Name	Initial value	R/W	Description
4	RE	0	R/W	Receive Enable Enables or disables the SCIF serial receiver. 0: Receiver disabled*[1] 1: Receiver enabled*[2] Notes: 1. Clearing RE to 0 does not affect the receive flags (DR, ER, BRK, RDF, FER, PER, and ORER). These flags retain their previous values. 2. Serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock input is detected in synchronous mode. Select the receive format in SCSMR and SCFCR and reset the receive FIFO before setting RE to 1.
3	REIE	0	R	Receive Error Interrupt Enable Enables or disables the receive-error (ERI) interrupts and break (BRI) interrupts. The setting of REIE bit is valid only when RIE bit is set to 0. 0: Receive-error interrupt (ERI) and break interrupt (BRI) requests are disabled 1: Receive-error interrupt (ERI) and break interrupt (BRI) requests are enabled* Note: * ERI or BRI interrupt requests can be cleared by reading the ER, BR or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0. Even if RIE is set to 0, when REIE is set to 1, ERI or BRI interrupt requests are enabled. Set so If SCIF wants to inform INTC of ERI or BRI interrupt requests during DMA transfer.

Bit	Bit Name	Initial value	R/W	Description
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	CKE1	0	R/W	Clock Enable 1, 0
0	CKE0	0	R/W	Select the SCIF clock source and enable or disable clock output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin can be used for serial clock output or serial clock input. If the serial clock output is set in synchronous mode, the communication mode bit (C/$\overline{A}$) in SCSMR2 is set to 1, and then CKE1 and CKE0 bits are set. • Asynchronous mode 00: Internal clock, SCK pin used for input pin (input signal is ignored) 01: Internal clock, SCK pin used for clock output (The output clock frequency is 16 times the bit rate.) 10: External clock, SCK pin used for clock input (The input clock frequency is 16 times the bit rate.) 11: Setting prohibited • Synchronous mode 00: Internal clock, SCK pin used for serial clock output 01: Internal clock, SCK pin used for serial clock output 10: External clock, SCK pin used for serial clock input 11: Setting prohibited

19.3.7 Serial Status Register (SCFSR)

The serial status register (SCFSR) is a 16-bit register. The upper 8 bits indicate the number of receives errors in the SCFRDR data, and the lower 8 bits indicate the status flag indicating SCIF operating state.

The CPU can always read and write to SCFSR, but cannot write 1 to the status flags (ER, TEND, TDFE, BRK, RDF, and DR). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 3 (FER) and 2 (PER) are read-only bits that cannot be written. SCFSR is initialized to H'0060 by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
15	PER3	0	R	Number of Parity Errors
14	PER2	0	R	Indicate the quantity of data including a parity error in the receive data stored in the receive FIFO data register (SCFRDR). The value indicated by bits 15 to 12 represents the number of parity errors in SCFRDR. When parity errors have occurred in all 16-byte receive data in SCFRDR, PER3 to PER0 show 0.
13	PER1	0	R	
12	PER0	0	R	
11	FER3	0	R	Number of Framing Errors
10	FER2	0	R	Indicate the quantity of data including a framing error in the receive data stored in SCFRDR. The value indicated by bits 11 to 8 represents the number of framing errors in SCFRDR. When framing errors have occurred in all 16-byte receive data in SCFRDR, FER3 to FER0 show 0.
9	FER1	0	R	
8	FER0	0	R	

Bit	Bit Name	Initial value	R/W	Description
7	ER	0	R/(W)*	Receive Error Indicates the occurrence of a framing error, or of a parity error when receiving data that includes parity. *1 0: Receiving is in progress or has ended normally [Clearing conditions] • ER is cleared to 0 a power-on reset • ER is cleared to 0 when the chip is when 0 is written after 1 is read from ER 1: A framing error or parity error has occurred. [Setting conditions] • ER is set to 1 when the stop bit is 0 after checking whether or not the last stop bit of the received data is 1 at the end of one data receive operation*2 • ER is set to 1 when the total number of 1s in the receive data plus parity bit does not match the even/odd parity specified by the O/$\overline{E}$ bit in SCSMR Notes: 1. Clearing the RE bit to 0 in SCSCR does not affect the ER bit, which retains its previous value. Even if a receive error occurs, the receive data is transferred to SCFRDR and the receive operation is continued. Whether or not the data read from SCRDR includes a receive error can be detected by the FER and PER bits in SCFSR. 2. In two stop bits mode, only the first stop bit is checked; the second stop bit is not checked.

Bit	Bit Name	Initial value	R/W	Description
6	TEND	0	R/(W)*	Transmit End Indicates that when the last bit of a serial character was transmitted, SCFTDR did not contain valid data, so transmission has ended. 0: Transmission is in progress [Clearing condition] • TEND is cleared to 0 when 0 is written after 1 is read from TEND after transmit data is written in SCFTDR 1: End of transmission [Setting conditions] • TEND is set to 1 when the chip is a power-on reset • TEND is set to 1 when TE is cleared to 0 in the serial control register (SCSCR) • TEND is set to 1 when SCFTDR does not contain receive data when the last bit of a one-byte serial character is transmitted Note: When the transmit FIFO data empty DMA transfer request is generated and transmit data is written to SCFTDR by the DMAC, do not use this flag as a transmit end flag.

Bit	Bit Name	Initial value	R/W	Description
5	TDFE	0	R/(W)*	Transmit FIFO Data Empty Indicates that data has been transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), the quantity of data in SCFTDR has become less than the transmission trigger number specified by the TTRG1 and TTRG0 bits in the FIFO control register (SCFCR), and writing of transmit data to SCFTDR is enabled. 0: The quantity of transmit data written to SCFTDR is greater than the specified transmission trigger number [Clearing conditions] • TDFE is cleared to 0 when data exceeding the specified transmission trigger number is written to SCFTDR after 1 is read from TDFE and then 0 is written • TDFE is cleared to 0 when DMAC write data exceeding the specified transmission trigger number to SCFTDR 1: The quantity of transmit data in SCFTDR is less than the specified transmission trigger number* [Setting conditions] • TDFE is set to 1 by a power-on reset • TDFE is set to 1 when the quantity of transmit data in SCFTDR becomes less than the specified transmission trigger number as a result of transmission Note: * Since SCFTDR is a 16-byte FIFO register, the maximum quantity of data that can be written when TDFE is 1 is "16 minus the specified transmission trigger number". If an attempt is made to write additional data, the data is ignored. The quantity of data in SCFTDR is indicated by the upper 8 bits of SCFDR.

Bit	Bit Name	Initial value	R/W	Description
4	BRK	0	R/(W)*	Break Detection Indicates that a break signal has been detected in receive data. 0: No break signal received [Clearing conditions] • BRK is cleared to 0 when the chip is a power-on reset • BRK is cleared to 0 when software reads BRK after it has been set to 1, then writes 0 to BRK 1: Break signal received* [Setting condition] • BRK is set to 1 when data including a framing error is received, and a framing error occurs with space 0 in the subsequent receive data Note: * When a break is detected, transfer of the receive data (H'00) to SCFRDR stops after detection. When the break ends and the receive signal becomes mark 1, the transfer of receive data resumes.
3	FER	0	R	Framing Error Indicates a framing error in the data read from the next receive FIFO data register (SCFRDR) in asynchronous mode. 0: No receive framing error occurred in the next data read from SCFRDR [Clearing conditions] • FER is cleared to 0 when the chip undergoes a power-on reset • FER is cleared to 0 when no framing error is present in the next data read from SCFRDR 1: A receive framing error occurred in the next data read from SCFRDR. [Setting condition] • FER is set to 1 when a framing error is present in the next data read from SCFRDR

Bit	Bit Name	Initial value	R/W	Description
2	PER	0	R	Parity Error Indicates a parity error in the data read from the next receive FIFO data register (SCFRDR) in asynchronous mode. 0: No receive parity error occurred in the next data read from SCFRDR [Clearing conditions] • PER is cleared to 0 when the chip undergoes a power-on reset • PER is cleared to 0 when no parity error is present in the next data read from SCFRDR 1: A receive parity error occurred in the data read from SCFRDR [Setting condition] • PER is set to 1 when a parity error is present in the next data read from SCFRDR

Bit	Bit Name	Initial value	R/W	Description
1	RDF	0	R/(W)*	Receive FIFO Data Full Indicates that receive data has been transferred to the receive FIFO data register (SCFRDR), and the quantity of data in SCFRDR has become more than the receive trigger number specified by the RTRG1 and RTRG0 bits in the FIFO control register (SCFCR). 0: The quantity of transmit data written to SCFRDR is less than the specified receive trigger number [Clearing conditions] • RDF is cleared to 0 by a power-on reset, standby mode • RDF is cleared to 0 when the SCFRDR is read until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number after 1 is read from RDF and then 0 is written • RDF is cleared to 0 when DMAC read SCFRDR until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number 1: The quantity of receive data in SCFRDR is more than the specified receive trigger number [Setting condition] • RDF is set to 1 when a quantity of receive data more than the specified receive trigger number is stored in SCFRDR* Note: * SCFTDR is a 16-byte FIFO register. When RDF is 1, the specified receive trigger number of data can be read. If an attempt is made to read after all the data in SCFRDR has been read, the data is undefined. The quantity of receive data in SCFRDR is indicated by the lower 8 bits of SCFDR.

Bit	Bit Name	Initial value	R/W	Description
0	DR	0	R/(W)*	Receive Data Ready Indicates that the quantity of data in the receive FIFO data register (SCFRDR) is less than the specified receive trigger number, and that the next data has not yet been received after the elapse of 15 ETU from the last stop bit in asynchronous mode. In clock synchronous mode, this bit is not set to 1. 0: Receiving is in progress, or no receive data remains in SCFRDR after receiving ended normally [Clearing conditions] • DR is cleared to 0 when the chip undergoes a power-on reset • DR is cleared to 0 when all receive data are read after 1 is read from DR and then 0 is written • DR is cleared to 0 when all receive data are read by DMAC 1: Next receive data has not been received [Setting condition] • DR is set to 1 when SCFRDR contains less data than the specified receive trigger number, and the next data has not yet been received after the elapse of 15 ETU from the last stop bit.* Note: * This is equivalent to 1.5 frames with the 8-bit, 1-stop-bit format. (ETU: elementary time unit)

Note: * The only value that can be written is 0 to clear the flag.

19.3.8 Bit Rate Register (SCBRR)

The bit rate register (SCBRR) is an 8-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR. SCBRR is initialized to H'FF by a power-on reset. Each channel has independent baud rate generator control, so different values can be set in three channels.

The SCBRR setting is calculated as follows:

- Asynchronous mode:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

- Synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 19.2.)

Table 19.2 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: The bit rate error in asynchronous is given by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64^{2n-1} \times 2} - 1 \right\} \times 100$$

Table 19.3 lists examples of SCBRR settings in asynchronous mode, and table 19.4 lists examples of SCBRR settings in synchronous mode.

Table 19.3 Bit Rates and SCBRR Settings in Asynchronous Mode

	Pϕ (MHz)								
	5			6			6.144		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	88	–0.25	2	106	–0.44	2	108	0.08
150	2	64	0.16	2	77	0.16	2	79	0.00
300	1	129	0.16	1	155	0.16	1	159	0.00
600	1	64	0.16	1	77	0.16	1	79	0.00
1200	0	129	0.16	0	155	0.16	0	159	0.00
2400	0	64	0.16	0	77	0.16	0	79	0.00
4800	0	32	–1.36	0	38	0.16	0	39	0.00
9600	0	15	1.73	0	19	–2.34	0	19	0.00
19200	0	7	1.73	0	9	–2.34	0	9	0.00
31250	0	4	0.00	0	5	0.00	0	5	2.40
38400	0	3	1.73	0	4	–2.34	0	4	0.00

	Pϕ (MHz)								
	7.3728			8			9.8304		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	130	–0.07	2	141	0.03	2	174	–0.26
150	2	95	0.00	2	103	0.16	2	127	0.00
300	1	191	0.00	1	207	0.16	1	255	0.00
600	1	95	0.00	1	103	0.16	1	127	0.00
1200	0	191	0.00	0	207	0.16	0	255	0.00
2400	0	95	0.00	0	103	0.16	0	127	0.00
4800	0	47	0.00	0	51	0.16	0	63	0.00
9600	0	23	0.00	0	25	0.16	0	31	0.00
19200	0	11	0.00	0	12	0.16	0	15	0.00
31250	0	6	5.33	0	7	0.00	0	9	–1.70
38400	0	5	0.00	0	6	–6.99	0	7	0.00

	Pϕ (MHz)											
	10			12			12.288			14.7456		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	–0.25	2	212	0.03	2	217	0.08	3	64	0.70
150	2	129	0.16	2	155	0.16	2	159	0.00	2	191	0.00
300	2	64	0.16	2	77	0.16	2	79	0.00	2	95	0.00
600	1	129	0.16	1	155	0.16	1	159	0.00	1	191	0.00
1200	1	64	0.16	1	77	0.16	1	79	0.00	1	95	0.00
2400	0	129	0.16	0	155	0.16	0	159	0.00	0	191	0.00
4800	0	64	0.16	0	77	0.16	0	79	0.00	0	95	0.00
9600	0	32	–1.36	0	38	0.16	0	39	0.00	0	47	0.00
19200	0	15	1.73	0	19	0.16	0	19	0.00	0	23	0.00
31250	0	9	0.00	0	11	0.00	0	11	2.40	0	14	–1.70
38400	0	7	1.73	0	9	–2.34	0	9	0.00	0	11	0.00

	Pϕ (MHz)											
	16			19.6608			20			24		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	70	0.03	3	86	0.31	3	88	–0.25	3	106	–0.44
150	2	207	0.16	2	255	0.00	3	64	0.16	3	77	0.16
300	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
600	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
1200	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
2400	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
4800	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
9600	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
19200	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16
31250	0	15	0.00	0	19	–1.70	0	19	0.00	0	23	0.00
38400	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34

	Pϕ (MHz)											
	24.576			28.7			30			33		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	108	0.08	3	126	0.31	3	132	0.13	3	145	0.33
150	3	79	0.00	3	92	0.46	3	97	–0.35	3	106	0.39
300	2	159	0.00	2	186	–0.08	2	194	0.16	2	214	-0.07
600	2	79	0.00	2	92	0.46	2	97	–0.35	2	106	0.39
1200	1	159	0.00	1	186	–0.08	1	194	0.16	1	214	-0.07
2400	1	79	0.00	1	92	0.46	1	97	–0.35	1	106	0.39
4800	0	159	0.00	0	186	–0.08	0	194	–1.36	0	214	-0.07
9600	0	79	0.00	0	92	0.46	0	97	–0.35	0	106	0.39
19200	0	39	0.00	0	46	–0.61	0	48	–0.35	0	53	-0.54
31250	0	24	–1.70	0	28	–1.03	0	29	0.00	0	32	0.00
38400	0	19	0.00	0	22	1.55	0	23	1.73	0	26	-0.54

Note: Settings with an error of 1% or less are recommended.

Table 19.4 Bit Rates and SCBRR Settings in Synchronous Mode

Bit Rate (bits/s)	Pϕ (MHz)											
	5		8		16		28.7		30		33	
	n	N	n	N	n	N	n	N	n	N	n	N
110	—	—	—	—	—	—	—	—	—	—	—	—
250	3	77	3	124	3	249	—	—	—	—	—	—
500	3	38	2	249	3	124	3	223	3	233	3	255
1k	2	77	2	124	2	249	3	111	3	116	3	125
2.5k	1	124	1	199	2	99	2	178	2	187	2	200
5k	0	249	1	99	1	199	2	89	2	93	2	100
10k	0	124	0	199	1	99	1	178	1	187	1	200
25k	0	49	0	79	0	159	1	71	1	74	1	80
50k	0	24	0	39	0	79	0	143	0	149	0	160
100k	—	—	0	19	0	39	0	71	0	74	0	80
250k	0	4	0	7	0	15	—	—	0	29	0	31
500k	—	—	0	3	0	7	—	—	0	14	0	15
1M	—	—	—	—	0	3	—	—	—	—	0	7
2M			—	—	—	—	—	—	—	—	—	—

[Legend]
Blank: No setting possible
—: Setting possible, but error occurs
Note: Set the BRR value that satisfies the external specifications.

Table 19.5 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Tables 19.6 and 19.7 list the maximum rates for external clock input.

Table 19.5 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pϕ (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
5	156250	0	0
4.9152	153600	0	0
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.6608	614400	0	0
20	625000	0	0
24	750000	0	0
24.576	768000	0	0
28.7	896875	0	0
30	937500	0	0
33	1031250	0	0

Table 19.6 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
5	1.2500	78125
4.9152	1.2288	76800
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
19.6608	4.9152	307200
20	5.0000	312500
24	6.0000	375000
24.576	6.1440	384000
28.7	7.1750	448436
30	7.5000	468750
33	8.25	515625

Table 19.7 Maximum Bit Rates with External Clock Input (Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
5	0.8333	833333.3
8	1.3333	1333333.3
16	2.6667	2666666.7
24	4.0000	4000000.0
28.7	4.7833	4783333.3
30	5.0000	5000000.0
33	5.5000	5500000.0

19.3.9 FIFO Control Register (SCFCR)

The FIFO control register (SCFCR) resets the quantity of data in the transmit and receive FIFO registers, sets the trigger data quantity, and contains an enable bit for loop-back testing. SCFCR can always be read and written to by the CPU. It is initialized to H'0000 by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	RSTRG2	0	R/W	RTS Output Active Trigger
9	RSTRG1	0	R/W	When the quantity of receive data in receive FIFO register (SCFRDR) becomes more than the number shown below, $\overline{\text{RTS}}$ signal is set to high.
8	RSTRG0	0	R/W	000: 15 001: 1 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14 Note: Set the trigger number to 1 when the receive data is transferred by the DMAC in synchronous mode. If the set trigger number is other than 1, the receive data remains in SCFRDR should be read by the CPU.

Bit	Bit Name	Initial value	R/W	Description
7	RTRG1	0	R/W	Receive FIFO Data Trigger
6	RTRG0	0	R/W	Set the quantity of receive data which sets the receive data full (RDF) flag in the serial status register (SCFSR). The RDF flag is set when the quantity of receive data stored in the receive FIFO register (SCFRDR) is increased more than the set trigger number shown below. • Asynchronous mode 00: 1 01: 4 10: 8 11: 14 • Synchronous mode 00: 1 01: 2 10: 8 11: 14
5	TTRG1	0	R/W	Transmit FIFO Data Trigger 1, 0
4	TTRG0	0	R/W	Set the quantity of remaining transmit data which sets the transmit FIFO data register empty (TDFE) flag in the serial status register (SCFSR). The TDFE flag is set when the quantity of transmit data in the transmit FIFO data register (SCFTDR) becomes less than the set trigger number shown below. 00: 8 (8)* 01: 4 (12)* 10: 2 (14)* 11: 0 (16)* Note: * Values in parentheses mean the number of empty bytes in SCFTDR when the TDFE flag is set to 1.
3	MCE	0	R/W	Modem Control Enable Enables modem control signals $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$. In synchronous mode, MCE bit should always be 0. 0: Modem signal disabled* 1: Modem signal enabled Note: * $\overline{\text{CTS}}$ is fixed at active 0 regardless of the input value, and $\overline{\text{RTS}}$ is also fixed at 0.

Bit	Bit Name	Initial value	R/W	Description
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Disables the transmit data in the transmit FIFO data register and resets the data to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a power-on reset.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Disables the receive data in the receive FIFO data register and resets the data to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a power-on reset.
0	LOOP	0	R/W	Loop-Back Test Internally connects the transmit output pin (TxD) and receive input pin (RxD) and enables loop-back testing. 0: Loop back test disabled 1: Loop back test enabled

19.3.10 FIFO Data Count Register (SCFDR)

SCFDR is a 16-bit register which indicates the quantity of data stored in the transmit FIFO data register (SCFTDR) and the receive FIFO data register (SCFRDR). It indicates the quantity of transmit data in SCFTDR with the upper 8 bits, and the quantity of receive data in SCFRDR with the lower 8 bits. SCFDR can always be read by the CPU. SCFDR is initialized to H'0000 by a power on reset.

Bit	Bit Name	Initial value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	T4	0	R	T4 to T0 bits indicate the quantity of non-transmitted data stored in SCFTDR. H'00 means no transmit data, and H'10 means that SCFTDR is full of transmit data.
11	T3	0	R	
10	T2	0	R	
9	T1	0	R	
8	T0	0	R	
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	R4	0	R	R4 to R0 bits indicate the quantity of receive data stored in SCFRDR. H'00 means no receive data, and H'10 means that SCFRDR full of receive data.
3	R3	0	R	
2	R2	0	R	
1	R1	0	R	
0	R0	0	R	

19.3.11 Serial Port Register (SCSPTR)

The serial port register (SCSPTR) controls input/output and data of pins multiplexed to SCIF function. Bits 1 and 0 can input data from RxD pin and output data to TxD pin, so they control break of serial transmitting/receiving. Bits 3 and 2 can control input/output data of SCK pin, bits 5 and 4 can control input/output data of $\overline{\text{CTS}}$ pin, and bits 7 and 6 can control input/output data of $\overline{\text{RTS}}$ pin.

The CPU can always read and write to SCSPTR. SCSPTR is initialized to H'0050 by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	RTSIO	0	R/W	RTS Port Input/Output Indicates the input or output of $\overline{\text{RTS}}$ pin. When $\overline{\text{RTS}}$ pin is used as port outputting the RTSDT bit, the MCE bit of FIFO control register (SCFCR) should be set to 0. 0: Not output the RTSDT bit to $\overline{\text{RTS}}$ pin 1: Output the RTSDT bit to $\overline{\text{RTS}}$ pin
6	RTSDT	1	R/W	RTS Port Data Indicates the data of $\overline{\text{RTS}}$ pin used as port. Input/output is specified by RTSIO bit. When output, the value of RTSDT bit is outputted to $\overline{\text{RTS}}$ pin. Whenever input or output, $\overline{\text{RTS}}$ pin status is read from RTSDT bit. However Port Function of PFC (Pin Function Controller) must be set to $\overline{\text{RTS}}$ input/output. 0: Input/output data is low level 1: Input/output data is high level
5	CTSIO	0	R/W	CTS Port Input/Output Indicates the input or output of $\overline{\text{CTS}}$ pin. When $\overline{\text{CTS}}$ pin is used as port outputting the CTSDT bit, the MCE bit of FIFO control register (SCFCR) should be set to 0. 0: Not output the CTSDT bit to $\overline{\text{CTS}}$ pin 1: Output the CTSDT bit to $\overline{\text{CTS}}$ pin
4	CTSDT	1	R/W	CTS Port Data Indicates the data of $\overline{\text{CTS}}$ pin used as port. Input/output is specified by CTSIO bit. When output, the value of CTSDT bit is outputted to $\overline{\text{CTS}}$ pin. Whenever input or output, $\overline{\text{CTS}}$ pin status is read from CTSDT bit. However Port Function of PFC (Pin Function Controller) must be set to $\overline{\text{CTS}}$ input/output. 0: Input/output data is low level 1: Input/output data is high level

Bit	Bit Name	Initial value	R/W	Description
3	SCKIO	0	R/W	SCK Port Input/Output Indicates the input or output of SCK pin. When SCK pin is used as port outputting the SCKDT bit, the CKE1, CKE0 bit of serial control register (SCSCR) should be set to 0. 0: Not output the SCKDT bit to SCK pin 1: Output the SCKDT bit to SCK pin
2	SCKDT	0	R/W	SCK Port Data Indicates the data of SCK pin used as port. Input/output is specified by SCKIO bit. When output, the value of SCKDT bit is outputted to SCK pin. Whenever input or output, SCK pin status is read from SCKDT bit. However Port Function of PFC (Pin Function Controller) must be set to SCK input/output. 0: Input/output data is low level 1: Input/output data is high level
1	SPB2IO	0	R/W	Serial Port Break Input/Output Indicates the input or output of TxD pin. When TxD pin is used as port outputting the SPB2DT bit, the TE bit of serial control register (SCSCR) should be set to 0. 0: Not output the SPB2DT bit to TxD pin 1: Output the SPB2DT bit to TxD pin
0	SPB2DT	0	R/W	Serial Port Break Data Indicates the input data of RxD pin and the output data of TxD pin used as port. Output of TxD pin is specified by SPB2IO bit. When output, the value of SPB2DT bit is outputted to TxD pin. Whenever input or output, RxD pin status is read from SPB2DT bit. However Port Function of PFC (Pin Function Controller) must be set to TxD output and RxD input. 0: Input/output data is low level 1: Input/output data is high level

19.3.12 Line Status Register (SCLSR)

The CPU can always read or write to SCLSR, but cannot write 1 to the ORER flag. This flag can be cleared to 0 only if it has first been read (after being set to 1). SCLSR is initialized to H'0000 by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ORER	0	R/(W)*	Overrun Error Indicates the occurrence of an overrun error. 0: Receiving is in progress or has ended normally*[1] [Clearing conditions] • ORER is cleared to 0 when the chip is a power-on reset • ORER is cleared to 0 when 0 is written after 1 is read from ORER. 1: An overrun error has occurred*[2] [Setting condition] • ORER is set to 1 when the next serial receiving is finished while the receive FIFO is full of 16-byte receive data. Notes: 1. Clearing the RE bit to 0 in SCSCR does not affect the ORER bit, which retains its previous value. 2. The receive FIFO data register (SCFRDR) hold the data before an overrun error is occurred, and the next receive data is extinguished. When ORER is set to 1, SCIF cannot continue the next serial receiving.

Note: * The only value that can be written is 0 to clear the flag.

19.4 Operation

19.4.1 Overview

For serial communication, the SCIF has an asynchronous mode in which characters are synchronized individually, and a synchronous mode in which communication is synchronized with clock pulses. The SCIF has a 16-byte FIFO buffer for both transmit and receive operations, reducing the overhead of the CPU, and enabling continuous high-speed communication. Moreover, it has $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ signals as modem control signals. The transmission format is selected in the serial mode register (SCSMR). The SCIF clock source is selected by the combination of the CKE1 and CKE0 bits in the serial control register (SCSCR).

Asynchronous Mode:

- Data length is selectable: 7 or 8 bits
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, receive FIFO data full, overrun errors, receive data ready, and breaks.
- The number of stored data bytes is indicated for both the transmit and receive FIFO registers.
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the on-chip baud rate generator.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Synchronous Mode:

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the on-chip baud rate generator, and outputs a serial clock signal to external devices.
  - When an external clock is selected, the SCIF operates on the input serial clock. The on-chip baud rate generator is not used.

Table 19.8 SCSMR Settings and SCIF Communication Formats

SCSMR Settings					SCIF Communication Format		
Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Parity Bit	Stop Bit Length
0	0	0	0	Asynchronous	8 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
	1	0	0		7 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
1	*	*	*	Synchronous	8 bits	Not set	None

Note: *: Don't care

Table 19.9 SCSMR and SCSCR Settings and SCIF Clock Source Selection

SCSMR	SCSCR Settings			SCIF Transmit/Receive Clock	
Bit 7 C/$\overline{A}$	Bit 1 CKE1	Bit 0 CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous	Internal	SCIF does not use the SCK pin
		1			Outputs a clock with a frequency 16 times the bit rate
	1	0		External	Inputs a clock with frequency 16 times the bit rate
1	0	*	Synchronous	Internal	Outputs the serial clock
	1	0		External	Inputs the serial clock

Note: *: Don't care

19.4.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCIF are independent, so full duplex communication is possible. The transmitter and receiver are 16-byte FIFO buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 19.2 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCIF monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCIF synchronizes at the falling edge of the start bit. The SCIF samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

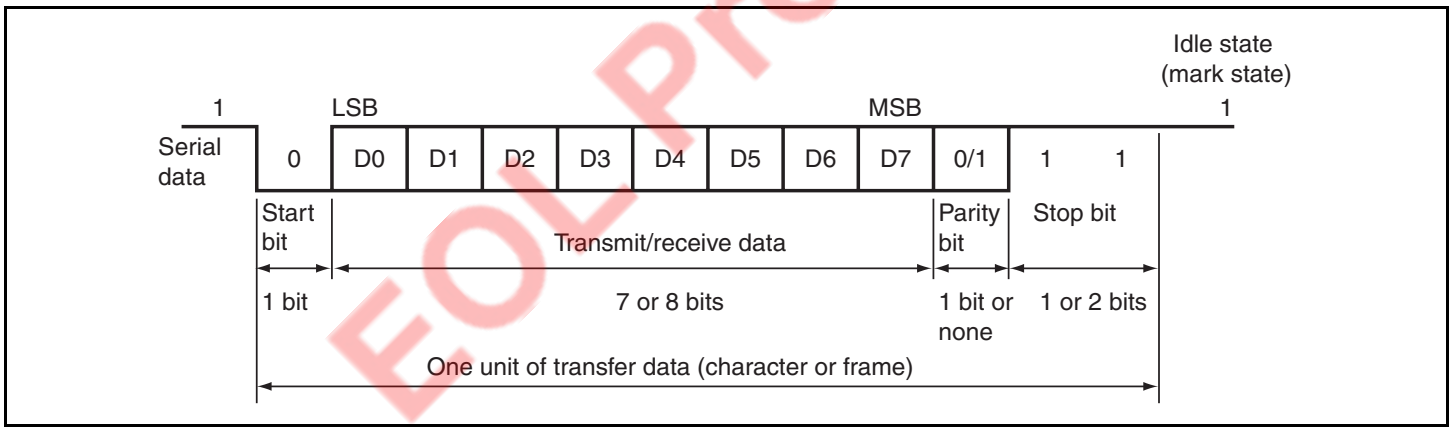


Figure 19.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

Transmit/Receive Formats: Table 19.10 lists the 8 communication formats that can be selected in asynchronous mode. The format is selected by settings in the serial mode register (SCSMR).

Table 19.10 Serial Communication Formats (Asynchronous Mode)

SCSMR Bits			Serial Transmit/Receive Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	START	8-bit data								STOP		
0	0	1	START	8-bit data								STOP	STOP	
0	1	0	START	8-bit data								P	STOP	
0	1	1	START	8-bit data								P	STOP	STOP
1	0	0	START	7-bit data							STOP			
1	0	1	START	7-bit data							STOP	STOP		
1	1	0	START	7-bit data							P	STOP		
1	1	1	START	7-bit data							P	STOP	STOP	

[Legend]
START: Start bit
STOP: Stop bit
P: Parity bit

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCIF transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the serial control register (SCSCR) (table 19.9).

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCIF operates on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to 16 times the desired bit rate.

Transmitting and Receiving Data:

- **SCIF Initialization (Asynchronous Mode)**

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF as follows.

When changing the operation mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing TE and RE to 0, however, does not initialize the serial status register (SCFSR), transmit FIFO data register (SCFTDR), or receive FIFO data register (SCFRDR), which retain their previous contents. Clear TE to 0 after all transmit data has been transmitted and the TEND flag in the SCFSR is set. The TE bit can be cleared to 0 during transmission, but the transmit data goes to the Mark state after the bit is cleared to 0. Set the TFRST bit in SCFCR to 1 and reset SCFTDR before TE is set again to start transmission.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCIF operation becomes unreliable if the clock is stopped.

Figure 19.3 shows a sample flowchart for initializing the SCIF.

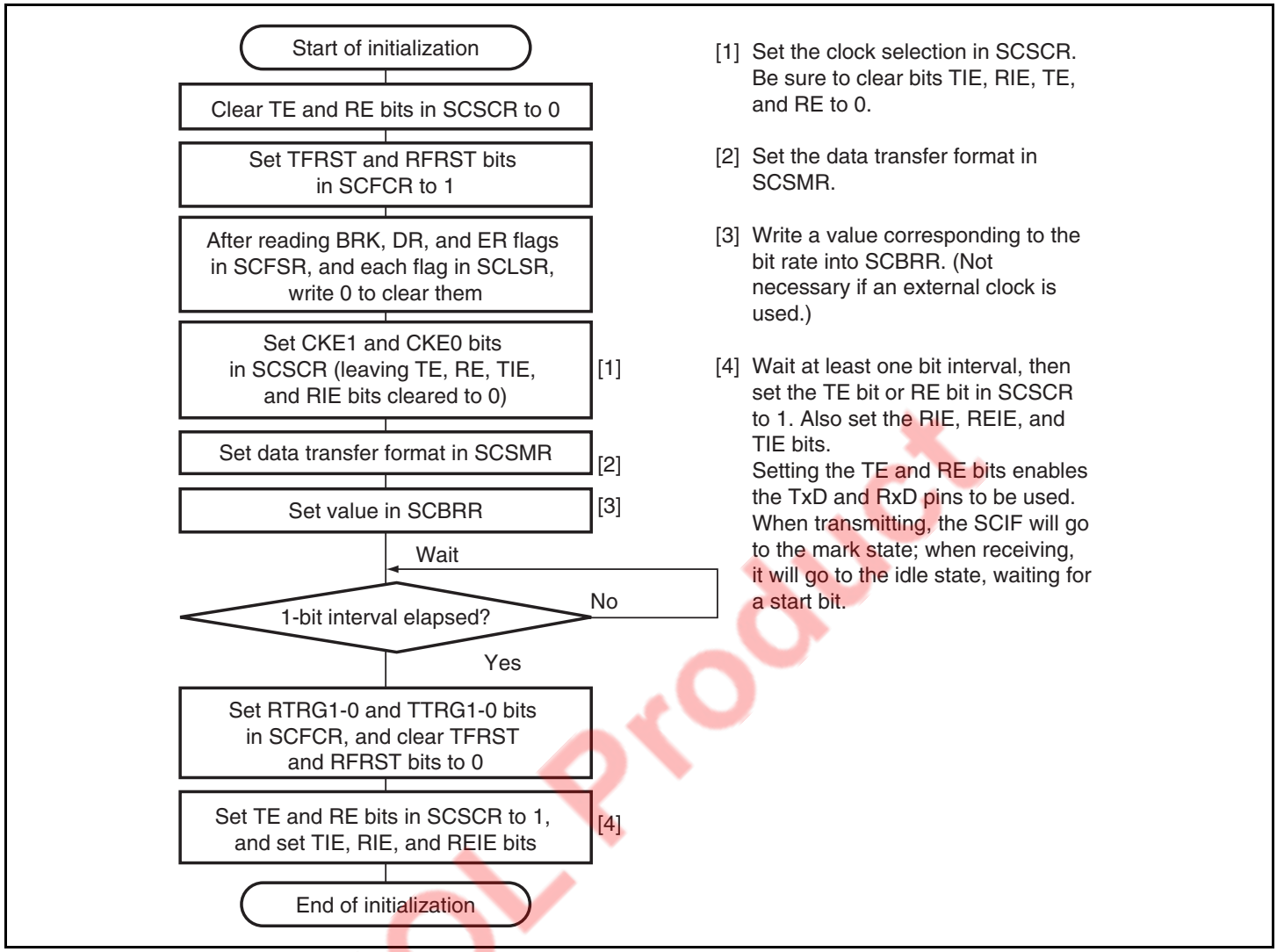


Figure 19.3 Sample Flowchart for SCIF Initialization

- **Transmitting Serial Data (Asynchronous Mode)**

Figure 19.4 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

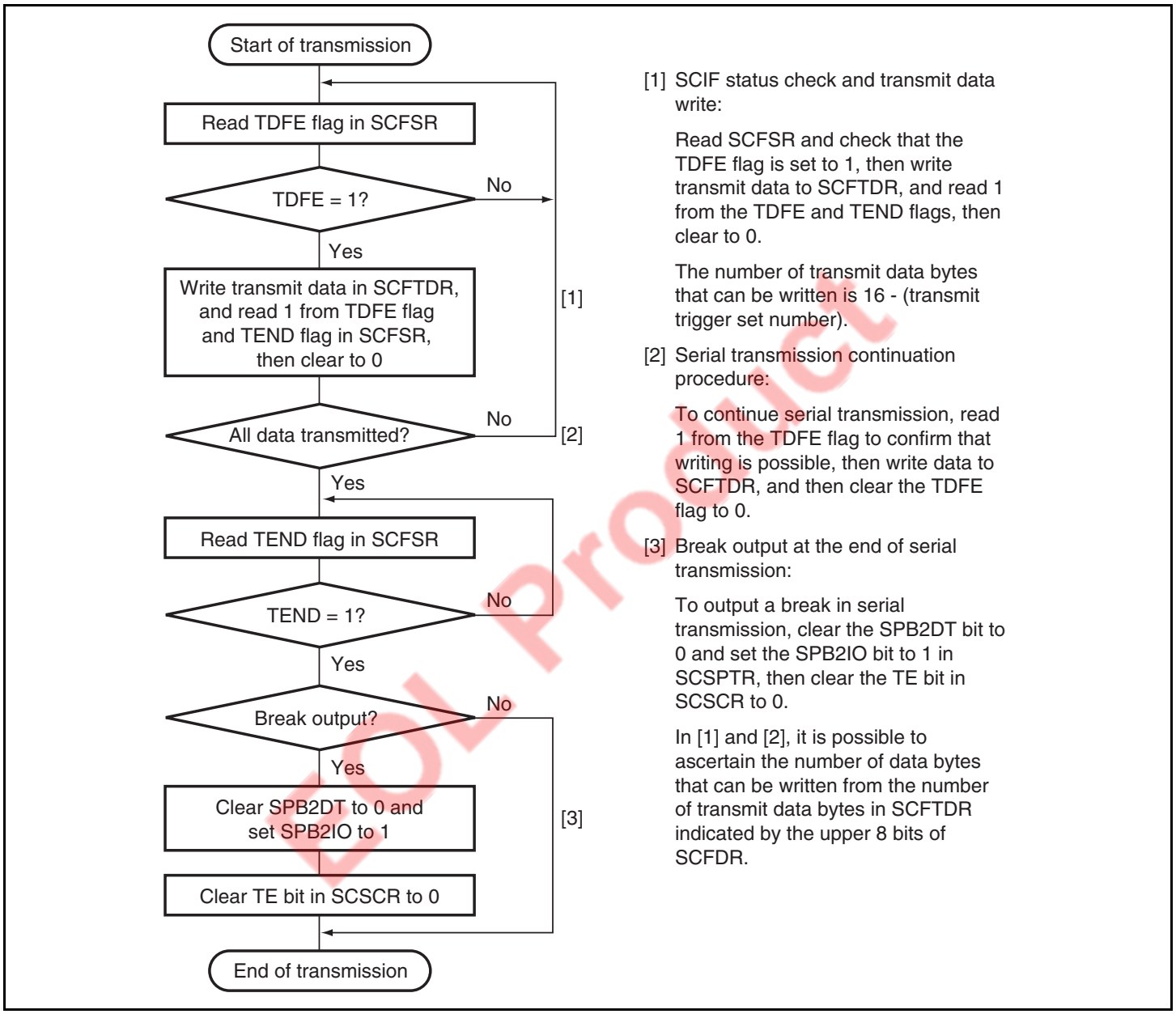


Figure 19.4 Sample Flowchart for Transmitting Serial Data

In serial transmission, the SCIF operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the SCIF transfers the data from SCFTDR to the transmit shift register (SCTSR) and starts transmitting. Confirm that the TDFE flag in the serial status register (SCFSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (16 – transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the TxD pin in the following order.

   A. Start bit: One-bit 0 is output.
   B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
   C. Parity bit: One parity bit (even or odd parity) is output. (A format in which a parity bit is not output can also be selected.)
   D. Stop bit(s): One or two 1 bits (stop bits) are output.
   E. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started. If there is no transmit data, the TEND flag in SCFSR is set to 1, the stop bit is sent, and then the line goes to the mark state in which 1 is output continuously.

Figure 19.5 shows an example of the operation for transmission.

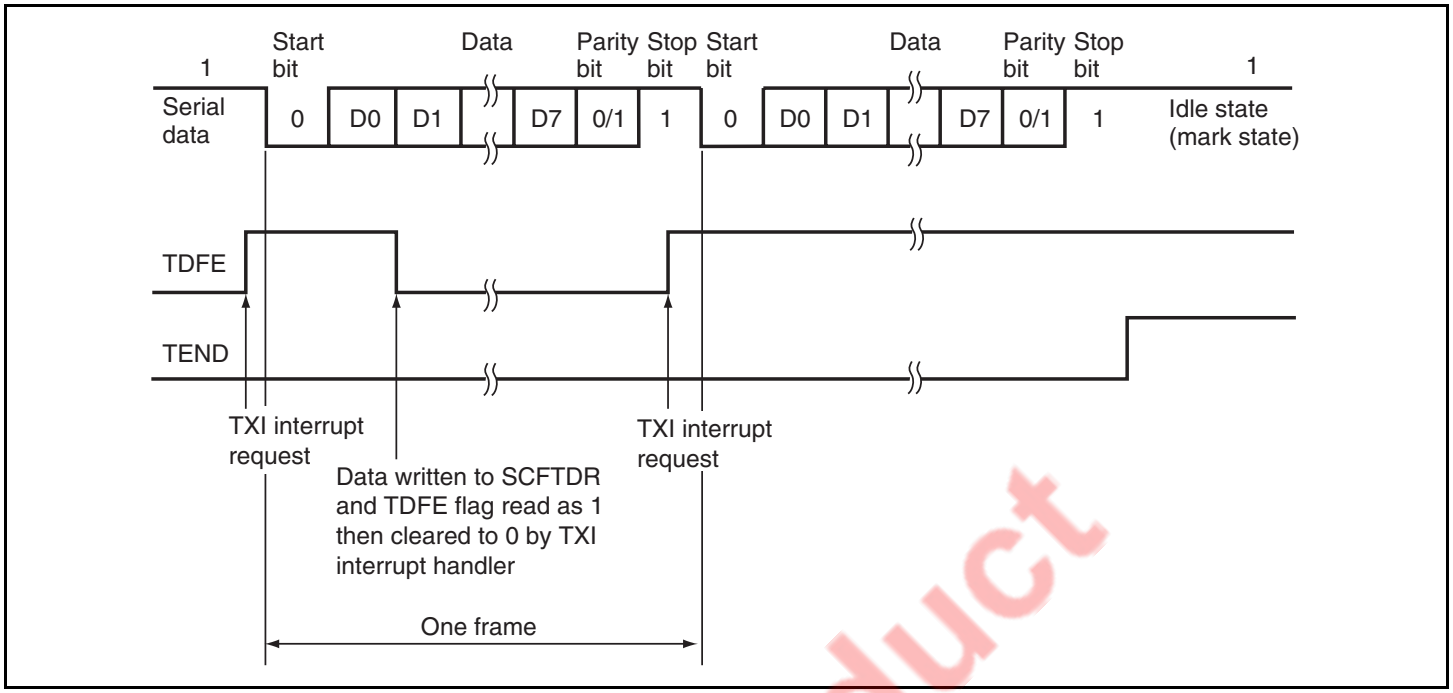


Figure 19.5 Example of Transmit Operation (8-Bit Data, Parity, One Stop Bit)

4. When modem control is enabled, transmission can be stopped and restarted in accordance with the $\overline{\text{CTS}}$ input value. When $\overline{\text{CTS}}$ is set to 1, if transmission is in progress, the line goes to the mark state after transmission of one frame. When $\overline{\text{CTS}}$ is set to 0, the next transmit data is output starting from the start bit.

Figure 19.6 shows an example of the operation when modem control is used.

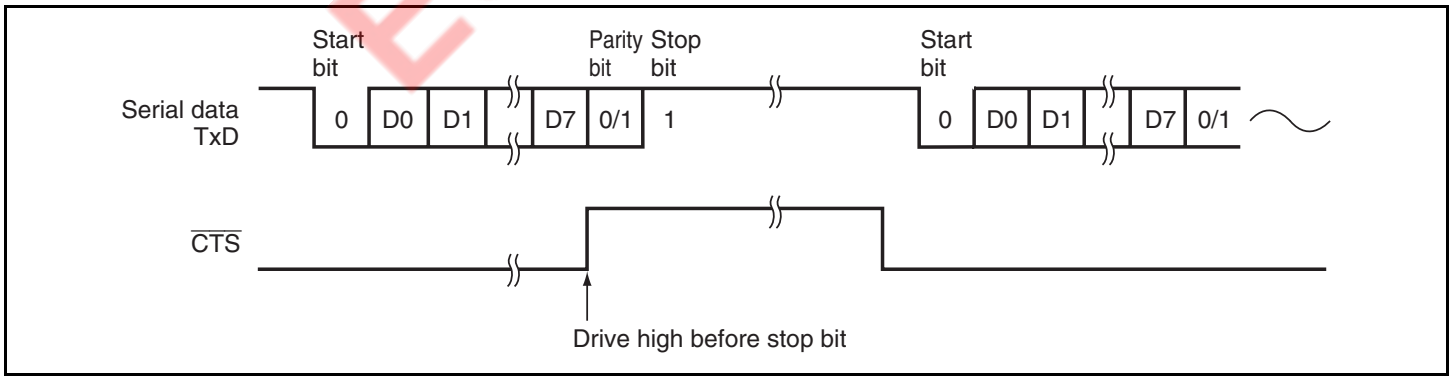


Figure 19.6 Example of Operation Using Modem Control ($\overline{\text{CTS}}$)

- **Receiving Serial Data (Asynchronous Mode)**

Figures 19.7 and 19.8 show a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

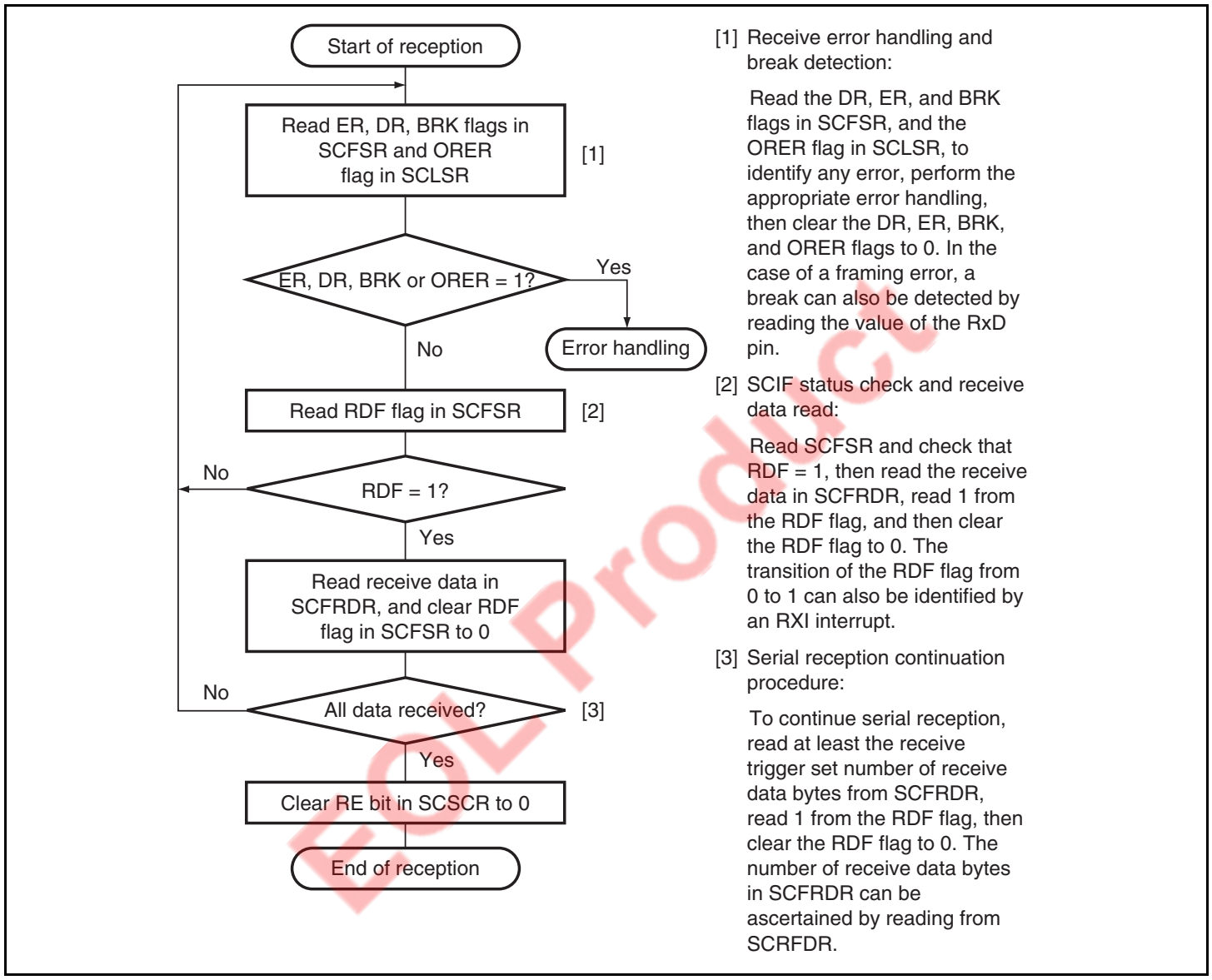
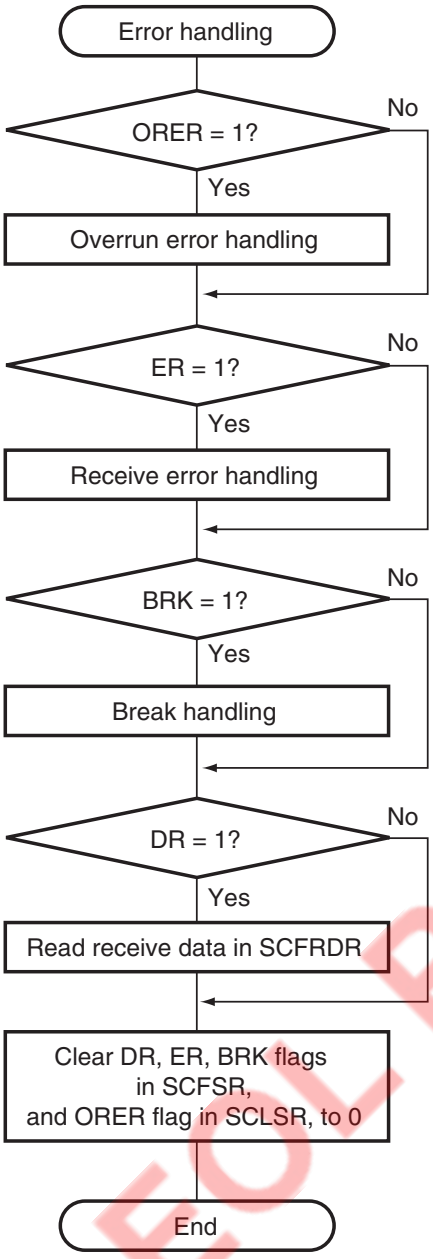


Figure 19.7 Sample Flowchart for Receiving Serial Data



[1] Whether a framing error or parity error has occurred in the receive data that is to be read from SCFRDR can be ascertained from the FER and PER bits in SCFSR.

[2] When a break signal is received, receive data is not transferred to SCFRDR while the BRK flag is set. However, note that the last data in SCFRDR is H'00, and the break data in which a framing error occurred is stored.

Figure 19.8 Sample Flowchart for Receiving Serial Data (cont)

In serial reception, the SCIF operates as described below.

1. The SCIF monitors the transmission line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR in LSB-to-MSB order.
3. The parity bit and stop bit are received.
   After receiving these bits, the SCIF carries out the following checks.
   A. Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
   B. The SCIF checks whether receive data can be transferred from the receive shift register (SCRSR) to SCFRDR.
   C. Overrun check: The SCIF checks that the ORER flag is 0, indicating that the overrun error has not occurred.
   D. Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not set.

   If all the above checks are passed, the receive data is stored in SCFRDR.

   Note: When a parity error or a framing error occurs, reception is not suspended.
4. If the RIE bit in SCSCR is set to 1 when the RDF or DR flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated. If the RIE bit or the REIE bit in SCSCR is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated. If the RIE bit or the REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes to 1, a break reception interrupt (BRI) request is generated.

Figure 19.9 shows an example of the operation for reception.

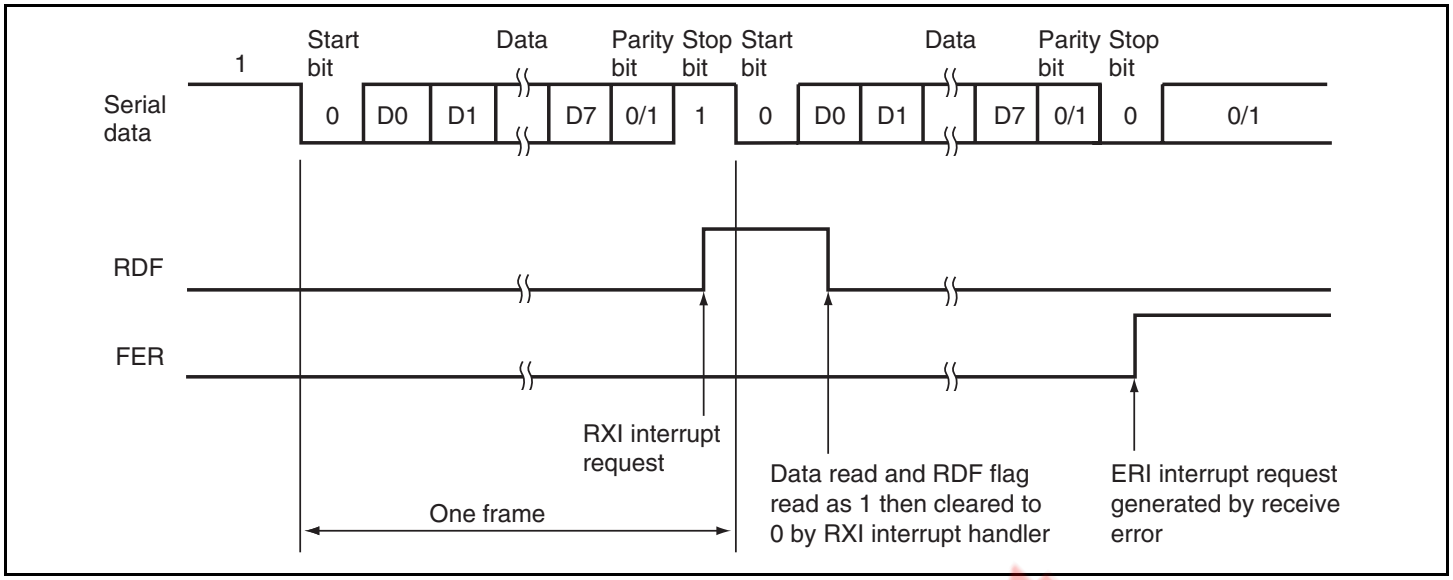


Figure 19.9 Example of SCIF Receive Operation (8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{RTS}}$ signal is output depending on the empty status of SCFRDR. When $\overline{\text{RTS}}$ is 0, reception is possible. When $\overline{\text{RTS}}$ is 1, this indicates that SCFRDR exceeds the number set for the RTS output active trigger.

Figure 19.10 shows an example of the operation when modem control is used.

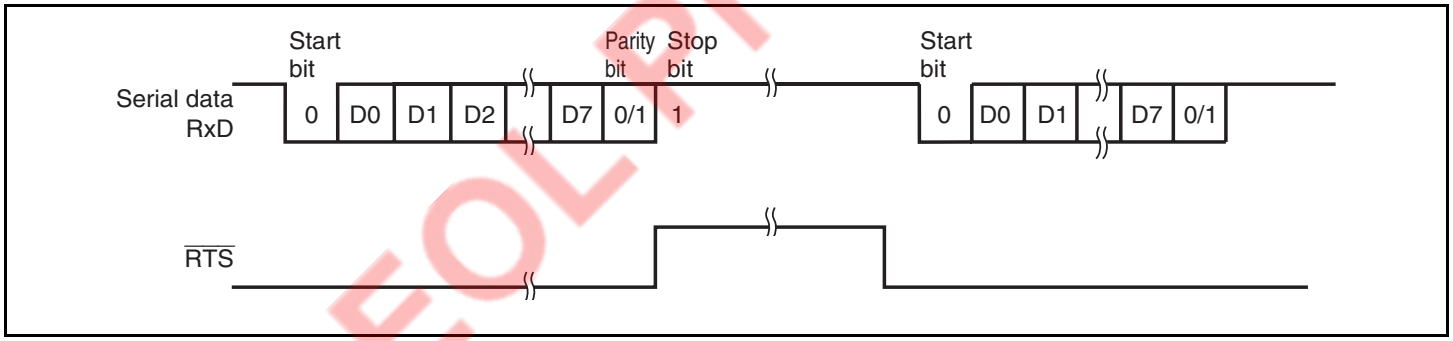


Figure 19.10 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

19.4.3 Synchronous Operation

In synchronous mode, the SCIF transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCIF transmitter and receiver are independent, so full-duplex communication is possible while sharing the same clock. The transmitter and receiver are also 16-byte FIFO buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 19.11 shows the general format in synchronous serial communication.

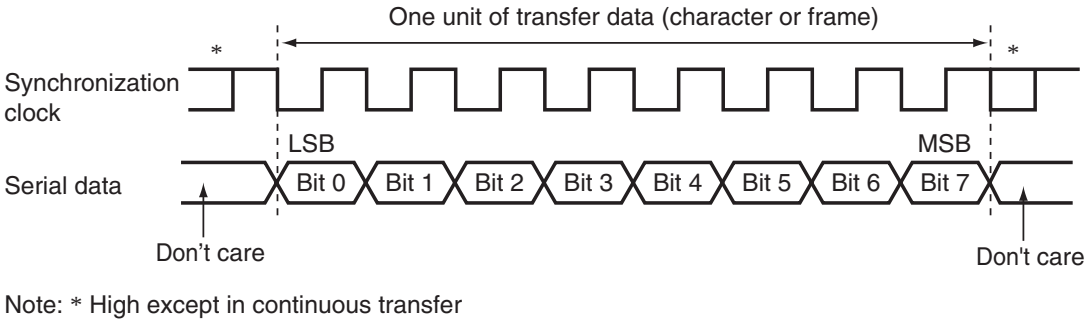


Figure 19.11 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In synchronous mode, the SCIF transmits or receives data by synchronizing with the rising edge of the serial clock.

Transmit/Receive Formats: The data length is fixed at eight bits. No parity bit can be added.

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCIF transmit/receive clock.

When the SCIF operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCIF is not transmitting or receiving, the clock signal remains in the high state. When only receiving, the clock signal outputs while the RE bit of SCSCR is 1 and the number of data in receive FIFO is less than the receive FIFO data trigger number.

Transmitting and Receiving Data:

- **SCIF Initialization (Synchronous Mode)**

Before transmitting, receiving, or changing the mode or communication format, the software must clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF. Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RDF, PER, FER, and ORER flags and receive data register (SCRDR), which retain their previous contents.

Figure 19.12 shows a sample flowchart for initializing the SCIF.

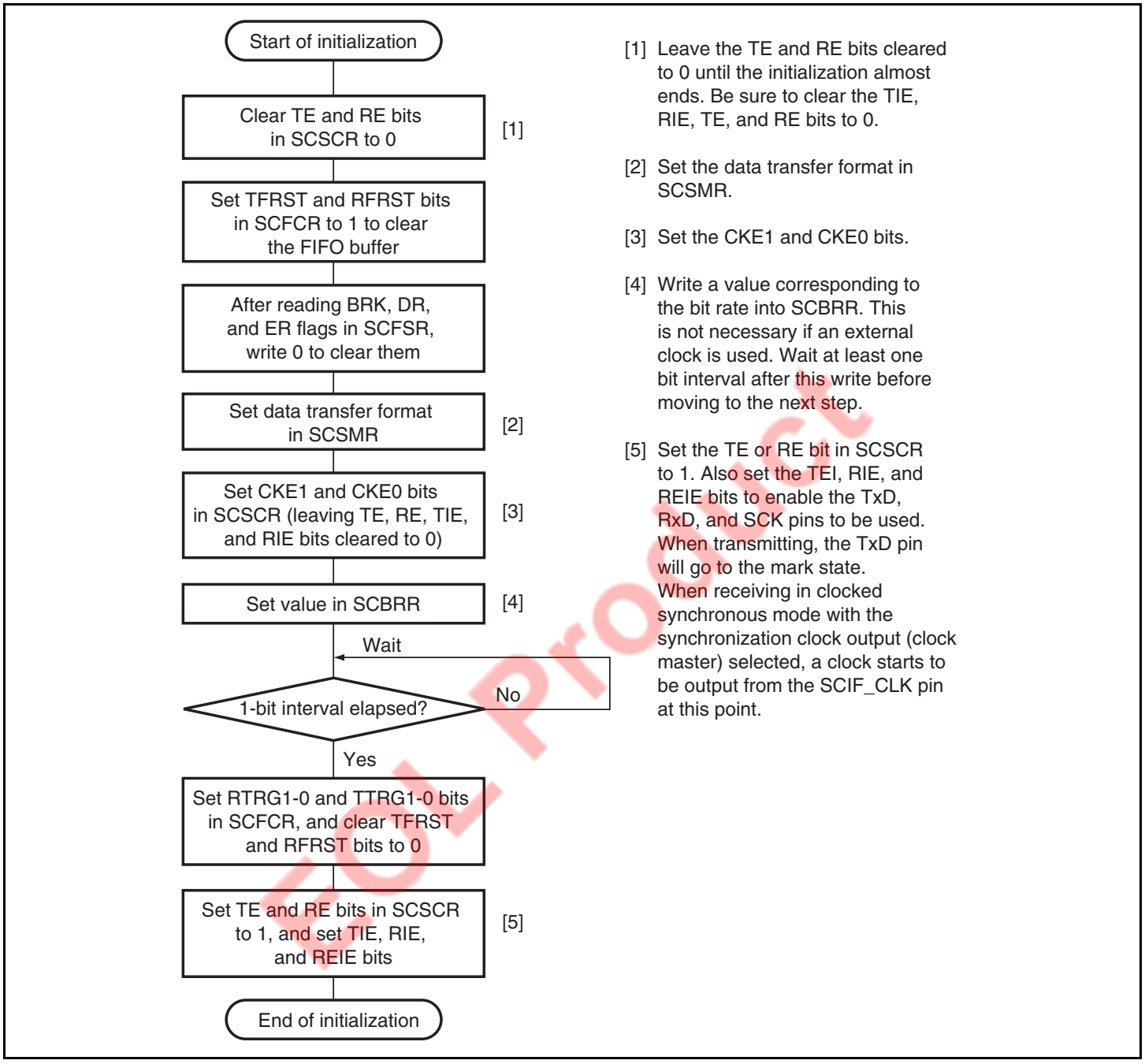


Figure 19.12 Sample Flowchart for SCIF Initialization

- **Transmitting Serial Data (Synchronous Mode)**

Figure 19.13 shows a sample flowchart for transmitting serial data.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

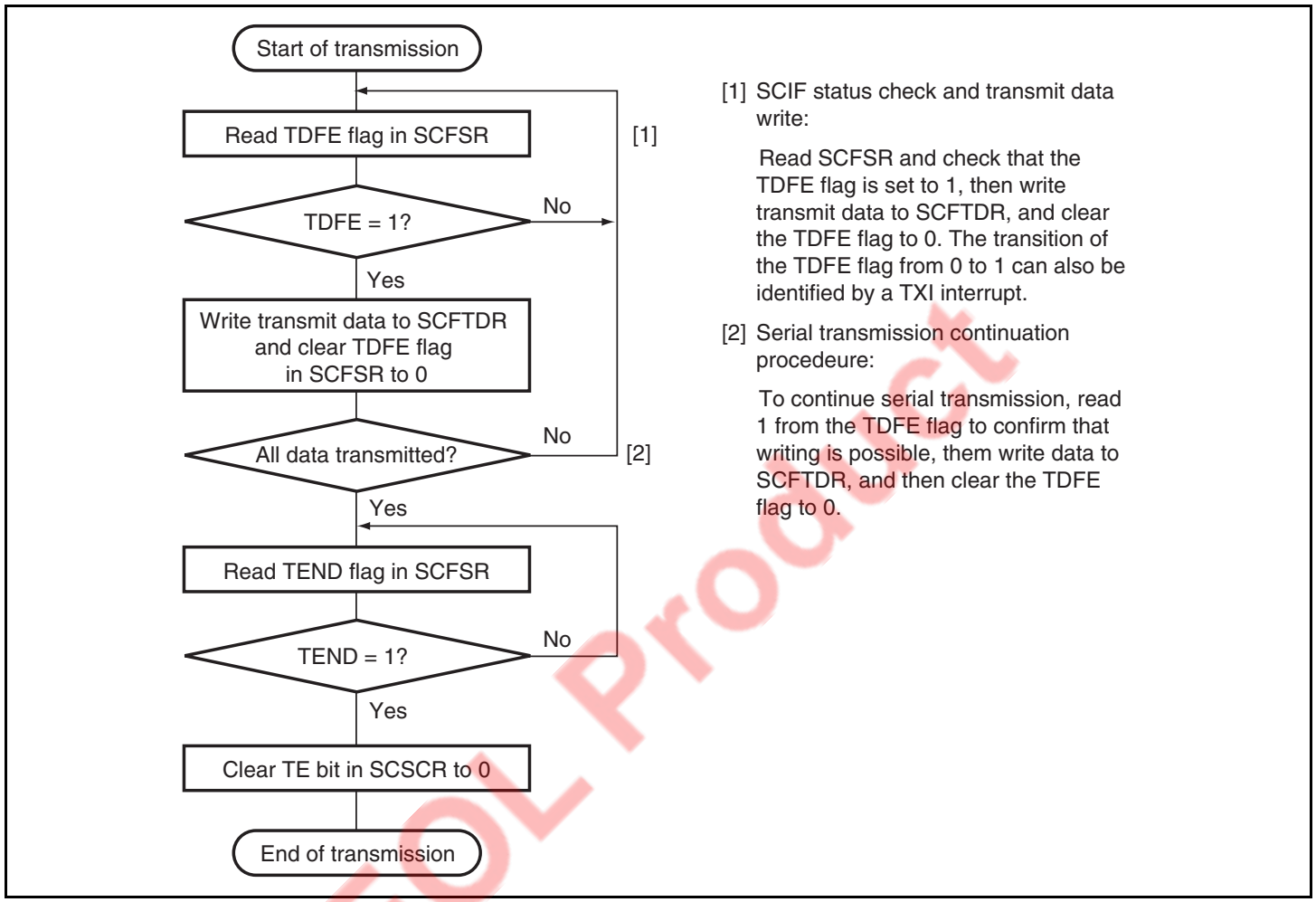


Figure 19.13 Sample Flowchart for Transmitting Serial Data

In serial transmission, the SCIF operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the SCIF transfers the data from SCFTDR to the transmit shift register (SCTSR) and starts transmitting. Confirm that the TDFE flag in the serial status register (SCFSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (16 – transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   If clock output mode is selected, the SCIF outputs eight synchronous clock pulses. If an external clock source is selected, the SCIF outputs data in synchronization with the input clock. Data is output from the TxD pin in order from the LSB (bit 0) to the MSB (bit 7).
3. The SCIF checks the SCFTDR transmit data at the timing for sending the MSB (bit 7). If data is present, the data is transferred from SCFTDR to SCTSR, the MSB (bit 7) is sent, and then serial transmission of the next frame is started. If there is no transmit data, the TEND flag in SCFSR is set to 1, the MSB (bit 7) is sent, and then the TxD pin holds the states.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 19.14 shows an example of SCIF transmit operation.

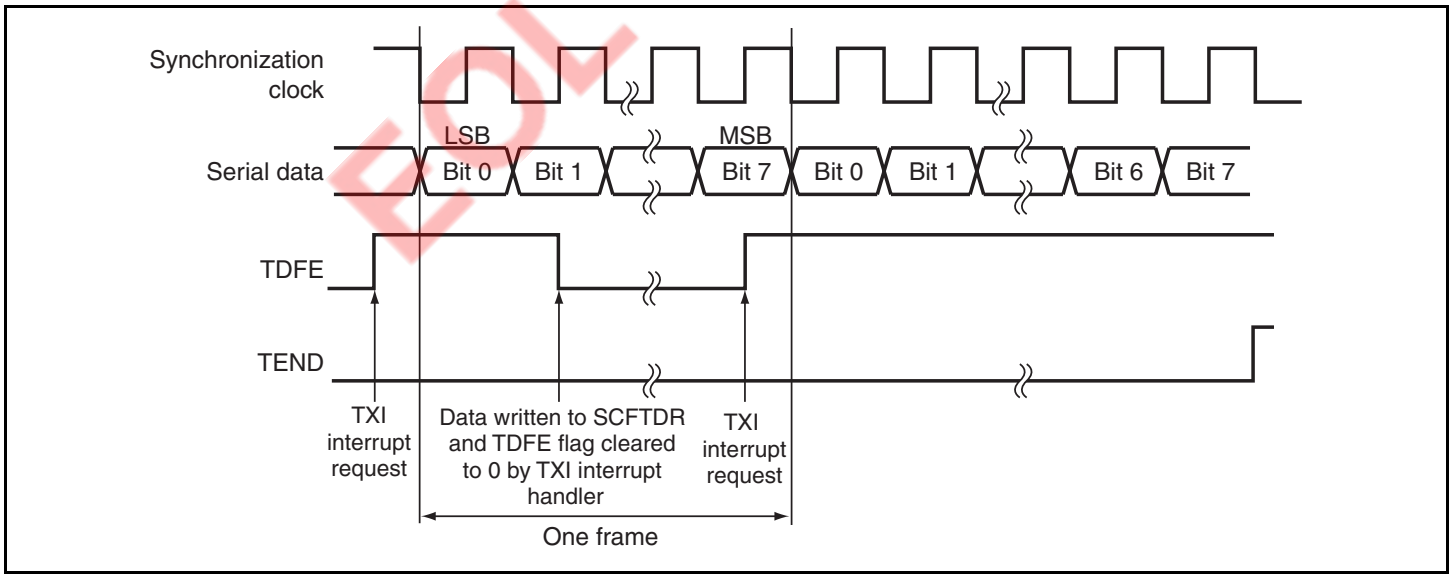


Figure 19.14 Example of SCIF Transmit Operation

- **Receiving Serial Data (Synchronous Mode)**

Figure 19.15 shows a sample flowchart for receiving serial data. When switching from asynchronous mode to synchronous mode without SCIF initialization, make sure that ORER, PER, and FER are cleared to 0.

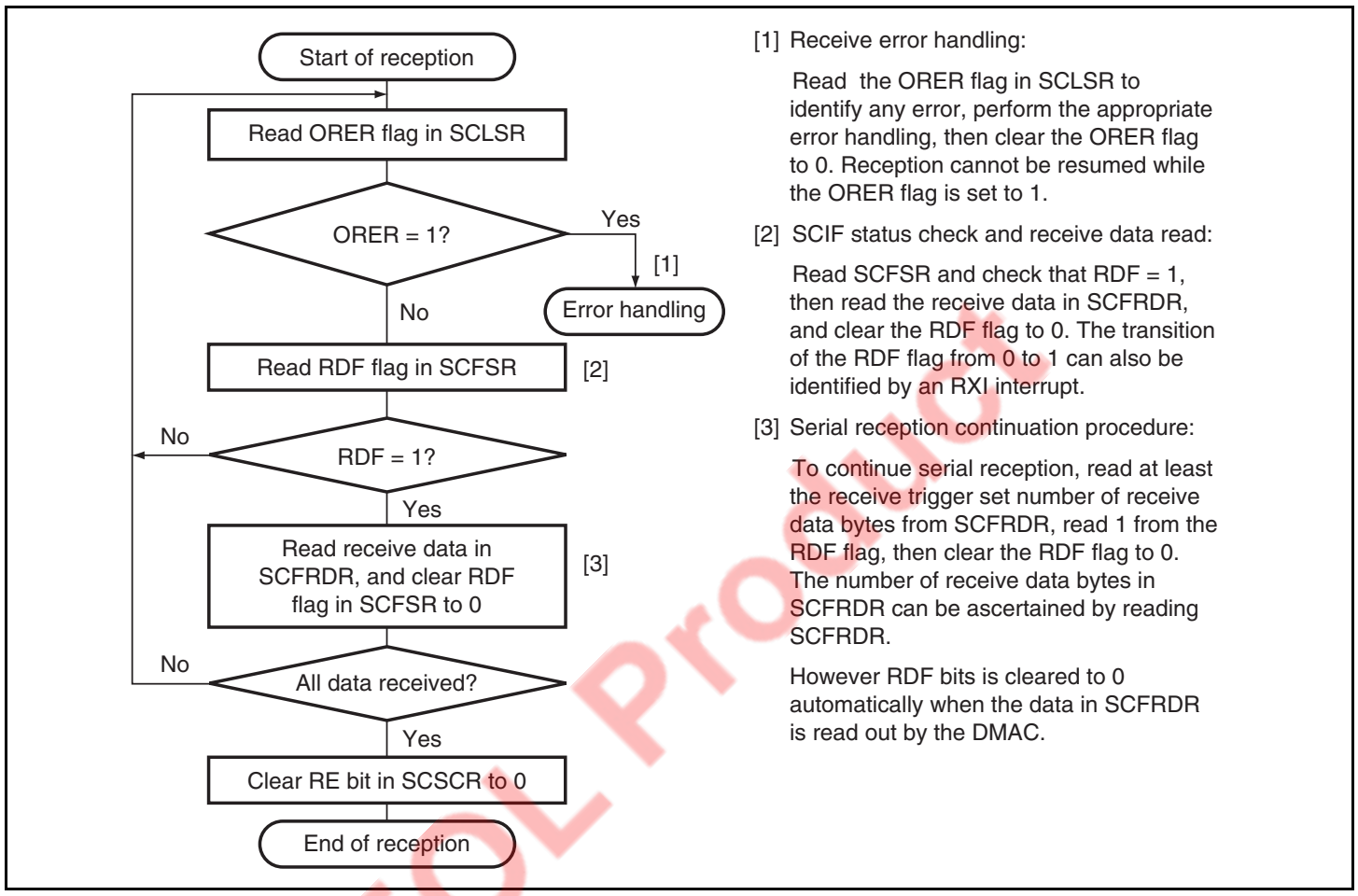


Figure 19.15 Sample Flowchart for Receiving Serial Data (1)

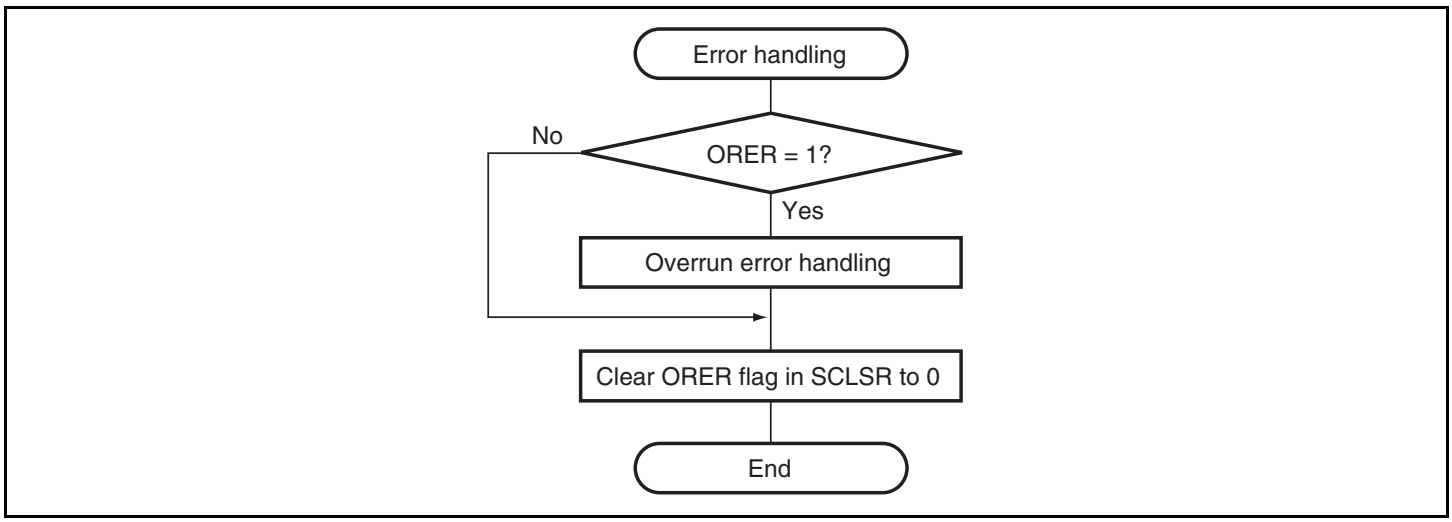


Figure 19.16 Sample Flowchart for Receiving Serial Data (2)

In serial reception, the SCIF operates as described below.

1. The SCIF synchronizes with serial clock input or output and starts the reception.
2. Receive data is shifted into SCRSR in order from the LSB to the MSB. After receiving the data, the SCIF checks the receive data can be loaded from SCRSR into SCFRDR or not. If this check is passed, the SCIF stores the received data in SCFRDR. If the check is not passed (overrun error is detected), further reception is prevented.
3. After setting RDF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in SCSCR, the SCIF requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the receive-data-full interrupt enable bit (RIE) or the receive error interrupt enable bit (REIE) in SCSCR is also set to 1, the SCIF requests a break interrupt (BRI).

Figure 19.17 shows an example of SCIF receive operation.

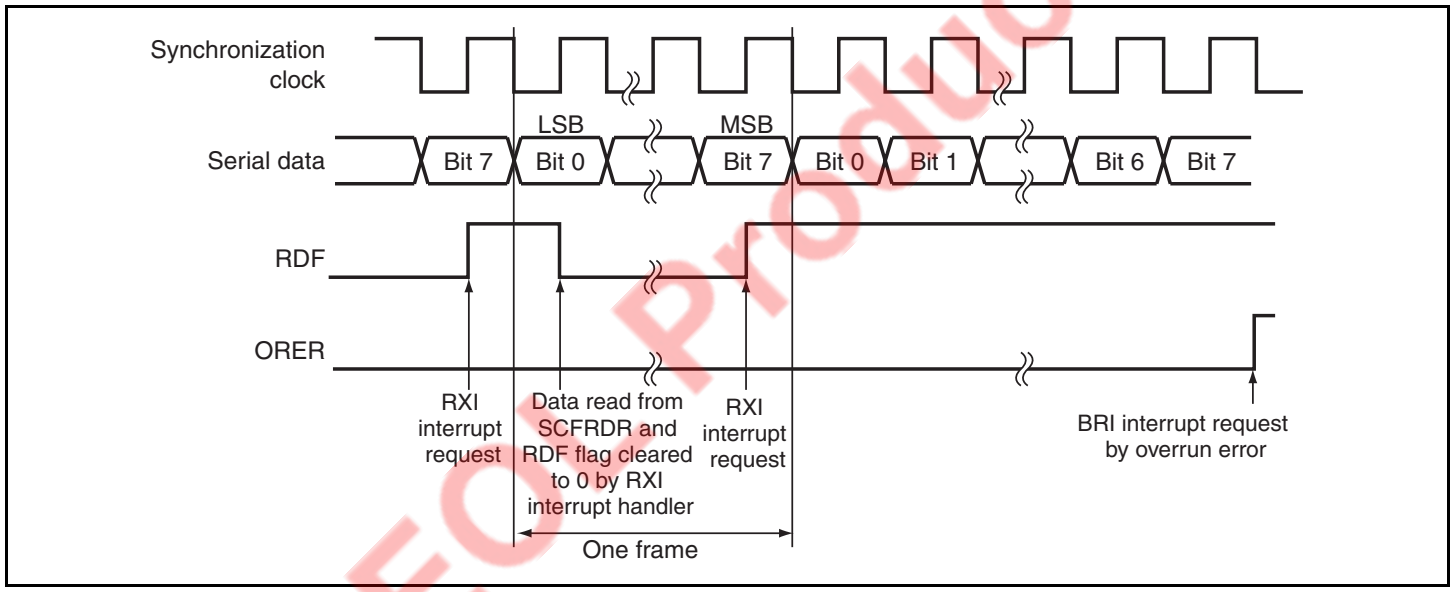


Figure 19.17 Example of SCIF Receive Operation

- **Transmitting and Receiving Serial Data Simultaneously (Synchronous Mode)**

Figure 19.18 shows a sample flowchart for transmitting and receiving serial data simultaneously.

Use the following procedure for the simultaneous transmission/reception of serial data, after enabling the SCIF for transmission/reception.

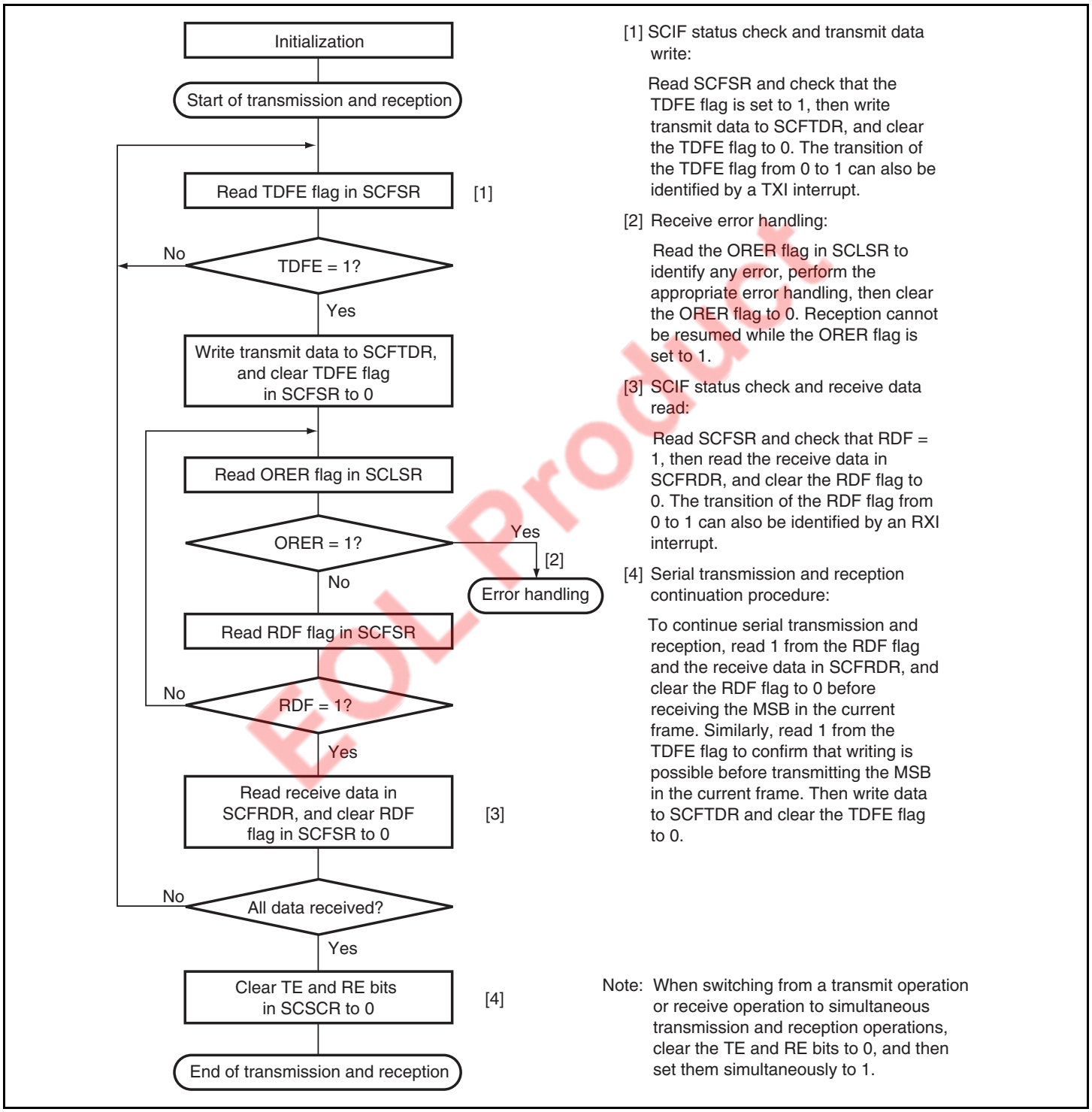


Figure 19.18 Sample Flowchart for Transmitting/Receiving Serial Data

19.5 SCIF Interrupts and DMAC

The SCIF has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive-data-full (RXI), and break (BRI).

Table 19.11 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE, RIE, and REIE bits in SCSCR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

When TXI request is enabled by TIE bit and the TDFE flag in the serial status register (SCFSR) is set to 1, a TXI interrupt request and transmit FIFO data empty DMA transfer request are generated. When TXI request is disabled by TIE bit and the TDFE flag is set to 1, transmit FIFO data empty DMA transfer request is generated. The DMAC can be activated and data transfer performed by the transmit FIFO data empty DMA transfer request.

When RXI request is enabled by RIE bit and the RDF or DR flag in SCFSR is set to 1, an RXI interrupt request and receive FIFO data full DMA transfer request are generated. When RXI request is disabled by RIE bit and the RDF or DR flag in SCFSR is set to 1, receive FIFO data full DMA transfer request is generated. The DMAC can be activated and data transfer performed by the receive FIFO data full DMA transfer request. The RXI interrupt request or receive FIFO data full DMA transfer request caused by DR flag is generated only in asynchronous mode.

When the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1, a BRI interrupt request is generated.

When the ER flag in SCFSR is set to 1, an ERI interrupt request is generated.

When transmitting or receiving data are transferred by DMAC, DMAC should be set enable at first, and then SCIF should be set enable. SCIF should be set not to request RXI or TXI interrupt to INTC. If SCIF is set to request the interrupt, DMA transfer clears the request to INTC independently of interrupt handling program.

When the RIE bit is set to 0 and the REIE bit is set to 1, SCIF request ERI interrupt and BRI interrupt without requesting RXI interrupt.

The TXI interrupt indicates that transmit data can be written, and the RXI interrupt indicates that there is receive data in SCFRDR.

Table 19.11 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
ERI	Interrupt initiated by receive error (ER)	Not possible	High
RXI	Interrupt initiated by receive data FIFO full (RDF) or data ready (DR)*	Possible	
BRI	Interrupt initiated by break (BRK) or overrun error (ORER)	Not possible	
TXI	Interrupt initiated by transmit FIFO data empty (TDFE)	Possible	Low

Note: * RXI interrupt by DR is only possible in the asynchronous mode.

19.6 Usage Notes

Note the following when using the SCIF.

1. SCFTDR Writing and TDFE Flag

   The TDFE flag in the serial status register (SCFSR) is set when the number of transmit data bytes written in the transmit FIFO data register (SCFTDR) has fallen below the transmit trigger number set by bits TTRG1 and TTRG0 in the FIFO control register (SCFCR). After TDFE is set, transmit data up to the number of empty bytes in SCFTDR can be written, allowing efficient continuous transmission.

   However, if the number of data bytes written in SCFTDR is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again after being read as 1 and cleared to 0. TDFE clearing should therefore be carried out when SCFTDR contains more than the transmit trigger number of transmit data bytes.

   The number of transmit data bytes in SCFTDR can be found from the upper 8 bits of the FIFO data count register (SCFDR).

2. SCFRDR Reading and RDF Flag

   The RDF flag in the serial status register (SCFSR) is set when the number of receive data bytes in the receive FIFO data register (SCFRDR) has become equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in the FIFO control register (SCFCR). After RDF is set, receive data equivalent to the trigger number can be read from SCFRDR, allowing efficient continuous reception.

   However, if the number of data bytes in SCFRDR is equal to or greater than the trigger number, the RDF flag will be set to 1 again if it is cleared to 0. RDF should therefore be cleared to 0 after being read as 1 after all the receive data has been read.

The number of receive data bytes in SCFRDR can be found from the lower 8 bits of the FIFO data count register (SCFDR).

3. Break Detection and Processing

   Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. Note that, although transfer of receive data to SCFRDR is halted in the break state, the SCIF receiver continues to operate.

4. Sending a Break Signal

   The I/O condition and level of the TxD pin are determined by the SPB2IO and SPB2DT bits in the serial port register (SCSPTR). This feature can be used to send a break signal.

   Until TE bit is set to 1 (enabling transmission) after initializing, TxD pin does not work. During the period, mark status is performed by SPB2DT bit. Therefore, the SPB2IO and SPB2DT bits should be set to 1 (high level output).

   To send a break signal during serial transmission, clear the SPB2DT bit to 0 (designating low level), then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, and 0 is output from the TxD pin.

5. Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

   The SCIF operates on a base clock with a frequency of 16 times the transfer rate. In reception, the SCIF synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. The timing is shown in figure 19.19.

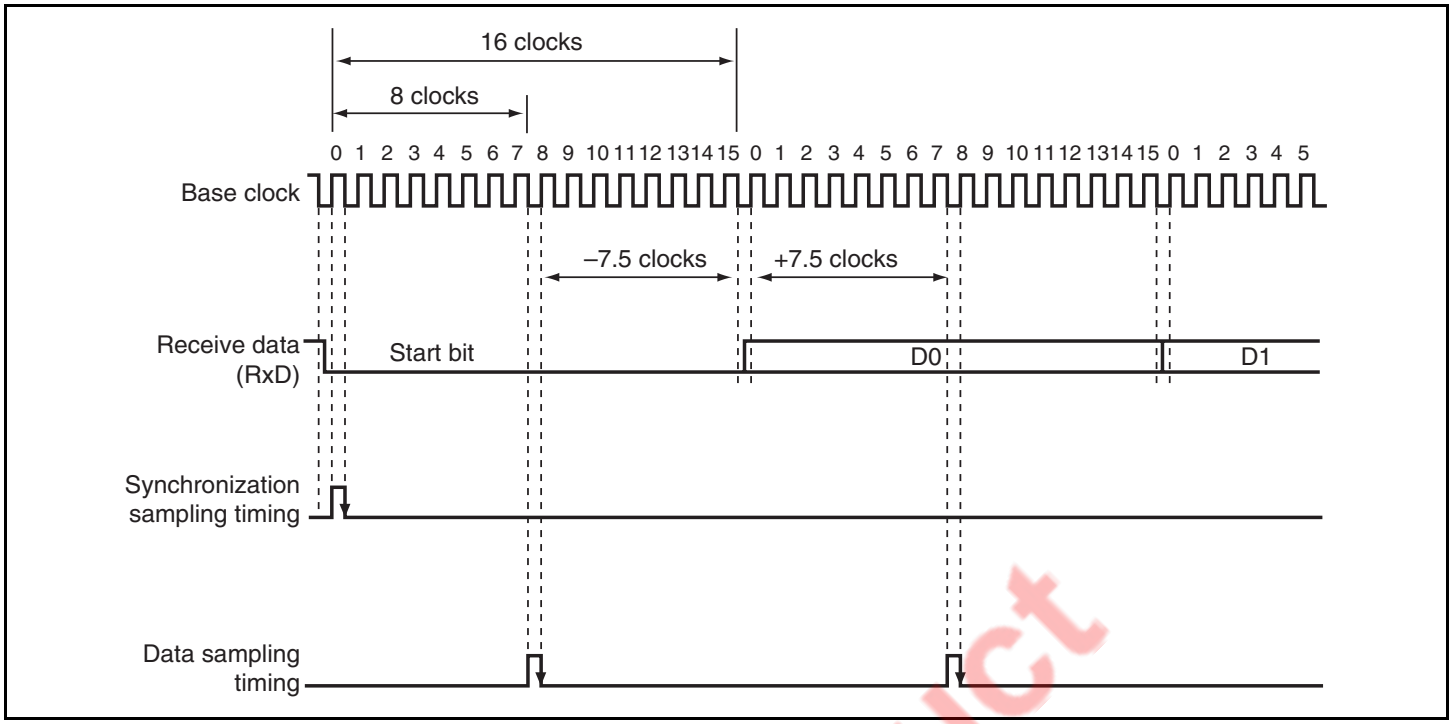


Figure 19.19 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

Equation 1:

$$M = \left| (0.5 - \frac{1}{2N}) = (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1+F) \right| \times 100\ \%$$

Where: M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation 2.

Equation 2:

When D = 0.5 and F = 0:

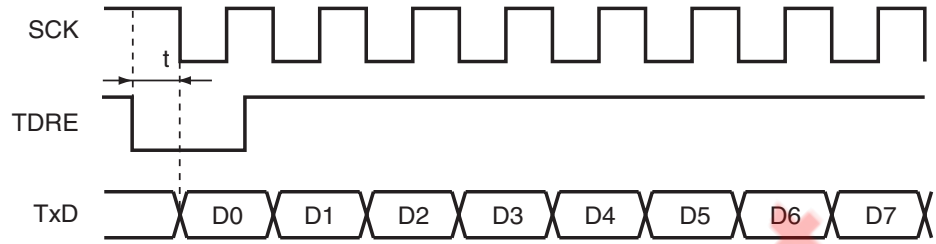
$M = (0.5 - 1/(2 \times 16)) \times 100\%$
$= 46.875\%$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

6. When Using the DMAC

— Using an External Clock in Chock Synchronous Mode:

When using an external clock as the synchronization clock, after SCFTDR is updated by the DMAC, an external clock should be input after at least five peripheral clock cycles. A malfunction may occur when the transfer clock is input within four cycles after updating SCFTDR (figure 19.20).



Note: When the SCIF is operated on an external clock, set t > 4.

Figure 19.20 DMA Transfer Example in the Synchronization Clock

— DMA Transfer Request:

When a DMA transfer is requested from the SCIF of which transfer request is allowed by the DMAC, the transfer request from the SCIF is held in the DMAC. This transfer request is cleared after it is actually transferred.

Even if the DME bit of the DMA operation register (DMAOR) and the DE bit of the DMA channel control register (CHCR) are cleared, the DMA transfer request from the SCIF is retained. In this state, note that the DMA transfer is done for one time without any DMA transfer request from the SCIF when the DMAC allows the transfer request from the SCIF.

— TEND Flag:

When the transmit FIFO data empty DMA transfer request is generated and the transmit data is written to SCFTDR by the DMAC, the value indicated by the TEND flag is undefined. Thus, do not use the TEND flag as a transmit end flag.

Section 20 USB Function Module

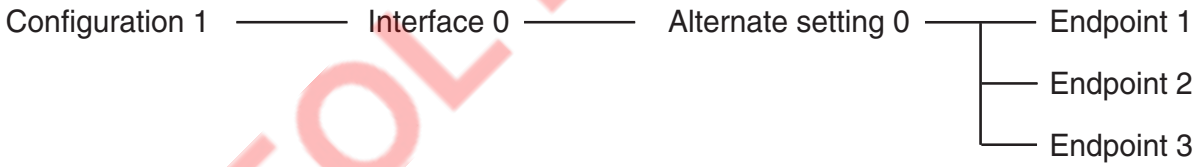
20.1 Features

- Incorporates UDC (USB device controller) conforming to the USB standard

  Automatic processing of USB protocol

  Automatic processing of USB standard commands for endpoint 0 (some commands and class/vendor commands require decoding and processing by firmware)
- Transfer speed: Full-speed
- 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	8	8	—
	EP0o	Control OUT	8	8	—
Endpoint 1	EP1	Bulk OUT	64	128	Possible
Endpoint 2	EP2	Bulk IN	64	128	Possible
Endpoint 3	EP3	Interrupt	8	8	—



- Interrupt requests: generates various interrupt signals necessary for USB transmission/reception
- Clock: External input (48 MHz)
- Power-down mode

  Power consumption can be reduced by stopping UDC internal clock when USB cable is disconnected

  Automatic transition to/recovery from suspend state
- In on-chip transceiver bypass mode (the XVEROFF bit of USBXVERCR resister is 1), a Philips PDIUSBP11 Series transceiver or compatible product can be connected (when using a compatible product, carry out evaluation and investigation with the manufacturer supplying the transceiver beforehand)

- Power mode: Self-powered, bus-powered

20.1.1 Block Diagram

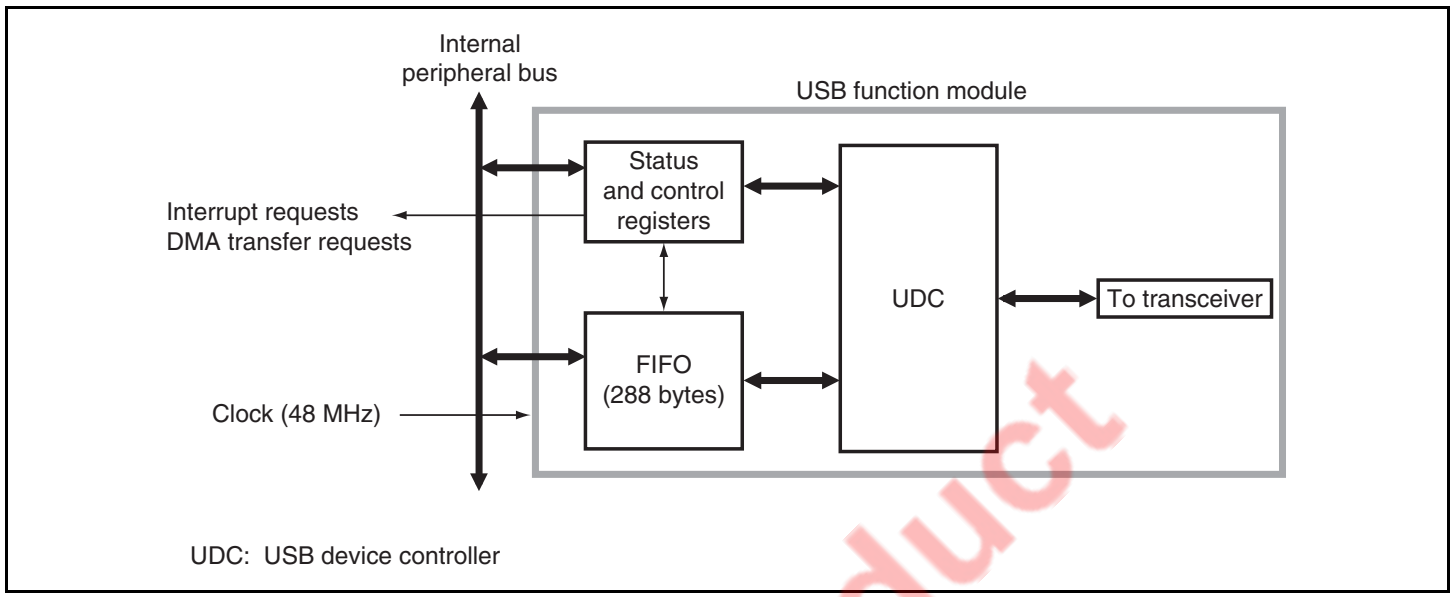


Figure 20.1 Block Diagram of USB

20.2 Pin Configuration

Table 20.1 Pin Configuration and Functions

Pin Name	I/O	Function	XVEROFF Conditions
XVDATA	Input	Input pin for receive data from differential receiver	1
DPLS	Input	Input pin to driver for D+ signal from receiver	1
DMNS	Input	Input pin to driver for D– signal from receiver	1
TXDPLS	Output	D+ transmit output pin to driver	1
TXDMNS	Output	D– transmit output pin to driver	1
TXENL	Output	Driver output enable pin	1
VBUS	Input	USB cable connection monitor pin	1/0
SUSPND	Output	Transceiver suspend state output pin	1/0
UCLK	Input	USB clock input pin (48 MHz input)	1/0
DP	I/O	On-chip transceiver D + signal	0
DM	I/O	On-chip transceiver D - signal	0

In on-chip transceiver bypass mode (the XVEROFF bit of the USBXVERCR register is 1), a Philips PDIUSBP11 Series transceiver or compatible product can be connected (when using a compatible product, carry out evaluation and investigation with the manufacturer supplying the transceiver beforehand).

20.3 Register Descriptions

The USB has the following registers.

- USB interrupt flag register 0 (USBIFR0)
- USB interrupt flag register 1 (USBIFR1)
- USB interrupt flag register 2 (USBIFR2)
- USB interrupt select register 0 (USBISR0)
- USB interrupt select register 1 (USBISR1)
- USB interrupt enable register 0 (USBIER0)
- USB interrupt enable register 1 (USBIER1)
- USB interrupt enable register 2 (USBIER2)
- USBEP0i data register (USBEPDR0i)
- USBEP0o data register (USBEPDR0o)
- USBEP0s data register (USBEPDR0s)
- USBEP1 data register (USBEPDR1)
- USBEP2 data register (USBEPDR2)
- USBEP3 data register (USBEPDR3)
- USBEP0o receive data size register (USBEPSZ0o)
- USBEP1 receive data size register (USBEPSZ1)
- USB trigger register (USBTRG)
- USB data status register (USBDASTS)
- USB FIFO clear register (USBFCLR)
- USB DMA transfer setting register (USBDMAR)
- USB endpoint stall register (USBEPSTL)
- USB transceiver control register (USBXVERCR)
- USB bus power control register (USBCTRL)

20.3.1 USB Interrupt Flag Register 0 (USBIFR0)

Together with USB interrupt flag registers 1 (USBIFR1) and 2 (USBIFR2), USBIFR0 indicates interrupt status information required by the application. When an interrupt occurs, the corresponding bit is set to 1 and an interrupt request is sent to the CPU according to the combination with USB interrupt enable register 0 (USBIER0). Clearing is performed by writing 0 to the bit to be cleared, and 1 to the other bits. However, EP1 FULL and EP2 EMPTY are status bits, and cannot be cleared.

USBIFR0 is initialized to H'10 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7	BRST	0	R/W	Bus Reset Set to 1 when the bus reset signal is detected on the USB bus.
6	EP1FULL	0	R	EP1 FIFO Full This bit is set when endpoint 1 receives one packet of data normally from the host, and holds a value of 1 as long as there is valid data in the FIFO buffer. EP1 FULL is a status bit, and cannot be cleared.
5	EP2TR	0	R/W	EP2 Transfer Request This bit is set if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 2 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
4	EP2EMPTY	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. EP2 EMPTY is a status bit, and cannot be cleared.
3	SETUPTS	0	R/W	Setup Command Receive Complete This bit is set to 1 when endpoint 0 receives normally a setup command requiring decoding on the application side, and returns an ACK handshake to the host.

Bit	Bit Name	Initial Value	R/W	Description
2	EP0oTS	0	R/W	EP0o Receive Complete This bit is set to 1 when endpoint 0 receives data from the host normally, stores the data in the FIFO buffer, and returns an ACK handshake to the host.
1	EP0iTR	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 NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
0	EP0iTS	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.

20.3.2 USB Interrupt Flag Register 1 (USBIFR1)

Together with USB interrupt flag registers 0 (USBIFR0) and 2 (USBIFR2), USBIFR1 indicates interrupt status information required by the application. When an interrupt occurs, the corresponding bit is set to 1 and an interrupt request is sent to the CPU according to the combination with USB interrupt enable register 1 (USBIER1). Clearing is performed by writing 0 to the bit to be cleared, and 1 to the other bits. However, VBUSMN is a status bit, and cannot be cleared.

USBIFR1 is initialized to H'20 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved The write value should always be 0.
3	VBUSMN	0	R	Status bit for monitoring the status of the VBUS pin. The status of the VBUS pin is reflected. 0: Disconnected 1: Connected

Bit	Bit Name	Initial Value	R/W	Description
2	EP3TR	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 NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
1	EP3TS	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.
0	VBUS	0	R/W	UBS Disconnection Detection This bit is set to 1 when a function is connected to or disconnected from the USB bus. Use the VBUSCNT pin of this module to detect connection/disconnection.

20.3.3 USB Interrupt Flag Register 2 (USBIFR2)

Together with USB interrupt flag registers 0 (USBIFR0) and 1 (USBIFR1), USBIFR2 indicates interrupt status information required by the application. When an interrupt occurs, the corresponding bit is set to 1 and an interrupt request is sent to the CPU according to the combination with USB interrupt enable register 2 (USBIER2). Clearing is performed by writing 0 to the bit to be cleared, and 1 to the other bits. However, CFGV is a status bit, and cannot be cleared. USBIFR2 is initialized to H'20 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	The write value should always be 0.
5	—	1	R	
4	—	0	R	
3	AWAKE	0	R/W	Awake Signal Detection This bit is set to 1 when the resume or bus reset signal is detected on the USB bus in the suspend state with USBCTRL/SUSPEND = 1.
2	SUSPS	0	R/W	USB Suspend Signal Detection This bit is set to 1 when the USB suspend signal is detected with USBCTRL/SUSPEND = 1.

Bit	Bit Name	Initial Value	R/W	Description
1	CFGV	0	R	Configuration Value Status bit for monitoring the configuration value. This is a status bit and cannot be cleared.
0	SETC	0	R/W	SET_CONFIGURATION Request Detection This bit is set to 1 when the SET_CONFIGURATION request is received.

20.3.4 USB Interrupt Select Register 0 (USBISR0)

USBISR0 selects the vector numbers of the interrupt requests indicated in USB interrupt flag register 0 (USBIFR0). If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR0 is cleared to 0, the interrupt will be USI0 (USB interrupt 0). If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR0 is set to 1, the interrupt will be USI1 (USB interrupt 1). If interrupts occur simultaneously, USI0 has priority by default.

USBISR0 is initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7	BRST	0	R/W	Bus reset
6	EP1FULL	0	R/W	EP1FIFO full
5	EP2TR	0	R/W	EP2 transfer request
4	EP2EMPTY	0	R/W	EP2 FIFO empty
3	SETUPTS	0	R/W	Setup command receive completion
2	EP0oTS	0	R/W	EPOo receive completion
1	EP0iTR	0	R/W	EPOi transfer request
0	EP0iTS	0	R/W	EPOi transmit completion

20.3.5 USB Interrupt Select Register 1 (USBISR1)

USBISR1 selects the vector numbers of the interrupt requests indicated in USB interrupt flag register 1 (USBIFR1). If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR1 is cleared to 0, the interrupt will be USI0 (USB interrupt 0). If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR1 is set to 1, the interrupt will be USI1 (USB interrupt 1). If interrupts occur simultaneously, USI0 has priority by default.

USBISR1 is initialized to H'07 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved The write value should always be 0.
2	EP3TR	0	R/W	EP3 transfer request
1	EP3TS	0	R/W	EP3 transmission completion
0	VBUSF	0	R/W	USB bus connection

20.3.6 USB Interrupt Enable Register 0 (USBIER0)

USBIER0 enables the interrupt requests indicated in USB interrupt flag register 0 (USBIFR0). When an interrupt flag is set while the corresponding bit in USBIER0 is set to 1, an interrupt request is sent to the CPU. The interrupt vector number is decided by the contents of USB interrupt select register 0 (USBISR0).

USBIER0 is initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7	BRST	0	R/W	Bus reset
6	EP1FULL	0	R/W	EP1FIFO full
5	EP2TR	0	R/W	EP2 transfer request
4	EP2EMPTY	0	R/W	EP2 FIFO empty
3	SETUPTS	0	R/W	Setup command receive completion
2	EP0oTS	0	R/W	EPOo receive completion
1	EP0iTR	0	R/W	EPOi transfer request
0	EP0iTS	0	R/W	EPOi transmit completion

20.3.7 USB Interrupt Enable Register 1 (USBIER1)

USBIER1 enables the interrupt requests indicated in USB interrupt flag register 1 (USBIFR1). When an interrupt flag is set while the corresponding bit in USBIER1 is set to 1, an interrupt request is sent to the CPU. The interrupt vector number is decided by the contents of USB interrupt select register 1 (USBISR1).

USBIER1 is initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved The write value should always be 0.
2	EP3TR	0	R/W	EP3 transfer request
1	EP3TS	0	R/W	EP3 transmit completion
0	VBUS	0	R/W	USB bus connection

20.3.8 USB Interrupt Enable Register 2 (USBIER2)

USBIER2 enables the interrupt requests detected by SET_CONFIGURATION in USB interrupt flag register 2 (USBIFR2). When the USBIFR2/SETC flag is set while the corresponding bit in USBIER2 is set to 1, a USIHP interrupt request is sent to the CPU.

USBIER2 is initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved The write value should always be 0.
0	SETC	0	R/W	SET_CONFIGURATION request detection

20.3.9 USBEP0i Data Register (USBEPDR0i)

USBEPDR0i is an 8-byte transmit FIFO buffer for endpoint 0, holding one packet of transmit data for control IN. Transmit data is fixed by writing one packet of data and setting the EP0iPKTE bit in the trigger register. When an ACK handshake is returned from the host after the data has been transmitted, bit 0 (EP0iTS) in USB interrupt flag register 0 is set.

USBEP0I can be initialized by means of the EP0iCLR bit in USBFCLR.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for control IN transfer

20.3.10 USBEP0o Data Register (USBEPDR0o)

USBEPDR0o is an 8-byte receive FIFO buffer for endpoint 0. USBEPDR0o holds endpoint 0 receive data other than setup commands. When data is received normally, the EP0oTS bit in USB interrupt flag register 0 is set, and the number of receive bytes is indicated in the EP0o receive data size register. After the data has been read, setting the EP0oRDFN bit in the trigger register enables the next packet to be received.

USBEPDR0o can be initialized by means of the EP0oCLR bit in USBFCLR.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	R	Data register for control OUT transfer

20.3.11 USBEP0s Data Register (USBEPDR0s)

USBEPDR0s is an 8-byte FIFO buffer specifically for endpoint 0 setup command reception and stores an 8-byte command data that is sent in the setup stage. USBEPDR0s receives only commands requiring processing on the microcomputer (firmware) side. Commands that this module automatically processes are not stored. When command data is received normally, the SETUPTS bit in USB interrupt flag register 0 is set.

As a setup command must be received without fail, 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 and the read data is invalid.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	R	Register for storing the setup command on control OUT transfer

20.3.12 USBEP1 Data Register (USBEPDR1)

USBEPDR1 is a 128-byte receive FIFO buffer for endpoint 1. USBEPDR1 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. When one packet of data is received normally from the host, the EP1FULL bit in USB interrupt flag register 0 is set. 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 again by writing 1 to the EP1RDFN bit in the USB trigger register. The receive data in this FIFO buffer can be transferred by DMA (dual address transfer byte by byte).

USBEPDR1 can be initialized by means of the EP1CLR bit in USBFCLR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0*	D31 to D0	Undefined	R	Data register for endpoint 1 transfer

Note: * 7 to 0 bits for DMA transfer.

20.3.13 USBEP2 Data Register (USBEPDR2)

USBEPDR2 is a 128-byte transmit FIFO buffer for endpoint 2. USBEPDR2 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 the EP2PKTE bit in the USB trigger register is set, one packet of transmit data is fixed, and the dual buffer is switched over. Transmit data for this FIFO buffer can be transferred by DMA (dual address transfer byte by byte).

USBEPDR2 can be initialized by means of the EP2CLR bit in USBFCLR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0*	D31 to D0	Undefined	W	Data register for endpoint 2 transfer

Note: * 7 to 0 bits for DMA transfer.

20.3.14 USBEP3 Data Register (USBEPDR3)

USBEPDR3 is an 8-byte transmit FIFO buffer for endpoint 3, holding one packet of transmit data in endpoint 3 interrupt transfer. Transmit data is fixed by writing one packet of data and setting the EP3PKTE bit in the USB trigger register. When an ACK handshake is received from the host after one packet of data has been transmitted normally, the EP3TS bit in the USB interrupt flag register 0 is set.

USBEPDR3 can be initialized by means of the EP3CLR bit in USBFCLR.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for endpoint 3 transfer

20.3.15 USBEP0o Receive Data Size Register (USBEPSZ0o)

USBEPSZ0o indicates, in bytes, the amount of data received from the host by endpoint 0o.

USBEPSZ0o can be initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Number of bytes received by endpoint 0

20.3.16 USBEP1 Receive Data Size Register (USBEPSZ1)

USBEPSZ1 indicates, in bytes, the amount of data received from the host by endpoint 1. The endpoint 1 FIFO buffer has a dual-FIFO configuration. The receive data size indicated by this register refers to the currently selected FIFO (that can be read by CPU).

USBEPSZ1 can be initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Number of bytes received by endpoint 1

20.3.17 USB Trigger Register (USBTRG)

USBTRG generates one-shot triggers to control the transmit/receive sequence for each endpoint.

USBTRG can be initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	EP3PKTE	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.
5	EP1RDFN	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-FIFO configuration. Writing 1 to this bit initializes the FIFO that was read, enabling the next packet to be received.
4	EP2PKTE	0	W	EP2 Packet Enable After one packet of data has been written to the endpoint 2 FIFO buffer, the transmit data is fixed by writing 1 to this bit.
3	—	0	R	Reserved The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	EP0sRDFN	0	W	EP0s Read Complete Write 1 to this bit after EP0s command FIFO data has been read. Writing 1 to this bit enables transmission/reception of data in the following data stage. A NACK handshake is returned in response to transmit/receive requests from the host in the data stage until 1 is written to this bit.
1	EP0oRDFN	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	EP0iPKTE	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.

20.3.18 USB Data Status Register (USBDASTS)

USBDASTS indicates whether the transmit FIFO buffers contain valid data. A bit is set to 1 when data is written to the corresponding FIFO buffer and the packet enable state is set. This bit is cleared when all data has been transmitted to the host.

In the case of dual-FIFO buffer for endpoint 2, this bit is cleared when all data on two FIFOs has been transmitted to the host.

USBDASTS can be initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved The write value should always be 0.
5	EP3DE	0	R	EP3 Data Present This bit is set when the endpoint 3 FIFO buffer contains valid data.

Bit	Bit Name	Initial Value	R/W	Description
4	EP2DE	0	R	EP2 Data Present This bit is set when the endpoint 2 FIFO buffer contains valid data
3 to 1	—	All 0	R	Reserved The write value should always be 0.
0	EP0iDE	0	R	EP0i Data Present This bit is set when the endpoint 0 transmit FIFO buffer contains valid data.

20.3.19 USBFIFO Clear Register (USBFCLR)

USBFCLR is provided to initialize the FIFO buffers for each endpoint. Writing 1 to a bit clears all the data in the corresponding FIFO buffer. The corresponding interrupt flag is not cleared. Do not clear a FIFO buffer during transmission/reception.

USBFCLR can be initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved The write value should always be 0.
6	EP3CLR	0	W	EP3 Clear When 1 is written to this bit, the endpoint 3 transmit FIFO buffer is initialized.
5	EP1CLR	0	W	EP1 Clear When 1 is written to this bit, both FIFOs in the endpoint 1 receive FIFO buffer are initialized.
4	EP2CLR	0	W	EP2 Clear When 1 is written to this bit, both FIFOs in the endpoint 2 transmit FIFO buffer are initialized.
3, 2	—	All 0	—	Reserved The write value should always be 0.
1	EP0oCLR	0	W	EP0o Clear When 1 is written to this bit, the endpoint 0 receive FIFO buffer is initialized.

Bit	Bit Name	Initial Value	R/W	Description
0	EP0iCLR	0	W	EP0i Clear When 1 is written to this bit, the endpoint 0 transmit FIFO buffer is initialized.

20.3.20 USBDMA Transfer Setting Register (USBDMAR)

USBDMAR enables DMA transfer between the endpoint 1 and endpoint 2 data registers and memory by means of the on-chip DMA controller (DMAC). Dual address transfer is performed with the transfer size of only on a per-byte basis. In order to start DMA transfer, DMAC settings must be made in addition to the settings in this register. For details of DMA transfer, see section 20.7, DMA Transfer.

USBDMAR can be initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved The write value should always be 0.
1	EP2DMAE	0	R/W	Endpoint 2 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 space in the FIFO buffer, a transfer request is asserted for the DMA controller. 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. If there is still space in the other of the two FIFOs, a transfer request is asserted for the DMA controller 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. Also, as EP2-related interrupt requests to the CPU are not automatically masked, interrupt requests should be masked as necessary in the interrupt enable register.

Bit	Bit Name	Initial Value	R/W	Description
0	EP1DMAE	0	R/W	Endpoint 1 DMA Transfer Enable When this bit is set, 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, a transfer request is asserted for the DMA controller. In DMA transfer, when all the received data is read, EP1 is read automatically and the completion trigger operates. Also, as EP1-related interrupt requests to the CPU are not automatically masked, interrupt requests should be masked as necessary in the interrupt enable register.

20.3.21 USB Endpoint Stall Register (USBEPSTL)

The bits in USBEPSTL are used to forcibly stall the endpoints on the application side. While a bit is set to 1, the corresponding endpoint returns a stall handshake to the host. The stall bit for endpoint 0 (EP0STL) is cleared automatically on reception of 8-bit command data for which decoding is performed in this function module. When the SETUPTS flag in USBIFR0 is set, writing 1 to the EP0STL bit is ignored. For details, see section 20.6, Stall Operations. When ASCE = 1 is specified, the EPxSTL bit is automatically cleared.

USBEPSTL can be initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved The write value should always be 0.
4	ASCE	0	R/W	Auto-Stall Clear Enable When this bit is set to 1, the stall setting bit (USBEPSTLR/ESxSTL) of the USB endpoint is automatically cleared after a stall handshake is returned to the host. This bit cannot be set for each endpoint.
3	EP3STL	0	R/W	EP3 Stall When this bit is set to 1, endpoint 3 is placed in the stall state.

Bit	Bit Name	Initial Value	R/W	Description
2	EP2STL	0	R/W	EP2 Stall When this bit is set to 1, endpoint 2 is placed in the stall state.
1	EP1STL	0	R/W	EP1 Stall When this bit is set to 1, endpoint 1 is placed in the stall state.
0	EP0STL	0	R/W	EP0 Stall When this bit is set to 1, endpoint 0 is placed in the stall state.

20.3.22 USB Transceiver Control Register (USBXVERCR)

The USB transceiver control register (USBXVERCR) selects the on-chip transceiver or the external transceiver. Make sure to check if USBIFR1/VBUSMN=0 (VBUS pin disconnection) to overwrite this register.

USBXVERCR can be initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved The write value should always be 0.
0	XVEROFF	0	R/W	Transceiver Control 1: The on-chip transceiver operates. 0: The on-chip transceiver function stops, and digital signals for the external transceiver are output from the port.

20.3.23 USB Bus Power Control Register (USBCTRL)

This LSI can operate using a bus power control method. For details of the bus power control method, see section 20.9, USB Bus Power Control Method.

USBCTRL can be initialized to H'00 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved The write value should always be 0.
1	SUSPEND	0	R/W	USB Suspend Enable Allows an interrupt by the USB suspend signal or awake signal detection.
0	PWMD	0	R/W	Power Mode Changes how the power is supplied. 0: Self-powered 1: Bus-powered

20.4 Operation

20.4.1 Cable Connection

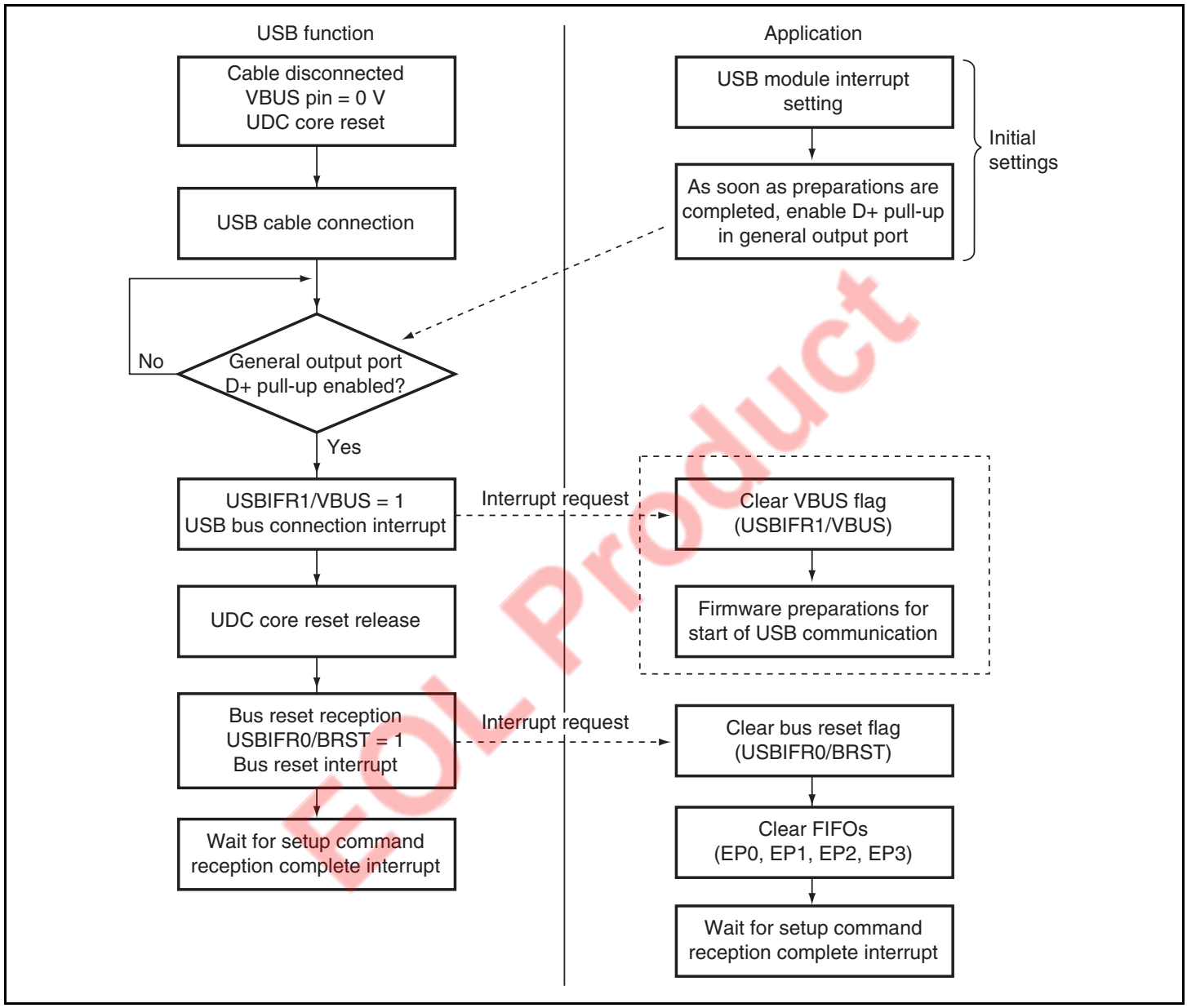


Figure 20.2 Cable Connection Operation

The flowchart in figure 20.2 shows the operation in the case for section 20.8, 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.

Also, in applications that require connection detection regardless of D+ pull-up control, detection should be carried out using IRQx or a general input port. For details, see section 20.8, Example of USB External Circuitry.

20.4.2 Cable Disconnection

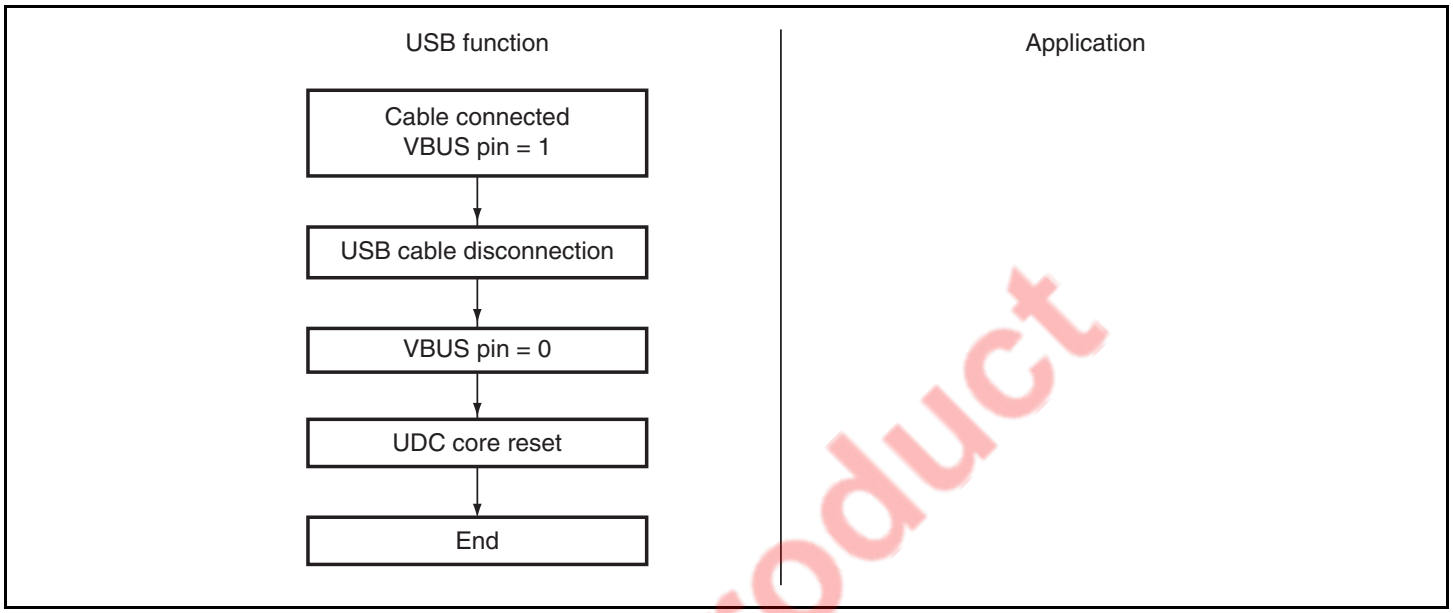


Figure 20.3 Cable Disconnection Operation

The flowchart in figure 20.3 shows the operation in the case for section 20.8, Example of USB External Circuitry.

20.4.3 Control Transfer

Control transfer consists of three stages: setup, data (not always included), and status (figure 20.4). The data stage comprises a number of bus transactions. Operation flowcharts for each stage are shown below.

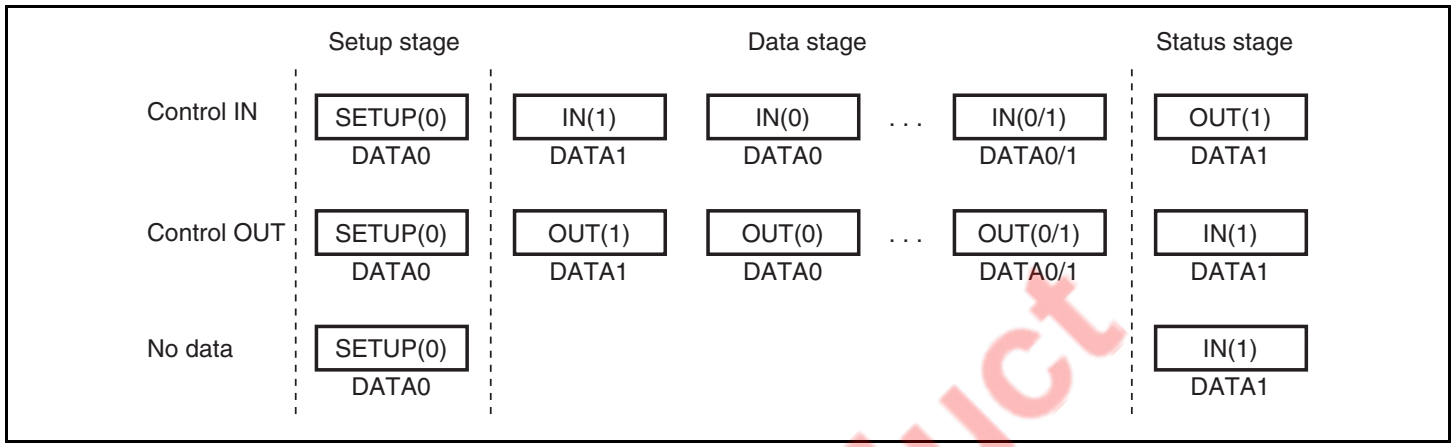
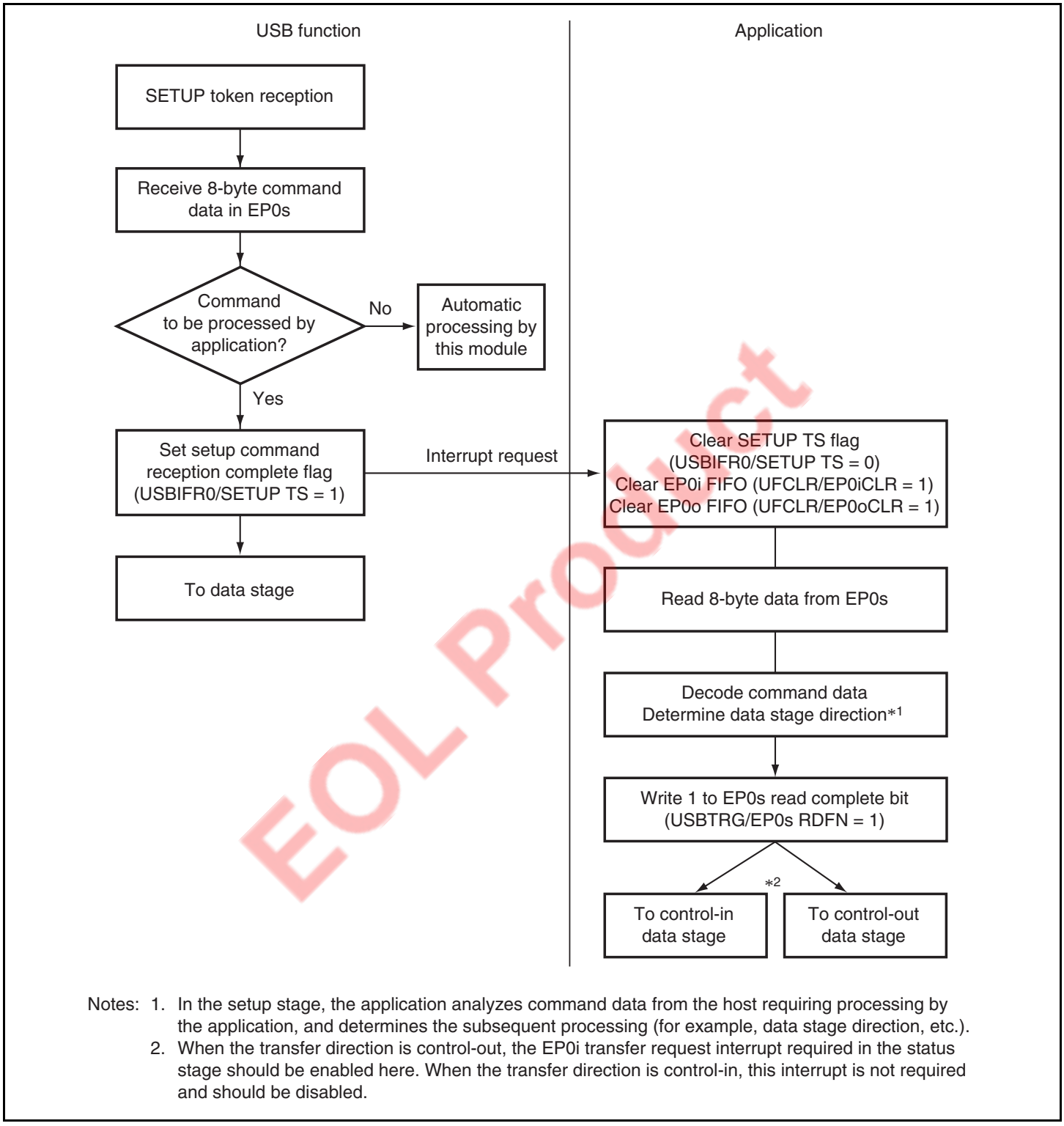


Figure 20.4 Transfer Stages in Control Transfer

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 20.5 Setup Stage Operation

Data Stage (Control-IN): 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 (USBIFR0/EP0iTS = 1).

The end of the data stage is identified when the host transmits an OUT token and the status stage is entered.

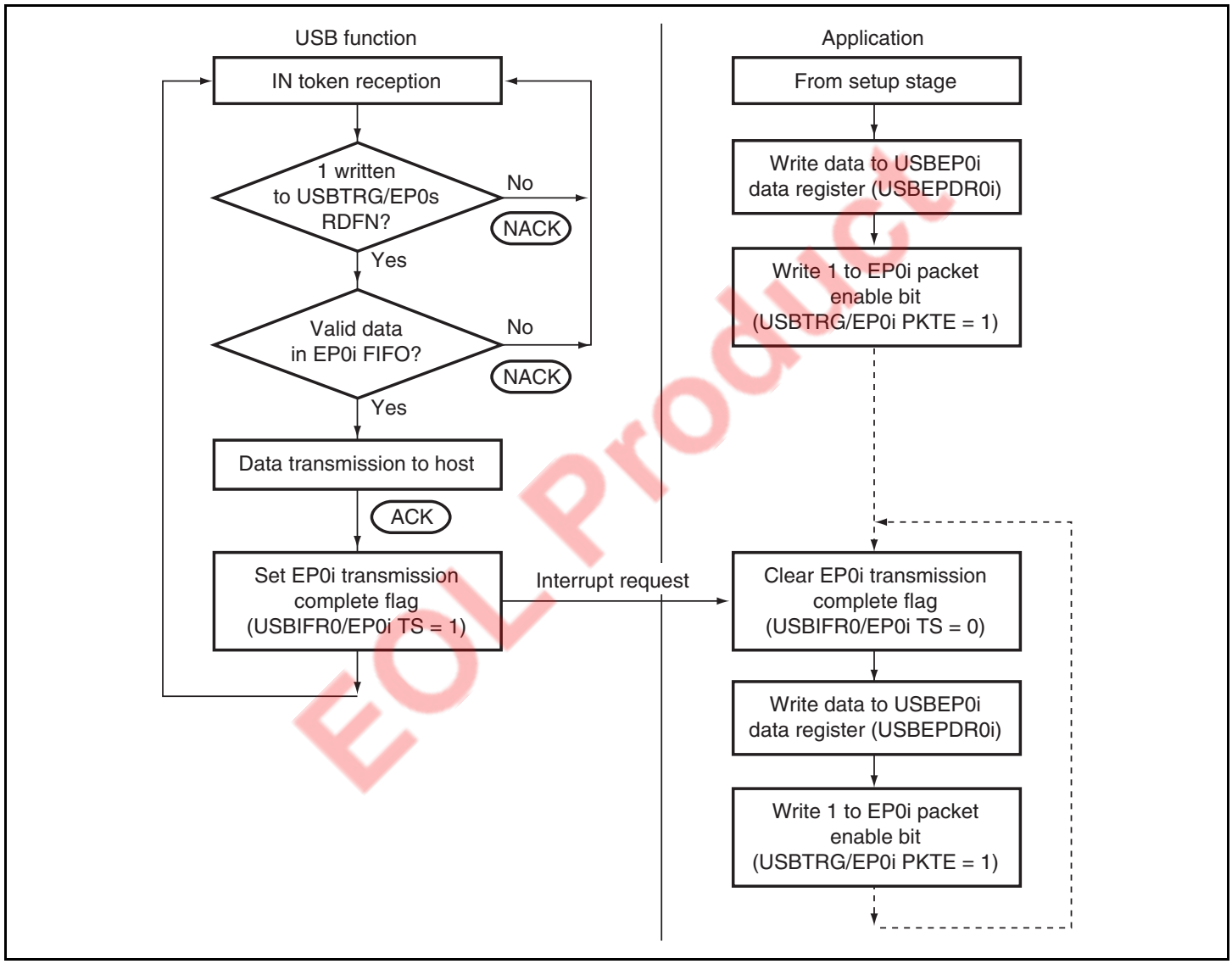


Figure 20.6 Data Stage (Control-IN) Operation

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.

Data Stage (Control-OUT): 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 (USBIFR0/EP0oTS = 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.

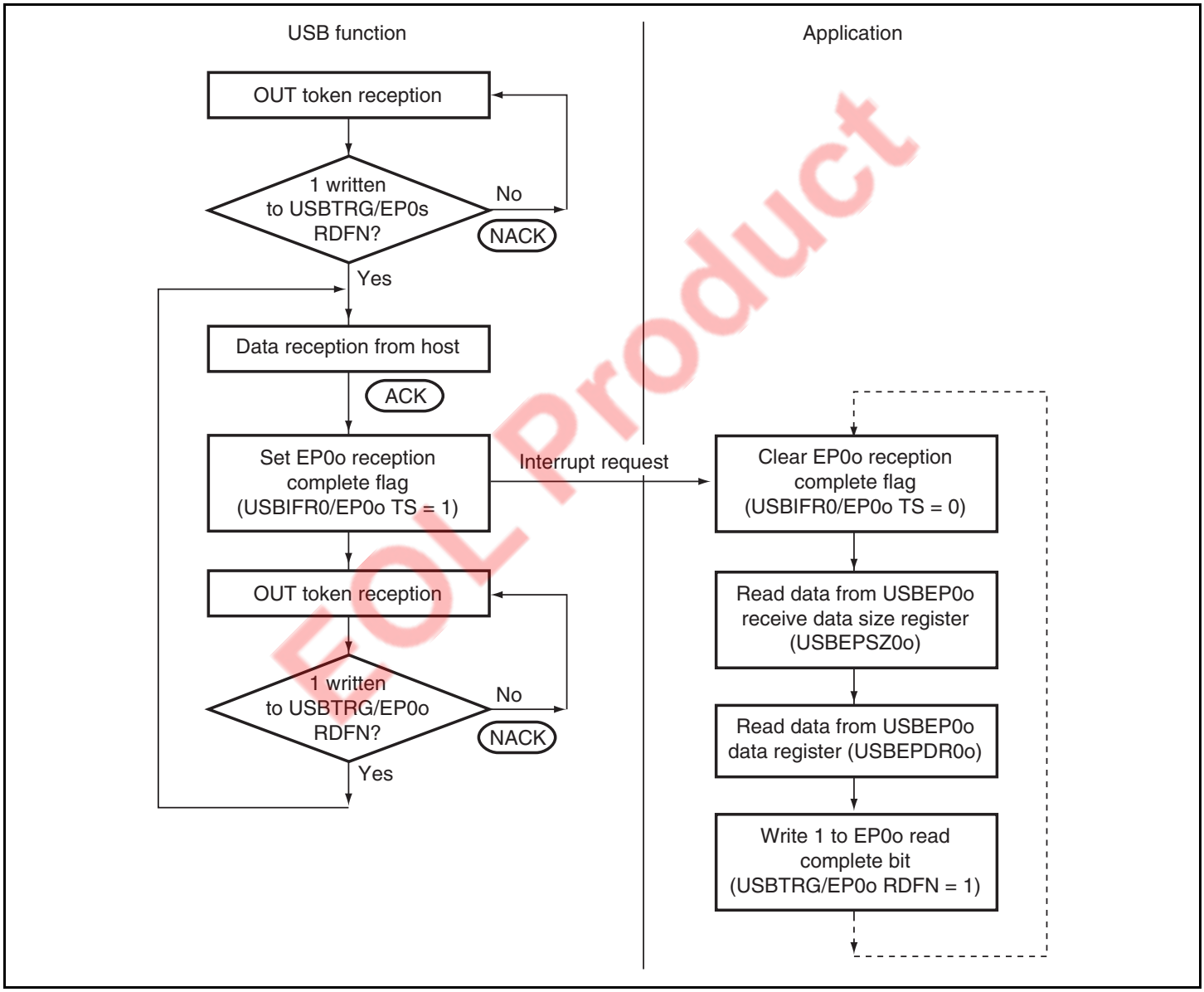


Figure 20.7 Data Stage (Control-OUT) Operation

Status Stage (Control-IN): 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.

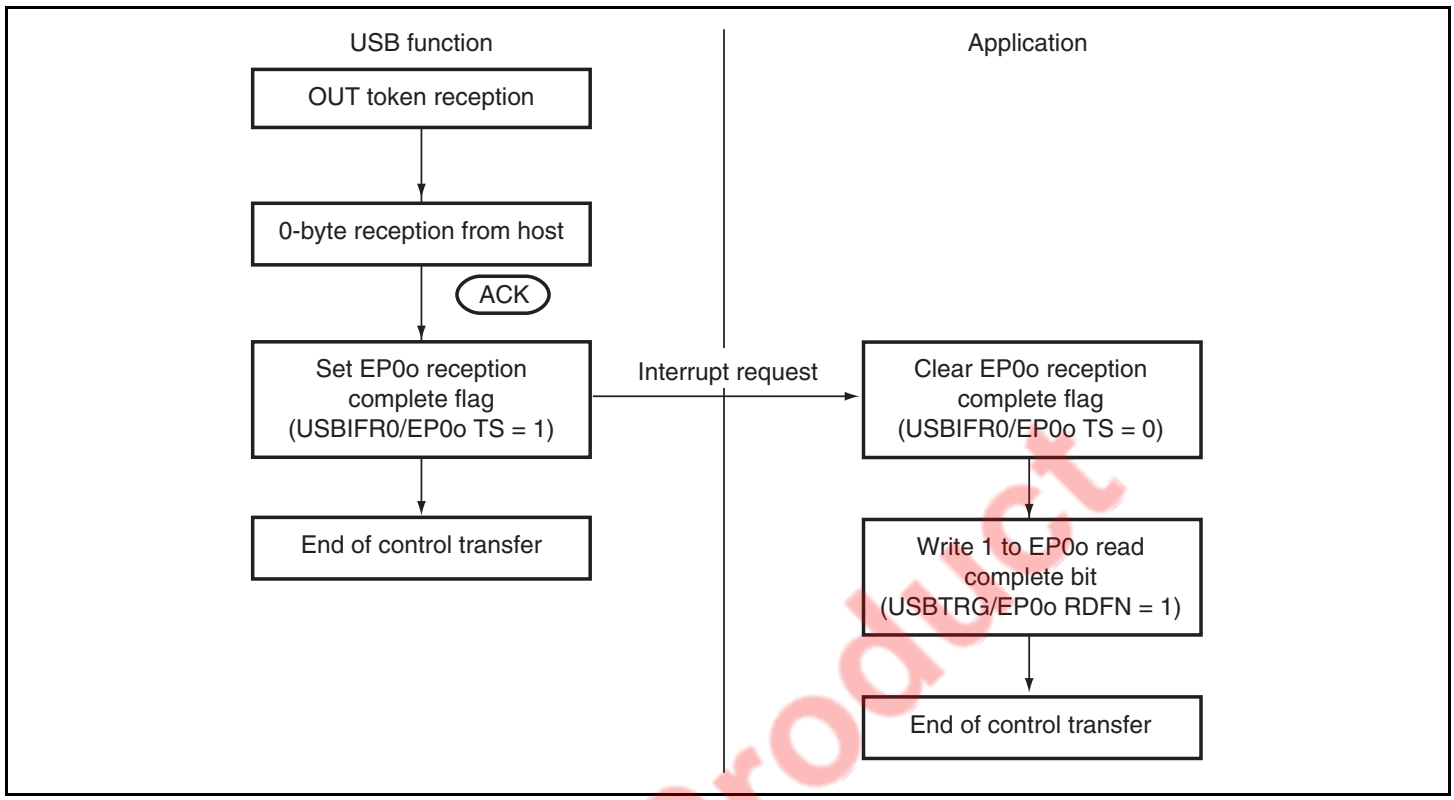


Figure 20.8 Status Stage (Control-IN) Operation

Status Stage (Control-OUT): 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 EP0iFIFO, 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.

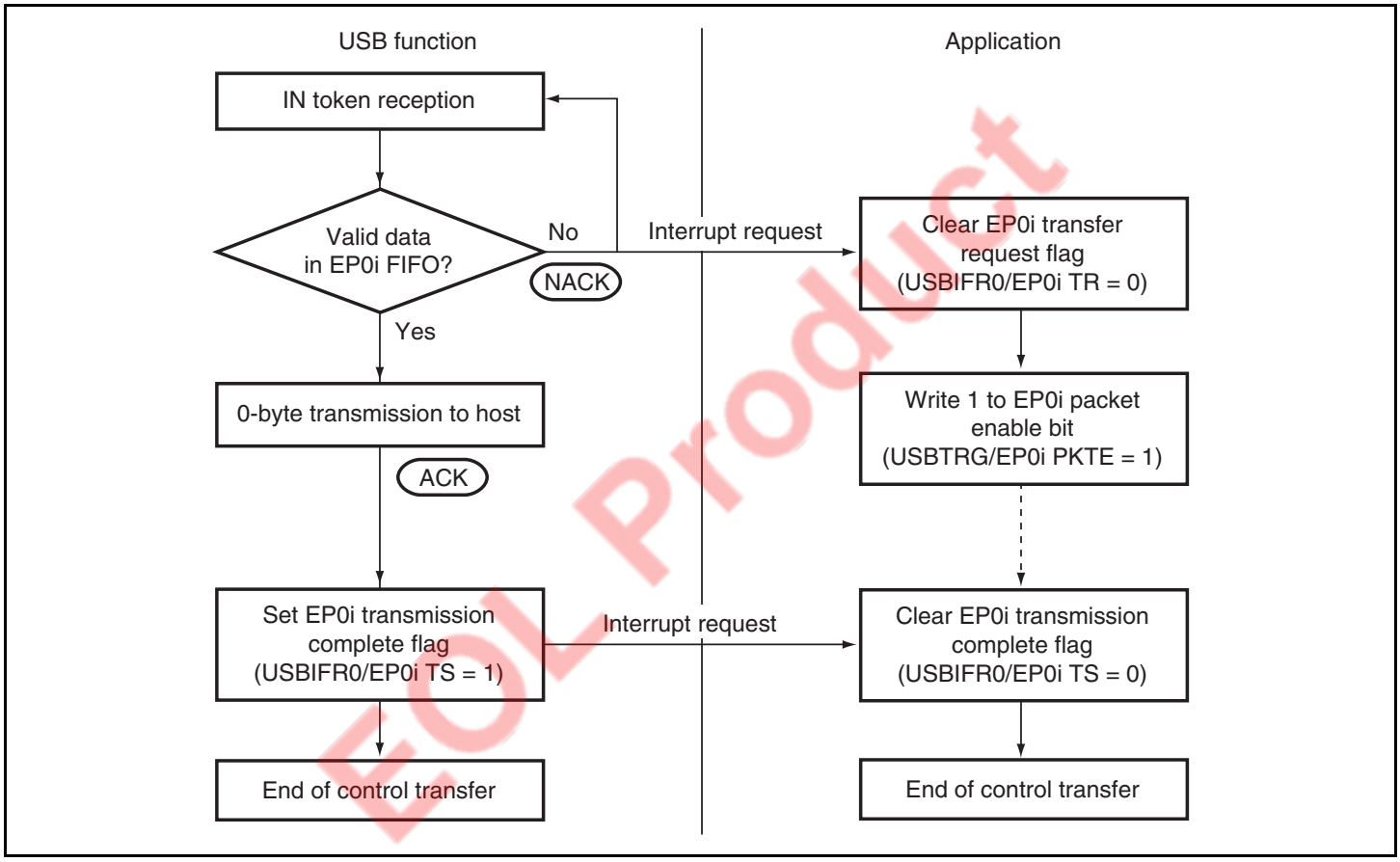


Figure 20.9 Status Stage (Control-OUT) Operation

20.4.4 EP1 Bulk-OUT Transfer (Dual FIFOs)

EP1 has two 64-byte FIFOs, but the user can perform data reception and receive data reads without being aware of this dual-FIFO configuration.

When one FIFO is full after reception is completed, the USBIFR0/EP1 FULL bit 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, NACK is returned to the host automatically. When reading of the receive data is completed following data reception, 1 is written to the USBTRG/EP1 RDFN bit. This operation empties the FIFO that has just been read, and makes it ready to receive the next packet.

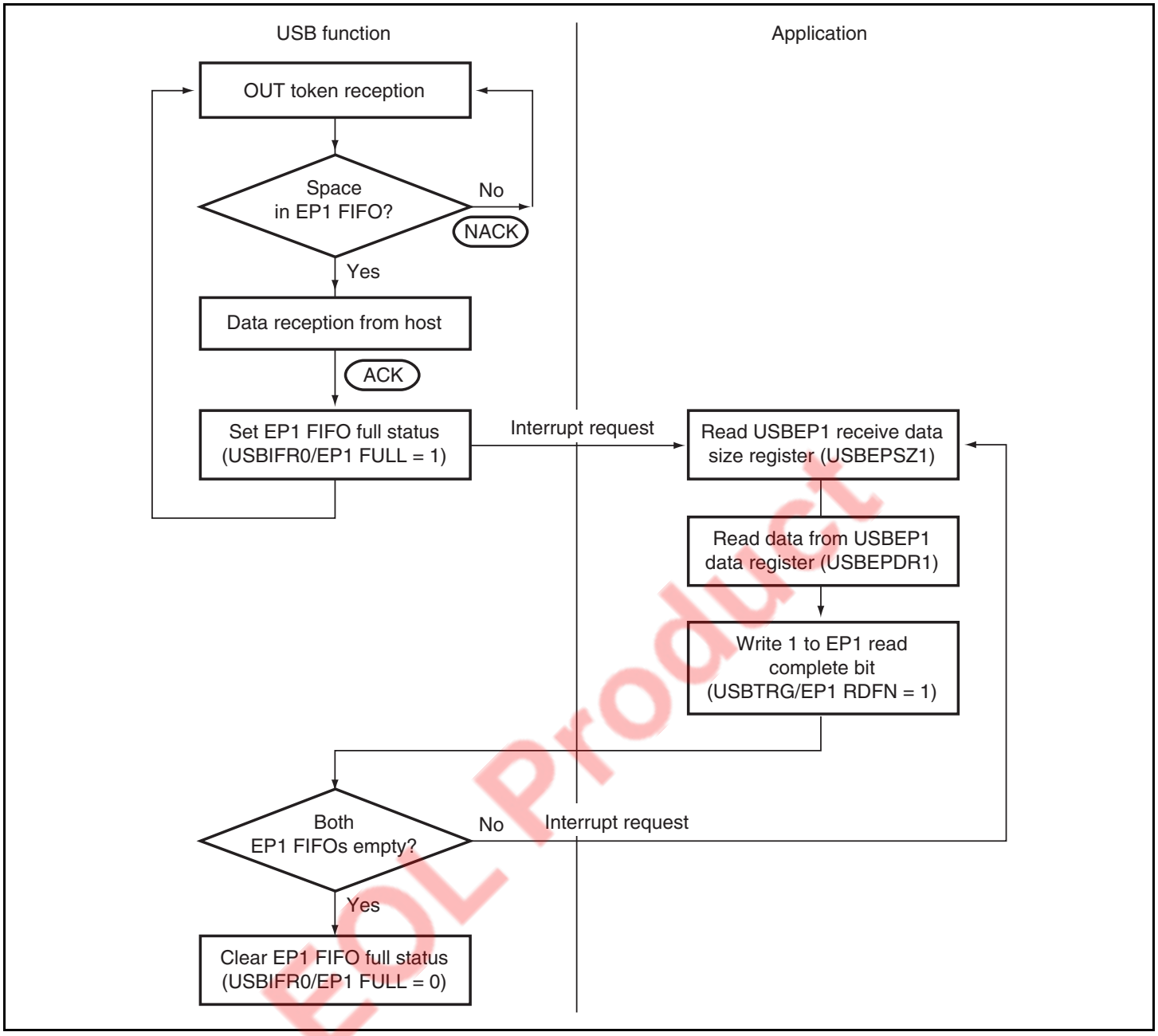


Figure 20.10 EP1 Bulk-OUT Transfer Operation

20.4.5 EP2 Bulk-IN Transfer (Dual FIFOs)

EP2 has two 64-byte FIFOs, but the user can perform data transmission and transmit data writes 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, a USBIFR0/EP2 TR interrupt is requested. With this interrupt, 1 is written to the USBIER0/EP2EMPTY bit, 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, EP2EMPTY is cleared to 0. If at least one FIFO is empty, USBIFR0/EP2EMPTY 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 USBIER0/EP2EMPTY and disable interrupt requests.

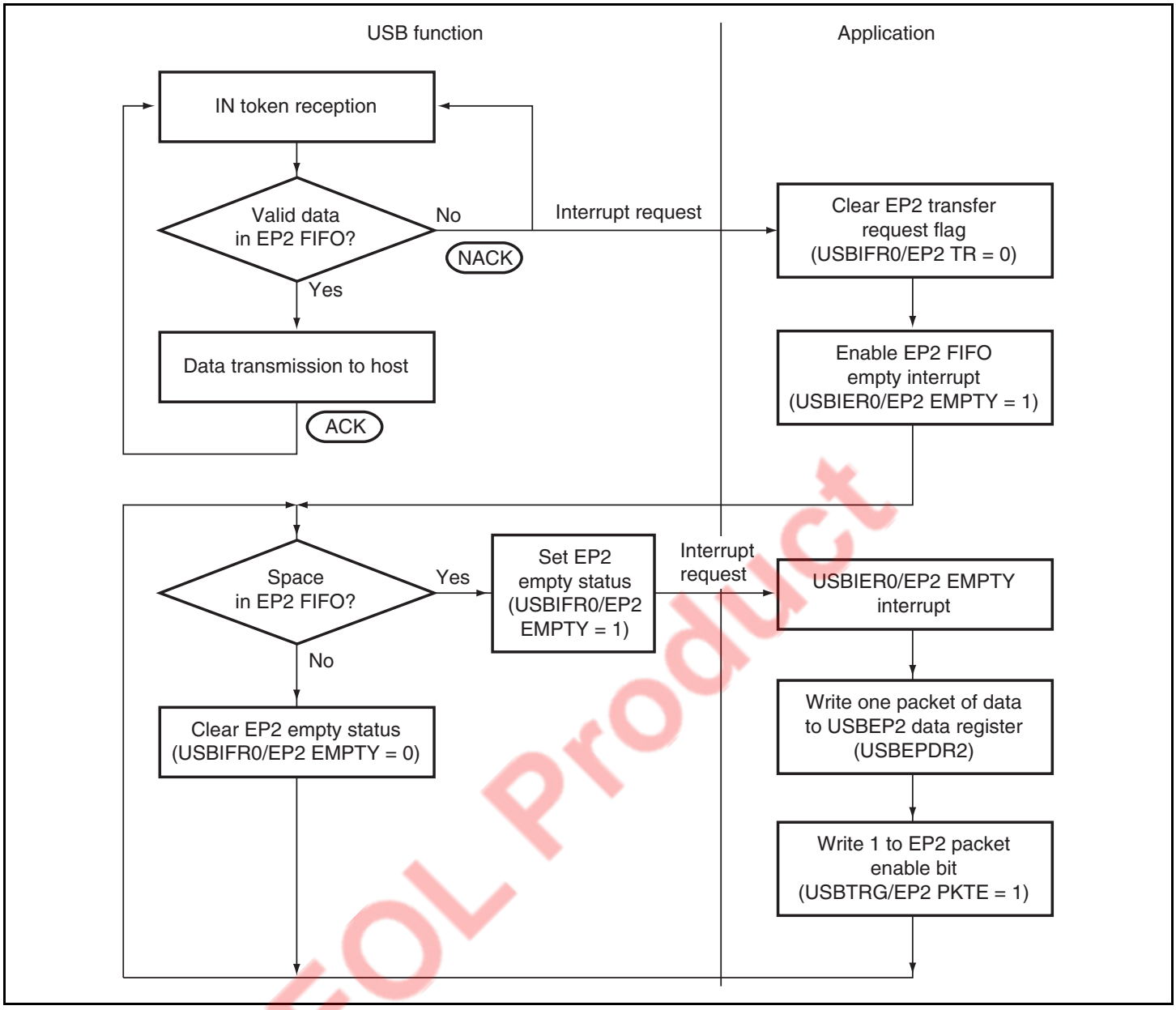
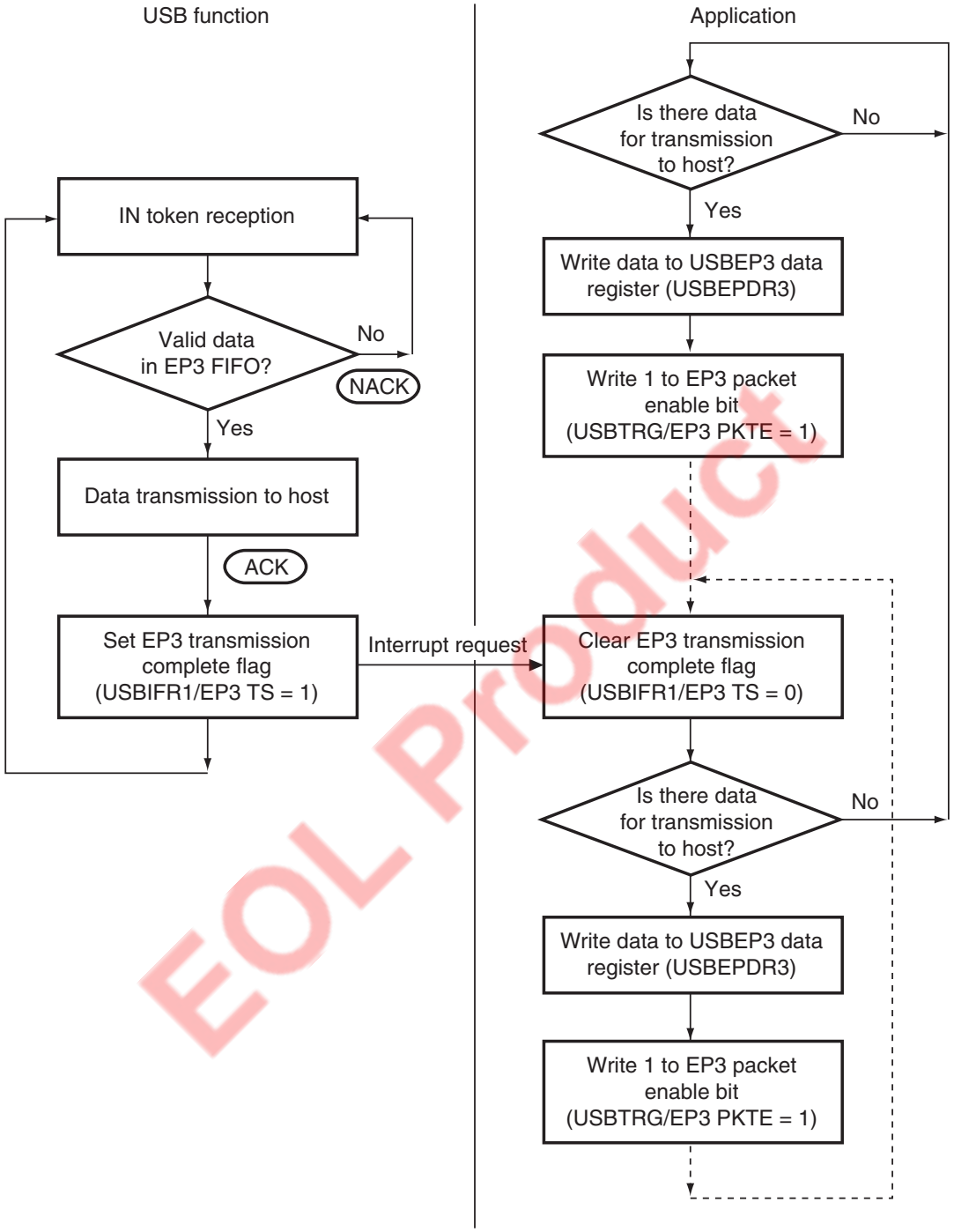


Figure 20.11 EP2 Bulk-IN Transfer Operation

20.4.6 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 USB data status register is referenced to confirm that the FIFO is empty, and then data is written to the FIFO.

Figure 20.12 EP3 Interrupt-IN Transfer Operation

20.5 Processing of USB Standard Commands and Class/Vendor Commands

20.5.1 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 20.2 below.

Table 20.2 Command Decoding on Application Side

Decoding not Necessary on Application Side	Decoding Necessary on Application Side
Clear feature	Get Descriptor
Get configuration	Synch Frame
Get interface	Set Descriptor
Get status	Class/Vendor command
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, the USB function module stores the command in the EP0s FIFO. After normal reception is completed, the USBIER0/SETUP TS flag is set and an interrupt request is generated. In the interrupt routine, 8 bytes of data must be read from the EP0s data register (USBEPDR0S) 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.

20.6 Stall Operations

This section describes stall operations in the USB function 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. The internal status bit for EP0 is automatically cleared only when the setup command is received.

20.6.1 Forcible Stall by Application

The application uses USBEPSTL 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 USBEPSTL (1-1 in figure 20.13). The internal status bits are not changed. When a transaction is sent from the host for the endpoint for which the USBEPSTL bit was set, the USB function module references the internal status bit, and if this is not set, references the corresponding bit in USBEPSTL (1-2 in figure 20.13). If the corresponding bit in USBEPSTL is set, the USB function module sets the internal status bit and returns a stall handshake to the host (1-3 in figure 20.13). If the corresponding bit in USBEPSTL 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 USBEPSTL register. Even after a bit is cleared by the Clear Feature command (3-1 in figure 20.13), the USB function module continues to return a stall handshake while the bit in USBEPSTL is set, since the internal status bit is set each time a transaction is executed for the corresponding endpoint (1-2 in figure 20.13). To clear a stall, therefore, it is necessary for the corresponding bit in USBEPSTL 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 20.13).

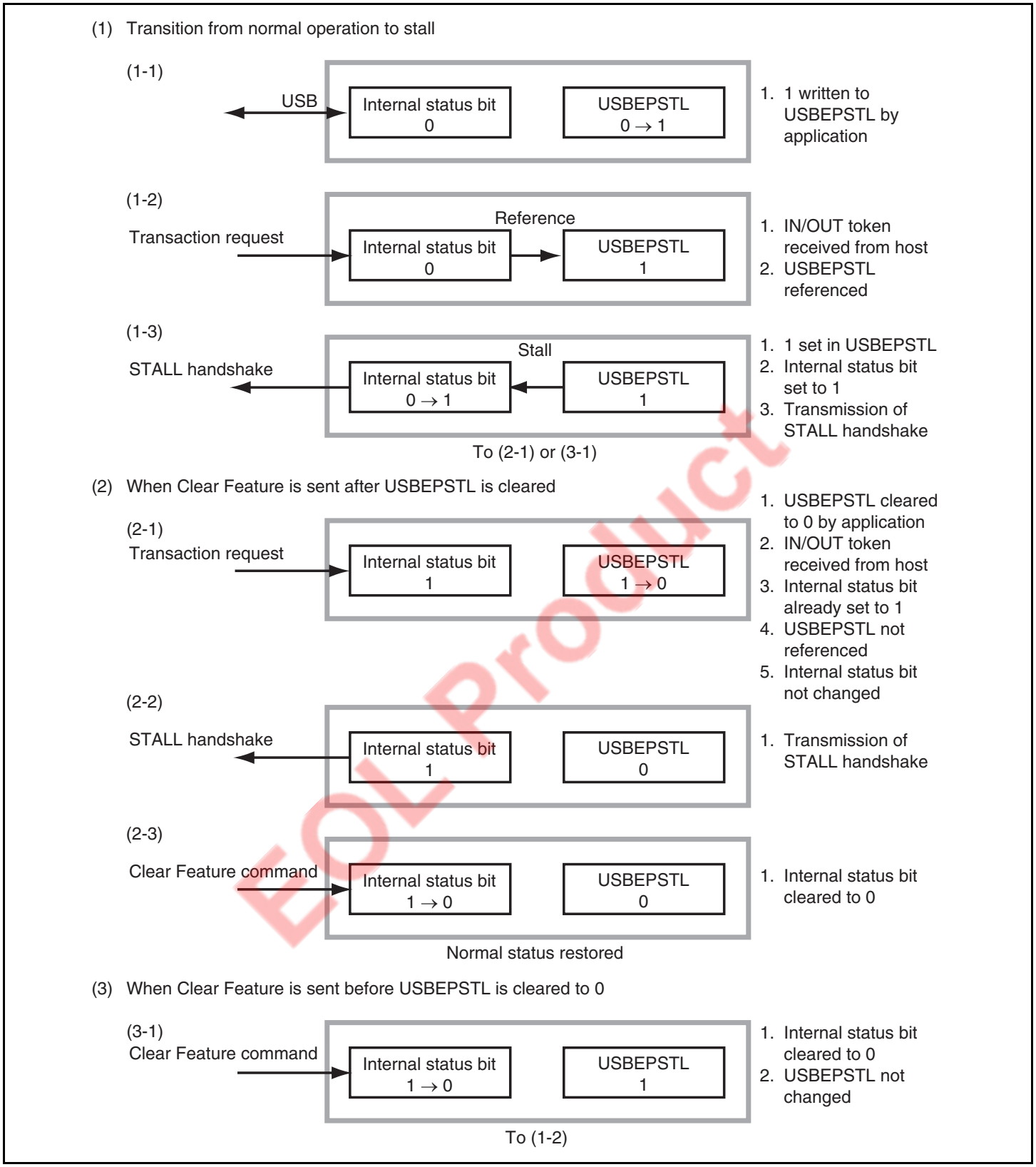


Figure 20.13 Forcible Stall by Application

20.6.2 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 USBEPSTL register, and returns a stall handshake (1-1 in figure 20.14).

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to USBEPSTL register. After a bit is cleared by the Clear Feature command, USBEPSTL is referenced (3-1 in figure 20.14). 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 20.14). To clear a stall, therefore, the internal status bit must be cleared with a Clear Feature command (3-1 in figure 20.14). If set by the application, USBEPSTL should also be cleared (2-1 in figure 20.14).

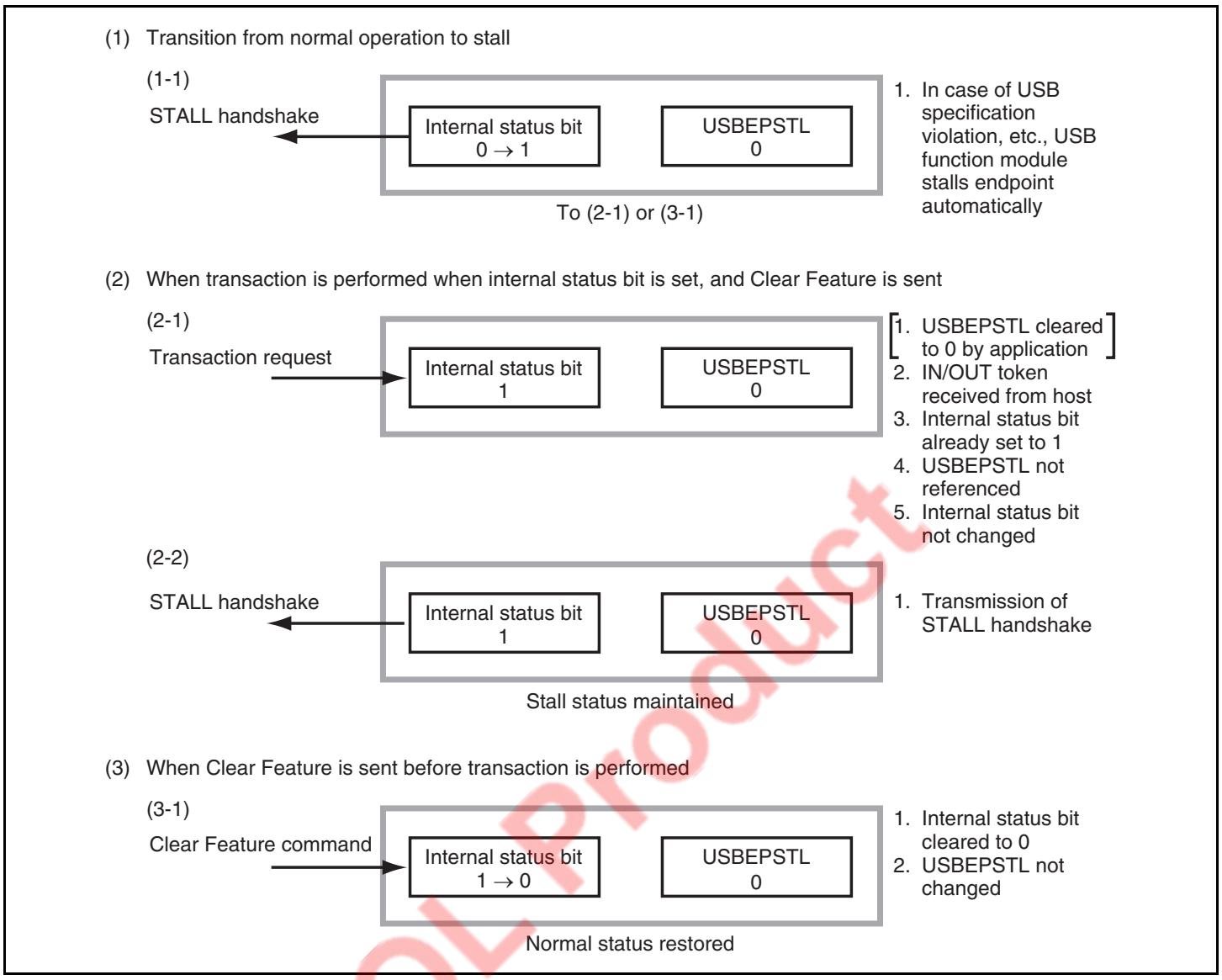


Figure 20.14 Automatic Stall by USB Function Module

20.7 DMA Transfer

This module allows DMAC transfer for endpoints 1 and 2, excluding transfer of word and longword.

If endpoint 1 contains at least one byte of valid receive data, a DMA transfer request is issued to endpoint 1. If there is no valid data in endpoint 2, a DMA transfer request is issued to endpoint 2.

When EP1 DMAE in the USBDMA setting register is set to 1 to allow DMA transfer, 0-length data received for endpoint 1 is ignored. When DMA transfer is set, it is unnecessary to write 1 to the EP1 USBTRG/RDFN and EP2 USBTRG/PKTE bits. (1 must be written to the USBTRG/PKTE bit for data that consists of the maximum number of bytes or less.) For EP1, the FIFO buffer automatically becomes empty when all the received data is read. For EP2, the FIFO automatically becomes full when the maximum number of bytes (64 bytes) is written to the FIFO and then the data in the FIFO is transmitted. (See figures 20.15 and 20.16.)

20.7.1 DMA Transfer for Endpoint 1

If the received data for EP1 is transferred by DMA when the data on the currently selected FIFO becomes empty, an equivalent processing of writing 1 to the USBTRG/RDFN bit is automatically performed in the module. Therefore, do not write 1 to the EP1RDFN bit in USBTRG after reading the data on one side of the FIFO. Correct operation cannot be guaranteed.

For example, if 150 bytes of data are received from the host, the equivalent processing of writing 1 to the USBTRG/RDFN bit is automatically performed internally in the three places in figure 20.15. This processing is done when the data on the currently selected FIFO becomes empty meaning that the processing is to be automatically performed even if 64 bytes of data or less than that are transferred.

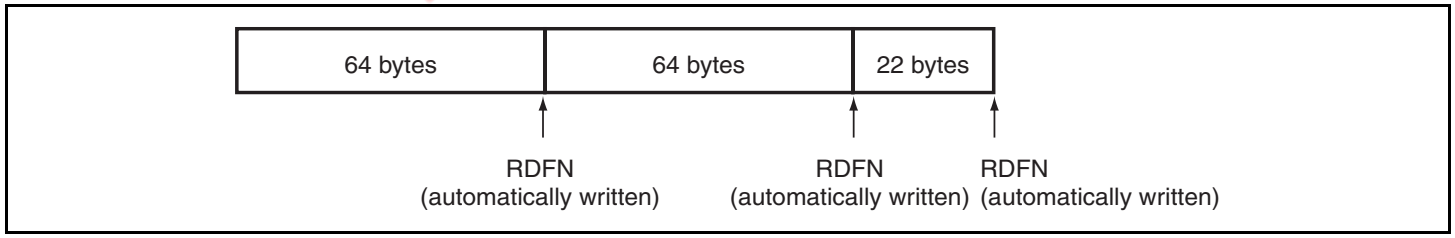


Figure 20.15 EP1 RDFN Operation

20.7.2 DMA Transfer for Endpoint 2

When the transmitted data for EP2 is transferred by DMA when the data on one side of FIFO (64 bytes) becomes full an equivalent processing of writing 1 to the USBTRG/PKTE bit is automatically performed in the module. Therefore, when data to be transferred is a multiple of 64 bytes, writing 1 to the USBTRG/PKTE bit is not necessary.

For the data less than 64 bytes, a 1 should be written to the USBTRG/PKTE bit by a DMA transfer end interrupt of the DMAC. If a 1 is written to the USBTRG/PKTE bit for transferring the maximum number of bytes (64 bytes), the correct operation cannot be guaranteed.

For example, if 150 bytes of data are transmitted to the host, the equivalent processing if writing 1 to the USBTRG/PKTE bit is automatically performed internally in the two places in figure 20.16. This processing is done when the data on the currently selected FIFO becomes full meaning that the processing is to be automatically performed only when 64 bytes of data are transferred.

When the last 22 bytes are transferred, write 1 to the USBTRG/PKTE bit because this is not automatically written to. There is no data to be transferred in the application side, but this module outputs the DMA transfer request for EP2 as long as the FIFO has a space. When all the data is transferred by DMA, write 0 to the USBDMA/EP2DMAE bit to cancel the DMA transfer request for EP2.

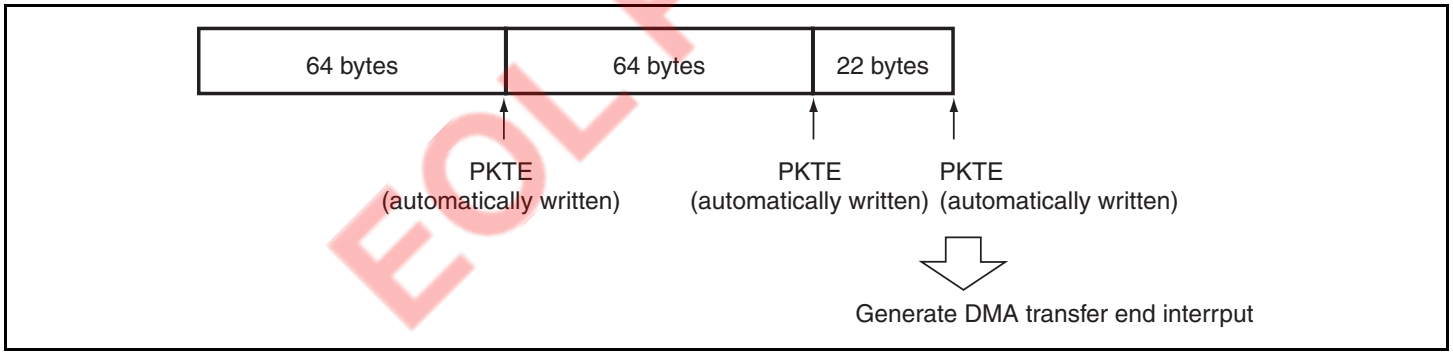


Figure 20.16 EP2 PKTE Operation

20.8 Example of USB External Circuitry

USB Transceiver: When an on-chip transceiver is not used, a USB transceiver IC (such as a PDIUSBP11) must be connected externally. The USB transceiver manufacturer should be consulted concerning the recommended circuit from the USB transceiver to the USB connector, etc.

D+ Pull-Up Control: In a system where it is wished to delay USB host/hub connection notification (D+ pull-up) (during high-priority processing or initialization processing, for example), D+ pull-up is controlled using a general output port. When the USB cable has been connected to the host or hub and D+ pull-up is inhibited, D+ and D- are placed in the low level state (D+ and D- are pull down on the host or hub side) and the USB module recognizes as if the USB bus reset has been received from the host. In that case, the D+ pull-up control signal and VBUS pin input signal should be controlled using a general output port and the USB cable VBUS (AND circuit) as shown in figure 20.17. (The UDC core of this LSI holds the powered state independent of D+ and D- state when the VBUS pin is low level.)

Detection of USB Cable Connection/Disconnection: As USB states 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 on-chip function LSI power is off, a voltage (5 V) will be applied from the USB host/hub. Therefore, an IC (HD74LV1G08A, 2G08A, etc.) that allows voltage application when the system power is off should be connected externally.

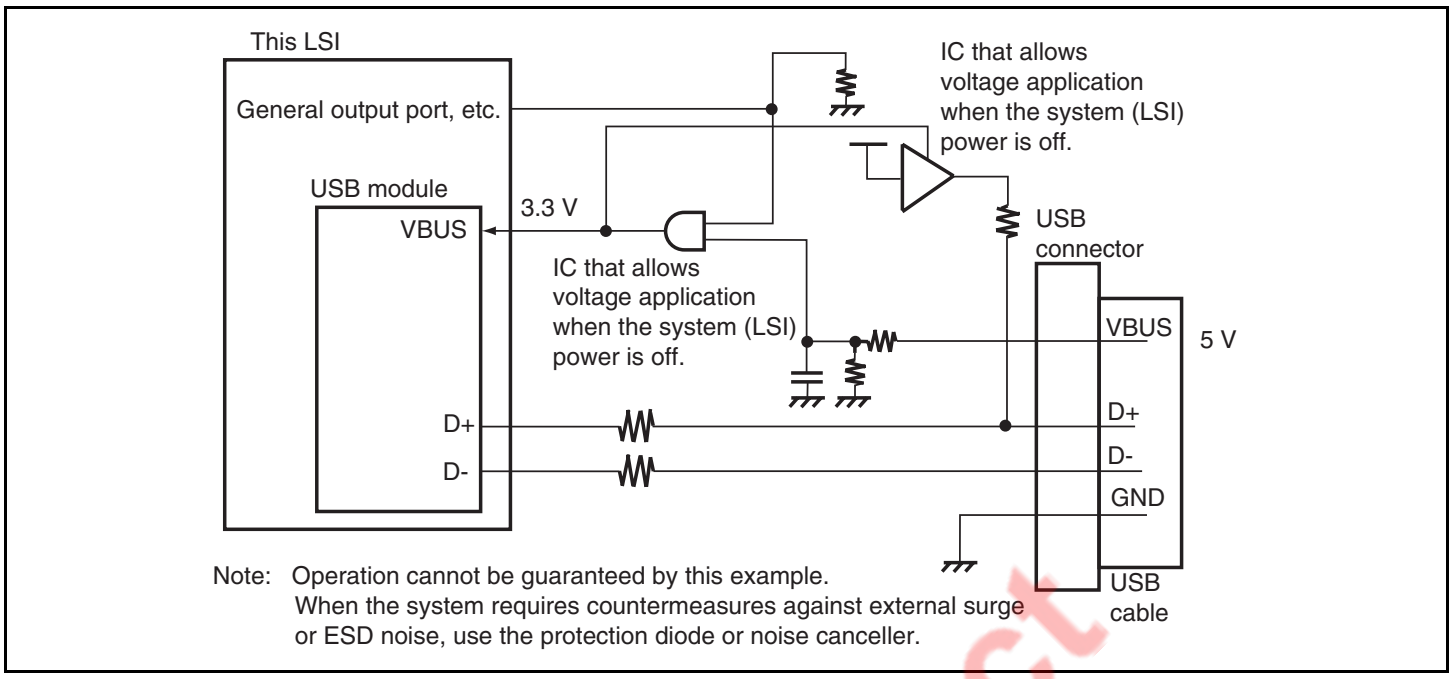
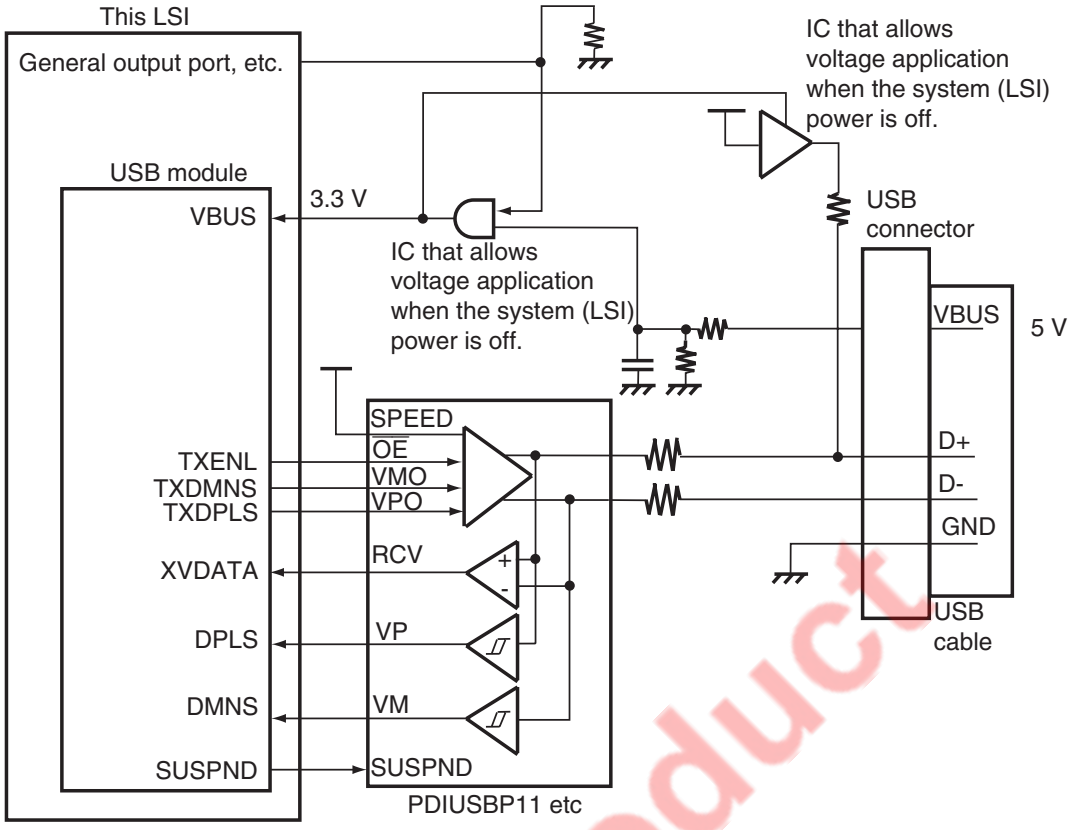


Figure 20.17 Example of USB Function Module External Circuitry (For On-Chip Transceiver)



Note: Operation cannot be guaranteed by this example.
When the system requires countermeasures against external surge or ESD noise, use the protection diode or noise canceller.

Figure 20.18 Example of USB Function Module External Circuitry (For External Transceiver)

20.9 USB Bus Power Control Method

20.9.1 USB Bus Power Control Operation

This LSI can operate using a USB bus power control method.

The following describes notes on the LSI using the USB bus power control method.

Changing to High-Power Function: According to the USB standard, the startup operation (from connecting cables to completing enumeration) is handled as a low-power function. Changing to the high-power function can be checked by detecting reception of a SET_CONFIGURATION request from USBIFR2 and IFRIER2/SETC and confirming USBIFR2/CFGV = 1.

Suspending: In this LSI, an interrupt by detecting the USB suspend signal or awake signal can be shared with an $\overline{\text{IRQ0}}$ or $\overline{\text{IRQ1}}$ interrupt by specifying USBCTRL/SUSPEND = 1. (See figure 20.19.)

This causes an $\overline{\text{IRQ1}}$ interrupt to occur by specifying USBIFR2/SUSPS = 1 to change the LSI state to the standby mode. An $\overline{\text{IRQ0}}$ interrupt occurs by specifying USBIFR2/AWAKE=1, the LSI can be returned from the standby mode. Since the LSI must enter the USB suspend state within 10 ms after the USB suspend signal is detected, an $\overline{\text{IRQ1}}$ interrupt must be set to be processed prior to other interrupts. When the $\overline{\text{IRQ0}}$ interrupt priority is lower than the interrupt request mask level (SR/I[3:0]), the LSI cannot be returned from the suspend state. Make sure that the $\overline{\text{IRQ0}}$ interrupt priority is higher than the interrupt request mask level (SR/I[3:0]). Figure 20.20 shows the operation timing.

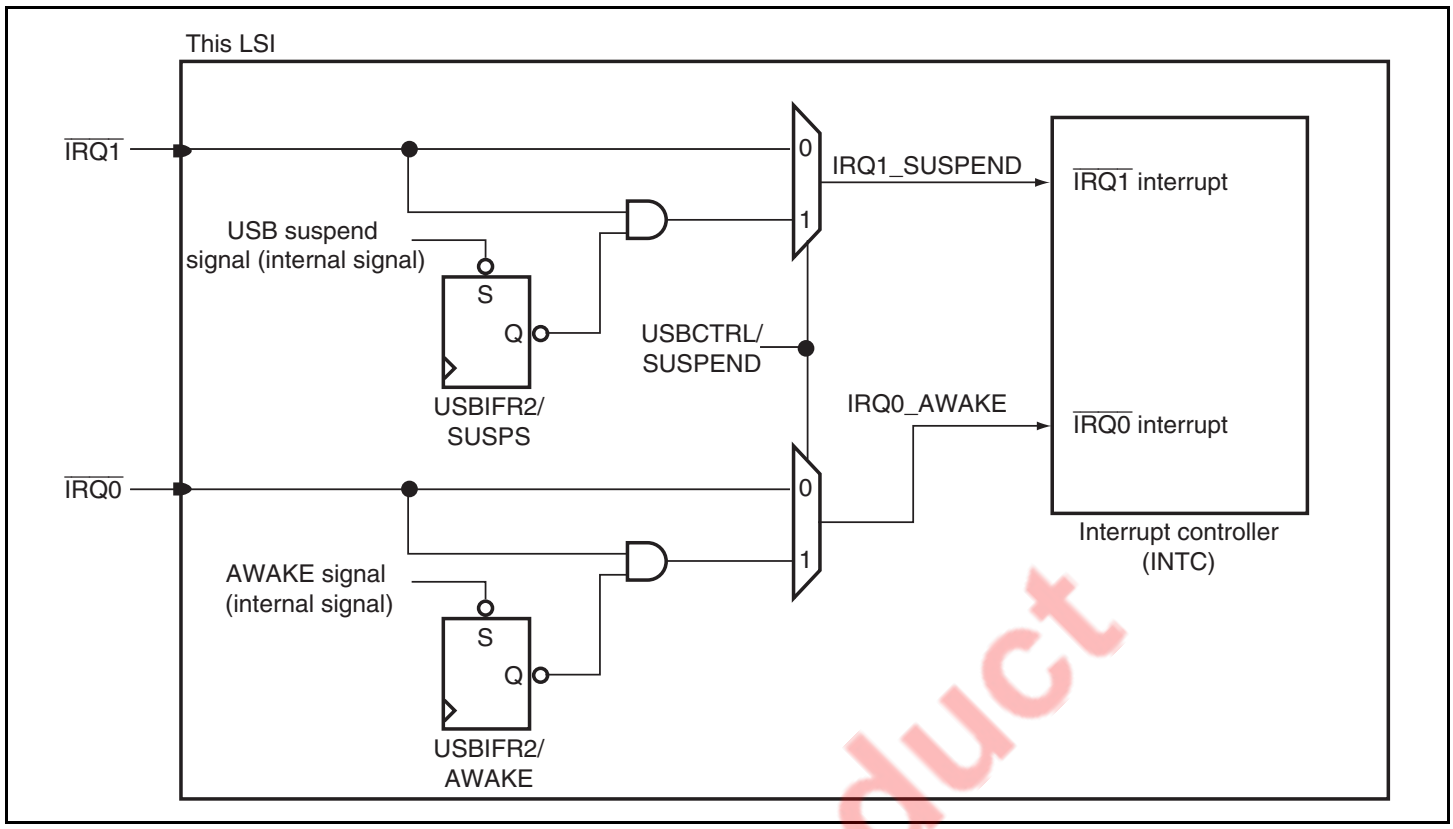


Figure 20.19 $\overline{\text{IRQ0}}$ and $\overline{\text{IRQ1}}$ Interrupt Circuitry

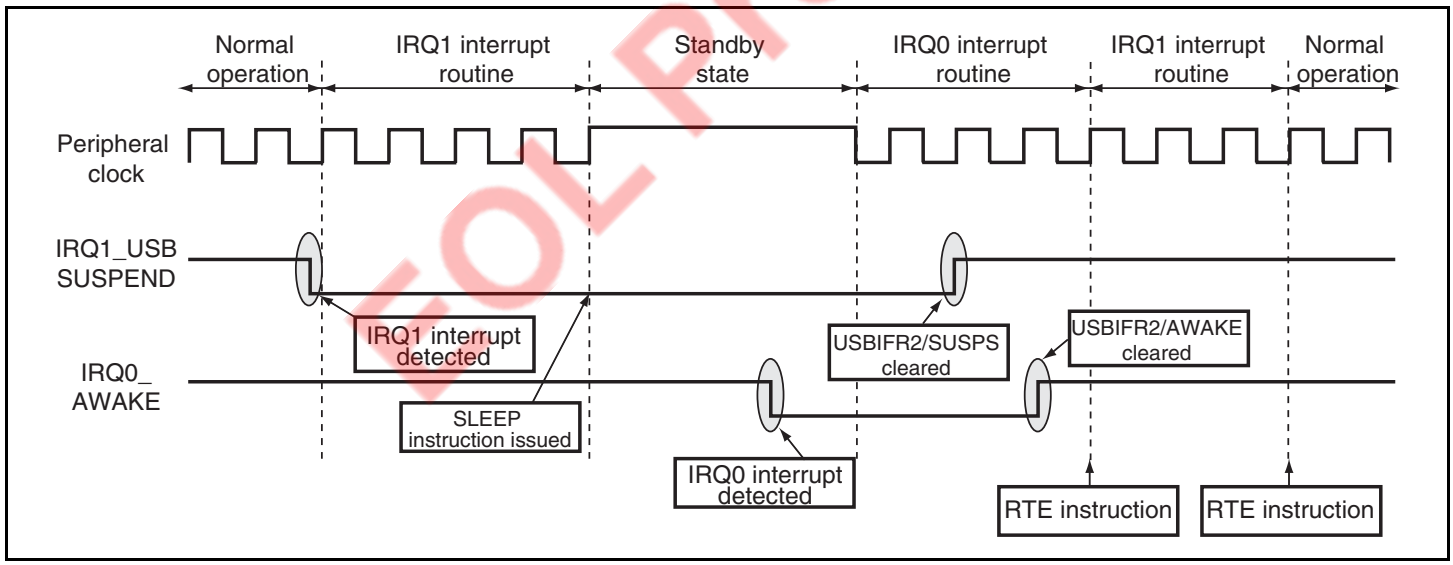


Figure 20.20 USB Standby Operation Timing

20.9.2 Usage Example of USB Bus Power Control Method

Figures 20.21 to 20.23 show flowcharts for initializing, entering standby mode and awaking when using the USB bus power control method.

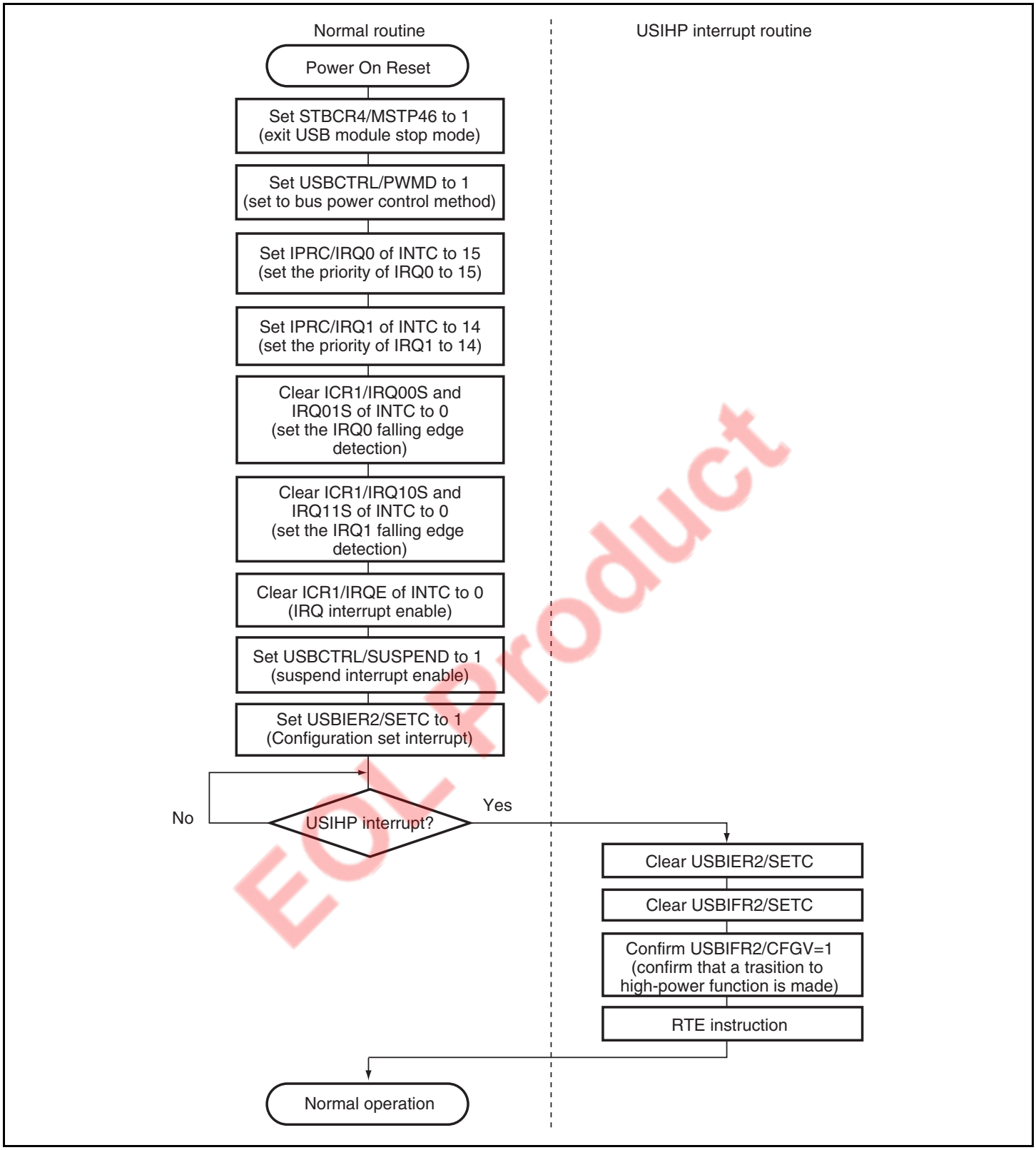


Figure 20.21 Sample Flowchart for Initialization of the USB Bus Power Control Method

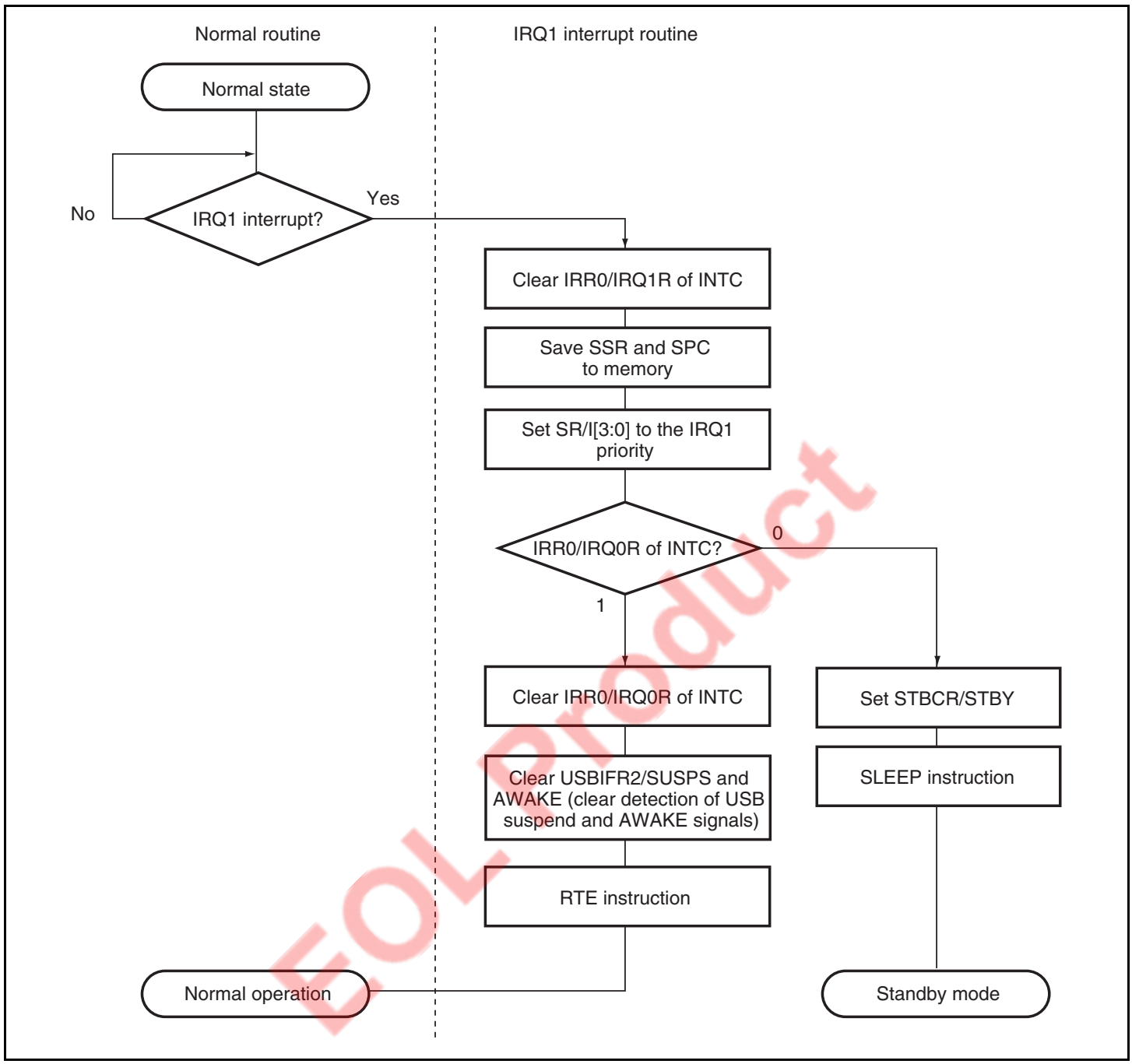


Figure 20.22 Sample Flowchart for Changing the State from USB Suspend to Standby

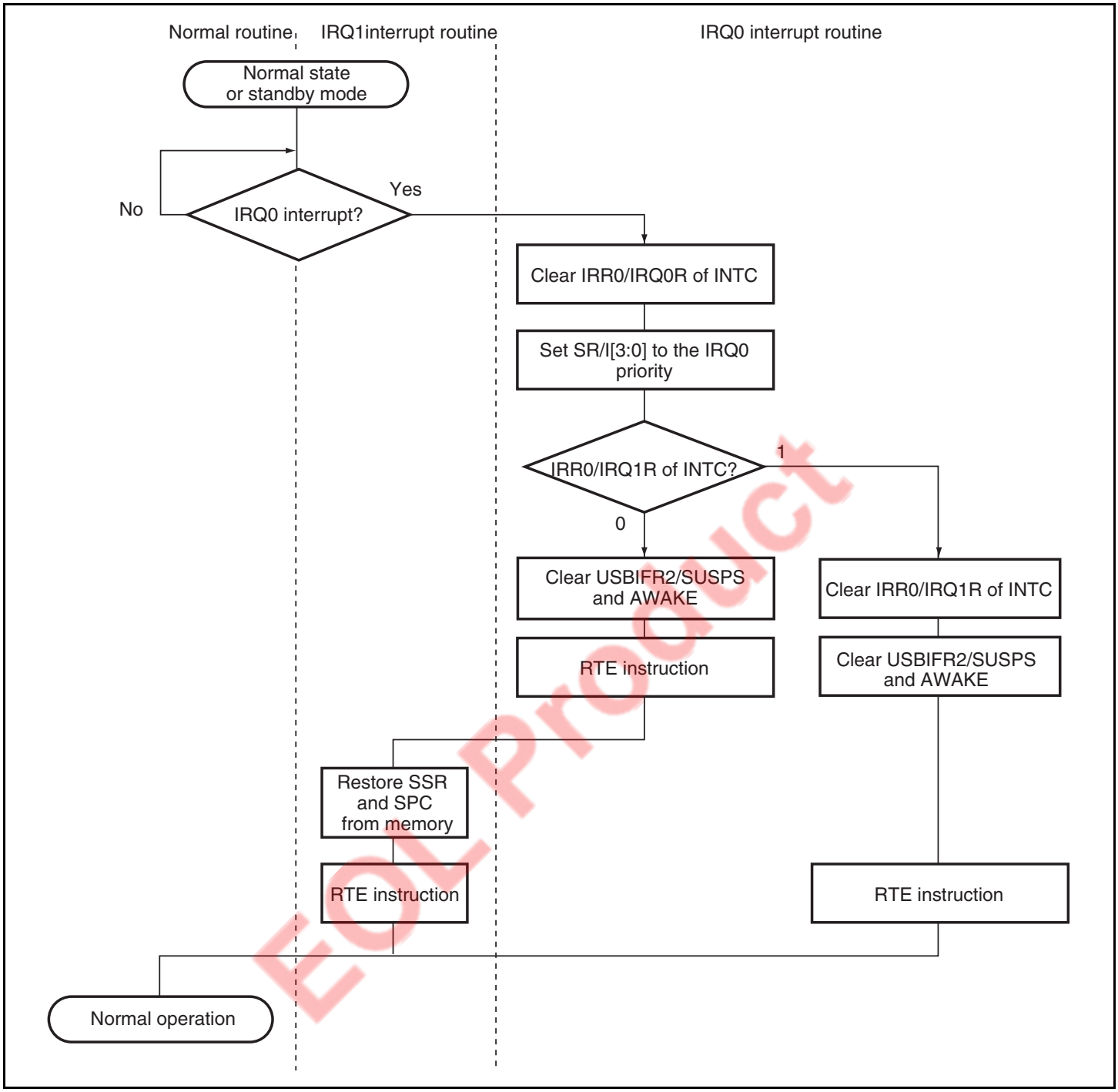


Figure 20.23 Sample Flowchart for AWAKE

20.10 Notes on Usage

20.10.1 Receiving Setup Data

Note that the following when 8-byte setup data is received by USBEPDR0s.

1. The USB must always receive the setup command. Therefore, writing from the USB bus has priority over reading from the CPU. When the USB starts receiving the next setup command while the CPU is reading data after data reception, the USB forcibly invalidates reading from the CPU to start writing. The value that is read after starting reception is undefined.
2. USBEPDR0s must be read in 8-byte unit. When reading is stopped in the middle, the data that is received by the next setup command cannot be read correctly.

20.10.2 Clearing FIFO

If the connected USB cable is disconnected during communication, the data being received or transmitted may remain in the FIFO. Therefore, clear the FIFO immediately after connecting the USB cable.

Do not clear the FIFO that is receiving or transmitting data from or to the host.

20.10.3 Overreading or Overwriting Data Register

Note that the following when reading or writing the data register of this module:

Receive Data Register: Do not read the number of data which exceeds that of valid receive data from the receive data register, i.e., data that exceeds the number of bytes indicated by the receive data size register must not be read. For USBEPDR1 that has two FIFOs, the maximum number of bytes that can be read at once is 64 bytes. After reading the data on the currently selected side, write 1 to USBTRG/EP1RDFN to change the current side to another side. This allows the number of bytes for the new side to be used as the receive data size, enabling the next data to be read.

Transmit Data Register: Do not write the number of data that exceeds the maximum packet size to the transmit data register. For USBEPDR2 that has two FIFOs, the data to be written at one time must be the maximum packet size or less. After writing the data, write 1 to TRG/PKTE to change the currently selected side to another in the module to allow the next data to be written to the new side. Therefore, do not write data to one side of FIFO right after the other side.

20.10.4 Assigning Interrupt Source for EP0

Interrupt sources (bits 0 to 3) for EP0 that are assigned to USBIFR0 of this module must be assigned to the same interrupt pin using USBISR0.

20.10.5 Clearing FIFO when Setting DMA Transfer

Clearing the endpoint 1 data register (USBEPDR1) is impossible when DMA transfer is enabled (USBDMAR/EP1DMAE = 1) for endpoint 1. To clear this register, cancel DMA transfer.

20.10.6 Manual Reset for DMA Transfer

Do not input a manual reset during DMA transfer for endpoints 1 and 2. Correct operation cannot be guaranteed.

20.10.7 USB Clock

Input the USB clock (UCLK) before setting the register in this module.

20.10.8 Using TR Interrupt

Note that the following when using the transfer request interrupt (TR interrupt) for interrupt-IN transfer of EP0i/EP2/EP3.

The TR interrupt flag is set when the IN token is sent from the USB host and there is no data in the FIFO of the EP. However, TR interrupts occur continuously at the timing shown in figure 20.24. Make sure that no malfunction occurs in these cases.

Note: This module checks NAK acknowledgement if there is no data in the FIFO of the EP when receiving the IN token. However the TR interrupt flag is set after transmitting the NAK handshake. Therefore, when writing the USBTRG/PKTE bit is later than the next IN token, the TR interrupt flag is set again.

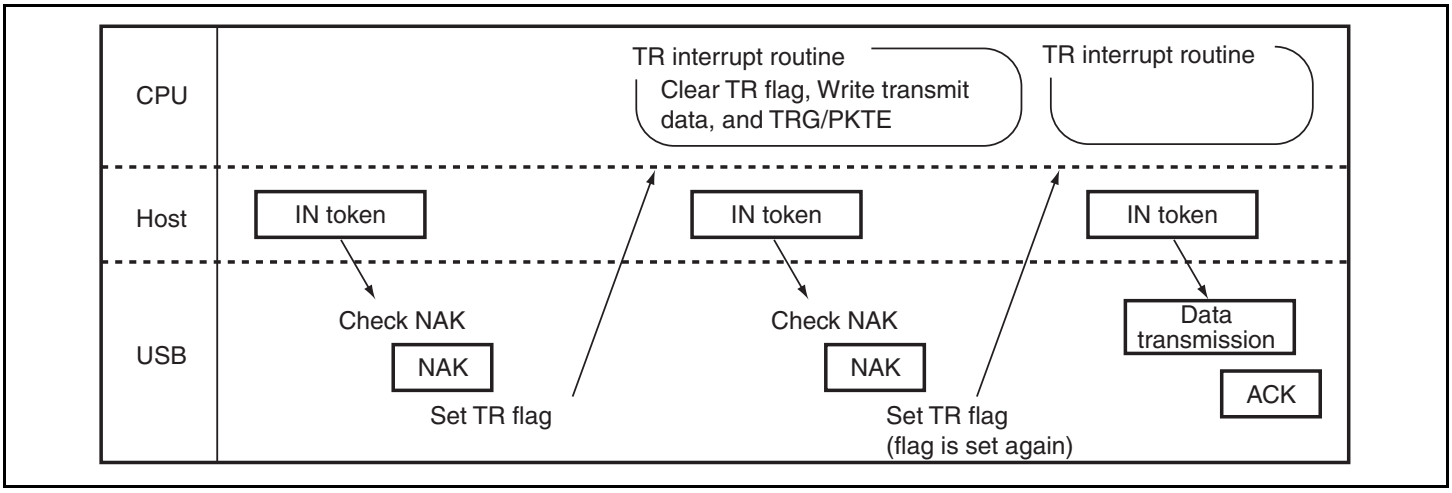


Figure 20.24 Timing for Setting the TR Interrupt Flag

Section 21 A/D Converter

This LSI includes a 10-bit successive-approximation A/D converter allowing selection of up to eight analog input channels.

The A/D converter is composed of two independent modules, A/D0 and A/D1.

21.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Eight input channels (4 channels × 2)
- High-speed conversion
  - Conversion time: maximum 4.4 µs per channel (in single mode, 146-state conversion (Typ.), Pϕ = 33 MHz operation)
- Three conversion modes
  - Single mode: A/D conversion on one channel
  - Multi mode: A/D conversion on one to four channels
  - Scan mode: Continuous A/D conversion on one to four channels
- Conversion can be carried out simultaneously on two channels.
- Two conversion start methods
  - Software or timer conversion start trigger (MTU) can be selected
- Eight 16-bit data registers
  - A/D conversion results are transferred for storage into 16-bit data registers corresponding to the channels.
- Sample-and-hold function
- A/D interrupt requested at the end of conversion
  - An A/D conversion end interrupt (ADI0, ADI1) request can be generated on completion of A/D conversion.
  - The DMAC can be activated by A/D conversion end.

21.1.1 Block Diagram

Figure 21.1 shows a block diagram of the A/D converter.

AVcc and AVss for both A/D modules are common pins in the chip.

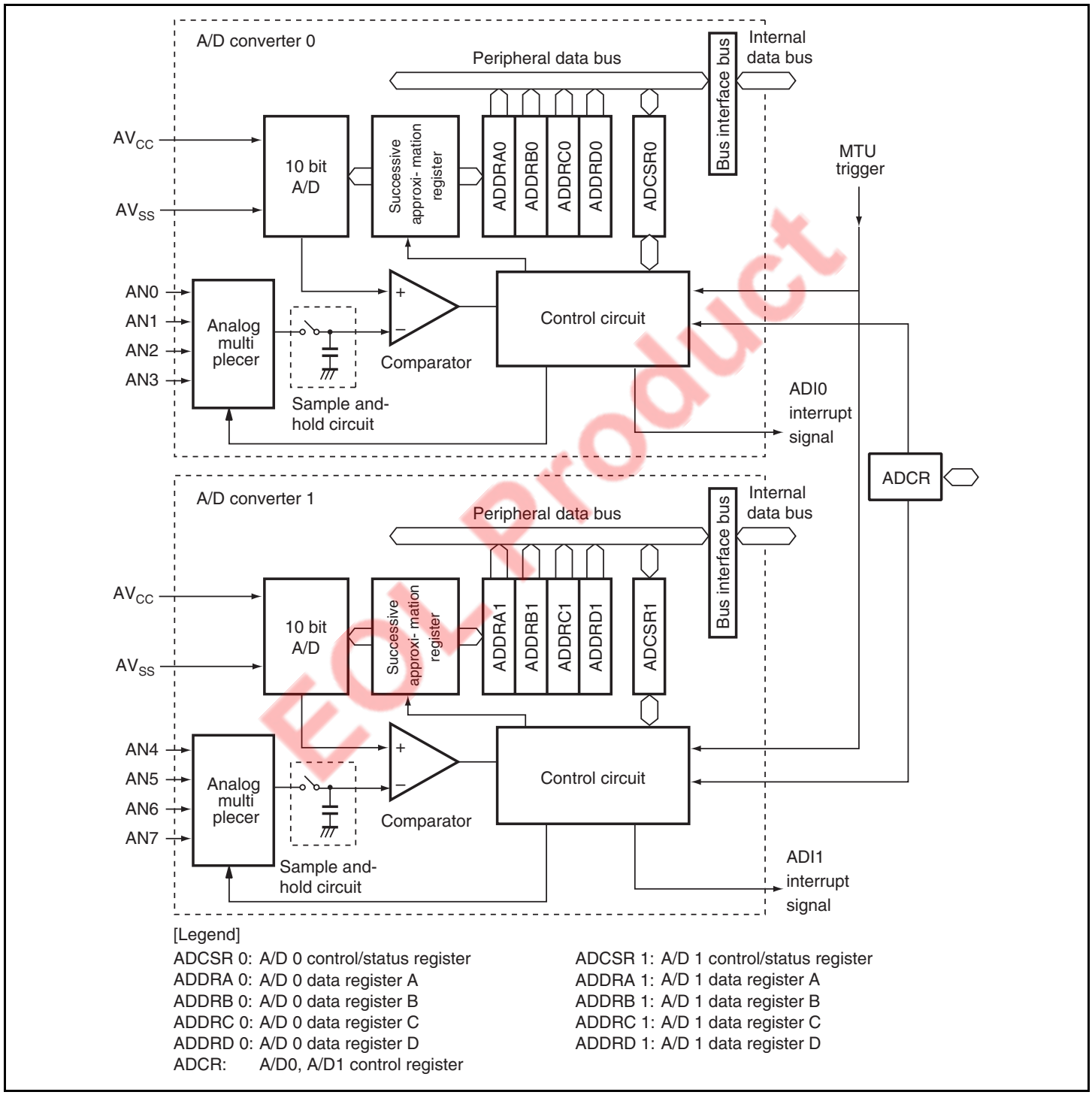


Figure 21.1 Block Diagram of A/D Converter

21.1.2 Input Pins

Table 21.1 summarizes the A/D converter's input pins. The eight analog input pins are divided into two groups: A/D0 (AN0 to AN3), and A/D1 (AN4 to AN7). AV_{CC} and AV_{SS} are the power supply inputs for the analog circuits in the A/D converter. AV_{CC} also functions as the A/D converter reference voltage pin. AV_{SS} also functions as the A/D converter reference ground pin.

Table 21.1 A/D Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AVcc	Input	Analog power supply and reference voltage for A/D conversion
Analog ground pin	AVss	Input	Analog ground and reference ground for A/D conversion
Analog input pin 0	AN0	Input	A/D0 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	A/D1 analog inputs
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	

21.1.3 Register Configuration

The A/D converter's registers are summarized below.

- A/D0 data register A (ADDRA0)
- A/D0 data register B (ADDRB0)
- A/D0 data register C (ADDRC0)
- A/D0 data register D (ADDRD0)
- A/D0 control/status register (ADCSR0)
- A/D1 data register A (ADDRA1)
- A/D1 data register B (ADDRB1)
- A/D1 data register C (ADDRC1)
- A/D1 data register D (ADDRD1)
- A/D1 control/status register (ADCSR1)
- A/D0 A/D1 control register (ADCR)

21.2 Register Descriptions

21.2.1 A/D Data Registers A to D (ADDRA0 to ADDRD0, ADDRA1 to ADDRD1)

The eight A/D data registers (ADDRA0 to ADDRD0, ADDRA1 to ADDRD1) are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D data register corresponding to the selected channel. The 10 bits of the result are stored in the upper bits (bits 15 to 6) of the A/D data register. Bits 5 to 0 of an A/D data register are reserved bits that are always read as 0. Table 21.2 indicates the pairings of analog input channels and A/D data registers.

The A/D data registers are initialized to H'0000 by a power-on reset and in standby mode.

Table 21.2 Analog Input Channels and A/D Data Registers

Analog Input Channel	A/D Data Register	Module
AN0	ADDRA0	A/D0
AN1	ADDRB0	
AN2	ADDRC0	
AN3	ADDRD0	
AN4	ADDRA1	A/D1
AN5	ADDRB1	
AN6	ADDRC1	
AN7	ADDRD1	

21.2.2 A/D Control/Status Registers (ADCSR0, ADCSR1)

ADCSR is a 16-bit readable/writable register that selects the mode, controls the A/D converter, and enable or disable starting of A/D conversion by external trigger input.

ADCSR is initialized to H'0040 by a power-on reset and in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	ADF	0	R/(W)*	A/D End Flag Indicates the end of A/D conversion. [Clearing conditions] • Cleared by reading ADF while ADF = 1, then writing 0 to ADF • Cleared when DMAC is activated by ADI interrupt and ADDR is read [Setting conditions] • Single mode: A/D conversion ends • Multi mode: A/D conversion ends cycling through the selected channels • Scan mode: A/D conversion ends cycling through the selected channels

Bit	Bit Name	Initial Value	R/W	Description
14	ADIE	0	R/W	A/D Interrupt Enable Enables or disables the interrupt (ADI) requested at the end of A/D conversion. Set the ADIE bit while A/D conversion is not being made. 0: A/D end interrupt request (ADI) is disabled 1: A/D end interrupt request (ADI) is enabled
13	ADST	0	R/W	A/D Start Starts or stops A/D conversion. The ADST bit remains set to 1 during A/D conversion. 0: A/D conversion is stopped 1: A/D conversion is started Single mode: A/D conversion starts; ADST is automatically cleared to 0 when conversion ends on all selected channels Multi mode: A/D conversion starts; when conversion is completed cycling through the selected channels, ADST is automatically cleared Scan mode: A/D conversion starts and continues, A/D conversion is continuously performed until ADST is cleared to 0 by software, by a power-on reset, or by a transition to standby mode
12	DMASL	0	R/W	DMAC Select Selects an interrupt due to the end of A/D conversion or activation of the DMAC. Set the DMASL bit while A/D conversion is not being made. 0: An interrupt by the end of A/D conversion is selected 1: Activation of the DMAC by the end of A/D conversion is selected
11	TRGE	0	R/W	A/D Trigger Enable This bit enables or disables starting of A/D conversion by MTU or CSL trigger. 0: Start of A/D conversion by MTU or CSL trigger input is disabled 1: A/D conversion is started MTU or CSL trigger input

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	CKS1	0	R/W	Clock Select
6	CKS0	1	R/W	Selects the A/D conversion time. Clear the ADST bit to 0 before changing the conversion time. 00: Conversion time = 151 states (maximum) clock = Pϕ/4 01: Conversion time = 285 states (maximum) clock = Pϕ/8 10: Conversion time = 545 states (maximum) clock = Pϕ/16 11: Reserved
5	MULTI1	0	R/W	These bits select single mode, multi mode, or scan mode. 00: Single mode 01: Reserved (setting prohibited) 10: Multi mode 11: Scan mode
4	MULTI0	0	R/W	
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CH1	0	R/W	Channel Select
0	CH0	0	R/W	These bits and the MULTI bit select the analog input channels. Clear the ADST bit to 0 before changing the channel selection. • In the case of ADCSR0 (A/D0) Single mode / Multi mode or scan mode 00: AN0 / AN0 01: AN1 / AN0, AN1 10: AN2 / AN0 to AN2 11: AN3 / AN0 to AN3 • In the case of ADSCR1 (A/D1) Single mode / Multi mode or scan mode 00: AN4 / AN4 01: AN5 / AN4, AN5 10: AN6 / AN4 to AN6 11: AN7 / AN4 to AN7

Note: * Clear this bit by writing 0.

21.2.3 A/D0, A/D1 Control Register (ADCR)

ADCR is a 16-bit readable/writable register that selects the simultaneous sampling of two channels. See section 21.3.4 Simultaneous Sampling Operation, for details on simultaneous sampling.

ADCR is initialized to H'0000 by a power-on reset and in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	DSMP	0	R/W	Selects A/D0 or A/D1 simultaneous sampling. Starts simultaneous sampling of two channels when the DSMP bit set to 1. The DSMP bit remains set to 1 during A/D conversion. DSMP is automatically cleared to 0 when conversion ends on all selected channels by each one mode. Note: Set the ADCSR registers before DSMP bit set.
14 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

21.3 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has three operating modes: single mode, multi mode, and scan mode.

21.3.1 Single Mode

Single mode should be selected when only one A/D conversion on one channel is required. A/D conversion starts when the ADST bit is set to 1 by software. The ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 when conversion ends.

When conversion ends the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is requested at this time. To clear the ADF flag to 0, first read ADF, then write 0 to ADF.

When the mode or analog input channel must be switched during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the mode or channel is changed.

Typical operations when channel 1 (AN1) is selected in single mode are described next.

Figure 21.2 shows a timing diagram for this example.

1. Single mode is selected (MULTI = 0), input channel AN1 is selected (CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred into ADDRB0. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1, ADIE = 1, and DMSL = 0 an ADI0 interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADF, and then writes 0 to the ADF flag.
6. The routine reads and processes the conversion result (ADDRB0).
7. Execution of the A/D interrupts handling routine ends. Then, when the ADST bit is set to 1, A/D conversion starts and steps 2 to 7 are executed.

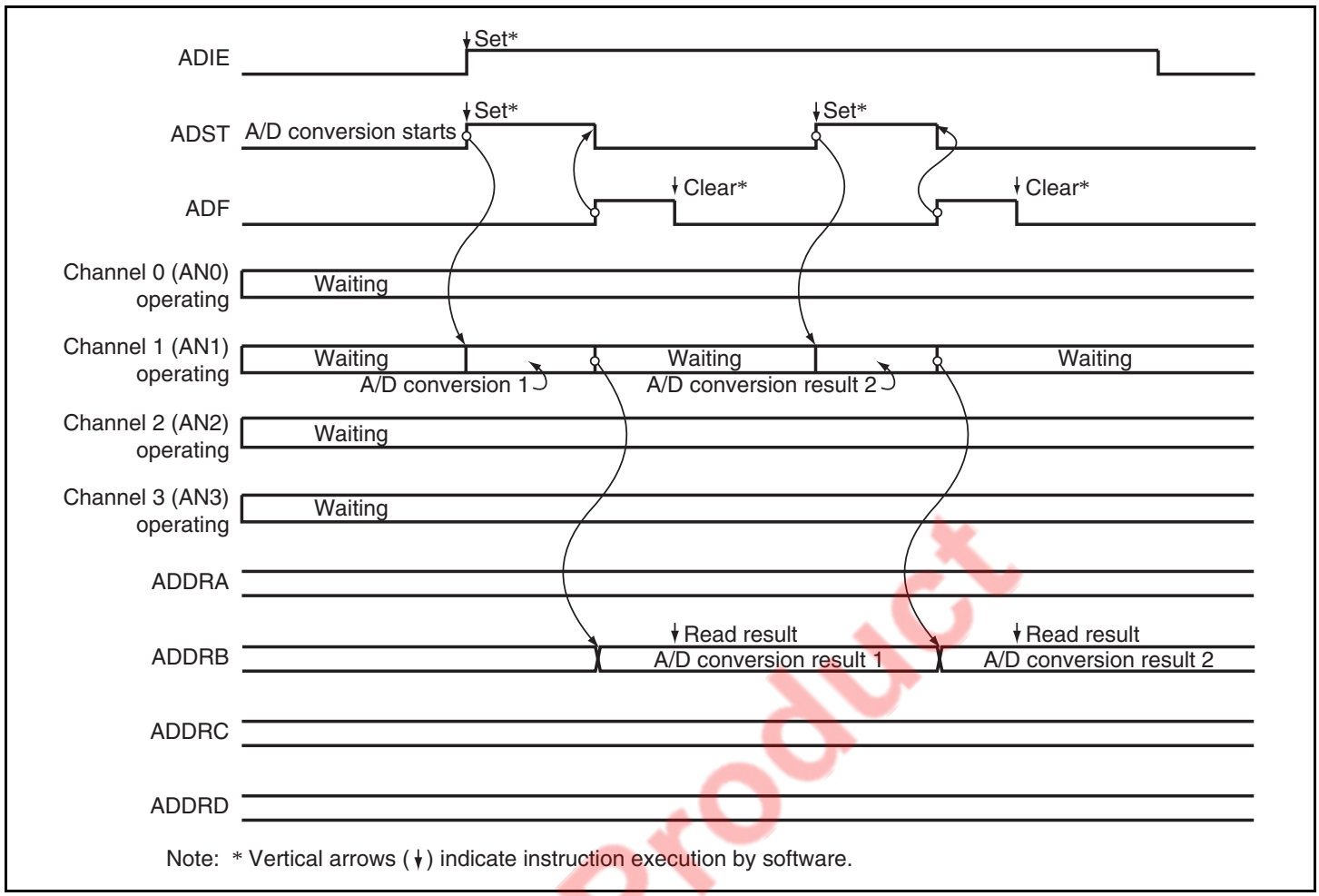


Figure 21.2 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

21.3.2 Multi Mode

Multi mode should be selected when performing A/D conversions on one or more channels. When the ADST bit is set to 1 by software, A/D conversion starts on the first channel in the group (A/D0 when AN0, A/D1 when AN4). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN1 or AN5) starts immediately. When A/D conversions end on the selected channels, the ADST bit is cleared to 0. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel selection must be changed during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels in A/D0 (AN0 to AN2) are selected in multi mode are described next. Figure 21.3 shows a timing diagram for this example.

1. Multi mode is selected (MULTI = 1), channel group A/D0 is selected, analog input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. When A/D conversion of the first channel (AN0) is completed, the result is transferred into ADDRA0.
3. Next, conversion of the second channel (AN1) starts automatically.
4. Conversion proceeds in the same way through the third channel (AN2).
5. When conversion of all selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and ADST bit is cleared to 0. If the ADIE bit is set to 1, an ADI0 interrupt is requested at this time.

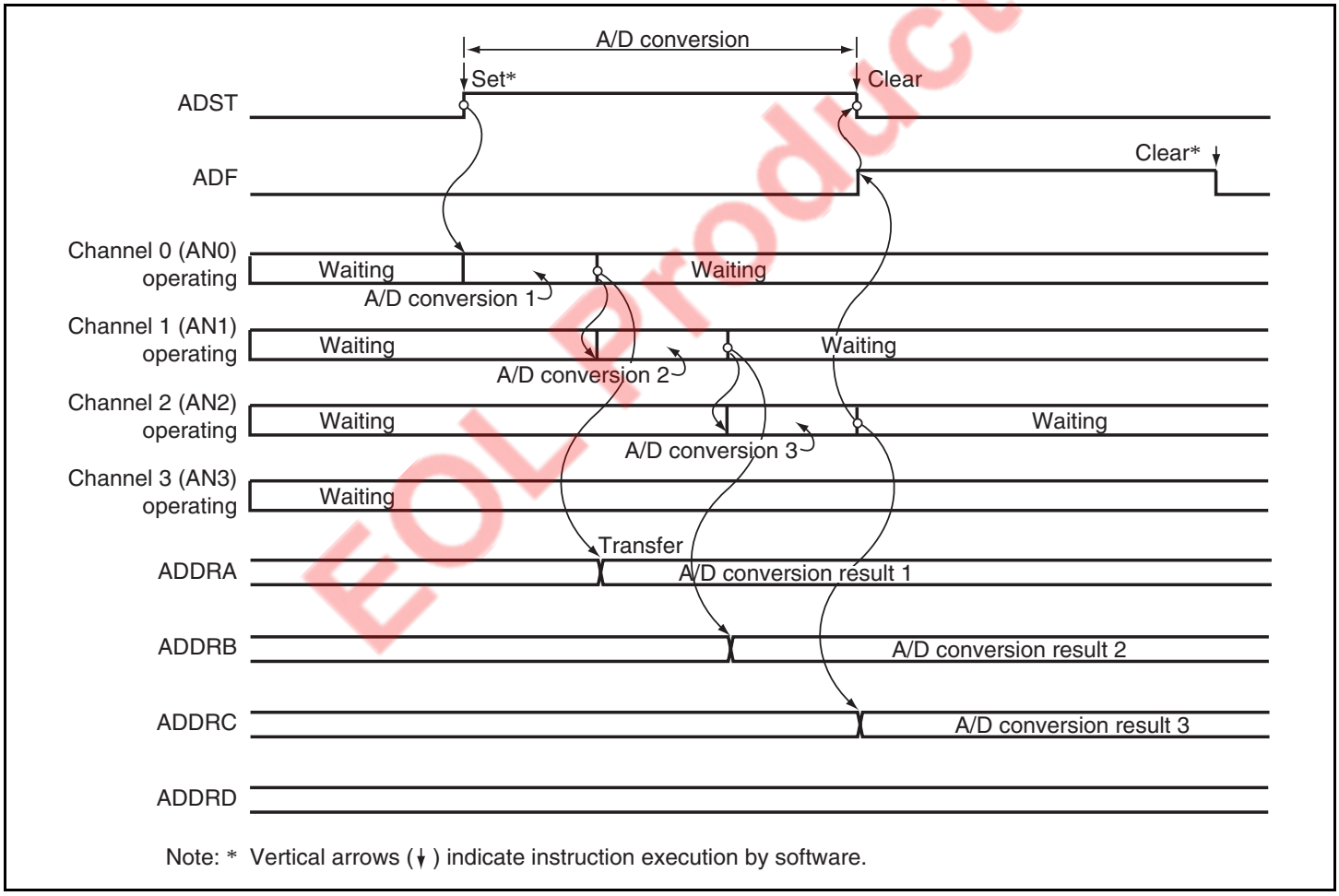


Figure 21.3 Example of A/D Converter Operation (Multi Mode, Channels AN0 to AN2 Selected)

21.3.3 Scan Mode

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit in the A/D control/status register (ADCSR0 or ADCSR1) is set to 1 by software, A/D conversion starts on the first channel in the group (A/D0 when AN0, A/D1 when AN4). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN1 or AN5) starts immediately. A/D conversion continues cyclically on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels (AN0 to AN2) are selected in scan mode are described next. Figure 21.4 shows a timing diagram for this example.

1. Scan modes are selected, analog input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. When A/D conversion of the first channel (AN0) is completed, the result is transferred into ADDRA0.
3. Next, conversion of the second channel (AN1) starts automatically.
4. Conversion proceeds in the same way through the third channel (AN2).
5. When conversion of all the selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to 1, an ADI0 interrupt is requested at this time.
6. The ADST bit is not cleared automatically. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When steps 2 to 4 are repeated, the ADF flag is keep to 1. When the ADST bit is cleared to 0, A/D conversion stops. The ADF bit cleared by reading ADF while ADF=1, then writing 0 to ADF.
7. If the ADIE bit is set to 1 and the ADF flag is set to 1 in steps 2 to 4 are repeated, an ADI0 interrupt is requested ad all times. When an ADI0 interrupt is requested at conversion ends of all the selected channels, the ADF bit is cleared to 0 after an ADI0 interrupt is requested.

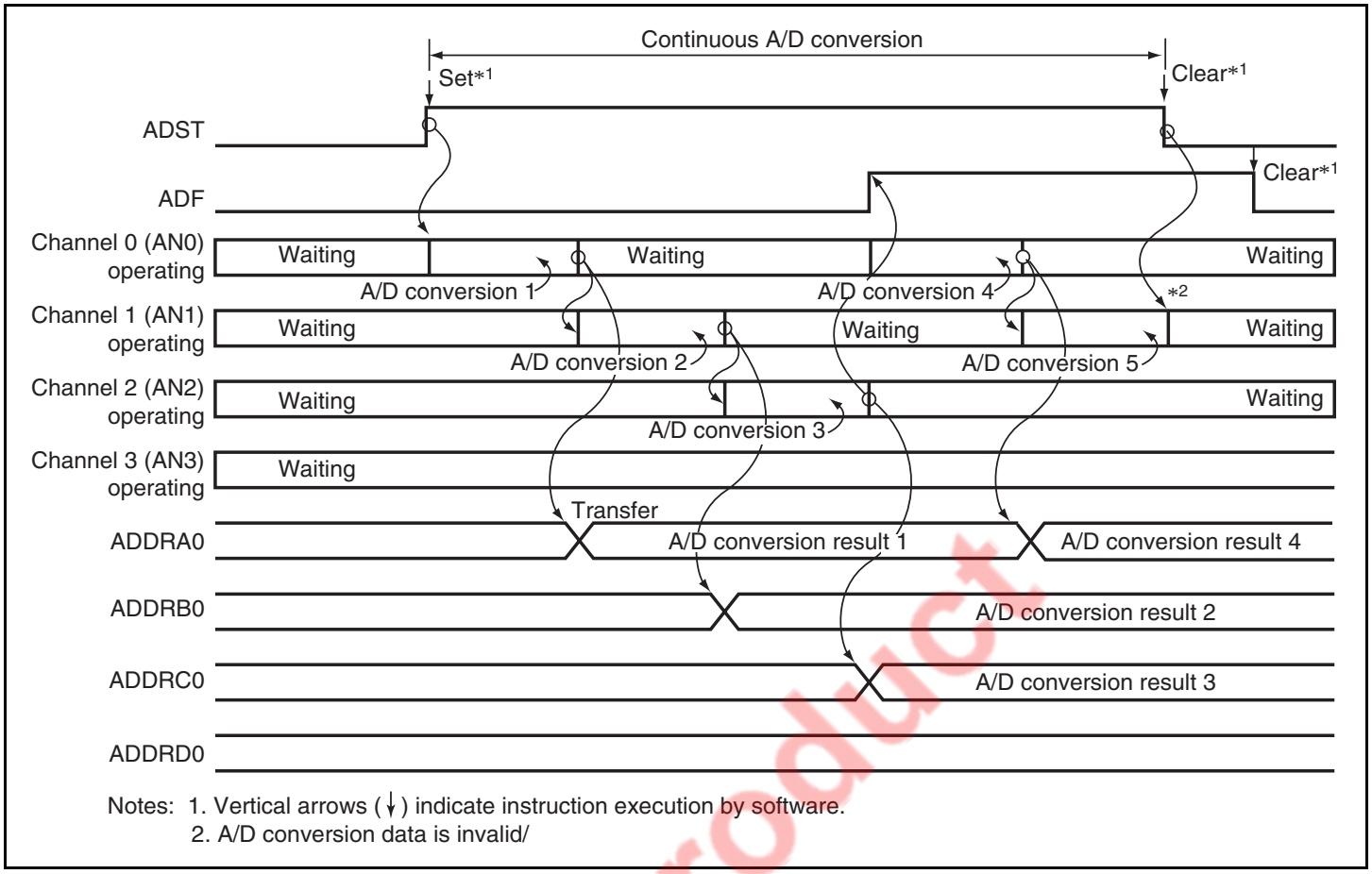


Figure 21.4 Example of A/D Converter Operation (Scan Mode, Channels AN0 to AN2 Selected)

21.3.4 Simultaneous Sampling Operation

With simultaneous sampling, conversion is conducted with sampling of the input voltages on two channels (channel in A/D0 and channel in A/D1) at the same time. Simultaneous sampling is valid in single mode and multi mode and scan mode. Channels for sampling are determined by the CH1 and CH0 bits of the ADCSR0 or ADCSR1.

Procedure for setting simultaneous sampling is shown the next. Select the ADCSR registers (conversion mode and input channels and conversion time), and then starts simultaneous sampling of two channels when the DSMP bit set to 1. When DSMP bit set to 1 during A/D conversion, not to start A/D conversion again. When the ADST bit is set, A/D conversion stops. The timing diagrams for simultaneous sampling are the same as for single mode and multi mode and scan mode.

21.3.5 A/D Converter Activation by MTU

The A/D converter can be independently activated by an A/D conversion request from the MTU or CSL.

To activate the A/D converter by the MTU, set the A/D trigger enable bit (TRGE). After this bit setting has been made, the ADST bit in ADCSR is automatically set to 1 and A/D conversion is started when an A/D conversion request from the MTU occurs. If the TRGE bit in both ADCSR0 and ADCSR1 is set to 1, starts simultaneous sampling of two channels. Channels for sampling are determined by the CH1 and CH0 bits of the ADCSR0 or ADCSR1. The timing from setting of the ADST bit until the start of A/D conversion is the same as when 1 is written to the ADST bit by software.

21.3.6 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at a time t_D after the ADST bit is set to 1, then starts conversion. Figure 21.5 shows the A/D conversion timing. Table 21.3 indicates the A/D conversion time.

As indicated in figure 21.5, the A/D conversion time includes t_D and the input sampling time. The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 21.3.

In multi mode and scan mode, the values given in table 21.3 apply to the first conversion. In the second and subsequent conversions time is the values given in table21.4.

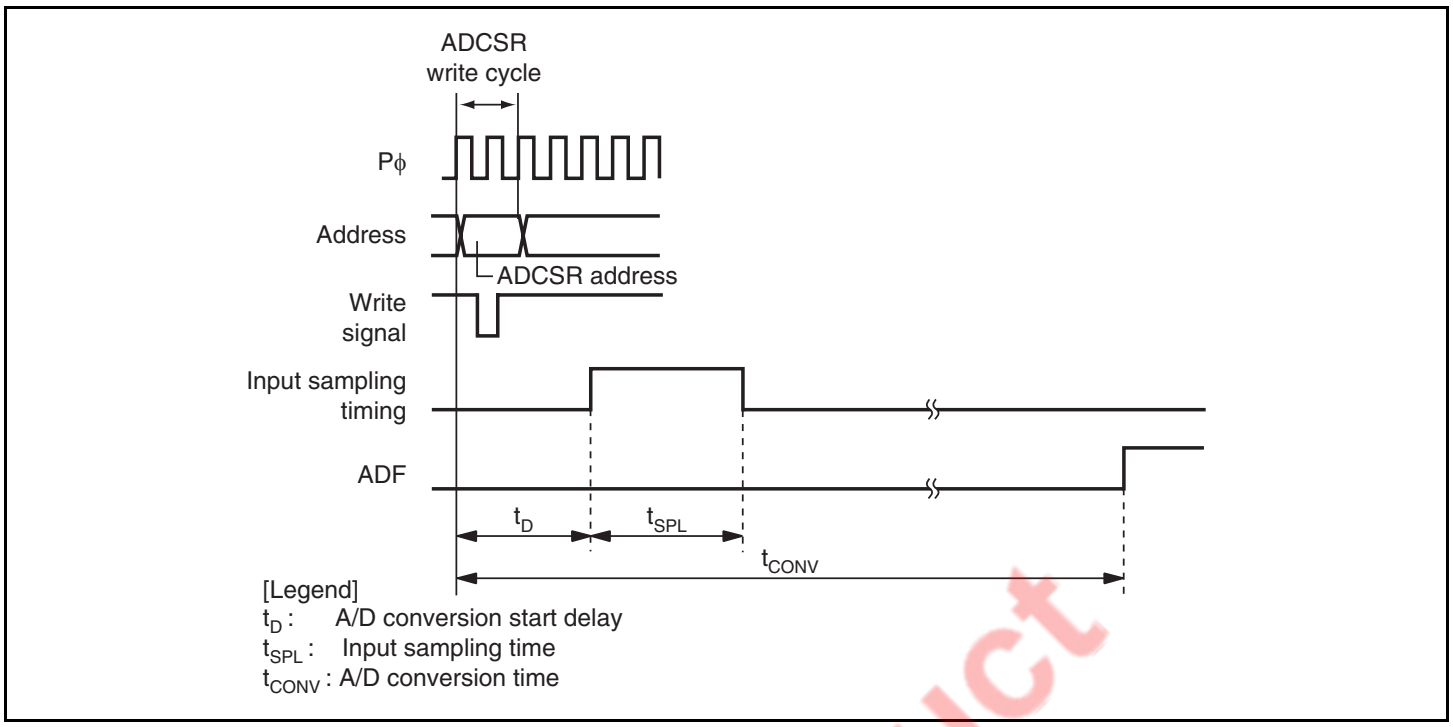


Figure 21.5 A/D Conversion Timing

Table 21.3 A/D Conversion Time (Single Mode)

		CKS1 = 1, CKS0 = 1			CKS1 = 1, CKS0 = 1			CKS1 = 1, CKS0 = 1		
	Symbol	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.
A/D conversion start delay	t_D	18	—	21	10	—	13	6	—	9
Input sampling time	t_{SPL}	—	129	—	—	65	—	—	33	—
A/D conversion time	t_{CONV}	535	—	545	275	—	285	141	—	151

Note: Values in the table are numbers of states (t_{cyc}).

Table 21.4 A/D Conversion Time (Multi Mode and Scan Mode)

CKS1	CKS0	Conversion Time (t_{cyc})
0	0	128 (constant)
	1	256 (constant)
1	0	512 (constant)
	1	Reserved

21.4 Interrupt and DMAC Transfer Request

The A/D converter generates an interrupt (ADI0 and ADI1) or DMAC activation signal at the end of A/D conversion. These requests are enabled or disabled by the ADIE bit or the DMASL bit in ADCSR.

When the DMAC is activated by an ADI interrupt, the ADF bit in the A/D control/status register (ADCSR0 and ADCSR1) is automatically cleared to 0 when an A/D register is accessed.

Table 21.5 Interrupt and DMAC Transfer Request

ADIE Bit	DMASL Bit	Interrupt	DMAC Transfer Request
0	0	Disabled	Disabled
	1	Disabled	Enabled
1	0	Enabled	Disabled
	1	Disabled	Enabled

21.5 Definitions of A/D Conversion Accuracy

The A/D converter compares an analog value input from an analog input channel with its analog reference value and converts it to 10-bit digital data. The absolute accuracy of this A/D conversion is the deviation between the input analog value and the output digital value. It includes the following errors:

- Offset error
- Full-scale error
- Quantization error
- Nonlinearity error

These four error quantities are explained below with reference to figure 21.6. In the figure, the 10 bits of the A/D converter have been simplified to 3 bits.

Offset error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the minimum (zero voltage) 0000000000 (000 in the figure) to 000000001 (001 in the figure)(figure 21.6, item (1)). Full-scale error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the 1111111110 (110 in the figure) to the maximum 1111111111 (111 in the figure)(figure 21.6, item (2)). Quantization error is the intrinsic error of the A/D converter and is expressed as 1/2 LSB (figure 21.6, item (3)). Nonlinearity error is the deviation between actual and ideal A/D conversion characteristics between zero voltage and full-scale voltage (figure 21.6, item (4)). Note that it does not include offset, full-scale, or quantization error.

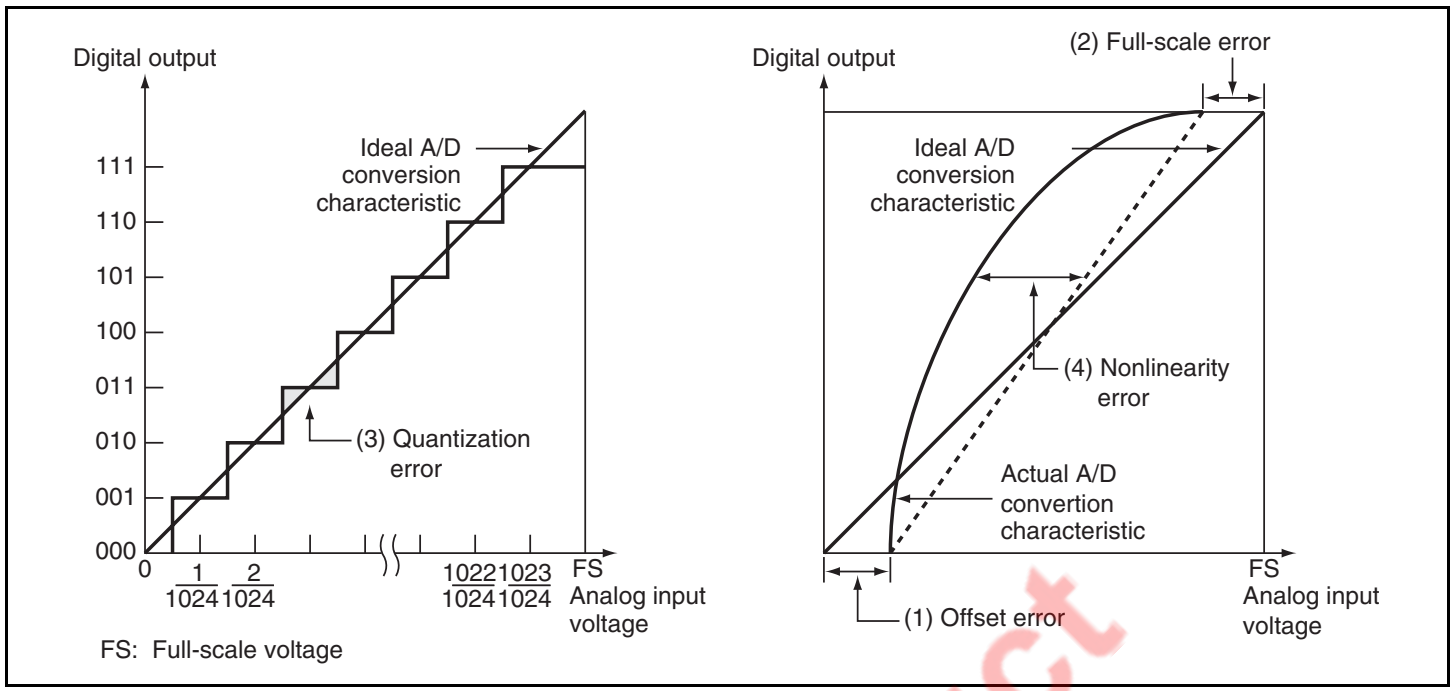


Figure 21.6 Definitions of A/D Conversion Accuracy

21.6 Usage Notes

When using the A/D converter, note the following points.

21.6.1 Setting Analog Input Voltage

Permanent damage to the LSI may result if the following voltage ranges are exceeded.

1. Analog input range: During A/D conversion, voltages on the analog input pins ANn should not go beyond the following range: AVss ≤ ANn ≤ AVcc (n = 0 to 7).
2. AVcc and AVss input voltages: Input voltages AVcc and AVss should be VccQ – 0.2 V ≤ AVcc ≤ VccQ and AVss = Vss. Do not leave the AVcc and AVss pins open when the A/D converter is not in use and during periods in standby mode; in these situations, connect AVcc to the power supply (VccQ) and AVss to the ground (VssQ).

21.6.2 Processing of Analog Input Pins

To prevent damage from voltage surges at the analog input pins (AN0 to AN7), connect an input protection circuit like the one shown in figure 21.7. The circuit shown also includes an RC filter to suppress noise. This circuit is shown as an example; the circuit constants should be selected according to actual application conditions. Section 25.4, A/D Converter Characteristics in section 25, Electrical Characteristics shows the analog input pin specifications and figure 21.8 shows an equivalent circuit diagram of the analog input ports.

21.6.3 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion precision is guaranteed for an input signal for which the signal source impedance is 5kΩ 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 5kΩ, charging may be insufficient and it may not be possible to guarantee A/D conversion precision. However, for A/D conversion in single mode with a large capacitance provided externally for A/D conversion in single mode, the input load will essentially comprise only the internal input resistance of 3kΩ, and the signal source impedance is ignored. However, as 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., 5mV/µs or greater) (see figure 21.9). When converting a high-speed analog signal, a low-impedance buffer should be inserted.

21.6.4 Influences on Absolute Precision

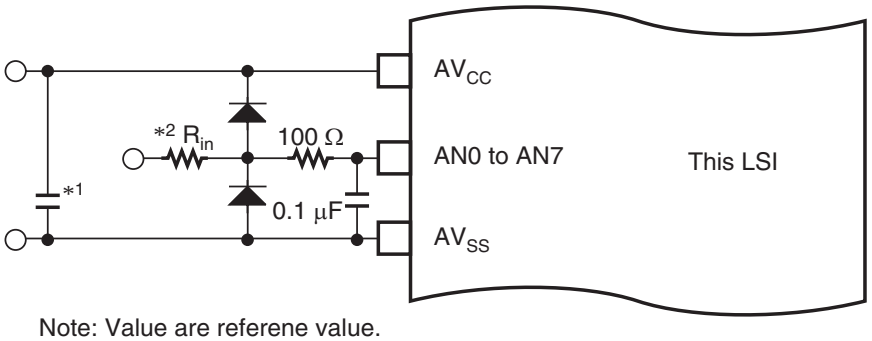
Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AVss.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board (i.e., acting as antennas).

21.6.5 Stop during A/D Conversion

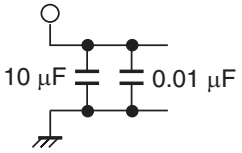
When A/D conversion is stopped with a program during A/D conversion, see the following notes.

1. In single mode, A/D conversion cannot be stopped with a program during A/D conversion.
2. In multi mode or scan mode, when A/D conversion is stopped with a program during A/D conversion, write 0 only to the ADST bit. If the ADST bit and the other bits are set simultaneously, the operation of A/D conversion cannot be guaranteed.
3. In multi mode or scan mode, when A/D conversion is stopped during A/D conversion, write 0 to the ADST bit. And then the ADST bit is set 1 after the time which is longer than the A/D conversion time has elapsed.
4. When the value of the ADST bit is changed, the period which is longer then one clock cycle selected by the CKS[1:0] bits in ADCSR0 and ADCSR1 must be kept because the ADST bit is sampling by the clock cycle selected by the setting of the CKS bits. If the period until the next change of the ADST bit is shorter, that change may not be detected correctly.



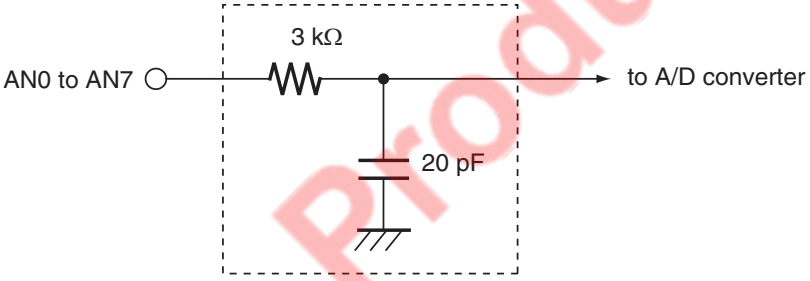
Note: Value are referene value.

1.



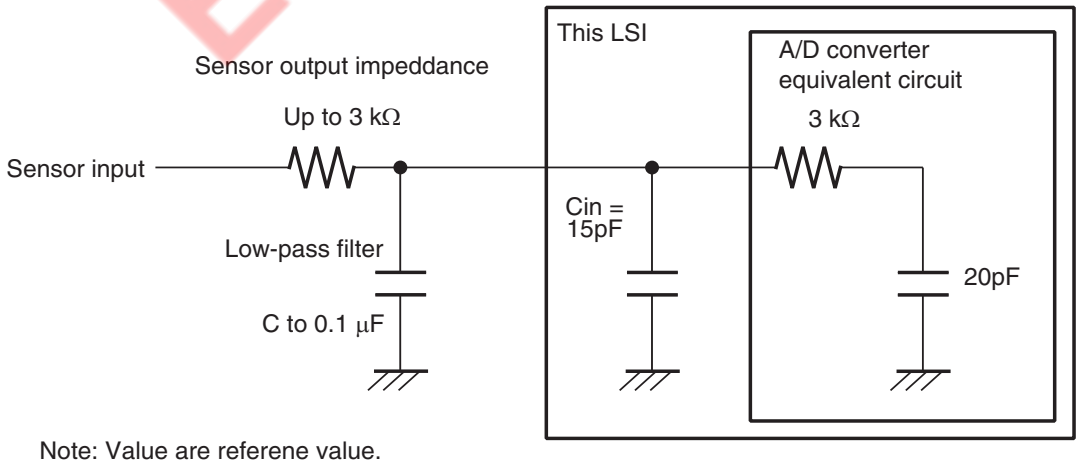
2. R_{in} : input impedance

Figure 21.7 Example of Analog Input Protection Circuit



Note: Value are referene value.

Figure 21.8 Analog Input Pin Equivalent Circuit



Note: Value are referene value.

Figure 21.9 Example of Analog Input Circuit

Section 22 Pin Function Controller (PFC)

The pin function controller (PFC) is composed of registers for selecting the function of multiplexed pins and the input/output direction. The pin function and input/output direction can be selected for each pin individually without regard to the operating mode of the chip. Table 22.1 lists the multiplexed pins.

Table 22.1 List of Multiplexed Pins

Port	Port Function (Related Module)	Other Function (Related Module)
A	PTA14 input/output (port)	A25 output (address bus)
	PTA13 input/output (port)	A24 output (address bus)
	PTA12 input/output (port)	A23 output (address bus)
	PTA11 input/output (port)	A22 output (address bus)
	PTA10 input/output (port)	A21 output (address bus)
	PTA9 input/output (port)	A20 output (address bus)
	PTA8 input/output (port)	A19 output (address bus)
	PTA7 input/output (port)	$\overline{\text{RASU}}$ output (BSC)
	PTA6 input/output (port)	$\overline{\text{RASL}}$ output (BSC)
	PTA5 input/output (port)	$\overline{\text{CASU}}$ output (BSC)
	PTA4 input/output (port)	$\overline{\text{CASL}}$ output (BSC)
	PTA3 input/output (port)	$\overline{\text{CS3}}$ output (BSC)
	PTA2 input/output (port)	$\overline{\text{CS2}}$ output (BSC)
	PTA1 input/output (port)	CKE output (BSC)
	PTA0 input/output (port)	A0 output (address bus)
B	PTB8 input/output (port)	DPLS input (USB)
	PTB7 input/output (port)	DMNS input (USB)
	PTB6 input/output (port)	TXDPLS output (USB)
	PTB5 input/output (port)	TXDMNS output (USB)
	PTB4 input/output (port)	TXENL output (USB)
	PTB3 input/output (port)	XVDATA input (USB)
	PTB2 input/output (port)	SUSPND output (USB)
	PTB1 input/output (port)	VBUS input (USB)
	PTB0 input/output (port)	UCLK input (USB)

Port	Port Function (Related Module)	Other Function (Related Module)
C	PTC15 input/output (port)	STATUS1 output (CPG)
	PTC14 input/output (port)	STATUS0 output (CPG)
	PTC13 input/output (port)	$\overline{\text{ASEBRKAK}}$ output (CPU)
	PTC12 input/output (port)	$\overline{\text{DACK1}}$ output (DMAC)
	PTC11 input/output (port)	$\overline{\text{DACK0}}$ output (DMAC)
	PTC10 input/output (port)	$\overline{\text{DREQ1}}$ input (DMAC)
	PTC9 input/output (port)	$\overline{\text{DREQ0}}$ input (DMAC)
	PTC8 input/output (port)	$\overline{\text{TEND}}$ output (DMAC)
	PTC7 input/output (port)	$\overline{\text{BACK}}$ output (BSC)
	PTC6 input/output (port)	$\overline{\text{BREQ}}$ input (BSC)
	PTC5 input/output (port)	$\overline{\text{FRAME}}$ output (BSC)
	PTC4 input/output (port)	$\overline{\text{CS6B}}$ output (BSC)
	PTC3 input/output (port)	$\overline{\text{CS6A}}$ output (BSC)
	PTC2 input/output (port)	$\overline{\text{CS5B}}$ output (BSC)
	PTC1 input/output (port)	$\overline{\text{CS5A}}$ output (BSC)
	PTC0 input/output (port)	$\overline{\text{CS4}}$ output (BSC)
D	PTD15 input/output (port)	D31 input/output (data bus)
	PTD14 input/output (port)	D30 input/output (data bus)
	PTD13 input/output (port)	D29 input/output (data bus)
	PTD12 input/output (port)	D28 input/output (data bus)
	PTD11 input/output (port)	D27 input/output (data bus)
	PTD10 input/output (port)	D26 input/output (data bus)
	PTD9 input/output (port)	D25 input/output (data bus)
	PTD8 input/output (port)	D24 input/output (data bus)
	PTD7 input/output (port)	D23 input/output (data bus)
	PTD6 input/output (port)	D22 input/output (data bus)
	PTD5 input/output (port)	D21 input/output (data bus)
	PTD4 input/output (port)	D20 input/output (data bus)
	PTD3 input/output (port)	D19 input/output (data bus)
	PTD2 input/output (port)	D18 input/output (data bus)
	PTD1 input/output (port)	D17 input/output (data bus)
	PTD0 input/output (port)	D16 input/output (data bus)

Port	Port Function (Related Module)	Other Function (Related Module)
E	PTE15 input/output (port)	TIOC0A input/output (MTU)
	PTE14 input/output (port)	TIOC0B input/output (MTU)
	PTE13 input/output (port)	TIOC0C input/output (MTU)
	PTE12 input/output (port)	TIOC0D input/output (MTU)
	PTE11 input/output (port)	TIOC1A input/output (MTU)
	PTE10 input/output (port)	TIOC1B input/output (MTU)
	PTE9 input/output (port)	TIOC2A input/output (MTU)
	PTE8 input/output (port)	TIOC2B input/output (MTU)
	PTE7 input/output (port)	TIOC3A input/output (MTU)
	PTE6 input/output (port)	TIOC3B input/output (MTU)
	PTE5 input/output (port)	TIOC3C input/output (MTU)
	PTE4 input/output (port)	TIOC3D input/output (MTU)
	PTE3 input/output (port)	TIOC4A input/output (MTU)
	PTE2 input/output (port)	TIOC4B input/output (MTU)
	PTE1 input/output (port)	TIOC4C input/output (MTU)
	PTE0 input/output (port)	TIOC4D input/output (MTU)
F	PTF15 input/output (port)	$\overline{\text{POE3}}$ input (MTU)
	PTF14 input/output (port)	$\overline{\text{POE2}}$ input (MTU)
	PTF13 input/output (port)	$\overline{\text{POE1}}$ input (MTU)
	PTF12 input/output (port)	$\overline{\text{POE0}}$ input (MTU)
	PTF11 input/output (port)	TCLKA input (MTU)
	PTF10 input/output (port)	TCLKB input (MTU)
	PTF9 input/output (port)	TCLKC input (MTU)
	PTF8 input/output (port)	TCLKD input (MTU)
	PTF7 input/output (port)	—
	PTF6 input/output (port)	—
	PTF5 input/output (port)	—
	PTF4 input/output (port)	—
	PTF3 input/output (port)	—
	PTF2 input/output (port)	—
	PTF1 input/output (port)	—
	PTF0 input/output (port)	—

Port	Port Function (Related Module)	Other Function (Related Module)
G	PTG13 input/output (port)	—
	PTG12 input/output (port)	—
	PTG11 input/output (port)	—
	PTG10 input/output (port)	SDA input/output (IIC2)
	PTG9 input/output (port)	SDL input/output (IIC2)
	PTG8 input/output (port)	—
	PTG7 input (port)	AN7 input (ADC)
	PTG6 input (port)	AN6 input (ADC)
	PTG5 input (port)	AN5 input (ADC)
	PTG4 input (port)	AN4 input (ADC)
	PTG3 input (port)	AN3 input (ADC)
	PTG2 input (port)	AN2 input (ADC)
	PTG1 input (port)	AN1 input (ADC)
	PTG0 input (port)	AN0 input (ADC)
H	PTH14 input/output (port)	$\overline{\text{RTS2}}$ input/output (SCIF2)
	PTH13 input/output (port)	RXD2 input (SCIF2)
	PTH12 input/output (port)	TXD2 output (SCIF2)
	PTH11 input/output (port)	$\overline{\text{CTS2}}$ input/output (SCIF2)
	PTH10 input/output (port)	SCK2 input/output (SCIF2)
	PTH9 input/output (port)	$\overline{\text{RTS1}}$ input/output (SCIF1)
	PTH8 input/output (port)	RXD1 input (SCIF1)
	PTH7 input/output (port)	TXD1 output (SCIF1)
	PTH6 input/output (port)	$\overline{\text{CTS1}}$ input/output (SCIF)
	PTH5 input/output (port)	SCK1 input/output (SCIF1)
	PTH4 input/output (port)	$\overline{\text{RTS0}}$ input/output (SCIF0)
	PTH3 input/output (port)	RXD0 input (SCIF0)
	PTH2 input/output (port)	TXD0 output (SCIF0)
	PTH1 input/output (port)	$\overline{\text{CTS0}}$ input/output (SCIF0)
	PTH0 input/output (port)	SCK0 input/output (SCIF0)

Port	Port Function (Related Module)	Other Function (Related Module)
J	PTJ12 input/output (port)	$\overline{\text{AUDSYNC}}$ output (AUD)
	PTJ11 input/output (port)	AUDATA3 output (AUD)
	PTJ10 input/output (port)	AUDATA2 output (AUD)
	PTJ9 input/output (port)	AUDATA1 output (AUD)
	PTJ8 input/output (port)	AUDATA0 output (AUD)
	PTJ7 input/output (port)	$\overline{\text{IRQ7}}$ input (INTC)
	PTJ6 input/output (port)	$\overline{\text{IRQ6}}$ input (INTC)
	PTJ5 input/output (port)	$\overline{\text{IRQ5}}$ input (INTC)
	PTJ4 input/output (port)	$\overline{\text{IRQ4}}$ input (INTC)
	PTJ3 input/output (port)	$\overline{\text{IRQ3}}$ input (INTC)
	PTJ2 input/output (port)	$\overline{\text{IRQ2}}$ input (INTC)
	PTJ1 input/output (port)	$\overline{\text{IRQ1}}$ input (INTC)
	PTJ0 input/output (port)	$\overline{\text{IRQ0}}$ input (INTC)

22.1 Register Descriptions

The registers of the pin function controller are shown below.

- Port A control register (PACR)
- Port B control register (PBCR)
- Port C control register (PCCR)
- Port D control register (PDCR)
- Port E control register (PECR)
- Port E I/O register (PEIOR)
- Port E MTU R/W enable register (PEMTURWER)
- Port F control register (PFCR)
- Port G control register (PGCR)
- Port H control register (PHCR)
- Port J control register (PJCR)

22.1.1 Port A Control Register (PACR)

PACR is a 32-bit readable/writable register that selects the pin functions. PACR is initialized to H'00000000 by a power-on reset, and it is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
29	PA14MD2	0	R/W	PAn Mode 2 and 1 The combination of bits PAnMD2 and PAnMD1 (n = 0 to 14) controls the pin functions. 00: Port input 01: Port output 10: Reserved (When set, correct operation cannot be guaranteed.) 11: Other functions (see table 22.1.)
28	PA14MD1	0	R/W	
27	PA13MD2	0	R/W	
26	PA13MD1	0	R/W	
25	PA12MD2	0	R/W	
24	PA12MD1	0	R/W	
23	PA11MD2	0	R/W	
22	PA11MD1	0	R/W	
21	PA10MD2	0	R/W	
20	PA10MD1	0	R/W	
19	PA9MD2	0	R/W	
18	PA9MD1	0	R/W	
17	PA8MD2	0	R/W	
16	PA8MD1	0	R/W	
15	PA7MD2	0	R/W	
14	PA7MD1	0	R/W	
13	PA6MD2	0	R/W	
12	PA6MD1	0	R/W	
11	PA5MD2	0	R/W	
10	PA5MD1	0	R/W	
9	PA4MD2	0	R/W	
8	PA4MD1	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	PA3MD2	0	R/W	PAn Mode 2 and 1
6	PA3MD1	0	R/W	The combination of bits PAnMD2 and PAnMD1 (n = 0 to 14) controls the pin functions.
5	PA2MD2	0	R/W	00: Port input
4	PA2MD1	0	R/W	01: Port output
3	PA1MD2	0	R/W	10: Reserved (When set, correct operation cannot be guaranteed.)
2	PA1MD1	0	R/W	11: Other functions (see table 22.1.)
1	PA0MD2	0	R/W	
0	PA0MD1	0	R/W	

Note: The initial function of the port A is port input after a power-on reset. When ROM with more than 256 kbytes is allocated to space0, strong pull-downs must be prepared on the user board to input 0 to the upper address bits of the ROM immediately after a power-on reset.

22.1.2 Port B Control Register (PBCR)

PBCR is a 32-bit readable/writable register that selects the pin functions. PBCR is initialized to H'00000000 by a power-on reset, and it is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	PB8MD2	0	R/W	PBn Mode 2 and 1 The combination of bits PBnMD2 and PBnMD1 (n = 0 to 8) controls the pin functions. 00: Port input 01: Port output 10: Reserved (When set, correct operation cannot be guaranteed.) 11: Other functions (see table 22.1.)
16	PB8MD1	0	R/W	
15	PB7MD2	0	R/W	
14	PB7MD1	0	R/W	
13	PB6MD2	0	R/W	
12	PB6MD1	0	R/W	
11	PB5MD2	0	R/W	
10	PB5MD1	0	R/W	
9	PB4MD2	0	R/W	
8	PB4MD1	0	R/W	
7	PB3MD2	0	R/W	
6	PB3MD1	0	R/W	
5	PB2MD2	0	R/W	
4	PB2MD1	0	R/W	
3	PB1MD2	0	R/W	
2	PB1MD1	0	R/W	
1	PB0MD2	0	R/W	
0	PB0MD1	0	R/W	

22.1.3 Port C Control Register (PCCR)

PCCR is a 32-bit readable/writable register that selects the pin functions. PCCR is initialized to H'0C000000 by a power-on reset, and it is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
31	PC15MD2	0	R/W	PCn Mode 2 and 1
30	PC15MD1	0	R/W	The combination of bits PCnMD2 and PCnMD1 (n = 0 to 15) controls the pin functions.
29	PC14MD2	0	R/W	00: Port input
28	PC14MD1	0	R/W	01: Port output
27	PC13MD2	1	R/W	10: Reserved (When set, correct operation cannot be guaranteed.)
26	PC13MD1	1	R/W	11: Other functions (see table22.1.)
25	PC12MD2	0	R/W	
24	PC12MD2	0	R/W	
23	PC11MD2	0	R/W	
22	PC11MD1	0	R/W	
21	PC10MD2	0	R/W	
20	PC10MD1	0	R/W	
19	PC9MD2	0	R/W	
18	PC9MD1	0	R/W	
17	PC8MD2	0	R/W	
16	PC8MD1	0	R/W	
15	PC7MD2	0	R/W	
14	PC7MD1	0	R/W	
13	PC6MD2	0	R/W	
12	PC6MD1	0	R/W	
11	PC5MD2	0	R/W	
10	PC5MD1	0	R/W	
9	PC4MD2	0	R/W	
8	PC4MD1	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	PC3MD2	0	R/W	PCn Mode 2 and 1
6	PC3MD1	0	R/W	The combination of bits PCnMD2 and PCnMD1 (n = 0 to 15) controls the pin functions.
5	PC2MD2	0	R/W	00: Port input
4	PC2MD1	0	R/W	01: Port output
3	PC1MD2	0	R/W	10: Reserved (When set, correct operation cannot be guaranteed.)
2	PC1MD1	0	R/W	11: Other functions (see table22.1.)
1	PC0MD2	0	R/W	
0	PC0MD1	0	R/W	

22.1.4 Port D Control Register (PDCR)

PDCR is a 32-bit readable/writable register that selects the pin functions. PDCR is initialized to H'00000000 (MD3 = 0, 16-bit bus width) or H'FFFFFFFF (MD3 = 1, 32-bit bus width) by a power-on reset, and it is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
31	PD15MD2	0/1	R/W	PDn Mode 2 and 1
30	PD15MD1	0/1	R/W	The combination of bits PDnMD2 and PDnMD1 (n = 0 to 15) controls the pin functions.
29	PD14MD2	0/1	R/W	00: Port input
28	PD14MD1	0/1	R/W	01: Port output
27	PD13MD2	0/1	R/W	10: Reserved (When set, correct operation cannot be guaranteed.)
26	PD13MD1	0/1	R/W	11: Other functions (see table22.1.)
25	PD12MD2	0/1	R/W	
24	PD12MD1	0/1	R/W	
23	PD11MD2	0/1	R/W	
22	PD11MD1	0/1	R/W	
21	PD10MD2	0/1	R/W	
20	PD10MD1	0/1	R/W	
19	PD9MD2	0/1	R/W	
18	PD9MD1	0/1	R/W	

Bit	Bit Name	Initial Value	R/W	Description
17	PD8MD2	0/1	R/W	PDn Mode 2 and 1
16	PD8MD1	0/1	R/W	The combination of bits PDnMD2 and PDnMD1 (n = 0 to 15) controls the pin functions.
15	PD7MD2	0/1	R/W	00: Port input
14	PD7MD1	0/1	R/W	01: Port output
13	PD6MD2	0/1	R/W	10: Reserved (When set, correct operation cannot be guaranteed.)
12	PD6MD1	0/1	R/W	11: Other functions (see table22.1.)
11	PD5MD2	0/1	R/W	
10	PD5MD1	0/1	R/W	
9	PD4MD2	0/1	R/W	
8	PD4MD1	0/1	R/W	
7	PD3MD2	0/1	R/W	
6	PD3MD1	0/1	R/W	
5	PD2MD2	0/1	R/W	
4	PD2MD1	0/1	R/W	
3	PD1MD2	0/1	R/W	
2	PD1MD1	0/1	R/W	
1	PD0MD2	0/1	R/W	
0	PD0MD1	0/1	R/W	

22.1.5 Port E Control Register (PECR)

PECR is a 32-bit readable/writable register that selects the pin functions. PECR is initialized to H'00000000 by a power-on reset, and it is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
31	PE15MD2	0	R/W	PEn Mode 2 and 1
30	PE15MD1	0	R/W	The combination of bits PEnMD2 and PEnMD1 (n = 0 to 15) controls the pin functions. 00: Port input 01: Port output 10: Reserved (When set, correct operation cannot be guaranteed.) 11: Other functions (see table 22.1.) When 11 (other functions) is set, the port E I/O register (PEIOR) controls input or output. For details, see section 22.1.6, Port E I/O Register (PEIOR).
29	PE14MD2	0	R/W	
28	PE14MD1	0	R/W	
27	PE13MD2	0	R/W	
26	PE13MD1	0	R/W	
25	PE12MD2	0	R/W	
24	PE12MD1	0	R/W	
23	PE11MD2	0	R/W	
22	PE11MD1	0	R/W	
21	PE10MD2	0	R/W	
20	PE10MD1	0	R/W	
19	PE9MD2	0	R/W	
18	PE9MD1	0	R/W	
17	PE8MD2	0	R/W	
16	PE8MD1	0	R/W	
15	PE7MD2	0	R/W	
14	PE7MD1	0	R/W	
13	PE6MD2	0	R/W	
12	PE6MD1	0	R/W	
11	PE5MD2	0	R/W	
10	PE5MD1	0	R/W	
9	PE4MD2	0	R/W	
8	PE4MD1	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	PE3MD2	0	R/W	PEn Mode 2 and 1
6	PE3MD1	0	R/W	The combination of bits PEnMD2 and PEnMD1 (n = 0 to 15) controls the pin functions.
5	PE2MD2	0	R/W	00: Port input
4	PE2MD1	0	R/W	01: Port output
3	PE1MD2	0	R/W	10: Reserved (When set, correct operation cannot be guaranteed.)
2	PE1MD1	0	R/W	11: Other functions (see table 22.1.)
1	PE0MD2	0	R/W	When 11 (other functions) is set, the port E I/O register (PEIOR) controls input or output. For details, see section 22.1.6, Port E I/O Register (PEIOR).
0	PE0MD1	0	R/W	

22.1.6 Port E I/O Register (PEIOR)

PEIOR is a 16-bit readable/writable register that selects the input/output direction of the port E pins.

The PE15IOR to PE0IOR bits correspond to the PE15/TIOC0A to PE0/TIOC4D pins. PEIOR is valid only when the port E pins function as the TIOC pins of the MTU (other functions). Otherwise, PEIOR is invalid. When the port E pins function as the TIOC pins of the MTU (other functions), setting a bit in PEIOR to 1 sets the pin to output and setting a bit in PEIOR to 0 sets the pin to input. PEIOR is initialized to H'0000 by a power-on reset, and it is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15	PE15IOR	0	R/W	When the port E pins function as the TIOC pins of the MTU (other functions): PEnIOR (n = 0 to 15) controls the input/output direction of the pins. 0: MTU input capture input 1: MTU output compare output PEIOR is invalid when the port E pins function as pins other than the TIOC pins of the MTU.
14	PE14IOR	0	R/W	
13	PE13IOR	0	R/W	
12	PE12IOR	0	R/W	
11	PE11IOR	0	R/W	
10	PE10IOR	0	R/W	
9	PE9IOR	0	R/W	
8	PE8IOR	0	R/W	
7	PE7IOR	0	R/W	
6	PE6IOR	0	R/W	
5	PE5IOR	0	R/W	
4	PE4IOR	0	R/W	
3	PE3IOR	0	R/W	
2	PE2IOR	0	R/W	
1	PE1IOR	0	R/W	
0	PE0IOR	0	R/W	

22.1.7 Port E MTU R/W Enable Register (PEMTURWER)

PEMTURWER is a 16-bit readable/writable register that allows access of the MTU registers. PEMTURWER is initialized to H'0001 by a power-on reset, and it is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	MTURWE	1	R/W	MTURWE allows access of the MTU registers. For details, see section 18, Multi-Function Timer Pulse Unit (MTU). 0: Disables access of the MTU registers. 1: Enables access of the MTU registers.

22.1.8 Port F Control Register (PFCR)

PFCR is a 32-bit readable/writable register that selects the pin functions. PFCR is initialized to H'00000000 by a power-on reset, and it is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
31	PF15MD2	0	R/W	PFn Mode 2 and 1
30	PF15MD1	0	R/W	The combination of bits PFnMD2 and PFnMD1controls the pin functions. (n = 8 to 15) 00: Port input 01: Port output 10: Reserved (When set, correct operation cannot be guaranteed.) 11: Other functions (see table22.1.)
29	PF14MD2	0	R/W	
28	PF14MD1	0	R/W	
27	PF13MD2	0	R/W	
26	PF13MD1	0	R/W	
25	PF12MD2	0	R/W	
24	PF12MD2	0	R/W	
23	PF11MD2	0	R/W	
22	PF11MD2	0	R/W	
21	PF10MD2	0	R/W	
20	PF10MD2	0	R/W	
19	PF9MD2	0	R/W	
18	PF9MD2	0	R/W	
17	PF8MD2	0	R/W	
16	PF8MD2	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
15	PF7MD2	0	R/W	PFn Mode 2,1
14	PF7MD2	0	R/W	The combination of bits PFnMD2 and PFnMD1 controls the pin functions. (n = 0 to 7)
13	PF6MD2	0	R/W	00: Port input
12	PF6MD2	0	R/W	01: Port output
11	PF5MD2	0	R/W	10, 11: Reserved (When set, correct operation cannot be guaranteed.)
10	PF5MD2	0	R/W	
9	PF4MD2	0	R/W	
8	PF4MD2	0	R/W	
7	PF3MD2	0	R/W	
6	PF3MD2	0	R/W	
5	PF2MD2	0	R/W	
4	PF2MD2	0	R/W	
3	PF1MD2	0	R/W	
2	PF1MD2	0	R/W	
1	PF0MD2	0	R/W	
0	PF0MD2	0	R/W	

22.1.9 Port G Control Register (PGCR)

PGCR is a 32-bit readable/writable register that selects the pin functions. PGCR is initialized to H'00000000 by a power-on reset, and it is not initialized by a manual reset, in the standby mode, or in the sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27	PG13MD2	0	R/W	PGn Mode 2 and 1
26	PG13MD1	0	R/W	The combination of bits PGnMD2 and PGnMD1 controls the pin functions. (n = 11 to 13) 00: Port input 01: Port output 10, 11: Reserved (When set, correct operation cannot be guaranteed.)
25	PG12MD2	0	R/W	
24	PG12MD2	0	R/W	
23	PG11MD2	0	R/W	
22	PG11MD2	0	R/W	
21	PG10MD2	0	R/W	PGn Mode 2 and 1
20	PG10MD2	0	R/W	The combination of bits PGnMD2 and PGnMD1 controls the pin functions. (n = 9 and 10) 00: Port input 01: Port output 10: Reserved (When set, correct operation cannot be guaranteed.) 11: Other functions (see table22.1.)
19	PG9MD2	0	R/W	
18	PG9MD2	0	R/W	
17	PG8MD2	0	R/W	PG8 mode 2 and 1
16	PG8MD2	0	R/W	The combination of bits PG8MD2 and PG8nMD1 controls the pin functions. 00: Port input 01: Port output 10, 11: Reserved (When set, correct operation cannot be guaranteed.)

Bit	Bit Name	Initial Value	R/W	Description
15	PG7MD2	0	R/W	PGn Mode 2 and 1
14	PG7MD2	0	R/W	The combination of bits PGnMD2 and PGnMD1 controls the pin functions. (n = 0 to 7)
13	PG6MD2	0	R/W	00: Port input/other functions (see table22.1.)
12	PG6MD2	0	R/W	01, 10, 11: Reserved (When set, correct operation cannot be guaranteed.)
11	PG5MD2	0	R/W	
10	PG5MD2	0	R/W	
9	PG4MD2	0	R/W	
8	PG4MD2	0	R/W	
7	PG3MD2	0	R/W	
6	PG3MD2	0	R/W	
5	PG2MD2	0	R/W	
4	PG2MD2	0	R/W	
3	PG1MD2	0	R/W	
2	PG1MD2	0	R/W	
1	PG0MD2	0	R/W	
0	PG0MD2	0	R/W	

Note: There is no bit for changing the function of port G (pins ANn: analog inputs for the A/D converter) between input and other functions because the function returns to input on completion of A/D conversion.

22.1.10 Port H Control Register (PHCR)

PHCR is a 32-bit readable/writable register that selects the pin functions. PHCR is initialized to H'00000000 by a power-on reset, and it is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
29	PH14MD2	0	R/W	PHn Mode 2 and 1 The combination of bits PHnMD2 and PHnMD1controls the pin functions. (n = 0 to 14) 00: Port input 01: Port output 10: Reserved (When set, correct operation cannot be guaranteed.) 11: Other functions (see table22.1.)
28	PH14MD1	0		
27	PH13MD2	0	R/W	
26	PH13MD1	0		
25	PH12MD2	0	R/W	
24	PH12MD2	0		
23	PH11MD2	0	R/W	
22	PH11MD2	0		
21	PH10MD2	0	R/W	
20	PH10MD2	0		
19	PH9MD2	0	R/W	
18	PH9MD2	0		
17	PH8MD2	0	R/W	
16	PH8MD2	0		
15	PH7MD2	0	R/W	
14	PH7MD2	0		
13	PH6MD2	0	R/W	
12	PH6MD2	0		
11	PH5MD2	0	R/W	
10	PH5MD2	0		
9	PH4MD2	0	R/W	
8	PH4MD2	0		

Bit	Bit Name	Initial Value	R/W	Description
7	PH3MD2	0	R/W	PHn Mode 2 and 1
6	PH3MD2	0		The combination of bits PHnMD2 and PHnMD1controls the pin functions. (n = 0 to 14)
5	PH2MD2	0	R/W	00: Port input
4	PH2MD2	0		01: Port output
3	PH1MD2	0	R/W	10: Reserved (When set, correct operation cannot be guaranteed.)
2	PH1MD2	0		11: Other functions (see table22.1.)
1	PH0MD2	0	R/W	
0	PH0MD2	0		

22.1.11 Port J Control Register (PJCR)

PJCR is a 32-bit readable/writable register that selects the pin functions. PJCR is initialized to H'00000000 by a power-on reset, and it is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	PJ12MD2	0	R/W	PJn Mode 2 and 1
24	PJ12MD2	0		The combination of bits PJnMD2 and PJnMD1controls the pin functions. (n = 0 to 12)
23	PJ11MD2	0	R/W	00: Port input 01: Port output 10: Reserved (When set, correct operation cannot be guaranteed.) 11: Other functions (see table 22.1)
22	PJ11MD2	0		
21	PJ10MD2	0	R/W	
20	PJ10MD2	0		
19	PJ9MD2	0	R/W	
18	PJ9MD2	0		
17	PJ8MD2	0	R/W	
16	PJ8MD2	0		
15	PJ7MD2	0	R/W	
14	PJ7MD2	0		

Bit	Bit Name	Initial Value	R/W	Description
13	PJ6MD2	0	R/W	PJn Mode 2 and 1
12	PJ6MD2	0		The combination of bits PJnMD2 and PJnMD1controls the pin functions. (n = 0 to 12)
11	PJ5MD2	0	R/W	00: Port input
10	PJ5MD2	0		01: Port output
9	PJ4MD2	0	R/W	10: Reserved (When set, correct operation cannot be guaranteed.)
8	PJ4MD2	0		11: Other functions (see table 22.1)
7	PJ3MD2	0	R/W	
6	PJ3MD2	0		
5	PJ2MD2	0	R/W	
4	PJ2MD2	0		
3	PJ1MD2	0	R/W	
2	PJ1MD2	0		
1	PJ0MD2	0	R/W	
0	PJ0MD2	0		

22.2 I/O Buffer Internal Block Diagram

22.2.1 I/O Buffer with Weak Keeper

All the I/O buffers except PTG10, PTG9, and PTG 7 to PTG 0 (IIC2 and analog pins) listed in table 22.1 have weak keepers that consist of two inverters to keep the status of the pin. Figure 22.1 shows the internal block diagram of the I/O buffer.

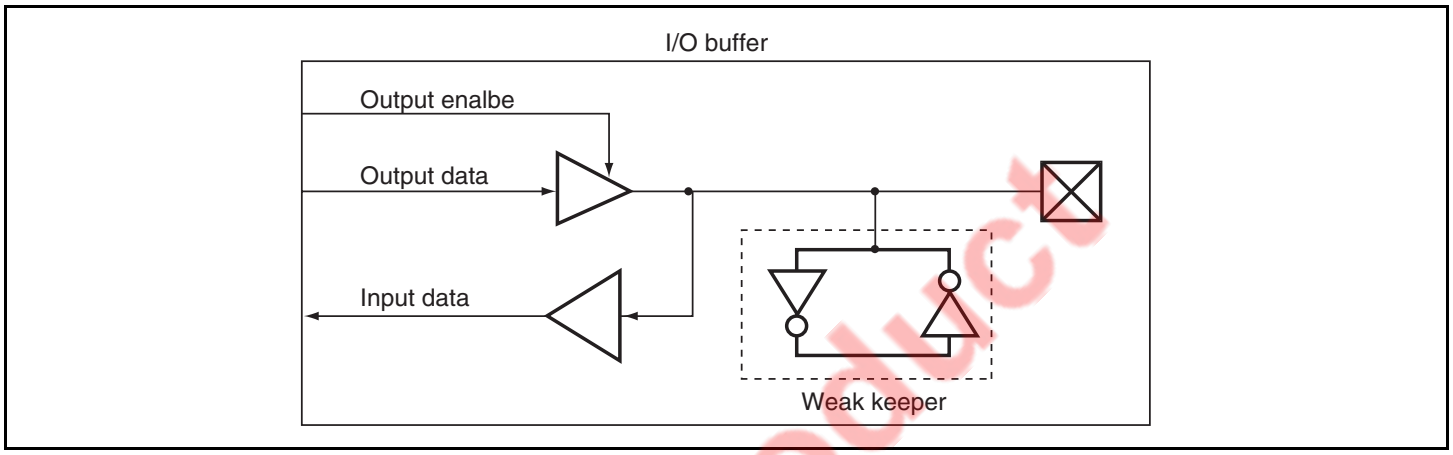


Figure 22.1 Internal Block Diagram of I/O Buffer with Weak Keeper

22.2.2 I/O Buffer with Open Drain Output

PTG10 and PTG9 are multiplexed with the IIC2 (SDA, SCL) pins and consist of the normal I/O buffer and the I/O buffer with an open drain output. Setting the port G control register (PGCR) to port input or port output enables the normal I/O buffer. Setting the PGCR to other function (IIC2) enables the I/O buffer with an open drain output.

Figure 22.2 shows the internal block diagram of the I/O buffer with an open drain output.

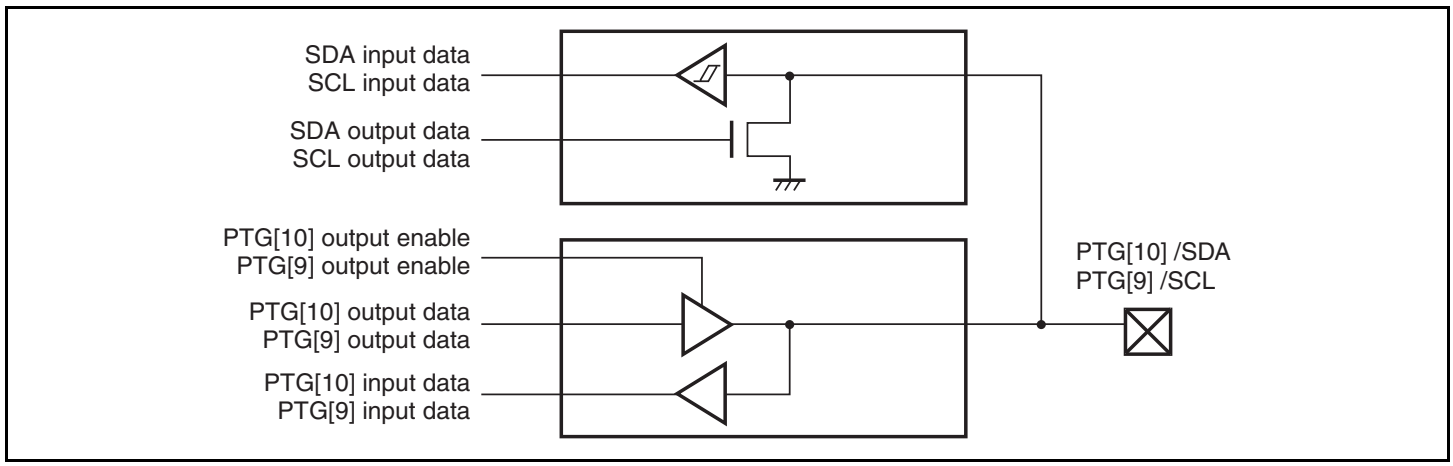


Figure 22.2 Internal Block Diagram of I/O Buffer with Open Drain

22.3 Notes on Usage

- Pins function as outputs when other function is selected by the port control register

  When the pin function (shown in table 22.1, List of Multiplexed Pins) is changed from other function (output) to port function (input), the weak keeper in figure 22.1 holds the value of the port data register of the pin.
- Pins function as inputs/outputs when other function is selected by the port control register

  When the pin function (shown in table 22.1, List of Multiplexed Pins) is changed from port function (input) to other function (output), the weak keeper in figure 22.1 holds the other function value of the pin.
- Pins PTG10 and PTG9

  The I/O buffers of PG10 and PTG9 have no weak keeper. When you do not use these pins, pull up or pull down them. If you use them as port input, do not apply mid-voltage.
- Pins with weak keepers

  Immediately after a power-on reset, the level of the pin which has a weak keeper is not undefined whether high or low. Thus, to fix the pin level, the pin needs to be pulled up or down.

  Reference pull-up and pull-down resistances are shown below. These resistances change according to the circuit configuration.

  Pull-up resistance (reference value) = 2 kΩ

  Pull-down resistance (reference value) = 8 kΩ

Section 23 I/O Ports

This LSI has nine 16-bit ports (ports A to J). All port pins are multiplexed with other pin functions (the pin function controller (PFC) handles the selection of pin functions). Each port has a data register which stores data for the pins.

23.1 Port A

Port A is a 15-bit input/output port with the pin configuration shown in figure 23.1. Each pin is controlled by the port A control register (PACR) in the PFC.

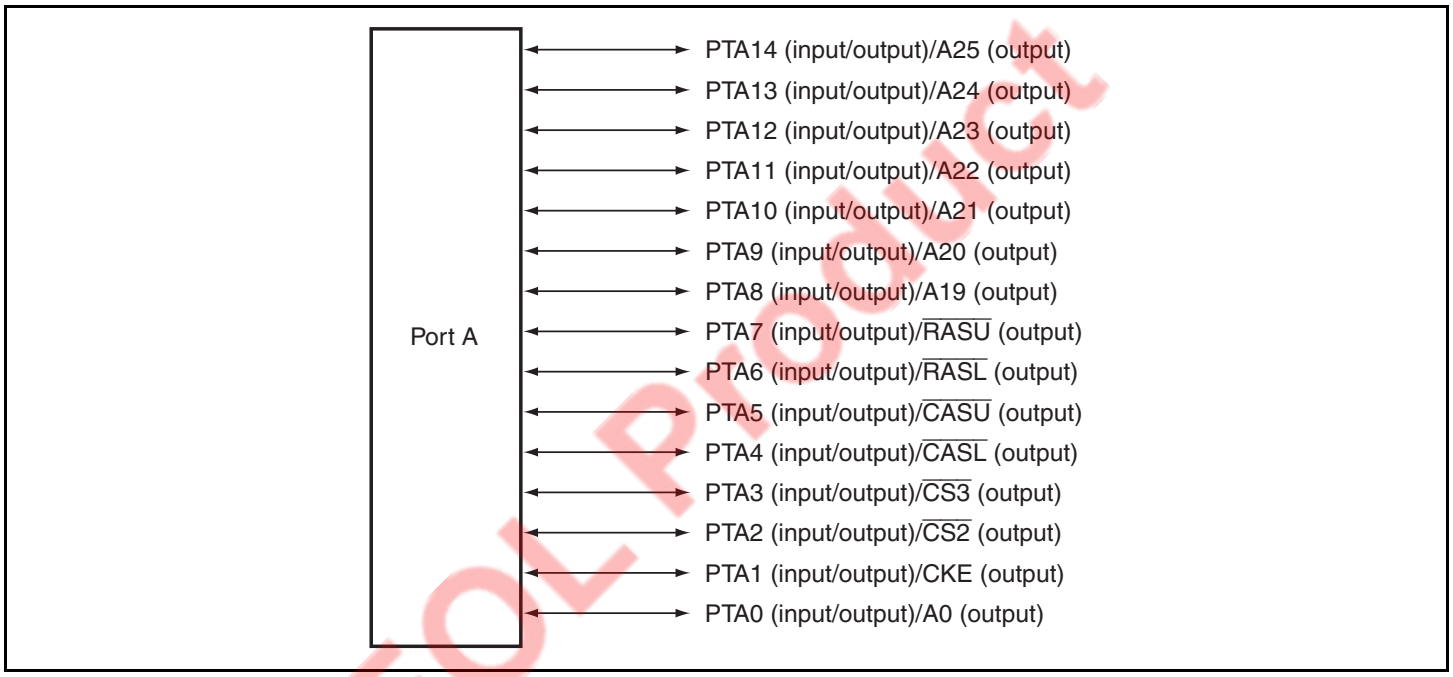


Figure 23.1 Port A

23.1.1 Register Description

Port A has the following register.

- Port A data register (PADR)

23.1.2 Port A Data Register (PADR)

PADR is a 15-bit readable/writable register with one reserved bit that stores data for pins PTA14 to PTA0. PADR is initialized to H'0000 by a power-on reset, but it retains its previous value by a manual reset, in standby mode or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	PA14DT	0	R/W	Bits PA14DT to PA0DT correspond to pins PTA14 to PTA0. When the pin function is general output port, the value of the corresponding PADR bit in PADR is returned directly by reading the port. When the function is general input port, the corresponding pin level is read by reading the port. Table 23.1 shows the function of PADR.
13	PA13DT	0	R/W	
12	PA12DT	0	R/W	
11	PA11DT	0	R/W	
10	PA10DT	0	R/W	
9	PA9DT	0	R/W	
8	PA8DT	0	R/W	
7	PA7DT	0	R/W	
6	PA6DT	0	R/W	
5	PA5DT	0	R/W	
4	PA4DT	0	R/W	
3	PA3DT	0	R/W	
2	PA2DT	0	R/W	
1	PA1DT	0	R/W	
0	PA0DT	0	R/W	

Table 23.1 Port A Data Register (PADR) Read/Write Operations

PAnMD2	PAnMD1	Pin Function	Read	Write
0	0	Input	Pin state	Data is written to PADR, but does not affect pin state.
	1	Output	PADR value	Data is written to PADR and the value is output from the pin.
1	0	Reserved	—	—
	1	Other functions	Pin state	Data is written to PADR, but does not affect pin state.

(n = 0 to 14)

23.2 Port B

Port B is a 9-bit input/output port with the pin configuration shown in figure 23.2. Each pin is controlled by the port B control register (PBCR) in the PFC.

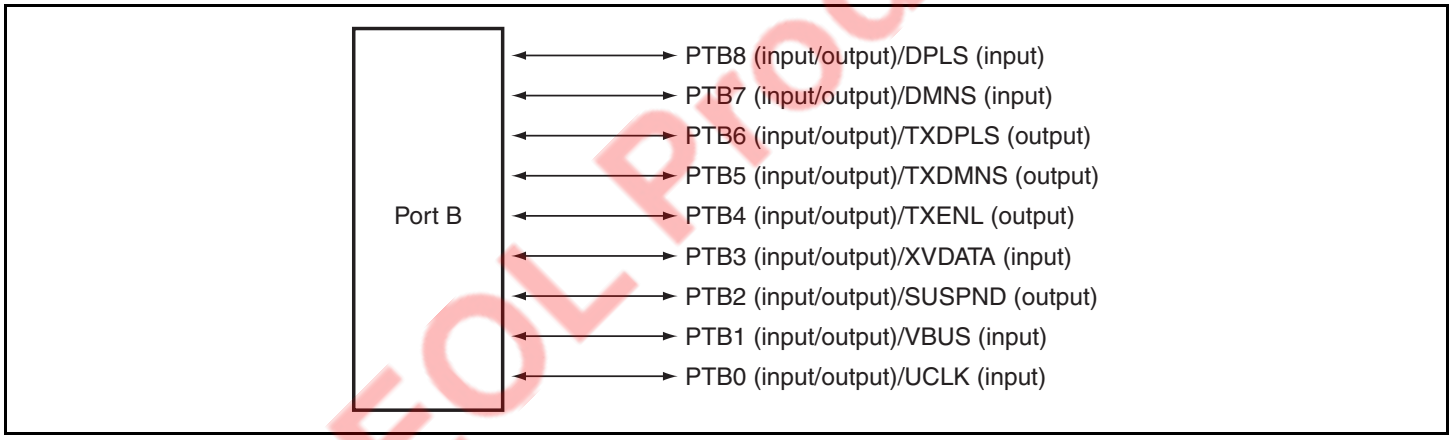


Figure 23.2 Port B

23.2.1 Register Description

Port B has the following register.

- Port B data register (PBDR)

23.2.2 Port B Data Register (PBDR)

PBDR is a 9-bit readable/writable register with seven reserved bits that stores data for pins PTB8 to PTB0. PBDR is initialized to H'0000 by a power-on reset, but it retains its previous value by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PB7DT	0	R/W	Bits PB8DT to PB0DT correspond to pins PTB8 to PTB0. When the pin function is general output port, the value of the corresponding bit in PBDR is returned directly by reading the port. When the function is general input port, the corresponding pin level is read by reading the port. Table 23.2 shows the function of PBDR.
6	PB6DT	0	R/W	
5	PB5DT	0	R/W	
4	PB4DT	0	R/W	
3	PB3DT	0	R/W	
2	PB2DT	0	R/W	
1	PB1DT	0	R/W	
0	PB0DT	0	R/W	

Table 23.2 Port B Data Register (PBDR) Read/Write Operations

PBnMD2	PBnMD1	Pin State	Read	Write
0	0	Input	Pin state	Data is written to PBDR, but does not affect pin state.
	1	Output	PBDR value	Data is written to PBDR and the value is output from the pin.
1	0	Reserved	—	—
	1	Other functions	Pin state	Data is written to PBDR, but does not affect pin state.

(n = 0 to 8)

23.3 Port C

Port C is a 16-bit input/output port with the pin configuration shown in figure 23.3. Each pin is controlled by the port C control register (PCCR) in the PFC.

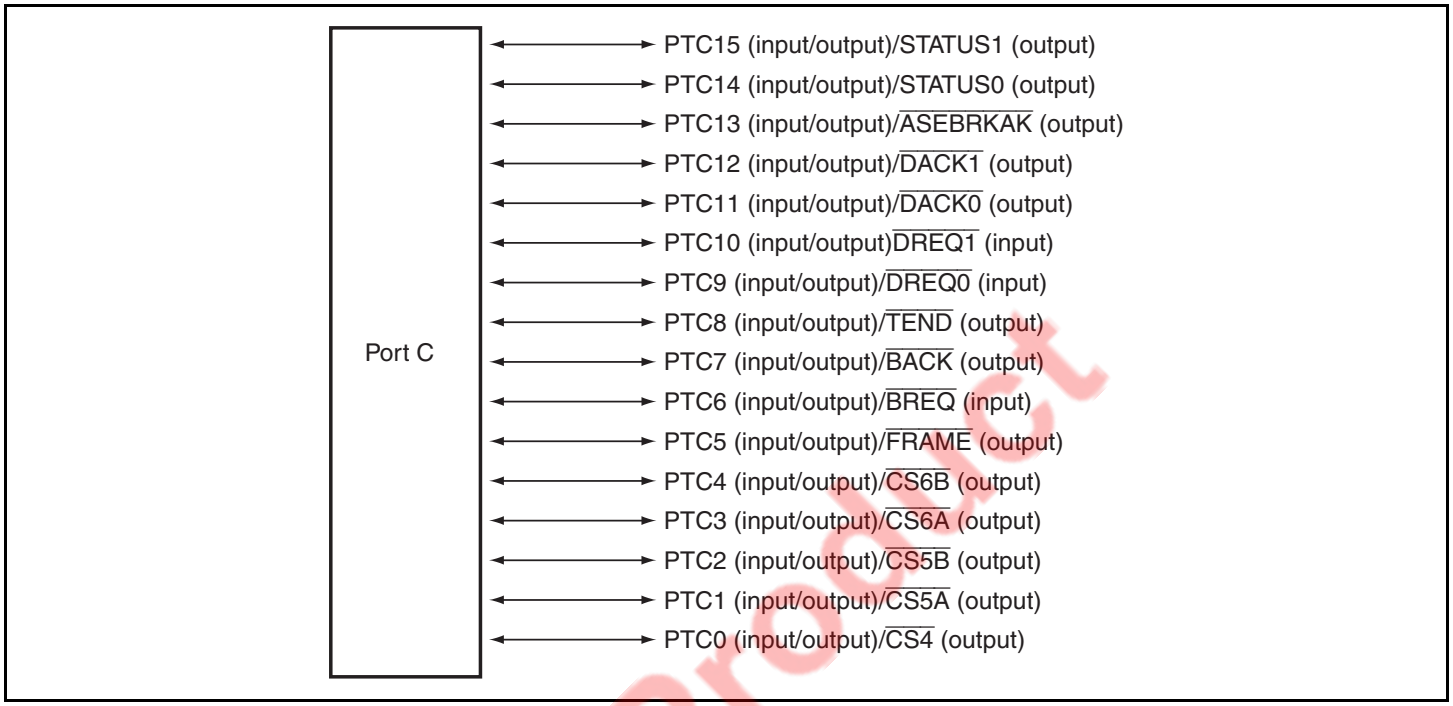


Figure 23.3 Port C

23.3.1 Register Description

Port C has the following register.

- Port C data register (PCDR)

23.3.2 Port C Data Register (PCDR)

PCDR is a 16-bit readable/writable register that stores data for pins PTC15 to PTC0. PCDR is initialized to H'0000 by a power-on reset, but it retains its previous value by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15	PC15DT	0	R/W	Bits PC15DT to PC0DT correspond to pins PTC15 to PTC0. When the pin function is general output port, the value of the corresponding bit in PCDR is returned directly by reading the port. When the function is general input port, the corresponding pin level is read by reading the port. Table 23.3 shows the function of PCDR.
14	PC14DT	0	R/W	
13	PC13DT	0	R/W	
12	PC12DT	0	R/W	
11	PC11DT	0	R/W	
10	PC10DT	0	R/W	
9	PC9DT	0	R/W	
8	PC8DT	0	R/W	
7	PC7DT	0	R/W	
6	PC6DT	0	R/W	
5	PC5DT	0	R/W	
4	PC4DT	0	R/W	
3	PC3DT	0	R/W	
2	PC2DT	0	R/W	
1	PC1DT	0	R/W	
0	PC0DT	0	R/W	

Table 23.3 Port C Data Register (PCDR) Read/Write Operations

PCnMD2	PCnMD1	Pin State	Read	Write
0	0	Input	Pin state	Data is written to PCDR, but does not affect pin state.
	1	Output	PCDR value	Data is written to PCDR and the value is output from the pin.
1	0	Reserved	—	—
	1	Other functions	Pin state	Data is written to PCDR, but does not affect pin state.

(n = 0 to 15)

23.4 Port D

Port D comprises a 16-bit input/output port with the pin configuration shown in figure 23.4. Each pin is controlled by the port D control register (PDCR) in the PFC.

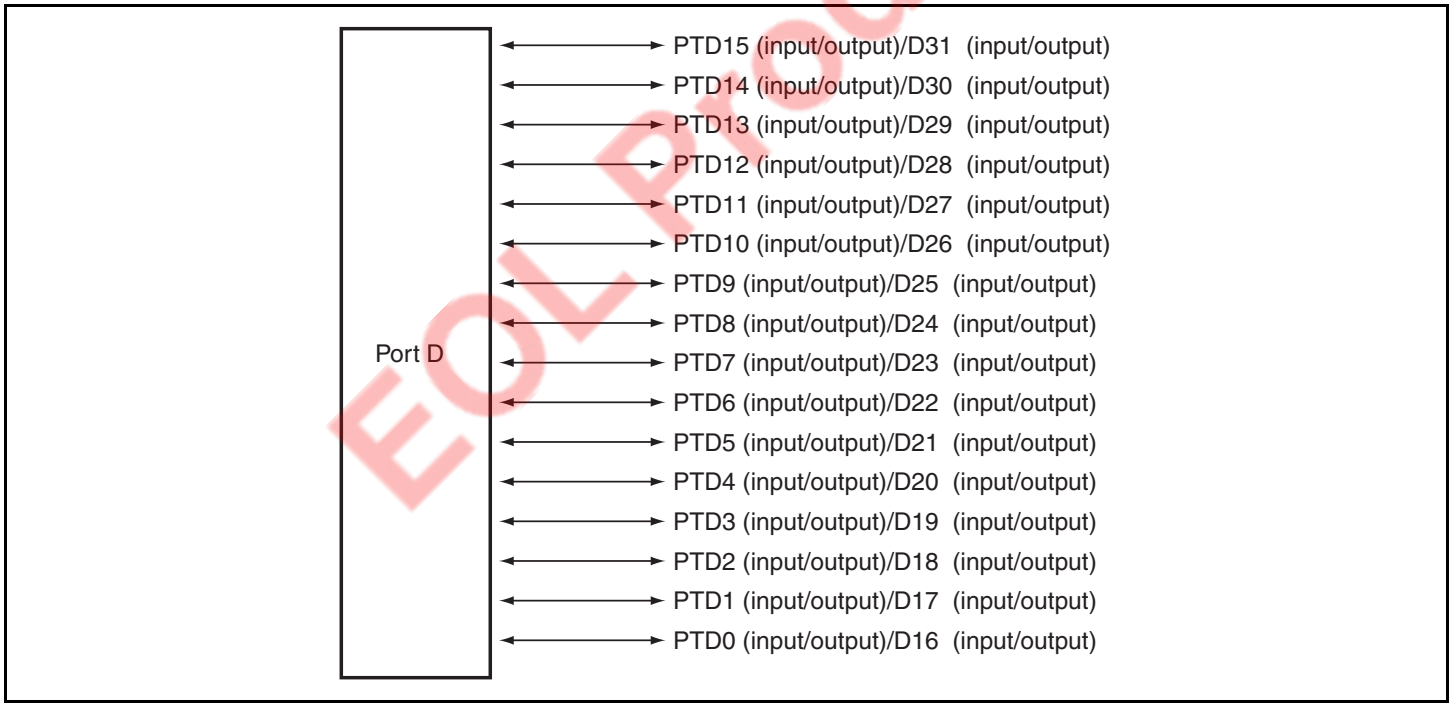


Figure 23.4 Port D

23.4.1 Register Description

Port D has the following register.

- Port D data register (PDDR)

23.4.2 Port D Data Register (PDDR)

PDDR is a 16-bit readable/writable register that stores data for pins PTD15 to PTD0. PDDR is initialized to H'0000 by a power-on reset, after which the general input port function is set as the initial pin function, and the corresponding pin levels are read when MD3 = 0 (16-bit bus width in CS0 space) is set. PDDR retains its previous value by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15	PD15DT	0	R/W	Bits PD15DT to PD0DT correspond to pins PTD15 to PTD0. When the pin function is general output port, the value of the corresponding bit in PDDR is returned directly by reading the port. When the function is general input port, the corresponding pin level is read by reading the port. Table 23.4 shows the function of PDDR.
14	PD14DT	0	R/W	
13	PD13DT	0	R/W	
12	PD12DT	0	R/W	
11	PD11DT	0	R/W	
10	PD10DT	0	R/W	
9	PD9DT	0	R/W	
8	PD8DT	0	R/W	
7	PD7DT	0	R/W	
6	PD6DT	0	R/W	
5	PD5DT	0	R/W	
4	PD4DT	0	R/W	
3	PD3DT	0	R/W	
2	PD2DT	0	R/W	
1	PD1DT	0	R/W	
0	PD0DT	0	R/W	

Table 23.4 Port D Data Register (PDDR) Read/Write Operations

PDnMD2	PDnMD1	Pin State	Read	Write
0	0	Input	Pin state	Data is written to PDDR, but does not affect pin state.
	1	Output	PDDR value	Data is written to PDDR and the value is output from the pin.
1	0	Reserved	—	—
	1	Other function	Pin state	Data is written to PDDR, but does not affect pin state.

(n = 0 to 15)

23.5 Port E

Port E is a 16-bit input/output port with the pin configuration shown in figure 23.5. Each pin is controlled by the port E control register (PECR) in the PFC.

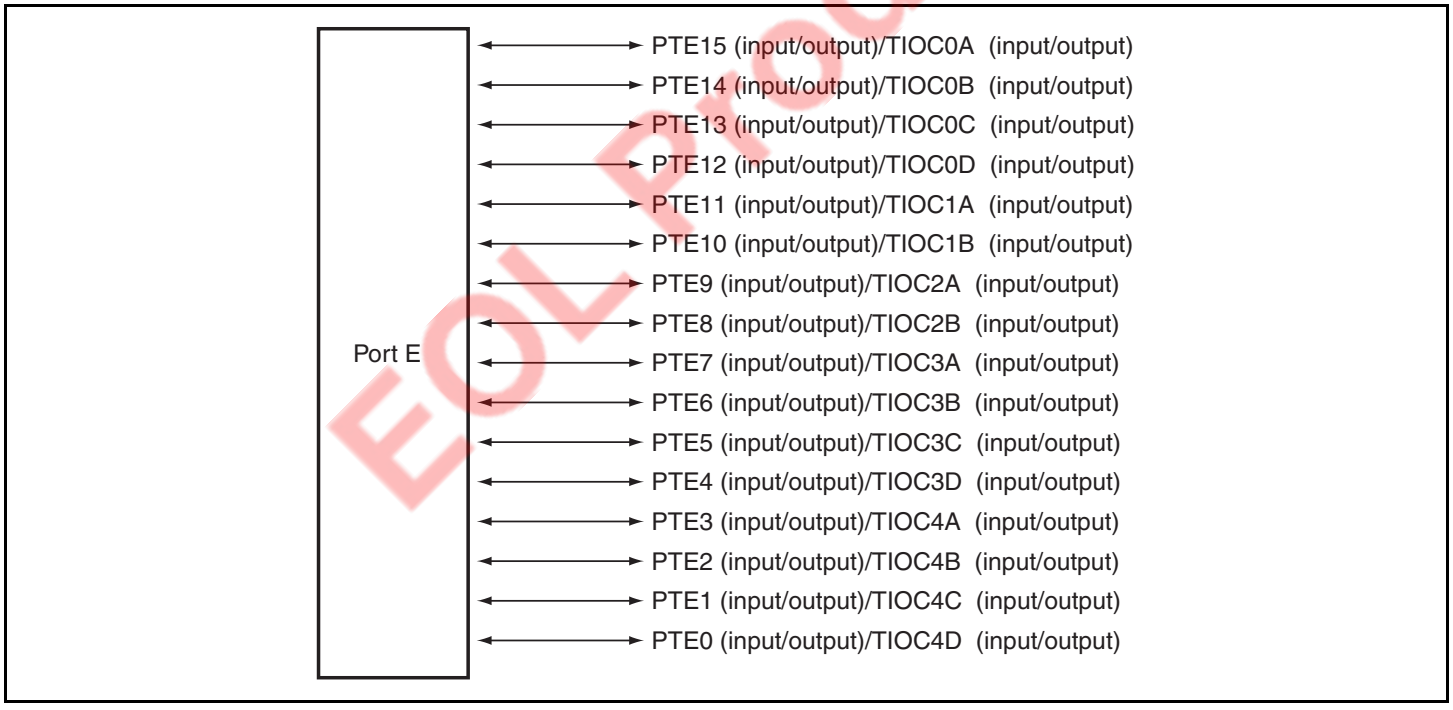


Figure 23.5 Port E

23.5.1 Register Description

Port E has the following register.

- Port E data register (PEDR)

23.5.2 Port E Data Register (PEDR)

PEDR is a 16-bit readable/writable register that stores data for pins PTE15 to PTE0. The PEDR is initialized to H'0000 by a power-on reset, but it retains its previous value by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15	PE15DT	0	R/W	Bits PE15DT to PE0DT correspond to pins PTE15 to PTE0. When the pin function is general output port, the value of the corresponding PEDR bit in PEDR is returned directly by reading the port. When the function is general input port, the corresponding pin level is read by reading the port. Table 23.5 shows the function of PEDR.
14	PE14DT	0	R/W	
13	PE13DT	0	R/W	
12	PE12DT	0	R/W	
11	PE11DT	0	R/W	
10	PE10DT	0	R/W	
9	PE9DT	0	R/W	
8	PE8DT	0	R/W	
7	PE7DT	0	R/W	
6	PE6DT	0	R/W	
5	PE5DT	0	R/W	
4	PE4DT	0	R/W	
3	PE3DT	0	R/W	
2	PE2DT	0	R/W	
1	PE1DT	0	R/W	
0	PE0DT	0	R/W	

Table 23.5 Port E Data Register (PEDR) Read/Write Operations

PEnMD2	PEnMD1	Pin State	Read	Write
0	0	Input	Pin state	Data is written to PEDR, but does not affect pin state.
	1	Output	PEDR value	Data is written to PEDR and the value is output from the pin.
1	0	Reserved	—	—
	1	Other function	Pin state	Data is written to PEDR, but does not affect pin state.

(n = 0 to 15)

23.6 Port F

Port F is a 16-bit input port with the pin configuration shown in figure 23.6. Each pin is controlled by the port F control register (PFCR) in the PFC.

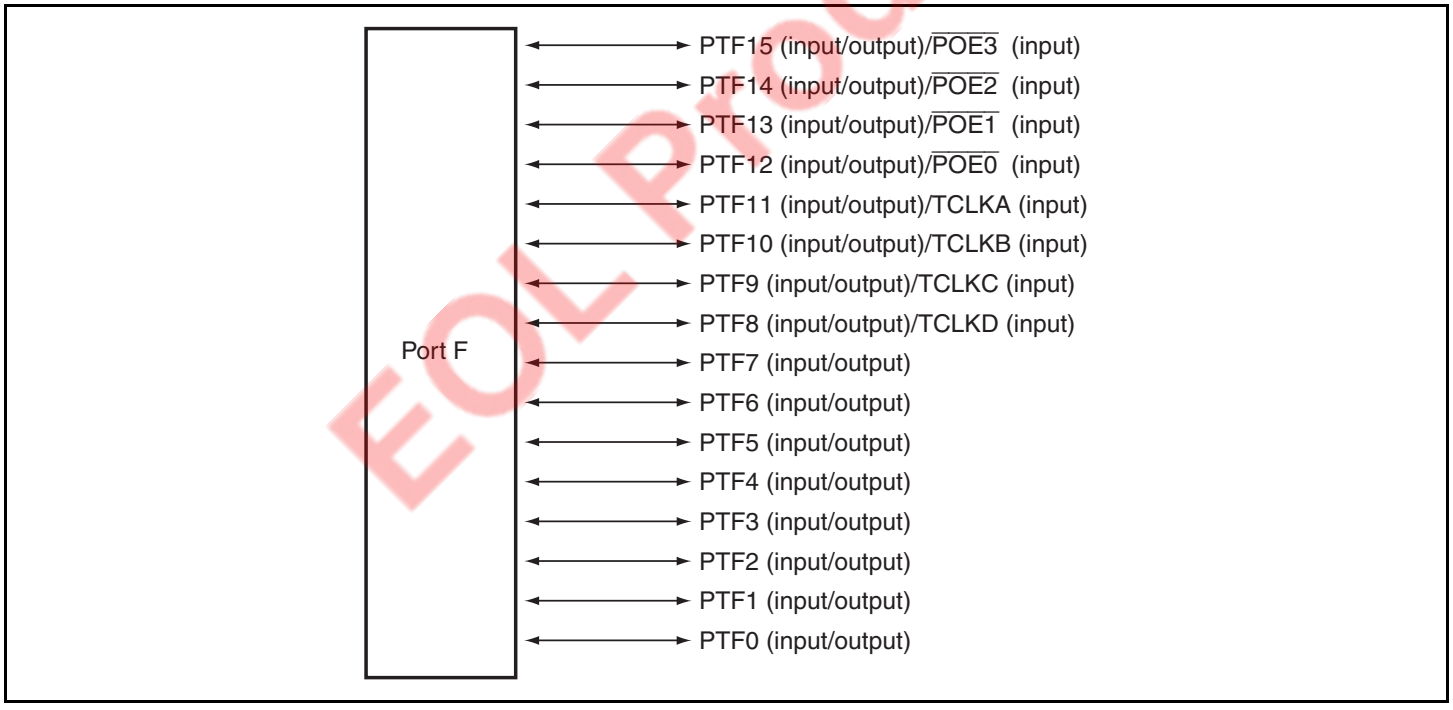


Figure 23.6 Port F

23.6.1 Register Description

Port F has the following register.

- Port F data register (PFDR)

23.6.2 Port F Data Register (PFDR)

PFDR is a 16-bit readable/writable register that stores data for pins PTF15 to PTF0. PFDR is initialized to H'0000 by a power-on reset, but it retains its previous value by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15	PF15DT	0	R/W	Bits PF15DT to PF0DT correspond to pins PTF15 to PTF0. When the function is general input port, the corresponding pin level is read by reading the port. Tables 23.6 and 23.7 show the function of PFDR.
14	PF14DT	0	R/W	
13	PF13DT	0	R/W	
12	PF12DT	0	R/W	
11	PF11DT	0	R/W	
10	PF10DT	0	R/W	
9	PF9DT	0	R/W	
8	PF8DT	0	R/W	
7	PF7DT	0	R	
6	PF6DT	0	R	
5	PF5DT	0	R/W	
4	PF4DT	0	R/W	
3	PF3DT	0	R/W	
2	PF2DT	0	R/W	
1	PF1DT	0	R/W	
0	PF0DT	0	R/W	

Table 23.6 Port F Data Register (PFDR) Read/Write Operations (PF15DT to PF8DT)

PFnMD2	PFnMD1	Pin State	Read	Write
0	0	Input	Pin state	Data is written to PFDR, but does not affect pin state.
	1	Output	PFDR value	Data is written to PFDR and the value is output from the pin.
1	0	Reserved	—	—
	1	Other function	Pin state	Data is written to PFDR, but does not affect pin state.

(n = 8 to 15)

Table 23.7 Port F Data Register (PFDR) Read/Write Operations (PF7DT to PF0DT)

PFnMD2	PFnMD1	Pin State	Read	Write
0	0	Input	Pin state	Data is written to PFDR, but does not affect pin state.
	1	Output	PFDR value	Data is written to PFDR and the value is output from the pin.
Other than above		Reserved	—	—

(n = 0 to 7)

23.7 Port G

Port G comprises a 6-bit input/output port and an 8-bit input port with the pin configuration shown in figure 23.7. Each pin is controlled by the port G control register (PGCR) in the PFC.

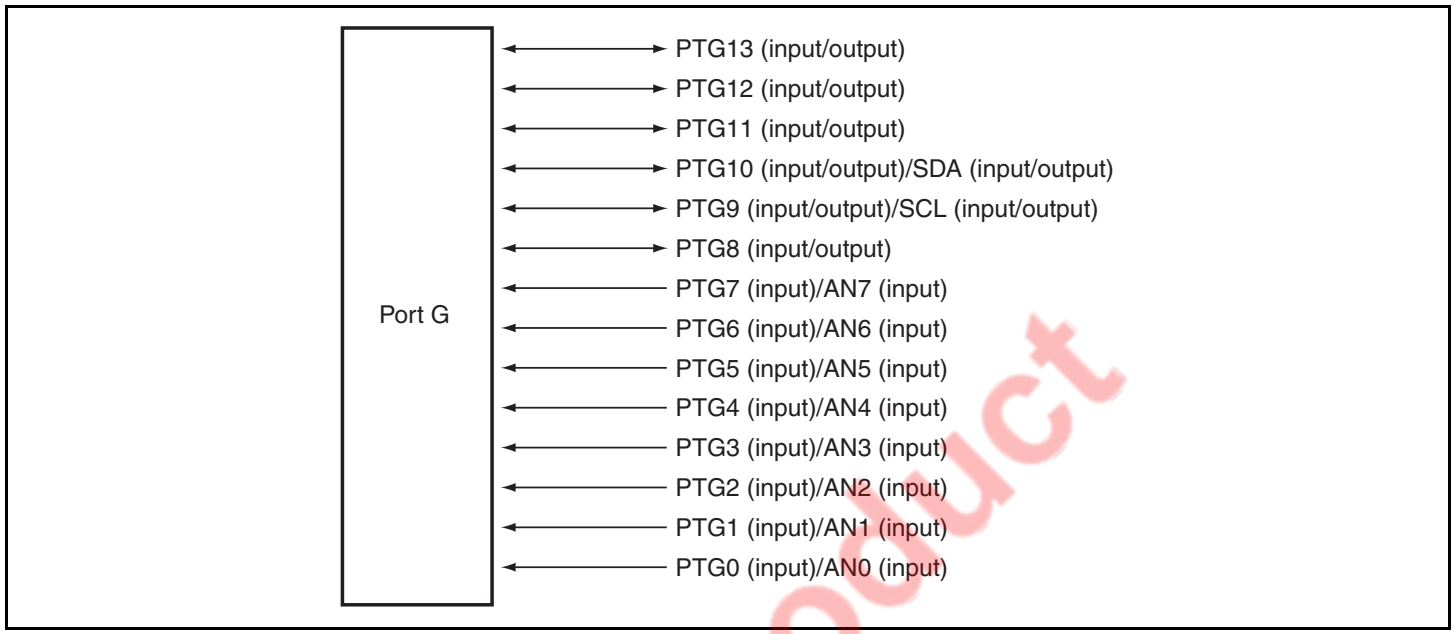


Figure 23.7 Port G

23.7.1 Register Description

Port G registers has the following register.

- Port G data register (PGDR)

23.7.2 Port G Data Register (PGDR)

PGDR a register that includes six readable/writable and eight readable bits with two reserved bits that store data for pins PTG13 to PTG0.

PGDR13 to PGDR8 are initialized to H'00 by a power-on reset, but they retain their previous values by a manual reset, in standby mode, or in sleep mode. PGDR7 to PGDR0 are not initialized by a power-on or manual reset, in standby mode, or in sleep mode. (The bit always indicates the status of the pin.)

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PG13DT	0	R/W	Bits PG13DT to PG8DT correspond to pins PTG13 to PTG8. When the function is general input port, the corresponding pin level is read by reading the port. Tables 23.8 and 23.9 show the function of PGDRs 13 to 8.
12	PG12DT	0	R/W	
11	PG11DT	0	R/W	
10	PG10DT	0	R/W	
9	PG9DT	0	R/W	
8	PG8DT	0	R/W	
7	PG7DT	0	R/W	Bits PG7DT to PG0DT correspond to pins PTG7 to PTG0. The values written to these bits are ignored and does not affect pin state. If these bits are read, the states of the pins are returned directly instead of the values of these bits. Do not read these bits when the A/D converter is used. Table 23.10 shows the function of PGDR.
6	PG6DT	0	R/W	
5	PG5DT	0	R/W	
4	PG4DT	0	R/W	
3	PG3DT	0	R/W	
2	PG2DT	0	R/W	
1	PG1DT	0	R/W	
0	PG0DT	0	R/W	

Note: * The initial value depends on the status of the pin at reading.

Table 23.8 Port G Data Register (PGDR) Read/Write Operations (PG13DT to PG11DT, PG8DT)

PGnMD2	PGnMD1	Pin State	Read	Write
0	0	Input	Pin state	Data is written to PGDR, but does not affect pin state.
	1	Output	PGDR value	Data is written to PGDR and the value is output from the pin.
Other than above		Reserved	—	—

(n = 8, 11 to 13)

Table 23.9 Port G Data Register (PGDR) Read/Write Operations (PG10DT to PG9DT)

PGnMD2	PGnMD1	Pin State	Read	Write
0	0	Input	Pin state	Data is written to PGDR, but does not affect pin state.
	1	Output	PGDR value	Data is written to PGDR and the value is output from the pin.
1	0	Reserved	—	—
	1	Other function	Pin state	Data is written to PGDR, but does not affect pin state.

(n = 9, 10)

Table 23.10 Port G Data Register (PGDR) Read/Write Operations (PG7DT to PG0DT)

PGnMD2	Pin State	Read	Write
0	Input/other function (The A/D converter is used.)	Prohibited	Prohibited
	Input/other function (The A/D converter is not used.)	Pin state	Ignored (does not affect pin state.)
1	Reserved	—	—

(n = 0 to 7)

23.7.3 Port G Internal Block Diagram

Pins PTG7 to PTG0 are multiplexed with the A/D converter. (See section 22, Pin Function Controller (PFC).) The statuses of these pins are read only when the PGDR is read, but are always input to the A/D converter.

Figure 23.8 shows the internal block diagram of PG7DT to PG0DT.

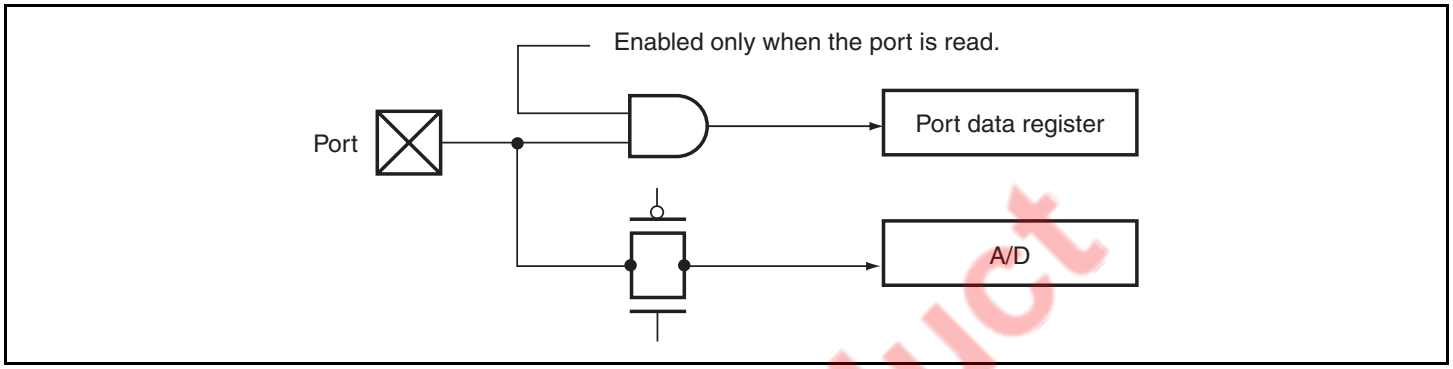


Figure 23.8 Internal Block Diagram of PG7DT to PG0DT

23.8 Port H

Port H comprises a 15-bit input/output port with the pin configuration shown in figure 23.9. Each pin is controlled by the port H control register (PHCR) in the PFC.

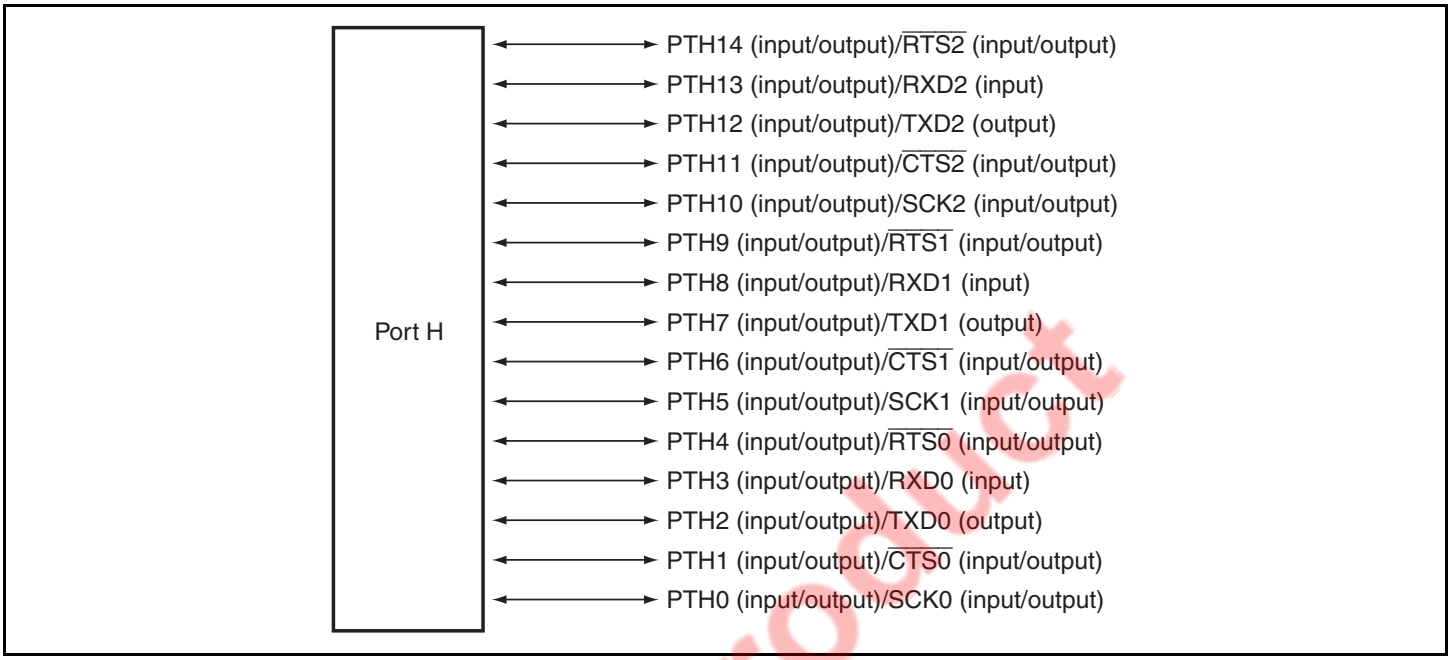


Figure 23.9 Port H

23.8.1 Register Description

Port H has the following register.

- Port H data register (PHDR)

23.8.2 Port H Data Register (PHDR)

PHDR is a 15-bit readable/writable register with one reserved bit that stores data for pins PTH14 to PTH0. PHDR is initialized to H'0000 by a power-on reset, but it retains its previous value by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	PH14DT	0	R/W	Bits PH14DT to PH0DT correspond to pins PTH14 to PTH0. When the pin function is general output port, the value of the corresponding bit in PHDR is returned directly by reading the port. When the function is general input port, the corresponding pin level is read by reading the port. Table 23.11 shows the function of PHDR.
13	PH13DT	0	R/W	
12	PH12DT	0	R/W	
11	PH11DT	0	R/W	
10	PH10DT	0	R/W	
9	PH9DT	0	R/W	
8	PH8DT	0	R/W	
7	PH7DT	0	R/W	
6	PH6DT	0	R/W	
5	PH5DT	0	R/W	
4	PH4DT	0	R/W	
3	PH3DT	0	R/W	
2	PH2DT	0	R/W	
1	PH1DT	0	R/W	
0	PH0DT	0	R/W	

Table 23.11 Port H Data Register (PHDR) Read/Write Operations

PHnMD2	PHnMD1	Pin State	Read	Write
0	0	Input	Pin state	Data is written to PHDR, but does not affect pin state.
	1	Output	PHDR value	Data is written to PHDR and the value is output from the pin.
1	0	Reserved	—	—
	1	Other functions	Pin state	Data is written to PHDR, but does not affect pin state.

(n = 0 to 14)

23.9 Port J

Port J is a 13-bit input/output port with the pin configuration shown in figure 23.10. Each pin is controlled by the port J control register (PJCR) in the PFC.

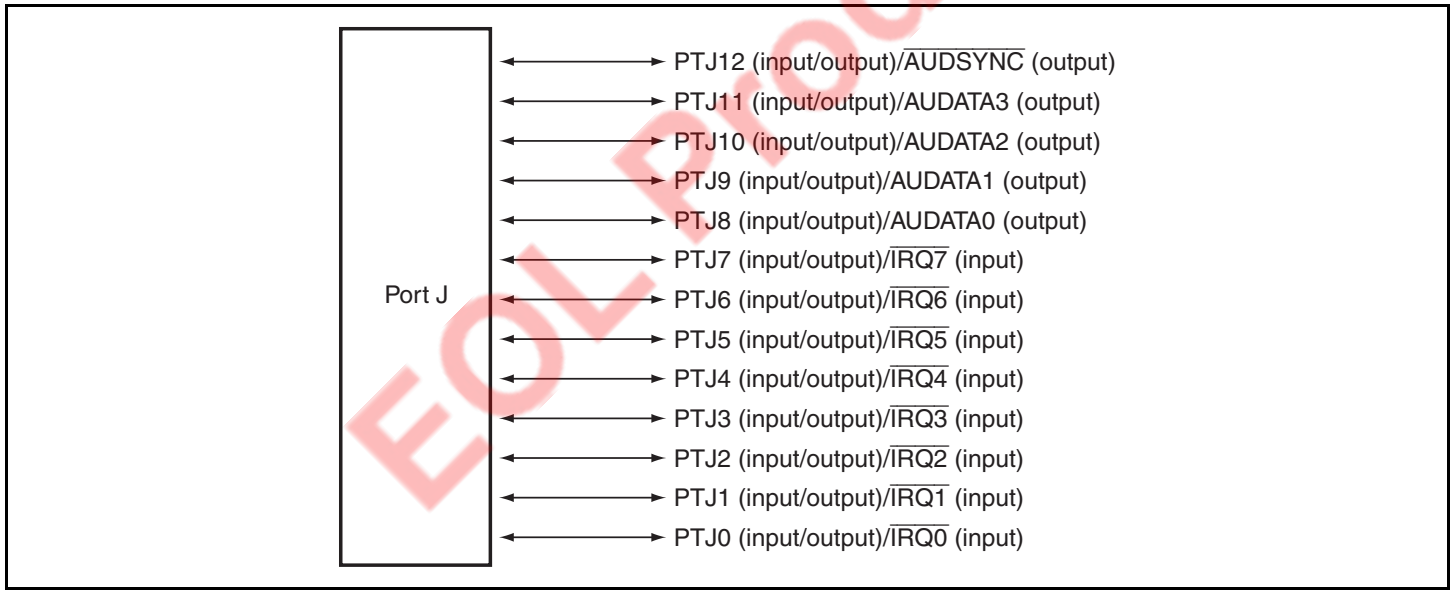


Figure 23.10 Port J

23.9.1 Register Description

Port J has the following register.

- Port J data register (PJDR)

23.9.2 Port J Data Register (PJDR)

PJDR is a 13-bit readable/writable register with three reserved bits that stores data for pins PTJ12 to PTJ0. The PJDR is initialized to H'0000 by a power-on reset, but it retains its previous value by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PJ12DT	0	R/W	Bits PJ12DT to PJ0DT correspond to pins PTJ12 to PTJ0. When the pin function is general output port, the value of the corresponding bit in PJDR is returned directly by reading the port. When the function is general input port, the corresponding pin level is read by reading the port. Table 23.12 shows the function of PJDR.
11	PJ11DT	0	R/W	
10	PJ10DT	0	R/W	
9	PJ9DT	0	R/W	
8	PJ8DT	0	R/W	
7	PJ7DT	0	R/W	
6	PJ6DT	0	R/W	
5	PJ5DT	0	R/W	
4	PJ4DT	0	R/W	
3	PJ3DT	0	R/W	
2	PJ2DT	0	R/W	
1	PJ1DT	0	R/W	
0	PJ0DT	0	R/W	

Table 23.12 Port J Data Register (PJDR) Read/Write Operations

PJnMD2	PJnMD1	Pin State	Read	Write
0	0	Input	Pin state	Data is written to PJDR, but does not affect pin state.
	1	Output	PJDR value	Data is written to PJDR and the value is output from the pin.
1	0	Reserved	—	—
	1	Other functions	Pin state	Data is written to PJDR, but does not affect pin state.

(n = 0 to 12)

Section 24 List of Registers

This section gives information on the on-chip I/O registers and is configured as described below.

1. Register Addresses (by functional module, in order of the corresponding section numbers)

- Descriptions by functional module, in order of the corresponding section numbers
  Entries that consist of - lines are for separation of the functional modules.
- Access to reserved addresses which are not described in this list is prohibited.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given.

2. Register Bits

- Bit configurations of the registers are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- Reserved bits are indicated by—in the bit name.
- No entry in the bit-name column indicates that the whole register is allocated as a counter or for holding data.
- When registers consist of 16 or 32 bits, bits are described from the MSB side.

3. Register States in Each Operating Mode

- Register states are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- For the initial state of each bit, refer to the description of the register in the corresponding section.
- The register states described are for the basic operating modes. If there is a specific reset for an on-chip module, refer to the section on that on-chip module.

24.1 Register Addresses (by functional module, in order of the corresponding section numbers)

Entries under Access size indicates numbers of bits.

Note: Access to undefined or reserved addresses is prohibited. Since operation or continued operation is not guaranteed when these registers are accessed, do not attempt such access.

Register Name	Abbreviation	Bit No.	Address	Module	Access States
Frequency control register	FRQCR	16	H'A415FF80	CPG	16
—	—	—	—	—	—
Watchdog timer counter	WTCNT	8	H'A415FF84	WDT	16*[1]
Watchdog timer control/status register	WTCSR	8	H'A415FF86		16*[2]
—	—	—	—	—	—
Standby control register	STBCR	8	H'A415FF82	Power-down modes	8
Standby control register 2	STBCR2	8	H'A415FF88		8
Standby control register 3	STBCR3	8	H'A40A0000		8
Standby control register 4	STBCR4	8	H'A40A0004		8
—	—	—	—	—	—
Cache control register 1	CCR1	32	H'FFFFFFEC	Cache	32
Cache control register 2	CCR2	32	H'A40000B0		32
—	—	—	—	—	—
Interrupt event register 2	INTEVT2	32	H'A400 0000	Exception handling	32
TRAPA exception register	TRA	32	H'FFFFFFD0		32
Exception event register	EXPEVT	32	H'FFFFFFD4		32
—	—	—	—	—	—
Interrupt priority registers F	IPRF	16	H'A408 0000	INTC	16
Interrupt priority registers G	IPRG	16	H'A408 0002		16
Interrupt priority registers H	IPRH	16	H'A408 0004		16
Interrupt priority registers I	IPRI	16	H'A408 0006		16

Register Name	Abbreviation	Bit No.	Address	Module	Access States
Interrupt mask register 0	IMR0	8	H'A408 0040	INTC	8
Interrupt mask register 1	IMR1	8	H'A408 0042		8
Interrupt mask register 2	IMR2	8	H'A408 0044		8
Interrupt mask register 4	IMR4	8	H'A408 0048		8
Interrupt mask register 5	IMR5	8	H'A408 004A		8
Interrupt mask register 6	IMR6	8	H'A408 004C		8
Interrupt mask register 7	IMR7	8	H'A408 004E		8
Interrupt mask register 8	IMR8	8	H'A408 0050		8
Interrupt mask register 9	IMR9	8	H'A408 0052		8
Interrupt mask register 10	IMR10	8	H'A408 0054		8
Interrupt mask clear register 0	IMCR0	8	H'A408 0060		8
Interrupt mask clear register 1	IMCR1	8	H'A408 0062		8
Interrupt mask clear register 2	IMCR2	8	H'A408 0064		8
Interrupt mask clear register 4	IMCR4	8	H'A408 0068		8
Interrupt mask clear register 5	IMCR5	8	H'A408 006A		8
Interrupt mask clear register 6	IMCR6	8	H'A408 006C		8
Interrupt mask clear register 7	IMCR7	8	H'A408 006E		8
Interrupt mask clear register 8	IMCR8	8	H'A408 0070		8
Interrupt mask clear register 9	IMCR9	8	H'A408 0072		8
Interrupt mask clear register 10	IMCR10	8	H'A408 0074		8
Interrupt request register 0	IRR0	8	H'A414 0004		8
Interrupt control register 1	ICR1	16	H'A414 0010		16
Interrupt control register 3	ICR3	16	H'A414 0020		16
Interrupt priority registers C	IPRC	16	H'A414 0016		16
Interrupt priority registers D	IPRD	16	H'A414 0018		16
Interrupt priority registers E	IPRE	16	H'A414 001A		16
Interrupt priority registers J	IPRJ	16	H'A414 0030		16
Interrupt control register 0	ICR0	16	H'A414 FEE0		16
Interrupt priority registers B	IPRB	16	H'A414 FEE4		16
—	—	—	—	—	—

Register Name	Abbreviation	Bit No.	Address	Module	Access States
Break data register B	BDRB	32	H'A4FFFF90	UBC	32
Break data mask register B	BDMRB	32	H'A4FFFF94		32
Break control register	BRCR	32	H'A4FFFF98		32
Execution Times Break Register	BETR	16	H'A4FFFF9C		16
Break address register B	BARB	32	H'A4FFFFA0		32
Break address mask register B	BAMRB	32	H'A4FFFFA4		32
Break bus cycle register B	BBRB	16	H'A4FFFFA8		16
Branch source register	BRSR	32	H'A4FFFFAC		32
Break address register A	BARA	32	H'A4FFFFB0		32
Break address mask register A	BAMRA	32	H'A4FFFFB4		32
Break bus cycle register A	BBRA	16	H'A4FFFFB8		16
Branch destination register	BRDR	32	H'A4FFFFBC		32
—	—	—	—	—	—
Common control register	CMNCR	32	H'A4FD0000	BSC	32
Bus control register for area 0	CS0BCR	32	H'A4FD0004		32
Bus control register for area 2	CS2BCR	32	H'A4FD0008		32
Bus control register for area 3	CS3BCR	32	H'A4FD000C		32
Bus control register for area 4	CS4BCR	32	H'A4FD0010		32
Bus control register for area 5A	CS5ABCR	32	H'A4FD0014		32
Bus control register for area 5B	CS5BBCR	32	H'A4FD0018		32
Bus control register for area 6A	CS6ABCR	32	H'A4FD001C		32
Bus control register for area 6B	CS6BBCR	32	H'A4FD0020		32
Wait control register for area 0	CS0WCR	32	H'A4FD0024		32
Wait control register for area 2	CS2WCR	32	H'A4FD0028		32
Wait control register for area 3	CS3WCR	32	H'A4FD002C		32
Wait control register for area 4	CS4WCR	32	H'A4FD0030		32
Wait control register for area 5A	CS5AWCR	32	H'A4FD0034		32
Wait control register for area 5B	CS5BWCR	32	H'A4FD0038		32
Wait control register for area 6A	CS6AWCR	32	H'A4FD003C		32
Wait control register for area 6B	CS6BWCR	32	H'A4FD0040		32
SDRAM control register	SDCR	32	H'A4FD0044		32

Register Name	Abbreviation	Bit No.	Address	Module	Access States
Refresh timer control/status register	RTCSR	16	H'A4FD0048	BSC	32*[3]
Refresh timer counter	RTCNT	16	H'A4FD004C		32*[3]
Refresh time constant register	RTCOR	16	H'A4FD0050		32*[3]
Reset wait counter	RWTCNT	16	H'A4FD0054		32*[3]
—	—	—	—	—	—
DMA source address register_0	SAR_0	32	H'A401 0020	DMAC	16/32
DMA destination address register_0	DAR_0	32	H'A401 0024		16/32
DMA transfer count register_0	DMATCR_0	32	H'A401 0028		16/32
DMA channel control register_0	CHCR_0	32	H'A401 002C		8/16/32
DMA source address register_1	SAR_1	32	H'A401 0030		16/32
DMA destination address register_1	DAR_1	32	H'A401 0034		16/32
DMA transfer count register_1	DMATCR_1	32	H'A401 0038		16/32
DMA channel control register _1	CHCR_1	32	H'A401 003C		8/16/32
DMA source address register_2	SAR_2	32	H'A401 0040		16/32
DMA destination address register_2	DAR_2	32	H'A401 0044		16/32
DMA transfer count register_2	DMATCR_2	32	H'A401 0048		16/32
DMA channel control register_2	CHCR_2	32	H'A401 004C		8/16/32
DMA source address register_3	SAR_3	32	H'A401 0050		16/32
DMA destination address register_3	DAR_3	32	H'A401 0054		16/32
DMA transfer count register_3	DMATCR_3	32	H'A401 0058		16/32
DMA channel control register_3	CHCR_3	32	H'A401 005C		8/16/32
DMA operation register	DMAOR	32	H'A401 0060		8/16/32
DMA extension resource selector 0	DMARS0	16	H'A409 0000		16
DMA extension resource selector 1	DMARS1	16	H'A409 0004		16
—	—	—	—	—	—
Instruction register	SDIR	16	H'A100 0200	H-UDI	16
ID Register	SDIDH	16	H'A100 0214		16/32
ID Register	SDIDL	16	H'A100 0216		16
—	—	—	—	—	—

Register Name	Abbreviation	Bit No.	Address	Module	Access States
I^2C bus control register 1	ICCR1	8	H'A447 0000	IIC2	8
I^2C bus control register 2	ICCR2	8	H'A447 0001		8
I^2C bus mode register	ICMR	8	H'A447 0002		8
I^2C bus interrupt enable register	ICIER	8	H'A447 0003		8
I^2C bus status register	ICSR	8	H'A447 0004		8
I^2C bus slave address register	SAR	8	H'A447 0005		8
I^2C bus transmit data register	ICDRT	8	H'A447 0006		8
I^2C bus receive data register	ICDRR	8	H'A447 0007		8
NF2CYC register	NF2CYC	8	H'A447 0008		8
—	—	—	—	—	—
Compare match timer start register_0	CMSTR_0	16	H'A44A 0000	CMT	16
Compare match timer control/ status register_0	CMCSR_0	16	H'A44A 0004		16
Compare match counter_0	CMCNT_0	16	H'A44A 0008		16
Compare match timer constant register_0	CMCOR_0	16	H'A44A 000C		16
Compare match timer start register_1	CMSTR_1	16	H'A44B 0000		16
Compare match timer control/ status register_1	CMCSR_1	16	H'A44B 0004		16
Compare match counter_1	CMCNT_1	16	H'A44B 0008		16
Compare match timer constant register_1	CMCOR_1	16	H'A44B 000C		16
—	—	—	—	—	—
Timer control register_3	TCR_3	8	H'A449 0000	MTU	8/16/32
Timer control register_4	TCR_4	8	H'A449 0001		8/16/32
Timer mode register_3	TMDR_3	8	H'A449 0002		8/16/32
Timer mode register_4	TMDR_4	8	H'A449 0003		8/16/32
Timer I/O control register H_3	TIORH_3	8	H'A449 0004		8/16/32
Timer I/O control register L_3	TIORL_3	8	H'A449 0005		8/16/32
Timer I/O control register H_4	TIORH_4	8	H'A449 0006		8/16/32
Timer I/O control register L_4	TIORL_4	8	H'A449 0007		8/16/32
Timer interrupt enable register_3	TIER_3	8	H'A449 0008		8/16/32
Timer interrupt enable register_4	TIER_4	8	H'A449 0009		8/16/32
Timer output master enable register	TOER	8	H'A449 000A		8/16/32

Register Name	Abbreviation	Bit No.	Address	Module	Access States
Timer output control register	TOCR	8	H'A449 000B	MTU	8/16/32
Timer gate control register	TGCR	8	H'A449 000D		8
Timer counter_3	TCNT_3	16	H'A449 0010		16/32
Timer counter_4	TCNT_4	16	H'A449 0012		16/32
Timer cycle data register	TCDR	16	H'A449 0014		16/32
Timer dead time data register	TDDR	16	H'A449 0016		16/32
Timer general register A_3	TGRA_3	16	H'A449 0018		16/32
Timer general register B_3	TGRB_3	16	H'A449 001A		16/32
Timer general register A_4	TGRA_4	16	H'A449 001C		16/32
Timer general register B_4	TGRB_4	16	H'A449 001E		16/32
Timer subcounter	TCNTS	16	H'A449 0020		16/32
Timer cycle buffer register	TCBR	16	H'A449 0022		16/32
Timer general register C_3	TGRC_3	16	H'A449 0024		16/32
Timer general register D_3	TGRD_3	16	H'A449 0026		16/32
Timer general register C_4	TGRC_4	16	H'A449 0028		16/32
Timer general register D_4	TGRD_4	16	H'A449 002A		16/32
Timer status register_3	TSR_3	8	H'A449 002C		8/16
Timer status register_4	TSR_4	8	H'A449 002D		8/16
Timer start register	TSTR	8	H'A449 0040		8/16
Timer synchro register	TSYR	8	H'A449 0041		8/16
Timer control register_0	TCR_0	8	H'A449 0060		8/16/32
Timer mode register_0	TMDR_0	8	H'A449 0061		8/16/32
Timer I/O control register H_0	TIORH_0	8	H'A449 0062		8/16/32
Timer I/O control register L_0	TIORL_0	8	H'A449 0063		8/16/32
Timer interrupt enable register_0	TIER_0	8	H'A449 0064		8/16/32
Timer status register_0	TSR_0	8	H'A449 0065		8/16/32
Timer counter_0	TCNT_0	16	H'A449 0066		16
Timer general register A_0	TGRA_0	16	H'A449 0068		16/32
Timer general register B_0	TGRB_0	16	H'A449 006A		16/32
Timer general register C_0	TGRC_0	16	H'A449 006C		16/32
Timer general register D_0	TGRD_0	16	H'A449 006E		16/32

Register Name	Abbreviation	Bit No.	Address	Module	Access States
Timer control register_1	TCR_1	8	H'A449 0080	MTU	8/16
Timer mode register_1	TMDR_1	8	H'A449 0081		8/16
Timer I/O control register _1	TIOR_1	8	H'A449 0082		8
Timer interrupt enable register_1	TIER_1	8	H'A449 0084		8/16/32
Timer status register_1	TSR_1	8	H'A449 0085		8/16/32
Timer counter_1	TCNT_1	16	H'A449 0086		8/16/32
Timer general register A_1	TGRA_1	16	H'A449 0088		16/32
Timer general register B_1	TGRB_1	16	H'A449 008A		16/32
Timer control register_2	TCR_2	8	H'A449 00A0		8/16
Timer mode register_2	TMDR_2	8	H'A449 00A1		8/16
Timer I/O control register_2	TIOR_2	8	H'A449 00A2		8
Timer interrupt enable register_2	TIER_2	8	H'A449 00A4		8/16/32
Timer status register_2	TSR_2	8	H'A449 00A5		8/16/32
Timer counter_2	TCNT_2	16	H'A449 00A6		16/32
Timer general register A_2	TGRA_2	16	H'A449 00A8		16/32
Timer general register B_2	TGRB_2	16	H'A449 00AA		16/32
Input level control/status register 1	ICSR1	16	H'A44C 0000		8/16/32
Output level control/status register	OCSR	16	H'A44C 0002		8/16/32
—	—	—	—	—	—
Serial mode register_0	SCSMR_0	16	H'A440 0000	SCIF	16
Bit rate register_0	SCBRR_0	8	H'A440 0004		8
Serial control register_0	SCSCR_0	16	H'A440 0008		16
Transmit FIFO data register_0	SCFTDR_0	8	H'A440 000C		8
Serial status register_0	SCFSR_0	16	H'A440 0010		16
Receive FIFO data register_0	SCFRDR_0	8	H'A440 0014		8
FIFO control register_0	SCFCR_0	16	H'A440 0018		16
FIFO data count register_0	SCFDR_0	16	H'A440 001C		16
Serial port register_0	SCSPTR_0	16	H'A440 0020		16
Line status register_0	SCLSR_0	16	H'A440 0024		16
Serial mode register_1	SCSMR_1	16	H'A441 0000		16

Register Name	Abbreviation	Bit No.	Address	Module	Access States
Bit rate register_1	SCBRR_1	8	H'A441 0004	SCIF	8
Serial control register_1	SCSCR_1	16	H'A441 0008		16
Transmit FIFO data register_1	SCFTDR_1	8	H'A441 000C		8
Serial status register_1	SCFSR_1	16	H'A441 0010		16
Receive FIFO data register_1	SCFRDR_1	8	H'A441 0014		8
FIFO control register_1	SCFCR_1	16	H'A441 0018		16
FIFO data count register_1	SCFDR_1	16	H'A441 001C		16
Serial port register_1	SCSPTR_1	16	H'A441 0020		16
Line status register_1	SCLSR_1	16	H'A441 0024		16
Serial mode register_2	SCSMR_2	16	H'A442 0000		16
Bit rate register_2	SCBRR_2	8	H'A442 0004		8
Serial control register_2	SCSCR_2	16	H'A442 0008		16
Transmit FIFO data register_2	SCFTDR_2	8	H'A442 000C		8
Serial status register_2	SCFSR_2	16	H'A442 0010		16
Receive FIFO data register_2	SCFRDR_2	8	H'A442 0014		8
FIFO control register_2	SCFCR_2	16	H'A442 0018		16
FIFO data count register_2	SCFDR_2	16	H'A442 001C		16
Serial port register_2	SCSPTR_2	16	H'A442 0020		16
Line status register_2	SCLSR_2	16	H'A442 0024		16
—	—	—	—	—	—
USB interrupt flag register 0	USBIFR0	8	H'A448 0000	USB	8
USB interrupt flag register 1	USBIFR1	8	H'A448 0001		8
USBEP0i data register	USBEPDR0i	8	H'A448 0002		8
USBEP0o data register	USBEPDR0o	8	H'A448 0003		8
USB trigger register	USBTRG	8	H'A448 0004		8
USB FIFO clear register	USBFCLR	8	H'A448 0005		8
USBEP0o receive data size register	USBEPSZ0o	8	H'A448 0006		8
USBEP0s data register	USBEPDR0s	8	H'A448 0007		8
USB data status register	USBDASTS	8	H'A448 0008		8
USB interrupt select register 0	USBISR0	8	H'A448 000A		8
USB endpoint stall register	USBEPSTL	8	H'A448 000B		8
USB interrupt enable register 0	USBIER0	8	H'A448 000C		8

Register Name	Abbreviation	Bit No.	Address	Module	Access States
USB interrupt enable register 1	USBIER1	8	H'A448 000D	USB	8
USBEP1 receive data size register	USBEPSZ1	8	H'A448 000F		8
USB interrupt select register 1	USBISR1	8	H'A448 0010		8
USB DMA transfer setting register	USBDMAR	8	H'A448 0011		8
USBEP3 data register	USBEPDR3	8	H'A448 0012		8
USBEP1 data register	USBEPDR1	8	H'A448 0014		8/32
USBEP2 data register	USBEPDR2	8	H'A448 0018		8/32
USB transceiver control register	USBXVERCR	8	H'A448 001C		8
USB interrupt flag register 2	USBIFR2	8	H'A448 001D		8
USB interrupt enable register 2	USBIER2	8	H'A448 001E		8
USB bus power control register	USBCTRL	8	H'A448 001F		8
—	—	—	—	—	—
A/D0 data register A	ADDRA0	16	H'A44E 0000	ADC	16
A/D0 data register B	ADDRB0	16	H'A44E 0002		16
A/D0 data register C	ADDRC0	16	H'A44E 0004		16
A/D0 data register D	ADDRD0	16	H'A44E 0006		16
A/D1 data register A	ADDRA1	16	H'A44E 0008		16
A/D1 data register B	ADDRB1	16	H'A44E 000A		16
A/D1 data register C	ADDRC1	16	H'A44E 000C		16
A/D1 data register D	ADDRD1	16	H'A44E 000E		16
A/D0 control/status register	ADCSR0	16	H'A44E 0010		16
A/D1 control/status register	ADCSR1	16	H'A44E 0012		16
A/D0 A/D1 control register	ADCR	16	H'A44E 0014		16
—	—	—	—	—	—
Port A control register	PACR	32	H'A443 0000	PFC	8/16/32
Port B control register	PBCR	32	H'A443 0004		8/16/32
Port C control register	PCCR	32	H'A443 0008		8/16/32
Port D control register	PDCR	32	H'A443 000C		8/16/32
Port E control register	PECR	32	H'A443 0010		8/16/32
Port F control register	PFCR	32	H'A443 0014		8/16/32
Port G control register	PGCR	32	H'A443 0018		8/16/32
Port H control register	PHCR	32	H'A443 001C		8/16/32

Register Name	Abbreviation	Bit No.	Address	Module	Access States
Port J control register	PJCR	32	H'A443 0020	PFC	8/16/32
Port E I/O register	PEIOR	16	H'A443 0038		8/16
Port E MTU R/W enable register	PEMTURWER	16	H'A443 003A		8/16
—	—	—	—	—	—
Port A data register	PADR	16	H'A443 0026	PORT	8/16
Port B data register	PBDR	16	H'A443 0028		8/16
Port C data register	PCDR	16	H'A443 002A		8/16
Port D data register	PDDR	16	H'A443 002C		8/16
Port E data register	PEDR	16	H'A443 002E		8/16
Port F data register	PFDR	16	H'A443 0030		8/16
Port G data register	PGDR	16	H'A443 0032		8/16
Port H data register	PHDR	16	H'A443 0034		8/16
Port J data register	PJDR	16	H'A443 0036		8/16

Notes: 1. This register only accepts 16-bit writing to prevent incorrect writing. In this case, the upper eight bits of the data must be H'5A, otherwise writing cannot be performed. When reading, read from the same address in bytes.

2. This register only accepts 16-bit writing to prevent incorrect writing. In this case, the upper eight bits of the data must be H'A5, otherwise writing cannot be performed. When reading, read from the same address in bytes.

3. This register only accepts 32-bit writing to prevent incorrect writing. In this case, the upper 16 bits of the data must be H'A55A, otherwise writing cannot be performed. When reading, read from the same address in unit of 32 bits. At this time, the upper 16 bits are read as 0s.

24.2 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 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
FRQCR	—	—	—	CKOEN	—	—	STC1	STC0	CPG
	—	—	IFC1	IFC0	—	—	PFC1	PFC0	
WTCNT									WDT
WTCSR	TME	WT/IT	RSTS	WOVF	IOVF	CKS2	CKS1	CKS0	
STBCR	STBY	—	—	—	—	—	—	—	Power-down modes
STBCR2	MSTP10	MSTP9	MSTP8	MSTP7	—	MSTP5	MSTP4	MSTP3	
STBCR3	HIZ	—	MSTP35	—	MSTP33	MSTP32	MSTP31	MSTP30	
STBCR4	—	MSTP46	MSTP45	MSTP44	MSTP43	MSTP42	—	—	
CCR1	—	—	—	—	—	—	—	—	Cache
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	CF	WB	WT	CE	
CCR2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	LE	
	—	—	—	—	—	—	W3LOAD	W3LOCK	
	—	—	—	—	—	—	W2LOAD	W2LOCK	
INTEVT2	—	—	—	—	—	—	—	—	Exception handling
	—	—	—	—	—	—	—	—	
	—	—	—	—					
TRA	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	imm	imm	
	imm	imm	imm	imm	imm	imm	—	—	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
EXPEVT	—	—	—	—	—	—	—	—	Exception handling
	—	—	—	—	—	—	—	—	
	—	—	—	—					
IPRF	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	INTC
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
IPRG	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
IPRH	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
IPRI	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
IMR0	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0	
IMR1	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0	
IMR2	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0	
IMR4	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0	
IMR5	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0	
IMR6	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0	
IMR7	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0	
IMR8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0	
IMR9	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0	
IMR10	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0	
IMCR0	IMC7	IMC6	IMC5	IMC4	IMC3	IMC2	IMC1	IMC0	
IMCR1	IMC7	IMC6	IMC5	IMC4	IMC3	IMC2	IMC1	IMC0	
IMCR2	IMC7	IMC6	IMC5	IMC4	IMC3	IMC2	IMC1	IMC0	
IMCR4	IMC7	IMC6	IMC5	IMC4	IMC3	IMC2	IMC1	IMC0	
IMCR5	IMC7	IMC6	IMC5	IMC4	IMC3	IMC2	IMC1	IMC0	
IMCR6	IMC7	IMC6	IMC5	IMC4	IMC3	IMC2	IMC1	IMC0	
IMCR7	IMC7	IMC6	IMC5	IMC4	IMC3	IMC2	IMC1	IMC0	
IMCR8	IMC7	IMC6	IMC5	IMC4	IMC3	IMC2	IMC1	IMC0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
IMCR9	IMC7	IMC6	IMC5	IMC4	IMC3	IMC2	IMC1	IMC0	INTC
IMCR10	IMC7	IMC6	IMC5	IMC4	IMC3	IMC2	IMC1	IMC0	
IRR0	IRQ7R	IRQ6R	IRQ5R	IRQ4R	IRQ3R	IRQ2R	IRQ1R	IRQ0R	
ICR1	—	IRQE	—	—	IRQ51S	IRQ50S	IRQ41S	IRQ40S	
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S	
ICR3	—	—	—	—	—	—	—	—	
	—	—	—	—	IRQ71S	IRQ70S	IRQ61S	IRQ60S	
IPRC	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
IPRD	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
IPRE	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
IPRJ	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
ICR0	NMIL	—	—	—	—	—	—	NMIE	
	—	—	—	—	—	—	—	—	
IPRB	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	UBC
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16	
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0	
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24	
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16	
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8	
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0	
BRCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—	
	DBEB	PCBB	—	—	SEQ	—	—	ETBE	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
BETR	—	—	—	—	BET11	BET10	BET9	BET8	UBC
	BET7	BET6	BET5	BET4	BET3	BET2	BET1	BET0	
BARB	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BAMRB	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24	
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16	
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8	
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0	
BBRB	—	—	—	—	—	—	XYE	XYS	
	CDB1	CDB0	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0	
BRSR	SVF	—	—	—	BSA27	BSA26	BSA25	BSA24	
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA16	
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8	
	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0	
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
BAMRA	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BAMA24	
	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA16	
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA8	
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA0	
BBRA	—	—	—	—	—	—	—	—	
	CDA1	CDA0	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0	
BRDR	DVF	—	—	—	BDA27	BDA26	BDA25	BDA24	
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16	
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
CMNCR	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	WAITSEL	—	—	MAP	BLOCK	DPRTY1	DPRTY0	DMAIW2	
	DMAIW1	DMAIW0	DMAIWA	—	—	CK2DRV	HIZMEM	HIZCNT	
CS0BCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS2BCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS3BCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS4BCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS5ABCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS5BBCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
CS6ABCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2	BSC
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS6BBCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS0WCR*[1]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS0WCR*[2]	—	—	—	—	—	—	—	—	
	—	—	—	BEN	—	—	BW1	BW0	
	—	—	—	—	—	W3	W2	W1	
	W0	WM	—	—	—	—	—	—	
CS0WCR*[3]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	BW1	BW0	
	—	—	—	—	—	W3	W2	W1	
	W0	WM	—	—	—	—	—	—	
CS2WCR*[1]	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	—	—	—	
	—	—	—	—	—	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	—	—	
CS2WCR*[4]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	A2CL1	
	A2CL0	—	—	—	—	—	—	—	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
CS3WCR*[1]	—	—	—	—	—	—	—	—	BSC
	—	—	—	BAS	—	—	—	—	
	—	—	—	—	—	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	—	—	
CS3WCR*[4]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	WTRP1	WTRP0	—	WTRCD1	WTRCD0	—	A3CL1	
	A3CL0	—	—	TRWL1	TRWL0	—	WTRC1	WTRC0	
CS4WCR*[1]	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	WW2	WW1	WW0	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS4WCR*[2]	—	—	—	—	—	—	—	—	
	—	—	—	BEN	—	—	BW1	BW0	
	—	—	—	SW1	SW0	W3	W2	W1	
	W0	WM	—	—	—	—	HW1	HW0	
CS5AWCR*[1]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	WW2	WW1	WW0	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS5BWCR*[1]	—	—	—	—	—	—	—	—	
	—	—	SZSEL	MPXW/BAS	—	WW2	WW1	WW0	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS6AWCR*[1]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
CS6BWCR*[1]	—	—	—	—	—	—	—	—	BSC
	—	—	—	BAS	—	—	—	—	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS6BWCR*[5]	—	—	—	—	—	—	—	—	
	—	—	MPXAW1	MPXAW0	MPXMD	—	BW1	BW0	
	—	—	—	—	—	W3	W2	W1	
	W0	WM	—	—	—	—	—	—	
SDCR	—	—	—	—	—	—	—	—	
	—	—	—	A2ROW1	A2ROW0	—	A2COL1	A2COL0	
	—	—	DEEP	SLOW	RFSH	RMODE	PDOWN	BACTV	
	—	—	—	A3ROW1	A3ROW0	—	A3COL1	A3COL0	
RTCSR	—	—	—	—	—	—	—	—	
	CMF	—	CKS2	CKS1	CKS0	RRC2	RRC1	RRC0	
RTCNT	—	—	—	—	—	—	—	—	
RTCOR	—	—	—	—	—	—	—	—	
RWTCNT	—	—	—	—	—	—	—	—	
	—								
SAR_0									DMAC
DAR_0									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
DMATCR_0									DMAC
CHCR_0	TC	—	—	—	—	—	—	—	
	DO	TL	—	—	—	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	DL	DS	TB	TS1	TS0	IE	TE	DE	
SAR_1									
DAR_1									
DMATCR_1									
CHCR_1	TC	—	—	—	—	—	—	—	
	DO	—	—	—	—	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	DL	DS	TB	TS1	TS0	IE	TE	DE	
SAR_2									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
DAR_2									DMAC
DMATCR_2									
CHCR_2	TC	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	—	TB	TS1	TS0	IE	TE	DE	
SAR_3									
DAR_3									
DMATCR_3									
CHCR_3	TC	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	—	TB	TS1	TS0	IE	TE	DE	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
DMAOR	—	—	CMS1	CMS0	—	—	PR1	PR0	DMAC
	—	—	—	—	—	AE	NMIF	DME	
	—	—	—	—	—	—	—	—	
	—	—	RC0	RC1	RC2	RC3	—	—	
DMARS0	C1MID5	C1MID4	C1MID3	C1MID2	C1MID1	C1MID0	C1RID1	C1RID0	
	C0MID5	C0MID4	C0MID3	C0MID2	C0MID1	C0MID0	C0RID1	C0RID0	
DMARS1	C3MID5	C3MID4	C3MID3	C3MID2	C3MID1	C3MID0	C3RID1	C3RID0	
	C2MID5	C2MID4	C2MID3	C2MID2	C2MID1	C2MID0	C2RID1	C2RID0	
SDIR	T17	T16	T15	T14	T13	T12	T11	T10	H-UDI
	—	—	—	—	—	—	—	—	
SDIDH	DID31	DID30	DID29	DID28	DID27	DID26	DID25	DID24	
	DID23	DID22	DID21	DID20	DID19	DID18	DID17	DID16	
SDIDL	DID15	DID14	DID13	DID12	DID11	DID10	DID9	DID8	
	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0	
ICCR1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC2
ICCR2	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—	
ICMR	MLS	—	—	—	BCWP	BS2	BC1	BC0	
ICIER	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ	
SAR	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICDRT									
ICDRR									
NF2CYC	—	—	—	—	—	—	—	NF2CYC	
CMSTR_0	—	—	—	—	—	—	—	—	CMT
	—	—	—	—	—	—	—	STR	
CMCSR_0	—	—	—	—	—	—	—	—	
	CMF	—	CMR1	CMR0	—	—	CKS1	CKS0	
CMCNT_0									
CMCOR_0									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
CMSTR_1	—	—	—	—	—	—	—	—	CMT
	—	—	—	—	—	—	—	STR	
CMCSR_1	—	—	—	—	—	—	—	—	
	CMF	—	CMR1	CMR0	—	—	CKS1	CKS0	
CMCNT_1									
CMCOR_1									
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	MTU
TCR_4	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TMDR_3	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TMDR_4	—	—	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	
TIORH_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_4	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_3	TTGE	TGFASEL	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TIER_4	TTGE	TGFASEL	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B	
TOCR	—	PSYE	—	—	—	—	OLSN	OLSP	
TGCR	—	BDC	N	P	FB	WF	VF	UF	
TCNT_3									
TCNT_4									
TCDR									
TDDR									
TGRA_3									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
TGRB_3									MTU
TGRA_4									
TGRB_4									
TCNTS									
TCBR									
TGRC_3									
TGRD_3									
TGRC_4									
TGRD_4									
TSR_3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TSR_4	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0	
TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0	
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_0	TTGE	TGFASEL	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
TGRA_0									MTU
TGRB_0									
TGRC_0									
TGRD_0									
TCR_1	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	TGFASEL	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1									
TGRA_1									
TGRB_1									
TCR_2	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	TGFASEL	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2									
TGRA_2									
TGRB_2									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
ICSR1	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE	MTU
	POE3M1	POE3M0	POE2M1	POE2M0	POE1M1	POE1M0	POE0M1	POE0M0	
OCSR	OSF	—	—	—	—	—	OCE	OIE	
	—	—	—	—	—	—	—	—	
SCSMR_0	—	—	—	—	—	—	—	—	SCIF
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR_0									
SCSCR_0	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_0									
SCFSR_0	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_0									
SCFCR_0	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCFDR_0	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_0	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT	
SCLSR_0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCSMR_1	—	—	—	—	—	—	—	—	
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR_1									
SCSCR_1	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_1									
SCFSR_1	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_1									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
SCFCR_1	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	SCIF
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCFDR_1	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_1	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT	
SCLSR_1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCSMR_2	—	—	—	—	—	—	—	—	
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR_2									
SCSCR_2	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_2									
SCFSR_2	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_2									
SCFCR2	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCFDR2	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR2	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT	
SCLSR2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
USBIFR0	BRST	EP1FULL	EP2TR	EP2EMPTY	SETUPTS	EP0oTS	EP0iTR	EP0iTS	USB
USBIFR1	—	—	—	—	VBUSMN	EP3TR	EP3TS	VBUSF	
USBEPDR0i	D7	D6	D5	D4	D3	D2	D1	D0	
USBEPDR0o	D7	D6	D5	D4	D3	D2	D1	D0	
USBTRG	—	EP3PKTE	EP1RDFN	EP2PKTE	—	EP0sRDFN	EP0oRDFN	EP0iPKTE	
USBFCLR	—	EP3CLR	EP1CLR	EP2CLR	—	—	EP0oCLR	EP0iCLR	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
USBEPSZ0o	—	—	—	—	—	—	—	—	USB
USBEPDR0s	D7	D6	D5	D4	D3	D2	D1	D0	
USBDASTS	—	—	EP3DE	EP2DE	—	—	—	EP0iDE	
USBISR0	BRST	EP1FULL	EP2TR	EP2EMPTY	SETUPTS	EP0oTS	EP0iTR	EP0iTS	
USBEPSTL	—	—	—	ASCE	EP3STL	EP2STL	EP1STL	EP0STL	
USBIER0	BRST	EP1FULL	EP2TR	EP2EMPTY	SETUPTS	EP0oTS	EP0iTR	EP0iTS	
USBIER1	—	—	—	—	—	EP3TR	EP3TS	VBUSF	
USBEPSZ1	—	—	—	—	—	—	—	—	
USBISR1	—	—	—	—	—	EP3TR	EP3TS	VBUSF	
USBDMAR	—	—	—	—	—	—	EP2DMAE	EP1DMAE	
USBEPDR3	D7	D6	D5	D4	D3	D2	D1	D0	
USBEPDR1	D7	D6	D5	D4	D3	D2	D1	D0	
USBEPDR2	D7	D6	D5	D4	D3	D2	D1	D0	
USBXVERCR	—	—	—	—	—	—	—	XVEROFF	
USBIFR2	—	—	—	—	AWAKE	SUSPS	CFGV	SETC	
USBIER2	—	—	—	—	—	—	—	SETC	
USBCTRL	—	—	—	—	—	—	SUSPEND	PWMD	
ADDRA0									ADC
			—	—	—	—	—	—	
ADDRB0									
			—	—	—	—	—	—	
ADDRC0									
			—	—	—	—	—	—	
ADDRD0									
			—	—	—	—	—	—	
ADDRA1									
			—	—	—	—	—	—	
ADDRB1									
			—	—	—	—	—	—	
ADDRC1									
			—	—	—	—	—	—	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
ADDRD1									ADC
			—	—	—	—	—	—	
ADCSR0	ADF	ADIE	ADST	DMASL	TRGE	—	—	—	
	CKS1	CKS0	MULTI1	MULTI0	—	—	CH1	CH0	
ADCSR1	ADF	ADIE	ADST	DMASL	TRGE	—	—	—	
	CKS1	CKS0	MULTI1	MULTI0	—	—	CH1	CH0	
ADCR	DSMP	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
PACR	—	—	PA14MD2	PA14MD1	PA13MD2	PA13MD1	PA12MD2	PA12MD1	PFC
	PA11MD2	PA11MD1	PA10MD2	PA10MD1	PA9MD2	PA9MD1	PA8MD2	PA8MD1	
	PA7MD2	PA7MD1	PA6MD2	PA6MD1	PA5MD2	PA5MD1	PA4MD2	PA4MD1	
	PA3MD2	PA3MD1	PA2MD2	PA2MD1	PA1MD2	PA1MD1	PA0MD2	PA0MD1	
PBCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	PB8MD2	PB8MD1	
	PB7MD2	PB7MD1	PB6MD2	PB6MD1	PB5MD2	PB5MD1	PB4MD2	PB4MD1	
	PB3MD2	PB3MD1	PB2MD2	PB2MD1	PB1MD2	PB1MD1	PB0MD2	PB0MD1	
PCCR	PC15MD2	PC15MD1	PC14MD2	PC14MD1	PC13MD2	PC13MD1	PC12MD2	PC12MD1	
	PC11MD2	PC11MD1	PC10MD2	PC10MD1	PC9MD2	PC9MD1	PC8MD2	PC8MD1	
	PC7MD2	PC7MD1	PC6MD2	PC6MD1	PC5MD2	PC5MD1	PC4MD2	PC4MD1	
	PC3MD2	PC3MD1	PC2MD2	PC2MD1	PC1MD2	PC1MD1	PC0MD2	PC0MD1	
PDCR	PD15MD2	PD15MD1	PD14MD2	PD14MD1	PD13MD2	PD13MD1	PD12MD2	PD12MD1	
	PD11MD2	PD11MD1	PD10MD2	PD10MD1	PD9MD2	PD9MD1	PD8MD2	PD8MD1	
	PD7MD2	PD7MD1	PD6MD2	PD6MD1	PD5MD2	PD5MD1	PD4MD2	PD4MD1	
	PD3MD2	PD3MD1	PD2MD2	PD2MD1	PD1MD2	PD1MD1	PD0MD2	PD0MD1	
PECR	PE15MD2	PE15MD1	PE14MD2	PE14MD1	PE13MD2	PE13MD1	PE12MD2	PE12MD1	
	PE11MD2	PE11MD1	PE10MD2	PE10MD1	PE9MD2	PE9MD1	PE8MD2	PE8MD1	
	PE7MD2	PE7MD1	PE6MD2	PE6MD1	PE5MD2	PE5MD1	PE4MD2	PE4MD1	
	PE3MD2	PE3MD1	PE2MD2	PE2MD1	PE1MD2	PE1MD1	PE0MD2	PE0MD1	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
PFCR	PF15MD2	PF15MD1	PF14MD2	PF14MD1	PF13MD2	PF13MD1	PF12MD2	PF12MD1	PFC
	PF11MD2	PF11MD1	PF10MD2	PF10MD1	PF9MD2	PF9MD1	PF8MD2	PF8MD1	
	PF7MD2	PF7MD1	PF6MD2	PF6MD1	PF5MD2	PF5MD1	PF4MD2	PF4MD1	
	PF3MD2	PF3MD1	PF2MD2	PF2MD1	PF1MD2	PF1MD1	PF0MD2	PF0MD1	
PGCR	—	—	—	—	PG13MD2	PG13MD1	PG12MD2	PG12MD1	
	PG11MD2	PG11MD1	PG10MD2	PG10MD1	PG9MD2	PG9MD1	PG8MD2	PG8MD1	
	PG7MD2	PG7MD1	PG6MD2	PG6MD1	PG5MD2	PG5MD1	PG4MD2	PG4MD1	
	PG3MD2	PG3MD1	PG2MD2	PG2MD1	PG1MD2	PG1MD1	PG0MD2	PG0MD1	
PHCR	—	—	PH14MD2	PH14MD1	PH13MD2	PH13MD1	PH12MD2	PH12MD1	
	PH11MD2	PH11MD1	PH10MD2	PH10MD1	PH9MD2	PH9MD1	PH8MD2	PH8MD1	
	PH7MD2	PH7MD1	PH6MD2	PH6MD1	PH5MD2	PH5MD1	PH4MD2	PH4MD1	
	PH3MD2	PH3MD1	PH2MD2	PH2MD1	PH1MD2	PH1MD1	PH0MD2	PH0MD1	
PJCR	—	—	—	—	—	—	PJ12MD2	PJ12MD1	
	PJ11MD2	PJ11MD1	PJ10MD2	PJ10MD1	PJ9MD2	PJ9MD1	PJ8MD2	PJ8MD1	
	PJ7MD2	PJ7MD1	PJ6MD2	PJ6MD1	PJ5MD2	PJ5MD1	PJ4MD2	PJ4MD1	
	PJ3MD2	PJ3MD1	PJ2MD2	PJ2MD1	PJ1MD2	PJ1MD1	PJ0MD2	PJ0MD1	
PEIOR	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR	
	PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR	
PEMTURWER	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	MTURWE	
PADR	—	PA14DT	PA13DT	PA12DT	PA11DT	PA10DT	PA9DT	PA8DT	PORT
	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT	
PBDR	—	—	—	—	—	—	—	PB8DT	
	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT	
PCDR	PC15DT	PC14DT	PC13DT	PC12DT	PC11DT	PC10DT	PC9DT	PC8DT	
	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT	
PDDR	PD15DT	PD14DT	PD13DT	PD12DT	PD11DT	PD10DT	PD9DT	PD8DT	
	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT	
PEDR	PE15DT	PE14DT	PE13DT	PE12DT	PE11DT	PE10DT	PE9DT	PE8DT	
	PE7DT	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE0DT	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
PFDR	PF15DT	PF14DT	PF13DT	PF12DT	PF11DT	PF10DT	PF9DT	PF8DT	PORT
	PF7DT	PF6DT	PF5DT	PF4DT	PF3DT	PF2DT	PF1DT	PF0DT	
PGDR	—	—	PG13DT	PG12DT	PG11DT	PG10DT	PG9DT	PG8DT	
	PG7DT	PG6DT	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG0DT	
PHDR	—	PH14DT	PH13DT	PH12DT	PH11DT	PH10DT	PH9DT	PH8DT	
	PH7DT	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT	
PJDR	—	—	—	PJ12DT	PJ11DT	PJ10DT	PJ9DT	PJ8DT	
	PJ7DT	PJ6DT	PJ5DT	PJ4DT	PJ3DT	PJ2DT	PJ1DT	PJ0DT	

Notes: 1. When the following memory is in use: normal memory, byte-selection SRAM, or MPX-IO (address/data multiplexed I/O)
2. When burst ROM (asynchronous) is in use
3. When burst ROM (synchronous) is in use
4. When SDRAM is in use
5. When burst MPX-IO is in use

24.3 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep	Module
FRQCR	Initialized*[1]	Retained	Retained	—	Retained	CPG
WTCNT	Initialized*[1]	Retained	Retained	—	Retained	WDT
WTCSR	Initialized*[1]	Retained	Retained	—	Retained	
STBCR	Initialized	Retained	Retained	—	Retained	Power-down modes
STBCR2	Initialized	Retained	Retained	—	Retained	
STBCR3	Initialized	Retained	Retained	—	Retained	
STBCR4	Initialized	Retained	Retained	—	Retained	
CCR1	Initialized	Initialized	Retained	Retained	Retained	Cache
CCR2	Initialized	Initialized	Retained	Retained	Retained	
INTEVT2	Initialized	Initialized	Retained	Retained	Retained	Exception handling
TRA	Initialized	Initialized	Retained	Retained	Retained	
EXPEVT	Initialized	Initialized	Retained	Retained	Retained	
IPRF	Initialized	Initialized	Retained	—	Retained	INTC
IPRG	Initialized	Initialized	Retained	—	Retained	
IPRH	Initialized	Initialized	Retained	—	Retained	
IPRI	Initialized	Initialized	Retained	—	Retained	
IMR0	Initialized	Initialized	Retained	—	Retained	
IMR1	Initialized	Initialized	Retained	—	Retained	
IMR2	Initialized	Initialized	Retained	—	Retained	
IMR4	Initialized	Initialized	Retained	—	Retained	
IMR5	Initialized	Initialized	Retained	—	Retained	
IMR6	Initialized	Initialized	Retained	—	Retained	
IMR7	Initialized	Initialized	Retained	—	Retained	
IMR8	Initialized	Initialized	Retained	—	Retained	
IMR9	Initialized	Initialized	Retained	—	Retained	
IMR10	Initialized	Initialized	Retained	—	Retained	

Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep	Module
IMCR0	Initialized	Initialized	Retained	—	Retained	INTC
IMCR1	Initialized	Initialized	Retained	—	Retained	
IMCR2	Initialized	Initialized	Retained	—	Retained	
IMCR4	Initialized	Initialized	Retained	—	Retained	
IMCR5	Initialized	Initialized	Retained	—	Retained	
IMCR6	Initialized	Initialized	Retained	—	Retained	
IMCR7	Initialized	Initialized	Retained	—	Retained	
IMCR8	Initialized	Initialized	Retained	—	Retained	
IMCR9	Initialized	Initialized	Retained	—	Retained	
IMCR10	Initialized	Initialized	Retained	—	Retained	
IRR0	Initialized	Initialized	Retained	—	Retained	
ICR1	Initialized	Initialized	Retained	—	Retained	
ICR3	Initialized	Initialized	Retained	—	Retained	
IPRC	Initialized	Initialized	Retained	—	Retained	
IPRD	Initialized	Initialized	Retained	—	Retained	
IPRE	Initialized	Initialized	Retained	—	Retained	
IPRJ	Initialized	Initialized	Retained	—	Retained	
ICR0	Initialized	Initialized	Retained	—	Retained	
IPRB	Initialized	Initialized	Retained	—	Retained	
BDRB	Initialized	Retained	Retained	Retained	Retained	UBC
BDMRB	Initialized	Retained	Retained	Retained	Retained	
BRCR	Initialized	Retained	Retained	Retained	Retained	
BETR	Initialized	Retained	Retained	Retained	Retained	
BARB	Initialized	Retained	Retained	Retained	Retained	
BAMRB	Initialized	Retained	Retained	Retained	Retained	
BBRB	Initialized	Retained	Retained	Retained	Retained	
BRSR	Undefined*[2]	Retained	Retained	Retained	Retained	
BARA	Initialized	Retained	Retained	Retained	Retained	
BAMRA	Initialized	Retained	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep	Module
BBRA	Initialized	Retained	Retained	Retained	Retained	UBC
BRDR	Undefined*[2]	Retained	Retained	Retained	Retained	
CMNCR	Initialized	Retained	Retained	—	Retained	BSC
CS0BCR	Initialized	Retained	Retained	—	Retained	
CS2BCR	Initialized	Retained	Retained	—	Retained	
CS3BCR	Initialized	Retained	Retained	—	Retained	
CS4BCR	Initialized	Retained	Retained	—	Retained	
CS5ABCR	Initialized	Retained	Retained	—	Retained	
CS5BBCR	Initialized	Retained	Retained	—	Retained	
CS6ABCR	Initialized	Retained	Retained	—	Retained	
CS6BBCR	Initialized	Retained	Retained	—	Retained	
CS0WCR	Initialized	Retained	Retained	—	Retained	
CS2WCR	Initialized	Retained	Retained	—	Retained	
CS3WCR	Initialized	Retained	Retained	—	Retained	
CS4WCR	Initialized	Retained	Retained	—	Retained	
CS5AWCR	Initialized	Retained	Retained	—	Retained	
CS5BWCR	Initialized	Retained	Retained	—	Retained	
CS6AWCR	Initialized	Retained	Retained	—	Retained	
CS6BWCR	Initialized	Retained	Retained	—	Retained	
SDCR	Initialized	Retained	Retained	—	Retained	
RTCSR	Initialized	Retained	Retained	—	Retained	
RTCNT	Initialized	Retained	Retained	—	Retained	
RTCOR	Initialized	Retained	Retained	—	Retained	
RWTCNT	Initialized	Retained	Retained	—	Retained	
SAR_0	Undefined	Undefined	Retained	Retained	Retained	DMAC
DAR_0	Undefined	Undefined	Retained	Retained	Retained	
DMATCR_0	Undefined	Undefined	Retained	Retained	Retained	
CHCR_0	Initialized	Initialized	Retained	Retained	Retained	
SAR_1	Undefined	Undefined	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep	Module
DAR_1	Undefined	Undefined	Retained	Retained	Retained	DMAC
DMATCR_1	Undefined	Undefined	Retained	Retained	Retained	
CHCR_1	Initialized	Initialized	Retained	Retained	Retained	
SAR_2	Undefined	Undefined	Retained	Retained	Retained	
DAR_2	Undefined	Undefined	Retained	Retained	Retained	
DMATCR_2	Undefined	Undefined	Retained	Retained	Retained	
CHCR_2	Initialized	Initialized	Retained	Retained	Retained	
SAR_3	Undefined	Undefined	Retained	Retained	Retained	
DAR_3	Undefined	Undefined	Retained	Retained	Retained	
DMATCR_3	Undefined	Undefined	Retained	Retained	Retained	
CHCR_3	Initialized	Initialized	Retained	Retained	Retained	
DMAOR	Initialized	Initialized	Retained	Retained	Retained	
DMARS0	Initialized	Initialized	Retained	Retained	Retained	
DMARS1	Initialized	Initialized	Retained	Retained	Retained	
SDIR	Initialized*[4]	Retained	Retained	Retained	Retained	H-UDI
SDIDH	Initialized	Retained	Retained	Retained	Retained	
SDIDL	Initialized	Retained	Retained	Retained	Retained	
ICCR1	Initialized	Retained	Retained	Retained	Retained	IIC2
ICCR2	Initialized	Retained	Retained	Retained	Retained	
ICMR	Initialized	Retained	Retained	Retained	Retained	
ICIER	Initialized	Retained	Retained	Retained	Retained	
ICSR	Initialized	Retained	Retained	Retained	Retained	
SAR	Initialized	Retained	Retained	Retained	Retained	
ICDRT	Undefined	Retained	Retained	Retained	Retained	
ICDRR	Undefined	Retained	Retained	Retained	Retained	
NF2CYC	Initialized	Retained	Retained	Retained	Retained	
CMSTR_0	Initialized	Retained	Retained	Retained	Retained	CMT
CMCSR_0	Initialized	Retained	Retained	Retained	Retained	
CMCNT_0	Initialized	Retained	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep	Module
CMCOR_0	Initialized	Retained	Retained	Retained	Retained	CMT
CMSTR_1	Initialized	Retained	Retained	Retained	Retained	
CMCSR_1	Initialized	Retained	Retained	Retained	Retained	
CMCNT_1	Initialized	Retained	Retained	Retained	Retained	
CMCOR_1	Initialized	Retained	Retained	Retained	Retained	
TCR_3	Initialized	Retained	Initialized	Initialized	Retained	MTU
TCR_4	Initialized	Retained	Initialized	Initialized	Retained	
TMDR_3	Initialized	Retained	Initialized	Initialized	Retained	
TMDR_4	Initialized	Retained	Initialized	Initialized	Retained	
TIORH_3	Initialized	Retained	Initialized	Initialized	Retained	
TIORL_3	Initialized	Retained	Initialized	Initialized	Retained	
TIORH_4	Initialized	Retained	Initialized	Initialized	Retained	
TIORL_4	Initialized	Retained	Initialized	Initialized	Retained	
TIER_3	Initialized	Retained	Initialized	Initialized	Retained	
TIER_4	Initialized	Retained	Initialized	Initialized	Retained	
TOER	Initialized	Retained	Initialized	Initialized	Retained	
TOCR	Initialized	Retained	Initialized	Initialized	Retained	
TGCR	Initialized	Retained	Initialized	Initialized	Retained	
TCNT_3	Initialized	Retained	Initialized	Initialized	Retained	
TCNT_4	Initialized	Retained	Initialized	Initialized	Retained	
TCDR	Initialized	Retained	Initialized	Initialized	Retained	
TDDR	Initialized	Retained	Initialized	Initialized	Retained	
TGRA_3	Initialized	Retained	Initialized	Initialized	Retained	
TGRB_3	Initialized	Retained	Initialized	Initialized	Retained	
TGRA_4	Initialized	Retained	Initialized	Initialized	Retained	
TGRB_4	Initialized	Retained	Initialized	Initialized	Retained	
TCNTS	Initialized	Retained	Initialized	Initialized	Retained	
TCBR	Initialized	Retained	Initialized	Initialized	Retained	
TGRC_3	Initialized	Retained	Initialized	Initialized	Retained	

Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep	Module
TGRD_3	Initialized	Retained	Initialized	Initialized	Retained	MTU
TGRC_4	Initialized	Retained	Initialized	Initialized	Retained	
TGRD_4	Initialized	Retained	Initialized	Initialized	Retained	
TSR_3	Initialized	Retained	Initialized	Initialized	Retained	
TSR_4	Initialized	Retained	Initialized	Initialized	Retained	
TSTR	Initialized	Retained	Initialized	Initialized	Retained	
TSYR	Initialized	Retained	Initialized	Initialized	Retained	
TCR_0	Initialized	Retained	Initialized	Initialized	Retained	
TMDR_0	Initialized	Retained	Initialized	Initialized	Retained	
TIORH_0	Initialized	Retained	Initialized	Initialized	Retained	
TIORL_0	Initialized	Retained	Initialized	Initialized	Retained	
TIER_0	Initialized	Retained	Initialized	Initialized	Retained	
TSR_0	Initialized	Retained	Initialized	Initialized	Retained	
TCNT_0	Initialized	Retained	Initialized	Initialized	Retained	
TGRA_0	Initialized	Retained	Initialized	Initialized	Retained	
TGRB_0	Initialized	Retained	Initialized	Initialized	Retained	
TGRC_0	Initialized	Retained	Initialized	Initialized	Retained	
TGRD_0	Initialized	Retained	Initialized	Initialized	Retained	
TCR_1	Initialized	Retained	Initialized	Initialized	Retained	
TMDR_1	Initialized	Retained	Initialized	Initialized	Retained	
TIOR_1	Initialized	Retained	Initialized	Initialized	Retained	
TIER_1	Initialized	Retained	Initialized	Initialized	Retained	
TSR_1	Initialized	Retained	Initialized	Initialized	Retained	
TCNT_1	Initialized	Retained	Initialized	Initialized	Retained	
TGRA_1	Initialized	Retained	Initialized	Initialized	Retained	
TGRB_1	Initialized	Retained	Initialized	Initialized	Retained	
TCR_2	Initialized	Retained	Initialized	Initialized	Retained	
TMDR_2	Initialized	Retained	Initialized	Initialized	Retained	
TIOR_2	Initialized	Retained	Initialized	Initialized	Retained	

Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep	Module
TIER_2	Initialized	Retained	Initialized	Initialized	Retained	MTU
TSR_2	Initialized	Retained	Initialized	Initialized	Retained	
TCNT_2	Initialized	Retained	Initialized	Initialized	Retained	
TGRA_2	Initialized	Retained	Initialized	Initialized	Retained	
TGRB_2	Initialized	Retained	Initialized	Initialized	Retained	
ICSR1	Initialized	Retained	Retained	Retained	Retained	
OCSR	Initialized	Retained	Retained	Retained	Retained	
SCSMR_0	Initialized	Retained	Retained	Retained	Retained	SCIF
SCBRR_0	Initialized	Retained	Retained	Retained	Retained	
SCSCR_0	Initialized	Retained	Retained	Retained	Retained	
SCFTDR_0	Undefined	Retained	Retained	Retained	Retained	
SCFSR_0	Initialized	Retained	Retained	Retained	Retained	
SCFRDR_0	Undefined	Retained	Retained	Retained	Retained	
SCFCR_0	Initialized	Retained	Retained	Retained	Retained	
SCFDR_0	Initialized	Retained	Retained	Retained	Retained	
SCSPTR_0	Initialized	Retained	Retained	Retained	Retained	
SCLSR_0	Initialized	Retained	Retained	Retained	Retained	
SCSMR_1	Initialized	Retained	Retained	Retained	Retained	
SCBRR_1	Initialized	Retained	Retained	Retained	Retained	
SCSCR_1	Initialized	Retained	Retained	Retained	Retained	
SCFTDR_1	Undefined	Retained	Retained	Retained	Retained	
SCFSR_1	Initialized	Retained	Retained	Retained	Retained	
SCFRDR_1	Undefined	Retained	Retained	Retained	Retained	
SCFCR_1	Initialized	Retained	Retained	Retained	Retained	
SCFDR_1	Initialized	Retained	Retained	Retained	Retained	
SCSPTR_1	Initialized	Retained	Retained	Retained	Retained	
SCLSR_1	Initialized	Retained	Retained	Retained	Retained	
SCSMR_2	Initialized	Retained	Retained	Retained	Retained	
SCBRR_2	Initialized	Retained	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep	Module
SCSCR_2	Initialized	Retained	Retained	Retained	Retained	SCIF
SCFTDR_2	Undefined	Retained	Retained	Retained	Retained	
SCFSR_2	Initialized	Retained	Retained	Retained	Retained	
SCFRDR_2	Undefined	Retained	Retained	Retained	Retained	
SCFCR_2	Initialized	Retained	Retained	Retained	Retained	
SCFDR_2	Initialized	Retained	Retained	Retained	Retained	
SCSPTR_2	Initialized	Retained	Retained	Retained	Retained	
SCLSR_2	Initialized	Retained	Retained	Retained	Retained	
USBIFR0	Initialized	Retained	Retained	Retained	Retained	USB
USBIFR1	Initialized	Retained	Retained	Retained	Retained	
USBEPDR0i	Undefined	Retained	Retained	Retained	Retained	
USBEPDR0o	Undefined	Retained	Retained	Retained	Retained	
USBTRG	Initialized	Retained	Retained	Retained	Retained	
USBFCLR	Initialized	Retained	Retained	Retained	Retained	
USBEPSZ0o	Initialized	Retained	Retained	Retained	Retained	
USBEPDR0s	Undefined	Retained	Retained	Retained	Retained	
USBDASTS	Initialized	Retained	Retained	Retained	Retained	
USBISR0	Initialized	Retained	Retained	Retained	Retained	
USBEPSTL	Initialized	Retained	Retained	Retained	Retained	
USBIER0	Initialized	Retained	Retained	Retained	Retained	
USBIER1	Initialized	Retained	Retained	Retained	Retained	
USBEPSZ1	Initialized	Retained	Retained	Retained	Retained	
USBISR1	Initialized	Retained	Retained	Retained	Retained	
USBDMAR	Initialized	Retained	Retained	Retained	Retained	
USBEPDR3	Undefined	Retained	Retained	Retained	Retained	
USBEPDR1	Undefined	Retained	Retained	Retained	Retained	
USBEPDR2	Undefined	Retained	Retained	Retained	Retained	
USBXVERCR	Initialized	Retained	Retained	Retained	Retained	
USBIFR2	Initialized	Retained	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep	Module
USBIER2	Initialized	Retained	Retained	Retained	Retained	USB
USBCTRL	Initialized	Retained	Retained	Retained	Retained	
ADDRA0	Initialized	Retained	Initialized	Initialized	Retained	ADC
ADDRB0	Initialized	Retained	Initialized	Initialized	Retained	
ADDRC0	Initialized	Retained	Initialized	Initialized	Retained	
ADDRD0	Initialized	Retained	Initialized	Initialized	Retained	
ADDRA1	Initialized	Retained	Initialized	Initialized	Retained	
ADDRB1	Initialized	Retained	Initialized	Initialized	Retained	
ADDRC1	Initialized	Retained	Initialized	Initialized	Retained	
ADDRD1	Initialized	Retained	Initialized	Initialized	Retained	
ADCSR0	Initialized	Retained	Initialized	Initialized	Retained	
ADCSR1	Initialized	Retained	Initialized	Initialized	Retained	
ADCR	Initialized	Retained	Initialized	Initialized	Retained	
PACR	Initialized	Retained	Retained	—	Retained	PFC
PBCR	Initialized	Retained	Retained	—	Retained	
PCCR	Initialized	Retained	Retained	—	Retained	
PDCR	Initialized	Retained	Retained	—	Retained	
PECR	Initialized	Retained	Retained	—	Retained	
PFCR	Initialized	Retained	Retained	—	Retained	
PGCR	Initialized	Retained	Retained	—	Retained	
PHCR	Initialized	Retained	Retained	—	Retained	
PJCR	Initialized	Retained	Retained	—	Retained	
PEIOR	Initialized	Retained	Retained	—	Retained	
PEMTURWER	Initialized	Retained	Retained	—	Retained	
PADR	Initialized	Retained	Retained	—	Retained	PORT
PBDR	Initialized	Retained	Retained	—	Retained	
PCDR	Initialized	Retained	Retained	—	Retained	
PDDR	Initialized	Retained	Retained	—	Retained	
PEDR	Initialized	Retained	Retained	—	Retained	

Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep	Module
PFDR	Initialized	Retained	Retained	—	Retained	PORT
PGDR	Initialized*[3]	Retained	Retained	—	Retained	
PHDR	Initialized	Retained	Retained	—	Retained	
PJDR	Initialized	Retained	Retained	—	Retained	

Notes: 1. Not initialized by a power-on reset.
2. Some bits are initialized.
3. Some bits are not initialized.
4. Initialized by $\overline{\text{TRST}}$ assertion or when the TAP controller is in the test-logic-reset state.

Section 25 Electrical Characteristics

The specifications shown in this section are preliminary. After the characteristics have been evaluated, the specifications may be changed without notice.

25.1 Absolute Maximum Ratings

Table 25.1 lists the absolute maximum ratings.

Table 25.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage (I/O)	$V_{CC}Q$	–0.3 to 3.8	V
Power supply voltage (Internal)	V_{CC} V_{CC} (PLL1) V_{CC} (PLL2)	–0.3 to 2.1	V
Input voltage (other than ports G7 to G0)	V_{in}	–0.3 to $V_{CC}Q$ +0.3	V
Input voltage (ports G7 to G0)	V_{in}	–0.3 to AV_{CC} (A/D) +0.3	V
Analog power supply voltage (A/D)	AV_{CC} (A/D)	–0.3 to 3.8	V
Analog input voltage (A/D)	V_{AN}	–0.3 to AV_{CC} (A/D) +0.3	V
Operating temperature	T_{opr}	–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.

25.1.1 Power-On Sequence

Supply the power so that VccQ (3.3-V system) and Vcc (1.8-V system) are supplied simultaneously or Vcc is supplied after VccQ is supplied.

Recommended values for the power-on procedure are shown below.

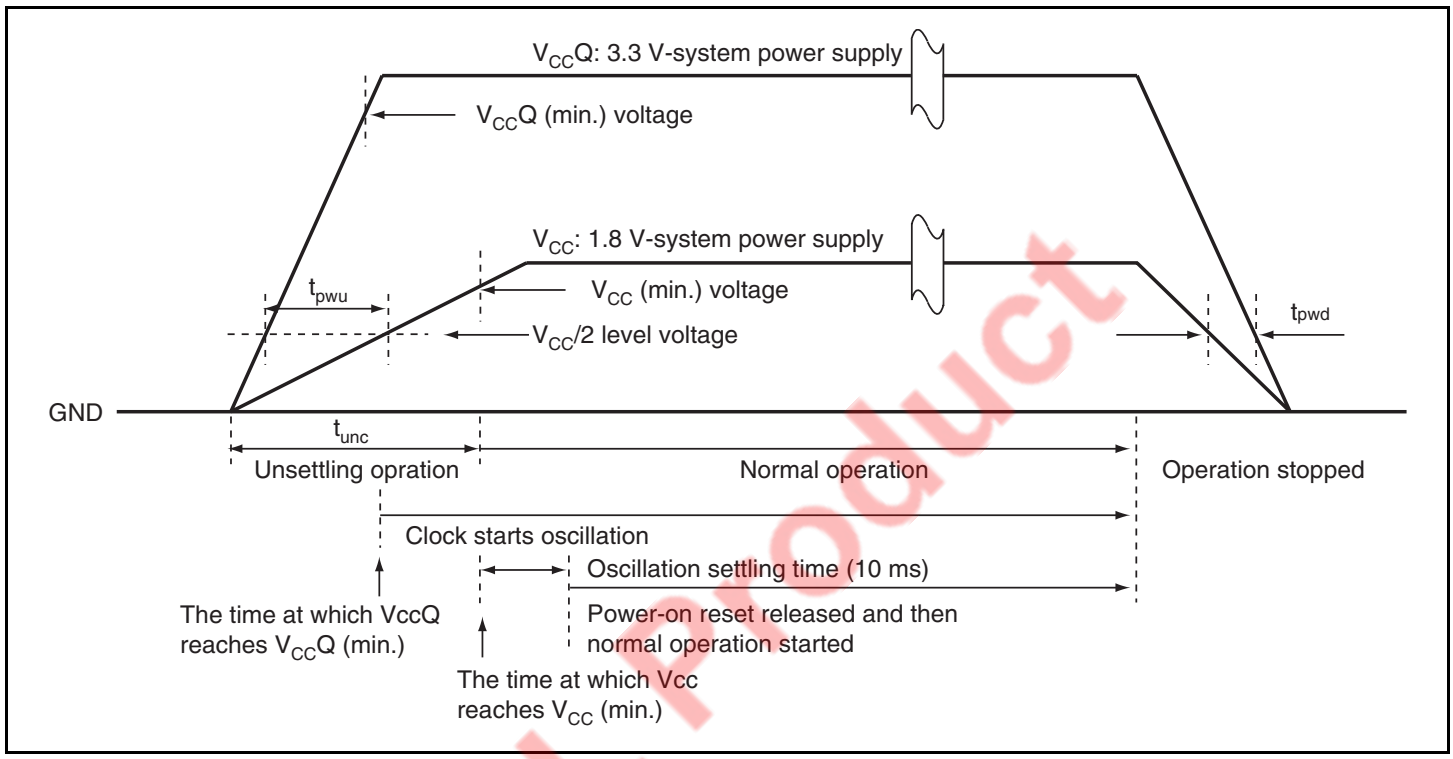


Figure 25.1 Power-On Sequence

Table 25.2 Recommended Values for Power-On/Off Sequence

Item	Symbol	Max. Permissible Value	Unit
Time lag between VccQ and Vcc when turning on	tpwu	1	ms
Time lag between VccQ and Vcc when turning off	tpwd	1	ms
Unsettling operation time	tunc	100	ms

Notes:
1. The figures shown above are recommended values, so they represent guidelines rather than strict requirements.
2. The system design must, however, ensure that the undefined states of internal circuits and pin states do not cause erroneous system operation.
3. The negative values in the maximum permissible value column indicate the allowed difference in the time the voltages supplied as VccQ and Vcc take to rise. Therefore, these figures do not allow the power supply in the reverse sequence: Vcc then VccQ.
4. When Vcc (1.8-V power) rises more quickly than VccQ (3.3-V power), the figure in the maximum permissible value column is a negative value.
5. The time over which the internal state is undefined means time over which the first supplying of power is in a transient state.
6. The pin states become defined when VccQ (min.) is reached. The power-on reset ($\overline{\text{RESETP}}$) is accepted after Vcc reaches Vcc (min.) and after the clock oscillation settling time elapsed.
7. Ensure that the period over which the internal state is undefined is less than or equal to 100 ms.

25.2 DC Characteristics

Tables 25.3 and 25.4 list DC characteristics.

Table 25.3 DC Characteristics (1) [Common Items]

Conditions: Ta = –40°C to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Current consumption*[1]	Normal operation	I_{CC}*[2]	—	300	400	mA	V_{CC} = 1.8 V Iϕ = 100 MHz Pϕ = 33 MHz
		$I_{CC}Q$*[3]	—	10	20	mA	$V_{CC}Q$ = 3.3 V Bϕ = 50 MHz
	Standby mode	I_{stby}*[2]	—	200	1000	µA	Ta = 25°C
		$I_{stby}Q$*[3]	—	5	20	µA	$V_{CC}Q$ = 3.3 V
	Sleep mode	I_{sleep}*[2]	—	50	110	mA	V_{CC} = 1.8 V Bϕ = 50 MHz Pϕ = 33 MHz
Input leakage current	All input pins	$\|I_{in}\|$	—	—	1.0	µA	V_{in} = 0.5 to $V_{CC}Q$ – 0.5 V
Three-state leakage current	All input/output pins, all output pins (except for pins with weak keeper) (off state)	$\|I_{STI}\|$	—	—	1.0	µA	Vin = 0.5 to $V_{CC}Q$ – 0.5 V
Input capacitance	All pins	C_{in}	—	—	20	pF	
Analog power supply voltage (A/D)		AV_{CC} (AD)	3.0	3.3	3.6	V	
Analog power supply current (A/D)	During A/D conversion	AI_{CC} (AD)	—	2	5	mA	
	Idle		—	600	1000	µA	

Caution: When the A/D converter is not in use, the AV_{CC} and AV_{SS} pins should not be open.

Note: 1. Current consumption values are when all output pins are unloaded.

2. I_{CC} I_{sleep} and I_{stby}, respectively, represents the total currents consumed in each Vcc, V_{CC} (PLL1) and Vcc (PLL2)

3. $I_{CC}Q$ and $I_{stby}Q$, respectively, represents the total currents consumed in each $V_{CC}Q$ and AV_{CC}.

Table 25.3 DC Characteristics (2) [Except for I²C- and USB-Related Pins]

Conditions: V_{CC} = V_{CC} (PLL1, PLL2) = 1.8 V ±5%, $V_{CC}Q$ = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{SS} = V_{SS} (PLL1, PLL2) = AV_{SS} = 0 V, Ta = –40°C to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply		$V_{CC}Q$	3.0	3.3	3.6	V	
		V_{CC} V_{CC} (PLL1) V_{CC} (PLL2)	1.71	1.8	1.89		
Input high voltage	$\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, $\overline{\text{NMI}}$, MD3, MD2 MD0, $\overline{\text{ASEMD0}}$, $\overline{\text{TRST}}$	V_{IH}	$V_{CC}Q \times 0.9$	—	$V_{CC}Q + 0.3$	V	
	EXTAL, CKIO		$V_{CC}Q - 0.3$	—	$V_{CC}Q + 0.3$		
	Ports G7 to G0		2.3	—	$V_{CC}Q + 0.3$		
	Input pins other than above (excluding Schmitt pins)		2.3	—	$V_{CC}Q + 0.3$		
Input low voltage	$\overline{\text{RESETP}}$, TCK, $\overline{\text{RESETM}}$, $\overline{\text{NMI}}$, MD3, MD2 MD0, $\overline{\text{ASEMD0}}$, $\overline{\text{TRST}}$	V_{IL}	–0.3	—	$V_{CC}Q \times 0.1$	V	
	EXTAL, CKIO,		–0.3	—	$V_{CC}Q \times 0.2$		
	Ports G7 to G0		–0.3	—	$V_{CC}Q \times 0.2$		
	Input pins other than above (excluding Schmitt pins)		–0.3	—	$V_{CC}Q \times 0.2$		

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input characteristics	TIOC0A to TIOC0D, TIOC1A, TIOC1B, TIOC2A, TIOC2B, TIOC3A to TIOC3D, TIOC4A to TIOC4D, TCLKA to TCLKD, SCK0 to SCK2, RxD0 to RxD2, $\overline{\text{CTS0}}$ to $\overline{\text{CTS2}}$, $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	V_T^+	$V_{CC}Q \times 0.9$	—	—	V	
		V_T^-	—	—	$V_{CC}Q \times 0.2$	V	
		$V_T^+ - V_T^-$	$V_{CC}Q \times 0.05$	—	—	V	
Output high voltage	All output pins*	V_{OH}	2.4	—	—	V	I_{OH} = –200 µA
			2.0	—	—		I_{OH} = –2 mA
Output low voltage	PE0 to PE4, PE6	V_{OL}	—	—	1.5	V	I_{OL} = 15 mA
	All pins except for above pins, SCL, SDA*		—	—	0.4		I_{OL} = 2.0 mA
RAM standby voltage		V_{RAM}	1.0	—	—	V	Measured by Vcc (= PLL1, PLL2) as parameter

Note: * The SCL and SDA pins (open-drain pins)
When the port functions are selected as the general inputs or outputs, however, the outputs of these pins show the usual V_{OH}/V_{OL} and V_{IH}/V_{IL} characteristics.

Table 25.3 DC Characteristics (3) [I²C-Related Pins*]

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.8 V ±5%, $V_{SS}Q = V_{SS}$ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply	$V_{CC}Q$	3.0	3.3	3.6	V	
Input high voltage	V_{IH}	$V_{CC}Q \times 0.7$	—	$V_{CC}Q + 0.3$	V	
Input low voltage	V_{IL}	–0.3	—	$V_{CC}Q \times 0.3$	V	
Schmitt trigger input characteristics	$V_{IH} - V_{IL}$	$V_{CC}Q \times 0.05$	—	—	V	
Output low voltage	V_{OL}	0	—	0.4	V	I_{OL} = 3.0 mA

Note: * The SCL and SDA pins (open-drain pins)
When the port functions are selected as the general inputs or outputs, however, these pins have the usual V_{OH}/V_{OL} and V_{IH}/V_{IL} characteristics.

Table 25.3 DC Characteristics (4) [USB-Related Pins*]

Conditions: Ta = –40°C to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply	$V_{CC}Q$	3.0	3.3	3.6	V	
Input high voltage	V_{IH}	2.3	—	$V_{CC}Q + 0.3$	V	
Input low voltage	V_{IL}	–0.3	—	$V_{CC}Q \times 0.2$	V	
Input high voltage (UCLK)	V_{IH}(UCLK)	$V_{CC}Q$ – 0.3	—	$V_{CC}Q + 0.3$	V	
Input low voltage (UCLK)	V_{IL}(UCLK)	–0.3	—	$V_{CC}Q \times 0.2$	V	
Output high voltage	V_{OH}	2.4	—	—	V	$V_{CC}Q$ = 3.0 V, I_{OH} = –200 µA
		2.0	—	—		$V_{CC}Q$ = 3.0 V, I_{OH} = –2.0 mA
Output low voltage	V_{OL}	—	—	0.4	V	$V_{CC}Q$ = 3.6 V, I_{OL} = 2.0 mA

Note: * The XVDATA, DPLS, DMNS, TXDPLS, TXDMNS, TXENL, VBUS, SUSPND, and UCLK pins

Table 25.3 DC Characteristics (5) [USB Transceiver-Related Pins*]

Conditions: Ta = –40°C to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Differential input sensitivity	V_{DI}	0.2	—	—	V	|(DP) – (DM)|
Differential common mode range	V_{CM}	0.8	—	2.5	V	
Single ended receiver threshold voltage	V_{SE}	0.8	—	2.0	V	
Output high voltage	V_{OH}	2.8	—	$V_{CC}Q$	V	
Output low voltage	V_{OL}	—	—	0.3	V	
Tri-state leak current	I_{LO}	–10	—	10	µA	0 V < V_{IN} < 3.3 V

Note: * The DP and DM pins

Table 25.4 Permissible Output Currents

Conditions: V_{CC} = 1.8 V ± 5%, $V_{CC}Q$ = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = $PLLV_{SS}$ = AV_{SS} = 0 V, Ta = –40°C to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	SCL, SDA	I_{OL}	—	—	10	mA
	PE0 to PE4, PE6				15	
	Other than above				2	
Permissible output low current (total)		ΣI_{OL}	—	—	120	mA
Permissible output high current (per pin)		$-I_{OH}$	—	—	2	mA
Permissible output high current (total)		$\Sigma -I_{OH}$	—	—	40	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in table 25.4.

25.3 AC Characteristics

Signals input to this LSI are basically handled as signals in synchronization with a clock. The setup and hold times for input pins must be followed.

Table 25.5 Maximum Operating Frequency

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.8 V ± 5%, AV_{CC} = 3.0 V to 3.6 V, Ta = –40°C to +85°C

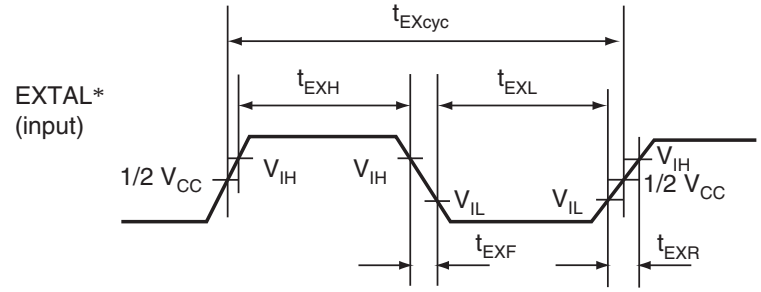
Item		Symbol	Min.	Typ.	Max.	Unit	Remarks
Operating frequency	CPU, cache (Iϕ)	f	20	—	100	MHz	
	External buss (Bϕ)		20	—	50		
	Peripheral module (Pϕ)		5	—	33		

25.3.1 Clock Timing

Table 25.6 Clock Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.8 V ± 5%, AV_{CC} = 3.0 V to 3.6 V, $V_{SS}Q = V_{SS} = AV_{SS}$ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure(s)
EXTAL clock input frequency	f_{EX}	10	25	MHz	25.2
EXTAL clock input cycle time	t_{EXcyc}	40	100	ns	
EXTAL clock input pulse low width	t_{EXL}	7	—	ns	
EXTAL clock input pulse high width	t_{EXH}	7	—	ns	
EXTAL clock input rising time	t_{EXR}	—	4	ns	
EXTAL clock falling time	t_{EXF}	—	4	ns	
CKIO clock input frequency	f_{CK}	20	50	MHz	25.3
CKIO clock input cycle time	t_{CKcyc}	20	50	ns	
CKIO clock input low pulse width	t_{CKIL}	7	—	ns	
CKIO clock input high pulse width	t_{CKIH}	7	—	ns	
CKIO clock input rising time	t_{CKIr}	—	3	ns	
CKIO clock input falling time	t_{CKIf}	—	3	ns	
CKIO, CKIO2 clock output frequency	f_{OP}	20	50	MHz	25.4
CKIO, CKIO2 clock output cycle time	t_{cyc}	20	50	ns	
CKIO, CKIO2 clock output pulse low width	t_{CKOL}	7	—	ns	
CKIO, CKIO2 clock output pulse high width	t_{CKOH}	7	—	ns	
CKIO, CKIO2 clock output rising time	t_{CKOR}	—	5	ns	
CKIO, CKIO2 clock falling time	t_{CKOF}	—	5	ns	
Oscillation settling time (after power-on reset)	t_{OSC1}	10	—	ms	25.5
Phase difference between CKIO and CKIO2	$t_{phckio2}$	—	3	ns	25.6
Oscillation settling time 1 (after standby mode)	t_{OSC2}	10	—	ms	25.7
Oscillation settling time 2 (after standby mode)	t_{OSC3}	10	—	ms	25.8



Note: * When the clock is input on the EXTAL pin.

Figure 25.2 EXTAL Clock Input Timing

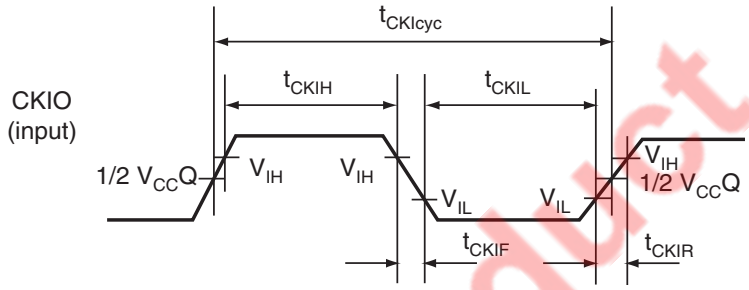


Figure 25.3 CKIO Clock Input Timing

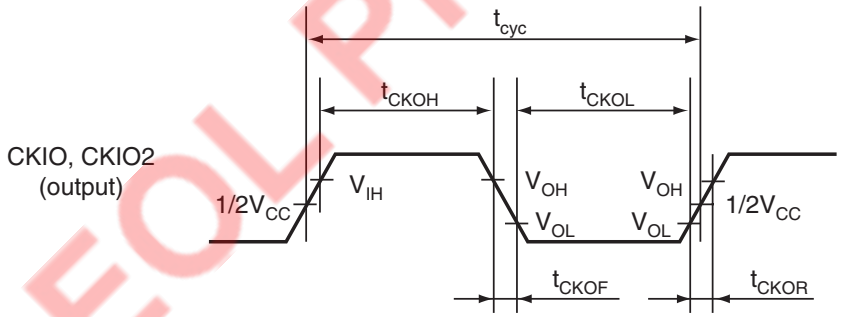
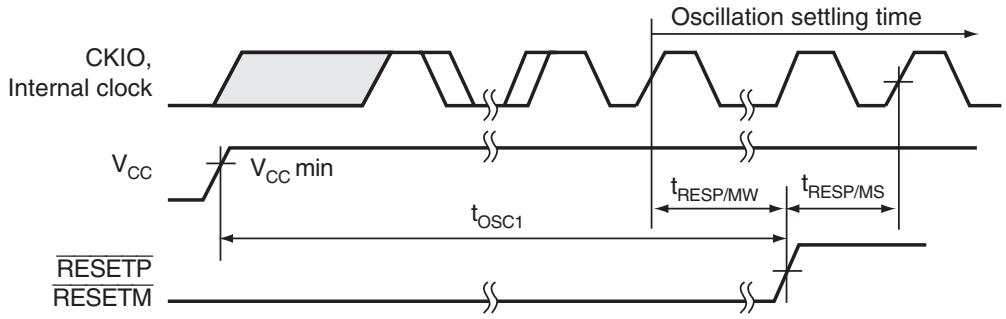


Figure 25.4 CKIO and CKIO2 Clock Input Timing



Note: Oscillation settling time when the internal oscillator is used.

Figure 25.5 Oscillation Settling Timing (Power-On)

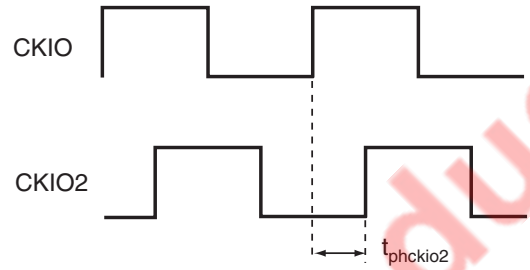
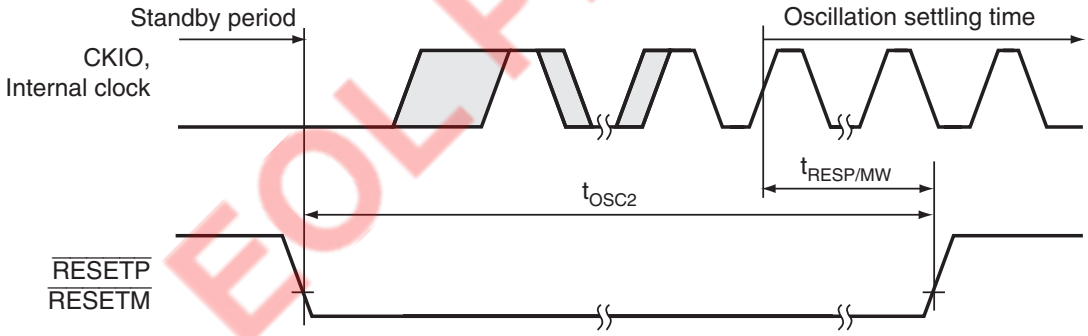
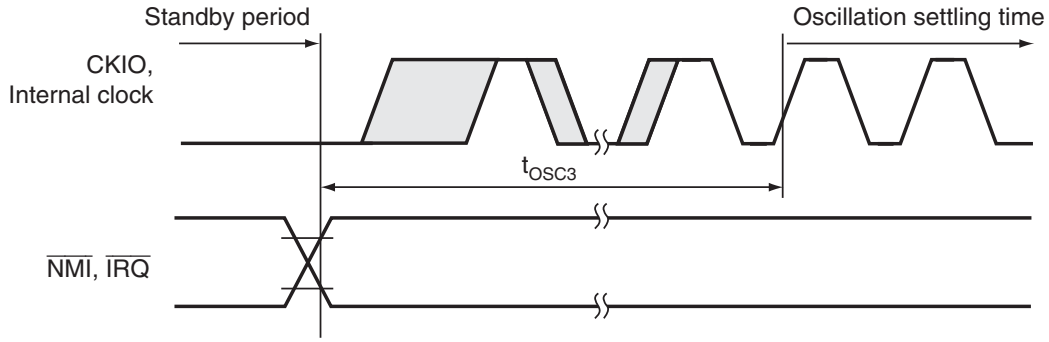


Figure 25.6 Phase Difference between CKIO and CKIO2



Note: Oscillation settling time when the internal oscillator is used.

Figure 25.7 Oscillation Settling Timing (Standby Mode Canceled by Reset)



Note: Oscillation settling time when the internal oscillator is used.

Figure 25.8 Oscillation Settling Timing (Standby Mode Canceled by $\overline{NMI}$ or $\overline{IRQ}$)

25.3.2 Control Signal Timing

Table 25.7 Control Signal Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.8 V ±5%, AV_{CC} = 2.7 V to 3.6 V, $V_{SS}Q$ = V_{SS} = AV_{SS} = 0 V, Ta = –40°C to +85°C

		Bϕ = 50 MHz*[2]			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure(s)**
$\overline{\text{RESETP}}$ pulse width	t_{RESPW}	20*[2]	—	Bcyc*[4]	25.5, 25.6, 25.9, and 25.10
$\overline{\text{RESETP}}$ setup time*[1]	t_{RESPS}	22	—	ns	
$\overline{\text{RESETP}}$ hold time	t_{RESPH}	2	—	ns	
$\overline{\text{RESETM}}$ pulse width	t_{RESMW}	12*[3]	—	Bcyc*[4]	
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	22	—	ns	
$\overline{\text{RESETM}}$ hold time	t_{RESMH}	12	—	ns	
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	1/2t_{cyc} + 10	—	ns	25.11
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	1/2t_{cyc} + 10	—	ns	
$\overline{\text{NMI}}$ setup time*[1]	t_{NMIS}	30	—	ns	25.10
$\overline{\text{NMI}}$ hold time	t_{NMIH}	30	—	ns	
$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ setup time*[1]	t_{IRQS}	30	—	ns	
$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ hold time	t_{IRQH}	30	—	ns	
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	1/2t_{cyc} + 13	ns	25.11, 25.12
STATUS1, STATUS0 delay time	t_{STD}	—	100	ns	
Bus tri-state delay time 1	t_{BOFF1}	0	100	ns	
Bus tri-state delay time 2	t_{BOFF2}	0	100	ns	
Bus buffer on time 1	t_{BON1}	0	30	ns	
Buss buffer on time 2	t_{BON2}	0	30	ns	

Notes:
1. The $\overline{\text{RESETP}}$, $\overline{\text{NMI}}$ and $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ signals are asynchronous signals. When the setup time is satisfied, change of signal level is detected at the rising edge of the clock. If not, the detection is delayed until the rising edge of the clock.
2. In standby mode, t_{RESP} = t_{OSC2} (10 ms). When multiplier of the clock is changed, t_{REPW} = t_{PLL1} (100 µs)
3. In standby mode, t_{RESP} = t_{OSC2} (10 ms). When multiplier of the clock is changed, $\overline{\text{RESETM}}$ must be held low until signals STATUS0 and STATUS1 indicate the reset state (HH).
4. Bcyc indicates external clock cycle time. (B clock cycle)

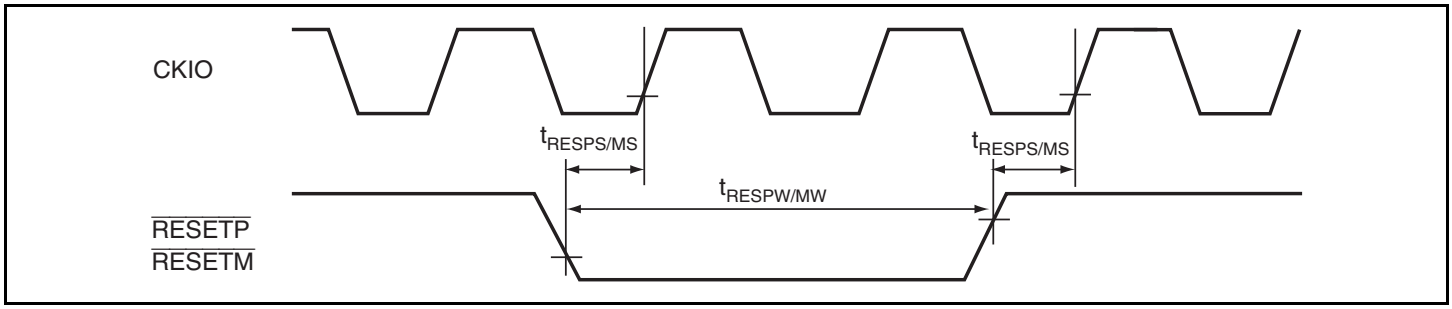


Figure 25.9 Reset Input Timing

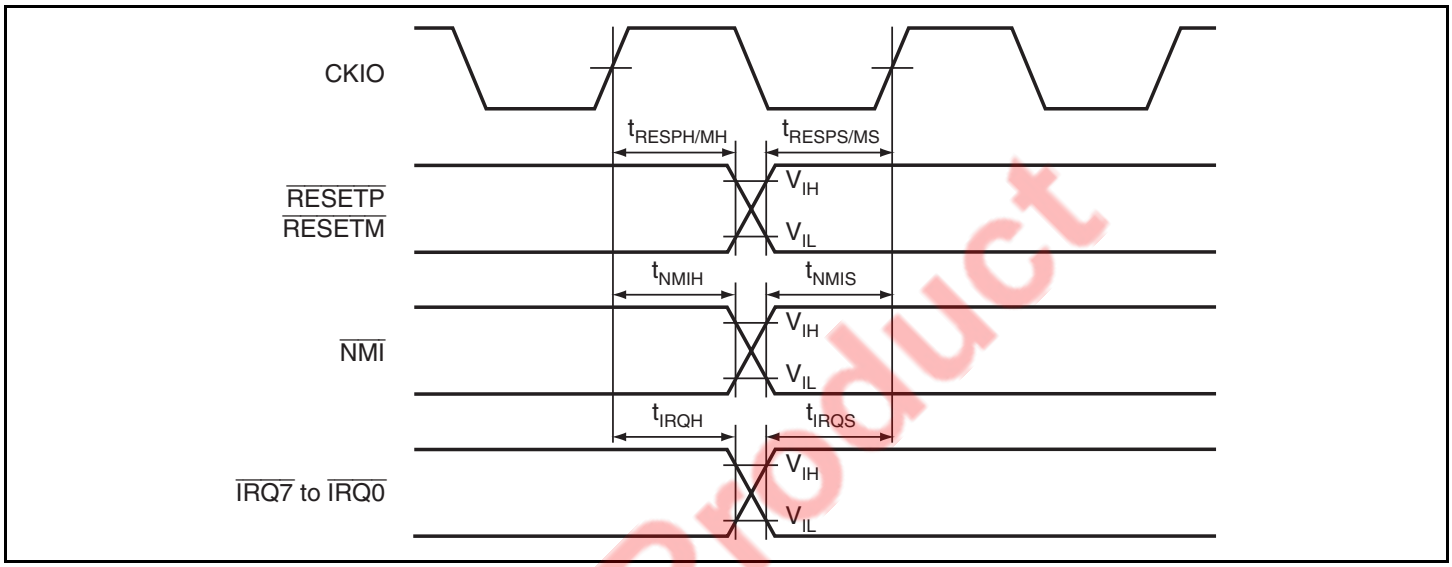


Figure 25.10 Interrupt Input Timing

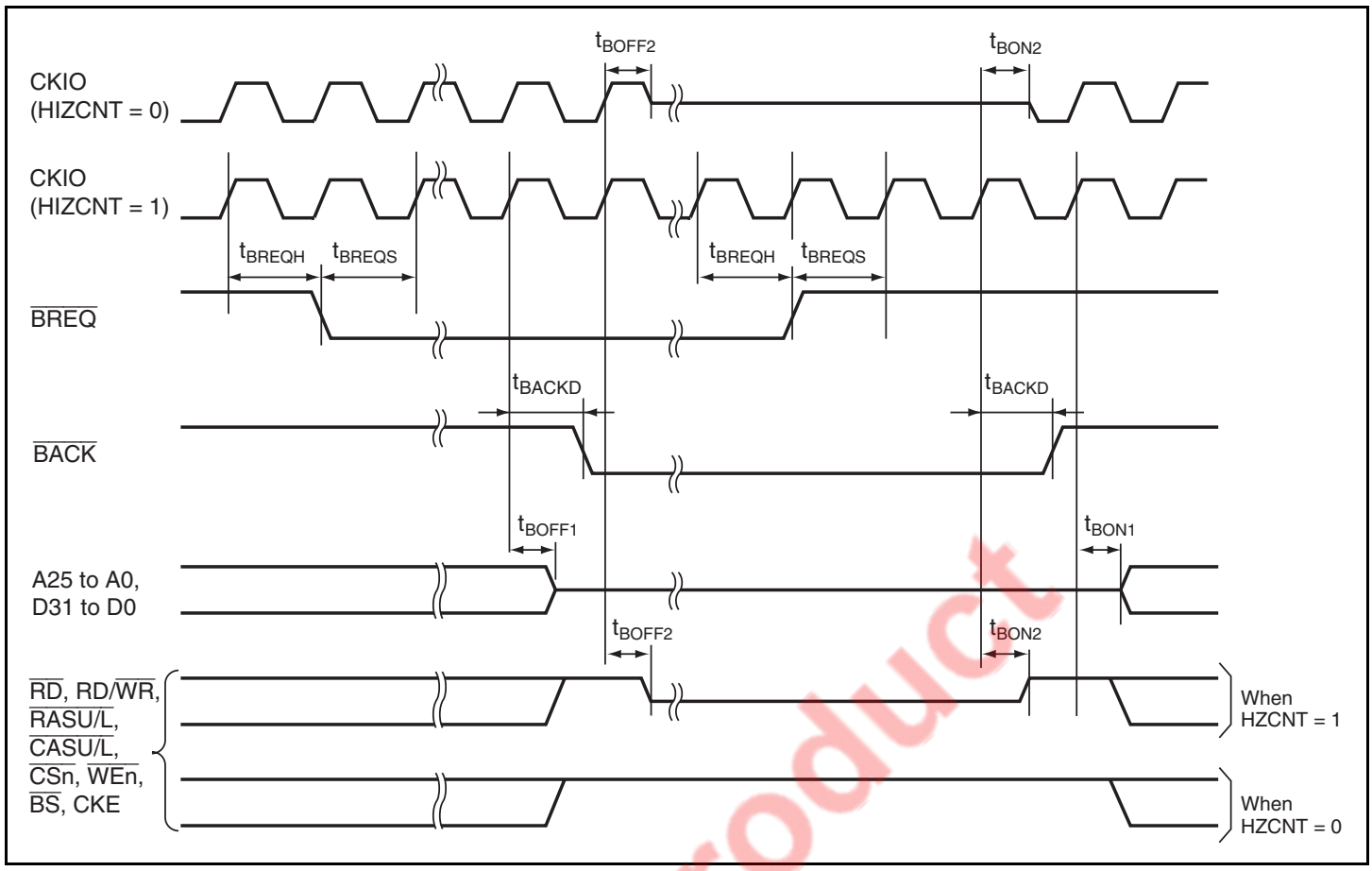


Figure 25.11 Bus Release Timing

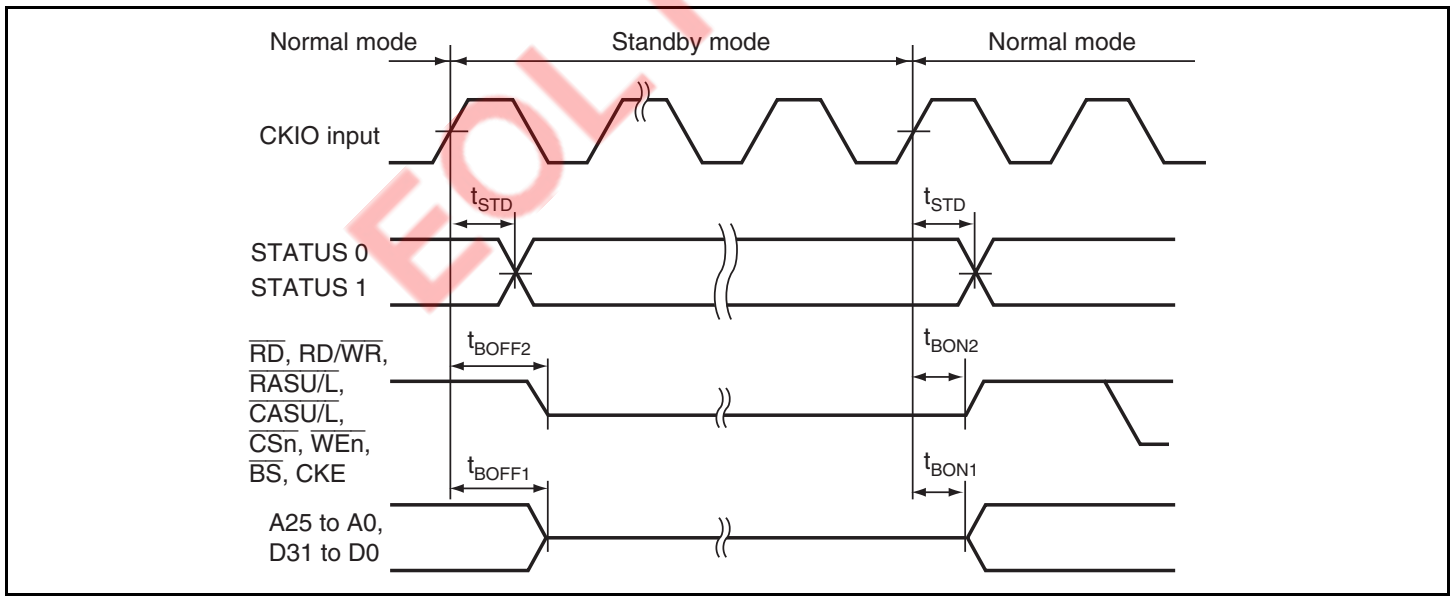


Figure 25.12 Pin Driving Timing in Standby Mode

25.3.3 AC Bus Timing

Table 25.8 Bus Timing

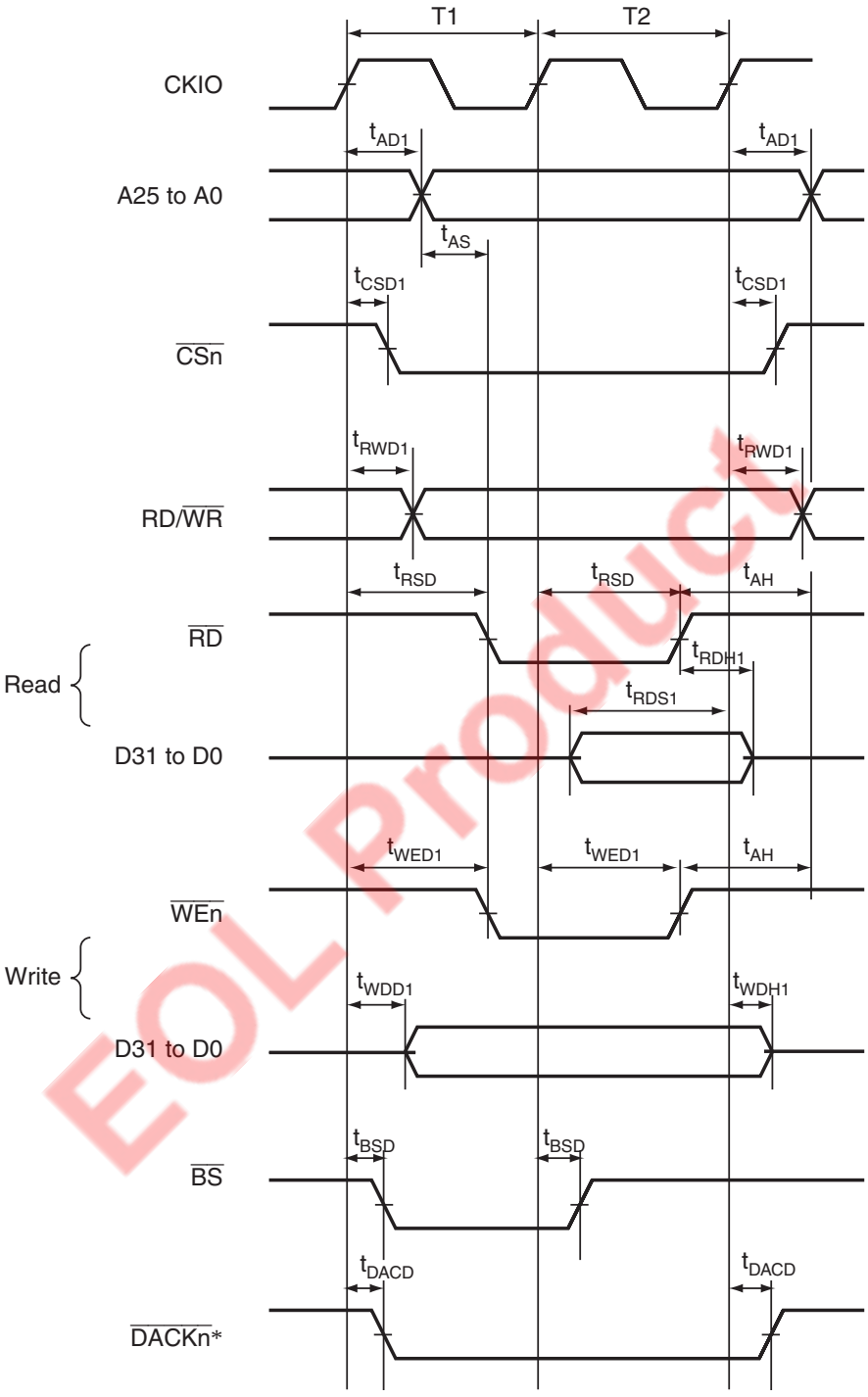
Conditions: Clock mode 2/6/7, $V_{CC}Q$ = 3.0 V to 3.6 V, $V_{SS}Q$ = 0 V, Ta = –40°C to +85°C

		Bϕ = 50 MHz*			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure(s)**
Address delay time 1	t_{AD1}	1	12	ns	25.13 to 25.39
Address delay time 2	t_{AD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 12	ns	25.22
Address delay time 3	t_{AD3}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 12	ns	25.40, 25.41
Address setup time	t_{AS}	0	—	ns	25.13 to 25.18
Address hold time	t_{AH}	0	—	ns	25.13 to 25.17
$\overline{BS}$ delay time	t_{BSD}	—	12	ns	25.13 to 25.36
$\overline{CS}$ delay time 1	t_{CSD1}	1	12	ns	25.13 to 25.39
$\overline{CS}$ delay time 2	t_{CSD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 12	ns	25.40, 25.41
Read write delay time 1	t_{RWD1}	1	12	ns	25.13 to 25.39
Read write delay time 2	t_{RWD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 12	ns	25.40, 25.41
Read strobe delay time	t_{RSD}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 12	ns	25.13 to 25.18, 25.20 to 25.22
Read data setup time 1	t_{RDS1}	$1/2t_{cyc}$ + 8	—	ns	25.13 to 25.18, 25.20, 25.21
Read data setup time 2	t_{RDS2}	8	—	ns	25.23 to 25.26, 25.31 to 25.33
Read data setup time 3	t_{RDS3}	$1/2t_{cyc}$ + 8	—	ns	25.22
Read data setup time 4	t_{RDS4}	$1/2t_{cyc}$ + 8	—	ns	25.40
Read data hold time 1	t_{RDH1}	0	—	ns	25.13 to 25.18, 25.20
Read data hold time 2	t_{RDH2}	2	—	ns	25.23 to 25.26, 25.31 to 25.33
Read data hold time 3	t_{RDH3}	0	—	ns	25.22
Read data hold time 4	t_{RDH4}	$1/2t_{cyc}$ + 5	—	ns	25.40
Write enable delay time 1	t_{WED1}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 12	ns	25.13 to 25.18, 25.20
Write enable delay time 2	t_{WED2}	—	12	ns	25.21

Item	Symbol	Bϕ = 50 MHz* Min.	Max.	Unit	Figure(s)
Write data delay time 1	t_{WDD1}	—	14	ns	25.13 to 25.21
Write data delay time 2	t_{WDD2}	—	14	ns	25.27 to 25.30, 25.34 to 25.36
Write data delay time 3	t_{WDD3}	—	$1/2t_{cyc}$ + 14	ns	25.40
Write enable hold time 1	t_{WDH1}	1	—	ns	25.13 to 25.21
Write enable hold time 2	t_{WDH2}	1	—	ns	25.27 to 25.30, 25.34 to 25.36
Write enable hold time 3	t_{WDH3}	$1/2t_{cyc}$	—	ns	25.40
$\overline{WAIT}$ setup time 1	t_{WTS1}	$1/2t_{cyc}$ + 8	—	ns	25.14, 25.15, 25.17 to 25.22
$\overline{WAIT}$ setup time 2	t_{WTS2}	8	—	ns	25.16
$\overline{WAIT}$ hold time 1	t_{WTH1}	$1/2t_{cyc}$ + 4	—	ns	25.14, 25.15, 25.17 to 25.22
$\overline{WAIT}$ hold time 2	t_{WTH2}	4	—	ns	25.16
$\overline{RAS}$ delay time 1	t_{RASD1}	1	12	ns	25.23 to 25.34, 25.36 to 25.39
$\overline{RAS}$ delay time 2	t_{RASD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 12	ns	25.40, 25.41
$\overline{CAS}$ delay time 1	t_{CASD1}	1	12	ns	25.23 to 25.39
$\overline{CAS}$ delay time 2	t_{CASD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 12	ns	25.40, 25.41
DQM delay time 1	t_{DQMD1}	1	12	ns	25.23 to 25.36
DQM delay time 2	t_{DQMD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 12	ns	25.40, 25.41
CKE delay time 1	t_{CKED1}	1	12	ns	25.38
CKE delay time 2	t_{CKED2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 12	ns	25.41
$\overline{AH}$ delay time	t_{AHD}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 12	ns	25.18
Multiplexed address delay time	t_{MAD}	—	12	ns	25.18
Multiplexed address hold time	t_{MAH}	0	—	ns	25.18
$\overline{DACK}$, $\overline{TEND}$ delay time	t_{DACD}	—	Refer to peripheral modules	ns	25.13 to 25.34
$\overline{FRAME}$ delay time	t_{FMD}	1	12	ns	25.19

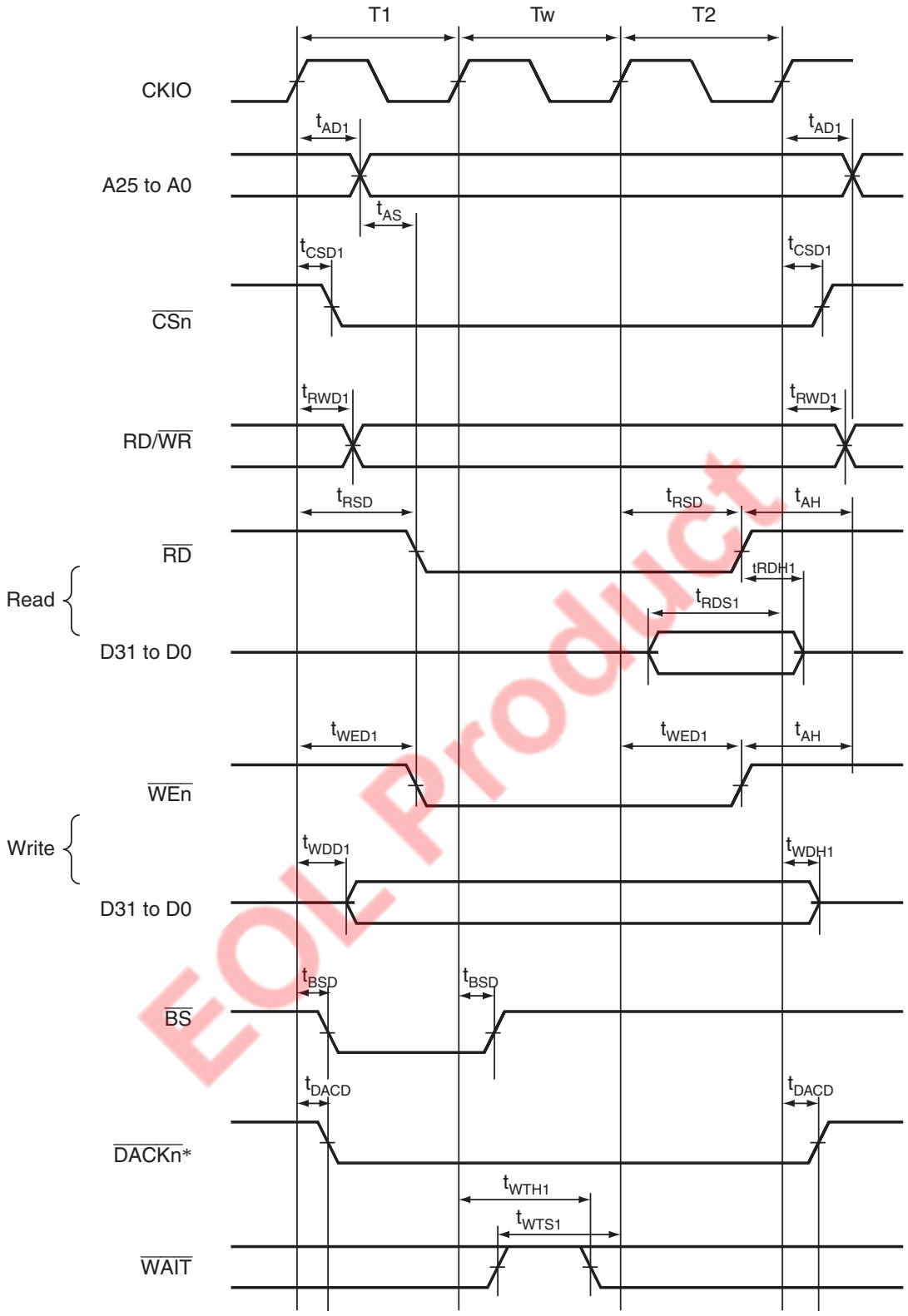
Note: * The maximum value (f_{max}) of Bϕ (external bus clock) depends on the number of wait cycles and the system configuration of your board.

25.3.4 Basic Timing



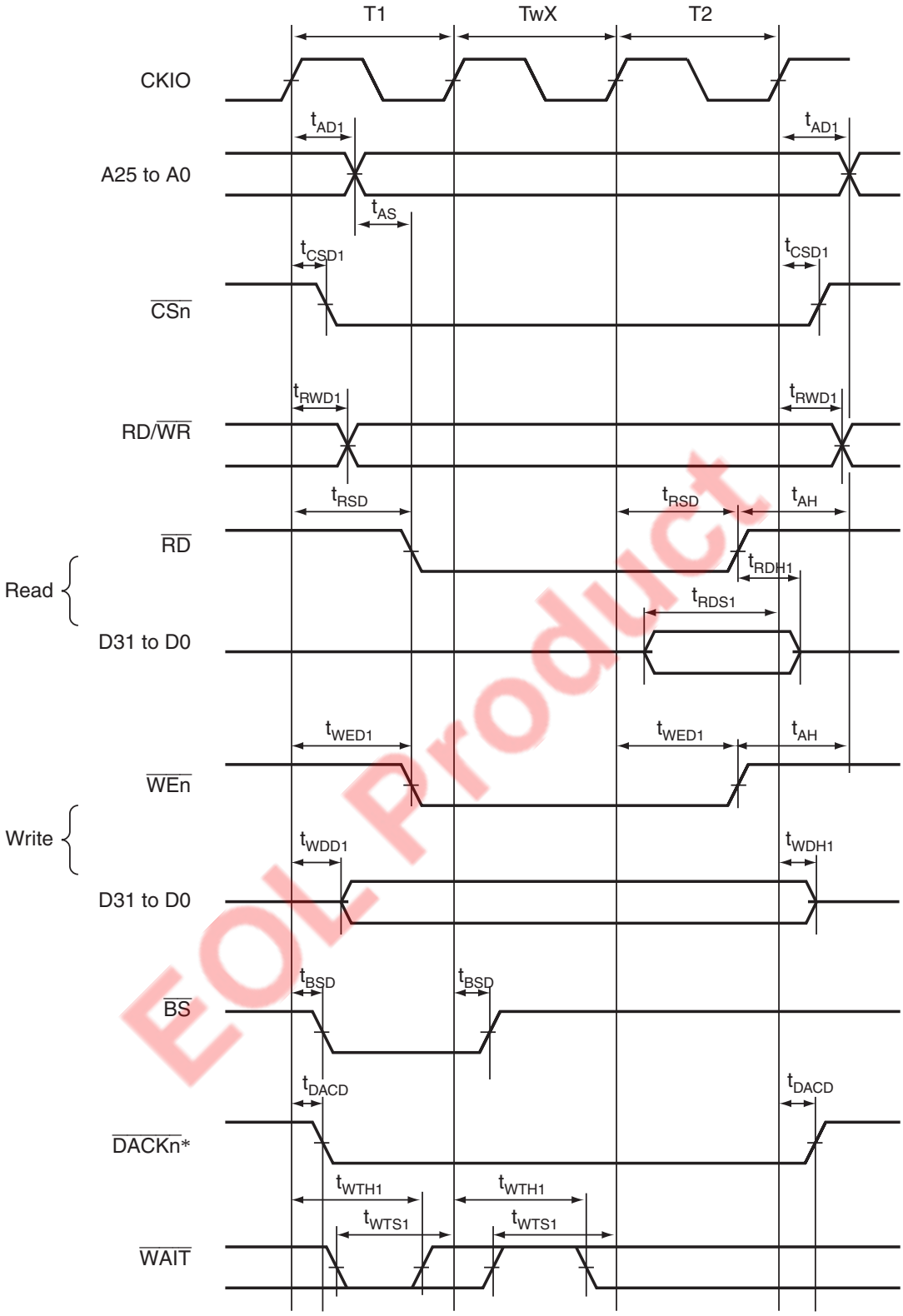
Note: * Waveform for DACKn when active low is selected.

Figure 25.13 Basic Bus Timing for Normal Space (No Wait)



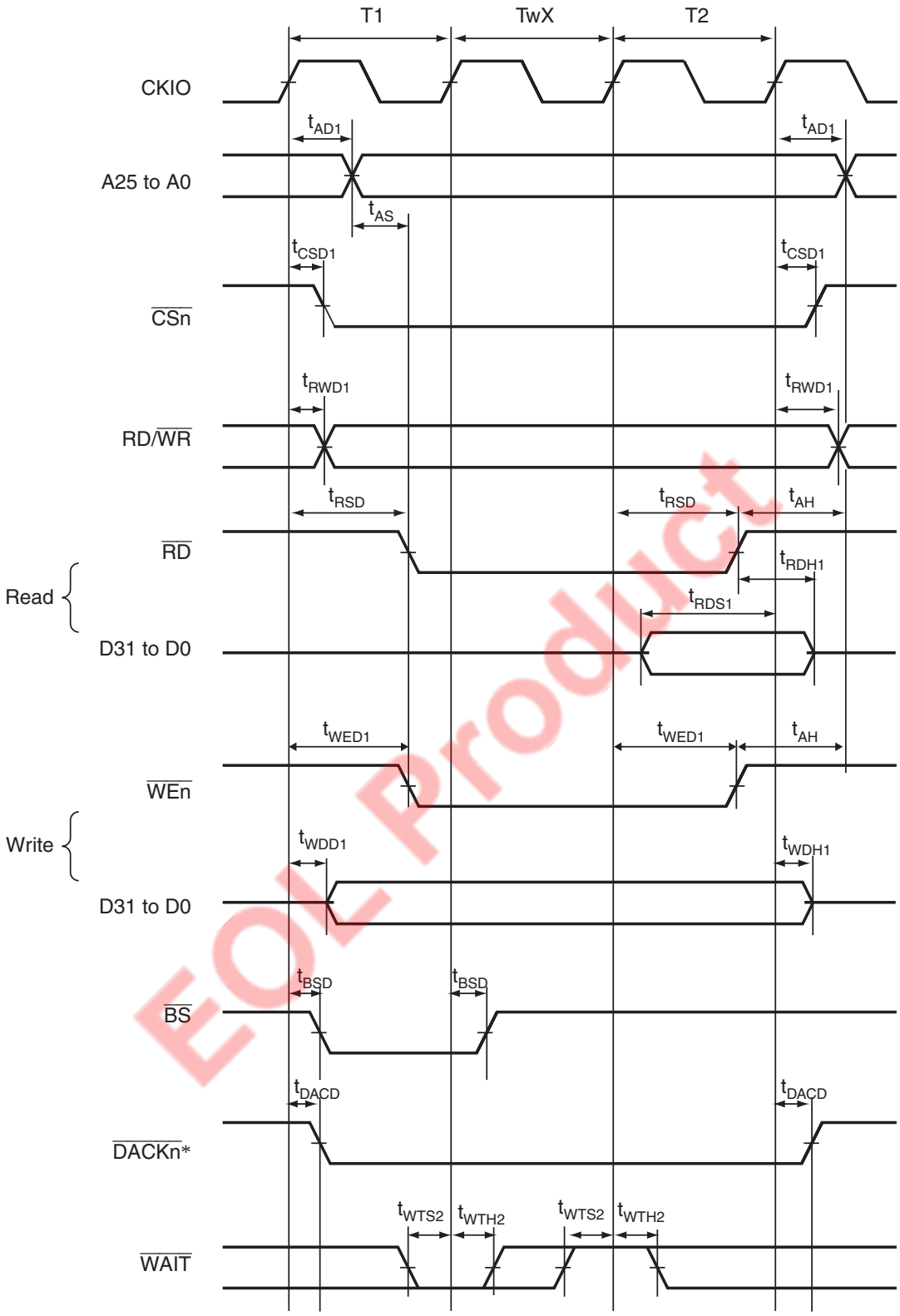
Note: * Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.14 Basic Bus Timing for Normal Space (Software 1 Wait)



Note: * Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.15 Basic Bus Timing for Normal Space (One Cycle of Externally Input/WAITSEL = 0)



Note: * Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.16 Basic Bus Timing for Normal Space (One Cycle of Externally Input/WAITSEL = 1)

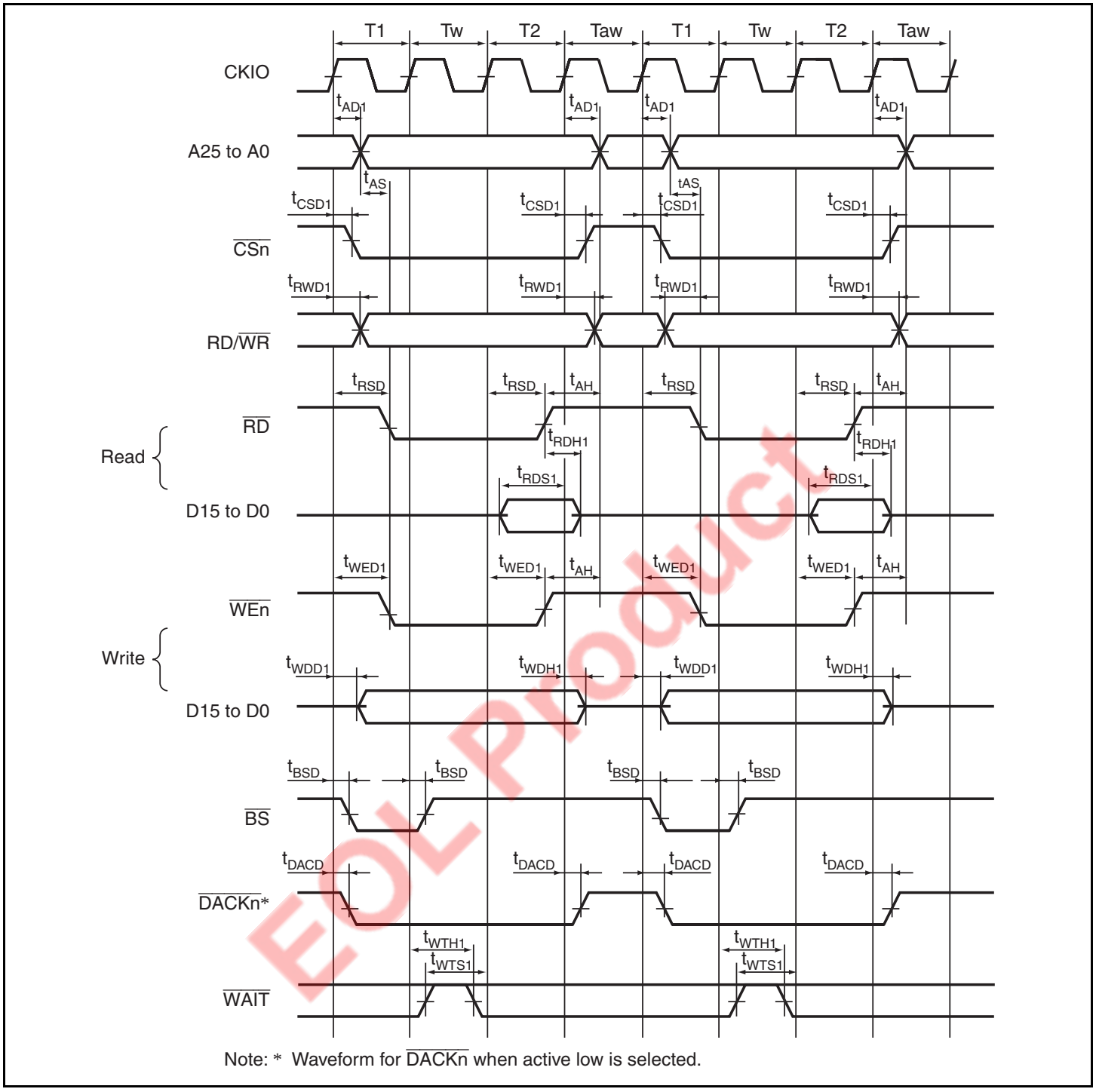


Figure 25.17 Basic Bus Timing for Normal Space
(One Cycle of Software Wait, External Wait Cycle Valid (WM Bit = 0), No Idle Cycle)

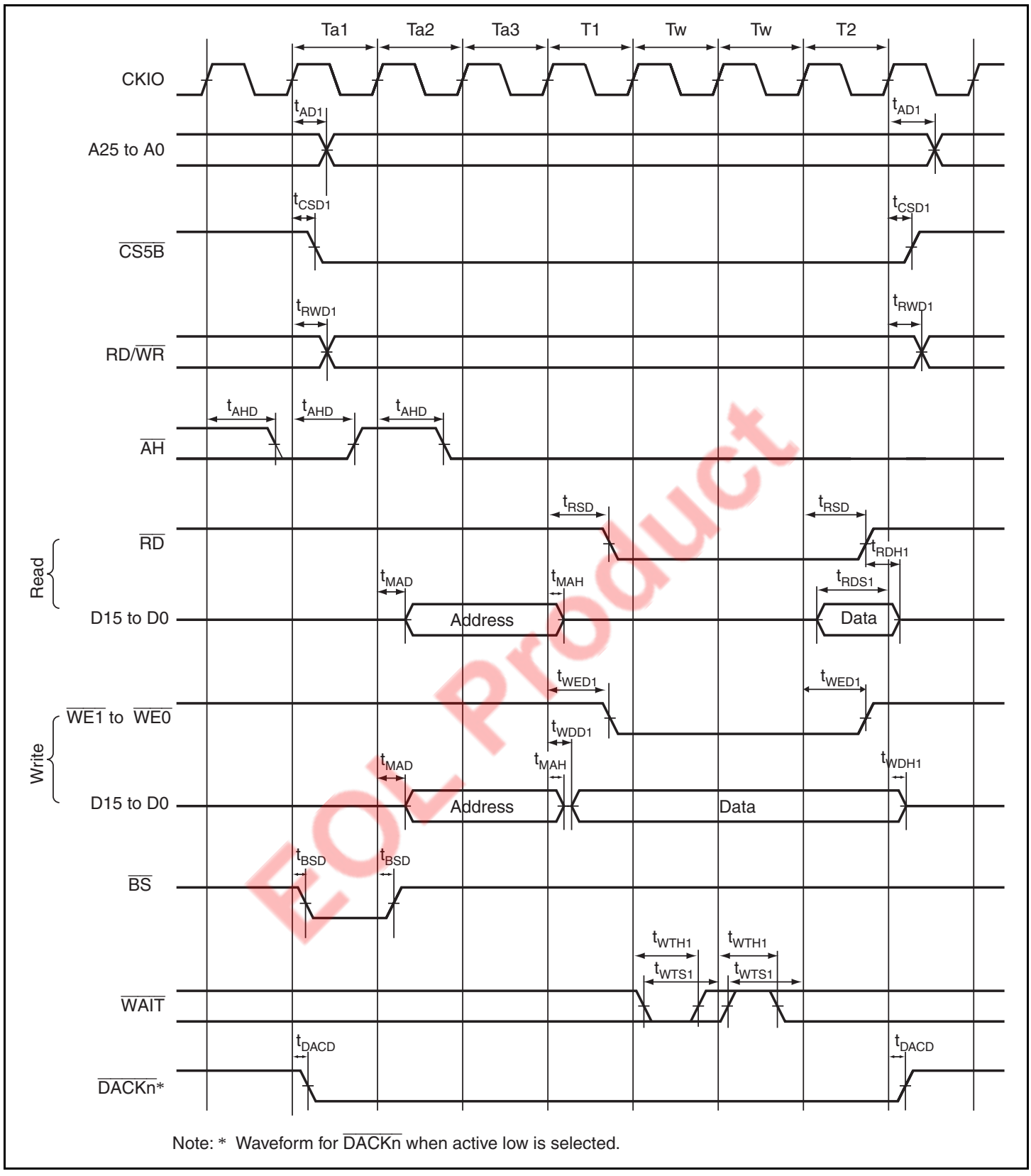
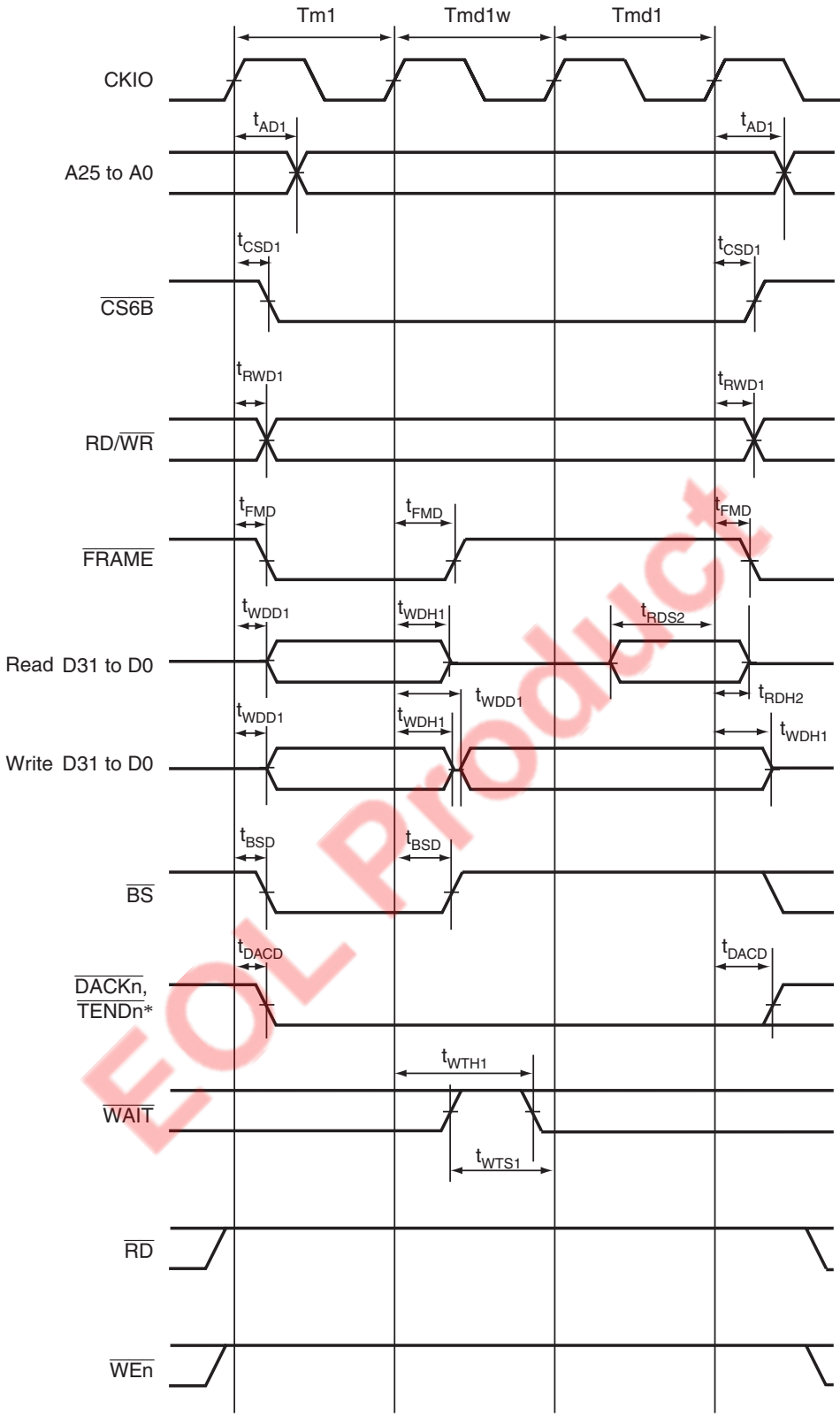


Figure 25.18 MPX-IO Interface Bus Cycle
(Three Address Cycles, One Software Wait Cycle, One External Wait Cycle)



Note: * Waveform for $\overline{\text{DACKn}}$ and $\overline{\text{TENDn}}$ when active low is selected.

Figure 25.19 Burst MPX-IO Interface Bus Cycle Single Read Write (One Address Cycle, One Software Wait)

25.3.5 Bus Cycle of Byte-Selection SRAM

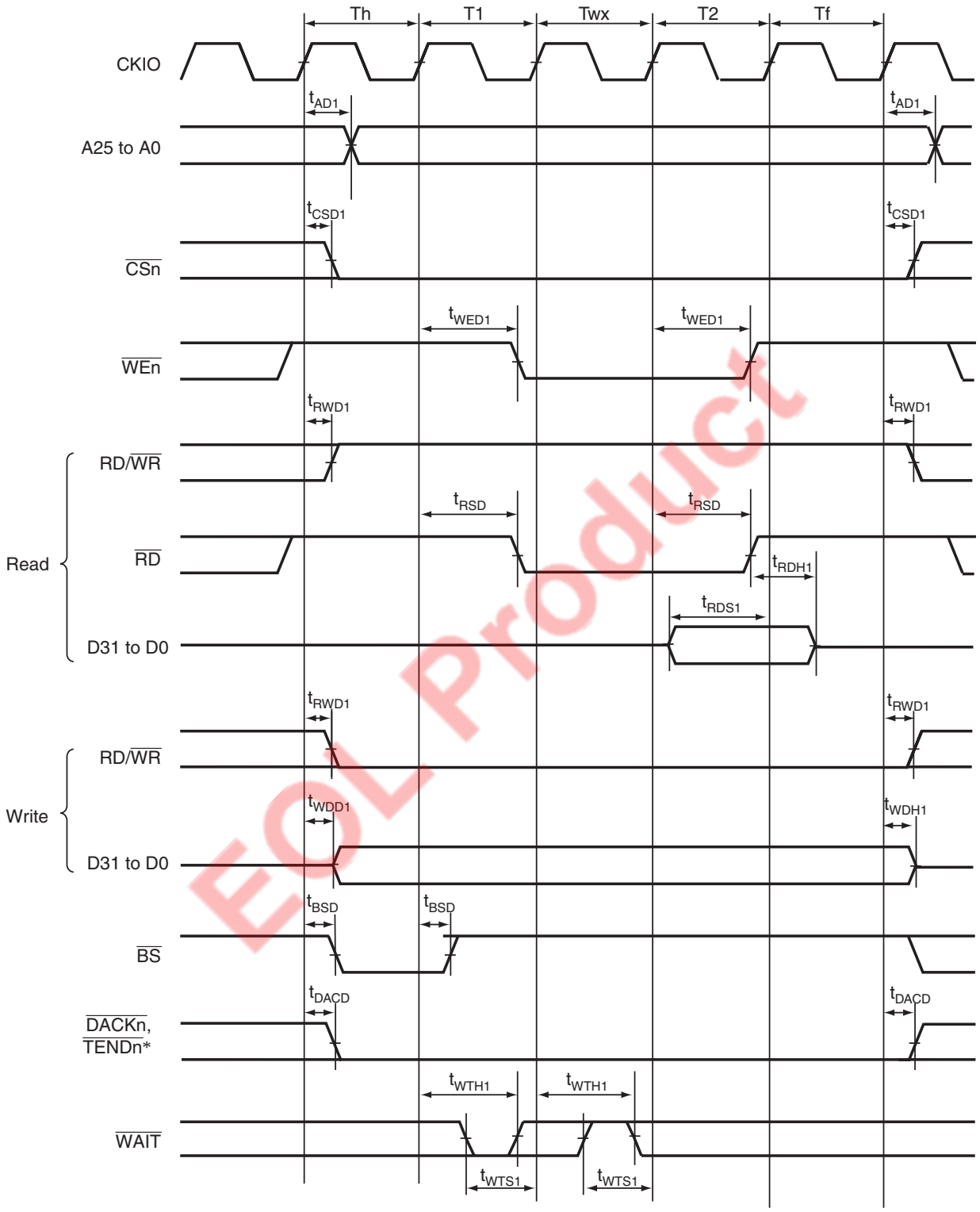
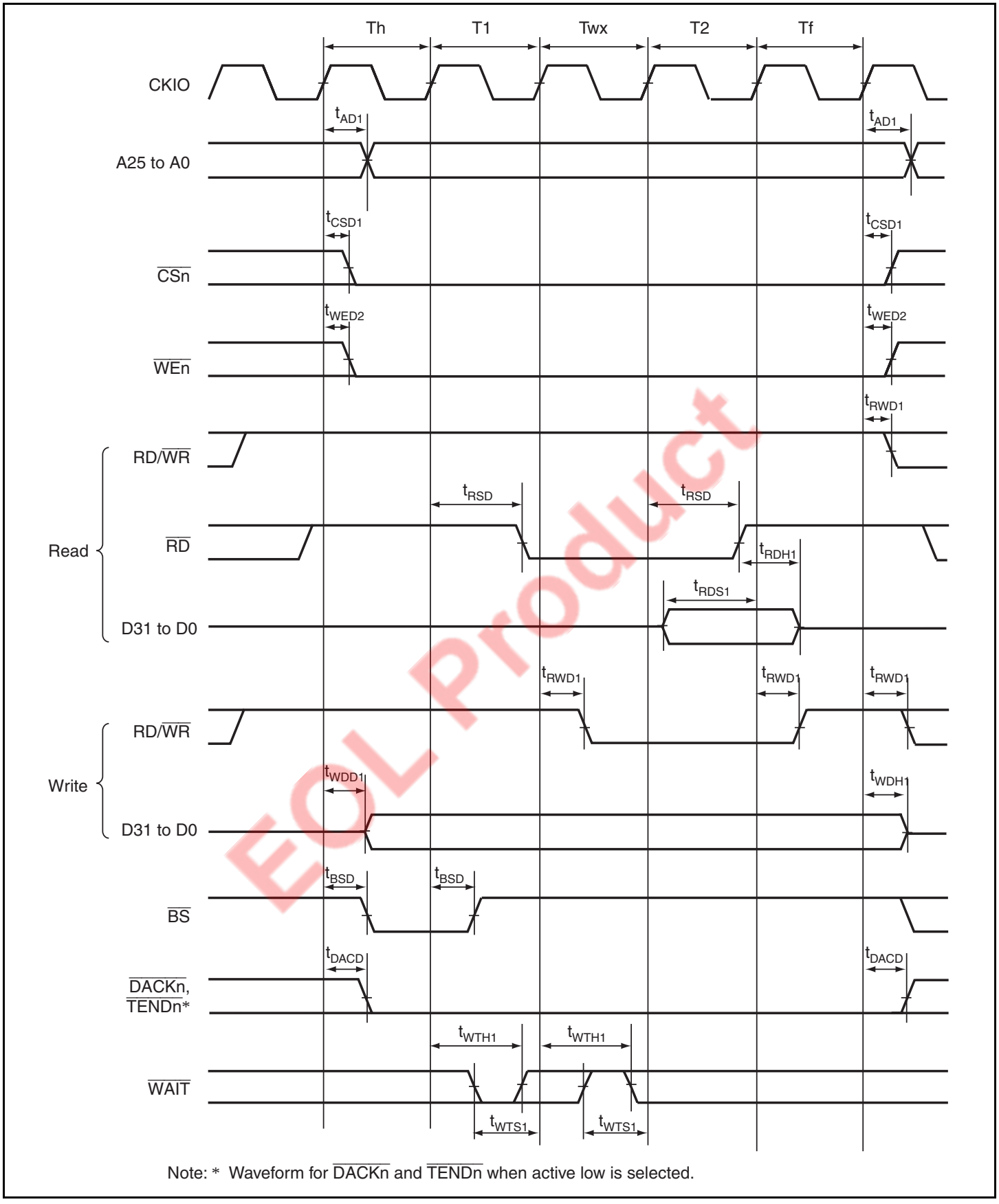


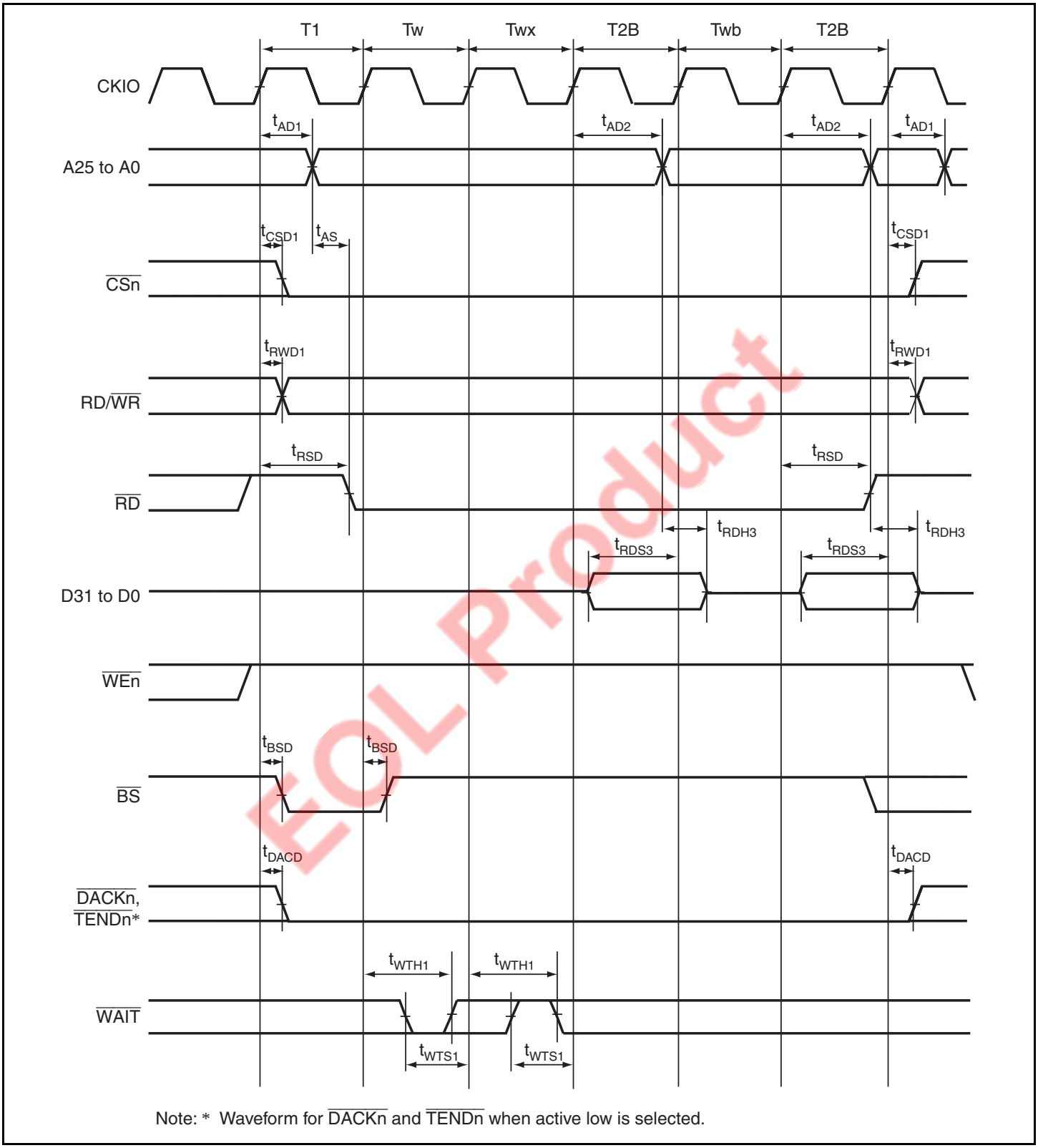
Figure 25.20 Byte-Selection SRAM Bus Cycle (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 0 (Write Cycle UB/LB Control))



Note: * Waveform for DACKn and TENDn when active low is selected.

Figure 25.21 Byte-Selection SRAM Bus Cycle (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 1 (Write Cycle WE Control))

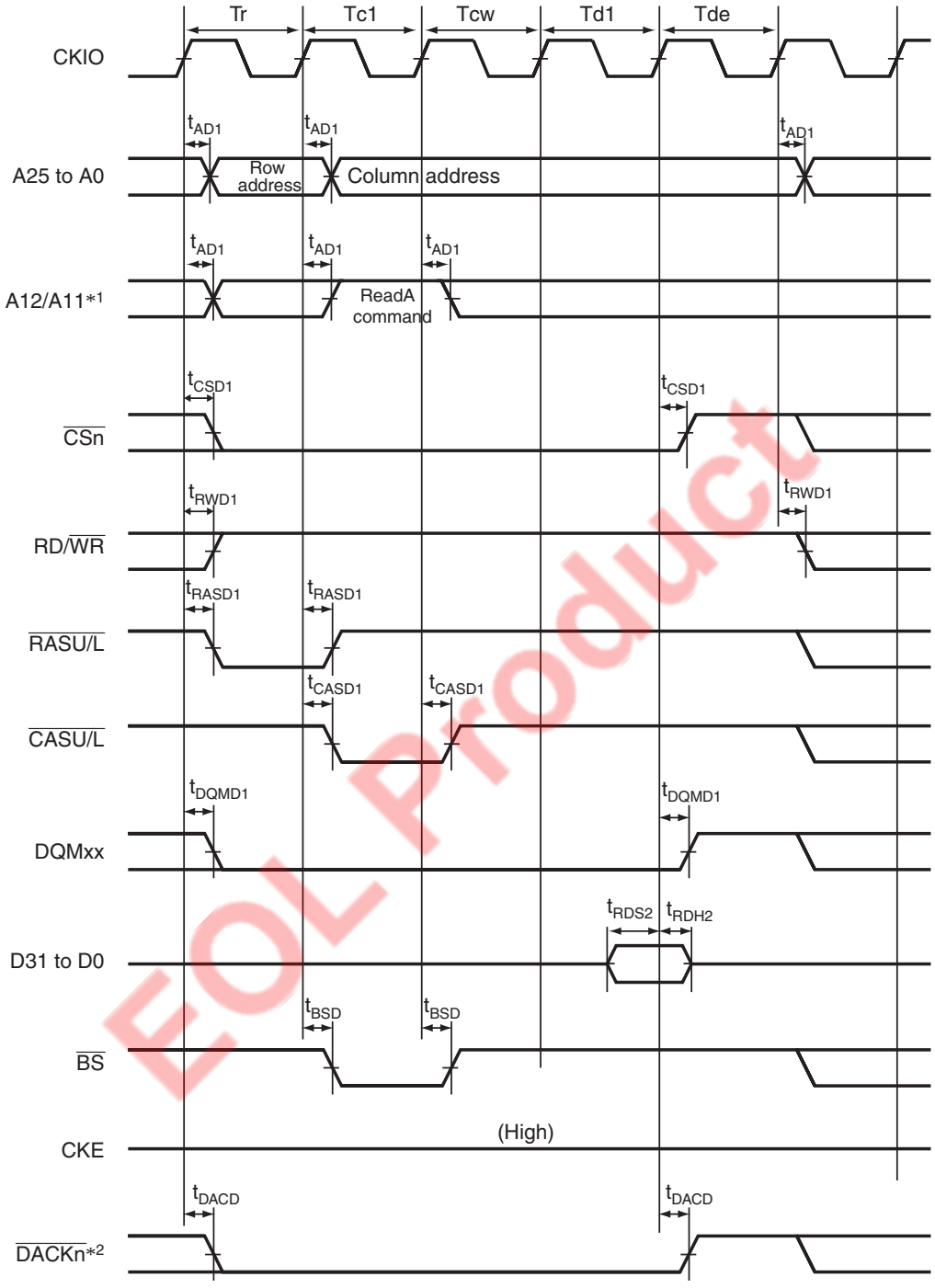
25.3.6 Burst ROM Read Cycle



Note: * Waveform for $\overline{\text{DACKn}}$ and $\overline{\text{TENDn}}$ when active low is selected.

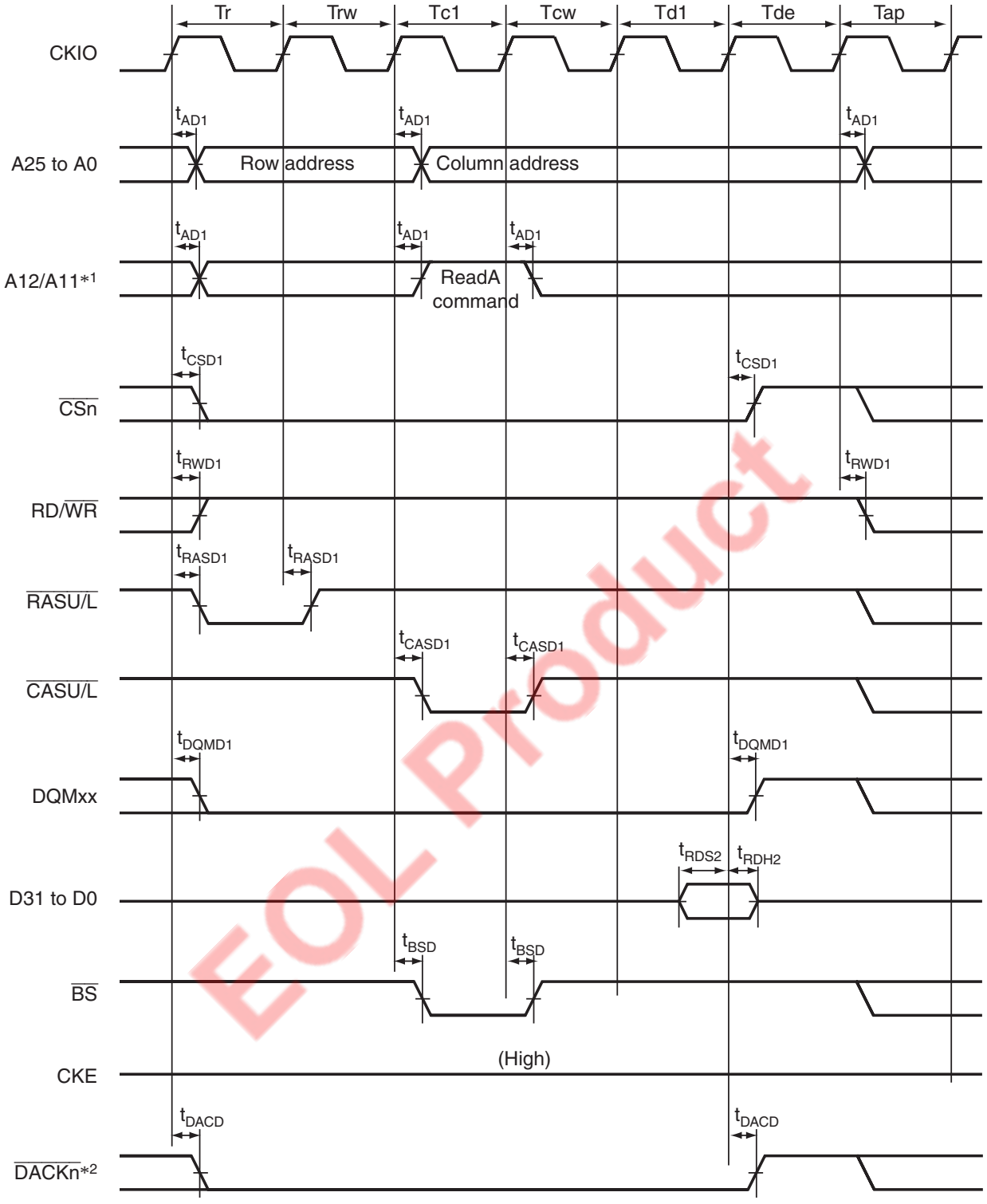
Figure 25.22 Burst ROM Read Cycle
(One Software Wait Cycle, One Asynchronous External Burst Wait Cycle, Two Burst)

25.3.7 Synchronous DRAM Timing



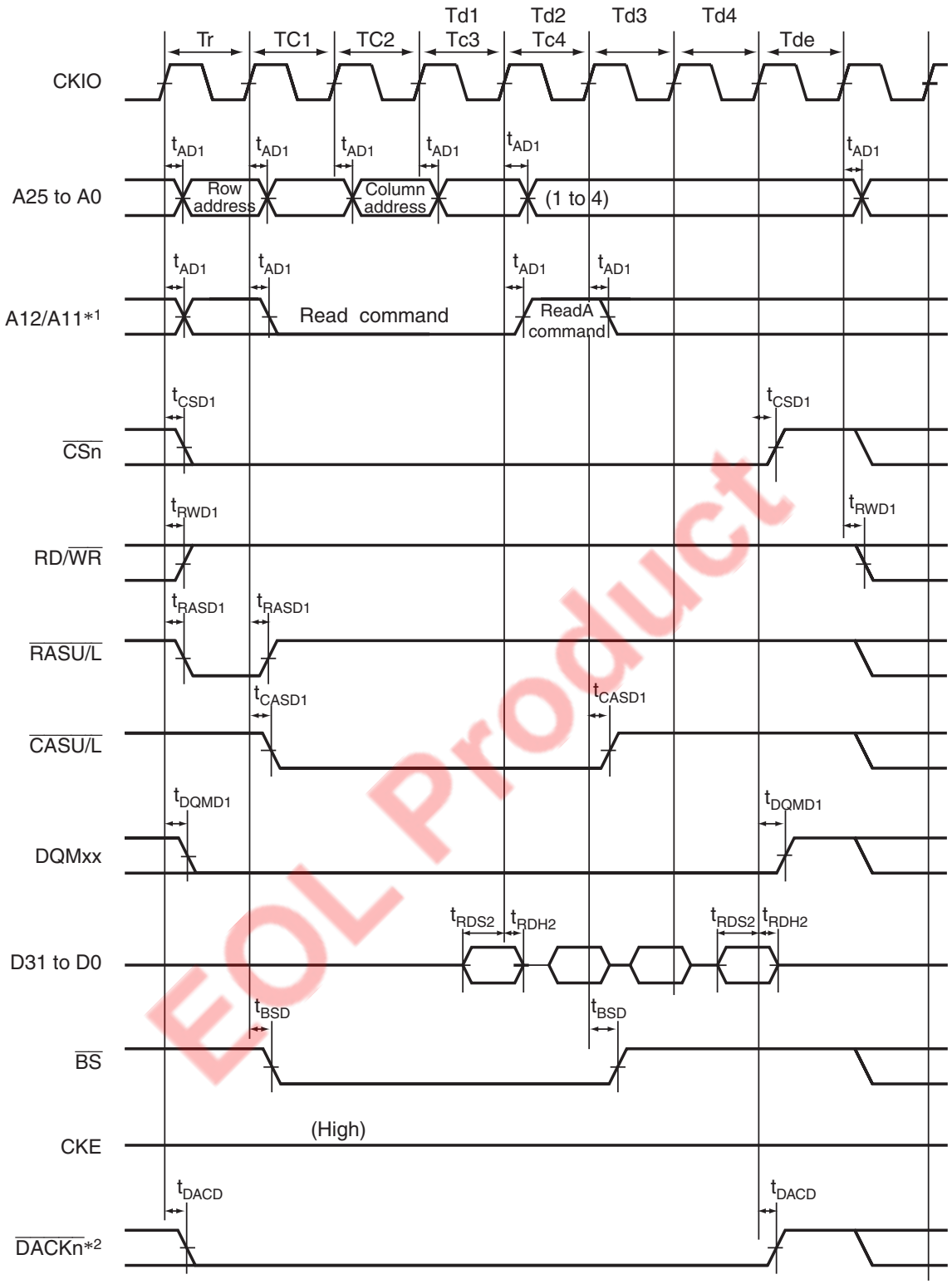
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.23 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 0 Cycle)



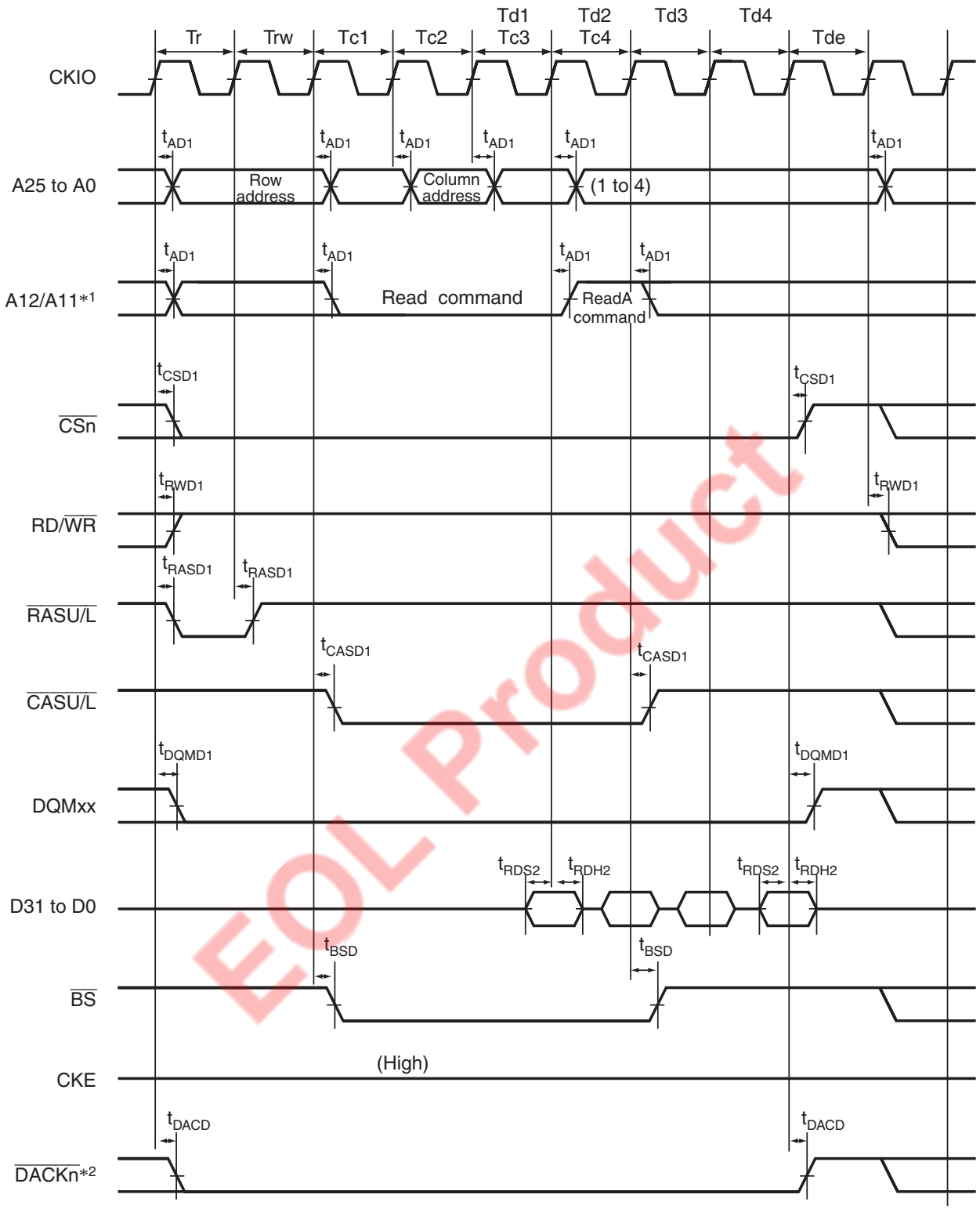
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for DACKn when active low is selected.

Figure 25.24 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 1 Cycle)



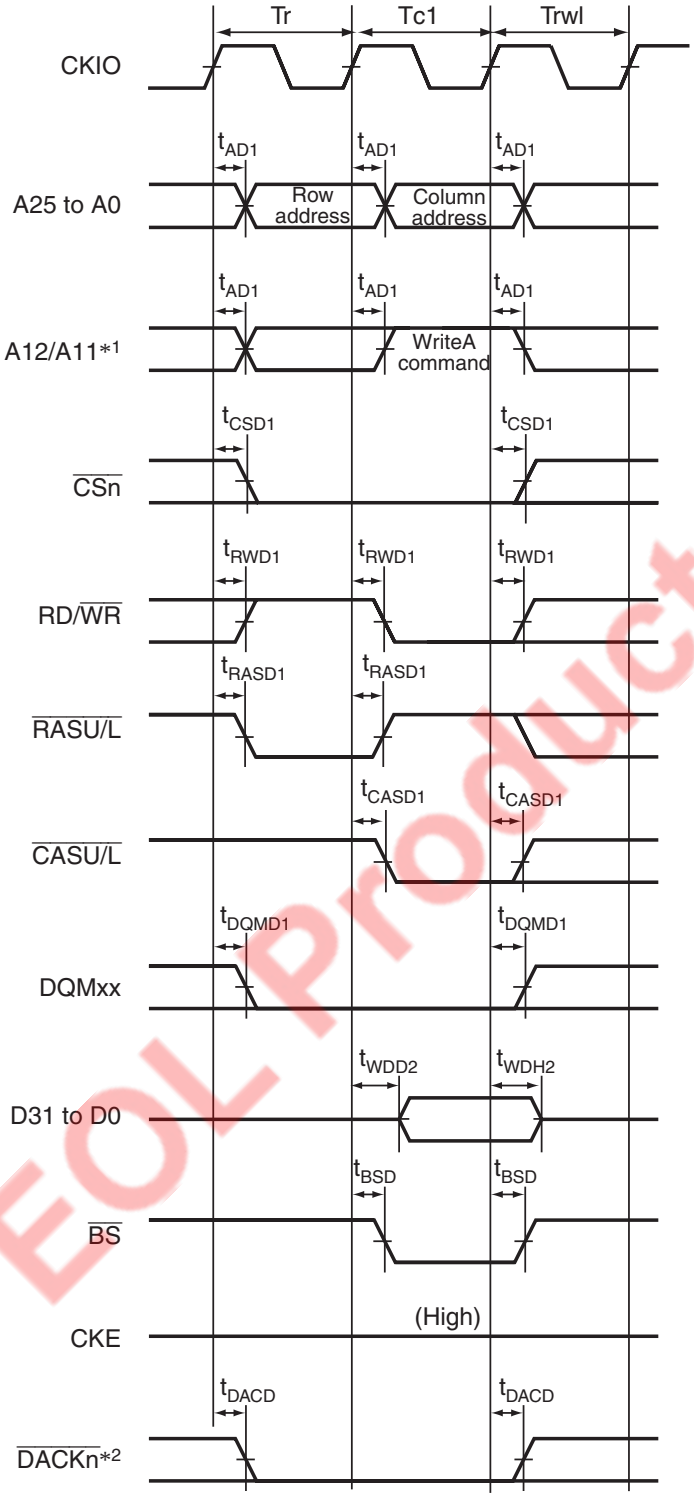
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.25 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 1 Cycle)



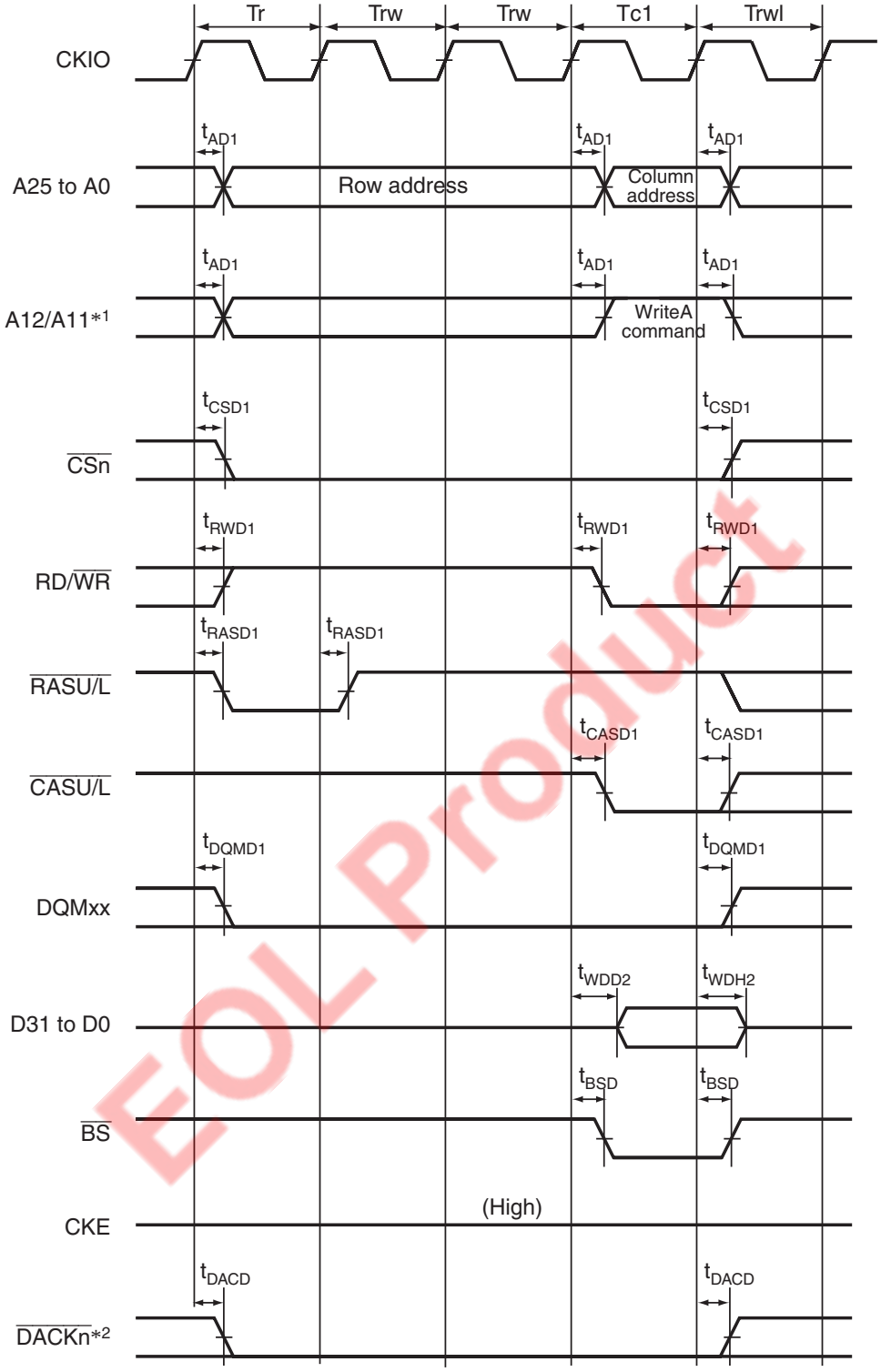
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.26 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 0 Cycle)



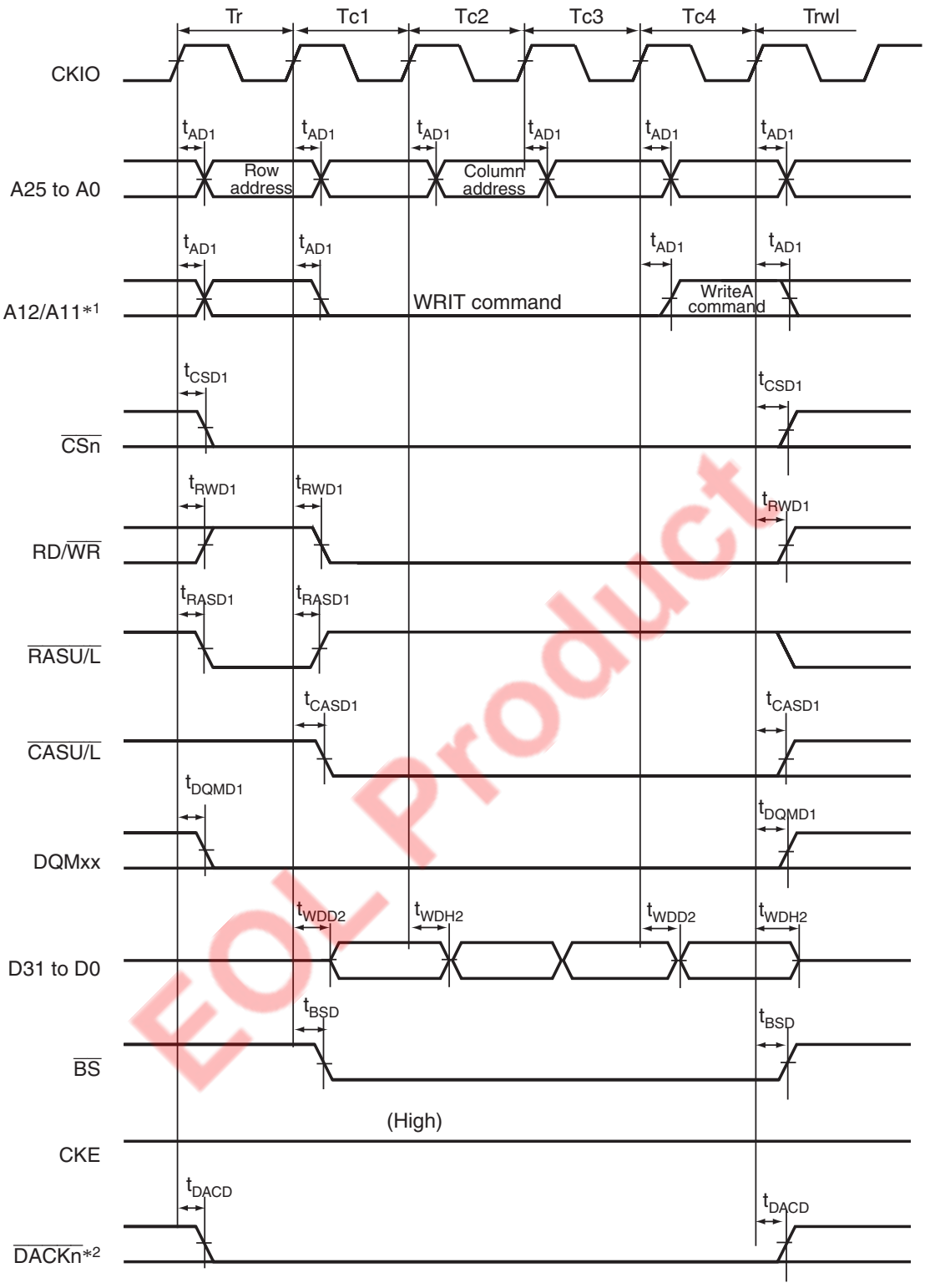
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for DACKn when active low is selected.

Figure 25.27 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRWL = 1 Cycle)



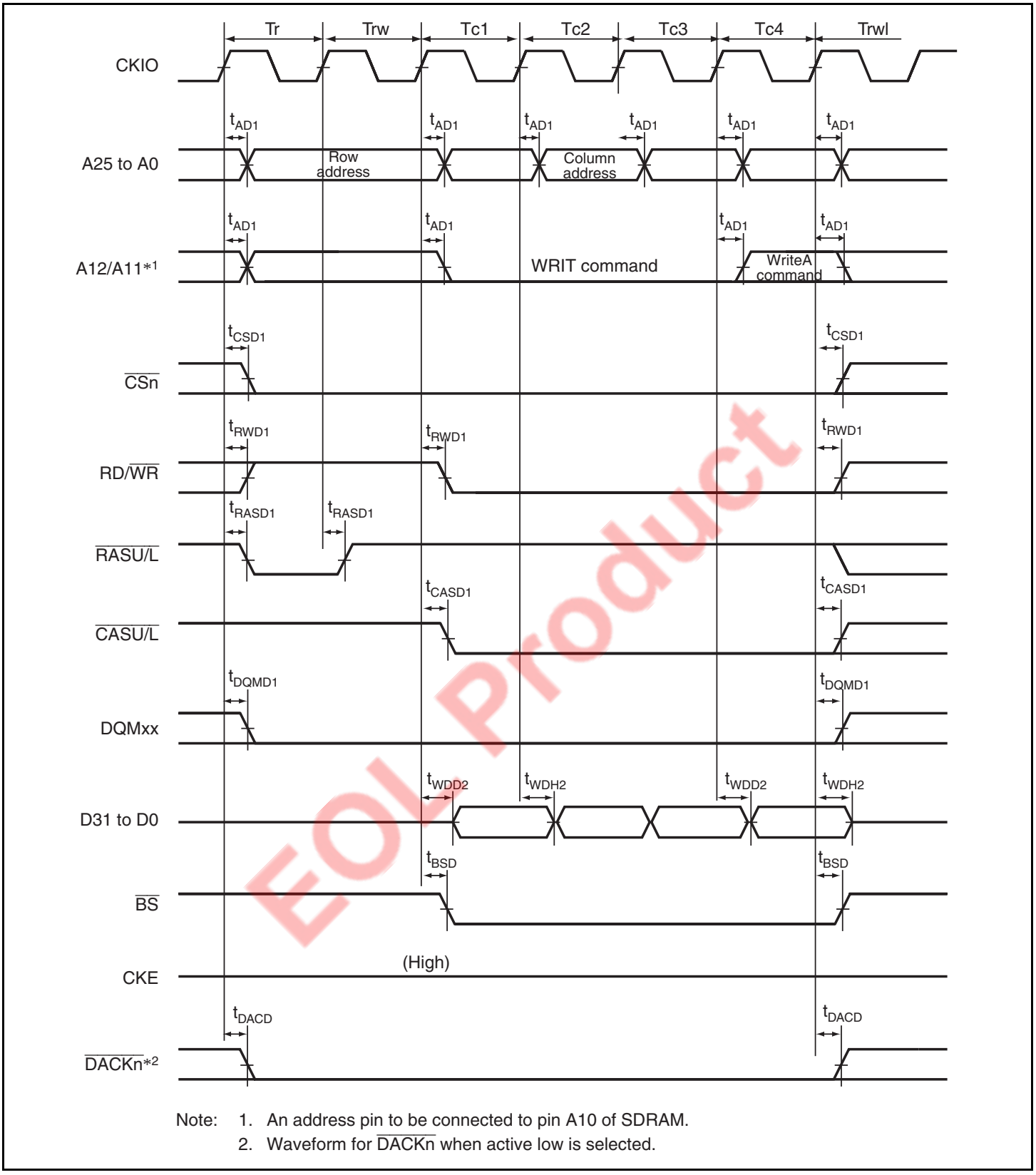
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for DACKn when active low is selected.

Figure 25.28 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle)



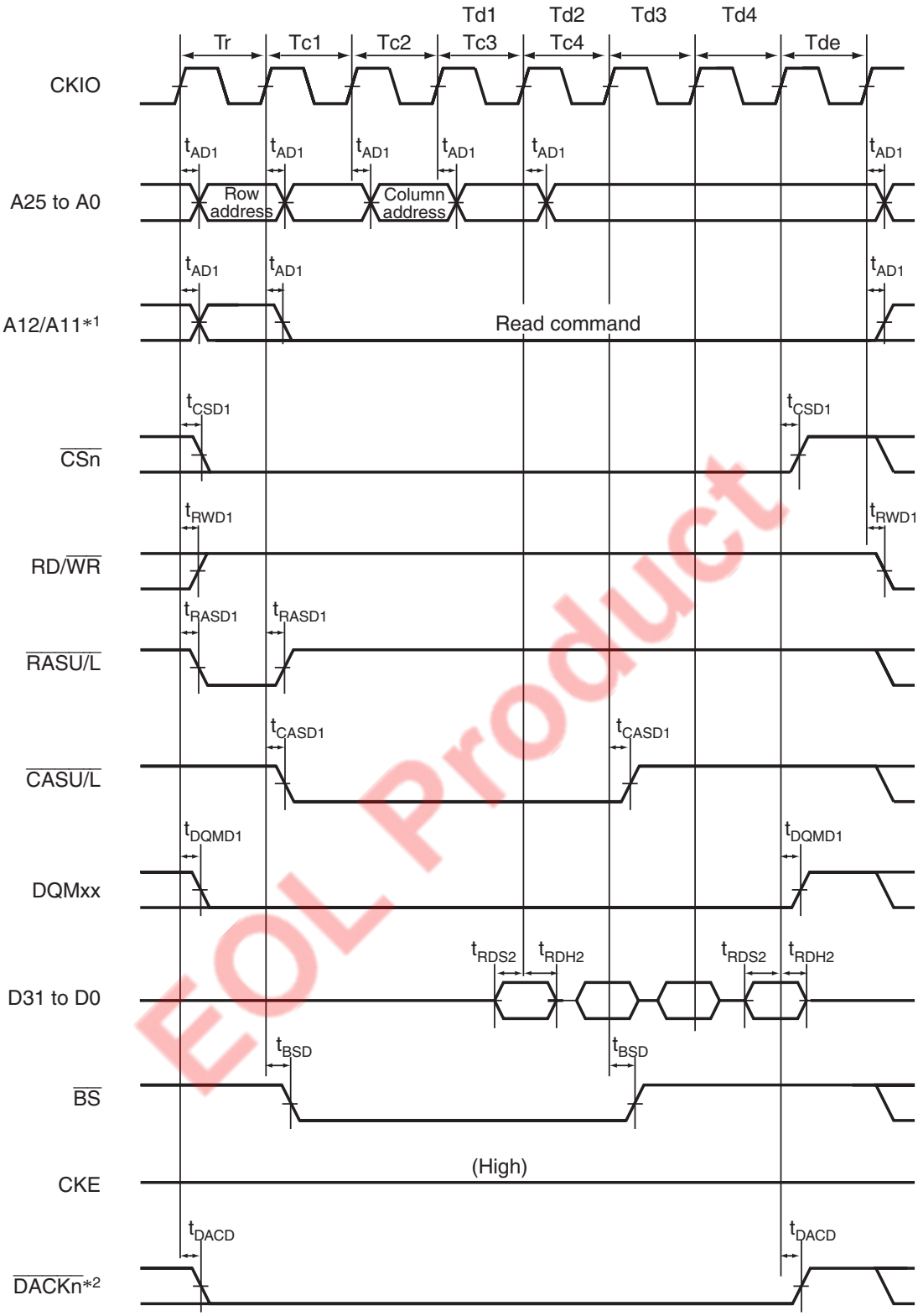
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.29 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)



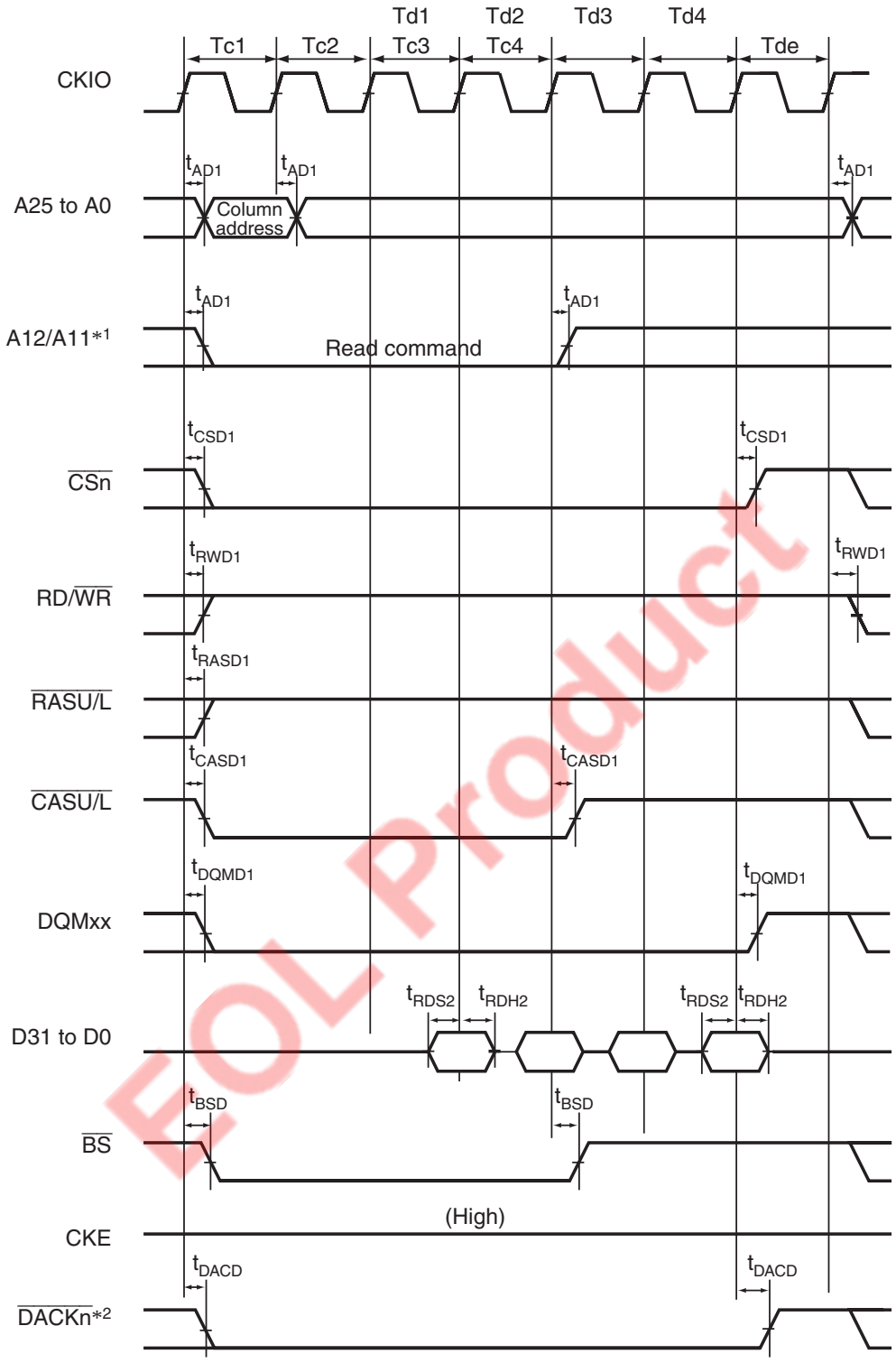
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.30 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)



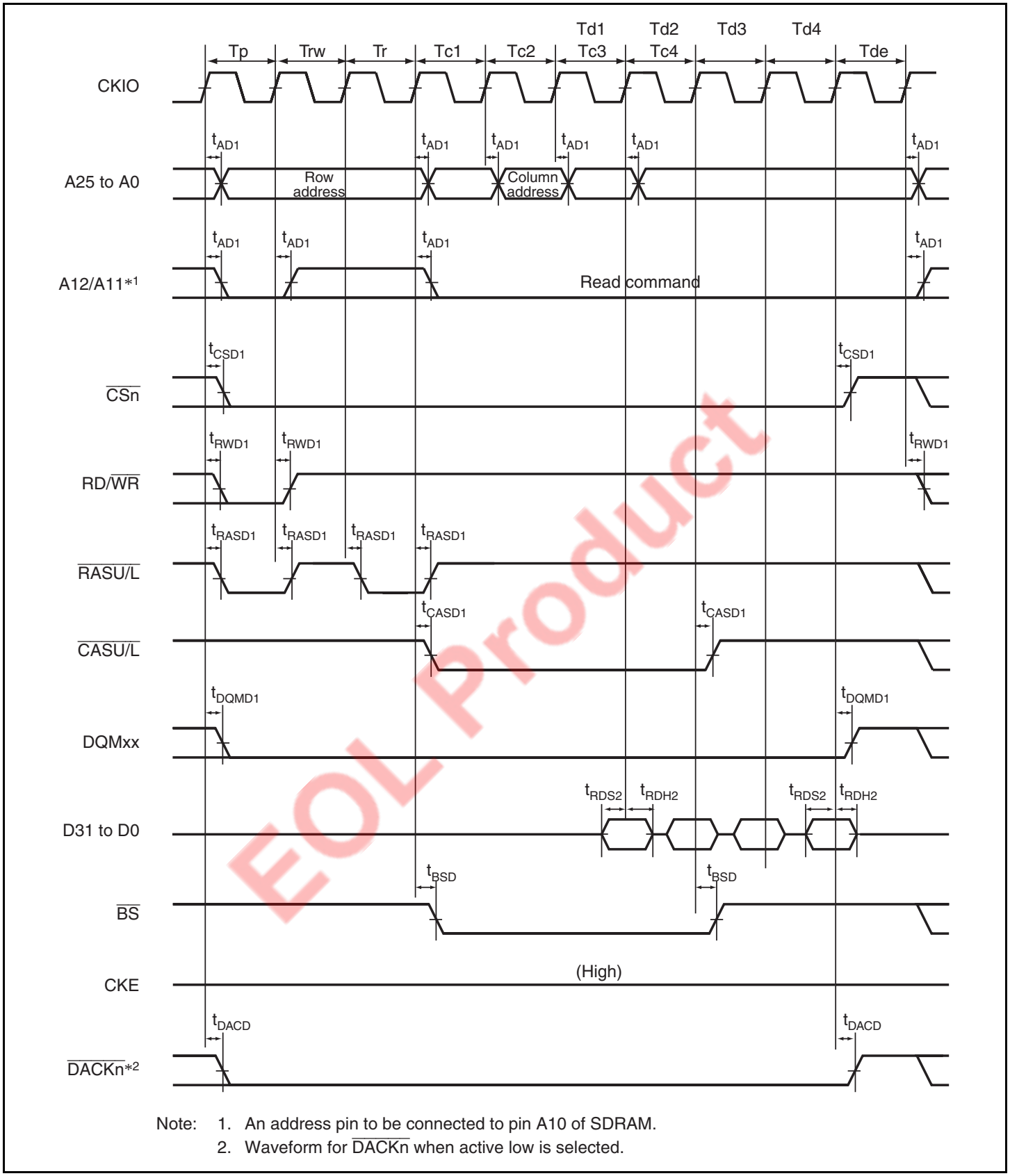
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.31 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles)
(Bank Active Mode: ACT + READ Commands, CAS Latency 2, WTRCD = 0 Cycle)



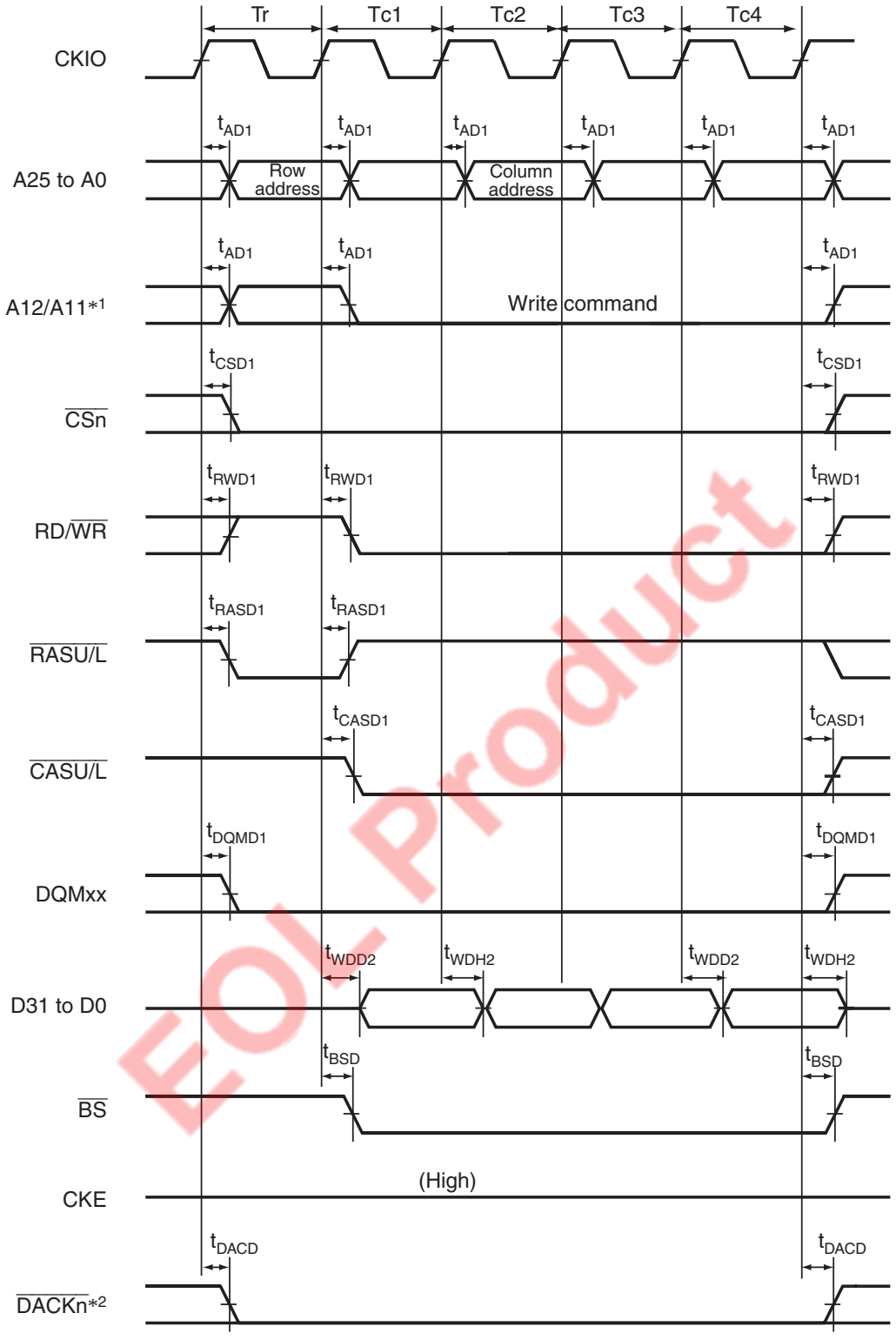
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.32 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles)
(Bank Active Mode: READ Command, Same Row Address, CAS Latency 2, WTRCD = 0 Cycle)



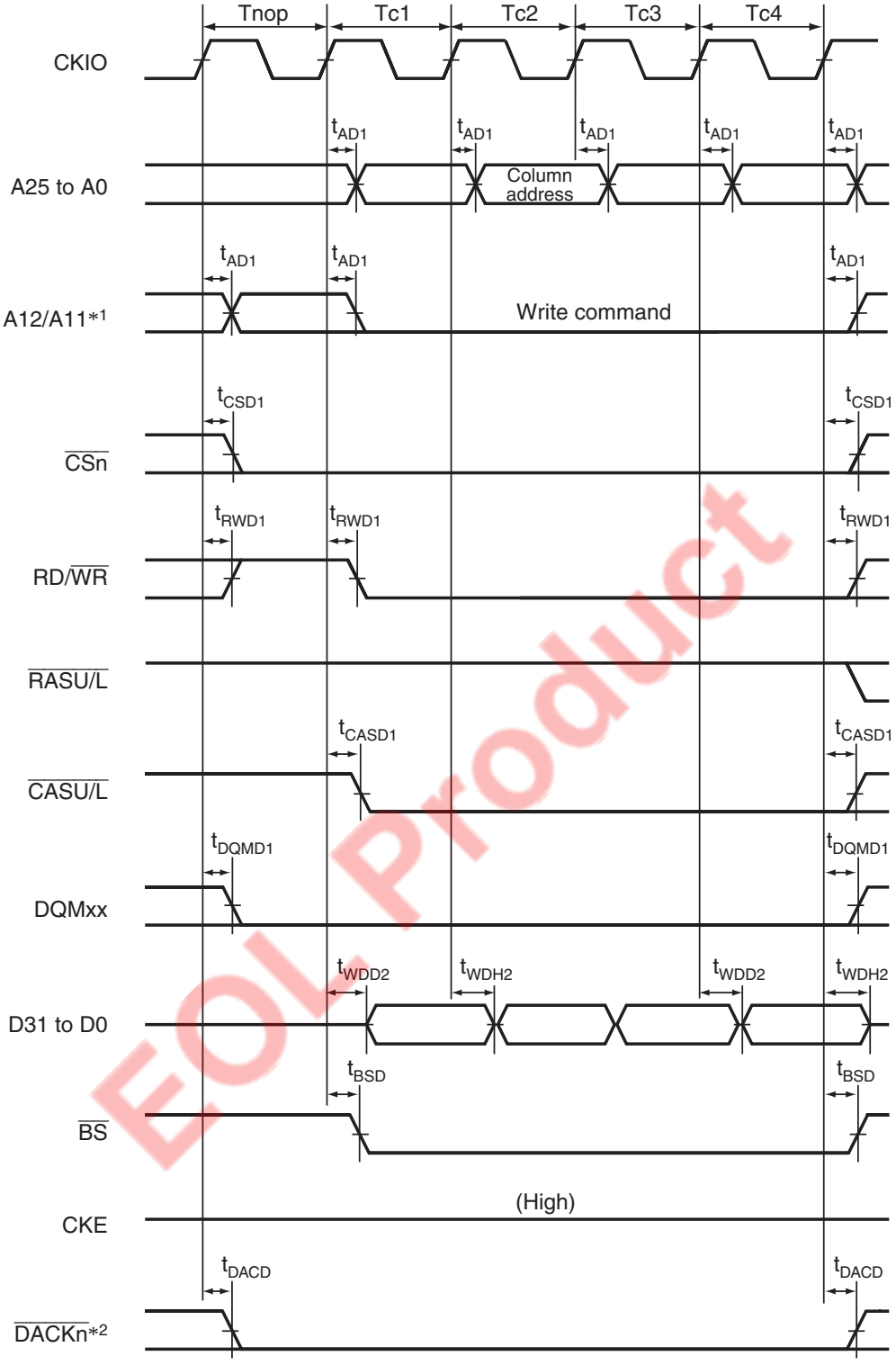
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.33 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Bank Active Mode: PRE + ACT + READ Commands, Different Row Addresses, CAS Latency 2, WTRCD = 0 Cycle)



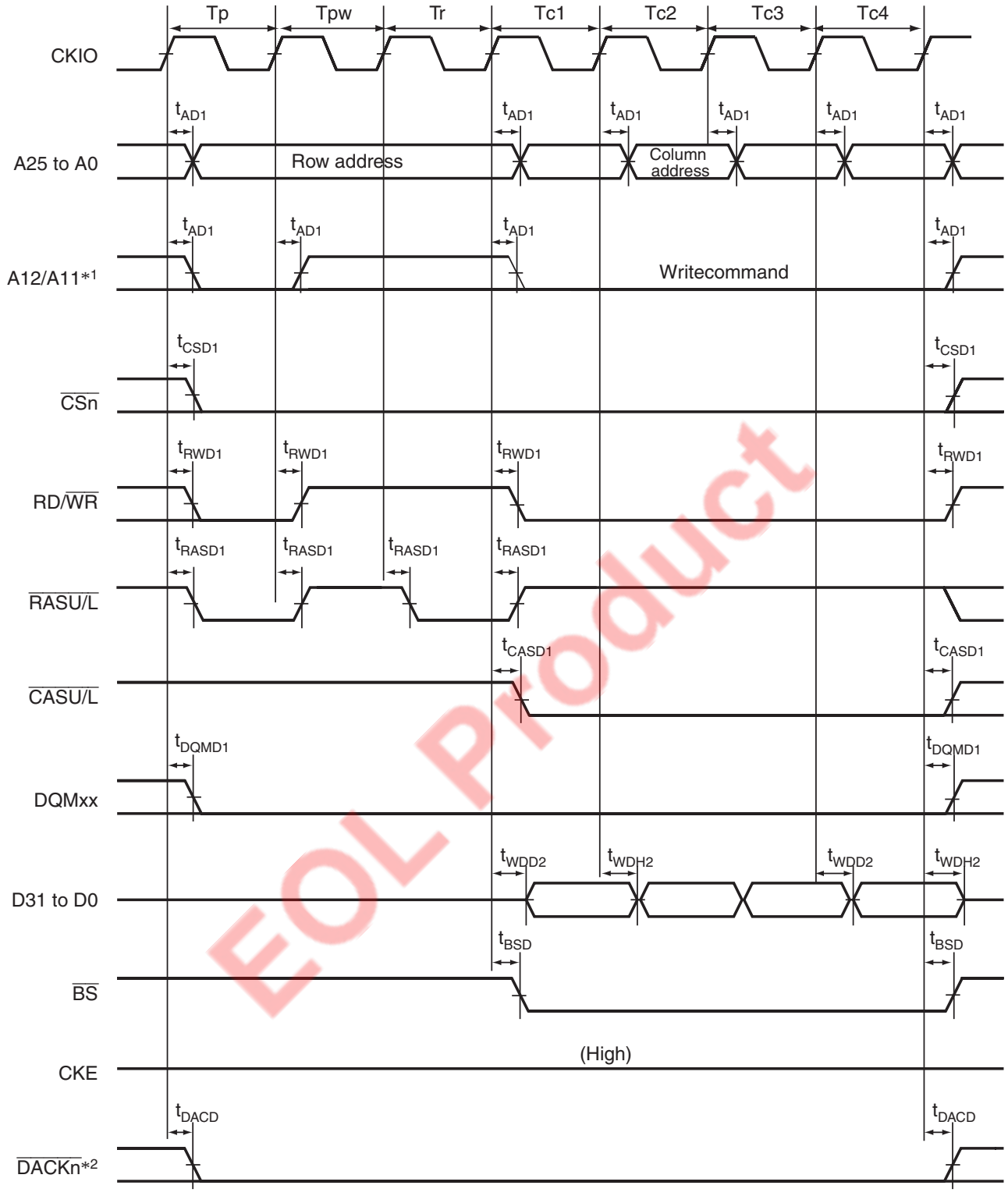
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.34 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles)
(Bank Active Mode: ACT + WRITE Commands, WTRCD = 0 Cycle, TRWL = 0 Cycle)



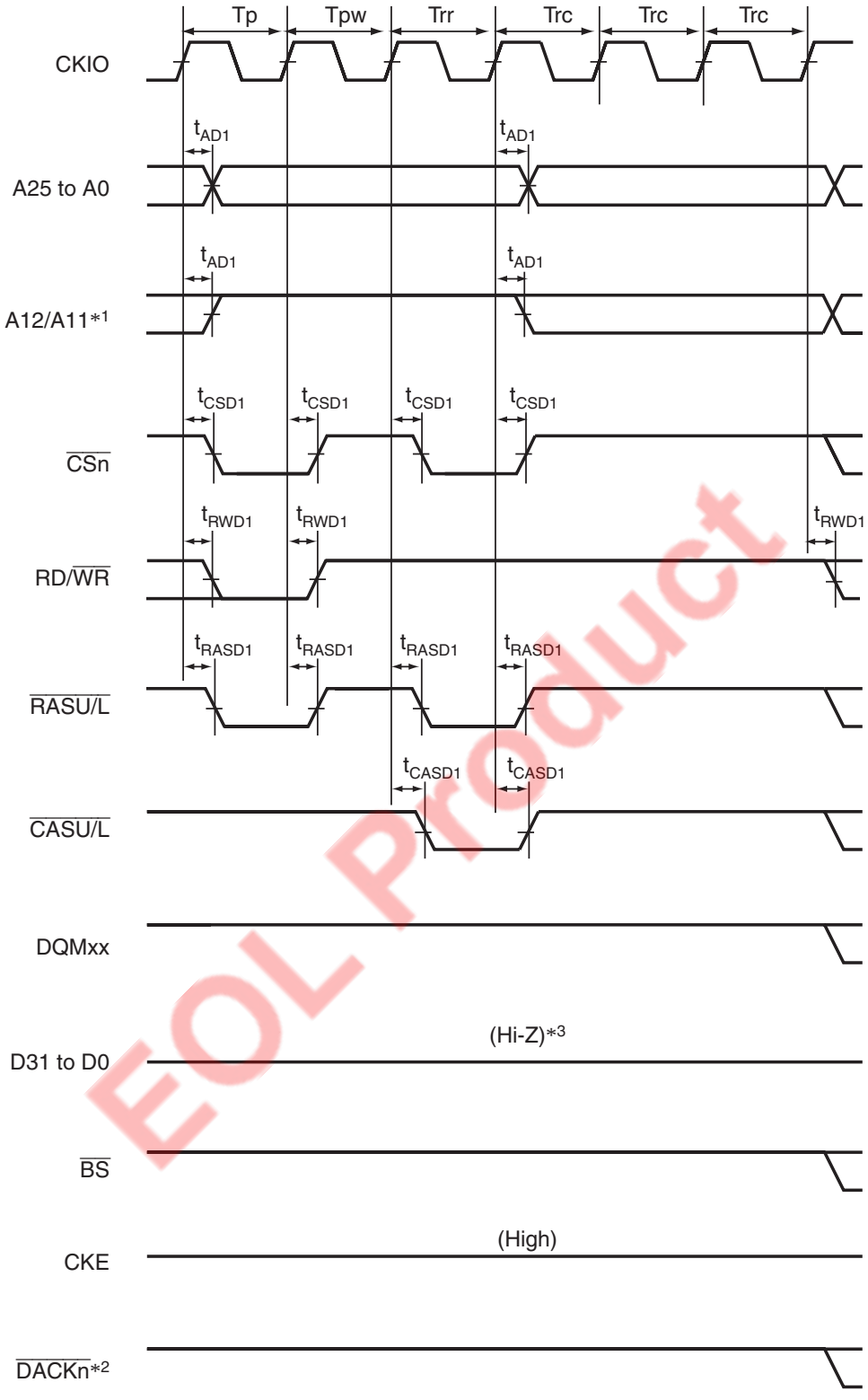
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.35 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 Cycle, TRWL = 0 Cycle)



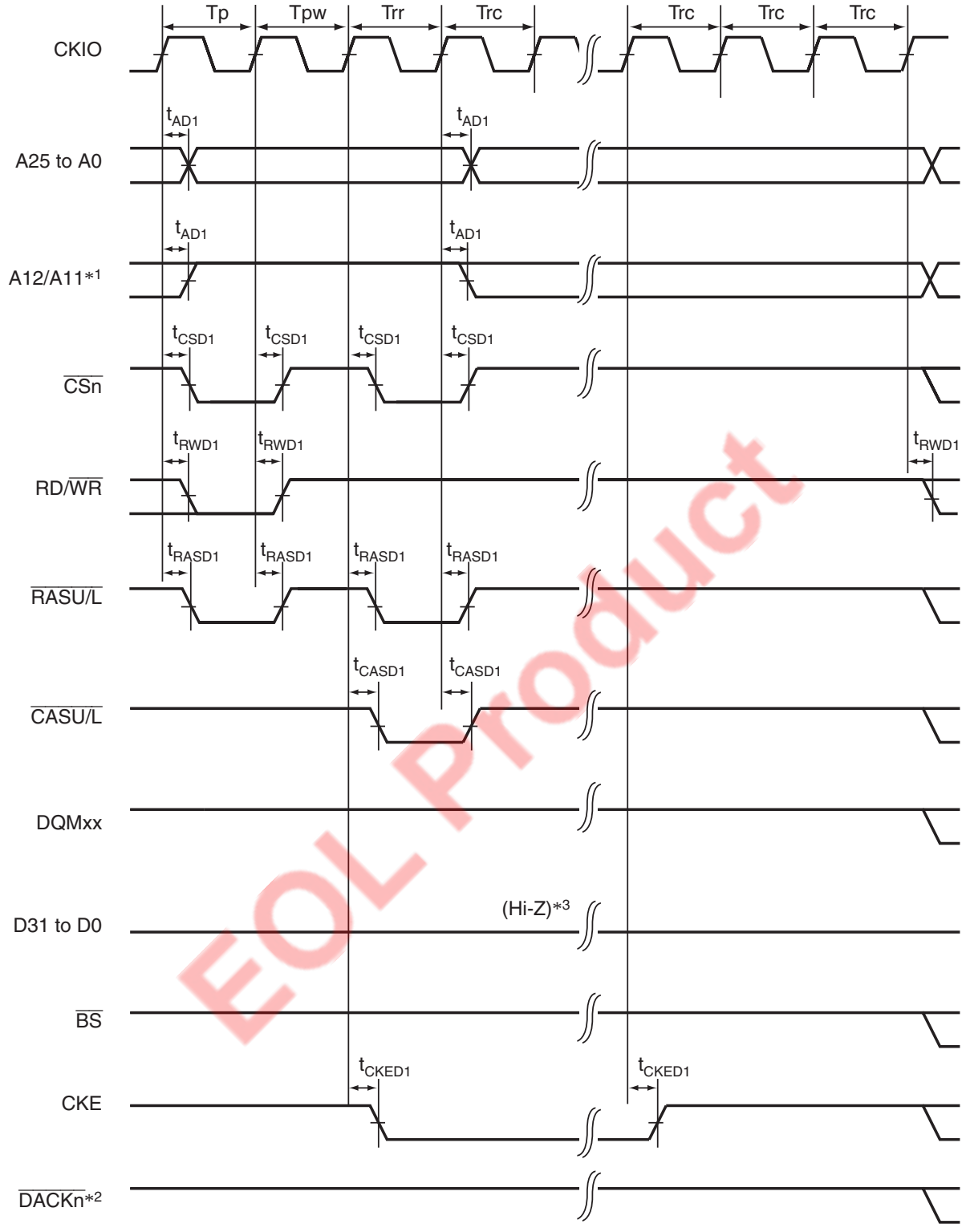
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for $\overline{\text{DACKn}}$ when active low is selected.

Figure 25.36 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: PRE + ACT + WRITE Commands, Different Row Addresses, WTRCD = 0 Cycle, TRWL = 0 Cycle)



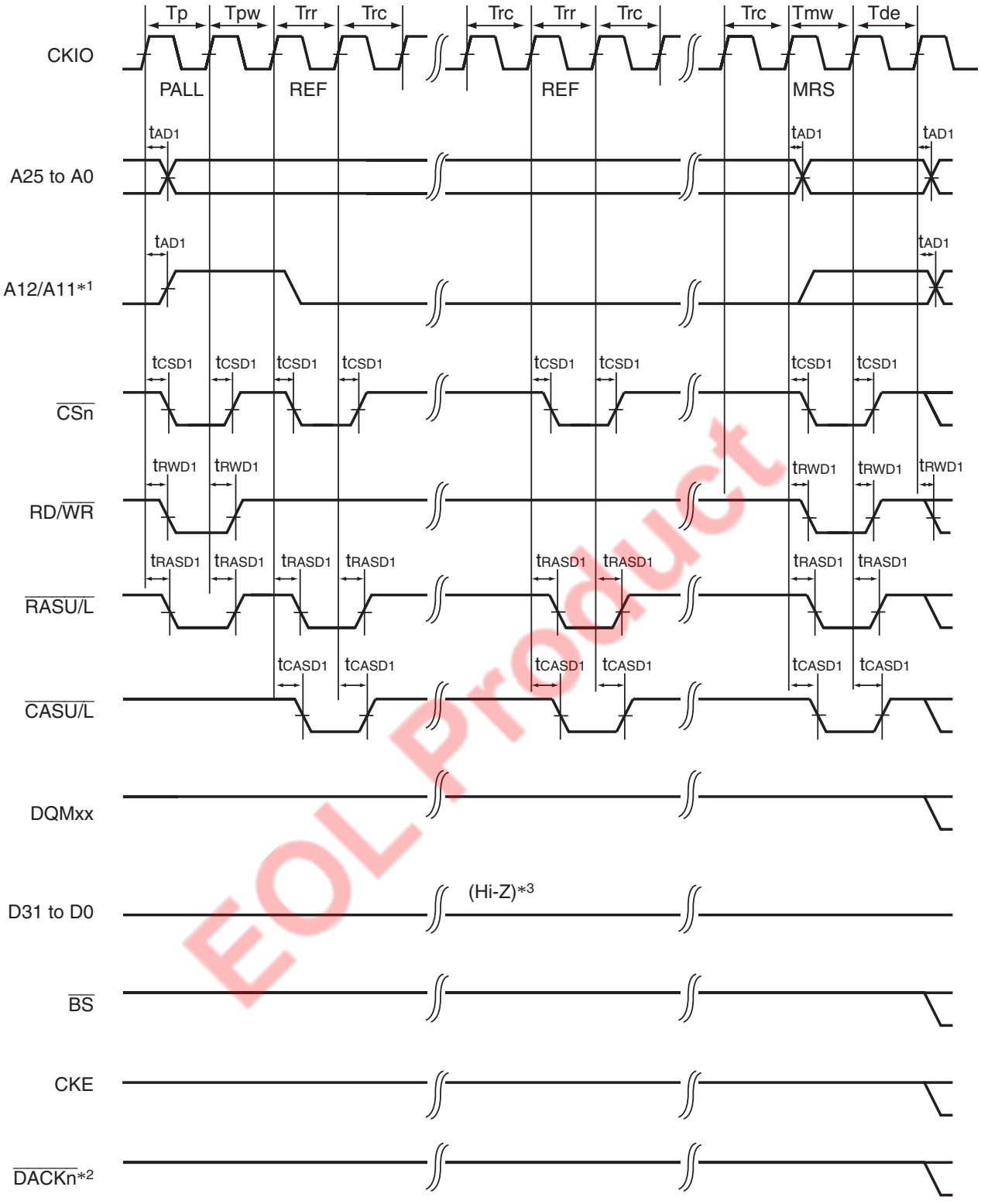
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for $\overline{\text{DACKn}}$ when active low is selected.
3. Pins D31 to D16 with weak keeper are retained as weak keepers.

Figure 25.37 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles)



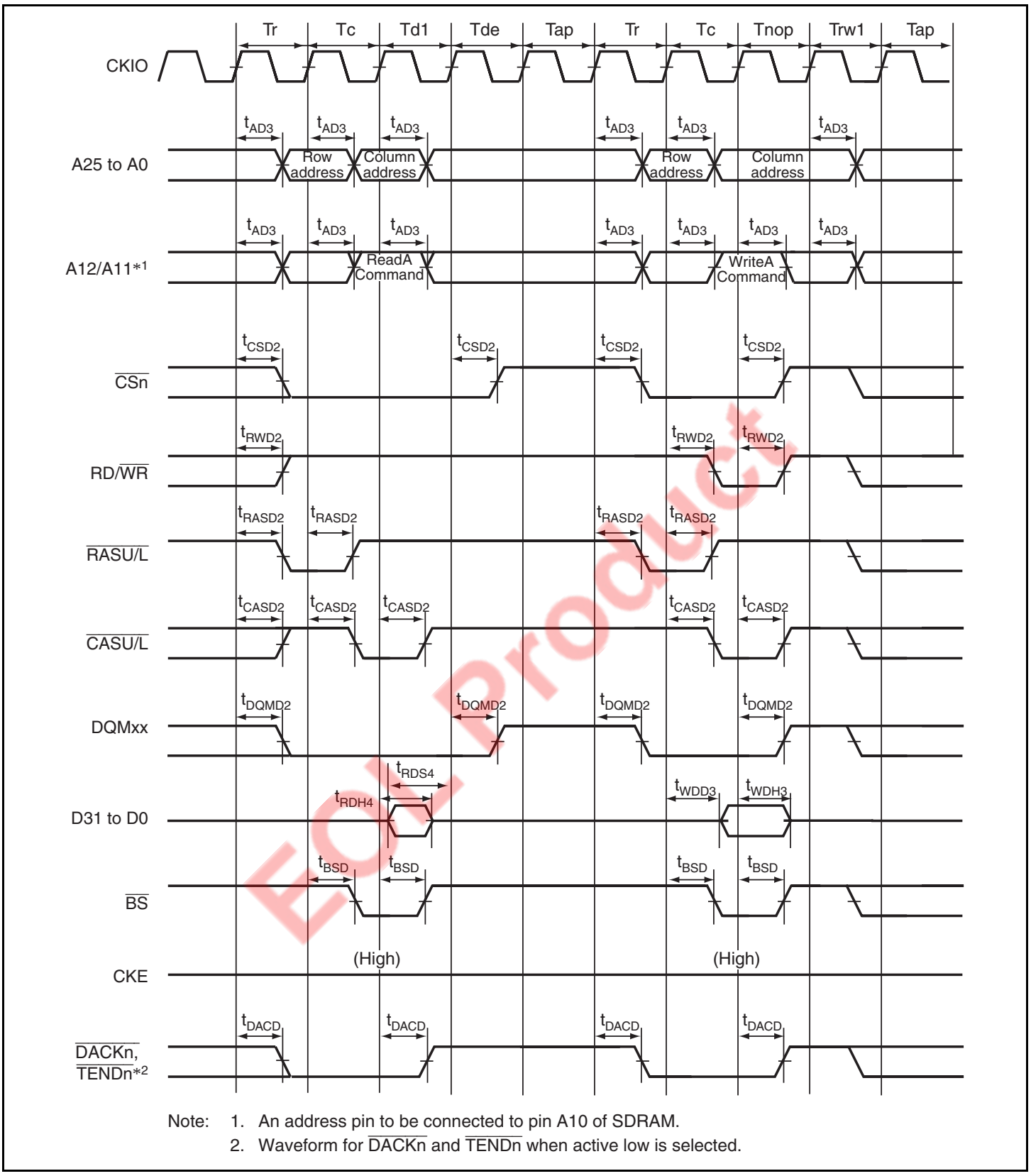
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for DACKn when active low is selected.
3. Pins D31 to D16 with weak keeper are retained as weak keepers.

Figure 25.38 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle)



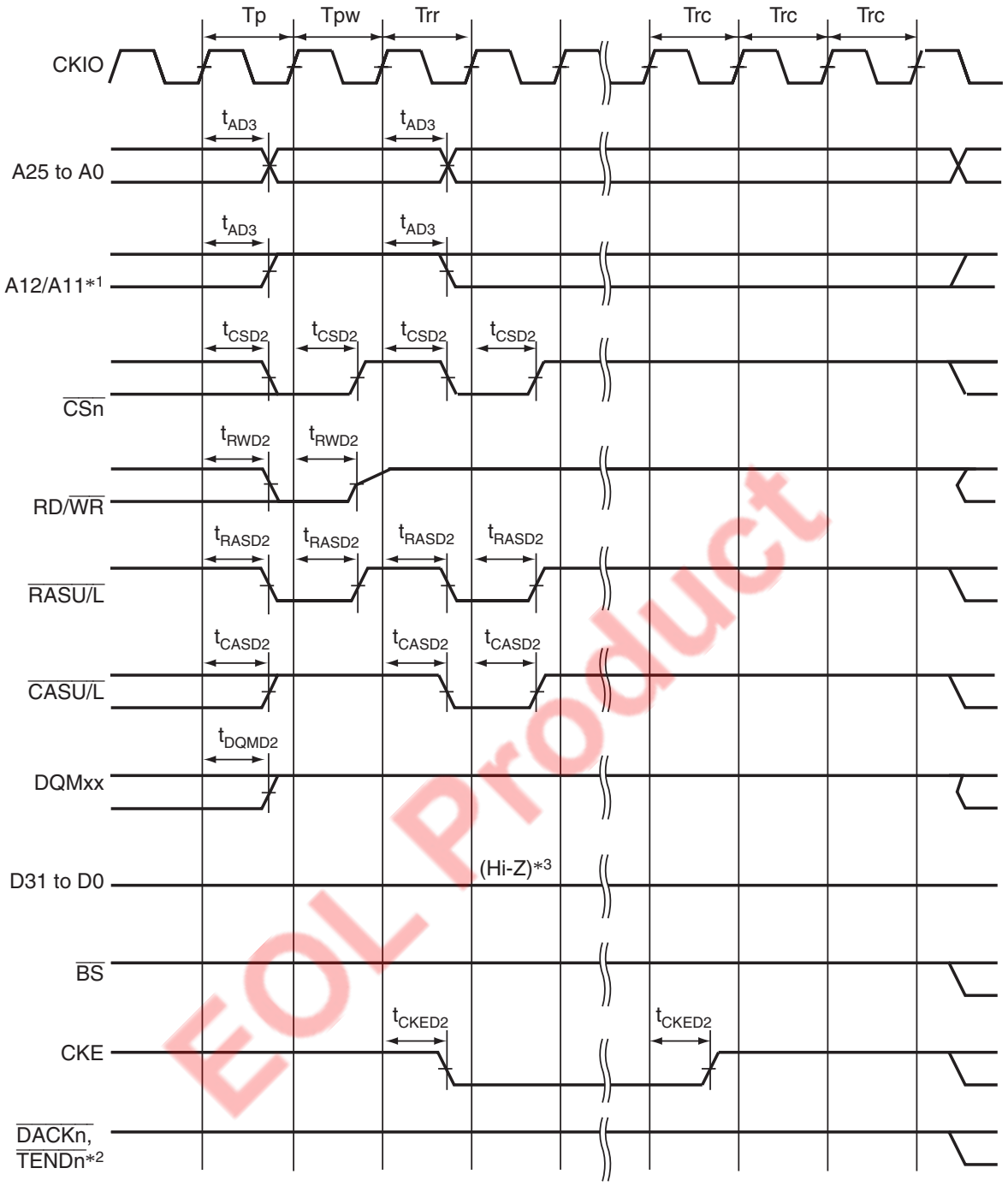
Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for $\overline{\text{DACKn}}$ when active low is selected.
3. Pins D31 to D16 with weak keeper are retained as weak keepers.

Figure 25.39 Synchronous DRAM Mode Register Write Timing (WTRP = 1 Cycle)



Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for DACKn and TENDn when active low is selected.

Figure 25.40 Synchronous DRAM Access Timing in Low-Frequency Mode (Auto-Precharge, TRWL = 2 Cycles)



Note: 1. An address pin to be connected to pin A10 of SDRAM.
2. Waveform for DACKn and TENDn when active low is selected.
3. Pins D31 to D16 with weak keeper are retained as weak keepers.

Figure 25.41 Synchronous DRAM Self-Refreshing Timing in Low-Frequency Mode (WTRP = 2 Cycles)

25.3.8 Peripheral Module Signal Timing

Table 25.9 Peripheral Module Signal Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.8 V ± 5%, AV_{CC} = 3.0 V to 3.6 V,
$V_{SS} = V_{SS}Q = AV_{SS}$ = 0 V, Ta = –40°C to +85°C

Module	Item		Symbol	Min.	Max.	Unit	Figure(s)
SCIF	Input clock cycle	(synchronous)	t_{Scyc}	16	—	t_{Pcyc}	25.42
		(asynchronous)		4	—	t_{Pcyc}	25.42
	Input clock rising time		t_{SCKR}	—	1.5	t_{Pcyc}	25.42
	Input clock falling time		t_{SCKF}	—	1.5	t_{Pcyc}	25.42
	Input clock width		t_{SCKW}	0.4	0.6	t_{Scyc}	25.42
	Transmit data delay time (synchronous)		t_{TXD}	—	3 t_{Pcyc} + 15	ns	25.43
	Receive data setup time (synchronous)		t_{RXS}	4 t_{Pcyc} + 15	—	ns	25.43
	Receive data hold time (synchronous)		t_{RXH}	100	—	ns	25.43
PORT	Output data delay time		t_{PORTD}	—	100	ns	25.44
	Input data setup time		t_{PORTS2}	100	—		
	Input data hold time		t_{PORTH2}	100	—		
DMAC	$\overline{\text{DREQ}}$ setup time		t_{DREQ}	8	—	ns	25.45
	$\overline{\text{DREQ}}$ hold time		t_{DREQH}	8	—		
	$\overline{\text{DACK}}$, $\overline{\text{TEND}}$ delay time		t_{DACD}	—	12		25.46

Note: * t_{Pcyc} indicate Pclock cycle.

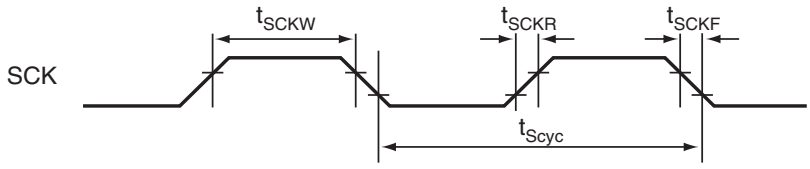


Figure 25.42 SCK Input Clock Timing

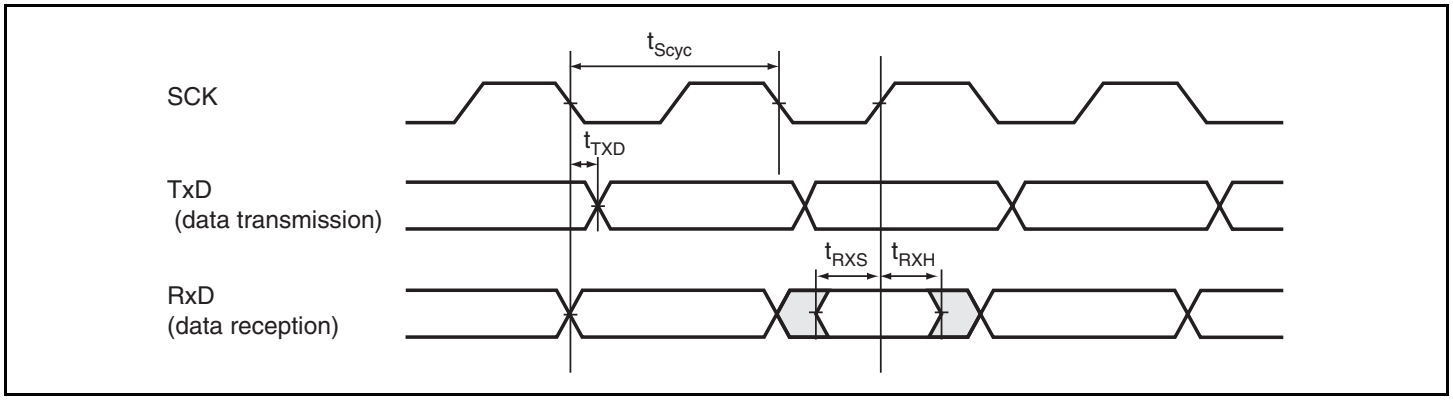


Figure 25.43 SCIF Input/Output Timing in Synchronous Mode

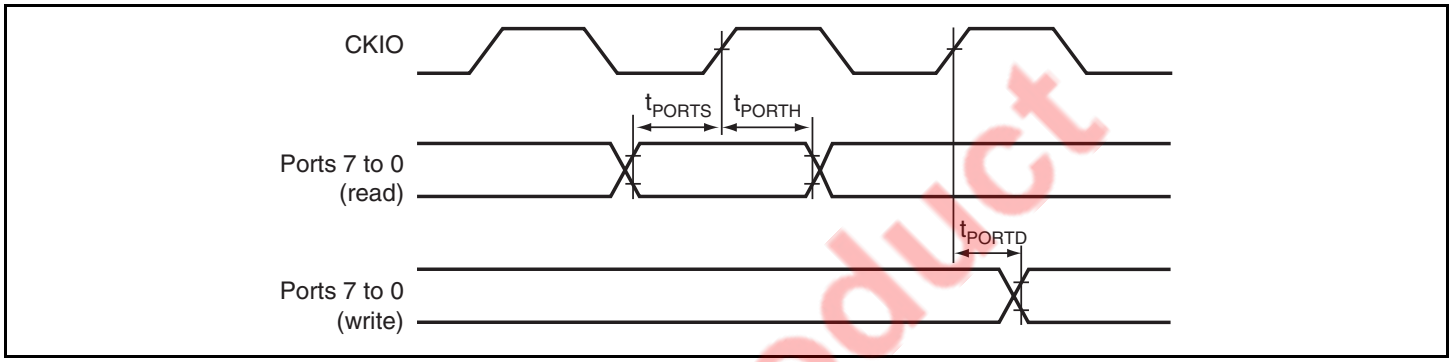


Figure 25.44 I/O Port Timing

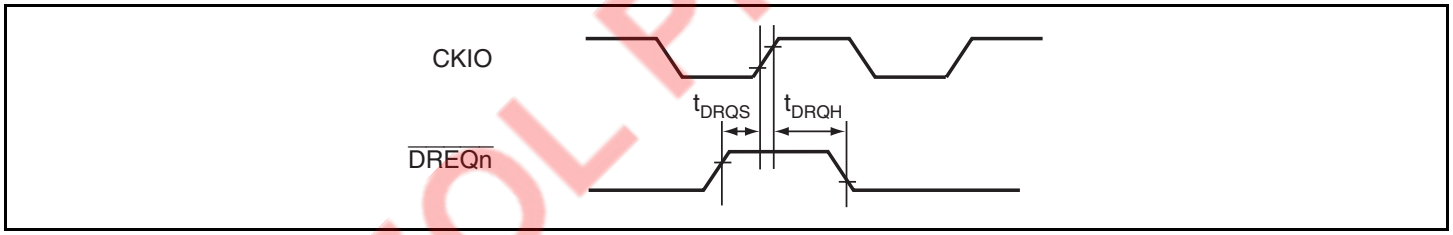


Figure 25.45 $\overline{\text{DREQ}}$ Input Timing

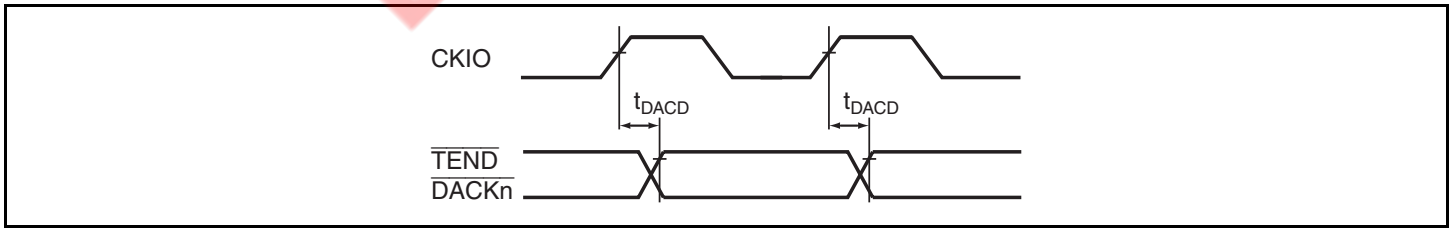


Figure 25.46 $\overline{\text{DACK}}$, $\overline{\text{TEND}}$ Output Timing

25.3.9 Multi Function Timer Pulse Unit Timing

Table 25.10 lists the multi function timer pulse unit timing.

Table 25.10 Multi Function Timer Pulse Unit Timing

Conditions: V_{CC} = 1.8 V ± 5%, $V_{CC}Q = AV_{CC}$ = 3.0 V to 3.6 V, $V_{SS} = V_{SS}Q = AV_{SS}$ = 0 V,
Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure(s)
Output compare output delay time	t_{TOCD}	—	$B_{cyc}/2 + 20$	ns	25.47
Input capture input setup time	t_{TICS}	$B_{cyc}/2 + 20$	—	ns	
Timer input setup time	t_{TCKS}	$B_{cyc}/2 + 20$	—	ns	25.48
Timer clock pulse width (single edge)	$t_{TCKWH/L}$	1.5	—	t_{pcyc}	
Timer clock pulse width (both edges)	$t_{TCKWH/L}$	2.5	—	t_{pcyc}	
Timer clock pulse width (phase counting mode)	$t_{TCKWH/L}$	2.5	—	t_{pcyc}	

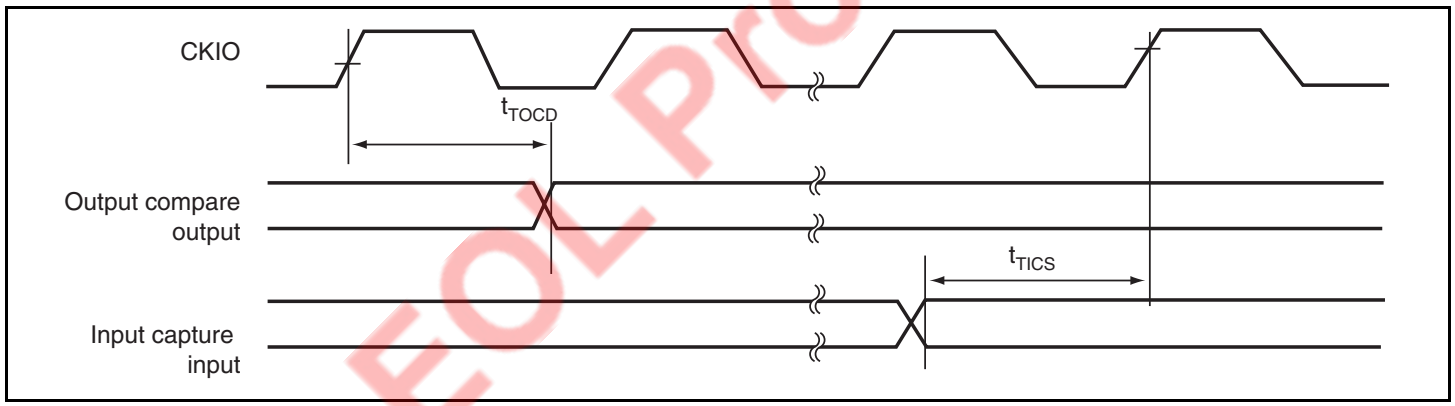


Figure 25.47 MTU Input/Output Timing

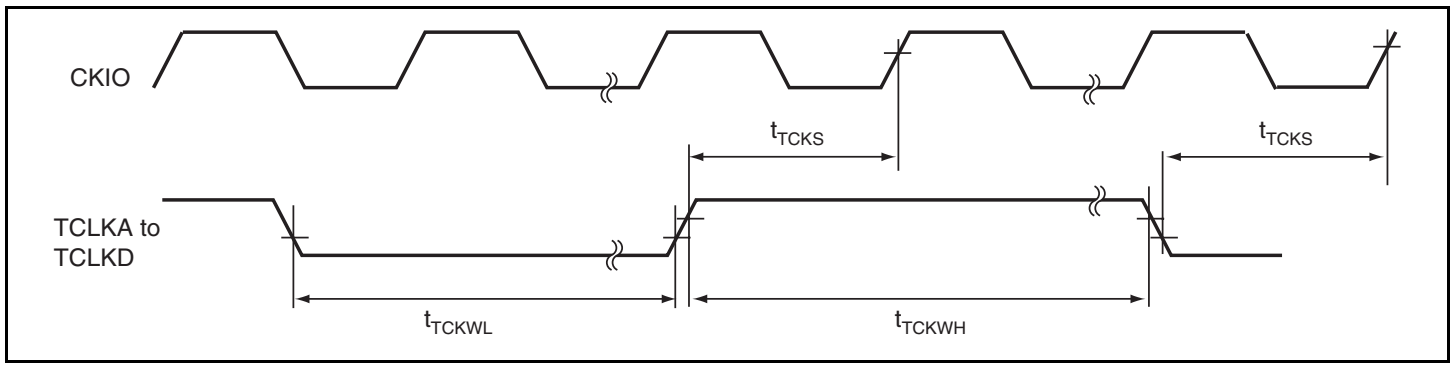


Figure 25.48 MTU Clock Input Timing

25.3.10 POE Module Signal Timing

Table 25.11 Output Enable (POE) Timing

Conditions: V_{CC} = 1.8 V ± 5%, $V_{CC}Q = AV_{CC}$ = 3.0 V to 3.6 V, $V_{SS} = V_{SS}Q = AV_{SS}$ = 0 V,
Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure(s)
$\overline{\text{POE}}$ input setup time	t_{POES}	B_{cyc}/2+10	—	ns	25.49
$\overline{\text{POE}}$ input pulse width	t_{POEW}	1.5	—	t_{pcyc}	

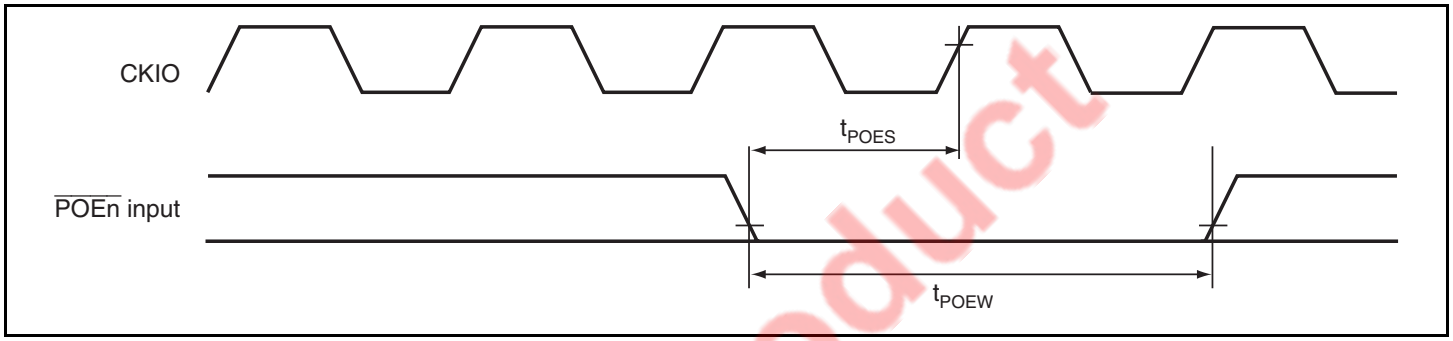


Figure 25.49 POE Input/Output Timing

25.3.11 I²C Module Signal Timing

Table 25.12 I²C Bus Interface Timing

Normal Conditions: V_{CC} = 1.8 V ± 5%, AV_{CC} = $V_{CC}Q$ = 3.0 V to 3.6 V, V_{SS} = AV_{SS} = $V_{SS}Q$ = 0 V, Ta = –40°C to +85°C

			Specifications				
Item	**Symbol**	**Test Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**	**Figure(s)**
SCL input cycle time	t_{SCL}		12 t_{Pcyc} + 600	—	—	ns	25.50
SCL input high pulse width	t_{SCLH}		3 t_{Pcyc} + 300	—	—	ns	
SCL input low pulse width	t_{SCLL}		5 t_{Pcyc} + 300	—	—	ns	
SCL, SDA input rising time	t_{SR}		—	—	300	ns	
SCL, SDA input falling time	t_{SF}		—	—	300	ns	
SCL, SDA input spike pulse removal time*[2]	t_{SP}		—	—	1.2	t_{Pcyc} *[1]	
SDA input bus free time	t_{BUF}		5 t_{Pcyc}	—	—	t_{Pcyc}	
Start condition input hold time	t_{STAH}		3 t_{Pcyc}	—	—	t_{Pcyc}	
Retransmit start condition input setup time	t_{STAS}		3 t_{Pcyc}	—	—	t_{Pcyc}	
Stop condition input setup time	t_{STOS}		3 t_{Pcyc}	—	—	t_{Pcyc}	
Data input setup time	t_{SDAS}		1 t_{Pcyc} + 20	—	—	ns	
Data input hold time	t_{SDAH}		0	—	—	ns	
SCL, SDA capacitive load	Cb		0	—	400	pF	
SCL, SDA output falling time	t_{SF}	$V_{CC}Q$ = 3.0 to 3.6 V	—	—	250*[3]	ns	

Note: 1. Pcyc indicates peripheral clock cycle.
2. Depends on the value of the register NF2CYC.
3. Indicates the I/O buffer characteristic.

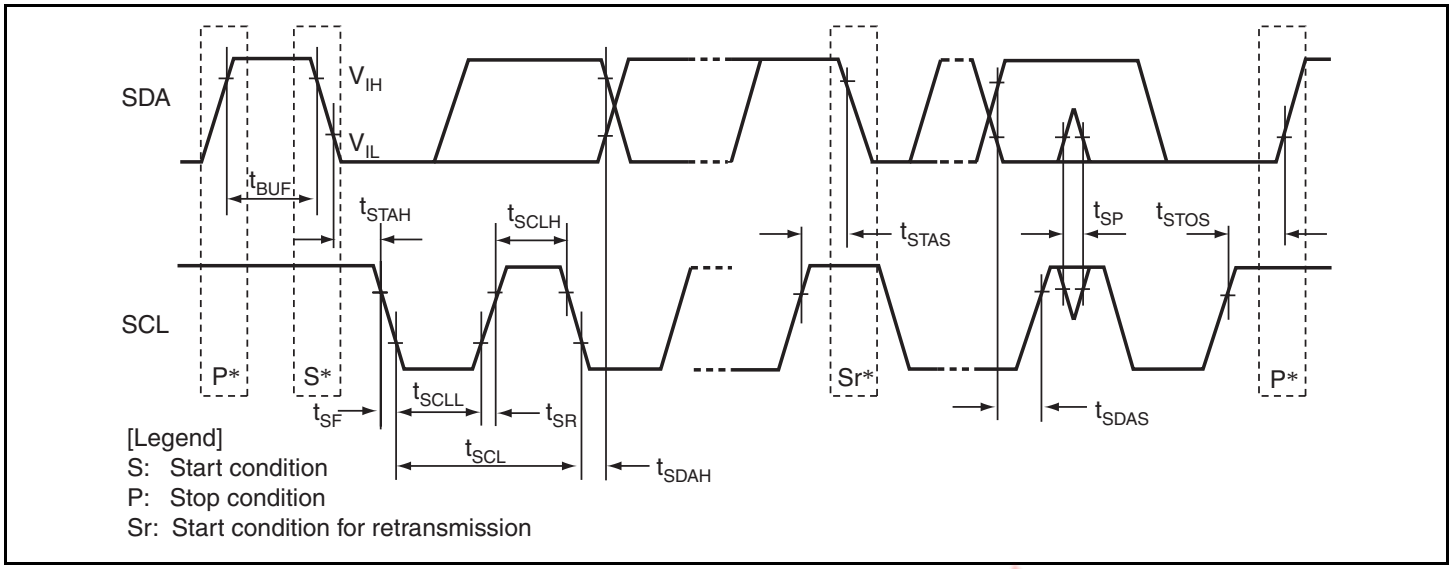


Figure 25.50 I^2C Bus Interface Input/Output Timing

25.3.12 H-UDI Related Pin Timing

Table 25.13 H-UDI Related Pin Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.8 V ± 5%, AV_{CC} = 3.0 V to 3.6 V,
$V_{SS} = V_{SS}Q = AV_{SS}$ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure(s)
TCK cycle time	t_{TCKcyc}	50	—	ns	25.51
TCK high pulse width	t_{TCKH}	0.4	0.6	tTckcyc	
TCK low pulse width	t_{TCKL}	0.4	0.6	tTckcyc	
$\overline{\text{TRST}}$ setup time	t_{TRSTS}	20	—	ns	25.52
$\overline{\text{TRST}}$ hold time	t_{TRSTH}	50	—	t_{cyc}	
TDI setup time	t_{TDIS}	10	—	ns	25.53
TDI hold time	t_{TDIH}	10	—	ns	
TMS setup time	t_{TMSS}	10	—	ns	
TMS hold time	t_{TMSH}	10	—	ns	
TDO delay time	t_{TDOD}	—	20	ns	
Capture register setup time	t_{CAPTS}	10	—	ns	25.54
Capture register hold time	t_{CAPTH}	10	—	ns	
Update register delay time	t_{UPDTED}	—	16	ns	

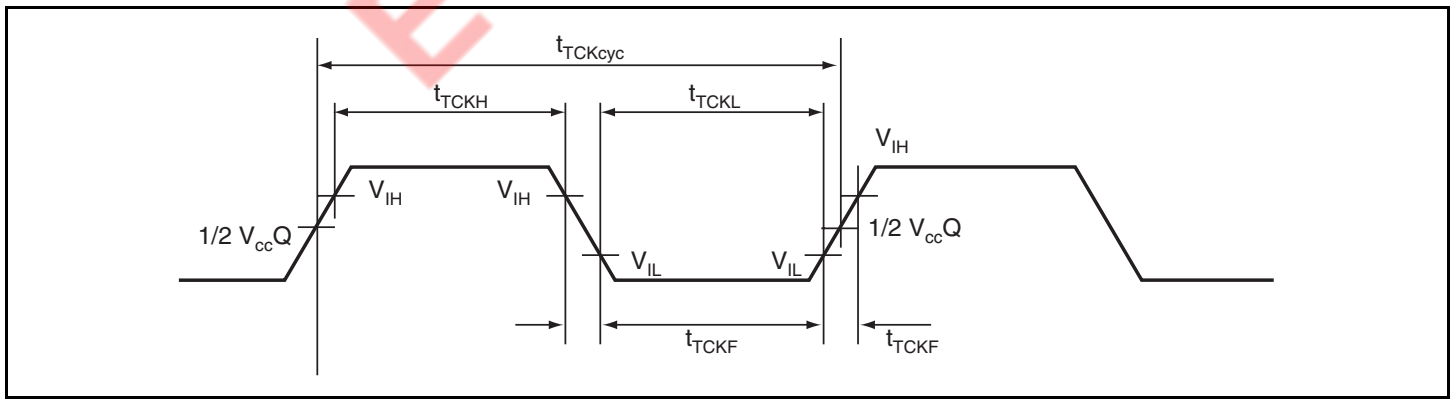


Figure 25.51 TCK Input Timing

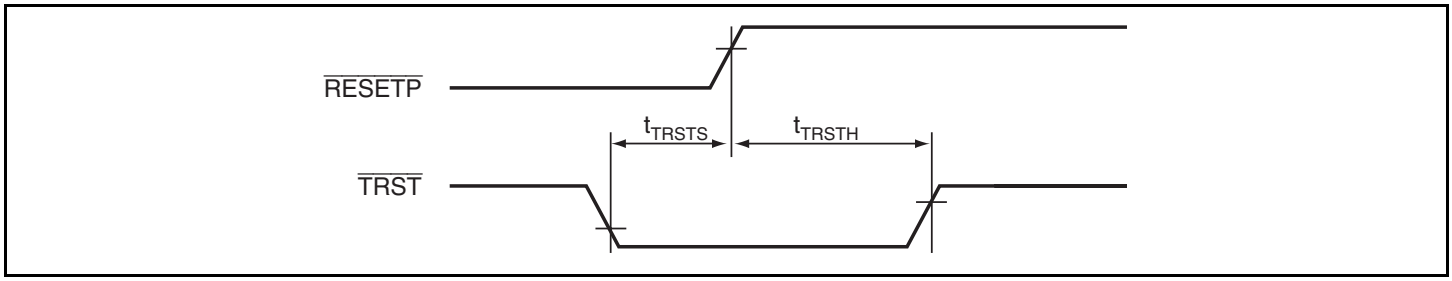


Figure 25.52 $\overline{\text{TRST}}$ Input Timing (Reset-Hold State)

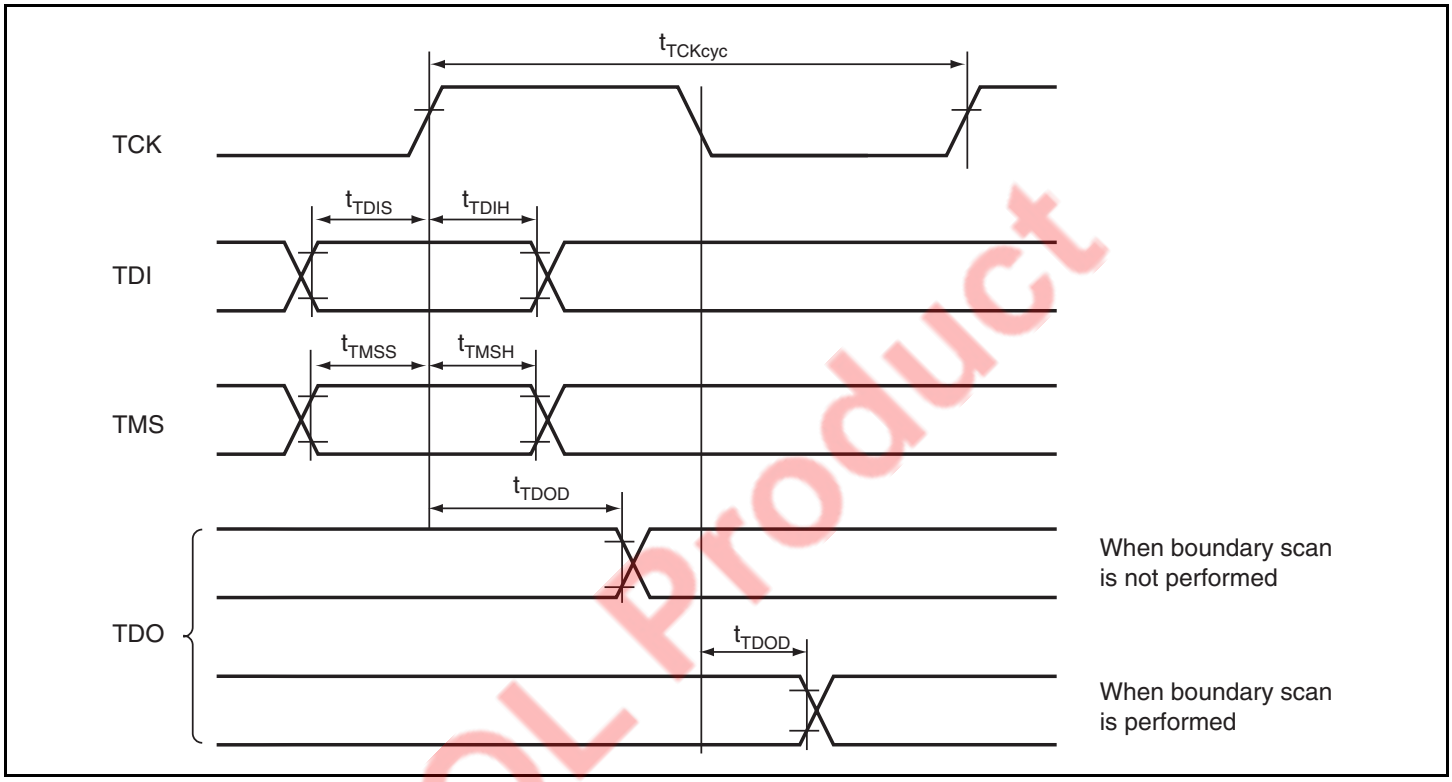


Figure 25.53 H-UDI Data Transfer Timing

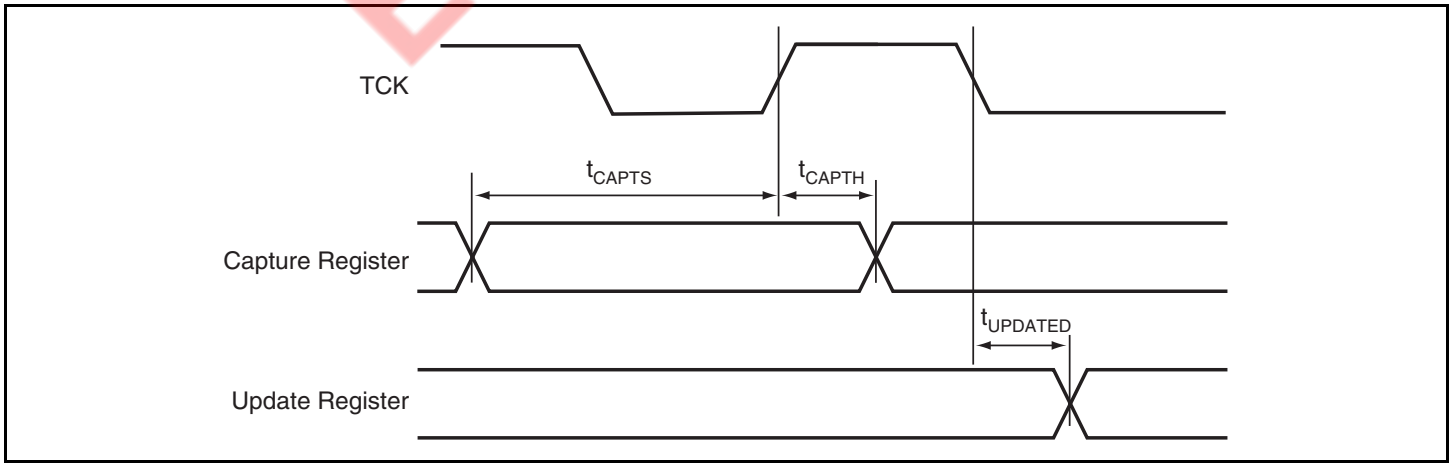


Figure 25.54 Boundary-Scan Input/Output Timing

25.3.13 USB Module Signal Timing

Table 25.14 USB Module Clock Timing

Conditions: V_{CC} = 1.8 V ±5%, $V_{CC}Q$ = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{SS} = $V_{SS}Q$ = AV_{SS} = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure(s)
Frequency (48 MHz)	t_{FREQ}	47.9	48.1	MHz	25.55
Clock rising time	t_{RAS}	—	4	ns	
Clock falling time	t_{FAS}	—	4	ns	
Duty cycle (t_{HIGH}/t_{LOW})	t_{DUTY}	90	110	%	

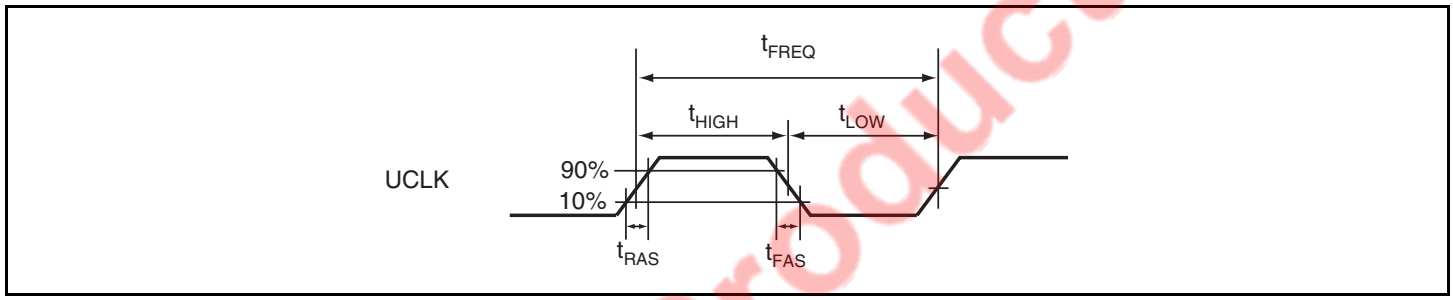


Figure 25.55 USB Clock Timing

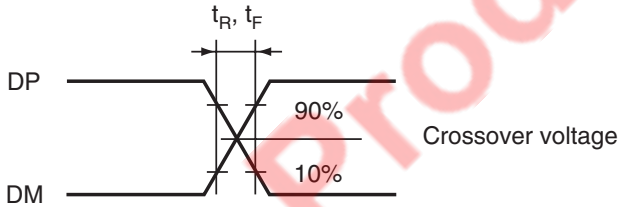
25.3.14 USB Transceiver Timing

Table 25.15 USB Transceiver Timing

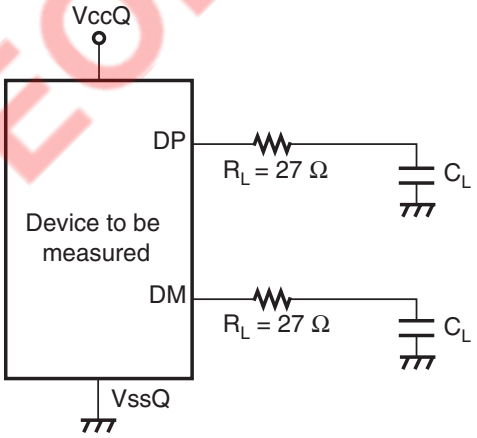
Conditions: V_{CC} = 1.8 V ±5%, $V_{CC}Q$ = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{SS} = $V_{SS}Q$ = AV_{SS} = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Rising time	t_R	4	—	20	ns	C_L = 50pF
Falling time	t_F	4	—	20	ns	C_L = 50pF
Rising/falling time ratio	t_R/t_F	90	—	110	%	
Output crossover voltage	V_{CRS}	1.3	—	2.0	V	C_L = 50pF
Output driver resistance	Z_{DRU}	28		44	Ω	

Notes: 1. Transceivers conform to the full-speed specification.
2. The resistance includes the value of the externally connected resistor (RS = 27 Ω ± 1%).



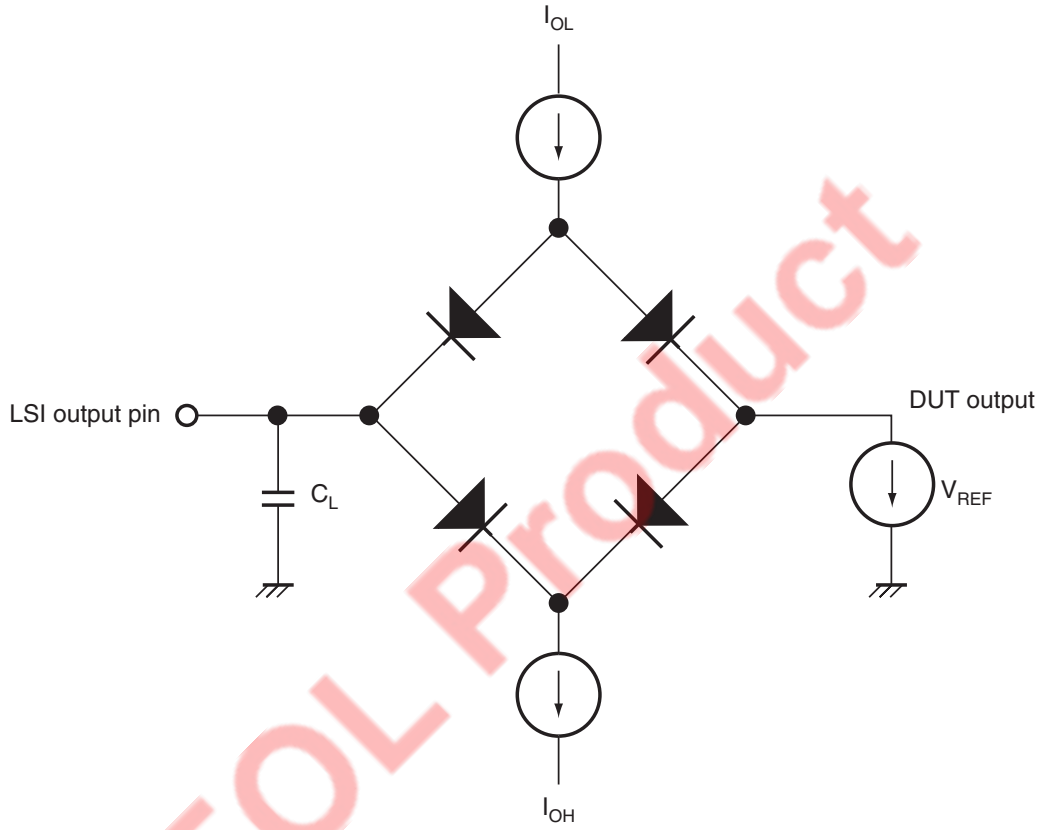
• Measurement circuit



Notes: 1. The Values of t_R and t_F are measured at 10% and 90% of amplitude.
2. The capacitance (C_L) includes stray capacitance of writing connection and probe input capacitance.

25.3.15 AC Characteristics Measurement Conditions

- I/O signal reference level: $V_{CC}Q/2$ ($V_{CC}Q$ = 3.0 to 3.6 V, V_{CC} = 1.8 V ± 5%)
- Input pulse level: $V_{SS}Q$ to 3.0 V (where $\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, $\overline{\text{ASEMD0}}$, $\overline{\text{NMI}}$, $\overline{\text{TRST}}$, EXTAL, CKIO, TCK, MD0, MD2, MD3, and Schmitt inputs are within $V_{SS}Q$ to $V_{CC}Q$)
- Input rising and falling times: 1 ns



Notes: 1. C_L is the total value that includes the capacitance of measurement tools. Each pin is set as follows:
30pF: CKIO, $\overline{\text{RASU/L}}$, $\overline{\text{CASU/L}}$, $\overline{\text{CS0}}$, $\overline{\text{CS2}}$ to $\overline{\text{CS6B}}$, and $\overline{\text{BACK}}$
50pF: All other pins
2. I_{OL} and I_{OH} are shown in table 25.4.

Figure 25.56 Output Load Circuit

25.4 A/D Converter Characteristics

Table 25.16 lists the A/D converter characteristics.

Table 25.16 A/D Converter Characteristics

Conditions: $V_{CC}Q$ = 3.0 to 3.6 V, V_{CC} = 1.8 V ± 5%, AV_{CC} = 2.7 V to 3.6 V,
$V_{SS}Q = V_{SS} = AV_{SS}$ = 0 V, Ta = –40°C to +85°C

Item		Min.	Typ.	Max.	Unit
Resolution		10	10	10	bits
Conversion time		—	—	10.5	µs
Analog input capacitance		—	—	20*[1]	pF
Permissible signal-source impedance (single-source)		—	—	5*[1]	kΩ
Nonlinearity error		—	—	±3.0*[1]	LSB
Offset error		—	—	±2.5*[1]	LSB
Full-scale error		—	—	±2.5*[1]	LSB
Quantization error		—	—	±0.5*[1]	LSB
Absolute accuracy (Pϕ = 33 MHz)	CKS1 = 0, CKS0 = 0*[2]	—	—	±8.0	LSB
	Other than above	—	—	±4.0	

Notes: 1. Reference values
2. The fastest conversion time is equivalent to 4.4 us.

Appendix

A. Pin States

A.1 When Other Function is Selected

Table A.1 Pin States in Reset State, Power Down Mode, and Bus-Released States When Other Function is Selected

		Reset State		Power Down Mode		Bus-Released Reset
Type	Pin Name	Power-On	Manual	Software Standby	Sleep	
Clock	EXTAL (clock modes 2 and 6)	I	I	I	I	I
	EXTAL (clock mode 7)	Z*[1]	Z*[1]	Z*[1]	Z*[1]	Z*[1]
	EXTAL (clock modes 2 and 6)	O	O	O	O	O
	EXTAL (clock mode 7)	O*[1]	O*[1]	O*[1]	O*[1]	O*[1]
	EXTAL (clock modes 2 and 6)	O	O	O/Z*[2]	O	O/Z*[2]
	EXTAL (clock mode 7)	I	I	I	I	I
	CKIO2	O	O	O/Z*[2]	O	O/Z*[2]
System control	$\overline{\text{RESETP}}$ $\overline{\text{RESETM}}$	I	I	I	I	I
	$\overline{\text{BREQ}}$	Z+	I+	Z+	I+	I+
	$\overline{\text{BACK}}$	Z+	O	Z+	O	L
	MD[3,2,0]	I	I*[5]	I*[5]	I*[5]	I*[5]
	STATUS[1:0]	Z+	O	O	O	O
Interrupt	$\overline{\text{IRQ}}$[7:0]	Z+	I+	I+	I+	I+
	$\overline{\text{NMI}}$	I	I	I	I	I
Address bus	A[25:19], A0	Z+	O	O/Z+*[3]	O	Z+*[6]
	A[18:1]	O	O	O/Z*[3]	O	Z

Type	Pin Name	Reset State		Power Down Mode		Bus-Released Reset
		Power-On	Manual	Software Standby	Sleep	
Data bus	D[15:0]	Z	I	Z	I	Z
	D[31:16]	Z+	I+	Z+	I+	Z+*6
Bus control	$\overline{CS0}$	H	O	Z/H*3	O	Z
	$\overline{CS6}$[A,B] $\overline{CS5}$[A,B] $\overline{CS}$[2:4]	Z+	O	Z+/H*3	O	Z+
	$\overline{BS}$	H	O	Z/H*3	O	Z
	$\overline{CAS}$[U,L] $\overline{RAS}$[U,L]	Z+	O	O/Z+*2	O	O/Z+*2*6
	$\overline{WE0}$/DQMLL $\overline{WE1}$/DQMLU $\overline{WE2}$/DQMUL $\overline{WE3}$/DQMUU/$\overline{AH}$	H	O	Z/H*3	O	Z
	RD/$\overline{WR}$ $\overline{RD}$	H	O	Z/H*3	O	Z
	CKE	Z+	O	O/Z+*2	O	O/Z+*2*6
	$\overline{WAIT}$	Z	I++	Z	I++	Z
	$\overline{FRAME}$	Z+	O	Z+/H*3	O	Z+
DMAC	$\overline{DREQ}$[1:0]	Z+	I+	Z+	I+	I+
	$\overline{DACK}$[1:0]	Z+	O	O/Z+*4	O	O
	$\overline{TEND}$	Z+	O	O/Z+*4	O	O
MTU	TCLK[A:D]	Z+	I+	Z+	I+	I+
	TIOC0[A:D]	Z+	I+/O	Z+/K*4	I+/O	I+/O
	TIOC1[A,B]	Z+	I+/O	Z+/K*4	I+/O	I+/O
	TIOC2[A,B]	Z+	I+/O	Z+/K*4	I+/O	I+/O
	TIOC3[A:D]	Z+	I+/O	Z+/K*4	I+/O	I+/O
	TIOC4[A:D]	Z+	I+/O	Z+/K*4	I+/O	I+/O
POE	$\overline{POE}$[3:0]	Z+	I+	Z+	I+	I+

Type	Pin Name	Reset State		Power Down Mode		Bus-Released Reset
		Power-On	Manual	Software Standby	Sleep	
SCIF[2:0]	RxD[2:0]	Z+	I+	Z+	I+	I+
	TxD[2:0]	Z+	O/Z+	O/Z+*[4]	O/Z+	O/Z+
	SCK[2:0]	Z+	I+/O	K/Z+*[4]	I+/O	I+/O
	$\overline{\text{RTS}}$[2:0]	Z+	I+/O	K/Z+*[4]	I+/O	I+/O
	$\overline{\text{CTS}}$[2:0]	Z+	I+/O	K/Z+*[4]	I+/O	I+/O
AUD	$\overline{\text{AUDSYNC}}$	Z+	O	O	O	O
	AUDCK	O	O	O	O	O
	AUDATA[3:0]	Z+	O	O	O	O
H-UDI*[8]	$\overline{\text{ASEBRKAK}}$	O	O	O	O	O
	$\overline{\text{ASEMD0}}$	I	I*[5]	I*[5]	I*[5]	I*[5]
	TCK	I++	I++	I++	I++	I++
	TDI	I++	I++	I++	I++	I++
	TMS	I++	I++	I++	I++	I++
	$\overline{\text{TRST}}$	I++	I++	I++	I++	I++
	TDO	O/Z*[7]	O/Z*[7]	O/Z*[7]	O/Z*[7]	O/Z*[7]
USB	TXDMNS TXDPLS	Z+	O	O/Z+*[4]	O	O
	DMNS DPLS VBUS	Z+	I+	I+	I+	I+
	SUSPND TXENL	Z+	O	O/Z+*[4]	O	O
	XVDATA UCLK	Z+	I+	I+	I+	I+
	DP DM	Z	I/O	I	I/O	I/O
A/D	AN[7:0]	Z	I	Z	I	I
IIC2	SCL SDA	Z	I/O	Z	I/O	I/O

[Legend]

I: Input

I+: Input with weak keeper

I++: Input with pull-up MOS

O: Output

L: Low level output

H: High level output

Z: Hi-Z (The pin must not be open since the intermediate level at this pin caused a pass though current in the LSI.)

Z+: Hi-Z with weak keeper

Z++: Hi-Z with pull-up MOS

K: Input becomes Hi-Z, output retains state

Notes:
1. The EXTAL pin must be pulled up and the XTAL pin must be open.
2. Controlled by the HIZCNT bit in the common control register of the BSC.
3. Controlled by the HIZMEM bit in the common control register of the BSC.
4. Controlled by the HIZ bit in the standby control register.
5. The pin must not be open since the intermediate level at this pin causes the path though current in the LSI.
6. The data register of the I/O port can be written to.
7. Hi-Z when the TAP controller of the H-UDI is neither Shift-DR nor Shift-IR state.
8. When the H-UDI is not used, pins $\overline{\text{ASEMD0}}$, TCK, TDI, and TMS must be pulled up, the TDO and $\overline{\text{ASEBRKAK}}$ pins must be open, and the $\overline{\text{TRST}}$ pin must be connected to the $\overline{\text{RESETP}}$ pin or ground.
   For use of emulators, the board must be designed following instructions in the emulator manual.

A.2 When I/O Port is Selected

Table A.2 Pin States in Reset State, Power Down Mode, and Bus-Released States When I/O Port is Selected

	Reset State		Power Down Mode		
Pin Name	Power-On	Manual	Software Standby	Sleep	Bus-Released Reset
PTA[14:0]	Z+	I+/O	Z+/K*	I+/O	I+/O
PTB[8:0]	Z+	I+/O	Z+/K*	I+/O	I+/O
PTC[15,14,12:0]	Z+	I+/O	Z+/K*	I+/O	I+/O
PTC[13]	O	I+/O	Z+/K*	I+/O	I+/O
PTD[15:0]	Z+	I+/O	Z+/K*	I+/O	I+/O
PTE[15:0]	Z+	I+/O	Z+/K*	I+/O	I+/O
PTF[15:0]	Z+	I+/O	Z+/K*	I+/O	I+/O
PTG[13:11,8]	Z+	I+/O	Z+/K*	I+/O	I+/O
PTG[10:9]	Z	I/O	Z+/K*	I/O	I/O
PTG[7:0]	Z	I	Z	I	I
PTH[14:0]	Z+	I+/O	Z+/K*	I+/O	I+/O
PTJ[12:0]	Z+	I+/O	Z+/K*	I+/O	I+/O

[Legend]

I: Input

I+: Input with weak keeper

O: Output

Z: Hi-Z (The pin must not be open since the intermediate level at this pin causes a path though current in the LSI.)

Z+: Hi-Z with weak keeper

K: Input becomes Hi-Z, output retains state

Note: * Controlled by the HIZ bit in the standby control register.

B. Product Lineup

Product	Model	Package (Code)
SH7641	HD6417641BP100 (100 MHz version)	P-LFBGA1717-256*

Note: * For details of packages, please contact your nearest Renesas Technology sales representative.

C. Package Dimensions

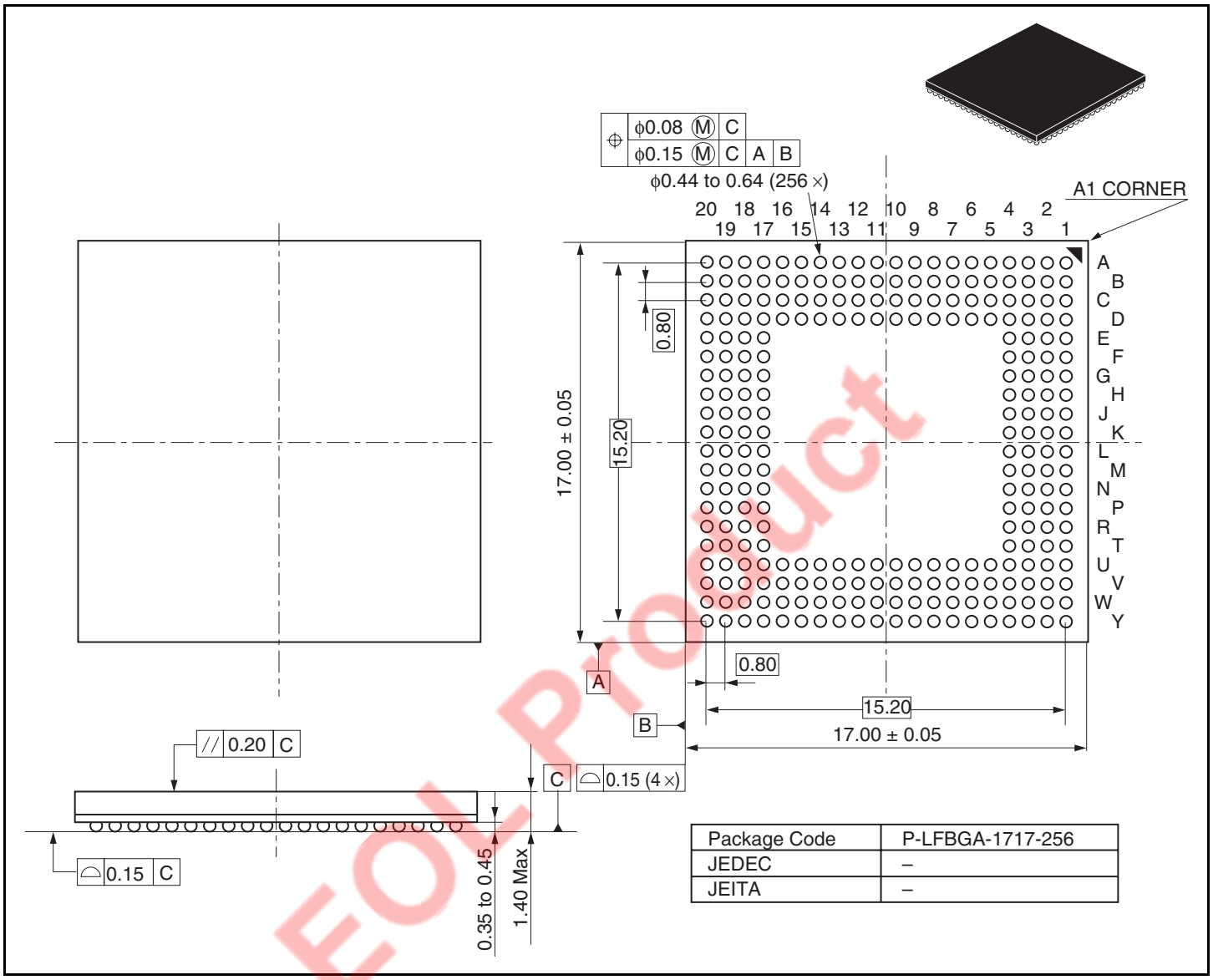


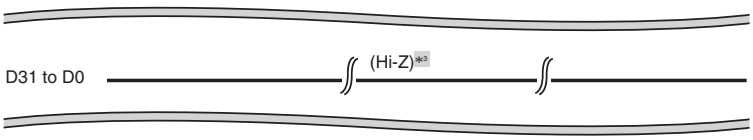
Figure C.1 Package Dimensions

Main Revisions and Additions in this Edition

Item	Page	Revisions (See Manual for Details)
General Precautions on Handling of Product	iv	5. added.

Section 9 Exception Handling
9.5 Note on Initializing this LSI
	217	;
		MOV.W #H'FF40,R10;
		MOV.L #H'A4FC0000,R8;
		MOV #H'10,R9;
		MOV.B R10,@R10;
		MOV.B R10,@R10;
		MOV.B R10,@R10;
		MOV.L R9,@R8;
		;
		MOV.L #H'FC000000,R1;
		MOV.W @R1,R0;
		;
		MOV #H'00,R9;
		MOV.B R10,@R10;
		MOV.B R10,@R10;
		MOV.B R10,@R10;

Section 13 Direct Memory Access Controller (DMAC) 13.4.8 Notes On DREQ Sampling When DACK is Divided in External Access	446	Added.

Item	Page	Revisions (See Manual for Details)
Section 25 Electrical Characteristics Figure 25.37 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles) Figure 25.38 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle) Figure 25.39 Synchronous DRAM Mode Register Write Timing (WTRP = 1 Cycle) Figure 25.41 Synchronous DRAM Self-Refreshing Timing in Low-Frequency Mode (WTRP = 2 Cycles)	949 to 951 and 953	 Note: 1. An address pin to be connected to pin A10 of SDRAM. 2. Waveform for $\overline{\text{DACKn}}$ when active low is selected. 3. Pins D31 to D16 with weak keeper are retained as weak keepers.

Index

E

F

G

H

I

L

M

N

O

P

Q

R

Register

S

T

U

V

W

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7641

Publication Date: Rev.1.00 Sep 19, 2003
Rev.4.00 Sep 14, 2005
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, United Kingdom
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.
Unit2607 Ruijing Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> 2-796-3115, Fax: <82> 2-796-2145

Renesas Technology Malaysia Sdn. Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

Colophon 3.0

SH7641
Hardware Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan