User's Manual

32

SH7205 Group

User's Manual: Hardware

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family / SH7200 Series

SH7205 R5S72050W200BG

Renesas Electronics
www.renesas.com

Rev.3.00 Mar 2014

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

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

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

(2012.4)

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

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

   — The input pins of CMOS products are generally in the high-impedance state. In operation with unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

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

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Configuration of This Manual

This manual comprises the following items:

1. General Precautions in the Handling of MPU/MCU Products
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
   - Product Type, Package Dimensions, etc.
10. Main Revisions for 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

This LSI is an RISC (Reduced Instruction Set Computer) microcomputer which includes a Renesas-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. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users.
Refer to the SH-2A, SH2A-FPU Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.
- In order to understand the details of the CPU's functions
  Read the SH-2A, SH2A-FPU Software Manual.
- In order to understand the details of a register when its name is known
  Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 32, List of Registers.

- Description of Numbers and Symbols

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

(1) Overall notation

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

(2) Register notation

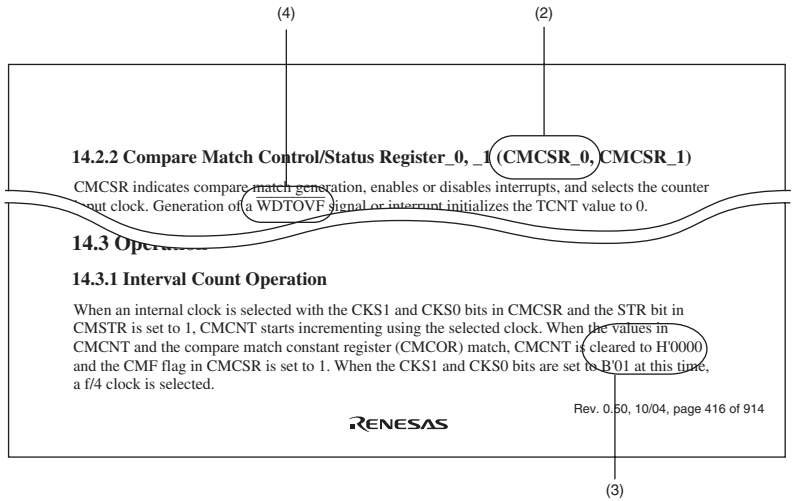
The style "register name"_"instance number" is used in cases where there is more than one instance of the same function or similar functions.
[Example] CMCSR_0: Indicates the CMCSR register for the compare-match timer of channel 0.

(3) Number notation

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

(4) Notation for active-low

An overbar on the name indicates that a signal or pin is active-low.
[Example] $\overline{\text{WDTOVF}}$

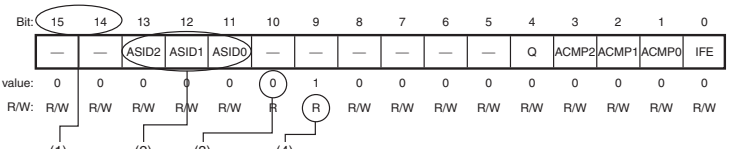


Note: The bit names and sentences in the above figure are examples and do not refer to specific data in this manual.

- Description of Registers

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

[Bit Chart]



[Table of Bits]

(1) (2) (3) (4) (5)

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

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

(1) Bit
Indicates the bit number or numbers.
In the case of a 32-bit register, the bits are arranged in order from 31 to 0. In the case of a 16-bit register, the bits are arranged in order from 15 to 0.

(2) Bit name
Indicates the name of the bit or bit field.
When the number of bits has to be clearly indicated in the field, appropriate notation is included (e.g., ASID[3:0]).
A reserved bit is indicated by "–".
Certain kinds of bits, such as those of timer counters, are not assigned bit names. In such cases, the entry under Bit Name is blank.

(3) Initial value
Indicates the value of each bit immediately after a power-on reset, i.e., the initial value.
0: The initial value is 0
1: The initial value is 1
–: The initial value is undefined

(4) R/W
For each bit and bit field, this entry indicates whether the bit or field is readable or writable, or both writing to and reading from the bit or field are impossible.
The notation is as follows:

R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable.
However, writing is only performed to flag clearing.
R: The bit or field is readable.
"R" is indicated for all reserved bits. When writing to the register, write the value under Initial Value in the bit chart to reserved bits or fields.
W: The bit or field is writable.

(5) Description
Describes the function of the bit or field and specifies the values for writing.

Contents

Section 1 Overview

1.1 SH7205 Features

This LSI is a single-chip RISC (reduced instruction set computer) microcontroller that includes two Renesas-original RISC CPUs as its cores, and the peripheral functions required to configure a system.

This LSI features a multi-processor architecture, that is, a dual-core architecture that includes two units of SH-2A CPU, which provides upward compatibility for SH-1, SH-2, and SH-2E CPUs at object code level. The SH-2A CPU has a RISC-type instruction set and uses a superscalar architecture and a Harvard architecture, which greatly improves instruction execution speed. In addition, the 32-bit internal-bus architecture that is independent from the direct memory access controller (DMAC) enhances data processing power. This CPU brings the user the ability to set up high-performance systems with strong functionality at less expense than was achievable with previous microcontrollers, and is even able to handle realtime control applications requiring high-speed characteristics.

This LSI includes a floating-point unit (FPU) and cache for each of the CPU cores (CPU0 and CPU1). In addition, this LSI has on-chip peripheral functions necessary for system configuration: 64-Kbyte (CPU0) and 32-Kbyte (CPU1) RAM for high-speed operation, 16-Kbyte RAM for data retention, an interrupt controller (INTC), a multi-function timer pulse unit 2 (MTU2), a compare match timer (CMT), a realtime clock (RTC), a serial communication interface with FIFO (SCIF), a synchronous serial communication unit (SSU), an I^2C bus interface 3 (IIC3), a serial sound interface with FIFO (SSIF), a controller area network (RCAN-TL1), an A/D converter (ADC), a D/A converter (DAC), an AND/NAND flash memory controller (FLCTL), a USB2.0 host/function module supporting two ports (USB), an AT attachment packet interface (ATAPI), a 2D engine (2DG), and I/O ports.

This LSI also provides an external memory access support function to enable direct connection to various memory devices or peripheral LSIs. These on-chip functions significantly reduce costs of designing and manufacturing application systems. Furthermore, I/O pins in this LSI have weak keeper circuits that prevent the pin voltage from entering an intermediate potential range. Therefore, no external circuits for fixing the input level are required, which reduces the number of parts considerably.

The features of this LSI are listed in table 1.1.

Table 1.1 SH7205 Features

Items	Specification
CPU	• Renesas original SuperH architecture • Compatible with SH-1, SH-2, and SH-2E at object code level • 32-bit internal data bus • Support of an abundant register-set — Sixteen 32-bit general registers — Four 32-bit control registers — Four 32-bit system registers — Register bank for high-speed response to interrupts • RISC-type instruction set (upward compatible with SH series) — Instruction length: 16-bit fixed-length basic instructions for improved code efficiency and 32-bit instructions for high performance and usability — Load/store architecture — Delayed branch instructions — Instruction set based on C language • Superscalar architecture including an FPU that allows execution of two instructions in parallel • Instruction execution time: Up to two instructions/cycle • Address space: 4 Gbytes • Internal multiplier • Five-stage pipeline • Harvard architecture

Items	Specification
Floating-point unit (FPU)	• Floating-point co-processor included • Supports single-precision (32-bit) and double-precision (64-bit) • Supports data types and exceptions that conform to IEEE754 standard • Two rounding modes: Round to the nearest and round to zero • Handling of denormalized numbers: Flush to zero • Floating-point registers — Sixteen 32-bit floating-point registers (single-precision × 16 words or double-precision × 8 words) — Two 32-bit floating-point system registers • Supports FMAC (multiplication and accumulation) instruction • Supports FDIV (division) and FSQRT (square root) instructions • Supports FLDI0/FLDI1 (load constant 0/1) instructions • Instruction execution time — Latency (FAMC/FADD/FSUB/FMUL): Three cycles (single-precision), eight cycles (double-precision) — Pitch (FAMC/FADD/FSUB/FMUL): One cycle (single-precision), six cycles (double-precision) Note: FMAC only supports single-precision • Five-stage pipeline
Exclusive control and memory sharing	• Supports exclusive control between two CPUs Semaphore control through registers provided between two CPUs Exclusive control by TAS.B instruction
Clock pulse generator (CPG)	• Clock mode: Input clock source can be selected from external input (EXTAL, CKIO, or USB_X1) or crystal resonator • Input clock can be multiplied by 16 (max.) by the internal PLL circuit • Four types of clocks generated: — CPU0 clock: Maximum 200 MHz — CPU1 clock: Maximum 200 MHz — Bus clock: Maximum 66 MHz (CPU0 bus, CPU1 bus and peripheral buses 1, 2, 3) — Peripheral clock: Maximum 33 MHz (peripheral bus 0)

Items	Specification
Interrupt controller (INTC)	• Inter-processor interrupts for synchronization control • Seventeen external interrupt pins (NMI, IRQ7 to IRQ0, and PINT7 to PINT0) • On-chip peripheral interrupts: Priority level set for each module • Programmable 16 priority levels • Register bank enabling fast register saving and restoring in interrupt processing
User break controller (UBC)	• Two break channels • Addresses, data values, access modes, and data size can be set as break conditions
Cache memory	• Instruction cache: 8 Kbytes × 2 cores (CPU0, CPU1) • Operand cache: 8 Kbytes × 2 cores (CPU0, CPU1) • 128-entry/way, 4-way set-associative, 16-byte block length configuration each for the instruction cache and operand cache • Write-back, write-through, and LRU replacement algorithm • Way lock function available (only for operand cache); ways 2 and 3 can be locked

Items	Specification
Bus state controller (BSC)	• Address space — Six areas of CS0 to CS5 and SDRAM space, each a maximum of 64 Mbytes — Data bus width selectable for each area (8, 16, or 32 bits) — Cycle wait function: Up to 31 wait cycles (up to 7 cycles for page access) • Normal space interface — Wait control: Assertion/negation timing of chip select signals, assertion/negation timing of read/write strobe signals, start/end timing of data outputs, and extension of chip select signals can be set — Write access mode: 1-write strobe/byte-write strobe mode — Page access mode: Page read/write supported (64-bit, 128-bit, and 256-bit page sizes are available) • SDRAM interface — Up to two areas can be allocated as SDRAM space (each area may be maximum of 64 Mbytes) — Refreshing: Auto refresh (with internal programmable refresh counter) or self refresh — Access timing: Row-column latency, column latency, row-active period can be set — Initialization sequencer: Power-down, deep power-down, and mode register setting functions

Items	Specification
Direct memory access controller (DMAC)	• 14 channels: Two-dimensional addressing supported on eight channels • Transfer requests Software trigger, on-chip peripheral I/O request, and requests from external pins • Number of bytes for transfer: 64 Mbytes at maximum • Transfer data size Single-data transfer: 8 bits, 16 bits, or 32 bits Single-operand transfer: 1,2, 4, 8, 16, 32, 64, 128 units of data Non-stop transfer: Until the byte counter reaches 0 • Transfer mode — Cycle stealing mode (dual-address transfer): 3 clock cycles at minimum per data transfer — Pipelined transfer (dual-address transfer): 1 clock cycle at minimum per data transfer • Address direction control Fixed, incremental, decremental, rotate, and two-dimensional addressing • Selectable DMA transfer condition Single-operand transfer, consecutive operand transfer, and non-stop transfer • Reloading: source address, destination address, byte counter

Items	Specification
Multi-function timer pulse unit 2 (MTU2)	• Maximum 16 lines of pulse inputs/outputs based on the five channels of 16-bit timers • 18 output-compare and input-capture registers • Input capture function • Pulse output modes Toggle, PWM, complementary PWM, and reset-synchronized PWM modes • Synchronization of multiple counters • Complementary PWM output mode — Non-overlapping waveforms output for 3-phase inverter control — Automatic dead time setting — 0% to 100% PWM duty cycle value specifiable — A/D converter start request delaying function — Interrupt skipping at crest or trough • Reset-synchronized PWM mode Three-phase PWM waveforms in positive and negative phases can be output with a desired duty value • Phase counting mode Two-phase encoder pulse counting available
Compare match timer (CMT)	• Four channels of 16-bit counters • Four types of clock can be selected (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) • DMA transfer request or interrupt request can be issued when a compare match occurs
Watchdog timer (WDT)	• One-channel watchdog timer × 2 cores • A counter overflow can reset the LSI
Realtime clock (RTC)	• Internal clock, calendar function, alarm function • Interrupts can be generated at intervals of 1/256 s by the 32.768-kHz on-chip crystal oscillator
Serial communication interface with FIFO (SCIF)	• Six channels • Clock synchronous or asynchronous mode selectable • Simultaneous transmission and reception (full-duplex communication) • Dedicated baud rate generator • Separate 16-byte FIFO registers for transmission and reception • Modem control function (in asynchronous mode)

Items	Specification
Synchronous serial communication unit (SSU)	• Master mode and slave mode selectable • Standard mode and bidirectional mode selectable • Transmit/receive data length can be selected from 8, 16, and 32 bits. • Simultaneous transmission and reception (full-duplex communication) • Consecutive serial communication • Two channels
I^2C bus interface 3 (IIC3)	• Four channels • Master mode and slave mode supported
Serial sound interface with FIFO (SSIF)	• Bidirectional serial transfer on six channels • Support of various serial audio formats • Support of master and slave functions • Programmable generation of word clock and bit clock • Multi-channel formats • Support of 8, 16, 18, 20, 22, 24, and 32-bit data formats • 8-stage FIFO for transmission and reception
Controller area network (RCAN-TL1)	• Two channels • TTCAN level 1 supported on both channels • BOSCH 2.0B active compatible • 31 transmit/receive mailboxes and one receive-only mailbox • Multiple RCAN-TL1 channels can be assigned to one bus to increase number of buffers with a granularity of 32 channels
A/D converter (ADC)	• 10-bit resolution • Eight input channels • A/D conversion initiated by the external trigger or timer trigger
D/A converter (DAC)	• 8-bit resolution • Two output channels
AND/NAND flash memory controller (FLCTL)	• Interface for direct connection with AND-/NAND-type flash memory • Read/write in sectors • Two types of transfer modes: Command access mode and sector access mode (512-byte data + 16-byte management code: with ECC) • Interrupt request and DMAC transfer request • Supports 5-byte address (up to 2 Gbits) of flash memory

Items	Specification
USB2.0 host/function module (USB)	• Conforms to the Universal Serial Bus Specification Revision 2.0 • USB bus supporting two ports 2-port host mode and 1-port function mode • 480-Mbps, 12-Mbps, and 1.5-Mbps transfer rates (host mode) • 480-Mbps and 12-Mbps transfer rates (function mode) • Includes 10-Kbyte RAM as communication buffers
AT attachment packet interface (ATAPI)	• Primary channel supported • Master/slave supported • PIO modes 0 to 4, multiword DMA modes 0 to 2, ultra DMA modes 0 to 2 supported • Includes dedicated DMAC • 32-byte double buffering • 3.3-V I/O interface
2D engine (2DG)	• Accelerative functions 2-input 1-output blit, fill, chroma keying, logical operation, color gradation handling, variable blending • Resizing function — Blit: bilinear/nearest-selectable independently for horizontal and vertical directions, transformation rate settable in the range from ×1/2 to ×2, pre-filtering for moiré prevention can be turned on/off — Image output: Cubic algorithm in horizontal direction only, transformation rate settable in the range from ×1/3 to ×1 • Motion picture input BT656 format (NTSC/PAL) input • Superimposition of motion picture Alpha-blends a graphic plane and motion picture and outputs in RGB666 at a constant rate • Blit input formats αRGB444 (16 bits), αRGB555 (16 bits), and α (4 bits) • Blit output formats αRGB444 (16 bits) and αRGB555 (16 bits) • Resolution of final output image WQVGA (480 × 234) or QVGA (320 × 240) • Video-out Built-in DAC (6-bit resolution, operating frequency of 5 MHz to 12.5 MHz)

Items	Specification
On-chip RAM	• CPU0: 64-Kbyte memory for high-speed operation (16 Kbytes × 4) • CPU1: 32-Kbyte memory for high-speed operation (16 Kbytes × 2) • 16-Kbyte memory for data retention
I/O ports	• 96 I/Os and 11 inputs • Input or output can be selected for each bit • Internal weak keeper circuit
Power-down modes	• Six power-down modes provided to reduce power consumption of this LSI — Dual-processor mode — Single-processor mode — Dual sleep mode — Software standby mode — Deep standby mode — Module standby mode
User debugging interface (H-UDI)	• E10A emulator support • JTAG-standard pin assignment
Power supply voltage	• Vcc: 1.1 to 1.3 V • PVcc: 3.0 to 3.6 V
Packages	• 272-pin BGA, 17-mm square, and 0.8-mm pitch JEITA Package Code: P-FBGA272-17×17-0.80 Renesas Code: PRBG0272GA-A

1.2 Product Lineup

Table 1.2 Product Lineup

Product Classification	Product Code	Package
SH7205	R5S72050W200BG	272-pin BGA

1.3 Block Diagram

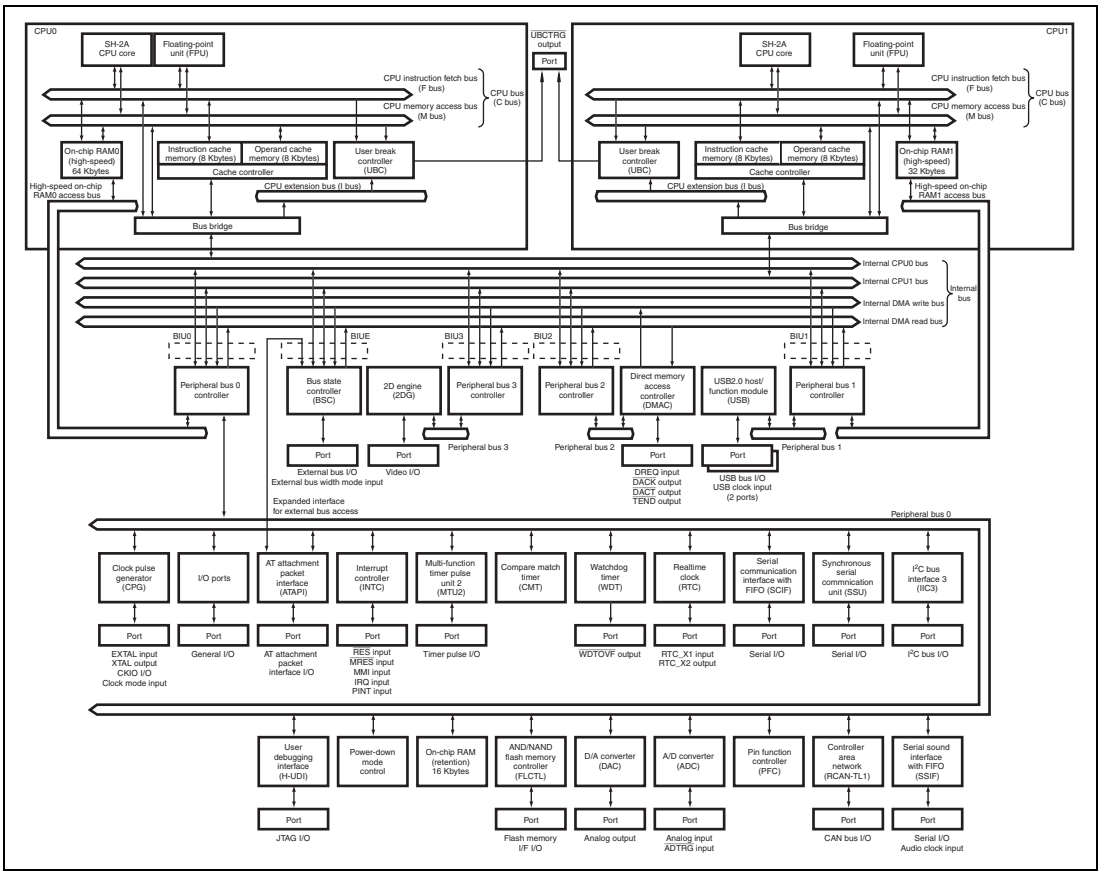


Figure 1.1 Block Diagram

1.4 Pin Assignment

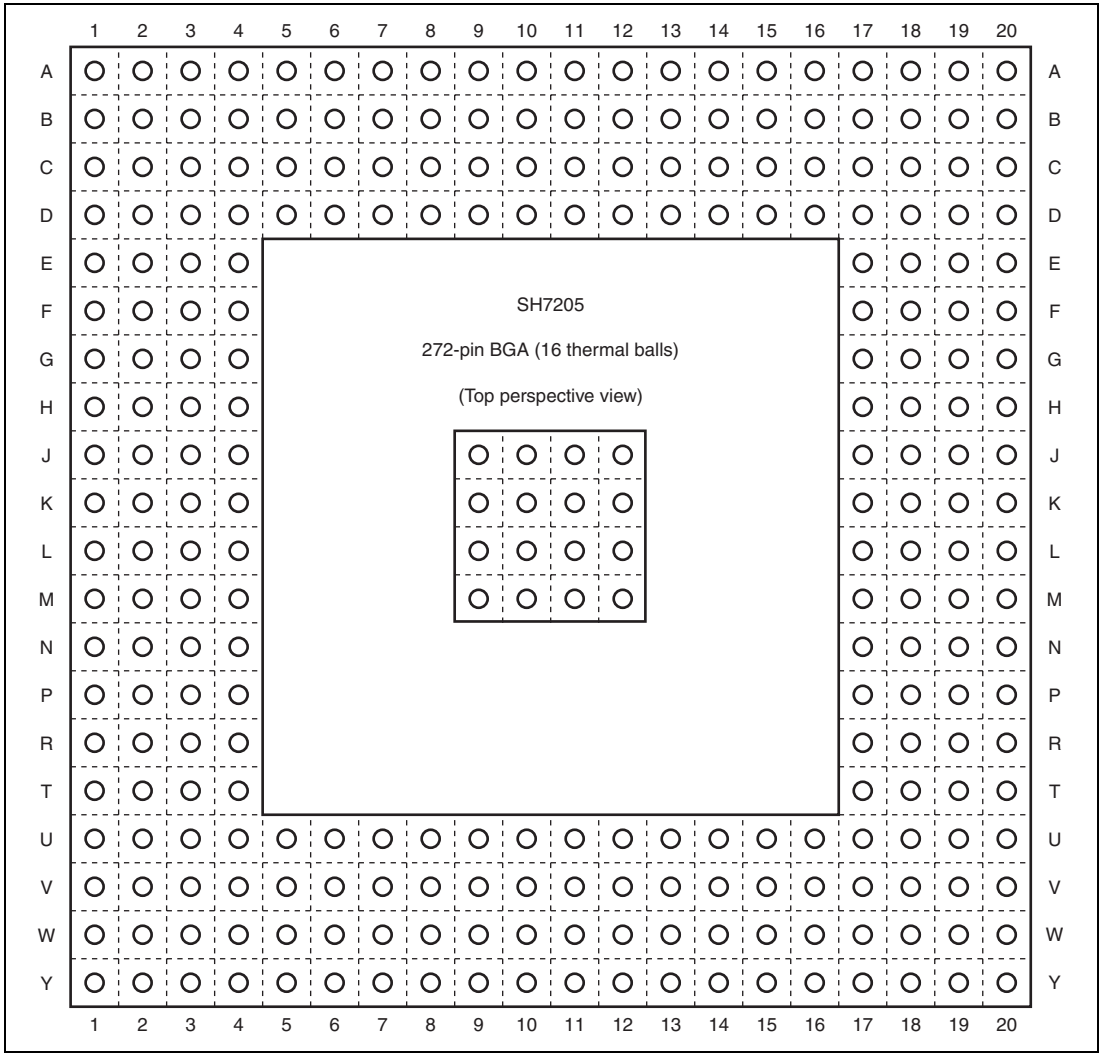


Figure 1.2 Pin Assignment

Table 1.3 Pin Numbers and Corresponding Pin Names

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
A1	Vss							
A2	CKIO	I/O	—	—	—	—	—	—
A3	PB9	I/O	$\overline{WE0}$/$\overline{BC0}$/DQM0	O	—	—	—	—
A4	D14	I/O	—	—	—	—	—	—
A5	D12	I/O	—	—	—	—	—	—
A6	D10	I/O	—	—	—	—	—	—
A7	D8	I/O	—	—	—	—	—	—
A8	PA14	I/O	D30	I/O	IDED14	I/O	—	—
A9	PA12	I/O	D28	I/O	IDED12	I/O	$\overline{DACK1}$	O
A10	PA10	I/O	D26	I/O	IDED10	I/O	$\overline{TEND0}$	O
A11	PA8	I/O	D24	I/O	IDED8	I/O	DREQ0	I(s)
A12	PA6	I/O	D22	I/O	IDED6	I/O	TCLKC	I(s)
A13	PA4	I/O	D20	I/O	IDED4	I/O	TCLKA	I(s)

Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Function 8 Symbol	Function 8 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
A1											
A2	—	—	—	—	—	—	—	—			Figure 1.3 (9)
A3	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
A4	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
A5	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
A6	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
A7	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
A8	TIOC4C	I(s)/O	PINT6	I(s)	—	—	—	—	Yes		Figure 1.3 (14)
A9	TIOC4A	I(s)/O	PINT4	I(s)	—	—	$\overline{DACT1}$	O	Yes		Figure 1.3 (14)
A10	TIOC3C	I(s)/O	PINT2	I(s)	—	—	—	—	Yes		Figure 1.3 (14)
A11	TIOC3A	I(s)/O	PINT0	I(s)	—	—	—	—	Yes		Figure 1.3 (14)
A12	TIOC2A	I(s)/O	IRQ6	I(s)	SSO1	I(s)/O	—	—	Yes		Figure 1.3 (14)
A13	TIOC1A	I(s)/O	IRQ4	I(s)	SSCK1	I(s)/O	—	—	Yes		Figure 1.3 (14)

Pin	Function 1		Function 2		Function 3		Function 4	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
A14	MD0	I(s)	—	—	—	—	—	—
A15	PC3	I(s)/O	IODACK#	O	TCLKD	I(s)	NAF1	I(s)/O
A16	PC7	I(s)/O	IDEIORDY	I	TIOC4D	I(s)/O	NAF5	I(s)/O
A17	PF0	I(s)/O	—	—	SCL3	I(s)/O(o)	—	—
A18	PF2	I/O	—	—	—	—	CTx0	O
A19	PF4	I/O	—	—	$\overline{\text{DACK3}}$	O	CTx1	O
A20	PVcc							
B1	PB11	I/O	$\overline{\text{WE2}}$/$\overline{\text{BC2}}$/DQM2	O	IDECS#0	O	$\overline{\text{FWE}}$	O
B2	Vss							
B3	PB10	I/O	$\overline{\text{WE1}}$/$\overline{\text{BC1}}$/DQM1	O	—	—	—	—
B4	D15	I/O	—	—	—	—	—	—
B5	D13	I/O	—	—	—	—	—	—
B6	D11	I/O	—	—	—	—	—	—
B7	D9	I/O	—	—	—	—	—	—

Pin	Function 5		Function 6		Function 7		Function 8		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
A14	—	—	—	—	—	—	—	—			Figure 1.3 (1)
A15	IRQ3	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (12)
A16	PINT3	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (14)
A17	—	—	—	—	—	—	—	—			Figure 1.3 (16)
A18	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
A19	CTx0&CTx1	O	$\overline{\text{DACT3}}$	O	—	—	—	—	Yes		Figure 1.3 (10)
A20											
B1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
B2											
B3	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
B4	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
B5	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
B6	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
B7	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)

Pin	Function 1		Function 2		Function 3		Function 4	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
B8	PA15	I/O	D31	I/O	IDED15	I/O	$\overline{\text{ADTRG}}$	I(s)
B9	PA13	I/O	D29	I/O	IDED13	I/O	$\overline{\text{TEND1}}$	O
B10	PA11	I/O	D27	I/O	IDED11	I/O	DREQ1	I(s)
B11	PA9	I/O	D25	I/O	IDED9	I/O	$\overline{\text{DACK0}}$	O
B12	PA7	I/O	D23	I/O	IDED7	I/O	TCLKD	I(s)
B13	PA5	I/O	D21	I/O	IDED5	I/O	TCLKB	I(s)
B14	PC0	I(s)/O	IDEA0	O	TCLKA	I(s)	FOE	O
B15	PC4	I(s)/O	IODREQ	I	TIOC4A	I(s)/O	NAF2	I(s)/O
B16	PC8	I(s)/O	IDEINT	I	DREQ1	I(s)	NAF6	I(s)/O
B17	PF1	I(s)/O	—	—	SDA3	I(s)/O(o)	CRx0	I(s)
B18	PF3	I/O	—	—	DREQ3	I(s)	CRx1	I(s)
B19	PVcc							
B20	Vcc							
C1	PB13	I/O	$\overline{\text{RAS}}$	O	—	—	—	—

Pin	Function 5		Function 6		Function 7		Function 8		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
B8	TIOC4D	I(s)/O	PINT7	I(s)	—	—	—	—	Yes		Figure 1.3 (14)
B9	TIOC4B	I(s)/O	PINT5	I(s)	—	—	—	—	Yes		Figure 1.3 (14)
B10	TIOC3D	I(s)/O	PINT3	I(s)	—	—	—	—	Yes		Figure 1.3 (14)
B11	TIOC3B	I(s)/O	PINT1	I(s)	—	—	$\overline{\text{DACT0}}$	O	Yes		Figure 1.3 (14)
B12	TIOC2B	I(s)/O	IRQ7	I(s)	$\overline{\text{SCS1}}$	I(s)/O	—	—	Yes		Figure 1.3 (14)
B13	TIOC1B	I(s)/O	IRQ5	I(s)	SSI1	I(s)/O	—	—	Yes		Figure 1.3 (14)
B14	IRQ0	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (12)
B15	PINT0	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (14)
B16	PINT4	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (14)
B17	IERxD	I(s)	—	—	—	—	—	—			Figure 1.3 (16)
B18	CRx0/CRx1	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (14)
B19											
B20											
C1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)

Pin	Function 1		Function 2		Function 3		Function 4	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
C2	PB12	I/O	$\overline{WE3}$/$\overline{BC3}$/DQM3	O	IDECS#1	O	FCDE	O
C3	Vss							
C4	Vcc							
C5	Vcc							
C6	D6	I/O	—	—	—	—	—	—
C7	D4	I/O	—	—	—	—	—	—
C8	D2	I/O	—	—	—	—	—	—
C9	D0	I/O	—	—	—	—	—	—
C10	PVcc							
C11	Vcc							
C12	PA2	I/O	D18	I/O	IDED2	I/O	—	—
C13	PA0	I/O	D16	I/O	IDED0	I/O	DREQ2	I(s)
C14	PC1	I(s)/O	IDEA1	O	TCLKB	I(s)	FSC	O
C15	PC5	I(s)/O	IDEIOWR#	O	TIOC4B	I(s)/O	NAF3	I(s)/O

Pin	Function 5		Function 6		Function 7		Function 8		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
C2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
C3											
C4											
C5											
C6	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
C7	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
C8	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
C9	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
C10											
C11											
C12	TIOC0C	I(s)/O	IRQ2	I(s)	SSO0	I(s)/O	—	—	Yes		Figure 1.3 (14)
C13	TIOC0A	I(s)/O	IRQ0	I(s)	SSCK0	I(s)/O	—	—	Yes		Figure 1.3 (14)
C14	IRQ1	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (12)
C15	PINT1	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (12)

Pin	Function 1		Function 2		Function 3		Function 4	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
C16	PC9	I(s)/O	IDERST#	O	$\overline{\text{DACK1}}$	O	NAF7	I(s)/O
C17	PVcc							
C18	PVcc							
C19	Vcc							
C20	RTC_X2	O	—	—	—	—	—	—
D1	$\overline{\text{CS0}}$	O	—	—	—	—	—	—
D2	PB14	I/O	$\overline{\text{CAS}}$	O	—	—	—	—
D3	PVcc							
D4	Vss							
D5	Vcc							
D6	D7	I/O	—	—	—	—	—	—
D7	D5	I/O	—	—	—	—	—	—
D8	D3	I/O	—	—	—	—	—	—
D9	D1	I/O	—	—	—	—	—	—

Pin	Function 5		Function 6		Function 7		Function 8		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
C16	PINT5	I(s)	$\overline{\text{DACT1}}$	O	—	—	—	—	Yes		Figure 1.3 (12)
C17											
C18											
C19											
C20	—	—	—	—	—	—	—	—			Figure 1.3 (18)
D1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
D2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
D3											
D4											
D5											
D6	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
D7	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
D8	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
D9	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)

Pin	Function 1		Function 2		Function 3		Function 4	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
D10	PVcc							
D11	Vcc							
D12	PA3	I/O	D19	I/O	IDED3	I/O	—	—
D13	PA1	I/O	D17	I/O	IDED1	I/O	$\overline{\text{DACK2}}$	O
D14	PC2	I(s)/O	IDEA2	O	TCLKC	I(s)	NAF0	I(s)/O
D15	PC6	I(s)/O	IDEIORD#	O	TIOC4C	I(s)/O	NAF4	I(s)/O
D16	PC10	I(s)/O	DIRECTION	O	$\overline{\text{TEND1}}$	O	$\overline{\text{FCE}}$	O
D17	PVcc							
D18	Vcc							
D19	PE0	I(s)/O	RxD0	I(s)	SSCK0	I(s)/O	—	—
D20	RTC_X1	I	—	—	—	—	—	—
E1	PB4	I/O	$\overline{\text{CS1}}$	O	—	—	—	—
E2	PB6	I/O	$\overline{\text{CS3}}$	O	RD_$\overline{\text{WR}}$/$\overline{\text{WE}}$	O	—	—
E3	PVcc							

Pin	Function 5		Function 6		Function 7		Function 8		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
D10											
D11											
D12	TIOC0D	I(s)/O	IRQ3	I(s)	$\overline{\text{SCS0}}$	I(s)/O	—	—	Yes		Figure 1.3 (14)
D13	TIOC0B	I(s)/O	IRQ1	I(s)	SSI0	I(s)/O	$\overline{\text{DACT2}}$	O	Yes		Figure 1.3 (14)
D14	IRQ2	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (12)
D15	PINT2	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (12)
D16	PINT6	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (12)
D17											
D18											
D19	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
D20	—	—	—	—	—	—	—	—			Figure 1.3 (18)
E1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
E2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
E3											

Pin	Function 1		Function 2		Function 3		Function 4	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
E4	PVcc							
E17	Vcc							
E18	Vcc							
E19	PE2	I(s)/O	$\overline{CTS0}$	I(s)/O	SSO0	I(s)/O	—	—
E20	PE1	I(s)/O	TxD0	O	SSI0	I(s)/O	—	—
F1	PB5	I/O	$\overline{CS2}$	O	—	—	—	—
F2	PB7	I/O	$\overline{CS4}$	O	$\overline{SDCS0}$	O	—	—
F3	PB15	I/O	CKE	O	—	—	—	—
F4	PB16	I/O	$\overline{SDWE}$	O	—	—	—	—
F17	PE6	I(s)/O	TxD1	O	SSO1	I(s)/O	—	—
F18	PE5	I(s)/O	RxD1	I(s)	SSI1	I(s)/O	—	—
F19	PE4	I(s)/O	SCK0	I(s)/O	SSCK1	I(s)/O	—	—
F20	PE3	I(s)/O	$\overline{RTS0}$	I(s)/O	$\overline{SCS0}$	I(s)/O	—	—
G1	PB1	I/O	A1	O	—	—	—	—

Pin	Function 5		Function 6		Function 7		Function 8		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
E4											
E17											
E18											
E19	TIOC2A	I(s)/O	—	—	—	—	—	—	Yes		Figure 1.3 (12)
E20	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
F1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
F2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
F3	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
F4	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
F17	SSIWS5	I(s)/O	—	—	—	—	—	—	Yes		Figure 1.3 (12)
F18	SSISCK5	I(s)/O	—	—	—	—	—	—	Yes		Figure 1.3 (12)
F19	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
F20	TIOC2B	I(s)/O	—	—	—	—	—	—	Yes		Figure 1.3 (12)
G1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
G2	A2	O	—	—	—	—	—	—
G3	PB8	I(s)/O	$\overline{CS5}$	O	$\overline{SDCS1}$	O	$\overline{MRES}$	I(s)
G4	PB17	I/O	$\overline{WAIT}$	I	—	—	—	—
G17	PE10	I(s)	RxD3	I(s)	SCL1	I(s)/O(o)	—	—
G18	PE9	I(s)/O	TxD2	O	SDA0	I(s)/O(o)	—	—
G19	PE8	I(s)	RxD2	I(s)	SCL0	I(s)/O(o)	—	—
G20	PE7	I(s)/O	SCK1	I(s)/O	$\overline{SCS1}$	I(s)/O	—	—
H1	A3	O	—	—	—	—	—	—
H2	A4	O	—	—	—	—	—	—
H3	$\overline{RD}$	O	—	—	—	—	—	—
H4	PB0	I/O	A0	O	RD_$\overline{WR}$/$\overline{WE}$	O	—	—
H17	PE13	I(s)/O	TxD4	O	SDA2	I(s)/O(o)	—	—
H18	PE12	I(s)	RxD4	I(s)	SCL2	I(s)/O(o)	—	—
H19	PE11	I(s)/O	TxD3	O	SDA1	I(s)/O(o)	—	—

Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Function 8 Symbol	Function 8 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
G2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
G3	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
G4	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
G17	—	—	—	—	—	—	—	—			Figure 1.3 (15)
G18	—	—	—	—	—	—	—	—			Figure 1.3 (16)
G19	—	—	—	—	—	—	—	—			Figure 1.3 (15)
G20	SSIDATA5	I(s)/O	—	—	—	—	—	—	Yes		Figure 1.3 (12)
H1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
H2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
H3	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
H4	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
H17	—	—	—	—	—	—	—	—			Figure 1.3 (16)
H18	—	—	—	—	—	—	—	—			Figure 1.3 (15)
H19	—	—	—	—	—	—	—	—			Figure 1.3 (16)

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
H20	TDI	I	—	—	—	—	—	—
J1	A5	O	—	—	—	—	—	—
J2	A6	O	—	—	—	—	—	—
J3	A7	O	—	—	—	—	—	—
J4	A8	O	—	—	—	—	—	—
J9	Vss							
J10	Vss							
J11	Vss							
J12	Vss							
J17	$\overline{\text{RES}}$	I(s)	—	—	—	—	—	—
J18	NMI	I(s)	—	—	—	—	—	—
J19	TCK	I	—	—	—	—	—	—
J20	TDO	O	—	—	—	—	—	—
K1	A9	O	—	—	—	—	—	—

Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Function 8 Symbol	Function 8 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
H20	—	—	—	—	—	—	—	—		Yes	Figure 1.3 (3)
J1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
J2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
J3	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
J4	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
J9											
J10											
J11											
J12											
J17	—	—	—	—	—	—	—	—			Figure 1.3 (1)
J18	—	—	—	—	—	—	—	—			Figure 1.3 (1)
J19	—	—	—	—	—	—	—	—		Yes	Figure 1.3 (3)
J20	—	—	—	—	—	—	—	—			Figure 1.3 (7)
K1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)

Pin	Function 1		Function 2		Function 3		Function 4	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
K2	A10	O	—	—	—	—	—	—
K3	Vcc							
K4	Vcc							
K9	Vss							
K10	Vss							
K11	Vss							
K12	Vss							
K17	PVcc							
K18	PVcc							
K19	$\overline{\text{ASEMD}}$	I(s)	—	—	—	—	—	—
K20	$\overline{\text{TRST}}$	I(s)	—	—	—	—	—	—
L1	A11	O	—	—	—	—	—	—
L2	A12	O	—	—	—	—	—	—
L3	PVcc							

Pin	Function 5		Function 6		Function 7		Function 8		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
K2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
K3											
K4											
K9											
K10											
K11											
K12											
K17											
K18											
K19	—	—	—	—	—	—	—	—			Figure 1.3 (1)
K20	—	—	—	—	—	—	—	—		Yes	Figure 1.3 (2)
L1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
L2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
L3											

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
L4	PVcc							
L9	Vss							
L10	Vss							
L11	Vss							
L12	Vss							
L17	Vcc							
L18	Vcc							
L19	TMS	I	—	—	—	—	—	—
L20	$\overline{\text{ASEBRKAK/ASEBRK}}$	I(s)/O	—	—	—	—	—	—
M1	A13	O	—	—	—	—	—	—
M2	A14	O	—	—	—	—	—	—
M3	A15	O	—	—	—	—	—	—
M4	A16	O	—	—	—	—	—	—
M9	Vss							

Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Function 8 Symbol	Function 8 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
L4											
L9											
L10											
L11											
L12											
L17											
L18											
L19	—	—	—	—	—	—	—	—		Yes	Figure 1.3 (3)
L20	—	—	—	—	—	—	—	—		Yes	Figure 1.3 (13)
M1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
M2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
M3	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
M4	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
M9											

Pin	Function 1		Function 2		Function 3		Function 4	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
M10	Vss							
M11	Vss							
M12	Vss							
M17	PVcc							
M18	PVcc							
M19	Vss							
M20	DP1	I/O	—	—	—	—	—	—
N1	A17	O	—	—	—	—	—	—
N2	A18	O	—	—	—	—	—	—
N3	A19	O	—	—	—	—	—	—
N4	A20	O	—	—	—	—	—	—
N17	Vss							
N18	Vss							
N19	Vss							

Pin	Function 5		Function 6		Function 7		Function 8		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
M10											
M11											
M12											
M17											
M18											
M19											
M20	—	—	—	—	—	—	—	—			
N1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
N2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
N3	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
N4	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (8)
N17											
N18											
N19											

Pin	Function 1		Function 2		Function 3		Function 4	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
N20	DM1	I/O	—	—	—	—	—	—
P1	PB2	I(s)/O	A21	O	RxD2	I(s)	—	—
P2	PB3	I/O	A22	O	TxD2	O	—	—
P3	PD2	I(s)/O	$\overline{\text{TEND0}}$	O	A23	O	SCK2	I(s)/O
P4	PD1	I(s)/O	$\overline{\text{DACK0}}$	O	A24	O	$\overline{\text{DACT0}}$	O
P17	Vss							
P18	Vss							
P19	Vss							
P20	DP0	I/O	—	—	—	—	—	—
R1	PH0	I(s)/O	SSISCK0	I(s)/O	—	—	—	—
R2	PH1	I(s)/O	SSIWS0	I(s)/O	—	—	—	—
R3	PH2	I(s)/O	SSIDATA0	I(s)/O	—	—	—	—
R4	PD0	I(s)/O	DREQ0	I(s)	A25	O	$\overline{\text{ADTRG}}$	I(s)
R17	REFRIN	I	—	—	—	—	—	—

Pin	Function 5		Function 6		Function 7		Function 8		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
N20	—	—	—	—	—	—	—	—			
P1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
P2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (10)
P3	IRQ6	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (12)
P4	IRQ5	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (12)
P17											
P18											
P19											
P20	—	—	—	—	—	—	—	—			
R1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
R2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
R3	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
R4	IRQ4	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (12)
R17	—	—	—	—	—	—	—	—			

Pin	Function 1		Function 2		Function 3		Function 4	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
R18	USBAPVcc							
R19	VBUS	I	—	—	—	—	—	—
R20	DM0	I/O	—	—	—	—	—	—
T1	PH3	I(s)/O	SSISCK1	I(s)/O	DREQ2	I(s)	—	—
T2	PH4	I(s)/O	SSIWS1	I(s)/O	$\overline{\text{DACK2}}$	O	$\overline{\text{DACT2}}$	O
T3	Vcc							
T4	Vcc							
T17	PVcc							
T18	PVcc							
T19	USBAPVss							
T20	USBDVcc							
U1	EXTAL	I	—	—	—	—	—	—
U2	PB18	I/O	$\overline{\text{WDTOVF}}$	O	$\overline{\text{UBCTRG}}$	O	—	—
U3	Vcc							

Pin	Function 5		Function 6		Function 7		Function 8		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
R18											
R19	—	—	—	—	—	—	—	—			
R20	—	—	—	—	—	—	—	—			
T1	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
T2	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
T3											
T4											
T17											
T18											
T19											
T20											
U1	—	—	—	—	—	—	—	—			Figure 1.3 (17)
U2	—	—	—	—	—	—	—	—		Yes	Figure 1.3 (11)
U3											

Pin	Function 1		Function 2		Function 3		Function 4	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
U4	PVcc							
U5	PH7	I(s)/O	SSIWS2	I(s)/O	—	—	—	—
U6	PH10	I(s)/O	SSIWS3	I(s)/O	—	—	—	—
U7	PH14	I(s)/O	SSIDATA4	I(s)/O	SCK5	I(s)/O	—	—
U8	PJ3	I(s)/O	IRQ7	I(s)	TIOC0D	I(s)/O	—	—
U9	PJ7	I/O	VIDATA2	I	TIOC1A	I(s)/O	—	—
U10	Vcc							
U11	PVcc							
U12	PJ12	I/O	VIDATA7	I	$\overline{\text{SCS1}}$	I(s)/O	—	—
U13	PG3	I(s)	IRQ3	I(s)	AN3	I(a)	—	—
U14	2DGAPVss0							
U15	2DGAPVcc0							
U16	Vcc							
U17	Vss							

Pin	Function 5		Function 6		Function 7		Function 8		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
U4											
U5	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
U6	NAF1	I(s)/O	—	—	—	—	—	—	Yes		Figure 1.3 (12)
U7	NAF5	I(s)/O	—	—	—	—	—	—	Yes		Figure 1.3 (12)
U8	—	—	TxD3	O	—	—	—	—	Yes		Figure 1.3 (12)
U9	NAF6	I(s)/O	—	—	AUDATA0	O	—	—	Yes		Figure 1.3 (14)
U10											
U11											
U12	FRB	I(s)	—	—	—	—	—	—	Yes		Figure 1.3 (14)
U13	TCLKD	I(s)	—	—	—	—	—	—			Figure 1.3 (6)
U14											
U15											
U16											
U17											

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
U18	PVcc							
U19	USBAVcc							
U20	Vss							
V1	XTAL	O	—	—	—	—	—	—
V2	Vcc							
V3	PVcc							
V4	PVcc							
V5	PH6	I(s)/O	SSISCK2	I(s)/O	—	—	—	—
V6	PH9	I(s)/O	SSISCK3	I(s)/O	—	—	—	—
V7	PH13	I(s)/O	SSIWS4	I(s)/O	TxD5	O	—	—
V8	PJ2	I(s)/O	IRQ6	I(s)	TIOC0C	I(s)/O	—	—
V9	PJ6	I/O	VIDATA1	I	$\overline{\text{TEND3}}$	O	—	—
V10	Vcc							
V11	PVcc							

Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Function 8 Symbol	Function 8 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
U18											
U19											
U20											
V1	—	—	—	—	—	—	—	—			Figure 1.3 (17)
V2											
V3											
V4											
V5	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
V6	NAF0	I(s)/O	—	—	—	—	—	—	Yes		Figure 1.3 (12)
V7	NAF4	I(s)/O	—	—	—	—	—	—	Yes		Figure 1.3 (12)
V8	—	—	RxD3	I(s)	—	—	—	—	Yes		Figure 1.3 (12)
V9	FCE	O	—	—	AUDSYNC	O	—	—	Yes		Figure 1.3 (10)
V10											
V11											

Pin	Function 1		Function 2		Function 3		Function 4	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
V12	PG0	I(s)	IRQ0	I(s)	AN0	I(a)	FRB	I(s)
V13	PG2	I(s)	IRQ2	I(s)	AN2	I(a)	—	—
V14	2DGAPVss1							
V15	2DGAPVcc1							
V16	Vcc							
V17	Vcc							
V18	Vss							
V19	USBAVss							
V20	USB_X1	I	—	—	—	—	—	—
W1	Vcc							
W2	PVcc							
W3	PH15	I(s)/O	AUDIO_CLK	I(s)	—	—	—	—
W4	MD_CLK1	I(s)	—	—	—	—	—	—
W5	PH5	I(s)/O	SSIDATA1	I(s)/O	$\overline{\text{TEND2}}$	O	—	—

Pin	Function 5		Function 6		Function 7		Function 8		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
V12	TCLKA	I(s)	—	—	—	—	—	—			Figure 1.3 (6)
V13	TCLKC	I(s)	—	—	—	—	—	—			Figure 1.3 (6)
V14											
V15											
V16											
V17											
V18											
V19											
V20	—	—	—	—	—	—	—	—			Figure 1.3 (17)
W1											
W2											
W3	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
W4	—	—	—	—	—	—	—	—			Figure 1.3 (1)
W5	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)

Pin	Function 1		Function 2		Function 3		Function 4	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
W6	PH8	I(s)/O	SSIDATA2	I(s)/O	—	—	—	—
W7	PH12	I(s)/O	SSISCK4	I(s)/O	RxD5	I(s)	—	—
W8	PJ1	I(s)/O	IRQ5	I(s)	TIOC0B	I(s)/O	—	—
W9	PJ5	I/O	VIDATA0	I	$\overline{\text{DACK3}}$	O	$\overline{\text{DACT3}}$	O
W10	PJ9	I/O	VIDATA4	I	SSCK1	I(s)/O	—	—
W11	PJ11	I/O	VIDATA6	I	SSO1	I(s)/O	—	—
W12	PK0	I/O	CSYNC	O	—	—	—	—
W13	PG4	I	VICLKENB	I	AN4	I(a)	—	—
W14	PG6	I	VIVSYNC	I	AN6	I(a)	DA0	O(a)
W15	PG7	I	VIHSYNC	I	AN7	I(a)	DA1	O(a)
W16	AVref							
W17	REXT	I	—	—	—	—	—	—
W18	CBU	O	—	—	—	—	—	—
W19	Vss							

Pin	Function 5		Function 6		Function 7		Function 8		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
W6	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
W7	NAF3	I(s)/O	—	—	—	—	—	—	Yes		Figure 1.3 (12)
W8	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
W9	FSC	O	TxD4	O	AUDCK	O	—	—	Yes		Figure 1.3 (10)
W10	—	—	—	—	AUDATA2	O	—	—	Yes		Figure 1.3 (14)
W11	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (14)
W12	FWE	O	—	—	—	—	—	—	Yes		Figure 1.3 (10)
W13	—	—	—	—	—	—	—	—			Figure 1.3 (4)
W14	—	—	—	—	—	—	—	—			Figure 1.3 (5)
W15	—	—	—	—	—	—	—	—			Figure 1.3 (5)
W16											
W17	—	—	—	—	—	—	—	—			
W18	—	—	—	—	—	—	—	—			
W19											

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
W20	USB_X2	O	—	—	—	—	—	—
Y1	PVcc							
Y2	PLLVss							
Y3	PLLVcc							
Y4	MD_CLK0	I(s)	—	—	—	—	—	—
Y5	AUDIO_X1	I	—	—	—	—	—	—
Y6	AUDIO_X2	O	—	—	—	—	—	—
Y7	PH11	I(s)/O	SSIDATA3	I(s)/O	—	—	—	—
Y8	PJ0	I(s)/O	IRQ4	I(s)	TIOC0A	I(s)/O	—	—
Y9	PJ4	I/O	VICLK	I	DREQ3	I(s)	—	—
Y10	PJ8	I/O	VIDATA3	I	TIOC1B	I(s)/O	—	—
Y11	PJ10	I/O	VIDATA5	I	SSI1	I(s)/O	—	—
Y12	PK1	I/O	DCLKIN	I	—	—	—	—
Y13	PG1	I(s)	IRQ1	I(s)	AN1	I(a)	—	—

Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Function 8 Symbol	Function 8 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
W20	—	—	—	—	—	—	—	—			Figure 1.3 (17)
Y1											
Y2											
Y3											
Y4	—	—	—	—	—	—	—	—			Figure 1.3 (1)
Y5	—	—	—	—	—	—	—	—			Figure 1.3 (17)
Y6	—	—	—	—	—	—	—	—			Figure 1.3 (17)
Y7	NAF2	I(s)/O	—	—	—	—	—	—	Yes		Figure 1.3 (12)
Y8	—	—	—	—	—	—	—	—	Yes		Figure 1.3 (12)
Y9	FOE	O	RxD4	I(s)	—	—	—	—	Yes		Figure 1.3 (14)
Y10	NAF7	I(s)/O	—	—	AUDATA1	O	—	—	Yes		Figure 1.3 (14)
Y11	—	—	—	—	AUDATA3	O	—	—	Yes		Figure 1.3 (14)
Y12	FCDE	O	—	—	—	—	—	—	Yes		Figure 1.3 (10)
Y13	TCLKB	I(s)	—	—	—	—	—	—			Figure 1.3 (6)

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
Y14	PG5	I	—	—	AN5	I(a)	—	—
Y15	AVss							
Y16	AVcc							
Y17	B	O(a)	—	—	—	—	—	—
Y18	G	O(a)	—	—	—	—	—	—
Y19	R	O(a)	—	—	—	—	—	—
Y20	Vss							

Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Function 8 Symbol	Function 8 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
Y14	—	—	—	—	—	—	—	—			Figure 1.3 (4)
Y15											
Y16											
Y17	—	—	—	—	—	—	—	—			
Y18	—	—	—	—	—	—	—	—			
Y19	—	—	—	—	—	—	—	—			
Y20											

[Legend]
(s): Schmitt
(a): Analog
(o): Open drain

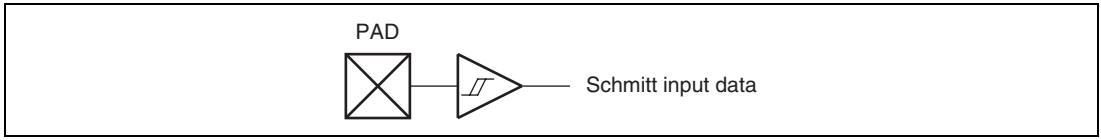


Figure 1.3 (1) Simplified Circuit Diagram (Schmitt Input Buffer)

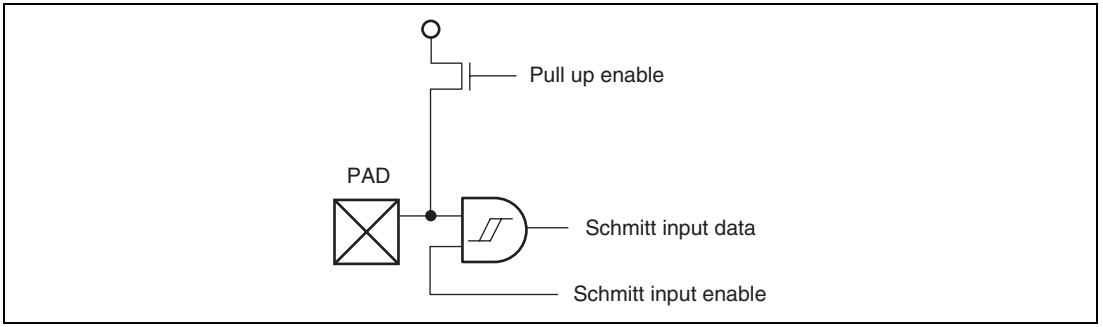


Figure 1.3 (2) Simplified Circuit Diagram (Schmitt AND Input Buffer with Pull-Up)

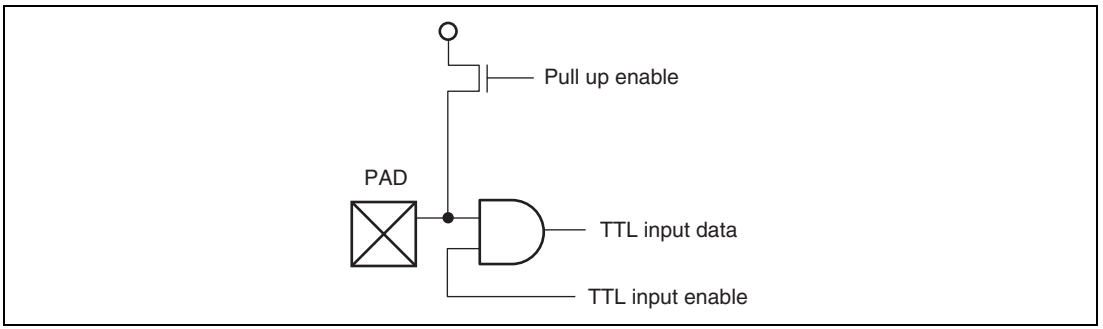


Figure 1.3 (3) Simplified Circuit Diagram (TTL AND Input Buffer with Pull-Up)

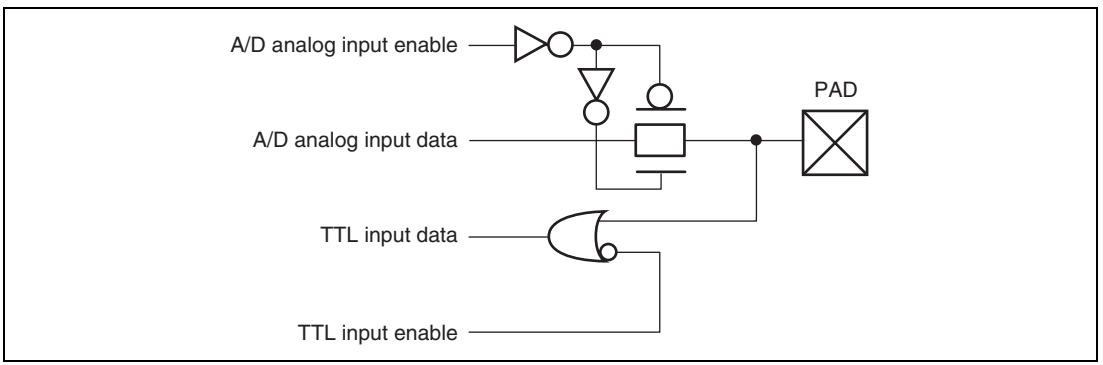


Figure 1.3 (4) Simplified Circuit Diagram (TTL OR Input and A/D Input Buffer)

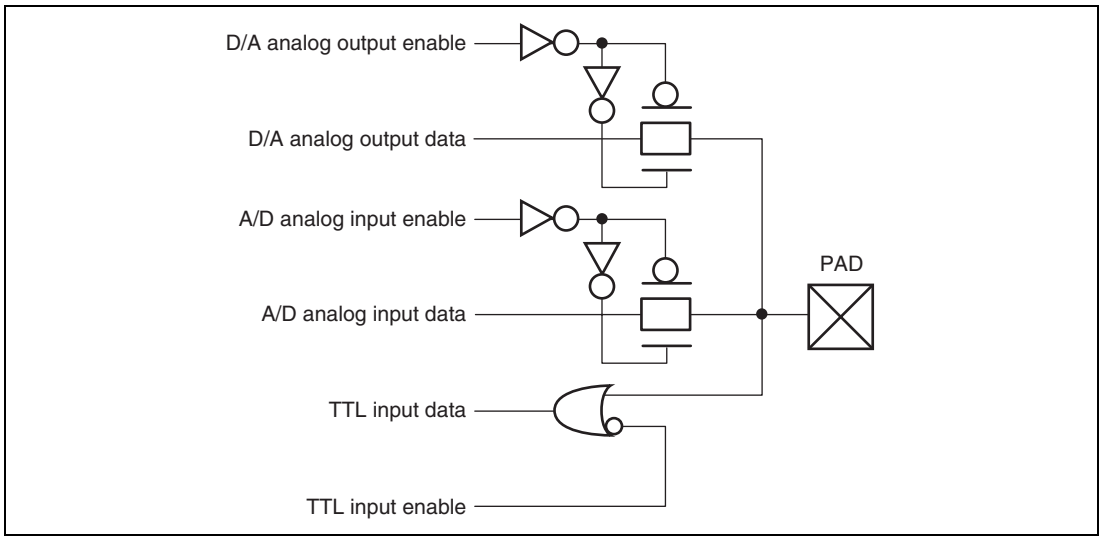


Figure 1.3 (5) Simplified Circuit Diagram (TTL OR Input and A/D Input and D/A Output Buffer)

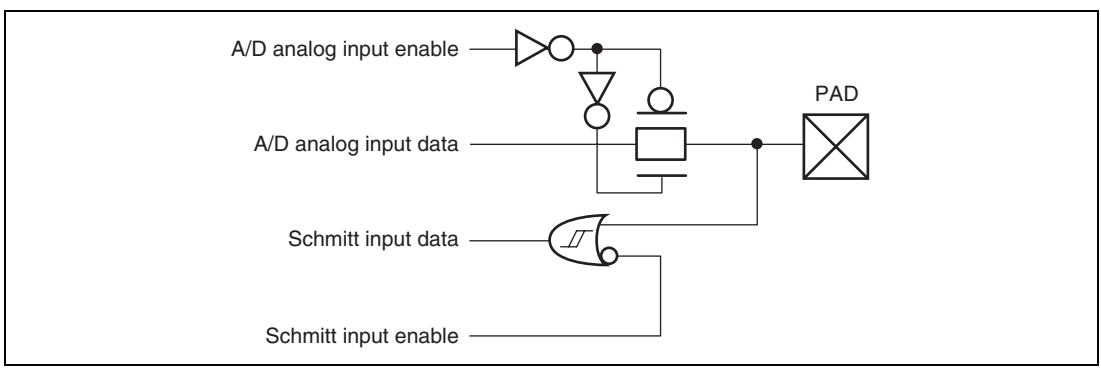


Figure 1.3 (6) Simplified Circuit Diagram (Schmitt OR Input and A/D Input Buffer)

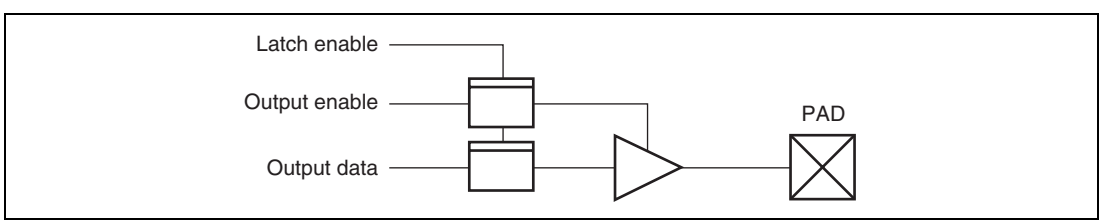


Figure 1.3 (7) Simplified Circuit Diagram (Output Buffer with Enable, with Latch)

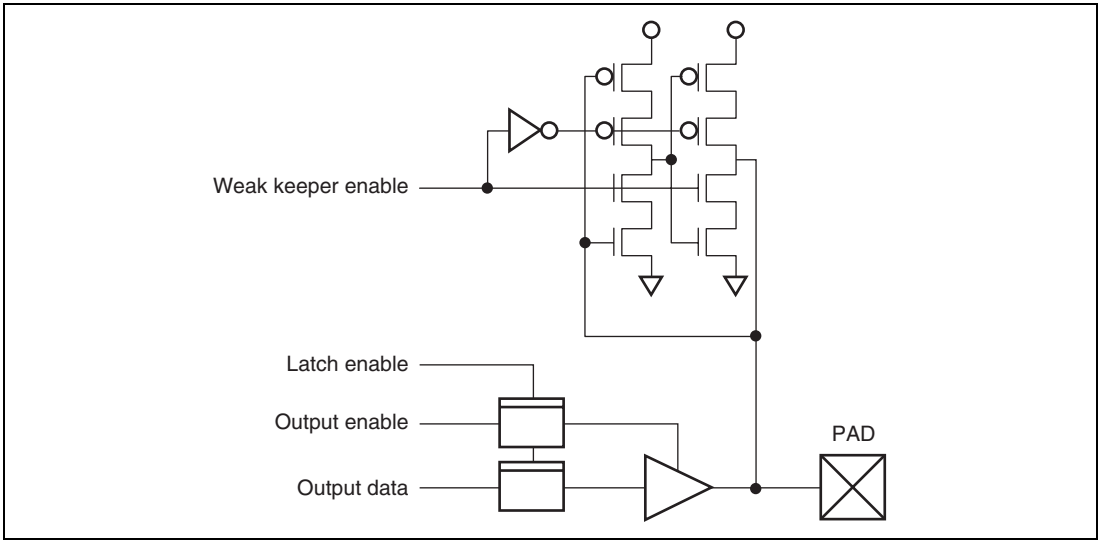


Figure 1.3 (8) Simplified Circuit Diagram (Output Buffer with Enable, with Latch and Weak Keeper)

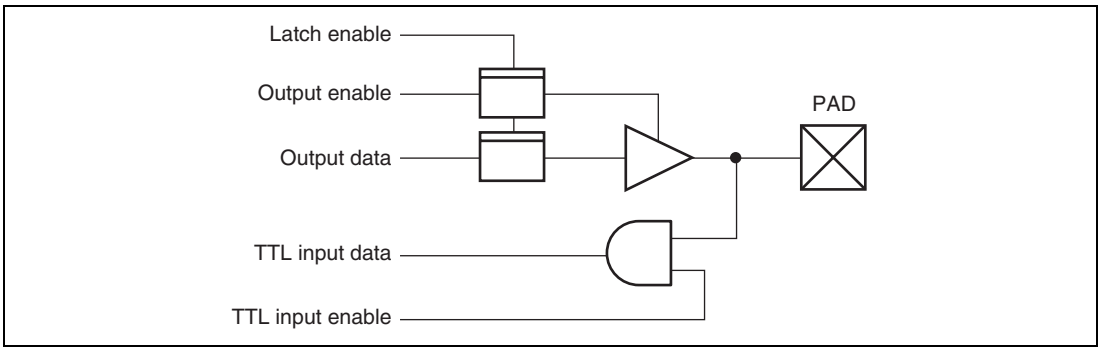


Figure 1.3 (9) Simplified Circuit Diagram (Bidirectional Buffer, TTL AND Input, with Latch)

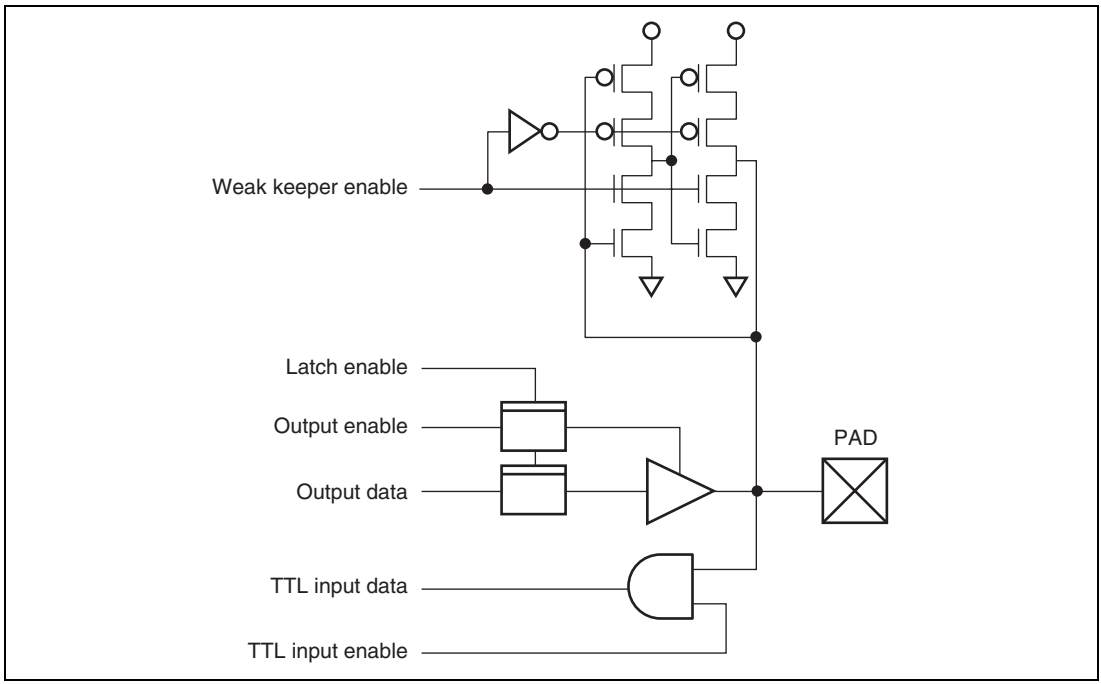


Figure 1.3 (10) Simplified Circuit Diagram (Bidirectional Buffer, TTL AND Input, with Latch and Weak Keeper)

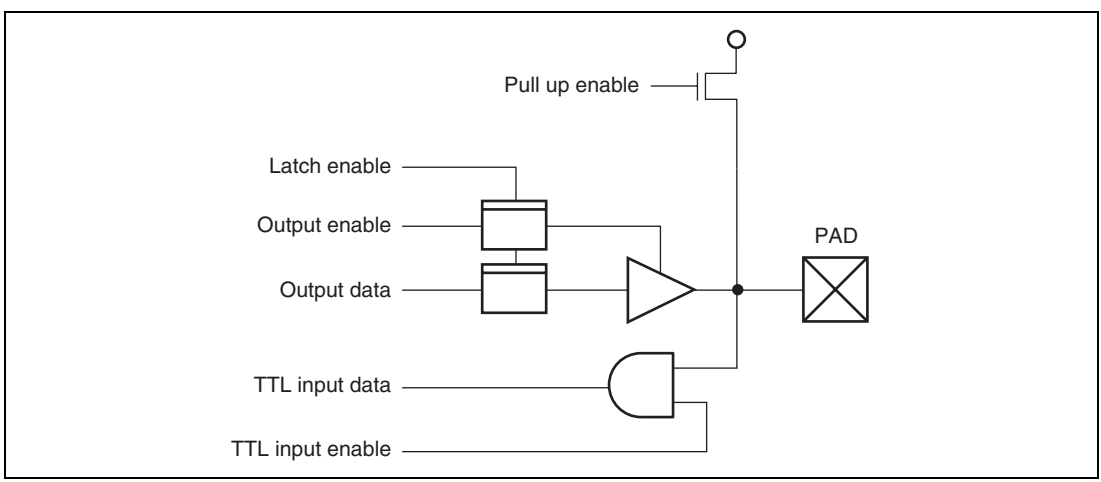


Figure 1.3 (11) Simplified Circuit Diagram (Bidirectional Buffer, TTL AND Input, with Latch and Pull-up)

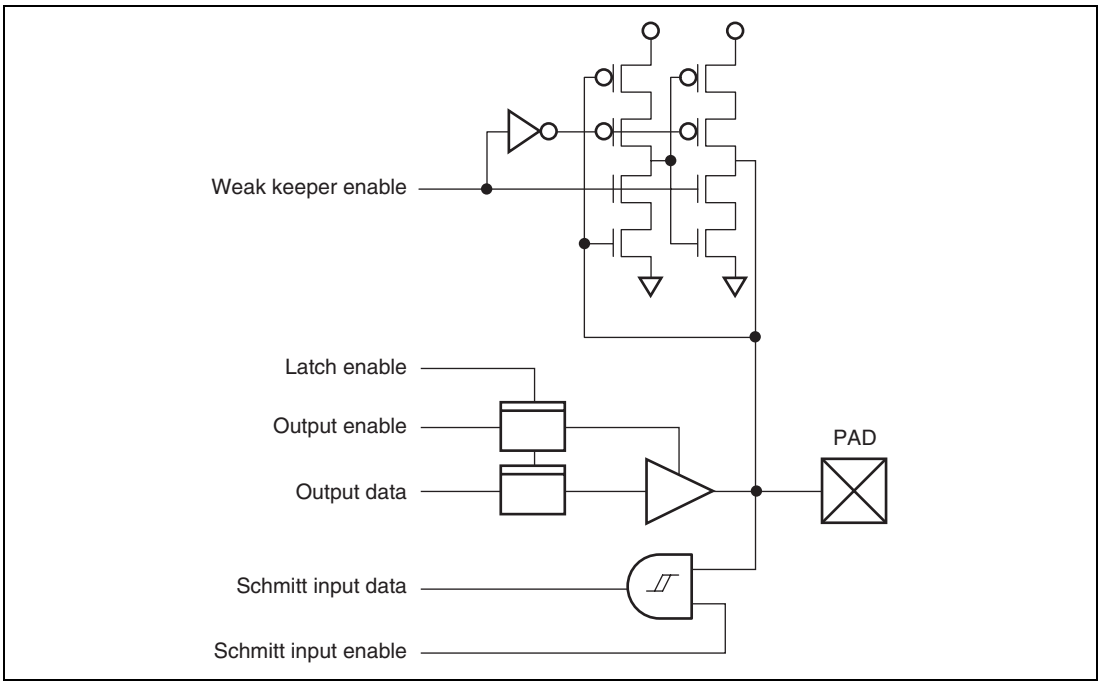


Figure 1.3 (12) Simplified Circuit Diagram (Bidirectional Buffer, Schmitt AND Input, with Latch and Weak Keeper)

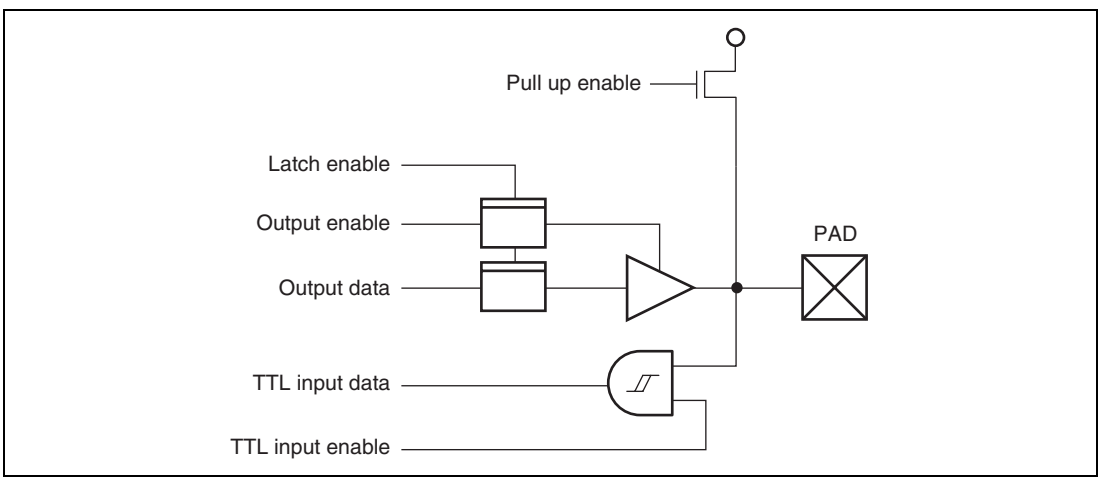


Figure 1.3 (13) Simplified Circuit Diagram (Bidirectional Buffer, Schmitt AND Input, with Latch and Pull-up)

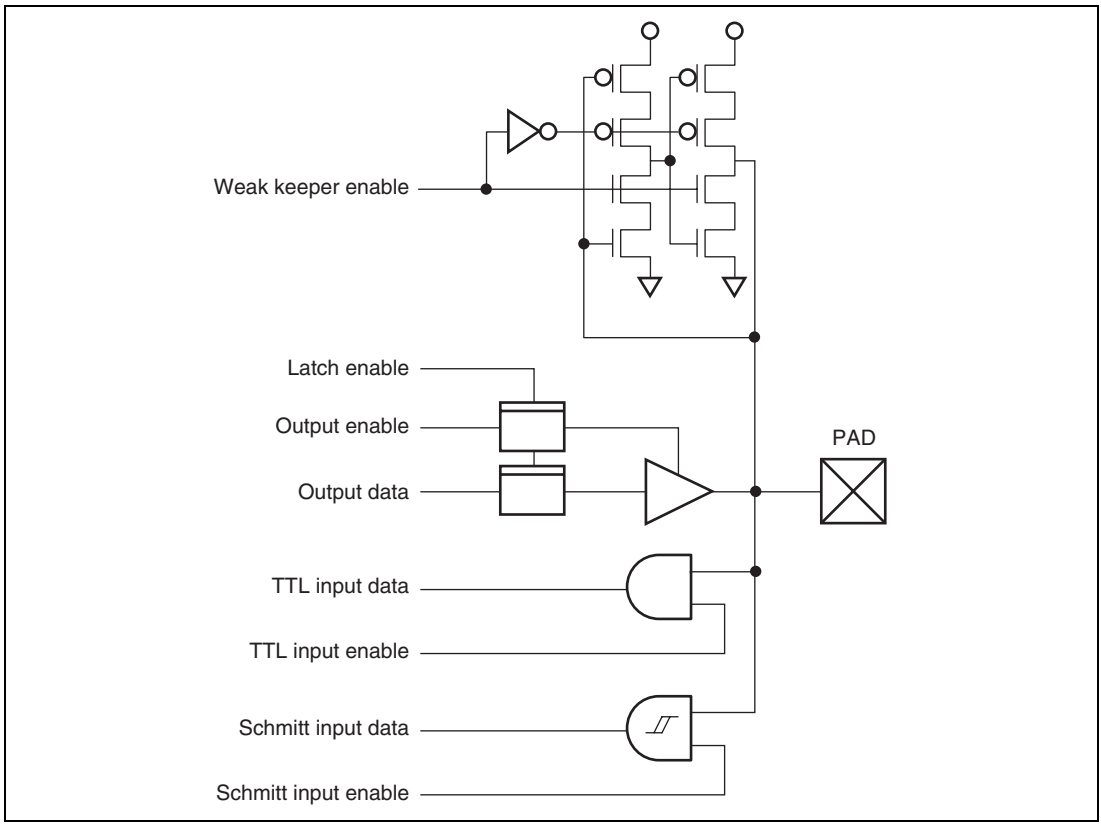


Figure 1.3 (14) Simplified Circuit Diagram
(Bidirectional Buffer, TTL AND Input, Schmitt AND Input, with Latch and Weak Keeper)

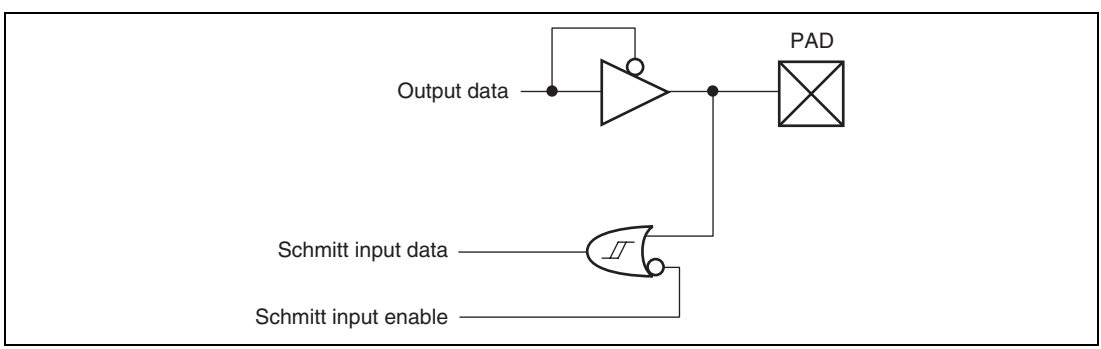


Figure 1.3 (15) Simplified Circuit Diagram
(Open Drain Output and Schmitt OR Input Buffer)

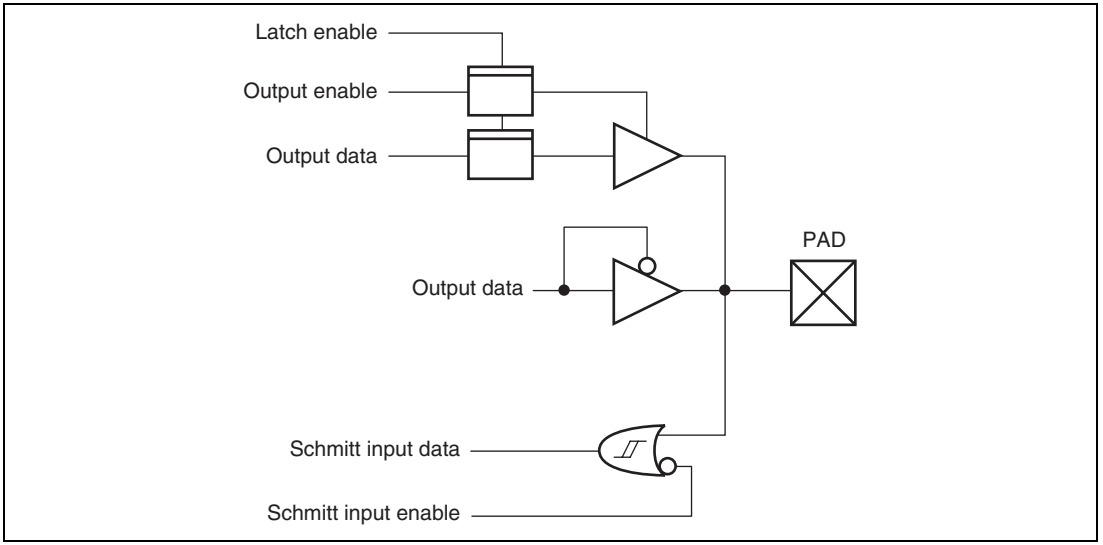


Figure 1.3 (16) Simplified Circuit Diagram (Open Drain/Normal Output and Schmitt OR Input Buffer, with Latch)

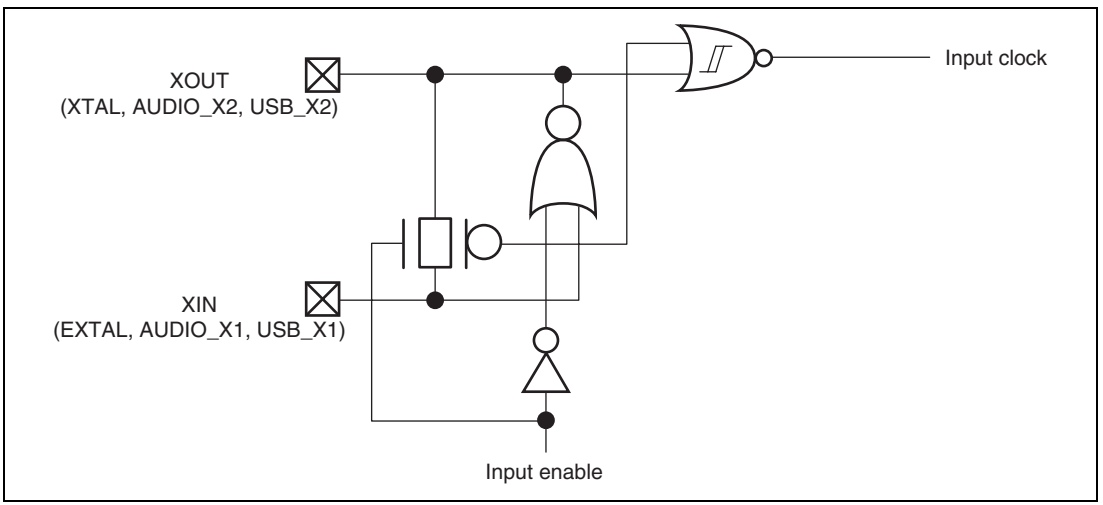


Figure 1.3 (17) Simplified Circuit Diagram (Oscillation Buffer 1)

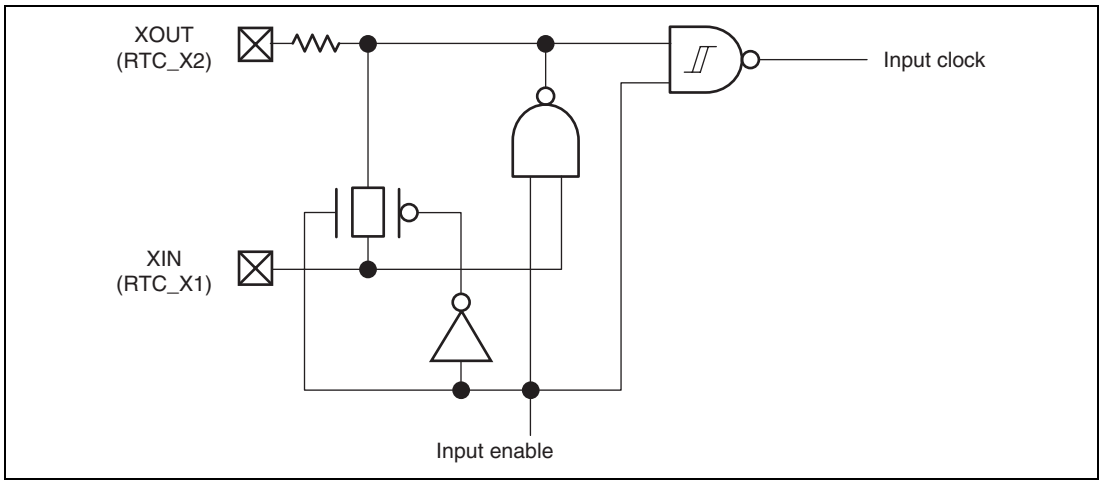


Figure 1.3 (18) Simplified Circuit Diagram (Oscillation Buffer 2)

1.5 Pin Functions

Table 1.4 Pin Functions

Classification	Symbol	I/O	Name	Function
Power supply	Vcc	I	Power supply	Power supply pins. All the Vcc pins must be connected to the system power supply. This LSI does not operate correctly if there is any pin left open.
	PVcc	I	Power supply for I/O circuits	Power supply for I/O pins. All the PVcc pins must be connected to the system power supply. This LSI does not operate correctly if there is any pin left open.
	Vss	I	Ground	Ground pins. All the Vss pins must be connected to the system power supply (0 V). This LSI does not operate correctly if there is any pin left open.
	PLLVcc	I	Power supply for PLL	Power supply for the on-chip PLL oscillator.
	PLLVss	I	Ground for PLL	Ground pin for the on-chip PLL oscillator.
Clock	EXTAL	I	External clock	Connected to a crystal resonator. An external clock signal may also be input to the EXTAL pin.
	XTAL	O	Crystal	Connected to a crystal resonator.
	CKIO	I/O	System clock I/O	Inputs an external clock or supplies the system clock to external devices.
Operating mode control	MD0	I	Mode set	Sets the operating mode. Do not change the signal level on this pin during operation.
	MD_CLK1, MD_CLK0	I	Clock mode set	These pins set the clock operating mode. Do not change the signal levels on these pins during operation.

Classification	Symbol	I/O	Name	Function
Operating mode control	$\overline{\text{ASEMD}}$	I	ASE mode	During $\overline{\text{RES}}$ pin assertion period, input a low level to operate this LSI in ASE mode. To operate it in product chip mode, apply a high level to this pin. Enables the E10A-USB emulator functions in ASE mode. Fix a high level mode when not using emulator functions.
System control	$\overline{\text{RES}}$	I	Power-on reset	This LSI enters the power-on reset state when this signal goes low.
	$\overline{\text{MRES}}$	I	Manual reset	This LSI enters the manual reset state when this signal goes low.
	$\overline{\text{WDTOVF}}$	O	Watchdog timer overflow	Outputs an overflow signal from the WDT.
Interrupts	NMI	I	Non-maskable interrupt	Non-maskable interrupt request pin. Fix it high when not in use.
	IRQ7 to IRQ0	I	Interrupt requests 7 to 0	Maskable interrupt request pins. Level-input or edge-input detection can be selected. When the edge-input detection is selected, the rising edge, falling edge, or both edges can also be selected.
	PINT7 to PINT0	I	Interrupt requests 7 to 0	Maskable interrupt request pins. Only level-input detection can be selected.
User break controller (UBC)	$\overline{\text{UBCTRG}}$	O	User break trigger output	Trigger output pin for UBC condition match.
Address bus	A25 to A0	O	Address bus	Outputs addresses.
Data bus	D31 to D0	I/O	Data bus	Bidirectional data bus.

Classification	Symbol	I/O	Name	Function
Bus control	$\overline{CS5}$ to $\overline{CS0}$	O	Chip select 5 to 0	Chip-select signals for external memory or devices.
	$\overline{RD}$	O	Read	Indicates that data is read from an external device.
	RD_$\overline{WR}$/$\overline{WE}$	O	Read/write	Read/write signal.
	$\overline{WAIT}$	I	Wait	Inserts wait cycles into bus cycles during access to the external space.
	$\overline{WE0}$/$\overline{BC0}$/ DQM0	O	Byte select	Indicates a write access to bits 7 to 0 of data in external memory or device.
	$\overline{WE1}$/$\overline{BC1}$/ DQM1	O	Byte select	Indicates a write access to bits 15 to 8 of data in external memory or device.
	$\overline{WE2}$/$\overline{BC2}$/ DQM2	O	Byte select	Indicates a write access to bits 23 to 16 of data in external memory or device.
	$\overline{WE3}$/$\overline{BC3}$/ DQM3	O	Byte select	Indicates a write access to bits 31 to 24 of data in external memory or device.
	$\overline{RAS}$	O	RAS	Connected to the $\overline{RAS}$ pin when SDRAM is connected.
	$\overline{CAS}$	O	CAS	Connected to the $\overline{CAS}$ pin when SDRAM is connected.
	$\overline{SDCS1}$, $\overline{SDCS0}$	O	Chip select	Connected to the $\overline{CS}$ pin when SDRAM is connected.
	$\overline{SDWE}$	O	SDRAM write enable	Connects to the WE pin of SDRAM if SDRAM is connected ($\overline{SDWE}$).
	CKE	O	CK enable	Connected to the CKE pin when SDRAM is connected.

Classification	Symbol	I/O	Name	Function
Direct memory access controller (DMAC)	DREQ3 to DREQ0	I	DMA-transfer request	Input pins to receive external requests for DMA transfer.
	$\overline{\text{DACK3}}$ to $\overline{\text{DACK0}}$	O	DMA-transfer request accept	Output pins for signals indicating acceptance of external requests from external devices.
	$\overline{\text{DACT3}}$ to $\overline{\text{DACT0}}$	O	DMA-transfer request active	Output pins for signals indicating that DMA is active in response to external requests from external devices.
	$\overline{\text{TEND3}}$ to $\overline{\text{TEND0}}$	O	DMA-transfer end output	Output pins for DMA transfer end.
Multi-function timer pulse unit 2 (MTU2)	TCLKA, TCLKB, TCLKC, TCLKD	I	MTU2 timer clock input	External clock input pins for the timer.
	TIOC0A, TIOC0B, TIOC0C, TIOC0D	I/O	MTU2 input capture/output compare (channel 0)	The TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
	TIOC1A, TIOC1B	I/O	MTU2 input capture/output compare (channel 1)	The TGRA_1 and TGRB_1 input capture input/output compare output/PWM output pins.
	TIOC2A, TIOC2B	I/O	MTU2 input capture/output compare (channel 2)	The TGRA_2 and TGRB_2 input capture input/output compare output/PWM output pins.
	TIOC3A, TIOC3B, TIOC3C, TIOC3D	I/O	MTU2 input capture/output compare (channel 3)	The TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.
	TIOC4A, TIOC4B, TIOC4C, TIOC4D	I/O	MTU2 input capture/output compare (channel 4)	The TGRA_4 and TGRB_4 input capture input/output compare output/PWM output pins.

Classification	Symbol	I/O	Name	Function
Realtime clock (RTC)	RTC_X1	I	Crystal oscillator for RTC/ external clock	Connected to 32.768-kHz crystal resonator. RTC_X1 can also be used to input an external clock.
	RTC_X2	O		
Serial communication interface with FIFO (SCIF)	TxD5 to TxD0	O	Transmit data	Data output pins.
	RxD5 to RxD0	I	Receive data	Data input pins.
	SCK5, SCK2 to SCK0	I/O	Serial clock	Clock input/output pins.
	$\overline{\text{RTS0}}$	O	Transmit request	Modem control pin.
	$\overline{\text{CTS0}}$	I	Transmit enable	Modem control pin.
Synchronous serial communication unit (SSU)	SSO1, SSO0	I/O	Data	Data I/O pin.
	SSI1, SSI0	I/O	Data	Data I/O pin.
	SSCK1, SSCK0	I/O	Clock	Clock I/O pin.
	$\overline{\text{SCS1}}$, $\overline{\text{SCS0}}$	I/O	Chip select	Chip select I/O pin.
I^2C bus interface 3 (IIC3)	SCL3 to SCL0	I/O	Serial clock pin	Serial clock I/O pin.
	SDA3 to SDA0	I/O	Serial data pin	Serial data I/O pin.
Serial sound interface with FIFO (SSIF)	SSIDATA5 to SSIDATA0	I/O	SSI data I/O	I/O pins for serial data.
	SSISCK5 to SSISCK0	I/O	SSI clock I/O	I/O pins for serial clocks.
	SSIWS5 to SSIWS0	I/O	SSI clock LR I/O	I/O pins for word selection.
	AUDIO_CLK	I	External clock for SSI audio	Input pin of external clock for SSI audio. A clock input to the divider is selected from an oscillation clock input on this pin or pins AUDIO_X1 and AUDIO_X2.
	AUDIO_X1	I	Crystal resonator/ external clock for SSI audio	Pins for connecting a crystal resonator for SSI audio. AUDIO_X1 can also be used to input an external clock. A clock input to the divider is selected from an oscillation clock input on these pins or the AUDIO_CLK pin.
	AUDIO_X2	O		

Classification	Symbol	I/O	Name	Function
Controller area network (RCAN-TL1)	CTx1, CTx0	O	CAN bus transmit data	Output pin for transmit data on the CAN bus.
	CRx1, CRx0	I	CAN bus receive data	Output pin for receive data on the CAN bus.
A/D converter (ADC)	AN7 to AN0	I	Analog input pins	Analog input pins.
	$\overline{\text{ADTRG}}$	I	A/D conversion trigger input	External trigger input pin for starting A/D conversion.
D/A converter (DAC)	DA1, DA0	O	Analog output pins	Analog output pins.
Common to analog circuits	AVcc	I	Analog power supply	Power supply pins for the A/D converter and D/A converter.
	AVss	I	Analog ground	Ground pins for the A/D converter and D/A converter.
	AVref	I	Analog reference voltage	Analog reference voltage pins for the A/D converter and D/A converter.
AND/NAND flash memory controller (FLCTL)	FOE	O	Flash memory output enable	Address latch enable: Asserted for address output and negated for data I/O. Output enable: Asserted for data input or status read.
	FSC	O	Flash memory serial clock	Read enable: Data is read on the falling edge of this signal. Serial clock: Data is inputs/output in synchronization with this signal.
	$\overline{\text{FCE}}$	O	Flash memory chip enable	Chip enable: Enables the flash memory connected to this LSI.
	FCDE	O	Flash memory command data enable	Command latch enable: Asserted at command output. Command data enable: Asserted at command output.

Classification	Symbol	I/O	Name	Function
AND/NAND flash memory controller (FLCTL)	FRB	I	Flash memory ready/busy	Ready/busy: High level indicates ready state and low level indicates busy state.
	$\overline{\text{FWE}}$	O	Flash memory write enable	Write enable: Flash memory latches commands, addresses, and data on the falling edge.
	NAF7 to NAF0	I/O	Flash memory data	Data I/O pins.
USB2.0 host/function module (USB)	DP1, DP0	I/O	USB D+ data	USB bus D+ data.
	DM1, DM0	I/O	USB D– data	USB bus D– data.
	VBUS	I	VBUS input	Connected to Vbus on USB bus.
	REFRIN	I	Reference input	Connected to USBAPVss via 5.6±1%-kΩ resistance.
	USB_X1	I	Crystal resonator/ external clock for USB	Connected to a crystal resonator for USB. An external clock signal may also be input to the USB_X1 pin.
	USB_X2	O		
	USBAPVcc	I	Power supply for transceiver analog pins	Power supply for pins.
	USBAPVss	I	Ground for transceiver analog pins	Ground for pins.
	USBAVcc	I	Power supply for transceiver analog core	Power supply for core.
	USBAVss	I	Ground for transceiver analog core	Ground for core.
	USBDVcc	I	Power supply for transceiver digital core	Power supply for core.

Classification	Symbol	I/O	Name	Function
AT attachment packet interface (ATAPI)	IDED15 to IDED0	I/O	Data bus	Bidirectional data bus
	IDEA2 to IDEA0	O	Address bus	Address bus
	IODACK#	O	DMA acknowledge	Primary channel DMA acknowledge
	IODREQ	O	DMA request	Primary channel DMA request
	IDECS#1, IDECS#0	O	Chip select	Primary channel chip select
	IDEIOWR#	O	Write	Primary channel disk write
	IDEIORD#	O	Read	Primary channel disk read
	IDEIORDY	I	Ready	Primary channel ready signal
	IDEINT	I	Interrupt request	Primary channel interrupt request
	IDERST#	O	Reset	Primary channel ATAPI device reset
	DIRECTION	O	Direction	External level shifter direction signal
2D engine (2DG)	R, G, B	O	RGB output	RGB analog output pins
	REXT	I	External reference input	External reference input pin for D/A converter
	CBU	O	External capacitance output	External capacitance output pin for D/A converter
	VIHSYNC	I	HSYNC signal	HSYNC signal input
	VIVSYNC	I	VSYNC signal	VSYNC signal input
	VICLK	I	Pixel clock	Pixel clock input
	VIDATA7 to VIDATA0	I	Image data	Image data input
	VICLKENB	I	Clock enable	Indicates that pixel data is valid
	CSYNC	O	CSYNC signal	Composite sync signal after graphic processing
	DCLKIN	I	Dot clock	Dot clock input
	2DGAPVcc0, 2DGAPVcc1	I	2DG D/A converter analog power supply	Power supply pin of 2DG D/A converter
	2DGAPVss0, 2DGAPVss1	I	2DG D/A converter analog ground	Ground pin of 2DG D/A converter

Classification	Symbol	I/O	Name	Function
I/O ports	PA15 to PA0, PB18 to PB0, PC10 to PC0, PD2 to PD0, PE13, PE11, PE9, PE7 to PE0, PF4 to PF0, PJ12 to PJ0, PH15 to PH0, PK1, PK0	I/O	General port	96 bits of general I/O port pins.
	PE12, PE10, PE8, PG7 to PG0	I	General port	11 bits of general input port pin.
User debugging interface (H-UDI)	TCK	I	Test clock	Test-clock input pin.
	TMS	I	Test mode select	Test-mode select signal input pin.
	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.
Emulator interface	AUDATA3 to AUDATA0	O	AUD data	Branch source or destination address output pins.
	AUDCK	O	AUD clock	Sync-clock output pin.
	$\overline{\text{AUDSYNC}}$	O	AUD sync signal	Data start-position acknowledge-signal output pin.
	$\overline{\text{ASEBRKAK}}$/ $\overline{\text{ASEBRK}}$	I/O	Break mode acknowledge/ break request	Indicates that the E10A-USB emulator has entered its break mode/E10A-USB emulator break input pin.

1.6 Bus Structure

The bus structure of this LSI largely consists of CPU buses, internal buses, and peripheral buses.

The bus master of the CPU bus is a CPU. Each of the CPUs (CPU0 and CPU1) is provided with a CPU bus for its own use, allowing both CPUs to run independently. A CPU bus actually consists of two buses: an instruction-fetch bus and a memory-access bus (Harvard architecture).

The circuit has multiple (four) internal buses. The master modules of the internal bus are the two CPUs and the DMAC. CPU0 and CPU1 are connected to the internal bus via the CPU bus and a bus bridge. The read port and write port of the DMAC act as master modules for the corresponding buses. That is, CPU0, CPU1, the DMA read port, and the DMA write port are individually connected to the corresponding internal buses. This allows each of the master modules to occupy its own internal bus without bus arbitration.

The slave modules of the internal buses are multiple peripheral buses (including the external bus and high-speed on-chip RAM access bus). On each internal bus, arbitration for bus mastership is performed between internal buses (master modules), after which access to the individual peripheral bus proceeds. In this LSI, internal modules called bus interface units (BIUs) perform this bus mastership arbitration. Since the BIUs perform arbitration per slave module, multiple accesses can proceed in parallel as long as access by each master module is to a different BIU. However, if more than one attempt at access to a given BIU is made at the same time, arbitration between the master modules is performed. The master module that failed to gain bus mastership is kept waiting until it succeeds, and thus the multiple accesses are executed one after another. The order of priority in bus-mastership arbitration is as follows: DMA write port > DMA read port > CPU. The priority order of CPU0 and CPU1 alternates in a round-robin manner.

The peripheral buses are used for the connections with the on-chip peripheral modules.

Section 2 CPU

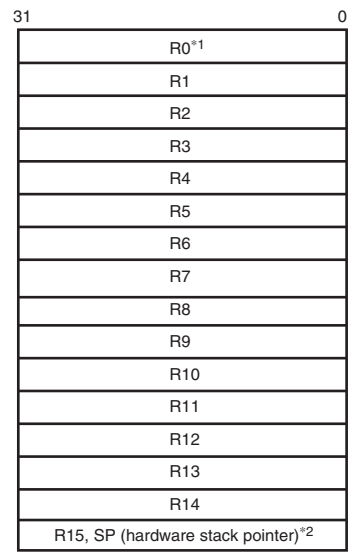
2.1 Register Configuration

The register set consists of sixteen 32-bit general registers, four 32-bit control registers, and four 32-bit system registers.

2.1.1 General Registers

Figure 2.1 shows the general registers.

The sixteen 32-bit general registers are numbered R0 to R15. General registers are used for data processing and address calculation. R0 is also used as an index register. Several instructions have R0 fixed as their only usable register. R15 is used as the hardware stack pointer (SP). Saving and restoring the status register (SR) and program counter (PC) in exception handling is accomplished by referencing the stack using R15.



Notes: 1. R0 functions as an index register in the indexed register indirect addressing mode and indexed GBR indirect addressing mode. In some instructions, R0 functions as a fixed source register or destination register.
2. R15 functions as a hardware stack pointer (SP) during exception processing.

Figure 2.1 General Registers

2.1.2 Control Registers

The control registers consist of four 32-bit registers: the status register (SR), the global base register (GBR), the vector base register (VBR), and the jump table base register (TBR).

The status register indicates instruction processing states.

The global base register functions as a base address for the GBR indirect addressing mode to transfer data to the registers of on-chip peripheral modules.

The vector base register functions as the base address of the exception handling vector area (including interrupts).

The jump table base register functions as the base address of the function table area.

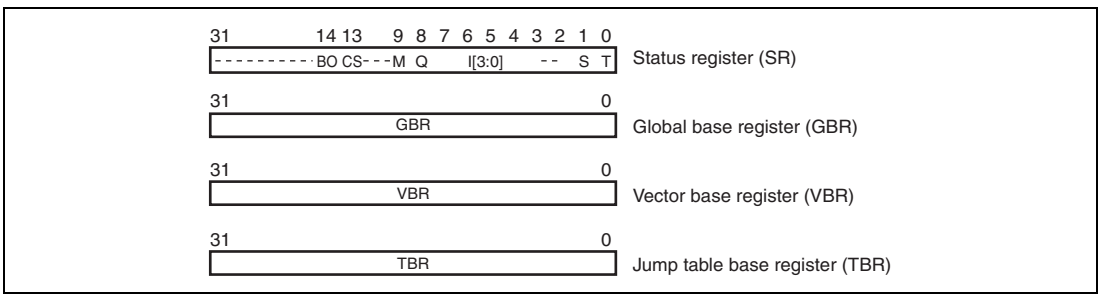


Figure 2.2 Control Registers

(1) Status Register (SR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	BO	CS	-	-	-	M	Q	I[3:0]				-	-	S	T
Initial value:	0	0	0	0	0	0	-	-	1	1	1	1	0	0	-	-
R/W:	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

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	BO	0	R/W	BO Bit Indicates that a register bank has overflowed.
13	CS	0	R/W	CS Bit Indicates that, in CLIP instruction execution, the value has exceeded the saturation upper-limit value or fallen below the saturation lower-limit value.
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	M	—	R/W	M Bit
8	Q	—	R/W	Q Bit Used by the DIV0S, DIV0U, and DIV1 instructions.
7 to 4	I[3:0]	1111	R/W	Interrupt Mask Level
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	S	—	R/W	S Bit Specifies a saturation operation for a MAC instruction.
0	T	—	R/W	T Bit True/false condition or carry/borrow bit

(2) Global Base Register (GBR)

GBR is referenced as the base address in a GBR-referencing MOV instruction.

(3) Vector Base Register (VBR)

VBR is referenced as the branch destination base address in the event of an exception or an interrupt.

(4) Jump Table Base Register (TBR)

TBR is referenced as the start address of a function table located in memory in a JSR/N@@(disp8,TBR) table-referencing subroutine call instruction.

2.1.3 System Registers

The system registers consist of four 32-bit registers: the high and low multiply and accumulate registers (MACH and MACL), the procedure register (PR), and the program counter (PC). MACH and MACL store the results of multiply or multiply and accumulate operations. PR stores the return address from a subroutine procedure. PC indicates the four bytes ahead of the current instruction being executed.

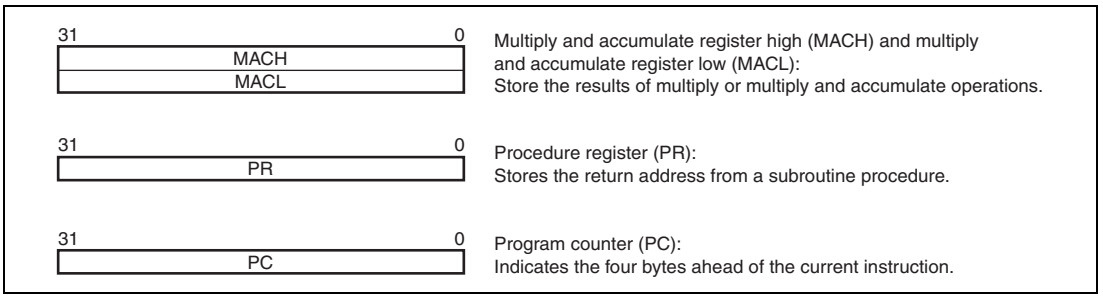


Figure 2.3 System Registers

(1) Multiply and Accumulate Register High (MACH) and Multiply and Accumulate Register Low (MACL)

MACH and MACL are used as the addition value in a MAC instruction, and store the result of a MAC or MUL instruction.

(2) Procedure Register (PR)

PR stores the return address of a subroutine call using a BSR, BSRF, or JSR instruction, and is referenced by a subroutine return instruction (RTS).

(3) Program Counter (PC)

PC indicates the four bytes ahead of the current instruction being executed.

2.1.4 Register Banks

For the nineteen 32-bit registers comprising general registers R0 to R14, control register GBR, and system registers MACH, MACL, and PR, high-speed register saving and restoration can be carried out using a register bank. The register contents are automatically saved in the bank after the CPU accepts an interrupt that uses a register bank. Restoration from the bank is executed by issuing a RESBANK instruction in an interrupt processing routine.

The LSI has two sets of 15 banks, one set for each CPU core. For details, see the SH-2A, SH2A-FPU Software Manual and section 7.8, Register Banks.

2.1.5 Initial Values of Registers

Table 2.1 lists the values of the registers after a reset.

Table 2.1 Initial Values of Registers

Classification	Register	Initial Value
General registers	R0 to R14	Undefined
	R15 (SP)	Value of the stack pointer in the vector address table
Control registers	SR	Bits I[3:0] are 1111 (H'F), BO and CS are 0, reserved bits are 0, and other bits are undefined
	GBR, TBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	Value of the program counter in the vector address table

2.2 Data Formats

2.2.1 Data Format in Registers

Register operands are always longwords (32 bits). If the size of memory operand is a byte (8 bits) or a word (16 bits), it is changed into a longword by expanding the sign-part when loaded into a register.

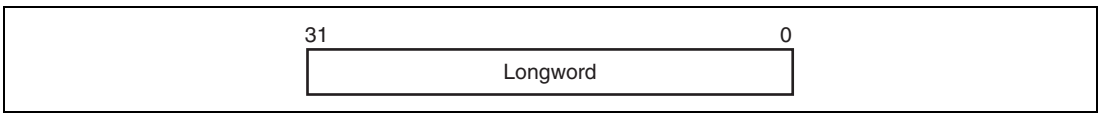


Figure 2.4 Data Format in Registers

2.2.2 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be accessed in 8-bit bytes, 16-bit words, or 32-bit longwords. A memory operand of fewer than 32 bits is stored in a register in sign-extended or zero-extended form.

A word operand should be accessed at a word boundary (an even address of multiple of two bytes: address 2n), and a longword operand at a longword boundary (an even address of multiple of four bytes: address 4n). Otherwise, an address error will occur. A byte operand can be accessed at any address.

Only big-endian byte order can be selected for the data format.

Data formats in memory are shown in figure 2.5.

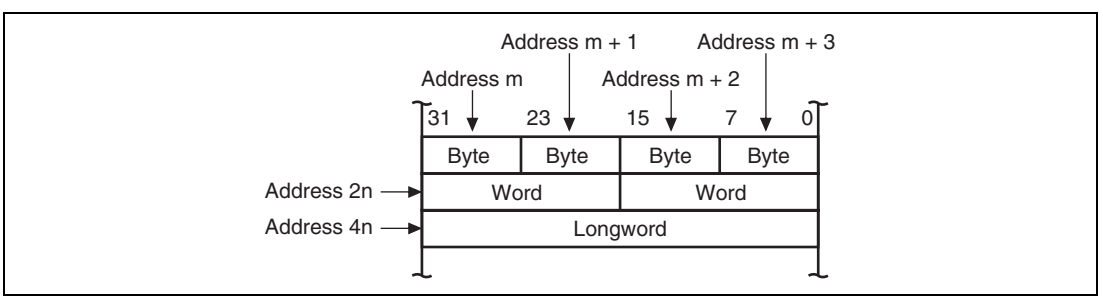


Figure 2.5 Data Formats in Memory

2.2.3 Immediate Data Format

Byte (8-bit) immediate data is located in an instruction code. Immediate data accessed by the MOV, ADD, and CMP/EQ instructions is sign-extended and handled in registers as longword data. Immediate data accessed by the TST, AND, OR, and XOR instructions is zero-extended and handled as longword data. Consequently, AND instructions with immediate data always clear the upper 24 bits of the destination register.

20-bit immediate data is located in the code of a MOVI20 or MOVI20S 32-bit transfer instruction. The MOVI20 instruction stores immediate data in the destination register in sign-extended form. The MOVI20S instruction shifts immediate data by eight bits in the upper direction, and stores it in the destination register in sign-extended form.

Word or longword immediate data is not located in the instruction code, but rather is stored in a memory table. The memory table is accessed by an immediate data transfer instruction (MOV) using the PC relative addressing mode with displacement.

See examples given in section 2.3.1 (10), Immediate Data.

2.3 Instruction Features

2.3.1 RISC-Type Instruction Set

Instructions are RISC type. This section details their functions.

(1) 16-Bit Fixed-Length Instructions

Basic instructions have a fixed length of 16 bits, improving program code efficiency.

(2) 32-Bit Fixed-Length Instructions

The SH-2A additionally features 32-bit fixed-length instructions, improving performance and ease of use.

(3) One Instruction per State

Each basic instruction can be executed in one cycle using the pipeline system.

(4) Data Length

Longword is the standard data length for all operations. Memory can be accessed in bytes, words, or longwords. Byte or word data in memory is sign-extended and handled as longword data. Immediate data is sign-extended for arithmetic operations or zero-extended for logic operations. It is also handled as longword data.

Table 2.2 Sign Extension of Word Data

SH2-A CPU		Description	Example of Other CPU
`MOV.W`	`@(disp,PC),R1`	Data is sign-extended to 32 bits, and R1 becomes H'00001234. It is next operated upon by an ADD instruction.	`ADD.W #H'1234,R0`
`ADD`	`R1,R0`		
	`.........`		
`.DATA.W`	`H'1234`		

Note: @(disp, PC) accesses the immediate data.

(5) Load-Store Architecture

Basic operations are executed between registers. For operations that involve memory access, data is loaded to the registers and executed (load-store architecture). Instructions such as AND that manipulate bits, however, are executed directly in memory.

(6) Delayed Branch Instructions

With the exception of some instructions, unconditional branch instructions, etc., are executed as delayed branch instructions. With a delayed branch instruction, the branch is taken after execution of the instruction immediately following the delayed branch instruction. This reduces disturbance of the pipeline control when a branch is taken.

In a delayed branch, the actual branch operation occurs after execution of the slot instruction. However, instruction execution such as register updating excluding the actual branch operation, is performed in the order of delayed branch instruction → delay slot instruction. 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 2.3 Delayed Branch Instructions

SH-2A CPU		Description	Example of Other CPU	
BRA	TRGET	Executes the ADD before branching to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

(7) Unconditional Branch Instructions with No Delay Slot

The SH-2A additionally features unconditional branch instructions in which a delay slot instruction is not executed. This eliminates unnecessary NOP instructions, and so reduces the code size.

(8) Multiply/Multiply-and-Accumulate Operations

16-bit × 16-bit → 32-bit multiply operations are executed in one to two cycles. 16-bit × 16-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to three cycles. 32-bit × 32-bit → 64-bit multiply and 32-bit × 32-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to four cycles.

(9) T Bit

The T bit in the status register (SR) changes according to the result of the comparison. Whether a conditional branch is taken or not taken depends upon the T bit condition (true/false). The number of instructions that change the T bit is kept to a minimum to improve the processing speed.

Table 2.4 T Bit

SH-2A CPU		Description	Example of Other CPU	
CMP/GE	R1,R0	T bit is set when R0 ≥ R1.	CMP.W	R1,R0
BT	TRGET0	The program branches to TRGET0 when R0 ≥ R1 and to TRGET1 when R0 < R1.	BGE	TRGET0
BF	TRGET1		BLT	TRGET1
ADD	#–1,R0	T bit is not changed by ADD.	SUB.W	#1,R0
CMP/EQ	#0,R0	T bit is set when R0 = 0.	BEQ	TRGET
BT	TRGET	The program branches if R0 = 0.		

(10) Immediate Data

Byte immediate data is located in an instruction code. Word or longword immediate data is not located in instruction codes but in a memory table. The memory table is accessed by an immediate data transfer instruction (MOV) using the PC relative addressing mode with displacement.

With the SH-2A, 17- to 28-bit immediate data can be located in an instruction code. However, for 21- to 28-bit immediate data, an OR instruction must be executed after the data is transferred to a register.

Table 2.5 Immediate Data Accessing

Classification	SH-2A CPU		Example of Other CPU	
8-bit immediate	MOV	#H'12,R0	MOV.B	#H'12,R0
16-bit immediate	MOVI20	#H'1234,R0	MOV.W	#H'1234,R0
20-bit immediate	MOVI20	#H'12345,R0	MOV.L	#H'12345,R0
28-bit immediate	MOVI20S	#H'12345,R0	MOV.L	#H'1234567,R0
	OR	#H'67,R0		
32-bit immediate	MOV.L	@(disp,PC),R0	MOV.L	#H'12345678,R0
		.....		
	.DATA.L	H'12345678		

Note: @(disp, PC) accesses the immediate data.

(11) Absolute Address

When data is accessed by an absolute address, the absolute address value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in register indirect addressing mode.

With the SH-2A, when data is referenced using an absolute address not exceeding 28 bits, it is also possible to transfer immediate data located in the instruction code to a register and to reference the data in register indirect addressing mode. However, when referencing data using an absolute address of 21 to 28 bits, an OR instruction must be used after the data is transferred to a register.

Table 2.6 Absolute Address Accessing

Classification	SH-2A CPU		Example of Other CPU
Up to 20 bits	MOVI20	#H'12345,R1	MOV.B @H'12345,R0
	MOV.B	@R1,R0	
21 to 28 bits	MOVI20S	#H'12345,R1	MOV.B @H'1234567,R0
	OR	#H'67,R1	
	MOV.B	@R1,R0	
29 bits or more	MOV.L	@(disp,PC),R1	MOV.B @H'12345678,R0
	MOV.B	@R1,R0	
		.....	
	.DATA.L	H'12345678	

(12) 16-Bit/32-Bit Displacement

When data is accessed by 16-bit or 32-bit displacement, the displacement value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in the indexed indirect register addressing mode.

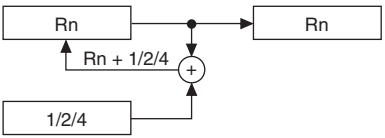
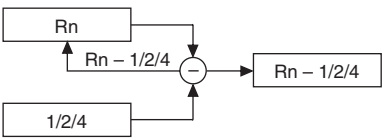
Table 2.7 Displacement Accessing

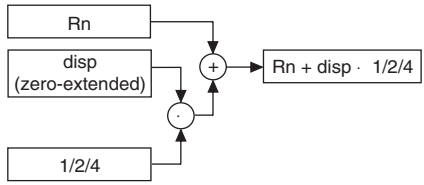
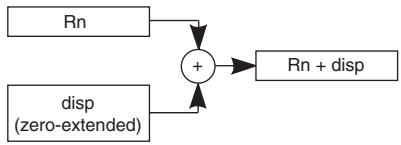
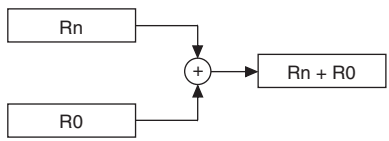
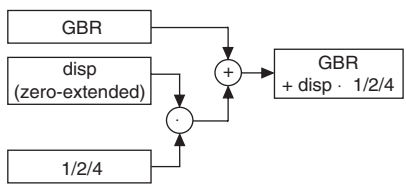
Classification	SH-2A CPU		Example of Other CPU
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W @(H'1234,R1),R2
	MOV.W	@(R0,R1),R2	
		.....	
	.DATA.W	H'1234	

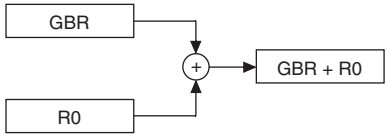
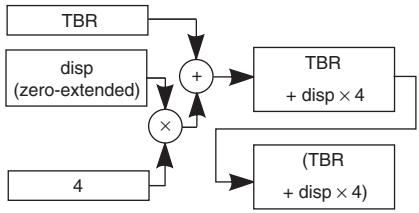
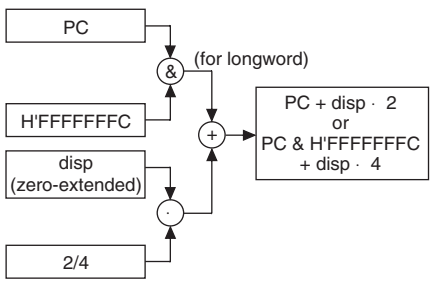
2.3.2 Addressing Modes

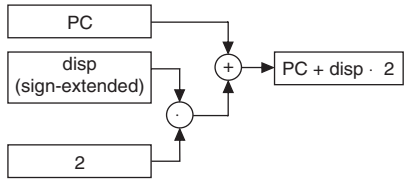
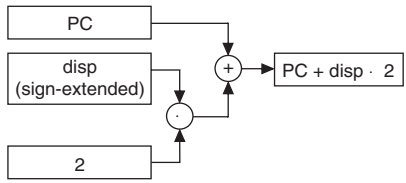
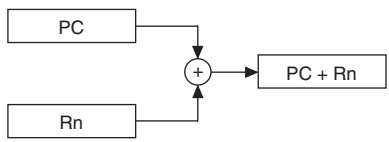
Addressing modes and effective address calculation are as follows:

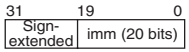
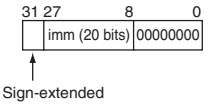
Table 2.8 Addressing Modes and Effective Addresses

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Register direct	`Rn`	The effective address is register Rn. (The operand is the contents of register Rn.)	—
Register indirect	`@Rn`	The effective address is the contents of register Rn. 	Rn
Register indirect with post-increment	`@Rn+`	The effective address is the contents of register Rn. A constant is added to the contents of Rn after the instruction is executed. 1 is added for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Rn (After instruction execution) Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn
Register indirect with pre-decrement	`@-Rn`	The effective address is the value obtained by subtracting a constant from Rn. 1 is subtracted for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn (Instruction is executed with Rn after this calculation)

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Register indirect with displacement	@(disp:4, Rn)	The effective address is the sum of Rn and a 4-bit displacement (disp). The value of disp is zero-extended, and remains unchanged for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: Rn + disp Word: Rn + disp × 2 Longword: Rn + disp × 4
Register indirect with displacement	@(disp:12, Rn)	The effective address is the sum of Rn and a 12-bit displacement (disp). The value of disp is zero-extended. 	Byte: Rn + disp Word: Rn + disp Longword: Rn + disp
Indexed register indirect	@(R0,Rn)	The effective address is the sum of Rn and R0. 	Rn + R0
GBR indirect with displacement	@(disp:8, GBR)	The effective address is the sum of GBR value and an 8-bit displacement (disp). The value of disp is zero-extended, and remains unchanged for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: GBR + disp Word: GBR + disp × 2 Longword: GBR + disp × 4

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Indexed GBR indirect	@(R0, GBR)	The effective address is the sum of GBR value and R0. 	GBR + R0
TBR duplicate indirect with displacement	@@ (disp:8, TBR)	The effective address is the sum of TBR value and an 8-bit displacement (disp). The value of disp is zero-extended, and is multiplied by 4. 	Contents of address (TBR + disp × 4)
PC indirect with displacement	@(disp:8, PC)	The effective address is the sum of PC value and an 8-bit displacement (disp). The value of disp is zero-extended, and is doubled for a word operation, and quadrupled for a longword operation. For a longword operation, the lowest two bits of the PC value are masked. 	Word: PC + disp × 2 Longword: PC & H'FFFFFFFC + disp × 4

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
PC relative	disp:8	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 8-bit displacement (disp). 	PC + disp × 2
	disp:12	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 12-bit displacement (disp). 	PC + disp × 2
	Rn	The effective address is the sum of PC value and Rn. 	PC + Rn

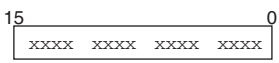
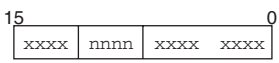
Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Immediate	#imm:20	The 20-bit immediate data (imm) for the MOVI20 instruction is sign-extended. 	—
		The 20-bit immediate data (imm) for the MOVI20S instruction is shifted by eight bits to the left, the upper bits are sign-extended, and the lower bits are padded with zero. 	—
	#imm:8	The 8-bit immediate data (imm) for the TST, AND, OR, and XOR instructions is zero-extended.	—
	#imm:8	The 8-bit immediate data (imm) for the MOV, ADD, and CMP/EQ instructions is sign-extended.	—
	#imm:8	The 8-bit immediate data (imm) for the TRAPA instruction is zero-extended and then quadrupled.	—
	#imm:3	The 3-bit immediate data (imm) for the BAND, BOR, BXOR, BST, BLD, BSET, and BCLR instructions indicates the target bit location.	—

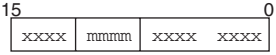
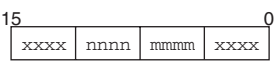
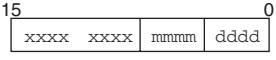
2.3.3 Instruction Format

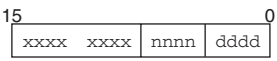
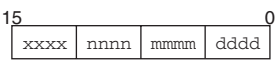
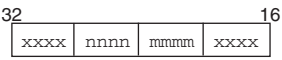
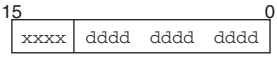
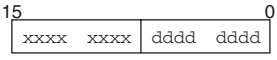
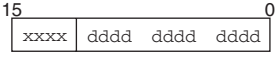
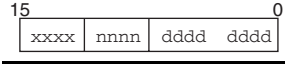
The instruction formats and the meaning of source and destination operands are described below. The meaning of the operand depends on the instruction code. The symbols used are as follows:

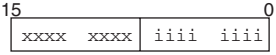
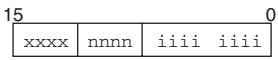
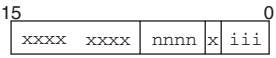
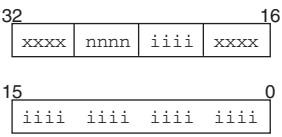
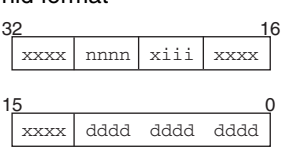
- xxxx: Instruction code
- mmmm: Source register
- nnnn: Destination register
- iiii: Immediate data
- dddd: Displacement

Table 2.9 Instruction Formats

Instruction Formats	Source Operand	Destination Operand	Example
0 format 	—	—	`NOP`
n format 	—	nnnn: Register direct	`MOVT Rn`
	Control register or system register	nnnn: Register direct	`STS MACH,Rn`
	R0 (Register direct)	nnnn: Register direct	`DIVU R0,Rn`
	Control register or system register	nnnn: Register indirect with pre-decrement	`STC.L SR,@-Rn`
	mmmm: Register direct	R15 (Register indirect with pre-decrement)	`MOVMU.L Rm,@-R15`
	R15 (Register indirect with post-increment)	nnnn: Register direct	`MOVMU.L @R15+,Rn`
	R0 (Register direct)	nnnn: (Register indirect with post-increment)	`MOV.L R0,@Rn+`

Instruction Formats	Source Operand	Destination Operand	Example
m format 	mmmm: Register direct	Control register or system register	LDC Rm,SR
	mmmm: Register indirect with post-increment	Control register or system register	LDC.L @Rm+,SR
	mmmm: Register indirect	—	JMP @Rm
	mmmm: Register indirect with pre-decrement	R0 (Register direct)	MOV.L @-Rm,R0
	mmmm: PC relative using Rm	—	BRAF Rm
nm format 	mmmm: Register direct	nnnn: Register direct	ADD Rm,Rn
	mmmm: Register direct	nnnn: Register indirect	MOV.L Rm,@Rn
	mmmm: Register indirect with post-increment (multiply-and-accumulate) nnnn*: Register indirect with post-increment (multiply-and-accumulate)	MACH, MACL	MAC.W @Rm+,@Rn+
	mmmm: Register indirect with post-increment	nnnn: Register direct	MOV.L @Rm+,Rn
	mmmm: Register direct	nnnn: Register indirect with pre-decrement	MOV.L Rm,@-Rn
	mmmm: Register direct	nnnn: Indexed register indirect	MOV.L Rm,@(R0,Rn)
md format 	mmmmdddd: Register indirect with displacement	R0 (Register direct)	MOV.B @(disp,Rm),R0

Instruction Formats	Source Operand	Destination Operand	Example
nd4 format 	R0 (Register direct)	nnnndddd: Register indirect with displacement	MOV.B R0,@(disp,Rn)
nmd format 	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
nmd12 format  	mmmm: Register direct	nnnndddd: Register indirect with displacement	MOV.L Rm,@(disp12,Rn)
	mmmmdddd: Register indirect with displacement	nnnn: Register direct	MOV.L @(disp12,Rm),Rn
d format 	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: TBR duplicate indirect with displacement	—	JSR/N @@(disp8,TBR)
	dddddddd: PC relative	—	BF label
d12 format 	dddddddddddd: PC relative	—	BRA label (label = disp + PC)
nd8 format 	dddddddd: PC relative with displacement	nnnn: Register direct	MOV.L @(disp,PC),Rn

Instruction Formats	Source Operand	Destination Operand	Example
i format 	iiiiiiii: Immediate	Indexed GBR indirect	AND.B #imm,@(R0,GBR)
	iiiiiiii: Immediate	R0 (Register direct)	AND #imm,R0
	iiiiiiii: Immediate	—	TRAPA #imm
ni format 	iiiiiiii: Immediate	nnnn: Register direct	ADD #imm,Rn
ni3 format 	nnnn: Register direct iii: Immediate	—	BLD #imm3,Rn
	—	nnnn: Register direct iii: Immediate	BST #imm3,Rn
ni20 format 	iiiiiiiiiiiiiiii: Immediate	nnnn: Register direct	MOVI20 #imm20, Rn
nid format 	nnnndddddddddddd: Register indirect with displacement iii: Immediate	—	BLD.B #imm3,@(disp12,Rn)
	—	nnnndddddddddddd: Register indirect with displacement iii: Immediate	BST.B #imm3,@(disp12,Rn)

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

2.4 Instruction Set

2.4.1 Instruction Set by Classification

Table 2.10 lists the instructions according to their classification.

Table 2.10 Classification of Instructions

Classification	Types	Operation Code	Function	No. of Instructions
Data transfer	13	MOV	Data transfer Immediate data transfer Peripheral module data transfer Structure data transfer Reverse stack transfer	62
		MOVA	Effective address transfer	
		MOVI20	20-bit immediate data transfer	
		MOVI20S	20-bit immediate data transfer 8-bit left-shit	
		MOVML	R0–Rn register save/restore	
		MOVMU	Rn–R14 and PR register save/restore	
		MOVRT	T bit inversion and transfer to Rn	
		MOVT	T bit transfer	
		MOVU	Unsigned data transfer	
		NOTT	T bit inversion	
		PREF	Prefetch to operand cache	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of the middle of registers connected	

Classification	Types	Operation Code	Function	No. of Instructions
Arithmetic operations	26	ADD	Binary addition	40
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		CLIPS	Signed saturation value comparison	
		CLIPU	Unsigned saturation value comparison	
		DIVS	Signed division (32 ÷ 32)	
		DIVU	Unsigned division (32 ÷ 32)	
		DIV1	One-step division	
		DIV0S	Initialization of signed one-step division	
		DIV0U	Initialization of unsigned one-step division	
		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 operation	
		MUL	Double-precision multiply operation	
		MULR	Signed multiplication with result storage in Rn	
		MULS	Signed multiplication	
		MULU	Unsigned multiplication	
		NEG	Negation	
		NEGC	Negation with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow	

Classification	Types	Operation Code	Function	No. of Instructions
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	12	ROTL	One-bit left rotation	16
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAD	Dynamic arithmetic shift	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLD	Dynamic logical shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
Branch	10	BF	Conditional branch, conditional delayed branch (branch when T = 0)	15
		BT	Conditional branch, conditional delayed branch (branch when T = 1)	
		BRA	Unconditional delayed branch	
		BRAF	Unconditional delayed branch	
		BSR	Delayed branch to subroutine procedure	
		BSRF	Delayed branch to subroutine procedure	
		JMP	Unconditional delayed branch	
		JSR	Branch to subroutine procedure	
			Delayed branch to subroutine procedure	
		RTS	Return from subroutine procedure	
			Delayed return from subroutine procedure	
		RTV/N	Return from subroutine procedure with Rm → R0 transfer	

Classification	Types	Operation Code	Function	No. of Instructions
System control	14	CLRT	T bit clear	36
		CLRMAC	MAC register clear	
		LDBANK	Register restoration from specified register bank entry	
		LDC	Load to control register	
		LDS	Load to system register	
		NOP	No operation	
		RESBANK	Register restoration from register bank	
		RTE	Return from exception handling	
		SETT	T bit set	
		SLEEP	Transition to power-down mode	
		STBANK	Register save to specified register bank entry	
		STC	Store control register data	
		STS	Store system register data	
		TRAPA	Trap exception handling	
Floating-point instructions	19	FABS	Floating-point absolute value	48
		FADD	Floating-point addition	
		FCMP	Floating-point comparison	
		FCNVDS	Conversion from double-precision to single-precision	
		FCNVSD	Conversion from single-precision to double -precision	
		FDIV	Floating-point division	
		FLDI0	Floating-point load immediate 0	
		FLDI1	Floating-point load immediate 1	
		FLDS	Floating-point load into system register FPUL	
		FLOAT	Conversion from integer to floating-point	
		FMAC	Floating-point multiply and accumulate operation	
		FMOV	Floating-point data transfer	
		FMUL	Floating-point multiplication	
		FNEG	Floating-point sign inversion	

Classification	Types	Operation Code	Function	No. of Instructions
Floating-point instructions	19	FSCHG	SZ bit inversion	48
		FSQRT	Floating-point square root	
		FSTS	Floating-point store from system register FPUL	
		FSUB	Floating-point subtraction	
		FTRC	Floating-point conversion with rounding to integer	
FPU-related CPU instructions	2	LDS	Load into floating-point system register	8
		STS	Store from floating-point system register	
Bit manipulation	10	BAND	Bit AND	14
		BCLR	Bit clear	
		BLD	Bit load	
		BOR	Bit OR	
		BSET	Bit set	
		BST	Bit store	
		BXOR	Bit exclusive OR	
		BANDNOT	Bit NOT AND	
		BORNOT	Bit NOT OR	
		BLDNOT	Bit NOT load	
Total:	112			253

The table below shows the format of instruction codes, operation, and execution states. They are described by using this format according to their classification.

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.
[Legend] Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement*2	[Legend] mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ..... 1111: R15 iiii: Immediate data dddd: Displacement	[Legend] →, ←: 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. Instruction execution cycles: The execution cycles shown in the table are minimums. In practice, the number of instruction execution states will be increased in cases such as the following:
   a. When there is a conflict between an instruction fetch and a data access
   b. When the destination register of a load instruction (memory → register) is the same as the register used by the next instruction.
2. Depending on the operand size, displacement is scaled by ×1, ×2, or ×4. For details, refer to the SH-2A, SH2A-FPU Software Manual.

2.4.2 Data Transfer Instructions

Table 2.11 Data Transfer Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
MOV	#imm,Rn	1110nnnniiiiiiii	imm → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.W	@(disp,PC),Rn	1001nnnnddddddd	(disp × 2 + PC) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.L	@(disp,PC),Rn	1101nnnnddddddd	(disp × 4 + PC) → Rn	1	—	Yes	Yes	Yes
MOV	Rm,Rn	0110nnnnmmmm0011	Rm → Rn	1	—	Yes	Yes	Yes
MOV.B	Rm,@Rn	0010nnnnmmmm0000	Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.W	Rm,@Rn	0010nnnnmmmm0001	Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.L	Rm,@Rn	0010nnnnmmmm0010	Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.B	@Rm,Rn	0110nnnnmmmm0000	(Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.W	@Rm,Rn	0110nnnnmmmm0001	(Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.L	@Rm,Rn	0110nnnnmmmm0010	(Rm) → Rn	1	—	Yes	Yes	Yes
MOV.B	Rm,@-Rn	0010nnnnmmmm0100	Rn-1 → Rn, Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.W	Rm,@-Rn	0010nnnnmmmm0101	Rn-2 → Rn, Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.L	Rm,@-Rn	0010nnnnmmmm0110	Rn-4 → Rn, Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.B	@Rm+,Rn	0110nnnnmmmm0100	(Rm) → sign extension → Rn, Rm + 1 → Rm	1	—	Yes	Yes	Yes
MOV.W	@Rm+,Rn	0110nnnnmmmm0101	(Rm) → sign extension → Rn, Rm + 2 → Rm	1	—	Yes	Yes	Yes
MOV.L	@Rm+,Rn	0110nnnnmmmm0110	(Rm) → Rn, Rm + 4 → Rm	1	—	Yes	Yes	Yes
MOV.B	R0,@(disp,Rn)	10000000nnnndddd	R0 → (disp + Rn)	1	—	Yes	Yes	Yes
MOV.W	R0,@(disp,Rn)	10000001nnnndddd	R0 → (disp × 2 + Rn)	1	—	Yes	Yes	Yes
MOV.L	Rm,@(disp,Rn)	0001nnnnmmmmdddd	Rm → (disp × 4 + Rn)	1	—	Yes	Yes	Yes
MOV.B	@(disp,Rm),R0	10000100mmmmdddd	(disp + Rm) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.W	@(disp,Rm),R0	10000101mmmmdddd	(disp × 2 + Rm) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.L	@(disp,Rm),Rn	0101nnnnmmmmdddd	(disp × 4 + Rm) → Rn	1	—	Yes	Yes	Yes
MOV.B	Rm,@(R0,Rn)	0000nnnnmmmm0100	Rm → (R0 + Rn)	1	—	Yes	Yes	Yes
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm → (R0 + Rn)	1	—	Yes	Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
MOV.L	Rm,@(R0,Rn)	0000nnnnmmmm0110	Rm → (R0 + Rn)	1	—	Yes	Yes	Yes
MOV.B	@(R0,Rm),Rn	0000nnnnmmmm1100	(R0 + Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.W	@(R0,Rm),Rn	0000nnnnmmmm1101	(R0 + Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.L	@(R0,Rm),Rn	0000nnnnmmmm1110	(R0 + Rm) → Rn	1	—	Yes	Yes	Yes
MOV.B	R0,@(disp,GBR)	11000000dddddddd	R0 → (disp + GBR)	1	—	Yes	Yes	Yes
MOV.W	R0,@(disp,GBR)	11000001dddddddd	R0 → (disp × 2 + GBR)	1	—	Yes	Yes	Yes
MOV.L	R0,@(disp,GBR)	11000010dddddddd	R0 → (disp × 4 + GBR)	1	—	Yes	Yes	Yes
MOV.B	@(disp,GBR),R0	11000100dddddddd	(disp + GBR) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.W	@(disp,GBR),R0	11000101dddddddd	(disp × 2 + GBR) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp × 4 + GBR) → R0	1	—	Yes	Yes	Yes
MOV.B	R0,@Rn+	0100nnnn10001011	R0 → (Rn), Rn + 1 → Rn	1	—			Yes
MOV.W	R0,@Rn+	0100nnnn10011011	R0 → (Rn), Rn + 2 → Rn	1	—			Yes
MOV.L	R0,@Rn+	0100nnnn10101011	R0 → Rn), Rn + 4 → Rn	1	—			Yes
MOV.B	@-Rm,R0	0100mmmm11001011	Rm-1 → Rm, (Rm) → sign extension → R0	1	—			Yes
MOV.W	@-Rm,R0	0100mmmm11011011	Rm-2 → Rm, (Rm) → sign extension → R0	1	—			Yes
MOV.L	@-Rm,R0	0100mmmm11101011	Rm-4 → Rm, (Rm) → R0	1	—			Yes
MOV.B	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0000dddddddddddd	Rm → (disp + Rn)	1	—			Yes
MOV.W	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0001dddddddddddd	Rm → (disp × 2 + Rn)	1	—			Yes
MOV.L	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0010dddddddddddd	Rm → (disp × 4 + Rn)	1	—			Yes
MOV.B	@(disp12,Rm),Rn	0011nnnnmmmm0001 0100dddddddddddd	(disp + Rm) → sign extension → Rn	1	—			Yes
MOV.W	@(disp12,Rm),Rn	0011nnnnmmmm0001 0101dddddddddddd	(disp × 2 + Rm) → sign extension → Rn	1	—			Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
MOV.L	@(disp12,Rm),Rn	0011nnnnmmmm0001 0110ddddddddddd	(disp × 4 + Rm) → Rn	1	—			Yes
MOVA	@(disp,PC),R0	11000111dddddddd	disp × 4 + PC → R0	1	—	Yes	Yes	Yes
MOVI20	#imm20,Rn	0000nnnniiii0000 iiiiiiiiiiiiiiii	imm → sign extension → Rn	1	—			Yes
MOVI20S	#imm20,Rn	0000nnnniiii0001 iiiiiiiiiiiiiiii	imm << 8 → sign extension → Rn	1	—			Yes
MOVML.L	Rm,@-R15	0100mmmm11110001	R15-4 → R15, Rm → (R15) R15-4 → R15, Rm-1 → (R15) : R15-4 → R15, R0 → (R15) Note: When Rm = R15, read Rm as PR	1 to 16	—			Yes
MOVML.L	@R15+,Rn	0100nnnn11110101	(R15) → R0, R15 + 4 → R15 (R15) → R1, R15 + 4 → R15 : (R15) → Rn Note: When Rn = R15, read Rn as PR	1 to 16	—			Yes
MOVMU.L	Rm,@-R15	0100mmmm11110000	R15-4 → R15, PR → (R15) R15-4 → R15, R14 → (R15) : R15-4 → R15, Rm → (R15) Note: When Rm = R15, read Rm as PR	1 to 16	—			Yes
MOVMU.L	@R15+,Rn	0100nnnn11110100	(R15) → Rn, R15 + 4 → R15 (R15) → Rn + 1, R15 + 4 → R15 : (R15) → R14, R15 + 4 → R15 (R15) → PR Note: When Rn = R15, read Rn as PR	1 to 16	—			Yes
MOVRT	Rn	0000nnnn00111001	~T → Rn	1	—			Yes
MOVT	Rn	0000nnnn00101001	T → Rn	1	—	Yes	Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
MOVU.B	@(disp12,Rm),Rn	0011nnnnmmmm0001 1000dddddddddddd	(disp + Rm) → zero extension → Rn	1	—			Yes
MOVU.W	@(disp12,Rm),Rn	0011nnnnmmmm0001 1001dddddddddddd	(disp × 2 + Rm) → zero extension → Rn	1	—			Yes
NOTT		0000000001101000	~T → T	1	Operation result			Yes
PREF	@Rn	0000nnnn10000011	(Rn) → operand cache	1	—		Yes	Yes
SWAP.B	Rm,Rn	0110nnnnmmmm1000	Rm → swap lower 2 bytes → Rn	1	—	Yes	Yes	Yes
SWAP.W	Rm,Rn	0110nnnnmmmm1001	Rm → swap upper and lower words → Rn	1	—	Yes	Yes	Yes
XTRCT	Rm,Rn	0010nnnnmmmm1101	Rm:Rn middle 32 bits → Rn	1	—	Yes	Yes	Yes

2.4.3 Arithmetic Operation Instructions

Table 2.12 Arithmetic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
ADD	Rm,Rn	0011nnnnmmmm1100	Rn + Rm → Rn	1	—	Yes	Yes	Yes
ADD	#imm,Rn	0111nnnniiiiiiii	Rn + imm → Rn	1	—	Yes	Yes	Yes
ADDC	Rm,Rn	0011nnnnmmmm1110	Rn + Rm + T → Rn, carry → T	1	Carry	Yes	Yes	Yes
ADDV	Rm,Rn	0011nnnnmmmm1111	Rn + Rm → Rn, overflow → T	1	Over-flow	Yes	Yes	Yes
CMP/EQ	#imm,R0	10001000iiiiiiii	When R0 = imm, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	When Rn = Rm, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/HS	Rm,Rn	0011nnnnmmmm0010	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/GE	Rm,Rn	0011nnnnmmmm0011	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/HI	Rm,Rn	0011nnnnmmmm0110	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/GT	Rm,Rn	0011nnnnmmmm0111	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/PL	Rn	0100nnnn00010101	When Rn > 0, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/PZ	Rn	0100nnnn00010001	When Rn ≥ 0, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/STR	Rm,Rn	0010nnnnmmmm1100	When any bytes are equal, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
CLIPS.B	Rn	0100nnnn10010001	When Rn > (H'0000007F), (H'0000007F) → Rn, 1 → CS when Rn < (H'FFFFFF80), (H'FFFFFF80) → Rn, 1 → CS	1	—			Yes
CLIPS.W	Rn	0100nnnn10010101	When Rn > (H'00007FFF), (H'00007FFF) → Rn, 1 → CS When Rn < (H'FFFF8000), (H'FFFF8000) → Rn, 1 → CS	1	—			Yes
CLIPU.B	Rn	0100nnnn10000001	When Rn > (H'000000FF), (H'000000FF) → Rn, 1 → CS	1	—			Yes
CLIPU.W	Rn	0100nnnn10000101	When Rn > (H'0000FFFF), (H'0000FFFF) → Rn, 1 → CS	1	—			Yes
DIV1	Rm,Rn	0011nnnnmmmm0100	1-step division (Rn ÷ Rm)	1	Calculation result	Yes	Yes	Yes
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	1	Calculation result	Yes	Yes	Yes
DIV0U		0000000000011001	0 → M/Q/T	1	0	Yes	Yes	Yes
DIVS	R0,Rn	0100nnnn10010100	Signed operation of Rn ÷ R0 → Rn 32 ÷ 32 → 32 bits	36	—			Yes
DIVU	R0,Rn	0100nnnn10000100	Unsigned operation of Rn ÷ R0 → Rn 32 ÷ 32 → 32 bits	34	—			Yes
DMULS.L	Rm,Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2	—	Yes	Yes	Yes
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2	—	Yes	Yes	Yes
DT	Rn	0100nnnn00010000	Rn – 1 → Rn When Rn is 0, 1 → T When Rn is not 0, 0 → T	1	Comparison result	Yes	Yes	Yes
EXTS.B	Rm,Rn	0110nnnnmmmm1110	Byte in Rm is sign-extended → Rn	1	—	Yes	Yes	Yes
EXTS.W	Rm,Rn	0110nnnnmmmm1111	Word in Rm is sign-extended → Rn	1	—	Yes	Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
EXTU.B	Rm,Rn	0110nnnnmmmm1100	Byte in Rm is zero-extended → Rn	1	—	Yes	Yes	Yes
EXTU.W	Rm,Rn	0110nnnnmmmm1101	Word in Rm is zero-extended → Rn	1	—	Yes	Yes	Yes
MAC.L	@Rm+,@Rn+	0000nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 32 × 32 + 64 → 64 bits	4	—	Yes	Yes	Yes
MAC.W	@Rm+,@Rn+	0100nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 16 × 16 + 64 → 64 bits	3	—	Yes	Yes	Yes
MUL.L	Rm,Rn	0000nnnnmmmm0111	Rn × Rm → MACL 32 × 32 → 32 bits	2	—	Yes	Yes	Yes
MULR	R0,Rn	0100nnnn10000000	R0 × Rn → Rn 32 × 32 → 32 bits	2				Yes
MULS.W	Rm,Rn	0010nnnnmmmm1111	Signed operation of Rn × Rm → MACL 16 × 16 → 32 bits	1	—	Yes	Yes	Yes
MULU.W	Rm,Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MACL 16 × 16 → 32 bits	1	—	Yes	Yes	Yes
NEG	Rm,Rn	0110nnnnmmmm1011	0-Rm → Rn	1	—	Yes	Yes	Yes
NEGC	Rm,Rn	0110nnnnmmmm1010	0-Rm-T → Rn, borrow → T	1	Borrow	Yes	Yes	Yes
SUB	Rm,Rn	0011nnnnmmmm1000	Rn-Rm → Rn	1	—	Yes	Yes	Yes
SUBC	Rm,Rn	0011nnnnmmmm1010	Rn-Rm-T → Rn, borrow → T	1	Borrow	Yes	Yes	Yes
SUBV	Rm,Rn	0011nnnnmmmm1011	Rn-Rm → Rn, underflow → T	1	Over-flow	Yes	Yes	Yes

2.4.4 Logic Operation Instructions

Table 2.13 Logic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
AND	Rm,Rn	0010nnnnmmmm1001	Rn & Rm → Rn	1	—	Yes	Yes	Yes
AND	#imm,R0	11001001iiiiiiii	R0 & imm → R0	1	—	Yes	Yes	Yes
AND.B	#imm,@(R0,GBR)	11001101iiiiiiii	(R0 + GBR) & imm → (R0 + GBR)	3	—	Yes	Yes	Yes
NOT	Rm,Rn	0110nnnnmmmm0111	~Rm → Rn	1	—	Yes	Yes	Yes
OR	Rm,Rn	0010nnnnmmmm1011	Rn | Rm → Rn	1	—	Yes	Yes	Yes
OR	#imm,R0	11001011iiiiiiii	R0 | imm → R0	1	—	Yes	Yes	Yes
OR.B	#imm,@(R0,GBR)	11001111iiiiiiii	(R0 + GBR) | imm → (R0 + GBR)	3	—	Yes	Yes	Yes
TAS.B	@Rn	0100nnnn00011011	When (Rn) is 0, 1 → T Otherwise, 0 → T, 1 → MSB of(Rn)	3	Test result	Yes	Yes	Yes
TST	Rm,Rn	0010nnnnmmmm1000	Rn & Rm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	Yes	Yes
TST	#imm,R0	11001000iiiiiiii	R0 & imm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	Yes	Yes
TST.B	#imm,@(R0,GBR)	11001100iiiiiiii	(R0 + GBR) & imm When the result is 0, 1 → T Otherwise, 0 → T	3	Test result	Yes	Yes	Yes
XOR	Rm,Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	1	—	Yes	Yes	Yes
XOR	#imm,R0	11001010iiiiiiii	R0 ^ imm → R0	1	—	Yes	Yes	Yes
XOR.B	#imm,@(R0,GBR)	11001110iiiiiiii	(R0 + GBR) ^ imm → (R0 + GBR)	3	—	Yes	Yes	Yes

2.4.5 Shift Instructions

Table 2.14 Shift Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
ROTL	Rn	0100nnnn00000100	T ← Rn ← MSB	1	MSB	Yes	Yes	Yes
ROTR	Rn	0100nnnn00000101	LSB → Rn → T	1	LSB	Yes	Yes	Yes
ROTCL	Rn	0100nnnn00100100	T ← Rn ← T	1	MSB	Yes	Yes	Yes
ROTCR	Rn	0100nnnn00100101	T → Rn → T	1	LSB	Yes	Yes	Yes
SHAD	Rm,Rn	0100nnnnmmmm1100	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [MSB → Rn]	1	—		Yes	Yes
SHAL	Rn	0100nnnn00100000	T ← Rn ← 0	1	MSB	Yes	Yes	Yes
SHAR	Rn	0100nnnn00100001	MSB → Rn → T	1	LSB	Yes	Yes	Yes
SHLD	Rm,Rn	0100nnnnmmmm1101	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [0 → Rn]	1	—		Yes	Yes
SHLL	Rn	0100nnnn00000000	T ← Rn ← 0	1	MSB	Yes	Yes	Yes
SHLR	Rn	0100nnnn00000001	0 → Rn → T	1	LSB	Yes	Yes	Yes
SHLL2	Rn	0100nnnn00001000	Rn << 2 → Rn	1	—	Yes	Yes	Yes
SHLR2	Rn	0100nnnn00001001	Rn >> 2 → Rn	1	—	Yes	Yes	Yes
SHLL8	Rn	0100nnnn00011000	Rn << 8 → Rn	1	—	Yes	Yes	Yes
SHLR8	Rn	0100nnnn00011001	Rn >> 8 → Rn	1	—	Yes	Yes	Yes
SHLL16	Rn	0100nnnn00101000	Rn << 16 → Rn	1	—	Yes	Yes	Yes
SHLR16	Rn	0100nnnn00101001	Rn >> 16 → Rn	1	—	Yes	Yes	Yes

2.4.6 Branch Instructions

Table 2.15 Branch Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
BF	label	10001011dddddddd	When T = 0, disp × 2 + PC → PC, When T = 1, nop	3/1*	—	Yes	Yes	Yes
BF/S	label	10001111dddddddd	Delayed branch When T = 0, disp × 2 + PC → PC, When T = 1, nop	2/1*	—	Yes	Yes	Yes
BT	label	10001001dddddddd	When T = 1, disp × 2 + PC → PC, When T = 0, nop	3/1*	—	Yes	Yes	Yes
BT/S	label	10001101dddddddd	Delayed branch When T = 1, disp × 2 + PC → PC, When T = 0, nop	2/1*	—	Yes	Yes	Yes
BRA	label	1010dddddddddddd	Delayed branch, disp × 2 + PC → PC	2	—	Yes	Yes	Yes
BRAF	Rm	0000mmmm00100011	Delayed branch, Rm + PC → PC	2	—	Yes	Yes	Yes
BSR	label	1011dddddddddddd	Delayed branch, PC → PR, disp × 2 + PC → PC	2	—	Yes	Yes	Yes
BSRF	Rm	0000mmmm00000011	Delayed branch, PC → PR, Rm + PC → PC	2	—	Yes	Yes	Yes
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm → PC	2	—	Yes	Yes	Yes
JSR	@Rm	0100mmmm00001011	Delayed branch, PC → PR, Rm → PC	2	—	Yes	Yes	Yes
JSR/N	@Rm	0100mmmm01001011	PC-2 → PR, Rm → PC	3	—			Yes
JSR/N	@@(disp8,TBR)	10000011dddddddd	PC-2 → PR, (disp × 4 + TBR) → PC	5	—			Yes
RTS		0000000000001011	Delayed branch, PR → PC	2	—	Yes	Yes	Yes
RTS/N		0000000001101011	PR → PC	3	—			Yes
RTV/N	Rm	0000mmmm01111011	Rm → R0, PR → PC	3	—			Yes

Note: * One cycle when the program does not branch.

2.4.7 System Control Instructions

Table 2.16 System Control Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
CLRT		0000000000001000	0 → T	1	0	Yes	Yes	Yes
CLRMAC		0000000000101000	0 → MACH,MACL	1	—	Yes	Yes	Yes
LDBANK	@Rm,R0	0100mmmm11100101	(Specified register bank entry) → R0	6	—			Yes
LDC	Rm,SR	0100mmmm00001110	Rm → SR	3	LSB	Yes	Yes	Yes
LDC	Rm,TBR	0100mmmm01001010	Rm → TBR	1	—			Yes
LDC	Rm,GBR	0100mmmm00011110	Rm → GBR	1	—	Yes	Yes	Yes
LDC	Rm,VBR	0100mmmm00101110	Rm → VBR	1	—	Yes	Yes	Yes
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm) → SR, Rm + 4 → Rm	5	LSB	Yes	Yes	Yes
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm) → GBR, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm) → VBR, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDS	Rm,MACH	0100mmmm00001010	Rm → MACH	1	—	Yes	Yes	Yes
LDS	Rm,MACL	0100mmmm00011010	Rm → MACL	1	—	Yes	Yes	Yes
LDS	Rm,PR	0100mmmm00101010	Rm → PR	1	—	Yes	Yes	Yes
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm) → MACH, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm) → MACL, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm) → PR, Rm + 4 → Rm	1	—	Yes	Yes	Yes
NOP		0000000000001001	No operation	1	—	Yes	Yes	Yes
RESBANK		0000000001011011	Bank → R0 to R14, GBR, MACH, MACL, PR	9*	—			Yes
RTE		0000000000101011	Delayed branch, stack area → PC/SR	6	—	Yes	Yes	Yes
SETT		0000000000011000	1 → T	1	1	Yes	Yes	Yes
SLEEP		0000000000011011	Sleep	5	—	Yes	Yes	Yes
STBANK	R0,@Rn	0100nnnn11100001	R0 → (specified register bank entry)	7	—			Yes
STC	SR,Rn	0000nnnn00000010	SR → Rn	2	—	Yes	Yes	Yes
STC	TBR,Rn	0000nnnn01001010	TBR → Rn	1	—			Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
STC	GBR,Rn	0000nnnn00010010	GBR → Rn	1	—	Yes	Yes	Yes
STC	VBR,Rn	0000nnnn00100010	VBR → Rn	1	—	Yes	Yes	Yes
STC.L	SR,@-Rn	0100nnnn00000011	Rn-4 → Rn, SR → (Rn)	2	—	Yes	Yes	Yes
STC.L	GBR,@-Rn	0100nnnn00010011	Rn-4 → Rn, GBR → (Rn)	1	—	Yes	Yes	Yes
STC.L	VBR,@-Rn	0100nnnn00100011	Rn-4 → Rn, VBR → (Rn)	1	—	Yes	Yes	Yes
STS	MACH,Rn	0000nnnn00001010	MACH → Rn	1	—	Yes	Yes	Yes
STS	MACL,Rn	0000nnnn00011010	MACL → Rn	1	—	Yes	Yes	Yes
STS	PR,Rn	0000nnnn00101010	PR → Rn	1	—	Yes	Yes	Yes
STS.L	MACH,@-Rn	0100nnnn00000010	Rn-4 → Rn, MACH → (Rn)	1	—	Yes	Yes	Yes
STS.L	MACL,@-Rn	0100nnnn00010010	Rn-4 → Rn, MACL → (Rn)	1	—	Yes	Yes	Yes
STS.L	PR,@-Rn	0100nnnn00100010	Rn-4 → Rn, PR → (Rn)	1	—	Yes	Yes	Yes
TRAPA	#imm	11000011iiiiiiii	PC/SR → stack area, (imm × 4 + VBR) → PC	5	—	Yes	Yes	Yes

Notes: Instruction execution cycles: The execution cycles shown in the table are minimums. In practice, the number of instruction execution states in cases such as the following:

a. When there is a conflict between an instruction fetch and a data access

b. When the destination register of a load instruction (memory → register) is the same as the register used by the next instruction.

* In the event of bank overflow, the number of cycles is 19.

2.4.8 Floating-Point Operation Instructions

Table 2.17 Floating-Point Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
FABS	FRn	1111nnnn01011101	|FRn| → FRn	1	—	Yes	Yes	Yes
FABS	DRn	1111nnn001011101	|DRn| → DRn	1	—		Yes	Yes
FADD	FRm, FRn	1111nnnnmmmm0000	FRn + FRm → FRn	1	—	Yes	Yes	Yes
FADD	DRm, DRn	1111nnn0mmm00000	DRn + DRm → DRn	6	—		Yes	Yes
FCMP/EQ	FRm, FRn	1111nnnnmmmm0100	(FRn = FRm)? 1:0 → T	1	Comparison result	Yes	Yes	Yes
FCMP/EQ	DRm, DRn	1111nnn0mmm00100	(DRn = DRm)? 1:0 → T	2	Comparison result		Yes	Yes
FCMP/GT	FRm, FRn	1111nnnnmmmm0101	(FRn > FRm)? 1:0 → T	1	Comparison result	Yes	Yes	Yes
FCMP/GT	DRm, DRn	1111nnn0mmm00101	(DRn > DRm)? 1:0 → T	2	Comparison result		Yes	Yes
FCNVDS	DRm, FPUL	1111mmm010111101	(float) DRm → FPUL	2	—		Yes	Yes
FCNVSD	FPUL, DRn	1111nnn010101101	(double) FPUL → DRn	2	—		Yes	Yes
FDIV	FRm, FRn	1111nnnnmmmm0011	FRn/FRm → FRn	10	—	Yes	Yes	Yes
FDIV	DRm, DRn	1111nnn0mmm00011	DRn/DRm → DRn	23	—		Yes	Yes
FLDI0	FRn	1111nnnn10001101	0 × 00000000 → FRn	1	—	Yes	Yes	Yes
FLDI1	FRn	1111nnnn10011101	0 × 3F800000 → FRn	1	—	Yes	Yes	Yes
FLDS	FRm, FPUL	1111mmmm00011101	FRm → FPUL	1	—	Yes	Yes	Yes
FLOAT	FPUL,FRn	1111nnnn00101101	(float)FPUL → FRn	1	—	Yes	Yes	Yes
FLOAT	FPUL,DRn	1111nnn000101101	(double)FPUL → DRn	2	—		Yes	Yes
FMAC	FR0,FRm,FRn	1111nnnnmmmm1110	FR0 × FRm+FRn → FRn	1	—	Yes	Yes	Yes
FMOV	FRm, FRn	1111nnnnmmmm1100	FRm → FRn	1	—	Yes	Yes	Yes
FMOV	DRm, DRn	1111nnn0mmm01100	DRm → DRn	2	—		Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
FMOV.S	@(R0, Rm), FRn	1111nnnnmmmm0110	(R0 + Rm) → FRn	1	—	Yes	Yes	Yes
FMOV.D	@(R0, Rm), DRn	1111nnn0mmmm0110	(R0 + Rm) → DRn	2	—		Yes	Yes
FMOV.S	@Rm+, FRn	1111nnnnmmmm1001	(Rm) → FRn, Rm+=4	1	—	Yes	Yes	Yes
FMOV.D	@Rm+, DRn	1111nnn0mmmm1001	(Rm) → DRn, Rm += 8	2	—		Yes	Yes
FMOV.S	@Rm, FRn	1111nnnnmmmm1000	(Rm) → FRn	1	—	Yes	Yes	Yes
FMOV.D	@Rm, DRn	1111nnn0mmmm1000	(Rm) → DRn	2	—		Yes	Yes
FMOV.S	@(disp12,Rm),FRn	0011nnnnmmmm0001 0111dddddddddddd	(disp × 4 + Rm) → FRn	1	—			Yes
FMOV.D	@(disp12,Rm),DRn	0011nnn0mmmm0001 0111dddddddddddd	(disp × 8 + Rm) → DRn	2	—			Yes
FMOV.S	FRm, @(R0,Rn)	1111nnnnmmmm0111	FRm → (R0 + Rn)	1	—	Yes	Yes	Yes
FMOV.D	DRm, @(R0,Rn)	1111nnnnmmm00111	DRm → (R0 + Rn)	2	—		Yes	Yes
FMOV.S	FRm, @-Rn	1111nnnnmmmm1011	Rn-=4, FRm → (Rn)	1	—	Yes	Yes	Yes
FMOV.D	DRm, @-Rn	1111nnnnmmm01011	Rn-=8, DRm → (Rn)	2	—		Yes	Yes
FMOV.S	FRm, @Rn	1111nnnnmmmm1010	FRm → (Rn)	1	—	Yes	Yes	Yes
FMOV.D	DRm, @Rn	1111nnnnmmm01010	DRm → (Rn)	2	—		Yes	Yes
FMOV.S	FRm, @(disp12,Rn)	0011nnnnmmmm0001 0011dddddddddddd	FRm → (disp × 4 + Rn)	1	—			Yes
FMOV.D	DRm, @(disp12,Rn)	0011nnnnmmm00001 0011dddddddddddd	DRm → (disp × 8 + Rn)	2	—			Yes
FMUL	FRm, FRn	1111nnnnmmmm0010	FRn × FRm → FRn	1	—	Yes	Yes	Yes
FMUL	DRm, DRn	1111nnn0mmm00010	DRn × DRm → DRn	6	—		Yes	Yes
FNEG	FRn	1111nnnn01001101	-FRn → FRn	1	—	Yes	Yes	Yes
FNEG	DRn	1111nnn001001101	-DRn → DRn	1	—		Yes	Yes
FSCHG		1111001111111101	FPSCR.SZ=~FPSCR.SZ	1	—		Yes	Yes
FSQRT	FRn	1111nnnn01101101	√FRn → FRn	9	—		Yes	Yes
FSQRT	DRn	1111nnn001101101	√DRn → DRn	22	—		Yes	Yes
FSTS	FPUL,FRn	1111nnnn00001101	FPUL → FRn	1	—	Yes	Yes	Yes
FSUB	FRm, FRn	1111nnnnmmmm0001	FRn-FRm → FRn	1	—	Yes	Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
FSUB	DRm, DRn	1111nnn0mmm00001	DRn-DRm → DRn	6	—		Yes	Yes
FTRC	FRm, FPUL	1111mmmm00111101	(long)FRm → FPUL	1	—	Yes	Yes	Yes
FTRC	DRm, FPUL	1111mmm000111101	(long)DRm → FPUL	2	—		Yes	Yes

2.4.9 FPU-Related CPU Instructions

Table 2.18 FPU-Related CPU Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
LDS	Rm,FPSCR	0100mmmm01101010	Rm → FPSCR	1	—	Yes	Yes	Yes
LDS	Rm,FPUL	0100mmmm01011010	Rm → FPUL	1	—	Yes	Yes	Yes
LDS.L	@Rm+, FPSCR	0100mmmm01100110	(Rm) → FPSCR, Rm+=4	1	—	Yes	Yes	Yes
LDS.L	@Rm+, FPUL	0100mmmm01010110	(Rm) → FPUL, Rm+=4	1	—	Yes	Yes	Yes
STS	FPSCR, Rn	0000nnnn01101010	FPSCR → Rn	1	—	Yes	Yes	Yes
STS	FPUL,Rn	0000nnnn01011010	FPUL → Rn	1	—	Yes	Yes	Yes
STS.L	FPSCR,@-Rn	0100nnnn01100010	Rn-=4, FPCSR → (Rn)	1	—	Yes	Yes	Yes
STS.L	FPUL,@-Rn	0100nnnn01010010	Rn-=4, FPUL → (Rn)	1	—	Yes	Yes	Yes

2.4.10 Bit Manipulation Instructions

Table 2.19 Bit Manipulation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
BAND.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0100dddddddddddd	(imm of (disp + Rn)) & T →	3	Operation result			Yes
BANDNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1100dddddddddddd	~(imm of (disp + Rn)) & T → T	3	Operation result			Yes
BCLR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0000dddddddddddd	0 → (imm of (disp + Rn))	3	—			Yes
BCLR	#imm3,Rn	10000110nnnn0iii	0 → imm of Rn	1	—			Yes
BLD.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0011dddddddddddd	(imm of (disp + Rn)) →	3	Operation result			Yes
BLD	#imm3,Rn	10000111nnnn1iii	imm of Rn → T	1	Operation result			Yes
BLDNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1011dddddddddddd	~(imm of (disp + Rn)) → T	3	Operation result			Yes
BOR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0101dddddddddddd	(imm of (disp + Rn)) | T → T	3	Operation result			Yes
BORNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1101dddddddddddd	~(imm of (disp + Rn)) | T → T	3	Operation result			Yes
BSET.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0001dddddddddddd	1 → (imm of (disp + Rn))	3	—			Yes
BSET	#imm3,Rn	10000110nnnn1iii	1 → imm of Rn	1	—			Yes
BST.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0010dddddddddddd	T → (imm of (disp + Rn))	3	—			Yes
BST	#imm3,Rn	10000111nnnn0iii	T → imm of Rn	1	—			Yes
BXOR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0110dddddddddddd	(imm of (disp + Rn)) ^ T → T	3	Operation result			Yes

2.5 Processing States

The LSI has four CPU processing states: reset, dual-processor active, single-processor active, and power-down. Figure 2.6 shows the transitions between the states.

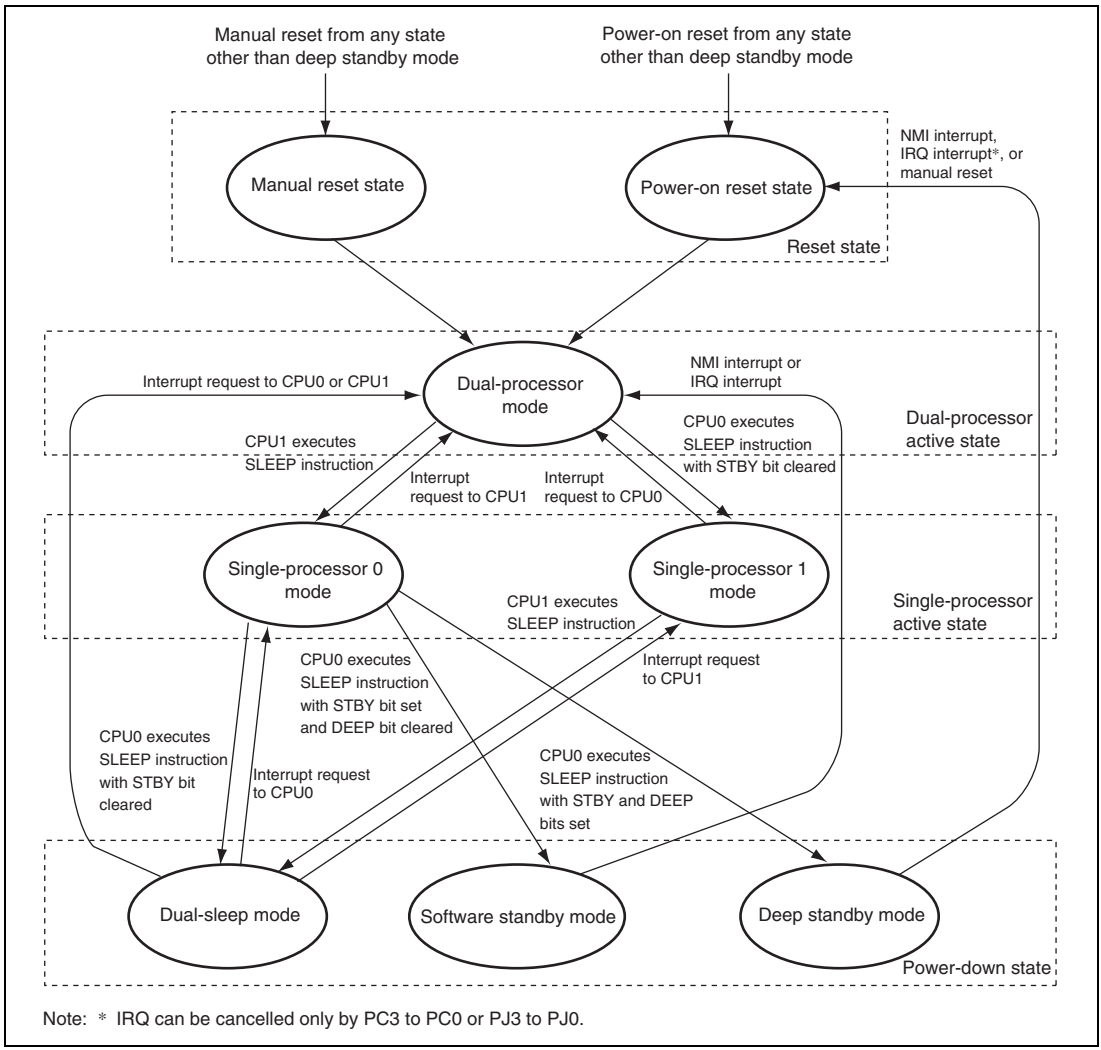


Figure 2.6 Transitions between Processing States

(1) Reset State

In the reset state, the CPU is reset. There are two kinds of reset, power-on reset and manual reset.

(2) Dual-Processor Active State

In this state, CPU0 and CPU1 sequentially execute their own programs.

(3) Single-Processor Active State

In this state, either CPU0 or CPU1 operates.
In single-processor 0 mode, CPU0 is active and CPU1 is in the sleep state.
In single-processor 1 mode, CPU1 is active and CPU0 is in the sleep state.

(4) Power-Down State

In the power-down state, both CPUs stop operating to reduce power consumption.

Section 3 Floating-Point Unit (FPU)

3.1 Features

The FPU has the following features.

- Conforms to IEEE754 standard
- 16 single-precision floating-point registers (can also be referenced as eight double-precision registers)
- Two rounding modes: Round to nearest and round to zero
- Denormalization modes: Flush to zero
- Five exception sources: Invalid operation, divide by zero, overflow, underflow, and inexact
- Comprehensive instructions: Single-precision, double-precision, and system control

3.2 Data Formats

3.2.1 Floating-Point Format

A floating-point number consists of the following three fields:

- Sign (s)
- Exponent (e)
- Fraction (f)

This LSI can handle single-precision and double-precision floating-point numbers, using the formats shown in figures 3.1 and 3.2.

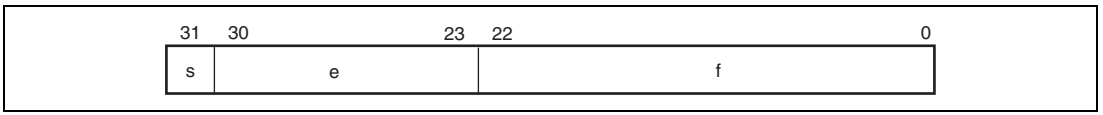


Figure 3.1 Format of Single-Precision Floating-Point Number

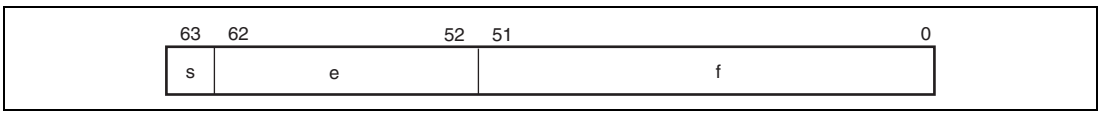


Figure 3.2 Format of Double-Precision Floating-Point Number

The exponent is expressed in biased form, as follows:

$$e = E + bias$$

The range of unbiased exponent E is $E_{min} - 1$ to $E_{max} + 1$. The two values $E_{min} - 1$ and $E_{max} + 1$ are distinguished as follows. $E_{min} - 1$ indicates zero (both positive and negative sign) and a denormalized number, and $E_{max} + 1$ indicates positive or negative infinity or a non-number (NaN). Table 3.1 shows E_{min} and E_{max} values.

Table 3.1 Floating-Point Number Formats and Parameters

Parameter	Single-Precision	Double-Precision
Total bit width	32 bits	64 bits
Sign bit	1 bit	1 bit
Exponent field	8 bits	11 bits
Fraction field	23 bits	52 bits
Precision	24 bits	53 bits
Bias	+127	+1023
E_{max}	+127	+1023
E_{min}	–126	–1022

Floating-point number value v is determined as follows:

If $E = E_{max} + 1$ and $f \neq 0$, v is a non-number (NaN) irrespective of sign s
If $E = E_{max} + 1$ and $f = 0$, $v = (-1)^s$ (infinity) [positive or negative infinity]
If $E_{min} \leq E \leq E_{max}$, $v = (-1)^s 2^E$ (1.f) [normalized number]
If $E = E_{min} - 1$ and $f \neq 0$, $v = (-1)^s 2^{Emin}$ (0.f) [denormalized number]
If $E = E_{min} - 1$ and $f = 0$, $v = (-1)^s 0$ [positive or negative zero]

Table 3.2 shows the ranges of the various numbers in hexadecimal notation.

Table 3.2 Floating-Point Ranges

Type	Single-Precision	Double-Precision
Signaling non-number	H'7FFF FFFF to H'7FC0 0000	H'7FFF FFFF FFFF FFFF to H'7FF8 0000 0000 0000
Quiet non-number	H'7FBF FFFF to H'7F80 0001	H'7FF7 FFFF FFFF FFFF to H'7FF0 0000 0000 0001
Positive infinity	H'7F80 0000	H'7FF0 0000 0000 0000
Positive normalized number	H'7F7F FFFF to H'0080 0000	H'7FEF FFFF FFFF FFFF to H'0010 0000 0000 0000
Positive denormalized number	H'007F FFFF to H'0000 0001	H'000F FFFF FFFF FFFF to H'0000 0000 0000 0001
Positive zero	H'0000 0000	H'0000 0000 0000 0000
Negative zero	H'8000 0000	H'8000 0000 0000 0000
Negative denormalized number	H'8000 0001 to H'807F FFFF	H'8000 0000 0000 0001 to H'800F FFFF FFFF FFFF
Negative normalized number	H'8080 0000 to H'FF7F FFFF	H'8010 0000 0000 0000 to H'FFEF FFFF FFFF FFFF
Negative infinity	H'FF80 0000	H'FFF0 0000 0000 0000
Quiet non-number	H'FF80 0001 to H'FFBF FFFF	H'FFF0 0000 0000 0001 to H'FFF7 FFFF FFFF FFFF
Signaling non-number	H'FFC0 0000 to H'FFFF FFFF	H'FFF8 0000 0000 0000 to H'FFFF FFFF FFFF FFFF

3.2.2 Non-Numbers (NaN)

Figure 3.3 shows the bit pattern of a non-number (NaN). A value is NaN in the following case:

- Sign bit: Don't care
- Exponent field: All bits are 1
- Fraction field: At least one bit is 1

The NaN is a signaling NaN (sNaN) if the MSB of the fraction field is 1, and a quiet NaN (qNaN) if the MSB is 0.

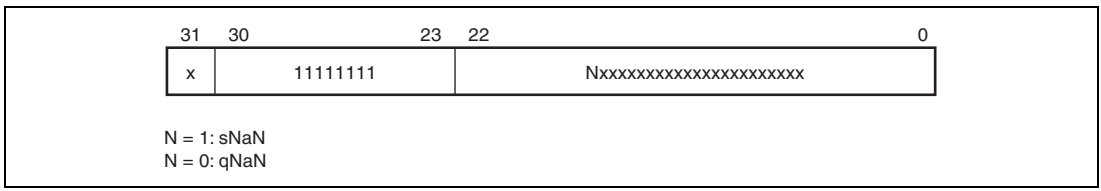


Figure 3.3 Single-Precision NaN Bit Pattern

An sNaN is input in an operation, except copy, FABS, and FNEG, that generates a floating-point value.

- When the EN.V bit in FPSCR is 0, the operation result (output) is a qNaN.
- When the EN.V bit in FPSCR is 1, an invalid operation exception will generate FPU exception processing. In this case, the contents of the operation destination register are unchanged.

If a qNaN is input in an operation that generates a floating-point value, and an sNaN has not been input in that operation, the output will always be a qNaN irrespective of the setting of the EN.V bit in FPSCR. An exception will not be generated in this case.

The qNAN values as operation results are as follows:

- Single-precision qNaN: H'7FBF FFFF
- Double-precision qNaN: H'7FF7 FFFF FFFF FFFF

See the individual instruction descriptions for details of floating-point operations when a non-number (NaN) is input.

3.2.3 Denormalized Numbers

For a denormalized number floating-point value, the exponent field is expressed as 0, and the fraction field as a non-zero value.

In the SH2A-FPU, the DN bit in the status register FPSCR is always set to 1, therefore a denormalized number (source operand or operation result) is always flushed to 0 in a floating-point operation that generates a value (an operation other than copy, FNEG, or FABS).

When the DN bit in FPSCR is 0, a denormalized number (source operand or operation result) is processed as it is. See the individual instruction descriptions for details of floating-point operations when a denormalized number is input.

3.3 Register Descriptions

3.3.1 Floating-Point Registers

Figure 3.4 shows the floating-point register configuration. There are sixteen 32-bit floating-point registers FPR0 to FPR15, referenced by specifying FR0 to FR15, DR0/2/4/6/8/10/12/14. The correspondence between FRPn and the reference name is determined by the PR and SZ bits in FPSCR. Refer to figure 3.4.

1. Floating-point registers, FPRi (16 registers)
   FPR0 to FPR15
2. Single-precision floating-point registers, FRi (16 registers)
   FR0 to FR15 indicate FPR0 to FPR15
3. Double-precision floating-point registers or single-precision floating-point vector registers in pairs, DRi (8 registers)
   A DR register comprises two FR registers.
   DR0 = {FR0, FR1}, DR2 = {FR2, FR3}, DR4 = {FR4, FR5}, DR6 = {FR6, FR7},
   DR8 = {FR8, FR9}, DR10 = {FR10, FR11}, DR12 = {FR12, FR13}, DR14 = {FR14, FR15}

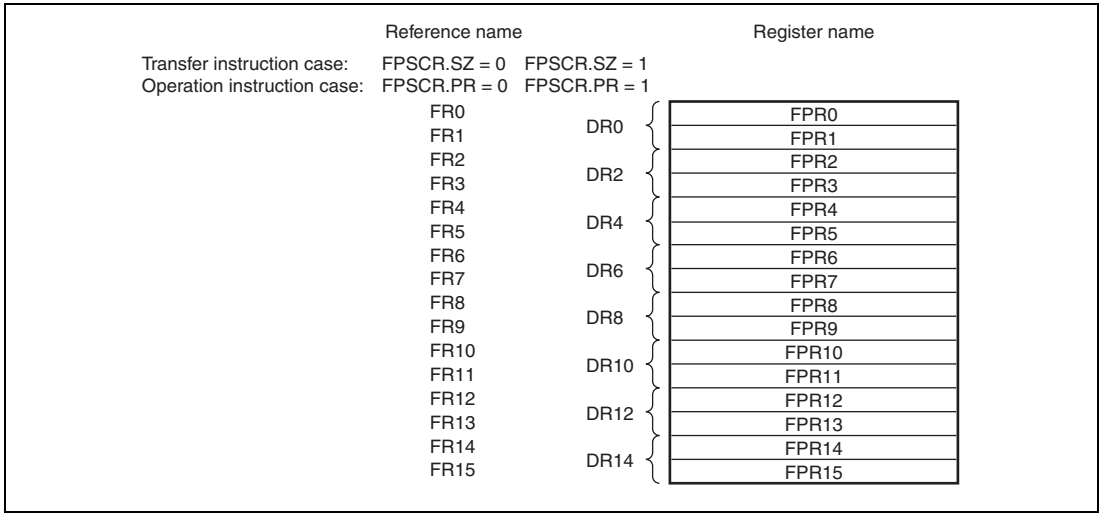


Figure 3.4 Floating-Point Registers

3.3.2 Floating-Point Status/Control Register (FPSCR)

FPSCR is a 32-bit register that controls floating-point instructions, sets FPU exceptions, and selects the rounding mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	QIS	-	SZ	PR	DN	Cause	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cause				Enable					Flag					RM1	RM0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22	QIS	0	R/W	Nonnunerical Processing Mode 0: Processes qNaN or ±∞ as such 1: Treats qNaN or ±∞ as the same as sNaN (valid only when FPSCR.Enable.V = 1)
21	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
20	SZ	0	R/W	Transfer Size Mode 0: Data size of FMOV instruction is 32-bits 1: Data size of FMOV instruction is a 32-bit register pair (64 bits)
19	PR	0	R/W	Precision Mode 0: Floating-point instructions are executed as single-precision operations 1: Floating-point instructions are executed as double-precision operations (graphics support instructions are undefined)
18	DN	1	R	Denormalization Mode (Always fixed to 1 in SH2A-FPU) 1: Denormalized number is treated as zero

Bit	Bit Name	Initial Value	R/W	Description
17 to 12	Cause	All 0	R/W	FPU Exception Cause Field FPU Exception Enable Field FPU Exception Flag Field Each time floating-point operation instruction is executed, the FPU exception cause field is cleared to 0 first. When an FPU exception on floating-point operation occurs, the bits corresponding to the FPU exception cause field and FPU exception flag field are set to 1. The FPU exception flag field remains set to 1 until it is cleared to 0 by software. As the bits corresponding to FPU exception enable filed are sets to 1, FPU exception processing occurs. For bit allocations of each field, see table 3.3.
11 to 7	Enable	All 0	R/W	
6 to 2	Flag	All 0	R/W	
1	RM1	0	R/W	Rounding Mode These bits select the rounding mode. 00: Round to Nearest 01: Round to Zero 10: Reserved 11: Reserved
0	RM0	1	R/W	

Table 3.3 Bit Allocation for FPU Exception Handling

Field Name		FPU Error (E)	Invalid Operation (V)	Division by Zero (Z)	Overflow (O)	Underflow (U)	Inexact (I)
Cause	FPU exception cause field	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12
Enable	FPU exception enable field	None	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7
Flag	FPU exception flag field	None	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2

Note: No FPU error occurs in the SH2A-FPU.

3.3.3 Floating-Point Communication Register (FPUL)

Information is transferred between the FPU and CPU via FPUL. FPUL is a 32-bit system register that is accessed from the CPU side by means of LDS and STS instructions. For example, to convert the integer stored in general register R1 to a single-precision floating-point number, the processing flow is as follows:

R1 → (LDS instruction) → FPUL → (single-precision FLOAT instruction) → FR1

3.4 Rounding

In a floating-point instruction, rounding is performed when generating the final operation result from the intermediate result. Therefore, the result of combination instructions such as FMAC will differ from the result when using a basic instruction such as FADD, FSUB, or FMUL. Rounding is performed once in FMAC, but twice in FADD, FSUB, and FMUL.

Which of the two rounding methods is to be used is determined by the RM bits in FPSCR.

FPSCR.RM[1:0] = 00: Round to Nearest
FPSCR.RM[1:0] = 01: Round to Zero

(1) Round to Nearest

The operation result is rounded to the nearest expressible value. If there are two nearest expressible values, the one with an LSB of 0 is selected.

If the unrounded value is $2^{Emax} (2 - 2^{-P})$ or more, the result will be infinity with the same sign as the unrounded value. The values of Emax and P, respectively, are 127 and 24 for single-precision, and 1023 and 53 for double-precision.

(2) Round to Zero

The digits below the round bit of the unrounded value are discarded.

If the unrounded value is larger than the maximum expressible absolute value, the value will become the maximum expressible absolute value.

3.5 FPU Exceptions

3.5.1 FPU Exception Sources

FPU exceptions may occur on floating-point operation instruction and the exception sources are as follows:

- FPU error (E): When FPSCR.DN = 0 and a denormalized number is input (No error occurs in the SH2A-FPU)
- Invalid operation (V): In case of an invalid operation, such as NaN input
- Division by zero (Z): Division with a zero divisor
- Overflow (O): When the operation result overflows
- Underflow (U): When the operation result underflows
- Inexact exception (I): When overflow, underflow, or rounding occurs

The FPU exception cause field in FPSCR contains bits corresponding to all of above sources E, V, Z, O, U, and I, and the FPU exception flag and enable fields in FPSCR contain bits corresponding to sources V, Z, O, U, and I, but not E. Thus, FPU errors cannot be disabled.

When an FPU exception occurs, the corresponding bit in the FPU exception cause field is set to 1, and 1 is added to the corresponding bit in the FPU exception flag field. When an FPU exception does not occur, the corresponding bit in the FPU exception cause field is cleared to 0, but the corresponding bit in the FPU exception flag field remains unchanged.

3.5.2 FPU Exception Handling

FPU exception handling is initiated in the following cases:

- FPU error (E): FPSCR.DN = 0 and a denormalized number is input (No error occurs in the SH2A-FPU)
- Invalid operation (V): FPSCR.Enable.V = 1 and invalid operation
- Division by zero (Z): FPSCR.Enable.Z = 1 and division with a zero divisor
- Overflow (O): FPSCR.Enable.O = 1 and instruction with possibility of operation result overflow
- Underflow (U): FPSCR.Enable.U = 1 and instruction with possibility of operation result underflow
- Inexact exception (I): FPSCR.Enable.I = 1 and instruction with possibility of inexact operation result

These possibilities of each exceptional handling on floating-point operation are shown in the individual instruction descriptions. All exception events that originate in the floating-point operation are assigned as the same FPU exceptional handling event. The meaning of an exception generated by floating-point operation is determined by software by reading from FPSCR and interpreting the information it contains. Also, the destination register is not changed when FPU exception handling operation occurs.

Except for the above, the FPU disables exception handling. In every processing, the bit corresponding to source V, Z, O, U, or I is set to 1, and a default value is generated as the operation result.

- Invalid operation (V): qNaN is generated as the result.
- Division by zero (Z): Infinity with the same sign as the unrounded value is generated.
- Overflow (O):
  When rounding mode = RZ, the maximum normalized number, with the same sign as the unrounded value, is generated.
  When rounding mode = RN, infinity with the same sign as the unrounded value is generated.
- Underflow (U):
  Zero with the same sign as the unrounded value is generated.
- Inexact exception (I): An inexact result is generated.

Section 4 Multi-Core Processor

This LSI includes two SH2A CPUs (CPU0 and CPU1). Dual CPUs provide this LSI with strong levels of performance (through distribution of load) and functionality (through distribution of functions) in processing, which cannot be achieved by a single CPU.

4.1 Features

- Synchronization control between CPUs

  Inter-processor interrupts support control of synchronization between the two CPUs (see section 7, Interrupt Controller (INTC)).
- Exclusive control for shared resources

  Semaphore control registers support exclusive control for shared resources.
- Floating-point unit (FPU), cache memory, and high-speed on-chip RAM are provided for each CPU

  Each CPU has its own FPU, cache memory, and high-speed on-chip RAM.
  The high-speed on-chip RAM can be configured as shared RAM space or as CPU-specific RAM space by enabling or disabling access from the other CPU (see section 29, On-Chip RAM).
- Power-down modes (see section 30, Power-Down Modes)

  To reduce power consumption, this LSI can move between dual-processor mode, where both CPUs are operating, single-processor mode, where one CPU is in the sleep state, and dual-sleep mode, where both CPUs are in the sleep state. By making transitions between these modes in accordance with the load, power consumption can be reduced while high performance is maintained.
- Multiple-internal-bus structure (see section 1, Overview)

  To prevent deterioration in performance due to both CPUs and the DMAC not being able to get bus mastership, multiple (four) internal buses are provided.

4.2 Register Descriptions

The following registers are provided for control of the multi-core processor.

Table 4.1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
CPU ID register	CPUIDR	R	*1	H'FFFC1404	32
Semaphore register 0	SEMR0	R*2/W	H'00	H'FFFC1E00	8
Semaphore register 1	SEMR1	R*2/W	H'00	H'FFFC1E04	8
Semaphore register 2	SEMR2	R*2/W	H'00	H'FFFC1E08	8
Semaphore register 3	SEMR3	R*2/W	H'00	H'FFFC1E0C	8
Semaphore register 4	SEMR4	R*2/W	H'00	H'FFFC1E10	8
Semaphore register 5	SEMR5	R*2/W	H'00	H'FFFC1E14	8
Semaphore register 6	SEMR6	R*2/W	H'00	H'FFFC1E18	8
Semaphore register 7	SEMR7	R*2/W	H'00	H'FFFC1E1C	8
Semaphore register 8	SEMR8	R*2/W	H'00	H'FFFC1E20	8
Semaphore register 9	SEMR9	R*2/W	H'00	H'FFFC1E24	8
Semaphore register 10	SEMR10	R*2/W	H'00	H'FFFC1E28	8
Semaphore register 11	SEMR11	R*2/W	H'00	H'FFFC1E2C	8
Semaphore register 12	SEMR12	R*2/W	H'00	H'FFFC1E30	8
Semaphore register 13	SEMR13	R*2/W	H'00	H'FFFC1E34	8
Semaphore register 14	SEMR14	R*2/W	H'00	H'FFFC1E38	8
Semaphore register 15	SEMR15	R*2/W	H'00	H'FFFC1E3C	8
Semaphore register 16	SEMR16	R*2/W	H'00	H'FFFC1E40	8
Semaphore register 17	SEMR17	R*2/W	H'00	H'FFFC1E44	8
Semaphore register 18	SEMR18	R*2/W	H'00	H'FFFC1E48	8
Semaphore register 19	SEMR19	R*2/W	H'00	H'FFFC1E4C	8
Semaphore register 20	SEMR20	R*2/W	H'00	H'FFFC1E50	8
Semaphore register 21	SEMR21	R*2/W	H'00	H'FFFC1E54	8
Semaphore register 22	SEMR22	R*2/W	H'00	H'FFFC1E58	8
Semaphore register 23	SEMR23	R*2/W	H'00	H'FFFC1E5C	8
Semaphore register 24	SEMR24	R*2/W	H'00	H'FFFC1E60	8

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Semaphore register 25	SEMR25	R*[2]/W	H'00	H'FFFC1E64	8
Semaphore register 26	SEMR26	R*[2]/W	H'00	H'FFFC1E68	8
Semaphore register 27	SEMR27	R*[2]/W	H'00	H'FFFC1E6C	8
Semaphore register 28	SEMR28	R*[2]/W	H'00	H'FFFC1E70	8
Semaphore register 29	SEMR29	R*[2]/W	H'00	H'FFFC1E74	8
Semaphore register 30	SEMR30	R*[2]/W	H'00	H'FFFC1E78	8
Semaphore register 31	SEMR31	R*[2]/W	H'00	H'FFFC1E7C	8

Notes: 1. The values H'10111000 and H'50110800, respectively, are read out in response to reading by CPU0 and CPU1.
2. After being read, the register is cleared to H'00.

4.2.1 CPU ID Register (CPUIDR)

The CPU ID register indicates the CPU number (CPU0 or CPU1).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	ID	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	*	R	Reserved A fixed value is read from this bit.
30	ID	*	R	Indicates the CPU number. “0” is read by CPU0, and “1” is read by CPU1.
29 to 0	—	*	R	Reserved A fixed value is read from these bits.

Note: * Overall values of H'10111000 and H'50110800, respectively, are read out in response to reading by CPU0 and CPU1.

4.2.2 Semaphore Registers 0 to 31 (SEMR0 to SEMR31)

Semaphore registers 0 to 31 (SEMR0 to SEMR31) support exclusive control for resource access by the two CPUs.

Access to SEMR0 to SEMR31 by a given CPU does not interfere with the operation of the other CPU or the DMAC.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	SEMF
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 1.
0	SEMF	0	R/W	Used to support exclusive control for the two CPUs. The value written to this bit is retained. If this bit is read, the value of this bit is read out to the CPU and this bit is cleared automatically.

4.3 Operation

4.3.1 Initializing This LSI

Use the following procedure to initialize this LSI. A sample program for the procedure is given in figure 4.1.

1. After exit from the power-on reset state, the two CPUs execute power-on reset exception handling. The CPUs should execute the same exception-handling routine.
2. In the power-on reset exception handling routine, each CPU should identify itself as CPU0 or CPU1 by reading out CPUIDR and testing the value of the ID bit. The value read from the ID bit is 0 for CPU0 and 1 for CPU1.
3. Each CPU then branches to the corresponding processing routine.

```
; In the power-on reset exception handling routine,
; read ID bit in CPUIDR and check the value.
    MOVI20    #H'FFFC1404, R0
    MOV.L     @R0, R1
    MOV.L     #H'40000000, R2
    AND       R2, R1
    CMP/EQ    R2, R1
    BF        CPU0_ROUTINE
    BRA       CPU1_ROUTINE
    NOP

; Processing routine for CPU0
CPU0_ROUTINE:
    :
    :

; Processing routine for CPU1
CPU1_ROUTINE:
    :
    :
```

Figure 4.1 Example of a Program for Initialization of This LSI

4.3.2 Exclusive Control for CPUs

(1) Using the Semaphore Registers for Exclusive Control of CPU Access to Resources

A procedure for exclusive control of resource access by the two CPUs is given below. A sample program for this procedure is shown in figure 4.2.

1. In the initialization routine for either of the CPUs, set all of the SEMF bits in SEMR0 to SEMR31 to 1 (this indicates that all resources are free).
2. For example, assume that SEMR0 is used for semaphore control of resource A and that CPU0 wants to use resource A. In this case, CPU0 should read the SEMF bit in SEMR0 repeatedly until the bit is read as 1
3. CPU0 recognizes that it has read 1 from the SEMF bit in SEMR0. This clears the SEMF bit to 0.
4. CPU0 then uses resource A. While resource A is in use by CPU0, CPU1 can only read 0 (resource A is in use) from the SEMF bit of SEMR0, and thus cannot use resource A.
5. After CPU0 has finished using resource A, it sets the SEMF bit in SEMR0 to 1 (resource A is free).

```
; Initialization routine

    Make initial settings

; 1. Initialize SEMR0 to SEMR31

    MOVI20    #H'FFFC1E00, R0
    MOV       #H'01, R1
    MOV.B     R1,@(H'000, R0) ; SEMR0.SEMF = 1
    MOV.B     R1,@(H'004, R0) ; SEMR1.SEMF = 1
    MOV.B     R1,@(H'008, R0) ; SEMR2.SEMF = 1
    MOV.B     R1,@(H'00C, R0) ; SEMR3.SEMF = 1
     :
     :
    MOV.B     R1,@(H'07C, R0) ; SEMR31.SEMF = 1

; 2. Read SEMR0
; 3. Make sure that 1 has been read from SEMF bit in SEMR0

     MOVI20   #H'FFFC1E00, R0
 LOOP:
    BLD.B     #0,@(H'000, R0)
    BF        LOOP

; 4. Use resource A

    :
    :

; 5. Set SEMF bit in SEMR0 to 1

    MOVI20    #H'FFFC1E00, R0
    MOV       #H'01, R1
    MOV.B     R1,@(H'000, R0)
```

Figure 4.2 Example of a Program for Exclusive Control

(2) Notes

As a general precaution in exclusive control, pay attention to ensuring that the system does not enter a deadlock. For example, the system enters a deadlock in the following case.

1. CPU0 is to use currently available resource A and thus reads 1 from the SEMF bit in SEMR0 (this operation clears the SEMF bit in SEMR0 to 0).
2. CPU1 is to use currently available resource B and thus reads 1 from the SEMF bit in SEMR1 (this operation clears the SEMF bit in SEMR1 to 0).
3. CPU0 is to use resource B and thus reads the SEMF bit in SEMR1, but it keeps reading the bit as 0 since it was cleared in step 2.
4. CPU1 is to use resource A and thus reads the SEMF bit in SEMR0, but it keeps reading the bit as 0 since it was cleared in step 1.

Section 5 Clock Pulse Generator (CPG)

This LSI has a clock pulse generator (CPG) that generates CPU0 clock ($I_0\phi$), CPU1 clock ($I_1\phi$), a peripheral clock (Pϕ), and a bus clock (Bϕ). The CPG consists of a crystal oscillator, PLL circuits, and divider circuits.

5.1 Features

- Four clock operating modes

  The mode is selected from among the four clock operating modes based on the frequency range to be used and the input clock type: the clock from the crystal resonator, the external clock or the clock for USB.
- Three clocks generated independently

  CPU0 clock ($I_0\phi$) and CPU1 clock ($I_1\phi$) for the CPU and cache, a peripheral clock (Pϕ) for peripheral modules, and a bus clock (Bϕ = CKIO) for the external bus interface can be generated independently.
- Frequency change function

  CPU0 clock, CPU1 clock, and peripheral clock frequencies can be changed independently using the PLL (phase locked loop) circuits and divider circuits within the CPG. Frequencies are changed by software using the settings of the frequency control registers 0 and 1 (FRQCR0 and FRQCR1).
- Power-down mode control

  The clock can be stopped and specific modules can be stopped using the module standby function in power-down modes. For details on clock control in power-down mode, see section 30, Power-Down Modes.

Figure 5.1 shows a block diagram of the clock pulse generator.

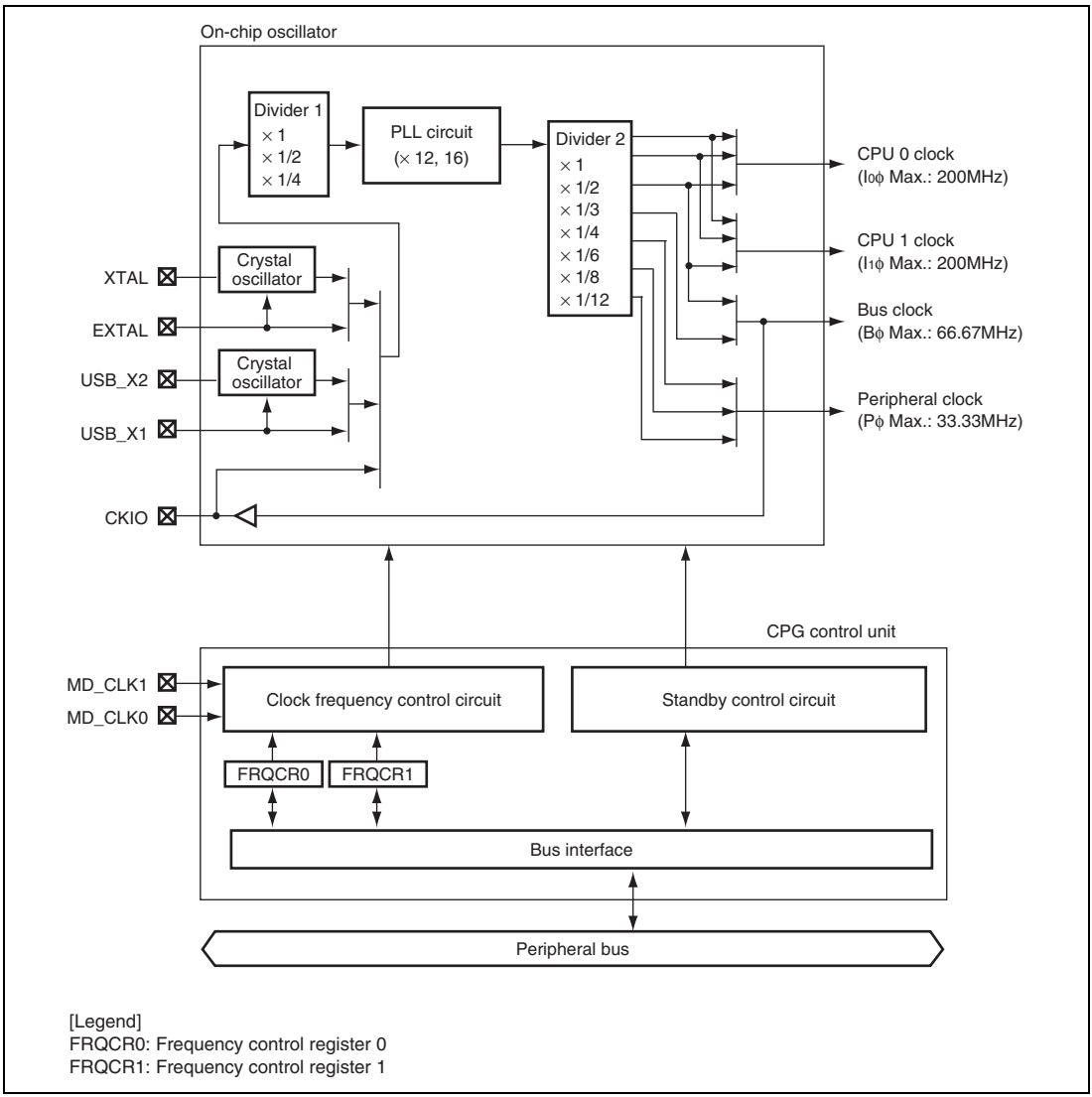


Figure 5.1 Block Diagram of Clock Pulse Generator

The clock pulse generator blocks function as follows:

(1) Crystal Oscillator

The crystal oscillator is used in which the crystal resonator is connected to the XTAL/EXTAL pin or USB_X1/USB_X2 pin. One of them is selected according to the clock operating mode.

(2) Divider 1

Divider 1 divides the output from the crystal oscillator or the external clock input. The division ratio depends on the clock operating mode.

(3) PLL Circuit

PLL circuit multiplies the frequency of the output from the divider 1. The multiplication ratio is set by the frequency control register.

(4) Divider 2

Divider 2 generates a clock signal whose operating frequency can be used for the CPU0 clock, CPU1 clock, peripheral clock, and bus clock. The division ratio of the CPU0 clock, CPU1 clock, and peripheral clock are set by the frequency control register. The division ratio of the bus clock is determined by the clock operating mode and the PLL multiplication ratio.

(5) Clock Frequency Control Circuit

The clock frequency control circuit controls the clock frequency using the MD_CLK0 and MD_CLK1 pins and the frequency control registers 0 and 1 (FRQCR0 and FRQCR1).

(6) Standby Control Circuit

The standby control circuit controls the states of the on-chip oscillation circuit and other modules in power-down modes.

In addition, the standby control register is provided to control power-down mode of other modules. For details on the standby control register, see section 30, Power-Down Modes.

(7) Frequency Control Register 0 (FRQCR0)

The frequency control register 0 (FRQCR0) has control bits assigned for the following functions: clock output/non-output from the CKIO pin during software standby mode, the frequency multiplication rate of the PLL circuit, and the frequency division ratio of the CPU0 clock ($I_0\phi$) and the peripheral clock ($P\phi$).

(8) Frequency Control Register 1 (FRQCR1)

The frequency control register 1 (FRQCR1) has control bits assigned for the following function: frequency division ratio of the CPU1 clock ($I_1\phi$).

5.2 Input/Output Pins

Table 5.1 lists the clock pulse generator pins and their functions.

Table 5.1 Pin Configuration and Functions of the Clock Pulse Generator

Pin Name	Symbol	I/O	Function (Clock Operating Mode 0, 1)	Function (Clock Operating Mode 2)	Function (Clock Operating Mode 3)
Mode control pins	MD_CLK0	Input	Sets clock operating mode.		
	MD_CLK1	Input	Sets clock operating mode.		
Crystal input/output pins (clock input pins)	XTAL	Output	Connected to the crystal resonator. (Leave this pin open when the crystal resonator is not in use.)	Leave this pin open.	Leave this pin open.
	EXTAL	Input	Connected to the crystal resonator or used to input external clock.	Fix this pin (pull up, pull down, connect to power supply, or connect to ground).	Fix this pin (pull up, pull down, connect to power supply, or connect to ground).
Clock input/output pin	CKIO	I/O	Clock output pin	Clock input pin	Clock output pin
Crystal input/output pins for USB (clock input pins)	USB_X1	Input	Connected to the crystal resonator to input the clock for USB only, or used to input external clock. When USB is not used, this pin should be fixed (pulled up, pulled down, connected to power supply, or connected to ground).	Connected to the crystal resonator to input the clock for USB only, or used to input external clock. When USB is not used, this pin should be fixed (pulled up, pulled down, connected to power supply, or connected to ground).	Connected to the crystal resonator to input the clock for both USB and the LSI, or used to input external clock.
	USB_X2	Output	Connected to the crystal resonator for USB. (Leave this pin open when the crystal resonator is not in use.)	Connected to the crystal resonator for USB. (Leave this pin open when the crystal resonator is not in use.)	Connected to the crystal resonator for both USB and the LSI. (Leave this pin open when the crystal resonator is not in use.)

5.3 Clock Operating Modes

Table 5.2 shows the relationship between the combinations of the mode control pins (MD_CLK1 and MD_CLK0) and the clock operating modes. Table 5.3 shows the usable frequency ranges in clock operating modes.

Table 5.2 Clock Operating Modes

	Pin Values		Clock I/O				
Mode	MD_CLK1	MD_CLK0	Source	Output	Divider 1	PLL Circuit On/Off	CKIO Frequency
0	0	0	EXTAL or crystal resonator	CKIO	1	ON (12, 16)	(EXTAL or crystal resonator) × 4
1	0	1	EXTAL or crystal resonator	CKIO	1/2	ON (12, 16)	(EXTAL or crystal resonator) × 2
2	1	0	CKIO	—	1/4	ON (12, 16)	(CKIO)
3	1	1	USB_X1 or crystal resonator	CKIO	1/4	ON (12, 16)	(USB_X1 or crystal resonator)

- Mode 0

  In mode 0, a clock is input from the EXTAL pin or the crystal oscillator. The PLL circuit shapes waveforms and the frequency is multiplied according to the frequency control register setting before the clock is supplied to the LSI. The oscillating frequency for the crystal resonator and EXTAL pin input clock ranges from 10 to 16.67 MHz. The frequency range of CKIO is from 40 to 66.66 MHz. The internal clock frequency is the EXTAL pin frequency multiplied by the frequency multiplication rate of the PLL circuit and the division ratio of the divider 2. To reduce current, fix the USB_X1 pin (pull up, pull down, connect to power supply, or connect to ground) and open the USB_X2 pin when USB is not used.
- Mode 1

  In mode 1, a clock is input from the EXTAL pin or the crystal oscillator. The PLL circuit shapes waveforms and the frequency is multiplied according to the frequency control register setting before the clock is supplied to the LSI. The oscillating frequency for the crystal resonator and EXTAL pin input clock ranges from 20 to 33.33 MHz. The frequency range of CKIO is from 40 to 66.66 MHz. The internal clock frequency is half the EXTAL pin frequency multiplied by the frequency multiplication rate of the PLL circuit and the division ratio of the divider 2. To reduce current, fix the USB_X1 pin (pull up, pull down, connect to power supply, or connect to ground) and open the USB_X2 pin when USB is not used.

- Mode 2

  In mode 2, the CKIO pin functions as an input pin and draws an external clock signal. The PLL circuit shapes waveforms and the frequency is multiplied according to the frequency control register setting before the clock is supplied to the LSI. The frequency range of CKIO is from 40 to 66.66 MHz. The internal clock frequency is quarter the CKIO pin frequency multiplied by the frequency multiplication rate of the PLL circuit and the division ratio of the divider 2. To reduce current, fix the EXTAL pin (pull up, pull down, connect to power supply, or connect to ground) and open the XTAL pin when the LSI is used in mode 2. When USB is not used, fix the USB_X1 pin (pull up, pull down, connect to power supply, or connect to ground) and open the USB_X2 pin.

- Mode 3

  In mode 3, a clock is input from the USB_X1 pin or the crystal oscillator. The external clock is input through this pin and a waveform is shaped in the PLL circuit. Then the frequency is multiplied according to the frequency control register setting before the clock is supplied to the LSI. The frequency of CKIO is 48 MHz (USB_X1/crystal resonator). The internal clock frequency is quarter the USB_X1 pin frequency multiplied by the frequency multiplication rate of the PLL circuit and the division ratio of the divider 2. To reduce current, fix the EXTAL pin (pull up, pull down, connect to power supply, or connect to ground) and open the XTAL pin when the LSI is used in mode 3. When the USB crystal resonator is not used, open the USB_X2 pin.

Table 5.3 Relationship between Clock Operating Mode and Frequency Range

Clock Operating Mode	FRQCR0 Register Setting*1	FRQCR1 Register Setting	PLL Multiplication Rate PLL Circuit	Ratio of Internal Clock Frequencies (I_0:I_1:B:P)*2	Selectable Frequency Range (MHz) Input Clock*3	CPU0 Clock ($I_0\phi$)	CPU1 Clock ($I_1\phi$)	Bus Clock (Bϕ = CKIO Pin)	Peripheral Clock (Pϕ)
0	H'x104	H'0000	ON (×12)	12:12:4:2	10 to 16.67	120 to 200	120 to 200	40 to 66.66	20 to 33.33
	H'x104	H'0020	ON (×12)	12:4:4:2	10 to 16.67	120 to 200	40 to 66.66	40 to 66.66	20 to 33.33
	H'x106	H'0000	ON (×12)	12:12:4:1	10 to 16.67	120 to 200	120 to 200	40 to 66.66	10 to 16.67
	H'x106	H'0020	ON (×12)	12:4:4:1	10 to 16.67	120 to 200	40 to 66.66	40 to 66.66	10 to 16.67
	H'x124	H'0000	ON (×12)	4:12:4:2	10 to 16.67	40 to 66.66	120 to 200	40 to 66.66	20 to 33.33
	H'x124	H'0020	ON (×12)	4:4:4:2	10 to 16.67	40 to 66.66	40 to 66.66	40 to 66.66	20 to 33.33
	H'x126	H'0000	ON (×12)	4:12:4:1	10 to 16.67	40 to 66.66	120 to 200	40 to 66.66	10 to 16.67
	H'x126	H'0020	ON (×12)	4:4:4:1	10 to 16.67	40 to 66.66	40 to 66.66	40 to 66.66	10 to 16.67
	H'x205	H'0000	ON (×16)	16:16:4:2	10 to 12.5	160 to 200	160 to 200	40 to 50	20 to 25
	H'x205	H'0010	ON (×16)	16:8:4:2	10 to 12.5	160 to 200	80 to 100	40 to 50	20 to 25
	H'x206	H'0000	ON (×16)	16:16:4:4/3	10 to 12.5	160 to 200	160 to 200	40 to 50	13.33 to 16.67
	H'x206	H'0010	ON (×16)	16:8:4:4/3	10 to 12.5	160 to 200	80 to 100	40 to 50	13.33 to 16.67
	H'x215	H'0000	ON (×16)	8:16:4:2	10 to 12.5	40 to 50	160 to 200	40 to 50	20 to 25
	H'x215	H'0010	ON (×16)	8:8:4:2	10 to 12.5	40 to 50	80 to 100	40 to 50	20 to 25
	H'x216	H'0000	ON (×16)	8:16:4:4/3	10 to 12.5	40 to 50	160 to 200	40 to 50	13.33 to 16.67
	H'x216	H'0010	ON (×16)	8:8:4:4/3	10 to 12.5	40 to 50	80 to 100	40 to 50	13.33 to 16.67
1	H'x104	H'0000	ON (×12)	6:6:2:1	20 to 33.33	120 to 200.0	120 to 200.0	40 to 66.66	20 to 33.33
	H'x104	H'0020	ON (×12)	6:2:2:1	20 to 33.33	120 to 200.0	40 to 66.66	40 to 66.66	20 to 33.33
	H'x106	H'0000	ON (×12)	6:6:2:1/2	20 to 33.33	120 to 200.0	120 to 200.0	40 to 66.66	10 to 16.67
	H'x106	H'0020	ON (×12)	6:2:2:1/2	20 to 33.33	120 to 200.0	40 to 66.66	40 to 66.66	10 to 16.67
	H'x124	H'0000	ON (×12)	2:6:2:1	20 to 33.33	40 to 66.66	120 to 200.0	40 to 66.66	20 to 33.33
	H'x124	H'0020	ON (×12)	2:2:2:1	20 to 33.33	40 to 66.66	40 to 66.66	40 to 66.66	20 to 33.33
	H'x126	H'0000	ON (×12)	2:6:2:1/2	20 to 33.33	40 to 66.66	120 to 200.0	40 to 66.66	10 to 16.67
	H'x126	H'0020	ON (×12)	2:2:2:1/2	20 to 33.33	40 to 66.66	40 to 66.66	40 to 66.66	10 to 16.67
	H'x205	H'0000	ON (×16)	8:8:2:1	20 to 25	160 to 200	160 to 200	40 to 50	20 to 25
	H'x205	H'0010	ON (×16)	8:4:2:1	20 to 25	160 to 200	80 to 100	40 to 50	20 to 25

Clock Operating Mode	FRQCR0 Register Setting*1	FRQCR1 Register Setting	PLL Multiplication Rate / PLL Circuit	Ratio of Internal Clock Frequencies (I_0:I_1:B:P)*2	Selectable Frequency Range (MHz)				
					Input Clock*3	CPU0 Clock ($I_0\phi$)	CPU1 Clock ($I_1\phi$)	Bus Clock (Bϕ = CKIO Pin)	Peripheral Clock (Pϕ)
1	H'x206	H'0000	ON (×16)	8:8:2:2/3	20 to 25	160 to 200	160 to 200	40 to 50	13.33 to 16.67
	H'x206	H'0010	ON (×16)	8:4:2:2/3	20 to 25	160 to 200	80 to 100	40 to 50	13.33 to 16.67
	H'x215	H'0000	ON (×16)	4:8:2:1	20 to 25	80 to 100	160 to 200	40 to 50	20 to 25
	H'x215	H'0010	ON (×16)	4:4:2:1	20 to 25	80 to 100	80 to 100	40 to 50	20 to 25
	H'x216	H'0000	ON (×16)	4:8:2:2/3	20 to 25	80 to 100	160 to 200	40 to 50	13.33 to 16.67
	H'x216	H'0010	ON (×16)	4:4:2:2/3	20 to 25	80 to 100	80 to 100	40 to 50	13.33 to 16.67
2	H'x104	H'0000	ON (×12)	3:3:1:1/2	40 to 66.66	120 to 200.0	120 to 200.0	40 to 66.66	20 to 33.33
	H'x104	H'0020	ON (×12)	3:1:1:1/2	40 to 66.66	120 to 200.0	40 to 66.66	40 to 66.66	20 to 33.33
	H'x106	H'0000	ON (×12)	3:3:1:1/4	40 to 66.66	120 to 200.0	120 to 200.0	40 to 66.66	10 to 16.67
	H'x106	H'0020	ON (×12)	3:1:1:1/4	40 to 66.66	120 to 200.0	40 to 66.66	40 to 66.66	10 to 16.67
	H'x124	H'0000	ON (×12)	1:3:1:1/2	40 to 66.66	40 to 66.66	120 to 200.0	40 to 66.66	20 to 33.33
	H'x124	H'0020	ON (×12)	1:1:1:1/2	40 to 66.66	40 to 66.66	40 to 66.66	40 to 66.66	20 to 33.33
	H'x126	H'0000	ON (×12)	1:3:1:1/4	40 to 66.66	40 to 66.66	120 to 200.0	40 to 66.66	10 to 16.67
	H'x126	H'0020	ON (×12)	1:1:1:1/4	40 to 66.66	40 to 66.66	40 to 66.66	40 to 66.66	10 to 16.67
	H'x205	H'0000	ON (×16)	4:4:1:1/2	40 to 50	160 to 200	160 to 200	40 to 50	20 to 25
	H'x205	H'0010	ON (×16)	4:2:1:1/2	40 to 50	160 to 200	80 to 100	40 to 50	20 to 25
	H'x206	H'0000	ON (×16)	4:4:1:1/4	40 to 50	160 to 200	160 to 200	40 to 50	13.33 to 16.67
	H'x206	H'0010	ON (×16)	4:2:1:1/4	40 to 50	160 to 200	80 to 100	40 to 50	13.33 to 16.67
	H'x215	H'0000	ON (×16)	2:4:1:1/2	40 to 50	80 to 100	160 to 200	40 to 50	20 to 25
	H'x215	H'0010	ON (×16)	2:2:1:1/2	40 to 50	80 to 100	80 to 100	40 to 50	20 to 25
	H'x216	H'0000	ON (×16)	2:4:1:1/4	40 to 50	80 to 100	160 to 200	40 to 50	13.33 to 16.67
	H'x216	H'0010	ON (×16)	2:2:1:1/4	40 to 50	80 to 100	80 to 100	40 to 50	13.33 to 16.67

Clock Operating Mode	FRQCR0 Register Setting*1	FRQCR1 Register Setting	PLL Multiplication Rate PLL Circuit	Ratio of Internal Clock Frequencies (I_0:I_1:B:P)*2	Selectable Frequency Range (MHz) Input Clock*3	CPU0 Clock ($I_0\phi$)	CPU1 Clock ($I_1\phi$)	Bus Clock (Bϕ = CKIO Pin)	Peripheral Clock (Pϕ)
3	H'x104	H'0000	ON (×12)	3:3:1:1/2	48	144	144	48	24
	H'x104	H'0020	ON (×12)	3:1:1:1/2	48	144	48	48	24
	H'x106	H'0000	ON (×12)	3:3:1:1/4	48	144	144	48	12
	H'x106	H'0020	ON (×12)	3:1:1:1/4	48	144	48	48	12
	H'x124	H'0000	ON (×12)	1:3:1:1/2	48	48	144	48	24
	H'x124	H'0020	ON (×12)	1:1:1:1/2	48	48	48	48	24
	H'x126	H'0000	ON (×12)	1:3:1:1/4	48	48	144	48	12
	H'x126	H'0020	ON (×12)	1:1:1:1/4	48	48	48	48	12
	H'x205	H'0000	ON (×16)	4:4:1:1/2	48	192	192	48	24
	H'x205	H'0010	ON (×16)	4:2:1:1/2	48	192	96	48	24
	H'x206	H'0000	ON (×16)	4:4:1:1/4	48	192	192	48	12
	H'x206	H'0010	ON (×16)	4:2:1:1/4	48	192	96	48	12
	H'x215	H'0000	ON (×16)	2:4:1:1/2	48	96	192	48	24
	H'x215	H'0010	ON (×16)	2:2:1:1/2	48	96	96	48	24
	H'x216	H'0000	ON (×16)	2:4:1:1/4	48	96	192	48	12
	H'x216	H'0010	ON (×16)	2:2:1:1/4	48	96	96	48	12

Notes: 1. x in the FRQCR0 register setting depends on the set value in bits 12, 13, and 14.
2. The ratio of clock frequencies, where the input clock frequency is assumed to be 1.
3. In mode 0 or 1, the frequency of the EXTAL pin input clock or the crystal resonator
In mode 2, the frequency of the CKIO pin input clock.
In mode 3, the frequency of the USB_X1 pin input clock or the crystal resonator

Cautions: Do not use this LSI for frequency settings other than those in table 5.3.

5.4 Register Descriptions

The clock pulse generator has the following registers.

Table 5.4 Register Configuration

Register Name	Abbreviation	R/W	Initial Value: Clock Modes 0, 1, 2	Initial Value: Clock Mode 3	Address	Access Size
Frequency control register	FRQCR0	R/W	H'0124	H'0215	H'FFFE0010	16
	FRQCR1	R/W	H'0020	H'0010	H'FFFE0012	16

5.4.1 Frequency Control Registers 0 and 1 (FRQCR0 and FRQCR1)

(1) FRQCR0

FRQCR0 is a 16-bit readable/writable register used to specify whether a clock is output from the CKIO pin during normal operation mode, changes in the multiplication rate of the PLL circuit, software standby mode, and standby mode cancellation. The register also specifies the frequency multiplication rate of the PLL circuit and the frequency division ratio for the CPU0 clock ($I_0\phi$) and peripheral clock ($P\phi$). The FRQCR0 register should be changed only from CPU0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	CKOEN2	CKOEN[1:0]		-	-	STC[1:0]		-	-	IFC[1:0]		-	PFC[2:0]		
Initial value:	0	0	0	0	0	0	0/1*	0/1*	0	0	0/1*	0/1*	0	1	0	0/1*
R/W:	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

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.

Bit	Bit Name	Initial Value	R/W	Description
14	CKOEN2	0	R/W	Clock Output Enable 2 Specifies whether the CKIO pin outputs clock signals or is fixed at low during a change of the frequency multiplication rate of the PLL circuit. If this bit is set to 1, the CKIO pin is fixed at low during a change of the frequency multiplication rate of the PLL circuit. Therefore, the malfunction of an external circuit caused by an unstable CKIO clock during a change of the frequency multiplication rate of the PLL circuit can be prevented. In clock operating mode 2, the CKIO pin functions as an input regardless of the value of these bits. 0: An unstable clock is output. 1: A low level is output.
13, 12	CKOEN [1:0]	00	R/W	Clock Output Enable Specifies whether the CKIO pin outputs a clock signal, or is tied to a fixed level or high impedance (Hi-Z) during normal operation, standby mode, and exit from standby mode. If these bits are set to 01, the CKIO pin is fixed at low during standby mode or cancellation of standby mode. Therefore, the malfunction of an external circuit caused by an unstable CKIO clock during exit from standby mode can be prevented. In clock operating mode 2, the CKIO pin functions as an input regardless of the value of these bits. In standby mode, the state in normal mode is retained. The settings are shown under the CKOEN[1:0] bits in table 5.5.
11, 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, 8	STC[1:0]	01/10*	R/W	Frequency Multiplication Rate of PLL Circuit • Clock modes 0, 1, and 2 00: Reserved (setting prohibited) 01: × 12 (initial value) 10: × 16 11: Reserved (setting prohibited) • Clock mode 3 00: Reserved (setting prohibited) 01: × 12 10: × 16 (initial value) 11: Reserved (setting prohibited)
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	IFC[1:0]	10/01*	R/W	Division Ratio of CPU0 Clock Frequency ($I_0\phi$) Specify the division ratio for the CPU0 clock, which is used in division of the output frequency of the PLL circuit. • Clock modes 0, 1, and 2 00: × 1 01: × 1/2 10: × 1/3 (initial value) 11: Reserved (setting prohibited) • Clock mode 3 00: × 1 01: × 1/2 (initial value) 10: × 1/3 11: Reserved (setting prohibited)
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 to 0	PFC[2:0]	100/ 101*	R/W	Peripheral Clock Frequency Division Ratio (Pϕ) Specify the division ratio for the peripheral clock, which is used in division of the output frequency of the PLL circuit. • Clock modes 0, 1, and 2 000: Reserved (setting prohibited) 001: Reserved (setting prohibited) 010: Reserved (setting prohibited) 011: Reserved (setting prohibited) 100: × 1/6 (initial value) 101: × 1/8 110: × 1/12 • Clock mode 3 000: Reserved (setting prohibited) 001: Reserved (setting prohibited) 010: Reserved (setting prohibited) 011: Reserved (setting prohibited) 100: × 1/6 101: × 1/8 (initial value) 110: × 1/12

Note: * The initial value depends on clock mode.

Table 5.5 CKOEN[1:0] Settings

Setting	Normal Operation	Software Standby Mode	Deep Standby Mode
00	Output	Output off (Hi-Z)	Low-level or high-level output
01	Output	Low-level output	Low-level or high-level output
10	Output	Output (unstable clock output)	Low-level or high-level output
11	Output off (Hi-Z)	Output off (Hi-Z)	Output off (Hi-Z)

(2) FRQCR1

FRQCR1 is a 16-bit readable/writable register used to specify the frequency division ratio of the CPU1 clock ($I_1\phi$). The FRQCR1 register should be changed only from CPU1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	IFC[1:0]		-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0/1*	0/1*	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	IFC[1:0]	10/01*	R/W	Division Ratio of CPU1 Clock Frequency ($I_1\phi$) Specify the division ratio for the CPU1 clock, which is used in division of the output frequency of the PLL circuit. • Clock modes 0, 1, and 2 00: × 1 01: × 1/2 10: × 1/3 (initial value) 11: Reserved (setting prohibited) • Clock mode 3 00: × 1 01: × 1/2 (initial value) 10: × 1/3 11: Reserved (setting prohibited)
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * The initial value depends on clock mode.

5.5 Changing the Frequency

The frequencies of the CPU0 clock ($I_0\phi$), CPU1 clock ($I_1\phi$), and peripheral clock ($P\phi$) can be changed either by changing the multiplication rate of the PLL circuit or by changing the division ratio of the divider. All of these are controlled by software through the frequency control registers 0 and 1 (FRQCR0 and FRQCR1). The methods are described below.

The multiplication rate and division ratio must be changed so that the register values satisfy the conditions shown in table 5.3. Otherwise, the operation is not guaranteed.

5.5.1 Changing the Multiplication Rate

An oscillation stabilization time is required when the multiplication rate of the PLL circuit is changed. On-chip WDT0 counts the stabilization time. The oscillation stabilization time becomes the same time as that of recovery from the software standby mode.

When changing the multiplication rate, after setting IFC1 and IFC0 in FRQCR1 to B'00 and specifying CPU1 not to be interrupted, execute the SLEEP command from CPU1, confirm that single processor 0 mode (CPU1 is in sleep mode) (for details, see section 30, Power-Down Modes), and perform the following procedure from CPU0.

1. In the initial state, the multiplication rate of the PLL circuit is 12 times in clock modes 0, 1, and 2 or 16 times in clock mode 3.
2. Set a value that will produce the specified oscillation stabilization time in WDT0 for CPU0 and stop WDT0. The following must be set:
   WTCSR0.TME = 0: WDT stops
   WTCSR0.CKS[2:0]: Division ratio for WDT counter clock
   WTCNT0 counter: Initial counter value
   (The WDT0 count is incremented using the clock after the setting.)
3. Set the desired value in the STC[1:0] bits of FRQCR0. The division ratios can also be set in the IFC[1:0] and PFC[2:0] of FRQCR0.
4. This LSI pauses temporarily and WDT0 starts to increment. A clock is supplied to the WDT only, and the other internal clocks all stop. The clock will continue to be output at the CKIO pin. Low level output can also be selected by setting CKOEN2 of FRQCR0.
   This state is the same as software standby mode. Whether or not registers are initialized depends on the module. For details, see section 32.3, Register States in Each Operating Mode.
5. Supply of the clock that has been set begins at WDT0 count overflow, and CPU0 of this LSI begins to operate again. WDT0 stops after it overflows. At this time, WOVF of WRCR0 is not set. The counter (WTCNT0) stops at H'00.

6. To change the WTCNT0 value after executing a frequency changing instruction, read WTCNT0 to make sure that it holds a value of H'00.
7. Since the CPU1 stays in the sleep state after execution of the frequency changing instruction, wake it up by using an interrupt or other means before using it.

Figure 5.2 shows a sample procedure for changing the multiplication rate of the PLL circuit from ×12 to ×16 in clock mode 0.

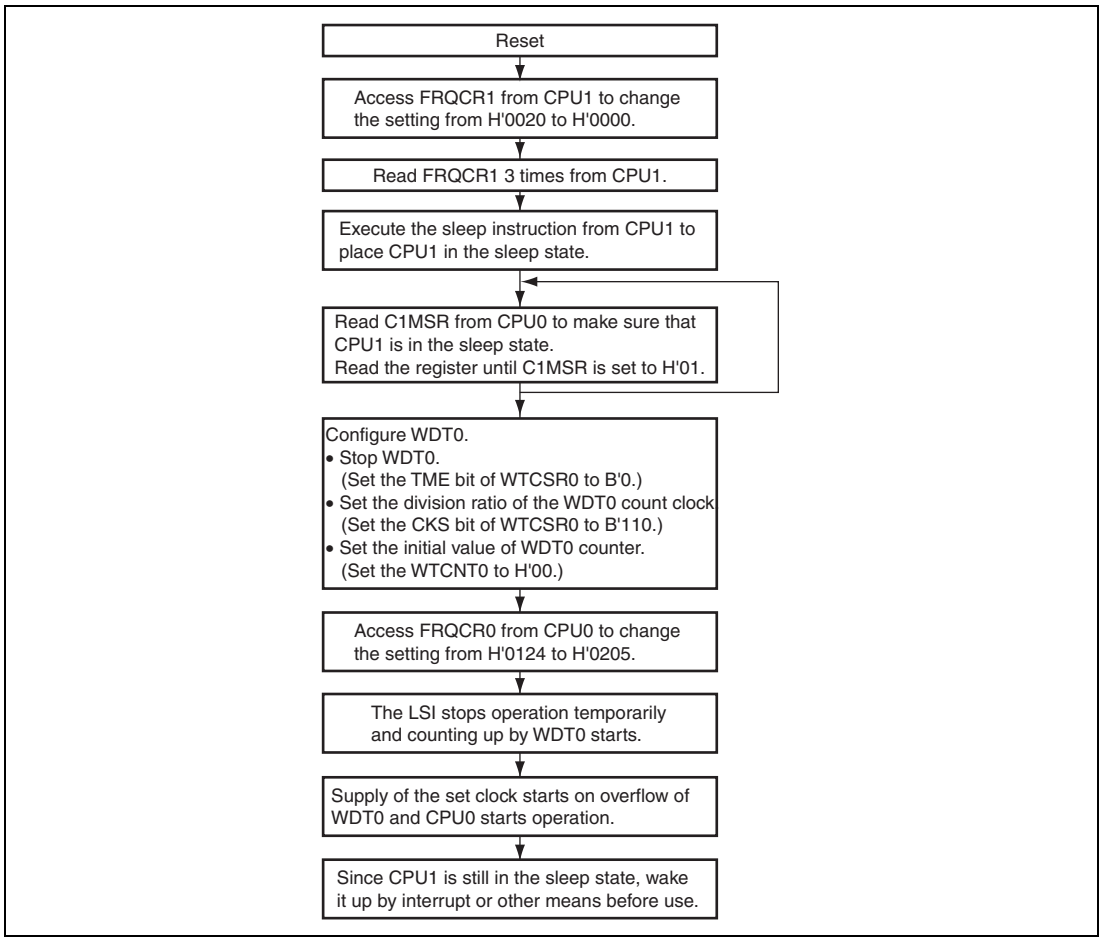


Figure 5.2 Sample Procedure for Changing the Multiplication Rate of the PLL Circuit from ×12 to ×16

5.5.2 Changing the Division Ratio

Counting by WDT0 does not proceed if the frequency division ratio is changed but the multiplication rate is not. However, when changing the division ratio of a peripheral clock, after specifying CPU1 not to be interrupted, place CPU1 in the sleep state and perform the operation from CPU0. When changing the division ratio of only the CPU1 clock, it is not necessary to place CPU1 in the sleep state.

- When the CPU0 clock is changed from CPU0
  1. The initial state depends on clock mode. See table 5.4.
  2. Set the desired values in the IFC1 and IFC0 bits and the PFC2 to PFC0 bits of FRQCR0. The values that can be set are limited by the clock operating mode and the multiplication rate of the PLL circuit. Note that if the wrong value is set, this LSI will malfunction.
  3. After the register bits (IFC[1:0] and PFC[2:0] of FRQCR0) have been set, the clock generated with the new division ratio is supplied.

- When the CPU1 clock is changed from CPU1
  1. The initial state depends on clock mode. See table 5.4.
  2. Set the desired value in IFC[1:0] of FRQCR1. The values that can be set are limited by clock operating mode and the multiplication rate of the PLL circuit. Note that if the wrong value is set, this LSI will malfunction.
  3. After the register bits (IFC[1:0] of FRQCR1) have been set, the clock generated with the new division ratio is supplied.

Note: When executing the SLEEP instruction after changing the frequency, read the frequency control register 0 (FRQCR0) or frequency control register 1 (FRQCR1) three times and then execute the SLEEP instruction.

5.5.3 Notes on Changing the Multiplication Rate and Division Ratio

1. When the division ratio for the CPU1 clock is changed, if IFC[1:0] of FRQCR1 are changed while CPU1 is in the sleep state, the change is not reflected. To prevent malfunction, always change the FRQCR1 register from CPU1.
2. When the multiplication rate or division ratio is changed through the frequency control registers 0 and 1 (FRQCR0 and FRQCR1) while the DMAC is transferring data, the DMA transfer is not guaranteed because the frequency is changed without waiting for the completion of the DMA transfer. Therefore, to change the multiplication rate or division ratio through the frequency control registers 0 and 1 (FRQCR0 and FRQCR1), wait for the completion of the DMA transfer or stop the DMA transfer and then change the frequency control registers 0 and 1 (FRQCR0 and FRQCR1).

5.6 Usage of the Clock Pins

For the connection of a crystal resonator or the input of a clock signal, this LSI circuit has the pins listed in table 5.6. With regard to these pins, take care on the following points. Furthermore, Xin pin and Xout pin are used in this section to refer to the pins listed in the table.

Table 5.6 Clock Pins

Xin Pins (Used for Connection of a Crystal Resonator and Input of External Clock Signals)	Xout Pins (Used for Connection of a Crystal Resonator)
EXTAL	XTAL
USB_X1	USB_X2
AUDIO_X1	AUDIO_X2
RTC_X1	RTC_X2

5.6.1 In the Case of Inputting an External Clock

An example of the connection of an external clock is shown in figure 5.3. In cases where the Xout pin is left open state, take the parasitic capacitance as less than 10 pF.

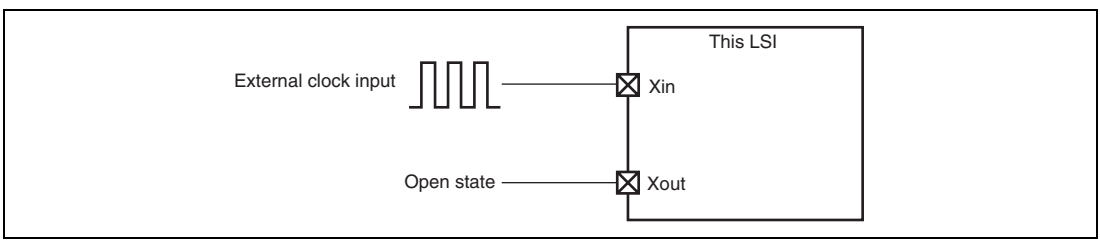


Figure 5.3 Example of the Connection of an External Clock

5.6.2 In the Case of Using a Crystal Resonator

An example of the connection of crystal resonator is shown in figure 5.4.

Place the crystal resonator and capacitors (CL1 and CL2) as close to pins Xin and Xout as possible. Furthermore, to avoid inductance so that oscillation is correct, use the points where the capacitors are connected to the crystal resonator in common and do not place wiring patterns close to these components.

Since the design of the user board is closely connected with the effective characteristics of the crystal resonator, refer to the example of connection of the crystal resonator that is introduced in this section and perform thorough evaluation on the user side as well. The rated value of the crystal resonator will vary with the floating capacitances and so on of the crystal resonator and mounted circuit, so proceed with decisions on the basis of full discussions with the maker of the crystal resonator. Ensure that voltages applied to the clock pins do not exceed the maximum rated values.

Although the feedback resistor is included in this LSI, an external feedback resistor may be required in some cases. This depends on the characteristics of the crystal resonator.

Set the parameters (of resistors and capacitors) with thorough evaluation on the user side.

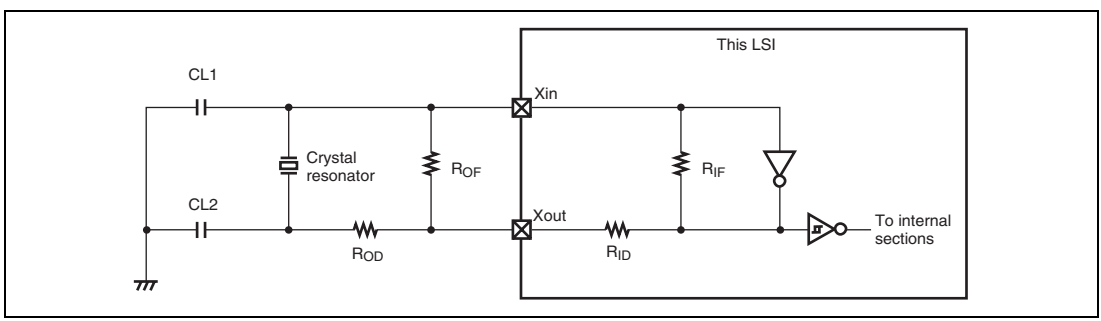


Figure 5.4 Example of the Connection of a Crystal Resonator

5.6.3 In the Case of Not Using the Clock Pin

In cases where the pins are not in use, fix the level on the Xin pin (pull it up or down, or connect it to the power-supply or ground level), and leave the Xout pin open state.

5.7 Oscillation Stabilizing Time

5.7.1 Oscillation Stabilizing Time of the On-chip Crystal Oscillator

In the case of using a crystal resonator, please wait longer than the oscillation stabilizing time at the following cases, to keep the oscillation stabilizing time of the on-chip crystal oscillator (In the case of inputting an external clock input, it is not necessary).

- Power on
- Canceling software standby mode or deep standby mode by using the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin
- Changing from halting oscillation to running oscillation by power-on reset or register setting (AUDIO_X1, RTC_X1)

5.7.2 Oscillation Stabilizing Time of the PLL circuit

In clock mode 0 or 1 the clock input on EXTAL, in clock mode 2 the clock input on CKIO, and in clock mode 3 the clock input on USB_X1 is supplied to the PLL circuit. So, regardless of whether using a crystal resonator or inputting an external clock from EXTAL (clock mode 0 and 1) or USB_X1 (clock mode 3), please wait longer than the oscillation stabilizing time at the following cases, to keep the oscillation stabilizing time of the PLL circuit.

- Power on (in the case of using the crystal resonator)/start inputting external clock (in the case of inputting the external clock)
- Canceling software standby mode or deep standby mode by using the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin
- Changing the multiplication ratio of the PLL circuit by power-on reset from $\overline{\text{RES}}$ pin

[Remarks]

The oscillation stabilizing time is kept by the counter running in the LSI at the following cases.

- Canceling software standby mode or deep standby mode by using the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin

5.8 Notes on Board Design

5.8.1 Note on Using a PLL Oscillation Circuit

In the PLLVcc and PLLVss 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.

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 PVcc should not supply the same resources on the board if at all possible.

Ensure that PLLVcc has the same electric potential as Vcc.

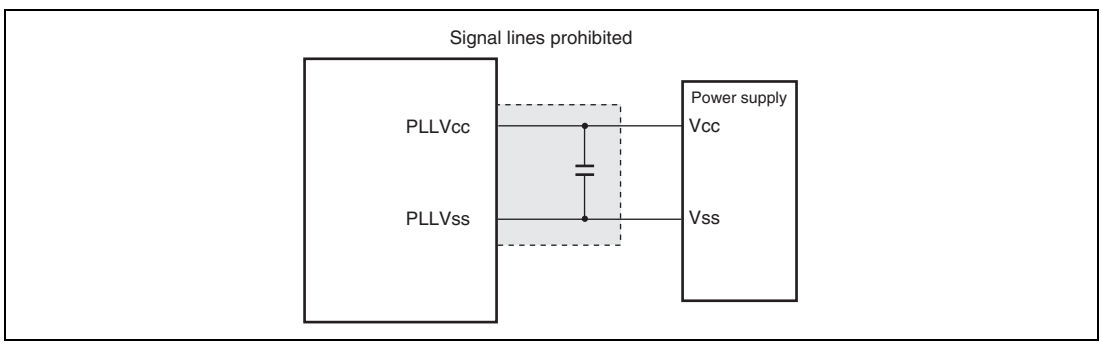


Figure 5.5 Note on Using a PLL Oscillation Circuit

Section 6 Exception Handling

6.1 Overview

6.1.1 Types of Exception Handling and Priority

Exception handling is started by sources, such as resets, address errors, register bank errors, interrupts, and instructions as shown in table 6.1. Since the exception sources have their priorities as shown in table 6.1, when multiple exception source events coincide, they are processed according to the priority shown below.

Table 6.1 Types of Exception Handling and Priority Order

Type	Exception Handling	Priority
Reset	Power-on reset	High
	Manual reset	
Address error	CPU address error	
Instruction	FPU exception	
	Integer division exception (division by zero)	
	Integer division exception (overflow)	
Register bank error	Bank underflow	
	Bank overflow	
Sleep error	Sleep error	
Interrupt	NMI	
	User break	
	H-UDI	
	Inter-processor	
	IRQ	
	PINT	
	On-chip peripheral modules	Low

Type	Exception Handling	Priority
Instruction	Trap instruction (TRAPA instruction)	High
	General illegal instructions (undefined code)	
	Slot illegal instructions (undefined code placed immediately after a delayed branch instruction*1, instruction that rewrites the PC*2, 32-bit instruction*3, RESBANK instruction, DIVS instruction, and DIVU instruction)	Low

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF

2. Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, JSR/N, RTV/N

3. 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR.B, BORNOT.B, BSET.B, BST.B, BXOR.B, MOV.B@disp12, MOV.W@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, MOVU.W

6.1.2 Exception Handling Operations

Exception sources are detected and their corresponding processing is started with the timing shown in table 6.2.

Table 6.2 Timing of Exception Source Detection and Start of Exception Handling

Exception Source		Timing of Source Detection and Start of Handling
Reset	Power-on reset	Starts on a low-to-high transition on the $\overline{\text{RES}}$ pin, when the H-UDI reset negate command is set after the H-UDI reset assert command has been set, or when the WDT overflows.
	Manual reset	Starts on a low-to-high transition on the $\overline{\text{MRES}}$ pin or when the WDT overflows.
Address error		Detected during decoding of an instruction and the handling starts when the execution of the previous instruction is completed.
Interrupts		
Register bank error	Bank underflow	Starts upon an attempt to execute a RESBANK instruction when saving to register bank has not been performed.
	Bank overflow	In the state where saving has been performed to all register bank areas, the handling starts when acceptance of register bank overflow exception has been set by the interrupt controller (the BOVE bit in IBNR of the INTC is 1) and an interrupt that uses a register bank has occurred and been accepted by the CPU.
Sleep error		Starts when the SLEEP instruction is executed by CPU0 when the sleep error enable bit (SLPERE) of standby control register 1 (STBCR1) is 1. For STBCR1, see section 30, Power-Down Modes.
Instructions	Trap instruction	Starts on the execution of a TRAPA instruction.
	General illegal instructions	Starts from the decoding of an undefined code placed anywhere except immediately after a delayed branch instruction (delay slot) (including an FPU instruction or FPU-related CPU instruction in FPU module standby state).
	Slot illegal instructions	Starts from the decoding of an undefined code, an instruction that rewrites the PC, a 32-bit instruction, the RESBANK instruction, the DIVS instruction, or the DIVU instruction placed directly after a delayed branch instruction (delay slot) (including an FPU instruction or FPU-related CPU instruction in FPU module standby state).
	Integer division exception	Starts upon detection of division-by-zero exception or overflow caused by division of the negative maximum value (H'80000000) by −1.

Exception Handling		Timing of Source Detection and Start of Handling
Instructions	FPU exception	Starts upon detection of invalid operation exception defined by IEEE standard 754, division-by-zero, overflow, underflow, or inexact exception. Also starts when qNaN or ±∞ is input to the source for a floating point operation instruction when the QIS bit in FPSCR is set.

When exception handling starts, the CPU operates as follows:

(1) Exception Handling Triggered by Reset

The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception handling vector table (PC and SP are respectively at the H'00000000 and H'00000004 addresses for power-on resets and the H'00000008 and H'0000000C addresses for manual resets). See section 6.1.3, Exception Handling Vector Table, for more information. The vector base register (VBR) is then initialized to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized to 0. The BN bit in IBNR of the interrupt controller (INTC) is also initialized to 0. The program begins running from the PC address fetched from the exception handling vector table.

(2) Exception Handling Triggered by Address Errors, Sleep Errors, Register Bank Errors, Interrupts, and Instructions

SR and PC are saved to the stack indicated by R15. In the case of interrupt exception handling other than NMI or UBC with usage of the register banks enabled, general registers R0 to R14, control register GBR, system registers MACH, MACL, and PR, and the vector table address offset of the interrupt exception handling to be executed are saved to the register banks. In the case of exception handling due to an address error, sleep error, register bank error, NMI interrupt, UBC interrupt, or instruction, saving to a register bank is not performed. When saving is performed to all register banks, automatic saving to the stack is performed instead of register bank saving. In this case, an interrupt controller setting must have been made so that register bank overflow exceptions are not accepted (the BOVE bit in IBNR of the INTC is 0). If a setting to accept register bank overflow exceptions has been made (the BOVE bit in IBNR of the INTC is 1), register bank overflow exception will be generated. In the case of interrupt exception handling, the interrupt priority level is written to the I3 to I0 bits in SR. In the case of exception handling due to an address error, sleep error, register bank error, or instruction, the I3 to I0 bits are not affected. The start address is then fetched from the exception handling vector table and the program begins running from that address.

6.1.3 Exception Handling Vector Table

Before exception handling begins running, the exception handling vector table must be set in memory. The exception handling vector table stores the start addresses of exception service routines. (The reset exception handling table holds the initial values of PC and SP.)

All exception sources are given different vector numbers and vector table address offsets, from which the vector table addresses are calculated. During exception handling, the start addresses of the exception service routines are fetched from the exception handling vector table, which is indicated by this vector table address.

Table 6.3 shows the vector numbers and vector table address offsets. Table 6.4 shows how vector table addresses are calculated.

Table 6.3 Exception Handling Vector Table

Exception Sources		Vector Numbers	Vector Table Address Offset	Receipt CPU Selection*[1]
Power-on reset	PC	0	H'00000000 to H'00000003	Both CPUs
	SP	1	H'00000004 to H'00000007	Both CPUs
Manual reset	PC	2	H'00000008 to H'0000000B	Both CPUs
	SP	3	H'0000000C to H'0000000F	Both CPUs
General illegal instruction		4	H'00000010 to H'00000013	Each CPU
(Reserved by system)		5	H'00000014 to H'00000017	—
Slot illegal instruction		6	H'00000018 to H'0000001B	Each CPU
(Reserved by system)		7	H'0000001C to H'0000001F	—
		8	H'00000020 to H'00000023	—
CPU address error		9	H'00000024 to H'00000027	Each CPU
(Reserved by system)		10	H'00000028 to H'0000002B	—
Interrupts	NMI	11	H'0000002C to H'0000002F	User selection A
	User break	12	H'00000030 to H'00000033	Each CPU
FPU exception		13	H'00000034 to H'00000037	Each CPU
H-UDI		14	H'00000038 to H'0000003B	User selection A
Bank overflow		15	H'0000003C to H'0000003F	Each CPU
Bank underflow		16	H'00000040 to H'00000043	Each CPU
Integer division exception (division by zero)		17	H'00000044 to H'00000047	Each CPU

Exception Sources	Vector Numbers	Vector Table Address Offset	Receipt CPU Selection*1
Integer division exception (overflow)	18	H'00000048 to H'0000004B	Each CPU
Sleep error	19	H'0000004C to H'0000004F	User selection A
(Reserved by system)	20	H'00000050 to H'00000053	—
Inter-processor interrupt (CPU0, CPU1)	21	H'00000054 to H'00000057	Each CPU
	:	:	
	28	H'00000070 to H'00000073	
(Reserved by system)	29	H'00000074 to H'00000077	—
	:	:	
	31	H'0000007C to H'0000007F	
Trap instruction (user vector)	32	H'00000080 to H'00000083	Each CPU
	:	:	
	63	H'000000FC to H'000000FF	
External interrupt (IRQ, PINT), on-chip peripheral module*2	64	H'00000100 to H'00000103	User selection B
	:	:	
	255	H'000003FC to H'000003FF	

Notes: 1. Both CPUs: Indicates that exception handling is performed on both CPU0 and CPU1.
Each CPU: Indicates that exception handling is performed on CPU0 when the exception source occurred on CPU0 or performed on CPU1 when the exception source occurred on CPU1.
User selection A: Indicates that only CPU0, only CPU1, or both CPU0 and CPU1 can be selected to perform exception handling.
User selection B: Indicates that only CPU0 or only CPU1 can be selected to perform exception handling.
2. The vector numbers and vector table address offsets for each external interrupt and on-chip peripheral module interrupt are given in table 7.8.

Table 6.4 Calculating Exception Handling Vector Table Addresses

Exception Source	Vector Table Address Calculation
Resets	Vector table address = (vector table address offset) = (vector number) × 4
Address errors, sleep errors, register bank errors, interrupts, instructions	Vector table address = VBR + (vector table address offset) = VBR + (vector number) × 4

Notes: 1. Vector table address offset: See table 6.3.
2. Vector number: See table 6.3.

6.2 Resets

6.2.1 Input/Output Pins

Table 6.5 shows the reset-related pin configuration.

Table 6.5 Pin Configuration

Pin Name	Symbol	I/O	Function
Power-on reset	$\overline{\text{RES}}$	Input	When this pin is driven low, this LSI shifts to the power-on reset processing.
Manual reset	$\overline{\text{MRES}}$	Input	When this pin is driven low, this LSI shifts to the manual reset processing.

6.2.2 Types of Reset

A reset is the highest-priority exception source. There are two kinds of reset, power-on and manual. As shown in table 6.6, the CPU state is initialized in both a power-on reset and a manual reset. On-chip peripheral module registers are also initialized by a power-on reset, but not by a manual reset.

Table 6.6 Reset States

	Conditions for Transition to Reset State					Internal States			
Type	$\overline{\text{RES}}$	H-UDI Command	$\overline{\text{MRES}}$	WDT0 Overflow	WDT1 Overflow	CPU	Other Modules	On-Chip High-Speed RAM	On-Chip Data Retention RAM
Power-on reset	Low	—	—	—	—	Initialized	Initialized	Initialized or retained contents*2	Initialized
	High	H-UDI reset assert command is set	—	—	—	Initialized	Initialized	Initialized or retained contents*2	Initialized
	High	Command other than H-UDI reset assert is set	—	Power-on reset	Power-on reset	Initialized	*1	Initialized or retained contents*2	Initialized
Manual reset	High	Command other than H-UDI reset assert is set	Low	—	—	Initialized	*1	Retained contents	Retained contents
	High	Command other than H-UDI reset assert is set	High	Manual reset	Manual reset	Initialized	*1	Retained contents	Retained contents

Notes: 1. See section 32.3, Register States in Each Operating Mode.
2. Data are retained when the setting of either the RAME or RAMWE bit is disabled.

6.2.3 Power-On Reset

(1) Power-On Reset by Means of $\overline{\text{RES}}$ Pin

When the $\overline{\text{RES}}$ pin is driven low, this LSI enters the power-on reset state. To reliably reset this LSI, the $\overline{\text{RES}}$ pin should be kept at the low level for the duration of the oscillation settling time at power-on or in software standby mode (when the clock is halted), or at least 20-tcyc when the clock is running. In the power-on reset state, the internal state of the CPU and all the on-chip peripheral module registers are initialized. See appendix A, Pin States, for the status of individual pins during the power-on reset state.

In the power-on reset state, power-on reset exception handling starts when the $\overline{\text{RES}}$ pin is first driven low for a fixed period and then returned to high. The CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized to 0. The BN bit in IBNR of the INTC is also initialized to 0.
4. The values fetched from the exception handling vector table are set in the PC and SP, and the program begins executing.

Be certain to always perform power-on reset processing when turning the system power on. For a recommended flow of power-on reset processing, see section 4, Multi-Core Processor.

(2) Power-On Reset by Means of H-UDI Reset Assert Command

When the H-UDI reset assert command is set, this LSI enters the power-on reset state. Power-on reset by means of an H-UDI reset assert command is equivalent to power-on reset by means of the $\overline{\text{RES}}$ pin. Setting the H-UDI reset negate command cancels the power-on reset state. The time required between an H-UDI reset assert command and H-UDI reset negate command is the same as the time to keep the $\overline{\text{RES}}$ pin low to initiate a power-on reset. In the power-on reset state generated by an H-UDI reset assert command, setting the H-UDI reset negate command starts power-on reset exception handling. The CPU operates in the same way as when a power-on reset was caused by the $\overline{\text{RES}}$ pin.

(3) Power-On Reset Initiated by WDT

Each CPU has a watchdog timer (WDT).

When either or both of the WDTs are set so that a power-on reset occurs in watchdog timer mode, and the WTCNT (or WTCNTs) of the WDT (or WDTs) overflows, this LSI enters the power-on reset state.

In this case, WRCSR of the WDT and FRQCR of the CPG are not initialized by the reset signal generated by the WDT.

If a reset caused by the $\overline{\text{RES}}$ pin or the H-UDI reset assert command occurs simultaneously with a reset caused by WDT overflow, the reset caused by the $\overline{\text{RES}}$ pin or the H-UDI reset assert command has priority, and the WOVF bit in WRCSR is cleared to 0. When power-on reset exception handling is started by the WDT, the CPU operates in the same way as when a power-on reset was caused by the $\overline{\text{RES}}$ pin.

6.2.4 Manual Reset

(1) Manual Reset by Means of $\overline{\text{MRES}}$ Pin

When the $\overline{\text{MRES}}$ pin is driven low, this LSI enters the manual reset state. To reset this LSI without fail, the $\overline{\text{MRES}}$ pin should be kept at the low level for at least 20-tcyc. In the manual reset state, the CPU's internal state is initialized, but the on-chip peripheral module registers are not initialized. In the manual reset state, manual reset exception handling starts when the $\overline{\text{MRES}}$ pin is first driven low for a fixed period and then returned to high. The CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized to 0. The BN bit in IBNR of the INTC is also initialized to 0.
4. The values fetched from the exception handling vector table are set in the PC and SP, and the program begins executing.

(2) Manual Reset Initiated by WDT

Each CPU has a watchdog timer (WDT).

When either or both of the WDTs are set so that a manual reset occurs in watchdog timer mode, and the WTCNT (or WTCNTs) of the WDT (or WDTs) overflows, this LSI enters the manual reset state.

When manual reset exception handling is started by the WDT, the CPU operates in the same way as when a manual reset was caused by the $\overline{\text{MRES}}$ pin.

(3) Note in Manual Reset

When a manual reset is generated, the bus cycle is retained, but if a manual reset occurs during DMAC burst transfer, manual reset exception handling will be deferred until the CPU acquires the bus mastership. The CPU and the BN bit in IBNR of the INTC are initialized by a manual reset. The FPU and other modules are not initialized.

6.3 Address Errors

6.3.1 Address Error Sources

Address errors occur when instructions are fetched or data read or written, as shown in table 6.7.

Table 6.7 Bus Cycles and Address Errors

Bus Cycle Type	Bus Cycle Bus Master	Bus Cycle Description	Address Errors
Instruction fetch	CPU	An instruction is fetched from an even address.	None (normal)
		An instruction is fetched from an odd address.	Address error
		An instruction is fetched from other than H'F0000000 to H'F5FFFFFF in the cache address array space and the reserved area.*	None (normal)
		An instruction is fetched from H'F0000000 to H'F5FFFFFF in the cache address array space and the reserved area. *	Address error
Data read/write	CPU	Word data is accessed from an even address.	None (normal)
		Word data is accessed from an odd address.	Address error
		Longword data is accessed from a longword boundary.	None (normal)
		Longword data is accessed from other than a long-word boundary.	Address error
		Double longword data accessed from a double longword boundary	None (normal)
		Double longword data accessed from other than a double longword boundary	Address error
		Byte or word data is accessed in on-chip peripheral module space*.	None (normal)
		Longword data is accessed in 16-bit on-chip peripheral module space*.	None (normal)
		Longword data is accessed in 8-bit on-chip peripheral module space*.	None (normal)

Note: * For details on the cache address array space and on-chip peripheral module space, see section 10, Bus State Controller (BSC).

6.3.2 Address Error Exception Handling

When an address error occurs, the bus cycle in which the address error occurred ends*, the executing instruction finishes, and address error exception handling starts. The CPU operates as follows:

1. The exception service routine start address which corresponds to the address error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
4. After jumping to the address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

Note: * In the case of address error related to data read/write. In the case of address error related to instruction fetch, if the bus cycle in which the address error occurred doesn't end until the entire three above-mentioned operations end, the CPU will start address error exception handling again until the bus cycle in which the address error occurred ends.

6.4 Register Bank Errors

6.4.1 Register Bank Error Sources

(1) Bank Overflow

In the state where saving has already been performed to all register bank areas, bank overflow occurs when acceptance of register bank overflow exception has been set by the interrupt controller (the BOVE bit in IBNR of the INTC is set to 1) and an interrupt that uses a register bank has occurred and been accepted by the CPU.

(2) Bank Underflow

Bank underflow occurs when an attempt is made to execute an RESBANK instruction while saving has not been performed to register banks.

6.4.2 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. The CPU operates as follows:

1. The exception service routine start address which corresponds to the register bank error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction for a bank overflow, and the start address of the executed RESBANK instruction for a bank underflow.

   To prevent multiple interrupts from occurring at a bank overflow, the interrupt priority level that caused the bank overflow is written to the interrupt mask level bits (I3 to I0) of the status register (SR).
4. After jumping to the address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

6.5 Sleep Errors

6.5.1 Sleep Error Source

A sleep error occurs if issuance of the sleep instruction by CPU0 is detected when the sleep error occurrence notification for CPU0 is set and the sleep error enable bit (SLPERE) of the standby control register 1 (STBCR1) is 1. For details, see section 30, Power-Down Modes.

6.5.2 Sleep Error Exception Handling

When a sleep error occurs, sleep error exception handling starts after the bus cycle in which the sleep error occurred ends and the execution of the current instruction is completed. The CPU operates as follows:

1. The exception service routine start address which corresponds to the sleep error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
4. After jumping to the address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.
5. Clear the sleep error enable bit (SLPERE) of the standby control register 1 (STBCR1) to 0 in the sleep error exception handling routine.

To detect a sleep error again, set the sleep error enable bit of the standby control register 1 to 1 after the corresponding sleep instruction of CPU0.

6.6 Interrupts

6.6.1 Interrupt Sources

The interrupt sources that trigger interrupt exception handling are NMI, user breaks, H-UDI, inter-processor interrupts, IRQ, PINT, and on-chip peripheral modules.

Each interrupt source is assigned a different vector number and vector table offset. See table 7.8, for more information on vector numbers and vector table address offsets.

6.6.2 Interrupt Priority Level

The interrupt priority order is predetermined. When multiple interrupts occur simultaneously (overlap), the interrupt controller (INTC) determines their relative priorities and starts exception handling according to the results.

The priority order of interrupts is expressed as priority levels 0 to 16, with priority 0 the lowest and priority 16 the highest. The NMI interrupt has priority 16 and cannot be masked, so it is always accepted. The user break interrupt and H-UDI interrupt priority level is 15. The inter-processor interrupt has an interrupt priority level of 15 to 8 according to the interrupt source. Priority levels of IRQ interrupts, PINT interrupts, and on-chip peripheral module interrupts can be set freely using the interrupt priority registers of the INTC (table 6.8). The priority levels that can be set are 0 to 15. Level 16 cannot be set. See section 7.3.1, Interrupt Priority Registers 01, 02, 05 to 21 (C0IPR01, C0IPR02, C0IPR05 to C0IPR21, C1IPR01, C1IPR02, C1IPR05 to C1IPR21), for details of the interrupt priority registers.

Table 6.8 Interrupt Priority Order

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be masked.
User break	15	Fixed priority level
H-UDI	15	Fixed priority level
Inter-processor interrupt	15 to 8	Fixed priority level
IRQ	0 to 15	Set by interrupt priority registers
PINT		
On-chip peripheral module		

6.6.3 Interrupt Exception Handling

When an interrupt occurs, its priority level is determined by the interrupt controller (INTC). NMI is always accepted, but other interrupts are only accepted if they have a priority level higher than the priority level set in the interrupt mask level bits (I3 to I0) of the status register (SR).

When an interrupt is accepted, interrupt exception handling begins. In interrupt exception handling, the CPU fetches the exception service routine start address which corresponds to the accepted interrupt from the exception handling vector table, and saves the SR and the program counter (PC) to the stack. In the case of interrupt exception handling other than NMI or UBC with usage of the register banks enabled, general registers R0 to R14, control register GBR, system registers MACH, MACL, and PR, and the vector table address offset of the interrupt exception handling to be executed are saved in the register banks. In the case of exception handling due to an address error, NMI interrupt, UBC interrupt, or instruction, saving is not performed to the register banks. If saving has been performed to all register banks (0 to 14), automatic saving to the stack is performed instead of register bank saving. In this case, an interrupt controller setting must have been made so that register bank overflow exceptions are not accepted (the BOVE bit in IBNR of the INTC is 0). If a setting to accept register bank overflow exceptions has been made (the BOVE bit in IBNR of the INTC is 1), register bank overflow exception occurs. Next, the priority level value of the accepted interrupt is written to the I3 to I0 bits in SR. For NMI, however, the priority level is 16, but the value set in the I3 to I0 bits is H'F (level 15). Then, after jumping to the start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch. See section 7.6, Operation, for further details of interrupt exception handling.

6.7 Exceptions Triggered by Instructions

6.7.1 Types of Exceptions Triggered by Instructions

Exception handling can be triggered by the trap instruction, slot illegal instructions, general illegal instructions, integer division exceptions, and FPU exceptions, as shown in table 6.9.

Table 6.9 Types of Exceptions Triggered by Instructions

Type	Source Instruction	Comment
Trap instruction	TRAPA	
Slot illegal instructions	Undefined code placed immediately after a delayed branch instruction (delay slot) (including an FPU instruction or FPU-related CPU instruction in FPU module standby state), instructions that rewrite the PC, 32-bit instructions, RESBANK instruction, DIVS instruction, and DIVU instruction	Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, JSR/N, RTV/N 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR.B, BORNOT.B, BSET.B, BST.B, BXOR.B, MOV.B@disp12, MOV.W@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, MOVU.W
General illegal instructions	Undefined code anywhere besides in a delay slot (including an FPU instruction or FPU-related CPU instruction in FPU module standby state)	
Integer division exceptions	Division by zero	DIVU, DIVS
	Negative maximum value ÷ (–1)	DIVS
FPU exceptions	Instruction causing invalid operation defined by IEEE754 standard or, division-by-zero exception, instruction causing overflow, underflow, or inexact exception.	FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, FSQRT

6.7.2 Trap Instruction

When a TRAPA instruction is executed, trap instruction exception handling starts. The CPU operates as follows:

1. The exception service routine start address which corresponds to the vector number specified in the TRAPA instruction is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the TRAPA instruction.
4. After jumping to the address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

6.7.3 Slot Illegal Instructions

An instruction placed immediately after a delayed branch instruction is said to be placed in a delay slot. When the instruction placed in the delay slot is undefined code, an instruction that rewrites the PC, a 32-bit instruction, an RESBANK instruction, a DIVS instruction, or a DIVU instruction, slot illegal exception handling starts when such kind of instruction is decoded. In slot illegal instruction exception handling, the CPU operates as follows:

1. The exception service routine start address is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the jump address of the delayed branch instruction immediately before the undefined code, the instruction that rewrites the PC, the 32-bit instruction, the RESBANK instruction, the DIVS instruction, or the DIVU instruction.
4. After jumping to the address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

6.7.4 General Illegal Instructions

When an undefined code placed anywhere other than immediately after a delayed branch instruction, i.e., in a delay slot, is decoded, general illegal instruction exception handling starts. In general illegal instruction exception handling, the CPU handles general illegal instructions in the same way as slot illegal instructions. Unlike processing of slot illegal instruction exception handling, however, the program counter value stored is the start address of the undefined code.

6.7.5 Integer Division Exceptions

When an integer division instruction performs division by zero or the result of integer division overflows, integer division instruction exception handling starts. The instructions that may become the source of division-by-zero exception are DIVU and DIVS. The only source instruction of overflow exception is DIVS, and overflow exception occurs only when the negative maximum value is divided by −1. The CPU operates as follows:

1. The exception service routine start address which corresponds to the integer division exception that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the integer division instruction at which the exception occurred.
4. After jumping to the address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

6.7.6 FPU Exceptions

An FPU exception handling is generated when the V, Z, O, U or I bit in the FPU enable field (Enable) of the floating point status register (FPSCR) is set. This indicates the occurrence of an invalid operation exception defined by the IEEE standard 754, a division-by-zero exception, overflow (in the case of an instruction for which this is possible), underflow (in the case of an instruction for which this is possible), or inexact exception (in the case of an instruction for which this is possible).

The floating-point operation instructions that may cause generation of an FPU exception handling are FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, and FSQRT.

An FPU exception handling is generated only when the corresponding FPU exception enable bit (enabled) is set. When the FPU detects an exception source by a floating-point operation, FPU operation is halted and FPU exception handling generation is reported to the CPU. When exception handling is started, the CPU operations are as follows.

1. The start address of the exception service routine corresponding to the FPU exception handling that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.

4. After jumping to the address fetched from the exception handling vector table, program execution starts. This jump is not a delayed branch.

The FPU exception flag field (Flag) of FPSCR is always updated regardless of whether or not an FPU exception handling has been accepted, and remains set until explicitly cleared by the user through an instruction. The FPU exception source field (Cause) of FPSCR changes each time a floating-point operation is executed. When the V bit in the FPU exception enable field (Enable) of FPSCR is set and the QIS bit in FPSCR is also set, FPU exception handling is generated when qNaN or ±∞ is input to a floating point operation instruction source.

6.8 When Exception Sources Are Not Accepted

When an address error, sleep error, FPU exception, register bank error (overflow), or interrupt is generated immediately after a delayed branch instruction, it is sometimes not accepted immediately but stored instead, as shown in table 6.10. When this happens, it will be accepted when an instruction that can accept the exception is decoded.

Table 6.10 Exception Source Generation Immediately after Delayed Branch Instruction

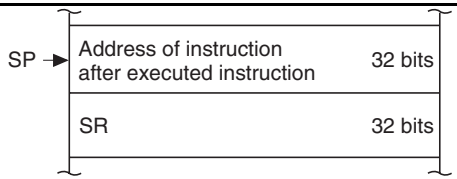
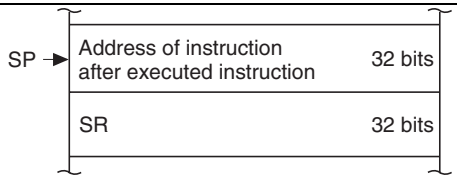
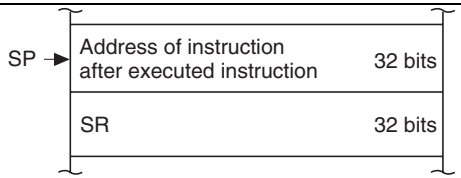
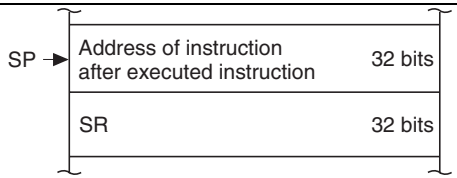
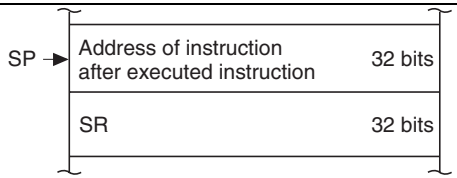
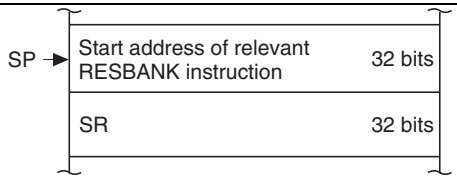
	Exception Source				
Point of Occurrence	**Address Error**	**Sleep Error**	**FPU Exception**	**Register Bank Error (Overflow)**	**Interrupt**
Immediately after a delayed branch instruction*	Not accepted	Not accepted	Not accepted	Not accepted	Not accepted

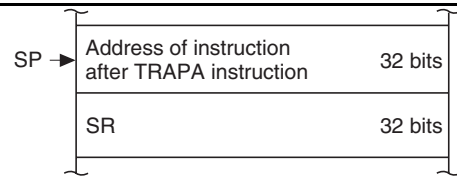
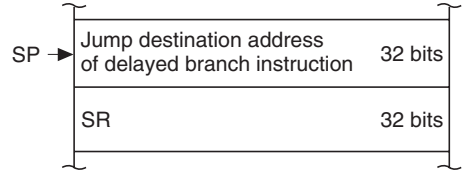
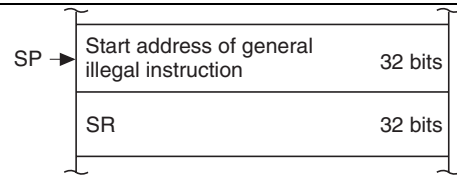
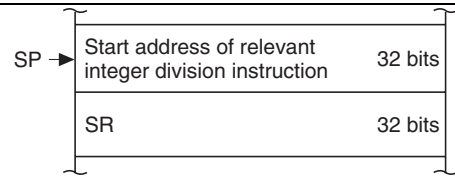
Note: * Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF

6.9 Stack Status after Exception Handling Ends

The status of the stack after exception handling ends is as shown in table 6.11.

Table 6.11 Stack Status after Exception Handling Ends

Exception Type	Stack Status
Address error	
Interrupt	
Sleep error	
FPU exception	
Register bank error (overflow)	
Register bank error (underflow)	

Exception Type	Stack Status
Trap instruction	
Slot illegal instruction	
General illegal instruction	
Integer division exception	

6.10 Usage Notes

6.10.1 Value of Stack Pointer (SP)

The value of the stack pointer must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception handling.

6.10.2 Value of Vector Base Register (VBR)

The value of the vector base register must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception handling.

6.10.3 Address Errors Caused by Stacking of Address Error Exception Handling

When the stack pointer is not a multiple of four, an address error will occur during stacking of the exception handling (interrupts, etc.) and address error exception handling will start up as soon as the first exception handling is ended. Address errors will then also occur in the stacking for this address error exception handling. To ensure that address error exception handling does not go into an endless loop, no address errors are accepted at that point. This allows program control to be shifted to the address error exception service routine and enables error processing.

When an address error occurs during exception handling stacking, the stacking bus cycle (write) is executed. During stacking of the status register (SR) and program counter (PC), the SP is decremented by 4 for both, so the value of SP will not be a multiple of four after the stacking either. The address value output during stacking is the SP value, so the address where the error occurred itself is output. This means the write data stacked will be undefined.

6.10.4 Interrupt Control via Modification of Interrupt Mask Bits

When enabling interrupts by changing the Interrupt Mask bits (I3-I0) of the Status Register (SR) using the LDC or LDC.L instructions, interrupts might not be accepted during the execution of the 5 instructions immediately after the LDC/LDC.L instruction.

Therefore, when enabling/disabling interrupts by changing the Interrupt Mask bits (I3-I0) of the Status Register (SR) using LDC/LDC.L instructions, please place at least 5 instructions between the interrupt-enable instruction and the interrupt-disable instruction.

6.10.5 Note before Exception Handling Begins Running

Before exception handling begins running, the exception handling vector table must be stored in a memory, and the CPU must be able to access the memory. So, if the exception handling is generated

- Ex. 1: when the exception handling vector table is stored in an external address space, but the settings of bus state controller and general I/O ports to access the external address space have been not completed yet, or
- Ex. 2: when the exception handling vector table is stored in the on-chip RAM, but the vector base register (VBR) has been not changed to the on-chip RAM address yet,

the CPU fetches an unintended value as the execution start address, and starts executing programs from unintended address.

Section 7 Interrupt Controller (INTC)

The interrupt controller (INTC) identifies the priorities of interrupt sources and controls interrupt requests to the CPU. The INTC has registers used to set interrupt priorities; interrupt requests are processed according to the priorities set in these registers by the user.

7.1 Features

- 16 levels of interrupt priority can be set.

  By setting 19 interrupt priority registers, the priorities of IRQ interrupts, PINT interrupts, and on-chip peripheral module interrupts can be selected from 16 levels for individual request sources.
- NMI noise canceller function

  An NMI input-level bit indicates the NMI pin state. The pin state can be checked by reading this bit in the interrupt exception service routine, and this LSI can be used for the noise canceller function.
- Register banks

  The LSI has two sets of register banks, one set for each CPU core, that enable register contents to be saved and restoration processing to be performed at high speed for the interrupt processing.
- Inter-processor interrupts

  By configuring the inter-processor interrupt control registers, inter-processor interrupts can be generated with programmed priority levels of 15 to 8.

Figure 7.1 shows a block diagram of the INTC.

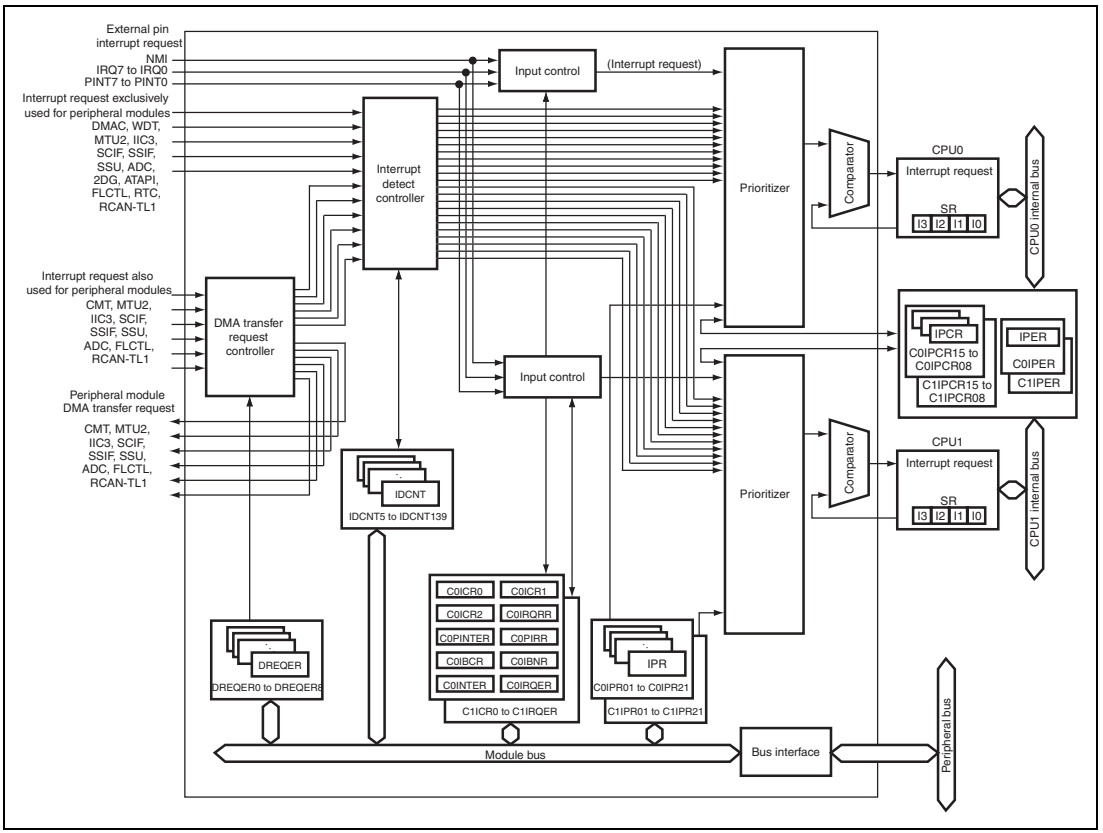


Figure 7.1 Block Diagram of INTC

7.2 Input/Output Pins

Table 7.1 shows the pin configuration of the INTC.

Table 7.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Nonmaskable interrupt input pin	NMI	Input	Input of nonmaskable interrupt request signal
Interrupt request input pins	IRQ7 to IRQ0	Input	Input of maskable interrupt request signals
	PINT7 to PINT0	Input	

7.3 Register Descriptions

The INTC has the following registers. These registers are used to set the interrupt priorities and control detection of the external interrupt input signals. The registers are classified as the following: CPU0-dedicated, CPU1-dedicated, and shared.

(1) CPU0-Dedicated Registers

Table 7.2 CPU0-Dedicated Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt control register 0	C0ICR0	R/W	*1	H'FFFD9400	16, 32
Interrupt control register 1	C0ICR1	R/W	H'0000	H'FFFD9402	16, 32
Interrupt control register 2	C0ICR2	R/W	H'0000	H'FFFD9404	16, 32
IRQ interrupt request register	C0IRQRR	R/(W)*2	H'0000	H'FFFD9406	16, 32
PINT interrupt enable register	C0PINTER	R/W	H'0000	H'FFFD9408	16, 32
PINT interrupt request register	C0PIRR	R	H'0000	H'FFFD940A	16, 32
Bank control register	C0IBCR	R/W	H'0000	H'FFFD940C	16, 32
Bank number register	C0IBNR	R/W	H'0000	H'FFFD940E	16, 32
Interrupt priority register 01	C0IPR01	R/W	H'0000	H'FFFD9418	16, 32
Interrupt priority register 02	C0IPR02	R/W	H'0000	H'FFFD941A	16, 32
Interrupt priority register 05	C0IPR05	R/W	H'0000	H'FFFD9420	16, 32

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt enable control register	C0INTER	R/W	H'E000	H'FFFD9428	16, 32
IRQ interrupt enable control register	C0IRQER	R/W	H'FFFF	H'FFFD942A	16, 32
Inter-processor interrupt control register 15	C0IPCR15	R/W	H'0000	H'FFFC1C00	16
Inter-processor interrupt control register 14	C0IPCR14	R/W	H'0000	H'FFFC1C02	16
Inter-processor interrupt control register 13	C0IPCR13	R/W	H'0000	H'FFFC1C04	16
Inter-processor interrupt control register 12	C0IPCR12	R/W	H'0000	H'FFFC1C06	16
Inter-processor interrupt control register 11	C0IPCR11	R/W	H'0000	H'FFFC1C08	16
Inter-processor interrupt control register 10	C0IPCR10	R/W	H'0000	H'FFFC1C0A	16
Inter-processor interrupt control register 9	C0IPCR09	R/W	H'0000	H'FFFC1C0C	16
Inter-processor interrupt control register 8	C0IPCR08	R/W	H'0000	H'FFFC1C0E	16
Inter-processor interrupt enable register	C0IPER	R/W	H'0000	H'FFFC1C10	16
Interrupt priority register 06	C0IPR06	R/W	H'0000	H'FFFD9800	16, 32
Interrupt priority register 07	C0IPR07	R/W	H'0000	H'FFFD9802	16, 32
Interrupt priority register 08	C0IPR08	R/W	H'0000	H'FFFD9804	16, 32
Interrupt priority register 09	C0IPR09	R/W	H'0000	H'FFFD9806	16, 32
Interrupt priority register 10	C0IPR10	R/W	H'0000	H'FFFD9808	16, 32
Interrupt priority register 11	C0IPR11	R/W	H'0000	H'FFFD980A	16, 32
Interrupt priority register 12	C0IPR12	R/W	H'0000	H'FFFD980C	16, 32
Interrupt priority register 13	C0IPR13	R/W	H'0000	H'FFFD980E	16, 32
Interrupt priority register 14	C0IPR14	R/W	H'0000	H'FFFD9810	16, 32
Interrupt priority register 15	C0IPR15	R/W	H'0000	H'FFFD9812	16, 32
Interrupt priority register 16	C0IPR16	R/W	H'0000	H'FFFD9814	16, 32

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt priority register 17	C0IPR17	R/W	H'0000	H'FFFD9816	16, 32
Interrupt priority register 18	C0IPR18	R/W	H'0000	H'FFFD9818	16, 32
Interrupt priority register 19	C0IPR19	R/W	H'0000	H'FFFD981A	16, 32
Interrupt priority register 20	C0IPR20	R/W	H'0000	H'FFFD981C	16, 32
Interrupt priority register 21	C0IPR21	R/W	H'0000	H'FFFD981E	16, 32

Notes: 1. The initial value is either H'8000 when the NMI pin is high, or H'0000 when the NMI pin is low.
2. To clear the flag, only 0 can be written after 1 is read.

(2) CPU1-Dedicated Registers

Table 7.3 CPU1-Dedicated Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt control register 0	C1ICR0	R/W	*1	H'FFFD9500	16, 32
Interrupt control register 1	C1ICR1	R/W	H'0000	H'FFFD9502	16, 32
Interrupt control register 2	C1ICR2	R/W	H'0000	H'FFFD9504	16, 32
IRQ interrupt request register	C1IRQRR	R/(W)*2	H'0000	H'FFFD9506	16, 32
PINT interrupt enable register	C1PINTER	R/W	H'0000	H'FFFD9508	16, 32
PINT interrupt request register	C1PIRR	R	H'0000	H'FFFD950A	16, 32
Bank control register	C1IBCR	R/W	H'0000	H'FFFD950C	16, 32
Bank number register	C1IBNR	R/W	H'0000	H'FFFD950E	16, 32
Interrupt priority register 01	C1IPR01	R/W	H'0000	H'FFFD9518	16, 32
Interrupt priority register 02	C1IPR02	R/W	H'0000	H'FFFD951A	16, 32
Interrupt priority register 05	C1IPR05	R/W	H'0000	H'FFFD9520	16, 32
Interrupt enable control register	C1INTER	R/W	H'0000	H'FFFD9528	16, 32
IRQ interrupt enable control register	C1IRQER	R/W	H'0000	H'FFFD952A	16, 32
Inter-processor interrupt control register 15	C1IPCR15	R/W	H'0000	H'FFFC1C20	16
Inter-processor interrupt control register 14	C1IPCR14	R/W	H'0000	H'FFFC1C22	16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Inter-processor interrupt control register 13	C1IPCR13	R/W	H'0000	H'FFFC1C24	16
Inter-processor interrupt control register 12	C1IPCR12	R/W	H'0000	H'FFFC1C26	16
Inter-processor interrupt control register 11	C1IPCR11	R/W	H'0000	H'FFFC1C28	16
Inter-processor interrupt control register 10	C1IPCR10	R/W	H'0000	H'FFFC1C2A	16
Inter-processor interrupt control register 9	C1IPCR09	R/W	H'0000	H'FFFC1C2C	16
Inter-processor interrupt control register 8	C1IPCR08	R/W	H'0000	H'FFFC1C2E	16
Inter-processor interrupt enable register	C1IPER	R/W	H'0000	H'FFFC1C30	16
Interrupt priority register 06	C1IPR06	R/W	H'0000	H'FFFD9900	16, 32
Interrupt priority register 07	C1IPR07	R/W	H'0000	H'FFFD9902	16, 32
Interrupt priority register 08	C1IPR08	R/W	H'0000	H'FFFD9904	16, 32
Interrupt priority register 09	C1IPR09	R/W	H'0000	H'FFFD9906	16, 32
Interrupt priority register 10	C1IPR10	R/W	H'0000	H'FFFD9908	16, 32
Interrupt priority register 11	C1IPR11	R/W	H'0000	H'FFFD990A	16, 32
Interrupt priority register 12	C1IPR12	R/W	H'0000	H'FFFD990C	16, 32
Interrupt priority register 13	C1IPR13	R/W	H'0000	H'FFFD990E	16, 32
Interrupt priority register 14	C1IPR14	R/W	H'0000	H'FFFD9910	16, 32
Interrupt priority register 15	C1IPR15	R/W	H'0000	H'FFFD9912	16, 32
Interrupt priority register 16	C1IPR16	R/W	H'0000	H'FFFD9914	16, 32
Interrupt priority register 17	C1IPR17	R/W	H'0000	H'FFFD9916	16, 32
Interrupt priority register 18	C1IPR18	R/W	H'0000	H'FFFD9918	16, 32
Interrupt priority register 19	C1IPR19	R/W	H'0000	H'FFFD991A	16, 32
Interrupt priority register 20	C1IPR20	R/W	H'0000	H'FFFD991C	16, 32
Interrupt priority register 21	C1IPR21	R/W	H'0000	H'FFFD991E	16, 32

Notes: 1. The initial value is either H'8000 when the NMI pin is high, or H'0000 when the NMI pin is low.
2. To clear the flag, only 0 can be written after 1 is read.

(3) Shared Registers

Table 7.4 Shared Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt detect control register 6	IDCNT6	R/W	H'4100	H'FFFD9C0C	16
Interrupt detect control register 7	IDCNT7	R/W	H'4100	H'FFFD9C0E	16
Interrupt detect control register 8	IDCNT8	R/W	H'4100	H'FFFD9C10	16
Interrupt detect control register 9	IDCNT9	R/W	H'4100	H'FFFD9C12	16
Interrupt detect control register 10	IDCNT10	R/W	H'4100	H'FFFD9C14	16
Interrupt detect control register 11	IDCNT11	R/W	H'4100	H'FFFD9C16	16
Interrupt detect control register 12	IDCNT12	R/W	H'4100	H'FFFD9C18	16
Interrupt detect control register 13	IDCNT13	R/W	H'4100	H'FFFD9C1A	16
Interrupt detect control register 14	IDCNT14	R/W	H'4100	H'FFFD9C1C	16
Interrupt detect control register 15	IDCNT15	R/W	H'4100	H'FFFD9C1E	16
Interrupt detect control register 16	IDCNT16	R/W	H'4100	H'FFFD9C20	16
Interrupt detect control register 17	IDCNT17	R/W	H'4100	H'FFFD9C22	16
Interrupt detect control register 18	IDCNT18	R/W	H'4100	H'FFFD9C24	16
Interrupt detect control register 19	IDCNT19	R/W	H'4100	H'FFFD9C26	16
Interrupt detect control register 20	IDCNT20	R/W	H'4100	H'FFFD9C28	16
Interrupt detect control register 21	IDCNT21	R/W	H'4100	H'FFFD9C2A	16
Interrupt detect control register 22	IDCNT22	R/W	H'4100	H'FFFD9C2C	16
Interrupt detect control register 23	IDCNT23	R/W	H'4100	H'FFFD9C2E	16
Interrupt detect control register 24	IDCNT24	R/W	H'4100	H'FFFD9C30	16
Interrupt detect control register 25	IDCNT25	R/W	H'4100	H'FFFD9C32	16
Interrupt detect control register 26	IDCNT26	R/W	H'4100	H'FFFD9C34	16
Interrupt detect control register 27	IDCNT27	R/W	H'4100	H'FFFD9C36	16
Interrupt detect control register 28	IDCNT28	R/W	H'4100	H'FFFD9C38	16
Interrupt detect control register 29	IDCNT29	R/W	H'4100	H'FFFD9C3A	16
Interrupt detect control register 30	IDCNT30	R/W	H'4100	H'FFFD9C3C	16
Interrupt detect control register 31	IDCNT31	R/W	H'4100	H'FFFD9C3E	16
Interrupt detect control register 32	IDCNT32	R/W	H'4100	H'FFFD9C40	16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt detect control register 33	IDCNT33	R/W	H'4100	H'FFFD9C42	16
Interrupt detect control register 34	IDCNT34	R/W	H'4100	H'FFFD9C44	16
Interrupt detect control register 35	IDCNT35	R/W	H'4100	H'FFFD9C46	16
Interrupt detect control register 36	IDCNT36	R/W	H'4100	H'FFFD9C48	16
Interrupt detect control register 37	IDCNT37	R/W	H'4100	H'FFFD9C4A	16
Interrupt detect control register 38	IDCNT38	R/W	H'4100	H'FFFD9C4C	16
Interrupt detect control register 39	IDCNT39	R/W	H'4100	H'FFFD9C4E	16
Interrupt detect control register 40	IDCNT40	R/W	H'4100	H'FFFD9C50	16
Interrupt detect control register 41	IDCNT41	R/W	H'4100	H'FFFD9C52	16
Interrupt detect control register 42	IDCNT42	R/W	H'4100	H'FFFD9C54	16
Interrupt detect control register 43	IDCNT43	R/W	H'4100	H'FFFD9C56	16
Interrupt detect control register 44	IDCNT44	R/W	H'4100	H'FFFD9C58	16
Interrupt detect control register 45	IDCNT45	R/W	H'4100	H'FFFD9C5A	16
Interrupt detect control register 46	IDCNT46	R/W	H'4100	H'FFFD9C5C	16
Interrupt detect control register 47	IDCNT47	R/W	H'4100	H'FFFD9C5E	16
Interrupt detect control register 48	IDCNT48	R/W	H'4100	H'FFFD9C60	16
Interrupt detect control register 49	IDCNT49	R/W	H'4100	H'FFFD9C62	16
Interrupt detect control register 50	IDCNT50	R/W	H'4100	H'FFFD9C64	16
Interrupt detect control register 51	IDCNT51	R/W	H'4100	H'FFFD9C66	16
Interrupt detect control register 52	IDCNT52	R/W	H'4100	H'FFFD9C68	16
Interrupt detect control register 53	IDCNT53	R/W	H'4100	H'FFFD9C6A	16
Interrupt detect control register 54	IDCNT54	R/W	H'4100	H'FFFD9C6C	16
Interrupt detect control register 55	IDCNT55	R/W	H'4100	H'FFFD9C6E	16
Interrupt detect control register 56	IDCNT56	R/W	H'4100	H'FFFD9C70	16
Interrupt detect control register 57	IDCNT57	R/W	H'4100	H'FFFD9C72	16
Interrupt detect control register 58	IDCNT58	R/W	H'4100	H'FFFD9C74	16
Interrupt detect control register 59	IDCNT59	R/W	H'4100	H'FFFD9C76	16
Interrupt detect control register 60	IDCNT60	R/W	H'4100	H'FFFD9C78	16
Interrupt detect control register 61	IDCNT61	R/W	H'4100	H'FFFD9C7A	16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt detect control register 62	IDCNT62	R/W	H'4100	H'FFFD9C7C	16
Interrupt detect control register 63	IDCNT63	R/W	H'4100	H'FFFD9C7E	16
Interrupt detect control register 64	IDCNT64	R/W	H'4100	H'FFFD9C80	16
Interrupt detect control register 66	IDCNT66	R/W	H'4100	H'FFFD9C84	16
Interrupt detect control register 67	IDCNT67	R/W	H'4100	H'FFFD9C86	16
Interrupt detect control register 68	IDCNT68	R/W	H'4100	H'FFFD9C88	16
Interrupt detect control register 69	IDCNT69	R/W	H'4100	H'FFFD9C8A	16
Interrupt detect control register 70	IDCNT70	R/W	H'4100	H'FFFD9C8C	16
Interrupt detect control register 71	IDCNT71	R/W	H'4100	H'FFFD9C8E	16
Interrupt detect control register 72	IDCNT72	R/W	H'4100	H'FFFD9C90	16
Interrupt detect control register 73	IDCNT73	R/W	H'4100	H'FFFD9C92	16
Interrupt detect control register 74	IDCNT74	R/W	H'4100	H'FFFD9C94	16
Interrupt detect control register 75	IDCNT75	R/W	H'4100	H'FFFD9C96	16
Interrupt detect control register 76	IDCNT76	R/W	H'4100	H'FFFD9C98	16
Interrupt detect control register 77	IDCNT77	R/W	H'4100	H'FFFD9C9A	16
Interrupt detect control register 78	IDCNT78	R/W	H'4100	H'FFFD9C9C	16
Interrupt detect control register 79	IDCNT79	R/W	H'4100	H'FFFD9C9E	16
Interrupt detect control register 80	IDCNT80	R/W	H'4100	H'FFFD9CA0	16
Interrupt detect control register 81	IDCNT81	R/W	H'4100	H'FFFD9CA2	16
Interrupt detect control register 82	IDCNT82	R/W	H'4100	H'FFFD9CA4	16
Interrupt detect control register 83	IDCNT83	R/W	H'4100	H'FFFD9CA6	16
Interrupt detect control register 84	IDCNT84	R/W	H'4100	H'FFFD9CA8	16
Interrupt detect control register 85	IDCNT85	R/W	H'4100	H'FFFD9CAA	16
Interrupt detect control register 86	IDCNT86	R/W	H'4100	H'FFFD9CAC	16
Interrupt detect control register 87	IDCNT87	R/W	H'4100	H'FFFD9CAE	16
Interrupt detect control register 88	IDCNT88	R/W	H'4100	H'FFFD9CB0	16
Interrupt detect control register 89	IDCNT89	R/W	H'4100	H'FFFD9CB2	16
Interrupt detect control register 90	IDCNT90	R/W	H'4100	H'FFFD9CB4	16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt detect control register 91	IDCNT91	R/W	H'4100	H'FFFD9CB6	16
Interrupt detect control register 92	IDCNT92	R/W	H'4100	H'FFFD9CB8	16
Interrupt detect control register 93	IDCNT93	R/W	H'4100	H'FFFD9CBA	16
Interrupt detect control register 94	IDCNT94	R/W	H'4100	H'FFFD9CBC	16
Interrupt detect control register 95	IDCNT95	R/W	H'4100	H'FFFD9CBE	16
Interrupt detect control register 96	IDCNT96	R/W	H'4100	H'FFFD9CC0	16
Interrupt detect control register 97	IDCNT97	R/W	H'4100	H'FFFD9CC2	16
Interrupt detect control register 98	IDCNT98	R/W	H'4100	H'FFFD9CC4	16
Interrupt detect control register 99	IDCNT99	R/W	H'4100	H'FFFD9CC6	16
Interrupt detect control register 100	IDCNT100	R/W	H'4100	H'FFFD9CC8	16
Interrupt detect control register 101	IDCNT101	R/W	H'4100	H'FFFD9CCA	16
Interrupt detect control register 102	IDCNT102	R/W	H'4100	H'FFFD9CCC	16
Interrupt detect control register 103	IDCNT103	R/W	H'4100	H'FFFD9CCE	16
Interrupt detect control register 104	IDCNT104	R/W	H'4100	H'FFFD9CD0	16
Interrupt detect control register 105	IDCNT105	R/W	H'4100	H'FFFD9CD2	16
Interrupt detect control register 106	IDCNT106	R/W	H'4100	H'FFFD9CD4	16
Interrupt detect control register 107	IDCNT107	R/W	H'4100	H'FFFD9CD6	16
Interrupt detect control register 108	IDCNT108	R/W	H'4100	H'FFFD9CD8	16
Interrupt detect control register 109	IDCNT109	R/W	H'4100	H'FFFD9CDA	16
Interrupt detect control register 110	IDCNT110	R/W	H'4100	H'FFFD9CDC	16
Interrupt detect control register 111	IDCNT111	R/W	H'4100	H'FFFD9CDE	16
Interrupt detect control register 112	IDCNT112	R/W	H'4100	H'FFFD9CE0	16
Interrupt detect control register 113	IDCNT113	R/W	H'4100	H'FFFD9CE2	16
Interrupt detect control register 114	IDCNT114	R/W	H'4100	H'FFFD9CE4	16
Interrupt detect control register 115	IDCNT115	R/W	H'4100	H'FFFD9CE6	16
Interrupt detect control register 116	IDCNT116	R/W	H'4100	H'FFFD9CE8	16
Interrupt detect control register 117	IDCNT117	R/W	H'4100	H'FFFD9CEA	16
Interrupt detect control register 118	IDCNT118	R/W	H'4100	H'FFFD9CEC	16
Interrupt detect control register 119	IDCNT119	R/W	H'4100	H'FFFD9CEE	16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt detect control register 120	IDCNT120	R/W	H'4100	H'FFFD9CF0	16
Interrupt detect control register 121	IDCNT121	R/W	H'4100	H'FFFD9CF2	16
Interrupt detect control register 122	IDCNT122	R/W	H'4100	H'FFFD9CF4	16
Interrupt detect control register 123	IDCNT123	R/W	H'4100	H'FFFD9CF6	16
Interrupt detect control register 124	IDCNT124	R/W	H'4100	H'FFFD9CF8	16
Interrupt detect control register 125	IDCNT125	R/W	H'4100	H'FFFD9CFA	16
Interrupt detect control register 126	IDCNT126	R/W	H'4100	H'FFFD9CFC	16
Interrupt detect control register 130	IDCNT130	R/W	H'4100	H'FFFD9D04	16
Interrupt detect control register 131	IDCNT131	R/W	H'4100	H'FFFD9D06	16
Interrupt detect control register 132	IDCNT132	R/W	H'4100	H'FFFD9D08	16
Interrupt detect control register 133	IDCNT133	R/W	H'4100	H'FFFD9D0A	16
Interrupt detect control register 134	IDCNT134	R/W	H'4100	H'FFFD9D0C	16
Interrupt detect control register 135	IDCNT135	R/W	H'4100	H'FFFD9D0E	16
Interrupt detect control register 136	IDCNT136	R/W	H'4100	H'FFFD9D10	16
Interrupt detect control register 137	IDCNT137	R/W	H'4100	H'FFFD9D12	16
Interrupt detect control register 138	IDCNT138	R/W	H'4100	H'FFFD9D14	16
Interrupt detect control register 139	IDCNT139	R/W	H'4100	H'FFFD9D16	16
DMA transfer request enable register 0	DREQER0	R/W	H'00	H'FFFE0800	8, 16, 32
DMA transfer request enable register 1	DREQER1	R/W	H'00	H'FFFE0801	8
DMA transfer request enable register 2	DREQER2	R/W	H'00	H'FFFE0802	8, 16
DMA transfer request enable register 3	DREQER3	R/W	H'00	H'FFFE0803	8
DMA transfer request enable register 4	DREQER4	R/W	H'00	H'FFFE0804	8, 16, 32

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
DMA transfer request enable register 5	DREQER5	R/W	H'00	H'FFFE0805	8
DMA transfer request enable register 6	DREQER6	R/W	H'00	H'FFFE0806	8, 16
DMA transfer request enable register 7	DREQER7	R/W	H'00	H'FFFE0807	8
DMA transfer request enable register 8	DREQER8	R/W	H'00	H'FFFE0808	8

7.3.1 Interrupt Priority Registers 01, 02, 05 to 21 (C0IPR01, C0IPR02, C0IPR05 to C0IPR21, C1IPR01, C1IPR02, C1IPR05 to C1IPR21)

C0IPR01, C0IPR02, and C0IPR05 to C0IPR21 and C1IPR01, C1IPR02, and C1IPR05 to C0IPR21 are 16-bit readable/writable registers in which priority levels from 0 to 15 are set for IRQ interrupts, PINT interrupts, and on-chip peripheral module interrupts. Table 7.5 shows the correspondence between the interrupt request sources and the bits in C0IPR01, C0IPR02, and C0IPR05 to C0IPR21 and C1IPR01, C1IPR02, and C1IPR05 to C0IPR21.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.5 Interrupt Request Sources and C0IPR01, C0IPR02, C0IPR05 to C0IPR21, C1IPR01, C1IPR02, and C1IPR05 to C0IPR21

	Bit			
Register	**15 to 12**	**11 to 8**	**7 to 4**	**3 to 0**
Interrupt priority register 01	IRQ0	IRQ1	IRQ2	IRQ3
Interrupt priority register 02	IRQ4	IRQ5	IRQ6	IRQ7
Interrupt priority register 05	PINT0 to PINT7	Reserved	Reserved	Reserved
Interrupt priority register 06	DMAC0	DMAC1	DMAC2	DMAC3
Interrupt priority register 07	DMAC4	DMAC5	DMAC6	DMAC7
Interrupt priority register 08	DMAC8	DMAC9	DMAC10	DMAC11
Interrupt priority register 09	DMAC12	DMAC13	DMAC-shared	USB
Interrupt priority register 10	CMT0	CMT1	CMT2	CMT3
Interrupt priority register 11	WDT0	WDT1	MTU0 (TGI0A to TGI0D)	MTU0 (TCI0V, TGI0E, TGI0F)
Interrupt priority register 12	MTU1 (TGI1A, TGI1B)	MTU1 (TCI1V, TCI1U)	MTU2 (TGI2A, TGI2B)	MTU2 (TCI2V, TCI2U)
Interrupt priority register 13	MTU3 (TGI3A to TGI3D)	MTU3 (TCI3V)	MTU4 (TGI4A to TGI4D)	MTU4 (TCI4V)
Interrupt priority register 14	SSIF0	SSIF1	SSIF2	SSIF3
Interrupt priority register 15	SSIF4	SSIF5	Reserved	Reserved
Interrupt priority register 16	IIC3_0	IIC3_1	IIC3_2	IIC3_3
Interrupt priority register 17	SCIF0	SCIF1	SCIF2	SCIF3
Interrupt priority register 18	SCIF4	SCIF5	Reserved	Reserved
Interrupt priority register 19	SSU0	SSU1	ADC	2DG
Interrupt priority register 20	ATAPI	FLCTL	RTC	Reserved
Interrupt priority register 21	RCAN0	RCAN1	Reserved	Reserved

As shown in table 7.5, by setting the 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, and bits 3 to 0) with values from H'0 (0000) to H'F (1111), the priority of each corresponding interrupt is set. Setting of H'0 means priority level 0 (the lowest level) and H'F means priority level 15 (the highest level).

7.3.2 Interrupt Control Registers 0 (C0ICR0, C1ICR0)

C0ICR0 and C1ICR0 are 16-bit registers that set the input signal detection mode for the external interrupt input pin NMI, and indicate the input level at the NMI pin.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMIL	-	-	-	-	-	-	NMIS	-	-	-	-	-	-	-	-
Initial value:	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	*	R	NMI Input Level This bit sets the level of the signal input to the NMI pin. The NMI pin level can be obtained by reading this bit. This bit cannot be modified. 0: Low level is input to NMI pin. 1: High level is input to NMI pin.
14 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	NMIS	0	R/W	NMI Edge Select This bit selects whether interrupt request signals are detected on the falling or rising edge of NMI input. 0: Interrupt request is detected on falling edge of NMI input. 1: Interrupt request is detected on rising edge of NMI input. Note: C0ICR0 and C1ICR0 must be the same in the value set in this bit.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * The initial value is either 1 when the NMI pin is high, or 0 when the NMI pin is low.

7.3.3 Interrupt Control Registers 1 (C0ICR1, C1ICR1)

C0ICR1 and C1ICR1 are 16-bit registers that specify the detection mode for external interrupt input pins IRQ7 to IRQ0 individually: falling edge, rising edge, both edges, or low level.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ71S	0	R/W	IRQ Sense Select
14	IRQ70S	0	R/W	These bits select whether interrupt signals input to pins IRQ7 to IRQ0 are detected on a low level, falling edge, rising edge, or both edges.
13	IRQ61S	0	R/W	
12	IRQ60S	0	R/W	00: Interrupt request is detected on low level of IRQn input.
11	IRQ51S	0	R/W	01: Interrupt request is detected on falling edge of IRQn input.
10	IRQ50S	0	R/W	
9	IRQ41S	0	R/W	10: Interrupt request is detected on rising edge of IRQn input.
8	IRQ40S	0	R/W	
7	IRQ31S	0	R/W	11: Interrupt request is detected on both edges of IRQn input.
6	IRQ30S	0	R/W	Note: C0ICR1 and C1ICR1 must be the same in each value set in IRQ71S to IRQ0S.
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	

[Legend]
n = 7 to 0

7.3.4 Interrupt Control Registers 2 (C0ICR2, C1ICR2)

C0ICR2 and C1ICR2 are 16-bit registers that specify the detection mode for external interrupt input pins PINT7 to PINT0 individually: low level or high level.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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	PINT7S	0	R/W	PINT Sense Select These bits select whether interrupt signals input to pins PINT7 to PINT0 are detected on a low level or high level. 0: Interrupt request is detected on low level of PINTn input. 1: Interrupt request is detected on high level of PINTn input. Note: C0ICR2 and C1ICR2 must be the same in each value set in PINT7S to PINT0S.
6	PINT6S	0	R/W	
5	PINT5S	0	R/W	
4	PINT4S	0	R/W	
3	PINT3S	0	R/W	
2	PINT2S	0	R/W	
1	PINT1S	0	R/W	
0	PINT0S	0	R/W	

[Legend]

n = 7 to 0

7.3.5 IRQ Interrupt Request Registers (C0IRQRR, C1IRQRR)

C0IRQRR and C1IRQRR are 16-bit registers that indicate interrupt requests from external interrupt input pins IRQ7 to IRQ0. If edge detection is set for the IRQ7 to IRQ0 interrupts, the retained interrupt requests can be cancelled by reading 1 from the IRQ7F to IRQ0F bits and then writing 0 to these bits. However, this register is enabled only when C0IRQER and C1IRQER accept an interrupt request input. When an interrupt request input is disabled, this register always becomes 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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	IRQ7F	0	R/W	IRQ Interrupt Request These bits indicate the status of the IRQ7 to IRQ0 interrupt requests. Level detection: 0: IRQn interrupt request has not occurred. [Clearing condition] • IRQn input is high 1: IRQn interrupt request has occurred. [Setting condition] • IRQn input is low Edge detection: 0: IRQn interrupt request is not detected. [Clearing conditions] • Cleared by reading 1 from IRQnF, and then writing 0 to IRQnF • Cleared by executing IRQn interrupt exception handling 1: IRQn interrupt request is detected. [Setting condition] • Edge corresponding to IRQn1S or IRQn0S of C0ICR1 or C1ICR1 has occurred at IRQn pin
6	IRQ6F	0	R/W	
5	IRQ5F	0	R/W	
4	IRQ4F	0	R/W	
3	IRQ3F	0	R/W	
2	IRQ2F	0	R/W	
1	IRQ1F	0	R/W	
0	IRQ0F	0	R/W	

[Legend]
n = 7 to 0

7.3.6 PINT Interrupt Enable Registers (C0PINTER, C1PINTER)

C0PINTER and C1PINTER are 16-bit registers that enable interrupt request inputs to external interrupt input pins PINT7 to PINT0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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	PINT7E	0	R/W	PINT Enable These bits select whether to enable interrupt request inputs to external interrupt input pins PINT7 to PINT0. 0: Interrupt request input to PINTn is disabled. 1: Interrupt request input to PINTn is enabled.
6	PINT6E	0	R/W	
5	PINT5E	0	R/W	
4	PINT4E	0	R/W	
3	PINT3E	0	R/W	
2	PINT2E	0	R/W	
1	PINT1E	0	R/W	
0	PINT0E	0	R/W	

[Legend]

n = 7 to 0

7.3.7 PINT Interrupt Request Registers (C0PIRR, C1PIRR)

C0PIRR and C1PIRR are 16-bit registers that indicate interrupt requests from external interrupt input pins PINT7 to PINT0.

However, this register is enabled only when C0PINTER and C1PINTER accept an interrupt request input. When an interrupt request input is disabled, this register always becomes 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PINT7R	PINT6R	PINT5R	PINT4R	PINT3R	PINT2R	PINT1R	PINT0R
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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	PINT7R	0	R	PINT Interrupt Request These bits indicate the status of the PINT7 to PINT0 interrupt requests. 0: No interrupt request at PINTn pin 1: Interrupt request at PINTn pin
6	PINT6R	0	R	
5	PINT5R	0	R	
4	PINT4R	0	R	
3	PINT3R	0	R	
2	PINT2R	0	R	
1	PINT1R	0	R	
0	PINT0R	0	R	

[Legend]
n = 7 to 0

7.3.8 Bank Control Registers (C0IBCR, C1IBCR)

C0IBCR and C1IBCR are 16-bit registers that enable or disable the use of register banks for each interrupt priority level.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15	E15	0	R/W	Enable These bits enable or disable the use of register banks for interrupt priority levels 15 to 1. However, the use of register banks is always disabled for the user break interrupts. 0: Use of register banks is disabled. 1: Use of register banks is enabled.
14	E14	0	R/W	
13	E13	0	R/W	
12	E12	0	R/W	
11	E11	0	R/W	
10	E10	0	R/W	
9	E9	0	R/W	
8	E8	0	R/W	
7	E7	0	R/W	
6	E6	0	R/W	
5	E5	0	R/W	
4	E4	0	R/W	
3	E3	0	R/W	
2	E2	0	R/W	
1	E1	0	R/W	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

7.3.9 Bank Number Registers (C0IBNR, C1IBNR)

C0IBNR and C1IBNR are 16-bit registers that enable or disable the use of register banks and register bank overflow exception. In bits BN3 to BN0, C0IBNR and C1IBNR indicate the number of the bank to which saving is performed next.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BE[1:0]		BOVE	-	-	-	-	-	-	-	-	-	BN[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	BE[1:0]	00	R/W	Register Bank Enable These bits enable or disable the use of register banks. 00: Use of register banks is disabled for all interrupts. The settings of C0IBCR and C1IBCR are ignored. 01: Use of register banks is enabled for all interrupts except NMI and user break. The settings of C0IBCR and C1IBCR are ignored. 10: Reserved (setting prohibited) 11: Use of register banks is controlled by the settings of C0IBCR and C1IBCR.
13	BOVE	0	R/W	Register Bank Overflow Enable This bit enables or disables register bank overflow exception. 0: Generation of register bank overflow exception is disabled. 1: Generation of register bank overflow exception is enabled.
12 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	BN[3:0]	0000	R	Bank Number These bits indicate the number of the bank to which saving is performed next. When an interrupt is accepted and register banks are used for the interrupt, saving is performed to the register bank indicated by these bits, and BN is incremented by 1. After BN is decremented by 1 due to execution of a RESBANK (restore from register bank) instruction, restoration from the register bank is performed.

7.3.10 Inter-Processor Interrupt Control Registers 15 to 08 (C0IPCR15 to C0IPCR08, C1IPCR15 to C1IPCR08)

C0IPCR15 to C0IPCR08 and C1IPCR15 to C1IPCR08 are 16-bit registers that generate inter-processor interrupts when 1 is written to any of the CI bits. Each CI bit remains 1 until the target processor accepts interrupt processing, and the bit is cleared to 0 upon completion of the acceptance.

An inter-processor interrupt request made from CPU0 is set in one of C1IPCR15 to C1IPCR08. An inter-processor interrupt request made from CPU1 to CPU0 is set in one of C0IPCR15 to C0IPCR08.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CI
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

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	CI	0	R/W	Inter-processor Interrupt Request 0: Inter-processor interrupt request is not set. 1: Inter-processor interrupt request is set.

Note: Although 0 can be written to the CI bit, an inter-processor interrupt request is held pending internally, and cannot be cleared.

The interrupt priorities for the registers are predefined as shown below.

Table 7.6 Interrupt Priorities for Registers

Register Name		Level	Priority
C0IPCR15	C1IPCR15	Level 15	High
C0IPCR14	C1IPCR14	Level 14	
C0IPCR13	C1IPCR13	Level 13	
C0IPCR12	C1IPCR12	Level 12	
C0IPCR11	C1IPCR11	Level 11	
C0IPCR10	C1IPCR10	Level 10	
C0IPCR09	C1IPCR09	Level 9	
C0IPCR08	C1IPCR08	Level 8	Low

7.3.11 Inter-processor Interrupt Enable Registers (C0IPER, C1IPER)

C0IPER and C1IPER are 16-bit registers that enable or disable inter-processor interrupts of each interrupt priority level. The interrupt controller decides whether to accept interrupts, according to the inter-processor interrupt enable settings. C0IPER enables or disables interrupts to CPU0, while C1IPER enables or disables interrupts to CPU1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CIPE15	CIPE14	CIPE13	CIPE12	CIPE11	CIPE10	CIPE9	CIPE8	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	CIPE15	0	R/W	Inter-processor Interrupt Enable These bits enable or disable inter-processor interrupt requests of priority levels 15 to 8. 0: Inter-processor interrupt is disabled. 1: Inter-processor interrupt is enabled.
14	CIPE14	0	R/W	
13	CIPE13	0	R/W	
12	CIPE12	0	R/W	
11	CIPE11	0	R/W	
10	CIPE10	0	R/W	
9	CIPE9	0	R/W	
8	CIPE8	0	R/W	
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

7.3.12 Interrupt Enable Control Registers (C0INTER, C1INTER)

C0INTER and C1INTER are 16-bit registers that control whether to enable or disable acceptance of interrupt requests by processors CPU0 and CPU1. If the same bits in both registers C0INTER and C1INTER are set to 0, the acceptance by CPU0 is enabled.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMIE	UDIE	SLPEE	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	*	*	*	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	NMIE	*	R/W	NMI Interrupt Enable This bit selects whether to enable NMI interrupt request inputs. 0: NMI interrupt request input is disabled. 1: NMI interrupt request input is enabled.
14	UDIE	*	R/W	UDI Interrupt Enable This bit selects whether to enable interrupt request inputs from UDI. 0: Interrupt request input from UDI is disabled. 1: Interrupt request input from UDI is enabled.
13	SLPEE	*	R/W	Sleep Error Interrupt Enable This bit selects whether to enable interrupt request inputs for sleep errors. 0: Interrupt request input for sleep error is disabled. 1: Interrupt request input for sleep error is enabled.
12 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * The initial value is 1 for C0INTER and 0 for C1INTER.

7.3.13 IRQ Interrupt Enable Control Registers (C0IRQER, C1IRQER)

C0IRQER and C1IRQER are 16-bit registers that control whether to enable or disable acceptance of IRQ interrupt requests by processors CPU0 and CPU1. If the same bits in both registers C0IRQER and C1IRQER are set to 0, the acceptance by CPU0 is enabled.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	*	R	Reserved In C0IRQER, these bits are always read as 1. The write value should always be 1. In C1IRQER, these bits are always read as 0. The write value should always be 0.
7	IRQ7	*	R/W	IRQn Interrupt Enable These bits select whether to enable IRQn interrupt request inputs. 0: IRQn interrupt request input is disabled. 1: IRQn interrupt request input is enabled.
6	IRQ6	*	R/W	
5	IRQ5	*	R/W	
4	IRQ4	*	R/W	
3	IRQ3	*	R/W	
2	IRQ2	*	R/W	
1	IRQ1	*	R/W	
0	IRQ0	*	R/W	

[Legend]

n = 7 to 0

Note: * The initial value is 1 for C0IRQER and 0 for C1IRQER.

7.3.14 Interrupt Detect Control Registers (IDCNT6 to IDCNT139)

IDCNT6 to IDCNT139 (except IDCNT65 and 127 to 129) are 16-bit registers that control whether to enable interrupt requests from on-chip peripheral modules and also control which CPU should accept the requests.

Table 7.7 shows the correspondence between the sources of on-chip peripheral module interrupt requests and the IDCNT registers.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	CPUN	INTEN	-	-	-	MON	-	-	-	-
Initial value:	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

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	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
13 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	CPUN	0	R/W	CPU Accepting Interrupt Request This bit specifies which CPU should accept interrupt requests from on-chip peripheral modules. 0: CPU0 accepts interrupt request from on-chip peripheral module. 1: CPU1 accepts interrupt request from on-chip peripheral module.
8	INTEN	1	R/W	Interrupt Request Input Enable This bit enables or disables acceptance of interrupt requests from on-chip peripheral modules. 0: Interrupt request input from on-chip peripheral module is disabled. 1: Interrupt request input from on-chip peripheral module is enabled.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	MON	0	R	Interrupt Request Monitor This bit is used to monitor the status of interrupt requests from on-chip peripheral modules. 0: No interrupt request has been made from on-chip peripheral module. 1: Interrupt request has been made from on-chip peripheral module.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 7.7 Correspondence between Sources of On-Chip Peripheral Module Interrupt Requests and IDCNT Registers

Interrupt Source		Corresponding IDCNT Register
DMAC	DMINT0	IDCNT6
	DMINT1	IDCNT7
	DMINT2	IDCNT8
	DMINT3	IDCNT9
	DMINT4	IDCNT10
	DMINT5	IDCNT11
	DMINT6	IDCNT12
	DMINT7	IDCNT13
	DMINT8	IDCNT14
	DMINT9	IDCNT15
	DMINT10	IDCNT16
	DMINT11	IDCNT17
	DMINT12	IDCNT18
	DMINT13	IDCNT19
	DMINTA	IDCNT20
USB	USBI	IDCNT21
CMT0	CMI0	IDCNT22
	CMI1	IDCNT23
CMT1	CMI2	IDCNT24
	CMI3	IDCNT25
WDT0	ITI0	IDCNT26
WDT1	ITI1	IDCNT27
MTU0	TGI0A	IDCNT28
	TGI0B	IDCNT29
	TGI0C	IDCNT30
	TGI0D	IDCNT31
	TCI0V	IDCNT32
	TGI0E	IDCNT33
	TGI0F	IDCNT34
MTU1	TGI1A	IDCNT35
	TGI1B	IDCNT36
	TCI1V	IDCNT37
	TCI1U	IDCNT38
MTU2	TGI2A	IDCNT39
	TGI2B	IDCNT40
	TCI2V	IDCNT41
	TCI2U	IDCNT42
MTU3	TGI3A	IDCNT43
	TGI3B	IDCNT44
	TGI3C	IDCNT45
	TGI3D	IDCNT46
	TCI3V	IDCNT47
MTU4	TGI4A	IDCNT48
	TGI4B	IDCNT49
	TGI4C	IDCNT50
	TGI4D	IDCNT51
	TCI4V	IDCNT52
SSIF0	SSII0	IDCNT53
	SSIRTI0	IDCNT54
SSIF1	SSII1	IDCNT55
	SSIRTI1	IDCNT56
SSIF2	SSII2	IDCNT57
	SSIRTI2	IDCNT58
SSIF3	SSII3	IDCNT59
	SSIRTI3	IDCNT60
SSIF4	SSII4	IDCNT61
	SSIRTI4	IDCNT62

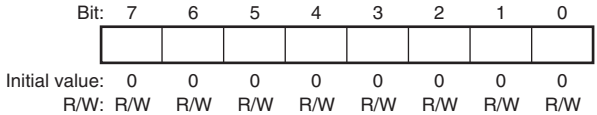
Interrupt Source		Corresponding IDCNT Register
SSIF5	SSII5	IDCNT63
	SSIRTI5	IDCNT64
IIC3_0	STPI0	IDCNT66
	NAKI0	IDCNT67
	RXI0	IDCNT68
	TXI0	IDCNT69
	TEI0	IDCNT70
IIC3_1	STPI1	IDCNT71
	NAKI1	IDCNT72
	RXI1	IDCNT73
	TXI1	IDCNT74
	TEI1	IDCNT75
IIC3_2	STPI2	IDCNT76
	NAKI2	IDCNT77
	RXI2	IDCNT78
	TXI2	IDCNT79
	TEI2	IDCNT80
IIC3_3	STPI3	IDCNT81
	NAKI3	IDCNT82
	RXI3	IDCNT83
	TXI3	IDCNT84
	TEI3	IDCNT85
SCIF0	BRI0	IDCNT86
	ERI0	IDCNT87
	RXI0	IDCNT88
	TXI0	IDCNT89
SCIF1	BRI1	IDCNT90
	ERI1	IDCNT91
	RXI1	IDCNT92
	TXI1	IDCNT93
SCIF2	BRI2	IDCNT94
	ERI2	IDCNT95
	TXI2	IDCNT97
SCIF3	BRI3	IDCNT98
	ERI3	IDCNT99
	RXI3	IDCNT100
	TXI3	IDCNT101
SCIF4	BRI4	IDCNT102
	ERI4	IDCNT103
	RXI4	IDCNT104
	TXI4	IDCNT105
SCIF5	BRI5	IDCNT106
	ERI5	IDCNT107
	RXI5	IDCNT108
	TXI5	IDCNT109
SSU0	SSERI0	IDCNT110
	SSRXI0	IDCNT111
	SSTXI0	IDCNT112
SSU1	SSERI1	IDCNT113
	SSRXI1	IDCNT114
	SSTXI1	IDCNT115
ADC	ADI	IDCNT116
2DG	BLT interrupt	IDCNT117
	Output interrupt	IDCNT118
ATAPI	ATAPII	IDCNT119
FLCTL	FLSTEI	IDCNT120
	FLTENDI	IDCNT121
	FLTREQ0I	IDCNT122
	FLTREQ1I	IDCNT123

Interrupt Source		Corresponding IDCNT Register
RTC	ARM	IDCNT124
	PRD	IDCNT125
	CUP	IDCNT126
RCAN0	ERS0	IDCNT130
	OVR0	IDCNT131
	RM00	IDCNT132
	RM10	IDCNT133
	SLE0	IDCNT134

Interrupt Source		Corresponding IDCNT Register
RCAN1	ERS1	IDCNT135
	OVR1	IDCNT136
	RM01	IDCNT137
	RM11	IDCNT138
	SLE1	IDCNT139

7.3.15 DMA Transfer Request Enable Registers 0 to 8 (DREQER0 to DREQER8)

DREQER0 to DREQER8 are 8-bit readable/writable registers that enable or disable DMA transfer requests from on-chip peripheral modules, and enable or disable CPU interrupts.



(1) DREQER0

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CMT CMI3	0	R/W	DMA Transfer Enable These bits enable or disable DMA transfer requests and CPU interrupt requests. 0: DMA transfer request is disabled and CPU interrupt request is enabled 1: DMA transfer request is enabled and CPU interrupt request is disabled
2	CMT CMI2	0	R/W	
1	CMT CMI1	0	R/W	
0	CMT CMI0	0	R/W	

(2) DREQER1

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	MTU TGI4A	0	R/W	DMA Transfer Enable These bits enable or disable DMA transfer requests and CPU interrupt requests. 0: DMA transfer request is disabled and CPU interrupt request is enabled. 1: DMA transfer request is enabled and CPU interrupt request is disabled.
3	MTU TGI3A	0	R/W	
2	MTU TGI2A	0	R/W	
1	MTU TGI1A	0	R/W	
0	MTU TGI0A	0	R/W	

(3) DREQER2

Bit	Bit Name	Initial Value	R/W	Description
7	IIC TXI3	0	R/W	DMA Transfer Enable These bits enable or disable DMA transfer requests and CPU interrupt requests. 0: DMA transfer request is disabled and CPU interrupt request is enabled. 1: DMA transfer request is enabled and CPU interrupt request is disabled.
6	IIC RXI3	0	R/W	
5	IIC TXI2	0	R/W	
4	IIC RXI2	0	R/W	
3	IIC TXI1	0	R/W	
2	IIC RXI1	0	R/W	
1	IIC TXI0	0	R/W	
0	IIC RXI0	0	R/W	

(4) DREQER3

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SCIF TXI5	0	R/W	DMA Transfer Enable These bits enable or disable DMA transfer requests and CPU interrupt requests. 0: DMA transfer request is disabled and CPU interrupt request is enabled. 1: DMA transfer request is enabled and CPU interrupt request is disabled.
2	SCIF RXI5	0	R/W	
1	SCIF TXI4	0	R/W	
0	SCIF RXI4	0	R/W	

(5) DREQER4

Bit	Bit Name	Initial Value	R/W	Description
7	SCIF TXI3	0	R/W	DMA Transfer Enable These bits enable or disable DMA transfer requests and CPU interrupt requests. 0: DMA transfer request is disabled and CPU interrupt request is enabled. 1: DMA transfer request is enabled and CPU interrupt request is disabled.
6	SCIF RXI3	0	R/W	
5	SCIF TXI2	0	R/W	
4	SCIF RXI2	0	R/W	
3	SCIF TXI1	0	R/W	
2	SCIF RXI1	0	R/W	
1	SCIF TXI0	0	R/W	
0	SCIF RXI0	0	R/W	

(6) DREQER5

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	SSI SSIRTI5	0	R/W	DMA Transfer Enable These bits enable or disable DMA transfer requests and CPU interrupt requests. 0: DMA transfer request is disabled and CPU interrupt request is enabled. 1: DMA transfer request is enabled and CPU interrupt request is disabled.
4	SSI SSIRTI4	0	R/W	
3	SSI SSIRTI3	0	R/W	
2	SSI SSIRTI2	0	R/W	
1	SSI SSIRTI1	0	R/W	
0	SSI SSIRTI0	0	R/W	

(7) DREQER6

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SSU TXI1	0	R/W	DMA Transfer Enable These bits enable or disable DMA transfer requests and CPU interrupt requests. 0: DMA transfer request is disabled and CPU interrupt request is enabled. 1: DMA transfer request is enabled and CPU interrupt request is disabled.
2	SSU RXI1	0	R/W	
1	SSU TXI0	0	R/W	
0	SSU RXI0	0	R/W	

(8) DREQER7

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ADC ADI	0	R/W	DMA Transfer Enable These bits enable or disable DMA transfer requests and CPU interrupt requests. 0: DMA transfer request is disabled and CPU interrupt request is enabled. 1: DMA transfer request is enabled and CPU interrupt request is disabled.

(9) DREQER8

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	RCAN RM01	0	R/W	DMA Transfer Enable These bits enable or disable DMA transfer requests and CPU interrupt requests. 0: DMA transfer request is disabled and CPU interrupt request is enabled. 1: DMA transfer request is enabled and CPU interrupt request is disabled.
0	RCAN RM00	0	R/W	

7.4 Interrupt Sources

There are six types of interrupt sources: NMI, user break, H-UDI, IRQ, PINT, and on-chip peripheral modules. Each interrupt has a priority level (0 to 16), with 0 the lowest and 16 the highest. When set to level 0, that interrupt is masked at all times.

7.4.1 NMI Interrupts

An NMI interrupt has a priority level of 16 and is accepted at all times. NMI pin inputs are edge-detected, and the NMI sense select bits (NMIS) in interrupt control registers 0 (C0ICR0 and C1ICR0) select whether interrupt requests are detected on the rising edge or falling edge. The CPU that should accept the NMI interrupts can be selected by the interrupt enable control registers (C0INTER and C1INTER).

Though the priority level of the NMI interrupt is 16, the NMI interrupt exception handling sets the interrupt mask bits (I3 to I0) in the status register (SR) to level 15.

7.4.2 User Break Interrupts

A user break interrupt has a priority level of 15, and occurs when a break condition set in the user break controller (UBC) is satisfied. User break interrupt requests are edge-detected and retained until they are accepted. The user break exception handling sets the I3 to I0 bits in the SR to level 15. For user break interrupts, see section 8, User Break Controller (UBC).

7.4.3 H-UDI Interrupts

A user debugging interface (H-UDI) interrupt has a priority level of 15, and occurs at serial input of an H-UDI interrupt instruction. H-UDI interrupt requests are edge-detected and retained until they are accepted. The CPU that should accept the H-UDI interrupts can be selected by the interrupt enable control registers (C0INTER and C1INTER). The H-UDI exception handling sets the I3 to I0 bits in the SR to level 15. For H-UDI interrupts, see section 31, User Debugging Interface (H-UDI).

7.4.4 IRQ Interrupts

IRQ interrupts are input from pins IRQ7 to IRQ0. For an explanation of how to configure pins IRQ7 to IRQ0, see section 27, Pin Function Controller (PFC). For IRQ7 to IRQ0, low-level, falling-edge, rising-edge, or both-edge detection can be selected individually for each pin by the IRQ sense select bits (IRQ71S to IRQ01S and IRQ70S to IRQ00S) in interrupt control registers 1 (C0ICR1 and C1ICR1). The priority level can be set individually in a range from 0 to 15 for each pin by interrupt priority registers 01 and 02 (C0IPR01, C0IPR02, C1IPR01, and C1IPR02). The CPU that should accept the IRQ interrupts can be selected by the IRQ interrupt enable control registers (C0IRQER and C1IRQER).

When low-level detection is used for IRQ interrupts, an interrupt request signal is sent to the INTC while the IRQ7 to IRQ0 pins are low. An interrupt request signal is no longer sent to the INTC when the IRQ7 to IRQ0 pins are driven high. The status of the interrupt requests can be checked by reading the IRQ interrupt request bits (IRQ7F to IRQ0F) in the IRQ interrupt request registers (C0IRQRR and C1IRQRR).

When edge-detection is used for IRQ interrupts, an interrupt request is detected due to change of the IRQ7 to IRQ0 pin states, and an interrupt request signal is sent to the INTC. The result of IRQ interrupt request detection is retained until that interrupt request is accepted. Whether IRQ interrupt requests have been detected or not can be checked by reading IRQ7F to IRQ0F in C0IRQRR and C1IRQRR. The result of IRQ interrupt request detection can be cleared by reading 1 from these bits and then writing 0 to them.

The IRQ interrupt exception handling sets the I3 to I0 bits in the SR to the priority level of the accepted IRQ interrupt.

When returning from the IRQ interrupt exception service routine, execute the RTE instruction after using C0IRQRR and C1IRQRR to ensure that the interrupt request has been cleared, so as not to accidentally receive the interrupt request again.

7.4.5 PINT Interrupts

PINT interrupts are input from pins PINT7 to PINT0. For an explanation of how to configure pins PINT7 to PINT0, see section 27, Pin Function Controller (PFC). Input of the interrupt requests is enabled by the PINT enable bits (PINT7E to PINT0E) in the PINT interrupt enable registers (C0PINTER and C1PINTER). For PINT7 to PINT0, low-level or high-level detection can be selected individually for each pin by the PINT sense select bits (PINT7S to PINT0S) in interrupt control registers 2 (C0ICR2 and C1ICR2). A single priority level in a range from 0 to 15 can be set for all PINT7 to PINT0 interrupts by bits 15 to 12 in interrupt priority registers 05 (C0IPR05 and C1IPR05).

When low-level detection is used for the PINT7 to PINT0 interrupts, an interrupt request signal is sent to the INTC while the PINT7 to PINT0 pins are low. An interrupt request signal is no longer sent to the INTC when the PINT7 to PINT0 pins are driven high. The status of the interrupt requests can be checked by reading the PINT interrupt request bits (PINT7R to PINT0R) in the PINT interrupt request registers (C0PIRR and C1PIRR). The above description also applies to a case in which high-level detection is used, except for the polarity being reversed. The PINT interrupt exception handling sets the I3 to I0 bits in the SR to the priority level of the PINT interrupt.

When returning from the PINT interrupt exception service routine, execute the RTE instruction after using C0PIRR and C1PIRR to ensure that the interrupt request has been cleared, so as not to accidentally receive the interrupt request again.

7.4.6 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are generated by the following on-chip peripheral modules:

- Direct memory access controller (DMAC)
- USB2.0 host/function module (USB)
- Compare match timer (CMT)
- Watchdog timer (WDT)
- Multi-function timer pulse unit 2 (MTU2)
- I^2C bus interface 3 (IIC3)
- Serial communications interface with FIFO (SCIF)
- Serial sound interface with FIFO (SSIF)
- Synchronous serial communications unit (SSU)
- A/D converter (ADC)
- 2D engine (2DG)
- AT attachment packet interface (ATAPI)
- AND/NAND flash memory controller (FLCTL)
- Realtime clock (RTC)
- Controller area network (RCAN-TL1)

As every source is assigned a different interrupt vector, the source does not need to be identified in the exception service routine. A priority level in a range from 0 to 15 can be set for each module by interrupt priority registers 06 to 21 (C0IPR06 to C0IPR21 and C1IPR06 to C1IPR21). The on-chip peripheral module interrupt exception handling sets the I3 to I0 bits in the SR to the priority level of the accepted on-chip peripheral module interrupt.

7.4.7 Inter-Processor Interrupts

Inter-processor interrupts are generated by setting the inter-processor interrupt control registers (C0IPCR15 to C0IPCR08 and C1IPCR15 to C1IPCR08). Interrupts can be generated from CPU0 to CPU1 and vise versa.

When the inter-processor interrupt enable registers (C0IPER and C1IPER) are set, interrupt requests from the inter-processor interrupt control registers are enabled and sent to the CPU.

7.5 Interrupt Exception Handling Vector Tables and Priorities

Table 7.8 lists interrupt sources and their vector numbers, vector table address offsets, and interrupt priorities.

Each interrupt source is allocated a different vector number and vector table address offset. Vector table addresses are calculated from the vector numbers and vector table address offsets. In interrupt exception handling, the exception service routine start address is fetched from the vector table indicated by the vector table address. For details of calculation of the vector table address, see table 6.4.

The priorities of IRQ interrupts, PINT interrupts, and on-chip peripheral module interrupts can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers 01, 02, and 05 to 21 (C0IPR01, C0IPR02, and C0IPR05 to C0IPR21 and C1IPR01, C1IPR02, and C1IPR05 to C1IPR21). However, if two or more interrupts specified by the same IPR setting among C0IPR05 to C0IPR21 and C1IPR05 to C1IPR21 occur, the priorities are defined as shown in the default priorities in table 7.8, and the priorities cannot be changed. A power-on reset assigns priority level 0 to IRQ interrupts, PINT interrupts, 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, they are processed according to the default priorities indicated in table 7.8.

Table 7.8 Interrupt Exception Vectors and Priorities

Interrupt Source		Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Default Priority
NMI		11	H'0000002C to H'0000002F	16	—	High
User break		12	H'00000030 to H'00000033	15	—	
H-UDI		14	H'00000038 to H'0000003B	15	—	
Inter-processor interrupt 15		21	H'00000054 to H'00000057	15		
Inter-processor interrupt 14		22	H'00000058 to H'0000005B	14		
Inter-processor interrupt 13		23	H'0000005C to H'0000005F	13		
Inter-processor interrupt 12		24	H'00000060 to H'00000063	12		
Inter-processor interrupt 11		25	H'00000064 to H'00000067	11		
Inter-processor interrupt 10		26	H'00000068 to H'0000006B	10		
Inter-processor interrupt 09		27	H'0000006C to H'0000006F	9		
Inter-processor interrupt 08		28	H'00000070 to H'00000073	8		
IRQ	IRQ0	64	H'00000100 to H'00000103	0 to 15 (0)	IPR01 (15 to 12)	
	IRQ1	65	H'00000104 to H'00000107	0 to 15 (0)	IPR01 (11 to 8)	
	IRQ2	66	H'00000108 to H'0000010B	0 to 15 (0)	IPR01 (7 to 4)	
	IRQ3	67	H'0000010C to H'0000010F	0 to 15 (0)	IPR01 (3 to 0)	
	IRQ4	68	H'00000110 to H'00000113	0 to 15 (0)	IPR02 (15 to 12)	
	IRQ5	69	H'00000114 to H'00000117	0 to 15 (0)	IPR02 (11 to 8)	
	IRQ6	70	H'00000118 to H'0000011B	0 to 15 (0)	IPR02 (7 to 4)	
	IRQ7	71	H'0000011C to H'0000011F	0 to 15 (0)	IPR02 (3 to 0)	
PINT	PINT0	80	H'00000140 to H'00000143	0 to 15 (0)	IPR05 (15 to 12)	
	PINT1	81	H'00000144 to H'00000147			
	PINT2	82	H'00000148 to H'0000014B			
	PINT3	83	H'0000014C to H'0000014F			
	PINT4	84	H'00000150 to H'00000153			
	PINT5	85	H'00000154 to H'00000157			
	PINT6	86	H'00000158 to H'0000015B			
	PINT7	87	H'0000015C to H'0000015F			Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Default Priority
DMAC	DMAC0	DMINT0	102	H'00000198 to H'0000019B	0 to 15 (0)	IPR06 (15 to 12)	High
	DMAC1	DMINT1	103	H'0000019C to H'0000019F	0 to 15 (0)	IPR06 (11 to 8)	
	DMAC2	DMINT2	104	H'000001A0 to H'000001A3	0 to 15 (0)	IPR06 (7 to 4)	
	DMAC3	DMINT3	105	H'000001A4 to H'000001A7	0 to 15 (0)	IPR06 (3 to 0)	
	DMAC4	DMINT4	106	H'000001A8 to H'000001AB	0 to 15 (0)	IPR07 (15 to 12)	
	DMAC5	DMINT5	107	H'000001AC to H'000001AF	0 to 15 (0)	IPR07 (11 to 8)	
	DMAC6	DMINT6	108	H'000001B0 to H'000001B3	0 to 15 (0)	IPR07 (7 to 4)	
	DMAC7	DMINT7	109	H'000001B4 to H'000001B7	0 to 15 (0)	IPR07 (3 to 0)	
	DMAC8	DMINT8	110	H'000001B8 to H'000001BB	0 to 15 (0)	IPR08 (15 to 12)	
	DMAC9	DMINT9	111	H'000001BC to H'000001BF	0 to 15 (0)	IPR08 (11 to 8)	
	DMAC10	DMINT10	112	H'000001C0 to H'000001C3	0 to 15 (0)	IPR08 (7 to 4)	
	DMAC11	DMINT11	113	H'000001C4 to H'000001C7	0 to 15 (0)	IPR08 (3 to 0)	
	DMAC12	DMINT12	114	H'000001C8 to H'000001CB	0 to 15 (0)	IPR09 (15 to 12)	
	DMAC13	DMINT13	115	H'000001CC to H'000001CF	0 to 15 (0)	IPR09 (11 to 8)	
	DMINTA		116	H'000001D0 to H'000001D3	0 to 15 (0)	IPR09 (7 to 4)	
USB	USBI		117	H'000001D4 to H'000001D7	0 to 15 (0)	IPR09 (3 to 0)	
CMT0	CMI0		118	H'000001D8 to H'000001DB	0 to 15 (0)	IPR10 (15 to 12)	
	CMI1		119	H'000001DC to H'000001DF	0 to 15 (0)	IPR10 (11 to 8)	
CMT1	CMI2		120	H'000001E0 to H'000001E3	0 to 15 (0)	IPR10 (7 to 4)	
	CMI3		121	H'000001E4 to H'000001E7	0 to 15 (0)	IPR10 (3 to 0)	
WDT0	ITI0		122	H'000001E8 to H'000001EB	0 to 15 (0)	IPR11 (15 to 12)	
WDT1	ITI1		123	H'000001EC to H'000001EF	0 to 15 (0)	IPR11 (11 to 8)	
MTU2	MTU0	TGI0A	124	H'000001F0 to H'000001F3	0 to 15 (0)	IPR11 (7 to 4)	
		TGI0B	125	H'000001F4 to H'000001F7			
		TGI0C	126	H'000001F8 to H'000001FB			
		TGI0D	127	H'000001FC to H'000001FF			
		TCI0V	128	H'00000200 to H'00000203	0 to 15 (0)	IPR11 (3 to 0)	
		TGI0E	129	H'00000204 to H'00000207			
		TGI0F	130	H'00000208 to H'0000020B			Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Default Priority
MTU2	MTU1	TGI1A	131	H'0000020C to H'0000020F	0 to 15 (0)	IPR12 (15 to 12)	High
		TGI1B	132	H'00000210 to H'00000213			
		TCI1V	133	H'00000214 to H'00000217	0 to 15 (0)	IPR12 (11 to 8)	
		TCI1U	134	H'00000218 to H'0000021B			
	MTU2	TGI2A	135	H'0000021C to H'0000021F	0 to 15 (0)	IPR12 (7 to 4)	
		TGI2B	136	H'00000220 to H'00000223			
		TCI2V	137	H'00000224 to H'00000227	0 to 15 (0)	IPR12 (3 to 0)	
		TCI2U	138	H'00000228 to H'0000022B			
	MTU3	TGI3A	139	H'0000022C to H'0000022F	0 to 15 (0)	IPR13 (15 to 12)	
		TGI3B	140	H'00000230 to H'00000233			
		TGI3C	141	H'00000234 to H'00000237			
		TGI3D	142	H'00000238 to H'0000023B			
		TCI3V	143	H'0000023C to H'0000023F	0 to 15 (0)	IPR13 (11 to 8)	
	MTU4	TGI4A	144	H'00000240 to H'00000243	0 to 15 (0)	IPR13 (7 to 4)	
		TGI4B	145	H'00000244 to H'00000247			
		TGI4C	146	H'00000248 to H'0000024B			
		TGI4D	147	H'0000024C to H'0000024F			
		TCI4V	148	H'00000250 to H'00000253	0 to 15 (0)	IPR13 (3 to 0)	
SSIF	SSIF0	SSII0	149	H'00000254 to H'00000257	0 to 15 (0)	IPR14 (15 to 12)	
		SSIRTI0	150	H'00000258 to H'0000025B			
	SSIF1	SSII1	151	H'0000025C to H'0000025F	0 to 15 (0)	IPR14 (11 to 8)	
		SSIRTI1	152	H'00000260 to H'00000263			
	SSIF2	SSII2	153	H'00000264 to H'00000267	0 to 15 (0)	IPR14 (7 to 4)	
		SSIRTI2	154	H'00000268 to H'0000026B			
	SSIF3	SSII3	155	H'0000026C to H'0000026F	0 to 15 (0)	IPR14 (3 to 0)	
		SSIRTI3	156	H'00000270 to H'00000273			
	SSIF4	SSII4	157	H'00000274 to H'00000277	0 to 15 (0)	IPR15 (15 to 12)	
		SSIRTI4	158	H'00000278 to H'0000027B			
	SSIF5	SSII5	159	H'0000027C to H'0000027F	0 to 15 (0)	IPR15 (11 to 8)	
		SSIRTI5	160	H'00000280 to H'00000283			Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Default Priority
IIC3	IIC3_0	STPI0	162	H'00000288 to H'0000028B	0 to 15 (0)	IPR16 (15 to 12)	High
		NAKI0	163	H'0000028C to H'0000028F			
		RXI0	164	H'00000290 to H'00000293			
		TXI0	165	H'00000294 to H'00000297			
		TEI0	166	H'00000298 to H'0000029B			
	IIC3_1	STPI1	167	H'0000029C to H'0000029F	0 to 15 (0)	IPR16 (11 to 8)	
		NAKI1	168	H'000002A0 to H'000002A3			
		RXI1	169	H'000002A4 to H'000002A7			
		TXI1	170	H'000002A8 to H'000002AB			
		TEI1	171	H'000002AC to H'000002AF			
	IIC3_2	STPI2	172	H'000002B0 to H'000002B3	0 to 15 (0)	IPR16 (7 to 4)	
		NAKI2	173	H'000002B4 to H'000002B7			
		RXI2	174	H'000002B8 to H'000002BB			
		TXI2	175	H'000002BC to H'000002BF			
		TEI2	176	H'000002C0 to H'000002C3			
	IIC3_3	STPI3	177	H'000002C4 to H'000002C7	0 to 15 (0)	IPR16 (3 to 0)	
		NAKI3	178	H'000002C8 to H'000002CB			
		RXI3	179	H'000002CC to H'000002CF			
		TXI3	180	H'000002D0 to H'000002D3			
		TEI3	181	H'000002D4 to H'000002D7			
SCIF	SCIF0	BRI0	182	H'000002D8 to H'000002DB	0 to 15 (0)	IPR17 (15 to 12)	
		ERI0	183	H'000002DC to H'000002DF			
		RXI0	184	H'000002E0 to H'000002E3			
		TXI0	185	H'000002E4 to H'000002E7			
	SCIF1	BRI1	186	H'000002E8 to H'000002EB	0 to 15 (0)	IPR17 (11 to 8)	
		ERI1	187	H'000002EC to H'000002EF			
		RXI1	188	H'000002F0 to H'000002F3			
		TXI1	189	H'000002F4 to H'000002F7			Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Default Priority
SCIF	SCIF2	BRI2	190	H'000002F8 to H'000002FB	0 to 15 (0)	IPR17 (7 to 4)	High
		ERI2	191	H'000002FC to H'000002FF			
		RXI2	192	H'00000300 to H'00000303			
		TXI2	193	H'00000304 to H'00000307			
	SCIF3	BRI3	194	H'00000308 to H'0000030B	0 to 15 (0)	IPR17 (3 to 0)	
		ERI3	195	H'0000030C to H'0000030F			
		RXI3	196	H'00000310 to H'00000313			
		TXI3	197	H'00000314 to H'00000317			
	SCIF4	BRI4	198	H'00000318 to H'0000031B	0 to 15 (0)	IPR18 (15 to 12)	
		ERI4	199	H'0000031C to H'0000031F			
		RXI4	200	H'00000320 to H'00000323			
		TXI4	201	H'00000324 to H'00000327			
	SCIF5	BRI5	202	H'00000328 to H'0000032B	0 to 15 (0)	IPR18 (11 to 8)	
		ERI5	203	H'0000032C to H'0000032F			
		RXI5	204	H'00000330 to H'00000333			
		TXI5	205	H'00000334 to H'00000337			
SSU	SSU0	SSERI0	206	H'00000338 to H'0000033B	0 to 15 (0)	IPR19 (15 to 12)	
		SSRXI0	207	H'0000033C to H'0000033F			
		SSTXI0	208	H'00000340 to H'00000343			
	SSU1	SSERI1	209	H'00000344 to H'00000347	0 to 15 (0)	IPR19 (11 to 8)	
		SSRXI1	210	H'00000348 to H'0000034B			
		SSTXI1	211	H'0000034C to H'0000034F			
ADC	ADI		212	H'00000350 to H'00000353	0 to 15 (0)	IPR19 (7 to 4)	
2DG	BLT interrupt		213	H'00000354 to H'00000357	0 to 15 (0)	IPR19 (3 to 0)	
	Output interrupt		214	H'00000358 to H'0000035B			
ATAPI	ATAPII		215	H'0000035C to H'0000035F	0 to 15 (0)	IPR20 (15 to 12)	
FLCTL	FLSTEI		216	H'00000360 to H'00000363	0 to 15 (0)	IPR20 (11 to 8)	
	FLTENDI		217	H'00000364 to H'00000367			
	FLTREQ0I		218	H'00000368 to H'0000036B			
	FLTREQ1I		219	H'0000036C to H'0000036F			Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Default Priority
RTC	ARM		220	H'00000370 to H'00000373	0 to 15 (0)	IPR20 (7 to 4)	High
	PRD		221	H'00000374 to H'00000377			
	CUP		222	H'00000378 to H'0000037B			
RCAN	RCAN0	ERS0	226	H'00000388 to H'0000038B	0 to 15 (0)	IPR21 (15 to 12)	
		OVR0	227	H'0000038C to H'0000038F			
		RM00	228	H'00000390 to H'00000393			
		RM10	229	H'00000394 to H'00000397			
		SLE0	230	H'00000398 to H'0000039B			
	RCAN1	ERS1	231	H'0000039C to H'0000039F	0 to 15 (0)	IPR21 (11 to 8)	
		OVR1	232	H'000003A0 to H'000003A3			
		RM01	233	H'000003A4 to H'000003A7			
		RM11	234	H'000003A8 to H'000003AB			
		SLE1	235	H'000003AC to H'000003AF			Low

7.6 Operation

7.6.1 Interrupt Operation Sequence

The interrupt operation sequence is described below. Figure 7.2 shows the operation flow.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest-priority interrupt from the sent interrupt requests, according to the priority levels set in interrupt priority registers 01, 02, and 05 to 21 (C0IPR01, C0IPR02, and C0IPR05 to C0IPR21 and C1IPR01, C1IPR02, and C1IPR05 to C1IPR21). Lower priority interrupts are ignored*. If two of more interrupts have the same priority level or if two or more interrupts specified by the same IPR setting occur, the interrupt with the highest priority is selected, according to the default priorities shown in table 7.8.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt mask level bits (I3 to I0) in the status register (SR) of the CPU. If the priority level is equal to or lower than the level set in bits I3 to I0, the interrupt is ignored. Only when the 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. The CPU detects the interrupt request sent from the interrupt controller when the CPU decodes the instruction to be executed. Instead of executing the decoded instruction, the CPU starts interrupt exception handling (figure 7.4).
5. The exception service routine start address is fetched from the exception handling vector table corresponding to the accepted interrupt.
6. The status register (SR) is saved onto the stack, and the priority level of the accepted interrupt is written to bits I3 to I0 in the SR.
7. The program counter (PC) is saved onto the stack.
8. The CPU jumps to the fetched exception service routine start address and starts executing the program. The jump that occurs is not a delayed branch.

Notes: The interrupt source flag should be cleared in the interrupt handler. After clearing the interrupt source flag, "time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in the SR, and sends interrupt request signal to CPU" shown in table 7.9 is required before the interrupt source sent to the CPU is actually cancelled. 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.

* Interrupt requests that are set for edge-detection are held pending until they are accepted. IRQ interrupts, however, can be cancelled by accessing the IRQ interrupt request registers (C0IRQRR and C1IRQRR). For details, see section 7.4.4, IRQ Interrupts.
  Interrupts held pending due to edge-detection are cleared by a power-on reset.

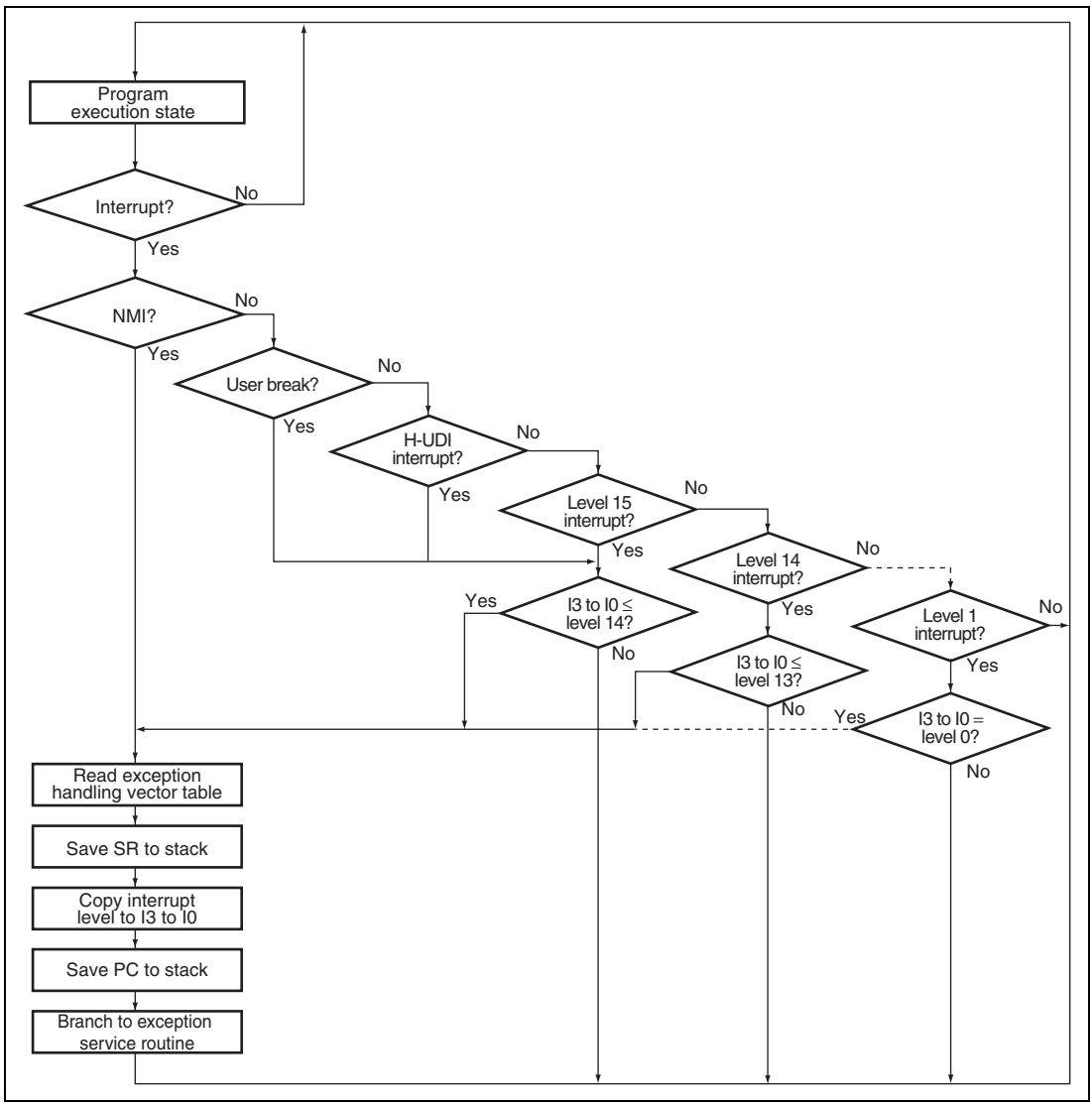
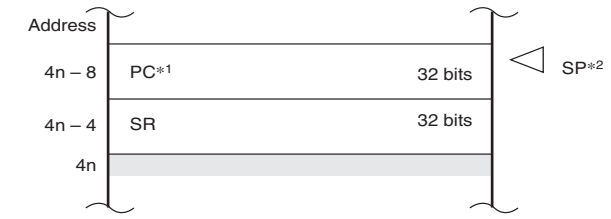


Figure 7.2 Interrupt Operation Flow

7.6.2 Stack Status after Interrupt Exception Handling

Figure 7.3 shows the stack status after interrupt exception handling.



Notes: 1. PC: Start address of the next instruction (return destination instruction) after the executed instruction
2. Always make sure that SP is a multiple of 4.

Figure 7.3 Stack Status after Interrupt Exception Handling

7.7 Interrupt Response Time

Table 7.9 lists the interrupt response time, which is the time from the occurrence of an interrupt request until the interrupt exception handling starts and fetching of the first instruction in the exception service routine begins. The interrupt processing operations differ in the cases when banking is disabled, when banking is enabled without register bank overflow, and when banking is enabled with register bank overflow. Figures 7.4 and 7.5 show examples of pipeline operation when banking is disabled. Figures 7.6 and 7.7 show examples of pipeline operation when banking is enabled without register bank overflow. Figures 7.8 and 7.9 show examples of pipeline operation when banking is enabled with register bank overflow.

Table 7.9 Interrupt Response Time

			Number of States*1						
Item			NMI	User Break	H-UDI	IRQ, PINT	USB	Peripheral Module (except USB)	Remarks
Time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU			2 Incyc + 2 Bcyc + 1 Pcyc	3 Incyc	2 Incyc + 1 Pcyc	2 Incyc + 3 Bcyc + 1 Pcyc	2 Incyc + 4 Bcyc	2 Incyc + 2 Bcyc	
Time from input of interrupt request signal to CPU until sequence currently being executed is completed, interrupt exception handling starts, and first instruction in exception service routine is fetched	No register banking used	Min.	3 Icyc + m1 + m2						Min. is obtained when the interrupt wait time is zero. Max. is obtained when a higher-priority interrupt request has occurred during interrupt exception handling.
		Max.	4 Icyc + 2 (m1 + m2) + m3						
	Register banking used without register bank overflow	Min.	—		3 Icyc + m1 + m2				Min. is obtained when the interrupt wait time is zero. Max. is obtained when an interrupt request has occurred during execution of the RESBANK instruction.
		Max.	—		12 Icyc + m1 + m2				
	Register banking used with register bank overflow	Min.	—		3 Icyc + m1 + m2				Min. is obtained when the interrupt wait time is zero. Max. is obtained when an interrupt request has occurred during execution of the RESBANK instruction.
		Max.	—		3 Icyc + m1 + m2 + 19(m4)				

Item			Number of States*1 NMI	User Break	H-UDI	IRQ, PINT	USB	Peripheral Module (except USB)	Remarks
Response time	No register banking used	Min.	5 Incyc + 2 Bcyc + 1 Pcyc + m1 + m2	6 Incyc + m1 + m2	5 Incyc + 1 Pcyc + m1 + m2	5 Incyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Incyc + 4 Bcyc + m1 + m2	5 Incyc + 2 Bcyc + m1 + m2	200-MHz operation*2: 0.040 to 0.110 µs
		Max.	6 Incyc + 2 Bcyc + 1 Pcyc + 2(m1 + m2) + m3	7 Incyc + 2(m1 + m2) + m3	6 Incyc + 1 Pcyc + 2(m1 + m2) + m3	6 Incyc + 3 Bcyc + 1 Pcyc + 2(m1 + m2) + m3	6 Incyc + 4 Bcyc + 2(m1 + m2) + m3	6 Incyc + 2 Bcyc + 2(m1 + m2) + m3	200-MHz operation*2: 0.060 to 0.130 µs
	Register banking used without register bank overflow	Min.	—	—	5 Incyc + 1 Pcyc + m1 + m2	5 Incyc + 3 Bcyc + 1 Pcyc + m1+m2	5 Incyc + 4 Bcyc + m1 + m2	5 Incyc + 2 Bcyc + m1 + m2	200-MHz operation*2: 0.070 to 0.110 µs
		Max.	—	—	14 Incyc + 1 Pcyc + m1 + m2	14 Incyc + 3 Bcyc + 1 Pcyc + m1 + m2	14 Incyc + 4 Bcyc + m1 + m2	14 Incyc + 2 Bcyc + m1 + m2	200-MHz operation*2: 0.120 to 0.155 µs
	Register banking used with register bank overflow	Min.	—	—	5 Incyc + 1 Pcyc + m1 + m2	5 Incyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Incyc + 4 Bcyc + m1 + m2	5 Incyc + 2 Bcyc + m1 + m2	200-MHz operation*2: 0.065 to 0.110 µs
		Max.	—	—	5 Incyc + 1 Pcyc + m1 + m2 + 19(m4)	5 Incyc + 3 Bcyc + 1 Pcyc + m1 + m2 + 19(m4)	5 Incyc + 4 Bcyc + m1 + m2 + 19(m4)	5 Incyc + 2 Bcyc + m1 + m2 + 19(m4)	200-MHz operation*2: 0.160 to 0.205 µs

Notes: m1 to m4 are the number of states needed for the following memory accesses.

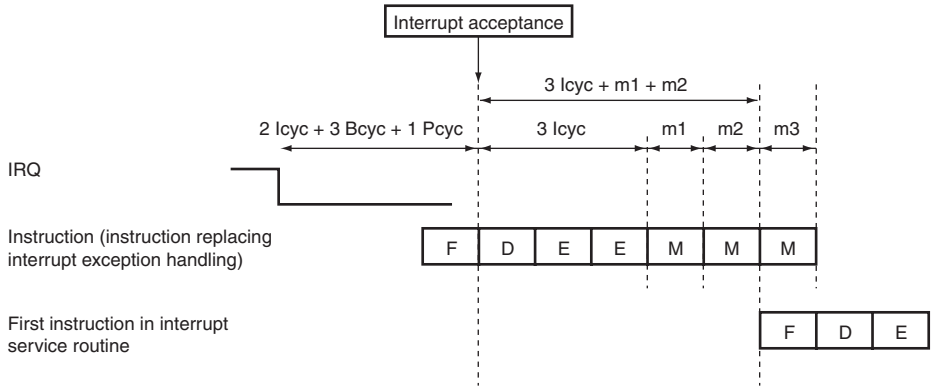
m1: Vector address read (longword read)

m2: SR save (longword write)

m3: PC save (longword write)

m4: Restoration of banked registers (R0 to R14, GBR, MACH, MACL, and PR) from the stack

1. n in Incyc indicates the number (0 or 1) of the CPU to which an interrupt request is sent.
2. Case where Iϕ:Bϕ:Pϕ = 200 MHz:66 MHz:33 MHz and m1 = m2 = m3 = m4 = 1 Incyc



[Legend]
m1: Vector address read
m2: Saving of SR (stack)
m3: Saving of PC (stack)
F: Instruction fetch. Instruction is fetched from memory in which program is stored.
D: Instruction decoding. Fetched instruction is decoded.
E: Instruction execution. Data operation or address calculation is performed in accordance with the result of decoding.
M: Memory access. Memory data access is performed.

Figure 7.4 Example of Pipeline Operation when IRQ Interrupt is Accepted (No Register Banking)

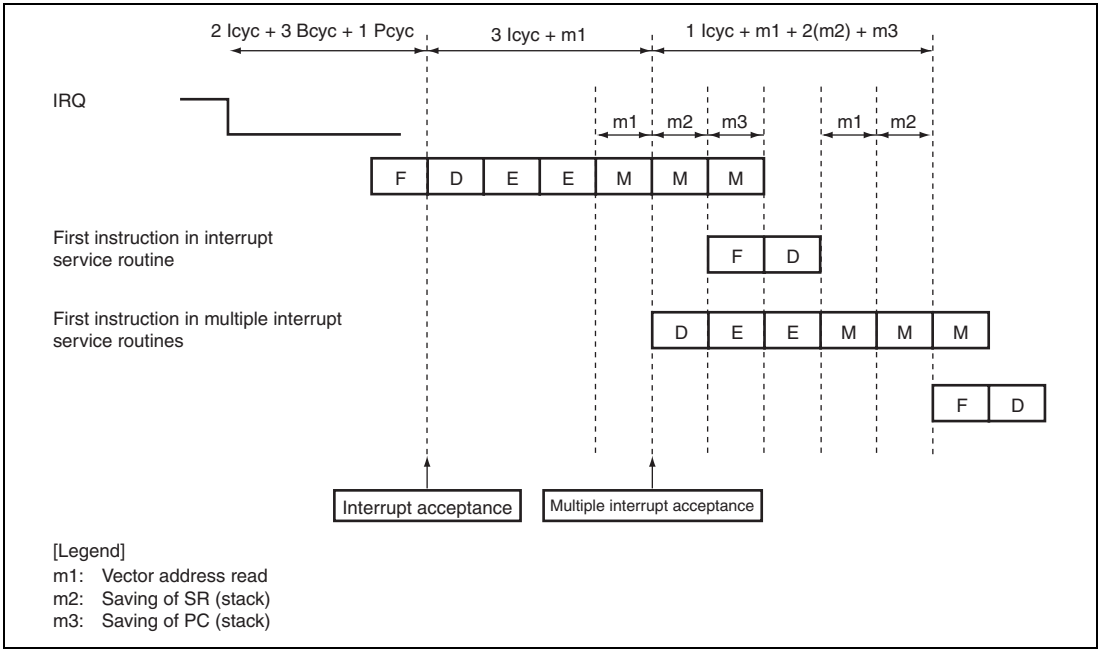


Figure 7.5 Example of Pipeline Operation for Multiple Interrupts (No Register Banking)

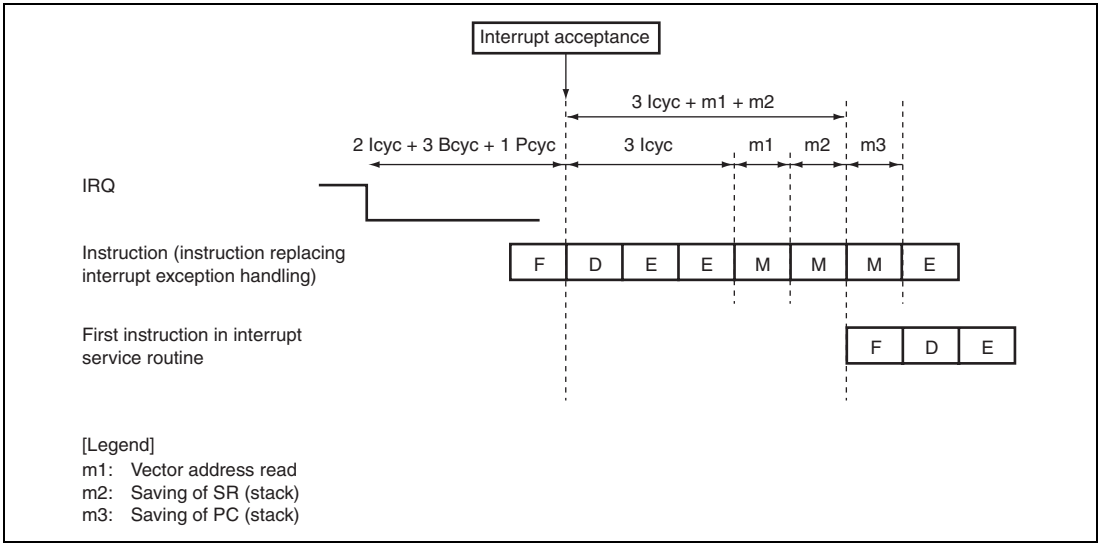


Figure 7.6 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking without Register Bank Overflow)

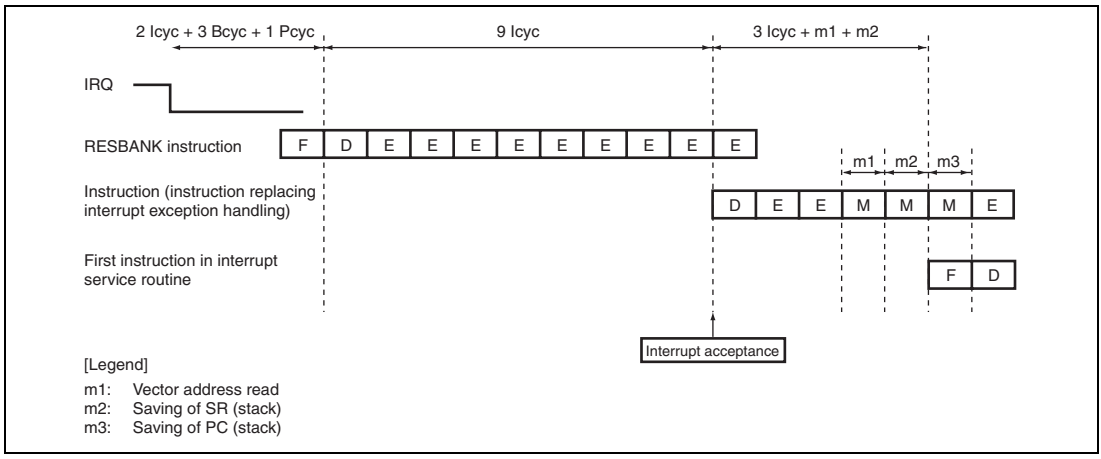


Figure 7.7 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking without Register Bank Overflow)

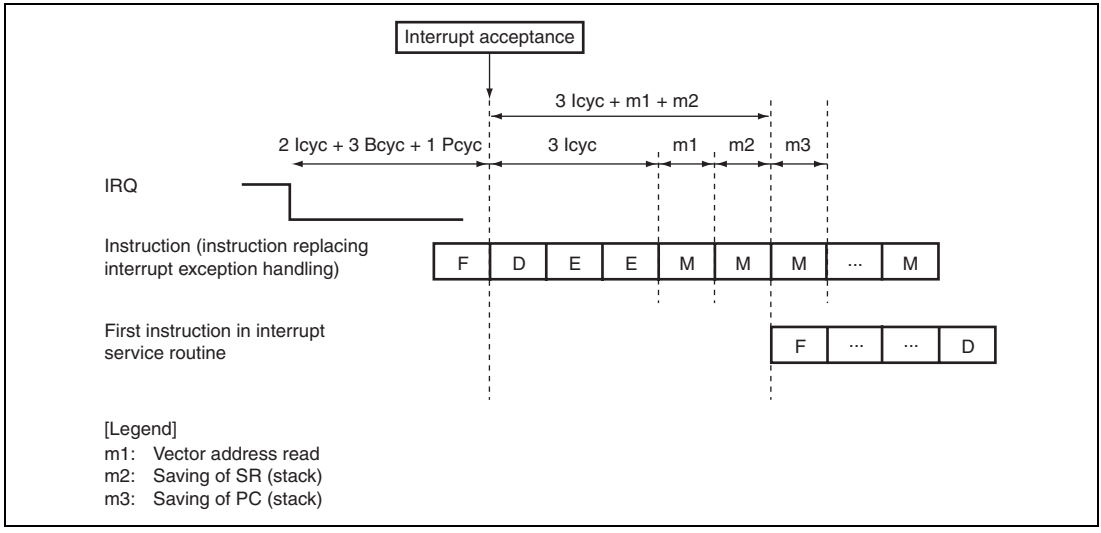


Figure 7.8 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking with Register Bank Overflow)

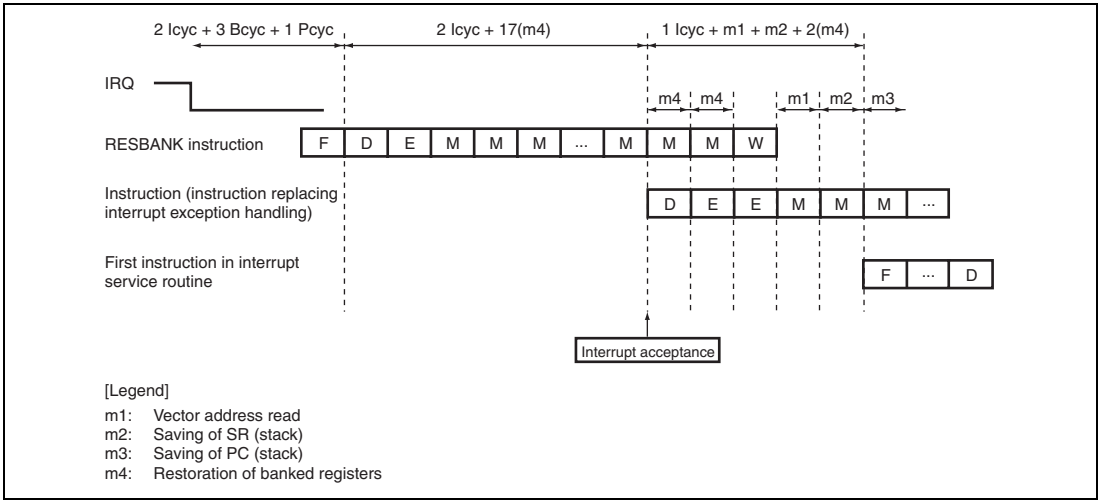


Figure 7.9 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking with Register Bank Overflow)

7.8 Register Banks

The LSI has two sets of 15 register banks, one set for each CPU core, to enable saving and restoring of register data at high speed during interrupt handling. Figure 7.10 shows the register bank configuration.

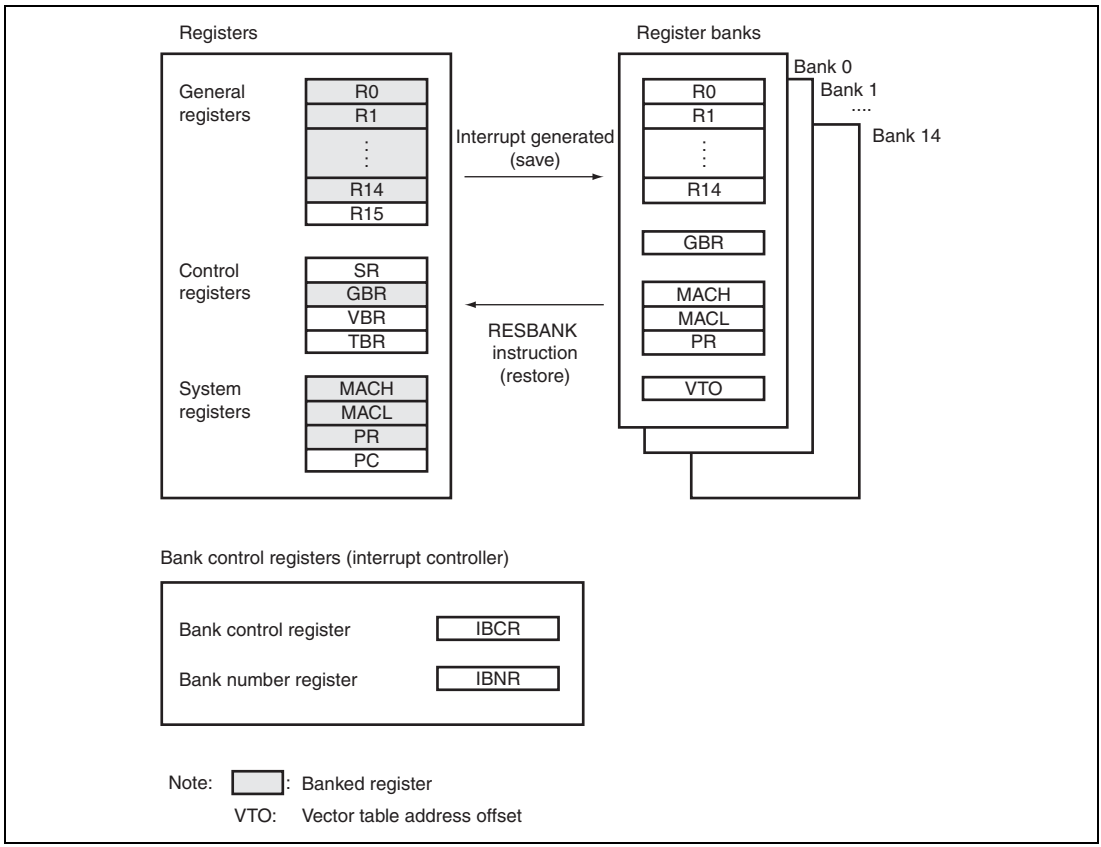


Figure 7.10 Overview of Register Bank Configuration

7.9 Register Banks and Bank Control Registers

(1) Banked Registers

The general registers (R0 to R14), global base register (GBR), multiply-and-accumulate registers (MACH and MACL), procedure register (PR), and the vector table address offset are banked.

(2) Input/Output of Banks

This LSI has 15 register banks, bank 0 to bank 14. Register banks are stacked in first-in last-out (FILO) sequence. Saving takes place in order, beginning from bank 0, and restoration takes place in the reverse order, beginning from the last bank saved to.

7.9.1 Bank Save and Restore Operations

(1) Saving to Bank

Figure 7.11 shows register bank save operation. The following operation is performed when the CPU accepts an interrupt and the use of register banks is enabled for that interrupt.

a. Assume that the values of the bank number bits (BN) in the bank number registers (C0IBNR and C1IBNR) are i before the interrupt is generated.
b. The values in registers R0 to R14, GBR, MACH, MACL, and PR, and the vector table address offset (VTO) of the accepted interrupt are saved to bank i indicated by BN.
c. The BN value is incremented by 1.

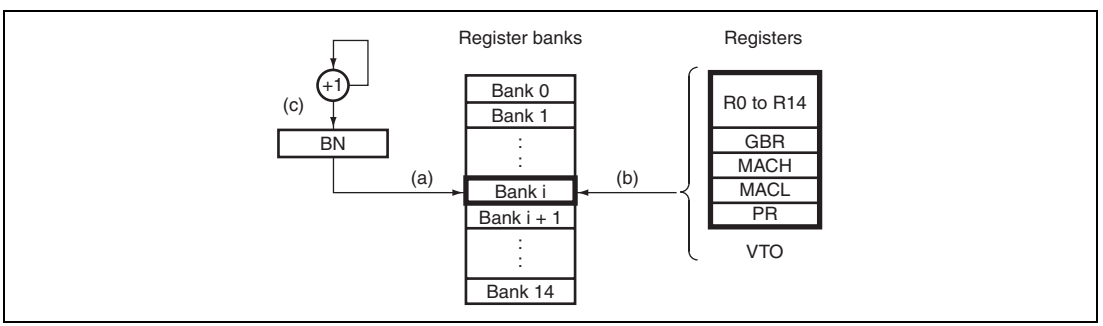


Figure 7.11 Bank Save Operation

Figure 7.12 shows the timing for saving to a register bank. Saving to a register bank takes place between the start of interrupt exception handling and the start of fetching the first instruction in the exception service routine.

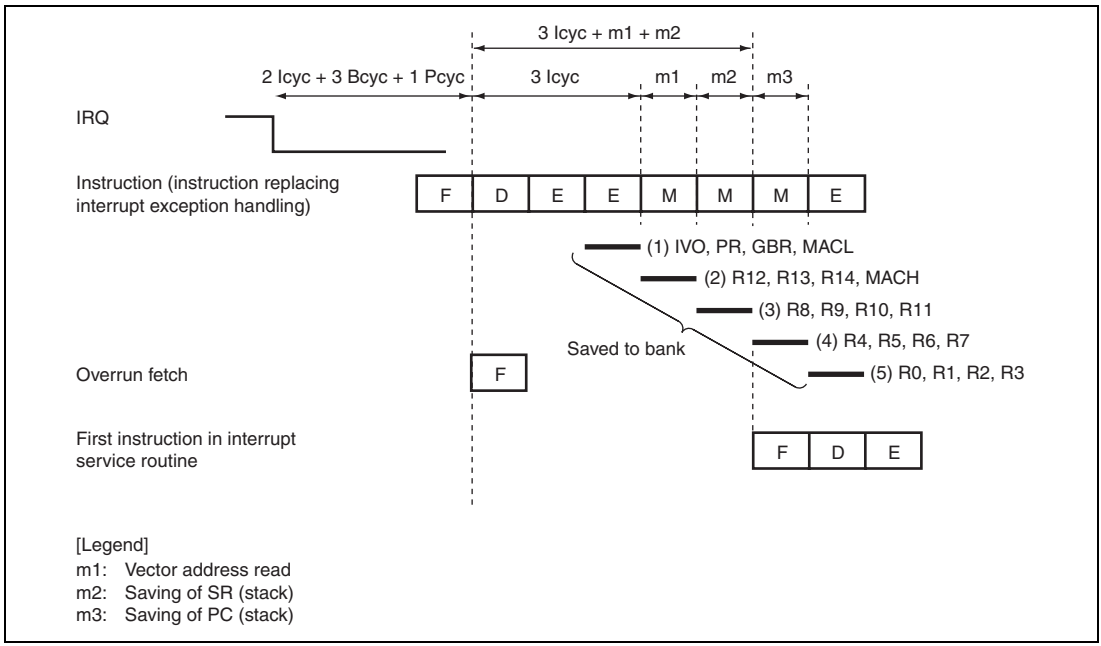


Figure 7.12 Bank Save Timing

(2) Restoration from Bank

The RESBANK (restore from register bank) instruction is used to restore data saved in a register bank. After restoring data from the register banks with the RESBANK instruction at the end of the interrupt service routine, execute the RTE instruction to return from the exception handling.

7.9.2 Save and Restore Operations after Saving to All Banks

If the CPU accepts an interrupt and the use of the register banks is enabled for that interrupt when saving to all register banks has been performed, automatic saving to the stack is performed instead of register bank saving if the BOVE bits in the bank number registers (C0IBNR and C1IBNR) are cleared to 0. If the BOVE bits in C0IBNR and C1IBNR are set to 1, a register bank overflow exception occurs and data is not saved to the stack.

Saving to the stack and restoration from the stack take place as described below:

(1) Saving to Stack

1. The status register (SR) and program counter (PC) are saved to the stack during interrupt exception handling.
2. The values in the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are saved to the stack. The register values are saved to the stack in the order of MACL, MACH, GBR, PR, R14, R13, …, R1, and R0.
3. The register bank overflow bit (BO) in the SR is set to 1.
4. The bank number bits (BN) in the bank number registers (C0IBNR and C1IBNR) remain set to the maximum value of 15.

(2) Restoration from Stack

When the RESBANK (restore from register bank) instruction is executed with the register bank overflow bit (BO) in the SR set to 1, the following operation is performed:

1. The values in the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are restored from the stack. The register values are restored from the stack in the order of R0, R1, …, R13, R14, PR, GBR, MACH, and MACL.
2. The bank number bits (BN) in the bank number registers (C0IBNR and C1IBNR) remain set to the maximum value of 15.

7.9.3 Register Bank Exceptions

There are two types of register bank exceptions (register bank errors): register bank overflow and register bank underflow.

(1) Register Bank Overflow

This exception occurs if, after data has been saved to all of the register banks, the CPU accepts an interrupt and the use of the register banks is enabled for that interrupt, and the BOVE bits in the bank number registers (C0IBNR and C1IBNR) are set to 1. In this case, the bank number bits (BN) in the bank number registers (C0IBNR and C1IBNR) remain set to the bank count of 15 and saving to the register bank is not performed.

(2) Register Bank Underflow

This exception occurs if the RESBANK (restore from register bank) instruction is executed when no data has been saved to the register banks. In this case, the values of R0 to R14, GBR, MACH, MACL, and PR do not change. In addition, the bank number bits (BN) in the bank number registers (C0IBNR and C1IBNR) remain set to 0.

7.10 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. When this happens, the CPU operates as follows:

1. The exception service routine start address is fetched from the exception handling vector table corresponding to the register bank error that has occurred.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. For a register bank overflow, the saved PC value is the start address of the instruction to be executed after the last executed instruction. For a register bank underflow, the saved PC value is the start address of the executed RESBANK instruction. To prevent multiple interrupts from occurring at a register bank overflow, the priority level of the interrupt that caused the register bank overflow is written to the interrupt mask level bits (I3 to I0) of the status register (SR).
4. Program execution starts from the exception service routine start address.

7.11 Data Transfer with Interrupt Request Signals

Interrupt request signals can be used to activate the DMAC and transfer data.

DMA transfer request enable registers 0 to 8 (DREQER0 to DREQER8) are used to specify whether the interrupt request signals start interrupt exception handling or activate the DMAC. When the bits corresponding to on-chip peripheral modules are set to 1, DMA transfer requests are generated; when these bits are set to 0, CPU interrupt requests are generated.

7.12 Usage Note

7.12.1 Timing to Clear an Interrupt Source

The interrupt source flags should be cleared in the interrupt handler. After clearing the interrupt source flag, "time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in the SR, and sends interrupt request signal to CPU" shown in table 7.9 is required before the interrupt source sent to the CPU is actually cancelled. 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.

Section 8 User Break Controller (UBC)

The user break controller (UBC) provides functions that simplify program debugging. These functions make it easy to design a self-monitoring debugger, enabling this LSI chip to debug programs without using an in-circuit emulator. Instruction fetch or data read/write of CPU, data size, data contents, address value, and stop timing in the case of instruction fetch are break conditions that can be set in the UBC. Since this LSI uses a Harvard architecture, instruction fetch on the CPU bus (C bus) is performed by issuing bus cycles on the instruction fetch bus (F bus), and data access on the C bus is performed by issuing bus cycles on the memory access bus (M bus). The UBC monitors the C bus and internal bus (I bus).

There are two UBCs: UBC0, which monitors the operation of CPU0, and UBC1, which monitors the operation of CPU1. These UBCs are quite the same. The control registers of UBC0 and UBC1 are mapped to the same addresses, but the registers for UBC0 are accessed when access from CPU0 is made and the registers for UBC1 are accessed when access from CPU1 is made. In this section, UBC0 and UBC1 are collectively called UBC.

8.1 Features

1. The following break comparison conditions can be set.
   Number of break channels: two channels (channels 0 and 1)
   User break can be requested as the independent condition on channels 0 and 1.
   — Address
     Comparison of the 32-bit address is maskable in 1-bit units.
     One of the three address buses (F address bus (FAB), M address bus (MAB), and I address bus (IAB)) can be selected.
   — Data
     Comparison of the 32-bit data is maskable in 1-bit units.
     One of the two data buses (M data bus (MDB) and I data bus (IDB)) can be selected.
   — Bus cycle
     Instruction fetch (only when C bus is selected) or data access
   — Read/write
   — Operand size
     Byte, word, and longword
2. In an instruction fetch cycle, it can be selected whether break is set before or after execution of an instruction.
3. When a break condition is satisfied, a trigger signal can be output from the $\overline{\text{UBCTRG}}$ pin.

Figure 8.1 shows a block diagram of the UBC.

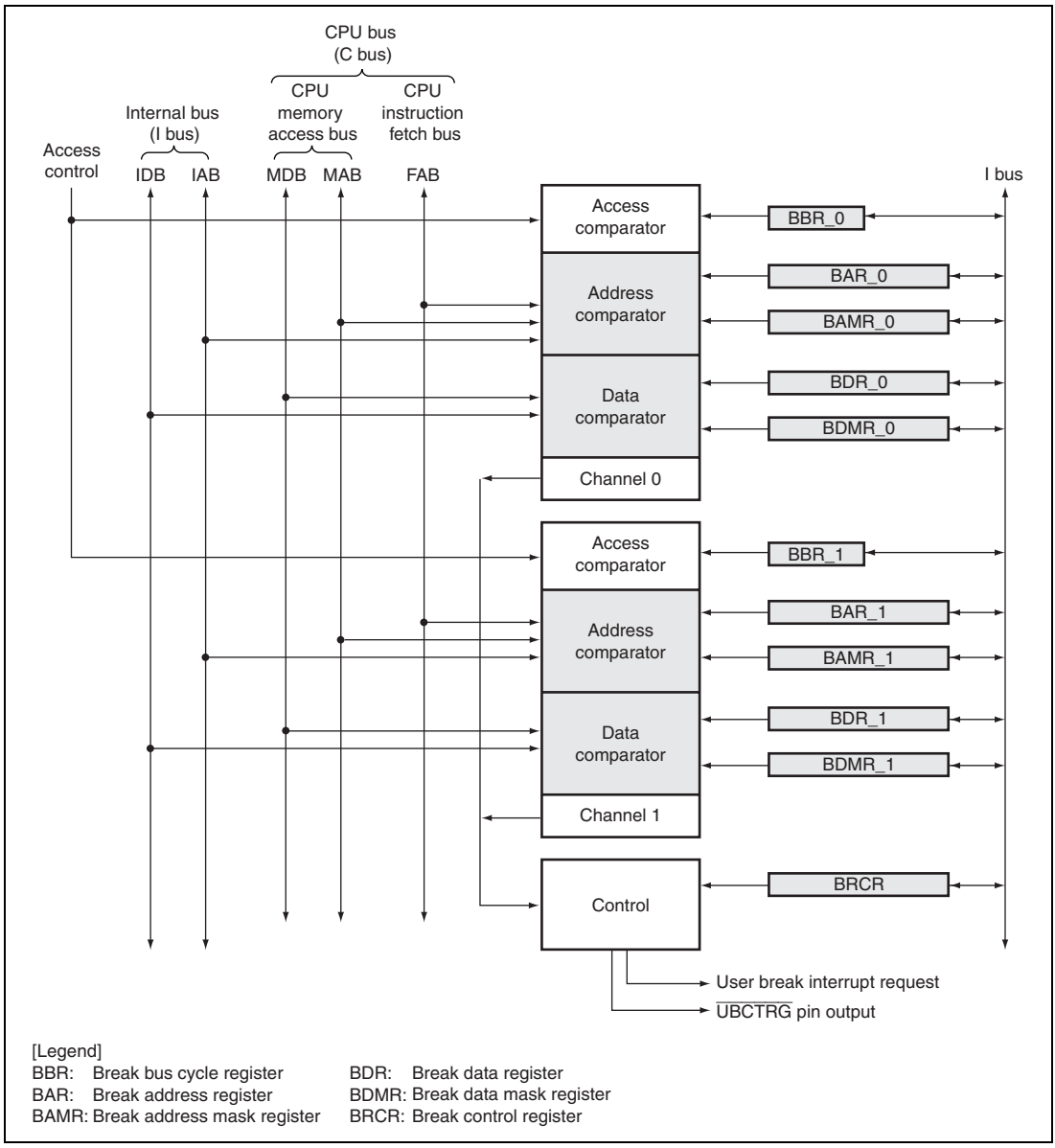


Figure 8.1 Block Diagram of UBC (for One CPU)

8.2 Input/Output Pin

Table 8.1 shows the pin configuration of the UBC.

Table 8.1 Pin Configuration

Pin Name	Symbol	I/O	Function
UBC trigger	$\overline{\text{UBCTRG}}$	Output	Indicates that a setting condition is satisfied on any one of channels 0 and 1 of UBC0 and UBC1.

8.3 Register Descriptions

The UBC has the following registers: five registers for each channel and a control register common to channels 0 and 1. These registers are provided for each of UBC0 and UBC1.

The channel of the UBC registers is indicated as follows: for example, BAR_0 represents the BAR register for channel 0.

Table 8.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Break address register_0	BAR_0	R/W	H'00000000	H'FFFC0400	32
	Break address mask register_0	BAMR_0	R/W	H'00000000	H'FFFC0404	32
	Break bus cycle register_0	BBR_0	R/W	H'0000	H'FFFC04A0	16
	Break data register_0	BDR_0	R/W	H'00000000	H'FFFC0408	32
	Break data mask register_0	BDMR_0	R/W	H'00000000	H'FFFC040C	32
1	Break address register_1	BAR_1	R/W	H'00000000	H'FFFC0410	32
	Break address mask register_1	BAMR_1	R/W	H'00000000	H'FFFC0414	32
	Break bus cycle register_1	BBR_1	R/W	H'0000	H'FFFC04B0	16
	Break data register_1	BDR_1	R/W	H'00000000	H'FFFC0418	32
	Break data mask register_1	BDMR_1	R/W	H'00000000	H'FFFC041C	32
Common	Break control register	BRCR	R/W	H'00000000	H'FFFC04C0	32

8.3.1 Break Address Register (BAR)

BAR is a 32-bit readable/writable register. BAR specifies the address used as a break condition in each channel. Control bits CD1 and CD0 in the break bus cycle register (BBR) select one of the three address buses for a break condition.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA31 to BA0	H'00000000	R/W	Break Address Store an address on the CPU address bus (FAB or MAB) or IAB specifying break conditions. When the C bus and instruction fetch cycle are selected by BBR, specify an FAB address in bits BA31 to BA0. When the C bus and data access cycle are selected by BBR, specify an MAB address in bits BA31 to BA0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR to 0.

8.3.2 Break Address Mask Register (BAMR)

BAMR is a 32-bit readable/writable register. BAMR specifies the bits to be masked of the break address bits specified by BAR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM31	BAM30	BAM29	BAM28	BAM27	BAM26	BAM25	BAM24	BAM23	BAM22	BAM21	BAM20	BAM19	BAM18	BAM17	BAM16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM15	BAM14	BAM13	BAM12	BAM11	BAM10	BAM9	BAM8	BAM7	BAM6	BAM5	BAM4	BAM3	BAM2	BAM1	BAM0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM31 to BAM0	H'00000000	R/W	Break Address Mask Specify the bits to be masked of the break address bits specified by BAR (BA31 to BA0). 0: Break address bit BAn is included in the break condition. 1: Break address bit BAn is masked and not included in the break condition. Note: n = 31 to 0

8.3.3 Break Data Register (BDR)

BDR is a 32-bit readable/writable register. Control bits CD1 and CD0 in the break bus cycle register (BBR) select one of the two data buses for a break condition.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BD31	BD30	BD29	BD28	BD27	BD26	BD25	BD24	BD23	BD22	BD21	BD20	BD19	BD18	BD17	BD16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BD15	BD14	BD13	BD12	BD11	BD10	BD9	BD8	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BD31 to BD0	H'00000000	R/W	Break Data Bits Store data which specifies a break condition. If the I bus is selected in BBR, specify the break data on IDB in bits BD31 to BD0. If the C bus is selected in BBR, specify the break data on MDB in bits BD31 to BD0.

Notes: 1. Set the operand size when specifying a value on a data bus as the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 31 to 24, 23 to 16, 15 to 8, and 7 to 0 in BDR as the break data. Similarly, when the word size is selected, the same word data must be set in bits 31 to 16 and 15 to 0.

8.3.4 Break Data Mask Register (BDMR)

BDMR is a 32-bit readable/writable register. BDMR specifies the bits to be masked of the break data bits specified by BDR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDM31	BDM30	BDM29	BDM28	BDM27	BDM26	BDM25	BDM24	BDM23	BDM22	BDM21	BDM20	BDM19	BDM18	BDM17	BDM16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDM15	BDM14	BDM13	BDM12	BDM11	BDM10	BDM9	BDM8	BDM7	BDM6	BDM5	BDM4	BDM3	BDM2	BDM1	BDM0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDM31 to BDM0	H'00000000	R/W	Break Data Mask Specify bits to be masked of the break data bits specified by BDR (BD31 to BD0). 0: Break data bit BDn is included in the break condition. 1: Break data bit BDn is masked and not included in the break condition. Note: n = 31 to 0

Notes: 1. Set the operand size when specifying a value on a data bus as the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 31 to 24, 23 to 16, 15 to 8, and 7 to 0 in BDMR as the break mask data. Similarly, when the word size is selected, the same word data must be set in bits 31 to 16 and 15 to 0.

8.3.5 Break Bus Cycle Register (BBR)

BBR is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) including or excluding of the data bus value, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID	DBE	-	-	-	CP	CD[1:0]		ID[1:0]		RW[1:0]		SZ[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID	0	R/W	User Break Interrupt Disable Disables or enables user break interrupt requests when a break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12	DBE	0	R/W	Data Break Enable Selects whether the data bus condition is included in the break conditions. 0: Data bus condition is not included in break conditions. 1: Data bus condition is included in break conditions.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	CP	0	R/W	I Bus Select Select permission or prohibition when the bus cycle of the break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). 0: The condition of the I bus cycle is not compared. 1: The condition of the I bus cycle is compared.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select Select the C bus cycle or I bus cycle as the bus cycle of the break condition. 00: Condition comparison is not performed. 01: Break condition is the C bus (F bus or M bus) cycle. 10: Break condition is the I bus cycle. 11: Break condition is the C bus (F bus or M bus) cycle.
5, 4	ID[1:0]	00	R/W	Instruction Fetch/Data Access Select Select the instruction fetch cycle or data access cycle as the bus cycle of the break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed. 01: Break condition is the instruction fetch cycle. 10: Break condition is the data access cycle. 11: Break condition is the instruction fetch cycle or data access cycle.
3, 2	RW[1:0]	00	R/W	Read/Write Select Select the read cycle or write cycle as the bus cycle of the break condition. 00: Condition comparison is not performed. 01: Break condition is the read cycle. 10: Break condition is the write cycle. 11: Break condition is the read cycle or write cycle.
1, 0	SZ[1:0]	00	R/W	Operand Size Select Select the operand size of the bus cycle for the break condition. 00: Break condition does not include operand size. 01: Break condition is byte access. 10: Break condition is word access. 11: Break condition is longword access.

8.3.6 Break Control Register (BRCR)

BRCR sets the following conditions:

1. Specifies whether a start of user break interrupt exception handling by instruction fetch cycle is set before or after instruction execution.
2. Specifies the pulse width of the $\overline{\text{UBCTRG}}$ output when a break condition is satisfied.

BRCR is a 32-bit readable/writable register that has break condition match flags and bits for setting other break conditions. For the condition match flags of bits 15 to 12, writing 1 is invalid (previous values are retained) and writing 0 is only possible. To clear the flag, write 0 to the flag bit to be cleared and 1 to all other flag bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCMFC 0	SCMFC 1	SCMFD 0	SCMFD 1	-	-	-	-	-	PCB1	PCB0	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R	R	R	R	R

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, 16	CKS[1:0]	00	R/W	Clock Select Specifies the pulse width output to the $\overline{\text{UBCTRG}}$ pin when a break condition is satisfied. 00: Pulse width of $\overline{\text{UBCTRG}}$ is one bus clock cycle. 01: Pulse width of $\overline{\text{UBCTRG}}$ is two bus clock cycles. 10: Pulse width of $\overline{\text{UBCTRG}}$ is four bus clock cycles. 11: Pulse width of $\overline{\text{UBCTRG}}$ is eight bus clock cycles.

Bit	Bit Name	Initial Value	R/W	Description
15	SCMFC0	0	R/W	C Bus Cycle Condition Match Flag 0 When the C bus cycle condition in the break conditions set for channel 0 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 0 does not match. 1: The C bus cycle condition for channel 0 matches.
14	SCMFC1	0	R/W	C Bus Cycle Condition Match Flag 1 When the C bus cycle condition in the break conditions set for channel 1 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 1 does not match. 1: The C bus cycle condition for channel 1 matches.
13	SCMFD0	0	R/W	I Bus Cycle Condition Match Flag 0 When the I bus cycle condition in the break conditions set for channel 0 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 0 does not match. 1: The I bus cycle condition for channel 0 matches.
12	SCMFD1	0	R/W	I Bus Cycle Condition Match Flag 1 When the I bus cycle condition in the break conditions set for channel 1 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 1 does not match. 1: The I bus cycle condition for channel 1 matches.
11 to 7	—	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
6	PCB1	0	R/W	PC Break Select 1 Selects the break timing of the instruction fetch cycle for channel 1 as before or after instruction execution. 0: PC break of channel 1 is generated before instruction execution. 1: PC break of channel 1 is generated after instruction execution.
5	PCB0	0	R/W	PC Break Select 0 Selects the break timing of the instruction fetch cycle for channel 0 as before or after instruction execution. 0: PC break of channel 0 is generated before instruction execution. 1: PC break of channel 0 is generated after instruction execution.
4 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

8.4 Operation

8.4.1 Flow of the User Break Operation

The flow from setting of break conditions to user break exception handling is described below:

1. The break address is set in the break address register (BAR). The masked address bits are set in the break address mask register (BAMR). The break data is set in the break data register (BDR). The masked data bits are set in the break data mask register (BDMR). The bus break conditions are set in the break bus cycle register (BBR). No user break will be generated if any one of the three control bit pairs in BBR (C bus cycle/I bus cycle select, instruction fetch/data access select, and read/write select) is set to 00. The break control settings are made in the bits of the break control register (BRCR). Make sure to set all registers related to breaks before setting BBR, and branch after reading from the last written register. The newly written register values become valid from the instruction at the branch destination.
2. If a the break condition is satisfied, UBC0 (UBC1) sends a user break request to CPU0 (CPU1) through the INTC, sets the C bus condition match flag (SCMFC) or I bus condition match flag (SCMFD) for the appropriate channel, and outputs a pulse to the $\overline{\text{UBCTRG}}$ pin with the width set by the CKS[1:0] bits. Setting the UBID bit in BBR to 1 enables external monitoring of the trigger output without requesting user break interrupts.
3. On receiving a user break interrupt request signal, the INTC determines its priority. Since the user break interrupt has a priority level of 15, it is accepted when the priority level set in the interrupt mask level bits (I3 to I0) of the status register (SR) is 14 or lower. If the I3 to I0 bits are set to a priority level of 15, the user break interrupt is not accepted, but the conditions are checked, and condition match flags are set if the conditions match. For details on ascertaining the priority, see section 7, Interrupt Controller (INTC).
4. Condition match flags (SCMFC and SCMFD) can be used to check which condition has been satisfied. Clear the condition match flags during the user break interrupt exception handling routine. The interrupt occurs again if this operation is not performed.
5. There is a possibility that the break set in channel 0 and the break set in channel 1 occur around the same time. In this case, there will be only one user break request to the INTC, but these two break channel match flags may both be set.
6. When selecting the I bus as the break condition, note as follows:
   — Whether or not the access the CPU issued on the C bus is issued on the I bus depends on the setting of the cache. As regard to the I bus operation that depends on cache conditions, see table 9.8.
   — When a break condition is specified for the I bus, only the data access cycle is monitored. The instruction fetch cycle (including cache update cycle) is not monitored.

— 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 user break interrupt request is to be accepted cannot be clearly defined.

8.4.2 Break on Instruction Fetch Cycle

1. When C bus/instruction fetch/read/word or longword is set in the break bus cycle register (BBR), the break condition is the FAB bus instruction fetch cycle. Whether a break is set before or after the execution of the instruction can be selected with the PCB0 or PCB1 bit in the break control register (BRCR) for the appropriate channel. If an instruction fetch cycle is set as a break condition, clear the BA0 bit in the break address register (BAR) to 0. A break cannot be generated as long as this bit is set to 1.
2. A break for instruction fetch which is set as a break before instruction execution occurs when it is confirmed that the instruction has been fetched and will be executed. This means a break does not occur for 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 user break interrupt request is not received until the execution of the first instruction at the branch destination.

Note: If a branch does not occur at a delayed branch instruction, the subsequent instruction is not recognized as a delay slot.

3. When setting a break condition for break after instruction execution, the instruction that matched the break condition is executed and then the break is generated prior to execution of the next instruction. As with pre-execution breaks, a break does not occur with overrun fetch instructions. When this kind of break is set for a delayed branch instruction and its delay slot, the user break interrupt request is not received until the first instruction at the branch destination.
4. When an instruction fetch cycle is set, the break data register (BDR) 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 setting is invalidated.

8.4.3 Break on Data Access Cycle

1. If the C 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 addresses (and data) of the data access cycles, and a break occurs if the condition is satisfied. For details on the CPU bus cycles issued on the I bus, see paragraph 6 in section 8.4.1, 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 8.3.

Table 8.3 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 (BAR), 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:

   When the data value is included in the break conditions, either longword, word, or byte is specified as the operand size in the break bus cycle register (BBR). 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 the four bytes at bits 31 to 24, 23 to 16, 15 to 8, and 7 to 0 of the break data register (BDR) and break data mask register (BDMR). To specify word data for this case, set the same data in the two words at bits 31 to 16 and 15 to 0.
4. Access by a PREF instruction is handled as read access in longword units without access data. Therefore, if data is included in the break condition, of PREF instruction, a break will not occur.
5. If the data access cycle is selected, the instruction at which the break will occur cannot be determined.

8.4.4 Value of Saved Program Counter

When a user break interrupt request is received, the address of the instruction from where execution is to be resumed is saved to the stack, and the exception handling state is entered. If the C bus (FAB)/instruction fetch cycle is specified as a break condition, the instruction at which the break should occur can be uniquely determined. If the C bus/data access cycle or I bus/data access cycle is specified as a break condition, the instruction at which the break should occur cannot be uniquely determined.

1. When C bus (FAB)/instruction fetch (before instruction execution) is specified as a break condition:

   The address of the instruction that matched the break condition is saved to the stack. 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 instruction is executed, and the branch destination address is saved to the stack.
2. When C bus (FAB)/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 to the stack. 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, the instruction is executed, and the branch destination address is saved to the stack.
3. When C bus/data access cycle or I bus/data access cycle is specified as a break condition:

   The address after executing several instructions of the instruction that matched the break condition is saved to the stack.

8.4.5 Usage Examples

(1) Break Condition Specified for C Bus Instruction Fetch Cycle

(Example 1-1)

- Register specifications

  BAR_0 = H'00000404, BAMR_0 = H'00000000, BBR_0 = H'0054, BAR_1 = H'00008010,
  BAMR_1 = H'00000006, BBR_1 = H'0054,
  BDR_1 = H'00000000, BDMR_1 = H'00000000, BRCR = H'00000020

  <Channel 0>

  Address: H'00000404, Address mask: H'00000000

  Bus cycle: C bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

  <Channel 1>

  Address: H'00008010, Address mask: H'00000006

  Data: H'00000000, Data mask: H'00000000

  Bus cycle: C 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

  BAR_0 = H'00027128, BAMR_0 = H'00000000, BBR_0 = H'005A, BAR_1= H'00031415,
  BAMR_1 = H'00000000, BBR_1 = H'0054,
  BDR_1 = H'00000000, BDMR_1 = H'00000000, BRCR = H'00000000

  <Channel 0>

  Address: H'00027128, Address mask: H'00000000

  Bus cycle: C bus/instruction fetch (before instruction execution)/write/word

  <Channel 1>

  Address: H'00031415, Address mask: H'00000000

  Data: H'00000000, Data mask: H'00000000

  Bus cycle: C bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

  On channel 0, a user break does not occur since instruction fetch is not a write cycle. On channel 1, a user break does not occur since instruction fetch is performed for an even address.

(Example 1-3)

- Register specifications

  BAR_0 = H'00008404, BAMR_0 = H'00000FFF, BBR_0 = H'0054, BAR_1= H'00008010,
  BAMR_1 = H'00000006, BBR_1 = H'0054,
  BDR_1 = H'00000000, BDMR_1 = H'00000000, BRCR = H'00000020

  <Channel 0>

  Address: H'00008404, Address mask: H'00000FFF

  Bus cycle: C bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

  <Channel 1>

  Address: H'00008010, Address mask: H'00000006

  Data: H'00000000, Data mask: H'00000000

  Bus cycle: C 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.

(2) Break Condition Specified for C Bus Data Access Cycle

(Example 2-1)

- Register specifications

  BAR_0 = H'00123456, BAMR_0 = H'00000000, BBR_0 = H'0064, BAR_1= H'000ABCDE,
  BAMR_1 = H'000000FF, BBR_1 = H'106A,
  BDR_1 = H'A512A512, BDMR_1 = H'00000000, BRCR = H'00000000

  <Channel 0>

  Address: H'00123456, Address mask: H'00000000

  Bus cycle: C bus/data access/read (operand size is not included in the condition)

  <Channel 1>

  Address: H'000ABCDE, Address mask: H'000000FF

  Data: H'0000A512, Data mask: H'00000000

  Bus cycle: C bus/data access/write/word

  On channel 0, 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 1, a user break occurs when word H'A512 is written in addresses H'000ABC00 to H'000ABCFE.

(3) Break Condition Specified for I Bus Data Access Cycle

(Example 3-1)

- Register specifications

  BAR_0 = H'00314156, BAMR_0 = H'00000000, BBR_0 = H'0094, BAR_1= H'00055555, BAMR_1 = H'00000000, BBR_1 = H'11A9,
  BDR_1 = H'78787878, BDMR_1 = H'0F0F0F0F, BRCR = H'00000000

  <Channel 0>

  Address: H'00314156, Address mask: H'00000000

  Bus cycle: I bus/instruction fetch/read (operand size is not included in the condition)

  <Channel 1>

  Address: H'00055555, Address mask: H'00000000

  Data: H'00000078, Data mask: H'0000000F

  Bus cycle: I bus/data access/write/byte

  On channel 0, the setting of I bus/instruction fetch is ignored. On channel 1, a user break occurs when the CPU writes byte data H'7x in address H'00055555 on the I bus.

8.5 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. The UBC cannot monitor access to the C bus and I bus cycles in the same channel.
3. When a user break and another exception source occur at the same instruction, which has higher priority is determined according to the priority levels defined in table 6.1. If an exception source with higher priority occurs, the user break interrupt request is not received.
4. Note the following when a break occurs in a delay slot.

   If a pre-execution break is set at a delay slot instruction, the break does not occur before execution of the branch destination.
5. User breaks are disabled during UBC module standby mode. Do not read from or write to the UBC registers during UBC module standby mode; the values are not guaranteed.
6. Do not set an address within an interrupt exception handling routine whose interrupt priority level is at least 15 (including user break interrupts) as a break address.
7. Do not set break after instruction execution for the SLEEP instruction or for the delayed branch instruction where the SLEEP instruction is placed at its delay slot.
8. When setting a break address for a 32-bit instruction, set the address where the upper 16 bits are placed. If the address of the lower 16 bits is set and a break before instruction execution is set as a break condition, the break is handled as a break after instruction execution.
9. Do not set a pre-execution break for the instruction that comes after the DIVU or DIVS instruction. If a pre-execution break is set for the instruction that comes after the DIVU or DIVS instruction and an exception or interrupt occurs during execution of the DIVU or DIVS instruction, a pre-execution break occurs even though execution of the DIVU or DIVS instruction is halted.

Section 9 Cache

9.1 Features

- Capacity
  Instruction cache: 8 Kbytes × 2 cores (CPU0/CPU1)
  Operand cache: 8 Kbytes × 2 cores (CPU0/CPU1)
- Structure: Instructions/data separated, 4-way set associative
- Way lock function (only for operand cache): Way 2 and way 3 are lockable
- Line size: 16 bytes
- Number of entries: 128 entries/way
- Write system: Write-back/write-through selectable
- Replacement method: Least-recently-used (LRU) algorithm

9.1.1 Cache Structure

The cache separates 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.

The address and data sections per way are divided into 128 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 2 Kbytes (16 bytes × 128 entries), which makes a total of 8 Kbytes as a whole cache (four ways).

There are two caches: cache 0 is incorporated in CPU0 and cache 1 is incorporated in CPU1. The two have the same functions.

Although the control registers for cache 0 and cache 1 are allocated to the same address, access from CUP0 will be to cache 0 and access from CPU1 will be to cache 1.

In this section, "cache" inclusively refers to both cache 0 and cache 1.

Figure 9.1 shows the operand cache structure. The instruction cache structure is the same as the operand cache structure except for not having the U bit.

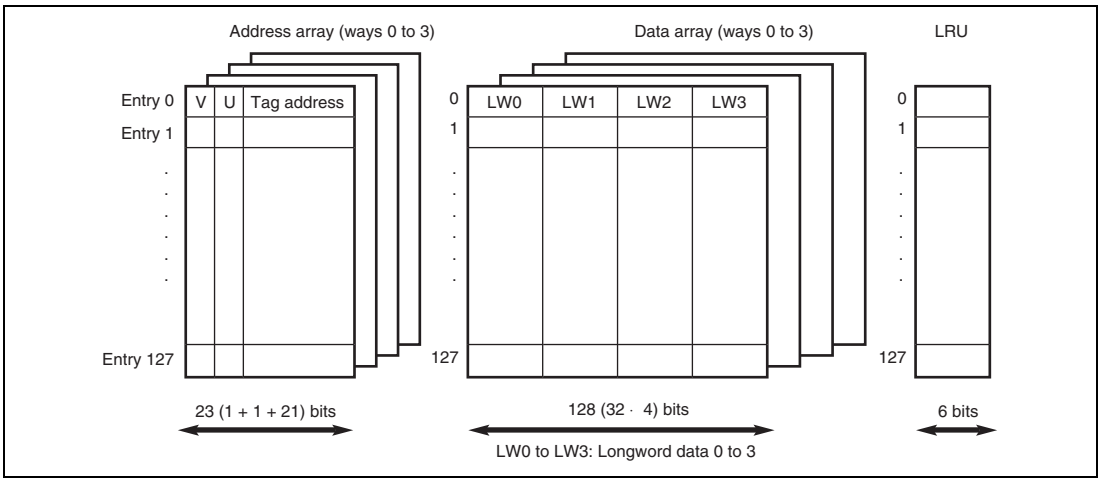


Figure 9.1 Operand Cache Structure

(1) 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 (only for operand cache) 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 tag address holds the physical address used in the external memory access. It consists of 21 bits (address bits 31 to 11) used for comparison during cache searches. In this LSI, the addresses of the cache-enabled space are H'00000000 to H'1FFFFFFF, and therefore the upper three bits of the tag address are cleared to 0.

The V and U bits are initialized to 0 by a power-on reset but not initialized by a manual reset or in standby mode. The tag address is not initialized by a power-on reset, manual reset, or in standby mode.

(2) 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 reset, manual reset, or in standby mode.

(3) LRU

With the 4-way set associative system, up to four instructions or data with the same entry address 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 in table 9.1 when the cache lock function (only for operand cache) is not used (concerning the case where the cache lock function is used, see section 9.2.2, Cache Control Register 2 (CCR2)). If a bit pattern other than those listed in table 9.1 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 9.1.

The LRU bits are initialized to B'000000 by a power-on reset but not initialized by a manual reset or in standby mode.

Table 9.1 LRU Bits and Way Replacement (Cache Lock Function Not Used)

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

9.2 Register Descriptions

The cache has the following registers.

Table 9.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Cache control register 1	CCR1	R/W	H'00000000	H'FFFC1000	32
Cache control register 2	CCR2	R/W	H'00000000	H'FFFC1004	32

9.2.1 Cache Control Register 1 (CCR1)

The instruction cache is enabled or disabled using the ICE bit. The ICF bit controls disabling of all instruction cache entries. The operand cache is enabled or disabled using the OCE bit. The OCF bit controls disabling of all operand cache entries. The WT bit selects either write-through mode or write-back mode for operand cache.

Programs that change the contents of CCR1 should be placed in a cache-disabled space, and a cache-enabled space should be accessed after reading the contents of CCR1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	ICF	-	-	ICE	-	-	-	-	OCF	-	WT	OCE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R	R	R/W	R	R	R	R	R/W	R	R/W	R/W

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.

Bit	Bit Name	Initial Value	R/W	Description
11	ICF	0	R/W	Instruction Cache Flush Writing 1 flushes all instruction cache entries (clears the V and LRU bits of all instruction cache entries to 0). Always reads 0. Write-back to external memory is not performed when the instruction cache is flushed.
10, 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ICE	0	R/W	Instruction Cache Enable Indicates whether the instruction cache function is enabled or disabled. 0: Instruction cache disabled. 1: Instruction cache enabled.
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	OCF	0	R/W	Operand Cache Flush Writing 1 flushes all operand cache entries (clears the V, U, and LRU bits of all operand cache entries to 0). Always reads 0. Write-back to external memory is not performed when the operand cache is flushed.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	WT	0	R/W	Write Through Selects write-back mode or write-through mode. 0: Write-back mode 1: Write-through mode
0	OCE	0	R/W	Operand Cache Enable Indicates whether the operand cache function is enabled or disabled. 0: Cache disabled. 1: Cache enabled.

9.2.2 Cache Control Register 2 (CCR2)

CCR2 is used to enable or disable the cache locking function for operand cache and is valid only in cache locking mode. In cache locking mode, the lock enable bit (LE bit) in CCR2 is set to 1. In 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 9.3. 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 9.4.

Programs that change the contents of CCR2 should be placed in a cache-disabled space, and a cache-enabled space should be accessed after reading the contents of CCR2.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	W3 LOAD*	W3 LOCK	-	-	-	-	-	-	W2 LOAD*	W2 LOCK
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

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

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 Controls cache locking mode. 0: Non-cache locking mode 1: Cache locking mode
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 cache miss 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 =1 in cache locking mode, the data is always loaded into way 2. Under any other condition, the cache miss data is loaded into the way to which LRU points.

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

Table 9.3 Way to Be Replaced When a Cache Miss Occurs in PREF Instruction

LE	W3LOAD*	W3LOCK	W2LOAD*	W2LOCK	Way to Be Replaced
0	×	×	×	×	Decided by LRU (table 9.1)
1	×	0	×	0	Decided by LRU (table 9.1)
1	×	0	0	1	Decided by LRU (table 9.5)
1	0	1	×	0	Decided by LRU (table 9.6)
1	0	1	0	1	Decided by LRU (table 9.7)
1	0	×	1	1	Way 2
1	1	1	0	×	Way 3

[Legend] x: Don't care

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

Table 9.4 Way to Be Replaced When a Cache Miss Occurs in Other Than PREF Instruction

LE	W3LOAD*	W3LOCK	W2LOAD*	W2LOCK	Way to Be Replaced
0	×	×	×	×	Decided by LRU (table 9.1)
1	×	0	×	0	Decided by LRU (table 9.1)
1	×	0	×	1	Decided by LRU (table 9.5)
1	×	1	×	0	Decided by LRU (table 9.6)
1	×	1	×	1	Decided by LRU (table 9.7)

[Legend] x: Don't care

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

Table 9.5 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 9.6 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 9.7 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

9.3 Operation

Operations for the operand cache are described here. Operations for the instruction cache are similar to those for the operand cache except for the address array not having the U bit, and there being no prefetch operation or write operation, or a write-back buffer.

9.3.1 Searching Cache

If the operand cache is enabled (OCE bit in CCR1 is 1), whenever data in a cache-enabled space is accessed, the cache will be searched to see if the desired data is in the cache. Figure 9.2 illustrates the method by which the cache is searched.

Entries are selected using bits 10 to 4 of the address used to access memory and the tag address of that entry is read. At this time, the upper three bits of the tag address are always cleared to 0. Bits 31 to 11 of the address used to access memory are compared with the read tag address. 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 9.2 shows a hit on way 1.

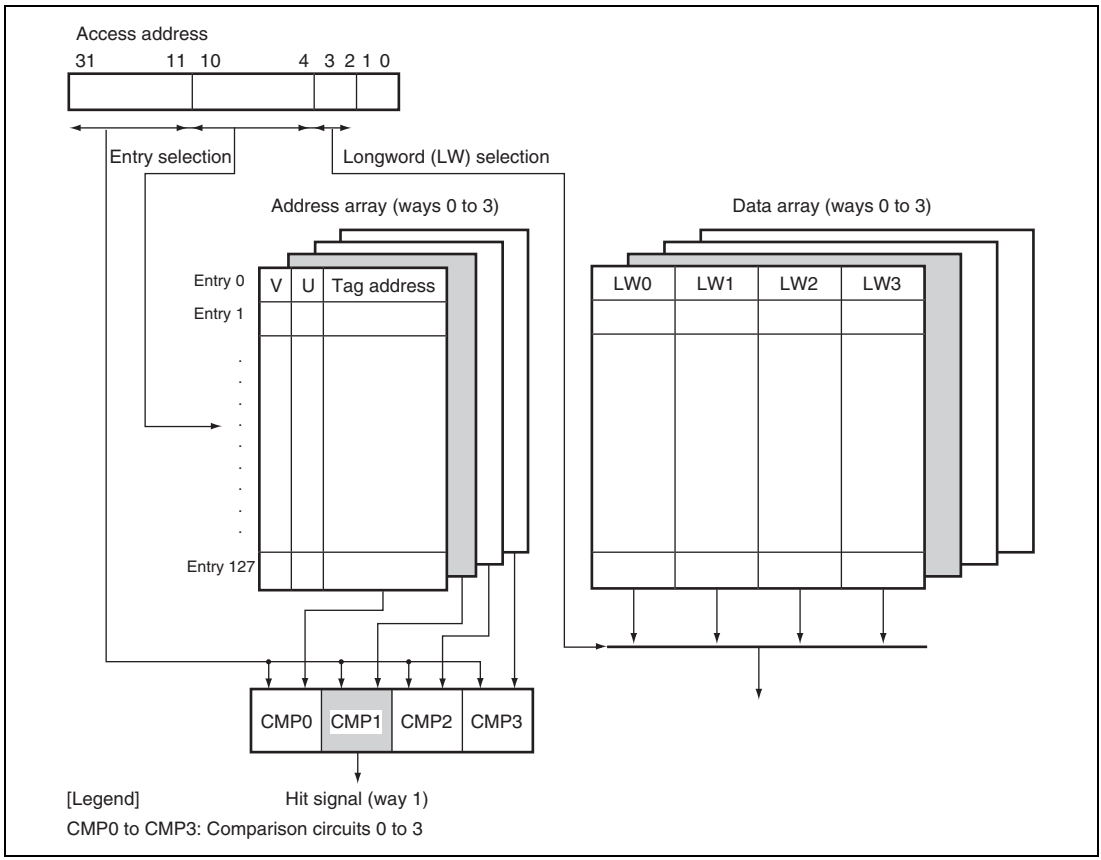


Figure 9.2 Cache Search Scheme

9.3.2 Read Access

(1) Read Hit

In a read access, data is transferred from the cache to the CPU. LRU is updated so that the hit way is the latest.

(2) Read Miss

An external bus cycle starts and the entry is updated. The way replaced follows table 9.4. Entries are updated in 16-byte units. When the desired data that caused the miss is loaded from external memory to the cache, the data is transferred to the CPU in parallel with being loaded to the cache. When it is loaded in the cache, the V bit is set to 1, and LRU is updated so that the replaced way becomes the latest. In operand cache, the U bit is additionally cleared to 0. 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. The update of cache and write-back to memory are performed in wrap around method. For example, the lower four bits of the address at which a read miss occurs indicate H'4, the update of cache and write-back to memory are performed in the order of H'4, H'8, H'C, H'0, which are the lower four bits of the address.

9.3.3 Prefetch Operation (Only for Operand Cache)

(1) Prefetch Hit

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

(2) Prefetch Miss

No data is transferred to the CPU. The way to be replaced follows table 9.3, Other operations are the same in case of read miss.

9.3.4 Write Operation (Only for Operand Cache)

(1) 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 hit way becomes the latest.

(2) 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 9.4. 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. Data is written to the cache, the U bit is set to 1, and the 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. The write-back unit is 16 bytes. The update of cache and write-back to memory are performed in wrap around method. For example, the lower four bits of the address at which a write miss occurs indicate H'4, the update of cache and write-back to memory are performed in the order of H'4, H'8, H'C, H'0, which are the lower four bits of the address.

In write-through mode, no write to cache occurs in a write miss; the write is only to the external memory.

9.3.5 Write-Back Buffer (Only for Operand Cache)

When the U bit of the entry to be replaced in 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 9.3 shows the configuration of the write-back buffer.

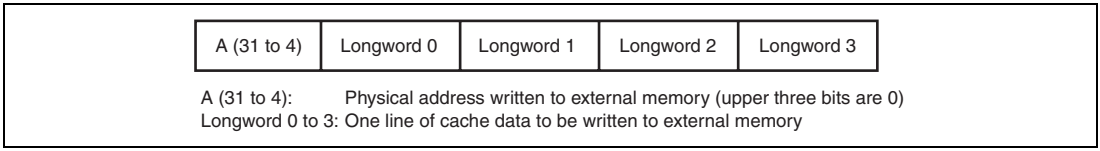


Figure 9.3 Write-Back Buffer Configuration

Table 9.8 summarizes the above operations in sections 9.3.2, Read Access to 9.3.5, Write-Back Buffer (Only for Operand Cache).

Table 9.8 Cache Operations

Cache	CPU Cycle	Hit/ Miss	Write-Back Mode/ Write Through Mode	U Bit	External Memory Accession	Cache Contents
Instruction cache	Instruction fetch	Hit	—	—	Not generated	Not renewed
		Miss	—	—	Cache renewal cycle is generated	Renewed to new values by cache renewal cycle
Operand cache	Prefetch/ read	Hit	Either mode is available	x	Not generated	Not renewed
		Miss	Write-through mode	—	Cache renewal cycle is generated	Renewed to new values by cache renewal cycle
			Write-back mode	0	Cache renewal cycle is generated	Renewed to new values by cache renewal cycle
				1	Cache renewal cycle is generated. Then write-back cycle in write-back buffer is generated.	Renewed to new values by cache renewal cycle
	Write	Hit	Write-through mode	—	Write cycle CPU issues is generated.	Renewed to new values by write cycle the CPU issues
			Write-back mode	x	Not generated	Renewed to new values by write cycle the CPU issues
		Miss	Write-through mode	—	Write cycle CPU issues is generated.	Not renewed*
			Write-back mode	0	Cache renewal cycle is generated	Renewed to new values by cache renewal cycle. Subsequently renewed again to new values in write cycle CPU issues.
				1	Cache renewal cycle is generated. Then write-back cycle in write-back buffer is generated.	Renewed to new values by cache renewal cycle. Subsequently renewed again to new values in write cycle CPU issues.

[Legend]

x: Don't care.

Notes: Cache renewal cycle: 16-byte read access

Write-back cycle in write-back buffer: 16-byte write access

* Neither LRU renewed. LRU is renewed in all other cases.

9.3.6 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 allocated in the cache-enabled space, operate the memory-allocated cache to invalidate and write back as required. Do the same operation for memory shared by the CPU on this LSI and the DMAC.

9.4 Memory-Allocated Cache

To allow software management of the cache, cache contents can be read and written by means of MOV instructions. The instruction cache address array is allocated onto addresses H'F0000000 to H'F07FFFFF, and the data array onto addresses H'F1000000 to H'F17FFFFF. The operand cache address array is allocated onto addresses H'F0800000 to H'F0FFFFFF, and the data array onto addresses H'F1800000 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.

9.4.1 Address Array

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.

In the address field, specify the entry address selecting the entry, The W bit for selecting the way, and the A bit for specifying the existence of associative operation. In the W bit, B'00 is way 0, B'01 is way 1, B'10 is way 2, and B'11 is way 3. Since the access size of the address array is fixed at longword, specify B'00 for bits 1 and 0 of the address.

The tag address, LRU bits, U bit (only for operand cache), and V bit are specified as data. Always specify 0 for the upper three bits (bits 31 to 29) of the tag address.

Refer to figure 9.4 regarding the address and data format.

The following three operations are possible for the address array.

(1) Address Array Read

The tag address, LRU bits, U bit (only for operand cache), and V bit are read from the entry address specified by the address and the entry corresponding to the way. For the read operation, associative operation is not performed regardless of whether the associative bit (A bit) specified by the address is 1 or 0.

(2) Address-Array Write (Non-Associative Operation)

When writing 0 to the associative bit (A bit) in the address field, the tag address, LRU bits, U bit (only for operand cache), and V bit, specified by the data field, are written to the entry address specified by the address and the entry corresponding to the way. When writing to a cache line for which the U bit = 1 and the V bit =1 in the operand cache address array, write the contents of the cache line back to memory, then write the tag address, LRU bits, U bit, and V bit specified by the data field. When 0 is written to the V bit, 0 must also be written to the U bit of that entry. The write-back to memory is performed in the order of H'0, H'4, H'8, H'C, which are the lower four bits of the address.

(3) Address-Array Write (Associative Operation)

When writing with the associative bit (A bit) in the address field set to 1, the addresses in the four ways for the entry specified by the address field are compared with the tag address that is specified by the data field. Write the U bit (only for operand cache) and the V bit specified by the data field to the entry of the way that has a hit. However, the tag address and LRU bits remain unchanged. When there is no way that has a hit, nothing is written and there is no operation. This function is used to invalidate a specific entry in the cache.

When the U bit of the entry that has had a hit is 1 in the operand cache, writing back should be performed. However, when 0 is written to the V bit, 0 must also be written to the U bit of that entry. The write-back to memory is performed in the order of H'0, H'4, H'8, H'C, which are the lower four bits of the address.

9.4.2 Data Array

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, the L bit indicating the longword position within the (16-byte) line, and the W bit for selecting the way. In the L bit, B'00 is longword 0, B'01 is longword 1, B'10 is longword 2, and B'11 is longword 3. In the W bit, B'00 is way 0, B'01 is way 1, B'10 is way 2, and B'11 is way 3. Since the access size of the data array is fixed at longword, specify B'00 for bits 1 and 0 of the address.

Refer to figure 9.4 regarding the address and data format.

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

(1) Data Array Read

The data specified by the L bit in the address is read from the entry address specified by the address and the entry corresponding to the way.

(2) Data Array Write

The longword data specified by the data is written to the position specified by the L bit in the address from the entry address specified by the address and the entry corresponding to the way.

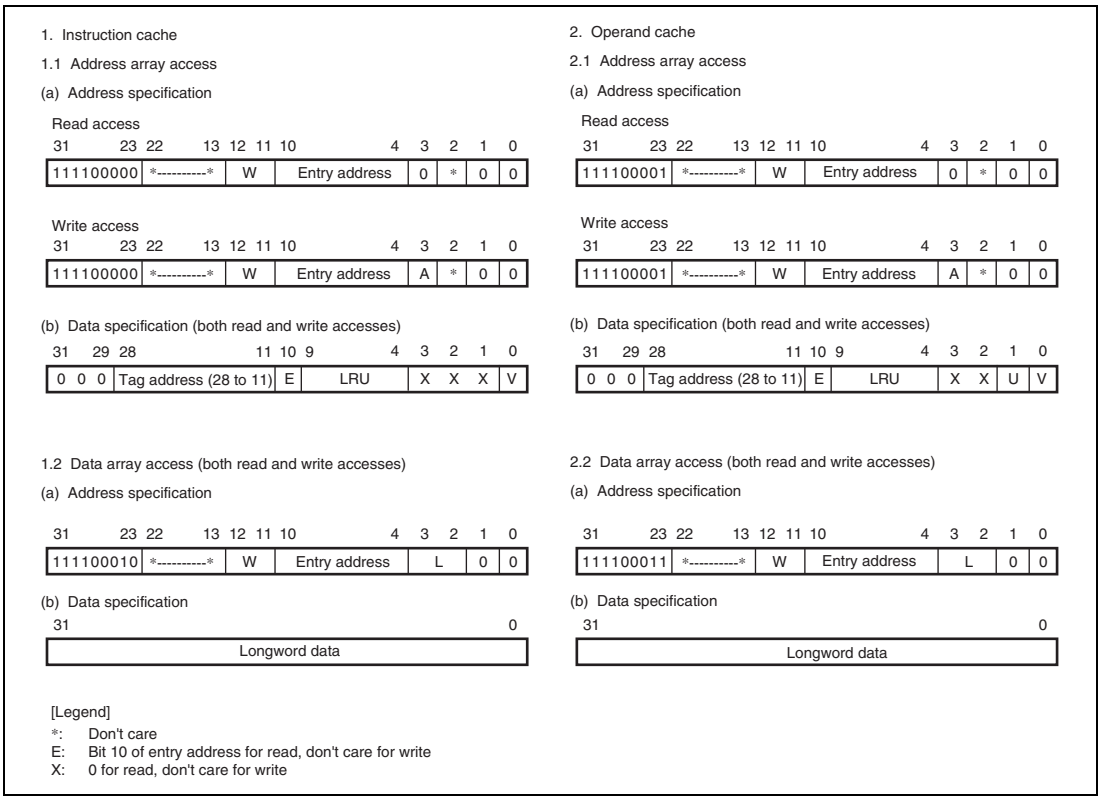


Figure 9.4 Specifying Address and Data for Memory-Allocated Cache Access

9.4.3 Usage Examples

(1) Invalidating Specific Entries

Specific cache entries can be invalidated by writing 0 to the entry's V bit in the memory allocating cache access. When the A bit is 1, the tag address specified by the write data is compared to the tag address 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 the write data is specified in R0 and an address is specified in R1 is shown below.

```
; R0=H'0110 0010; tag address(28 to 11)=B'0 0001 0001 0000 0000 0, U=0, V=0
; R1=H'F080 0088; operand cache address array access, entry=B'000 1000, A=1
;
 MOV.L  R0,@R1
```

(2) Reading the Data of a Specific Entry

The data section of a specific cache entry can be read by the memory allocating cache access. The longword indicated in the data field of the data array in figure 9.4 is read into the register.

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

```
; R0=H'F100 004C; instruction cache data array access, entry=B'000 0100, Way=0,
longword address=3
;
 MOV.L  @R0,R1
```

9.4.4 Notes

1. Programs that access memory-mapped cache of the operand cache should be placed in a cache-disabled space. Programs that access memory-mapped cache of the instruction cache should be placed in a cache-disabled space, and in each of the beginning and the end of that, two or more read accesses to on-chip peripheral modules or external address space (cache-disabled address) should be executed.
2. Rewriting the address array contents so that two or more ways are hit simultaneously is prohibited. Operation is not guaranteed if the address array contents are changed so that two or more ways are hit simultaneously.
3. Only the CPU can access memory-allocated cache; the DMAC cannot access it.

Section 10 Bus State Controller (BSC)

The bus state controller (BSC) outputs control signals for external devices and various types of memory that is connected to the external address space. This enables the LSI to connect directly with SRAM, SDRAM, and other memory storage devices, and external devices.

10.1 Features

1. External address space
   - Maximum of 64 Mbytes for the SDRAM and each for areas CS0 to CS5
   - Ability to select the data bus width (8, 16, or 32 bits) independently for each address space
2. Normal space interface
   - Supports an interface for direct connection to SRAM
   - Cycle wait function: Maximum of 31 wait states (maximum of seven wait states for page access cycles)
   - Wait control

     Ability to select the assert/negate timing for chip select signals

     Ability to select the assert/negate timing for the read strobe and write strobe signals

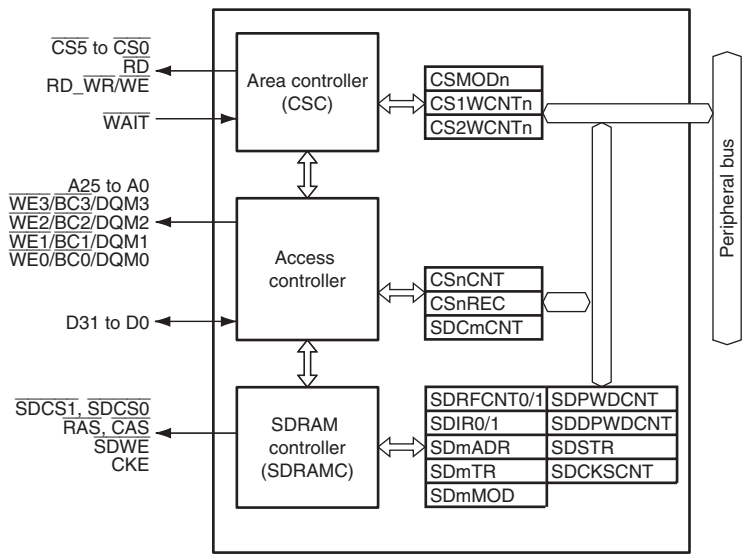
     Ability to select the data output start/end timing

     Ability to select the delay for chip select signals
   - Write access modes: One-write strobe and byte-write strobe modes
   - Page access mode: Support for page read and page write (64-bit, 128-bit, and 256-bit page units)
3. SDRAM interface
   - Ability to set SDRAM in up to two areas
   - Refresh functions

     Auto-refresh (on-chip programmable refresh counter)

     Self-refresh
   - Ability to select the access timing (support for row-column latency, column latency, and row-active interval settings)
   - Initialization sequencer function, power-down function, deep-power-down function, and mode register setting function implemented on-chip

Figure 10.1 shows a block diagram of the BSC. The BSC consists of an area controller (CSC), an access controller, and an SDRAM controller (SDRAMC). The CSC controls accessing normal space in the external address space (see table 10.2). The SDRAMC controls accesses to the SDRAM space. The access controller controls operations common to both the above-mentioned normal space and SDRAM space.



[Legend]

CSMODn:	CSn mode register
CS1WCNTn:	CSn wait control register 1
CS2WCNTn:	CSn wait control register 2
CSnCNT:	CSn control register
CSnREC:	CSn recovery cycle setting register
SDCmCNT:	SDRAMCm control register
SDRFCNT0/1:	SDRAM refresh control register 0/1
SDIR0/1:	SDRAM initialization register 0/1
SDmADR:	SDRAMm address register
SDmTR:	SDRAMm timing register
SDmMOD:	SDRAMm mode register
SDPWDCNT:	SDRAM power-down control register
SDDPWDCNT:	SDRAM deep-power-down control register
SDSTR:	SDRAM status register
SDCKSCNT:	SDRAM clock stop control signal setting register

Note: n = 0 to 5, m = 0 and 1

Figure 10.1 Block Diagram of BSC

10.2 Input/Output Pins

Table 10.1 shows the pin configuration of the BSC.

Table 10.1 Pin Configuration

Pin Name	I/O	Function
A25 to A0	Output	Address bus
D31 to D0	I/O	Data bus
$\overline{CS5}$ to $\overline{CS0}$	Output	Chip select
$\overline{RD}$	Output	Read pulse signal (read data output enable signal)
RD_$\overline{WR}$/$\overline{WE}$	Output	Read or write signal • Indicates either read or write access when a normal space is accessed in byte-write strobe access mode (RD_$\overline{WR}$). • Connects to the WE pin of a byte-select SRAM when a normal space is accessed in one-write strobe mode ($\overline{WE}$).
$\overline{WE3}$/$\overline{BC3}$/DQM3	Output	Controls access via D31 to D24. • Enables writing to the data area corresponding to D31 to D24 when a normal space is accessed in byte-write strobe mode ($\overline{WE3}$). • Connects to the byte select pin of a byte-select SRAM when a normal space is accessed in one-write strobe mode ($\overline{BC3}$). • Controls access to SDRAM if it is connected (DQM3).
$\overline{WE2}$/$\overline{BC2}$/DQM2	Output	Controls access via D23 to D16. • Enables writing to the data area corresponding to D23 to D16 when a normal space is accessed in byte-write strobe mode ($\overline{WE2}$). • Connects to the byte select pin of a byte-select SRAM when a normal space is accessed in one-write strobe mode ($\overline{BC2}$). • Controls access to SDRAM if it is connected (DQM2).

Pin Name	I/O	Function
$\overline{WE1}$/$\overline{BC1}$/DQM1	Output	Controls access via D15 to D8. • Enables writing to the data area corresponding to D15 to D8 when a normal space is accessed in byte-write strobe mode ($\overline{WE1}$). • Connects to the byte select pin of a byte-select SRAM when a normal space is accessed in one-write strobe mode ($\overline{BC1}$). • Controls access to SDRAM if it is connected (DQM1).
$\overline{WE0}$/$\overline{BC0}$/DQM0	Output	Controls access via D7 to D0. • Enables writing to the data area corresponding to D7 to D0 when a normal space is accessed in byte-write strobe mode ($\overline{WE0}$). • Connects to the byte select pin of a byte-select SRAM when a normal space is accessed in one-write strobe mode ($\overline{BC0}$). • Controls access to SDRAM if it is connected (DQM0).
$\overline{SDCS1}$, $\overline{SDCS0}$	Output	Connects to the $\overline{CS}$ pin if SDRAM is connected.
$\overline{RAS}$	Output	Connects to the $\overline{RAS}$ pin if SDRAM is connected.
$\overline{CAS}$	Output	Connects to the $\overline{CAS}$ pin if SDRAM is connected.
CKE	Output	Connects to the CKE pin if SDRAM is connected.
$\overline{SDWE}$	Output	Connects to the WE pin of SDRAM if SDRAM is connected ($\overline{SDWE}$).
$\overline{WAIT}$	Input	External wait input

10.3 Area Overview

10.3.1 Address Map

In the architecture, this LSI has a 32-bit address space, which is divided into cache-enabled, cache-disabled, and on-chip spaces (on-chip high-speed RAM, on-chip RAM for data retention, on-chip peripheral modules, and reserved areas) according to the upper bits of the address.

External address spaces CS5 to CS0, SDRAM0, and SDRAM1 are cache-enabled when internal address A29 = 0 and cache-disabled when A29 = 1.

The kind of memory to be connected and the data bus width are specified independently for each partial space. The address map for the external address space is shown below.

Table 10.2 Address Map

Internal Address	Space	Memory to be Connected	BIU*1	Size	Cache
H'00000000 to H'03FFFFFF	CS0	Normal space	BIU_E	64MB	Cache-enabled
H'04000000 to H'07FFFFFF	CS1	Normal space		64MB	
H'08000000 to H'0BFFFFFF	CS2	Normal space		64MB	
H'0C000000 to H'0FFFFFFF	CS3	Normal space		64MB	
H'10000000 to H'13FFFFFF	CS4	Normal space		64MB	
H'14000000 to H'17FFFFFF	CS5	Normal space		64MB	
H'18000000 to H'1BFFFFFF	SDRAM0	SDRAM space		64MB	
H'1C000000 to H'1FFFFFFF	SDRAM1	SDRAM space		64MB	
H'20000000 to H'23FFFFFF	CS0	Normal space (shadow)		64MB	Cache-disabled
H'24000000 to H'27FFFFFF	CS1	Normal space (shadow)		64MB	
H'28000000 to H'2BFFFFFF	CS2	Normal space (shadow)		64MB	
H'2C000000 to H'2FFFFFFF	CS3	Normal space (shadow)		64MB	
H'30000000 to H'33FFFFFF	CS4	Normal space (shadow)		64MB	
H'34000000 to H'37FFFFFF	CS5	Normal space (shadow)		64MB	
H'38000000 to H'3BFFFFFF	SDRAM0	SDRAM space (shadow)		64MB	
H'3C000000 to H'3FFFFFFF	SDRAM1	SDRAM space (shadow)		64MB	
H'40000000 to H'7FFFFFFF	Reserved	Reserved area	—	—	
H'80000000 to H'E7FFFFFF	Reserved	Reserved area	—	—	
H'E8000000 to H'EBFFFFFF	Other	On-chip peripheral modules	BIU_PB3	64MB	
H'EC000000 to H'EFFFFFFF	Reserved	Reserved area	—	—	
H'F0000000 to H'F1FFFFFF	Others	Cache address array space or other	*2	—	

Internal Address	Space	Memory to be Connected	BIU*1	Size	Cache
H'F2000000 to H'FEFFFFFF	Reserved	Reserved area	—	—	Cache-disabled
H'FF400000 to H'FF7FFFFF	Others	On-chip peripheral modules, reserved area	BIU_PB2	4MB	
H'FF800000 to H'FF9FFFFF	Others	On-chip peripheral modules, on-chip RAM for data retention, reserved area	BIU_PB0	2MB	
H'FFA00000 to H'FFBFFFFF	Others	On-chip peripheral modules, reserved area	BIU_PB1	2MB	
H'FFC00000 to H'FFD7FFFF	Reserved	Reserved area	—	—	
H'FFD80000 to H'FFD8FFFF	On-chip RAM0	On-chip high-speed RAM0 space (shadow)	BIU_PB0	64KB	
H'FFD90000 to H'FFD9FFFF	Reserved	Reserved area	—	—	
H'FFDA0000 to H'FFDA7FFF	On-chip RAM1	On-chip high-speed RAM1 space (shadow)	BIU_PB1	32KB	
H'FFDA8000 to H'FFF7FFFF	Reserved	Reserved area	—	—	
H'FFF80000 to H'FFF8FFFF	On-chip RAM0	On-chip high-speed RAM0 space	BIU_PB0	64KB	
H'FFF90000 to H'FFF9FFFF	Reserved	Reserved area	—	—	
H'FFFA0000 to H'FFFA7FFF	On-chip RAM1	On-chip high-speed RAM1 space	BIU_PB1	32KB	
H'FFFA8000 to H'FFFBFFFF	Reserved	Reserved area	—	—	
H'FFFC0000 to H'FFFCFFFF	Others	On-chip peripheral modules, reserved area	*2	64KB	
H'FFFD0000 to H'FFFEFFFF	Others	On-chip peripheral modules, reserved area	BIU_PB0	128KB	
H'FFFF0000 to H'FFFFFFFF	Others	On-chip peripheral modules, reserved area	BIU_PB1	64KB	

Notes: 1. The term BIU stands for Bus Interface Unit. BIUs are internal modules through which the CPU and DMAC accesses each address space. Described below are on-chip BIUs, and address spaces and internal buses connected to these BIUs.

BIU_E: External address spaces (normal and SDRAM spaces)

BIU_PB0: peripheral bus 0 (internal to the LSI)

BIU_PB1: peripheral bus 1 (internal to the LSI)

BIU_PB2: peripheral bus 2 (internal to the LSI)

BIU_PB3: peripheral bus 3 (internal to the LSI)

Pipelined DMA transfer is not available for transfer from a BIU to the same BIU. For details, see section 11, Direct Memory Access Controller (DMAC).

2. Cache address array space and some on-chip peripheral modules are not allocated to any BIU. These devices are accessed directly from individual CPUs without using the internal bus. The DMAC cannot access any of these devices.

10.3.2 Data Bus Width and Related Pin Settings for Each Area

The data bus width of area 0 can be set to either 16 or 32 bits by means of an external pin, but it cannot be changed by a program after startup. The initial data bus width of SDRAM0, SDRAM1, and areas 1 to 5 can be changed by a program by means of register settings.

Immediately after a power-on reset, certain address and data bus settings, and functions of $\overline{CS0}$ and $\overline{RD}$, are automatically selected as initial functions required to read the area 0 ROM. The other pins are set as general ports as the initial function setting, and they cannot be used until their functions are set by a program. Do not perform other than read access to area 0 until the pin function settings have been completed.

Table10.3 lists the external pin settings and the initial state of each area.

Note that sample access waveforms are shown in this section for pins such as RD_$\overline{WR}$ and $\overline{WEn}$, and these functions are available after they are selected through the pin function controller. For example, after startup with the 32-bit bus width, changing the bus width of the areas other than area 0 to 16 bits requires setting the function of the A1 pin, and changing the bus width to 8 bits requires setting the functions of the A1 and A0 pins.

For details on pin function settings, see section 27, Pin Function Controller (PFC).

Table 10.3 External Pin Setting (MD) and Initial State Of Each Area

MD0	Item	Area 0	SDRAM0, SDRAM1, Areas 1 to 5
1	Data bus width	Bus width fixed at 32 bits. Cannot be changed.	Initial bus width 32 bits. Can be changed by a program.
	BSC-related pin settings	Only A20 to A2, D31 to D0, $\overline{CS0}$, and $\overline{RD}$ pin functions are set automatically. The other pin functions must be set by a program.	
0	Data bus width	Bus width fixed at 16 bits. Cannot be changed.	Initial bus width 16 bits. Can be changed by a program.
	BSC-related pin settings	Only A20 to A1, D15 to D0, $\overline{CS0}$, and $\overline{RD}$ pin functions are set automatically. The other pin functions must be set by a program.	

Notes: 1. In area 0, when connecting ROM that uses address lines A21 and above, pull-down processing of address lines A21 and above must be performed on the board.

2. It is not possible to use a device with an 8-bit bus width, the RD_$\overline{WR}$/$\overline{WE}$ function, and area 3 all at the same time because the address line A0 function and RD_$\overline{WR}$/$\overline{WE}$ function are both assigned to a single pin (PB0), and the $\overline{CS3}$ function and RD_$\overline{WR}$/$\overline{WE}$ function are also both assigned to a single pin (PB6).

10.4 Register Descriptions

The BSC has the following registers. All registers are initialized by a power-on reset or in deep standby mode.

Do not access spaces other than area 0 until settings are completed for the connected memory interface.

Table 10.4 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
CS0 control register	CS0CNT	R/W	H'00010000/ H'00110000*	H'FF420000	8, 16, 32
CS0 recovery cycle setting register	CS0REC	R/W	H'00000000	H'FF420008	8, 16, 32
CS1 control register	CS1CNT	R/W	H'00000000	H'FF420010	8, 16, 32
CS1 recovery cycle setting register	CS1REC	R/W	H'00000000	H'FF420018	8, 16, 32
CS2 control register	CS2CNT	R/W	H'00000000	H'FF420020	8, 16, 32
CS2 recovery cycle setting register	CS2REC	R/W	H'00000000	H'FF420028	8, 16, 32
CS3 control register	CS3CNT	R/W	H'00000000	H'FF420030	8, 16, 32
CS3 recovery cycle setting register	CS3REC	R/W	H'00000000	H'FF420038	8, 16, 32
CS4 control register	CS4CNT	R/W	H'00000000	H'FF420040	8, 16, 32
CS4 recovery cycle setting register	CS4REC	R/W	H'00000000	H'FF420048	8, 16, 32
CS5 control register	CS5CNT	R/W	H'00000000	H'FF420050	8, 16, 32
CS5 recovery cycle setting register	CS5REC	R/W	H'00000000	H'FF420058	8, 16, 32
SDRAMC0 control register	SDC0CNT	R/W	H'00000000	H'FF420100	8, 16, 32
SDRAMC1 control register	SDC1CNT	R/W	H'00000000	H'FF420110	8, 16, 32
CS0 mode register	CSMOD0	R/W	H'00000000	H'FF421000	8, 16, 32
CS0 wait control register 1	CS1WCNT0	R/W	H'1F1F0707	H'FF421004	8, 16, 32
CS0 wait control register 2	CS2WCNT0	R/W	H'00000007	H'FF421008	8, 16, 32
CS1 mode register	CSMOD1	R/W	H'00000000	H'FF421010	8, 16, 32
CS1 wait control register 1	CS1WCNT1	R/W	H'1F1F0707	H'FF421014	8, 16, 32
CS1 wait control register 2	CS2WCNT1	R/W	H'00000007	H'FF421018	8, 16, 32
CS2 mode register	CSMOD2	R/W	H'00000000	H'FF421020	8, 16, 32
CS2 wait control register 1	CS1WCNT2	R/W	H'1F1F0707	H'FF421024	8, 16, 32
CS2 wait control register 2	CS2WCNT2	R/W	H'00000007	H'FF421028	8, 16, 32

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
CS3 mode register	CSMOD3	R/W	H'00000000	H'FF421030	8, 16, 32
CS3 wait control register 1	CS1WCNT3	R/W	H'1F1F0707	H'FF421034	8, 16, 32
CS3 wait control register 2	CS2WCNT3	R/W	H'00000007	H'FF421038	8, 16, 32
CS4 mode register	CSMOD4	R/W	H'00000000	H'FF421040	8, 16, 32
CS4 wait control register 1	CS1WCNT4	R/W	H'1F1F0707	H'FF421044	8, 16, 32
CS4 wait control register 2	CS2WCNT4	R/W	H'00000007	H'FF421048	8, 16, 32
CS5 mode register	CSMOD5	R/W	H'00000000	H'FF421050	8, 16, 32
CS5 wait control register 1	CS1WCNT5	R/W	H'1F1F0707	H'FF421054	8, 16, 32
CS5 wait control register 2	CS2WCNT5	R/W	H'00000007	H'FF421058	8, 16, 32
SDRAM refresh control register 0	SDRFCNT0	R/W	H'00000000	H'FF422000	8, 16, 32
SDRAM refresh control register 1	SDRFCNT1	R/W	H'0000xxxx	H'FF422004	16, 32
SDRAM initialization register 0	SDIR0	R/W	H'00000xxx	H'FF422008	8, 16, 32
SDRAM initialization register 1	SDIR1	R/W	H'00000000	H'FF42200C	8, 16, 32
SDRAM power-down control register	SDPWDCNT	R/W	H'00000000	H'FF422010	8, 16, 32
SDRAM deep-power-down control register	SDDPWDCNT	R/W	H'00000000	H'FF422014	8, 16, 32
SDRAM0 address register	SD0ADR	R/W	H'00000x0x	H'FF422020	8, 16, 32
SDRAM0 timing register	SD0TR	R/W	H'000xxx0x	H'FF422024	8, 16, 32
SDRAM0 mode register	SD0MOD	R/W	H'0000xxxx	H'FF422028	16, 32
SDRAM1 address register	SD1ADR	R/W	H'00000x0x	H'FF422040	8, 16, 32
SDRAM1 timing register	SD1TR	R/W	H'000xxx0x	H'FF422044	8, 16, 32
SDRAM1 mode register	SD1MOD	R/W	H'0000xxxx	H'FF422048	16, 32
SDRAM status register	SDSTR	R	H'00000000	H'FF4220E4	8, 16, 32
SDRAM clock stop control signal setting register	SDCKSCNT	R/W	H'0000000F	H'FF4220E8	8, 16, 32

Note: * Depends on the MD0 pin setting made at start-up.

10.4.1 CSn Control Register (CSnCNT) (n = 0 to 5)

CSnCNT selects the width of the external bus and controls the operation of the CSC interface.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	BSIZE[1:0]		-	-	-	EXENB
Initial value:	0	0	0	0	0	0	0	0	0	0	0*1	0*1	0	0	0	0*2
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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, 20	BSIZE[1:0]	00*1	R/W	External Bus Width Select These bits specify the width of the data bus for the external device corresponding to a CSC channel. The initial value for the data bus width for CSC channel 0 (CS0) differs depending on the MD0 pin setting. 10: 8-bit bus 00: 16-bit bus 01: 32-bit bus
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	EXENB	0*2	R/W	Operation Enable This bit enables or disables the operation for each corresponding CSC channel. The initial value corresponding to CS0 only is operation enabled (EXENB = 1). 0: Operation disabled 1: Operation enabled
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. The initial value of the BSIZE bits in CS0 differs depending on the MD0 pin setting.
2. The initial value of the EXENB bit in CS0 is 1.

To disable (EXENB = 0) the operation for each channel, forcibly write out data tentatively stored in internal write buffer. The procedure is as follows:

1. Execute read access to the channel whose operation is to be disabled.
2. Then, write 0 to the EXENB bit (operation disabled).

10.4.2 CSn Recovery Cycle Setting Register (CSnREC) (n = 0 to 5)

CSnREC specifies the number of data recovery cycles to be inserted after read or write accesses.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	WRCV[3:0]				-	-	-	-	RRCV[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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 to 24	WRCV[3:0]	0000	R/W	Post-Write Data Recovery Cycle Setting These bits specify the number of data recovery cycles to be inserted after write accesses to the external bus. If a value other than 0 is selected, 1 to 15 data recovery cycles are inserted when a write access to the external bus is followed by a read access to the external bus. (Data recovery cycles are inserted even when access is performed sequentially to the same CSC channel.) Note that if idle cycles occur between accesses to the external bus, the number of data recovery cycles inserted is reduced by the number of idle cycles. 0000: 0 cycles 0001: 1 cycle : 1111: 15 cycles

Bit	Bit Name	Initial Value	R/W	Description
23 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 16	RRCV[3:0]	0000	R/W	Post-Read Data Recovery Cycle Setting These bits specify the number of data recovery cycles to be inserted after read accesses to the external bus. If a value other than 0 is selected, data recovery cycles are inserted in the following cases: If a read access to the external bus is followed by a write access to the external bus. (Data recovery cycles are inserted even when access is performed sequentially to the same CSC channel.) If a read access to the external bus is followed by a read access to a different CSC channel. (No data recovery cycles are inserted in cases of sequential read accesses to the same CSC channel.) Note that if idle cycles occur between accesses to the external bus, the number of data recovery cycles inserted is reduced by the number of idle cycles. 0000: 0 cycles 0001: 1 cycle : 1111: 15 cycles
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. When accessing SDRAM, there is no danger of data collision on the bus due to timing. Consequently, there is no data recovery cycle setting for SDRAM. (The value is fixed at 0 cycles.)

2. Writing to the CSn recovery cycle setting register (CSnREC) must be done while the CSC for the corresponding channel is disabled (EXENB = 0). Only channel 0 (CS0) is allowed for writing to the register without disabling the CSC (EXENB = 1). To write to CS0REC with CSC enabled, satisfy all of the following conditions:
   - Stop the DMAC.
   - Keep the CPU other than the one that is going to rewrite the register from accessing CS0 (including access for instruction fetch). For example, if CPU0 is going to rewrite the register, make CPU1 stay looping by a program copied to on-chip memory, or put CPU1 in a sleep state.
   - Do not perform data write access to CS0 after a reset is released but before the register is rewritten.

10.4.3 SDRAMCm Control Register (SDCmCNT) (m = 0, 1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	BSIZE[1:0]		-	-	-	EXENB
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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, 20	BSIZE[1:0]	00	R/W	External Bus Width Select These bits specify the width of the data bus for the external device of the corresponding channel of SDRAMC. 10: 8-bit bus 00: 16-bit bus 01: 32-bit bus
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	EXENB	0	R/W	Operation Enable This bit enables or disables the operation for the corresponding channel of SDRAMC. 0: Operation disabled 1: Operation enabled
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

To disable the operation (EXENB = 0) for each channel, forcibly write out data tentatively stored in internal write buffer. The procedure is as follows:

1. Execute read access to the channel whose operation is to be disabled.
2. Then, write 0 to the EXENB bit (operation disabled).

10.4.4 CSn Mode Register (CSMODn) (n = 0 to 5)

CSMODn selects the mode for page read access and the bit boundary for page access, enables page read/write access and external wait, and selects the mode for write access.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PRMOD	-	PBCNT[1:0]		-	-	PWENB	PRENB	-	-	-	-	EWENB	-	-	WRMOD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R/W	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	PRMOD	0	R/W	Page Read Access Mode Select This bit selects operating mode for page read access. Clearing PRMOD to 0 selects normal access compatible mode. In this mode, the $\overline{RD}$ signal is negated and an $\overline{RD}$ assert wait is inserted each time a unit of data is read. Setting PRMOD to 1 selects external data read sequential assert mode. In this mode, $\overline{RD}$ is asserted continuously between page accesses. 0: Normal access compatible mode 1: External data read sequential assert mode
30	—	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
29, 28	PBCNT[1:0]	00	R/W	Page Access Bit Boundary Select These bits select the bit boundary for page access operation. When the bit boundary specified by PBCNT is exceeded during page access, page access operation is halted temporarily (the $\overline{CSn}$ signal is negated), and then page access operation begins again. The value written to these bits is valid only when either of the PWENB bit or the PRENB bit is set to 1. 00: 64-bit boundary 01: 128-bit boundary 10: 256-bit boundary 11: Setting prohibited
27, 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	PWENB	0	R/W	Page Write Access Enable This bit is used to enable page write access. 0: Page write access disabled 1: Page write access enabled
24	PRENB	0	R/W	Page Read Access Enable This bit is used to enable page read access. 0: Page read access disabled 1: Page read access enabled
23 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19	EWENB	0	R/W	External Wait Enable This bit is used to enable or disable external wait input. When EWENB is set to 1, external wait input is enabled and the number of wait states per cycle can be controlled using the external wait signal ($\overline{WAIT}$). In this case wait cycles are inserted while the $\overline{WAIT}$ signal is on a low level. When EWENB is cleared to 0, the $\overline{WAIT}$ signal is invalid. 0: External wait disabled 1: External wait enabled

Bit	Bit Name	Initial Value	R/W	Description
18, 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	WRMOD	0	R/W	Write Access Mode Select This bit selects operating mode for write access. Clearing WRMOD to 0 selects byte-write strobe mode. In this mode, data writes are controlled by multiple write signals ($\overline{WE3}$ to $\overline{WE0}$) that correspond to the individual byte positions. Setting WRMOD to 1 selects one-write strobe mode. In this mode, data writes are controlled by multiple byte control signals ($\overline{BC3}$ to $\overline{BC0}$) that correspond to the individual byte positions and a single write signal ($\overline{WE}$). 0: Byte-write strobe mode 1: One-write strobe mode
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Writing to the CSn mode register (CSMODn) must be done while the CSC for the corresponding channel is disabled (EXENB = 0). Only channel 0 (CS0) is allowed for writing to the register without disabling the CSC (EXENB = 1). To write to CSMOD0 with CSC enabled, satisfy all of the following conditions:

1. Stop the DMAC.
2. Keep the CPU other than the one that is going to rewrite the register from accessing CS0 (including access for instruction fetch). For example, if CPU0 is going to rewrite the register, make CPU1 stay looping by a program copied to on-chip memory, or put CPU1 in a sleep state.
3. Do not perform data write access to CS0 after a reset is released but before the register is updated.

10.4.5 CSn Wait Control Register 1 (CS1WCNTn) (n = 0 to 5)

CS1WCNTn specifies the number of wait states to be inserted into the read/write cycle or page read/page write cycle.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	CSRWAIT[4:0]					-	-	-	CSWWAIT[4:0]				
Initial value:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	CSPRWAIT[2:0]			-	-	-	-	-	CSPWWAIT[2:0]		
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 24	CSRWAIT [4:0]	11111	R/W	Read Cycle Wait Select These bits specify the number of wait states to be inserted into the initial normal read cycle and page read cycle. 00000: 0 wait states : 11111: 31 wait states
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20 to 16	CSWWAIT [4:0]	11111	R/W	Write Cycle Wait Select These bits specify the number of wait states to be inserted into the initial normal write cycle and page write cycle. 00000: 0 wait states : 11111: 31 wait states
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 to 8	CSPRWAIT [2:0]	111	R/W	Page Read Cycle Wait Select These bits specify the number of wait states to be inserted into the second and subsequent page read cycles. This setting is valid when the page read access enable bit (PRENB) is set to 1. 000: 0 wait states : 111: 7 wait states
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	CSPWWAIT [2:0]	111	R/W	Page Write Cycle Wait Select These bits specify the number of wait states to be inserted into the second and subsequent page write cycles. This setting is valid when the page write access enable bit (PWENB) is set to 1. 000: 0 wait states : 111: 7 wait states

Notes: 1. Make sure the page read and page write cycle wait select (CSPRWAIT and CSPWWAIT) settings are within the range defined by the read and write cycle wait select (CSRWAIT and CSWWAIT) settings. Select each number of wait states according to the configuration of your system.

2. Writing to the CSn wait control register 1 (CS1WCNTn) must be done while the CSC for the corresponding channel is disabled (EXENB = 0). Only channel 0 (CS0) is allowed for writing to the register without disabling the CSC (EXENB = 1). To write to CS1WCNT0 with CSC enabled, satisfy all of the following conditions:
   - Stop the DMAC.
   - Keep the CPU other than the one that is going to rewrite the register from accessing CS0 (including access for instruction fetch). For example, if CPU0 is going to rewrite the register, make CPU1 stay looping by a program copied to on-chip memory, or put CPU1 in a sleep state.
   - Do not perform data write access to CS0 after a reset is released but before the register is rewritten.

10.4.6 CSn Wait Control Register 2 (CS2WCNTn) (n = 0 to 5)

CS2WCNTn specifies the number of wait states and the number of delay cycles.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	CSON[2:0]			-	WDON[2:0]			-	WRON[2:0]			-	RDON[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	WDOFF[2:0]			-	CSWOFF[2:0]			-	CSROFF[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

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 to 28	CSON[2:0]	000	R/W	CS Assert Wait Select These bits specify the number of wait states to be inserted before the external chip select signal ($\overline{CSn}$) is asserted. 000: 0 wait states : 111: 7 wait states
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 24	WDON[2:0]	000	R/W	Write Data Output Wait Select These bits specify the number of wait states to be inserted before data is output to the external data bus. 000: 0 wait states : 111: 7 wait states
23	—	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
22 to 20	WRON[2:0]	000	R/W	WR Assert Wait Select These bits specify the number of wait states to be inserted before the external data write signal ($\overline{WE3}$ to $\overline{WE0}$) is asserted. 000: 0 wait states : 111: 7 wait states
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	RDON[2:0]	000	R/W	RD Assert Wait Select These bits specify the number of wait states inserted before the external data read signal ($\overline{RD}$) is asserted. 000: 0 wait states : 111: 7 wait states
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	WDOFF[2:0]	000	R/W	Write Data Output Delay Cycle Select These bits specify the number of cycles from the end of the wait cycle (negation of the $\overline{WE3}$ to $\overline{WE0}$ signals) to the negation of the external data bus during write operation. 000: 0 wait states : 111: 7 wait states
7	—	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
6 to 4	CSWOFF [2:0]	000	R/W	Write Operation CS Delay Cycle Select These bits specify the number of cycles from the end of the wait cycle (negation of the $\overline{WE3}$ to $\overline{WE0}$ signals) to the negation of the $\overline{CS5}$ to $\overline{CS0}$ signals during write access operation. 000: 0 wait states : 111: 7 wait states
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	CSROFF [2:0]	111	R/W	Read Operation CS Delay Cycle Select These bits specify the number of cycles from the end of the wait cycle (negation of the $\overline{RD}$ signal) to the negation of the $\overline{CS5}$ to $\overline{CS0}$ signals during read access operation. 000: 0 wait states : 111: 7 wait states

Notes: 1. Select each number of wait states or delay cycles according to the configuration of your system.

2. Writing to the CSn wait control register 2 (CS2WCNTn) must be done while the CSC for the corresponding channel is disabled (EXENB = 0). Only channel 0 (CS0) is allowed for writing to the register without disabling the CSC (EXENB = 1). To write to CS2WCNT0 with CSC enabled, satisfy all of the following conditions:
   - Stop the DMAC.
   - Keep the CPU other than the one that is going to rewrite the register from accessing CS0 (including access for instruction fetch). For example, if CPU0 is going to rewrite the register, make CPU1 stay looping by a program copied to on-chip memory, or put CPU1 in a sleep state.
   - Do not perform data write access to CS0 after a reset is released but before the register is rewritten.

3. Each bit must be set under the following restrictions.
   When page access is disabled (PRENB, PWENB = 0)
   CSON ≤ min (CSRWAIT, CSWWAIT), WDON ≤ CSWWAIT,
   WRON ≤ CSWWAIT, RDON ≤ CSRWAIT
   WDOFF ≤ CSWOFF
   When page access is enabled (PRENB = 1 or PWENB = 1)
   In addition to the restrictions for disabled page access, the following restrictions are required.

CSON ≤ min. (CSPRWAIT, CSPWWAIT)
WRON ≤ CSPWWAIT, RDON ≤ CSPRWAIT
WDON ≤ CSPWWAIT

10.4.7 SDRAM Refresh Control Register 0 (SDRFCNT0)

SDRFCNT0 controls self-refresh operation.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DSFEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DSFEN	0	R/W	SDRAM Common Self-Refresh Operation Enable This bit controls self-refresh operation for all channels simultaneously. Setting DSFEN to 1 performs auto-refresh cycle operation, immediately after which self-refresh operation begins. Clearing DSFEN to 0 ends self-refresh operation, and auto-refresh operation resumes immediately afterward. The value written to this bit is reflected when self-refresh operation starts, if DSFEN was set to 1, or when auto-refresh operation starts following the end of self-refresh operation, if DSFEN was cleared to 0. 0: Self-refresh disabled 1: Self-refresh enabled

10.4.8 SDRAM Refresh Control Register 1 (SDRFCNT1)

SDRFCNT1 controls auto-refresh operation.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DRFEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DREFW[3:0]				DRFC[11:0]											
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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	DRFEN	0	R/W	Auto-Refresh Operation Enable This bit controls auto-refresh operation for all channels simultaneously. When DRFEN is cleared to 0, auto-refresh operation does not take place. Auto-refresh operates when DRFEN is set to 1. Clearing this bit to 0 while auto-refresh is enabled causes the DRFEN bit to be cleared to 0, and auto-refresh operation to halt, after the end of the next auto-refresh cycle. Setting this bit to 1 while auto-refresh is disabled causes auto-refresh operation to commence as soon as the DRFEN bit is set to 1, and refresh requests are then generated at fixed intervals determined by a counter. The interval at which refresh requests are generated is determined by the set value of the auto-refresh request interval setting (DRFC) bits. Refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes. If an SDRAM access and refresh request are generated at the same time, the refresh request takes precedence. 0: Auto-refresh disabled 1: Auto-refresh enabled

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	DREFW [3:0]	Undefined	R/W	Auto-Refresh Cycle/Self-Refresh Clearing Cycle Count Setting These bits specify the number of auto-refresh cycles and the number of self-refresh clearing cycles. The DREFW bits can be written to at any time, regardless of the state of the auto-refresh operation enable (DRFEN) bit. If auto-refresh is disabled, the value written to these bits takes effect immediately. If auto-refresh is enabled, the value written to these bits takes effect immediately if an auto-refresh cycle is not in progress. If an auto-refresh cycle is in progress, the new value takes effect after the cycle completes. 0000: 1 cycle 0001: 2 cycles 0010: 3 cycles : 1111: 16 cycles
11 to 0	DRFC [11:0]	Undefined	R/W	Auto-Refresh Request Interval Setting These bits specify the auto-refresh interval. The DRFC bits can be written to at any time, regardless of the state of the auto-refresh operation enable (DRFEN) bit. If auto-refresh is disabled, the value written to these bits takes effect immediately. If auto-refresh is enabled, the value written to these bits is reflected in the operation of the refresh counter next time an auto-refresh request is generated. 000000000000: Setting prohibited 000000000001: 2 cycles 000000000010: 3 cycles : 111111111111: 4096 cycles

Note: Auto-refresh requests are not accepted while multiple read or write accesses are in progress, or during a transfer using DMAC, so the auto-refresh interval may become extended in some cases. Set the DRFC bits to an auto-refresh request interval value that satisfies the auto-refresh interval specification of the SDRAM being used. Furthermore, make sure to set the auto-refresh request interval to a duration longer than the auto-refresh cycle.

(a) Auto-Refresh Request Interval and DRFC Set Value

SDRAMC includes a 12-bit refresh counter that generates auto-refresh requests at fixed intervals. The following equation is used to calculate the set value for the DRFC bits from the auto-refresh request interval.

DRFC = (Auto-refresh request interval / Bus clock cycle) – 1

Auto-refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes. However, the counter value is updated regardless of whether or not the request was accepted. Note that if two or more auto-refresh requests are generated while SDRAM is being accessed, the second and subsequent requests are ignored.

10.4.9 SDRAM Initialization Register 0 (SDIR0)

SDIR0 specifies the SDRAM initialization sequence timing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	DPC[2:0]			DARFC[3:0]				DARFI[3:0]			
Initial value:	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	DPC[2:0]	Undefined	R/W	Initialization Precharge Cycle Count Setting These bits specify the number of precharge cycles in the SDRAM initialization sequence. 000: 3 cycles 001: 4 cycles : 111: 10 cycles

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	DARFC [3:0]	Undefined	R/W	Initialization Auto-Refresh Count These bits specify the number of times auto-refresh is to be performed in the SDRAM initialization sequence. 0000: Setting prohibited 0001: 1 time : 1111: 15 times
3 to 0	DARFI [3:0]	Undefined	R/W	Initialization Auto-Refresh Interval These bits specify the interval at which auto-refresh commands are issued in the SDRAM initialization sequence. 0000: 3 cycles 0001: 4 cycles 0010: 5 cycles : 1111: 18 cycles

Note: Make settings that satisfy the specifications of the connected SDRAM before starting the initialization sequence.

10.4.10 SDRAM Initialization Register 1 (SDIR1)

SDIR1 controls activation of the SDRAM initialization sequence.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DINIST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DINIRQ
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

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	DINIST	0	R/W	Initialization Status When set to 1, this bit indicates that an SDRAM initialization sequence is in progress for the channel for SDRAM0 or SDRAM1. 0: Initialization sequence not in progress 1: Initialization sequence in progress
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DINIRQ	0	R/W	Common Initialization Sequence Start Setting this bit to 1 causes the SDRAM initialization sequence to start and automatically sets the initialization status bit (DINIST) to 1. The initialization status bit (DINIST) is cleared automatically after the initialization sequence ends. The value written to the DINIRQ bit is not retained. 0: Invalid 1: Initialization sequence start

10.4.11 SDRAM Power-Down Control Register (SDPWDCNT)

SDPWDCNT controls transition to and recovery from power-down mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DPWD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DPWD	0	R/W	SDRAM Common Power-Down Enable This bit controls transition to and recovery from power-down mode for all channels simultaneously. Setting DPWD to 1 causes all channels to transfer to power-down mode. Clearing DPWD to 0 causes all channels to recover from power-down mode. If an auto-refresh is in progress, the transition to power-down mode is delayed until the auto-refresh completes. 0: Power-down disabled 1: Power-down enabled

10.4.12 SDRAM Deep-Power-Down Control Register (SDDPWDCNT)

SDDPWDCNT controls transition to and recovery from deep-power-down mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DDPD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DDPD	0	R/W	SDRAM Common Deep-Power-Down Enable This bit controls transition to and recovery from deep-power-down mode for all channels simultaneously. Setting DDPD to 1 causes all SDRAM channels to transfer to deep-power-down mode. Clearing DDPD to 0 causes all SDRAM channels to recover from deep-power-down mode. If an auto-refresh is in progress, the transition to deep-power-down mode is delayed until the auto-refresh completes. 0: Deep-power-down disabled 1: Deep-power-down enabled

10.4.13 SDRAMm Address Register (SDmADR) (m = 0, 1)

SDmADR specifies the data bus width and the channel size for SDRAM.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	DDBW[1:0]		-	-	-	-	-	DSZ[2:0]		
Initial value:	0	0	0	0	0	0	-	-	0	0	0	0	0	-	-	-
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	DDBW [1:0]	Undefined	R/W	SDRAM Data Bus Width Setting These bits specify the width of the SDRAM data bus. When accessing 32-bit data in SDRAM with a 16-bit bus width, the 16 bits at the first half of the address (A1 = 0) are accessed first, and then the 16 bits at the second half of the address (A1 = 1) are accessed. 00: 8 bits 01: 16 bits 10: 32 bits 11: Setting prohibited
7 to 3	—	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
2 to 0	DSZ[2:0]	Undefined	R/W	Channel Size Setting These bits specify the size for channels 0 and 1. If a size smaller than SDRAM area 0 or 1 is selected, the remaining portion is regarded as a shadow area. 000: Setting prohibited 001: 8MB 010: 16MB 011: 32MB 100: 64MB 101 to 111: Setting prohibited

10.4.14 SDRAMm Timing Register (SDmTR) (m = 0, 1)

SDmTR specifies the timing for read and write accesses to SDRAM.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	DRAS[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	DRCD[1:0]		DPCG[2:0]			DWR	-	-	-	-	-	DCL[2:0]		
Initial value:	0	0	-	-	-	-	-	-	0	0	0	0	0	-	-	-
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

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 to 16	DRAS[2:0]	Undefined	R/W	Row Active Interval Setting These bits specify the minimum interval that must elapse between the SDRAM row activation command (ACT) and deactivation (PRA). 000: 1 cycle : 111: 8 cycles
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	DRCD [1:0]	Undefined	R/W	Row Column Latency Setting These bits specify the SDRAM row column latency. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles

Bit	Bit Name	Initial Value	R/W	Description
11 to 9	DPCG [2:0]	Undefined	R/W	Row Precharge Interval Setting These bits specify the minimum interval that must elapse between the SDRAM deactivation (PRA) command and the next valid command. 000: 1 cycle : 111: 8 cycles
8	DWR	Undefined	R/W	Write Recovery Interval Setting This bit specifies the interval that must elapse between the SDRAM write command (WRITE) and deactivation (PRA). 0: 1 cycle 1: 2 cycles
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	DCL[2:0]	Undefined	R/W	SDRAM Controller Column Latency Setting These bits specify the column latency of the SDRAM controller. This setting only affects the latency setting on the SDRAM controller side. To specify the column latency for externally connected SDRAM, it is necessary to use the SDRAMm mode register (SDmMOD), which is described later. 000: Setting prohibited 001: 1 cycle 010: 2 cycles 011: 3 cycles 1xx: Setting prohibited

[Legend]

x: Don't care

10.4.15 SDRAMm Mode Register (SDmMOD) (m = 0, 1)

SDmMOD specifies the values to be written to the SDRAM mode register or extended mode register. Writing to this register causes a mode register set command or extended mode register set command to be issued automatically to SDRAM.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	DMR[14:0]														
Initial value:	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	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 to 0	DMR [14:0]	Undefined	R/W	Mode Register Setting Writing to these bits causes a mode register set command or extended mode register set command to be issued to SDRAM. The setting of the DMR bits is output as the A16 to A2 signal. SDRAM distinguishes between the mode register set command and the extended mode register set command based on their bank address. Write operation: A mode register set command is issued.

DMR bit	b14	b13	…	b0
	↓	↓		↓
A16 to A2 signal	A16	A15	…	A2

Notes: The following points should be kept in mind regarding SDRAMm mode register settings.

1. Make sure to set a burst length of 1 for SDRAM. Operation cannot be guaranteed with settings other than a burst length of 1.
2. The SDRAM column latency must match the setting of the SDRAM controller column latency setting bits (DCL) in SDRAMC. Operation cannot be guaranteed if the latency settings do not agree.
3. Make sure the status bits (DSRFST, DPWDST, DDPDST, and DMRSST) in the SDRAM status register (SDSTR) are all cleared to 0.

10.4.16 SDRAM Status Register (SDSTR)

SDSTR consists of the status flags that indicate the status of operation during self-refresh, initialization sequences, power-down mode, deep-power-down mode, and mode register setting.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	DSRFST	DINIST	DPWDST	DDPDST	DMRSST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	DSRFST	0	R	Self-Refresh Transition/Recovery Status When set to 1, this bit indicates that a transition to or recovery from self-refresh operation is in progress for the channel for SDRAM0 or SDRAM1. 0: Transition/recovery not in progress 1: Transition/recovery in progress
3	DINIST	0	R	Initialization Status When set to 1, this bit indicates that an SDRAM initialization sequence is in progress for the channel for SDRAM0 or SDRAM1. This bit has the same function as the DINIST bit in the SDIR1 register. 0: Initialization sequence not in progress 1: Initialization sequence in progress
2	DPWDST	0	R	Power-Down Transition/Recovery Status When set to 1, this bit indicates that a transition to or recovery from power-down mode is in progress on the channel for SDRAM0 or SDRAM1. 0: Transition/recovery not in progress 1: Transition/recovery in progress

Bit	Bit Name	Initial Value	R/W	Description
1	DDPDST	0	R	Deep-Power-Down Transition/Recovery Status When set to 1, this bit indicates that a transition to or recovery from deep-power-down mode is in progress on the channel for SDRAM0 or SDRAM1. 0: Transition/recovery not in progress 1: Transition/recovery in progress
0	DMRSST	0	R	Mode Register Setting Status When set to 1, this bit indicates that mode register setting is in progress for the channel for SDRAM0 or SDRAM1. 0: Mode register setting not in progress 1: Mode register setting in progress

"Transition or recovery in progress" refers to the interval from the point at which the bits listed in table 10.5 are written to until the corresponding commands are issued.

Table 10.5 List of Status Registers and Bits Requiring Checking

Function	Register Name	Bit Name
Self-refresh	SDRFCNT0	DSFENCm, DSFEN
Initialization sequence	SDIR1	DINIRQCm, DINIRQ
Power-down	SDPWDCNT	DPWDCm, DPWD
Deep-Power-down	SDDPDCNT	DDPDCm, DDPD
Mode register setting	SDmMOD	DMR

Note: Execution of a self-refresh, a transition to or recovery from power-down or deep-power-down mode, an initialization sequence, or mode register setting may only be performed when all status bits are cleared to 0.
Do not rewrite the registers (bits) listed in table 10.5 when any of the status bits (DSRFST, DINIST, DPWDST, DDPDST, DMRSST) is set to 1.

10.4.17 SDRAM Clock Stop Control Signal Setting Register (SDCKSCNT)

SDCKSCNT enables or disables the clock stop control signal (signal in the chip) and specifies the number of assert cycles.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DCKSEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	DCKSC[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	DCKSEN	0	R/W	Clock Stop Control Signal Enable This bit is used to enable or disable the clock stop control function. If enabled, the clock stop control function stops CKIO (low level) at transition to or recovery from deep-power-down mode. If disabled, the function does not stop CKIO. 0: Clock stop control function disabled 1: Clock stop control function enabled
15 to 8	—	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
7 to 0	DCKSC [7:0]	H'0F	R/W	Clock Stop Cycle Count Setting These bits specify the interval from the point at which the deep-power-down transition command is issued until the clock stop control function stops CKIO (low level), and the interval from the point at which CKIO starts operating until the recover command is issued. 00000000: 0 cycles : 00001111: 15 cycles : 11111111: 255 cycles

10.5 Operation

10.5.1 Accessing CS Space

(1) Normal Access

Normal read/write operation is used for all bus accesses when page read/write access is disabled (PRENB = 0, PWENB = 0). Even when page read/write access is enabled (PRENB = 1, PWENB = 1), normal read/write operation is employed in cases where page access cannot be used. Figure 10.2 shows the basic operation of the external bus control signals for read and write operations in byte-write strobe mode. Figure 10.3 shows the basic operation of these signals for read and write operations in one-write strobe mode. In these figures, $\overline{\text{DACTn}}$ is a DMA active output signal. For details, see section 11, Direct Memory Access Controller (DMAC).

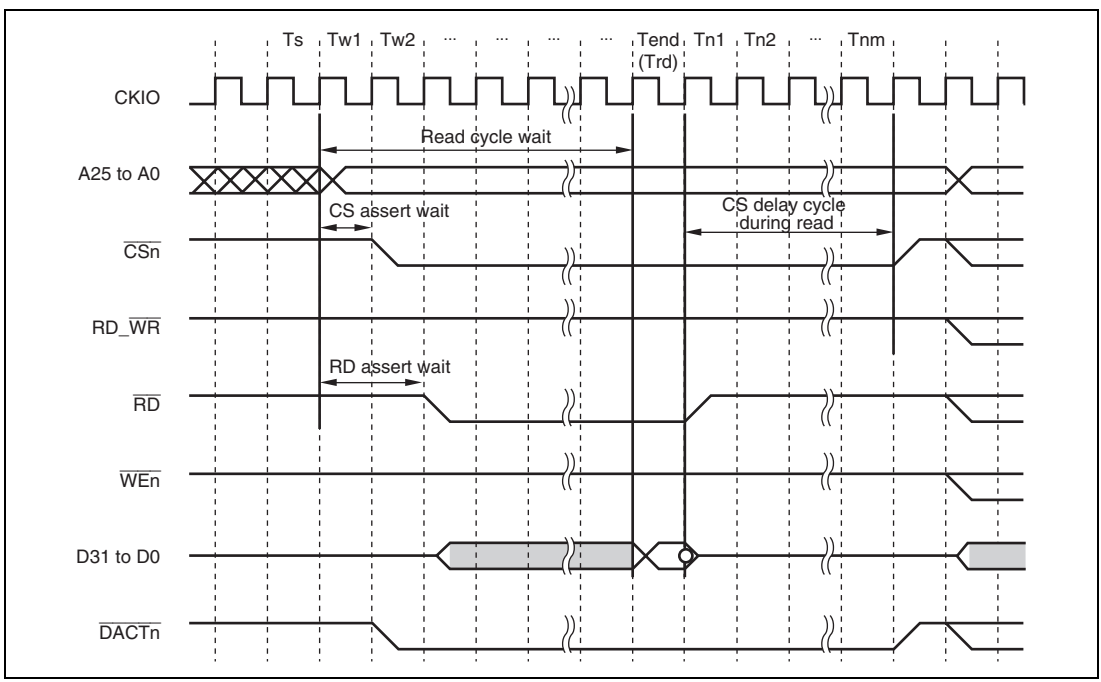


Figure 10.2 Basic Bus Timing (Read Operation in Byte-Write Strobe Mode) (a)

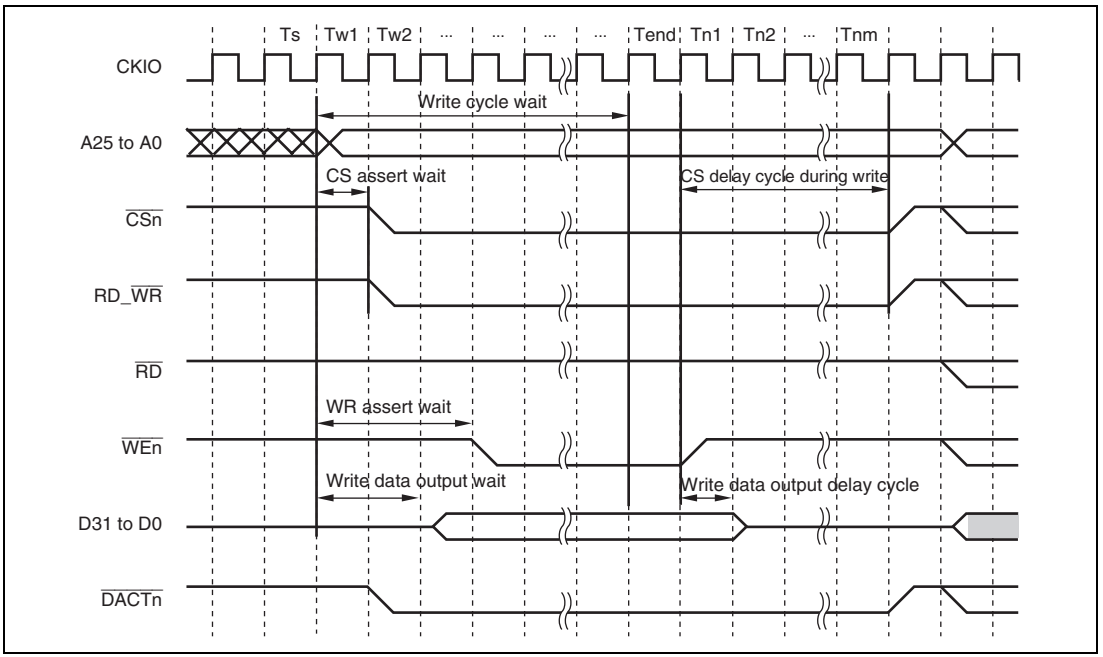


Figure 10.2 Basic Bus Timing (Write Operation in Byte-Write Strobe Mode) (b)

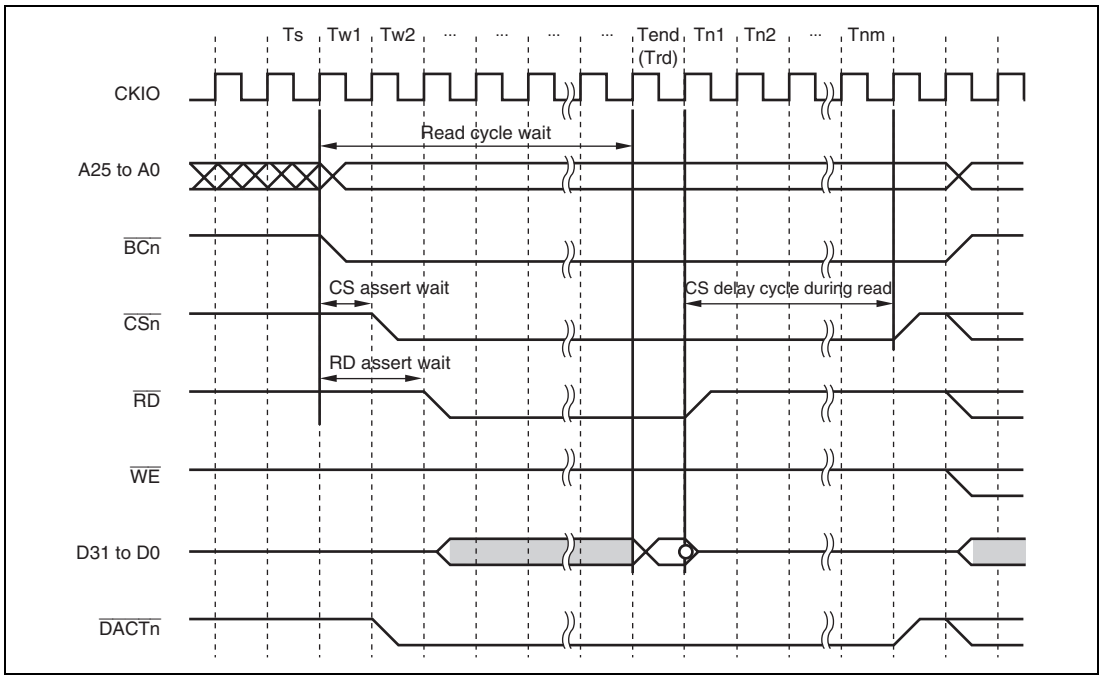


Figure 10.3 Basic Bus Timing (Read Operation in One-Write Strobe Mode) (a)

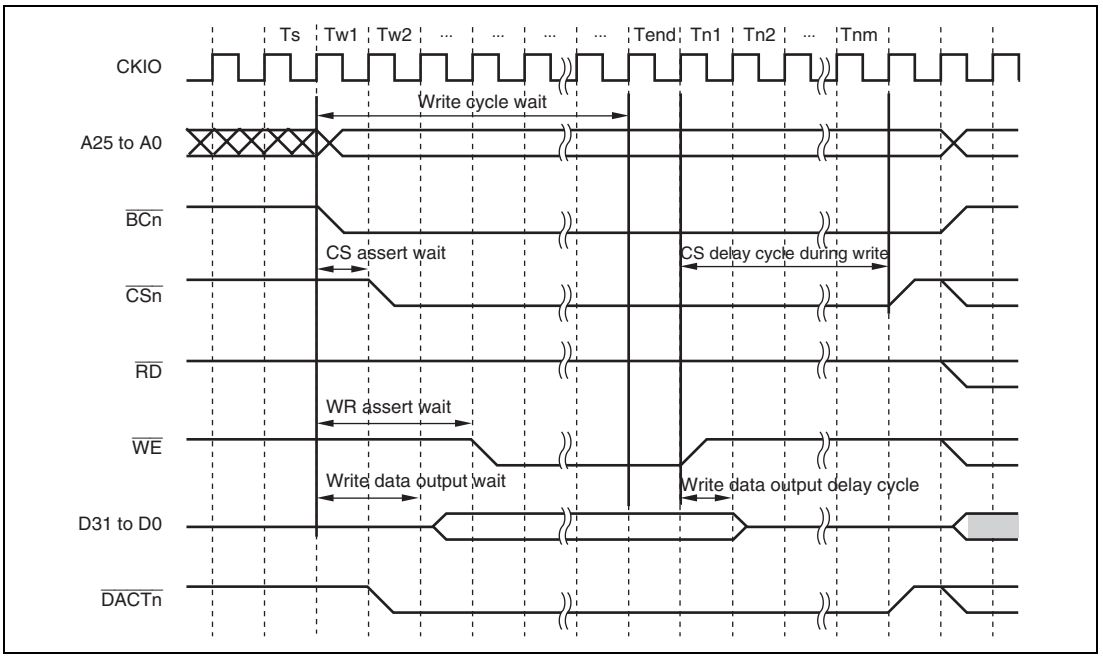


Figure 10.3 Basic Bus Timing (Write Operation in One-Write Strobe Mode) (b)

1. Ts (Internal Bus Access Start)

   This is a bus access request cycle initiated by the internal bus master to the external bus as the target. $\overline{\text{CSn}}$ is always high during this cycle. In the next cycle, A25 to A0, $\overline{\text{BCn}}$, and the write data change.

2. Tw1 to Twn (Read Cycle Wait, Write Cycle Wait)

   These are the cycles between internal bus access start and the wait end cycle. A duration of from 0 to 31 clocks may be selected. During this interval the $\overline{\text{CSn}}$, $\overline{\text{RD}}$, $\overline{\text{WEn}}$, and $\overline{\text{WE}}$ signals are asserted (low level) in accordance with the wait settings. The assert timing can be controlled using the CS assert wait, RD assert wait, WR assert wait, and write data output wait bits in CSn control registers 1 and 2. The number of wait cycles can be set to from 0 to 7 clocks, as counted from the cycle following internal bus access start (Ts). The number of clocks selected must be no greater than the number of read/write cycle wait cycles. The RD_$\overline{\text{WR}}$ signal operates with the same timing as for the $\overline{\text{CSn}}$ signal.

3. Tend (Wait End Cycle)

   This is the final cycle in a series of read cycle wait or write cycle wait cycles. The $\overline{\text{RD}}$, $\overline{\text{WEn}}$, or $\overline{\text{WE}}$ signal is negated (high level) in the next cycle.

4. Tn1 to Tnm (CS Delay Cycle)

   These are the cycles between the wait end cycle and when $\overline{\text{CSn}}$ is negated (high level). The negation timing can be controlled using write data output delay cycles. The number of cycles is counted beginning from the wait end cycle. In write access or if the number of CS delay cycles during a read is other than 0 or 1, the succeeding bus access can start from the cycle following the CS delay cycle end. If the number of CS delay cycles is 0 or 1 in read access, the succeeding bus access can start after the end of the read data sample cycle (see below).
5. Trd (Read Data Sample Cycle)

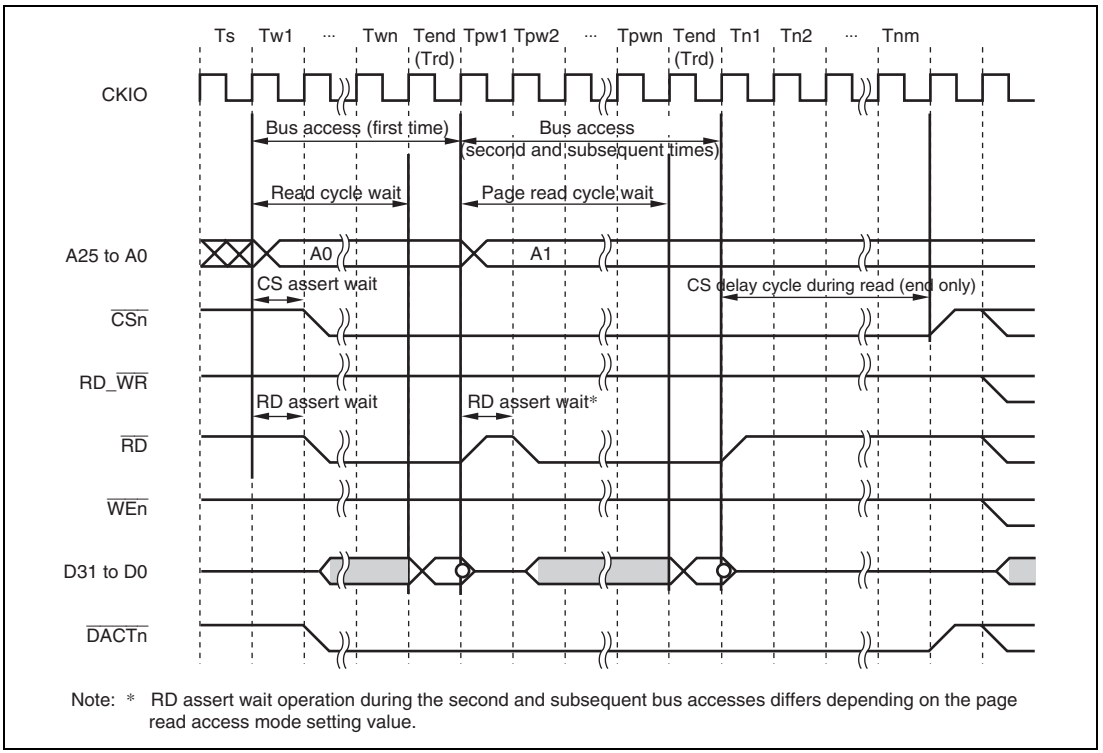
   This is the sample cycle for read data.

(2) Page Access

Page read and write operation is employed for bus accesses for which page access can be used if page write access enable (PWENB = 1) and page read access enable (PRENB = 1) have been selected. Page access is used in the following cases.

1. CPU cache replace (cache filling and write-back). Each time a transfer occurs, the address is incremented by the number of transferred bytes.
2. When longword (32-bit) access to an 8-bit or 16-bit external data bus has been performed.
3. When word (16-bit) access to an 8-bit external data bus has been performed.

Figure 10.4 shows the basic operation of the external bus control signals for page read operation, and figure 10.5 shows the basic operation of these signals for page write operation. If the single page access bit boundary setting made by the PBCNT bits in the mode register is smaller than the cache line size, a single cache replacement will trigger multiple page accesses. When the address exceeds the page boundary, page access stops once (the $\overline{\text{CSn}}$ signal is negated) and starts again. If the PBCNT bit setting is not smaller than the cache line size, a cache replacement is processed in a single page access.



Note: * RD assert wait operation during the second and subsequent bus accesses differs depending on the page read access mode setting value.

Figure 10.4 Basic Bus Timing (Page Read Operation in Byte-Write Strobe Mode)

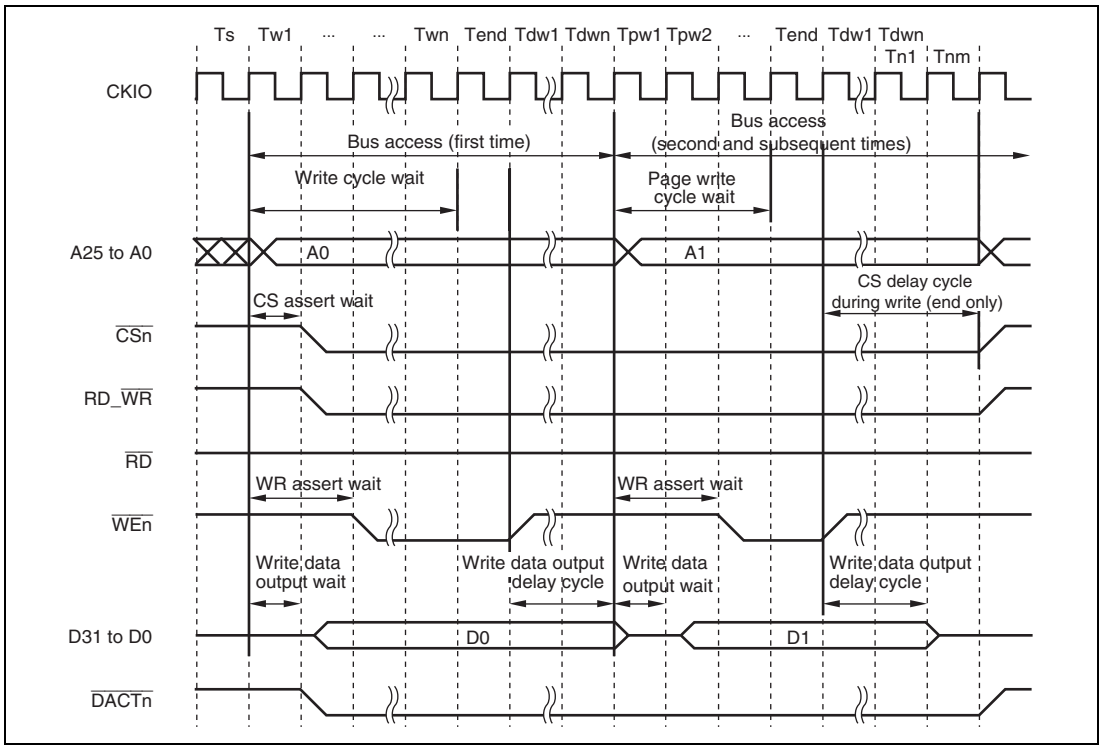


Figure 10.5 Basic Bus Timing (Page Write Operation in Byte-Write Strobe Mode)

1. Ts (Internal Bus Access Start)

   This is a bus access request cycle initiated by the internal bus master to the external bus as the target. $\overline{\text{CSn}}$ is always high during this cycle. In the next cycle, A25 to A0, $\overline{\text{BCn}}$, and the write data change.

2. Tw1 to Twn (Read Cycle Wait, Write Cycle Wait)

   For the first bus access in a page access, the control of the wait operation from internal bus access start to the wait end cycle is the same as in normal access.

3. Tend (First Wait End Cycle)

   This is the final cycle in the first series of read cycle wait or write cycle wait cycles. In write access, the second and subsequent page accesses start from the next cycle, unless a write data output delay cycle has been specified (with a value other than 0). The $\overline{RD}$, $\overline{WEn}$, or $\overline{WE}$ signal is negated (high level) in the next cycle if the RD assert wait or WR assert wait setting is other than 0. If the RD assert wait or WR assert wait setting is 0, the $\overline{RD}$, $\overline{WEn}$, or $\overline{WE}$ signal continues to be asserted (low level). The $\overline{CSn}$ signal is not negated and continues to be asserted (low level). The RD_$\overline{WR}$ signal operates with the same timing as for the $\overline{CSn}$ signal. In page read access, the succeeding bus access starts without waiting for the read data sample cycle (Trd).

4. Tdw1 to Tdwn (Write Data Output Delay Cycle)

   In write access, write data output delay cycles are inserted between the wait end cycle and the following page access if the write data output delay wait setting is other than 0. Assertion of the address and output data is extended for the duration of this interval. Also, the $\overline{WEn}$ and $\overline{WE}$ signals are negated (high level).

5. Tpw1 to Tpwn (Page Read Cycle Wait, Page Write Cycle Wait)

   For the second and subsequent bus cycles in a page access, the page read cycle wait and page write cycle wait settings are used in place of the read cycle wait and write cycle wait settings. The WR assert wait setting works in the same way as for the first bus cycle. The RD assert wait setting operates differently depending on the page read access mode (PRMOD) setting value.

   PRMOD = 0: RD assert wait setting operates in the same way as for the first bus cycle.

   PRMOD = 1: RD assert wait setting is invalid.

   Operation is the same as for RD assert wait setting of 0.

6. Tend/Tdw1 to Tdwn (Wait End Cycle/Write Data Output Delay Cycle)

   The operation is the same as for the first access (3 and 4 above).

7. Tn1 to Tnm (CS Delay Cycle)

   These are the cycles between the final wait end cycle and when $\overline{CSn}$ is negated (high level). The number of CS delay cycles is counted beginning from the wait end cycle.

8. Trd (Final Read Data Sample Cycle)

   This is the final sample cycle for read data.

(3) External Wait Function

The external wait signal ($\overline{\text{WAIT}}$) can be used to extend the wait cycle duration beyond the value specified by the cycle wait (CSRWAIT, CSWWAIT) or page access cycle wait (CSPRWAIT, CSPWWAIT) settings in the CSn wait control register (CSWCNTn). If external wait enable (EWENB = 1) has been selected, wait cycles are inserted for as long as the $\overline{\text{WAIT}}$ signal remains low level. The $\overline{\text{WAIT}}$ signal is disabled if external wait disable (EWENB = 0) has been selected.

Note that the wait cycles specified by the settings of the CSn wait control register (CSWCNTn) are inserted regardless of the state of the $\overline{\text{WAIT}}$ signal.

(a) Normal Read/Write Operation

The $\overline{\text{WAIT}}$ signal is sampled all the time and its result is reflected two cycles later. Thus, when the $\overline{\text{WAIT}}$ signal is low two cycles before the end of the wait cycles, external cycles are inserted. After the $\overline{\text{WAIT}}$ signal has gone high, the wait cycles end two cycles later.

(b) Page Access Operation

The initial data read/write operation is the same as a normal read/write operation. That is, when the $\overline{\text{WAIT}}$ signal is low two cycles before the end of the wait cycles (Tend), external wait cycles are inserted. After the $\overline{\text{WAIT}}$ signal has gone high, the wait cycles end (Tend) two cycles later.

In the second and subsequent read accesses, the page wait cycle is extended if the $\overline{\text{WAIT}}$ signal is low two cycles before the end of the page access wait cycle (Tend), and the page wait cycles end (Tend) two cycles after the $\overline{\text{WAIT}}$ signal has gone high.

Figures 10.6 and 10.7 show examples of external wait timing for page read access by longword (32-bit) access to a 16-bit channel. Figure 10.6 is an example in which one or more cycles of cycle wait state or page cycle wait state has been set. Figure 10.7 is an example in which no cycle wait or page cycle wait state has been set. Note that the value of the $\overline{\text{WAIT}}$ signal before the beginning of the bus cycle is reflected if there are only a few cycles of cycle wait states.

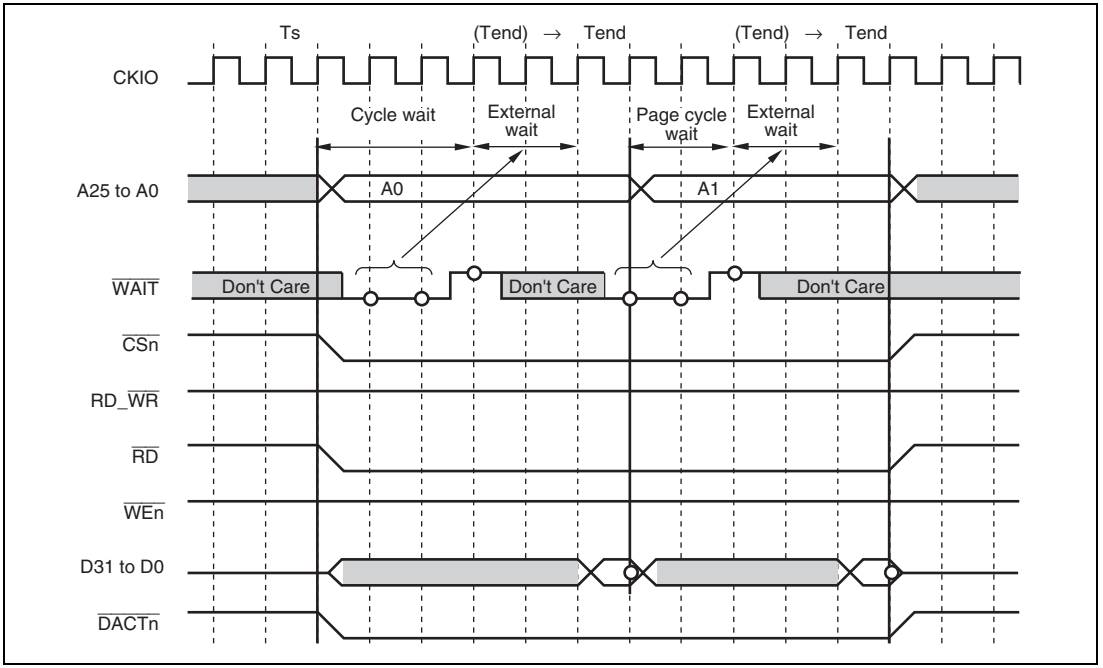


Figure 10.6 External Wait Timing Example (Page Read Access to 16-Bit Channel)

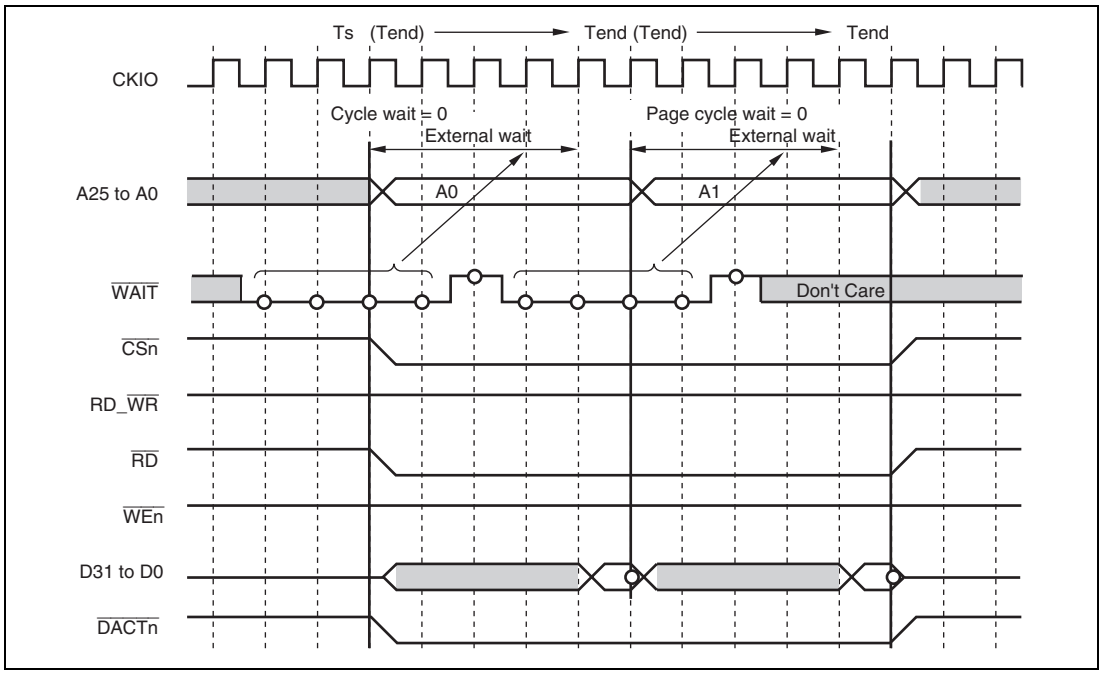


Figure 10.7 External Wait Timing Example (Page Read Access to 16-Bit Channel)

(4) Access Type and Data Alignment

(a) 32-Bit Bus Channel

If a 32-bit bus is selected by the external bus width select bits in the CSn control register, A25 to A2 are enabled as address signals for longword units and A1 and A0 are disabled (fixed at low level). Table 10.6 lists the data alignment corresponding to byte addresses for different data sizes.

If byte strobe mode (WRMOD = 0) is selected, the $\overline{WE3}$ to $\overline{WE0}$ signals indicate the bits to be accessed. For read access, however, all bits are access targets regardless of the state of the $\overline{WE3}$ to $\overline{WE0}$ signals.

If one-write strobe mode (WRMOD = 1) is selected, the $\overline{BC3}$ to $\overline{BC0}$ signals indicate access targets for both read and write accesses. For write access, the write strobe signal $\overline{WE}$ is also asserted.

Table 10.6 Data Alignment (32-Bit Bus Channel)

Data Size	Byte Address (Lower 2 Bits)	DATA [31:24]	DATA [23:16]	DATA [15:8]	DATA [7:0]	$\overline{WE}/\overline{BC}$ [3]	$\overline{WE}/\overline{BC}$ [2]	$\overline{WE}/\overline{BC}$ [1]	$\overline{WE}/\overline{BC}$ [0]
Byte	0	O	×	×	×	L	H	H	H
	1	×	O	×	×	H	L	H	H
	2	×	×	O	×	H	H	L	H
	3	×	×	×	O	H	H	H	L
Word	0	O	O	×	×	L	L	H	H
	2	×	×	O	O	H	H	L	L
Longword	0	O	O	O	O	L	L	L	L

Note: The valid bits on the data bus for each data size are indicated by circles (O).
Crosses (×) indicate bus data bits that are undefined.

(b) 16-Bit Bus Channel

If a 16-bit bus is selected by the external bus width select bits in the CSn control register, A25 to A1 are enabled as address signals for word units, and A0 is disabled (fixed at low level). Table 10.7 lists the data alignment corresponding to byte addresses for different data sizes.

If byte strobe mode (WRMOD = 0) is selected, the $\overline{WE1}$ and $\overline{WE0}$ signals indicate the bits to be accessed. For read access, however, all bits are access targets regardless of the state of the $\overline{WE1}$ and $\overline{WE0}$ signals.

If one-write strobe mode (WRMOD = 1) is selected, the $\overline{BC1}$ to $\overline{BC0}$ signals indicate access targets for both read and write. For write access, the write strobe signal $\overline{WE}$ is also asserted.

Table 10.7 Data Alignment (16-Bit Bus Channel)

		DATA				$\overline{WE}/\overline{BC}$			
Data Size	**Byte Address (Lower 2 Bits)**	**[31:24]**	**[23:16]**	**[15:8]**	**[7:0]**	**[3]**	**[2]**	**[1]**	**[0]**
Byte	0	×	×	O	×	*	*	L	H
	1	×	×	×	O	*	*	H	L
	2	×	×	O	×	*	*	L	H
	3	×	×	×	O	*	*	H	L
Word	0	×	×	O	O	*	*	L	L
	2	×	×	O	O	*	*	L	L
Longword	0 (1st)	×	×	O	O	*	*	L	L
	2 (2nd)	×	×	O	O	*	*	L	L

Note: The valid bits on the data bus for each data size are indicated by circles (O).
Crosses (×) indicate bus data bits that are undefined.
Asterisks (*) indicate write/byte control bits that are disabled (fixed at high level).

(c) 8-Bit Bus Channel

If an 8-bit bus is selected by the external bus width select bits in the CSn control register, A25 to A0 are enabled as address signals for byte units. Table 10.8 lists the data alignment corresponding to byte addresses for different data sizes.

If byte strobe mode (WRMOD = 0) is selected, the $\overline{WE0}$ signal is asserted only for write access; it is not asserted for read access.

If one-write strobe mode (WRMOD = 1) is selected, the $\overline{BC0}$ signal is asserted for both read and write accesses. For write access, the write strobe signal $\overline{WE}$ is also asserted.

Table 10.8 Data Alignment (8-Bit Bus Channel)

Data Size	Byte Address (Lower 2 Bits)	DATA [31:24]	DATA [23:16]	DATA [15:8]	DATA [7:0]	$\overline{WE}/\overline{BC}$ [3]	$\overline{WE}/\overline{BC}$ [2]	$\overline{WE}/\overline{BC}$ [1]	$\overline{WE}/\overline{BC}$ [0]
Byte	0	×	×	×	O	*	*	*	L
	1	×	×	×	O	*	*	*	L
	2	×	×	×	O	*	*	*	L
	3	×	×	×	O	*	*	*	L
Word	0 (1st)	×	×	×	O	*	*	*	L
	1 (2nd)	×	×	×	O	*	*	*	L
	2 (1st)	×	×	×	O	*	*	*	L
	3 (2nd)	×	×	×	O	*	*	*	L
Longword	0 (1st)	×	×	×	O	*	*	*	L
	1 (2nd)	×	×	×	O	*	*	*	L
	2 (3rd)	×	×	×	O	*	*	*	L
	3 (4th)	×	×	×	O	*	*	*	L

Note: The valid bits on the data bus for each data size are indicated by circles (O).
Crosses (×) indicate bus data bits that are undefined.
Asterisks (*) indicate write/byte control bits that are disabled (fixed at high level).

10.5.2 Accessing SDRAM

A description is provided here of the SDRAM controller (SDRAMC) operation enable and SDRAM bus width settings as well as operations involving SDRAM (read, write, auto-refresh, self-refresh, initialization sequence, and mode register settings).

(1) SDRAM Access Enable/Disable and SDRAM Bus Width Settings

Enabling and disabling SDRAM access is performed by making settings in the individual SDRAMCm control registers to enable or prohibit SDRAMC operation. SDRAM bus width settings are also performed by means of the SDRAMCm control registers.

Even if the SDRAMC control register is set to disable SDRAMC operation, refresh operation will still take place if self-refresh or auto-refresh operation is set as enabled.

(2) SDRAM Commands

SDRAMC controls SDRAM by issuing commands each bus cycle. These commands are defined by combinations of $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, CKE, $\overline{CS}$, etc.

Table 10.9 lists the commands issued by the SDRAMC.

Table 10.9 SDRAMC Commands

Mnemonic	Command	$\overline{SDCS}$	$\overline{RAS}$	$\overline{CAS}$	$\overline{SDWE}$	CKE	A16 (BA1)	A15 (BA0)
DSL	Deselect	H	X	X	X	X	X	X
ACT	Activate row and bank	L	L	H	H	H	V	V
RD	Read	L	H	L	H	H	V	V
WR	Write	L	H	L	L	H	V	V
PRA	Precharge all banks	L	L	H	L	H	X	X
RFA	Auto-refresh	L	L	L	H	H	X	X
MRS	Mode register set	L	L	L	L	H	L	L
EMRS	Extended mode register set	L	L	L	L	H	H	L
RFS	Self-refresh entry	L	L	L	H	H→L	X	X
RFX	Self-refresh exit	H	X	X	X	L→H	X	X
DPD	Deep-power-down	L	H	H	L	H→L	X	X
DPDX	Deep-power-down exit	X	X	X	X	L→H	X	X

[Legend]
H: High Level. L: Low Level. V: Valid. X: Don't Care.

(3) SDRAMC Register Setting Conditions

Rewriting of SDRAMC registers should only be performed when all of the conditions listed in table 10.10 are satisfied.

Table 10.10 Register Rewrite Conditions

Function/Operation	Register Name	Conditions
Self-refresh	SDRFCNT0	• SDRAM access disabled (set in SDRAMCm*[1]) • Auto-refresh enabled (DRFEN = 1) • Power-down disabled (DPWD/DPWDCI = 0) • Deep-power-down disabled (DDPD/DDPDCI = 0)
Auto-refresh	SDRFCNT1	• Self-refresh disabled (DSFEN/DSFENCI = 0) • Power-down disabled (DPWD/DPWDCI = 0)
Initialization sequence	SDIR0	• Before start of initialization sequence
	SDIR1	• After reset or after recovery from deep-power-down
Power-down	SDPWDCNT	• SDRAM access disabled (set in SDRAMCm*[1]) • Auto-refresh enabled (DRFEN = 1) • Self-refresh disabled (DSFEN/DSFENCI = 0) • Deep-power-down disabled (DDPD/DDPDCI = 0)
Deep-power-down	SDDPDCNT	• SDRAM access disabled (set in SDRAMCm*[1]) • Self-refresh disabled (DSFEN/DSFENCI = 0) • Auto-refresh disabled (DRFEN = 0) • Power-down disabled (DPWD/DPWDCI = 0)
Address register settings	SD0ADR, SD1ADR	• Auto-refresh disabled (DRFEN = 0) • SDRAM access disabled (set in SDRAMCm*[1]) • Self-refresh disabled (DSFEN/DSFENCI = 0) • Power-down disabled (DPWD/DPWDCI = 0) • Deep-power-down disabled (DDPD/DDPDCI = 0)
Timing register settings	SD0TR, SD1TR	• Self-refresh in progress (DSFEN/DSFENCI = 1) or • Self-refresh disabled (DSFEN/DSFENCI = 0) • Auto-refresh disabled (DRFEN = 0) • SDRAM access disabled (set in SDRAMCm*[1])

Function/Operation	Register Name	Conditions
Mode register settings	SD0MOD, SD1MOD*2	• SDRAM access disabled (set in SDRAMCm*1) • Self-refresh disabled (DSFEN/DSFENCI = 0) • Power-down disabled (DPWD/DPWDCI = 0) • Deep-power-down disabled (DDPD/DDPDCI = 0)
Clock stop control signal settings	SDCKSCNT	• Deep-power-down disabled (DDPD/DDPDCI = 0)

Notes: 1. After writing 0 to EXENB, make sure that the EXENB bit has been cleared to 0.
2. Before rewriting this bit, make sure that all status bits in the SDRAM status register (SDSTR) have been cleared to 0.

(4) Self-Refresh

Settings in SDRAM refresh control register 0 (SDRFCNT0) control transition to and recovery from self-refresh mode. A transition to and recovery from self-refresh mode takes place simultaneously for all channels. After settings for self-refresh mode have been made, this LSI continues in the self-refresh state even when it is placed on software standby or deep standby. The self-refreshing state is also maintained after interrupt-initiated recovery from standby state. However, the setting for the HIZBCS bit in the HIZCR register must be 0, and the CKE and other pins must be driven even in standby mode. With regard to the HIZCE register, refer to section 30, Power-Down Modes.

An auto-refresh cycle operation takes place immediately before transition to self-refresh mode. The CKE signal is at the low level during periods in self-refresh mode. Immediately after recovery from self-refresh mode, an auto-refresh cycle is triggered.

Figure 10.8 shows the timing of transition to self-refresh mode, and figure 10.9 shows the timing of recovery from self-refresh mode.

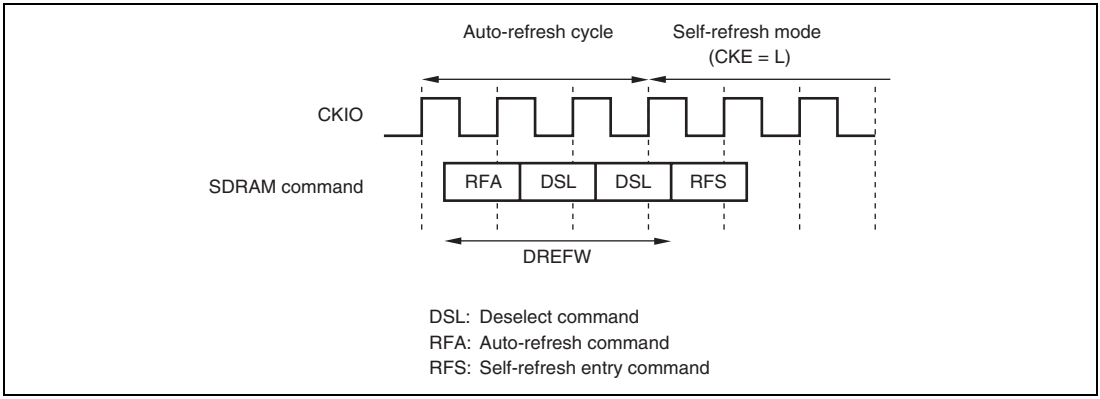


Figure 10.8 Example of Timing of Transition to Self-Refresh Mode (DREFW Bits = 0010)

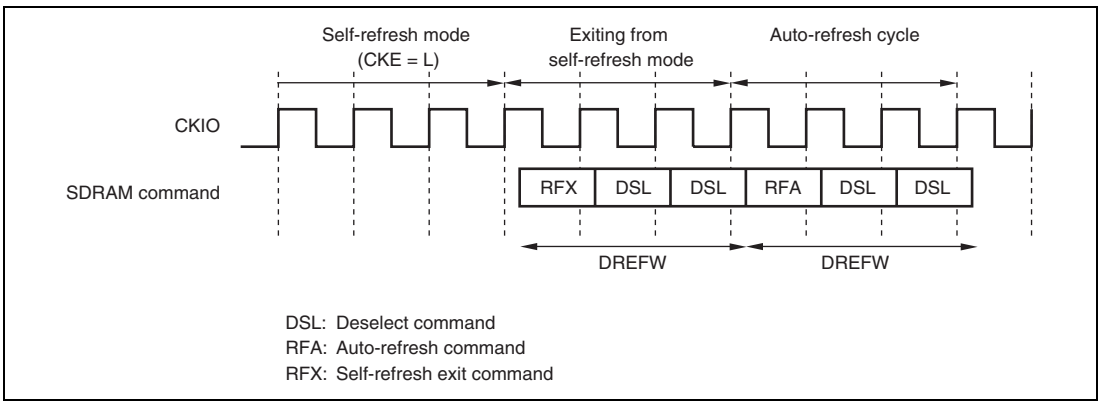


Figure 10.9 Example of Timing of Recovery from Self-Refresh Mode (DREFW Bits = 0010)

(5) Auto-Refresh

An auto-refresh cycle starts when the auto-refresh operation enable bit (DRFEN) in SDRAM refresh control register 1 (SDRFCNT1) is set to 1. After that, refresh requests are issued at fixed intervals according to the refresh counter, activating auto-refresh cycles. However, the activation of auto-refresh cycles may sometimes be delayed because refresh requests are not accepted during read or write accesses.

A refresh request is issued immediately when the auto-refresh operation enable bit (DRFEN) in SDRAM refresh control register 1 (SDRFCNT1) is set to 1 if auto-refresh is enabled.

The refresh counter is halted in self-refresh or deep-power-down mode. After recovery from self-refresh or deep-power-down mode, an auto-refresh cycle is activated, after which the counter value is reset and the counter begins operating again.

Make auto-refresh settings in SDRAM refresh control register 1 (SDRFCNT1). Note that refresh cycles affect SDRAM for all channels. Figure 10.10 shows an auto-refresh cycle timing example.

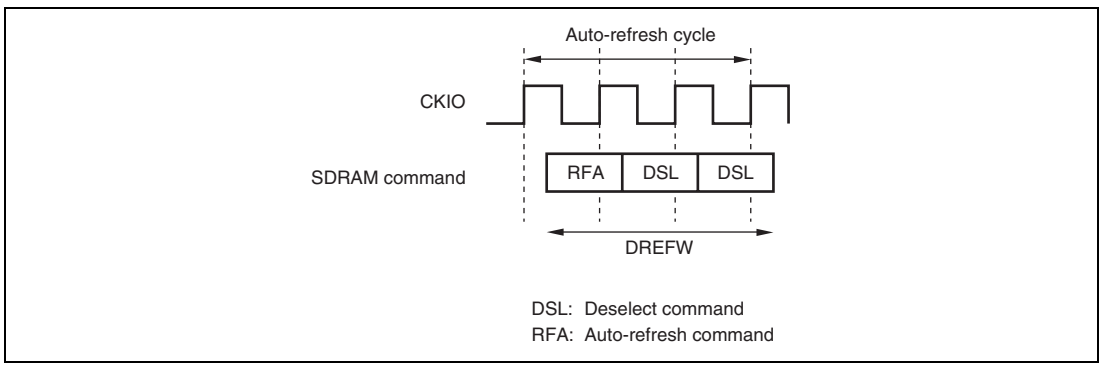


Figure 10.10 Auto-Refresh Cycle Timing Example (DREFW Bits = 0010)

(6) Initialization Sequencer

SDRAMC is provided with a sequencer for issuing the commands for SDRAM initialization. The initialization sequence should always be initiated a single time only following a reset (all channels) or following recovery from deep-power-down mode (individual channels). No normal operation is guaranteed if the initialization sequence is not performed or is performed more than once.

The SDRAM initialization sequencer issues the precharge-all-banks command and n (n = 1 to 15) auto-refresh commands in the stated order. Make timing settings for the initialization sequencer, using SDRAM initialization register 0 (SDIR0). To initiate initialization sequences, use SDRAM initialization register 1 (SDIR1).

Note that an initialization sequence for SDRAM0 and SDRAM1 is initiated simultaneously, using the DINIRQ bit.

Figure 10.11 shows a timing example for the initialization sequence. Setting DARFC to 2 or greater causes multiple initialization auto-refresh cycles to be performed.

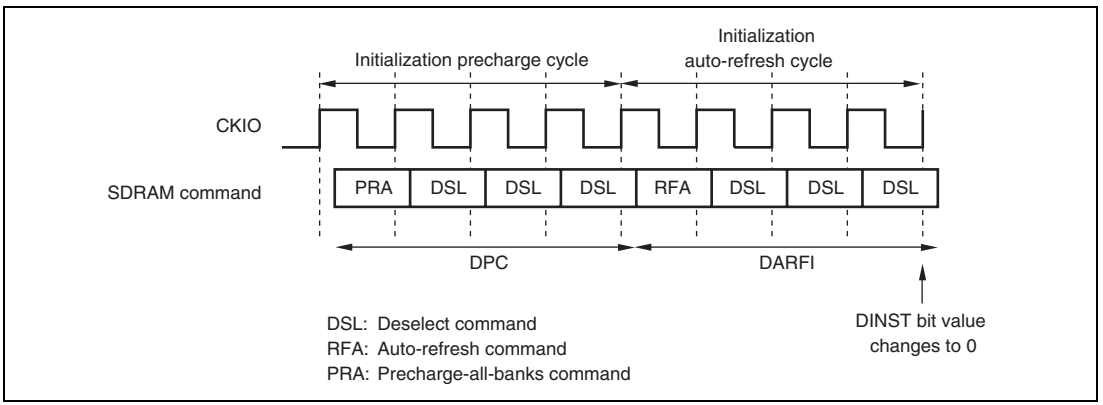


Figure 10.11 Initialization Sequence Timing Example (DPC Bits = 001, DARFI Bits = 0001, DARFC Bits = 001)

(7) Power-Down Mode

SDRAMC supports power-down mode for SDRAM. In power-down mode, it drives the CKE signal low. In power-down mode, auto-refresh operations are performed at the intervals specified by the auto-refresh request interval setting (DRFC) bits in SDRAM refresh control register 1 (SDRFCNT1). The CKE signal goes high only when an auto-refresh command is issued.

To perform transition to and recovery from power-down mode, use the SDRAM power-down control register (SDPWDCNT).

Setting the DPWD bit to 1 causes SDRAMC to enter power-down mode. Clearing the DPWD bit to 0 causes SDRAMC to exit power-down mode.

SDRAMC drives the CKE signal high after recovery from power-down mode.

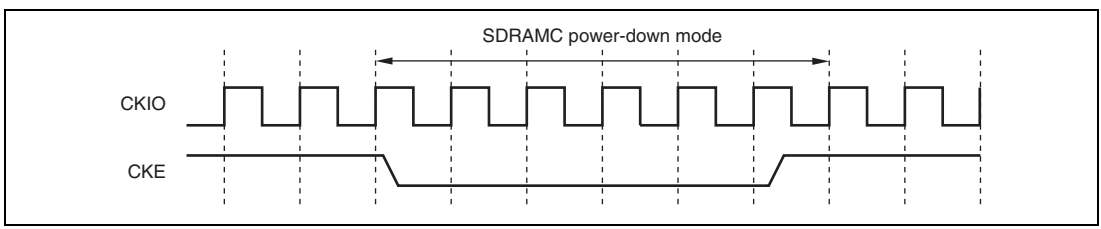


Figure 10.12 SDRAMC Power-Down Mode

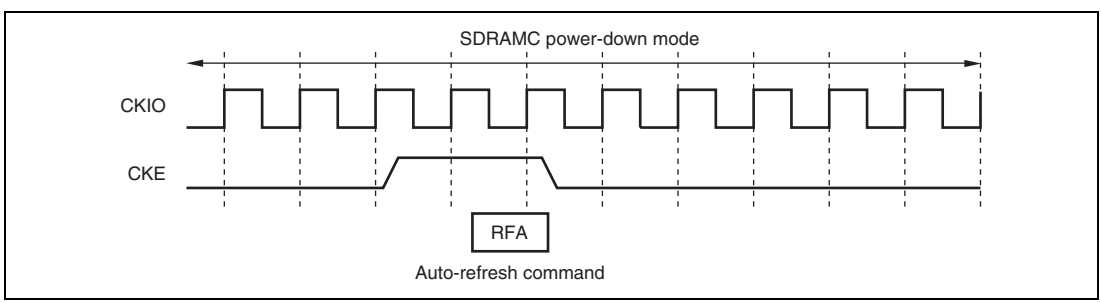


Figure 10.13 Auto-Refresh Operation in SDRAMC Power-Down Mode

(8) Deep-Power-Down Mode

SDRAMC supports deep-power-down mode for SDRAM. In deep-power-down mode, it issues a deep-power-down command to drive the CKE signal low.

To perform transition to and recovery from deep-power-down mode, use the SDRAM deep-power-down control register (SDDPDCNT).

Setting the DDPD bit to 1 causes SDRAM0 and SDRAM1 in SDRAM to enter deep-power-down mode. Clearing the DDPD bit to 0 causes SDRAMC to recover from deep-power-down mode.

After recovery from deep-power-down mode, SDRAMC issues a deep-power-down exit command to drive the CKE signal high.

After waiting for the duration designated for the SDRAM being used after recovery from deep-power-down mode, execute an initialization sequence.

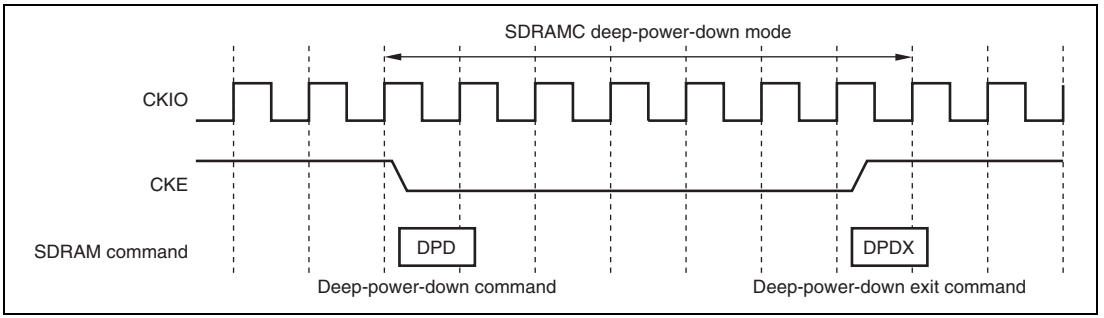


Figure 10.14 SDRAMC Deep-Power-Down Mode

(9) Read/Write Access

The following two types of read/write access are supported.

- Multiple reads/multiple writes
- Single read/single write

Multiple reads/multiple writes occur in the following cases.

1. CPU burst access (cache replace)
2. Longword (32-bit) access to the SDRAM data bus having 8-bit or 16-bit width
3. Word (16-bit) access to the SDRAM data bus having 8-bit width
4. Multiple data transfers by DMA pipeline transfer

The access timing can be set separately for each channel, using the SDRAMm timing register (SDmTR). Access timing examples are described below.

(a) Multiple Read/Multiple Write Accesses

Figure 10.15 shows a timing example for multiple reads of 4 units of data, and figure 10.16, for multiple writes of 4 units of data.

The number of DMA transfers performed will vary depending on factors such as the number of transfers and the transfer data size per operand and the SDRAM bus width. Read commands or write commands may or may not be issued consecutively in response to an access request from the bus master. If read commands or write commands are not issued consecutively, a deselect command is issued between them.

Furthermore, deactivation and activation are performed automatically when the SDRAM row address changes during a DMA transfer operation.

Figure 10.17 shows a timing example for multiple reads of 4 units of data, and figure 10.18, for multiple writes of 4 units of data, unless read/write commands are issued consecutively. Figure 10.19 shows a timing example for multiple writes with a row address change.

The access timing varies according to the settings in the SDRAMm timing register (SDmTR).

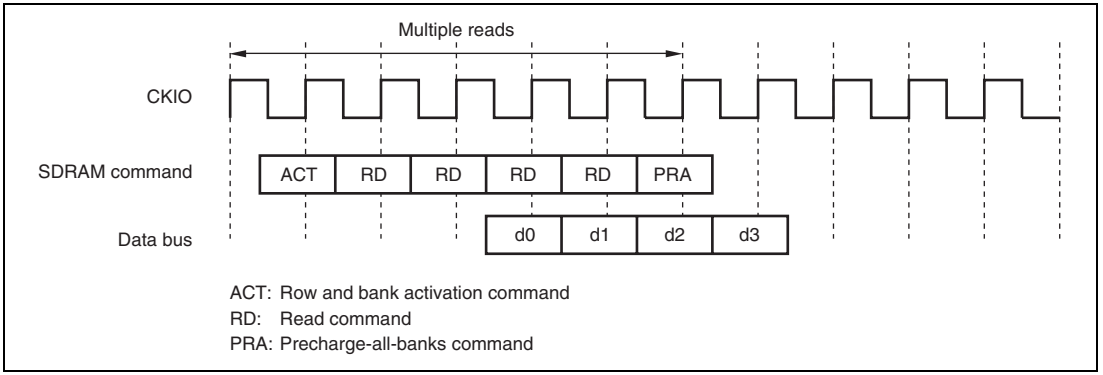


Figure 10.15 Multiple Read Timing Example (Multiple Reads of 4 Data Units, Shortest Timing Settings) Consecutive Read Commands Issued

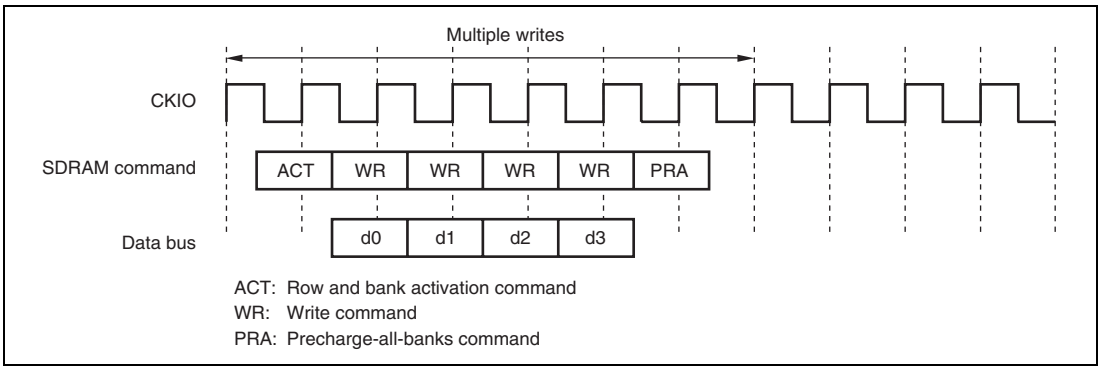


Figure 10.16 Multiple Write Timing Example (Multiple Writes of 4 Data Units, Shortest Timing Settings) Consecutive Write Commands Issued

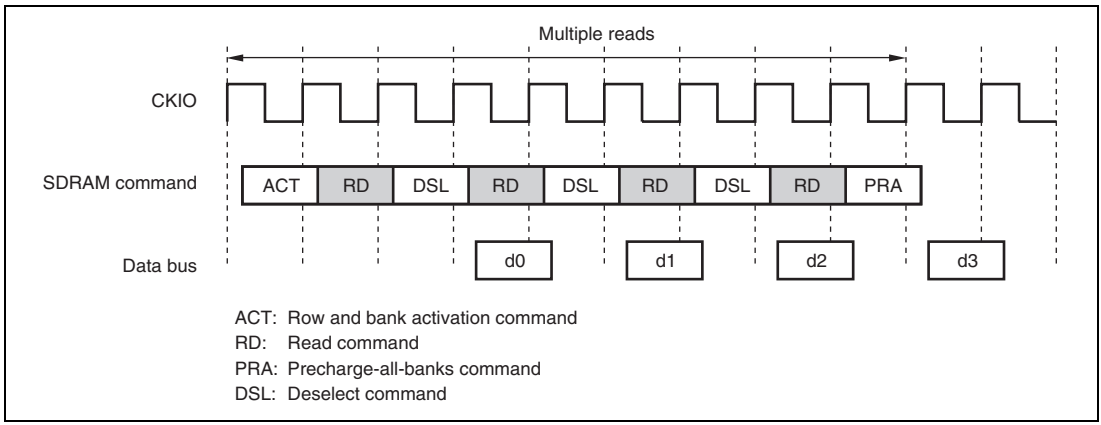


Figure 10.17 Multiple Read Timing Example (Multiple Reads of 4 Data Units, Shortest Timing Settings) Non-Consecutive Read Commands Issued

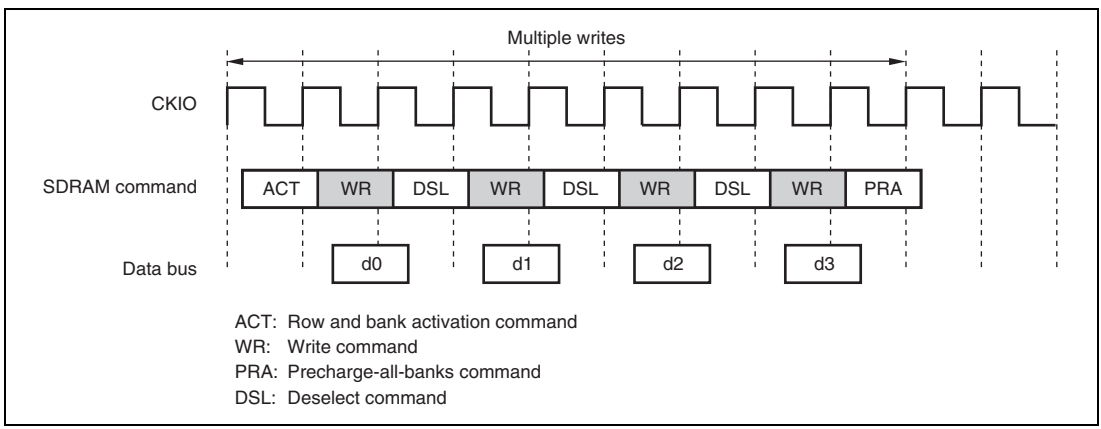


Figure 10.18 Multiple Write Timing Example (Multiple Writes of 4 Data Units, Shortest Timing Settings) Non-Consecutive Write Commands Issued

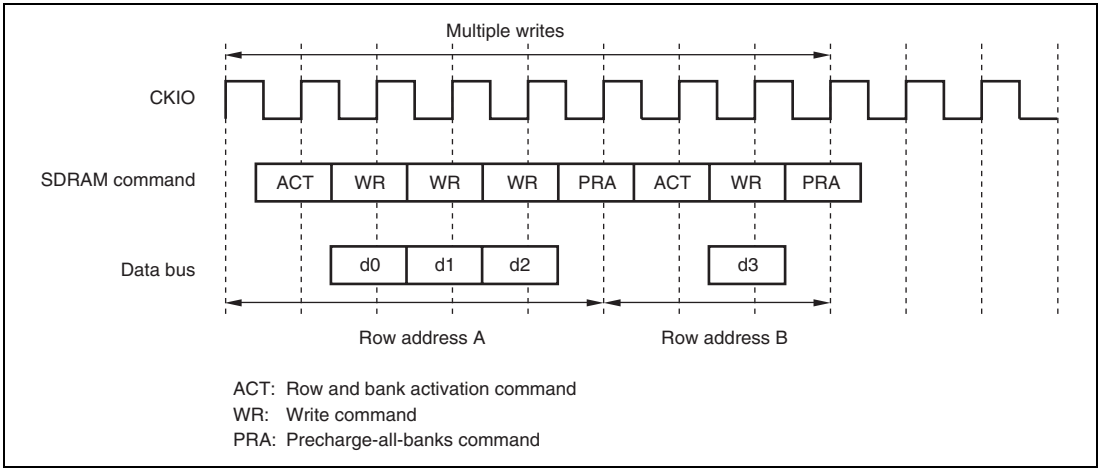


Figure 10.19 Multiple Write Timing Example (Multiple Writes of 4 Data Units, Shortest Timing Settings) Access Spanning Rows

(b) Single Read/Single Write Access

Figure 10.20 shows a timing example for single read operation and figure 10.21 for single write operation. The access timing is modified by means of settings in the SDRAMm timing register (SDmTR).

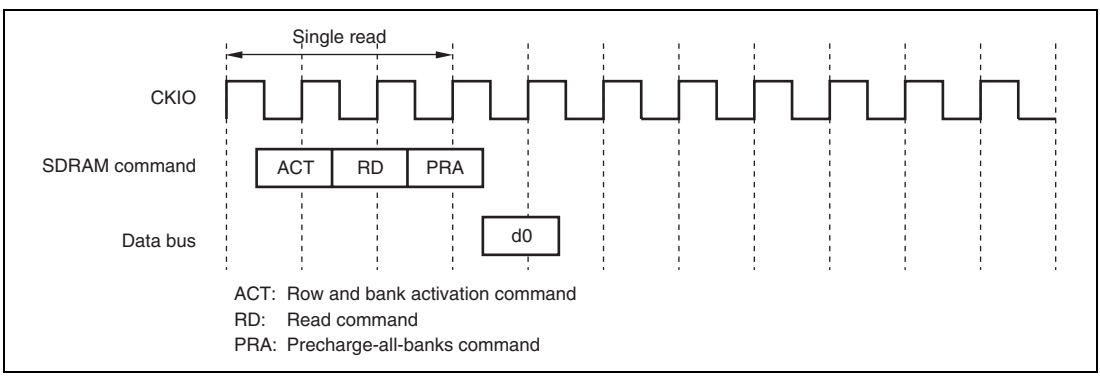


Figure 10.20 Single Read Timing Example (Shortest Timing Settings)

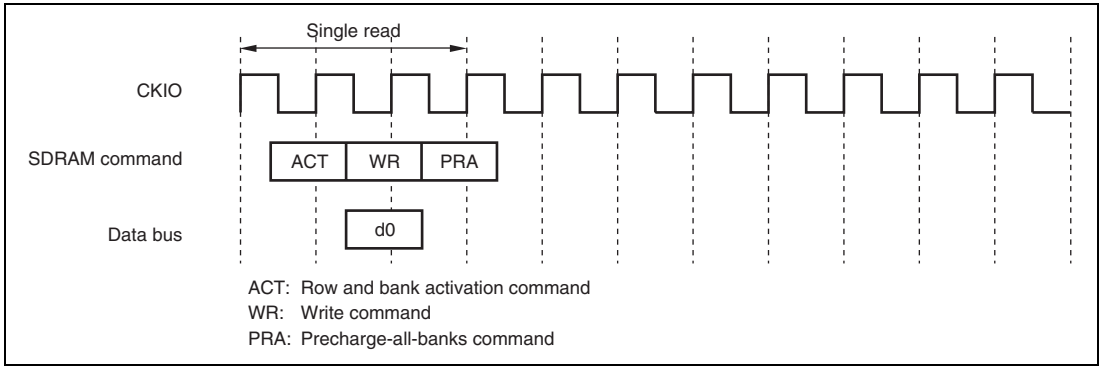


Figure 10.21 Single Write Timing Example (Shortest Timing Settings)

(10) Mode Register Setting

Writing to the SDRAMm mode register (SDmMOD) causes mode register set and extended mode register set commands to be issued to SDRAM for individual channels. Settings the SDRAMm mode register (SDmMOD) should be made separately for each channel.

Figure 10.22 shows the operation timing for mode register setting.

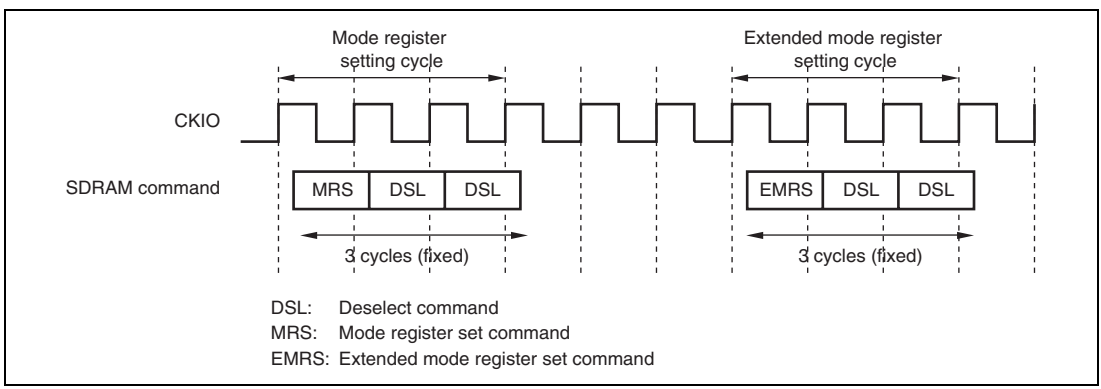


Figure 10.22 Operation Timing for Mode Register Setting

(11) Clock Stop Control Signal

The SDRAMC is provided with a clock stop control function, which can stop CKIO in deep-power-down mode. The function can be enabled or disabled using the DCKSEN bit in the SDRAM clock stop control signal setting register (SDCKSCNT).

CKIO runs continuously if the clock stop control function is disabled.

If the clock stop control function is enabled, CKIO stops or restarts operation in synchronization with transition to or from deep-power-down mode.

In transition to deep-power-down mode, CKIO is stopped (low level) after the deep-power-down entry command is issued. In recovery from deep-power-down mode, a deep-power-down exit command is issued when the clearing of the DDPD bit to 0 is accepted by SDRAMC, and CKIO restarts operating.

DCKSC, the period from the issuance of deep power-down entry (or exit) command until CKIO stops (or restarts) operating, can be set using the SDRAM clock stop control signal setting register.

Figures 10.23 and 10.24 show the operation timing of the clock stop control function.

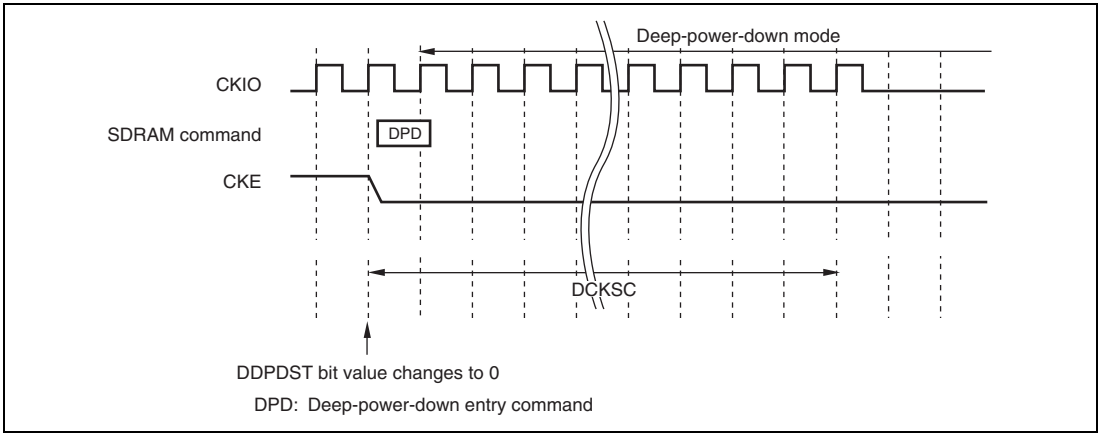


Figure 10.23 Clock Stop Control Function Operation Timing (Transition to Deep-Power-Down Mode)

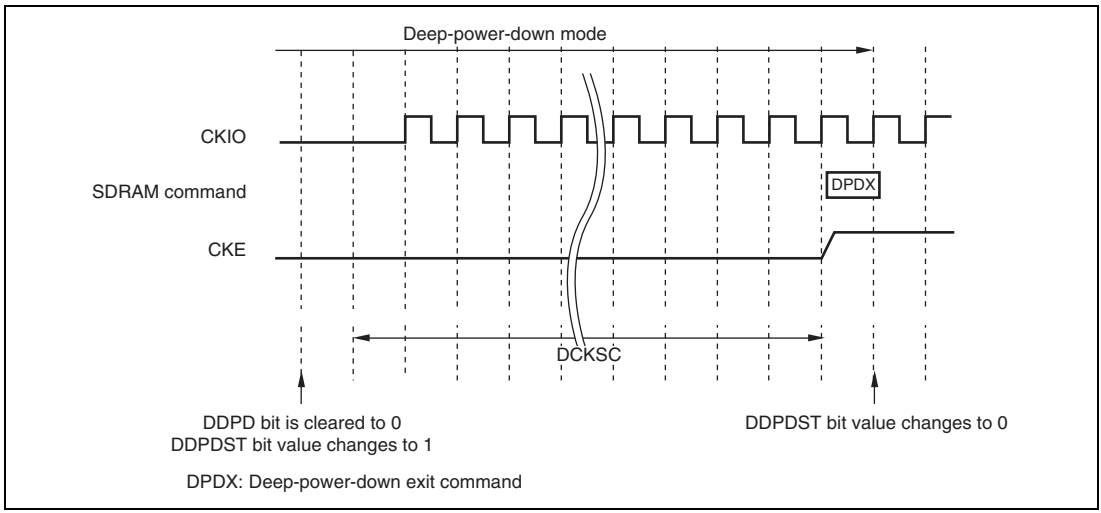


Figure 10.24 Clock Stop Control Function Operation Timing (Recovery from Deep-Power-Down Mode)

(12) SDRAMC Setting Examples

The SDRAMC setting procedure, timing register setting examples, and the procedure for transitioning to and recovering from self-refresh mode, power-down mode, and deep-power-down mode are described below.

(a) SDRAMC Setting Procedure

Figure 10.25 shows the SDRAMC setting procedure.

Note that the specifications such as the power-up sequence may vary with the SDRAM in use.

Study the SDRAM specifications carefully before designing your system. For example, when the SDRAM in use requires that the DQM pin be held "H" during the initialization sequence, set the SDRAM according to the procedure shown in figure 10.25 (b). Since the initialization sequence adopted for this LSI is compliant with the JEDEC standard, the value of DQM pin is not guaranteed from the power-up is supplied and through the initialization sequence.

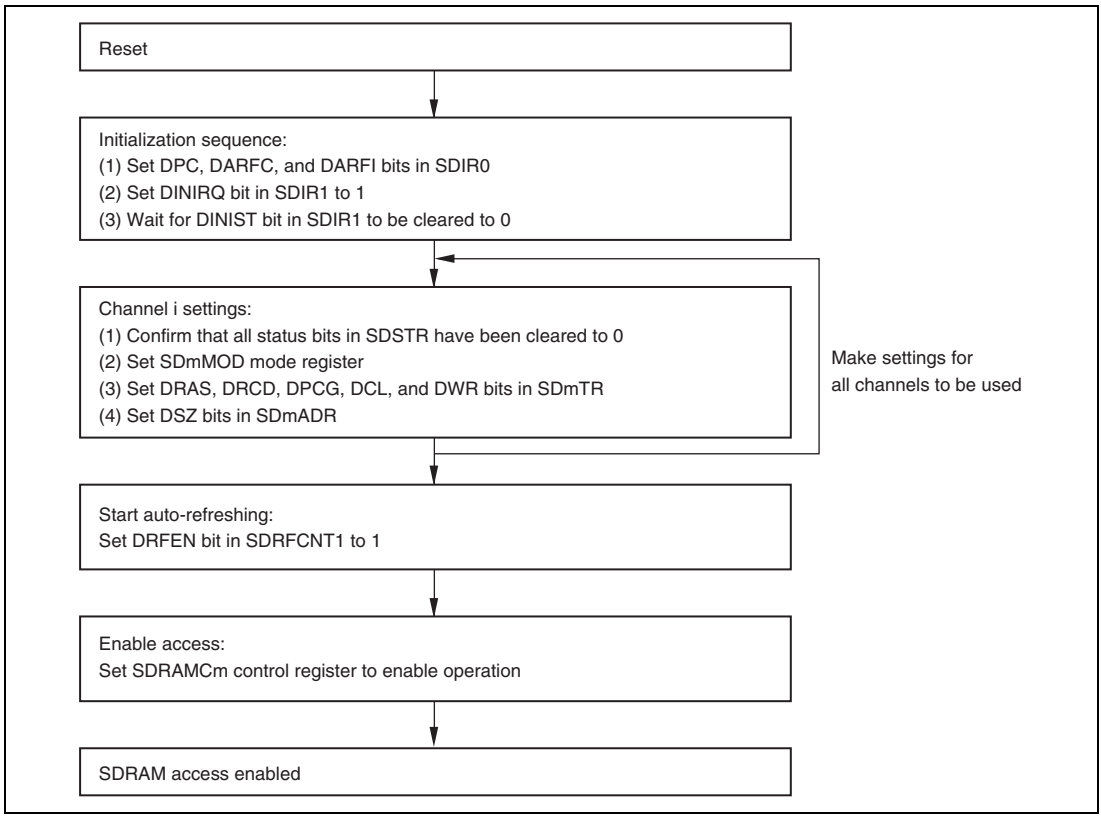


Figure 10.25 (a) SDRAMC Setting Procedure (Basic Setting Example)

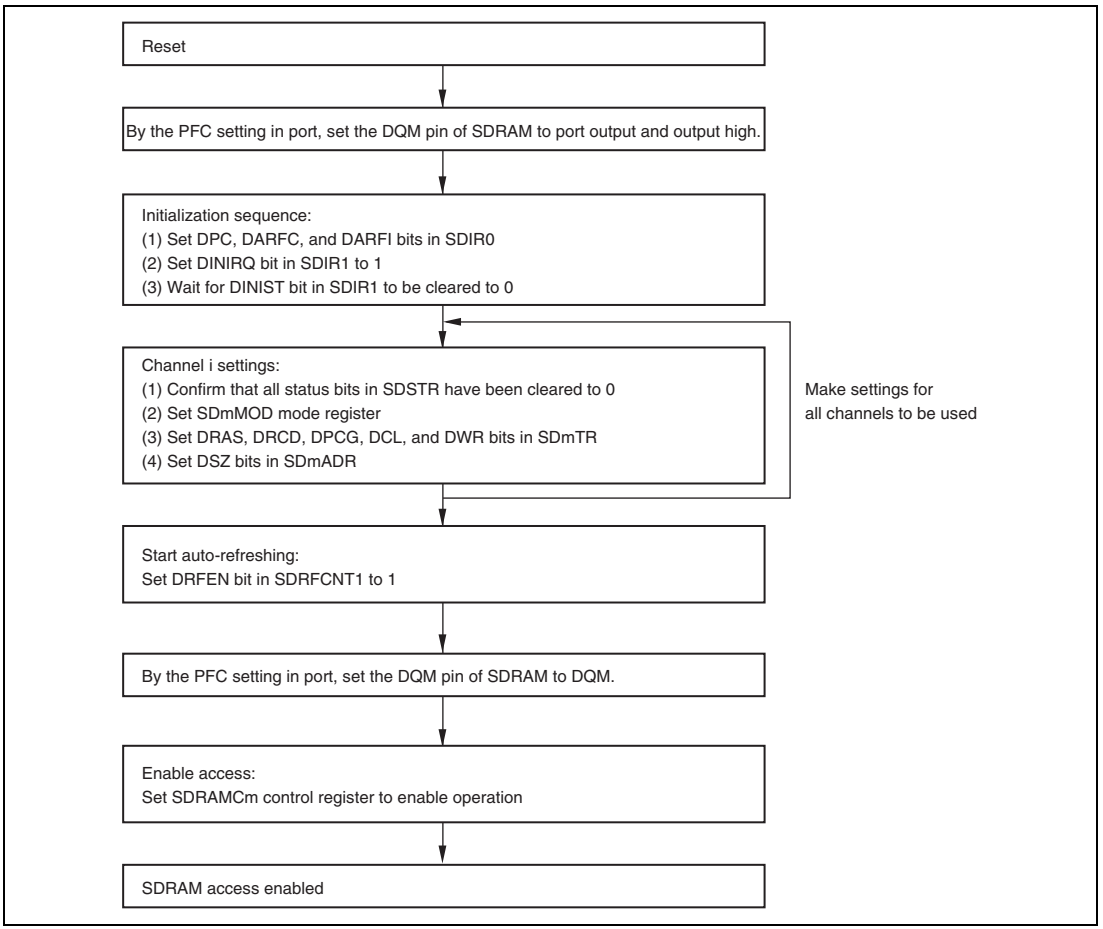


Figure 10.25 (b) SDRAMC Setting Procedure (when DQM Pin is Need to Keep "H" in the Initialization)

(b) Procedure for Transition to and Recovery from Self-Refresh Mode

Figure 10.26 shows the procedure for transitioning to and recovering from self-refresh mode.

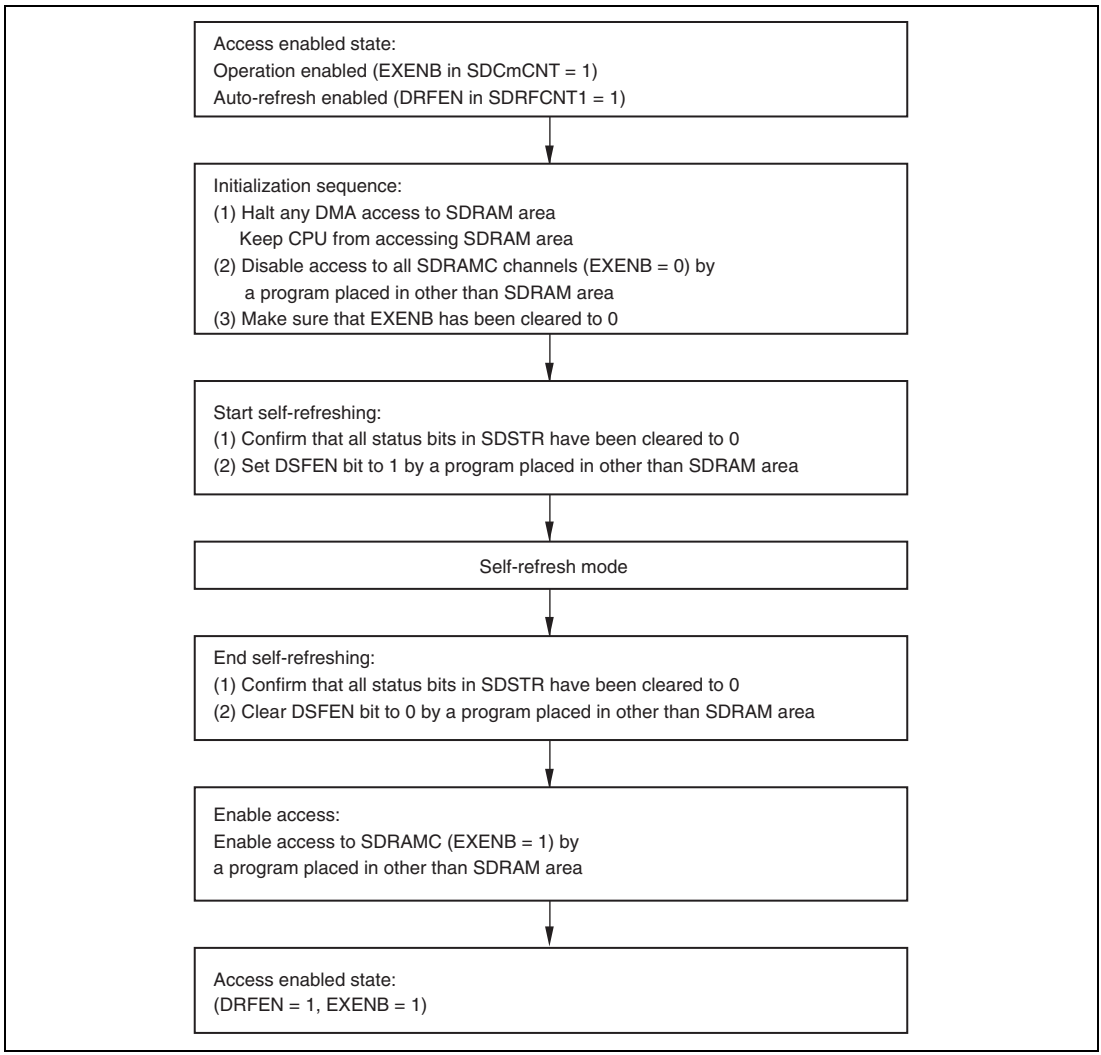


Figure 10.26 Procedure for Transition to and Recovery from Self-Refresh Mode

Notes: Before transitioning to or recovering from self-refresh mode it is necessary to halt SDRAM access to the affected channel. Consequently, it is not possible to transition to or recover from self-refresh mode while programs or DMA operations that access SDRAM are in progress. Pay attention to the following points when writing programs.

1. Before transitioning to self-refresh mode, disable any DMA channel transfers that access the SDRAM area of the channel.
2. If programs are to be executed during transition to self-refresh mode, in self-refresh mode, or during recovery from self-refresh mode, design them in such a way that they will not include operands accessing or fetching (including pre-fetching) instructions stored in the SDRAM area.

(c) Procedure for Transition to and Recovery from Deep-Power-Down Mode

Figure 10.27 shows the procedure for transitioning to deep-power-down mode.

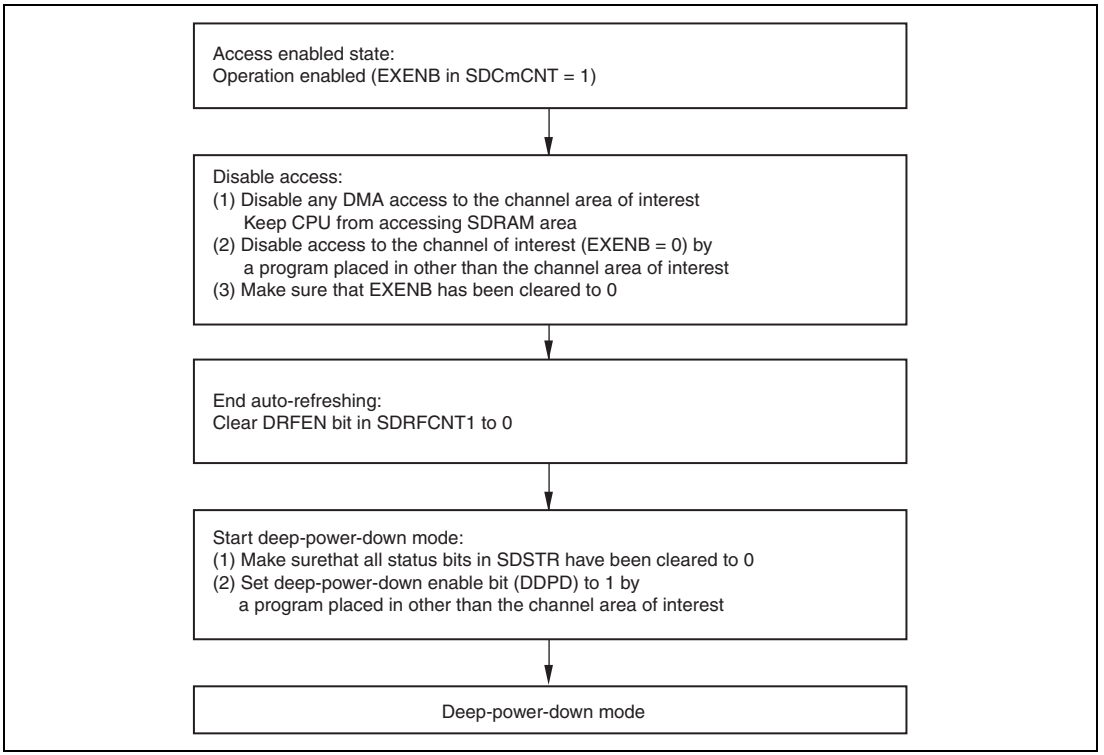


Figure 10.27 Procedure for Transition to Deep-Power-Down Mode

Figure 10.28 shows the procedure for recovering from deep-power-down mode.

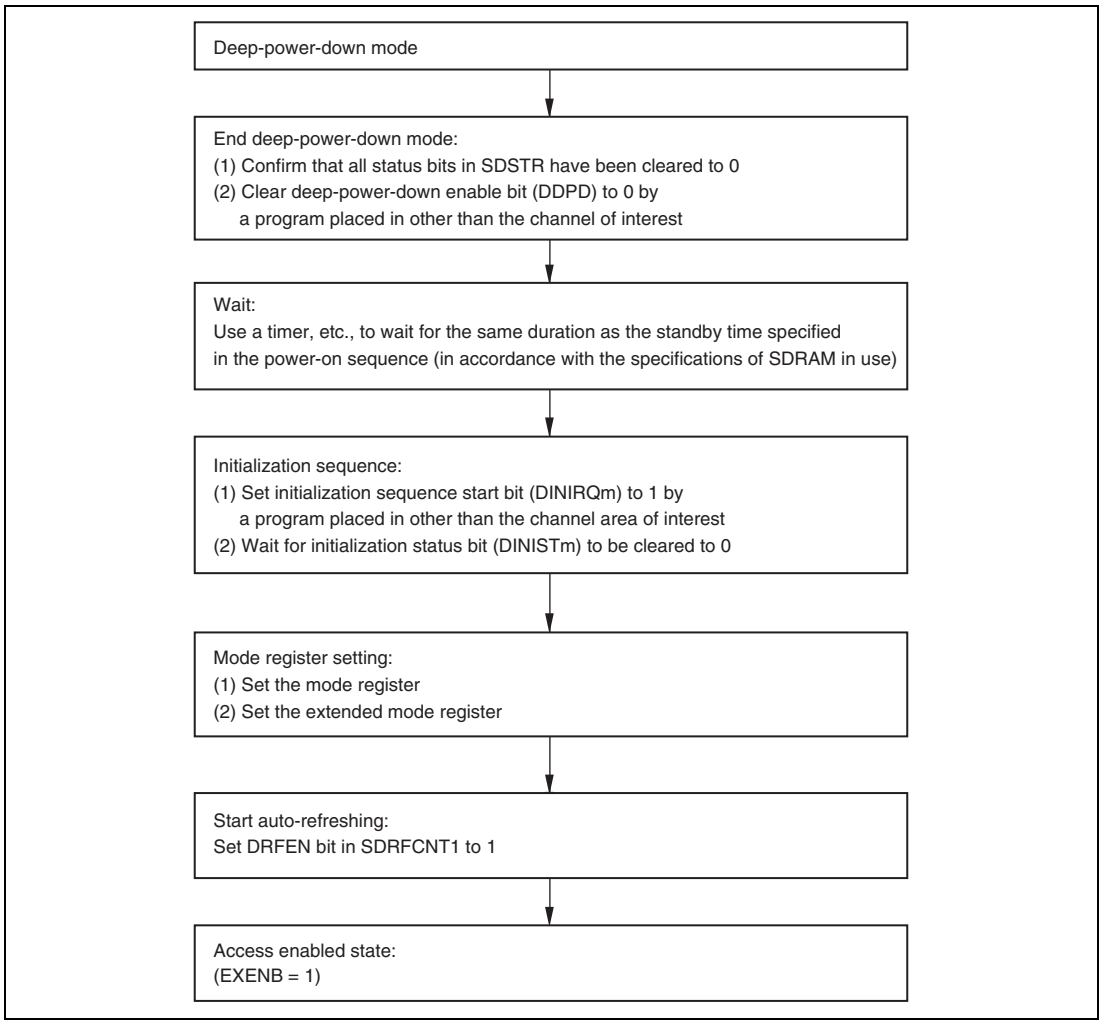


Figure 10.28 Procedure for Recovery from Deep-Power-Down Mode

Notes: Before transitioning to or recovering from deep-power-down mode it is necessary to halt SDRAM access to the affected channels. Consequently, it is not possible to transition to or recover from deep-power-down mode while programs or DMA operations that access SDRAM are in progress. Pay attention to the following points when writing programs.

1. Before transitioning to deep-power-down mode, prohibit any DMA channel transfers that access the SDRAM area of the affected channels.
2. If programs are to be executed during transition to deep-power-down mode, in deep-power-down mode, or during recovery from deep-power-down mode, design them in such a way that they will not include operands accessing or fetching (including pre-fetching) instructions stored in the SDRAM area.

(d) Timing Register Set Values and Access Timing

The correspondence between the SDRAMm timing register (SDmTR) set values and the read and write access timing is described below.

- Multiple Read Timing Setting Examples

Figures 10.29 to 10.31 show the correspondence between the timing of multiple read operations involving 4 data units and the set values of the SDRAMm timing register (SDmTR). Table 10.11 lists the SDRAMm timing register (SDmTR) set values for each figure.

Table 10.11 SDITR Set Value Correspondence Table (Multiple Read Timing)

Figure	DRAS	DRCD	DPCG	DCL
Figure 10.29	010	00	001	010
Figure 10.30	000	01	001	010
Figure 10.31	000	01	001	011

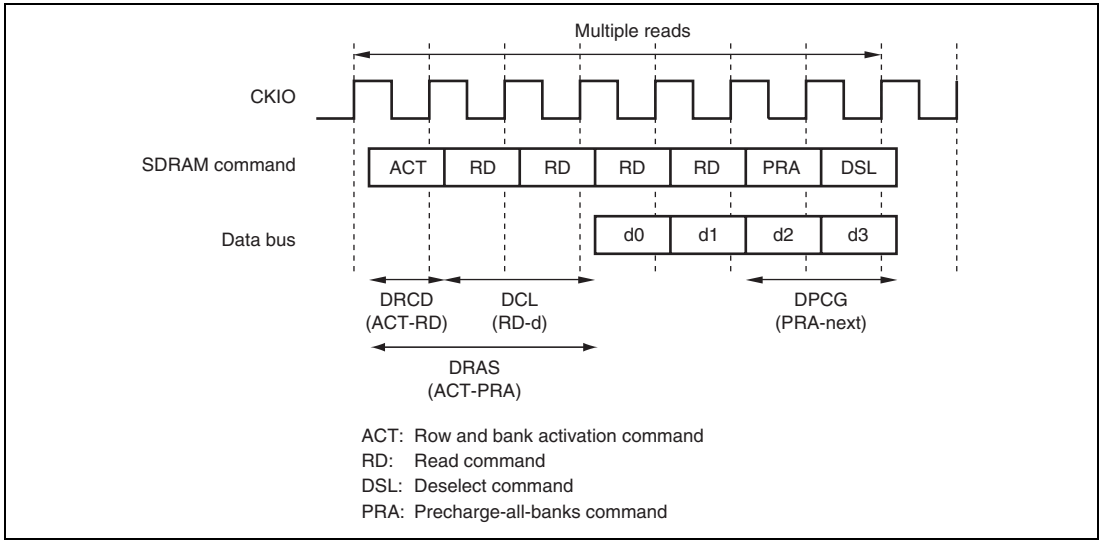


Figure 10.29 Multiple Read Timing Example 1

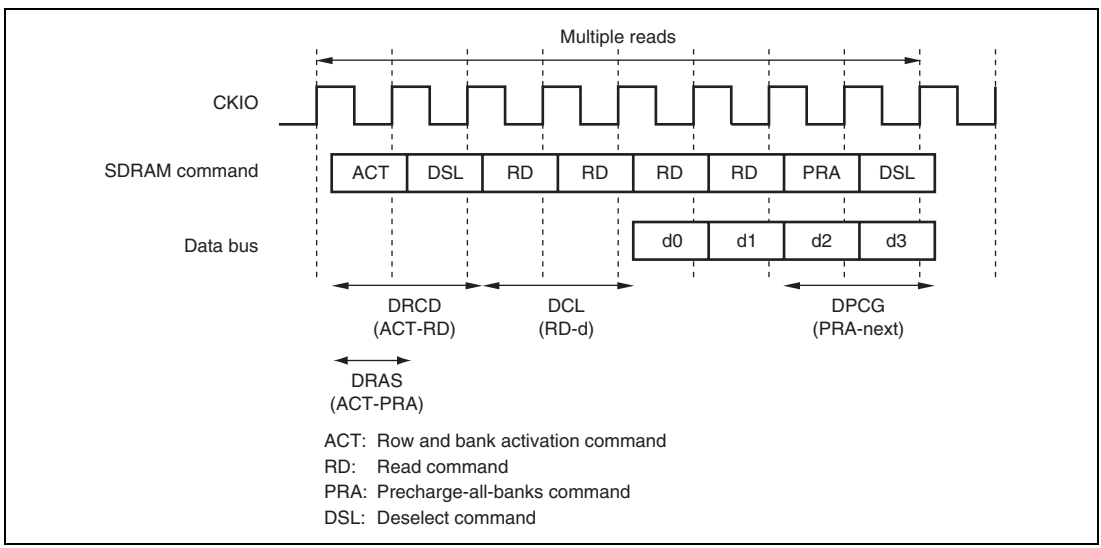


Figure 10.30 Multiple Read Timing Example 2

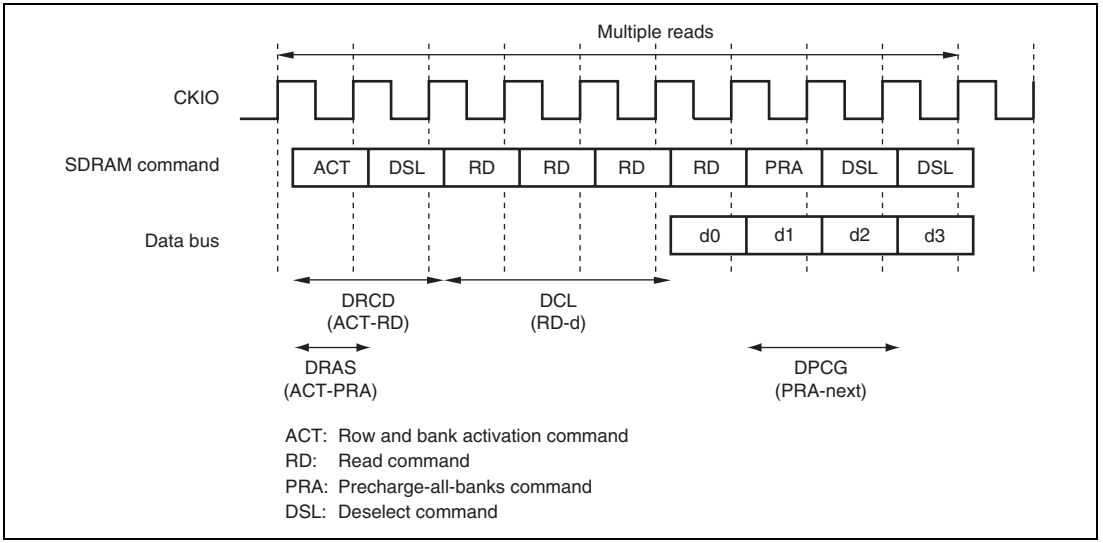


Figure 10.31 Multiple Read Timing Example 3

- Multiple Write Timing Setting Examples

Figures 10.32 to 10.34 show the correspondence between the timing of multiple write operations involving 4 data units and the set values of the SDRAMm timing register (SDmTR). Table 10.12 lists the SDRAMm timing register (SDmTR) set values for each figure.

Table 10.12 SDITR Set Value Correspondence Table (Multiple Write Timing)

Figure	DRAS	DRCD	DPCG	DWR
Figure 10.32	010	00	001	0
Figure 10.33	000	01	001	0
Figure 10.34	000	01	001	1

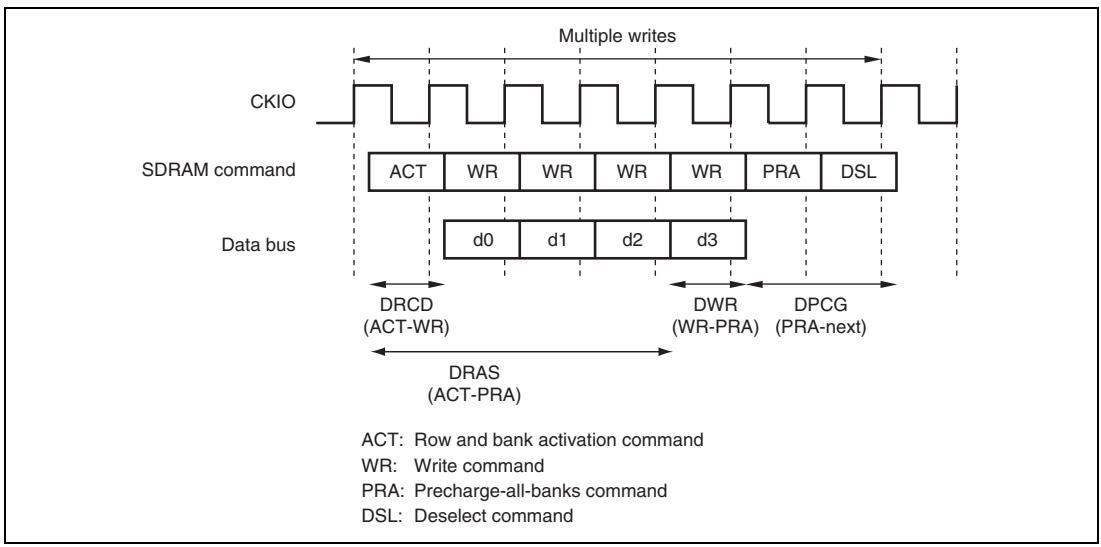


Figure 10.32 Multiple Write Timing Example 1

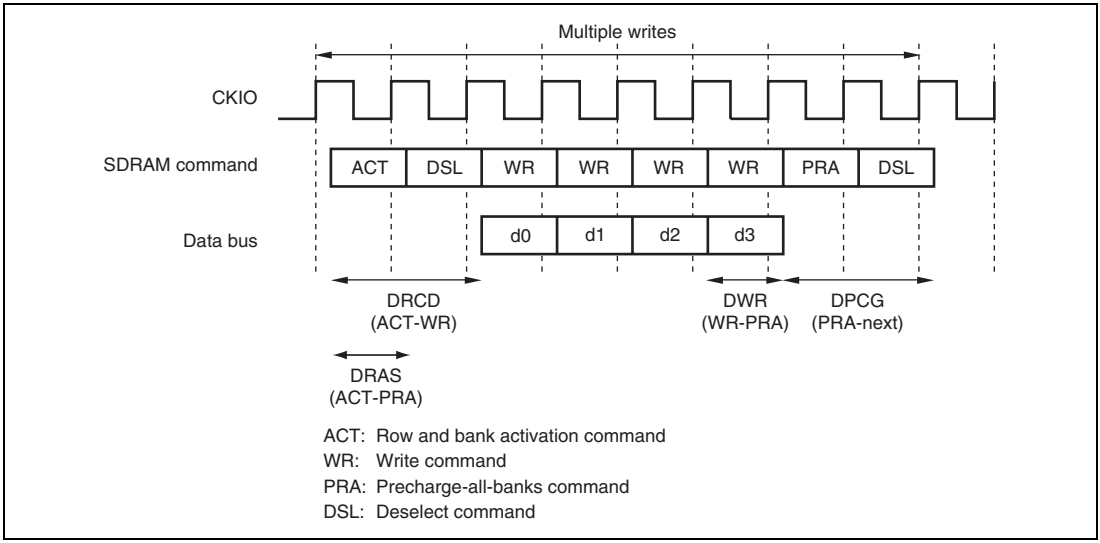


Figure 10.33 Multiple Write Timing Example 2

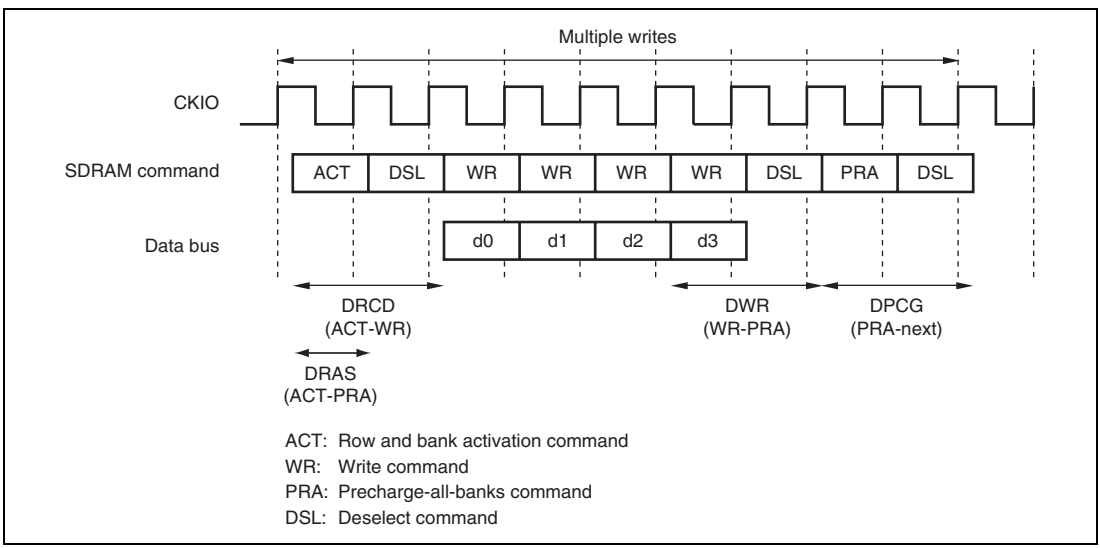


Figure 10.34 Multiple Write Timing Example 3

- Single Read Timing Setting Examples

Figures 10.35 to 10.37 show the correspondence between the timing of single read operations and the set values of the SDRAMm timing register (SDmTR). Table 10.13 lists the SDRAMm timing register (SDmTR) set values for each figure.

Table 10.13 SDITR Set Value Correspondence Table (Single Read Timing)

Figure	DRAS	DRCD	DPCG	DCL
Figure 10.35	010	00	001	010
Figure 10.36	000	01	001	010
Figure 10.37	000	01	001	011

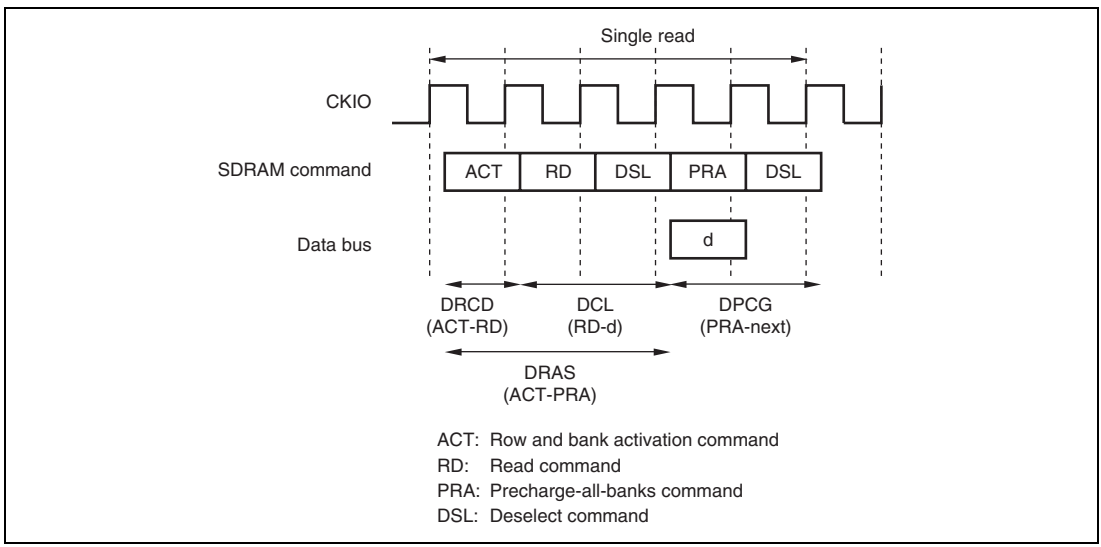
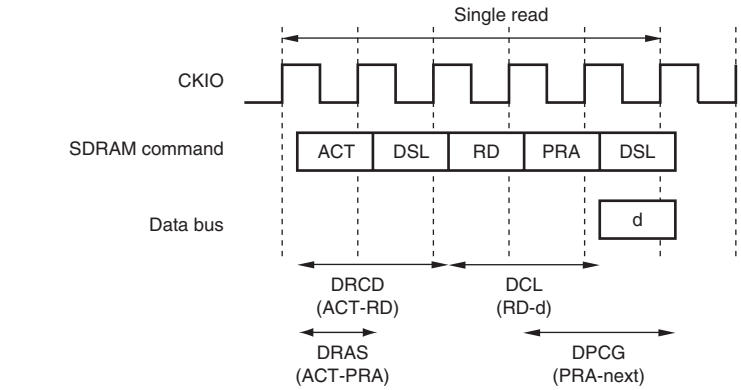


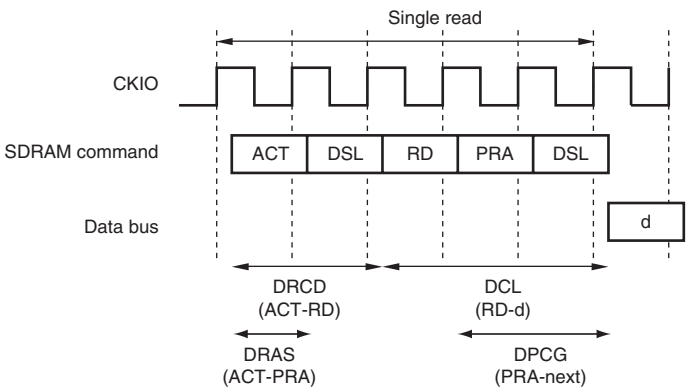
Figure 10.35 Single Read Timing Example 1



ACT: Row and bank activation command
RD: Read command
DSL: Deselect command
PRA: Precharge-all-banks command

Note: If the interval set in DRAS ends before RD, PRA is issued in the cycle after RD.

Figure 10.36 Single Read Timing Example 2



ACT: Row and bank activation command
RD: Read command
DSL: Deselect command
PRA: Precharge-all-banks command

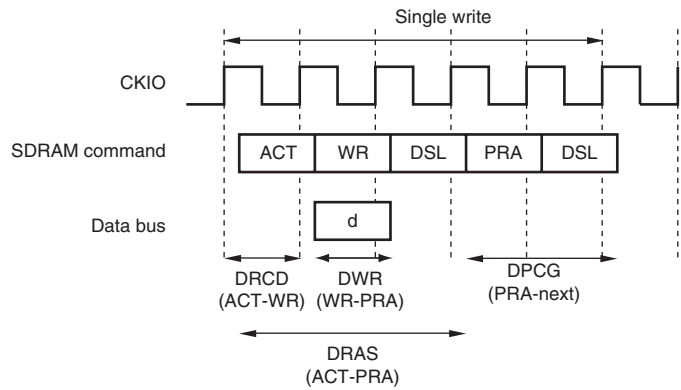
Figure 10.37 Single Read Timing Example 3

- Single Write Timing Setting Examples

Figures 10.38 to 10.40 show the correspondence between the timing of single write operations and the set values of the SDRAMm timing register (SDmTR). Table 10.14 lists the SDRAMm timing register (SDmTR) set values for each figure.

Table 10.14 SDITR Set Value Correspondence Table (Single Write Timing)

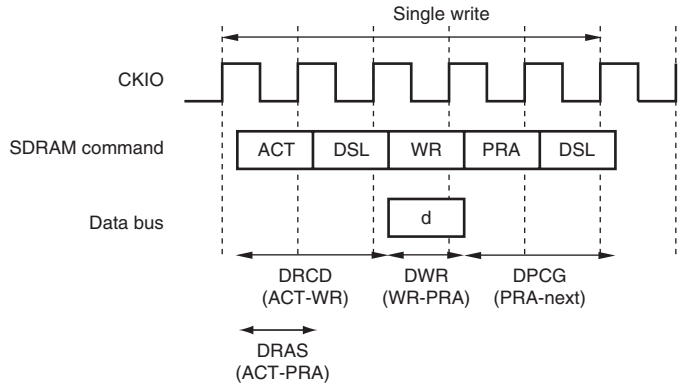
Figure	DRAS	DRCD	DPCG	DWR
Figure 10.38	010	00	001	0
Figure 10.39	000	01	001	0
Figure 10.40	000	01	001	1



ACT: Row and bank activation command
WR: Write command
DSL: Deselect command
PRA: Precharge-all-banks command

Note: If the interval set in DRAS is longer than the period from when the WR command is issued until the DWR interval elapses, the DRAS setting is used.

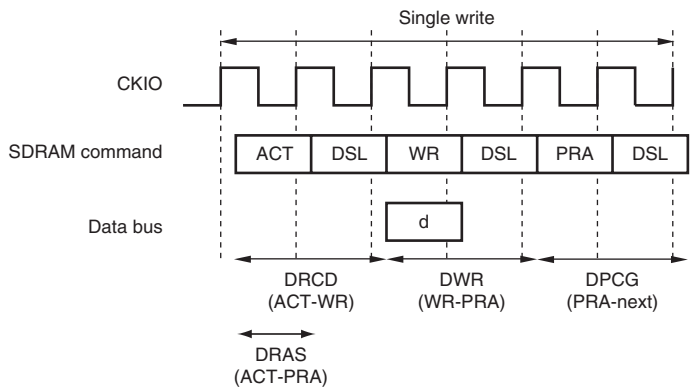
Figure 10.38 Single Write Timing Example 1



ACT: Row and bank activation command
WR: Write command
DSL: Deselect command
PRA: Precharge-all-banks command

Note: If the interval set in DRAS is shorter than the period from when the WR command is issued until the DWR interval elapses, the DWR setting is used.

Figure 10.39 Single Write Timing Example 2



ACT: Row and bank activation command
WR: Write command
DSL: Deselect command
PRA: Precharge-all-banks command

Figure 10.40 Single Write Timing Example 3

(13) External Address/SDRAM Address Signal Multiplexing

(a) Address Multiplexing

Either of addresses used for accessing an external device or SDRAM is output through external address pins.

Table 10.15 External Address/SDRAM Address Pins

Pin Name	Function
A[25]	External address
A[24]	External address
A[23]	External address
A[22]	External address
A[21]	External address
A[20]	External address
A[19]	External address
A[18]	External address
A[17]	External address
A[16]/ba[1]	External address/SDRAM bank address
A[15]/ba[0]	External address/SDRAM band address
A[14]/ma[12]	External address/SDRAM address
A[13]/ma[11]	External address/SDRAM address
A[12]/ma[10]	External address/SDRAM address
A[11]/ma[9]	External address/SDRAM address
A[10]/ma[8]	External address/SDRAM address
A[9]/ma[7]	External address/SDRAM address
A[8]/ma[6]	External address/SDRAM address
A[7]/ma[5]	External address/SDRAM address
A[6]/ma[4]	External address/SDRAM address
A[5]/ma[3]	External address/SDRAM address
A[4]/ma[2]	External address/SDRAM address
A[3]/ma[1]	External address/SDRAM address
A[2]/ma[0]	External address/SDRAM address
A[1]	External address
A[0]	External address

(14) Address Register Setting Values

(a) Supported SDRAM Configurations

Tables 10.16 to 10.21 list the SDRAM configurations supported for bus widths of 8, 16, and 32 bits. These tables are intended to help understand the relationships between the supported SDRAM configurations and address multiplexing.

addr[25:0] is the logical address used by the CPU and DMAC in access to the SDRAM. The table below lists how the settings of DSZ and DDBW determine which signals are output on the SDRAM-access pins.

Table 10.16 Case for 8-Bit External Data Bus Width (BSIZE*[1] = (1, 0))

SDRAM Type	64 Mbits (× 8)		128 Mbits (× 8)		256 Mbits (× 8)		512 Mbits (× 8)	
Number	1		1		1		1	
DSZ*[2]	001 (8 Mbytes)		010 (16 Mbytes)		011 (32 Mbytes)		100 (64 Mbytes)	
DDBW*[3]	00 (8 bits)		00 (8 bits)		00 (8 bits)		00 (8 bits)	
Output Pin of This LSI	Row Address Cycle	Column Address Cycle	Row Address Cycle	Column Address Cycle	Row Address Cycle	Column Address Cycle	Row Address Cycle	Column Address Cycle
A[16]/ba[1]*[4]	addr[22]	addr[22]	addr[23]	addr[23]	addr[24]	addr[24]	addr[25]	addr[25]
A[15]/ba[0]*[4]	addr[21]	addr[21]	addr[22]	addr[22]	addr[23]	addr[23]	addr[24]	addr[24]
A[14]/ma[12]*[4]	0	0	0	0	addr[22]	0	addr[23]	0
A[13]/ma[11]*[4]	addr[20]	0	addr[21]	0	addr[21]	0	addr[22]	addr[10]
A[12]/ma[10]*[4]	addr[19]	*[5]	addr[20]	*[5]	addr[20]	*[5]	addr[21]	*[5]
A[11]/ma[9]*[4]	addr[18]	0	addr[19]	addr[9]	addr[19]	addr[9]	addr[20]	addr[9]
A[10]/ma[8]*[4]	addr[17]	addr[8]	addr[18]	addr[8]	addr[18]	addr[8]	addr[19]	addr[8]
A[9]/ma[7]*[4]	addr[16]	addr[7]	addr[17]	addr[7]	addr[17]	addr[7]	addr[18]	addr[7]
A[8]/ma[6]*[4]	addr[15]	addr[6]	addr[16]	addr[6]	addr[16]	addr[6]	addr[17]	addr[6]
A[7]/ma[5]*[4]	addr[14]	addr[5]	addr[15]	addr[5]	addr[15]	addr[5]	addr[16]	addr[5]
A[6]/ma[4]*[4]	addr[13]	addr[4]	addr[14]	addr[4]	addr[14]	addr[4]	addr[15]	addr[4]
A[5]/ma[3]*[4]	addr[12]	addr[3]	addr[13]	addr[3]	addr[13]	addr[3]	addr[14]	addr[3]
A[4]/ma[2]*[4]	addr[11]	addr[2]	addr[12]	addr[2]	addr[12]	addr[2]	addr[13]	addr[2]
A[3]/ma[1]*[4]	addr[10]	addr[1]	addr[11]	addr[1]	addr[11]	addr[1]	addr[12]	addr[1]
A[2]/ma[0]*[4]	addr[9]	addr[0]	addr[10]	addr[0]	addr[10]	addr[0]	addr[11]	addr[0]

Notes: 1. BSIZE represents the BSIZE bit in the SDCmCNT control register.
2. DSZ represents the DSZ bit in the SDRAMm address register.
3. DDBW represents the DDBW bit in the SDRAMm address register.
4. ba[1:0] and ma[12:0] represent, respectively, the SDRAM bank address and SDRAM address.
5. When the RD, WR or PRA command is issued, this carries the pre-charge option signal.

Table 10.17 Case for 16-Bit External Data Bus Width (BSIZE*1 = (0, 0)) (1)

SDRAM Type	64 Mbits (× 16)		64 Mbits (× 8)		128 Mbits (× 16)		128 Mbits (× 8)	
Number	1		2		1		2	
DSZ*2	001 (8 Mbytes)		010 (16 Mbytes)		010 (16 Mbytes)		011 (32 Mbytes)	
DDBW*3	01 (16 bits)		00 (8 bits)		01 (16 bits)		00 (8 bits)	
Output Pin of This LSI	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address
A[16]/ba[1]*4	addr[22]	addr[22]	addr[23]	addr[23]	addr[23]	addr[23]	addr[24]	addr[24]
A[15]/ba[0]*4	addr[21]	addr[21]	addr[22]	addr[22]	addr[22]	addr[22]	addr[23]	addr[23]
A[14]/ma[12]*4	0	0	0	0	0	0	0	0
A[13]/ma[11]*4	addr[20]	0	addr[21]	0	addr[21]	0	addr[22]	0
A[12]/ma[10]*4	addr[19]	*5	addr[20]	*5	addr[20]	*5	addr[21]	*5
A[11]/ma[9]*4	addr[18]	0	addr[19]	0	addr[19]	0	addr[20]	addr[10]
A[10]/ma[8]*4	addr[17]	0	addr[18]	addr[9]	addr[18]	addr[9]	addr[19]	addr[9]
A[9]/ma[7]*4	addr[16]	addr[8]	addr[17]	addr[8]	addr[17]	addr[8]	addr[18]	addr[8]
A[8]/ma[6]*4	addr[15]	addr[7]	addr[16]	addr[7]	addr[16]	addr[7]	addr[17]	addr[7]
A[7]/ma[5]*4	addr[14]	addr[6]	addr[15]	addr[6]	addr[15]	addr[6]	addr[16]	addr[6]
A[6]/ma[4]*4	addr[13]	addr[5]	addr[14]	addr[5]	addr[14]	addr[5]	addr[15]	addr[5]
A[5]/ma[3]*4	addr[12]	addr[4]	addr[13]	addr[4]	addr[13]	addr[4]	addr[14]	addr[4]
A[4]/ma[2]*4	addr[11]	addr[3]	addr[12]	addr[3]	addr[12]	addr[3]	addr[13]	addr[3]
A[3]/ma[1]*4	addr[10]	addr[2]	addr[11]	addr[2]	addr[11]	addr[2]	addr[12]	addr[2]
A[2]/ma[0]*4	addr[9]	addr[1]	addr[10]	addr[1]	addr[10]	addr[1]	addr[11]	addr[1]

Notes: 1. BSIZE represents the BSIZE bit in the SDCmCNT control register.
2. DSZ represents the DSZ bit in the SDRAMm address register.
3. DDBW represents the DDBW bit in the SDRAMm address register.
4. ba[1:0] and ma[12:0] represent, respectively, the SDRAM bank address and SDRAM address.
5. When the RD, WR or PRA command is issued, this carries the pre-charge option signal.

Table 10.18 Case for 16-Bit External Data Bus Width (BSIZE*1 = (0, 0)) (2)

SDRAM Type	256 Mbits (× 16)		256 Mbits (× 8)		512 Mbits (× 16)	
Number	1		2		1	
DSZ*2	011 (32 Mbytes)		100 (64 Mbytes)		100 (64 Mbytes)	
DDBW*3	01 (16 bits)		00 (8 bits)		01 (16 bits)	
Output Pin of This LSI	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address
A[16]/ba[1]*4	addr[24]	addr[24]	addr[25]	addr[25]	addr[25]	addr[25]
A[15]/ba[0]*4	addr[23]	addr[23]	addr[24]	addr[24]	addr[24]	addr[24]
A[14]/ma[12]*4	addr[22]	0	addr[23]	0	addr[23]	0
A[13]/ma[11]*4	addr[21]	0	addr[22]	0	addr[22]	0
A[12]/ma[10]*4	addr[20]	*5	addr[21]	*5	addr[21]	*5
A[11]/ma[9]*4	addr[19]	0	addr[20]	addr[10]	addr[20]	addr[10]
A[10]/ma[8]*4	addr[18]	addr[9]	addr[19]	addr[9]	addr[19]	addr[9]
A[9]/ma[7]*4	addr[17]	addr[8]	addr[18]	addr[8]	addr[18]	addr[8]
A[8]/ma[6]*4	addr[16]	addr[7]	addr[17]	addr[7]	addr[17]	addr[7]
A[7]/ma[5]*4	addr[15]	addr[6]	addr[16]	addr[6]	addr[16]	addr[6]
A[6]/ma[4]*4	addr[14]	addr[5]	addr[15]	addr[5]	addr[15]	addr[5]
A[5]/ma[3]*4	addr[13]	addr[4]	addr[14]	addr[4]	addr[14]	addr[4]
A[4]/ma[2]*4	addr[12]	addr[3]	addr[13]	addr[3]	addr[13]	addr[3]
A[3]/ma[1]*4	addr[11]	addr[2]	addr[12]	addr[2]	addr[12]	addr[2]
A[2]/ma[0]*4	addr[10]	addr[1]	addr[11]	addr[1]	addr[11]	addr[1]

Notes: 1. BSIZE represents the BSIZE bit in the SDCmCNT control register.
2. DSZ represents the DSZ bit in the SDRAMm address register.
3. DDBW represents the DDBW bit in the SDRAMm address register.
4. ba[1:0] and ma[12:0] represent, respectively, the SDRAM bank address and SDRAM address.
5. When the RD, WR or PRA command is issued, this carries the pre-charge option signal.

Table 10.19 Case for 32-Bit External Data Bus Width (BSIZE*[1] = (0, 1)) (1)

SDRAM Type	64 Mbits (× 32)		64 Mbits (× 16)		128 Mbits (× 32)		64 Mbits (× 8)	
Number	1		2		1		4	
DSZ*[2]	001 (8 Mbytes)		010 (16 Mbytes)		010 (16 Mbytes)		011 (32 Mbytes)	
DDBW*[3]	10 (32 bits)		01 (16 bits)		10 (32 bits)		00 (8 bits)	
Output Pin of This LSI	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address
A[16]/ba[1]*[4]	addr[22]	addr[22]	addr[23]	addr[23]	addr[23]	addr[23]	addr[24]	addr[24]
A[15]/ba[0]*[4]	addr[21]	addr[21]	addr[22]	addr[22]	addr[22]	addr[22]	addr[23]	addr[23]
A[14]/ma[12]*[4]	0	0	0	0	0	0	0	0
A[13]/ma[11]*[4]	0	0	addr[21]	0	addr[21]	0	addr[22]	0
A[12]/ma[10]*[4]	addr[20]	*[5]	addr[20]	*[5]	addr[20]	*[5]	addr[21]	*[5]
A[11]/ma[9]*[4]	addr[19]	0	addr[19]	0	addr[19]	0	addr[20]	0
A[10]/ma[8]*[4]	addr[18]	0	addr[18]	0	addr[18]	0	addr[19]	addr[10]
A[9]/ma[7]*[4]	addr[17]	addr[9]	addr[17]	addr[9]	addr[17]	addr[9]	addr[18]	addr[9]
A[8]/ma[6]*[4]	addr[16]	addr[8]	addr[16]	addr[8]	addr[16]	addr[8]	addr[17]	addr[8]
A[7]/ma[5]*[4]	addr[15]	addr[7]	addr[15]	addr[7]	addr[15]	addr[7]	addr[16]	addr[7]
A[6]/ma[4]*[4]	addr[14]	addr[6]	addr[14]	addr[6]	addr[14]	addr[6]	addr[15]	addr[6]
A[5]/ma[3]*[4]	addr[13]	addr[5]	addr[13]	addr[5]	addr[13]	addr[5]	addr[14]	addr[5]
A[4]/ma[2]*[4]	addr[12]	addr[4]	addr[12]	addr[4]	addr[12]	addr[4]	addr[13]	addr[4]
A[3]/ma[1]*[4]	addr[11]	addr[3]	addr[11]	addr[3]	addr[11]	addr[3]	addr[12]	addr[3]
A[2]/ma[0]*[4]	addr[10]	addr[2]	addr[10]	addr[2]	addr[10]	addr[2]	addr[11]	addr[2]

Notes: 1. BSIZE represents the BSIZE bit in the SDCmCNT control register.
2. DSZ represents the DSZ bit in the SDRAMm address register.
3. DDBW represents the DDBW bit in the SDRAMm address register.
4. ba[1:0] and ma[12:0] represent, respectively, the SDRAM bank address and SDRAM address.
5. When the RD, WR or PRA command is issued, this carries the pre-charge option signal.

Table 10.20 Case for 32-Bit External Data Bus Width (BSIZE*[1] = (0, 1)) (2)

SDRAM Type	128 Mbits (× 16)		256 Mbits (× 32)		128 Mbits (× 8)		256 Mbits (× 16)	
Number	2		1		4		2	
DSZ*[2]	011 (32 Mbytes)		011 (32 Mbytes)		100 (64 Mbytes)		100 (64 Mbytes)	
DDBW*[3]	01 (16 bits)		10 (32 bits)		00 (8 bits)		01 (16 bits)	
Output Pin of This LSI	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address
A[16]/ba[1]*[4]	addr[24]	addr[24]	addr[24]	addr[24]	addr[25]	addr[25]	addr[25]	addr[25]
A[15]/ba[0]*[4]	addr[23]	addr[23]	addr[23]	addr[23]	addr[24]	addr[24]	addr[24]	addr[24]
A[14]/ma[12]*[4]	0	0	0	0	0	0	addr[23]	0
A[13]/ma[11]*[4]	addr[22]	0	addr[22]	0	addr[23]	0	addr[22]	0
A[12]/ma[10]*[4]	addr[21]	*[5]	addr[21]	*[5]	addr[22]	*[5]	addr[21]	*[5]
A[11]/ma[9]*[4]	addr[20]	0	addr[20]	0	addr[21]	addr[11]	addr[20]	0
A[10]/ma[8]*[4]	addr[19]	addr[10]	addr[19]	addr[10]	addr[20]	addr[10]	addr[19]	addr[10]
A[9]/ma[7]*[4]	addr[18]	addr[9]	addr[18]	addr[9]	addr[19]	addr[9]	addr[18]	addr[9]
A[8]/ma[6]*[4]	addr[17]	addr[8]	addr[17]	addr[8]	addr[18]	addr[8]	addr[17]	addr[8]
A[7]/ma[5]*[4]	addr[16]	addr[7]	addr[16]	addr[7]	addr[17]	addr[7]	addr[16]	addr[7]
A[6]/ma[4]*[4]	addr[15]	addr[6]	addr[15]	addr[6]	addr[16]	addr[6]	addr[15]	addr[6]
A[5]/ma[3]*[4]	addr[14]	addr[5]	addr[14]	addr[5]	addr[15]	addr[5]	addr[14]	addr[5]
A[4]/ma[2]*[4]	addr[13]	addr[4]	addr[13]	addr[4]	addr[14]	addr[4]	addr[13]	addr[4]
A[3]/ma[1]*[4]	addr[12]	addr[3]	addr[12]	addr[3]	addr[13]	addr[3]	addr[12]	addr[3]
A[2]/ma[0]*[4]	addr[11]	addr[2]	addr[11]	addr[2]	addr[12]	addr[2]	addr[11]	addr[2]

Notes: 1. BSIZE represents the BSIZE bit in the SDCmCNT control register.
2. DSZ represents the DSZ bit in the SDRAMm address register.
3. DDBW represents the DDBW bit in the SDRAMm address register.
4. ba[1:0] and ma[12:0] represent, respectively, the SDRAM bank address and SDRAM address.
5. When the RD, WR or PRA command is issued, this carries the pre-charge option signal.

Table 10.21 Case for 32-Bit External Data Bus Width (BSIZE*1 = (0, 1)) (3)

SDRAM Type, Number	512 Mbits (× 32), 1	
DSZ*2	100 (64 Mbytes)	
DDBW*3	10 (32 bits)	
Output Pin of This LSI	**Row Address**	**Column Address**
A[16]/ba[1]*4	addr[25]	addr[25]
A[15]/ba[0]*4	addr[24]	addr[24]
A[14]/ma[12]*4	addr[23]	0
A[13]/ma[11]*4	addr[22]	0
A[12]/ma[10]*4	addr[21]	*5
A[11]/ma[9]*4	addr[20]	0
A[10]/ma[8]*4	addr[19]	addr[10]
A[9]/ma[7]*4	addr[18]	addr[9]
A[8]/ma[6]*4	addr[17]	addr[8]
A[7]/ma[5]*4	addr[16]	addr[7]
A[6]/ma[4]*4	addr[15]	addr[6]
A[5]/ma[3]*4	addr[14]	addr[5]
A[4]/ma[2]*4	addr[13]	addr[4]
A[3]/ma[1]*4	addr[12]	addr[3]
A[2]/ma[0]*4	addr[11]	addr[2]

Notes: 1. BSIZE represents the BSIZE bit in the SDCmCNT control register.
2. DSZ represents the DSZ bit in the SDRAMm address register.
3. DDBW represents the DDBW bit in the SDRAMm address register.
4. ba[1:0] and ma[12:0] represent, respectively, the SDRAM bank address and SDRAM address.
5. When the RD, WR or PRA command is issued, this carries the pre-charge option signal.

10.6 Connection Examples

The following figures show examples of connecting SRAM or SDRAM to this LSI.

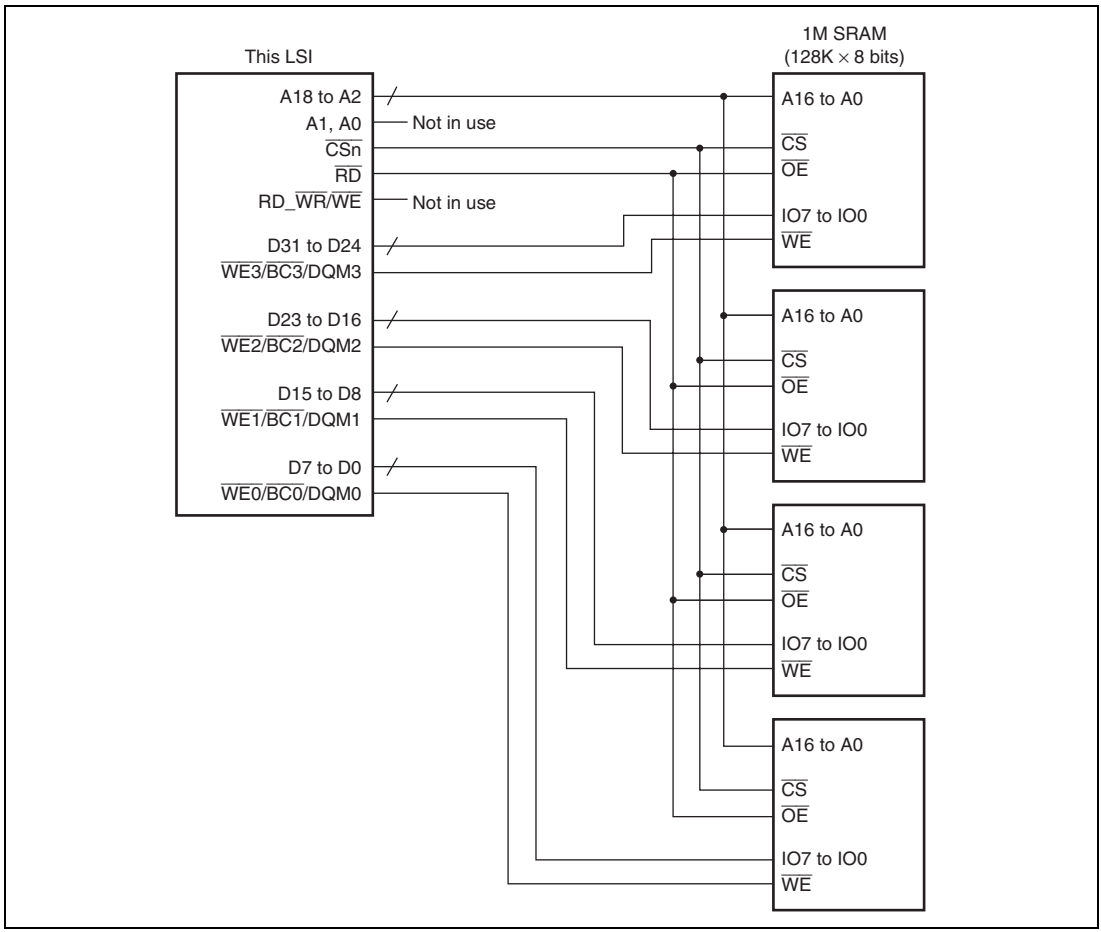


Figure 10.41 Example of Connecting a 32-Bit Data-Width SRAM

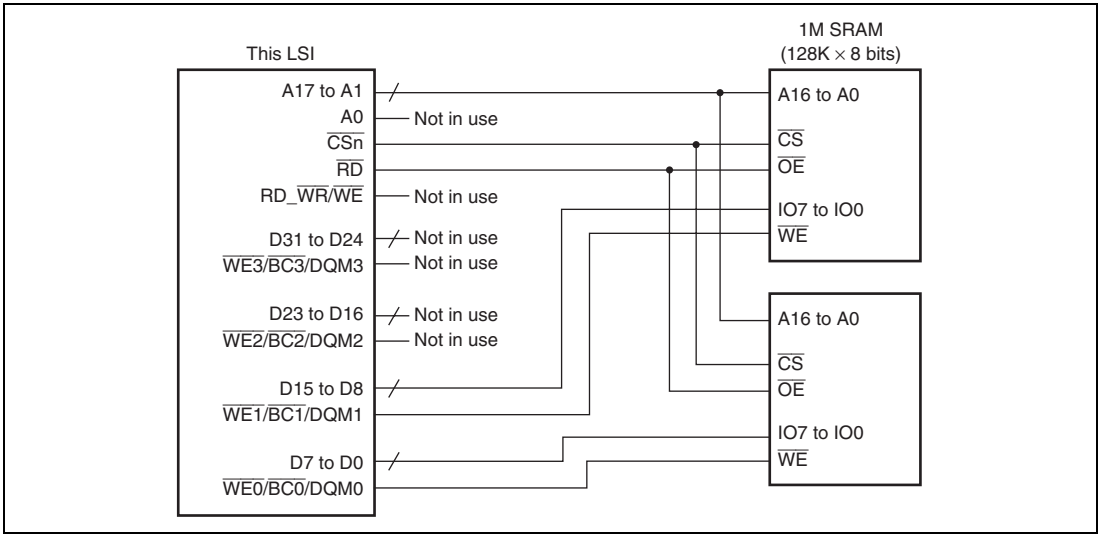


Figure 10.42 Example of Connecting a 16-Bit Data-Width SRAM

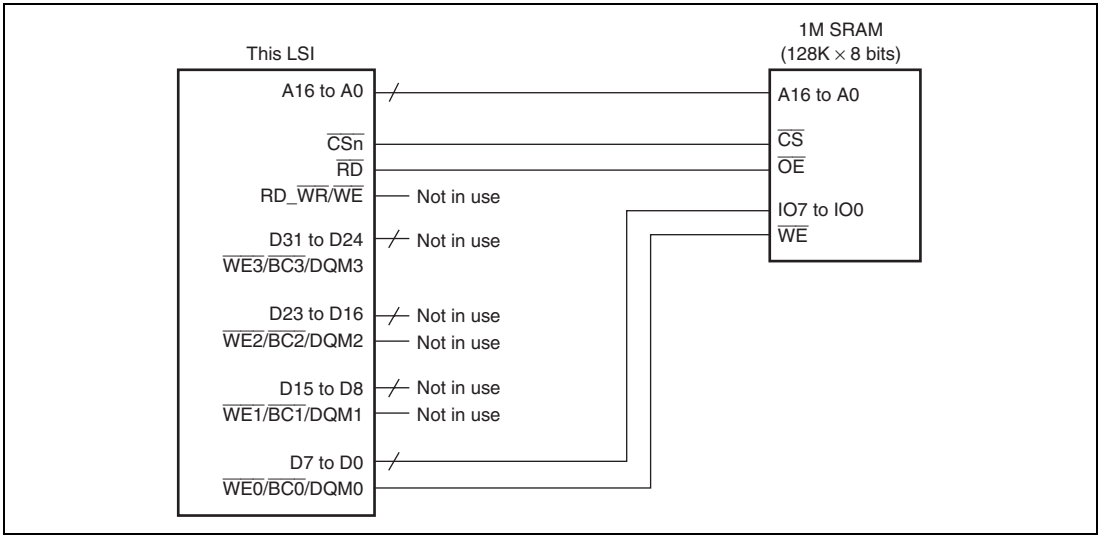


Figure 10.43 Example of Connecting an 8-Bit Data-Width SRAM

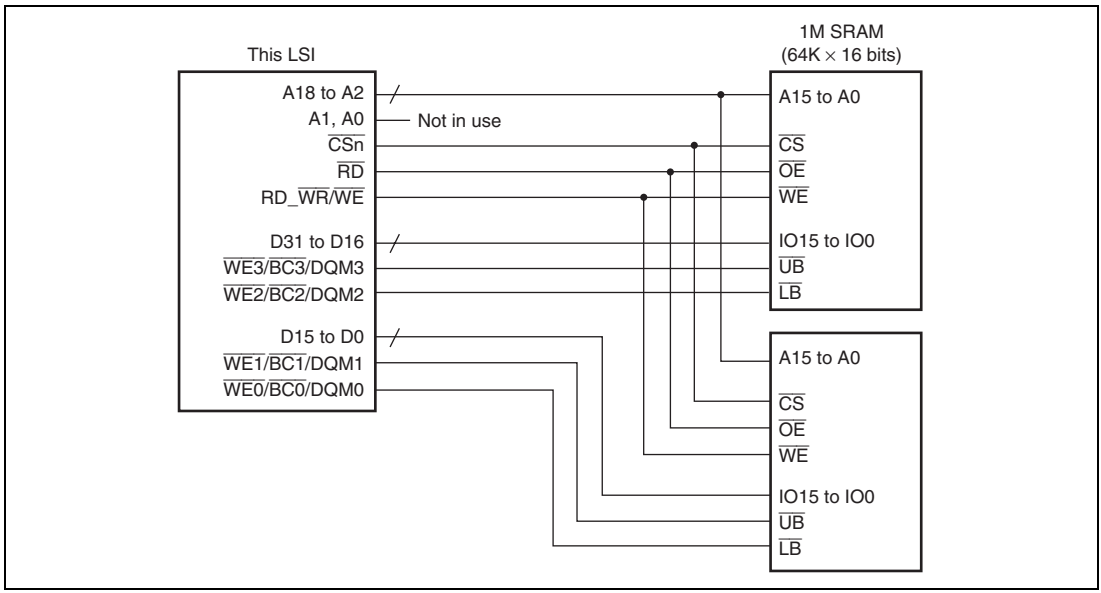


Figure 10.44 Example of Connecting a 32-Bit Data-Width SRAM (with Byte Control)

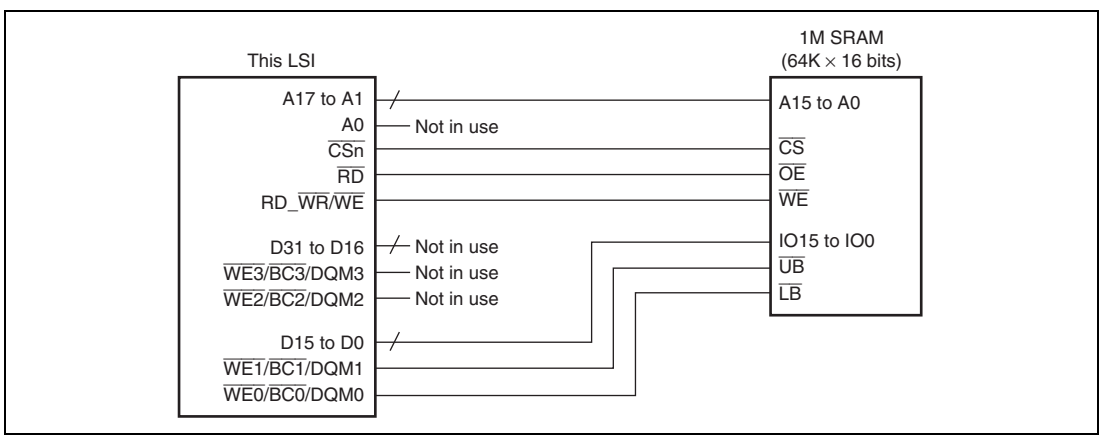


Figure 10.45 Example of Connecting a 16-Bit Data-Width SRAM (with Byte Control)

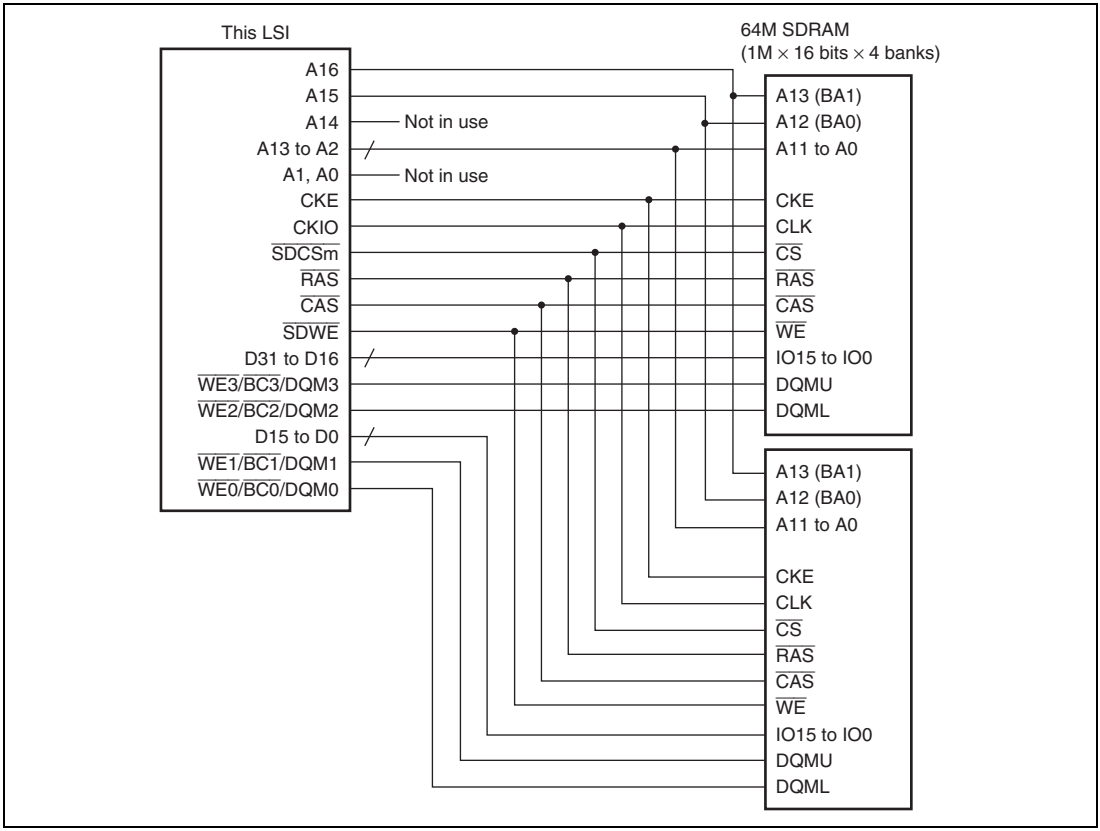


Figure 10.46 Example of Connecting a 32-Bit Data-Width SDRAM

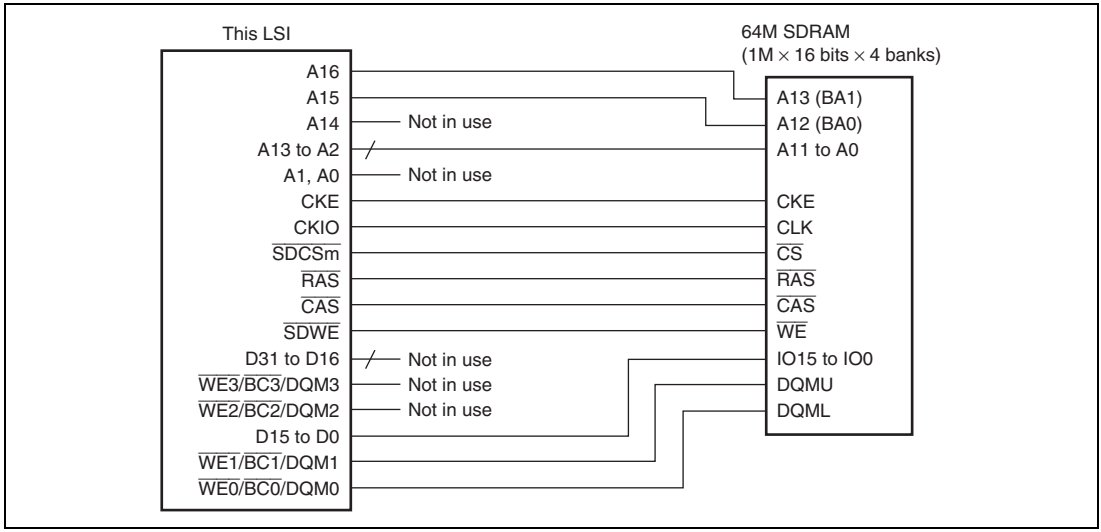


Figure 10.47 Example of Connecting a 16-Bit Data-Width SDRAM

10.7 Usage Notes

10.7.1 Write Buffer

In writing to locations in normal space or in SDRAM space, data to be written are held once in the internal write buffer of the BSC, after which writing to the device connected in external space or SDRAM space (external device) proceeds. Since writing of data from the write buffer to the external device is automatic, no processing by software is required.

At the same time, care is required on the following point. Writing by the CPU or DMAC appears to be complete at the point where the data are stored in the write buffer mentioned above. In other words, at the point where writing by the CPU or DMAC is completed, writing to the external device may in some cases not have been completed. To confirm completion of writing to the external device, execute dummy reading from the normal space or SDRAM space. Completion of dummy reading guarantees the completion of writing to the external device for previous write-access. The target device for dummy reading need not be the same as the target for writing. Furthermore, the space need not be the same.

10.7.2 Point for Caution at the Time of a Transition to Software Standby or deep Standby Mode

In cases where a transition to software standby or deep standby mode follows the execution of writing to locations in normal space or in SDRAM space, data may remain within the internal write buffer of the BSC. To ensure that data do not remain within the write buffer, execute dummy reading from an external device in the same way as described above.

Section 11 Direct Memory Access Controller (DMAC)

The DMA controller (DMAC) is a module that handles high-speed data transfer without CPU intervention in response to requests from software, on-chip peripheral modules, or external pins (external modules). The DMAC itself does not distinguish between requests from on-chip peripheral modules and those from external pins (external modules). The DMA supports data transfer between memories, between memory and on-chip peripheral modules, and between on-chip peripheral modules.

11.1 Features

- Number of channels: 14 channels (four channels can accept external requests; two-dimensional addressing supported on eight channels)
- Transfer requests: Software trigger and requests from on-chip peripheral modules (48 sources) and external pins (4 sources)
- Maximum transfer bytes: 64 Mbytes
- Address space: 4 Gbytes
- Transfer data sizes:
  - Single data transfer: 8, 16, and 32 bits
  - Single operand transfer: 1, 2, 4, 8, 16, 32, 64, and 128 data units
  - Non-stop transfer: Until the byte counter reaches "0"
- Transfer mode:
  - Cycle-stealing transfer
  - Piepelined transfer
- Maximum transfer speed
  - Cycle-stealing transfer: Minimum of three bus clock cycles per unit data transfer
  - Pipeline transfer: Minimum of one bus clock cycle per unit data transfer
- Transfer conditions

  The following transfer method can be selected.
  - Unit operand transfer: Transfers data of one operand per DMA request.
    Arbitrates channels per transfer of one operand.
    Requires request trigger per transfer of one operand.
  - Sequential operand transfer: Repeats transfer of one operand per DMA request until the byte count reaches "0".
    Arbitrates channels per transfer of one operand.
    Requires only the first request trigger.

— Non-stop transfer: Transfers data until the byte count reaches "0" per DMA request.
Does not arbitrate channels.
Requires only the first request trigger.

- Channel priorities: Channel 0 > channel 1 > ... > channel 12 > channel 13 (this priority order is fixed)
- Interrupt request: Two types of interrupt requests (generated when the byte counter reaches "0")
  — Interrupt request signal for each channel
  — Interrupt request signal common to multiple channels
- Reload function: Reloads the source address, destination address, and byte count.
- Rotate function: The address rotate function can be set.
- Two-dimensional addressing: This can be specified in channels 0 to 7.
- The DMAC suspend/restart/stop function can be set.

Note: Terminologies in this section are defined as follows:
Single data transfer: Transfer in one read cycle or one write cycle by the DMAC
Single operand transfer: Continuous data transfer by the DMAC on one channel (amount of data to be transferred is set in a register)
Single DMA transfer: Transfer of data by the number of data set in the byte count register from the start address to the end address
Channel number: n = 0 to 13
Two-dimensional addressing-supported channel number: m = 0 to 7
Request source number: k = 0 to 52

Figure 11.1 is a block diagram of the DMAC.

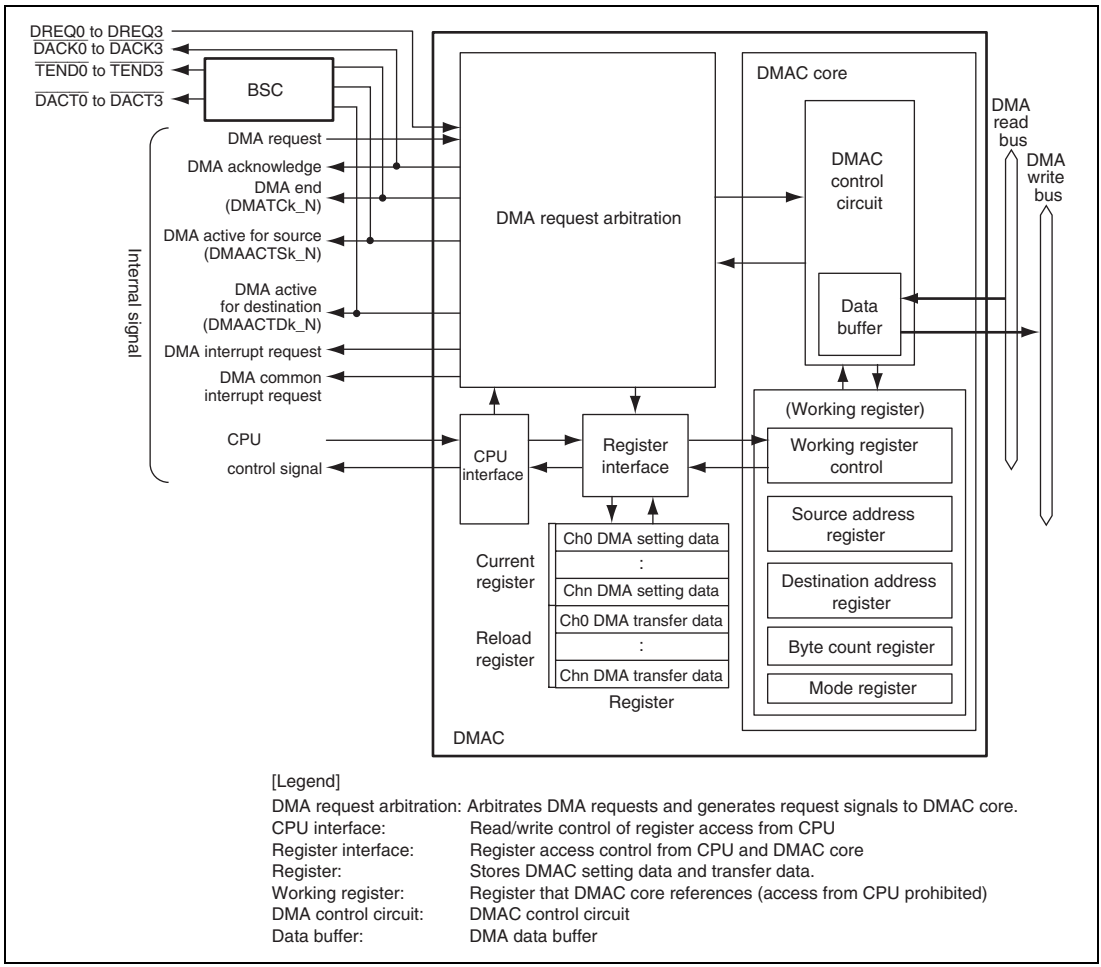


Figure 11.1 DMAC Block Diagram

11.2 Input/Output Pins

Table 11.1 shows DMAC pin functions.

Table 11.1 Pin Configuration

Pin Name	I/O	Function
DREQ0 to DREQ3	Input	External request for DMA transfer
$\overline{\text{DACK0}}$ to $\overline{\text{DACK3}}$	Output	DMA acknowledgment output signal (active low) for external request of DMA transfer These signals are output when an external request of DMA transfer is accepted.
$\overline{\text{DACT0}}$ to $\overline{\text{DACT3}}$	Output	DMA active output signal (active low) for external request of DMA transfer These signals are output during a normal DMA space access.
$\overline{\text{TEND0}}$ to $\overline{\text{TEND3}}$	Output	DMA end output signal (low at the end of DMA transfer) for external request of DMA transfer These signals are output during the last DMA access to normal space in cycle-stealing mode.

11.3 Register Descriptions

The DMAC has the registers shown in tables 11.2 and 11.3. All these registers are initialized by a power-on reset or in deep standby mode and the previous settings are lost.

Table 11.2 Register Configuration (Registers Not Related to Two-Dimensional Addressing)

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	DMA current source address register 0	DMCSADR0	R/W	Undefined	H'FF460000	32
	DMA current destination address register 0	DMCDADR0	R/W	Undefined	H'FF460004	32
	DMA current byte count register 0	DMCBCT0	R/W	Undefined	H'FF460008	32
	DMA mode register 0	DMMOD0	R/W	Undefined	H'FF46000C	32
	DMA reload source address register 0	DMRSADR0	R/W	Undefined	H'FF460200	32
	DMA reload destination address register 0	DMRDADR0	R/W	Undefined	H'FF460204	32
	DMA reload byte count register 0	DMRBCT0	R/W	Undefined	H'FF460208	32
	DMA control register A0	DMACNTA0	R/W	H'00000000	H'FF460400	8, 16, 32
	DMA control register B0	DMACNTB0	R/W	H'00000000	H'FF460404	8, 16, 32
1	DMA current source address register 1	DMCSADR1	R/W	Undefined	H'FF460010	32
	DMA current destination address register 1	DMCDADR1	R/W	Undefined	H'FF460014	32
	DMA current byte count register 1	DMCBCT1	R/W	Undefined	H'FF460018	32
	DMA mode register 1	DMMOD1	R/W	Undefined	H'FF46001C	32
	DMA reload source address register 1	DMRSADR1	R/W	Undefined	H'FF460210	32
	DMA reload destination address register 1	DMRDADR1	R/W	Undefined	H'FF460214	32
	DMA reload byte count register 1	DMRBCT1	R/W	Undefined	H'FF460218	32
	DMA control register A1	DMACNTA1	R/W	H'00000000	H'FF460408	8, 16, 32
	DMA control register B1	DMACNTB1	R/W	H'00000000	H'FF46040C	8, 16, 32
2	DMA current source address register 2	DMCSADR2	R/W	Undefined	H'FF460020	32
	DMA current destination address register 2	DMCDADR2	R/W	Undefined	H'FF460024	32
	DMA current byte count register 2	DMCBCT2	R/W	Undefined	H'FF460028	32
	DMA mode register 2	DMMOD2	R/W	Undefined	H'FF46002C	32
	DMA reload source address register 2	DMRSADR2	R/W	Undefined	H'FF460220	32
	DMA reload destination address register 2	DMRDADR2	R/W	Undefined	H'FF460224	32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2	DMA reload byte count register 2	DMRBCT2	R/W	Undefined	H'FF460228	32
	DMA control register A2	DMACNTA2	R/W	H'00000000	H'FF460410	8, 16, 32
	DMA control register B2	DMACNTB2	R/W	H'00000000	H'FF460414	8, 16, 32
3	DMA current source address register 3	DMCSADR3	R/W	Undefined	H'FF460030	32
	DMA current destination address register 3	DMCDADR3	R/W	Undefined	H'FF460034	32
	DMA current byte count register 3	DMCBCT3	R/W	Undefined	H'FF460038	32
	DMA mode register 3	DMMOD3	R/W	Undefined	H'FF46003C	32
	DMA reload source address register 3	DMRSADR3	R/W	Undefined	H'FF460230	32
	DMA reload destination address register 3	DMRDADR3	R/W	Undefined	H'FF460234	32
	DMA reload byte count register 3	DMRBCT3	R/W	Undefined	H'FF460238	32
	DMA control register A3	DMACNTA3	R/W	H'00000000	H'FF460418	8, 16, 32
	DMA control register B3	DMACNTB3	R/W	H'00000000	H'FF46041C	8, 16, 32
4	DMA current source address register 4	DMCSADR4	R/W	Undefined	H'FF460040	32
	DMA current destination address register 4	DMCDADR4	R/W	Undefined	H'FF460044	32
	DMA current byte count register 4	DMCBCT4	R/W	Undefined	H'FF460048	32
	DMA mode register 4	DMMOD4	R/W	Undefined	H'FF46004C	32
	DMA reload source address register 4	DMRSADR4	R/W	Undefined	H'FF460240	32
	DMA reload destination address register 4	DMRDADR4	R/W	Undefined	H'FF460244	32
	DMA reload byte count register 4	DMRBCT4	R/W	Undefined	H'FF460248	32
	DMA control register A4	DMACNTA4	R/W	H'00000000	H'FF460420	8, 16, 32
	DMA control register B4	DMACNTB4	R/W	H'00000000	H'FF460424	8, 16, 32
5	DMA current source address register 5	DMCSADR5	R/W	Undefined	H'FF460050	32
	DMA current destination address register 5	DMCDADR5	R/W	Undefined	H'FF460054	32
	DMA current byte count register 5	DMCBCT5	R/W	Undefined	H'FF460058	32
	DMA mode register 5	DMMOD5	R/W	Undefined	H'FF46005C	32
	DMA reload source address register 5	DMRSADR5	R/W	Undefined	H'FF460250	32
	DMA reload destination address register 5	DMRDADR5	R/W	Undefined	H'FF460254	32
	DMA reload byte count register 5	DMRBCT5	R/W	Undefined	H'FF460258	32
	DMA control register A5	DMACNTA5	R/W	H'00000000	H'FF460428	8, 16, 32
	DMA control register B5	DMACNTB5	R/W	H'00000000	H'FF46042C	8, 16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
6	DMA current source address register 6	DMCSADR6	R/W	Undefined	H'FF460060	32
	DMA current destination address register 6	DMCDADR6	R/W	Undefined	H'FF460064	32
	DMA current byte count register 6	DMCBCT6	R/W	Undefined	H'FF460068	32
	DMA mode register 6	DMMOD6	R/W	Undefined	H'FF46006C	32
	DMA reload source address register 6	DMRSADR6	R/W	Undefined	H'FF460260	32
	DMA reload destination address register 6	DMRDADR6	R/W	Undefined	H'FF460264	32
	DMA reload byte count register 6	DMRBCT6	R/W	Undefined	H'FF460268	32
	DMA control register A6	DMACNTA6	R/W	H'00000000	H'FF460430	8, 16, 32
	DMA control register B6	DMACNTB6	R/W	H'00000000	H'FF460434	8, 16, 32
7	DMA current source address register 7	DMCSADR7	R/W	Undefined	H'FF460070	32
	DMA current destination address register 7	DMCDADR7	R/W	Undefined	H'FF460074	32
	DMA current byte count register 7	DMCBCT7	R/W	Undefined	H'FF460078	32
	DMA mode register 7	DMMOD7	R/W	Undefined	H'FF46007C	32
	DMA reload source address register 7	DMRSADR7	R/W	Undefined	H'FF460270	32
	DMA reload destination address register 7	DMRDADR7	R/W	Undefined	H'FF460274	32
	DMA reload byte count register 7	DMRBCT7	R/W	Undefined	H'FF460278	32
	DMA control register A7	DMACNTA7	R/W	H'00000000	H'FF460438	8, 16, 32
	DMA control register B7	DMACNTB7	R/W	H'00000000	H'FF46043C	8, 16, 32
8	DMA current source address register 8	DMCSADR8	R/W	Undefined	H'FF460080	32
	DMA current destination address register 8	DMCDADR8	R/W	Undefined	H'FF460084	32
	DMA current byte count register 8	DMCBCT8	R/W	Undefined	H'FF460088	32
	DMA mode register 8	DMMOD8	R/W	Undefined	H'FF46008C	32
	DMA reload source address register 8	DMRSADR8	R/W	Undefined	H'FF460280	32
	DMA reload destination address register 8	DMRDADR8	R/W	Undefined	H'FF460284	32
	DMA reload byte count register 8	DMRBCT8	R/W	Undefined	H'FF460288	32
	DMA control register A8	DMACNTA8	R/W	H'00000000	H'FF460440	8, 16, 32
	DMA control register B8	DMACNTB8	R/W	H'00000000	H'FF460444	8, 16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
9	DMA current source address register 9	DMCSADR9	R/W	Undefined	H'FF460090	32
	DMA current destination address register 9	DMCDADR9	R/W	Undefined	H'FF460094	32
	DMA current byte count register 9	DMCBCT9	R/W	Undefined	H'FF460098	32
	DMA mode register 9	DMMOD9	R/W	Undefined	H'FF46009C	32
	DMA reload source address register 9	DMRSADR9	R/W	Undefined	H'FF460290	32
	DMA reload destination address register 9	DMRDADR9	R/W	Undefined	H'FF460294	32
	DMA reload byte count register 9	DMRBCT9	R/W	Undefined	H'FF460298	32
	DMA control register A9	DMACNTA9	R/W	H'00000000	H'FF460448	8, 16, 32
	DMA control register B9	DMACNTB9	R/W	H'00000000	H'FF46044C	8, 16, 32
10	DMA current source address register 10	DMCSADR10	R/W	Undefined	H'FF4600A0	32
	DMA current destination address register 10	DMCDADR10	R/W	Undefined	H'FF4600A4	32
	DMA current byte count register 10	DMCBCT10	R/W	Undefined	H'FF4600A8	32
	DMA mode register 10	DMMOD10	R/W	Undefined	H'FF4600AC	32
	DMA reload source address register 10	DMRSADR10	R/W	Undefined	H'FF4602A0	32
	DMA reload destination address register 10	DMRDADR10	R/W	Undefined	H'FF4602A4	32
	DMA reload byte count register 10	DMRBCT10	R/W	Undefined	H'FF4602A8	32
	DMA control register A10	DMACNTA10	R/W	H'00000000	H'FF460450	8, 16, 32
	DMA control register B10	DMACNTB10	R/W	H'00000000	H'FF460454	8, 16, 32
11	DMA current source address register 11	DMCSADR11	R/W	Undefined	H'FF4600B0	32
	DMA current destination address register 11	DMCDADR11	R/W	Undefined	H'FF4600B4	32
	DMA current byte count register 11	DMCBCT11	R/W	Undefined	H'FF4600B8	32
	DMA mode register 11	DMMOD11	R/W	Undefined	H'FF4600BC	32
	DMA reload source address register 11	DMRSADR11	R/W	Undefined	H'FF4602B0	32
	DMA reload destination address register 11	DMRDADR11	R/W	Undefined	H'FF4602B4	32
	DMA reload byte count register 11	DMRBCT11	R/W	Undefined	H'FF4602B8	32
	DMA control register A11	DMACNTA11	R/W	H'00000000	H'FF460458	8, 16, 32
	DMA control register B11	DMACNTB11	R/W	H'00000000	H'FF46045C	8, 16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
12	DMA current source address register 12	DMCSADR12	R/W	Undefined	H'FF4600C0	32
	DMA current destination address register 12	DMCDADR12	R/W	Undefined	H'FF4600C4	32
	DMA current byte count register 12	DMCBCT12	R/W	Undefined	H'FF4600C8	32
	DMA mode register 12	DMMOD12	R/W	Undefined	H'FF4600CC	32
	DMA reload source address register 12	DMRSADR12	R/W	Undefined	H'FF4602C0	32
	DMA reload destination address register 12	DMRDADR12	R/W	Undefined	H'FF4602C4	32
	DMA reload byte count register 12	DMRBCT12	R/W	Undefined	H'FF4602C8	32
	DMA control register A12	DMACNTA12	R/W	H'00000000	H'FF460460	8, 16, 32
	DMA control register B12	DMACNTB12	R/W	H'00000000	H'FF460464	8, 16, 32
13	DMA current source address register 13	DMCSADR13	R/W	Undefined	H'FF4600D0	32
	DMA current destination address register 13	DMCDADR13	R/W	Undefined	H'FF4600D4	32
	DMA current byte count register 13	DMCBCT13	R/W	Undefined	H'FF4600D8	32
	DMA mode register 13	DMMOD13	R/W	Undefined	H'FF4600DC	32
	DMA reload source address register 13	DMRSADR13	R/W	Undefined	H'FF4602D0	32
	DMA reload destination address register 13	DMRDADR13	R/W	Undefined	H'FF4602D4	32
	DMA reload byte count register 13	DMRBCT13	R/W	Undefined	H'FF4602D8	32
	DMA control register A13	DMACNTA13	R/W	H'00000000	H'FF460468	8, 16, 32
	DMA control register B13	DMACNTB13	R/W	H'00000000	H'FF46046C	8, 16, 32
Common	DMA activation control register	DMSCNT	R/W	H'00000000	H'FF460500	8, 16, 32
	DMA interrupt control register	DMICNT	R/W	H'00000000	H'FF460508	8, 16, 32
	DMA common interrupt control register	DMICNTA	R/W	H'00000000	H'FF46050C	8, 16, 32
	DMA interrupt status register	DMISTS	R	H'00000000	H'FF460510	8, 16, 32
	DMA transfer end detection register	DMEDET	R/W	H'00000000	H'FF460514	8, 16, 32
	DMA arbitration status register	DMASTS	R/W	H'00000000	H'FF460518	8, 16, 32

Table 11.3 Register Configuration (Registers Related to Two-Dimensional Addressing)

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	DMA two-dimensional addressing column setting register 0	DM2DCLM0	R/W	Undefined	H'FF460600	32
	DMA two-dimensional addressing row setting register 0	DM2DROW0	R/W	Undefined	H'FF460604	32
	DMA two-dimensional addressing block setting register 0	DM2DBLK0	R/W	Undefined	H'FF460608	32
	DMA two-dimensional addressing next row offset register 0	DM2DNROST0	R/W	Undefined	H'FF46060C	32
	DMA two-dimensional addressing next block offset register 0	DM2DNBOST0	R/W	Undefined	H'FF460610	32
	DMA two-dimensional addressing next line offset register 0	DM2DNLOST0	R/W	Undefined	H'FF460614	32
	DMA reload two-dimensional addressing column setting register 0	DMR2DCLM0	R/W	Undefined	H'FF460A00	32
	DMA reload two-dimensional addressing row setting register 0	DMR2DROW0	R/W	Undefined	H'FF460A04	32
	DMA reload two-dimensional addressing block setting register 0	DMR2DBLK0	R/W	Undefined	H'FF460A08	32
	DMA reload two-dimensional addressing next row offset register 0	DMR2DNROST0	R/W	Undefined	H'FF460A0C	32
	DMA reload two-dimensional addressing next block offset register 0	DMR2DNBOST0	R/W	Undefined	H'FF460A10	32
	DMA reload two-dimensional addressing next line offset register 0	DMR2DNLOST0	R/W	Undefined	H'FF460A14	32
1	DMA two-dimensional addressing column setting register 1	DM2DCLM1	R/W	Undefined	H'FF460620	32
	DMA two-dimensional addressing row setting register 1	DM2DROW1	R/W	Undefined	H'FF460624	32
	DMA two-dimensional addressing block setting register 1	DM2DBLK1	R/W	Undefined	H'FF460628	32
	DMA two-dimensional addressing next row offset register 1	DM2DNROST1	R/W	Undefined	H'FF46062C	32
	DMA two-dimensional addressing next block offset register 1	DM2DNBOST1	R/W	Undefined	H'FF460630	32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
1	DMA two-dimensional addressing next line offset register 1	DM2DNLOST1	R/W	Undefined	H'FF460634	32
	DMA reload two-dimensional addressing column setting register 1	DMR2DCLM1	R/W	Undefined	H'FF460A20	32
	DMA reload two-dimensional addressing row setting register 1	DMR2DROW1	R/W	Undefined	H'FF460A24	32
	DMA reload two-dimensional addressing block setting register 1	DMR2DBLK1	R/W	Undefined	H'FF460A28	32
	DMA reload two-dimensional addressing next row offset register 1	DMR2DNROST1	R/W	Undefined	H'FF460A2C	32
	DMA reload two-dimensional addressing next block offset register 1	DMR2DNBOST1	R/W	Undefined	H'FF460A30	32
	DMA reload two-dimensional addressing next line offset register 1	DMR2DNLOST1	R/W	Undefined	H'FF460A34	32
2	DMA two-dimensional addressing column setting register 2	DM2DCLM2	R/W	Undefined	H'FF460640	32
	DMA two-dimensional addressing row setting register 2	DM2DROW2	R/W	Undefined	H'FF460644	32
	DMA two-dimensional addressing block setting register 2	DM2DBLK2	R/W	Undefined	H'FF460648	32
	DMA two-dimensional addressing next row offset register 2	DM2DNROST2	R/W	Undefined	H'FF46064C	32
	DMA two-dimensional addressing next block offset register 2	DM2DNBOST2	R/W	Undefined	H'FF460650	32
	DMA two-dimensional addressing next line offset register 2	DM2DNLOST2	R/W	Undefined	H'FF460654	32
	DMA reload two-dimensional addressing column setting register 2	DMR2DCLM2	R/W	Undefined	H'FF460A40	32
	DMA reload two-dimensional addressing row setting register 2	DMR2DROW2	R/W	Undefined	H'FF460A44	32
	DMA reload two-dimensional addressing block setting register 2	DMR2DBLK2	R/W	Undefined	H'FF460A48	32
	DMA reload two-dimensional addressing next row offset register 2	DMR2DNROST2	R/W	Undefined	H'FF460A4C	32
	DMA reload two-dimensional addressing next block offset register 2	DMR2DNBOST2	R/W	Undefined	H'FF460A50	32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2	DMA reload two-dimensional addressing next line offset register 2	DMR2DNLOST2	R/W	Undefined	H'FF460A54	32
3	DMA two-dimensional addressing column setting register 3	DM2DCLM3	R/W	Undefined	H'FF460660	32
	DMA two-dimensional addressing row setting register 3	DM2DROW3	R/W	Undefined	H'FF460664	32
	DMA two-dimensional addressing block setting register 3	DM2DBLK3	R/W	Undefined	H'FF460668	32
	DMA two-dimensional addressing next row offset register 3	DM2DNROST3	R/W	Undefined	H'FF46066C	32
	DMA two-dimensional addressing next block offset register 3	DM2DNBOST3	R/W	Undefined	H'FF460670	32
	DMA two-dimensional addressing next line offset register 3	DM2DNLOST3	R/W	Undefined	H'FF460674	32
	DMA reload two-dimensional addressing column setting register 3	DMR2DCLM3	R/W	Undefined	H'FF460A60	32
	DMA reload two-dimensional addressing row setting register 3	DMR2DROW3	R/W	Undefined	H'FF460A64	32
	DMA reload two-dimensional addressing block setting register 3	DMR2DBLK3	R/W	Undefined	H'FF460A68	32
	DMA reload two-dimensional addressing next row offset register 3	DMR2DNROST3	R/W	Undefined	H'FF460A6C	32
	DMA reload two-dimensional addressing next block offset register 3	DMR2DNBOST3	R/W	Undefined	H'FF460A70	32
	DMA reload two-dimensional addressing next line offset register 3	DMR2DNLOST3	R/W	Undefined	H'FF460A74	32
4	DMA two-dimensional addressing column setting register 4	DM2DCLM4	R/W	Undefined	H'FF460680	32
	DMA two-dimensional addressing row setting register 4	DM2DROW4	R/W	Undefined	H'FF460684	32
	DMA two-dimensional addressing block setting register 4	DM2DBLK4	R/W	Undefined	H'FF460688	32
	DMA two-dimensional addressing next row offset register 4	DM2DNROST4	R/W	Undefined	H'FF46068C	32
	DMA two-dimensional addressing next block offset register 4	DM2DNBOST4	R/W	Undefined	H'FF460690	32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
4	DMA two-dimensional addressing next line offset register 4	DM2DNLOST4	R/W	Undefined	H'FF460694	32
	DMA reload two-dimensional addressing column setting register 4	DMR2DCLM4	R/W	Undefined	H'FF460A80	32
	DMA reload two-dimensional addressing row setting register 4	DMR2DROW4	R/W	Undefined	H'FF460A84	32
	DMA reload two-dimensional block setting register 4	DMR2DBLK4	R/W	Undefined	H'FF460A88	32
	DMA reload two-dimensional addressing next row offset register 4	DMR2DNROST4	R/W	Undefined	H'FF460A8C	32
	DMA reload two-dimensional addressing next block offset register 4	DMR2DNBOST4	R/W	Undefined	H'FF460A90	32
	DMA reload two-dimensional addressing next line offset register 4	DMR2DNLOST4	R/W	Undefined	H'FF460A94	32
5	DMA two-dimensional addressing column setting register 5	DM2DCLM5	R/W	Undefined	H'FF4606A0	32
	DMA two-dimensional addressing row setting register 5	DM2DROW5	R/W	Undefined	H'FF4606A4	32
	DMA two-dimensional addressing block setting register 5	DM2DBLK5	R/W	Undefined	H'FF4606A8	32
	DMA two-dimensional addressing next row offset register 5	DM2DNROST5	R/W	Undefined	H'FF4606AC	32
	DMA two-dimensional addressing next block offset register 5	DM2DNBOST5	R/W	Undefined	H'FF4606B0	32
	DMA two-dimensional addressing next line offset register 5	DM2DNLOST5	R/W	Undefined	H'FF4606B4	32
	DMA reload two-dimensional addressing column setting register 5	DMR2DCLM5	R/W	Undefined	H'FF460AA0	32
	DMA reload two-dimensional addressing row setting register 5	DMR2DROW5	R/W	Undefined	H'FF460AA4	32
	DMA reload two-dimensional block setting register 5	DMR2DBLK5	R/W	Undefined	H'FF460AA8	32
	DMA reload two-dimensional addressing next row offset register 5	DMR2DNROST5	R/W	Undefined	H'FF460AAC	32
	DMA reload two-dimensional addressing next block offset register 5	DMR2DNBOST5	R/W	Undefined	H'FF460AB0	32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
5	DMA reload two-dimensional addressing next line offset register 5	DMR2DNLOST5	R/W	Undefined	H'FF460AB4	32
6	DMA two-dimensional addressing column setting register 6	DM2DCLM6	R/W	Undefined	H'FF4606C0	32
	DMA two-dimensional addressing row setting register 6	DM2DROW6	R/W	Undefined	H'FF4606C4	32
	DMA two-dimensional addressing block setting register 6	DM2DBLK6	R/W	Undefined	H'FF4606C8	32
	DMA reload two-dimensional addressing next row offset register 6	DM2DNROST6	R/W	Undefined	H'FF4606CC	32
	DMA two-dimensional addressing next block offset register 6	DM2DNBOST6	R/W	Undefined	H'FF4606D0	32
	DMA two-dimensional addressing next line offset register 6	DM2DNLOST6	R/W	Undefined	H'FF4606D4	32
	DMA reload two-dimensional addressing column setting register 6	DMR2DCLM6	R/W	Undefined	H'FF460AC0	32
	DMA reload two-dimensional addressing row setting register 6	DMR2DROW6	R/W	Undefined	H'FF460AC4	32
	DMA reload two-dimensional block setting register 6	DMR2DBLK6	R/W	Undefined	H'FF460AC8	32
	DMA reload two-dimensional addressing next row offset register 6	DMR2DNROST6	R/W	Undefined	H'FF460ACC	32
	DMA reload two-dimensional addressing next block offset register 6	DMR2DNBOST6	R/W	Undefined	H'FF460AD0	32
	DMA reload two-dimensional addressing next line offset register 6	DMR2DNLOST6	R/W	Undefined	H'FF460AD4	32
7	DMA two-dimensional addressing column setting register 7	DM2DCLM7	R/W	Undefined	H'FF4606E0	32
	DMA two-dimensional addressing row setting register 7	DM2DROW7	R/W	Undefined	H'FF4606E4	32
	DMA two-dimensional addressing block setting register 7	DM2DBLK7	R/W	Undefined	H'FF4606E8	32
	DMA two-dimensional addressing row setting register 7	DM2DNROST7	R/W	Undefined	H'FF4606EC	32
	DMA two-dimensional addressing next block offset register 7	DM2DNBOST7	R/W	Undefined	H'FF4606F0	32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
7	DMA two-dimensional addressing next line offset register 7	DM2DNLOST7	R/W	Undefined	H'FF4606F4	32
	DMA reload two-dimensional addressing column setting register 7	DMR2DCLM7	R/W	Undefined	H'FF460AE0	32
	DMA reload two-dimensional addressing row setting register 7	DMR2DROW7	R/W	Undefined	H'FF460AE4	32
	DMA reload two-dimensional addressing block setting register 7	DMR2DBLK7	R/W	Undefined	H'FF460AE8	32
	DMA reload two-dimensional addressing next row offset register 7	DMR2DNROST7	R/W	Undefined	H'FF460AEC	32
	DMA reload two-dimensional addressing next block offset register 7	DMR2DNBOST7	R/W	Undefined	H'FF460AF0	32
	DMA reload two-dimensional addressing next line offset register 7	DMR2DNLOST7	R/W	Undefined	H'FF460AF4	32

11.3.1 DMA Current Source Address Registers (DMCSADRn)

DMCSADRn is a register used to specify the start address of the transfer source. The value in this register is transferred to the working source-address register when DMA transfer starts. The contents of the working source-address register are returned to this register when an operand transfer is completed. If the rotate setting (SAMOD = 011) is made for the source address, however, the contents of the working source-address register are not returned. If the source-address reload function is enabled, the contents stored in the DMA reload source address register (DMRSADRn) are returned to this register when DMA transfer is completed. This register must be set regardless of whether the reload function is enabled or disabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSA[31:16]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSA[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CSA[31:0]	Undefined	R/W	Holds source address bits A31 to A0.

Notes: 1. Set this register so that DMA transfer is performed for the following selected transfer data sizes within the correctly arranged address boundaries:
- When the transfer data size is set to 16 bits (SZSEL = 001): (bit 0) = 0
- When the transfer data size is set to 32 bits (SZSEL = 010): (bit 1, bit 0) = (0, 0)

2. Only write data to this register when the corresponding channel is not undergoing single operand transfer (the DASTS bit of the corresponding channel in the DMA arbitration status register (DMASTS) is 0) and DMA transfer is disabled (the DMST bit of the DMA activation control register (DMSCNT) is 0 or the DEN bit of DMA control register B (DMCNTBn) is 0). In other cases, operation is not guaranteed when data is written to this register.

11.3.2 DMA Current Destination Address Registers (DMCDADRn)

DMCDADRn is a register used to specify the start address of the transfer destination. The value in this register is transferred to the working destination-address register when DMA transfer is started. The contents of the working destination-address register are returned to this register when an operand transfer is completed. If the rotate setting (DAMOD = 011) is made for the destination address, however, the contents of the working destination-address register are not returned. If the destination-address reload function is enabled, the contents stored in the DMA reload destination address register (DMRDADRn) are returned to this register when DMA transfer is completed. This register must be set regardless of whether the reload function is enabled or disabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDA[31:16]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDA[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CDA[31:0]	Undefined	R/W	Holds destination address bits A31 to A0.

Notes: 1. Set this register so that DMA transfer is performed for the following selected transfer data sizes within the correctly arranged address boundaries:
- When the transfer data size is set to 16 bits (SZSEL = 001): (bit 0) = 0
- When the transfer data size is set to 32 bits (SZSEL = 010): (bit 1, bit 0) = (0, 0)

2. Only write data to this register when the corresponding channel is not undergoing single operand transfer (the DASTS bit of the corresponding channel in the DMA arbitration status register (DMASTS) is 0) and DMA transfer is disabled (the DMST bit of the DMA activation control register (DMSCNT) is 0 or the DEN bit of DMA control register B (DMCNTBn) is 0). In other cases, operation is not guaranteed when data is written to this register.

11.3.3 DMA Current Byte Count Register (DMCBCTn)

DMCBCTn is a register used to specify the number of bytes to be transferred by DMA. The value in this register is transferred to the working byte-count register when DMA transfer is started and decremented the number of bytes to be transferred per single data transfer. How much this value is decremented depends on the transfer data size as follows:

- When the transfer data size is set to 8 bits (SZSEL = 000): -1
- When the transfer data size is set to 16 bits (SZSEL = 001): -2
- When the transfer data size is set to 32 bits (SZSEL = 010): -4

When the value in the working byte count register reaches H'000 0000, DMA transfer ends (transfer end when the byte count reaches "0"). The corresponding bit of the DMA transfer end detection register (DMEDET) is set to 1. If the byte count reload function is disabled, the contents of the working byte count register are returned to this register when the channel for DMA transfer switches or DMA transfer ends. If the byte count reload function is enabled, the contents of the DMA reload byte count register (DMRBCTn) are returned to this register when DMA transfer is completed. This register must be set regardless of whether the reload function is enabled or disabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	CBC[25:16]									
Initial value:	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CBC[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 0	CBC[25:0]	Undefined	R/W	Number of bytes to be transferred by DMA

Notes: 1. Note that when the value in this register is H'000 0000, 64M bytes (maximum number of bytes to be transferred) are transferred.

2. Set this register so that the byte count becomes 0 as follows when the final data is sent in a DMA transfer:
   - When the transfer data size is set to 16 bits (SZSEL = 001): (bit 0) = 0
   - When the transfer data size is set to 32 bits (SZSEL = 010): (bit 1, bit 0) = (0, 0)
3. Only write data to this register when the corresponding channel is not undergoing single operand transfer (the DASTS bit of the corresponding channel in the DMA arbitration status register (DMASTS) is 0) and DMA transfer is disabled (the DMST bit of the DMA activation control register (DMSCNT) is 0 or the DEN bit of DMA control register B (DMCNTBn) is 0). In other cases, operation is not guaranteed when data is written to this register.

11.3.4 DMA Reload Source Address Register (DMRSADRn)

DMRSADRn is a register used to set an address to be reloaded to the DMA current source address register (DMCSADRn).

To enable the reload function, set the DMA source address reload function enable bit (SRLOD) in DMA control register A (DMCNTAn) to 1. In this case, set both the DMA current source address register (DMCSADRn) and DMA reload source address register (DMRSADRn).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSA[31:16]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSA[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RSA[31:0]	Undefined	R/W	Holds reload source address bits A31 to A0.

Note: Set this register so that DMA transfer is performed for the following selected transfer data sizes within the correctly arranged address boundaries:
- When the transfer data size is set to 16 bits (SZSEL = 001): (bit 0) = 0
- When the transfer data size is set to 32 bits (SZSEL = 010): (bit 1, bit 0) = (0, 0)

11.3.5 DMA Reload Destination Address Register (DMRDADRn)

DMRDADRn is a register used to set an address to be reloaded to the DMA current destination address register (DMCDADRn).

To enable the reload function, set the DMA destination address reload function enable bit (DRLOD) in DMA control register A (DMCNTAn) to 1. In this case, set both the DMA current destination address register (DMCDADRn) and DMA reload destination address register (DMRDADRn).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDA[31:16]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDA[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDA[31:0]	Undefined	R/W	Holds reload destination address bits A31 to A0.

Note: Set this register so that DMA transfer is performed for the following selected transfer data sizes within the correctly arranged address boundaries:
- When the transfer data size is set to 16 bits (SZSEL = 001): (bit 0) = 0
- When the transfer data size is set to 32 bits (SZSEL = 010): (bit 1, bit 0) = (0, 0)

11.3.6 DMA Reload Byte Count Register (DMRBCTn)

DMRBCTn is a register used to set the byte count to be reloaded to the DMA current byte count register (DMCBCTn). To enable the reload function, set the DMA byte count reload function enable bit (BRLOD) in DMA control register A (DMCNTAn) to 1. In this case, set both the DMA current byte count register (DMCBTn) and DMA reload byte count register (DMRBCTn).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	RBC[25:16]									
Initial value:	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBC[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 0	RBC[25:0]	Undefined	R/W	Number of DMA transfer bytes for reloading

Note: Set this register so that the byte count becomes 0 as follows when the final data is sent in a DMA transfer:

- When the transfer data size is set to 16 bits (SZSEL = 001): (bit 0) = 0
- When the transfer data size is set to 32 bits (SZSEL = 010): (bit 1, bit 0) = (0, 0)

11.3.7 DMA Mode Register (DMMODn)

DMMODn controls the amount of data, unit data size selection, address direction, and various signal outputs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	OPSEL[3:0]				-	-	-	-	-	SZSEL[2:0]		
Initial value:	0	0	0	0	-	-	-	-	0	0	0	0	0	-	-	-
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	SAMOD[2:0]			-	DAMOD[2:0]			-	-	-	-	SACT	DACT	DTCM[1:0]	
Initial value:	0	-	-	-	0	-	-	-	0	0	0	0	-	-	-	-
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

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.

Bit	Bit Name	Initial Value	R/W	Description
27 to 24	OPSEL [3:0]	Undefined	R/W	Single Operand Transfer Data Count Select These bits are used to set the number of data units to be transferred in single operand transfer. The amount of data set by theses bits is transferred continuously. Channel arbitration is not performed until this amount of data has been transferred (single operand transfer). These bits are invalid when non-stop transfer (DSEL = 11) is selected in the DMA transfer condition select bits (DSEL) of DMA control register A (DMCNTAn). 0000: 1 data unit 0001: 2 data units 0010: 4 data units 0011: 8 data units 0100: 16 data units 0101: 32 data units 0110: 64 data units 0111: 128 data units 1000 to 1111: Setting prohibited Note: Set the DMA current byte count register (DMCBCTn) so that it becomes H000 0000 when the last data is transferred via operand transfer. • When the transfer data size is set to 8 bits (SZSEL = 000) Integral multiple of the number of data units transferred in single operand transfer (×1, ×2, ×3, and so on) • When the transfer data size is set to 16 bits (SZSEL = 001) Number of data units transferred in single operand transfer multiplied by two (×2, ×4, ×6, and so on) • When the transfer data size is set to 32 bits (SZSEL = 010) Number of data units transferred in single operand transfer multiplied by four (×4, ×8, ×12, and so on) Operation is not guaranteed when values other than the above are set. For details, see section 11.3.3, DMA Current Byte Count Register (DMCBCTn) and section 11.3.6, DMA Reload Byte Count Register (DMRBCTn).

Bit	Bit Name	Initial Value	R/W	Description
23 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18 to 16	SZSEL [2:0]	Undefined	R/W	Transfer Data Size Select These bits are used to set the number of bits transferred in each single data transfer. A byte (8 bits), word (16 bits), or longword (32 bits) can be selected as the unit for transfer. (For details, see section 11.10, Units of Transfer and Transfer Byte Positions.) 000: Byte (8 bits) 001: Word (16 bits) 010: Longword (32 bits) 011 to 111: Setting prohibited
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	SAMOD [2:0]	Undefined	R/W	Source Address Direction Control These bits are used to set the source address counting direction. If these bits are set to 100 (two-dimensional addressing), the destination address direction control bits (DAMOD) cannot be set to 100. Two-dimensional addressing (100) can be set only in channels 0 to 7. Do not set two-dimensional addressing in other channels. 000: Fixed 001: Incrementation 010: Decrementation 011: Rotation 100: Two-dimensional addressing 101 to 111: Setting prohibited
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 to 8	DAMOD [2:0]	Undefined	R/W	Destination Address Direction Control These bits are used to set a destination address counting direction. If these bits are set to 100 (two-dimensional addressing), the source address direction control bits (SAMOD) cannot be set to 100. Two-dimensional addressing (100) can be set only in channels 0 to 7. Do not set two-dimensional addressing in other channels. 000: Fixed 001: Incrementation 010: Decrementation 011: Rotation 100: Two-dimensional addressing 101 to 111: Setting prohibited
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SACT	Undefined	R/W	Source DMA-active signal Output Control This bit is used to control the output of the DMA-active signal (DMAACTSk_N) for the source corresponding to the request source set in the DCTG bits. When this bit is set to 0, output of the DMAACTS_N signal is disabled and the signal is fixed high. When this bit is set to 1, a low-level DMAACTS_N is output (showing that DMA is active) from the next cycle after the start of the DMAC read cycle. When an on-chip peripheral module is selected as the DMA request source, be sure to set this bit to 1 (see table 11.5). 0: Disables output of the DMA-active signal for the source 1: Outputs DMA-active signal for the source during read access Note: In the context where indication of transfer request source number, k, is not necessary, the signal name is expressed as DMAACTS_N with k omitted.

Bit	Bit Name	Initial Value	R/W	Description
2	DACT	Undefined	R/W	Destination DMA-active signal Output Control This bit is used to control the output of the DMA-active signal (DMAACTDk_N) for the destination corresponding to the request source set in the DCTG bits. When this bit is set to 0, output of the DMAACTD_N signal is disabled and the signal is fixed high. When this bit is set to 1, a low-level DMAACTD_N is output (showing that DMA is active) from the next cycle after the start of the DMAC write cycle. When an on-chip peripheral module is selected as the DMA request source, be sure to set this bit to 1 (see table 11.6). 0: Disables output of the DMA-active signal for the destination 1: Outputs DMA-active signal for the destination during write access Note: In the context where indication of transfer request source number, k, is not necessary, the signal name is expressed as DMAACTD_N with k omitted.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	DTCM [1:0]	Undefined	R/W	DMA End Signal Output Control These bits are used to control the output of the DMA end signal (DMATCk_N) corresponding to the request source set in the DCTG bits when the DMA transfer end condition is detected. When these bits are set to 00, output of the DMATC_N signal is disabled and the signal is fixed high even if DMA transfer is completed. When these bits are set to 01, an active DMATC_N is output from the next cycle after the start of the read cycle immediately before completion of DMA transfer. When these bits are set to 10, an active DMATC _N is output from the next cycle after the start of the write cycle immediately before completion of DMA transfer. When these bits are set to 11, the DMATC_N signal goes active for one clock cycle to output a low pulse at the same timing as the DMA transfer end interrupt (for details, see figure 11.6). When selecting USB_0 or USB_1 as the request source, be sure to set these bit to 10 (see table 11.7). 00: Disables output of the DMA end signal. 01: Outputs the DMA end signal in the last read cycle. 10: Outputs the DMA end signal in the last write cycle. 11: Outputs the DMA end signal after DMA has been completed. Note: In the context where indication of transfer request source number, k, is not necessary, the signal name is expressed as DMATC_N with k omitted.

Note: Only write data to this register when the corresponding channel is not engaged in single operand transfer (the corresponding DASTS bit in the DMA arbitration status register (DMASTS) is 0) and DMA transfer is disabled (the DMST bit in the DMA activation control register (DMSCNT) is 0 or the DEN bit in DMA control register B (DMCNTBn) is 0). In other cases, operation is not guaranteed when data is written to this register.

Table 11.4 shows the counter increment and decrement of DMA source/destination address registers (for details on the “rotation” addressing mode, see section 11.12, Rotate Function). If two-dimensional addressing is specified in these bits (SAMOD and DAMOD), the settings of registers related to two-dimensional addressing (section 11.3.16, DMA Two-Dimensional Addressing Column Setting Register (DM2DCLMm), and after) become valid. When performing pipelined transfer to or from external devices and modules that support burst access, make sure to set the direction bits to select address incrementation (001), rotation (011), or two dimensions (100).

Table 11.4 Counter Increment/Decrement for DMA Source/Destination Address Registers

	Addressing Mode SAMOD or DAMOD				
Transfer Data Size Select Bits (SZSEL)	**000 (Fixed)**	**001 (Incremen-tation)**	**010 (Decremen-tation)**	**011 (Rotation)**	**100 (Two dimensions)**
000 (8 bits)	±0	+1	-1	+1	+1
001 (16 bits)	±0	+2	-2	+2	+2
010 (32 bits)	±0	+4	-4	+4	+4

Table 11.5 shows the relationship between DMA request sources and the DMA-active signal output control bit for the source. If the DREQ0 to DREQ3 pins are selected as the DMA request source, select "0: Stop" or "1: Output" as required. The signal corresponding to this setting is output to the $\overline{\text{DACT0}}$ to $\overline{\text{DACT3}}$ external pins (see section 11.9, DMA Acknowledge Signal Output and DMA-Active Signal Output). If the software trigger is selected, setting of this bit has no effect, so either 0 or 1 can be set. If other DMA request sources are selected, be sure to set "1: Output".

Table 11.5 Relationship between DMA Request Sources and DMA-active signal Output Control Bit for Source

	SACT Bit Setting		
DMA Request Source	**0: Stop**	**1: Output**	**DCTG Bit Setting**
Software trigger	—	—	000000
DREQ0 pin	O	O	000001
DREQ1 pin	O	O	000010
DREQ2 pin	O	O	000011
DREQ3 pin	O	O	000100
Other DMA request sources	×	O	Other than the above

[Legend]

O: Can be set

×: Setting prohibited

—: Setting ignored

Table 11.6 shows the relationship between DMA request sources and the DMA-active signal output control bit for the destination. If the DREQ0 to DREQ3 pins are selected as the DMA request source, select "0: Stop" or "1: Output" as required. The signal corresponding to this setting is output to the $\overline{\text{DACT0}}$ to $\overline{\text{DACT3}}$ external pins (see section 11.9, DMA Acknowledge Signal Output and DMA-Active Signal Output). If the software trigger is selected, setting of this bit has no effect, so either 0 or 1 can be set. If other DMA request sources are selected, be sure to set "1: Output".

Table 11.6 Relationship between DMA Request Sources and DMA-active signal Output Control Bit for Destination

	DACT Bit Setting		
DMA Request Source	**0: Stop**	**1: Output**	**DCTG Bit Setting**
Software trigger	—	—	000000
DREQ0 pin	O	O	000001
DREQ1 pin	O	O	000010
DREQ2 pin	O	O	000011
DREQ3 pin	O	O	000100
Other DMA request sources	×	O	Other than the above

[Legend]

O: Can be set

×: Setting prohibited

—: Setting ignored

Table 11.7 shows the relationship between DMA request sources and the DMA end signal output control bit. If the DREQ0 to DREQ3 pins are selected as the DMA request source, select 00, 01, 10, or 11 as required. The signal corresponding to this setting is output to the $\overline{\text{TEND0}}$ to $\overline{\text{TENDT3}}$ external pins (see section 11.5.3, DMA End Signal Output). If USB_0, USB_1, or a 2DG-related source is selected, be sure to select 11. If the software trigger or other DMA request source is selected, setting of this bit has no effect, so either 0 or 1 can be set.

Table 11.7 Relationship between DMA Request Sources and DMA End Signal Output Control Bit

	DTCM Bit Setting				
DMA Request Source	**00: Output Stop**	**01: Last Read Cycle**	**10: Last Write Cycle**	**11: After DMA has Ended**	**DCTG Bit Setting**
Software trigger	—	—	—	—	000000
DREQ0 pin	O	O	O	O	000010
DREQ1 pin	O	O	O	O	000010
DREQ2 pin	O	O	O	O	000011
DREQ3 pin	O	O	O	O	000100
USB_0	×	×	×	O	000101
USB_1	×	×	×	O	000110
2DG output	×	×	×	O	101111
2DG BLT input A	×	×	×	O	110000
2DG BLT input B	×	×	×	O	110001
2DG BLT output C	×	×	×	O	110010
Other DMA request sources	—	—	—	—	Other than the above

[Legend]

O: Can be set

×: Setting prohibited

—: Setting ignored

11.3.8 DMA Control Register A (DMCNTAn)

DMCNTAn is used to select transfer modes, transfer conditions, and DMA sources and control various reload functions.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	MDSEL[1:0]		-	-	DSEL[1:0]		-	-	-	-	-	-	STRG[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	2DRLOD	BRLOD	SRLOD	DRLOD	-	-	DCTG[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

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, 28	MDSEL [1:0]	00	R/W	DMA Transfer Mode Select These bits are used to set DMA transfer mode. Setting these bits to 00 selects cycle-stealing transfer mode. Setting these bits to 01 selects pipelined transfer mode. Do not set these bits to 10 or 11. Operation is not guaranteed if these settings are made (for details, see section 11.4.1, DMA Transfer Mode). 00: Cycle-stealing transfer 01: Piepelined transfer 10: Setting prohibited 11: Setting prohibited Note: If the source or destination is an SDRAM device when pipelined transfer mode (MDSE L = 01) is selected, non-stop transfer (DSEL = 11) cannot be set.
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, 24	DSEL[1:0]	00	R/W	DMA Transfer Condition Select These bits are used to set DMA transfer conditions. Setting these bits to 00 selects unit operand transfer. Setting these bits to 01 selects sequential operand transfer. Setting these bits to 11 selects non-stop transfer (for details, see section 11.4.2, DMA Transfer Conditions). Do not set these bits to 10. Operation is not guaranteed if this setting is made. 00: Unit operand transfer 01: Sequential operand transfer 10: Setting prohibited 11: Non-stop transfer
23 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	STRG[1:0]	00	R/W	Input Sense Mode Select These bits are used to set input sense mode for the DMA request signal to be input to the DMAC from the request source selected by the DMA request source select bit (DCTG). Table 11.8 shows the relationship between DMA request sources and input sense modes. If the software trigger (DCTG = 000000) is selected as the request source, set rising edge sense. If the DREQ pins (DCTG = 000001 to 000100) are selected, any input sense mode can be selected. If other request sources are selected, be sure to set low level sense. 00: Rising edge sense 01: High level sense 10: Falling edge sense 11: Low level sense
15 to 12	—	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
11	2DRLOD	0	R/W	Two-Dimensional Addressing Reload Function Enable This bit is used to enable or disable reloading to the six current two-dimensional addressing registers upon detection of the DMA transfer end condition. When this bit is cleared to 0, the contents of the six reload registers for two-dimensional addressing are not reloaded to the six current two-dimensional addressing registers. If the DMA transfer end condition is detected when this bit is set to 1, the contents of the six reload registers are reloaded to the six current two-dimensional addressing registers. This bit is only valid in channels 0 to 7. 0: Two-dimensional addressing reload function disabled 1: Two-dimensional addressing reload function enabled
10	BRLOD	0	R/W	DMA Byte Count Reload Function Enable This bit is used to select whether to reload the byte counter when the DMA transfer end condition is detected. When this bit is cleared to 0, reloading is not performed. If the DMA transfer end condition is detected when this bit is set to 1, the contents of the DMA reload byte count register (DMRBCTn) are reloaded to the DMA current byte count register (DMCBCTn). 0: Byte count reload function disabled 1: Byte count reload function enabled
9	SRLOD	0	R/W	DMA Source Address Reload Function Enable This bit is used to select whether to reload the source address when the DMA transfer end condition is detected. When this bit is cleared to 0, reloading is not performed. If the DMA transfer end condition is detected when this bit is set to 1, the contents of the DMA reload source address register (DMRSADRn) are reloaded to the DMA current source address register (DMCSADRn). 0: Source address reload function disabled 1: Source address reload function enabled

Bit	Bit Name	Initial Value	R/W	Description
8	DRLOD	0	R/W	DMA Destination Address Reload Function Enable This bit is used to select whether to reload the destination address when the DMA transfer end condition is detected. When this bit is cleared to 0, reloading is not performed. If the DMA transfer end condition is detected when this bit is set to 1, the contents of the DMA reload destination address register (DMRDADRn) are reloaded to the DMA current destination address register (DMCDADRn). 0: Destination address reload function disabled 1: Destination address reload function enabled
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	DCTG[5:0]	000000	R/W	DMA Request Source Select These bits are used to set DMA request sources. When selecting a DMA request source other than the software trigger, DREQ0 to DREQ3 pins, USB, FLCTL, and 2DG, set the DMA transfer request enable bits in DREQER0 to DREQER8 of the interrupt controller (INTC). For how to set DREQER0 to DREQER8, see section 7, Interrupt Controller (INTC). 000000: Software trigger 000001: DREQ0 pin 000010: DREQ1 pin 000011: DREQ2 pin 000100: DREQ3 pin 000101: USB_0 000110: USB_1 000111: CMT_0 001000: CMT_1 001001: CMT_2 001010: CMT_3 001011: MTU2_0 001100: MTU2_1

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	DCTG[5:0]	000000	R/W	(Continued) 001101: MTU2_2 001110: MTU2_3 001111: MTU2_4 010000: IIC3_0 reception 010001: IIC3_0 transmission 010010: IIC3_1 reception 010011: IIC3_1 transmission 010100: IIC3_2 reception 010101: IIC3_2 transmission 010110: IIC3_3 reception 010111: IIC3_3 transmission 011000: SCIF_0 reception 011001: SCIF_0 transmission 011010: SCIF_1 reception 011011: SCIF_1 transmission 011100: SCIF_2 reception 011101: SCIF_2 transmission 011110: SCIF_3 reception 011111: SCIF_3 transmission 100000: SCIF_4 reception 100001: SCIF_4 transmission 100010: SCIF_5 reception 100011: SCIF_5 transmission 100100: SSIF_0 transmission/reception 100101: SSIF_1 transmission/reception 100110: SSIF_2 transmission/reception 100111: SSIF_3 transmission/reception 101000: SSIF_4 transmission/reception 101001: SSIF_5 transmission/reception 101010: SSU_0 reception 101011: SSU_0 transmission

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	DCTG[5:0]	000000	R/W	(Continued) 101100: SSU_1 reception 101101: SSU_1 transmission 101110: A/D converter 101111: 2DG output 110000: 2DG BLT input A 110001: 2DG BLT input B 110010: 2DG BLT output C 110011: FLCTL_0 transmission/reception 110100: FLCTL_1 transmission/reception 110111: RM0_0 (RCAN) 111000: RM0_1 (RCAN) Other than the above: Setting prohibited

Note: Modify the settings of bits of this register other than the reload function enable bits (BRLOD, SRLOD, and DRLOD) only when the corresponding channel is not undergoing single operand transfer (the DASTS bit of the DMA arbitration status register (DMASTS) is 0) and DMA transfer is disabled (the DMST bit of the DMA activation control register (DMSCNT) is 0 or the DEN bit of DMA control register B (DMCNTBn) is 0). In other cases, operation is not guaranteed when data is written to this register.

Table 11.8 Relationship between DMA Request Sources and Input Sense Modes

DMA Request Source	STRG Bit Settings: 00: Rising Edge Sense	01: High Level Sense	10: Falling Edge Sense	11: Low Level Sense	DCTG Bit Settings
Software trigger	O	×	×	×	000000
DREQ0 pin	O	O	O	O	000001
DREQ1 pin	O	O	O	O	000010
DREQ2 pin	O	O	O	O	000011
DREQ3 pin	O	O	O	O	000100
Other DMA request sources	×	×	×	O	Other than the above

[Legend]

O: Can be set

×: Setting prohibited

Note: The input sense modes of other DMA request sources may be changed in the future because they are for the preliminary version.

11.3.9 DMA Control Register B (DMCNTBn)

DMCNTBn controls whether to enable or disable DMA transfer, transfer enable clearing, and internal status clearing. This register can also reference the DMA request status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	DEN	-	-	-	-	-	-	-	DREQ
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ECLR	-	-	-	-	-	-	-	DSCLR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

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.
24	DEN	0	R/W	DMA Transfer Enable This bit is used to enable or disable DMA transfer. Clearing this bit to 0 disables DMA transfer on the channel. Setting this bit to 1 enables DMA transfer on the channel (for DMA transfer activation, see section 11.4.3, DMA Activation). Even if this bit is cleared to 0, the DMA request bit (DREQ) changes according to the DMA request input to the DMAC. If the DMA transfer enable clear bit (ECLR) is 1, this bit is automatically cleared to 0 when the DMA transfer end condition is detected. Clearing this bit to 0 during DMA transfer also enables you to stop the channel after the current single operand transfer is completed (for details, see section 11.6, Suspending, Restarting, and Stopping of DMA Transfer). 0: DMA transfer disabled 1: DMA transfer enabled
23 to 17	—	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
16	DREQ	0	R/W	DMA Request This bit enables you to check whether a DMA request is currently present. If the software trigger (DCTG = 000000) is selected by the DMA request source select bits (DCTG), use this bit to operate DMA requests. 0: DMA request not currently present 1: DMA request currently present The value of this bit varies with the status of the DMA request signal input to the DMAC regardless of the settings of the DMAC module activate bit (DMST) and DMA transfer enable bit (DEN). The setting and clearing conditions of this bit vary with the settings of the DMA request source select bits (DCTG) and input sense mode select bits (STRG) as described below. (a) When the software trigger is selected by the DMA request source select bits (DCTG) • Setting condition This bit is set to 1 when software writes 1 to the bit. This generates a DMA request. • Clearing condition This bit is cleared to 0 when any of the following conditions is met: — Software writes 0 to this bit. — Operand transfer is started that corresponds to this bit. (b) When a request source other than the software trigger is selected by the DMA request source select bits (DCTG) and a level sense is selected • Setting condition This bit is set to 1 when the DMA request signal input level matches that specified in the input sense mode select bits (STRG), i.e. when a DMA request exists. • Clearing condition This bit is cleared to 0 when the level set in the input sense mode select bits (STRG) does not match the DMA request signal input level, i.e. when no DMA request exists. If a DMA request disappears before being accepted, it is not retained and the DMA request bit (DREQ) is cleared to 0. For this reason, to use the DMA request with a level sense, retain the request till it is accepted.

Bit	Bit Name	Initial Value	R/W	Description
16	DREQ	0	R/W	(Continued) (c) When a request source other than the software trigger is selected by the DMA request source select bits (DCTG) and an edge sense is selected • Setting condition This bit is set to 1 when the edge specified in the input sense mode select bits (STRG) is encountered (when a DMA request exists). Once set to 1, regardless of the status of subsequent DMA request signals, this bit remains set till any of the conditions for clearing this bit to 0 is met. • Clearing condition This bit is cleared to 0 when any of the following conditions is met: — Software writes 0 to this bit. — Operand transfer is started that corresponds to this bit. Notes: 1. If the selected request source is other than the software trigger, do not write 1 to this bit. If 1 is written to this bit, operation is not guaranteed. 2. When the DMA request source select bits (DCTG) and input sense mode select bits (STRG) of DMA control register A (DMCNTAn) are set, be sure to clear the DMA request bit (DREQ) of the set channel to 0 and then enable DMA transfer (DMST = 1, DEN = 1).
15 to 9	—	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
8	ECLR	0	R/W	DMA Transfer Enable Clear This bit is used to select whether to clear the DMA transfer enable bit (DEN) to 0 when the DMA transfer end condition is detected. Clearing this bit to 0 does not clear the DMA transfer enable bit (DEN) to 0 even when the DMA transfer end condition is detected. Setting this bit to 1 clears the DMA transfer enable bit (DEN) to 0 when the DMA transfer end condition is detected. 0: Does not clear the DMA transfer enable bit to 0 even when the DMA transfer end condition is detected. 1: Clears the DMA transfer enable bit to 0 when the DMA transfer end condition is detected. Note: If a value is written to the DMA transfer enable clear bit for the channel on which operand transfer is in progress, operation is not guaranteed.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DSCLR	0	R/W	DMAC Internal State Clear This bit enables you to stop the remaining DMA transfer operations in the middle of the current DMA transfer (before the byte count reaches 0) and initialize the DMAC internal state. Writing 1 to this bit clears the transfer status of the DMAC internal circuit but does not initialize each register. This bit is always read as 0. When read: This bit is read as 0. When written: 0: Ignored 1: Initializes the DMAC internal state. Note: Only write data to this register when the corresponding channel is not engaged in single operand transfer (the corresponding DASTS bit in the DMA arbitration status register (DMASTS) is 0) and DMA transfer is disabled (the DMST bit in the DMA activation control register (DMSCNT) is 0 or the DEN bit in DMA control register B (DMCNTBn) is 0). In other cases, operation is not guaranteed when data is written to this register.

Note: If the software trigger is selected as the DMA request source, the DMA request bit (DREQ) can be set to 1 regardless of the settings of the DMA transfer enable bit (DEN) and DMAC module activate bit (DMST) and the operand transfer status. However, even if the software trigger is selected as the DMA request source, clear the DMA request bit (DREQ) to 0 or write data to the DMAC internal state clear bit (DSCLR) only when the corresponding channel is not engaged in single operand transfer (the corresponding DASTS bit in the DMA arbitration status register (DMASTS) is 0) and DMA transfer is disabled (the DMST bit in the DMA activation control register (DMSCNT) is 0 or the DEN bit in DMA control register B (DMCNTBn) is 0). In other cases, operation is not guaranteed when data is written to this register.

11.3.10 DMA Activation Control Register (DMSCNT)

DMSCNT controls the entire DMAC operation.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DMST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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	DMST	0	R/W	DMAC Module Activate This bit is used to set DMAC module operation/stop. Clearing this bit to 0 stops the DMAC. Setting this bit to 1 activates the DMAC (for details, see section 11.4.3, DMA Activation, and section 11.6, Suspending, Restarting, and Stopping of DMA Transfer). 0: DMAC stops 1: DMAC operates
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

11.3.11 DMA Interrupt Control Register (DMICNT)

DMICNT controls DMA interrupt for each channel.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DIN TM0	DIN TM1	DIN TM2	DIN TM3	DIN TM4	DIN TM5	DIN TM6	DIN TM7	DIN TM8	DIN TM9	DIN TM10	DIN TM11	DIN TM12	DIN TM13	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	DINTM0 to DINTM13	H'0000	R/W	DMA Interrupt Control These bits are used to control whether to generate a DMA transfer end interrupt of each channel to the interrupt controller. Clearing these bits to 0 does not generate an interrupt request to the interrupt controller. Setting these bits to 1 generates an interrupt request to the interrupt controller when the DMA transfer end condition is detected (for details, see section 11.5.2, DMA Interrupt Requests). 0: Interrupt disabled 1: Interrupt enabled
17 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: Bits 31 to 18 correspond to channels 0 to 13.

11.3.12 DMA Common Interrupt Control Register (DMICNTA)

DMICNTA controls DMA interrupts for each channel.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DINTA0	DINTA1	DINTA2	DINTA3	DINTA4	DINTA5	DINTA6	DINTA7	DINTA8	DINTA9	DINTA10	DINTA11	DINTA12	DINTA13	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	DINTA0 to DINTA13	H'0000	R/W	DMA Common Interrupt Request Signal Control These bits are used to determine which channels contribute to the output of a common interrupt request signal. Only the channels for which these bits are set to 1 are grouped into one as the common interrupt request signal. The channels for which these bits are cleared to 0 do not contribute to the output of a common interrupt request signal. Also only the channels for which these bits are set to 1 are reflected in the DMA interrupt status register (DMISTS) when a common interrupt request signal is generated (for details, see section 11.5.2, DMA Interrupt Requests). 0: The channel is not involved in common interrupt request. 1: The channel is involved in common interrupt request.
17 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: Bits 31 to 18 correspond to channels 0 to 13.

11.3.13 DMA Interrupt Status Register (DMISTS)

DMISTS consists of DMA interrupt request status bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DIS TS0	DIS TS1	DIS TS2	DIS TS3	DIS TS4	DIS TS5	DIS TS6	DIS TS7	DIS TS8	DIS TS9	DIS TS10	DIS TS11	DIS TS12	DIS TS13	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	DISTS0 to DISTS13	H'0000	R	DMA Interrupt Request Status These bits enable you to reference the occurrence status of a common interrupt request for the interrupt controller. 0: No interrupt request occurred. 1: An interrupt request occurred. • Condition for setting these bits to 1 If the DMA transfer end condition is detected when the DMA common interrupt request signal control bit (DINTA) is set to 1, the bits of the corresponding channel are set to 1. The setting of the DMA interrupt control bit (DINTM) does not affect this setting. • Condition for clearing these bits to 0 Clearing the DMA transfer end condition detect bit (DEDET) of the DMA transfer end detection register (DMEDET) corresponding to the channel where the interrupt occurred to 0 clears these bits to 0 (for details, see section 11.5.2, DMA Interrupt Requests).
17 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. This register is read-only.
2. Bits 31 to 18 correspond to channels 0 to 13.

11.3.14 DMA Transfer End Detection Register (DMEDET)

DMEDET references the DMA transfer end detection status of each channel. Writing 0 to the DEDET bits is invalid and 1s written to these bits are not retained.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDET0	DEDET1	DEDET2	DEDET3	DEDET4	DEDET5	DEDET6	DEDET7	DEDET8	DEDET9	DEDET10	DEDET11	DEDET12	DEDET13	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	DEDET0 to DEDET13	H'0000	R/W	When read: DMA Transfer End Condition Detect 0: The DMA transfer end condition has not been detected. 1: The DMA transfer end condition was detected. When written: DMA Interrupt Request Status Clear 0: Invalid 1: Clears the DMA interrupt request status. These bits enable you to reference the detection status of DMA transfer end conditions of each channel. Reading this register does not provide automatic bit clearing. When set to 1, these bits always retain values unless they are cleared by software. • Condition for setting these bits to 1 When the DMA transfer end condition is detected, these bits are set to 1. • Condition for clearing these bits to 0 These bits are cleared to 0 by writing 1 to them. Write 0 to the bits not to be cleared. The bits to which 0 was written retain the values before write operation because they are not affected by write via software. To use the DMA transfer end interrupt, write 1 to the DMA transfer end condition detect bits (DEDET) of the channel where an interrupt request occurred in the interrupt handler. When the DMA transfer end condition detect bits (DEDET) are cleared to 0, the DMA interrupt request status bit (DISTS) is also cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
17 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: Bits 31 to 18 correspond to channels 0 to 13.

11.3.15 DMA Arbitration Status Register (DMASTS)

DMASTS is used to reference the DMA transfer status of each channel. Writing to this register is invalid.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DASTS0	DASTS1	DASTS2	DASTS3	DASTS4	DASTS5	DASTS6	DASTS7	DASTS8	DASTS9	DASTS10	DASTS11	DASTS12	DASTS13	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	DASTS0 to DASTS13	H'0000	R	When read: DMA Arbitration Status 0: Operand transfer not in progress 1: Operand transfer in progress These bits enable you to reference the DMA transfer status of each channel. • Condition for setting these bits to 1 The bit corresponding to the channel that has started operand transfer or non-stop transfer is set to 1. • Condition for clearing these bits to 0 When single operand transfer or non-stop transfer ends, the bit of the corresponding channel is cleared to 0. Note: In DMA transfer to external devices, the DMA arbitration status bit (DASTS) may be cleared before the end of external bus access (the last data-write operation is already started).
17 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: Bits 31 to 18 correspond to channels 0 to 13.

11.3.16 DMA Two-Dimensional Addressing Column Setting Register (DM2DCLMm)

DM2DCLMm is a register used to set the number of data columns in one block in two-dimensional addressing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCDN[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 to 0	DCDN [15:0]	Undefined	R/W	DMA Block Data Column Count These bits are used to set the number of data columns in one block. 00000000_00000000: 1 data column : 11111111_11111111: 65536 data columns Note: Set the number identical to the number of data units set in the single operand transfer data count select bits (OPSEL) or an integral multiple of it. Operation is not guaranteed if different setting is made.

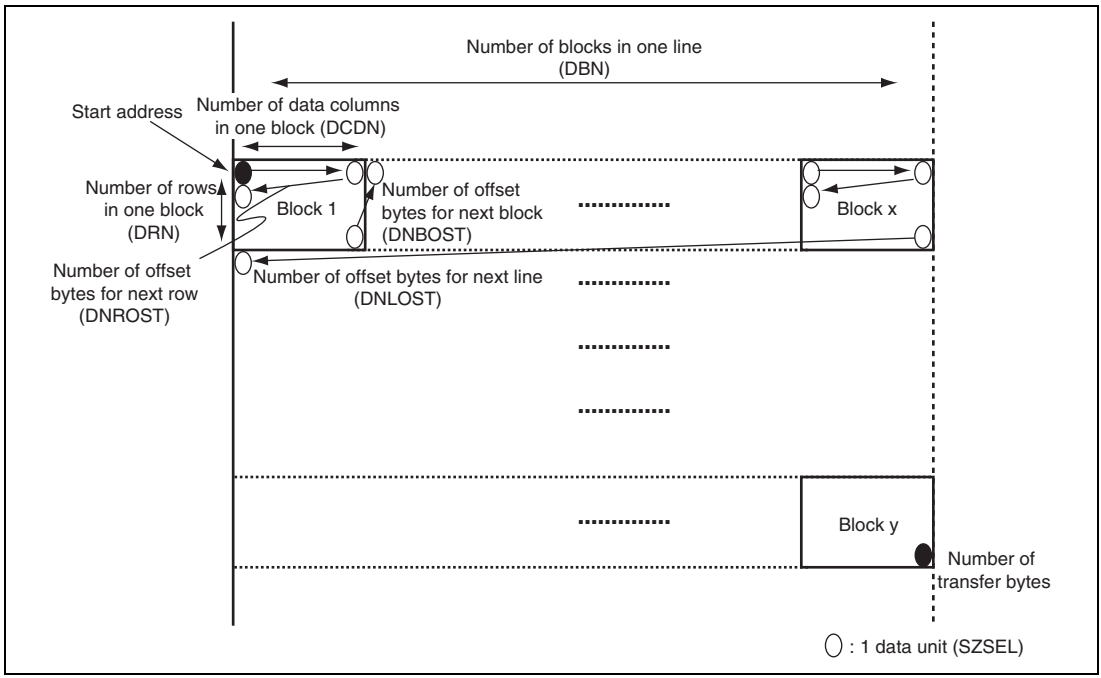


Figure 11.2 Specifying Two-Dimensional Blocks

11.3.17 DMA Two-Dimensional Addressing Row Setting Register (DM2DROWm)

DM2DROWm is a register used to set the number of rows in one block in two-dimensional addressing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRN[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 to 0	DRN [15:0]	Undefined	R/W	DMA Block Row Count These bits are used to set the number of rows in one block. 00000000_00000000: 1 row : 11111111_11111111: 65536 rows

11.3.18 DMA Two-Dimensional Addressing Block Setting Register (DM2DBLKm)

DM2DBLKm is a register used to set the number of blocks per line in two-dimensional addressing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	DBN[23:16]							
Initial value:	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DBN[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	DBN [23:0]	Undefined	R/W	DMA Block Count These bits are used to set the number of blocks in one line. 00000000_00000000_00000000: 1 block : 11111111_11111111_11111111: 16777216 blocks

11.3.19 DMA Two-Dimensional Addressing Next Row Offset Register (DM2DNROSTm)

DM2DNROSTm is a register used to set the offset for calculating the start address of the next row in two-dimensional addressing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DNROST[31:16]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DNROST[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DNROST [31:0]	Undefined	R/W	DMA2D Next Row Offset Byte Count In two-dimensional addressing, these bits are used to set the number of bytes to be added to the current source or destination address to calculate the start address of the next row when DMA transfer of one row in one block ends. Set a two's complement number in these bits.

11.3.20 DMA Two-Dimensional Addressing Next Block Offset Register (DM2DNBOSTm)

DM2DMBOSTm is a register used to set the offset for calculating the start address of the next block in two-dimensional addressing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DNBOST[31:16]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DNBOST[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DNBOST [31:0]	Undefined	R/W	DMA2D Next Block Offset Byte Count These bits are used to set the number of bytes to be added to the current source or destination address to calculate the start address of the next block when DMA transfer of one block ends in two-dimensional addressing. Set a two's complement number in these bits.

11.3.21 DMA Two-Dimensional Addressing Next Line Offset Register (DM2DNLOSTm)

DM2DNLOSTm is a register used to set the offset for calculating the start address of the next line in two-dimensional addressing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DNLOST[31:16]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DNLOST[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DNLOST [31:0]	Undefined	R/W	DMA2D Next Line Offset Byte Count These bits are used to set the number of bytes to be added to the current source or destination address to calculate the start address of the next line when DMA transfer of one line ends in two-dimensional addressing. Set a two's complement number in these bits.

11.3.22 DMA Reload Two-Dimensional Addressing Column Setting Register (DMR2DCLMm)

DMR2DCLMm is a register used to set the number of data columns to be reloaded to the DMA two-dimensional addressing column setting register (DM2DCLMm). To enable the reload function, set the two-dimensional reload function enable bit (2DRLOD) of DMA control register A (DMCNTAm) to 1. When enabled, it is necessary to set both the DMA two-dimensional addressing column setting register (DM2DCLMm) and DMA reload two-dimensional addressing column setting register (DMR2DCLMm).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRCDN[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 to 0	DRCDN [15:0]	Undefined	R/W	DMA Block Data Column Count for Reloading These bits are used to set the number of data columns in one block to be reloaded to the DMA two-dimensional addressing column setting register. 00000000_00000000: 1 data unit : 11111111_11111111: 65536 data units Note: Set the number of data units identical to the number of data units set in the single operand transfer data count select bits (OPSEL) or a number that is an integral multiple of the number of data units. Operation is not guaranteed if different settings are made.

11.3.23 DMA Reload Two-Dimensional Addressing Row Setting Register (DMR2DROWm)

DMR2DROWm is a register used to set the number of rows to be reloaded to the DMA two-dimensional addressing row setting register (DM2DROWm). To enable the reload function, set the two-dimensional reload function enable bit (2DRLOD) of DMA control register A (DMCNTAm) to 1. When enabled, it is necessary to set both the DMA two-dimensional addressing row setting register (DM2DROWm) and DMA reload two-dimensional addressing row setting register (DMR2DROWm).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRRN[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 to 0	DRRN [15:0]	Undefined	R/W	DMA Block Data Row Count for Reloading These bits are used to set the number of rows in one block to be reloaded to the DMA two-dimensional addressing row setting register. 00000000_00000000: 1 row : 11111111_11111111: 65536 rows

11.3.24 DMA Reload Two-Dimensional Addressing Block Setting Register (DMR2DBLKm)

DMR2DBLKm is a register used to set the number of blocks to be reloaded to the DMA two-dimensional addressing block setting register (DM2DBLKm). To enable the reload function, set the two-dimensional reload function enable bit (2DRLOD) of DMA control register A (DMCNTAm) to 1. When enabled, it is necessary to set both the DMA two-dimensional addressing block setting register (DM2DBLKn) and DMA reload two-dimensional addressing block setting register (DMR2DBLKm).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	DRBN[23:16]							
Initial value:	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRBN[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	DRBN [23:0]	Undefined	R/W	DMA Block Count for Reloading These bits are used to set the number of blocks in one line to be reloaded to the DMA two-dimensional addressing block setting register. 00000000_00000000_00000000: 1 block : 11111111_11111111_11111111: 16777216 blocks

11.3.25 DMA Reload Two-Dimensional Addressing Next Row Offset Register (DMR2DNROSTm)

DMR2DNROSTm is a register used to set the offset to be reloaded to the DMA two-dimensional row offset register (DM2DNROSTm). To enable the reload function, set the two-dimensional reload function enable bit (2DRLOD) of DMA control register A (DMCNTAm) to 1. When enabled, it is necessary to set both the DMA two-dimensional addressing row offset register (DM2DNROSTm) and DMA reload two-dimensional addressing row offset register (DMR2DNROSTm).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DRNROST[31:16]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRNROST[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DRNROST [31:0]	Undefined	R/W	DMA Next Row Offset Byte Count for Reloading These bits are used to set the number of offset bytes to be reloaded to the DMA two-dimensional addressing next row offset register. Set a two's complement number in these bits.

11.3.26 DMA Reload Two-Dimensional Addressing Next Block Offset Register (DMR2DNBOSTm)

DMR2DNBOSTm is a register used to set the offset to be reloaded to the DMA two-dimensional addressing next block offset register (DM2DNBOSTm). To enable the reload function, set the two-dimensional reload function enable bit (2DRLOD) of DMA control register A (DMCNTAm) to 1. When enabled, it is necessary to set both the DMA two-dimensional addressing next block offset register (DM2DNBOSTm) and DMA reload two-dimensional addressing next block offset register (DMR2DNBOSTm).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DRNBOST[31:16]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRNBOST[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DRNBOST [31:0]	Undefined	R/W	DMA Next Block Offset Byte Count for Reloading These bits are used to set the number of offset bytes to be reloaded to the DMA two-dimensional addressing next block offset register. Set a two's complement number in these bits.

11.3.27 DMA Reload Two-Dimensional Addressing Next Line Offset Register (DMR2DNLOSTm)

DMR2DNLOSTm is a register used to set the offset to be reloaded to the DMA two-dimensional addressing next line offset register (DM2DNLOSTm). To enable the reload function, set the two-dimensional reload function enable bit (2DRLOD) of DMA control register A (DMCNTAm) to 1. When enabled, it is necessary to set both the DMA two-dimensional addressing next line offset register (DM2DNLOSTm) and DMA reload two-dimensional addressing next line offset register (DMR2DNLOSTm).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DRNLOST[31:16]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRNLOST[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DRNLOST [31:0]	Undefined	R/W	DMA Next Line Offset Byte Count for Reloading These bits are used to set the number of offset bytes to be reloaded to the DMA two-dimensional addressing next line offset register. Set these bits by using a complement of 2.

11.4 Operation

11.4.1 DMA Transfer Mode

Two DMA transfer modes are available: cycle-stealing transfer mode and pipelined transfer mode. These modes can be selected by using the DMA transfer mode select bits (MDSEL) of DMA control register A (DMCNTAn).

Figure 11.3 shows how bus mastership alternates between the DMAC and CPU in DMA transfer modes.

(1) Cycle-Stealing Transfer Mode

Setting the DMA transfer mode select bits (MDSEL) to 00 selects cycle-stealing transfer mode.

In cycle-stealing transfer mode, the DMAC operates, leaving at least one cycle between the read and write access cycles (activations) of each single data transfer. For this reason, access from the CPU is possible during this interval (the CPU can access the BIU part of the source or destination target).

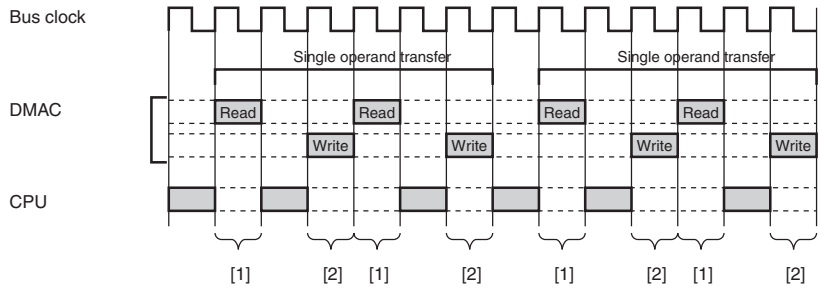
(2) Piepelined Transfer Mode

Setting the DMA transfer mode select bits (MDSEL) to 01 selects pipelined transfer mode.

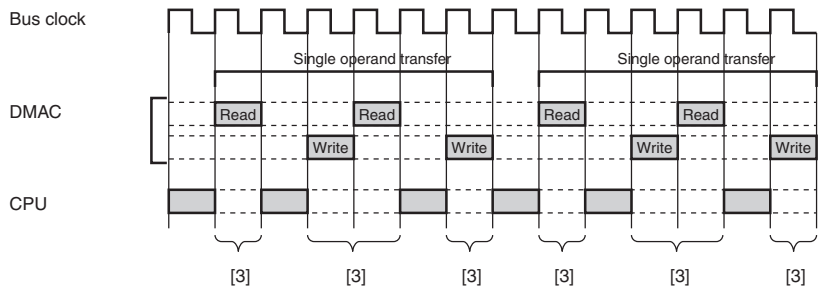
In pipelined transfer mode, the DMAC consecutively accesses the bus for read access or write access or both. For this reason, access from the CPU is not accepted till the current single operand transfer ends (the CPU cannot access the BIU part of the source or destination target).

Piepelined transfer through a single BIU is also not possible.

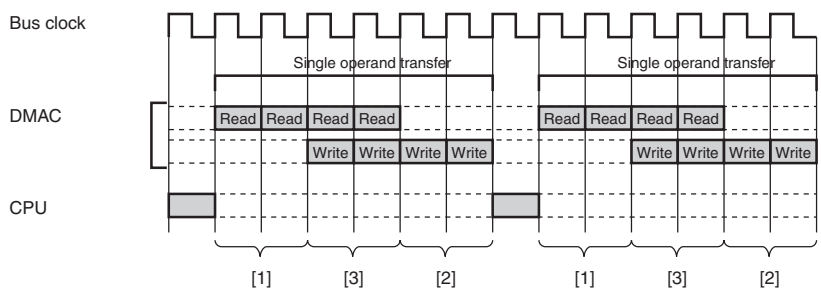
• Cycle-stealing transfer mode (transfer through different BIUs)



• Cycle-stealing transfer mode (transfer through the same BIU)



• Pipelined transfer mode (transfer through different BIUs)



This example shows that all DMA transfers ended in one cycle.
Of CPU access cycles, the colored cycles can access any BIUs.
Cycles [1] to [3] indicate the following operating states:

[1]: CPU can access BIUs other than the BIU on the reading side of the DMAC.
[2]: CPU can access BIUs other than the BIU on the writing side of the DMAC.
[3]: CPU can access BIUs other than the BIUs on the reading and writing sides of the DMAC.

However, when a DMA access to external address space followed by CPU access to external address space is occurred, CPU access next to DMA cycle may not be occurred.

Figure 11.3 Example of Bus Mastership Alternation between DMAC and CPU in Various DMA Transfer Modes

11.4.2 DMA Transfer Conditions

Three DMA transfer conditions are available: unit operand transfer, sequential operand transfer, and non-stop transfer. These conditions can be selected by using the DMA transfer condition select bits (DSEL) of DMA control register A (DMCNTAn). Each DMA transfer condition is described below. Table 11.9 lists DMA transfer conditions, which are illustrated in figure 11.4.

(1) Unit Operand Transfer

Setting the DMA transfer condition select bits (DSEL) to 00 selects unit operand transfer.

A single DMA request transfers data by the amount specified in the single operand transfer data count select bits (OPSEL) of the DMA mode register (DMMODn).

Each time a DMA transfer request is made, the DMAC repeats single operand transfer and ends single DMA transfer when the byte count reaches 0.

In the case that the DMA transfer condition is the unit operand transfer and the input sense mode of DMA request is the level sense, there is the mask period of the DMA request in the channel arbitration period after one operand transfer end (please refer to section 11.7.3, Sense Mode for DMA Requests for details). Therefore, in the channel arbitration period after one operand transfer end, in the case that there is no DMA request of the higher-priority channel than the transferring channel and there is the DMA request of the lower-priority channel than the transferring channel, the DMA transfer of the low-priority channel starts. To execute the DMA transfer of the high-priority channel in succession, please set the DMA transfer condition to the sequential operand transfer or the non-stop transfer.

(2) Sequential Operand Transfer

Setting the DMA transfer condition select bits (DSEL) to 01 selects sequential operand transfer.

A single DMA request transfers data in units of the number of data items set in the single operand transfer data count select bits (OPSEL) (single operand transfer). This data transfer is continued till a single DMA transfer ends (i.e., till the byte count reaches 0). Channel arbitration is performed each time the single operand transfer ends. Data transfer on the current channel is automatically continued if there is no DMA request from a higher-priority channel.

In the case that the DMA transfer condition is the sequential operand transfer, even if the input sense mode of DMA request is the level sense, there is no mask period before the byte count becomes 0. Therefore, the DMA transfer of the low-priority channel than the transferring channel cannot start.

(3) Non-Stop Transfer

Setting the DMA transfer condition select bits (DSEL) to 11 selects non-stop transfer.

A single DMA request continuously transfers data till a single DMA transfer ends (i.e., till the byte count reaches 0). During this interval, DMA requests from higher-priority channels are not accepted because channel arbitration is not performed.

In non-stop transfer, the settings of the single operand transfer data count select bits (OPSEL) are invalid and setting of two-dimensional addressing is prohibited.

Table 11.9 List of DMA Transfer Conditions

DMA Transfer Condition Select Bit (DSEL)	DMA Transfer Condition	Remarks
DSEL = 00	Unit operand transfer • Transfers data by the amount specified in the single operand transfer data count select bits (OPSEL) per DMA request. • Performs channel arbitration after a single operand transfer ends.	
DSEL = 01	Sequential operand transfer • Transfer data till the byte count reaches 0 per DMA request. • Performs channel arbitration each time a single operand transfer ends.	
DSEL = 11	Non-stop transfer • Continuously transfers data till the byte count reaches 0 per DMA request. • Does not perform channel arbitration after data transfer is started.	The OPSEL bits are disabled. Setting of two-dimensional addressing is prohibited.

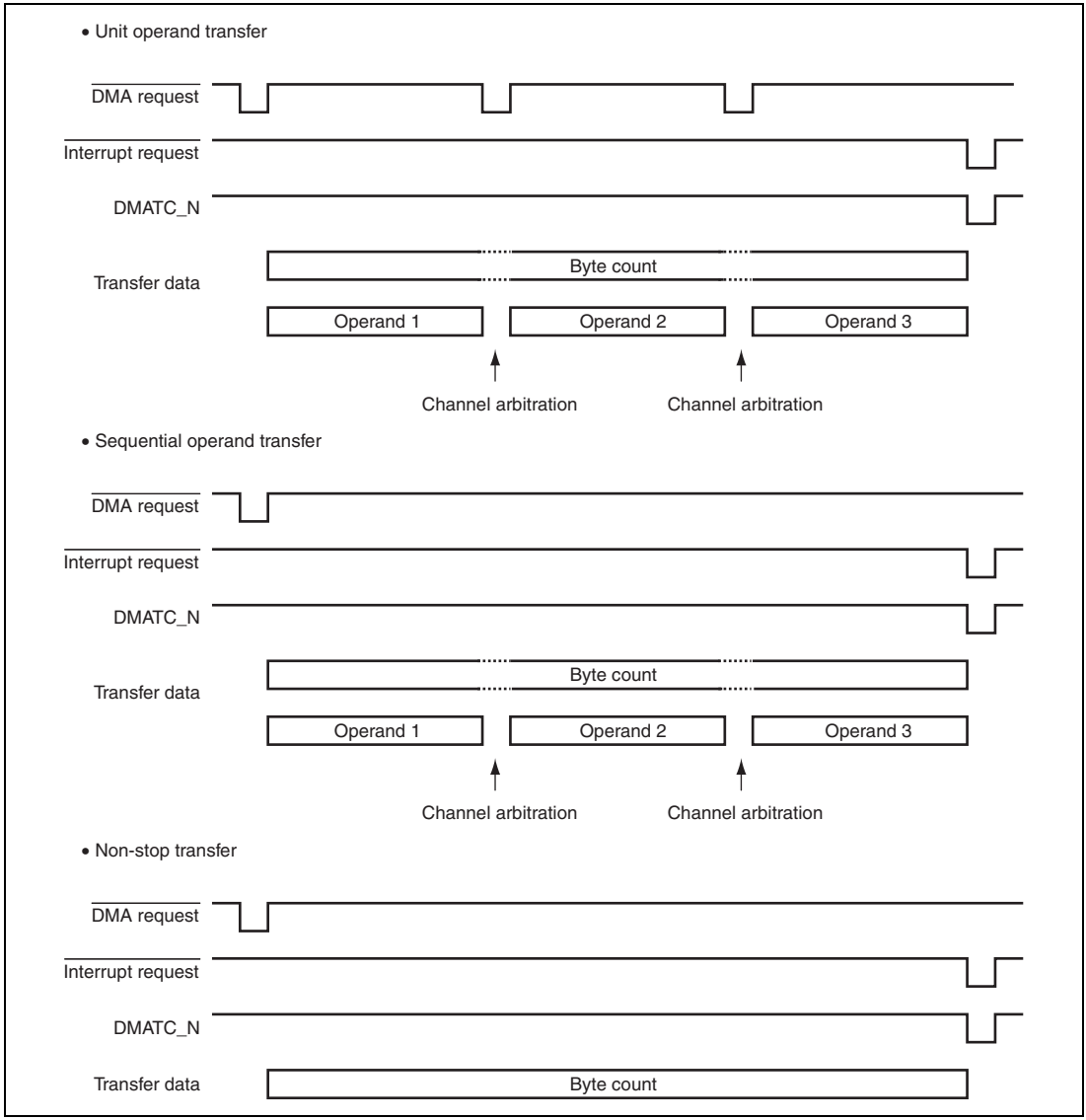


Figure 11.4 DMA Transfer Conditions

Table 11.10 shows the combinations of DMA transfer modes and DMA transfer conditions.

Table 11.10 Combinations of DMA Transfer Modes and DMA Transfer Conditions

		DMA Transfer Condition		
		Unit Operand Transfer DSEL = 00	**Sequential Operand Transfer DSEL = 01**	**Non-Stop Transfer DSEL = 11**
Transfer Mode	Cycle-Stealing Transfer MDSEL = 00	OK (between any two BIUs and the same BIU)	OK (between any two BIUs and the same BIU)	OK (between any two BIUs and the same BIU)
	Piepelined transfer MDSEL = 01	OK (between two different BIUs*2)	OK (between two different BIUs*2)	Partly OK*1 (between two different BIUs, excluding BIU_EIU*2)

Notes: 1. Non-stop transfer to BIU_E in pipelined transfer mode cannot be set.
2. Piepelined transfer to the same BIU is prohibited.

11.4.3 DMA Activation

(1) Initial setting of DMAC

Make the initial setting of each register before setting the DMA transfer enable bit (DEN) to 1. Once data transfer has been started, these settings cannot be changed.

The following shows an example of DMAC initial setting.

1. DMA mode register (DMMODn)
2. DMA control register A (DMCNTAn)
3. DMA control register B (DMCNTBn)
4. DMA current source address register (DMCSADRn)
5. DMA reload source address register (DMRSADRn)
   — When the reload function is used
6. DMA current destination address register (DMCDADRn)
7. DMA reload destination address register (DMRDADRn)
   — When the reload function is used
8. DMA current byte count register (DMCBCTn)
9. DMA reload byte count register (DMRBCTn)
   — When the reload function is used

10. DMA interrupt control register (DMICNT)
    — When an interrupt is used
11. DMA common interrupt control register (DMICNTA)
    — When an interrupt is used
12. DMA transfer enable bit (DEN) of DMA control register B (DMCNTBn)
13. DMA activation control register (DMSCNT)

(2) DMA Activation

To enable DMA transfer on a channel, set the DMA transfer enable bit (DEN) of DMA control register B (DMCNTBn) and the DMAC module activate bit (DMST) of the DMA activation control register (DMSCNT) corresponding to the channel to 1.

If multiple DMA transfer requests are present, channel priorities are judged, the DMA request corresponding to the highest-priority channel is accepted, and DMA transfer on the channel is started.

Whether DMA requests are present can be checked from the DMA request bit (DREQ) of DMA control register B (DMCNTBn).

When a DMA request is accepted and DMA transfer is started, the DMA arbitration status bit (DASTS) of the channel corresponding to the DMA arbitration status register (DMASTS) is set to 1.

(3) DMA Activation Sources and Restrictions

When the DMAC is activated by an on-chip peripheral module, the transfer source, transfer destination, operand size, data size, transfer conditions, and transfer mode (whether pipelined transfer is available) may be fixed. Table 11.11 shows the transfer methods that can be selected for each DMA request source.

Table 11.11 Transfer Methods That Can Be Selected in Each DMA Request Source

Request Source	Transfer Source	Transfer Destination	Operand Size (OPSEL)	Data Size (SZSEL)	Transfer Condition (DSEL)*1	Piepelined Transfer (MDSEL)
Software trigger	No restriction	No restriction	No restriction	No restriction	No restriction	Available
DREQ0 pin	No restriction	No restriction	No restriction	No restriction	No restriction	Available
DREQ1 pin						
DREQ2 pin						
DREQ3 pin						
USB_0	Write: No restriction Read: D0FIFO, D1FIFO	Write: D0FIFO, D1FIFO Read: No restriction	*2	*2	Unit	Not available
USB_1						
CMT_0	No restriction	No restriction	No restriction	No restriction	No restriction	Available
CMT_1						
CMT_2						
CMT_3						
MTU2_0	No restriction	No restriction	No restriction	No restriction	No restriction	Available
MTU2_1						
MTU2_2						
MTU2_3						
MTU2_4						
IIC3_0 reception	ICDRR_0	No restriction	1	8	Unit	Not available
IIC3_1 reception	ICDRR_1					
IIC3_2 reception	ICDRR_2					
IIC3_3 reception	ICDRR_3					

Request Source	Transfer Source	Transfer Destination	Operand Size (OPSEL)	Data Size (SZSEL)	Transfer Condition (DSEL)*1	Piepelined Transfer (MDSEL)
IIC3_0 transmission	No restriction	ICDRT_0	1	8	Unit	Not available
IIC3_1 transmission		ICDRT_1				
IIC3_2 transmission		ICDRT_2				
IIC3_3 transmission		ICDRT_3				
SCIF_0 reception	SCFRDR_0	No restriction	1	8	Unit	Not available
SCIF_1 reception	SCFRDR_1					
SCIF_2 reception	SCFRDR_2					
SCIF_3 reception	SCFRDR_3					
SCIF_4 reception	SCFRDR_4					
SCIF_5 reception	SCFRDR_5					
SCIF_0 transmission	No restriction	SDFTDR_0	1	8	Unit	Not available
SCIF_1 transmission		SDFTDR_1				
SCIF_2 transmission		SDFTDR_2				
SCIF_3 transmission		SDFTDR_3				
SCIF_4 transmission		SDFTDR_4				
SCIF_5 transmission		SDFTDR_5				
SSIF_0 transmission/reception	Reception: SSIFDR_n (n = channel number) Transmission: No restriction	Reception: No restriction Transmission: SSIFDR_n (n = channel number)	1, 2, 4	32	Unit	Not available
SSIF_1 transmission/reception						
SSIF_2 transmission/reception						
SSIF_3 transmission/reception						
SSIF_4 transmission/reception						
SSIF_5 transmission/reception						
SSU_0 reception	SSRDR0_0 to SSRDR3_0	No restriction	1	8, 16	Unit	Not available
SSU_1 reception	SSRDR0_1 to SSRDR3_1					

Request Source	Transfer Source	Transfer Destination	Operand Size (OPSEL)	Data Size (SZSEL)	Transfer Condition (DSEL)*1	Piepelined Transfer (MDSEL)
SSU_0 transmission	No restriction	SSTDR0_0 to SSTDR3_0	1	8, 16	Unit	Not available
SSU_1 transmission		SSTDR0_1 to SSTDR3_1				
A/D converter	ADDR	No restriction	1	16	Unit	Not available
2DG output	No restriction	SE buffer	No restriction	16, 32	Unit, sequential	Available
2DG BLT input A	No restriction	SA buffer	1 to 64			
2DG BLT input B	No restriction	SB buffer	1 to 64			
2DG BLT output C	DC buffer	No restriction	No restriction			
FLCTL_0 transmission/reception	Reception: FLDTFIFO	Reception: No restriction	1, 4, 32	32	Unit	Available
	Transmission: No restriction	Transmission: FLDTFIFO				
FLCTL_1 transmission/reception	Reception: FLDTFIFO	Reception: No restriction				
	Transmission: No restriction	Transmission: FLECFIFO				
RM0_0	RCAN_0 MB0	No restriction	1	8, 16, 32	Unit	Not available
RM0_1	RCAN_1 MB0	No restriction				

Notes: 1. Words in the Transfer Condition (DSEL) have the following meanings:
Unit: Only unit operand transfer can be specified.
Unit, sequential: Unit operand transfer or sequential operand transfer can be specified.
No restriction: Unit operand transfer, sequential operand transfer, or non-stop transfer can be specified.

2. For single data access mode: Operand size = 1, data size = 8, 16, 32
For 16-byte sequential access mode: Set so that operand size × data size = 16 bytes.
For 32-byte sequential access mode: Set so that operand size × data size = 32 bytes.

11.5 DMA Transfer End and Interrupts

11.5.1 DMA Transfer End

When the value in the DMA current byte count register (DMCBCTn) becomes H'000 0000 (transfer end of all data), the DMA transfer end condition is satisfied and one DMA transfer ends.

The following describes the operations performed when the DMA transfer end condition is detected.

- DMA transfer end detection

  The DMA transfer end condition detect bit (DEDET) of the channel corresponding to the DMA transfer end detection register (DMEDET) is set to 1.
- Interrupt request generation

  An interrupt request is generated for the interrupt controller according to the settings of the DMA interrupt control register (DMICNT) and the DMA common interrupt control register (DMICNTA).
- DMA end signal output

  The DMA end signal (DMATC_N) is output according to the setting of the DMA end signal output control bit (DTCM) of the DMA mode register (DMMODn).
- DMA transfer enable bit (DEN) clearing

  If the DMA transfer enable clear bit (ECLR) of DMA control register B (DMCNTBn) is set to 1, the DEN bit of DMA control register B (DMCNTBn) is cleared to 0 and the subsequent DMA transfer of the channel is suspended.

  If the DMA transfer enable clear bit (ECLR) is cleared to 0, the DEN bit is not cleared.
- Source address register reloading

  If the DMA source address reload function enable bit (SRLOD) of DMA control register A (DMCNTAn) is set to 1, the value in the DMA reload source address register (DMRSADRn) is reloaded to the DMA current source address register (DMCSADRn).
- Destination address register reloading

  If the DMA destination address reload function enable bit (DRLOD) of DMA control register A (DMCNTAn) is set to 1, the value in the DMA reload destination address register (DMRDADRn) is reloaded to the DMA current destination address register (DMCDADRn).
- Byte count register reloading

  If the DMA byte count reload function enable bit (BRLOD) of DMA control register A (DMCNTAn) is set to 1, the value in the DMA reload byte count register (DMRBCTn) is reloaded to the DMA current byte count register (DMCBCTn).

Note: If registers are not reloaded, set ECLR = 1 so that the DEN bit is cleared.

11.5.2 DMA Interrupt Requests

The DMAC provides two types of interrupt request signals for the interrupt controller: interrupt request signals per channel (DMINTn_N, n = 0 to 13) and a common interrupt request signal (DMINTA_N) that is a collection of interrupt requests per channel.

Figure 11.5 is a block diagram of generating interrupt request signals per channel and a common interrupt request signal.

If the DMA interrupt control bit (DINTM) of the channel corresponding to the DMA interrupt control register (DMICNT) is set to 1 when a DMA transfer ends, an interrupt request for the corresponding channel is generated.

Only those channels for which the DMA common interrupt request signal control bit (DINTA) of the DMA common interrupt control register (DMICNTA) is set to 1 are collected into one and output as the common interrupt request signal.

The generated interrupt request can be cleared to 0 by writing 1 to the DMA transfer end condition detect bit (DEDET) of the corresponding channel.

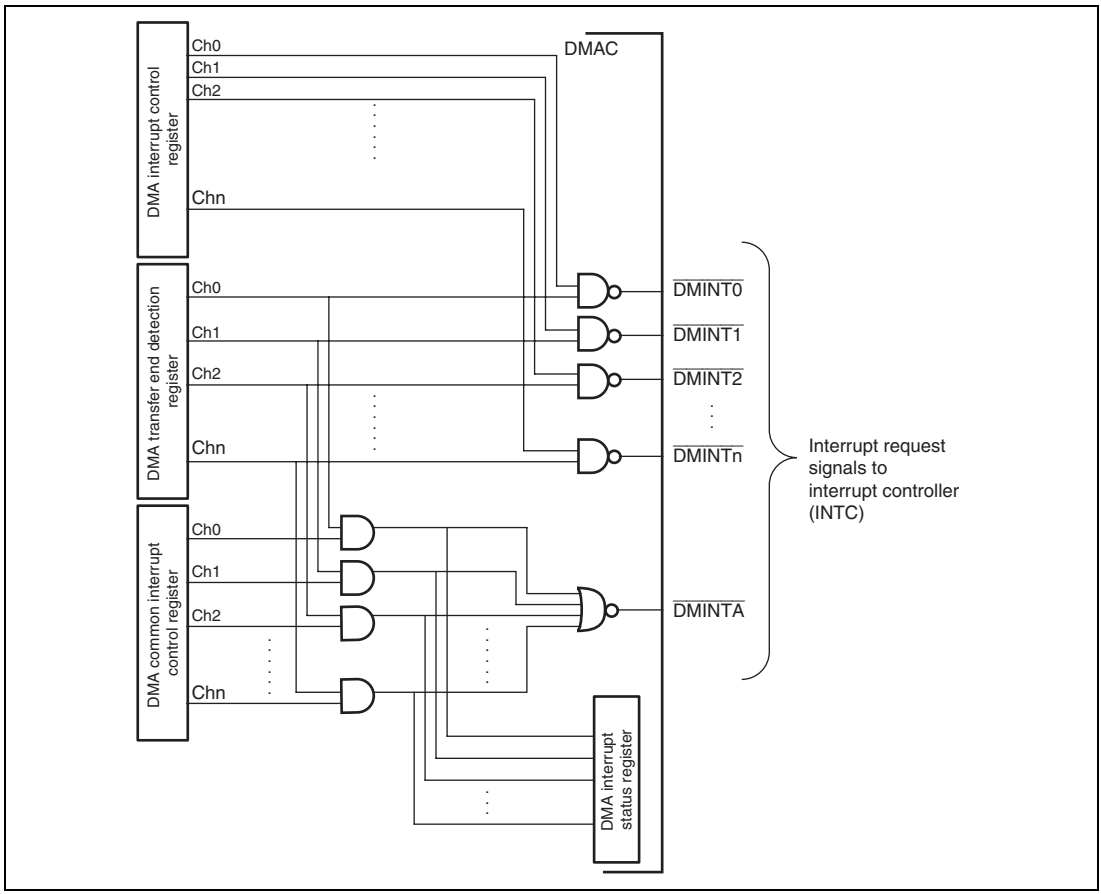


Figure 11.5 Block Diagram of Generating Interrupt Request Signal per Channel and Common Interrupt Request Signal

11.5.3 DMA End Signal Output

(1) LSI Internal Signal (DMATC_N)

When DMA transfer for the amount of data set in the DMA current byte count register (DMCBCTn) ends, the DMAC outputs the DMA end signal (DMATC_N) in the LSI. DMATC_N is the LSI internal signal, so it cannot be monitored from outside the LSI.

DMATC_N output depends on the setting of the DMA end signal output control bits (DTCM) in the DMA mode register (DMMODn) for the corresponding channel.

- If DTCM is set to 00, DMATC_N is not output and fixed at a high level even when DMA transfer ends.
- If DTCM is set to 01, DMATC_N is output in the read cycle immediately before the end of DMA transfer (read cycle of the last unit data transfer).
- If DTCM is set to 10, DMATC_N is output in the write cycle immediately before the end of DMA transfer (write cycle of the last unit data transfer).
- If DTCM is set to 11, the low pulse signal of one clock cycle is output as DMATC_N at the same timing as the DMA transfer end interrupt.

Figure 11.6 shows the output timing of the DMA end signal.

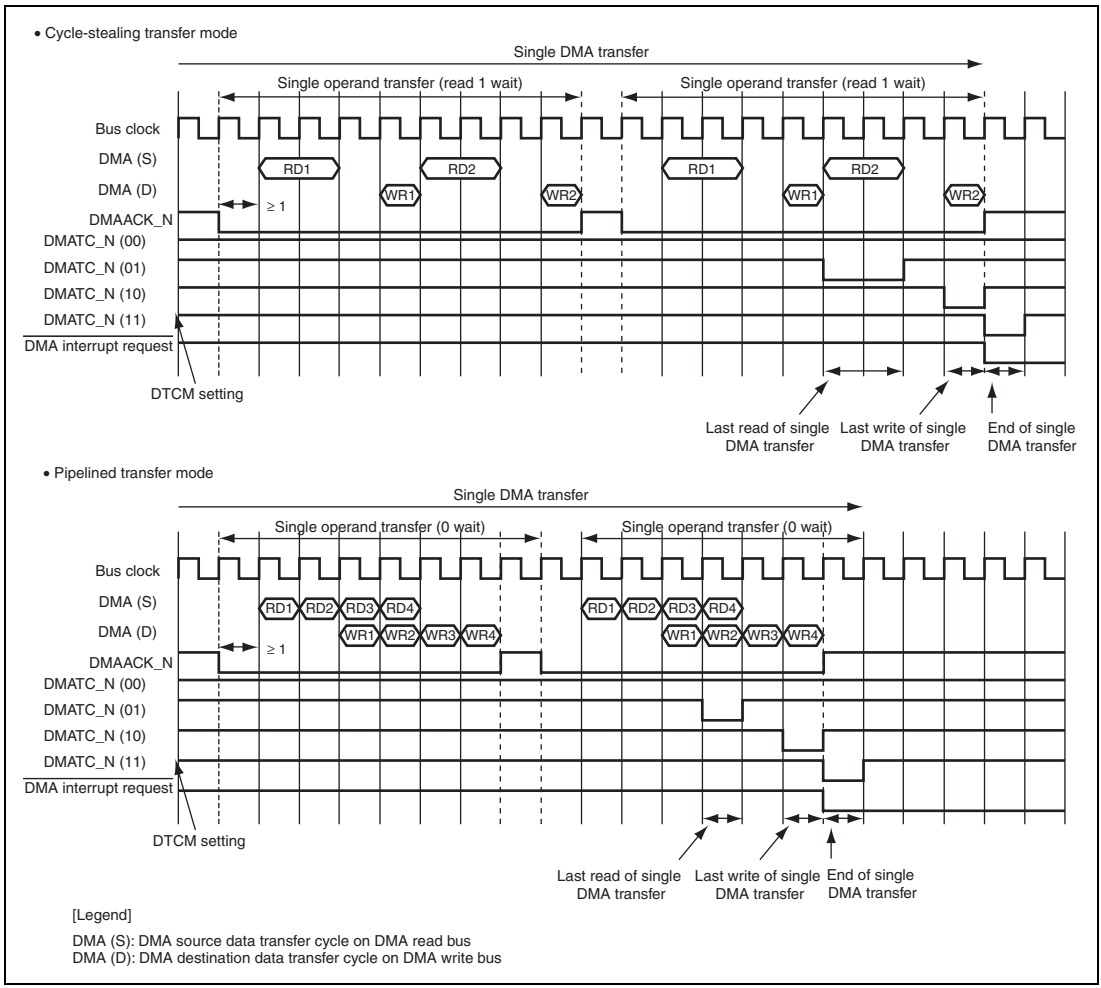


Figure 11.6 Output Timing of DMA End Signal

(2) DMA End Output Signal ($\overline{\text{TENDi}}$, i = 0 to 3)

If DMA transfer by the DREQi request from an external pin is selected, the read or write destination is normal space (CS0 to CS5), and transfer mode is cycle-stealing transfer mode, the DMA end output signal ($\overline{\text{TENDi}}$) can be output to outside the LSI. For access to LSI interior or SDRAM space or when transfer mode is pipelined transfer mode, the $\overline{\text{TENDi}}$ signal is not output.

- To read from normal space by DMA, set the DMA end signal output control bits (DTCM) of the DMA mode register (DMMODn) to select that an end signal is output in the last read cycle (DTCM = 01). $\overline{\text{TENDi}}$ is output when the last one data unit is read by DMA. The $\overline{\text{TENDi}}$ output timing is the same as the DMA-active signal ($\overline{\text{DACTi}}$) timing (see section 11.9, DMA Acknowledge Signal Output and DMA-Active Signal Output, and section 10, Bus State Controller (BSC)). If DTCM is set to other than 01, $\overline{\text{TENDi}}$ is not output.
- To write to normal space by DMA, set the DTCM bits of DMMODn to select that an end signal is output in the last write cycle (DTCM = 10). $\overline{\text{TENDi}}$ is output when the last one data unit is written by DMA. The $\overline{\text{TENDi}}$ output timing is the same as the DMA-active signal ($\overline{\text{DACTi}}$) timing (see section 11.9, DMA Acknowledge Signal Output and DMA-Active Signal Output, and section 10, Bus State Controller (BSC)). If DTCM is set to other than 10, $\overline{\text{TENDi}}$ is not output.

11.6 Suspending, Restarting, and Stopping of DMA Transfer

11.6.1 Suspending and Restarting of DMA Transfer

Clearing the DMAC module activate bit (DMST) of the DMA activation control register (DMSCNT) to 0 enables you to suspend data transfer on all channels of the DMAC. Clearing the DMA transfer enable bit (DEN) of DMA control register B (DMCNTBn) of the corresponding channel to 0 also enables you to suspend data transfer on the channel.

If the DMST or DEN bit is cleared to 0 when single operand transfer is in progress in the unit operand transfer condition or sequential operand transfer condition, DMA transfer is suspended after single operand transfer has ended without reference to each transfer mode (cycle-stealing mode or pipelined transfer mode).

If the DMST or DEN bit is cleared to 0 when DMA transfer is in progress in the non-stop transfer condition, DMA transfer is not suspended and continues till the DMA transfer end condition is detected (i.e., till the byte count reaches 0).

To restart the suspended channel, set the cleared DMST and DEN bits to 1 to restart DMA transfer.

11.6.2 Stopping of DMA Transfer on Any Channel

To stop DMA transfer on any channel, suspend DMA transfer on that channel and write 1 to the DMAC internal state clear bit (DSCLR) of DMA control register B (DMCNTBn) to initialize the DMAC interior. In this case, only the internal state of the DMAC internal circuit is initialized; each register is not initialized.

11.7 DMA Requests

11.7.1 DMA Request Sources

DMA request sources include software triggers and DMA request signal inputs. DMA request signal input sources are selected from the DMA request source select bits (DCTG) of DMA control register A (DMCNTAn) of each channel.

11.7.2 Synchronization Circuits for DMA Request Signal Inputs

Each DMAC channel is provided with a synchronization circuit to cope with asynchronously input DMA request signals. Therefore, a blank period of a few clock cycles appears during the period from the point when a DMA request signal input such as DREQ0 to DREQ3 goes active until the request is reflected in the DMA request bit (DREQ) in DMA control register B (DMCNTBn).

Figure 11.7 shows an example of the DMA request bit timing of DMA request signal input.

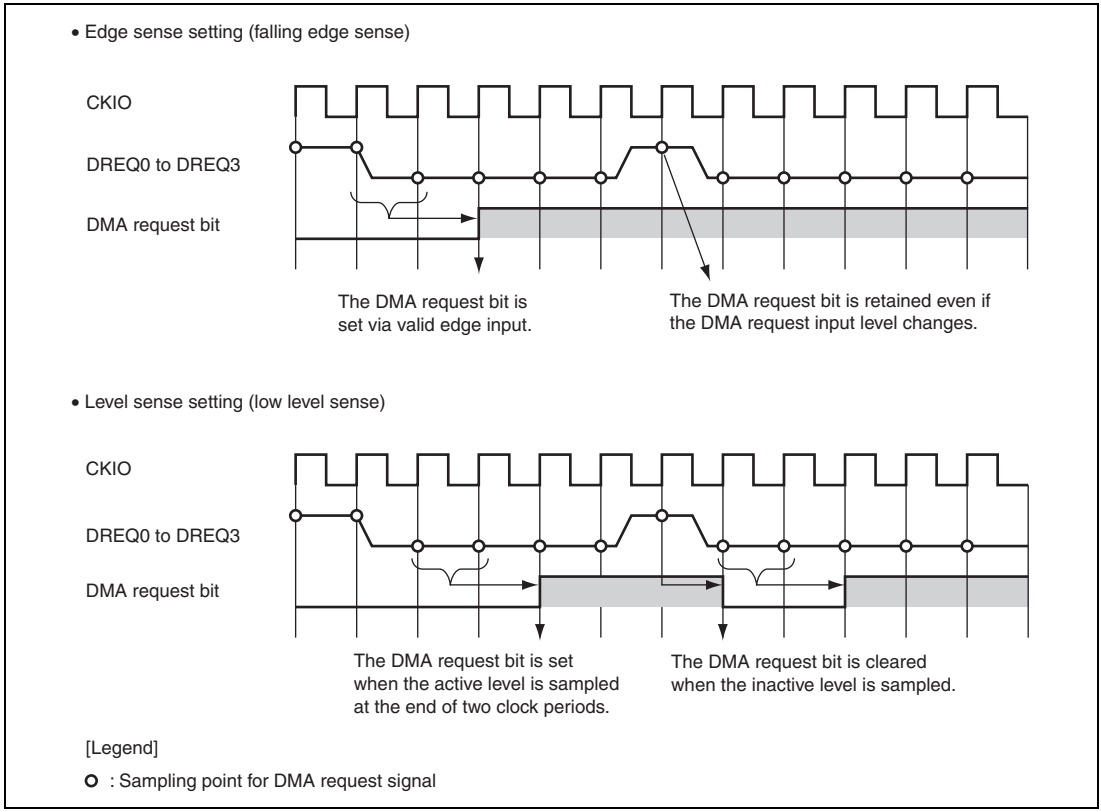


Figure 11.7 Example of DMA Request Bit Timing for DMA Request Signal Input

11.7.3 Sense Mode for DMA Requests

If the DREQ0 to DREQ3 pins (DCTG = 000001 to 000100) are selected by the DMA request source select bits (DCTG), a level sense (01 or 11) or an edge sense (00 or 10) can be selected from the input sense mode select bits (STRG) of DMA control register A (DMCNTAn).

If the software trigger (DCTG = 000000) is selected as the DMA request source, select the rising-edge sense (00). If the DMA request from the on-chip peripheral modules (DCTG = 000101 to 111001) is selected as the DMA request source, set the sense mode show in table 11.8.

The following describes the level sense and edge sense.

(1) Level Sense

If a level sense (STRG = 01 or 11) is selected, whether a DMA request is present is judged from the DMA request signal level. A DMA request is not retained in the DMAC, so retain the DMA request signal level till the acceptance of the DMA request is confirmed.

Figure 11.8 is an example of DMA request acceptance processing when a level sense is selected.

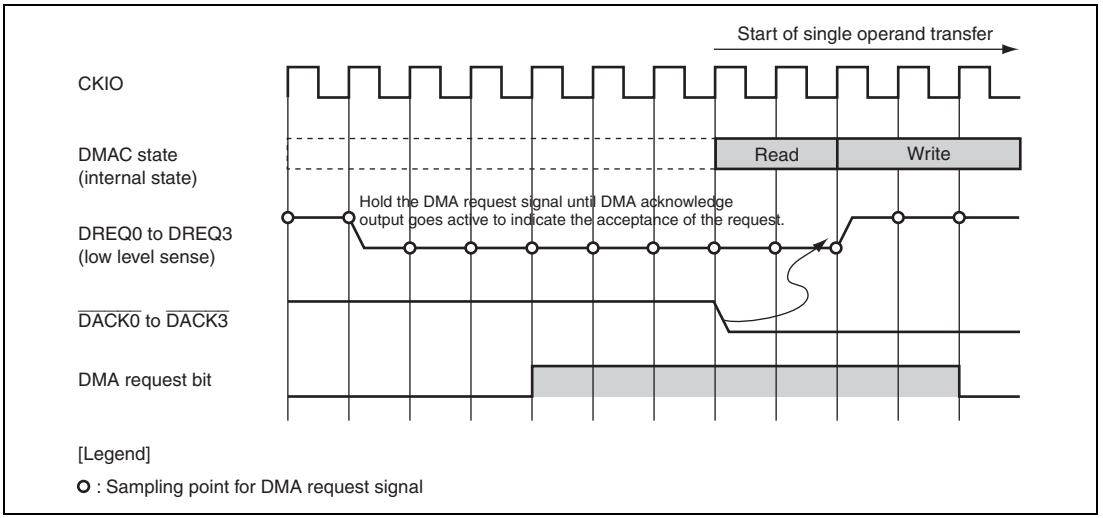


Figure 11.8 Example of DMA Request Acceptance Processing When a Level Sense Is Selected

If a level sense is selected, the DMA request signal of the channel is masked from the start of the last write access in a single operand transfer until two clock cycles after the single operand transfer has ended. This ensures a sufficient length of time for negation of the DMA request signal.

Figure 11.9 shows the DMA request signal mask period when a level sense is selected.

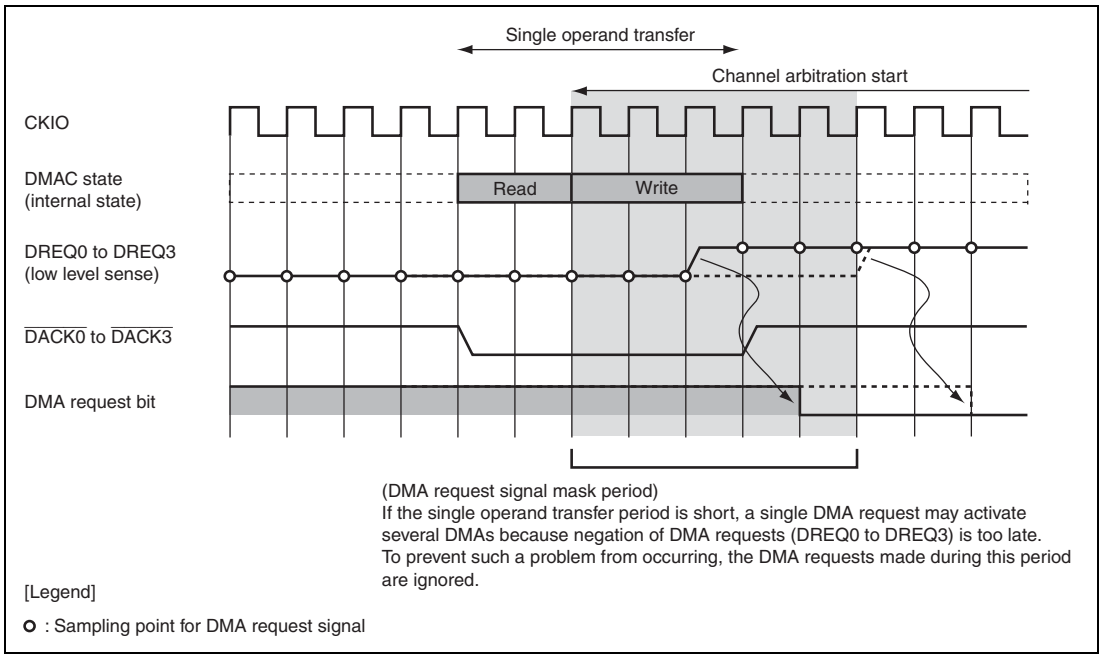


Figure 11.9 DMA Request Signal Mask Period When a Level Sense Is Selected

Therefore, even if a channel for which a level sense is selected retains the DMA request signal level (continues to request DMA transfer) as is after the DMA request is accepted, DMA requests from other channels are accepted, if any, because it is judged that no DMA request is made during the DMA request signal mask period.

For sequential operand transfer, however, the DMA request signal mask period becomes effective only if operand transfer ends when the byte count reaches 0. If operand transfer ends when the byte count is not 0, channel arbitration is performed without the DMA request masked. For non-stop transfer, this mask period becomes effective if DMA transfer ends when the byte count is 0.

If DMA transfer is not continuously performed, the DMA request must be canceled within three cycles after single operand transfer has ended.

(2) Edge Sense

If an edge sense (STRG = 00 or 10) is selected, transition to a rising or falling edge of a DMA request signal is recognized as a DMA request.

When a valid edge is detected, the DMA request bit (DREQ) of DMA control register B (DMCNTBn) is set to 1. The value of this bit is retained even if the input level of the DMA request signal changes later. When a DMA request is accepted and the DMA acknowledge signal is effectively output, the DMA request bit (DREQ) bit is automatically cleared to 0.

In this way, retention of DMA requests in edge sense mode is determined from the value of the DMA request bit (DREQ). For this reason, if the DMA request bit (DREQ) is set to 1, the edges selected according to new DMA request signals are ignored. Figure 11.10 shows an example of DMA acceptance processing when an edge sense is selected.

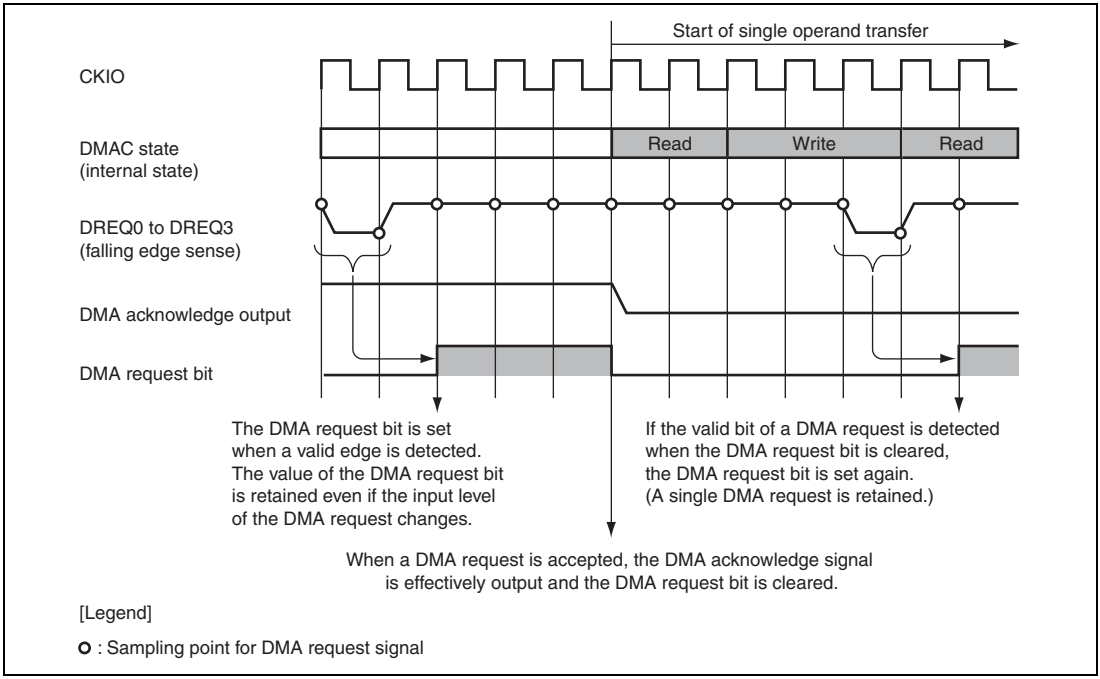


Figure 11.10 Example of DMA Request Acceptance Processing When an Edge Sense Is Selected

11.8 Determination of DMA Channel Priorities

11.8.1 Channel Priorities

Channel 0 has the highest priority. The priorities of channels are fixed in the following order:

Channel 0 > channel 1 > channel 2 > … > channel 12 > channel 13

11.8.2 Operation at Occurrence of Multiple DMA Requests

The DMAC determines DMA channel priorities per single operand transfer.

If a DMA request with a higher channel priority occurs during operand transfer on one channel, operand transfer on the higher-priority channel is started after operand transfer on the current channel has ended. Figure 11.11 shows an example of DMAC outline operation when multiple DMA requests occur. The thick lines in figure 11.11 indicate the period during which DMA request signals are at a low level (channel 0 (ch0), channel 2 (ch2), and channel 3 (ch3) are set to a level sense and channel 1 (ch1) is set to an edge sense).

1. Transfer on channel 3 is started because DMA requests are assumed to be non-existent in channel 2 because channel 2 is during the mask period.
2. Transfer on channel 0 is started because this channel has the highest channel priority.
3. Transfer on channel 2 is started because this channel has the highest channel priority at this point.
4. Transfer on channel 3 is started because there are no other requests at this point.
5. If DMA requests for channel 0, channel 1, and channel 3 occur at the same time, transfer on channel 0 is started because channel 0 has the highest priority.
6. When transfer on channel 0 ends, transfer on channel 1 is started because channel 1 has the second highest priority.
7. If a DMA request (low or high level request edge) occurs during DMA transfer on channel 1, DMA transfer on channel 1 is started again after DMA transfer on channel 1 ends. If an edge sense is set, no mask period exists.
8. When DMA transfer on channel 1 ends, DMA transfer on channel 3 is started because there is no other transfer request.
9. When channel 3 is during the mask period, DMA transfer is not started because there is no other transfer request. DMA transfer on channel 3 is started after the mask period ends.

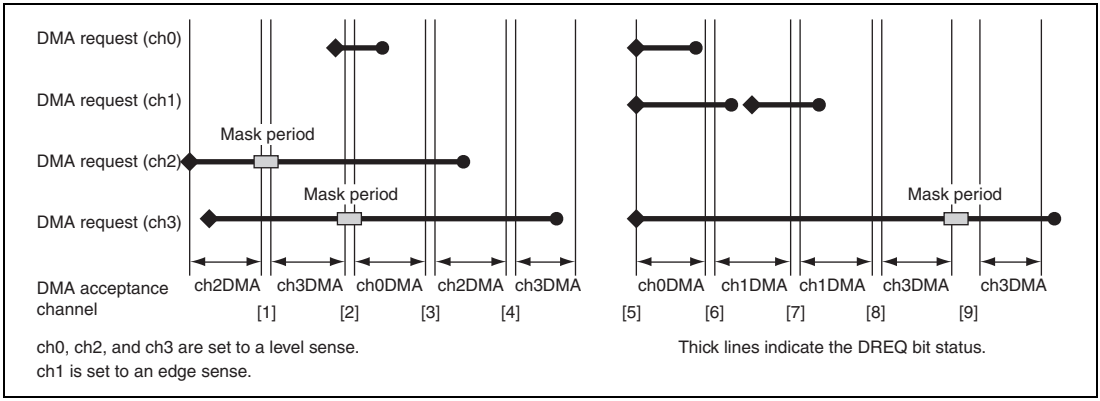


Figure 11.11 Example of Outline Operation When Multiple DMA Requests Occur

11.9 DMA Acknowledge Signal Output and DMA-Active Signal Output

(1) LSI Internal Signals

The DMAC outputs the DMA acknowledge signal (DMAACK_N) and the DMA-active signals (DMAACTS_N/DMAACTD_N) for the source and destination when a DMA request is accepted or DMA transfer is performed. These signals are LSI internal signals, so they cannot be monitored from outside the LSI. An on-chip peripheral module that requested DMA transfer recognizes that a DMA transfer request has been accepted and DMA transfer is being performed by monitoring these signals.

- DMA-active signals

  Outputs of DMAACTS_N and DMAACTD_N are enabled by setting the DMA-active signal output control bits (SACT/DACT) for source and destination in the DMA mode register of the corresponding channel.

  If SACT is set to 1, an active DMAACTS_N signal is output when read access is made.

  If DACT is set to 1, an active DMAACTD_N signal is output when write access is made.
- DMA acknowledge signal

  DMAACK_N is output during the period from start of single operand transfer to end of single operand transfer.

Figure 11.12 shows the output timing of the DMA acknowledge signal and DMA-active signals.

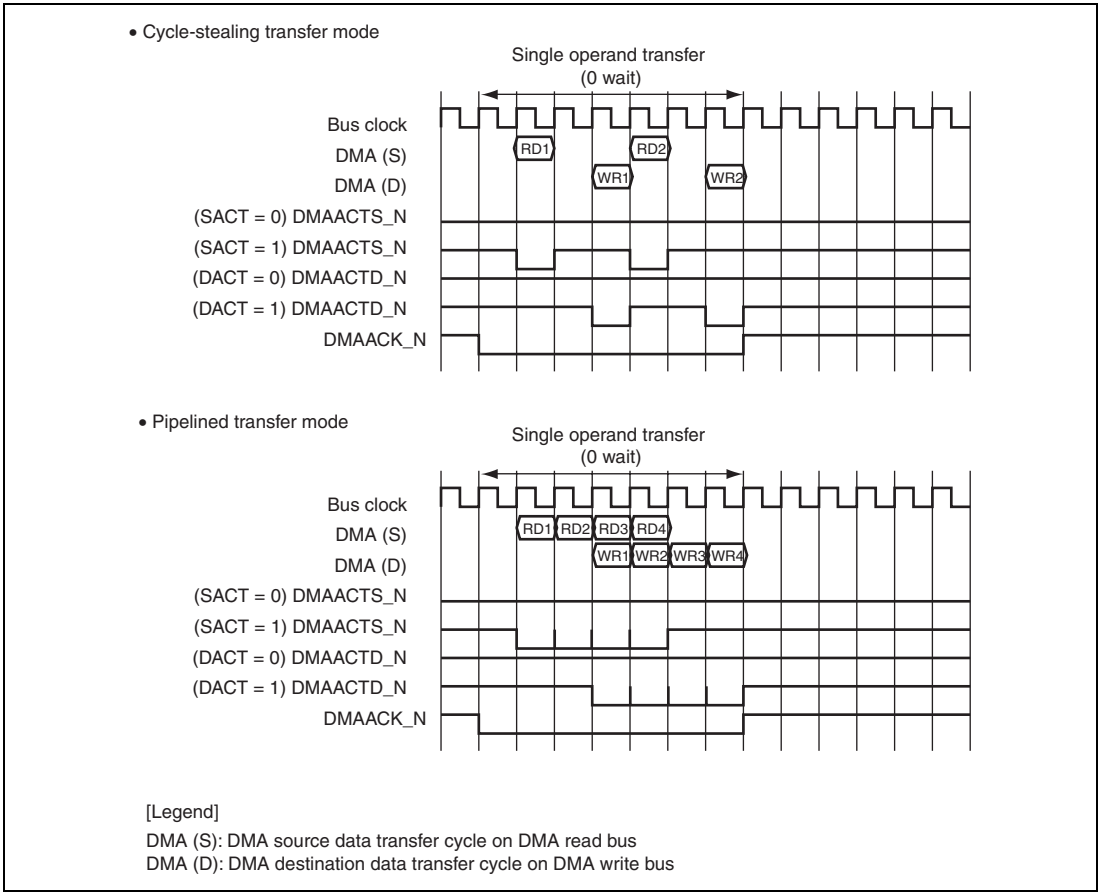


Figure 11.12 Output Timing of DMA Acknowledge Signal and DMA-active signals

(2) DMA Active Output Signal ($\overline{\text{DACT}}$, i = 0 to 3)

If DMA transfer is in response to the DREQi request from an external pin and the read or write destination is normal space (CS0 to CS5), the DMA active signal ($\overline{\text{DACTi}}$) can be output to outside the LSI. If the access destination is LSI interior or SDRAM space, the $\overline{\text{DACTi}}$ signal is not output.

- To read from normal space by DMA, set the source DMA-active signal output control bit (SACT) of the DMA mode register (DMMODn) to 1. $\overline{\text{DACTi}}$ is output when normal space is read by DMA. For the $\overline{\text{DACTi}}$ output timing, see section 10, Bus State Controller (BSC). If SACT is cleared to 0, $\overline{\text{DACTi}}$ is not output when normal space is read by DMA.
- To write to normal space by DMA, set the destination DMA-active signal output control bit (DACT) of the DMA module register (DMMODn) to 1. $\overline{\text{DACTi}}$ is output when normal space is written by DMA. See section 10, Bus State Controller (BSC). If DACT is cleared to 0, $\overline{\text{DACTi}}$ is not output when normal space is written by DMA.
- If both SACT and DACT are set to 1 in DMA transfer from one normal space to another, $\overline{\text{DACTi}}$ is output when normal space is read or written by DMA.

(3) DMA Acknowledge Output Signal ($\overline{\text{DACKi}}$, i = 0 to 3)

If DMA transfer is performed in response to the DREQi request from an external pin, the DMA acknowledge signal ($\overline{\text{DACKi}}$) is output. $\overline{\text{DACKi}}$ is output with the same timing as DMAACK_N that is the DMA acknowledge signal in the LSI.

Note: The $\overline{\text{DACKi}}$ signal indicates the DMA operation timing in the LSI. If normal space or SDRAM space is written by DMA, DMA write access observed outside the LSI may be delayed several cycles compared with DMA write access in the LSI. In this case, DMA write access may be observed outside the LSI after $\overline{\text{DACKi}}$ is negated.

11.10 Units of Transfer and Transfer Byte Positions

The bit size for single data transfer can be set to a byte (8 bits), word (16 bits), or longword (32 bits).

Figure 11.13 is an example of DMA data-byte control for 32-bit bus width.

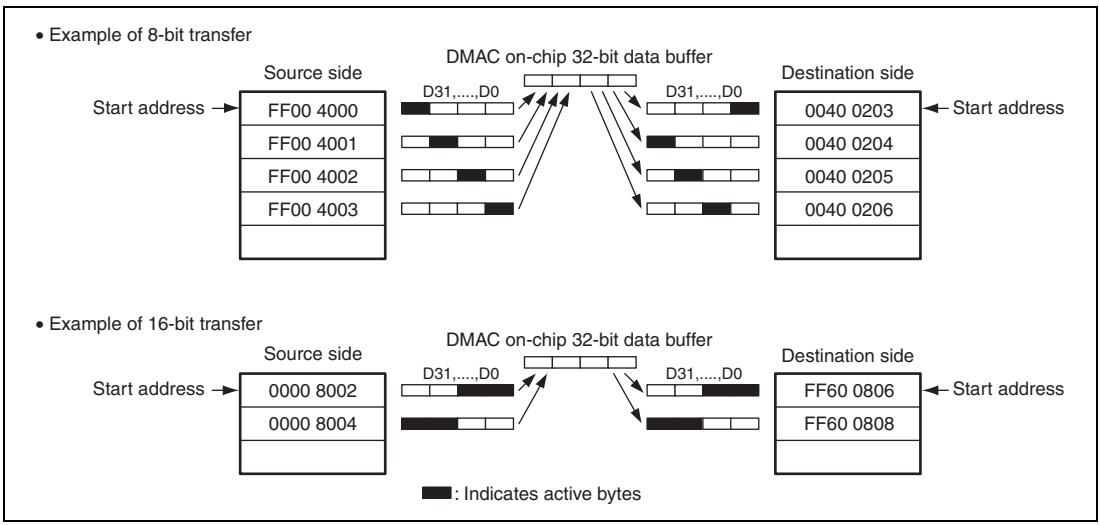


Figure 11.13 Example of DMA Data-Byte Control for 32-Bit Bus Width

11.11 Reload Function

To set the reload function, set each reload function enable bit of DMA control register A (DMCNTAn) per channel and per transfer parameter (source address, destination address or byte count).

To configure the reloading function on a channel supporting two-dimensional addressing, set the two-dimensional reload function enable bit (2DRLOD) of DMA control register A (DMCNTAn) for the corresponding channel. This enables reloading of the six registers related to two-dimensional addressing: column setting register, row setting register, block setting register, next row offset register, next block offset register, and next line offset register.

When the DMA transfer end condition is detected, DMA transfer parameters are automatically reloaded.

(1) Reload Register and Current Register

When not using the reload function, set data in the current register. When using the reload function, set data both in the reload register and current register.

Do not write data to the current register during single operand transfer. If data is written to the current register, operation is not guaranteed. The reload register can be set even during single operand transfer but this setting must be made before start of the last operand transfer (DMA transfer end). If the reload register is set after start of the last operand transfer, however, this setting may have not been reflected when data is reloaded after DMA transfer end.

(2) Continuous Transfer to Dispersed Areas

The reload function provides continuous transfer to dispersed areas. Writing values to the DMA reload source and destination address registers (DMRSADRn/DMRDADRn) and DMA reload byte count register (DMRBCTn) before DMA transfer ends enables you to prepare the next transfer parameters without affecting the current DMA transfer (current register). This enables you to continuously transfer several transfer blocks in different transfer areas and with a different number of bytes through the same channel.

Figure 11.14 shows an example of how the reload function transfers blocks between dispersed areas.

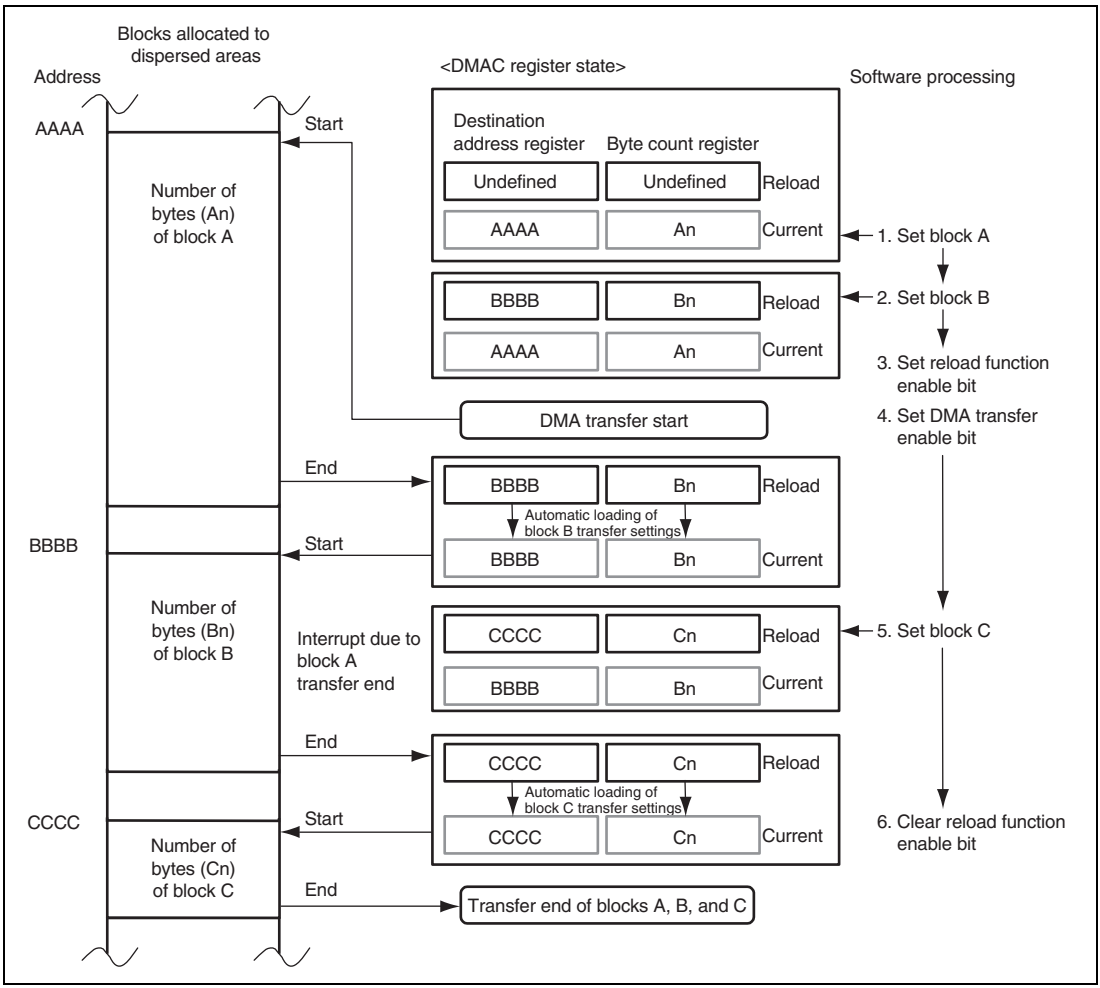


Figure 11.14 Example of Transferring Blocks between Dispersed Areas by Using Reload Function

11.12 Rotate Function

When rotation is selected in addressing mode, the address is incremented. After single operand transfer has ended, the contents of the working source address register or working destination address register for which rotation is set serve as the values in the DMA current source address register (DMCSADRn) or DMA current destination address register (DMCDADRn) set when DMA transfer was started.

Figure 11.15 is an example of transfer that uses the rotate function (source: rotation, destination: increment).

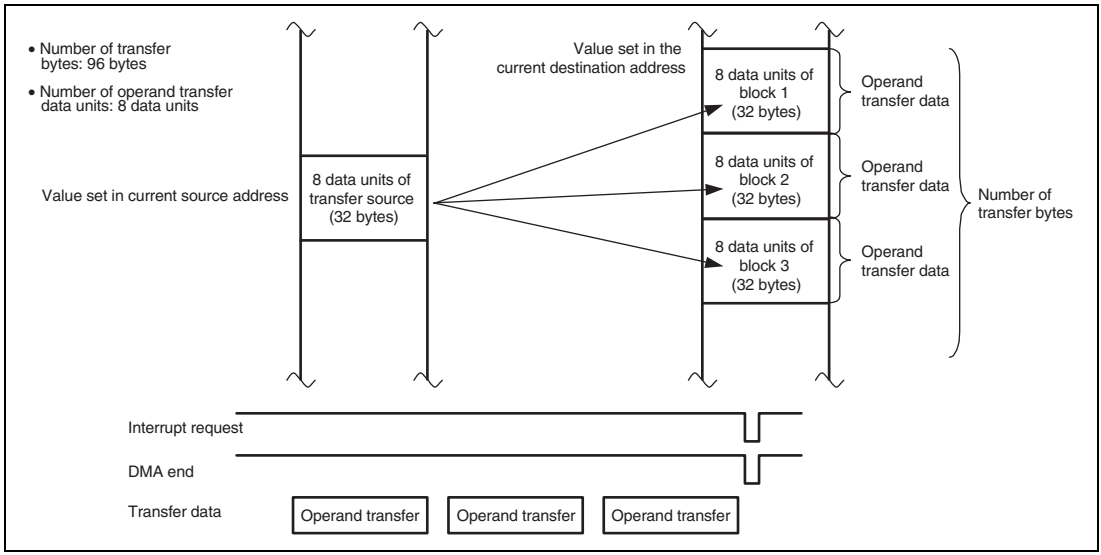


Figure 11.15 Example of Transfer That Uses Rotate Function (Source: Rotation, Destination: Increment)

11.13 Usage Note

11.13.1 Note on Transition to Software Standby Mode or Deep Standby Mode

If the SLEEP instruction for transition to software standby mode or deep standby mode during transfer by the DMAC, DMA transfer is not guaranteed because the DMAC stops without waiting for transfer end. Therefore, when making transition to software standby mode or deep standby mode, wait till DMA transfer ends or stop DMA transfer and then execute the SLEEP instruction. Also when changing the PLL multiplication rate, stop DMA transfer in advance.

Section 12 Multi-Function Timer Pulse Unit 2 (MTU2)

This LSI has an on-chip multi-function timer pulse unit 2 (MTU2) that comprises five 16-bit timer channels.

12.1 Features

- Maximum 16 pulse input/output lines
- Selection of eight counter input clocks for each channel
- The following operations can be set:
  - 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
- 28 interrupt sources
- Automatic transfer of register data
- A/D converter start trigger can be generated
- Module standby mode can be settable
- A total of six-phase waveform output, which includes complementary PWM output, and positive and negative phases of reset PWM output by interlocking operation of channels 3 and 4, is possible.
- AC synchronous motor (brushless DC motor) drive mode using complementary PWM output and reset PWM output is settable by interlocking operation of channels 0, 3, and 4, and the selection of two types of waveform outputs (chopping and level) is possible.
- In complementary PWM mode, interrupts at the crest and trough of the counter value and A/D converter start triggers can be skipped.

Table 12.1 MTU2 Functions

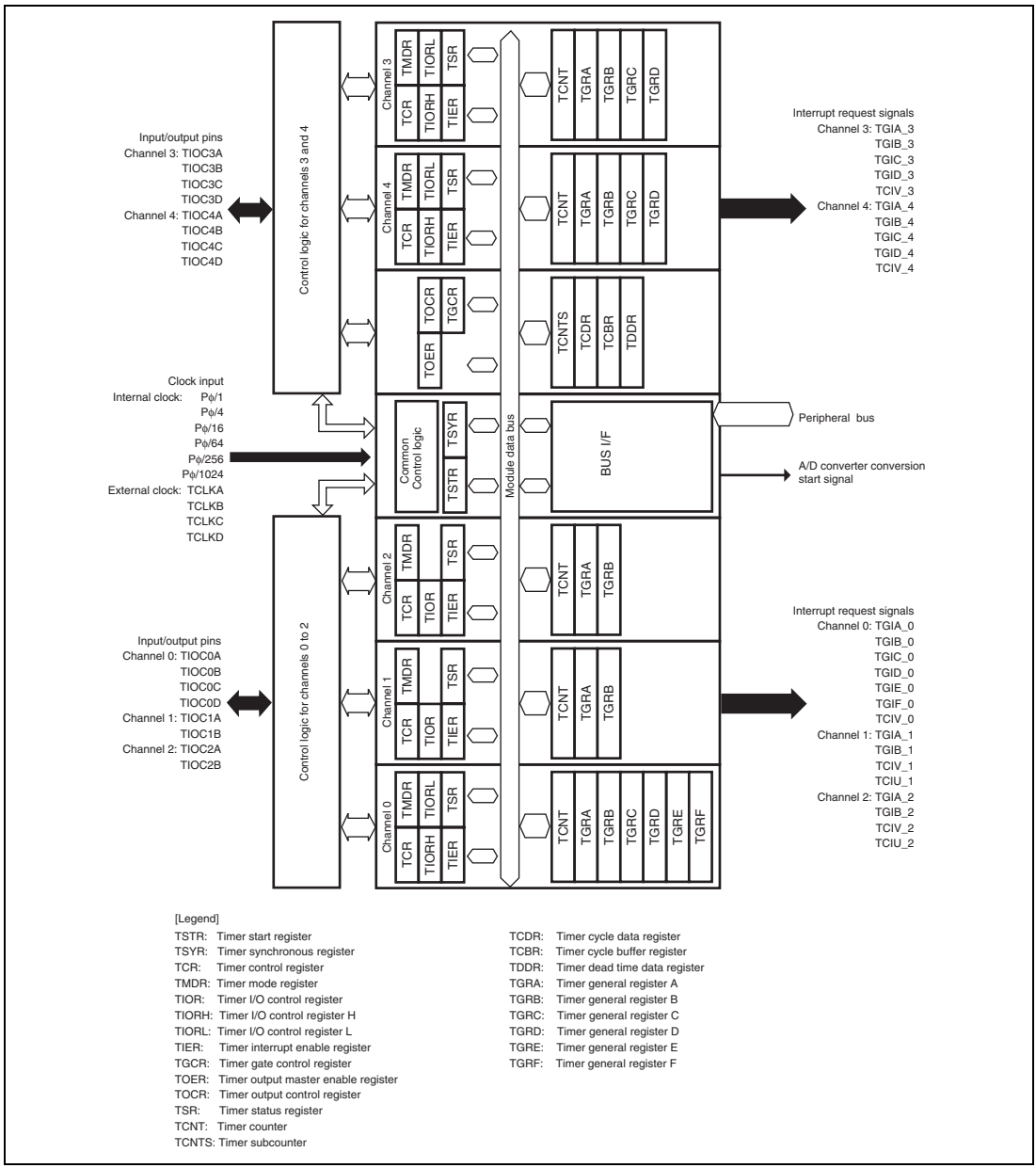
Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
Count clock		Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 TCLKA TCLKB TCLKC TCLKD	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/1024 TCLKA TCLKB TCLKC	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 Pϕ/1024 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 Pϕ/1024 TCLKA TCLKB
General registers		TGRA_0 TGRB_0 TGRE_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 TGRF_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	√	√	√	√	√
	1 output	√	√	√	√	√
	Toggle output	√	√	√	√	√
Input capture function		√	√	√	√	√
Synchronous operation		√	√	√	√	√
PWM mode 1		√	√	√	√	√
PWM mode 2		√	√	√	—	—
Complementary PWM mode		—	—	—	√	√
Reset PWM mode		—	—	—	√	√
AC synchronous motor drive mode		√	—	—	√	√

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
Phase counting mode	—	√	√	—	—
Buffer operation	√	—	—	√	√
DMAC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow or underflow
A/D converter start trigger	TGRA_0 compare match or input capture TGRE_0 compare match	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 TCNT_4 underflow (trough) in complementary PWM mode
Interrupt sources	7 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 • Compare match 0E • Compare match 0F • 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 or underflow

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
A/D converter start request delaying function	—	—	—	—	• A/D converter start request at a match between TADCORA_4 and TCNT_4 • A/D converter start request at a match between TADCORB_4 and TCNT_4
Interrupt skipping function	—	—	—	• Skips TGRA_3 compare match interrupts	• Skips TCIV_4 interrupts

[Legend]
√: Available
—: Not available

Figure 12.1 shows a block diagram of the MTU2.



[Legend]

TSTR: Timer start register
TSYR: Timer synchronous register
TCR: Timer control register
TMDR: Timer mode register
TIOR: Timer I/O control register
TIORH: Timer I/O control register H
TIORL: Timer I/O control register L
TIER: Timer interrupt enable register
TGCR: Timer gate control register
TOER: Timer output master enable register
TOCR: Timer output control register
TSR: Timer status register
TCNT: Timer counter
TCNTS: Timer subcounter
TCDR: Timer cycle data register
TCBR: Timer cycle buffer register
TDDR: Timer dead time data register
TGRA: Timer general register A
TGRB: Timer general register B
TGRC: Timer general register C
TGRD: Timer general register D
TGRE: Timer general register E
TGRF: Timer general register F

Figure 12.1 Block Diagram of MTU2

12.2 Input/Output Pins

Table 12.2 Pin Configuration

Channel	Pin Name	I/O	Function
Common	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

Note: For the pin configuration in complementary PWM mode, see table 12.54.

12.3 Register Descriptions

The MTU2 has the following registers. For details on register addresses and register states during each process, refer to section 32, List of Registers. To distinguish registers in each channel, an underscore and the channel number are added as a suffix to the register name; TCR for channel 0 is expressed as TCR_0.

Table 12.3 Register Descriptions

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
0	Timer control register_0	TCR_0	R/W	H'00	H'FFFE2300	8
	Timer mode register_0	TMDR_0	R/W	H'00	H'FFFE2301	8
	Timer I/O control register H_0	TIORH_0	R/W	H'00	H'FFFE2302	8
	Timer I/O control register L_0	TIORL_0	R/W	H'00	H'FFFE2303	8
	Timer interrupt enable register_0	TIER_0	R/W	H'00	H'FFFE2304	8
	Timer status register_0	TSR_0	R/W	H'C0	H'FFFE2305	8
	Timer counter_0	TCNT_0	R/W	H'0000	H'FFFE2306	16
	Timer general register A_0	TGRA_0	R/W	H'FFFF	H'FFFE2308	16
	Timer general register B_0	TGRB_0	R/W	H'FFFF	H'FFFE230A	16
	Timer general register C_0	TGRC_0	R/W	H'FFFF	H'FFFE230C	16
	Timer general register D_0	TGRD_0	R/W	H'FFFF	H'FFFE230E	16
	Timer general register E_0	TGRE_0	R/W	H'FFFF	H'FFFE2320	16
	Timer general register F_0	TGRF_0	R/W	H'FFFF	H'FFFE2322	16
	Timer interrupt enable register2_0	TIER2_0	R/W	H'00	H'FFFE2324	8
	Timer status register2_0	TSR2_0	R/W	H'C0	H'FFFE2325	8
	Timer buffer operation transfer mode register_0	TBTM_0	R/W	H'00	H'FFFE2326	8
1	Timer control register_1	TCR_1	R/W	H'00	H'FFFE2380	8
	Timer mode register_1	TMDR_1	R/W	H'00	H'FFFE2381	8
	Timer I/O control register_1	TIOR_1	R/W	H'00	H'FFFE2382	8
	Timer interrupt enable register_1	TIER_1	R/W	H'00	H'FFFE2384	8
	Timer status register_1	TSR_1	R/W	H'C0	H'FFFE2385	8

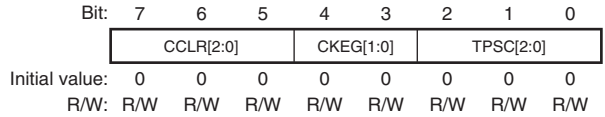
Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
1	Timer counter_1	TCNT_1	R/W	H'0000	H'FFFE2386	16
	Timer general register A_1	TGRA_1	R/W	H'FFFF	H'FFFE2388	16
	Timer general register B_1	TGRB_1	R/W	H'FFFF	H'FFFE238A	16
	Timer input capture control register	TICCR	R/W	H'00	H'FFFE2390	8
2	Timer control register_2	TCR_2	R/W	H'00	H'FFFE2000	8
	Timer mode register_2	TMDR_2	R/W	H'00	H'FFFE2001	8
	Timer I/O control register_2	TIOR_2	R/W	H'00	H'FFFE2002	8
	Timer interrupt enable register_2	TIER_2	R/W	H'00	H'FFFE2004	8
	Timer status register_2	TSR_2	R/W	H'C0	H'FFFE2005	8
	Timer counter_2	TCNT_2	R/W	H'0000	H'FFFE2006	16
	Timer general register A_2	TGRA_2	R/W	H'FFFF	H'FFFE2008	16
	Timer general register B_2	TGRB_2	R/W	H'FFFF	H'FFFE200A	16
3	Timer control register_3	TCR_3	R/W	H'00	H'FFFE2200	8
	Timer mode register_3	TMDR_3	R/W	H'00	H'FFFE2202	8
	Timer I/O control register H_3	TIORH_3	R/W	H'00	H'FFFE2204	8
	Timer I/O control register L_3	TIORL_3	R/W	H'00	H'FFFE2205	8
	Timer interrupt enable register_3	TIER_3	R/W	H'00	H'FFFE2208	8
	Timer status register_3	TSR_3	R/W	H'C0	H'FFFE222C	8
	Timer counter_3	TCNT_3	R/W	H'0000	H'FFFE2210	16
	Timer general register A_3	TGRA_3	R/W	H'FFFF	H'FFFE2218	16
	Timer general register B_3	TGRB_3	R/W	H'FFFF	H'FFFE221A	16
	Timer general register C_3	TGRC_3	R/W	H'FFFF	H'FFFE2224	16
	Timer general register D_3	TGRD_3	R/W	H'FFFF	H'FFFE2226	16
	Timer buffer operation transfer mode register_3	TBTM_3	R/W	H'00	H'FFFE2238	8
4	Timer control register_4	TCR_4	R/W	H'00	H'FFFE2201	8
	Timer mode register_4	TMDR_4	R/W	H'00	H'FFFE2203	8
	Timer I/O control register H_4	TIORH_4	R/W	H'00	H'FFFE2206	8
	Timer I/O control register L_4	TIORL_4	R/W	H'00	H'FFFE2207	8

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
4	Timer interrupt enable register_4	TIER_4	R/W	H'00	H'FFFE2209	8
	Timer status register_4	TSR_4	R/W	H'C0	H'FFFE222D	8
	Timer counter_4	TCNT_4	R/W	H'0000	H'FFFE2212	16
	Timer general register A_4	TGRA_4	R/W	H'FFFF	H'FFFE221C	16
	Timer general register B_4	TGRB_4	R/W	H'FFFF	H'FFFE221E	16
	Timer general register C_4	TGRC_4	R/W	H'FFFF	H'FFFE2228	16
	Timer general register D_4	TGRD_4	R/W	H'FFFF	H'FFFE222A	16
	Timer buffer operation transfer mode register_4	TBTM_4	R/W	H'00	H'FFFE2239	8
	Timer A/D converter start request control register	TADCR	R/W	H'0000	H'FFFE2240	16
	Timer A/D converter start request cycle set register A_4	TADCORA_4	R/W	H'FFFF	H'FFFE2244	16
	Timer A/D converter start request cycle set register B_4	TADCORB_4	R/W	H'FFFF	H'FFFE2246	16
	Timer A/D converter start request cycle set buffer register A_4	TADCOBRA_4	R/W	H'FFFF	H'FFFE2248	16
	Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	R/W	H'FFFF	H'FFFE224A	16
Common	Timer start register	TSTR	R/W	H'00	H'FFFE2280	8
	Timer synchronous register	TSYR	R/W	H'00	H'FFFE2281	8
	Timer read/write enable register	TRWER	R/W	H'01	H'FFFE2284	8

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Common to 3 and 4	Timer output master enable register	TOER	R/W	H'C0	H'FFFE220A	8
	Timer output control register 1	TOCR1	R/W	H'00	H'FFFE220E	8
	Timer output control register 2	TOCR2	R/W	H'00	H'FFFE220F	8
	Timer gate control register	TGCR	R/W	H80	H'FFFE220D	8
	Timer cycle control register	TCDR	R/W	H'FFFF	H'FFFE2214	16
	Timer dead time data register	TDDR	R/W	H'FFFF	H'FFFE2216	16
	Timer subcounter	TCNTS	R	H'0000	H'FFFE2220	16
	Timer cycle buffer register	TCBR	R/W	H'FFFF	H'FFFE2222	16
	Timer interrupt skipping set register	TITCR	R/W	H'00	H'FFFE2230	8
	Timer interrupt skipping counter	TITCNT	R	H'00	H'FFFE2231	8
	Timer buffer transfer set register	TBTER	R/W	H'00	H'FFFE2232	8
	Timer dead time enable register	TDER	R/W	H'01	H'FFFE2234	8
	Timer waveform control register	TWCR	R/W	H'00	H'FFFE2260	8
	Timer output level buffer register	TOLBR	R/W	H'00	H'FFFE2236	8

12.3.1 Timer Control Register (TCR)

The TCR registers are 8-bit readable/writable registers that control the TCNT operation for each channel. The MTU2 has a total of five TCR registers, one each for channels 0 to 4. TCR register settings should be conducted only when TCNT operation is stopped.



Bit	Bit Name	Initial Value	R/W	Description
7 to 5	CCLR[2:0]	000	R/W	Counter Clear 0 to 2 These bits select the TCNT counter clearing source. See tables 12.4 and 12.5 for details.
4, 3	CKEG[1:0]	00	R/W	Clock Edge 0 and 1 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 = Pϕ/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 Pϕ/4 or slower. When Pϕ/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
2 to 0	TPSC[2:0]	000	R/W	Time Prescaler 0 to 2 These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 12.6 to 12.9 for details.

[Legend]
x: Don't care

Table 12.4 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 12.5 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. It is always read as 0 and cannot be modified.

Table 12.6 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 12.7 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 12.8 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 12.9 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

12.3.2 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 MTU2 has five TMDR registers, one each for channels 0 to 4. TMDR register settings should be changed only when TCNT operation is stopped.

Bit:	7	6	5	4	3	2	1	0
	-	BFE	BFB	BFA	MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	BFE	0	R/W	Buffer Operation E Specifies whether TGRE_0 and TGRF_0 are to operate in the normal way or to be used together for buffer operation. TGRF compare match is generated when TGRF is used as the buffer register. In channels 1 to 4, this bit is reserved. It is always read as 0 and the write value should always be 0. 0: TGRE_0 and TGRF_0 operate normally 1: TGRE_0 and TGRF_0 used together for buffer operation

Bit	Bit Name	Initial Value	R/W	Description
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 a mode other than complementary PWM. In channels 1 and 2, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified. 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 a mode other than complementary PWM. TGRC compare match is generated when in complementary PWM mode. When compare match for channel 4 occurs during the Tb period in complementary PWM mode, TGFC is set. Therefore, set the TGIEC bit in the timer interrupt enable register 4 (TIER_4) to 0. In channels 1 and 2, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified. 0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation
3 to 0	MD[3:0]	0000	R/W	Modes 0 to 3 These bits are used to set the timer operating mode. See table 12.10 for details.

Table 12.10 Setting of Operation Mode by Bits MD0 to MD3

Bit 3 MD3	Bit 2 MD2	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
			1	Setting prohibited
		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	Setting prohibited
		1	X	Setting prohibited
	1	0	0	Setting prohibited
			1	Complementary PWM mode 1 (transmit at crest)*3
		1	0	Complementary PWM mode 2 (transmit at trough)*3
			1	Complementary PWM mode 2 (transmit at crest and trough)*3

[Legend]

X: Don't care

Notes: 1. PWM mode 2 cannot be set for channels 3 and 4.

2. Phase counting mode cannot be set for channels 0, 3, and 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, and 2.

12.3.3 Timer I/O Control Register (TIOR)

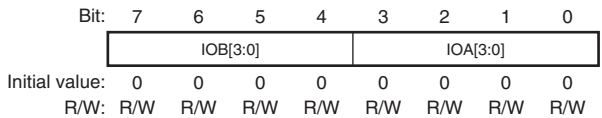
The TIOR registers are 8-bit readable/writable registers that control the TGR registers. The MTU2 has a total of eight TIOR registers, two each for channels 0, 3, and 4, one each for channels 1 and 2.

TIOR should be set while TMDR is set in normal operation, PWM mode, or phase counting mode.

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.

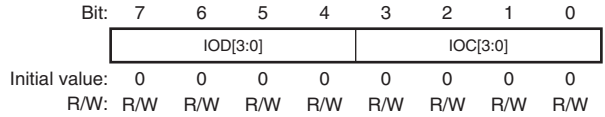
When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIORH_4



Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IOB[3:0]	0000	R/W	I/O Control B0 to B3 Specify the function of TGRB. See the following tables. TIORH_0: Table 12.11 TIOR_1: Table 12.13 TIOR_2: Table 12.14 TIORH_3: Table 12.15 TIORH_4: Table 12.17
3 to 0	IOA[3:0]	0000	R/W	I/O Control A0 to A3 Specify the function of TGRA. See the following tables. TIORH_0: Table 12.19 TIOR_1: Table 12.21 TIOR_2: Table 12.22 TIORH_3: Table 12.23 TIORH_4: Table 12.25

- TIORL_0, TIORL_3, TIORL_4



Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IOD[3:0]	0000	R/W	I/O Control D0 to D3 Specify the function of TGRD. See the following tables. TIORL_0: Table 12.12 TIORL_3: Table 12.16 TIORL_4: Table 12.18
3 to 0	IOC[3:0]	0000	R/W	I/O Control C0 to C3 Specify the function of TGRC. See the following tables. TIORL_0: Table 12.20 TIORL_3: Table 12.24 TIORL_4: Table 12.26

Table 12.11 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 retained*
			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 retained
			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: * After power-on reset, 0 is output until TIOR is set.

Table 12.12 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 retained*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 retained
			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. After power-on reset, 0 is output until TIOR is set.

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 12.13 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 retained*
			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 retained
			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: * After power-on reset, 0 is output until TIOR is set.

Table 12.14 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 retained*
			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 retained
			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: * After power-on reset, 0 is output until TIOR is set.

Table 12.15 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 retained*
			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 retained
			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: * After power-on reset, 0 is output until TIOR is set.

Table 12.16 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 retained*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 retained
			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. After power-on reset, 0 is output until TIOR is set.

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 12.17 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 retained*
			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 retained
			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: * After power-on reset, 0 is output until TIOR is set.

Table 12.18 TIORL_4 (Channel 4)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_4 Function**	**TIOC4D Pin Function**
0	0	0	0	Output compare register*2	Output retained*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 retained
			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. After power-on reset, 0 is output until TIOR is set.

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 12.19 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 retained*
			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 retained
			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: * After power-on reset, 0 is output until TIOR is set.

Table 12.20 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 retained*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 retained
			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. After power-on reset, 0 is output until TIOR is set.

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 12.21 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 retained*
			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 retained
			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: * After power-on reset, 0 is output until TIOR is set.

Table 12.22 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 retained*
			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 retained
			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: * After power-on reset, 0 is output until TIOR is set.

Table 12.23 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 retained*
			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 retained
			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: * After power-on reset, 0 is output until TIOR is set.

Table 12.24 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 retained*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 retained
			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. After power-on reset, 0 is output until TIOR is set.

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 12.25 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 retained*
			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 retained
			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: * After power-on reset, 0 is output until TIOR is set.

Table 12.26 TIORL_4 (Channel 4)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_4 Function**	**TIOC4C Pin Function**
0	0	0	0	Output compare register*2	Output retained*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 retained
			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. After power-on reset, 0 is output until TIOR is set.

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.

12.3.4 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 MTU2 has six TIER registers, two for channel 0 and one each for channels 1 to 4.

- TIER_0, TIER_1, TIER_2, TIER_3, TIER_4

Bit:	7	6	5	4	3	2	1	0
	TTGE	TTGE2	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TTGE	0	R/W	A/D Converter Start Request Enable Enables or disables generation of A/D converter start requests by TGRA input capture/compare match. 0: A/D converter start request generation disabled 1: A/D converter start request generation enabled
6	TTGE2	0	R/W	A/D Converter Start Request Enable 2 Enables or disables generation of A/D converter start requests by TCNT_4 underflow (trough) in complementary PWM mode. In channels 0 to 3, bit 6 is reserved. It is always read as 0 and the write value should always be 0. 0: A/D converter start request generation by TCNT_4 underflow (trough) disabled 1: A/D converter start request generation by TCNT_4 underflow (trough) enabled
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 the write value should always be 0. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled

Bit	Bit Name	Initial Value	R/W	Description
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0, 3, and 4. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and the write value should always be 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 the write value should always be 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 when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

- TIER2_0

Bit:	7	6	5	4	3	2	1	0
	TTGE2	-	-	-	-	-	TGIEF	TGIEE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TTGE2	0	R/W	A/D Converter Start Request Enable 2 Enables or disables generation of A/D converter start requests by compare match between TCNT_0 and TGRE_0. 0: A/D converter start request generation by compare match between TCNT_0 and TGRE_0 disabled 1: A/D converter start request generation by compare match between TCNT_0 and TGRE_0 enabled
6 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TGIEF	0	R/W	TGR Interrupt Enable F Enables or disables interrupt requests by compare match between TCNT_0 and TGRF_0. 0: Interrupt requests (TGIF) by TGFE bit disabled 1: Interrupt requests (TGIF) by TGFE bit enabled
0	TGIEE	0	R/W	TGR Interrupt Enable E Enables or disables interrupt requests by compare match between TCNT_0 and TGRE_0. 0: Interrupt requests (TGIE) by TGEE bit disabled 1: Interrupt requests (TGIE) by TGEE bit enabled

12.3.5 Timer Status Register (TSR)

The TSR registers are 8-bit readable/writable registers that indicate the status of each channel. The MTU2 has six TSR registers, two for channel 0 and one each for channels 1 to 4.

- TSR_0, TSR_1, TSR_2, TSR_3, TSR_4

Bit:	7	6	5	4	3	2	1	0
	TCFD	-	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1

Note: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

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 to 4. In channel 0, bit 7 is reserved. It is always read as 1 and the write value should always be 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)*1	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 the write value should always be 0. [Clearing condition] • When 0 is written to TCFU after reading TCFU = 1*2 [Setting condition] • When the TCNT value underflows (changes from H'0000 to H'FFFF)

Bit	Bit Name	Initial Value	R/W	Description
4	TCFV	0	R/(W)*1	Overflow Flag Status flag that indicates that TCNT overflow has occurred. Only 0 can be written, for flag clearing. [Clearing condition] • When 0 is written to TCFV after reading TCFV = 1*2 [Setting condition] • When the TCNT value overflows (changes from H'FFFF to H'0000) In channel 4, when the TCNT_4 value underflows (changes from H'0001 to H'0000) in complementary PWM mode, this flag is also set.
3	TGFD	0	R/(W)*1	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 the write value should always be 0. [Clearing condition] • When 0 is written to TGFD after reading TGFD = 1*2 [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

Bit	Bit Name	Initial Value	R/W	Description
2	TGFC	0	R/(W)*1	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 the write value should always be 0. [Clearing condition] • When 0 is written to TGFC after reading TGFC = 1*2 [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
1	TGFB	0	R/(W)*1	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. [Clearing condition] • When 0 is written to TGFB after reading TGFB = 1*2 [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

Bit	Bit Name	Initial Value	R/W	Description
0	TGFA	0	R/(W)*1	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. [Clearing conditions] • When DMAC is activated by TGIA interrupt • When 0 is written to TGFA after reading TGFA = 1*2 [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

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. If the next flag is set before cleared to 0 after reading 1, this bit remains 1 even when 0 is written to. In this case, read 1 again to clear to 0.

- TSR2_0

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TGFF	TGFE
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*1	R/(W)*1

Note: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

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 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TGFF	0	R/(W)*1	Compare Match Flag F Status flag that indicates the occurrence of compare match between TCNT_0 and TGRF_0. [Clearing condition] • When 0 is written to TGFF after reading TGFF = 1*2 [Setting condition] • When TCNT_0 = TGRF_0 and TGRF_0 is functioning as compare register
0	TGFE	0	R/(W)*1	Compare Match Flag E Status flag that indicates the occurrence of compare match between TCNT_0 and TGRE_0. [Clearing condition] • When 0 is written to TGFE after reading TGFE = 1*2 [Setting condition] • When TCNT_0 = TGRE_0 and TGRE_0 is functioning as compare register

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. If the next flag is set before cleared to 0 after reading 1, this bit remains 1 even when 0 is written to. In this case, read 1 again to clear to 0.

12.3.6 Timer Buffer Operation Transfer Mode Register (TBTM)

The TBTM registers are 8-bit readable/writable registers that specify the timing for transferring data from the buffer register to the timer general register in PWM mode. The MTU2 has three TBTM registers, one each for channels 0, 3, and 4.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	TTSE	TTSB	TTSA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TTSE	0	R/W	Timing Select E Specifies the timing for transferring data from TGRF_0 to TGRE_0 when they are used together for buffer operation. In channels 3 and 4, bit 2 is reserved. It is always read as 0 and the write value should always be 0. 0: When compare match E occurs in channel 0 1: When TCNT_0 is cleared
1	TTSB	0	R/W	Timing Select B Specifies the timing for transferring data from TGRD to TGRB in each channel when they are used together for buffer operation. 0: When compare match B occurs in each channel 1: When TCNT is cleared in each channel
0	TTSA	0	R/W	Timing Select A Specifies the timing for transferring data from TGRC to TGRA in each channel when they are used together for buffer operation. 0: When compare match A occurs in each channel 1: When TCNT is cleared in each channel

12.3.7 Timer Input Capture Control Register (TICCR)

TICCR is an 8-bit readable/writable register that specifies input capture conditions when TCNT_1 and TCNT_2 are cascaded. The MTU2 has one TICCR in channel 1.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	I2BE	I2AE	I1BE	I1AE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	I2BE	0	R/W	Input Capture Enable Specifies whether to include the TIOC2B pin in the TGRB_1 input capture conditions. 0: Does not include the TIOC2B pin in the TGRB_1 input capture conditions 1: Includes the TIOC2B pin in the TGRB_1 input capture conditions
2	I2AE	0	R/W	Input Capture Enable Specifies whether to include the TIOC2A pin in the TGRA_1 input capture conditions. 0: Does not include the TIOC2A pin in the TGRA_1 input capture conditions 1: Includes the TIOC2A pin in the TGRA_1 input capture conditions
1	I1BE	0	R/W	Input Capture Enable Specifies whether to include the TIOC1B pin in the TGRB_2 input capture conditions. 0: Does not include the TIOC1B pin in the TGRB_2 input capture conditions 1: Includes the TIOC1B pin in the TGRB_2 input capture conditions

Bit	Bit Name	Initial Value	R/W	Description
0	I1AE	0	R/W	Input Capture Enable Specifies whether to include the TIOC1A pin in the TGRA_2 input capture conditions. 0: Does not include the TIOC1A pin in the TGRA_2 input capture conditions 1: Includes the TIOC1A pin in the TGRA_2 input capture conditions

12.3.8 Timer A/D Converter Start Request Control Register (TADCR)

TADCR is a 16-bit readable/writable register that enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping operation. The MTU2 has one TADCR in channel 4.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BF[1:0]		-	-	-	-	-	-	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
Initial value:	0	0	0	0	0	0	0	0	0	0*	0	0*	0*	0*	0*	0*
R/W:	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Do not set to 1 when complementary PWM mode is not selected.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	BF[1:0]	00	R/W	TADCOBRA_4/TADCOBRB_4 Transfer Timing Select Select the timing for transferring data from TADCOBRA_4 and TADCOBRB_4 to TADCORA_4 and TADCORB_4. For details, see table 12.27.
13 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	UT4AE	0	R/W	Up-Count TRG4AN Enable Enables or disables A/D converter start requests (TRG4AN) during TCNT_4 up-count operation. 0: A/D converter start requests (TRG4AN) disabled during TCNT_4 up-count operation 1: A/D converter start requests (TRG4AN) enabled during TCNT_4 up-count operation

Bit	Bit Name	Initial Value	R/W	Description
6	DT4AE	0*	R/W	Down-Count TRG4AN Enable Enables or disables A/D converter start requests (TRG4AN) during TCNT_4 down-count operation. 0: A/D converter start requests (TRG4AN) disabled during TCNT_4 down-count operation 1: A/D converter start requests (TRG4AN) enabled during TCNT_4 down-count operation
5	UT4BE	0	R/W	Up-Count TRG4BN Enable Enables or disables A/D converter start requests (TRG4BN) during TCNT_4 up-count operation. 0: A/D converter start requests (TRG4BN) disabled during TCNT_4 up-count operation 1: A/D converter start requests (TRG4BN) enabled during TCNT_4 up-count operation
4	DT4BE	0*	R/W	Down-Count TRG4BN Enable Enables or disables A/D converter start requests (TRG4BN) during TCNT_4 down-count operation. 0: A/D converter start requests (TRG4BN) disabled during TCNT_4 down-count operation 1: A/D converter start requests (TRG4BN) enabled during TCNT_4 down-count operation
3	ITA3AE	0*	R/W	TGIA_3 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4AN) with TGIA_3 interrupt skipping operation. 0: Does not link with TGIA_3 interrupt skipping 1: Links with TGIA_3 interrupt skipping
2	ITA4VE	0*	R/W	TCIV_4 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4AN) with TCIV_4 interrupt skipping operation. 0: Does not link with TCIV_4 interrupt skipping 1: Links with TCIV_4 interrupt skipping
1	ITB3AE	0*	R/W	TGIA_3 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4BN) with TGIA_3 interrupt skipping operation. 0: Does not link with TGIA_3 interrupt skipping 1: Links with TGIA_3 interrupt skipping

Bit	Bit Name	Initial Value	R/W	Description
0	ITB4VE	0*	R/W	TCIV_4 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4BN) with TCIV_4 interrupt skipping operation. 0: Does not link with TCIV_4 interrupt skipping 1: Links with TCIV_4 interrupt skipping

Notes: 1. TADCR must not be accessed in eight bits; it should always be accessed in 16 bits.
2. When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), do not link A/D converter start requests with interrupt skipping operation (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).
3. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
* Do not set to 1 when complementary PWM mode is not selected.

Table 12.27 Setting of Transfer Timing by Bits BF1 and BF0

Bit 7	Bit 6	
BF1	**BF0**	**Description**
0	0	Does not transfer data from the cycle set buffer register to the cycle set register.
0	1	Transfers data from the cycle set buffer register to the cycle set register at the crest of the TCNT_4 count.*[1]
1	0	Transfers data from the cycle set buffer register to the cycle set register at the trough of the TCNT_4 count.*[2]
1	1	Transfers data from the cycle set buffer register to the cycle set register at the crest and trough of the TCNT_4 count.*[2]

Notes: 1. Data is transferred from the cycle set buffer register to the cycle set register when the crest of the TCNT_4 count is reached in complementary PWM mode, when compare match occurs between TCNT_3 and TGRA_3 in reset-synchronized PWM mode, or when compare match occurs between TCNT_4 and TGRA_4 in PWM mode 1 or normal operation mode.
2. These settings are prohibited when complementary PWM mode is not selected.

12.3.9 Timer A/D Converter Start Request Cycle Set Registers (TADCORA_4 and TADCORB_4)

TADCORA_4 and TADCORB_4 are 16-bit readable/writable registers. When the TCNT_4 count reaches the value in TADCORA_4 or TADCORB_4, a corresponding A/D converter start request will be issued.

TADCORA_4 and TADCORB_4 are initialized to H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TADCORA_4 and TADCORB_4 must not be accessed in eight bits; they should always be accessed in 16 bits.

12.3.10 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4)

TADCOBRA_4 and TADCOBRB_4 are 16-bit readable/writable registers. When the crest or trough of the TCNT_4 count is reached, these register values are transferred to TADCORA_4 and TADCORB_4, respectively.

TADCOBRA_4 and TADCOBRB_4 are initialized to H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TADCOBRA_4 and TADCOBRB_4 must not be accessed in eight bits; they should always be accessed in 16 bits.

12.3.11 Timer Counter (TCNT)

The TCNT counters are 16-bit readable/writable counters. The MTU2 has five TCNT counters, one each for channels 0 to 4.

The TCNT counters must not be accessed in eight bits; they should always be accessed in 16 bits.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: The TCNT counters must not be accessed in eight bits; they should always be accessed in 16 bits.

12.3.12 Timer General Register (TGR)

The TGR registers are 16-bit readable/writable registers. The MTU2 has eighteen TGR registers, six for channel 0, two each for channels 1 and 2, four each for channels 3 and 4.

TGRA, TGRB, TGRC, and TGRD function as either output compare or input capture registers. TGRC and TGRD for channels 0, 3, and 4 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.

TGRE_0 and TGRF_0 function as compare registers. When the TCNT_0 count matches the TGRE_0 value, an A/D converter start request can be issued. TGRF can also be designated for operation as a buffer register. TGR buffer register combination is TGRE and TGRF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: The TGR registers must not be accessed in eight bits; they should always be accessed in 16 bits.
TGR registers are initialized to H'FFFF.

12.3.13 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage of TCNT 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:	7	6	5	4	3	2	1	0
	CST4	CST3	-	-	-	CST2	CST1	CST0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CST4	0	R/W	Counter Start 4 and 3 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
6	CST3	0	R/W	
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 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_2 to TCNT_0 count operation is stopped 1: TCNT_2 to TCNT_0 performs count operation
1	CST1	0	R/W	
0	CST0	0	R/W	

12.3.14 Timer Synchronous 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:	7	6	5	4	3	2	1	0
	SYNC4	SYNC3	-	-	-	SYNC2	SYNC1	SYNC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	SYNC4	0	R/W	Timer Synchronous operation 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.

Bit	Bit Name	Initial Value	R/W	Description
2	SYNC2	0	R/W	Timer Synchronous operation 2 to 0 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
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

12.3.15 Timer Read/Write Enable Register (TRWER)

TRWER is an 8-bit readable/writable register that enables or disables access to the registers and counters which have write-protection capability against accidental modification in channels 3 and 4.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	RWE
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RWE	1	R/W	Read/Write Enable Enables or disables access to the registers which have write-protection capability against accidental modification. 0: Disables read/write access to the registers 1: Enables read/write access to the registers [Clearing condition] • When 0 is written to the RWE bit after reading RWE = 1

- Registers and counters having write-protection capability against accidental modification
  22 registers: TCR_3, TCR_4, TMDR_3, TMDR_4, TIORH_3, TIORH_4, TIORL_3, TIORL_4, TIER_3, TIER_4, TGRA_3, TGRA_4, TGRB_3, TGRB_4, TOER, TOCR1, TOCR2, TGCR, TCDR, TDDR, TCNT_3, and TCNT4.

12.3.16 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. Set TOER when count operation of TCNT channels 3 and 4 is halted.

Bit:	7	6	5	4	3	2	1	0
	-	-	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

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 MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
4	OE4C	0	R/W	Master Enable TIOC4C This bit enables/disables the TIOC4C pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
3	OE3D	0	R/W	Master Enable TIOC3D This bit enables/disables the TIOC3D pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
2	OE4B	0	R/W	Master Enable TIOC4B This bit enables/disables the TIOC4B pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
1	OE4A	0	R/W	Master Enable TIOC4A This bit enables/disables the TIOC4A pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled

Bit	Bit Name	Initial Value	R/W	Description
0	OE3B	0	R/W	Master Enable TIOC3B This bit enables/disables the TIOC3B pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled

Note: * The inactive level is determined by the settings in timer output control registers 1 and 2 (TOCR1 and TOCR2). For details, refer to section 12.3.17, Timer Output Control Register 1 (TOCR1), and section 12.3.18, Timer Output Control Register 2 (TOCR2). Set these bits to 1 to enable MTU2 output in other than complementary PWM or reset-synchronized PWM mode. When these bits are set to 0, low level is output.

12.3.17 Timer Output Control Register 1 (TOCR1)

TOCR1 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:	7	6	5	4	3	2	1	0
	-	PSYE	-	-	TOCL	TOCS	OLSN	OLSP
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R/(W)*	R/W	R/W	R/W

Note: * This bit can be set to 1 only once after a power-on reset. After 1 is written, 0 cannot be written to the bit.

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, 4	—	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
3	TOCL	0	R/(W)*3	TOC Register Write Protection*2 This bit selects the enable/disable of write access to the TOCS, OLSN, and OLSP bits in TOCR1. 0: Write access to the TOCS, OLSN, and OLSP bits is enabled 1: Write access to the TOCS, OLSN, and OLSP bits is disabled
2	TOCS	0	R/W	TOC Select This bit selects either the TOCR1 or TOCR2 setting to be used for the output level in complementary PWM mode and reset-synchronized PWM mode. 0: TOCR1 setting is selected 1: TOCR2 setting is selected
1	OLSN	0	R/W	Output Level Select N*2*4 This bit selects the reverse phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 12.28.
0	OLSP	0	R/W	Output Level Select P*2 This bit selects the positive phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 12.29.

Notes: 1. Setting the TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
2. Clearing the TOCS0 bit to 0 makes this bit setting valid.
3. After power-on reset, 1 can be written only once. After 1 has been written, 0 cannot be written.
4. If there is no dead time, the reverse phase output is the inversion of the forward phase. Set OLSP and OLSN to the same value.

Table 12.28 Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLSN	Initial Output	Active Level	Up Count	Down 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 12.29 Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLSP	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Figure 12.2 shows an example of complementary PWM mode output (1 phase) when OLSN = 1, OLSP = 1.

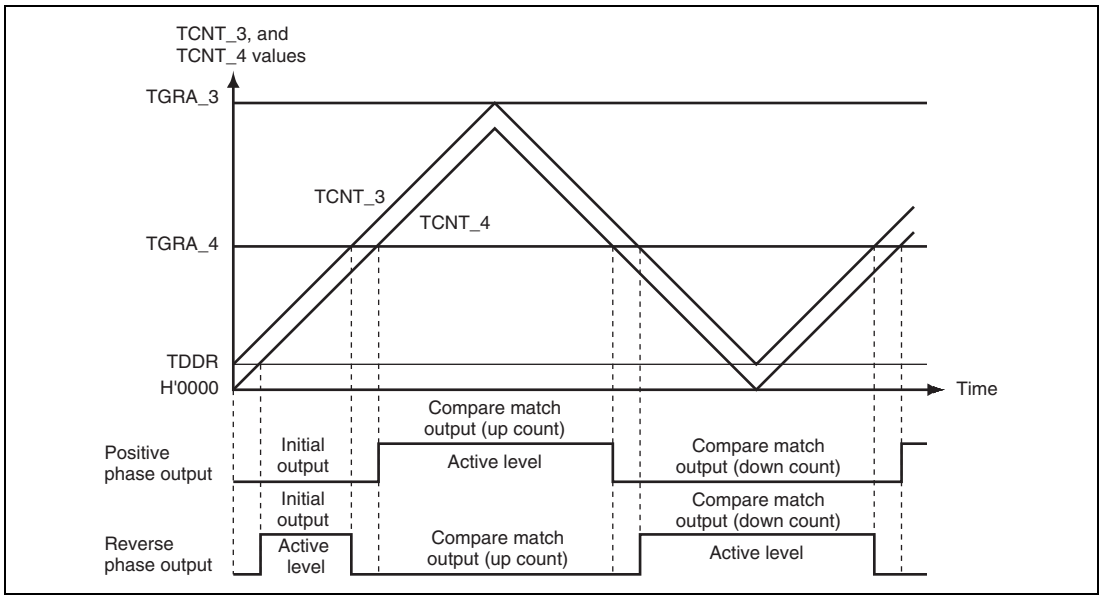


Figure 12.2 Complementary PWM Mode Output Level Example

12.3.18 Timer Output Control Register 2 (TOCR2)

TOCR2 is an 8-bit readable/writable register that controls output level inversion of PWM output in complementary PWM mode and reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7, 6	BF[1:0]	00	R/W	TOLBR Buffer Transfer Timing Select These bits select the timing for transferring data from TOLBR to TOCR2. For details, see table 12.30.
5	OLS3N	0	R/W	Output Level Select 3N* This bit selects the output level on TIOC4D in reset-synchronized PWM mode/complementary PWM mode. See table 12.31.
4	OLS3P	0	R/W	Output Level Select 3P* This bit selects the output level on TIOC4B in reset-synchronized PWM mode/complementary PWM mode. See table 12.32.
3	OLS2N	0	R/W	Output Level Select 2N* This bit selects the output level on TIOC4C in reset-synchronized PWM mode/complementary PWM mode. See table 12.33.
2	OLS2P	0	R/W	Output Level Select 2P* This bit selects the output level on TIOC4A in reset-synchronized PWM mode/complementary PWM mode. See table 12.34.
1	OLS1N	0	R/W	Output Level Select 1N* This bit selects the output level on TIOC3D in reset-synchronized PWM mode/complementary PWM mode. See table 12.35.

Bit	Bit Name	Initial value	R/W	Description
0	OLS1P	0	R/W	Output Level Select 1P* This bit selects the output level on TIOC3B in reset-synchronized PWM mode/complementary PWM mode. See table 12.36.

Note: * Setting the TOCS bit in TOCR1 to 1 makes this bit setting valid. If there is no dead time, the reverse phase output is the inversion of the forward phase. Set OLSiP and OLSiN to the same value (i = 1, 2, or 3).

Table 12.30 Setting of Bits BF1 and BF0

Bit 7	Bit 6	Description	
BF1	**BF0**	**Complementary PWM Mode**	**Reset-Synchronized PWM Mode**
0	0	Does not transfer data from the buffer register (TOLBR) to TOCR2.	Does not transfer data from the buffer register (TOLBR) to TOCR2.
0	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest of the TCNT_4 count.	Transfers data from the buffer register (TOLBR) to TOCR2 when TCNT_3/TCNT_4 is cleared
1	0	Transfers data from the buffer register (TOLBR) to TOCR2 at the trough of the TCNT_4 count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest and trough of the TCNT_4 count.	Setting prohibited

Table 12.31 TIOC4D Output Level Select Function

Bit 5	Function			
			Compare Match Output	
OLS3N	**Initial Output**	**Active Level**	**Up Count**	**Down 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 the active level after elapse of the dead time after count start.

Table 12.32 TIOC4B Output Level Select Function

Bit 4	Function			
			Compare Match Output	
OLS3P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 12.33 TIOC4C Output Level Select Function

Bit 3	Function			
			Compare Match Output	
OLS2N	Initial Output	Active Level	Up Count	Down 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 the active level after elapse of the dead time after count start.

Table 12.34 TIOC4A Output Level Select Function

Bit 2	Function			
			Compare Match Output	
OLS2P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 12.35 TIOC3D Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLS1N	Initial Output	Active Level	Up Count	Down 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 the active level after elapse of the dead time after count start.

Table 12.36 TIOC4B Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLS1P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

12.3.19 Timer Output Level Buffer Register (TOLBR)

TOLBR is an 8-bit readable/writable register that functions as a buffer for TOCR2 and specifies the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	OLS3N	0	R/W	Specifies the buffer value to be transferred to the OLS3N bit in TOCR2.
4	OLS3P	0	R/W	Specifies the buffer value to be transferred to the OLS3P bit in TOCR2.
3	OLS2N	0	R/W	Specifies the buffer value to be transferred to the OLS2N bit in TOCR2.
2	OLS2P	0	R/W	Specifies the buffer value to be transferred to the OLS2P bit in TOCR2.
1	OLS1N	0	R/W	Specifies the buffer value to be transferred to the OLS1N bit in TOCR2.
0	OLS1P	0	R/W	Specifies the buffer value to be transferred to the OLS1P bit in TOCR2.

Figure 12.3 shows an example of the PWM output level setting procedure in buffer operation.

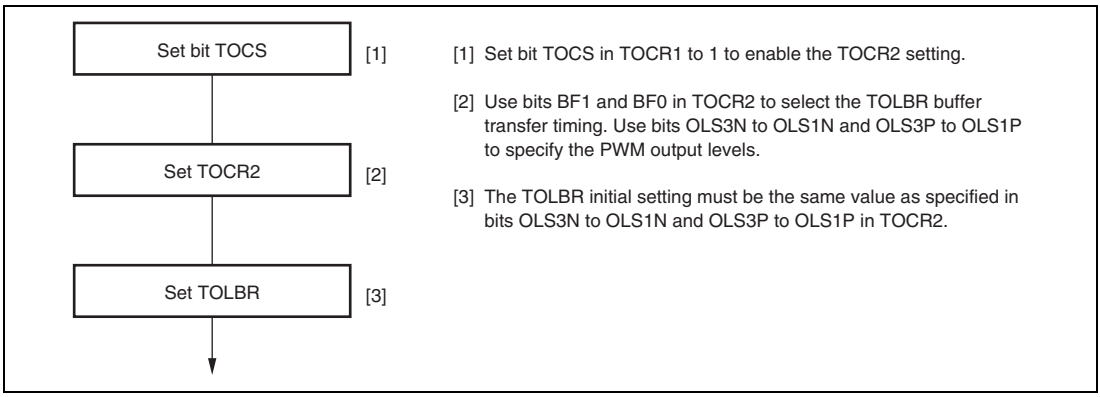


Figure 12.3 PWM Output Level Setting Procedure in Buffer Operation

12.3.20 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:	7	6	5	4	3	2	1	0
	-	BDC	N	P	FB	WF	VF	UF
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial value	R/W	Description
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 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 output. 0: Level output 1: Reset synchronized PWM/complementary PWM output
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 MTU2/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 external input (Input sources are channel 0 TGRA, TGRB, TGRC input capture signal) 1: Output switching is carried out by software (setting values of UF, VF, and WF in TGCR).
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 12.37.
1	VF	0	R/W	
0	UF	0	R/W	

Table 12.37 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

12.3.21 Timer Subcounter (TCNTS)

TCNTS is a 16-bit read-only counter that is used only in complementary PWM mode.

The initial value of TCNTS is H'0000.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: Accessing the TCNTS in 8-bit units is prohibited. Always access in 16-bit units.

12.3.22 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 register value is loaded into the TCNT_3 counter and the count operation starts.

The initial value of TDDR is H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Accessing the TDDR in 8-bit units is prohibited. Always access in 16-bit units.

12.3.23 Timer Cycle Data Register (TCDR)

TCDR is a 16-bit register used only in complementary PWM mode. Set half the PWM carrier sync value (a value of two times TDDR + 3 or greater) as the TCDR register 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).

The initial value of TCDR is H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Accessing the TCDR in 8-bit units is prohibited. Always access in 16-bit units.

12.3.24 Timer Cycle Buffer Register (TCBR)

TCBR is a 16-bit register used only in complementary PWM mode. It functions as a buffer register for the TCDR register. The TCBR register values are transferred to the TCDR register with the transfer timing set in the TMDR register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Accessing the TCBR in 8-bit units is prohibited. Always access in 16-bit units.

12.3.25 Timer Interrupt Skipping Set Register (TITCR)

TITCR is an 8-bit readable/writable register that enables or disables interrupt skipping and specifies the interrupt skipping count. The MTU2 has one TITCR.

Bit:	7	6	5	4	3	2	1	0
	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	T3AEN	0	R/W	T3AEN Enables or disables TGIA_3 interrupt skipping. 0: TGIA_3 interrupt skipping disabled 1: TGIA_3 interrupt skipping enabled
6 to 4	3ACOR[2:0]	000	R/W	These bits specify the TGIA_3 interrupt skipping count within the range from 0 to 7.* For details, see table 12.38.
3	T4VEN	0	R/W	T4VEN Enables or disables TCIV_4 interrupt skipping. 0: TCIV_4 interrupt skipping disabled 1: TCIV_4 interrupt skipping enabled

Bit	Bit Name	Initial value	R/W	Description
2 to 0	4VCOR[2:0]	000	R/W	These bits specify the TCIV_4 interrupt skipping count within the range from 0 to 7.* For details, see table 12.39.

Note: * When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed. Before changing the interrupt skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter (TICNT).

Table 12.38 Setting of Interrupt Skipping Count by Bits 3ACOR2 to 3ACOR0

Bit 6	Bit 5	Bit 4	
3ACOR2	3ACOR1	3ACOR0	Description
0	0	0	Does not skip TGIA_3 interrupts.
0	0	1	Sets the TGIA_3 interrupt skipping count to 1.
0	1	0	Sets the TGIA_3 interrupt skipping count to 2.
0	1	1	Sets the TGIA_3 interrupt skipping count to 3.
1	0	0	Sets the TGIA_3 interrupt skipping count to 4.
1	0	1	Sets the TGIA_3 interrupt skipping count to 5.
1	1	0	Sets the TGIA_3 interrupt skipping count to 6.
1	1	1	Sets the TGIA_3 interrupt skipping count to 7.

Table 12.39 Setting of Interrupt Skipping Count by Bits 4VCOR2 to 4VCOR0

Bit 2	Bit 1	Bit 0	
4VCOR2	4VCOR1	4VCOR0	Description
0	0	0	Does not skip TCIV_4 interrupts.
0	0	1	Sets the TCIV_4 interrupt skipping count to 1.
0	1	0	Sets the TCIV_4 interrupt skipping count to 2.
0	1	1	Sets the TCIV_4 interrupt skipping count to 3.
1	0	0	Sets the TCIV_4 interrupt skipping count to 4.
1	0	1	Sets the TCIV_4 interrupt skipping count to 5.
1	1	0	Sets the TCIV_4 interrupt skipping count to 6.
1	1	1	Sets the TCIV_4 interrupt skipping count to 7.

12.3.26 Timer Interrupt Skipping Counter (TITCNT)

TITCNT is an 8-bit readable/writable counter. The MTU2 has one TITCNT. TITCNT retains its value even after stopping the count operation of TCNT_3 and TCNT_4.

Bit:	7	6	5	4	3	2	1	0
	-	3ACNT[2:0]			-	4VCNT[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0.
6 to 4	3ACNT[2:0]	000	R	TGIA_3 Interrupt Counter While the T3AEN bit in TITCR is set to 1, the count in these bits is incremented every time a TGIA_3 interrupt occurs. [Clearing conditions] • When the 3ACNT2 to 3ACNT0 value in TITCNT matches the 3ACOR2 to 3ACOR0 value in TITCR • When the T3AEN bit in TITCR is cleared to 0 • When the 3ACOR2 to 3ACOR0 bits in TITCR are cleared to 0
3	—	0	R	Reserved This bit is always read as 0.
2 to 0	4VCNT[2:0]	000	R	TCIV_4 Interrupt Counter While the T4VEN bit in TITCR is set to 1, the count in these bits is incremented every time a TCIV_4 interrupt occurs. [Clearing conditions] • When the 4VCNT2 to 4VCNT0 value in TITCNT matches the 4VCOR2 to 4VCOR2 value in TITCR • When the T4VEN bit in TITCR is cleared to 0 • When the 4VCOR2 to 4VCOR2 bits in TITCR are cleared to 0

Note: To clear the TITCNT, clear the bits T3AEN and T4VEN in TITCR to 0.

12.3.27 Timer Buffer Transfer Set Register (TBTER)

TBTER is an 8-bit readable/writable register that enables or disables transfer from the buffer registers* used in complementary PWM mode to the temporary registers and specifies whether to link the transfer with interrupt skipping operation. The MTU2 has one TBTER.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	BTE[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	BTE[1:0]	00	R/W	These bits enable or disable transfer from the buffer registers* used in complementary PWM mode to the temporary registers and specify whether to link the transfer with interrupt skipping operation. For details, see table 12.40.

Note: * Applicable buffer registers:
TGRC_3, TGRD_3, TGRC_4, TGRD_4, and TCBR

Table 12.40 Setting of Bits BTE1 and BTE0

Bit 1	Bit 0	
BTE1	**BTE0**	**Description**
0	0	Enables transfer from the buffer registers to the temporary registers*[1] and does not link the transfer with interrupt skipping operation.
0	1	Disables transfer from the buffer registers to the temporary registers.
1	0	Links transfer from the buffer registers to the temporary registers with interrupt skipping operation.*[2]
1	1	Setting prohibited

Notes: 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For details, refer to section 12.4.8, Complementary PWM Mode.

2. When interrupt skipping is disabled (the T3AEN and T4VEN bits are cleared to 0 in the timer interrupt skipping set register (TITCR) or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0)), be sure to disable link of buffer transfer with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

12.3.28 Timer Dead Time Enable Register (TDER)

TDER is an 8-bit readable/writable register that controls dead time generation in complementary PWM mode. The MTU2 has one TDER in channel 3. TDER must be modified only while TCNT stops.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	TDER
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/(W)

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TDER	1	R/(W)	Dead Time Enable Specifies whether to generate dead time. 0: Does not generate dead time 1: Generates dead time* [Clearing condition] • When 0 is written to TDER after reading TDER = 1

Note: * TDDR must be set to 1 or a larger value.

12.3.29 Timer Waveform Control Register (TWCR)

TWCR is an 8-bit readable/writable register that controls the waveform when synchronous counter clearing occurs in TCNT_3 and TCNT_4 in complementary PWM mode and specifies whether to clear the counters at TGRA_3 compare match. The CCE bit and WRE bit in TWCR must be modified only while TCNT stops.

Bit:	7	6	5	4	3	2	1	0
	CCE	-	-	-	-	-	-	WRE
Initial value:	0*	0	0	0	0	0	0	0
R/W:	R/(W)	R	R	R	R	R	R	R/(W)

Note: * Do not set to 1 when complementary PWM mode is not selected.

Bit	Bit Name	Initial Value	R/W	Description
7	CCE	0*	R/(W)	Compare Match Clear Enable Specifies whether to clear counters at TGRA_3 compare match in complementary PWM mode. 0: Does not clear counters at TGRA_3 compare match 1: Clears counters at TGRA_3 compare match [Setting condition] • When 1 is written to CCE after reading CCE = 0
6 to 1	—	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
0	WRE	0	R/(W)	Initial Output Suppression Enable Selects the waveform output when synchronous counter clearing occurs in complementary PWM mode. The initial output is suppressed only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial value specified in TOCR is output regardless of the WRE bit setting. The initial value is also output when synchronous clearing occurs in the Tb interval at the trough immediately after TCNT_3 and TCNT_4 start operation. For the Tb interval at the trough in complementary PWM mode, see figure 12.40. 0: Outputs the initial value specified in TOCR 1: Suppresses initial output [Setting condition] • When 1 is written to WRE after reading WRE = 0

Note: * Do not set to 1 when complementary PWM mode is not selected.

12.3.30 Bus Master Interface

The timer counters (TCNT), general registers (TGR), timer subcounter (TCNTS), timer cycle buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register (TCDR), timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCOR), and timer A/D converter start request cycle set buffer registers (TADCOBR) 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.

12.4 Operation

12.4.1 Basic Functions

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, cycle counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Always select MTU2 external pins set function using the pin function controller (PFC).

(1) Counter Operation

When one of bits CST0 to CST4 in TSTR is set to 1, the TCNT counter for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

(a) Example of Count Operation Setting Procedure

Figure 12.4 shows an example of the count operation setting procedure.

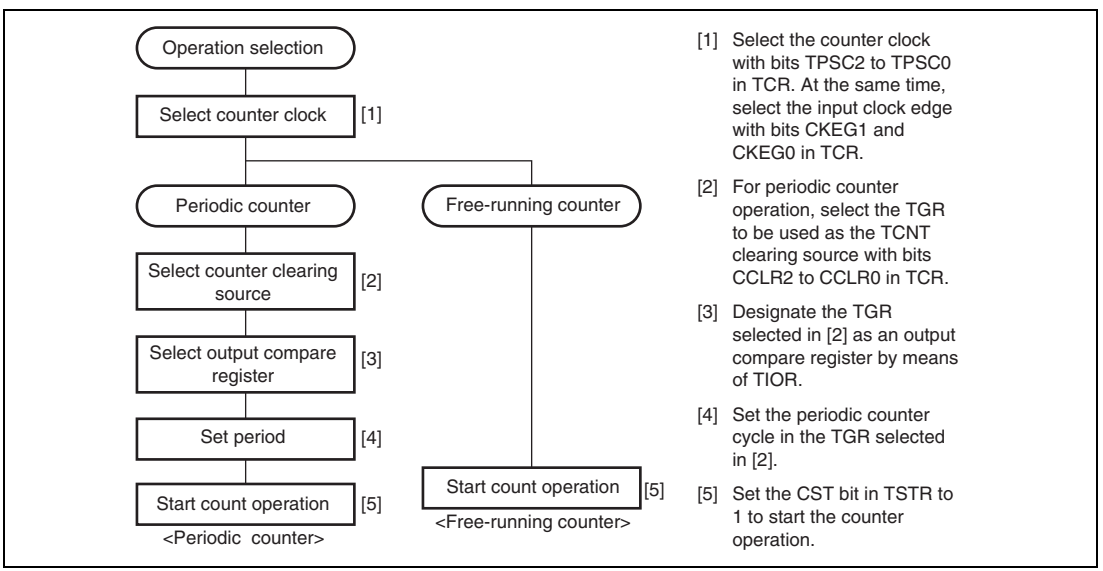


Figure 12.4 Example of Counter Operation Setting Procedure

(b) Free-Running Count Operation and Periodic Count Operation:

Immediately after a reset, the MTU2'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 MTU2 requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

Figure 12.5 illustrates free-running counter operation.

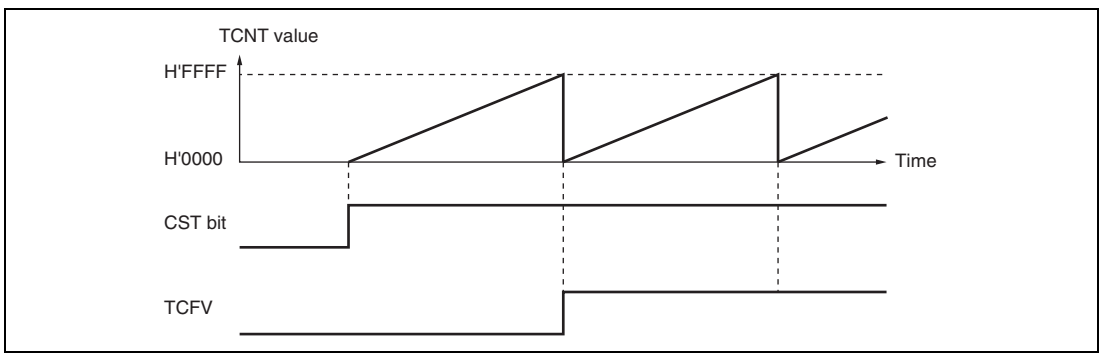


Figure 12.5 Free-Running Counter Operation

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 MTU2 requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 12.6 illustrates periodic counter operation.

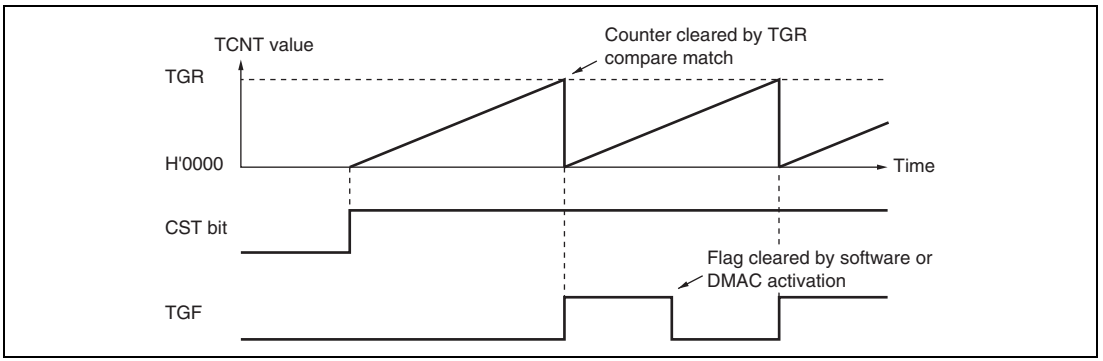


Figure 12.6 Periodic Counter Operation

(2) Waveform Output by Compare Match

The MTU2 can perform 0, 1, or toggle output from the corresponding output pin using compare match.

(a) Example of Setting Procedure for Waveform Output by Compare Match

Figure 12.7 shows an example of the setting procedure for waveform output by compare match

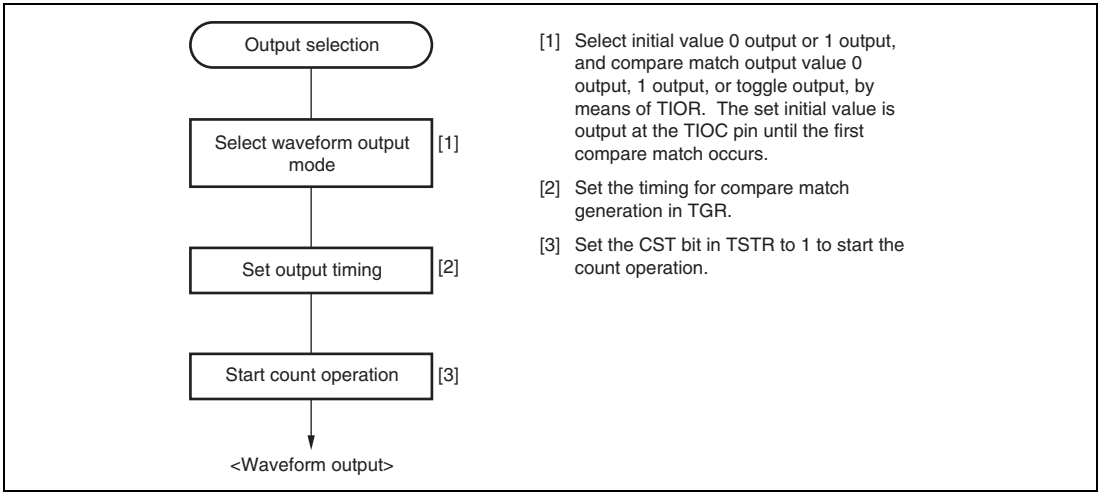


Figure 12.7 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of Waveform Output Operation:

Figure 12.8 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.

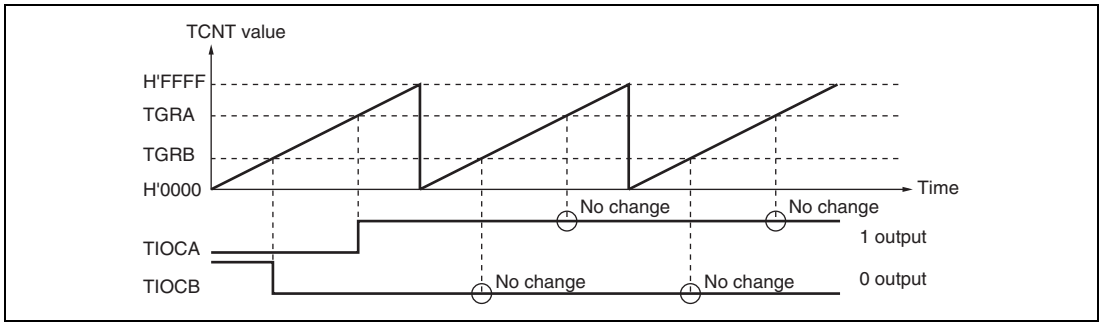


Figure 12.8 Example of 0 Output/1 Output Operation

Figure 12.9 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.

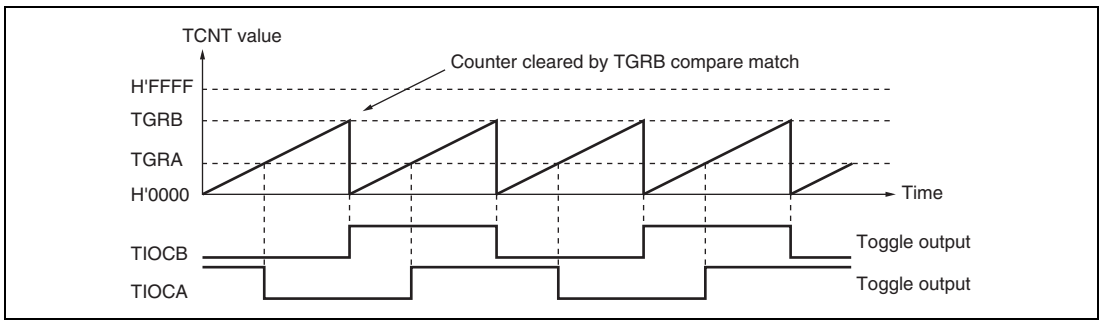


Figure 12.9 Example of Toggle Output Operation

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the 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, Pϕ/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if Pϕ/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 12.10 shows an example of the input capture operation setting procedure.

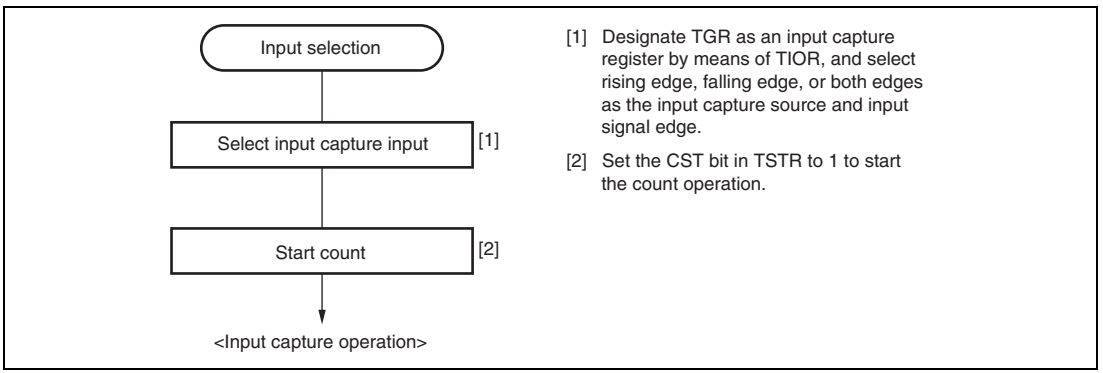


Figure 12.10 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 12.11 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.

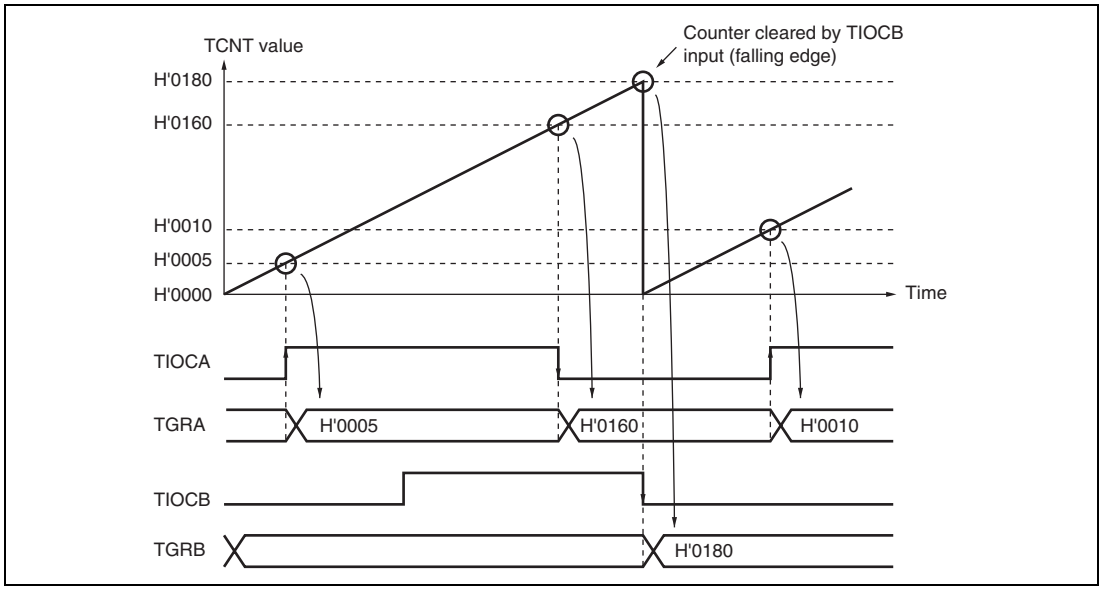


Figure 12.11 Example of Input Capture Operation

12.4.2 Synchronous Operation

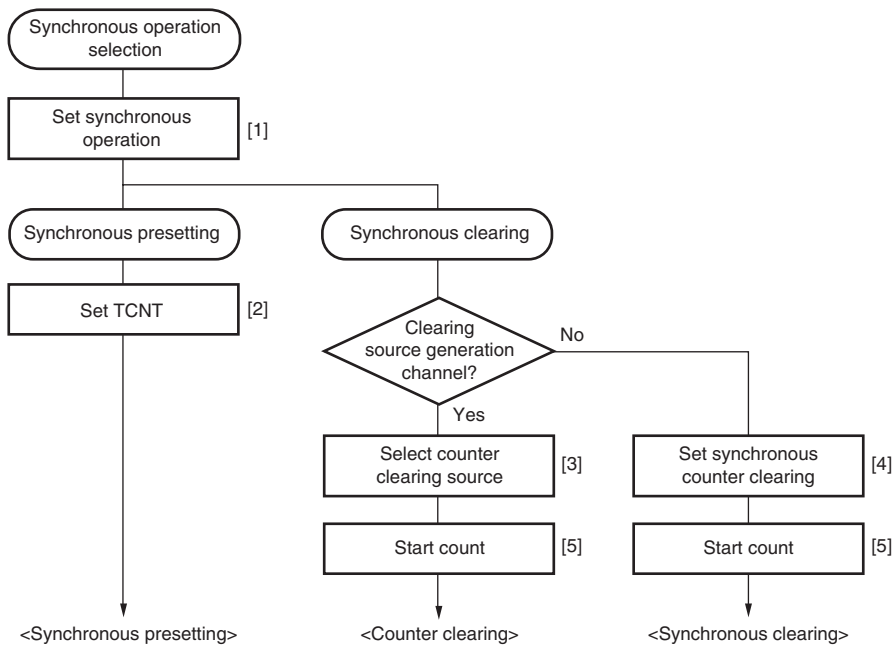
In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

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.

(1) Example of Synchronous Operation Setting Procedure

Figure 12.12 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 12.12 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 12.13 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 12.4.5, PWM Modes.

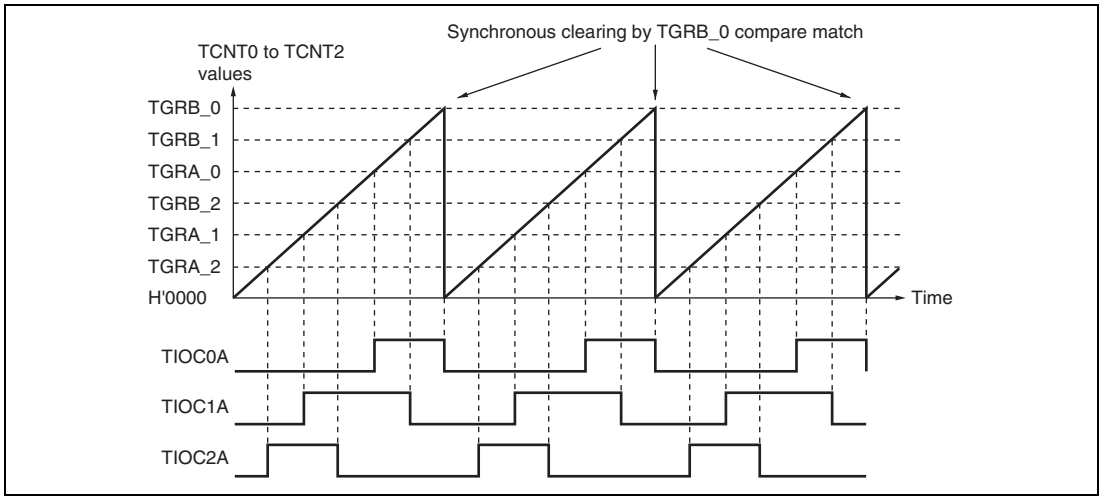


Figure 12.13 Example of Synchronous Operation

12.4.3 Buffer Operation

Buffer operation, provided for channels 0, 3, and 4, enables TGRC and TGRD to be used as buffer registers. In channel 0, TGRF can also be used as a buffer register.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Note: TGRE_0 cannot be designated as an input capture register and can only operate as a compare match register.

Table 12.41 shows the register combinations used in buffer operation.

Table 12.41 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
	TGRE_0	TGRF_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 12.14.

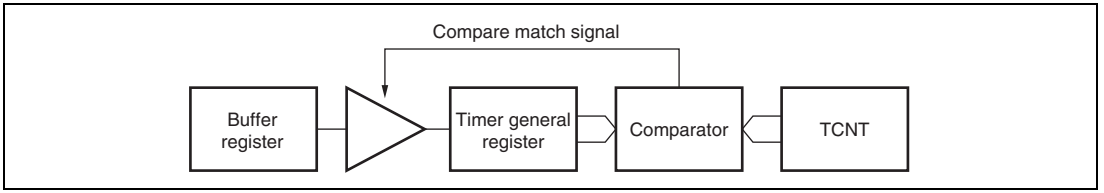


Figure 12.14 Compare Match Buffer Operation

- 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 12.15.

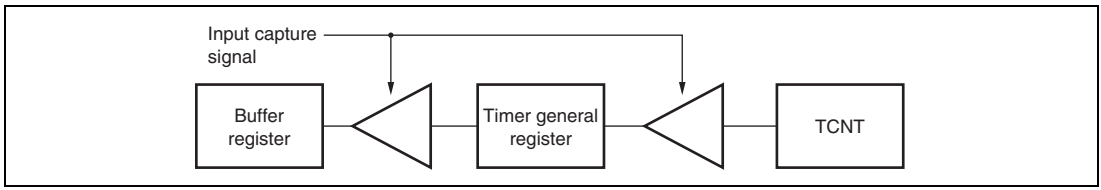


Figure 12.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 12.16 shows an example of the buffer operation setting procedure.

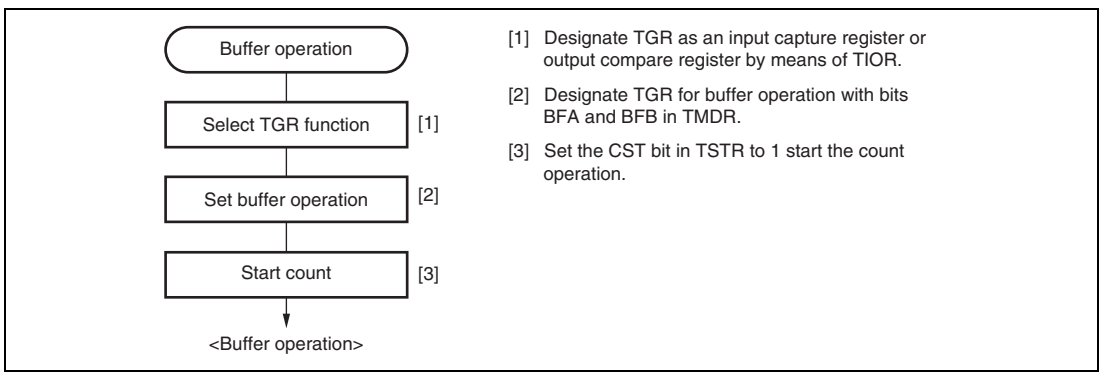


Figure 12.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an output compare register

Figure 12.17 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. In this example, the TTSA bit in TBTM is cleared to 0.

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 12.4.5, PWM Modes.

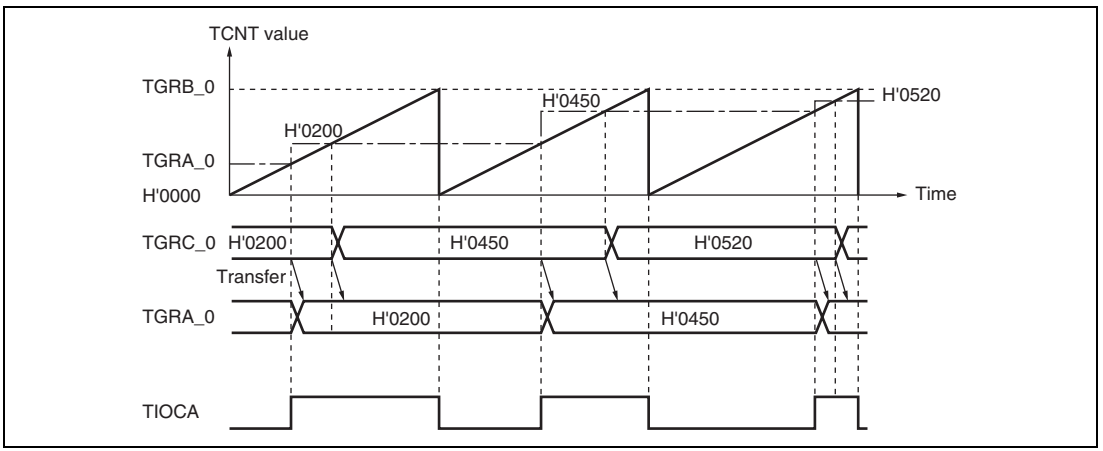


Figure 12.17 Example of Buffer Operation (1)

(b) When TGR is an input capture register

Figure 12.18 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.

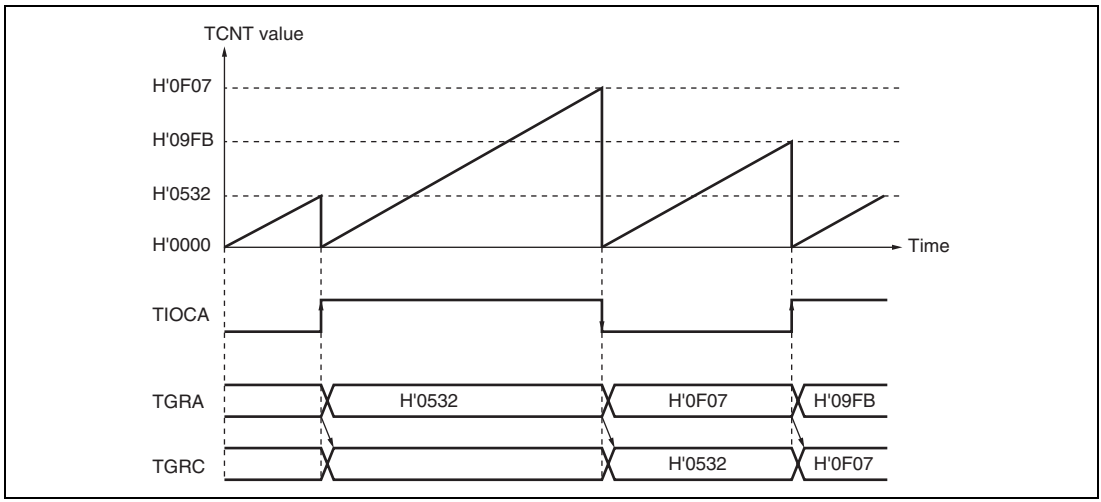


Figure 12.18 Example of Buffer Operation (2)

(3) Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation

The timing for transfer from buffer registers to timer general registers can be selected in PWM mode 1 or 2 for channel 0 or in PWM mode 1 for channels 3 and 4 by setting the buffer operation transfer mode registers (TBTM_0, TBTM_3, and TBTM_4). Either compare match (initial setting) or TCNT clearing can be selected for the transfer timing. TCNT clearing as transfer timing is one of the following cases.

- When TCNT overflows (H'FFFF to H'0000)
- When H'0000 is written to TCNT during counting
- When TCNT is cleared to H'0000 under the condition specified in the CCLR2 to CCLR0 bits in TCR

Note: TBTM must be modified only while TCNT stops.

Figure 12.19 shows an operation example in which PWM mode 1 is designated for channel 0 and buffer operation is designated for TGRA_0 and TGRC_0. The settings used in this example are TCNT_0 clearing by compare match B, 1 output at compare match A, and 0 output at compare match B. The TTSA bit in TBTM_0 is set to 1.

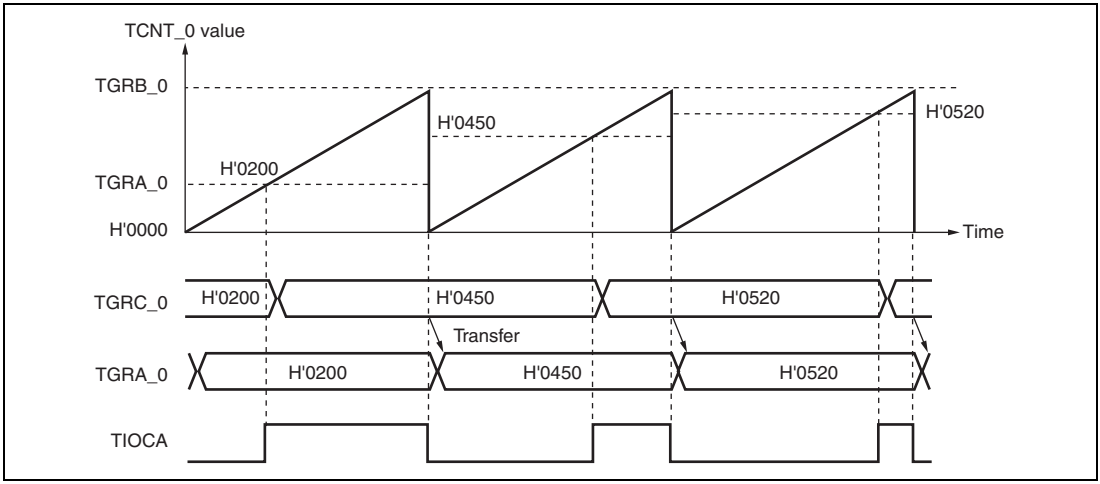


Figure 12.19 Example of Buffer Operation When TCNT_0 Clearing is Selected for TGRC_0 to TGRA_0 Transfer Timing

12.4.4 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 12.42 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1, the counter clock setting is invalid and the counters operates independently in phase counting mode.

Table 12.42 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2

For simultaneous input capture of TCNT_1 and TCNT_2 during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). The edge detection that is the condition for input capture uses a signal representing the logical OR of the original input pin and the added input pins. For details, see (4) Cascaded Operation Example (c).

For input capture in cascade connection, refer to section 12.7.22, Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection.

Table 12.43 shows the TICCR setting and input capture input pins.

Table 12.43 TICCR Setting and Input Capture Input Pins

Target Input Capture	TICCR Setting	Input Capture Input Pins
Input capture from TCNT_1 to TGRA_1	I2AE bit = 0 (initial value)	TIOC1A
	I2AE bit = 1	TIOC1A, TIOC2A
Input capture from TCNT_1 to TGRB_1	I2BE bit = 0 (initial value)	TIOC1B
	I2BE bit = 1	TIOC1B, TIOC2B
Input capture from TCNT_2 to TGRA_2	I1AE bit = 0 (initial value)	TIOC2A
	I1AE bit = 1	TIOC2A, TIOC1A
Input capture from TCNT_2 to TGRB_2	I1BE bit = 0 (initial value)	TIOC2B
	I1BE bit = 1	TIOC2B, TIOC1B

(1) Example of Cascaded Operation Setting Procedure

Figure 12.20 shows an example of the setting procedure for cascaded operation.

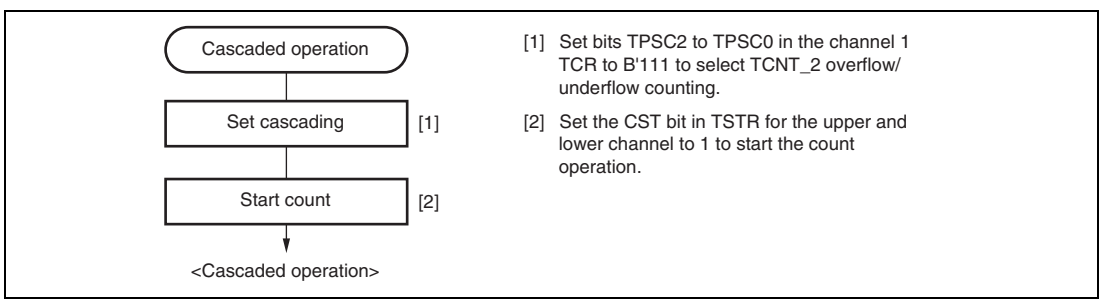


Figure 12.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 12.21 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.

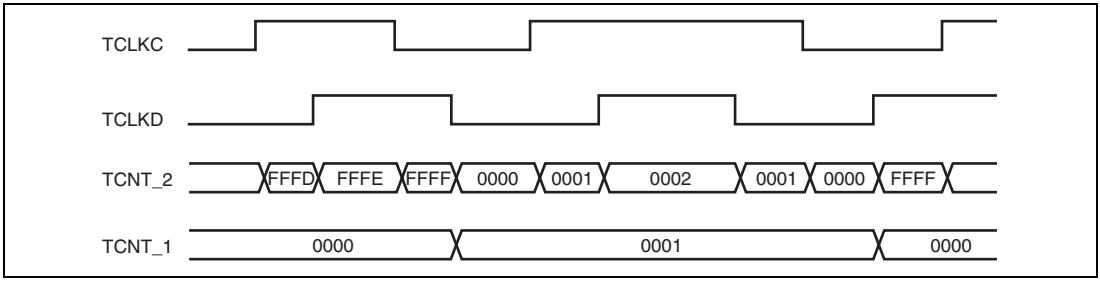


Figure 12.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 12.22 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE bit in TICCR has been set to 1 to include the TIOC2A pin in the TGRA_1 input capture conditions. In this example, the IOA0 to IOA3 bits in TIOR_1 have selected the TIOC1A rising edge for the input capture timing while the IOA0 to IOA3 bits in TIOR_2 have selected the TIOC2A rising edge for the input capture timing.

Under these conditions, the rising edge of both TIOC1A and TIOC2A is used for the TGRA_1 input capture condition. For the TGRA_2 input capture condition, the TIOC2A rising edge is used.

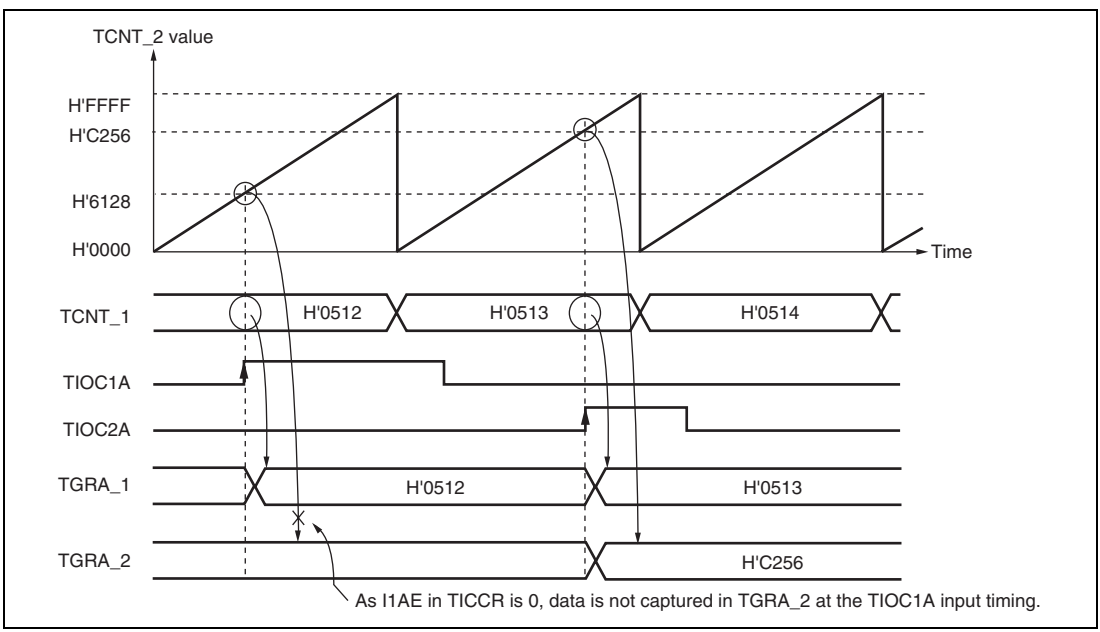


Figure 12.22 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 12.23 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE and I1AE bits in TICCR have been set to 1 to include the TIOC2A and TIOC1A pins in the TGRA_1 and TGRA_2 input capture conditions, respectively. In this example, the IOA0 to IOA3 bits in both TIOR_1 and TIOR_2 have selected both the rising and falling edges for the input capture timing. Under these conditions, the ORed result of TIOC1A and TIOC2A input is used for the TGRA_1 and TGRA_2 input capture conditions.

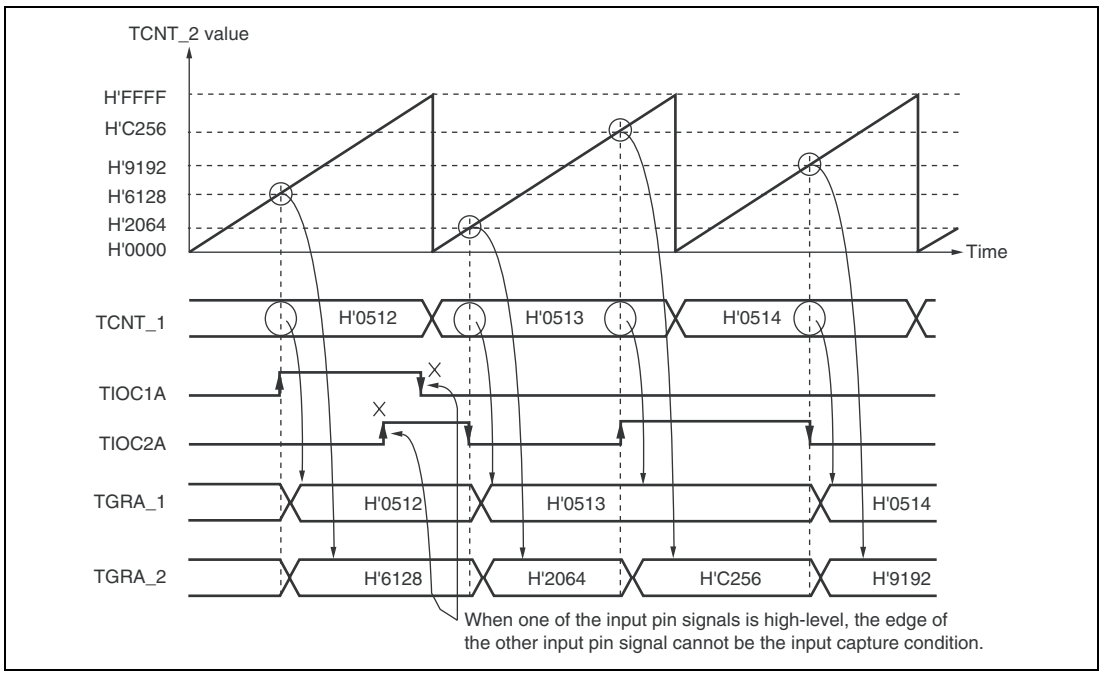


Figure 12.23 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 12.24 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE bit in TICCR has been set to 1 to include the TIOC2A pin in the TGRA_1 input capture conditions. In this example, the IOA0 to IOA3 bits in TIOR_1 have selected TGRA_0 compare match or input capture occurrence for the input capture timing while the IOA0 to IOA3 bits in TIOR_2 have selected the TIOC2A rising edge for the input capture timing.

Under these conditions, as TIOR_1 has selected TGRA_0 compare match or input capture occurrence for the input capture timing, the TIOC2A edge is not used for TGRA_1 input capture condition although the I2AE bit in TICCR has been set to 1.

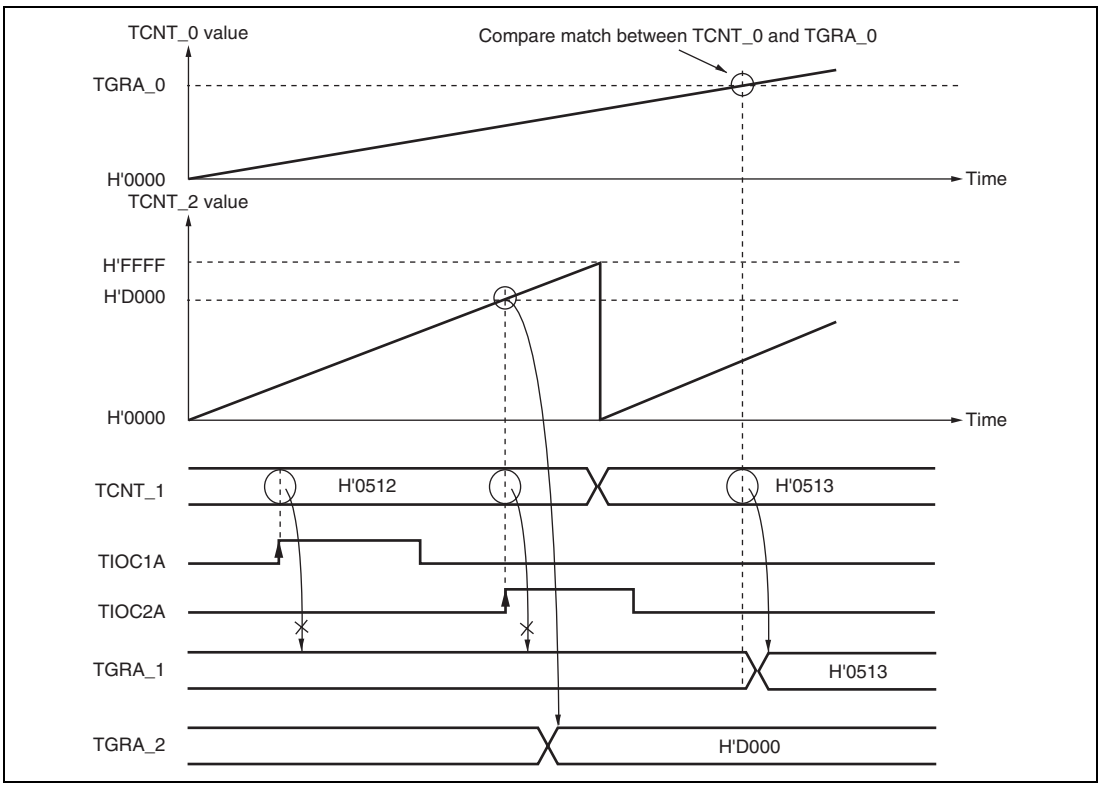


Figure 12.24 Cascaded Operation Example (d)

12.4.5 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 cycle 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 12.44.

Table 12.44 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	Cannot be set
	TGRB_3		Cannot be set
	TGRC_3	TIOC3C	Cannot be set
	TGRD_3		Cannot be set
4	TGRA_4	TIOC4A	Cannot be set
	TGRB_4		Cannot be set
	TGRC_4	TIOC4C	Cannot be set
	TGRD_4		Cannot be set

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

(1) Example of PWM Mode Setting Procedure

Figure 12.25 shows an example of the PWM mode setting procedure.

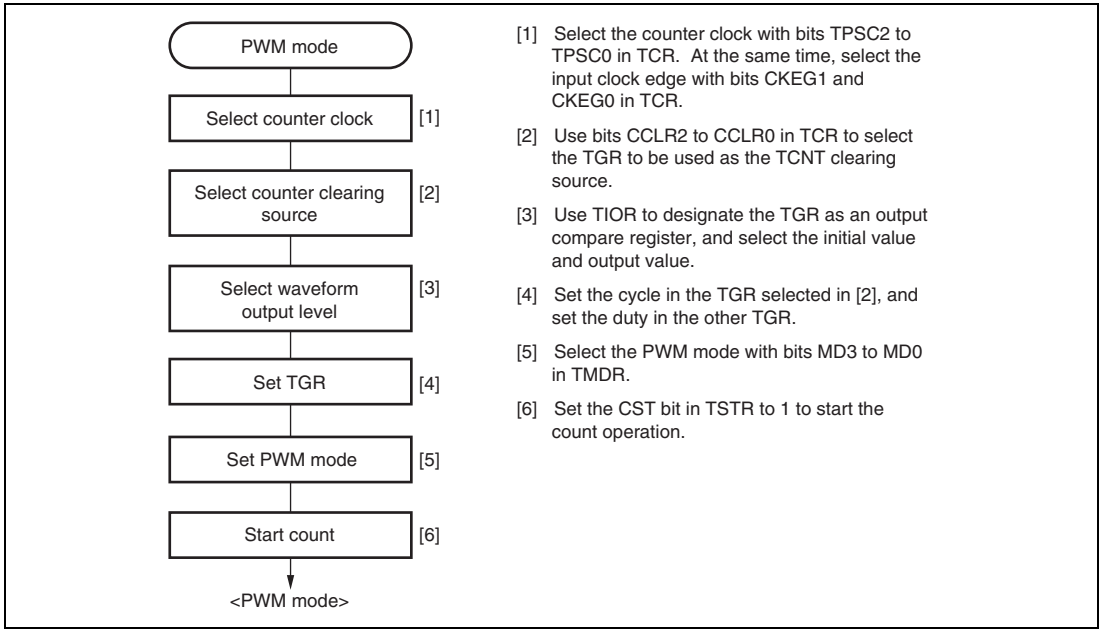


Figure 12.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 12.26 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 levels.

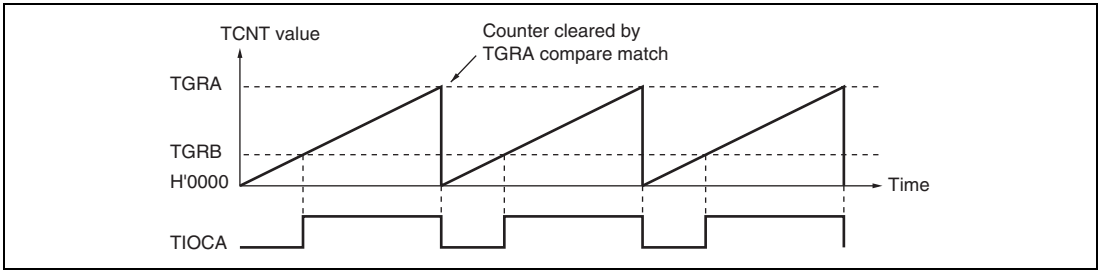


Figure 12.26 Example of PWM Mode Operation (1)

Figure 12.27 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.

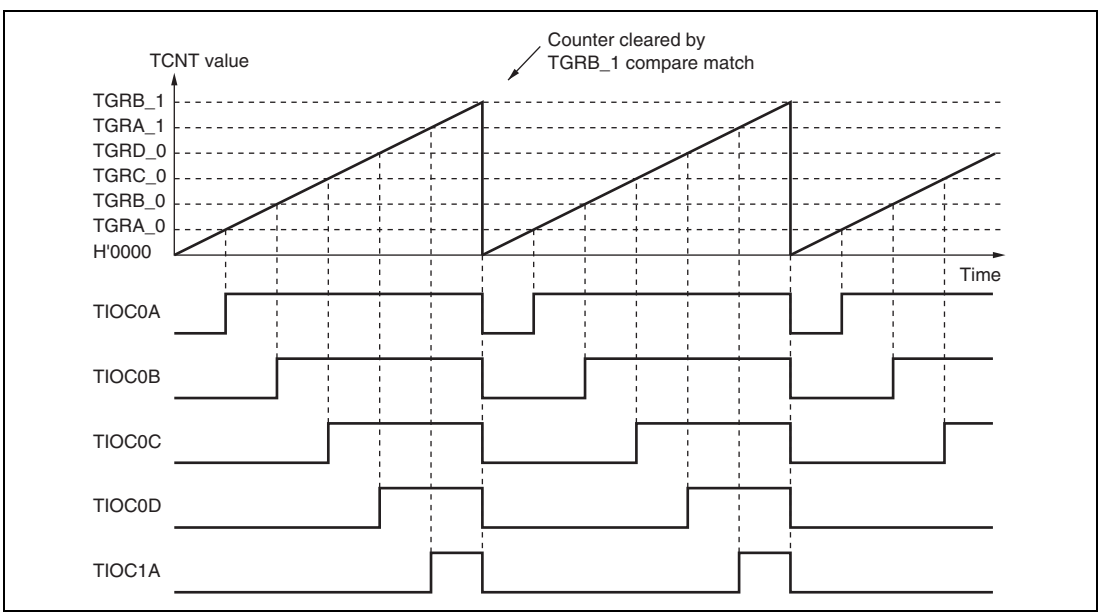


Figure 12.27 Example of PWM Mode Operation (2)

Figure 12.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

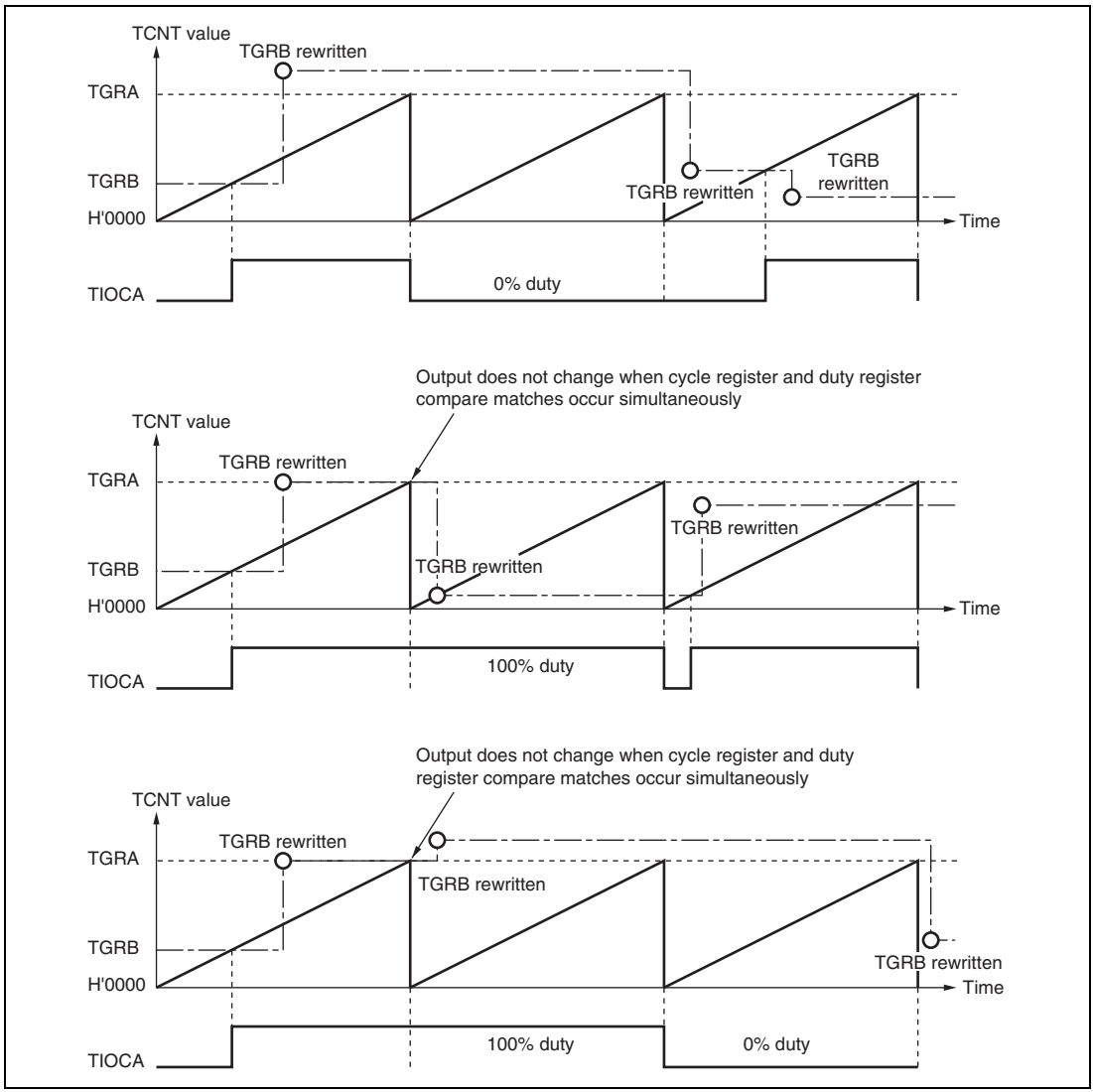


Figure 12.28 Example of PWM Mode Operation (3)

12.4.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1 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 12.45 shows the correspondence between external clock pins and channels.

Table 12.45 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

(1) Example of Phase Counting Mode Setting Procedure

Figure 12.29 shows an example of the phase counting mode setting procedure.

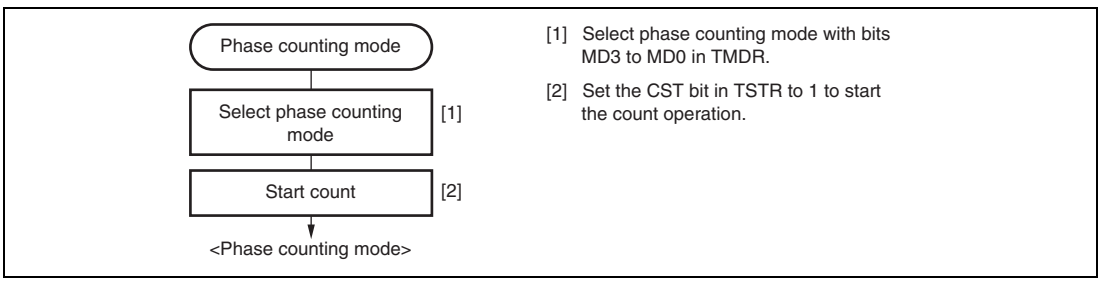


Figure 12.29 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

(a) Phase counting mode 1

Figure 12.30 shows an example of phase counting mode 1 operation, and table 12.46 summarizes the TCNT up/down-count conditions.

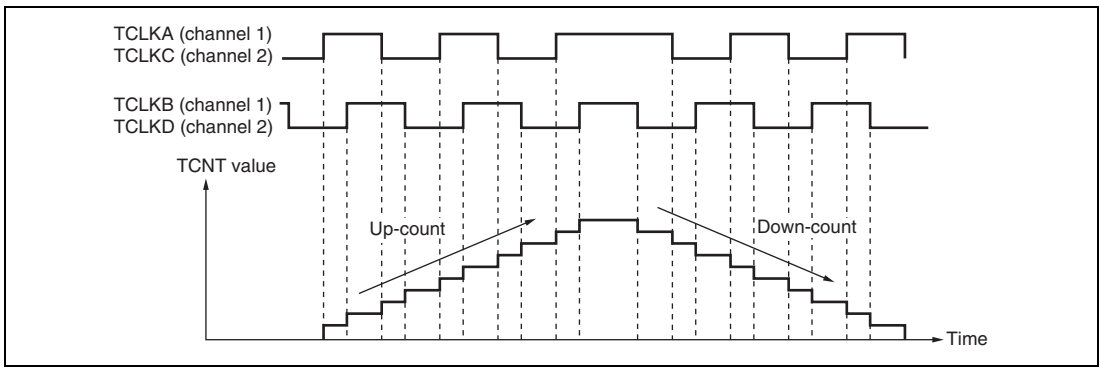


Figure 12.30 Example of Phase Counting Mode 1 Operation

Table 12.46 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

(b) Phase counting mode 2

Figure 12.31 shows an example of phase counting mode 2 operation, and table 12.47 summarizes the TCNT up/down-count conditions.

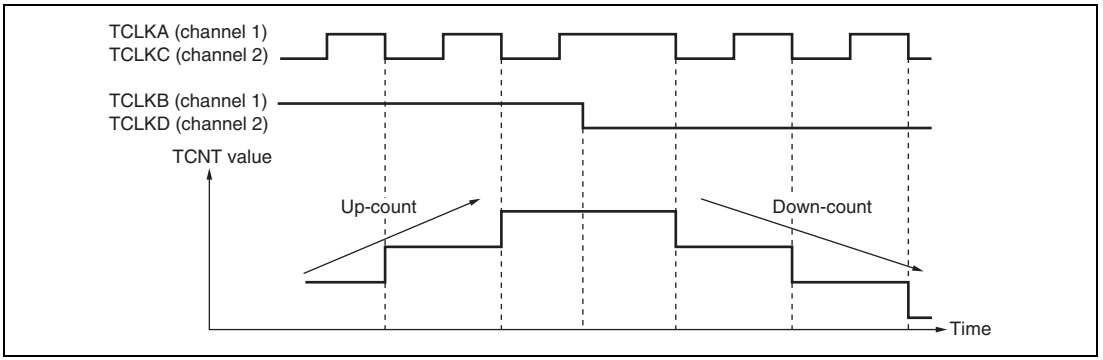


Figure 12.31 Example of Phase Counting Mode 2 Operation

Table 12.47 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

(c) Phase counting mode 3

Figure 12.32 shows an example of phase counting mode 3 operation, and table 12.48 summarizes the TCNT up/down-count conditions.

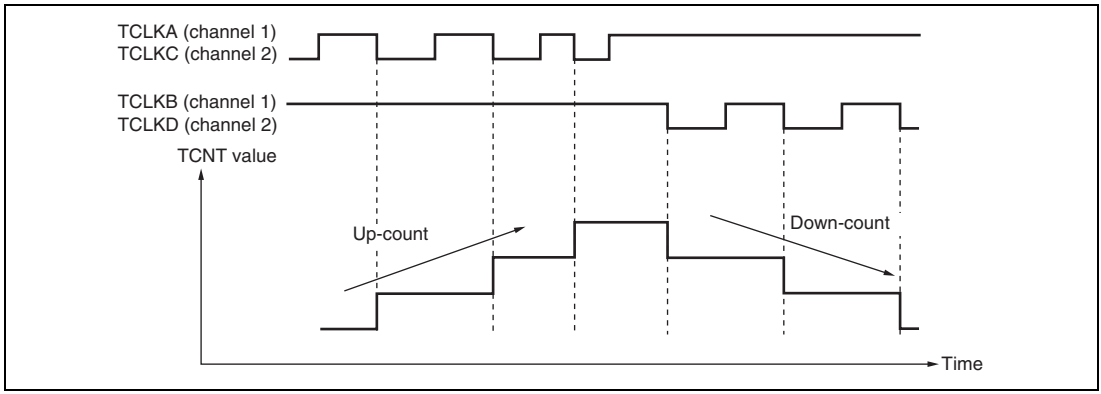


Figure 12.32 Example of Phase Counting Mode 3 Operation

Table 12.48 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

(d) Phase counting mode 4

Figure 12.33 shows an example of phase counting mode 4 operation, and table 12.49 summarizes the TCNT up/down-count conditions.

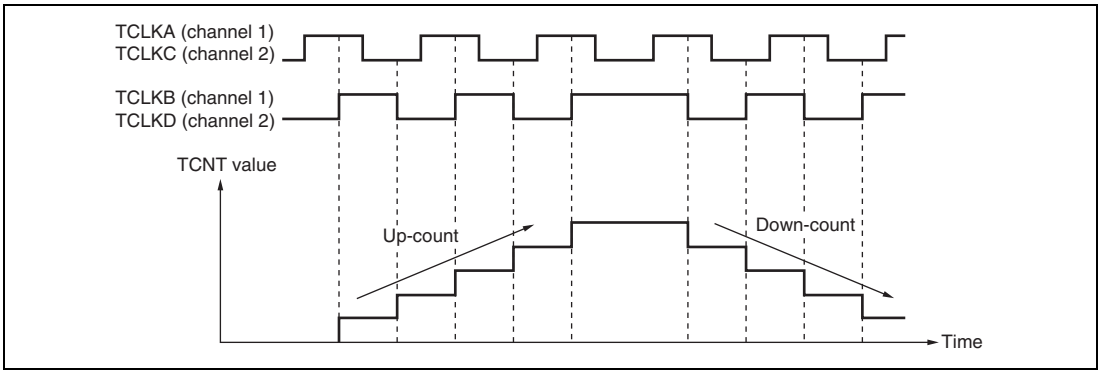


Figure 12.33 Example of Phase Counting Mode 4 Operation

Table 12.49 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

(3) Phase Counting Mode Application Example

Figure 12.34 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.

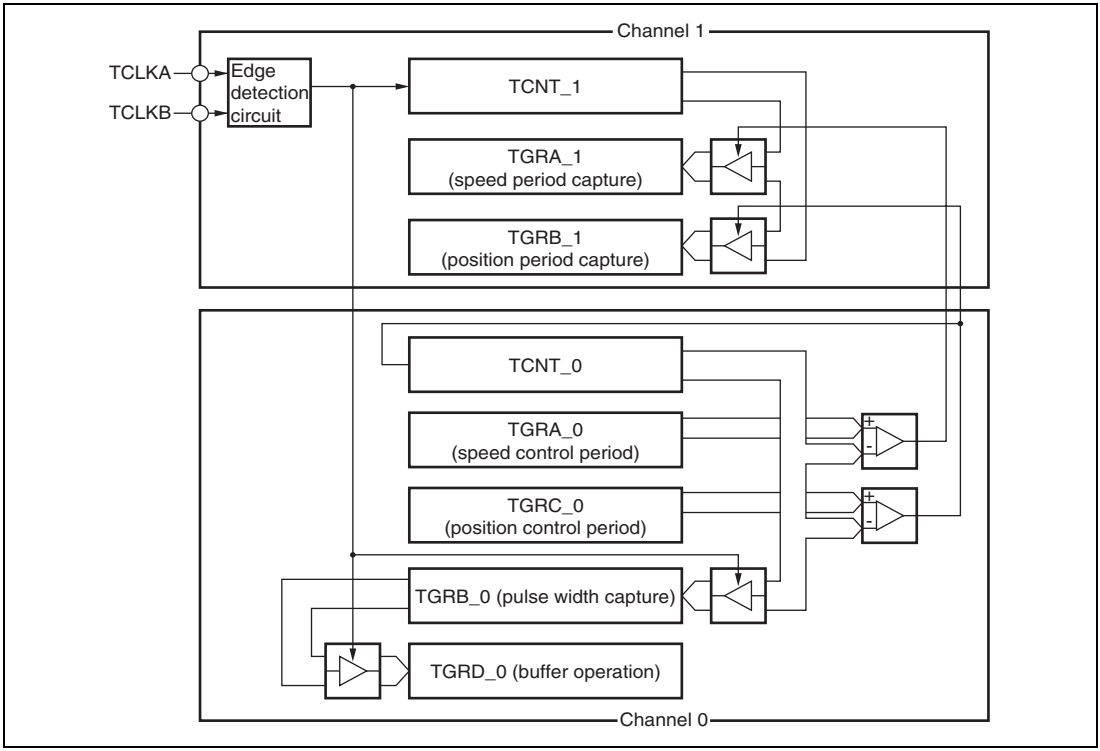


Figure 12.34 Phase Counting Mode Application Example

12.4.7 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 12.50 shows the PWM output pins used. Table 12.51 shows the settings of the registers.

Table 12.50 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 12.51 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

(1) Procedure for Selecting the Reset-Synchronized PWM Mode

Figure 12.35 shows an example of procedure for selecting the reset synchronized PWM mode.

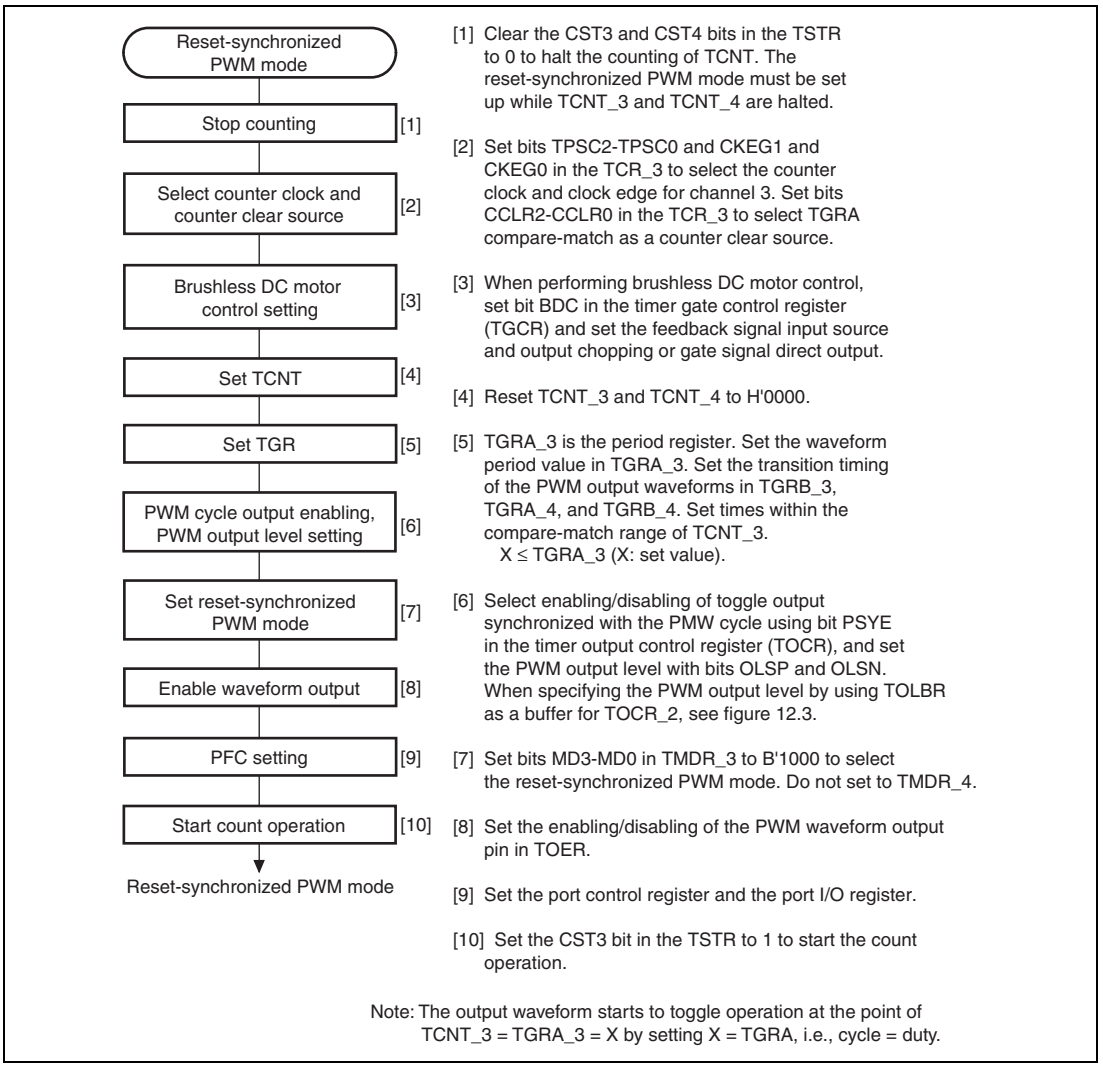


Figure 12.35 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Reset-Synchronized PWM Mode Operation

Figure 12.36 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 incrementing 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.

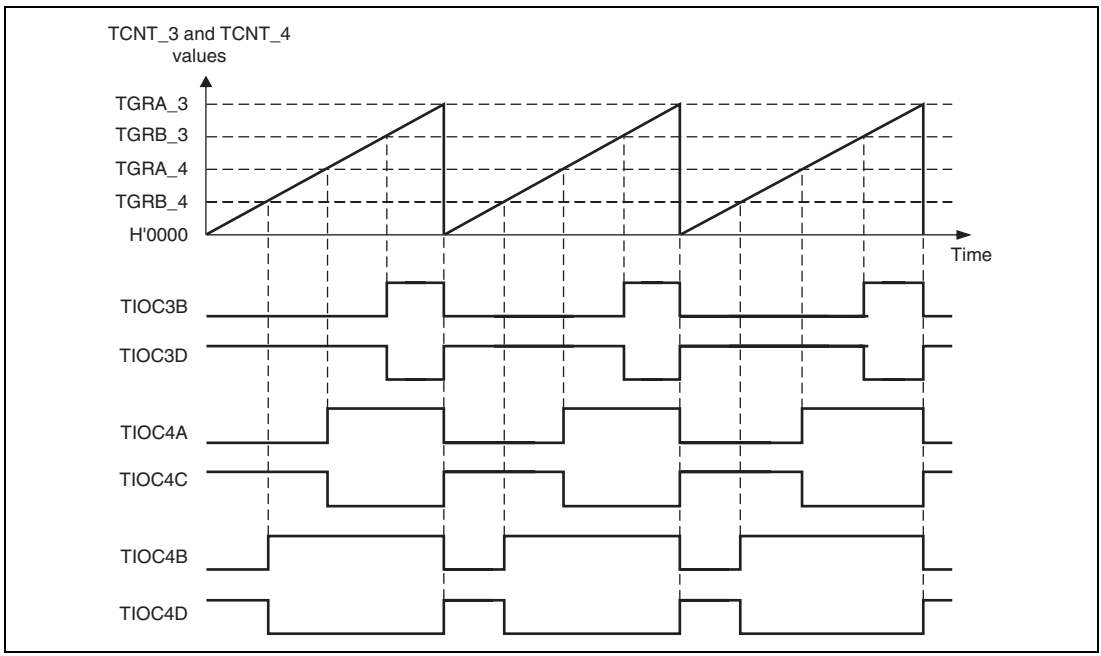


Figure 12.36 Reset-Synchronized PWM Mode Operation Example (When TOCR's OLSN = 1 and OLSP = 1)

12.4.8 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. PWM waveforms without non-overlapping interval are also available.

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 up/down counters.

Table 12.52 shows the PWM output pins used. Table 12.53 shows the settings of the registers used.

Table 12.52 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
3	TIOC3A	Toggle output synchronized with PWM period (or I/O port)
	TIOC3B	PWM output pin 1
	TIOC3C	I/O port*
	TIOC3D	PWM output pin 1' (non-overlapping negative-phase waveform of PWM output 1; PWM output without non-overlapping interval is also available)
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; PWM output without non-overlapping interval is also available)
	TIOC4D	PWM output pin 3' (non-overlapping negative-phase waveform of PWM output 3; PWM output without non-overlapping interval is also available)

Note: * Avoid setting the TIOC3C pin as a timer I/O pin in the complementary PWM mode.

Table 12.53 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 TRWER setting*
	TGRA_3	Set TCNT_3 upper limit value (1/2 carrier cycle + dead time)	Maskable by TRWER setting*
	TGRB_3	PWM output 1 compare register	Maskable by TRWER 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 TRWER setting*
	TGRA_4	PWM output 2 compare register	Maskable by TRWER setting*
	TGRB_4	PWM output 3 compare register	Maskable by TRWER 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 TRWER setting*
Timer cycle data register (TCDR)		Set TCNT_4 upper limit value (1/2 carrier cycle)	Maskable by TRWER 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 (RWE) in TRWER (timer read/write enable register).

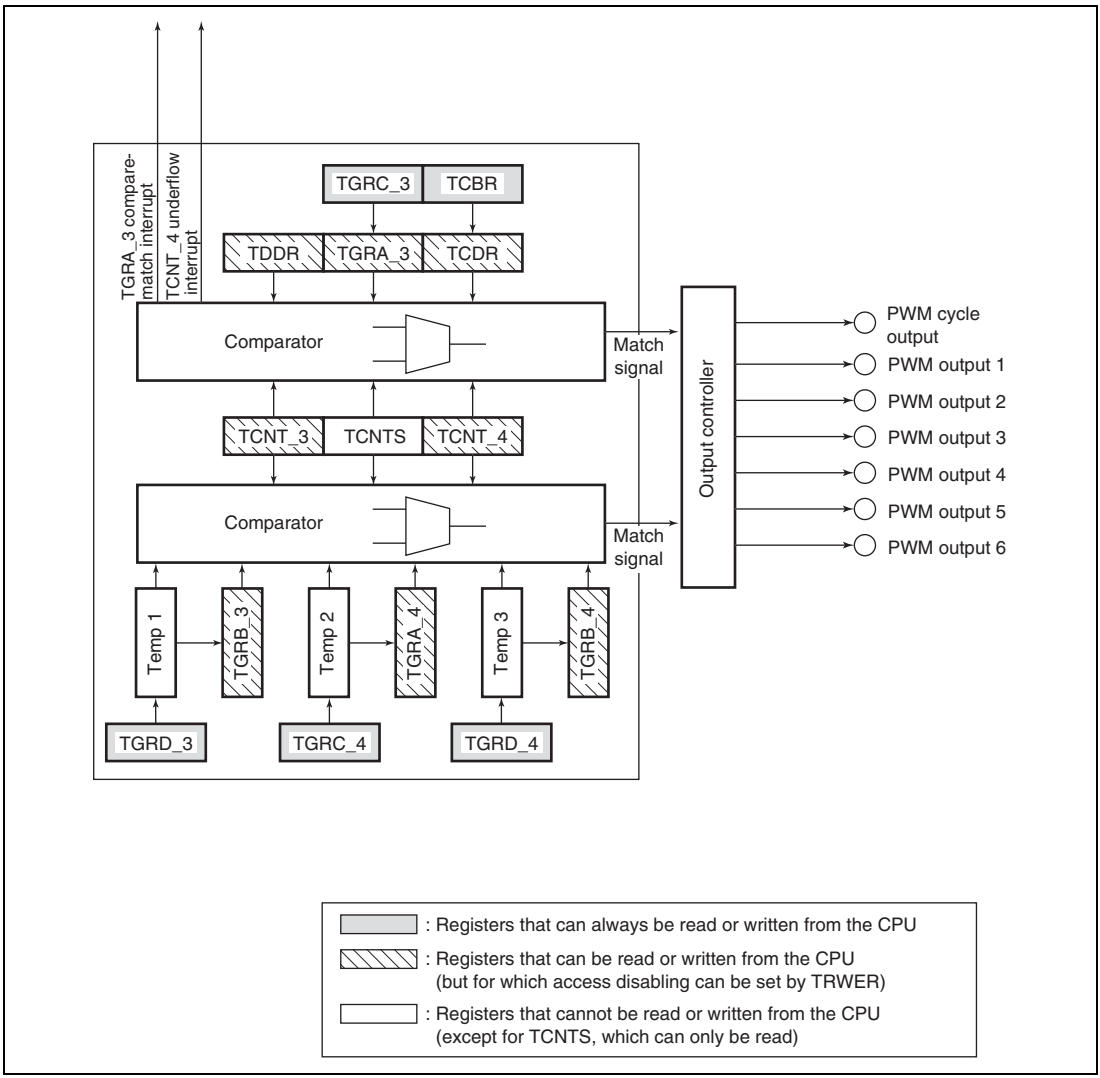
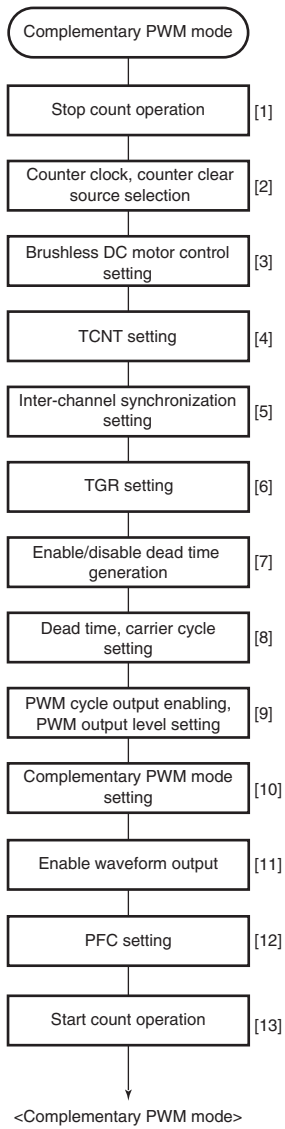


Figure 12.37 Block Diagram of Channels 3 and 4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

An example of the complementary PWM mode setting procedure is shown in figure 12.38.



[1] Clear bits CST3 and CST4 in the timer start register (TSTR) to 0, and halt timer counter (TCNT) operation. Perform complementary PWM mode setting when TCNT_3 and TCNT_4 are stopped.

[2] Set the same counter clock and clock edge for channels 3 and 4 with bits TPSC2-TPSC0 and bits CKEG1 and CKEG0 in the timer control register (TCR). Use bits CCLR2-CCLR0 to set synchronous clearing only when restarting by a synchronous clear from another channel during complementary PWM mode operation.

[3] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Set the dead time in TCNT_3. Set TCNT_4 to H'0000.

[5] Set only when restarting by a synchronous clear from another channel during complementary PWM mode operation. In this case, synchronize the channel generating the synchronous clear with channels 3 and 4 using the timer synchro register (TSYR).

[6] Set the output PWM duty in the duty registers (TGRB_3, TGRA_4, TGRB_4) and buffer registers (TGRD_3, TGRC_4, TGRD_4). Set the same initial value in each corresponding TGR.

[7] This setting is necessary only when no dead time should be generated. Make appropriate settings in the timer dead time enable register (TDER) so that no dead time is generated.

[8] Set the dead time in the dead time register (TDDR), 1/2 the timer cycle in the carrier cycle data register (TCDR) and timer cycle buffer register (TCBR), and 1/2 the carrier cycle plus the dead time in TGRA_3 and TGRC_3. When no dead time generation is selected, set 1 in TDDR and 1/2 the carrier cycle + 1 in TGRA_3 and TGRC_3.

[9] Select enabling/disabling of toggle output synchronized with the PWM cycle using bit PSYE in the timer output control register 1 (TOCR1), and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBR as a buffer for TOCR_2, see figure 12.3.

[10] Select complementary PWM mode in timer mode register 3 (TMDR_3). Do not set in TMDR_4.

[11] Set enabling/disabling of PWM waveform output pin output in the timer output master enable register (TOER).

[12] Set the port control register and the port I/O register.

[13] Set bits CST3 and CST4 in TSTR to 1 simultaneously to start the count operation.

Figure 12.38 Example of Complementary PWM Mode Setting Procedure

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, 6-phase PWM output is possible. Figure 12.39 illustrates counter operation in complementary PWM mode, and figure 12.40 shows an example of complementary PWM mode operation.

(a) 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.

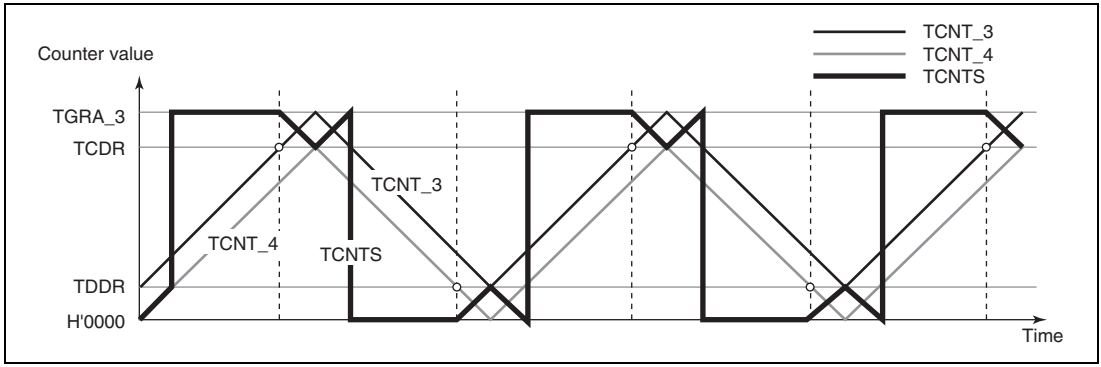


Figure 12.39 Complementary PWM Mode Counter Operation

(b) Register Operation

In complementary PWM mode, nine registers are used, comprising compare registers, buffer registers, and temporary registers. Figure 12.40 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 12.40 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 12.40) 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.

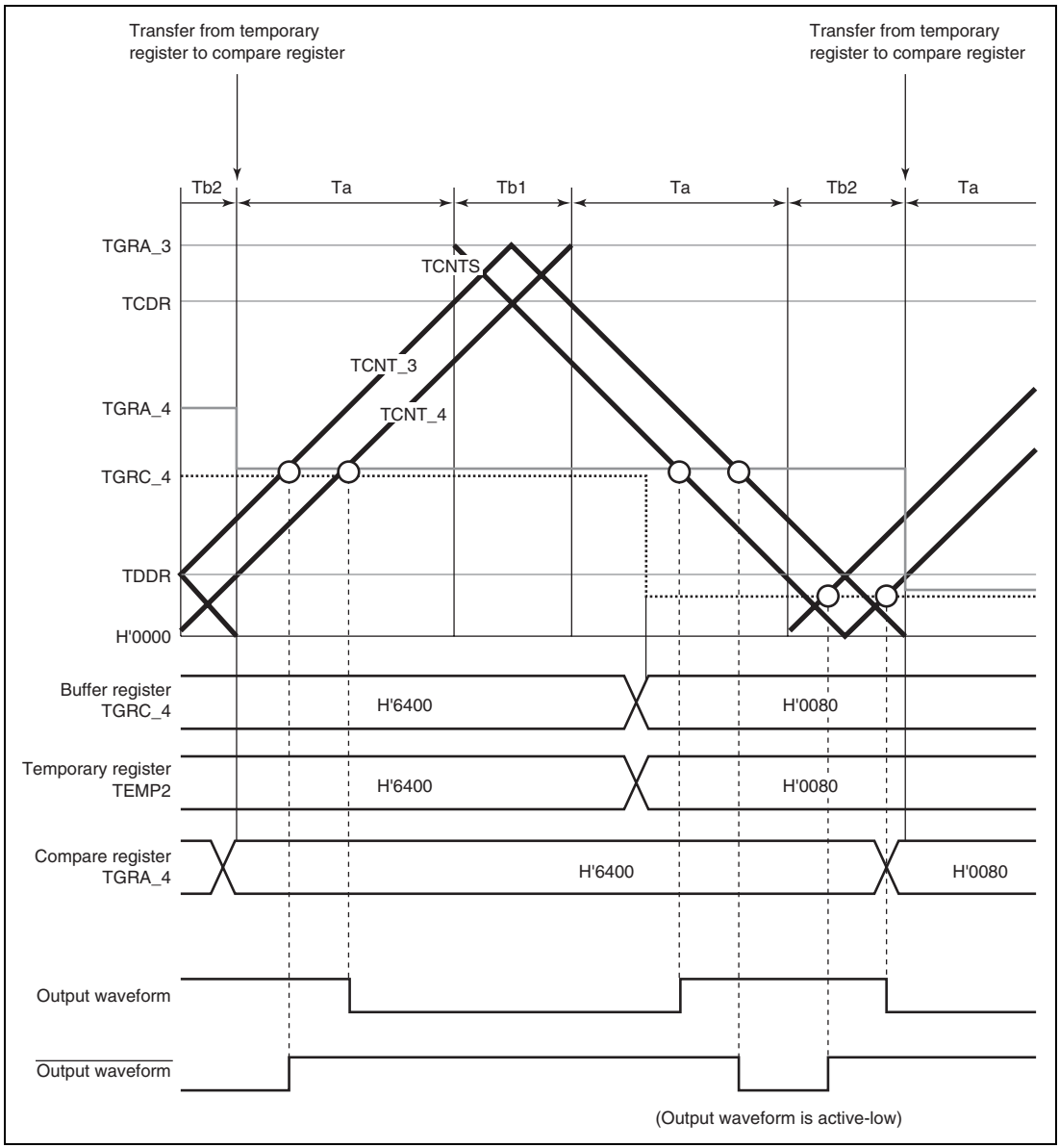


Figure 12.40 Example of Complementary PWM Mode Operation

(c) Initialization

In complementary PWM mode, there are six registers that must be initialized. In addition, there is a register that specifies whether to generate dead time (it should be used only when dead time generation should be disabled).

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).

When dead time is not needed, the TDER bit in the timer dead time enable register (TDER) should be cleared to 0, TGRC_3 and TGRA_3 should be set to 1/2 the PWM carrier cycle + 1, and TDDR should be set to 1.

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 12.54 Registers and Counters Requiring Initialization

Register/Counter	Set Value
TGRC_3	1/2 PWM carrier cycle + dead time Td (1/2 PWM carrier cycle + 1 when dead time generation is disabled by TDER)
TDDR	Dead time Td (1 when dead time generation is disabled by TDER)
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. When dead time generation is disabled by TDER, TGRC_3 must be set to 1/2 the PWM carrier cycle + 1.

(d) PWM Output Level Setting

In complementary PWM mode, the PWM pulse output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2).

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.

(e) 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.

(f) Dead Time Suppressing

Dead time generation is suppressed by clearing the TDER bit in the timer dead time enable register (TDER) to 0. TDER can be cleared to 0 only when 0 is written to it after reading TDER = 1.

TGRA_3 and TGRC_3 should be set to 1/2 PWM carrier cycle + 1 and the timer dead time data register (TDDR) should be set to 1.

By the above settings, PWM waveforms without dead time can be obtained. Figure 12.41 shows an example of operation without dead time.

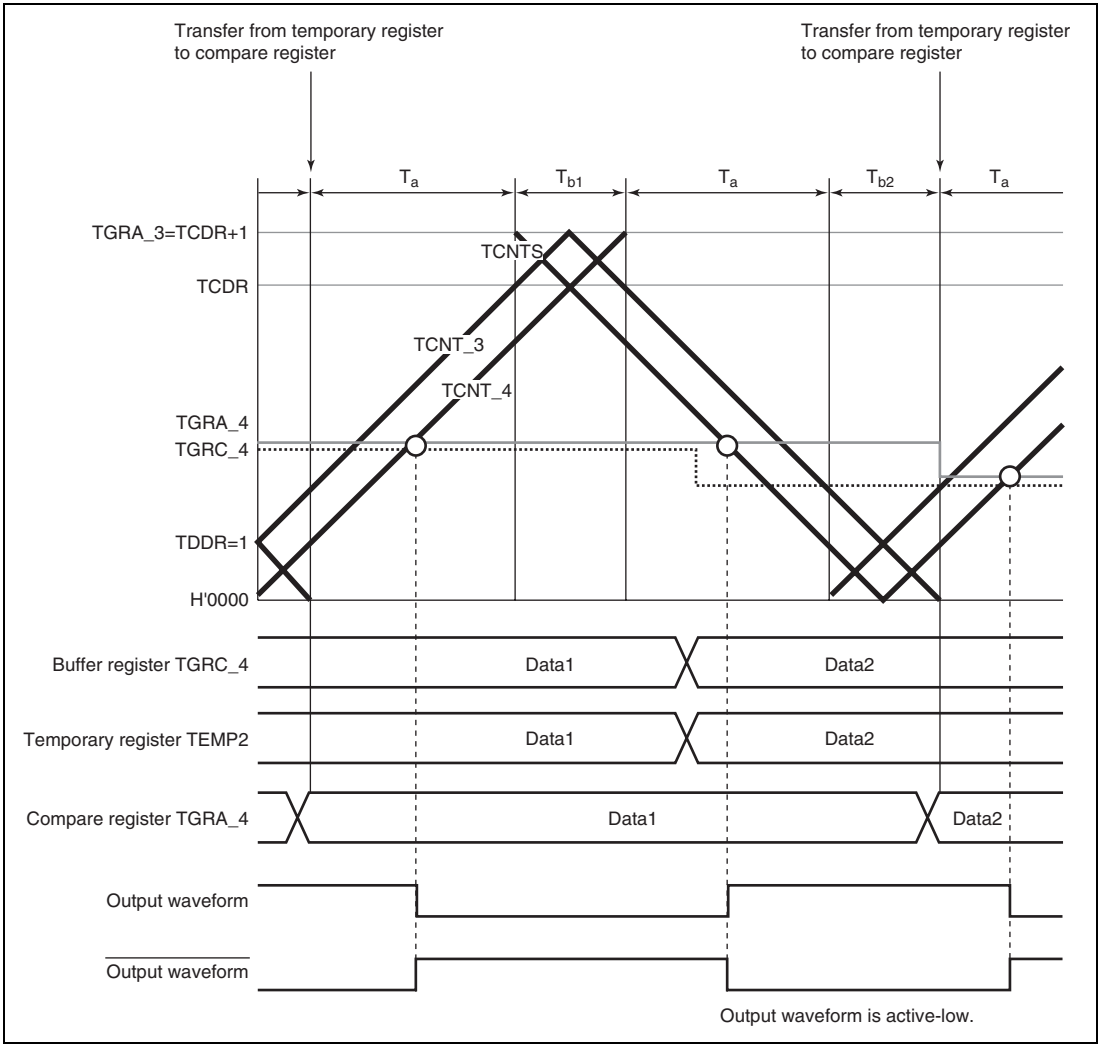


Figure 12.41 Example of Operation without Dead Time

(g) 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:

With dead time: TGRA_3 set value = TCDR set value + TDDR set value
TCDR set value > two times TDDR + 2
Without dead time: TGRA_3 set value = TCDR set value + 1
TCDR set value > 4

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 12.42 illustrates the operation when the PWM cycle is updated at the crest.

See the following section, Register Data Updating, for the method of updating the data in each buffer register.

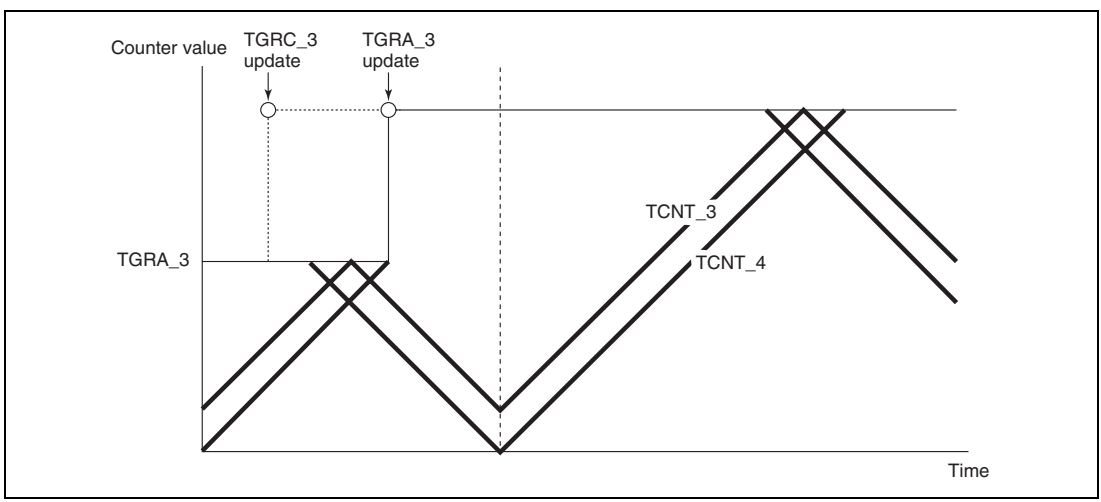


Figure 12.42 Example of PWM Cycle Updating

(h) 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 12.43 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.

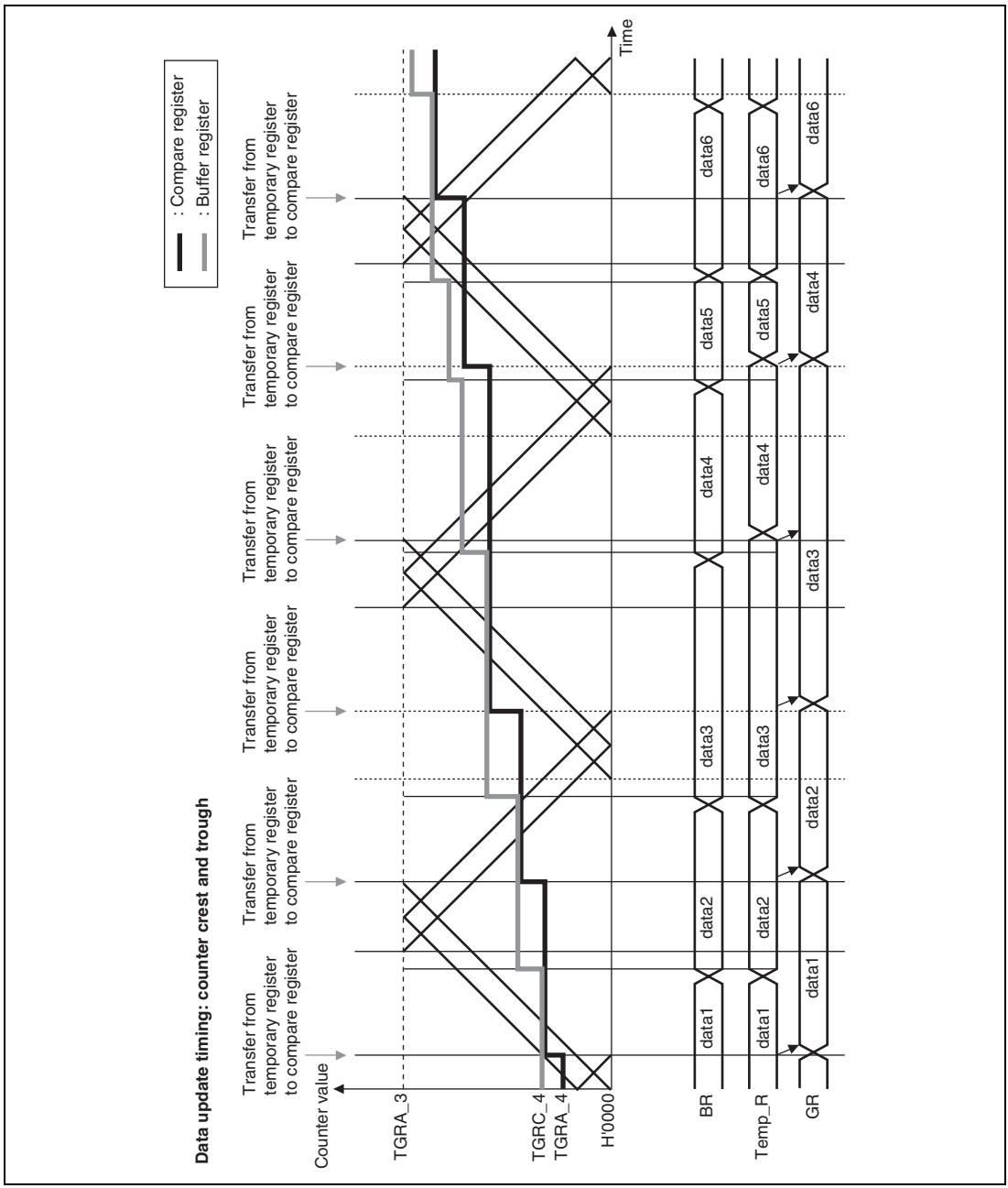


Figure 12.43 Example of Data Update in Complementary PWM Mode

(i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1N to OLS3N and OLS1P to OLS3P in timer output control register 2 (TOCR2).

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 12.44 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 12.45.

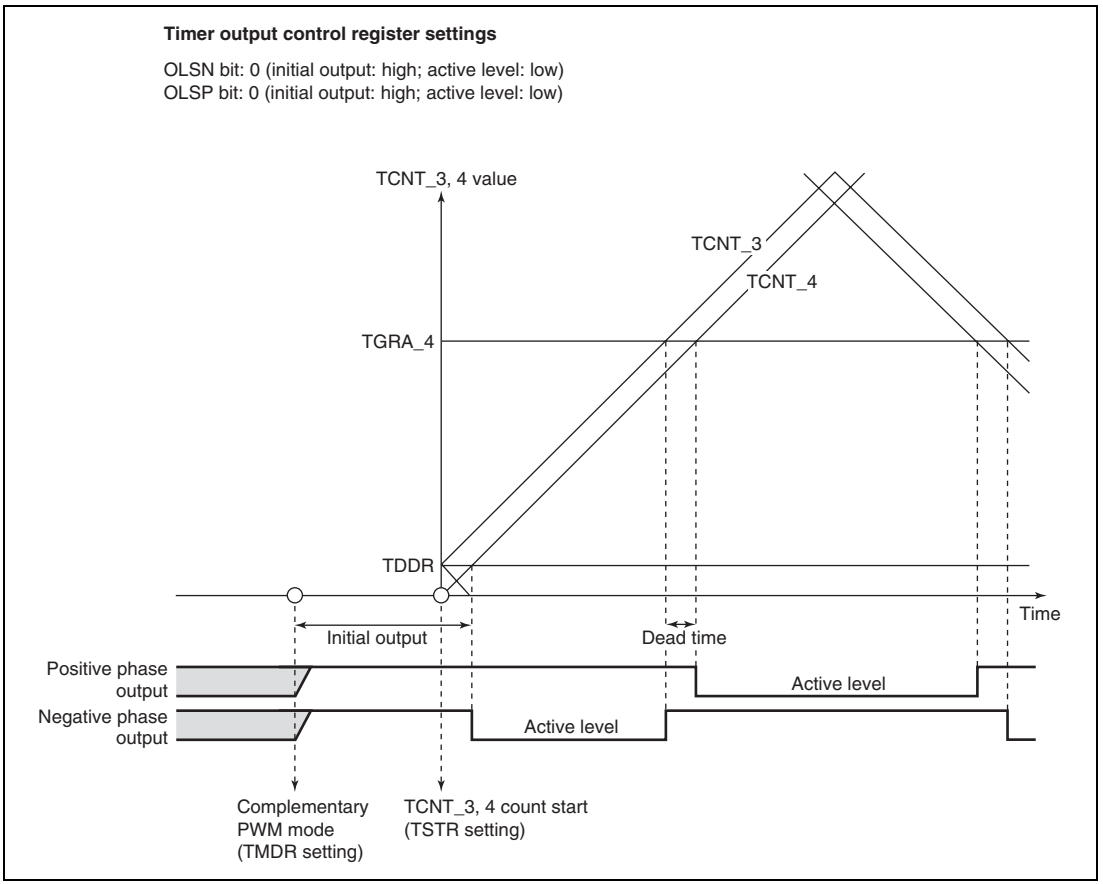


Figure 12.44 Example of Initial Output in Complementary PWM Mode (1)

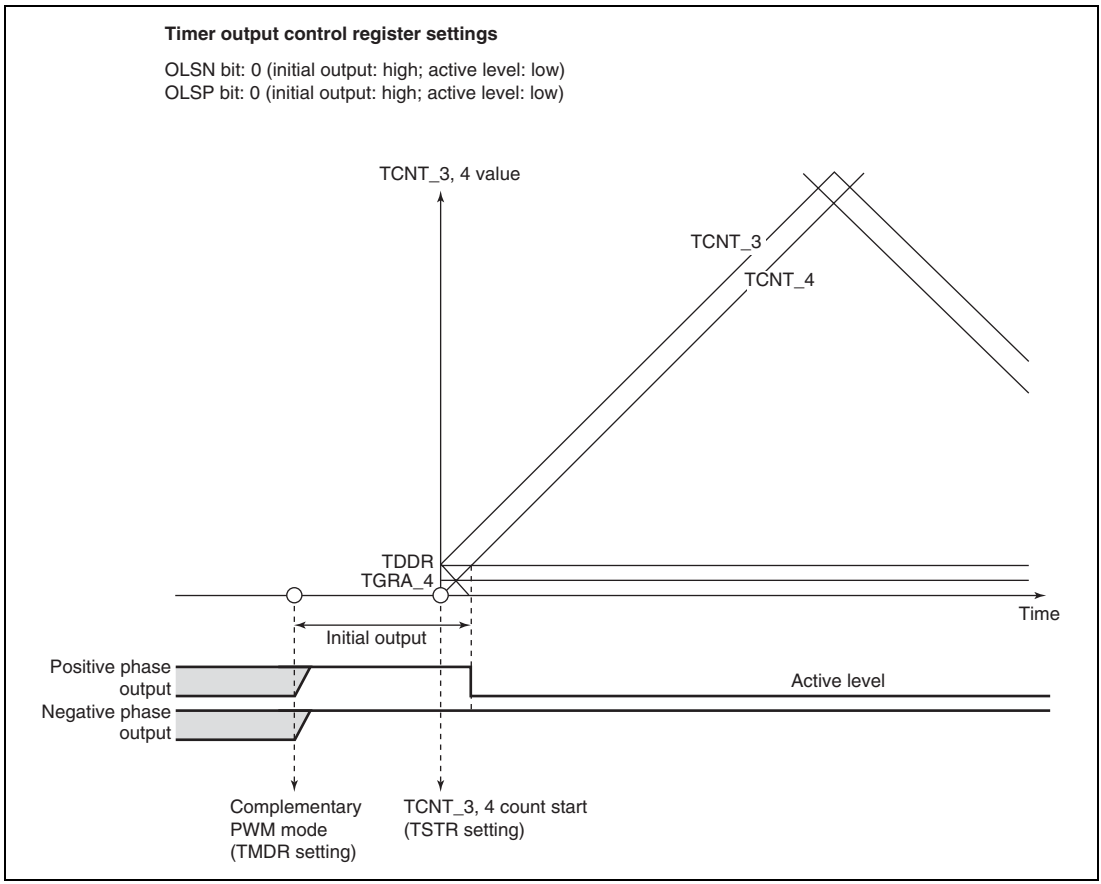


Figure 12.45 Example of Initial Output in Complementary PWM Mode (2)

(j) 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 compare register. While TCNTS is counting, compare 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 12.46 to 12.48 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 12.46.

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 12.47, 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 12.48, 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 negative 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.

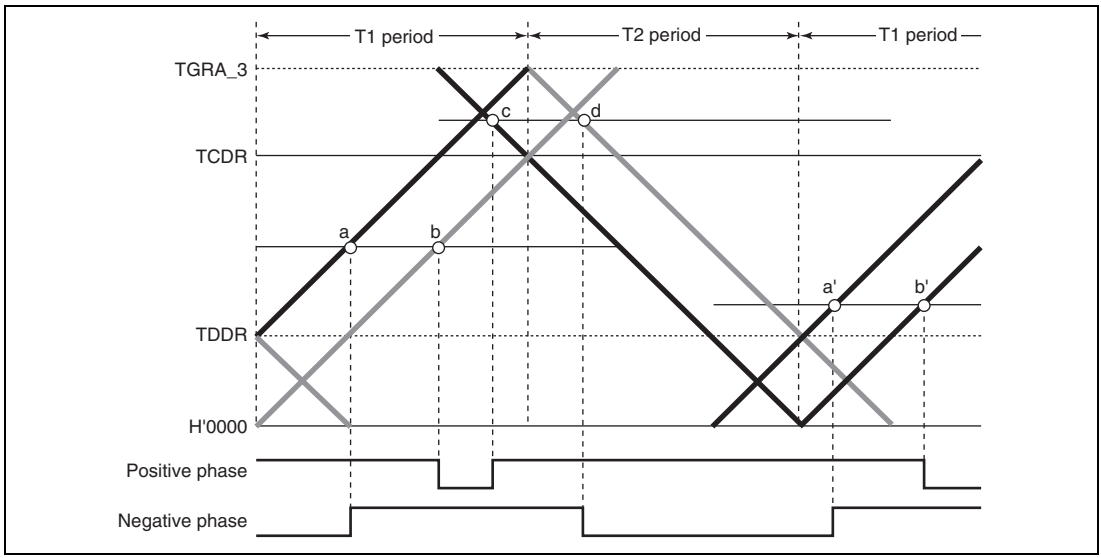


Figure 12.46 Example of Complementary PWM Mode Waveform Output (1)

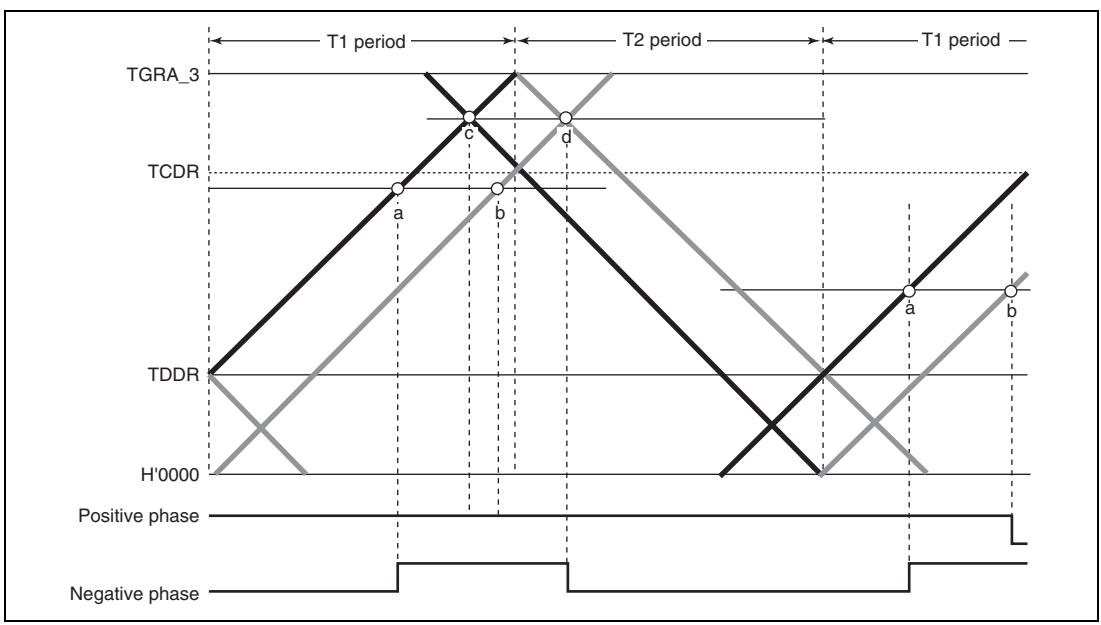


Figure 12.47 Example of Complementary PWM Mode Waveform Output (2)

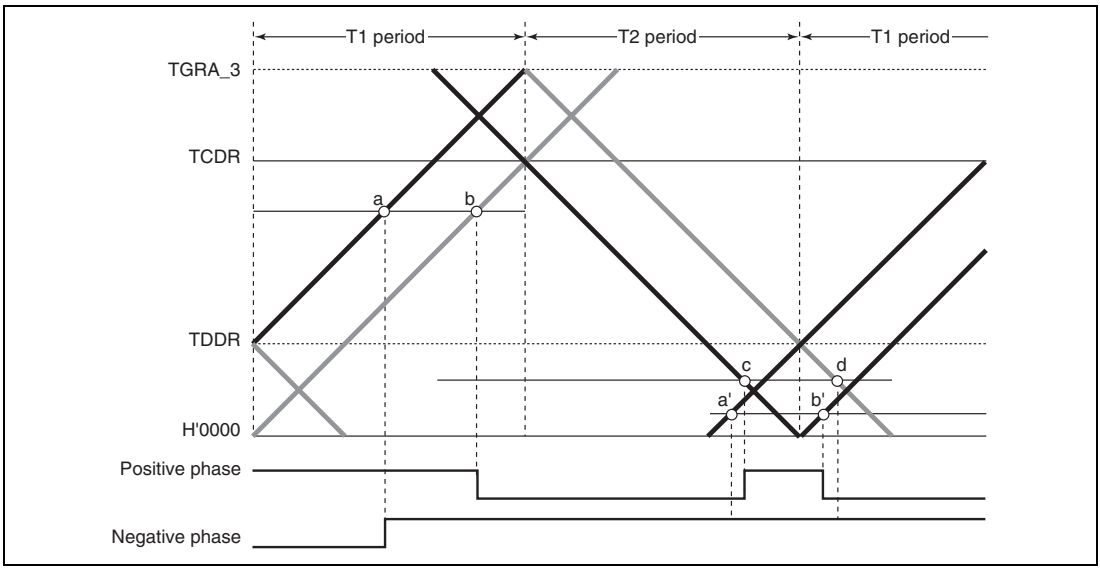


Figure 12.48 Example of Complementary PWM Mode Waveform Output (3)

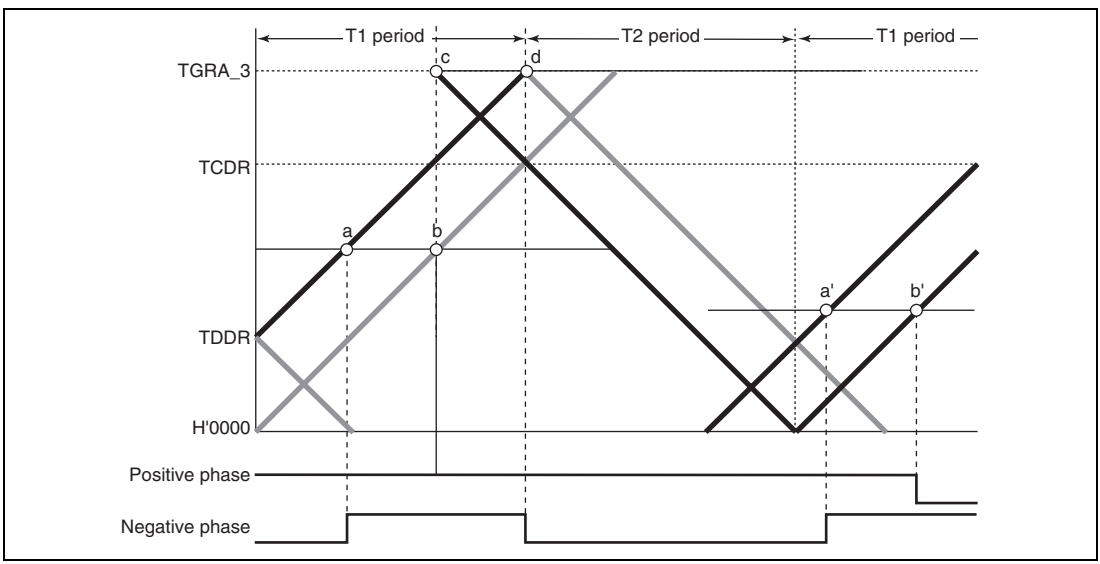


Figure 12.49 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)

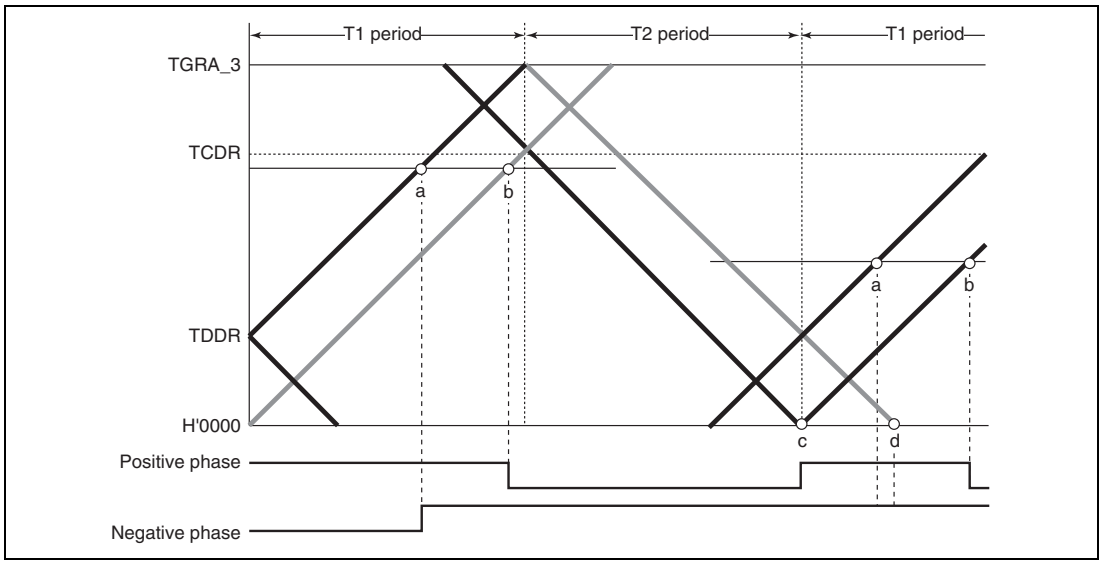


Figure 12.50 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)

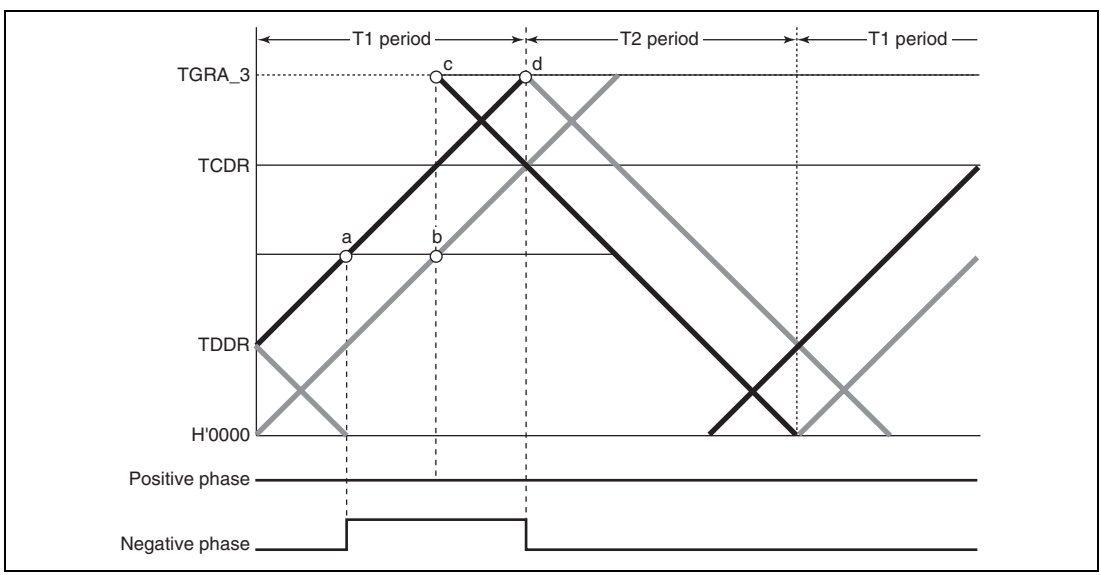


Figure 12.51 Example of Complementary PWM Mode 0% and 100% Waveform Output (3)

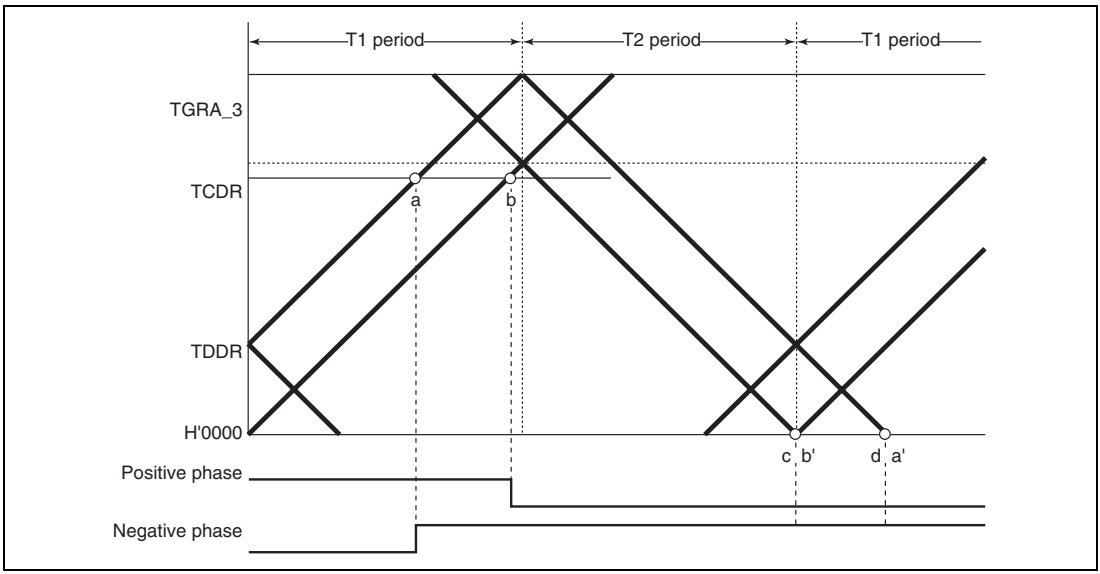


Figure 12.52 Example of Complementary PWM Mode 0% and 100% Waveform Output (4)

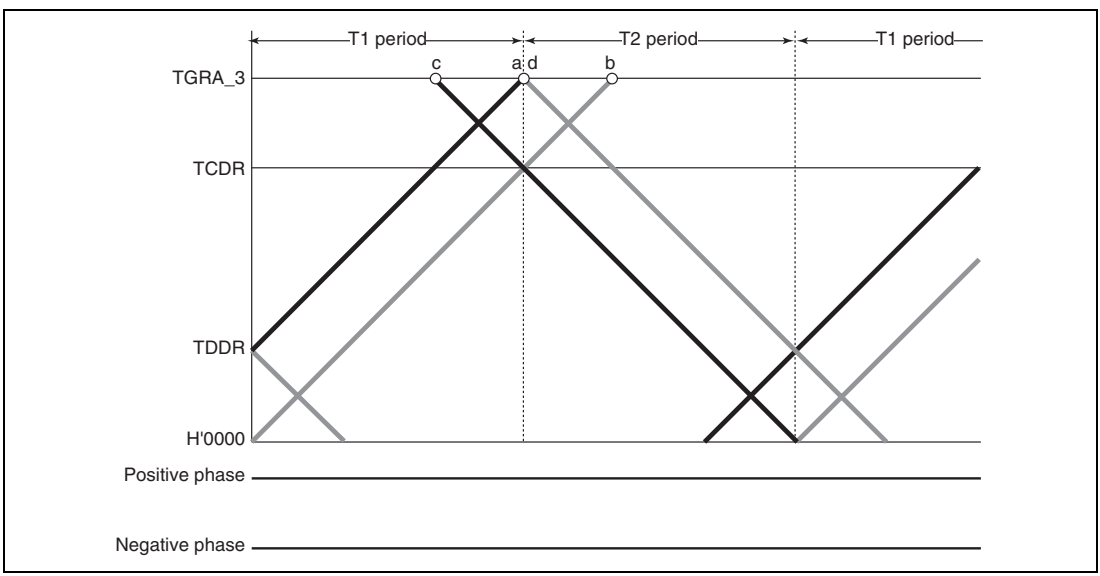


Figure 12.53 Example of Complementary PWM Mode 0% and 100% Waveform Output (5)

(k) Complementary PWM Mode 0% and 100% Duty Output

In complementary PWM mode, 0% and 100% duty cycles can be output as required. Figures 12.49 to 12.53 show output examples.

100% duty output is performed when the compare 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 compare 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.

(l) 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 12.54.

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.

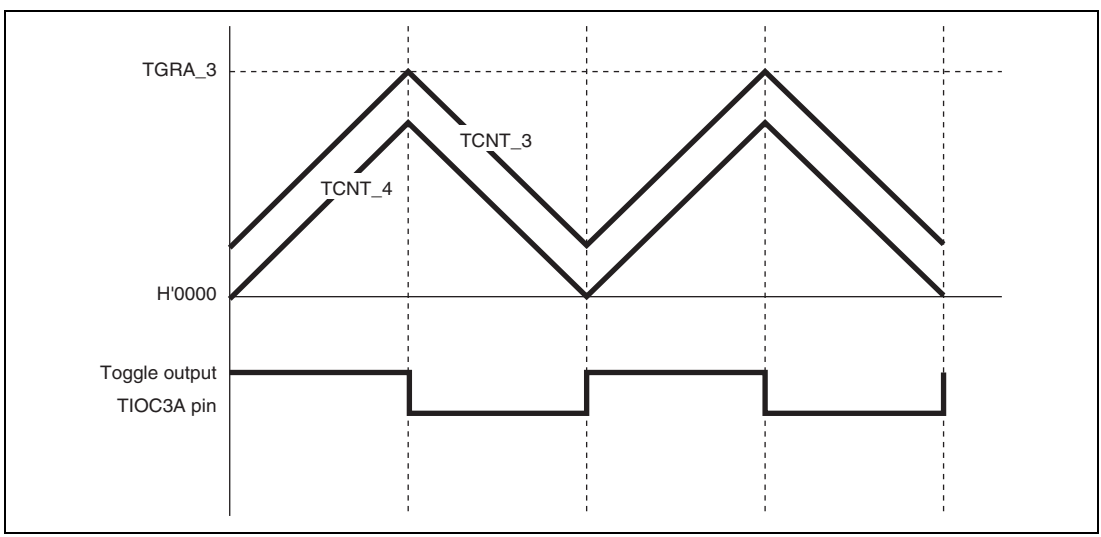


Figure 12.54 Example of Toggle Output Waveform Synchronized with PWM Output

(m) Counter Clearing by Another Channel

In complementary PWM mode, by setting a mode for synchronization with another channel by means of the timer synchronous 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 12.55 illustrates the operation.

Use of this function enables counter clearing and restarting to be performed by means of an external signal.

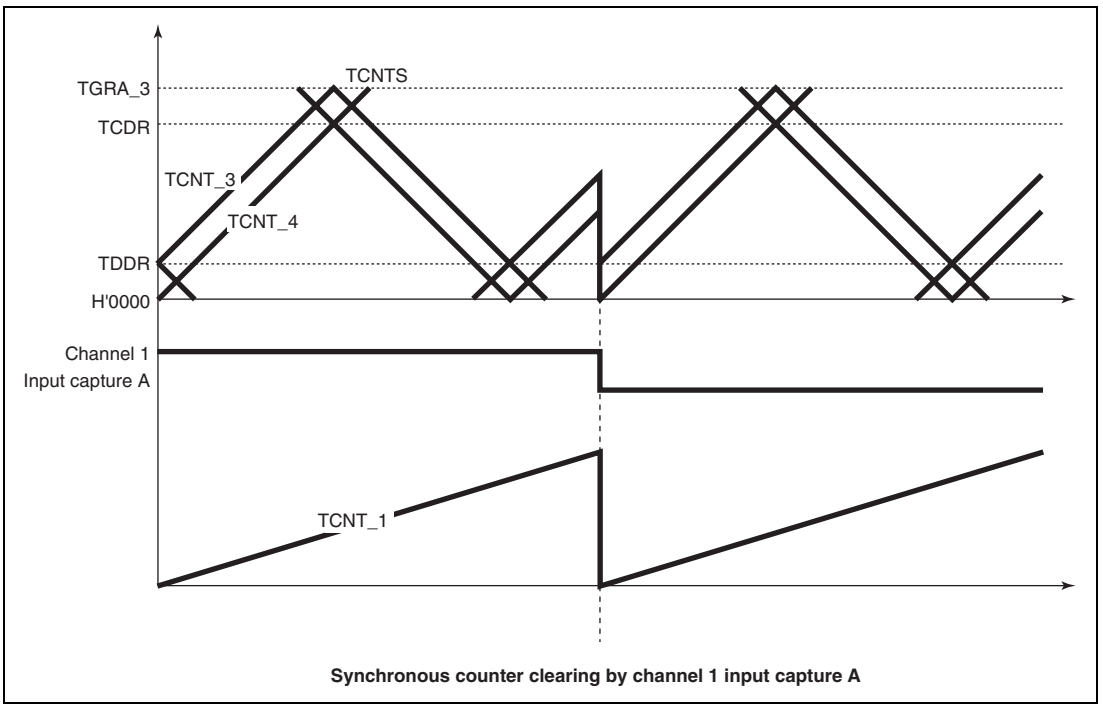


Figure 12.55 Counter Clearing Synchronized with Another Channel

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCR to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression is applicable only when synchronous clearing occurs in the Tb interval at the trough as indicated by (10) or (11) in figure 12.56. When synchronous clearing occurs outside that interval, the initial value specified by the OLS bits in TOCR is output. Even in the Tb interval at the trough, if synchronous clearing occurs in the initial value output period (indicated by (1) in figure 12.56) immediately after the counters start operation, initial value output is not suppressed.

When using the initial output suppression function, make sure to set compare registers TGRB_3, TGRA_4, and TGRB_4 to a value twice or more the setting of dead time data register TDDR. If synchronous clearing occurs with the compare registers set to a value less than twice the setting of TDDR, the PWM output dead time may be too short (or nonexistent) or illegal active-level PWM negative-phase output may occur during the initial output suppression interval. For details, see section 12.7.23, Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode.

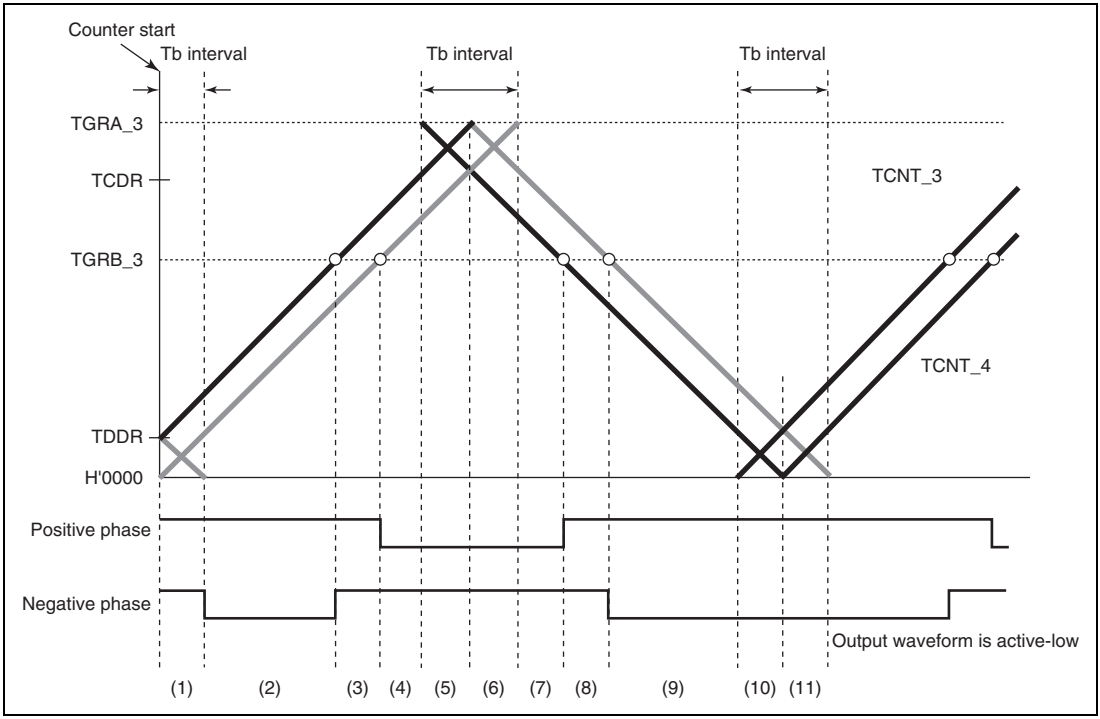


Figure 12.56 Timing for Synchronous Counter Clearing

- Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

  An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in figure 12.57.

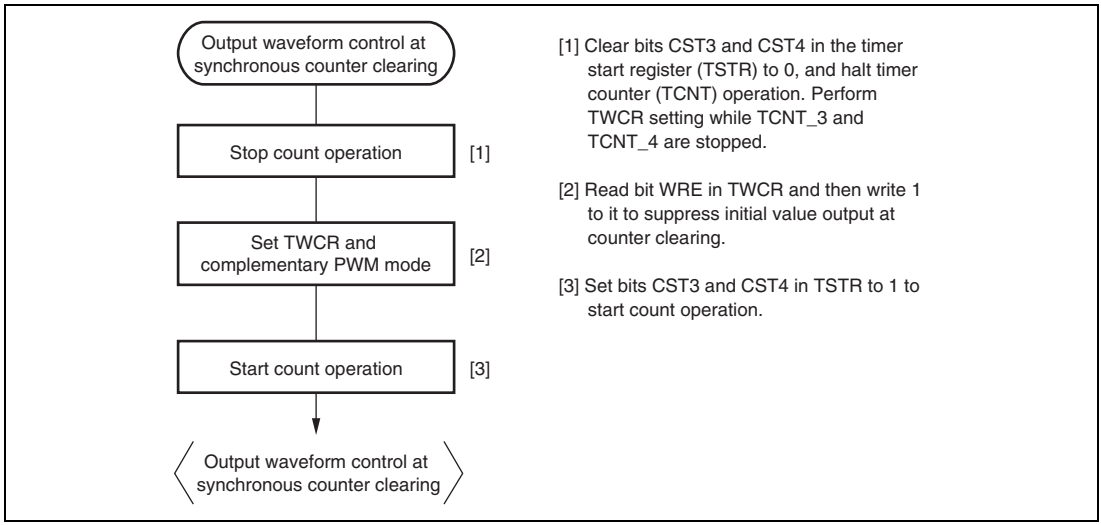


Figure 12.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

  Figures 12.58 to 12.61 show examples of output waveform control in which the MTU2 operates in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCR is set to 1. In the examples shown in figures 12.58 to 12.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figure 12.56, respectively.

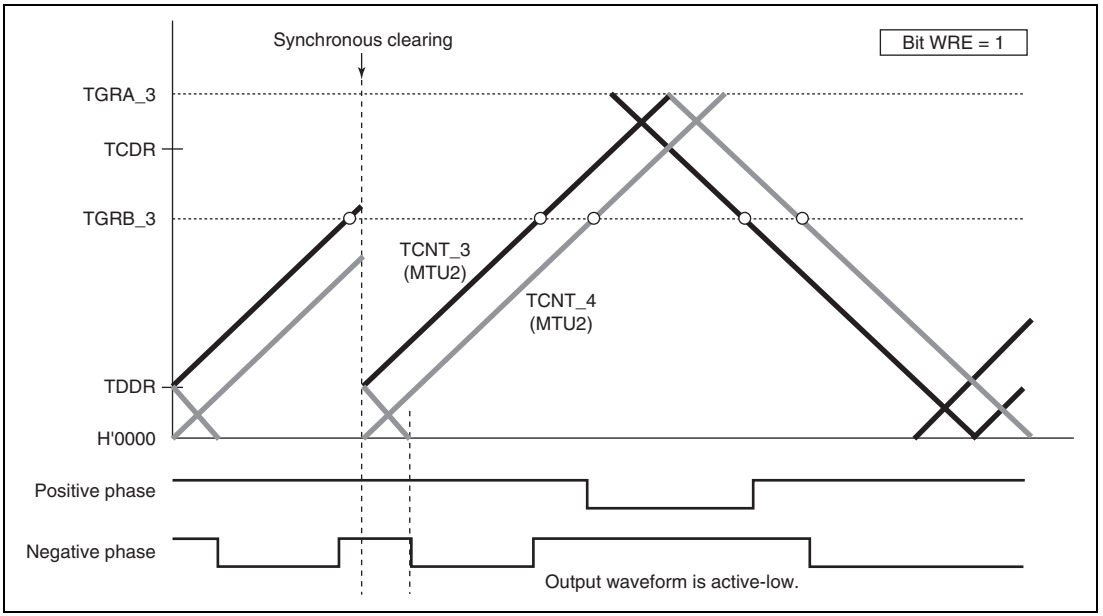


Figure 12.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 12.56; Bit WRE of TWCR in MTU2 is 1)

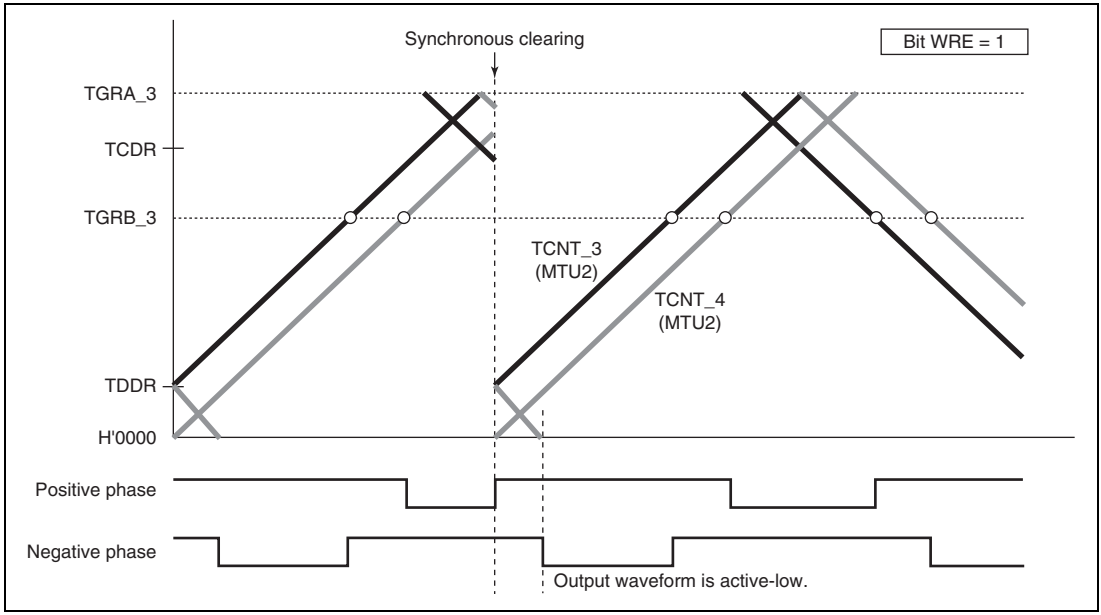


Figure 12.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 12.56; Bit WRE of TWCR in MTU2 is 1)

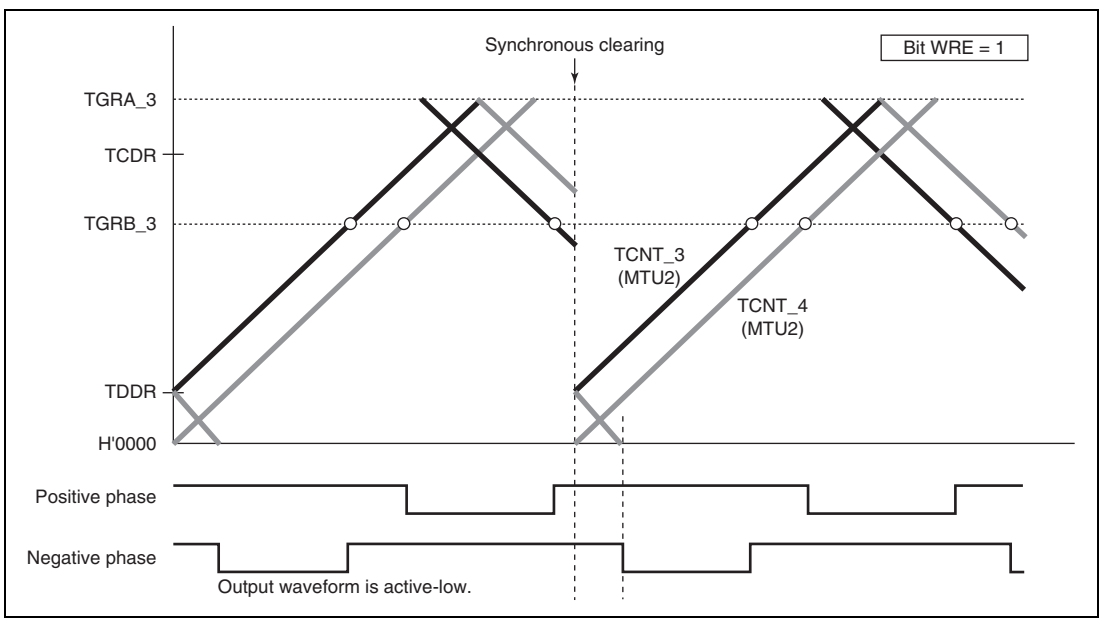


Figure 12.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 12.56; Bit WRE of TWCR is 1)

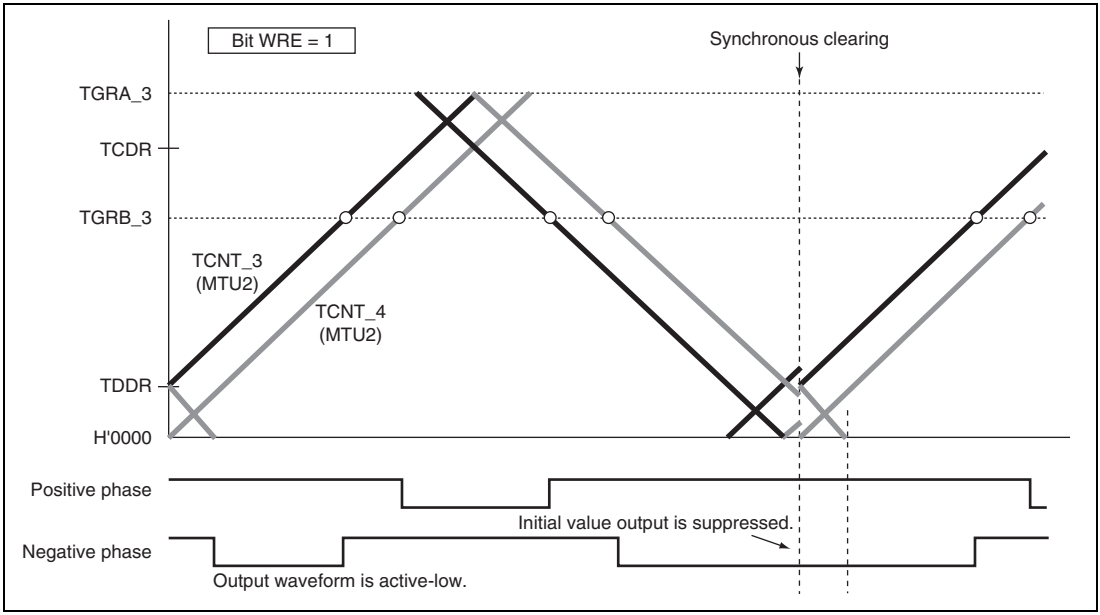


Figure 12.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 12.56; Bit WRE of TWCR is 1)

(o) Counter Clearing by TGRA_3 Compare Match

In complementary PWM mode, by setting the CCE bit in the timer waveform control register (TWCR), it is possible to have TCNT_3, TCNT_4, and TCNTS cleared by TGRA_3 compare match.

Figure 12.62 illustrates an operation example.

Notes: 1. Use this function only in complementary PWM mode 1 (transfer at crest)
2. Do not specify synchronous clearing by another channel (do not set the SYNC0 to SYNC4 bits in the timer synchronous register (TSYR) to 1.
3. Do not set the PWM duty value to H'0000.
4. Do not set the PSYE bit in timer output control register 1 (TOCR1) to 1.

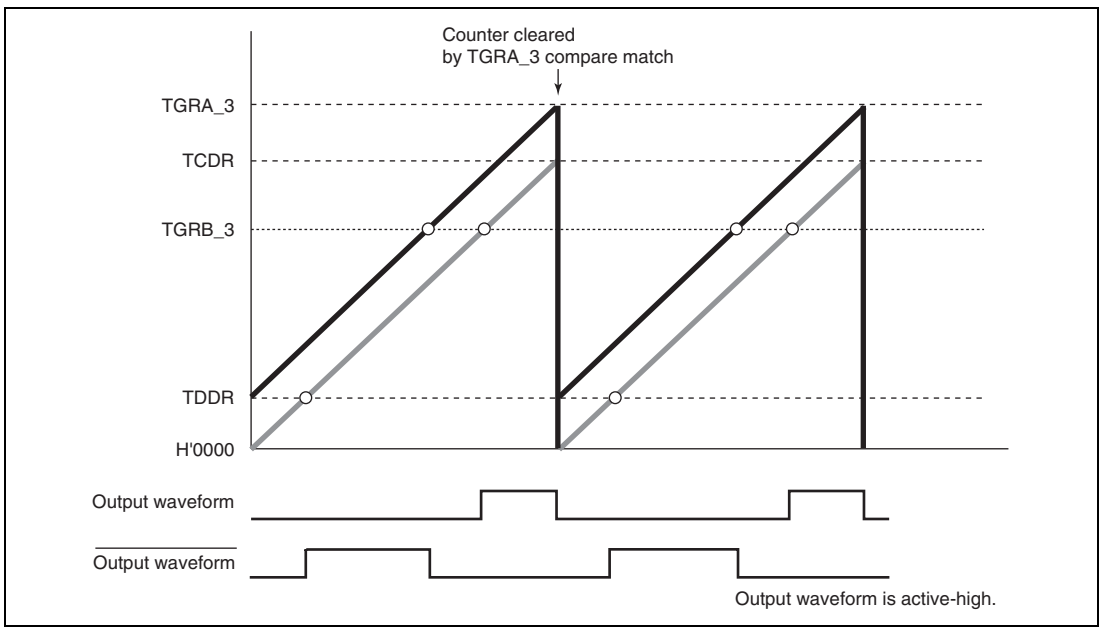


Figure 12.62 Example of Counter Clearing Operation by TGRA_3 Compare Match

(p) 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 12.63 to 12.66 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.

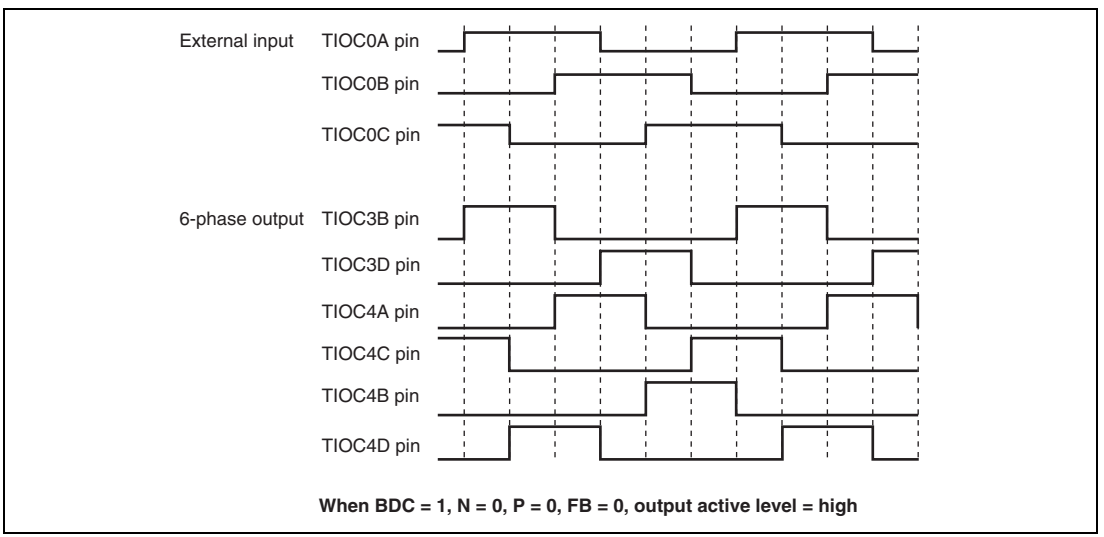


Figure 12.63 Example of Output Phase Switching by External Input (1)

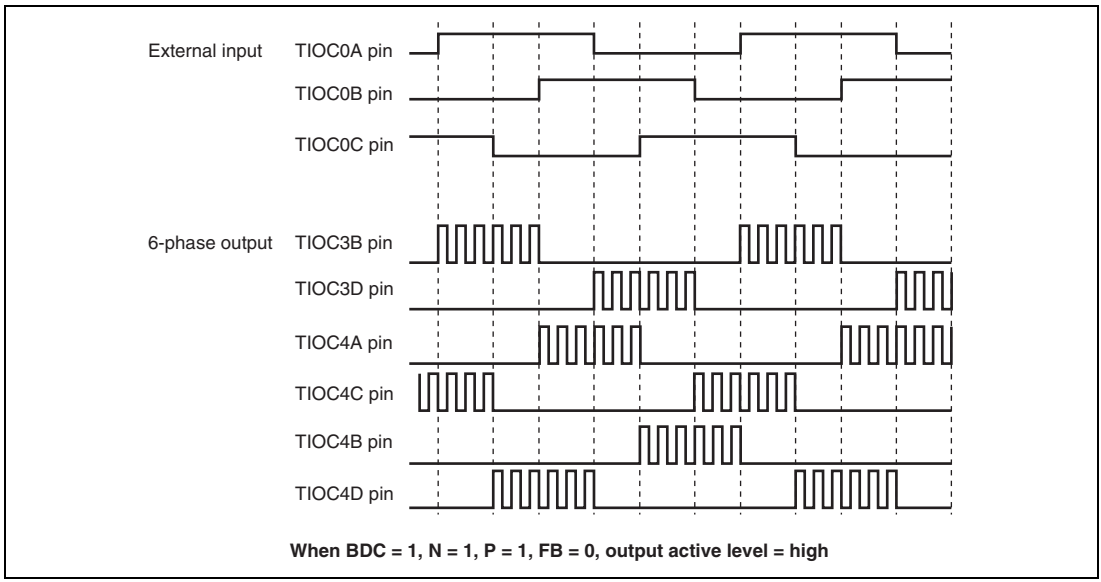


Figure 12.64 Example of Output Phase Switching by External Input (2)

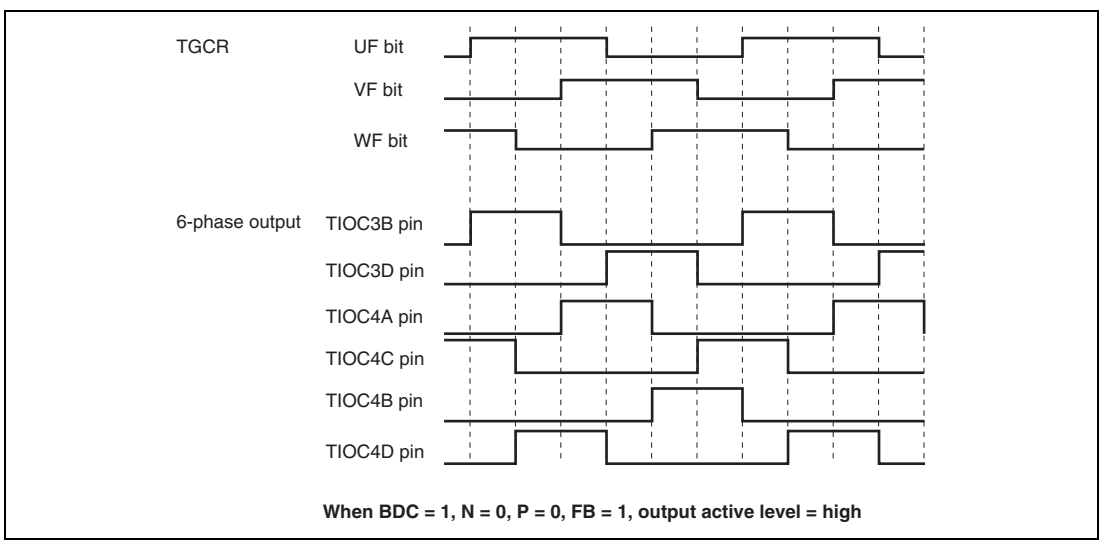


Figure 12.65 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1)

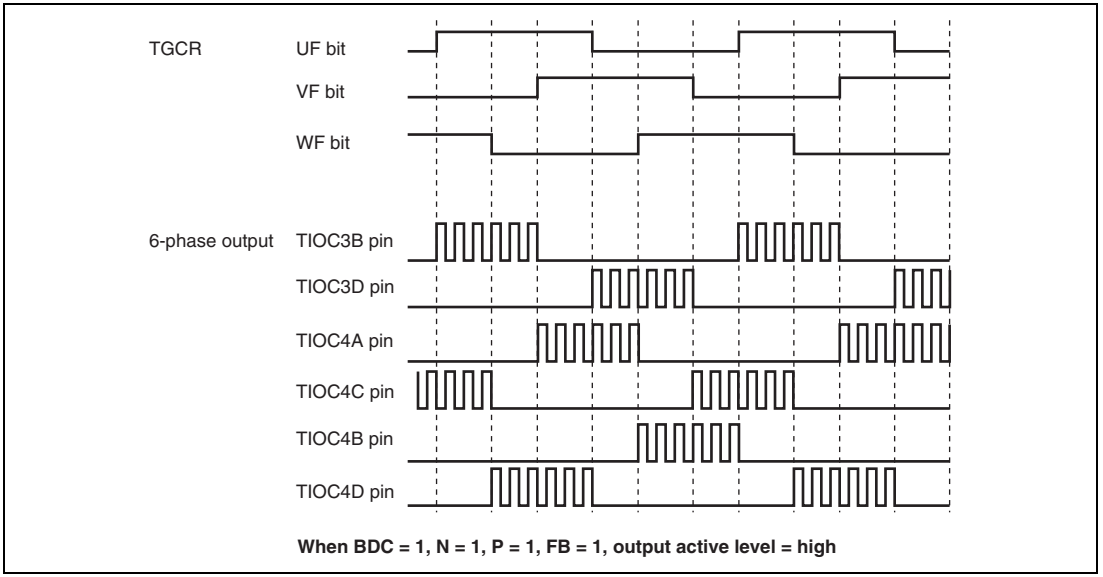


Figure 12.66 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2)

(q) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using a TGRA_3 compare-match, TCNT_4 underflow (trough), or compare-match on a channel other than channels 3 and 4.

When start requests using a TGRA_3 compare-match are specified, A/D conversion can be started at the crest of the TCNT_3 count.

A/D converter start requests can be set by setting the TTGE bit to 1 in the timer interrupt enable register (TIER). To issue an A/D converter start request at a TCNT_4 underflow (trough), set the TTGE2 bit in TIER_4 to 1.

(3) Interrupt Skipping in Complementary PWM Mode

Interrupts TGIA_3 (at the crest) and TCIV_4 (at the trough) in channels 3 and 4 can be skipped up to seven times by making settings in the timer interrupt skipping set register (TITCR).

Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the timer buffer transfer register (TBTER). For the linkage with buffer registers, refer to description (c), Buffer Transfer Control Linked with Interrupt Skipping, below.

A/D converter start requests generated by the A/D converter start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the timer A/D converter request control register (TADCR). For the linkage with the A/D converter start request delaying function, refer to section 12.4.9, A/D Converter Start Request Delaying Function.

The setting of the timer interrupt skipping setting register (TITCR) must be done while the TGIA_3 and TCIV_4 interrupt requests are disabled by the settings of TIER_3 and TIER_4 along with under the conditions in which TGFA_3 and TCFV_4 flag settings by compare match never occur. Before changing the skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Operation Setting Procedure

Figure 12.67 shows an example of the interrupt skipping operation setting procedure. Figure 12.68 shows the periods during which interrupt skipping count can be changed.

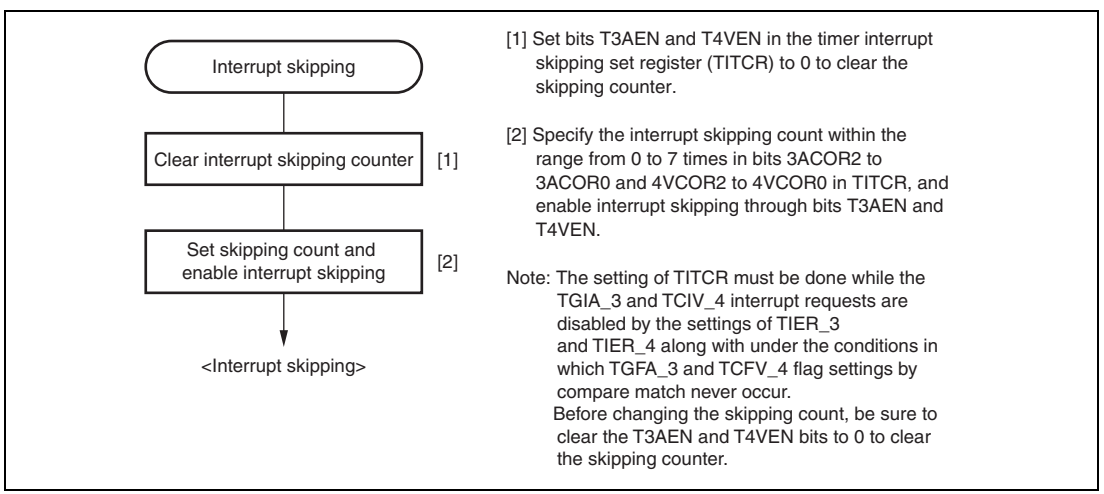


Figure 12.67 Example of Interrupt Skipping Operation Setting Procedure

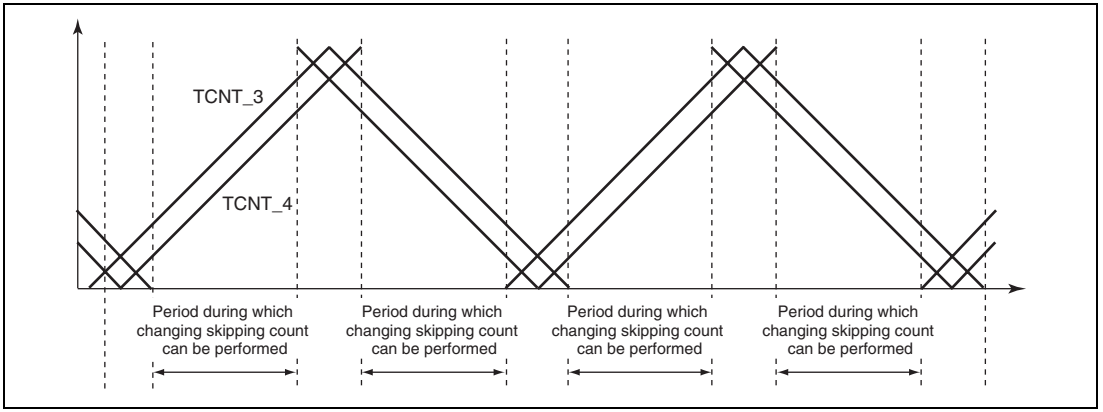


Figure 12.68 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Operation

Figure 12.69 shows an example of TGIA_3 interrupt skipping in which the interrupt skipping count is set to three by the 3ACOR bit and the T3AEN bit is set to 1 in the timer interrupt skipping set register (TITCR).

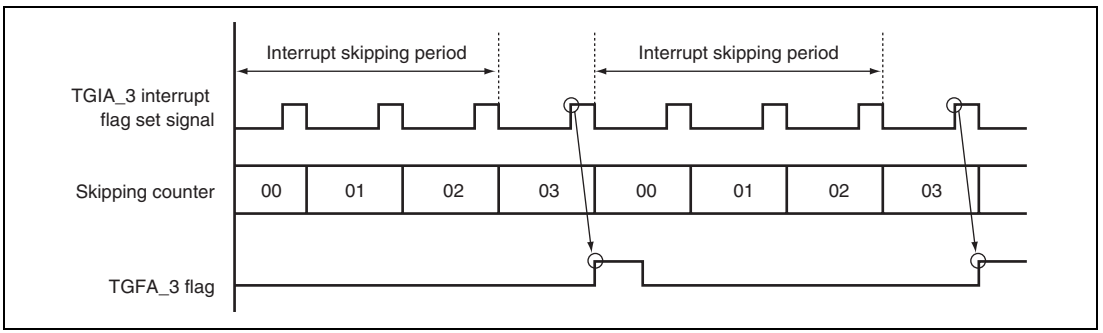


Figure 12.69 Example of Interrupt Skipping Operation

(c) Buffer Transfer Control Linked with Interrupt Skipping

In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE1 and BTE0 bits in the timer buffer transfer set register (TBTER).

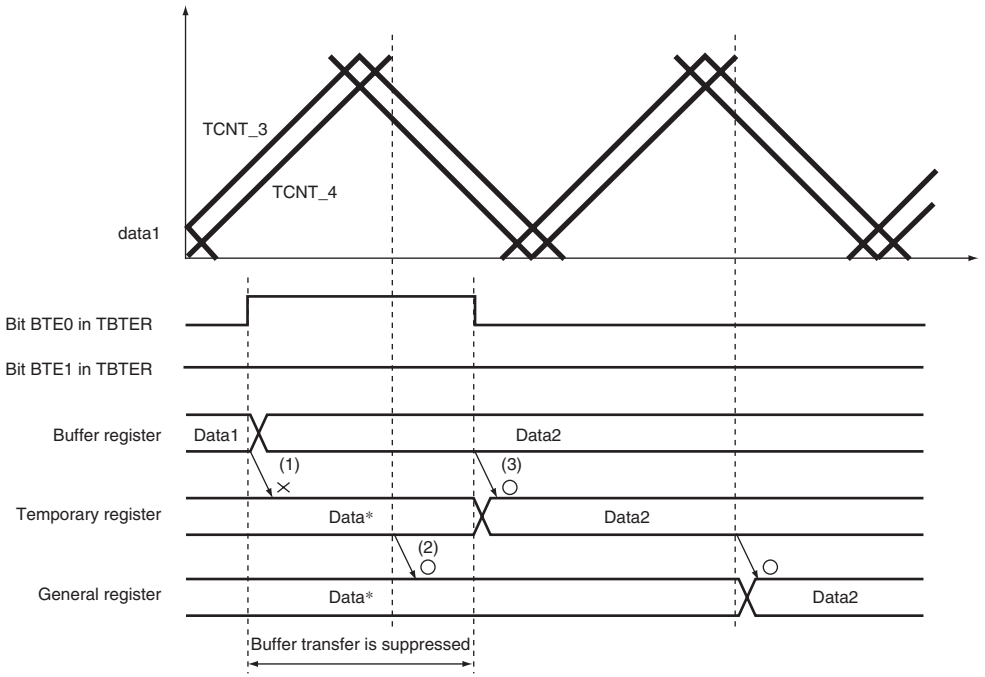
Figure 12.70 shows an example of operation when buffer transfer is suppressed (BTE1 = 0 and BTE0 = 1). While this setting is valid, data is not transferred from the buffer register to the temporary register.

Figure 12.71 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE1 = 1 and BET0 = 0). While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period.

Due to the buffer register rewrite timing after an interrupt, the timing of transfers from a buffer register to a temporary register differs from the timing of transfers from a temporary register to a general register.

Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR). Figure 12.72 shows the relationship between the T3AEN and T4VEN bit settings in TITCR and buffer transfer-enabled period.

Note: This function must always be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.



[Legend]

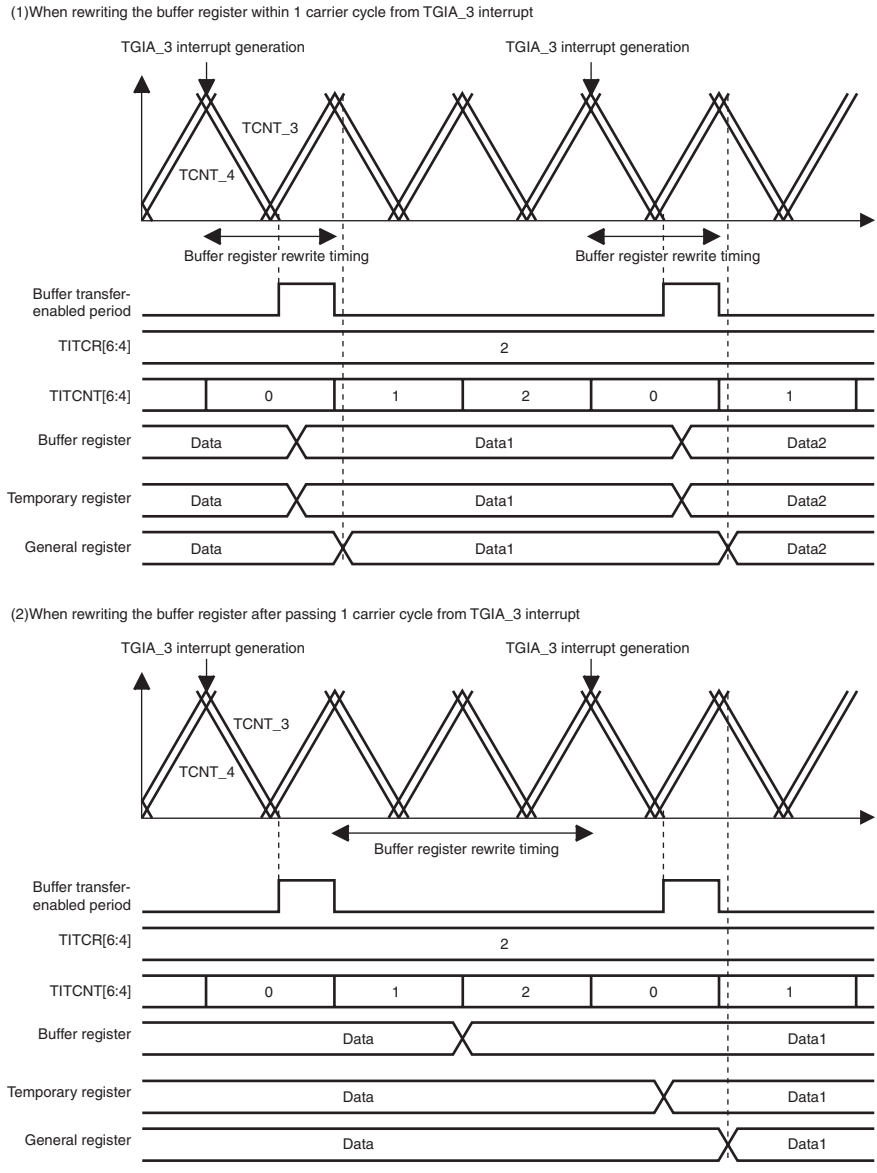
(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE1 and BTE0 in TBTER are set to 0 and 1, respectively).

(2) Data is transferred from the temporary register to the general register even in the buffer transfer-disabled period.

(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

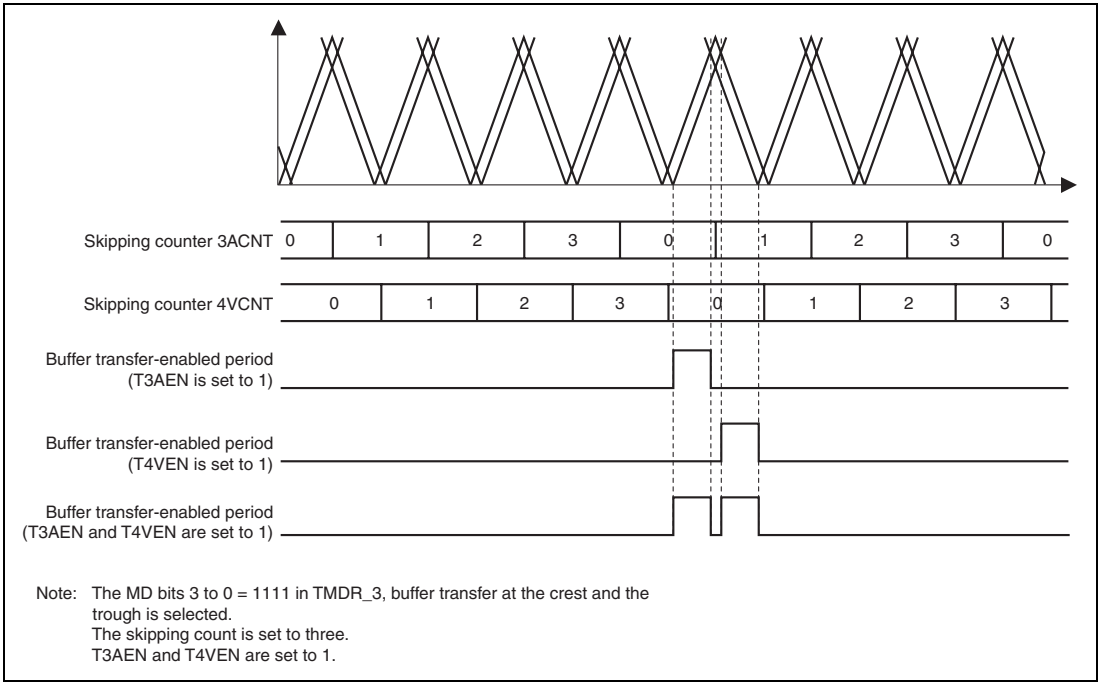
Note: * When buffer transfer at the crest is selected.

Figure 12.70 Example of Operation when Buffer Transfer is Suppressed (BTE1 = 0 and BTE0 = 1)



Note: The MD bits 3 to in TMDR3, buffer transfer at the crest is selected.
The skipping count is set to two.
T3AEN and T4VEN are set to 1 and 0.

Figure 12.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)



Note: The MD bits 3 to 0 = 1111 in TMDR_3, buffer transfer at the crest and the trough is selected.
The skipping count is set to three.
T3AEN and T4VEN are set to 1.

Figure 12.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period

(4) Complementary PWM Mode Output Protection Function

Complementary PWM mode output has the following protection function.

(a) 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 the RWE bit in the timer read/write enable register (TRWER). The applicable registers are some (21 in total) of the registers in channels 3 and 4 shown in the following:

- 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 miswriting due to CPU runaway to be prevented by disabling CPU access to the mode registers, control registers, and counters. When the applicable registers are read in the access-disabled state, undefined values are returned. Writing to these registers is ignored.

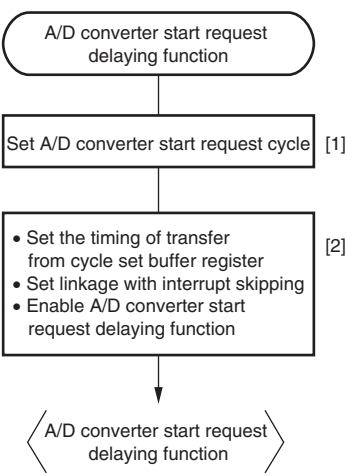
12.4.9 A/D Converter Start Request Delaying Function

A/D converter start requests can be issued in channel 4 by making settings in the timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCORA_4 and TADCORB_4), and timer A/D converter start request cycle set buffer registers (TADCOBRA_4 and TADCOBRB_4).

The A/D converter start request delaying function compares TCNT_4 with TADCORA_4 or TADCORB_4, and when their values match, the function issues a respective A/D converter start request (TRG4AN or TRG4BN).

A/D converter start requests (TRG4AN and TRG4BN) can be skipped in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in TADCR.

- Example of Procedure for Specifying A/D Converter Start Request Delaying Function
  Figure 12.73 shows an example of procedure for specifying the A/D converter start request delaying function.



[1] Set the cycle in the timer A/D converter start request cycle buffer register (TADCOBRA_4 or TADCOBRB_4) and timer A/D converter start request cycle register (TADCORA_4 or TADCORB_4). (The same initial value must be specified in the cycle buffer register and cycle register.)

[2] Use bits BF1 and BF2 in the timer A/D converter start request control register (TADCR) to specify the timing of transfer from the timer A/D converter start request cycle buffer register to A/D converter start request cycle register.
- Specify whether to link with interrupt skipping through bits ITA3AE, ITA4VE, ITB3AE, and ITB4VE.
- Use bits TU4AE, DT4AE, UT4BE, and DT4BE to enable A/D conversion start requests (TRG4AN or TRG4BN).

Notes: 1. Perform TADCR setting while TCNT_4 is stopped.
2. Do not set BF1 to 1 when complementary PWM mode is not selected.
3. Do not set ITA3AE, ITA4VE, ITB3AE, ITB4VE, DT4AE, or DT4BE to 1 when complementary PWM mode is not selected.

Figure 12.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

- Basic Operation Example of A/D Converter Start Request Delaying Function

  Figure 12.74 shows a basic example of A/D converter request signal (TRG4AN) operation when the trough of TCNT_4 is specified for the buffer transfer timing and an A/D converter start request signal is output during TCNT_4 down-counting.

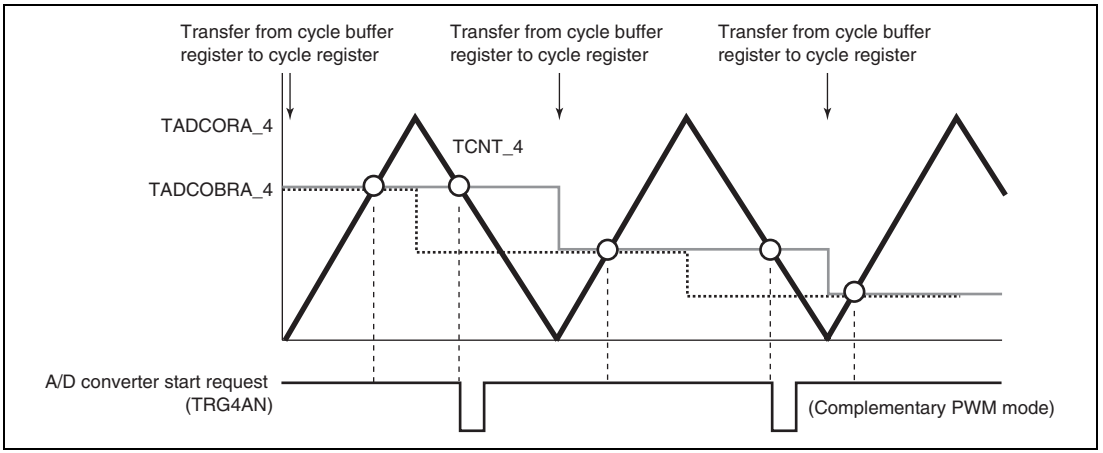


Figure 12.74 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

- Buffer Transfer

  The data in the timer A/D converter start request cycle set registers (TADCORA_4 and TADCORB_4) is updated by writing data to the timer A/D converter start request cycle set buffer registers (TADCOBRA_4 and TADCOBRB_4). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the BF1 and BF0 bits in the timer A/D converter start request control register (TADCR_4).

- A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

  A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR).

  Figure 12.75 shows an example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during TCNT_4 up-counting and down-counting and A/D converter start requests are linked with interrupt skipping.

  Figure 12.76 shows another example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during TCNT_4 up-counting and A/D converter start requests are linked with interrupt skipping.

Note: This function must be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), make sure that A/D converter start requests are not linked with interrupt skipping (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).

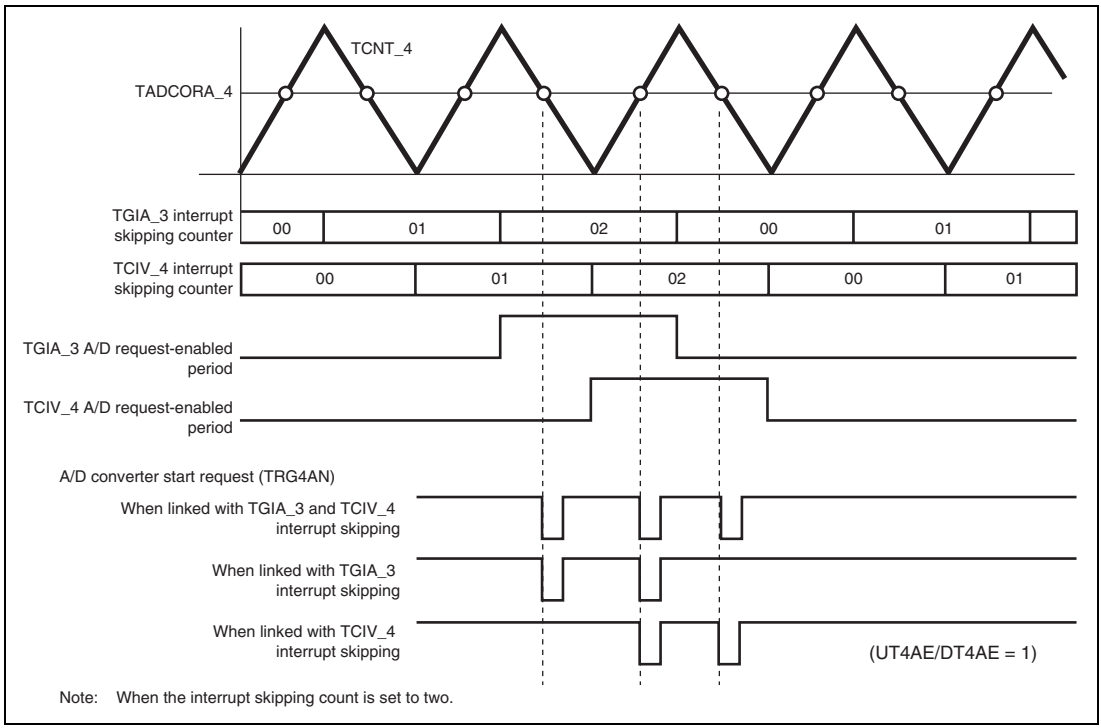


Figure 12.75 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

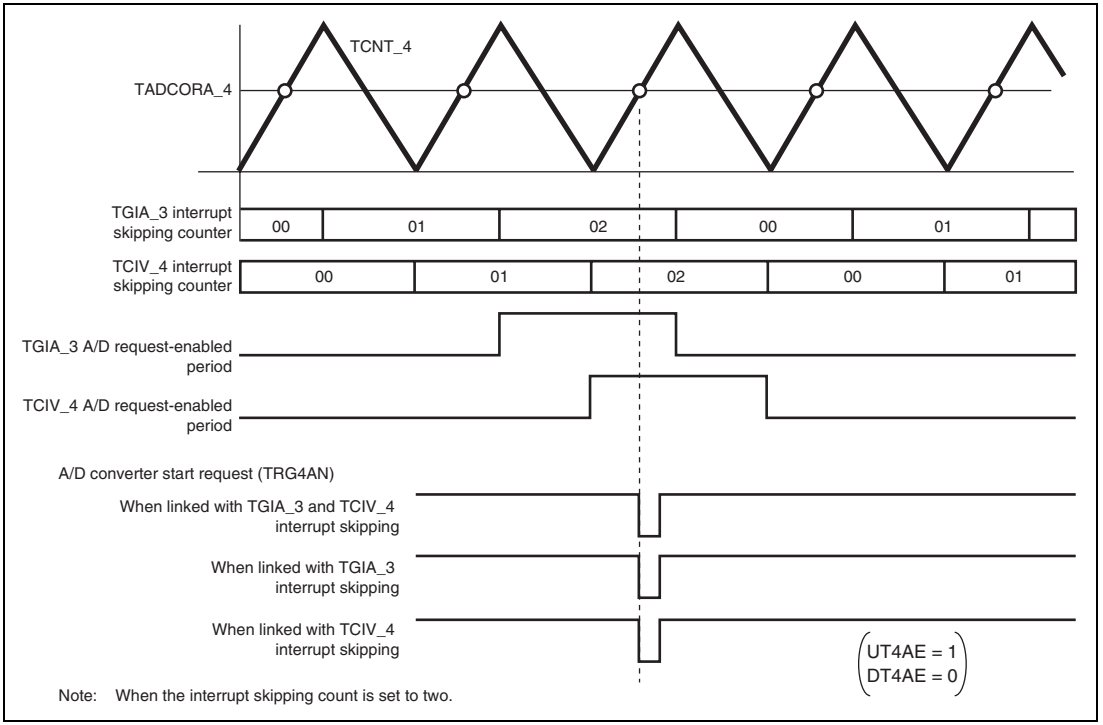


Figure 12.76 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

12.4.10 TCNT Capture at Crest and/or Trough in Complementary PWM Operation

The TCNT value is captured in TGR at either the crest or trough or at both the crest and trough during complementary PWM operation. The timing for capturing in TGR can be selected by TIOR.

Figure 12.77 shows an example in which TCNT is used as a free-running counter without being cleared, and the TCNT value is captured in TGR at the specified timing (either crest or trough, or both crest and trough).

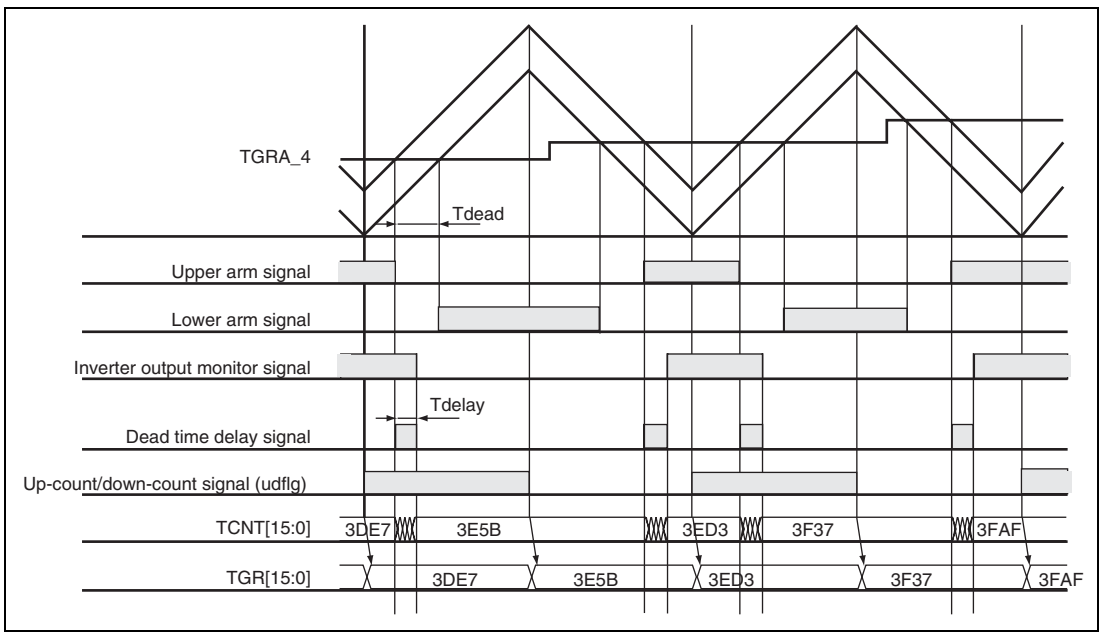


Figure 12.77 TCNT Capturing at Crest and/or Trough in Complementary PWM Operation

12.5 Interrupt Sources

12.5.1 Interrupt Sources and Priorities

There are three kinds of MTU2 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 priorities can be changed by the interrupt controller, however the priority order within a channel is fixed. For details, see section 7, Interrupt Controller (INTC).

Table 12.55 lists the MTU2 interrupt sources.

Table 12.55 MTU2 Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DMAC Activation	Priority
0	TGIA_0	TGRA_0 input capture/compare match	TGFA_0	Possible	High
	TGIB_0	TGRB_0 input capture/compare match	TGFB_0	Not possible	
	TGIC_0	TGRC_0 input capture/compare match	TGFC_0	Not possible	
	TGID_0	TGRD_0 input capture/compare match	TGFD_0	Not possible	
	TCIV_0	TCNT_0 overflow	TCFV_0	Not possible	
	TGIE_0	TGRE_0 compare match	TGFE_0	Not possible	
	TGIF_0	TGRF_0 compare match	TGFF_0	Not possible	
1	TGIA_1	TGRA_1 input capture/compare match	TGFA_1	Possible	
	TGIB_1	TGRB_1 input capture/compare match	TGFB_1	Not possible	
	TCIV_1	TCNT_1 overflow	TCFV_1	Not possible	
	TCIU_1	TCNT_1 underflow	TCFU_1	Not possible	
2	TGIA_2	TGRA_2 input capture/compare match	TGFA_2	Possible	
	TGIB_2	TGRB_2 input capture/compare match	TGFB_2	Not possible	
	TCIV_2	TCNT_2 overflow	TCFV_2	Not possible	
	TCIU_2	TCNT_2 underflow	TCFU_2	Not possible	Low

Channel	Name	Interrupt Source	Interrupt Flag	DMAC Activation	Priority
3	TGIA_3	TGRA_3 input capture/compare match	TGFA_3	Possible	High
	TGIB_3	TGRB_3 input capture/compare match	TGFB_3	Not possible	
	TGIC_3	TGRC_3 input capture/compare match	TGFC_3	Not possible	
	TGID_3	TGRD_3 input capture/compare match	TGFD_3	Not possible	
	TCIV_3	TCNT_3 overflow	TCFV_3	Not possible	
4	TGIA_4	TGRA_4 input capture/compare match	TGFA_4	Possible	
	TGIB_4	TGRB_4 input capture/compare match	TGFB_4	Not possible	
	TGIC_4	TGRC_4 input capture/compare match	TGFC_4	Not possible	
	TGID_4	TGRD_4 input capture/compare match	TGFD_4	Not possible	
	TCIV_4	TCNT_4 overflow/underflow	TCFV_4	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

(1) Input Capture/Compare Match Interrupt

An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The MTU2 has eighteen input capture/compare match interrupts, six for channel 0, four each for channels 3 and 4, and two each for channels 1 and 2. The TGFE_0 and TGFF_0 flags in channel 0 are not set by the occurrence of an input capture.

(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 MTU2 has five overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The MTU2 has two underflow interrupts, one each for channels 1 and 2.

12.5.2 DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt in each channel. For details, see section 11, Direct Memory Access Controller (DMAC).

In the MTU2, a total of five TGRA input capture/compare match interrupts can be used as DMAC activation sources, one each for channels 0 to 4.

12.5.3 A/D Converter Activation

The A/D converter can be activated by one of the following three methods in the MTU2. Table 12.56 shows the relationship between interrupt sources and A/D converter start request signals.

(1) A/D Converter Activation by TGRA Input Capture/Compare Match or at TCNT_4 Trough in Complementary PWM Mode

The A/D converter can be activated by the occurrence of a TGRA input capture/compare match in each channel. In addition, if complementary PWM operation is performed while the TTGE2 bit in TIER_4 is set to 1, the A/D converter can be activated at the trough of TCNT_4 count (TCNT_4 = H'0000).

A/D converter start request signal TRGAN is issued to the A/D converter under either one of the following conditions.

- When the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel while the TTGE bit in TIER is set to 1
- When the TCNT_4 count reaches the trough (TCNT_4 = H'0000) during complementary PWM operation while the TTGE2 bit in TIER_4 is set to 1

When either condition is satisfied, if A/D converter start signal TRGAN from the MTU2 is selected as the trigger in the A/D converter, A/D conversion will start.

(2) A/D Converter Activation by Compare Match between TCNT_0 and TGRE_0

The A/D converter can be activated by generating A/D converter start request signal TRG0N when a compare match occurs between TCNT_0 and TGRE_0 in channel 0.

When the TGFE flag in TSR2_0 is set to 1 by the occurrence of a compare match between TCNT_0 and TGRE_0 in channel 0 while the TTGE2 bit in TIER2_0 is set to 1, A/D converter start request TGR0N is issued to the A/D converter. If A/D converter start signal TGR0N from the MTU2 is selected as the trigger in the A/D converter, A/D conversion will start.

(3) A/D Converter Activation by A/D Converter Start Request Delaying Function

The A/D converter can be activated by generating A/D converter start request signal TRG4AN or TRG4BN when the TCNT_4 count matches the TADCORA or TADCORB value if the UT4AE, DT4AE, UT4BE, and DT4BE bit in the A/D converter start request control register (TADCR) is set to 1. For details, refer to section 12.4.9, A/D Converter Start Request Delaying Function.

A/D conversion will start if A/D converter start signal TRG4AN from the MTU2 is selected as the trigger in the A/D converter when TRG4AN is generated or if TRG4BN from the MTU2 is selected as the trigger in the A/D converter when TRG4BN is generated.

Table 12.56 Interrupt Sources and A/D Converter Start Request Signals

Target Registers	Interrupt Source	A/D Converter Start Request Signal
TGRA_0 and TCNT_0	Input capture/compare match	TRGAN
TGRA_1 and TCNT_1		
TGRA_2 and TCNT_2		
TGRA_3 and TCNT_3		
TGRA_4 and TCNT_4		
TCNT_4	TCNT_4 Trough in complementary PWM mode	
TGRE_0 and TCNT_0	Compare match	TRG0N
TADCORA and TCNT_4		TRG4AN
TADCORB and TCNT_4		TRG4BN

12.6 Operation Timing

12.6.1 Input/Output Timing

(1) TCNT Count Timing

Figure 12.78 shows TCNT count timing in internal clock operation, and figure 12.79 shows TCNT count timing in external clock operation (normal mode), and figure 12.80 shows TCNT count timing in external clock operation (phase counting mode).

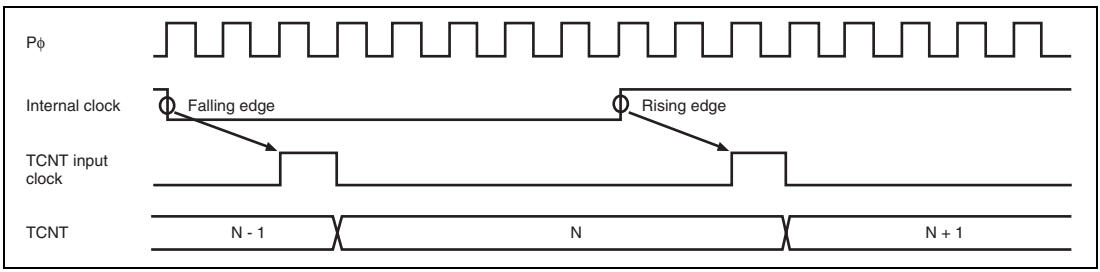


Figure 12.78 Count Timing in Internal Clock Operation

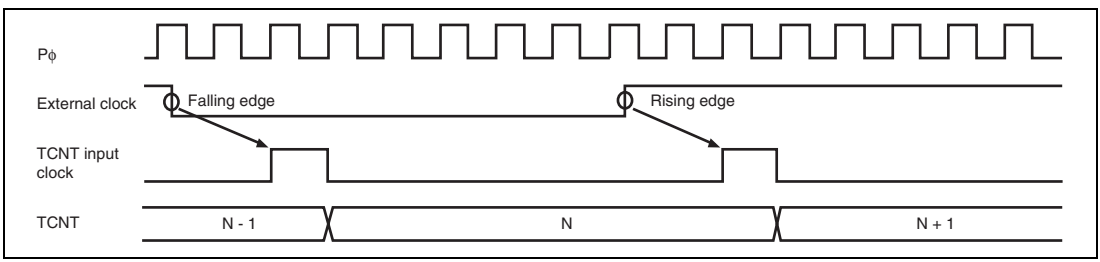


Figure 12.79 Count Timing in External Clock Operation

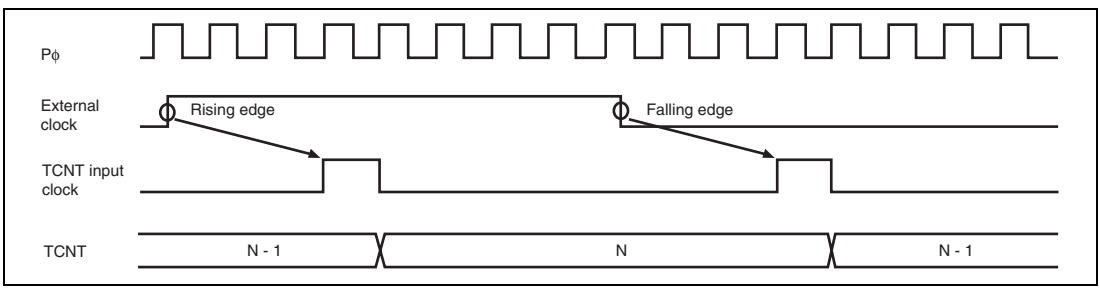


Figure 12.80 Count Timing in External Clock Operation (Phase Counting Mode)

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin (TIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 12.81 shows output compare output timing (normal mode and PWM mode) and figure 12.82 shows output compare output timing (complementary PWM mode and reset synchronous PWM mode).

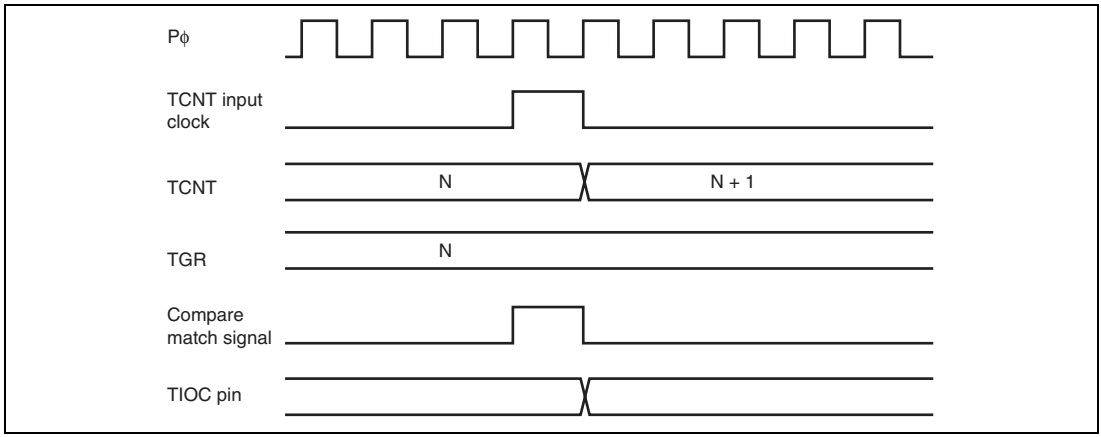


Figure 12.81 Output Compare Output Timing (Normal Mode/PWM Mode)

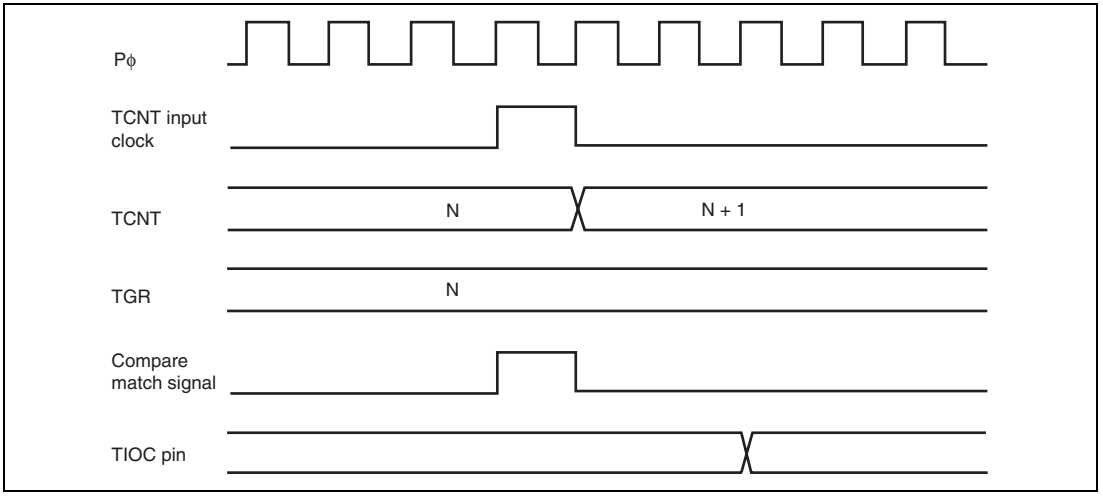


Figure 12.82 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)

(3) Input Capture Signal Timing

Figure 12.83 shows input capture signal timing.

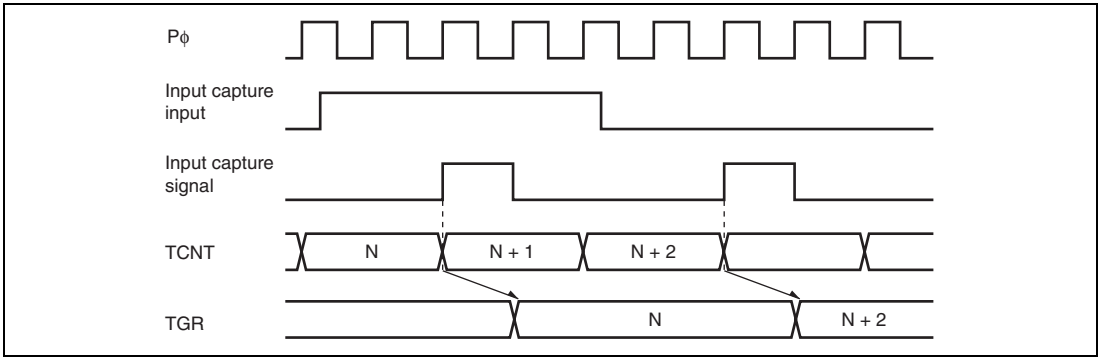


Figure 12.83 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 12.84 shows the timing when counter clearing on compare match is specified, and figure 12.85 shows the timing when counter clearing on input capture is specified.

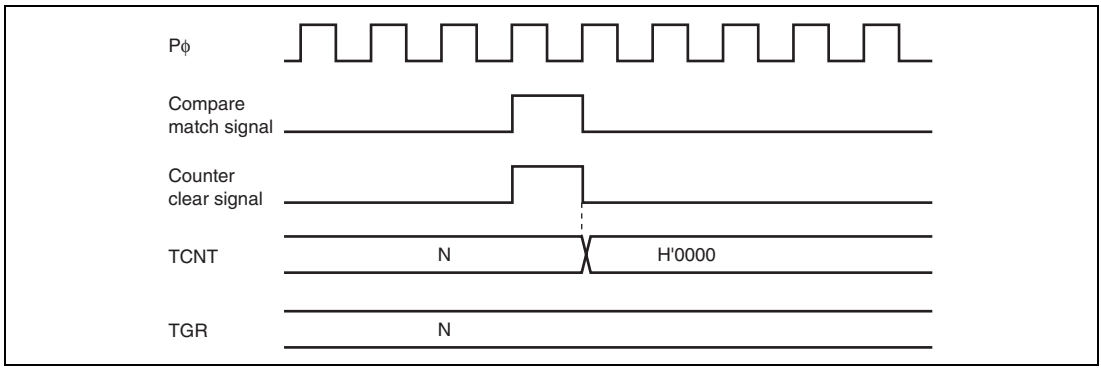


Figure 12.84 Counter Clear Timing (Compare Match)

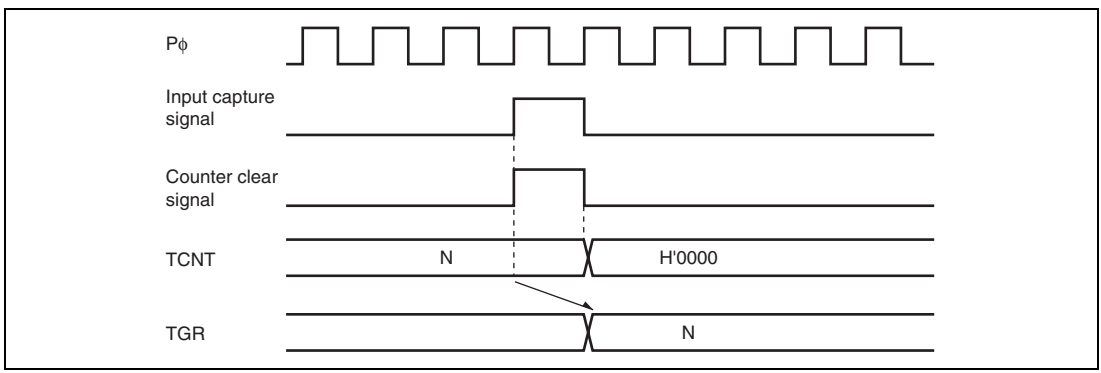


Figure 12.85 Counter Clear Timing (Input Capture)

(5) Buffer Operation Timing

Figures 12.86 to 12.88 show the timing in buffer operation.

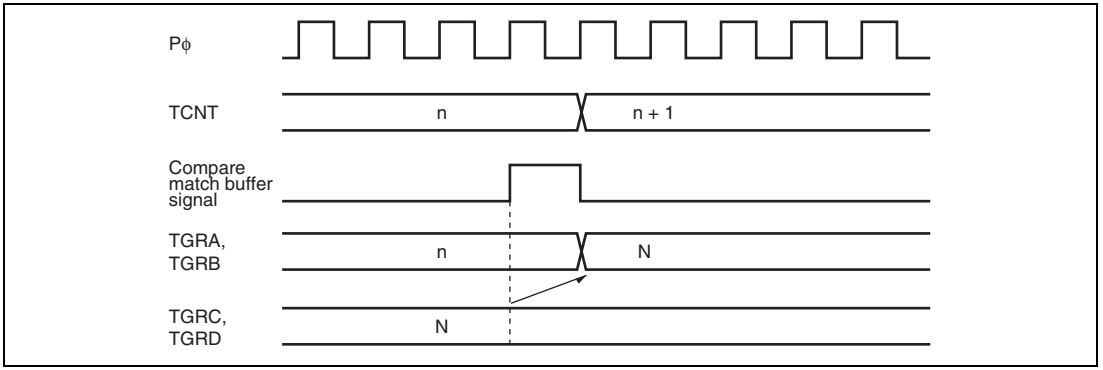


Figure 12.86 Buffer Operation Timing (Compare Match)

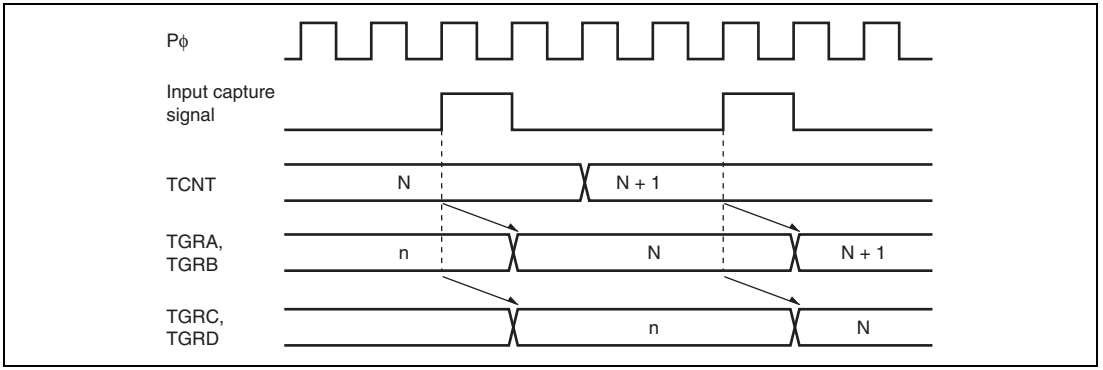


Figure 12.87 Buffer Operation Timing (Input Capture)

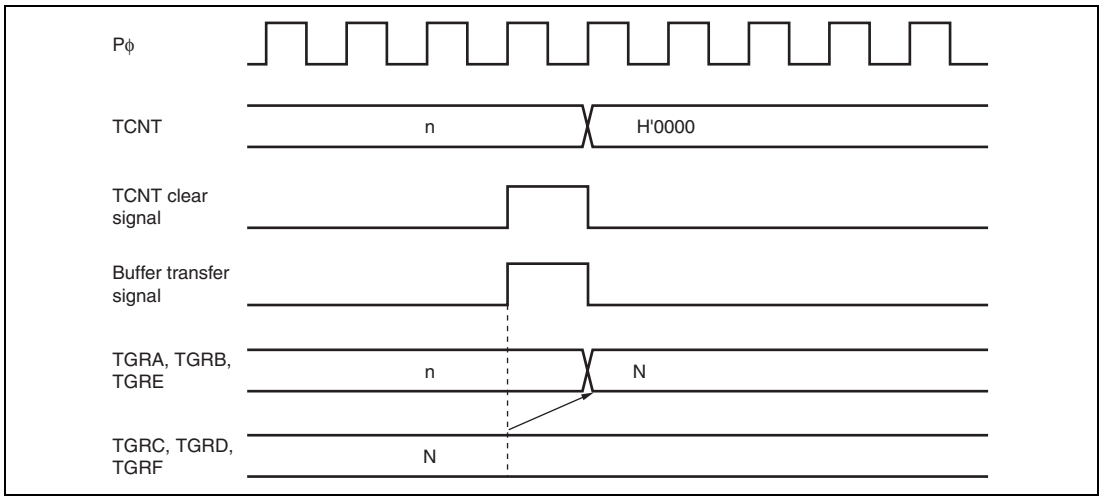


Figure 12.88 Buffer Transfer Timing (when TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figures 12.89 to 12.91 show the buffer transfer timing in complementary PWM mode.

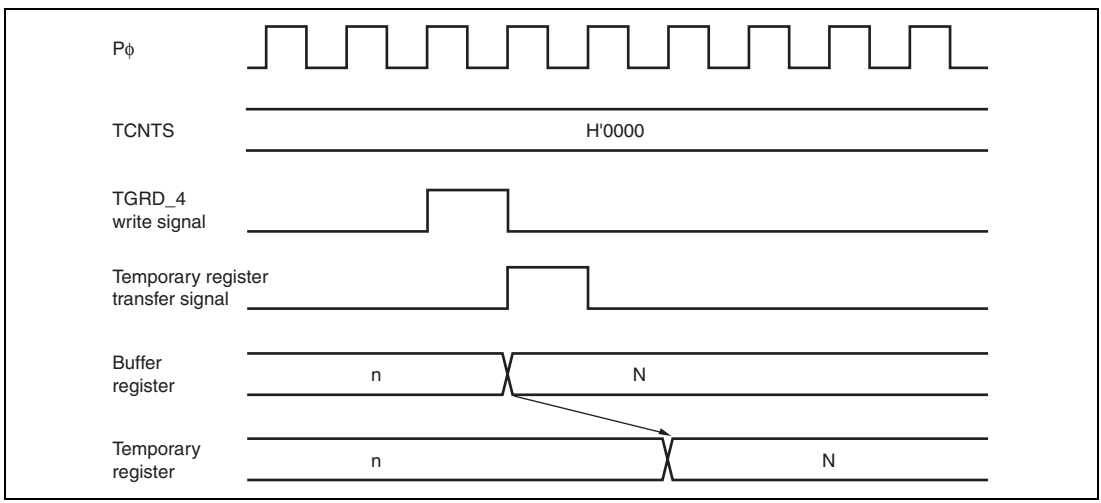


Figure 12.89 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

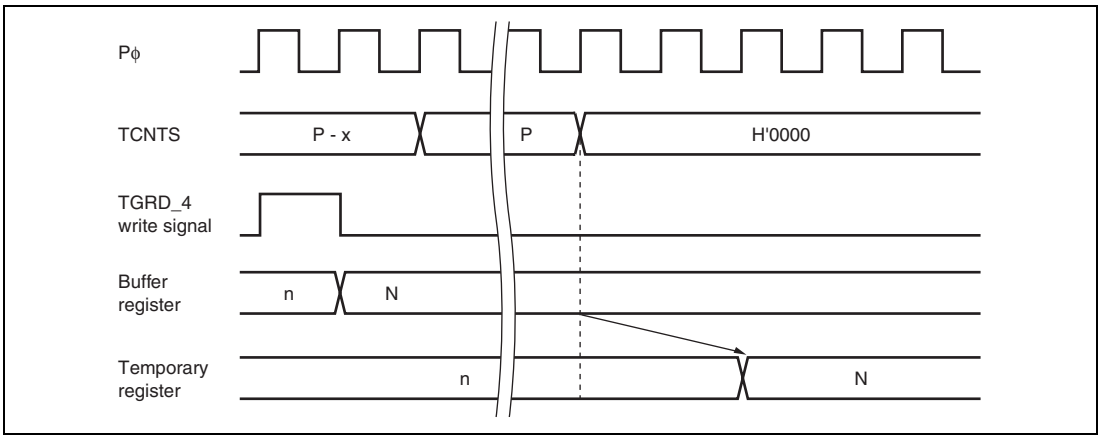


Figure 12.90 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

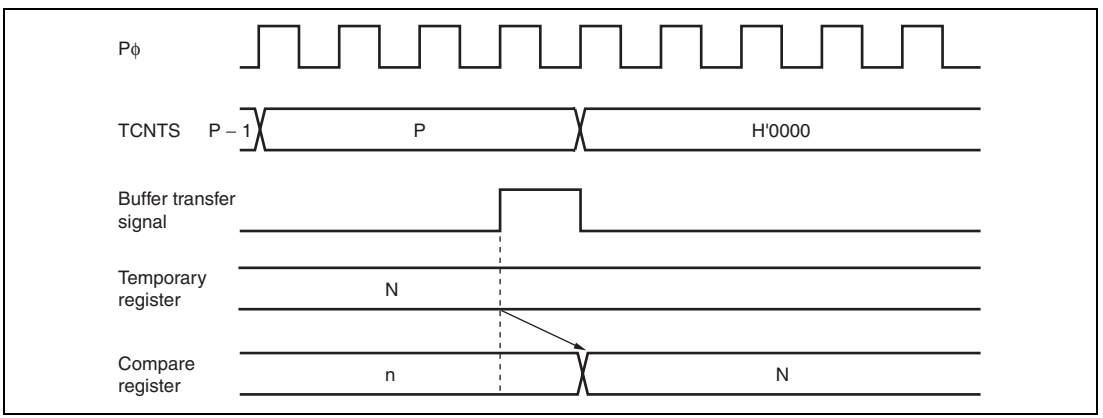


Figure 12.91 Transfer Timing from Temporary Register to Compare Register

12.6.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figure 12.92 shows the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

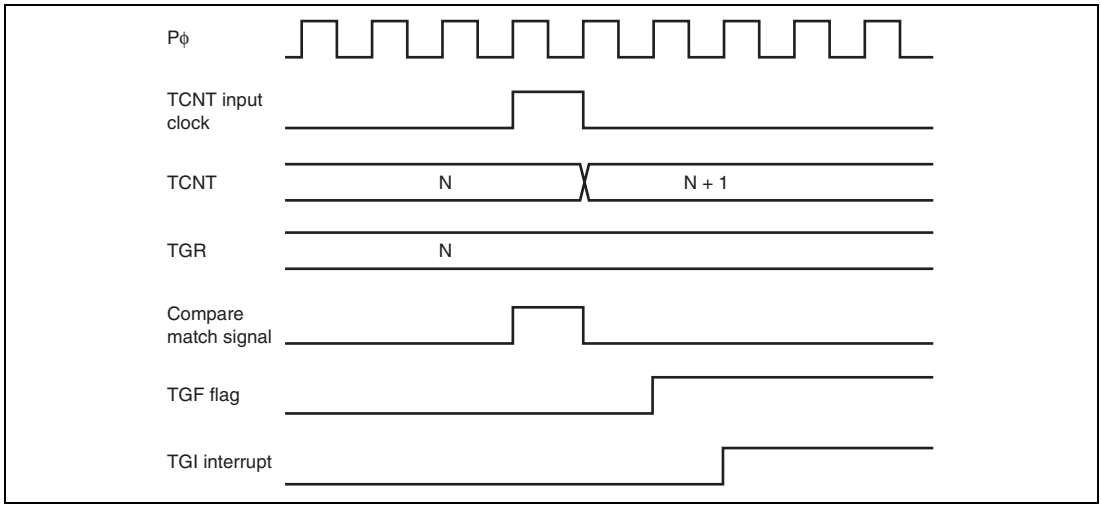


Figure 12.92 TGI Interrupt Timing (Compare Match)

(2) TGF Flag Setting Timing in Case of Input Capture

Figure 12.93 shows the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

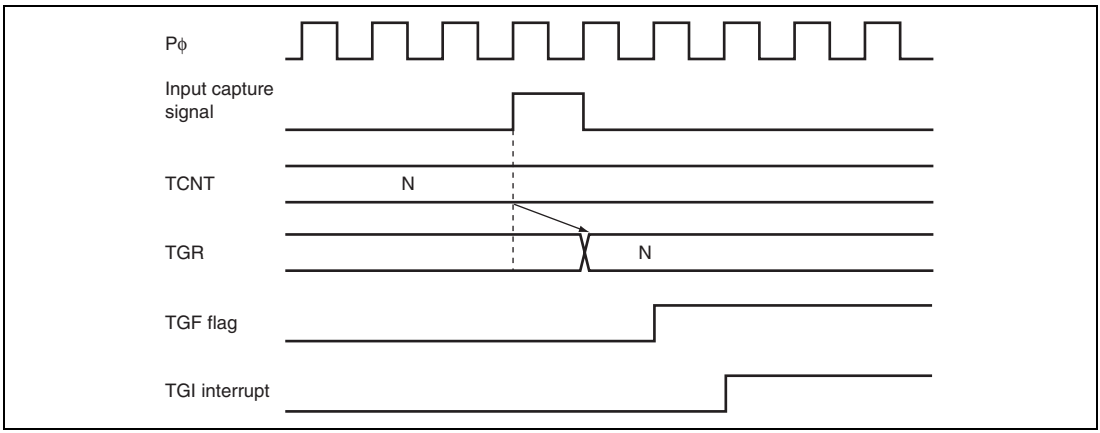


Figure 12.93 TGI Interrupt Timing (Input Capture)

(3) TCFV Flag/TCFU Flag Setting Timing

Figure 12.94 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 12.95 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

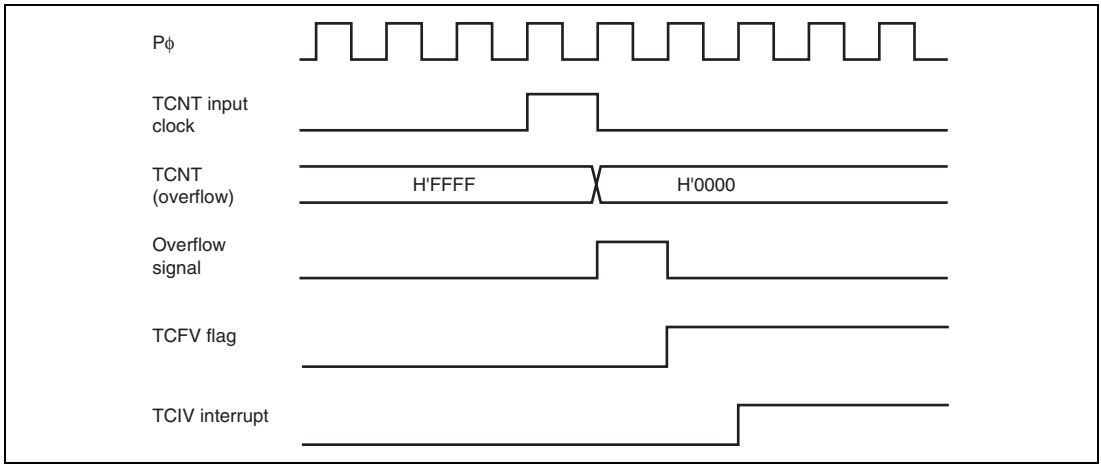


Figure 12.94 TCIV Interrupt Setting Timing

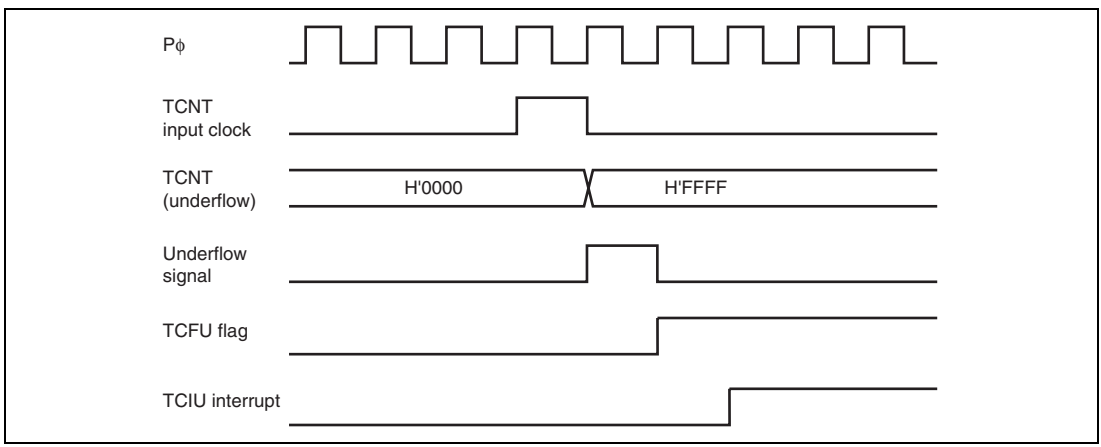


Figure 12.95 TCIU Interrupt Setting Timing

(4) Status Flag Clearing Timing

After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DMAC is activated, the flag is cleared automatically. Figure 12.96 shows the timing for status flag clearing by the CPU, and figure 12.97 shows the timing for status flag clearing by the DMAC.

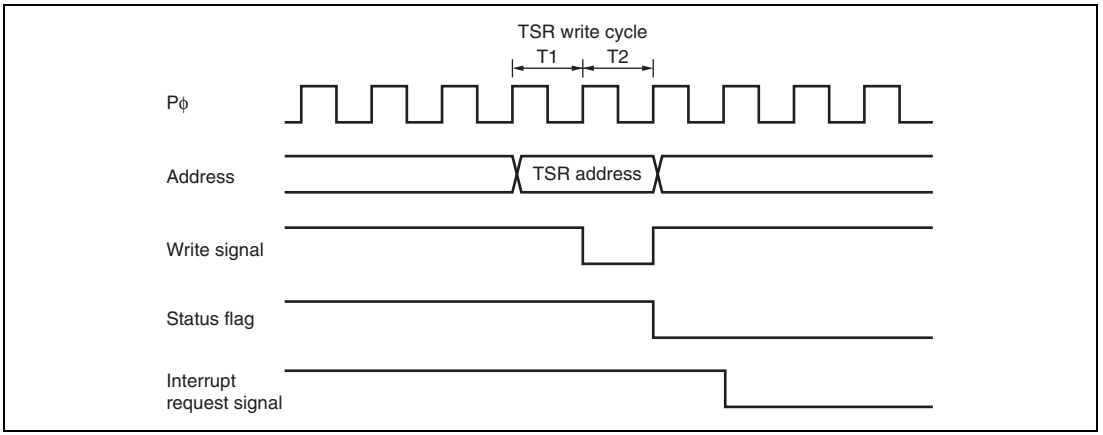


Figure 12.96 Timing for Status Flag Clearing by CPU

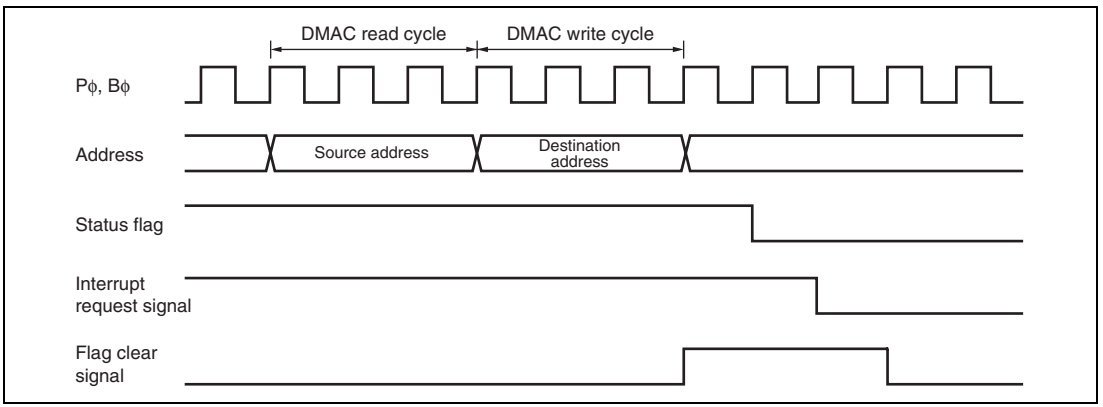


Figure 12.97 Timing for Status Flag Clearing by DMAC Activation

12.7 Usage Notes

12.7.1 Module Standby Mode Setting

MTU2 operation can be disabled or enabled using the standby control register. The initial setting is for MTU2 operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 30, Power-Down Modes.

12.7.2 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 MTU2 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 12.98 shows the input clock conditions in phase counting mode.

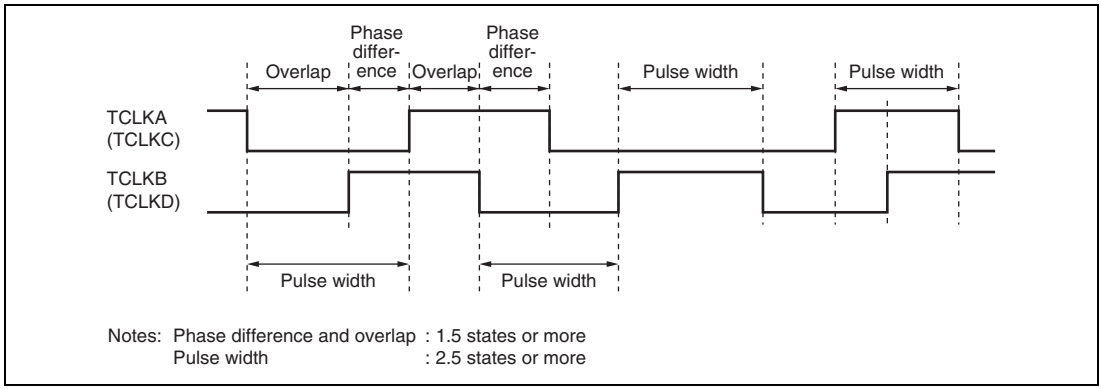


Figure 12.98 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

12.7.3 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

12.7.4 Contention 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 12.99 shows the timing in this case.

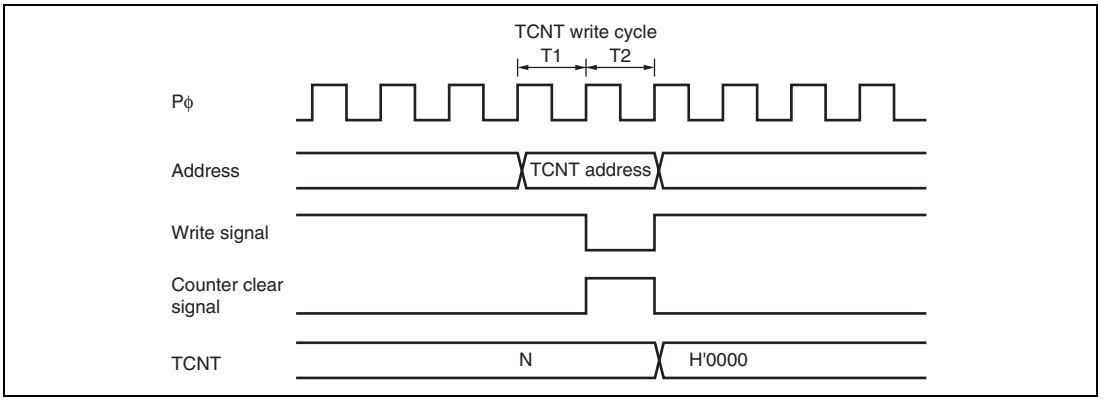


Figure 12.99 Contention between TCNT Write and Clear Operations

12.7.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 12.100 shows the timing in this case.

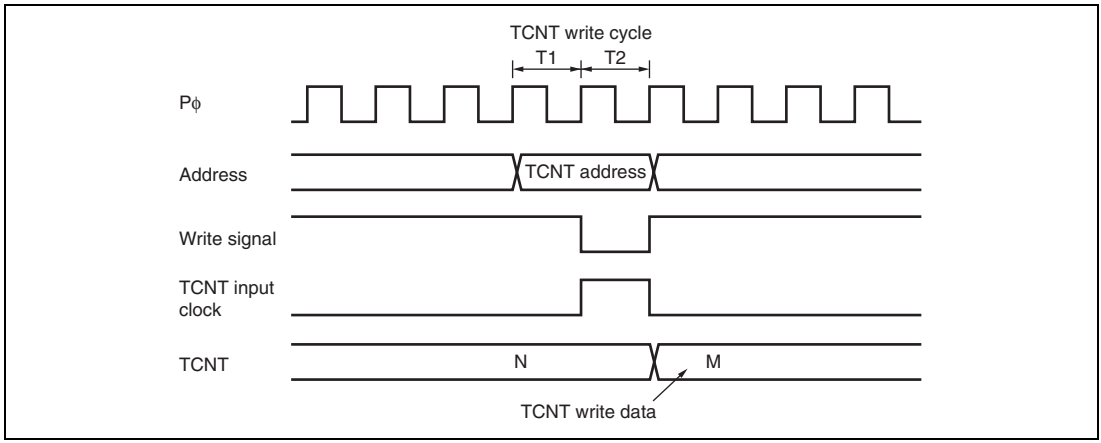


Figure 12.100 Contention between TCNT Write and Increment Operations

12.7.6 Contention between TGR Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write is executed and the compare match signal is also generated.

Figure 12.101 shows the timing in this case.

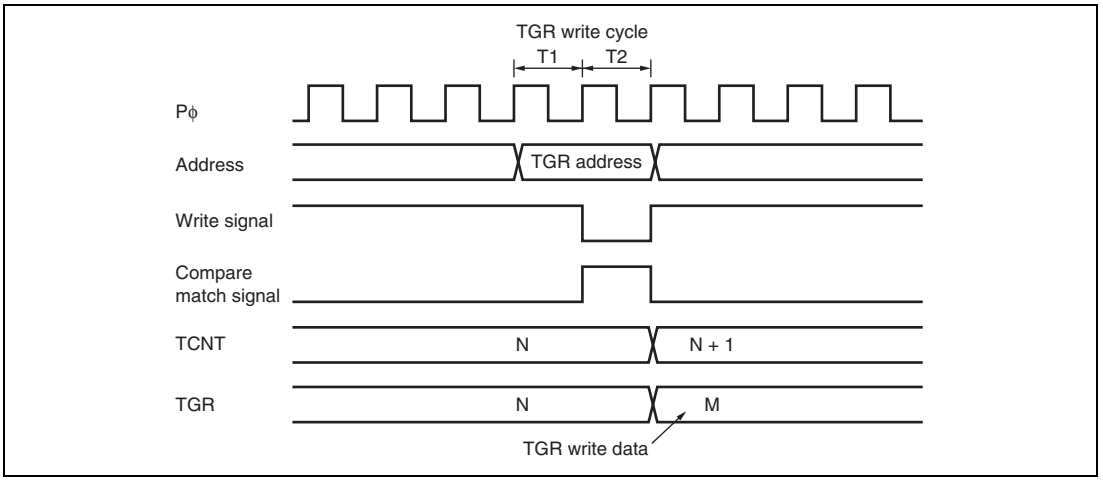


Figure 12.101 Contention between TGR Write and Compare Match

12.7.7 Contention between Buffer Register Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation is the data after write.

Figure 12.102 shows the timing in this case.

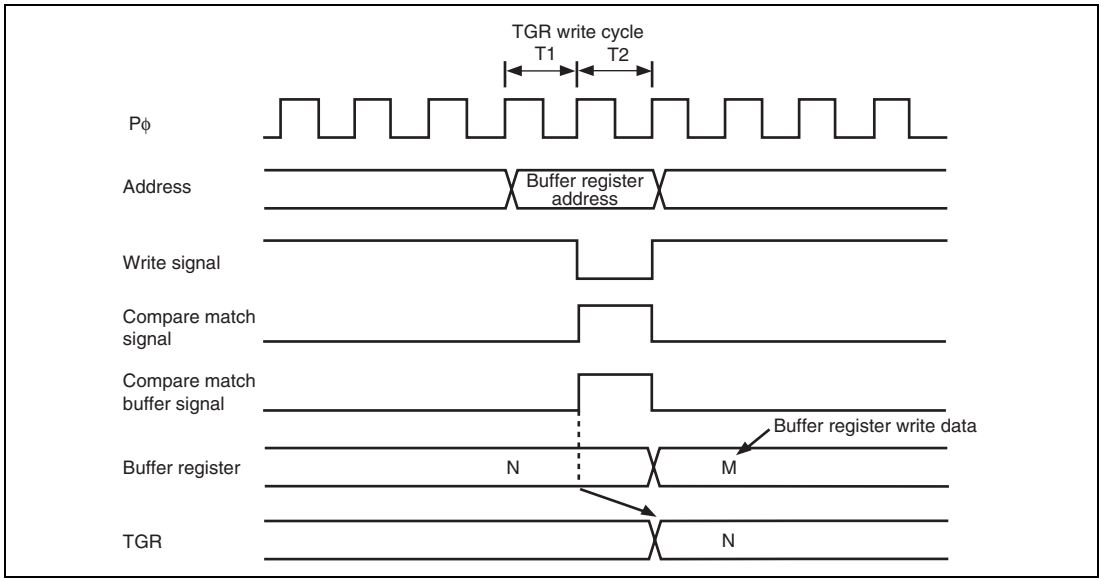


Figure 12.102 Contention between Buffer Register Write and Compare Match

12.7.8 Contention between Buffer Register Write and TCNT Clear

When the buffer transfer timing is set at the TCNT clear by the buffer transfer mode register (TBTM), if TCNT clear occurs in the T2 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation is the data before write.

Figure 12.103 shows the timing in this case.

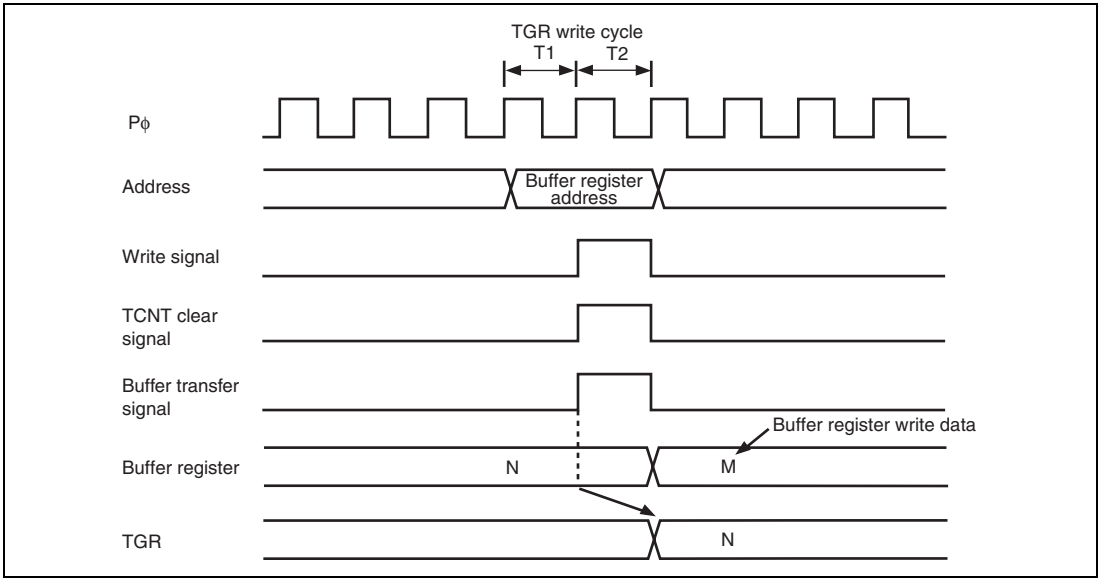


Figure 12.103 Contention between Buffer Register Write and TCNT Clear

12.7.9 Contention between TGR Read and Input Capture

If an input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data in the buffer before input capture transfer.

Figure 12.104 shows the timing in this case.

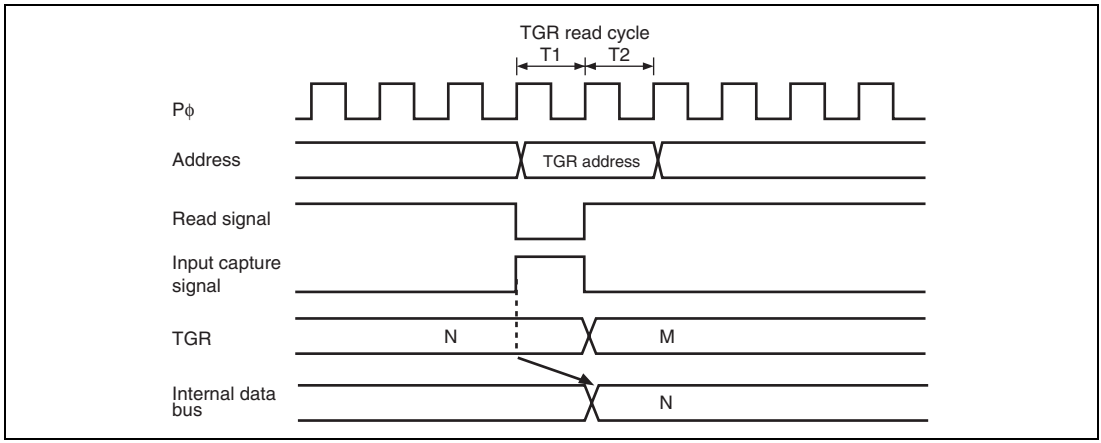


Figure 12.104 Contention between TGR Read and Input Capture

12.7.10 Contention 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 12.105 shows the timing in this case.

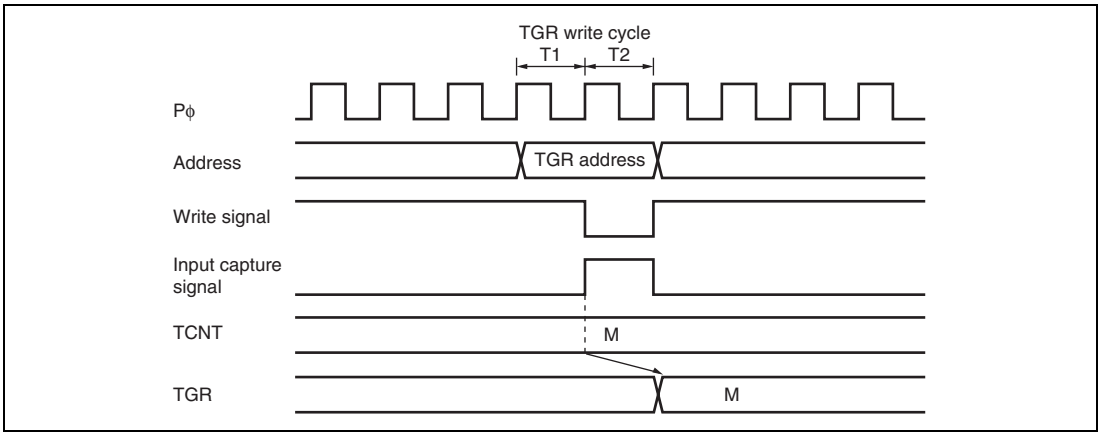


Figure 12.105 Contention between TGR Write and Input Capture

12.7.11 Contention 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 12.106 shows the timing in this case.

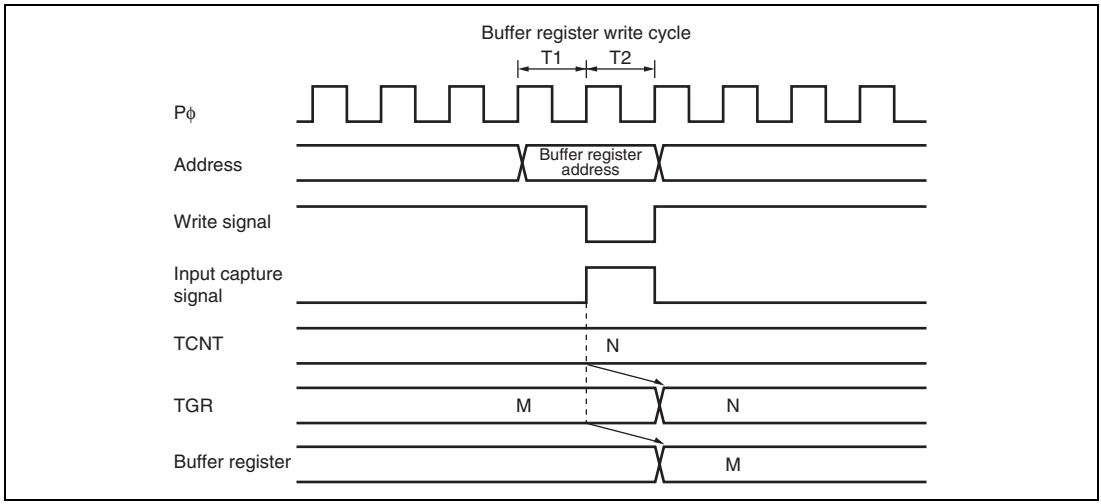


Figure 12.106 Contention between Buffer Register Write and Input Capture

12.7.12 TCNT2 Write and Overflow/Underflow Contention in Cascade Connection

With timer counters TCNT1 and TCNT2 in a cascade connection, when a contention 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 12.107.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setting TCNT clearing.

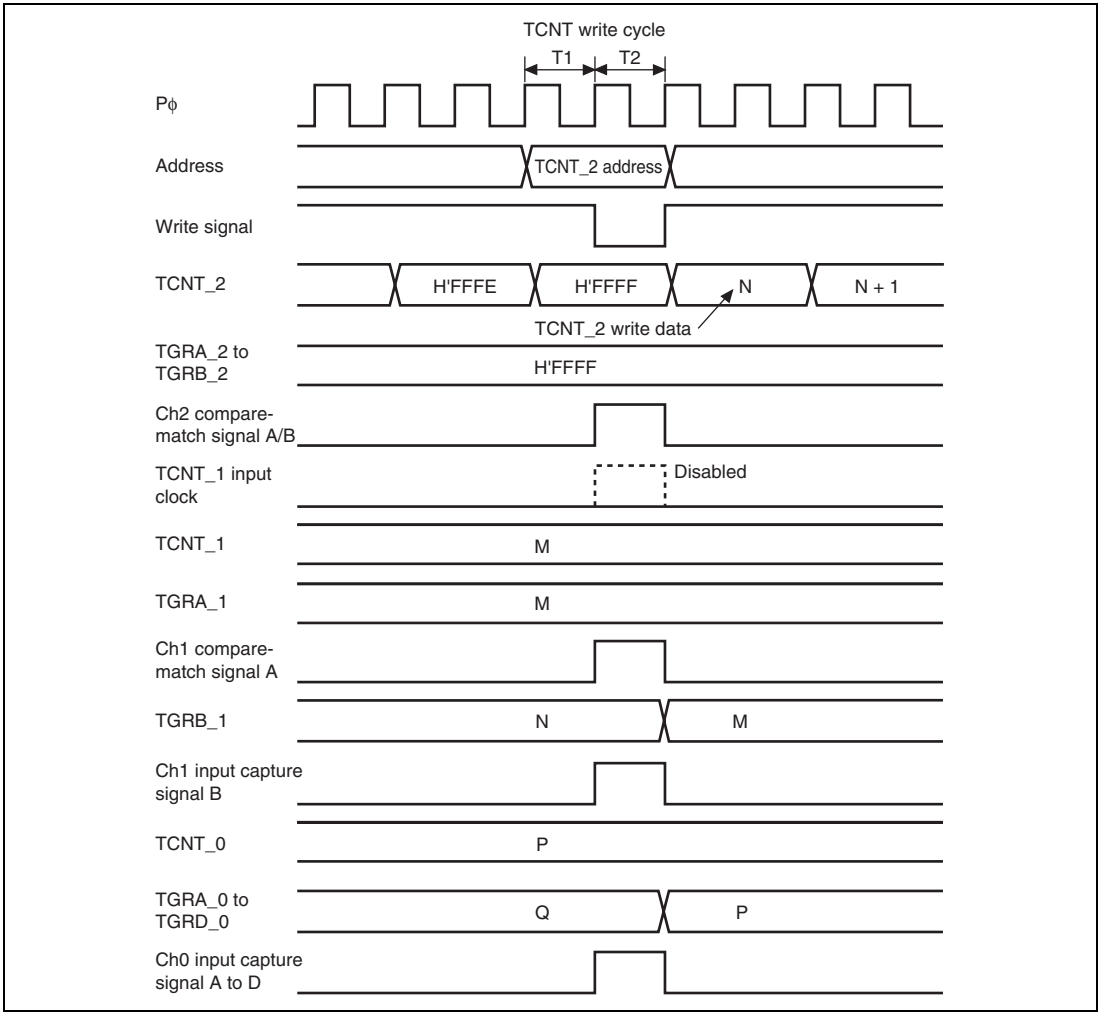


Figure 12.107 TCNT_2 Write and Overflow/Underflow Contention with Cascade Connection

12.7.13 Counter Value during Complementary PWM Mode Stop

When counting operation is suspended with TCNT_3 and TCNT_4 in complementary PWM mode, TCNT_3 has the timer dead time register (TDDR) value, and TCNT_4 is held at H'0000.

When restarting complementary PWM mode, counting begins automatically from the initialized state. This explanatory diagram is shown in figure 12.108.

When counting begins in another operating mode, be sure that TCNT_3 and TCNT_4 are set to the initial values.

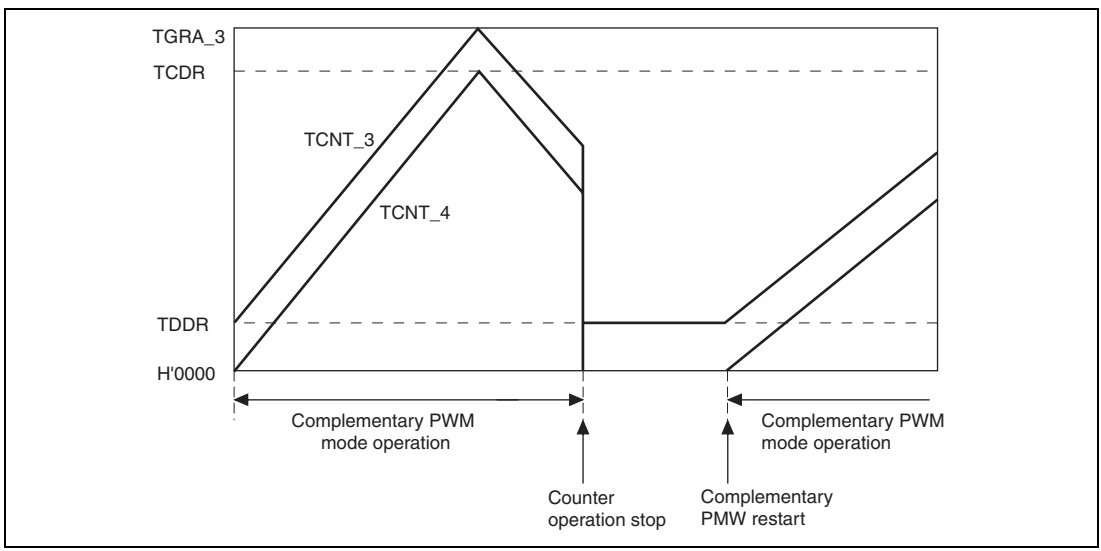


Figure 12.108 Counter Value during Complementary PWM Mode Stop

12.7.14 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, TGRA_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 TGRA_4, and TCBR functions as the TCDR's buffer register.

12.7.15 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 TGRA_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 12.109 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.

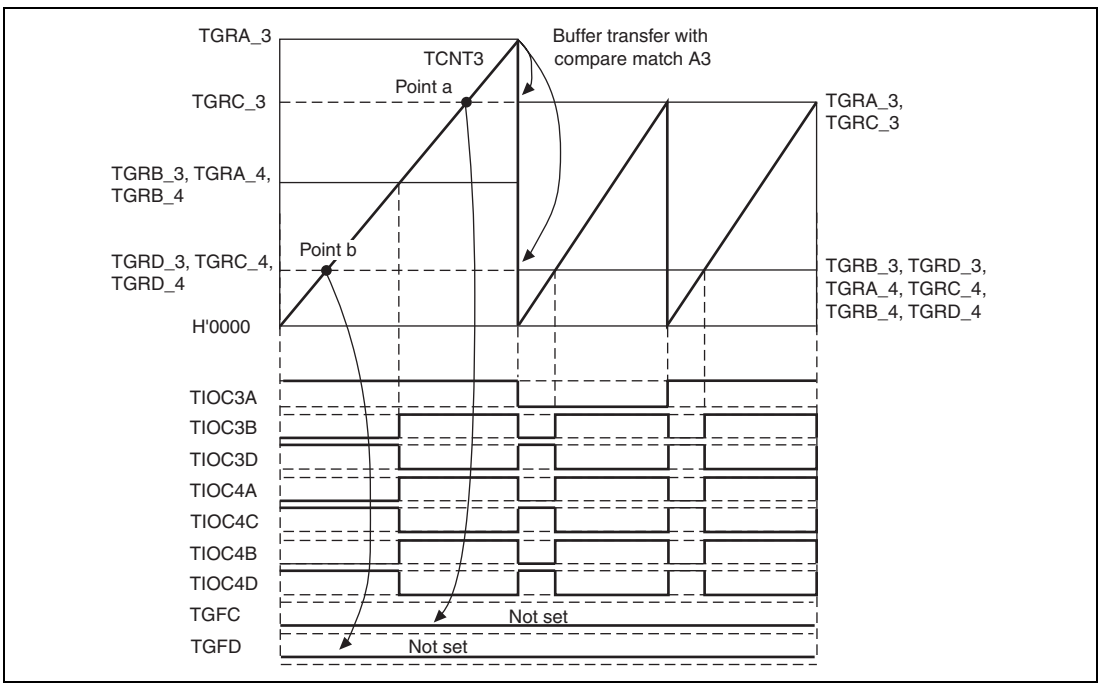


Figure 12.109 Buffer Operation and Compare-Match Flags in Reset Synchronous PWM Mode

12.7.16 Overflow Flags in Reset Synchronous PWM Mode

When set to reset synchronous 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 synchronous 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 12.110 shows a TCFV bit operation example in reset synchronous 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.

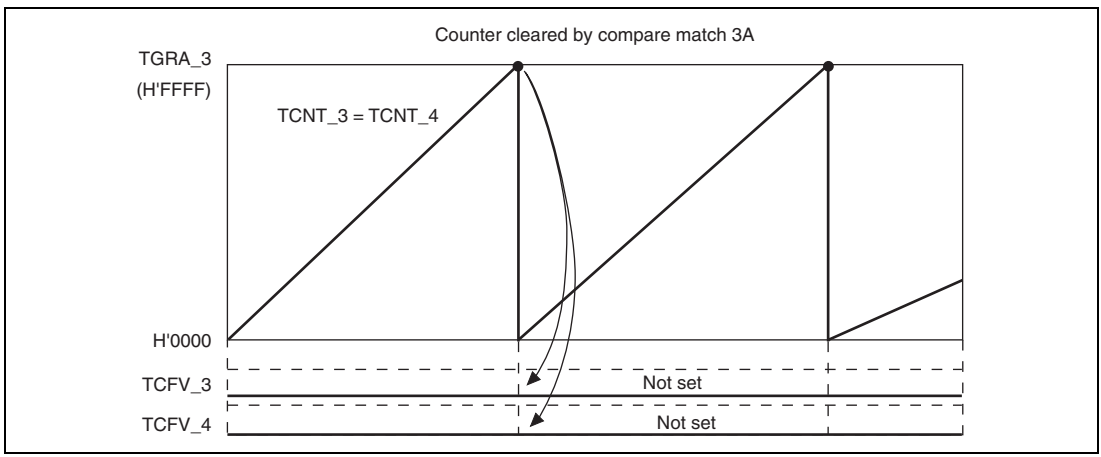


Figure 12.110 Reset Synchronous PWM Mode Overflow Flag

12.7.17 Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 12.111 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

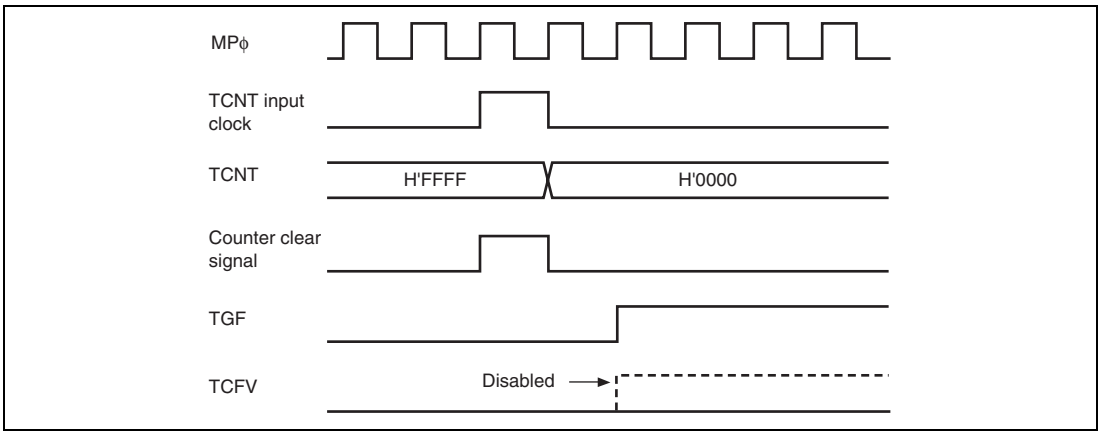


Figure 12.111 Contention between Overflow and Counter Clearing

12.7.18 Contention between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 12.112 shows the operation timing when there is contention between TCNT write and overflow.

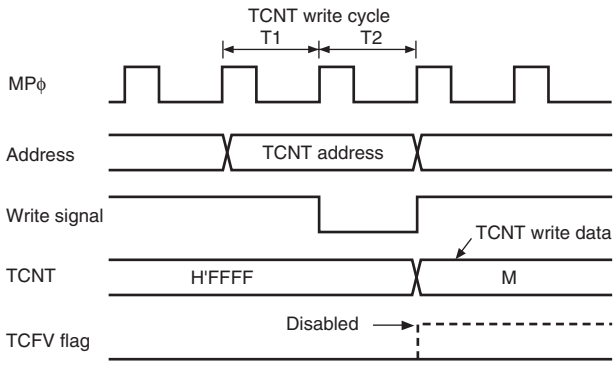


Figure 12.112 Contention between TCNT Write and Overflow

12.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from channel 3 or 4 normal operation or PWM mode 1 to reset-synchronized PWM mode, if the counter is halted with the output pins (TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, TIOC4D) in the high-level state, followed by the transition to reset-synchronized PWM mode and operation in that mode, the initial pin output will not be correct.

When making a transition from normal operation to reset-synchronized 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-synchronized 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-synchronized PWM mode.

12.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode

When channels 3 and 4 are in complementary PWM mode or reset-synchronized 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-synchronized PWM mode, TIOR should be set to H'00.

12.7.21 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 DMAC activation source. Interrupts should therefore be disabled before entering module standby mode.

12.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection

When timer counters 1 and 2 (TCNT_1 and TCNT_2) are operated as a 32-bit counter in cascade connection, the cascade counter value cannot be captured successfully even if input-capture input is simultaneously done to TIOC1A and TIOC2A or to TIOC1B and TIOC2B. This is because the input timing of TIOC1A and TIOC2A or of TIOC1B and TIOC2B may not be the same when external input-capture signals to be input into TCNT_1 and TCNT_2 are taken in synchronization with the internal clock. For example, TCNT_1 (the counter for upper 16 bits) does not capture the count-up value by overflow from TCNT_2 (the counter for lower 16 bits) but captures the count value before the count-up. In this case, the values of TCNT_1 = H'FFF1 and TCNT_2 = H'0000 should be transferred to TGRA_1 and TGRA_2 or to TGRB_1 and TGRB_2, but the values of TCNT_1 = H'FFF0 and TCNT_2 = H'0000 are erroneously transferred.

12.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode

In complementary PWM mode, when output waveform control during synchronous counter clearing is enabled (WRE in the TWCR register set to 1), the following problems may occur when condition (1) or condition (2), below, is satisfied.

- Dead time for the PWM output pins may be too short (or nonexistent).
- Active-level output from the PWM negative-phase pins may occur outside the correct active-level output interval

Condition (1): When synchronous clearing occurs in the PWM output dead time interval within initial output suppression interval (10) (figure 12.113).

Condition (2): When synchronous clearing occurs within initial output suppression interval (10) or (11) and TGRB_3 ≤ TDDR, TGRA_4 ≤ TDDR, or TGRB_4 ≤ TDDR is true (figure 12.114)

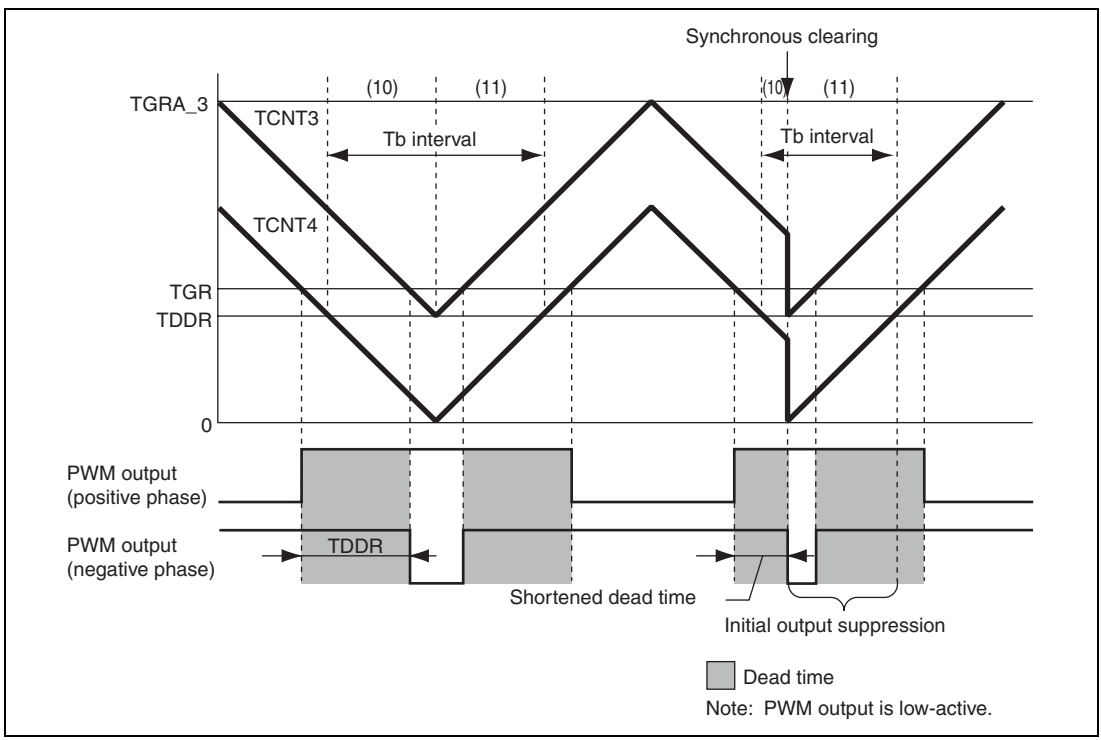


Figure 12.113 Condition (1) Synchronous Clearing Example

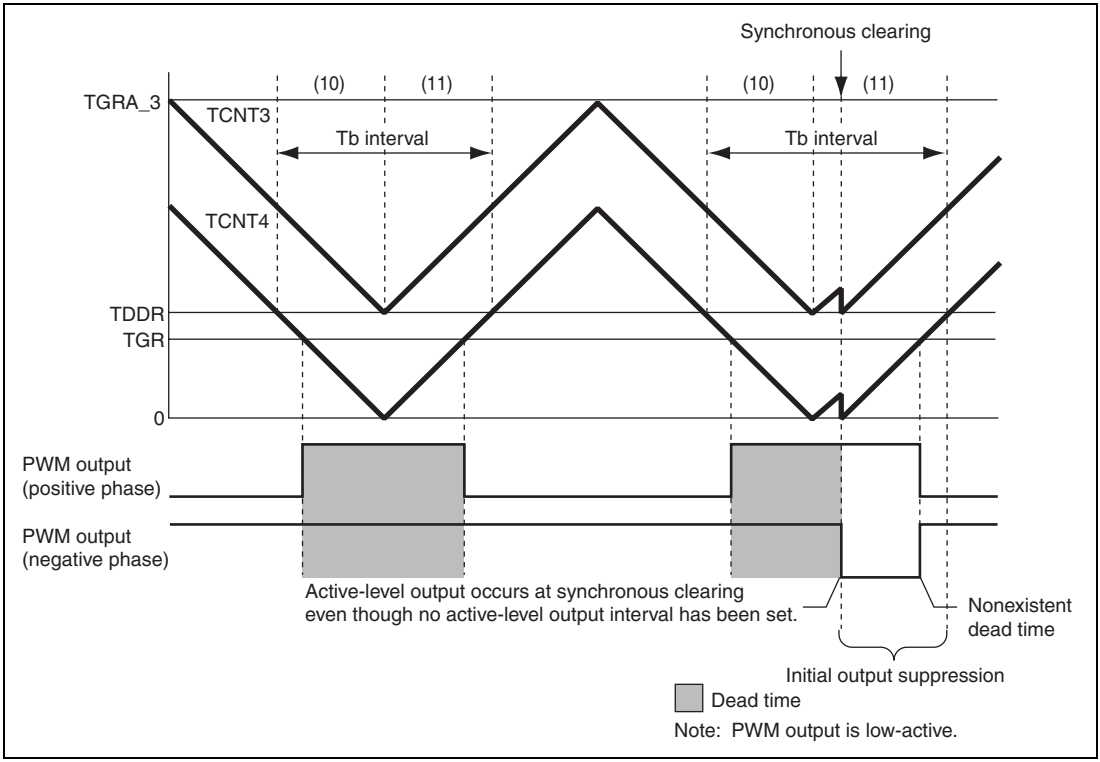


Figure 12.114 Condition (2) Synchronous Clearing Example

The following workaround can be used to avoid these problems.

When using synchronous clearing, make sure to set compare registers TGRB_3, TGRA_4, and TGRB_4 to a value twice or more the setting of dead time data register TDDR.

12.8 MTU2 Output Pin Initialization

12.8.1 Operating Modes

The MTU2 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-synchronized PWM mode (channels 3 and 4)

The MTU2 output pin initialization method for each of these modes is described in this section.

12.8.2 Reset Start Operation

The MTU2 output pins (TIOC*) are initialized low by a power-on reset and in deep standby mode. Since MTU2 pin function selection is performed by the pin function controller (PFC), when the PFC is set, the MTU2 pin states at that point are output to the ports. When MTU2 output is selected by the PFC immediately after a reset, the MTU2 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 MTU2 output pins is completed.

Note: Channel number and port notation are substituted for *.

12.8.3 Operation in Case of Re-Setting Due to Error During Operation, etc.

If an error occurs during MTU2 operation, MTU2 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. 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 MTU2 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 12.57.

Table 12.57 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-synchronized PWM mode

12.8.4 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).

Note: Channel number is substituted for * indicated in this article.

Pin initialization procedures are described below for the numbered combinations in table 12.57. The active level is assumed to be low.

(1) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Normal Mode

Figure 12.115 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

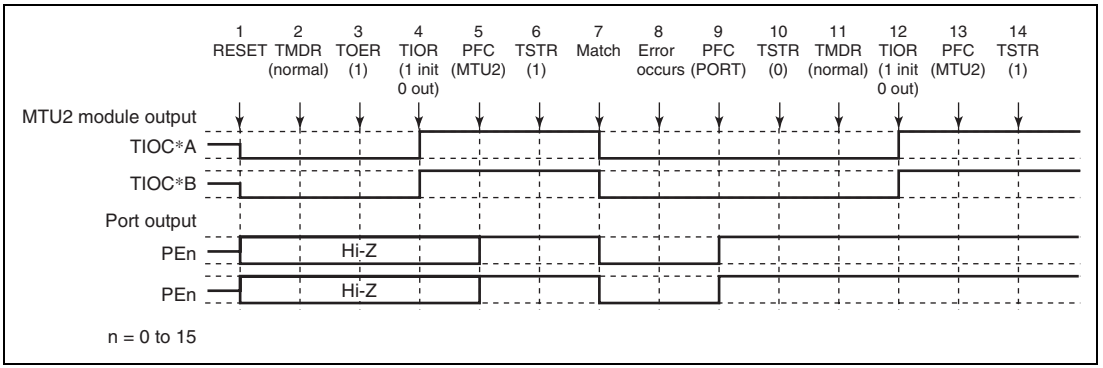


Figure 12.115 Error Occurrence in Normal Mode, Recovery in Normal Mode

1. After a reset, MTU2 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 MTU2 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 MTU2 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 12.116 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.

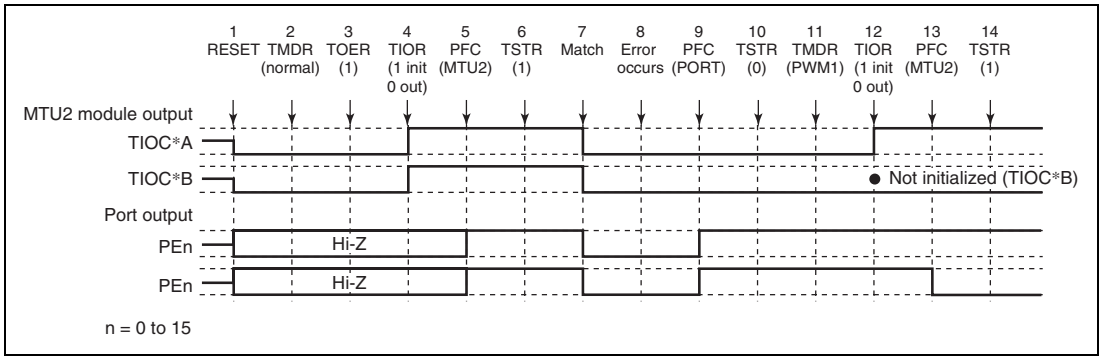


Figure 12.116 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.115.

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 MTU2 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 12.117 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.

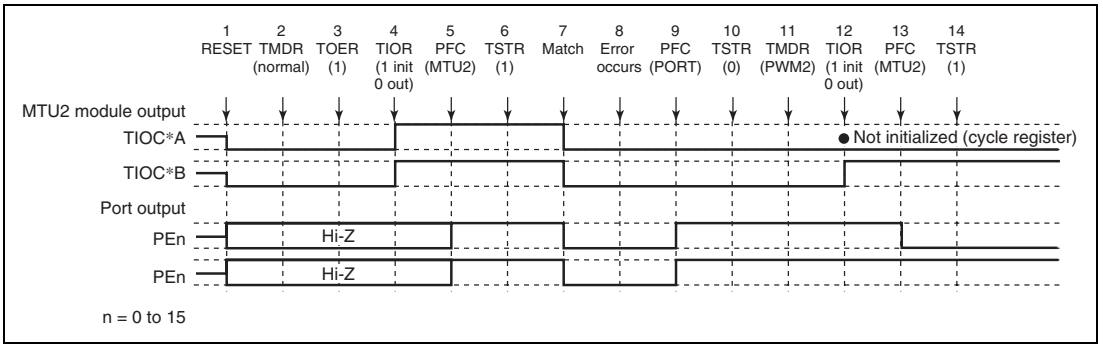


Figure 12.117 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

1 to 10 are the same as in figure 12.115.

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 MTU2 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 12.118 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.

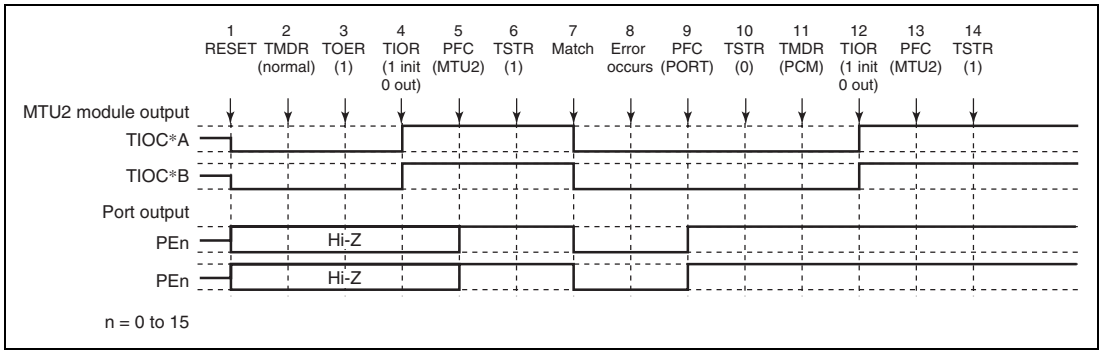


Figure 12.118 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 12.115.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU2 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 12.119 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.

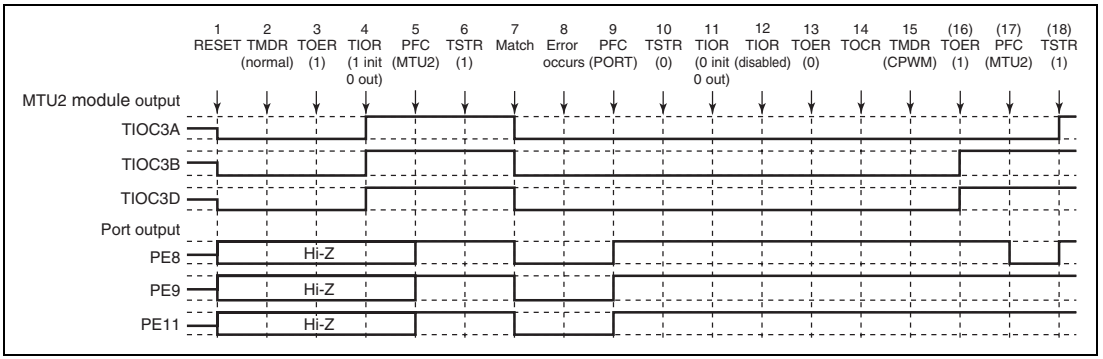


Figure 12.119 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.115.

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 MTU2 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-Synchronized PWM Mode

Figure 12.120 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

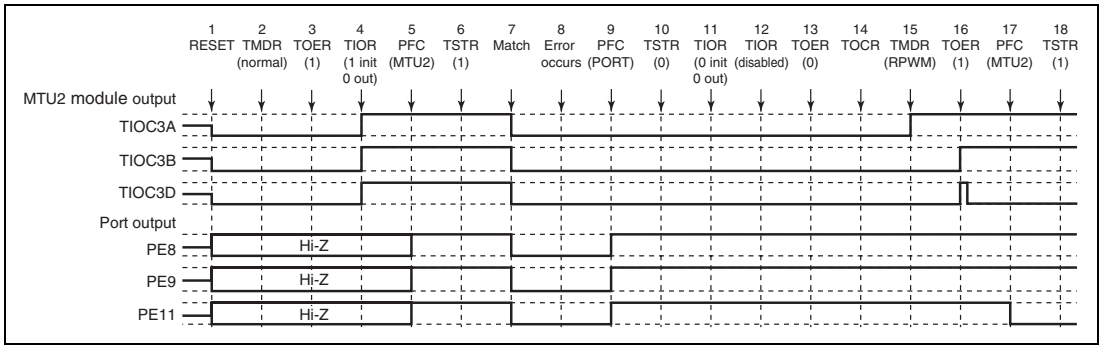


Figure 12.120 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

1 to 13 are the same as in figure 12.115.

14. Select the reset-synchronized PWM output level and cyclic output enabling/disabling with TOCR.
15. Set reset-synchronized PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set MTU2 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 12.121 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.

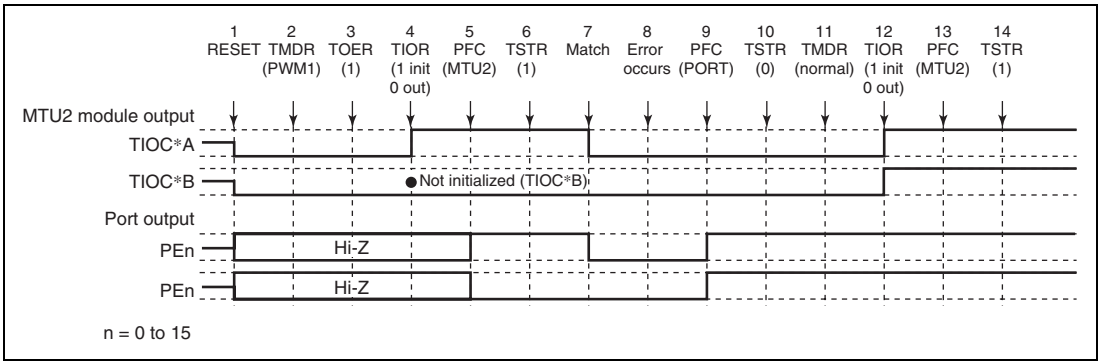


Figure 12.121 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

1. After a reset, MTU2 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 MTU2 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 MTU2 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 12.122 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.

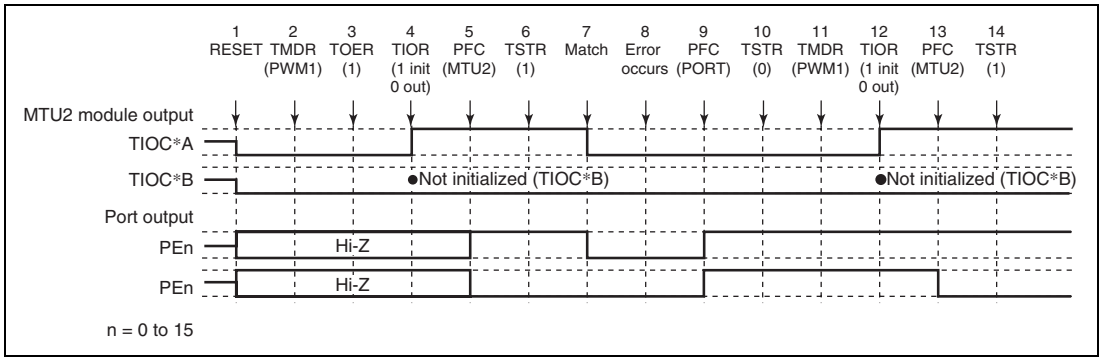


Figure 12.122 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.121.

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 MTU2 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 12.123 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.

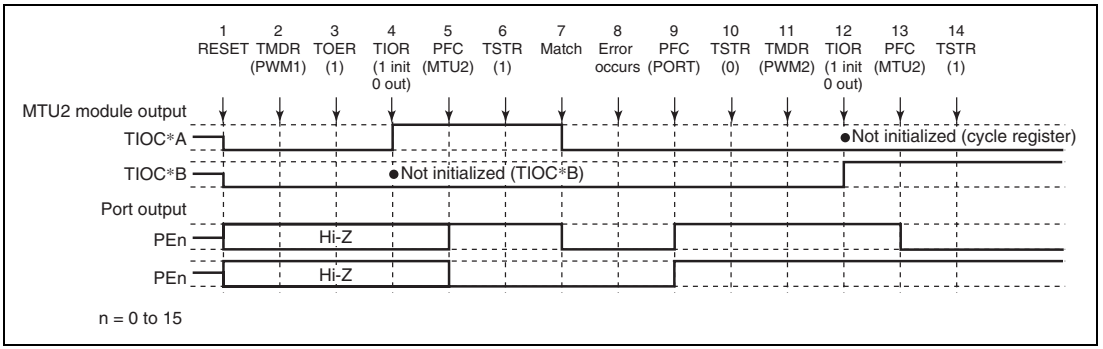


Figure 12.123 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

1 to 10 are the same as in figure 12.121.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
13. Set MTU2 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 12.124 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.

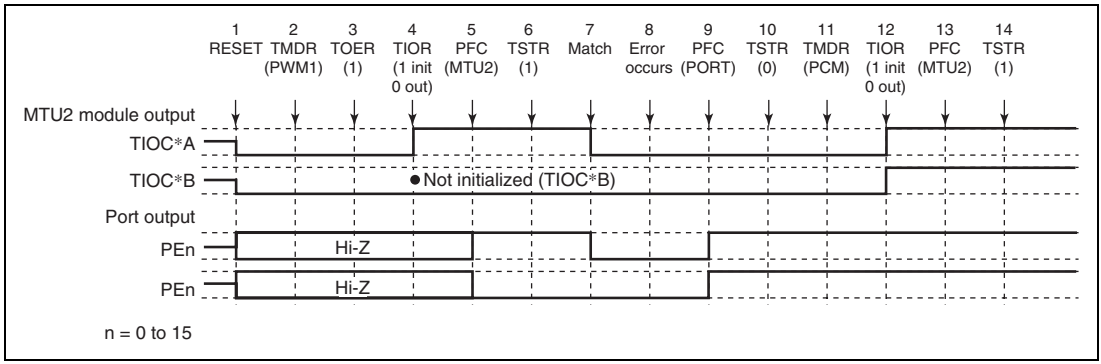


Figure 12.124 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 12.121.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU2 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 12.125 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.

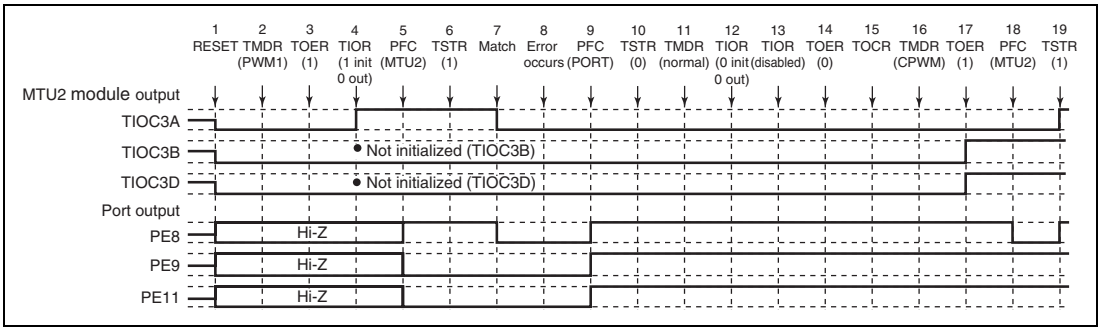


Figure 12.125 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.121.

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 MTU2 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-Synchronized PWM Mode

Figure 12.126 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

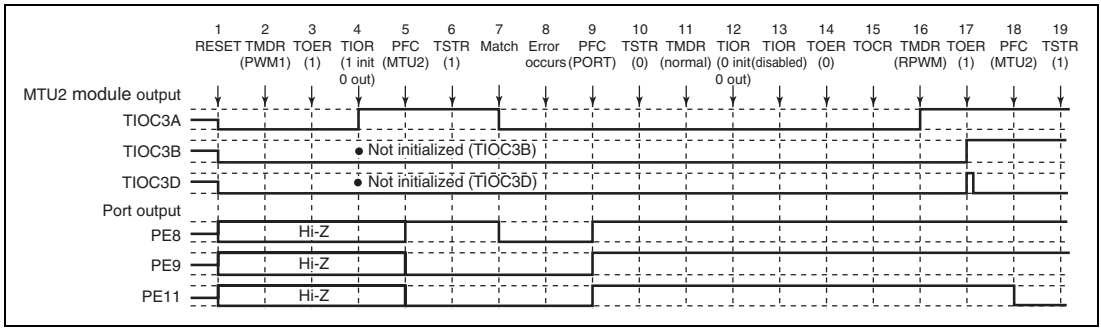


Figure 12.126 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

1 to 14 are the same as in figure 12.125.

15. Select the reset-synchronized PWM output level and cyclic output enabling/disabling with TOCR.
16. Set reset-synchronized PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set MTU2 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 12.127 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.

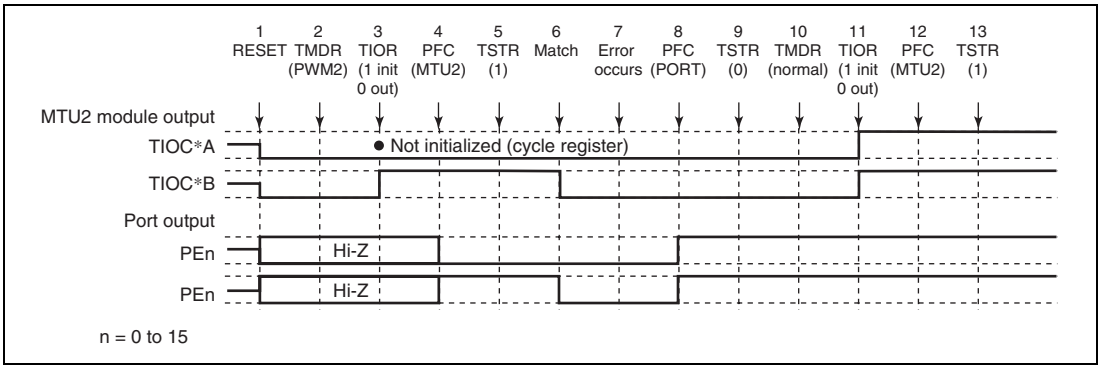


Figure 12.127 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

1. After a reset, MTU2 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 MTU2 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 MTU2 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 12.128 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.

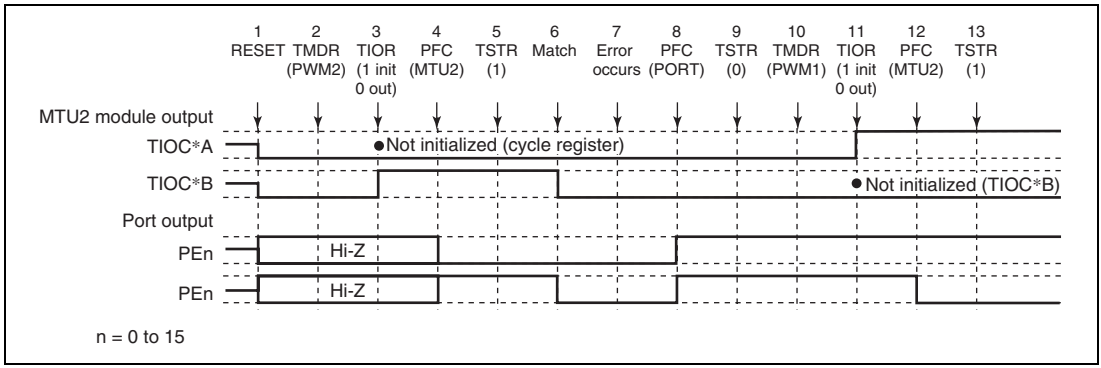


Figure 12.128 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

1 to 9 are the same as in figure 12.127.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
12. Set MTU2 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 12.129 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.

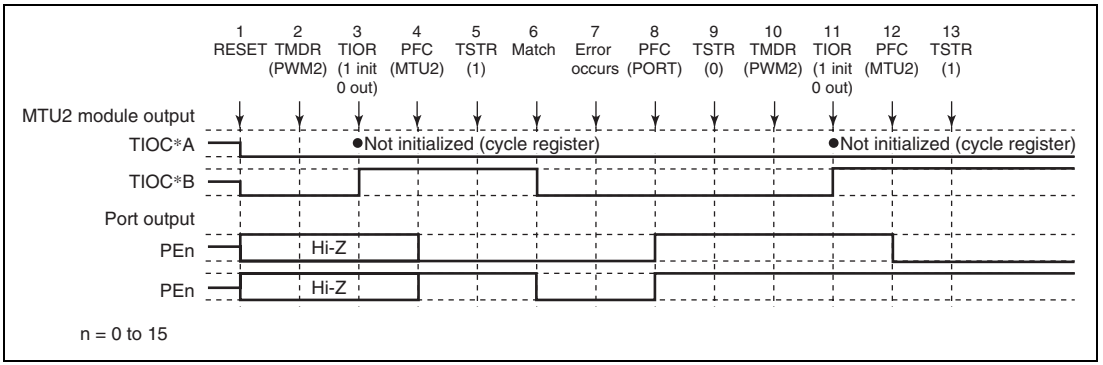


Figure 12.129 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

1 to 9 are the same as in figure 12.127.

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 MTU2 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 12.130 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.

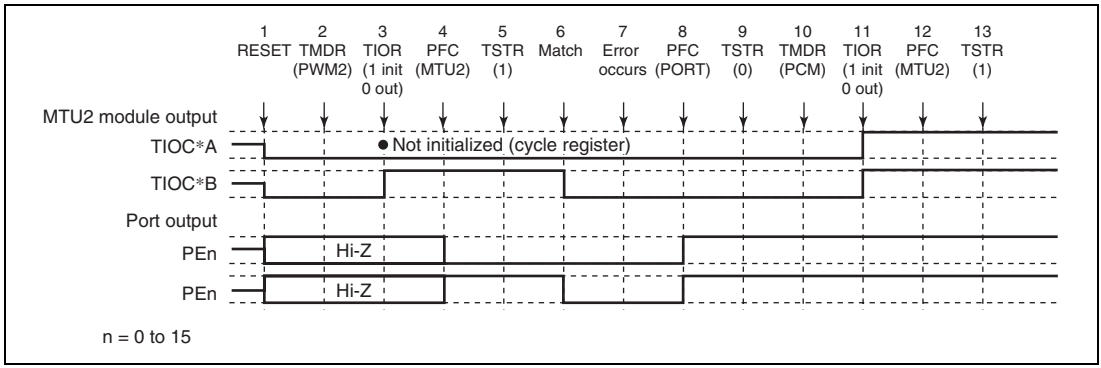


Figure 12.130 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 12.127.

10. Set phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU2 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 12.131 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.

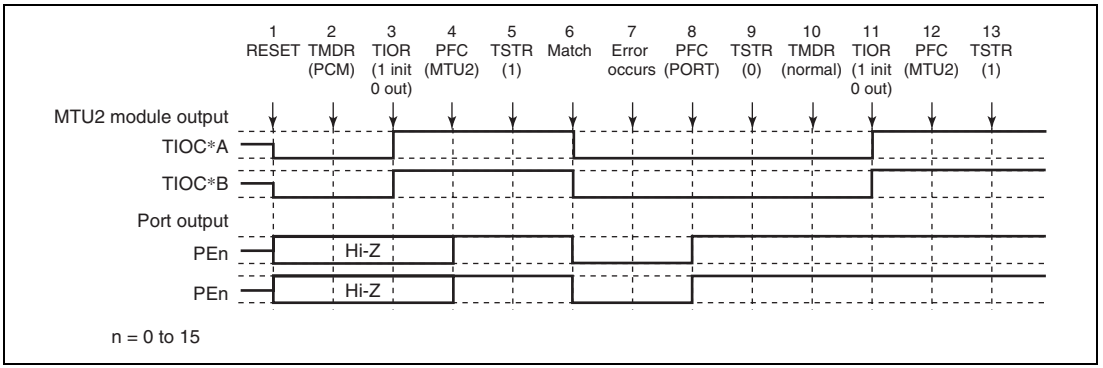


Figure 12.131 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

1. After a reset, MTU2 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 MTU2 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 MTU2 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 12.132 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.

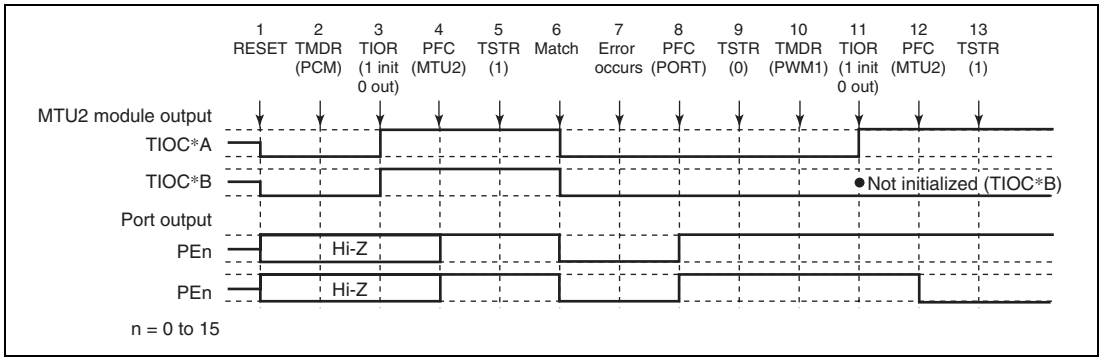


Figure 12.132 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

1 to 9 are the same as in figure 12.131.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
12. Set MTU2 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 12.133 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.

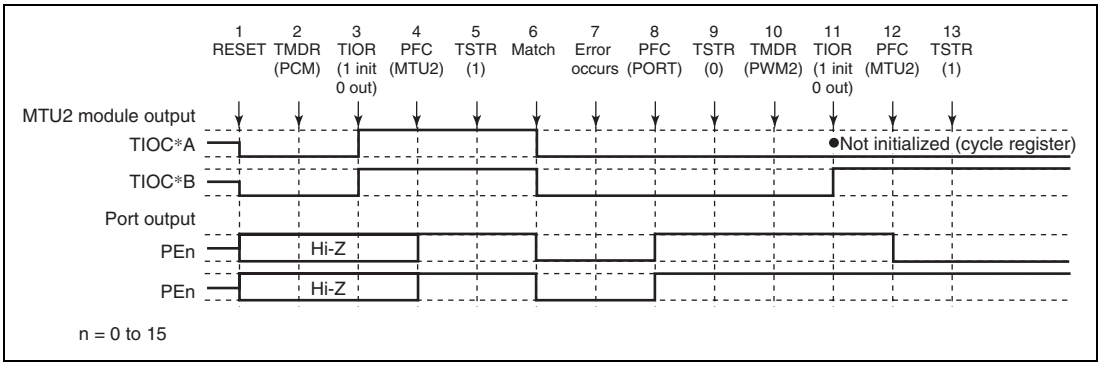


Figure 12.133 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

1 to 9 are the same as in figure 12.131.

10. Set PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set MTU2 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 12.134 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.

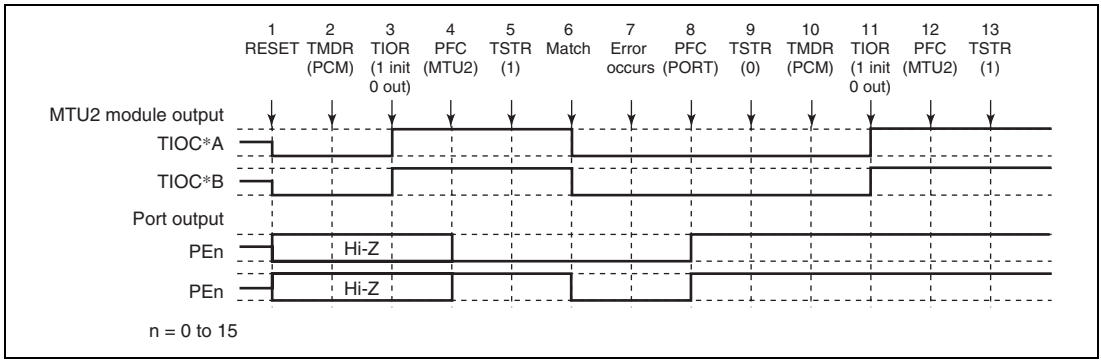


Figure 12.134 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 12.131.

10. Not necessary when restarting in phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU2 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 12.135 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.

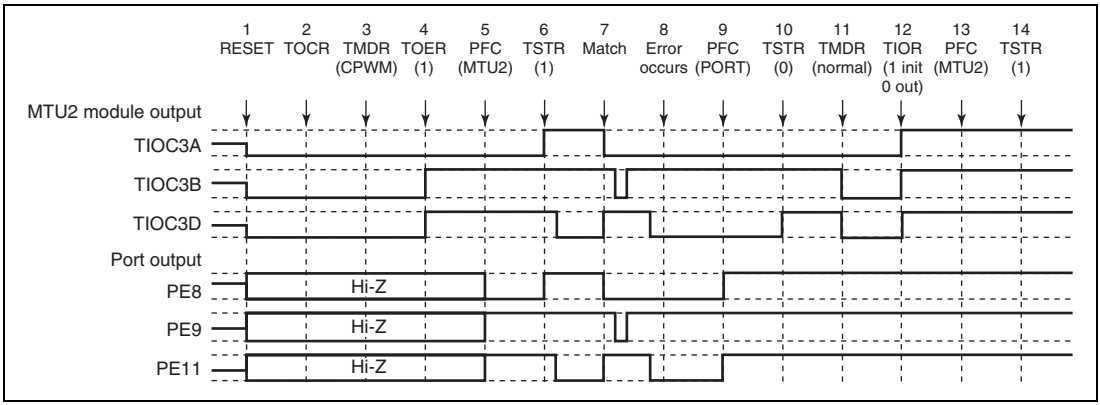


Figure 12.135 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

1. After a reset, MTU2 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 MTU2 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. (MTU2 output becomes the complementary PWM output initial value.)
11. Set normal mode. (MTU2 output goes low.)
12. Initialize the pins with TIOR.
13. Set MTU2 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 12.136 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.

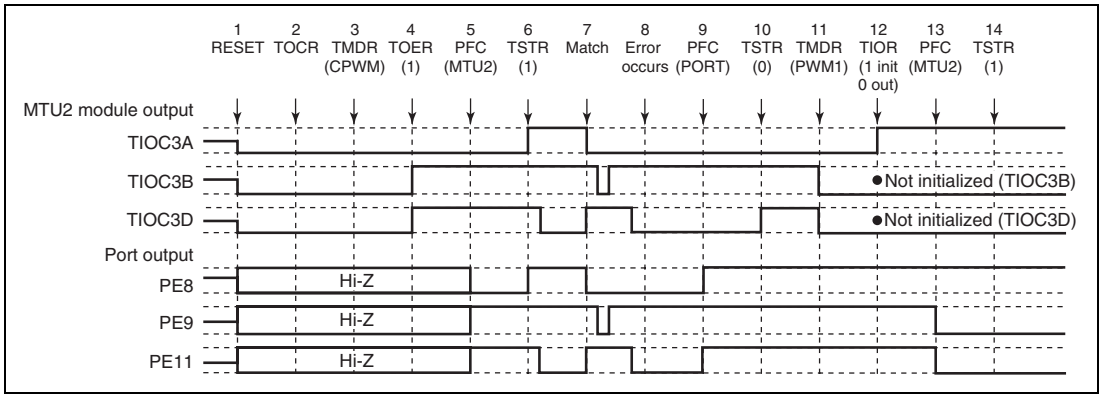


Figure 12.136 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.135.

11. Set PWM mode 1. (MTU2 output goes low.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
13. Set MTU2 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 12.137 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).

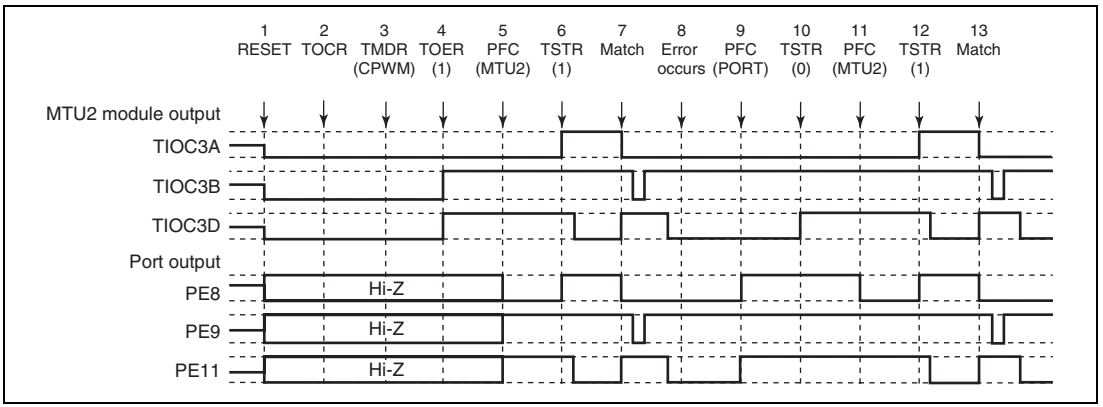


Figure 12.137 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.135.

11. Set MTU2 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 12.138 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).

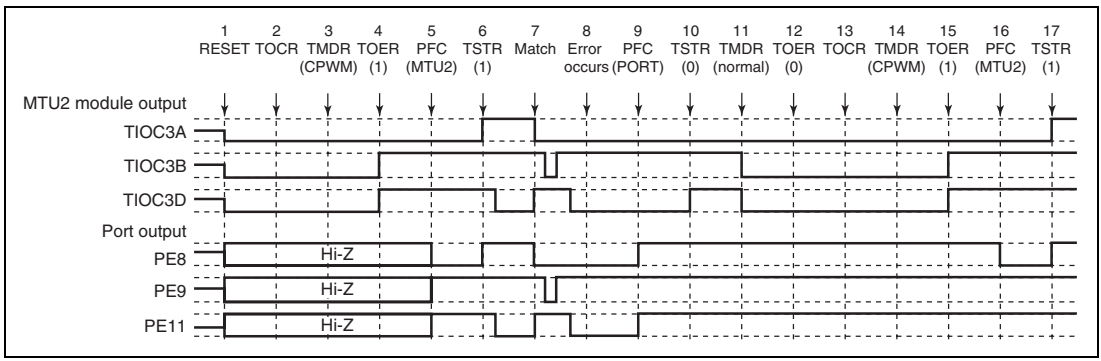


Figure 12.138 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.135.

11. Set normal mode and make new settings. (MTU2 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 MTU2 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-Synchronized PWM Mode

Figure 12.139 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode.

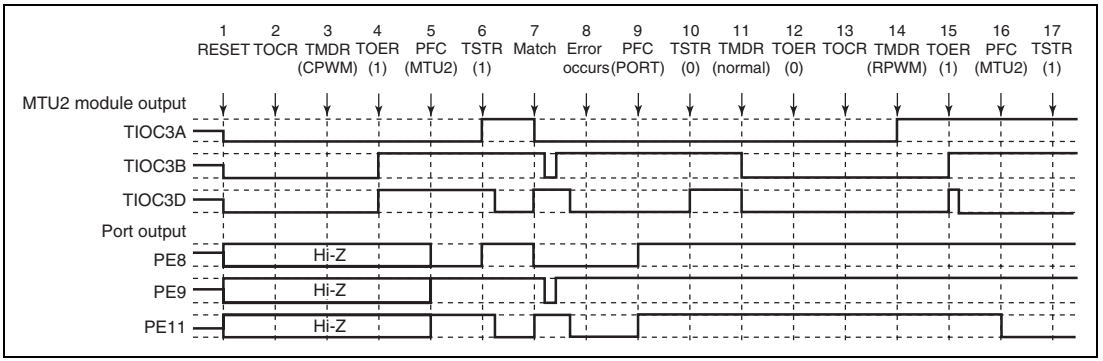


Figure 12.139 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 12.135.

11. Set normal mode. (MTU2 output goes low.)
12. Disable channel 3 and 4 output with TOER.
13. Select the reset-synchronized PWM mode output level and cyclic output enabling/disabling with TOCR.
14. Set reset-synchronized PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set MTU2 output with the PFC.
17. Operation is restarted by TSTR.

(26) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Normal Mode

Figure 12.140 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

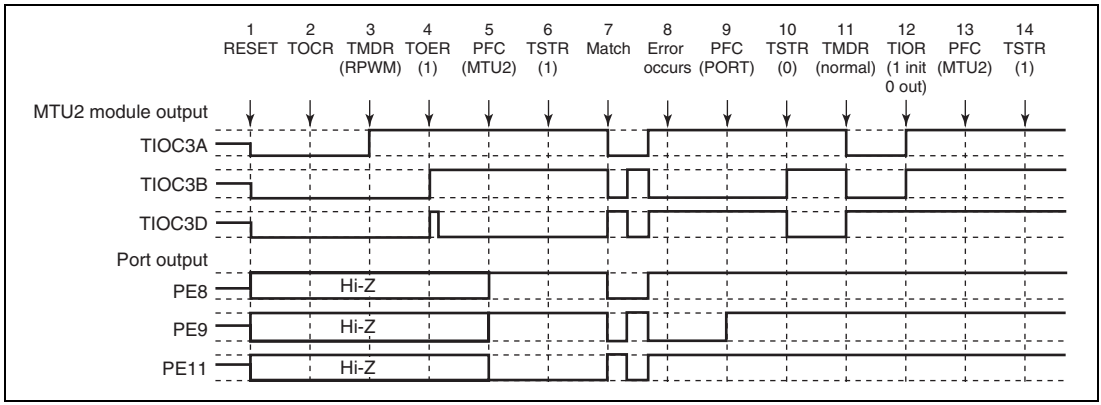


Figure 12.140 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Select the reset-synchronized PWM output level and cyclic output enabling/disabling with TOCR.
3. Set reset-synchronized PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. The reset-synchronized 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. (MTU2 output becomes the reset-synchronized PWM output initial value.)
11. Set normal mode. (MTU2 positive phase output is low, and negative phase output is high.)
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(27) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 12.141 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

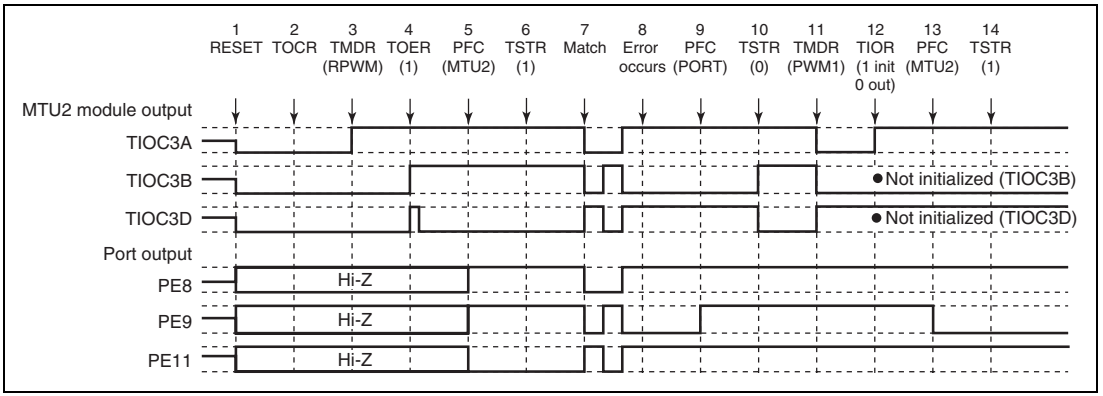


Figure 12.141 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.140.

11. Set PWM mode 1. (MTU2 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 MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(28) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 12.142 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

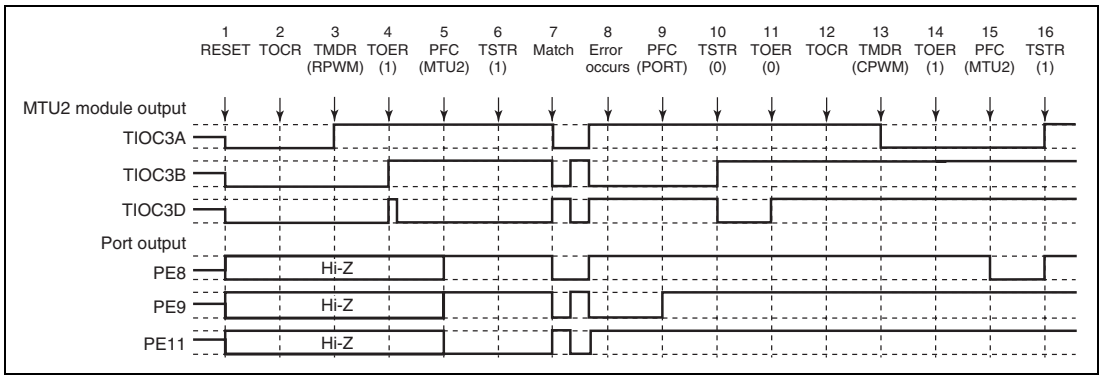


Figure 12.142 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.140.

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 MTU2 cyclic output pin goes low.)
14. Enable channel 3 and 4 output with TOER.
15. Set MTU2 output with the PFC.
16. Operation is restarted by TSTR.

(29) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 12.143 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

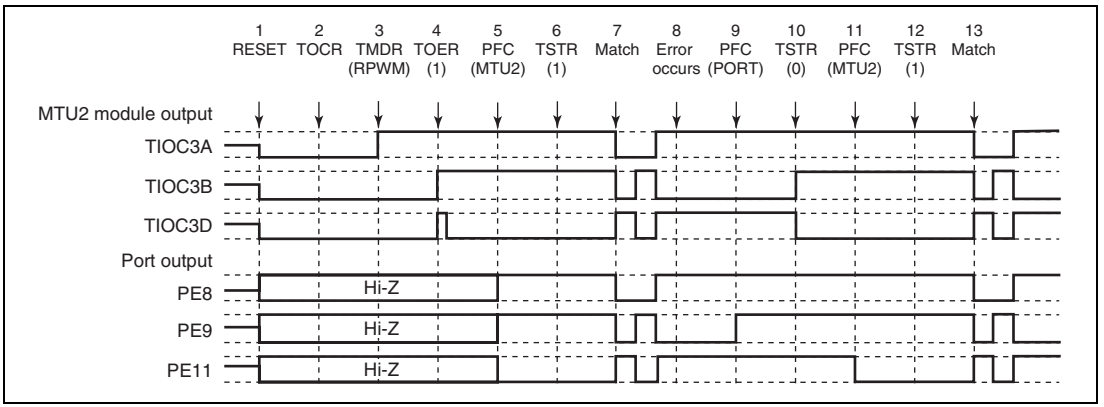


Figure 12.143 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 12.140.

11. Set MTU2 output with the PFC.
12. Operation is restarted by TSTR.
13. The reset-synchronized PWM waveform is output on compare-match occurrence.

Section 13 Compare Match Timer (CMT)

This LSI has an on-chip compare match timer module (CMT) consisting of two units of two-channel 16-bit timers, which makes a total of four channels. The CMT has a 16-bit counter, and can generate interrupts at set intervals.

13.1 Features

- Independent selection of four counter input clocks at two channels
  Any of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) can be selected.
- Selection of DMA transfer request or interrupt request generation on compare match by DMAC setting
- When not in use, the CMT can be stopped by halting its clock supply to reduce power consumption.

Figure 13.1 shows a block diagram of CMT.

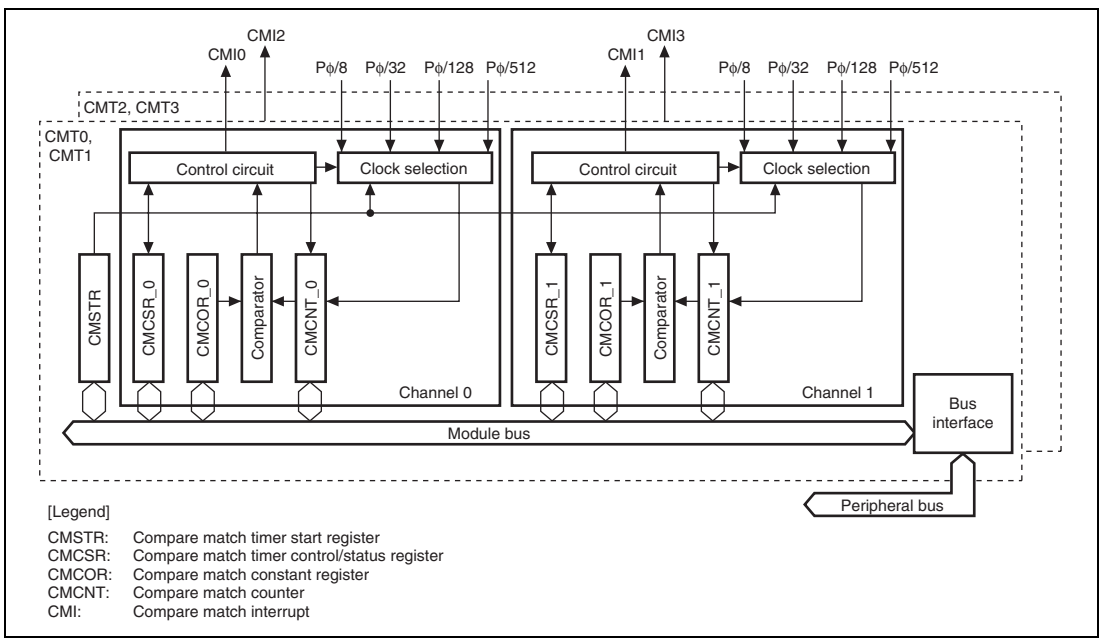


Figure 13.1 Block Diagram of CMT

13.2 Register Descriptions

The CMT has the following registers.

Table 13.1 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common (0, 1)	Compare match timer start register	CMSTR_01	R/W	H'0000	H'FFFE3000	16
0	Compare match timer control/ status register_0	CMCSR_0	R/W	H'0000	H'FFFE3002	16
	Compare match counter_0	CMCNT_0	R/W	H'0000	H'FFFE3004	8, 16
	Compare match constant register_0	CMCOR_0	R/W	H'FFFF	H'FFFE3006	8, 16
1	Compare match timer control/ status register_1	CMCSR_1	R/W	H'0000	H'FFFE3008	16
	Compare match counter_1	CMCNT_1	R/W	H'0000	H'FFFE300A	8, 16
	Compare match constant register_1	CMCOR_1	R/W	H'FFFF	H'FFFE300C	8, 16
Common (2, 3)	Compare match timer start register	CMSTR_23	R/W	H'0000	H'FFFE3400	16
2	Compare match timer control/ status register_2	CMCSR_2	R/W	H'0000	H'FFFE3402	16
	Compare match counter_2	CMCNT_2	R/W	H'0000	H'FFFE3404	8, 16
	Compare match constant register_2	CMCOR_2	R/W	H'FFFF	H'FFFE3406	8, 16
3	Compare match timer control/ status register_3	CMCSR_3	R/W	H'0000	H'FFFE3408	16
	Compare match counter_3	CMCNT_3	R/W	H'0000	H'FFFE340A	8, 16
	Compare match constant register_3	CMCOR_3	R/W	H'FFFF	H'FFFE340C	8, 16

13.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether compare match counter (CMCNT) operates or is stopped.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STR1/ STR3	STR0/ STR2
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	STR1/STR3	0	R/W	Count Start 1/3 Specifies whether compare match counter_1/3 operates or is stopped. 0: Counting by CMCNT_1/CMCNT_3 is stopped 1: Counting by CMCNT_1/CMCNT_3 is started
0	STR0/STR2	0	R/W	Count Start 0/2 Specifies whether compare match counter_0/2 operates or is stopped. 0: Counting by CMCNT_0/CMCNT_2 is stopped 1: Counting by CMCNT_0/CMCNT_2 is started

13.2.2 Compare Match Timer Control/Status Register (CMCSR)

CMCSR is a 16-bit register that indicates compare match generation, enables or disables interrupts, and selects the counter input clock.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMF	CMIE	-	-	-	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/W	R	R	R	R	R/W	R/W

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	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables compare match interrupt (CMI) generation when CMCNT and CMCOR values match (CMF = 1). 0: Compare match interrupt (CMI) disabled 1: Compare match interrupt (CMI) enabled
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, 0	CKS[1:0]	00	R/W	Clock Select These bits select the clock to be input to CMCNT from four internal clocks obtained by dividing the peripheral clock (Pϕ). When the STR bit in CMSTR is set to 1, CMCNT starts counting on the clock selected with bits CKS[1:0]. 00: Pϕ/8 01: Pϕ/32 10: Pϕ/128 11: Pϕ/512

Note: * Only 0 can be written to clear the flag after 1 is read.

13.2.3 Compare Match Counter (CMCNT)

CMCNT is a 16-bit register used as an up-counter. When the counter input clock is selected with bits CKS[1:0] 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 clearing any channels of the counter start bit from 1 to 0 in the compare match timer start register (CMSTR).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

13.2.4 Compare Match Constant Register (CMCOR)

CMCOR is a 16-bit register that sets the interval up to a compare match with CMCNT.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

13.3 Operation

13.3.1 Interval Count Operation

When an internal clock is selected with the CKS[1:0] 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. When the CMIE bit in CMCSR is set to 1 at this time, a compare match interrupt (CMI) is requested. CMCNT then starts counting up again from H'0000.

Figure 13.2 shows the operation of the compare match counter.

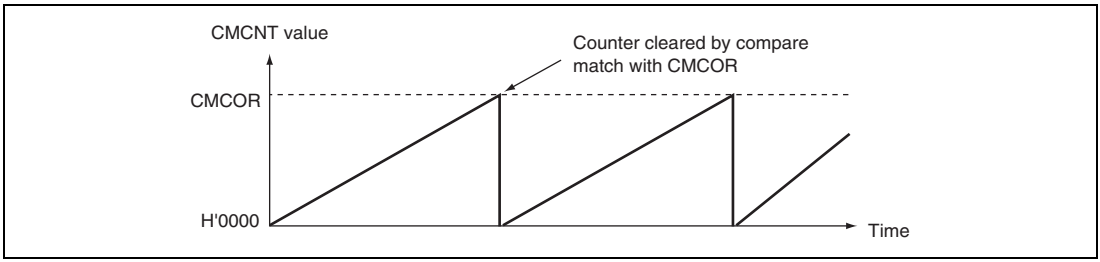


Figure 13.2 Counter Operation

13.3.2 CMCNT Count Timing

One of four clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) obtained by dividing the peripheral clock (Pϕ) can be selected with the CKS1 and CKS0 bits in CMCSR. Figure 13.3 shows the timing.

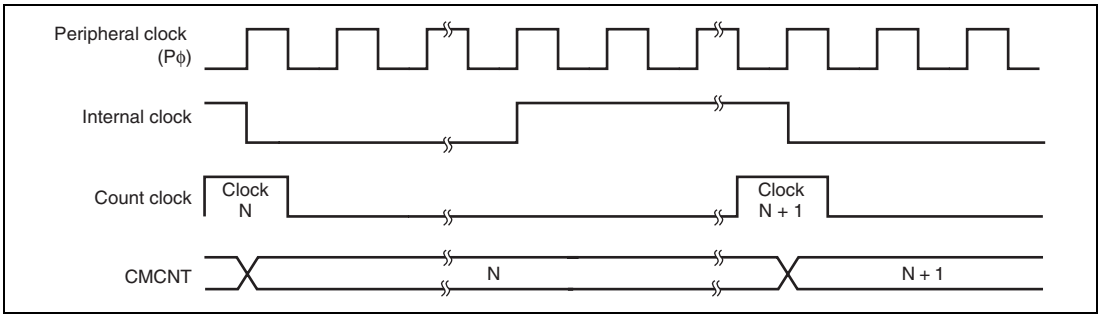


Figure 13.3 Count Timing

13.4 Interrupts

13.4.1 Interrupt Sources and DMA Transfer Requests

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt. When both the compare match flag (CMF) and the interrupt enable bit (CMIE) are set to 1, the corresponding interrupt request is output. When the interrupt is used to activate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 7, Interrupt Controller (INTC).

Clear the CMF bit to 0 by the user exception handling routine. If this operation is not carried out, another interrupt will be generated. By configuring the interrupt controller (INTC), the direct memory access controller (DMAC) can be activated when a compare match interrupt is requested. In this case, an interrupt is not issued to the CPU. If the setting to activate the DMAC has not been made, an interrupt request is sent to the CPU. The CMF bit is automatically cleared to 0 when data is transferred by the DMAC.

13.4.2 Timing of Compare Match Flag Setting

When CMCOR and CMCNT match, a compare match signal is generated at the last state in which the values match (the timing when the CMCNT value is updated to H'0000) and the CMF bit in CMCSR is set to 1. That is, after a match between CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT counter clock input. Figure 13.4 shows the timing of CMF bit setting.

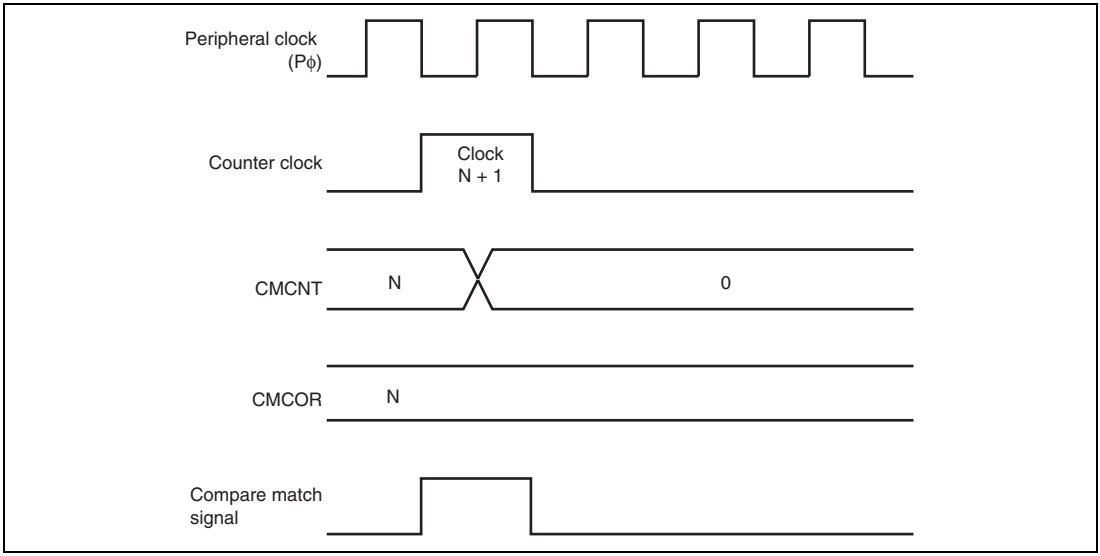


Figure 13.4 Timing of CMF Setting

13.4.3 Timing of Compare Match Flag Clearing

The CMF bit in CMCSR is cleared by first, reading as 1 then writing to 0. However, in the case of the DMAC being activated, the CMF bit is automatically cleared to 0 when data is transferred by the DMAC.

13.5 Usage Notes

13.5.1 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated in the T2 cycle while writing to CMCNT, clearing CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 13.5 shows the timing to clear the CMCNT counter.

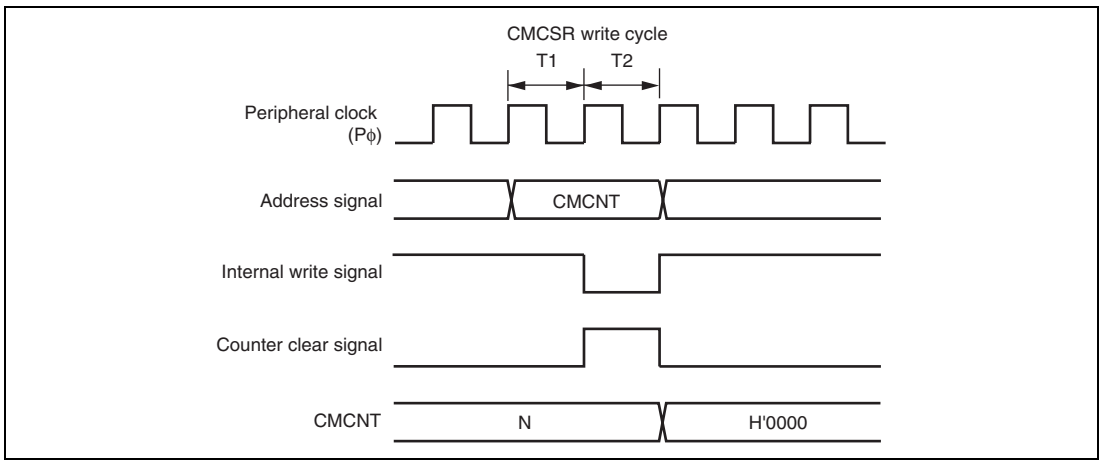


Figure 13.5 Conflict between Write and Compare Match Processes of CMCNT

13.5.2 Conflict between Word-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in words, the writing has priority over the count-up. In this case, the count-up is not performed. Figure 13.6 shows the timing to write to CMCNT in words.

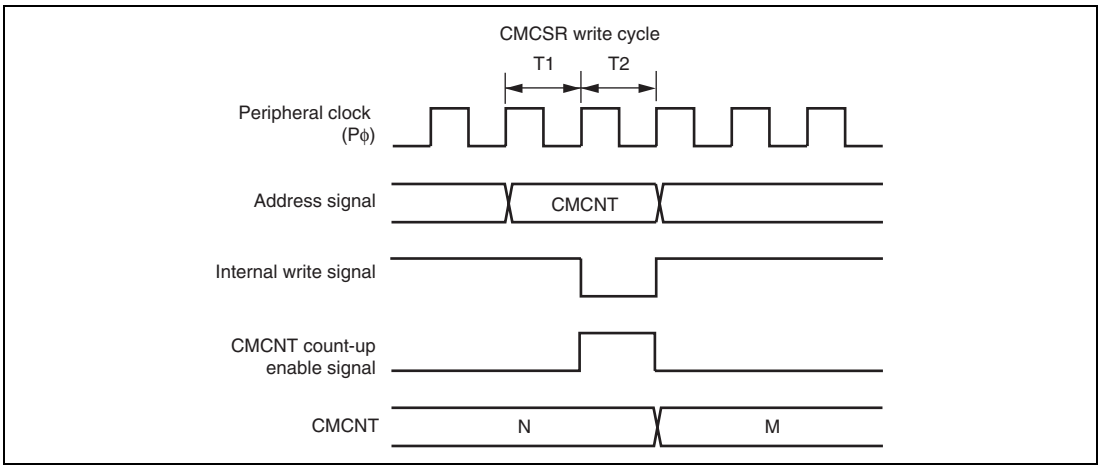


Figure 13.6 Conflict between Word-Write and Count-Up Processes of CMCNT

13.5.3 Conflict between Byte-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in bytes, the writing has priority over the count-up. In this case, the count-up is not performed. The byte data on the other side, which is not written to, is also not counted and the previous contents are retained.

Figure 13.7 shows the timing when the count-up occurs in the T2 cycle while writing to CMCNTH in bytes.

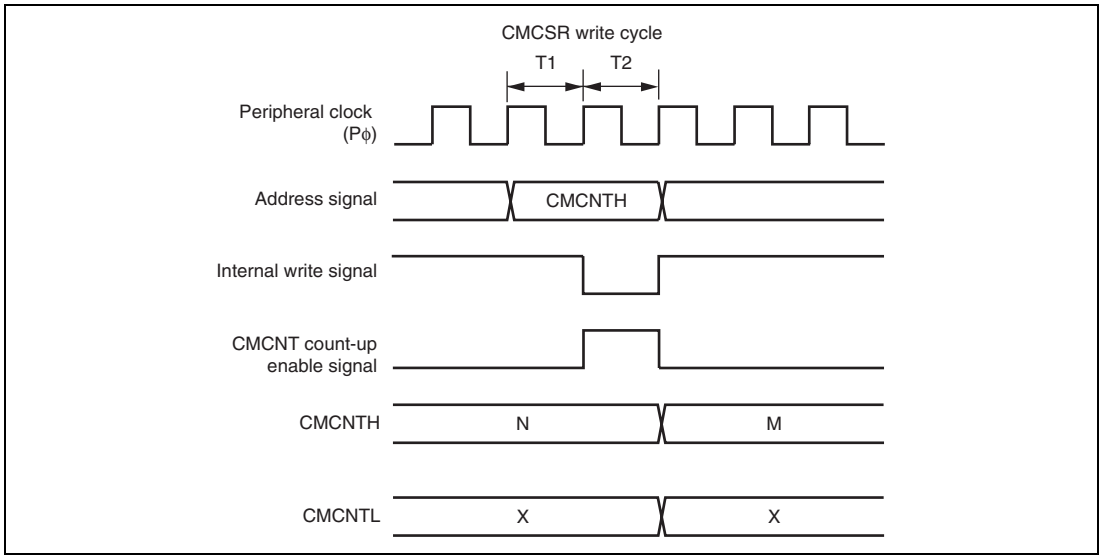


Figure 13.7 Conflict between Byte-Write and Count-Up Processes of CMCNT

13.5.4 Compare Match between CMCNT and CMCOR

Do not set the same value in CMCNT and CMCOR while CMCNT is not counting.

Section 14 Watchdog Timer (WDT)

This LSI includes the watchdog timer (WDT), which externally outputs an overflow signal ($\overline{\text{WDTOVF}}$) on overflow of the counter when the value of the counter has not been updated because of a system malfunction. The WDT can simultaneously generate an internal reset signal for the entire LSI.

The WDT is a single channel timer for each of CPU0 and CPU1, and WDT0 for CPU0 counts up the clock oscillation settling period when the system leaves software standby mode or the temporary standby periods that occur when the clock frequency is changed. Both WDT0 and WDT1 can be used as a general watchdog timer or interval timer.

14.1 Features

- Can be used to ensure the clock oscillation settling time (WDT0)
  The WDT is used in leaving software standby mode or the temporary standby periods that occur when the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode (WDT0, WDT1)
- Outputs $\overline{\text{WDTOVF}}$ signal in watchdog timer mode (WDT0, WDT1)
  When the counter overflows in watchdog timer mode, the $\overline{\text{WDTOVF}}$ signal is output externally. It is possible to select whether to reset the LSI internally when this happens. Either the power-on reset or manual reset signal can be selected as the internal reset type.
- Issues an interrupt in interval timer mode (WDT0, WDT1)
  An interval timer interrupt is issued when the counter overflows.
- Can select one of eight counter input clocks (WDT0, WDT1)
  Eight clocks (Pϕ × 1 to Pϕ × 1/16384) that are obtained by dividing the peripheral clock can be selected.

Figure 14.1 shows a block diagram of the WDT.

As shown in the figure, the reset output signals from WDT0 and WDT1 upon occurrence of an overflow are ORed and then output to both CPUs.

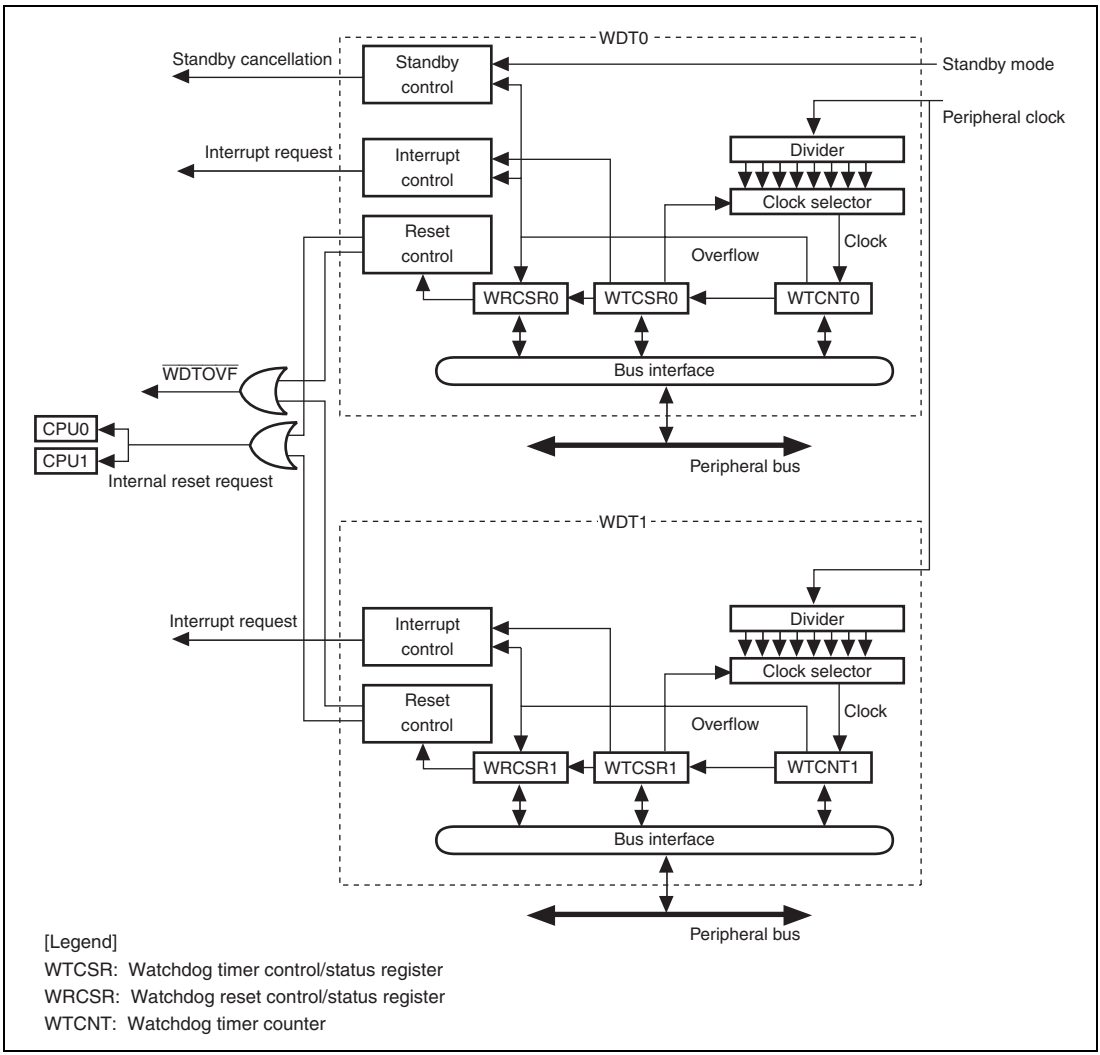


Figure 14.1 Block Diagram of WDT

14.2 Input/Output Pin

Table 14.1 shows the pin configuration of the WDT.

Table 14.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs the WDT0 or WDT1 counter overflow signal in watchdog timer mode.

14.3 Register Descriptions

The WDT has the following registers.

Table 14.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Watchdog timer counter 0	WTCNT0	R/W	H'00	H'FFFE0002	16*
Watchdog timer control/status register 0	WTCSR0	R/W	H'18	H'FFFE0000	16*
Watchdog reset control/status register 0	WRCSR0	R/W	H'1F	H'FFFE0004	16*
Watchdog timer counter 1	WTCNT1	R/W	H'00	H'FFFE000A	16*
Watchdog timer control/status register 1	WTCSR1	R/W	H'18	H'FFFE0008	16*
Watchdog reset control/status register 1	WRCSR1	R/W	H'3F	H'FFFE000C	16*

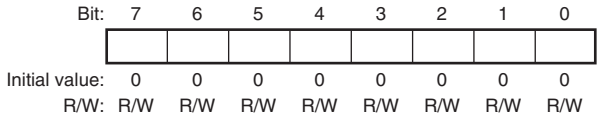
Note: * For the access size, see section 14.3.4, Notes on Register Access.

14.3.1 Watchdog Timer Counter (WTCNT0, WTCNT1)

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 watchdog timer overflow signal ($\overline{\text{WDTOVF}}$) in watchdog timer mode and an interrupt in interval timer mode.

Use word access to write to WTCNT with H'5A set in the upper byte. Use byte access to read from WTCNT.

Note: The method for writing to WTCNT differs from that for other registers to prevent erroneous writes. See section 14.3.4, Notes on Register Access, for details.



14.3.2 Watchdog Timer Control/Status Register (WTCSR0, WTCSR1)

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.

Use word access to write to WTCSR with H'A5 set in the upper byte. Use byte access to read from WTCSR.

Note: The method for writing to WTCSR differs from that for other registers to prevent erroneous writes. See section 14.3.4, Notes on Register Access, for details.

Bit:	7	6	5	4	3	2	1	0
	IOVF	WT/$\overline{IT}$	TME	-	-	CKS[2:0]		
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/(W)	R/W	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	IOVF	0	R/(W)	Interval Timer Overflow Indicates that WTCNT has overflowed in interval timer mode. This flag is not set in watchdog timer mode. 0: No overflow 1: WTCNT overflow in interval timer mode [Clearing condition] • When 0 is written to IOVF after reading IOVF
6	WT/$\overline{IT}$	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: When the WTCNT overflows in watchdog timer mode, the $\overline{WDTOVF}$ signal is output externally. If this bit is modified when the WDT is running, the up-count may not be performed correctly.

Bit	Bit Name	Initial Value	R/W	Description
5	TME	0	R/W	Timer Enable Starts and stops timer operation. Clear this bit to 0 when using the WDT in software standby mode or when changing the clock frequency. 0: Timer disabled Count-up stops and WTCNT value is retained. 1: Timer enabled
4, 3	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
2 to 0	CKS[2:0]	000	R/W	Clock Select 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 parentheses in the table is the value when the peripheral clock (Pϕ) is 33 MHz. Clock Ratio Overflow Cycle 000:1 × Pϕ (7.73 μs) 001:1/64 × Pϕ (496.5 μs) 010:1/128 × Pϕ (0.984 ms) 011:1/256 × Pϕ (1.97 ms) 100:1/512 × Pϕ (3.94 ms) 101:1/1024 × Pϕ (7.95 ms) 110:1/4096 × Pϕ (31.7 ms) 111:1/16384 × Pϕ (127.1 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.

14.3.3 Watchdog Reset Control/Status Register (WRCSR0, WRCSR1)

WRCSR is an 8-bit readable/writable register that controls output of the internal reset signal generated by watchdog timer counter (WTCNT) overflow.

Note: The method for writing to WRCSR differs from that for other registers to prevent erroneous writes. See section 14.3.4, Notes on Register Access, for details.

Bit:	7	6	5	4	3	2	1	0
	WOVF	RSTE	RSTS	-	-	-	-	-
Initial value:	0	0	*1	1	1	1	1	1
R/W:	R/(W)	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	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 [Clearing condition] • When 0 is written to WOVF after reading WOVF
6	RSTE	0	R/W	Reset Enable Selects whether to generate a signal to reset the LSI internally if WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored. 0: Not reset when WTCNT overflows*2 1: Reset when WTCNT overflows

Bit	Bit Name	Initial Value	R/W	Description
5	RSTS	*[1]	R/W	Reset Select (only valid for WDT0, reserved bit for WRCSR1) (The value of WRCSR0.RSTS determines the reset type regardless of the WDT in which an overflow occurs.) 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
4 to 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

Notes: 1. 0 for WRCSR0 and 1 for WRCSR1
2. The LSI is not reset, but WTCNT and WTCSR in WDT are reset.

14.3.4 Notes on Register Access

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

(1) 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 the WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as shown in figure 14.2. When writing to the 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 the WTCNT or WTCSR.

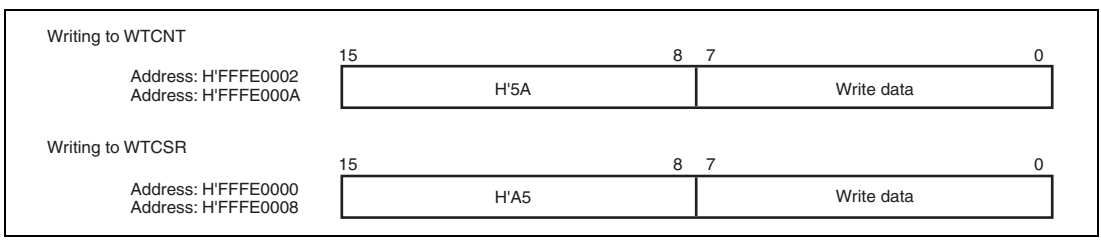


Figure 14.2 Writing to WTCNT and WTCSR

(2) Writing to WRCSR

WRCSR must be written by a word access to its address. It cannot be written by byte transfer or longword transfer instructions.

Procedures for writing 0 to WOVF (bit 7) and for writing to RSTE (bit 6) and RSTS (bit 5) are different, as shown in figure 14.3.

To write 0 to the WOVF bit, the write data must be H'A5 in the upper byte and H'00 in the lower byte. This clears the WOVF bit to 0. The RSTE and RSTS bits are not affected. To write to the RSTE and RSTS bits, the upper byte must be H'5A and the lower byte must be the write data. The values of bits 6 and 5 of the lower byte are transferred to the RSTE and RSTS bits, respectively. The WOVF bit is not affected.

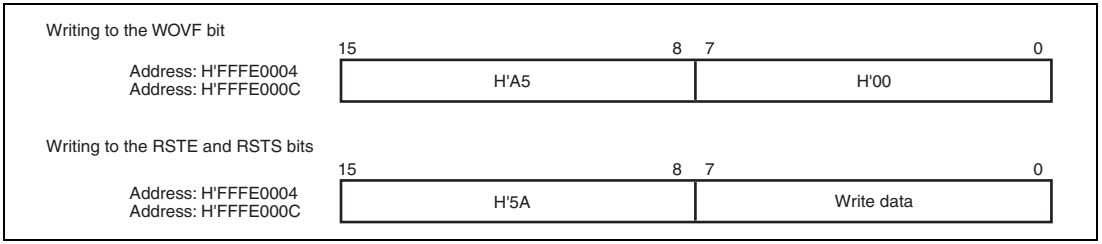


Figure 14.3 Writing to WRCSR

(3) Reading from WTCNT, WTCSR, and WRCSR

The registers of WDT0 are read in a method similar to other registers. WTCSR0 is allocated to address H'FFFE0000, WTCNT0 to address H'FFFE0002, and WRCSR0 to address H'FFFE0004. Byte transfer instructions must be used for reading from these registers.

The registers of WDT1 are read in a method similar to other registers. WTCSR1 is allocated to address H'FFFE0008, WTCNT1 to address H'FFFE000A, and WRCSR1 to address H'FFFE000C. Byte transfer instructions must be used for reading from these registers.

14.4 WDT Usage

14.4.1 Canceling Software Standby Mode

The WDT0 can be used to cancel software standby mode with an interrupt such as an NMI interrupt.

For details on the procedure, see section 30, Power-Down Modes.

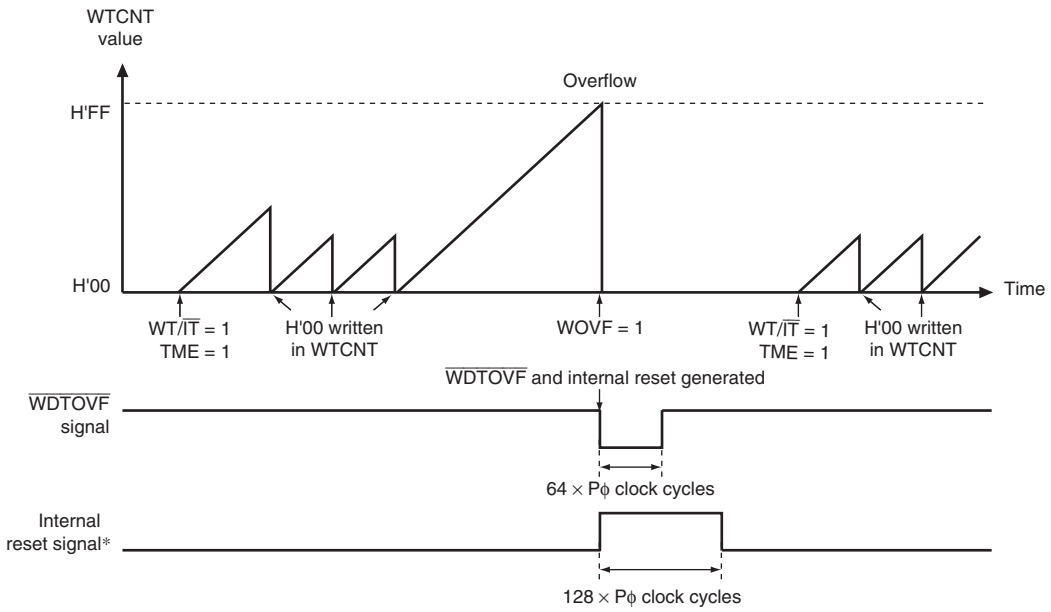
14.4.2 Changing the PLL Multiplication Ratio

When changing the clock frequency by the PLL, use the WDT0. When changing the frequency only by switching the divider, do not use the WDT. For details on the procedure, see section 5, Clock Pulse Generator (CPG).

14.4.3 Using Watchdog Timer Mode

WDT0 should be used for the watchdog of CPU0 and WDT1 for the watchdog of CPU1.

1. Set the WT/$\overline{IT}$ bit in WTCSR to 1, the type of count clock in the CKS2 to CKS0 bits in WTCSR, whether this LSI is to be reset internally or not in the RSTE bit in WRCSR, the reset type if it is generated in the RSTS bit in WRCSR0, and the initial value of the counter in 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. When the counter overflows, the WDT sets the WOVF flag in WRCSR to 1, and the $\overline{WDTOVF}$ signal is output outside the LSI (figure 14.4). The $\overline{WDTOVF}$ signal can be used to reset the system. The $\overline{WDTOVF}$ signal is output for 64 × Pϕ clock cycles.
4. If the RSTE bit in WRCSR is set to 1, a signal to reset the inside of this LSI can be generated simultaneously with the $\overline{WDTOVF}$ signal. Either power-on reset or manual reset can be selected for this interrupt by the RSTS bit in WRCSR0. The internal reset signal is output for 128 × Pϕ clock cycles.
5. When a WDT overflow reset is generated simultaneously with a reset input on the $\overline{RES}$ pin, the $\overline{RES}$ pin reset takes priority, and the WOVF bit in WRCSR is cleared to 0.



[Legend]
WT/IT: Timer mode select bit
TME: Timer enable bit

Note: * Internal reset signal occurs only when the RSTE bit is set to 1.

Figure 14.4 Operation in Watchdog Timer Mode

14.4.4 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 WTCSR to 0, set the type of count clock in the CKS[2:0] bits in WTCSR, and set the initial value of the counter in 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 bit in WTCSR to 1 and an interval timer interrupt request is sent to the INTC. The counter then resumes counting.

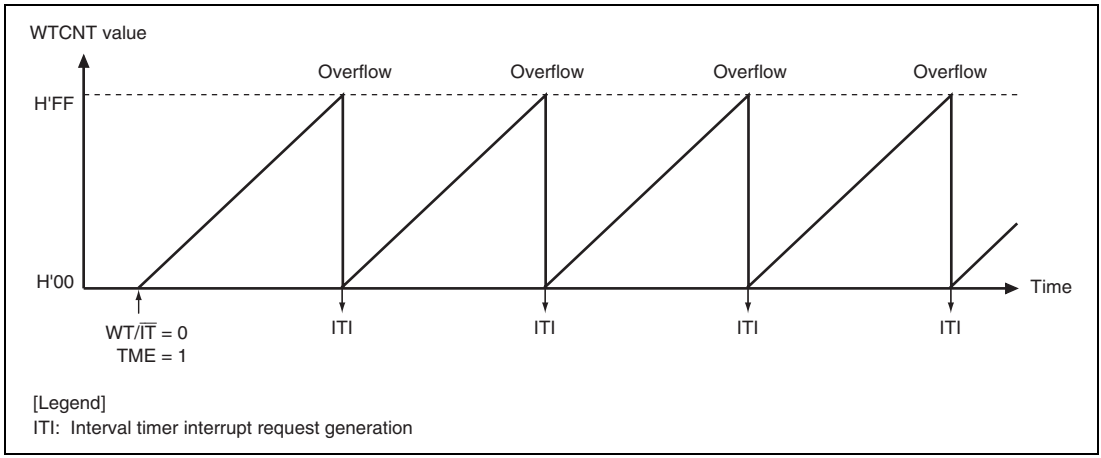


Figure 14.5 Operation in Interval Timer Mode

14.5 Usage Notes

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

14.5.1 Timer Variation

After timer operation has started, the period from the power-on reset point to the first count up timing of WTCNT varies depending on the time period that is set by the TME bit of WTCSR. 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 the CKS[2:0] bits. The timing of subsequent increment is in accordance with the selected frequency division ratio. Accordingly, this time difference is referred to as timer variation.

14.5.2 Prohibition against Setting H'FF to WTCNT

When the value in WTCNT reaches H'FF, the WDT assumes that an overflow has occurred. Accordingly, when H'FF is set in WTCNT, an interval timer interrupt or WDT reset will occur immediately, regardless of the current clock selection by the CKS[2:0] bits.

14.5.3 Interval Timer Overflow Flag

The IOVF bit in WTCSR cannot be cleared when the value in WTCNT is H'FF. Clear the IOVF bit when the value in WTCNT is set to H'00 or the value other than H'FF is rewritten.

14.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ signal is input to the $\overline{\text{RES}}$ pin of this LSI, this LSI cannot be initialized correctly.

Avoid input of the $\overline{\text{WDTOVF}}$ signal to the $\overline{\text{RES}}$ pin of this LSI through glue logic circuits. To reset the entire system with the $\overline{\text{WDTOVF}}$ signal, use the circuit shown in figure 14.6.

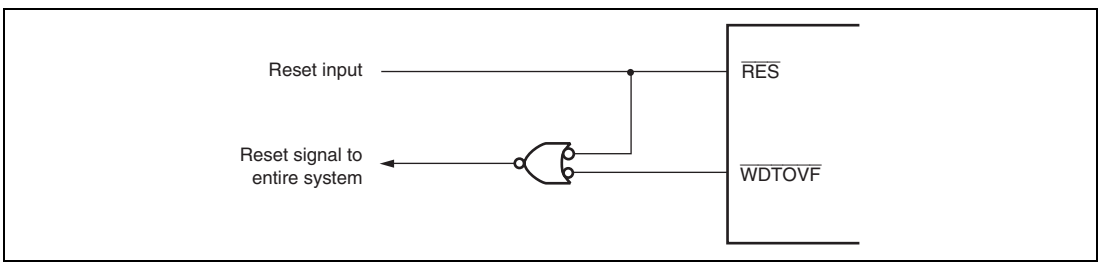


Figure 14.6 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal

14.5.5 Manual Reset in Watchdog Timer Mode

When a manual reset occurs in watchdog timer mode, the bus cycle is continued. If a manual reset occurs during burst transfer by the DMAC, manual reset exception handling will be pended until the CPU acquires the bus mastership.

14.5.6 Transition to Deep Standby Mode

The WDT does not directly perform transition to or release of deep standby mode. However, since the WDT may generate a watchdog timer reset or interval timer interrupt during transition to deep standby mode by CPU0 issuing the SLEEP instruction, clear the WTCSR0.TME and WTCSR1.TME bits to 0 to stop the WDT before issuance of the SLEEP instruction.

14.5.7 Internal Reset in Watchdog Timer Mode

When an internal reset occurs in watchdog timer mode due to an overflow of the watchdog timer counter (WTCNT0), the watchdog reset control/status register (WRCSR0) is not initialized and the WOVF bit of WDT0 stays set to 1. While the WOVF bit of WDT0 is set to 1, WDT0 cannot generate an internal reset even when an WTCNT0 overflow occurs.

In like manner, when an internal reset occurs in watchdog timer mode due to an overflow of the watchdog timer counter (WTCNT1), the watchdog reset control/status register (WRCSR1) is not initialized and the WOVF bit of WDT1 stays set to 1. While the WOVF bit of WDT1 is set to 1, WDT1 cannot generate an internal reset even when an WTCNT1 overflow occurs.

Section 15 Realtime Clock (RTC)

This LSI has a realtime clock (RTC) with its own 32.768-kHz crystal oscillator.

15.1 Features

- Clock and calendar functions (BCD format): Seconds, minutes, hours, date, day of the week, month, and year
- 1-Hz to 64-Hz timer (binary format)
  64-Hz counter indicates the state of the RTC divider circuit between 64 Hz and 1 Hz
- Start/stop function
- 30-second adjust function
- Alarm interrupt: Frame comparison of seconds, minutes, hours, date, day of the week, month, and year can be used as conditions for the alarm interrupt
- Periodic interrupts: the interrupt cycle may be 1/256 second, 1/64 second, 1/16 second, 1/4 second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: a carry interrupt indicates when a carry occurs during a counter read
- Automatic leap year adjustment

Figure 15.1 shows the block diagram of RTC.

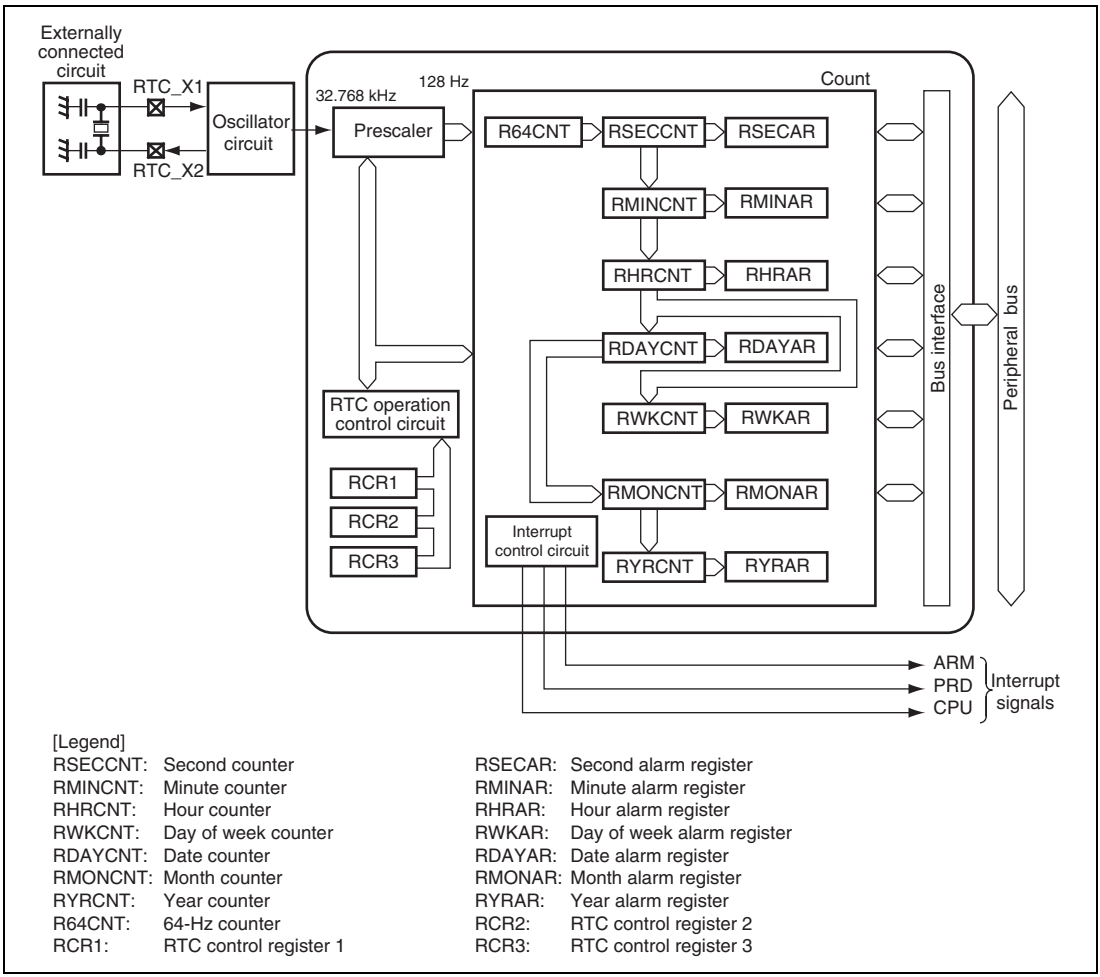


Figure 15.1 RTC Block Diagram

15.2 Input/Output Pin

Table 15.1 shows the RTC pin configuration.

Table 15.1 Pin Configuration

Pin Name	Symbol	I/O	Description
RTC oscillator crystal pin/ external clock	RTC_X1	Input	Connects 32.768-kHz crystal resonator for RTC, and enables to input the external clock to the RTC_X1 pin.
	RTC_X2	Output	

15.3 Register Descriptions

The RTC has the following registers.

Table 15.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
64-Hz counter	R64CNT	R	H'xx	H'FFFE1000	8
Second counter	RSECCNT	R/W	H'xx	H'FFFE1002	8
Minute counter	RMINCNT	R/W	H'xx	H'FFFE1004	8
Hour counter	RHRCNT	R/W	H'xx	H'FFFE1006	8
Day of week counter	RWKCNT	R/W	H'xx	H'FFFE1008	8
Date counter	RDAYCNT	R/W	H'xx	H'FFFE100A	8
Month counter	RMONCNT	R/W	H'xx	H'FFFE100C	8
Year counter	RYRCNT	R/W	H'xxxx	H'FFFE100E	16
Second alarm register	RSECAR	R/W	H'xx	H'FFFE1010	8
Minute alarm register	RMINAR	R/W	H'xx	H'FFFE1012	8
Hour alarm register	RHRAR	R/W	H'xx	H'FFFE1014	8
Day of week alarm register	RWKAR	R/W	H'xx	H'FFFE1016	8
Date alarm register	RDAYAR	R/W	H'xx	H'FFFE1018	8
Month alarm register	RMONAR	R/W	H'xx	H'FFFE101A	8
Year alarm register	RYRAR	R/W	H'xxxx	H'FFFE1020	16
RTC control register 1	RCR1	R/W	H'00	H'FFFE101C	8
RTC control register 2	RCR2	R/W	H'09	H'FFFE101E	8
RTC control register 3	RCR3	R/W	H'00	H'FFFE1024	8

15.3.1 64-Hz Counter (R64CNT)

R64CNT indicates the state of the divider circuit between 64 Hz and 1 Hz.

Reading this register, when carry from 128-Hz divider stage is generated, sets the CF bit in the RTC control register 1 (RCR1) to 1 so that the carrying and reading 64 Hz counter are performed at the same time is indicated. In this case, the R64CNT should be read again after writing 0 to the CF bit in RCR1 since the read value is not valid.

After the RESET bit or ADJ bit in the RTC control register 2 (RCR2) is set to 1, the RTC divider circuit is initialized and R64CNT is initialized.

Bit:	7	6	5	4	3	2	1	0
	-	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz
Initial value:	0	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

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	1 Hz	Undefined	R	Indicate the state of the divider circuit between 64 Hz and 1 Hz.
5	2 Hz	Undefined	R	
4	4 Hz	Undefined	R	
3	8 Hz	Undefined	R	
2	16 Hz	Undefined	R	
1	32 Hz	Undefined	R	
0	64 Hz	Undefined	R	

15.3.2 Second Counter (RSECCNT)

RSECCNT is used for setting/counting in the BCD-coded second section. The count operation is performed by a carry for each second of the 64-Hz counter.

The assignable range is from 00 through 59 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

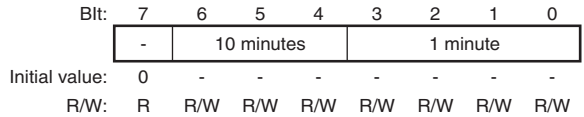
Bit:	7	6	5	4	3	2	1	0
	-	10 seconds			1 second			
Initial value:	0	-	-	-	-	-	-	-
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	10 seconds	Undefined	R/W	Counting Ten's Position of Seconds Counts on 0 to 5 for 60-seconds counting.
3 to 0	1 second	Undefined	R/W	Counting One's Position of Seconds Counts on 0 to 9 once per second. When a carry is generated, 1 is added to the ten's position.

15.3.3 Minute Counter (RMINCNT)

RMINCNT is used for setting/counting in the BCD-coded minute section. The count operation is performed by a carry for each minute of the second counter.

The assignable range is from 00 through 59 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.



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 to 4	10 minutes	Undefined	R/W	Counting Ten's Position of Minutes Counts on 0 to 5 for 60-minutes counting.
3 to 0	1 minute	Undefined	R/W	Counting One's Position of Minutes Counts on 0 to 9 once per second. When a carry is generated, 1 is added to the ten's position.

15.3.4 Hour Counter (RHRCNT)

RHRCNT is used for setting/counting in the BCD-coded hour section. The count operation is performed by a carry for each 1 hour of the minute counter.

The assignable range is from 00 through 23 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

Bit:	7	6	5	4	3	2	1	0
	-	-	10 hours		1 hour			
Initial value:	0	0	-	-	-	-	-	-
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	10 hours	Undefined	R/W	Counting Ten's Position of Hours Counts on 0 to 2 for ten's position of hours.
3 to 0	1 hour	Undefined	R/W	Counting One's Position of Hours Counts on 0 to 9 once per hour. When a carry is generated, 1 is added to the ten's position.

15.3.5 Day of Week Counter (RWKCNT)

RWKCNT is used for setting/counting day of week section. The count operation is performed by a carry for each day of the date counter.

The assignable range is from 0 through 6 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	Day		
Initial value:	0	0	0	0	0	-	-	-
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	Day	Undefined	R/W	Day-of-Week Counting Day-of-week is indicated with a binary code. 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Reserved (setting prohibited)

15.3.6 Date Counter (RDAYCNT)

RDAYCNT is used for setting/counting in the BCD-coded date section. The count operation is performed by a carry for each day of the hour counter.

The assignable range is from 01 through 31 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

The range of date changes with each month and in leap years. Confirm the correct setting. Leap years are recognized by dividing the year counter (RYRCNT) values by 400, 100, and 4 and obtaining a fractional result of 0.

BIt:	7	6	5	4	3	2	1	0
	-	-	10 days		1 day			
Initial value:	0	0	-	-	-	-	-	-
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	10 days	Undefined	R/W	Counting Ten's Position of Dates
3 to 0	1 day	Undefined	R/W	Counting One's Position of Dates Counts on 0 to 9 once per date. When a carry is generated, 1 is added to the ten's position.

15.3.7 Month Counter (RMONCNT)

RMONCNT is used for setting/counting in the BCD-coded month section. The count operation is performed by a carry for each month of the date counter.

The assignable range is from 01 through 12 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

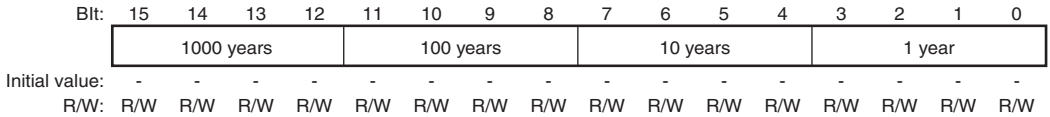
Bit:	7	6	5	4	3	2	1	0
	-	-	-	10 months	1 month			
Initial value:	0	0	0	-	-	-	-	-
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	10 months	Undefined	R/W	Counting Ten's Position of Months
3 to 0	1 month	Undefined	R/W	Counting One's Position of Months Counts on 0 to 9 once per month. When a carry is generated, 1 is added to the ten's position.

15.3.8 Year Counter (RYRCNT)

RYRCNT is used for setting/counting in the BCD-coded year section. The count operation is performed by a carry for each year of the month counter.

The assignable range is from 0000 through 9999 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.



Bit	Bit Name	Initial Value	R/W	Description
15 to 12	1000 years	Undefined	R/W	Counting Thousand's Position of Years
11 to 8	100 years	Undefined	R/W	Counting Hundred's Position of Years
7 to 4	10 years	Undefined	R/W	Counting Ten's Position of Years
3 to 0	1 year	Undefined	R/W	Counting One's Position of Years

15.3.9 Second Alarm Register (RSECAR)

RSECAR is an alarm register corresponding to the BCD coded second counter RSECCNT of the RTC. When the ENB bit is set to 1, a comparison with the RSECCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 00 through 59 + ENB bits (practically in BCD), otherwise operation errors occur.

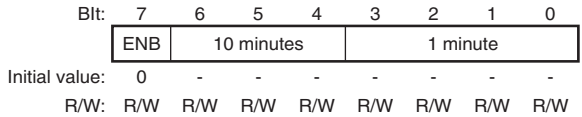
Bit:	7	6	5	4	3	2	1	0
	ENB	10 seconds			1 second			
Initial value:	0	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RSECCNT value is performed.
6 to 4	10 seconds	Undefined	R/W	Ten's position of seconds setting value
3 to 0	1 second	Undefined	R/W	One's position of seconds setting value

15.3.10 Minute Alarm Register (RMINAR)

RMINAR is an alarm register corresponding to the minute counter RMINCNT. When the ENB bit is set to 1, a comparison with the RMINCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 00 through 59 + ENB bits (practically in BCD), otherwise operation errors occur.



Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RMINCNT value is performed.
6 to 4	10 minutes	Undefined	R/W	Ten's position of minutes setting value
3 to 0	1 minute	Undefined	R/W	One's position of minutes setting value

15.3.11 Hour Alarm Register (RHRAR)

RHRAR is an alarm register corresponding to the BCD coded hour counter RHRCNT of the RTC. When the ENB bit is set to 1, a comparison with the RHRCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 00 through 23 + ENB bits (practically in BCD), otherwise operation errors occur.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	10 hours		1 hour			
Initial value:	0	0	-	-	-	-	-	-
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RHRCNT value is performed.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	10 hours	Undefined	R/W	Ten's position of hours setting value
3 to 0	1 hour	Undefined	R/W	One's position of hours setting value

15.3.12 Day of Week Alarm Register (RWKAR)

RWKAR is an alarm register corresponding to the BCD coded day of week counter RWKCNT. When the ENB bit is set to 1, a comparison with the RWKCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 0 through 6 + ENB bits (practically in BCD), otherwise operation errors occur.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	-	-	-	Day		
Initial value:	0	0	0	0	0	-	-	-
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RWKCNT value is performed.
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	Day	Undefined	R/W	Day of Week Setting Value 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Reserved (setting prohibited)

15.3.13 Date Alarm Register (RDAYAR)

RDAYAR is an alarm register corresponding to the BCD coded date counter RDAYCNT. When the ENB bit is set to 1, a comparison with the RDAYCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 01 through 31 + ENB bits (practically in BCD), otherwise operation errors occur.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	10 days		1 day			
Initial value:	0	0	-	-	-	-	-	-
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RDAYCNT value is performed.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	10 days	Undefined	R/W	Ten's position of dates setting value
3 to 0	1 day	Undefined	R/W	One's position of dates setting value

15.3.14 Month Alarm Register (RMONAR)

RMONAR is an alarm register corresponding to the BCD coded month counter RMONCNT. When the ENB bit is set to 1, a comparison with the RMONCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

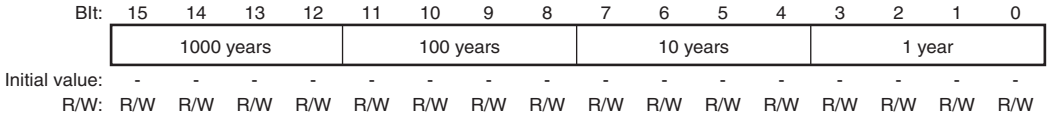
The assignable range is from 01 through 12 + ENB bits (practically in BCD), otherwise operation errors occur.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	-	10 months	1 month			
Initial value:	0	0	0	-	-	-	-	-
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RMONCNT value is performed.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	10 months	Undefined	R/W	Ten's position of months setting value
3 to 0	1 month	Undefined	R/W	One's position of months setting value

15.3.15 Year Alarm Register (RYRAR)

RYRAR is an alarm register corresponding to the year counter RYRCNT. The assignable range is from 0000 through 9999 (practically in BCD), otherwise operation errors occur.



Bit	Bit Name	Initial Value	R/W	Description
15 to 12	1000 years	Undefined	R/W	Thousand's position of years setting value
11 to 8	100 years	Undefined	R/W	Hundred's position of years setting value
7 to 4	10 years	Undefined	R/W	Ten's position of years setting value
3 to 0	1 year	Undefined	R/W	One's position of years setting value

15.3.16 RTC Control Register 1 (RCR1)

RCR1 is a register that affects carry flags and alarm flags. It also selects whether to generate interrupts for each flag.

The CF flag remains undefined until the divider circuit is reset (the RESET and ADJ bits in RCR2 are set to 1). When using the CF flag, make sure to reset the divider circuit beforehand.

Bit:	7	6	5	4	3	2	1	0
	CF	-	-	CIE	AIE	-	-	AF
Initial value:	-	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CF	Undefined	R/W	Carry Flag Status flag that indicates that a carry has occurred. CF is set to 1 when a count-up to 64-Hz occurs at the second counter carry or 64-Hz counter read. A count register value read at this time cannot be guaranteed; another read is required. 0: No carry of 64-Hz counter by second counter or 64-Hz counter [Clearing condition] When 0 is written to CF 1: Carry of 64-Hz counter by second counter or 64 Hz counter [Setting condition] When the second counter or 64-Hz counter is read during a carry occurrence by the 64-Hz counter, or 1 is written to CF.
6, 5	—	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
4	CIE	0	R/W	Carry Interrupt Enable Flag When the carry flag (CF) is set to 1, the CIE bit enables interrupts. 0: A carry interrupt is not generated when the CF flag is set to 1 1: A carry interrupt is generated when the CF flag is set to 1
3	AIE	0	R/W	Alarm Interrupt Enable Flag When the alarm flag (AF) is set to 1, the AIE bit allows interrupts. 0: An alarm interrupt is not generated when the AF flag is set to 1 1: An alarm interrupt is generated when the AF flag is set to 1
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	AF	0	R/W	Alarm Flag The AF flag is set when the alarm time, which is set by an alarm register (ENB bit in RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, or RYRAR is set to 1), and counter match. 0: Alarm register and counter not match [Clearing condition] When 0 is written to AF. 1: Alarm register and counter match* [Setting condition] When alarm register (only a register with ENB bit set to 1) and counter match Note: * Writing 1 holds previous value.

15.3.17 RTC Control Register 2 (RCR2)

RCR2 is a register for periodic interrupt control, 30-second adjustment ADJ, divider circuit RESET, and RTC count control.

RCR2 is initialized by a power-on reset or in deep standby mode. Bits other than the RTCEN and START bits are initialized by a manual reset.

Bit:	7	6	5	4	3	2	1	0
	PEF	PES[2:0]			RTCEN	ADJ	RESET	START
Initial value:	0	0	0	0	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PEF	0	R/W	Periodic Interrupt Flag Indicates interrupt generation with the period designated by the PES2 to PES0 bits. When set to 1, PEF generates periodic interrupts. 0: Interrupts not generated with the period designated by the bits PES2 to PES0. [Clearing condition] When 0 is written to PEF 1: Interrupts generated with the period designated by the PES2 to PES0 bits. [Setting condition] When an interrupt is generated with the period designated by the bits PES0 to PES2 or when 1 is written to the PEF flag
6 to 4	PES[2:0]	000	R/W	Interrupt Enable Flags These bits specify the periodic interrupt. 000: No periodic interrupts generated 001: Periodic interrupt generated every 1/256 second 010: Periodic interrupt generated every 1/64 second 011: Periodic interrupt generated every 1/16 second 100: Periodic interrupt generated every 1/4 second 101: Periodic interrupt generated every 1/2 second 110: Periodic interrupt generated every 1 second 111: Periodic interrupt generated every 2 seconds

Bit	Bit Name	Initial Value	R/W	Description
3	RTCEN	1	R/W	RTC_X1 Clock Control Controls the function of RTC_X1 pin. 0: Halts the on-chip crystal oscillator/disables the external clock input. 1: Runs the on-chip crystal oscillator/enables the external clock input.
2	ADJ	0	R/W	30-Second Adjustment When 1 is written to the ADJ bit, times of 29 seconds or less will be rounded to 00 seconds and 30 seconds or more to 1 minute. The divider circuit (RTC prescaler and R64CNT) will be simultaneously reset. This bit always reads 0. 0: Runs normally. 1: 30-second adjustment.
1	RESET	0	R/W	Reset Writing 1 to this bit initializes the divider circuit. In this case, the RESET bit is automatically reset to 0 after 1 is written to and the divider circuit (RTC prescaler and R64CNT) is reset. Thus, there is no need to write 1 to this bit. This bit is always read as 0. 0: Runs normally. 1: Divider circuit is reset.
0	START	1	R/W	Start Halts and restarts the counter (clock). 0: Second/minute/hour/day/week/month/year counter halts. 1: Second/minute/hour/day/week/month/year counter runs normally.

15.3.18 RTC Control Register 3 (RCR3)

When the ENB bit is set to 1, RCR3 performs a comparison with the RYRCNT. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, comparison of the year alarm register (RYRAR) and the year counter (RYRCNT) is performed.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

15.4 Operation

RTC usage is shown below.

15.4.1 Initial Settings of Registers after Power-On

All the registers should be set after the power is turned on.

15.4.2 Setting Time

Figure 15.2 shows how to set the time when the clock is stopped.

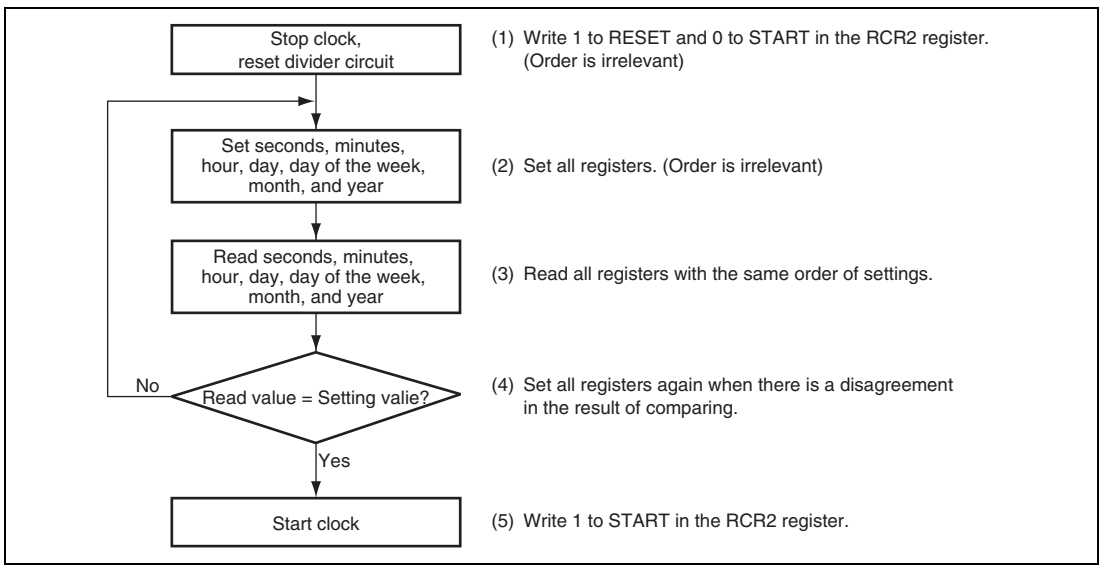


Figure 15.2 Setting Time

15.4.3 Reading Time

Figure 15.3 shows how to read the time.

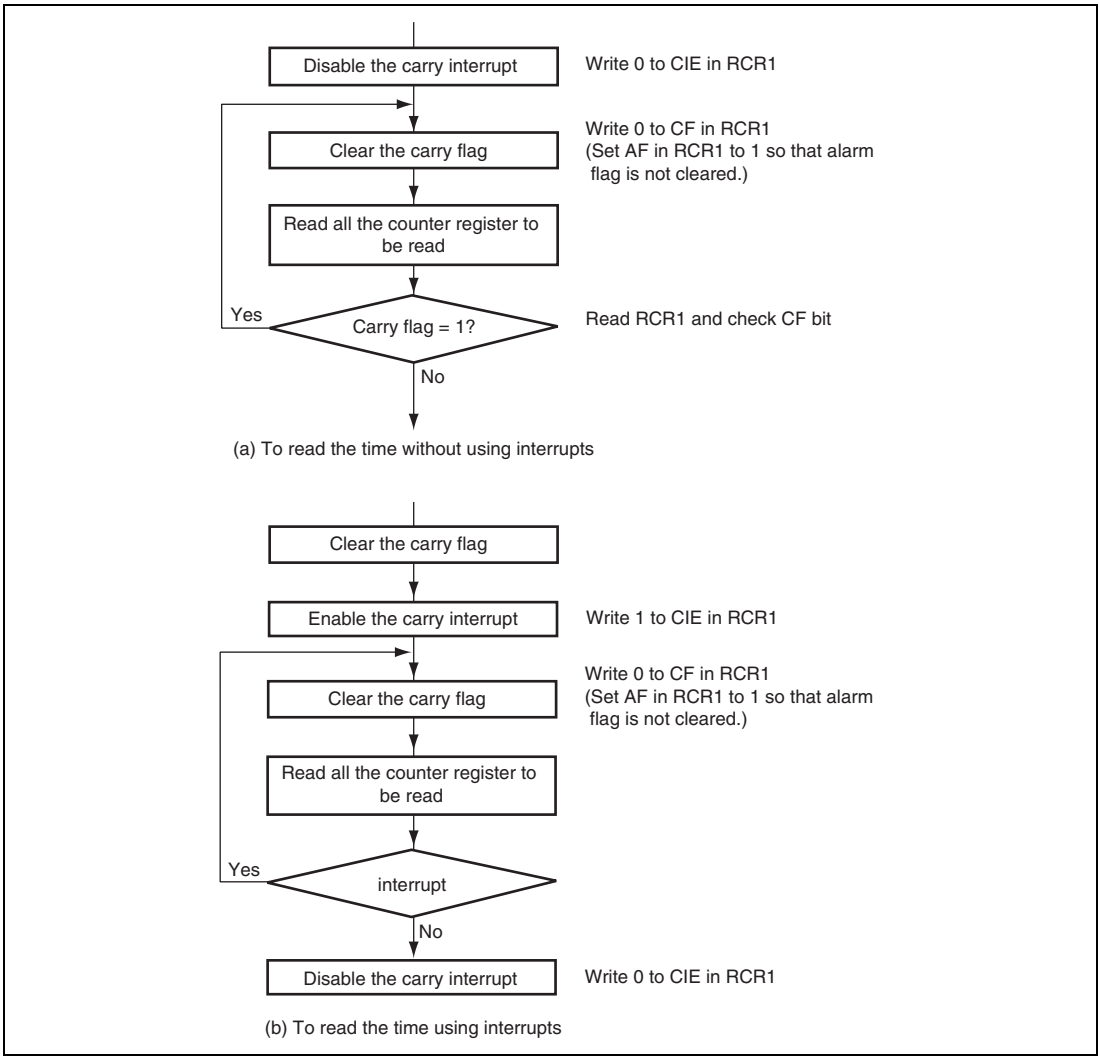


Figure 15.3 Reading Time

If a carry occurs while reading the time, the correct time will not be obtained, so it must be read again. Part (a) in figure 15.3 shows the method of reading the time without using interrupts; part (b) in figure 15.3 shows the method using carry interrupts. To keep programming simple, method (a) should normally be used.

15.4.4 Alarm Function

Figure 15.4 shows how to use the alarm function.

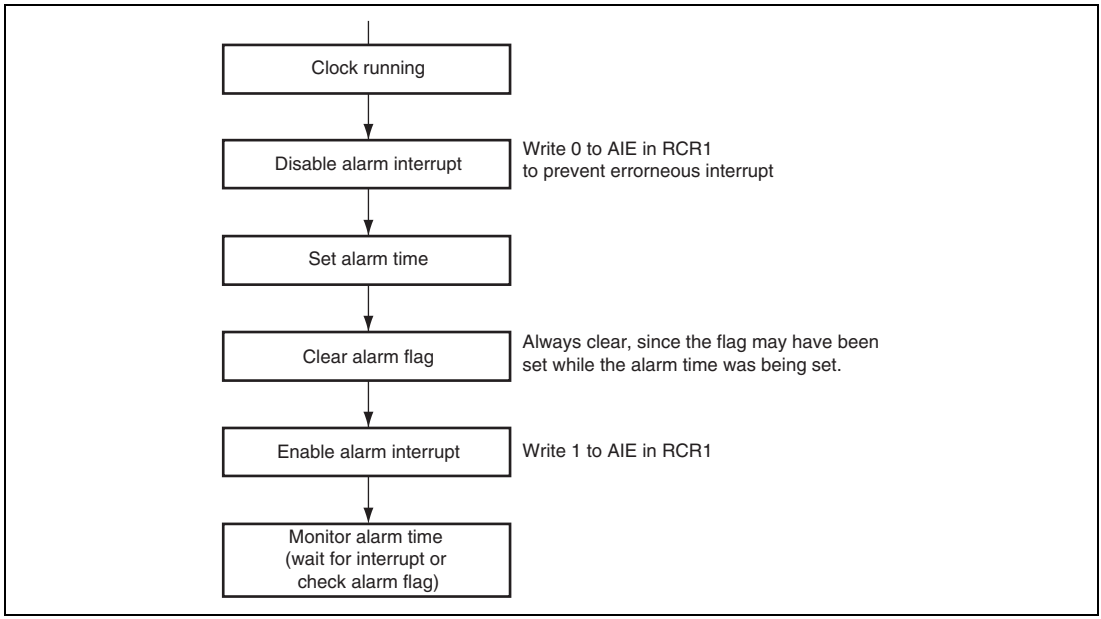


Figure 15.4 Using Alarm Function

Alarms can be generated using seconds, minutes, hours, day of the week, date, month, year, or any combination of these. Set the ENB bit in the register on which the alarm is placed to 1, and then set the alarm time in the lower bits. Clear the ENB bit in the register on which the alarm is not placed to 0.

When the clock and alarm times match, 1 is set in the AF bit in RCR1. Alarm detection can be checked by reading this bit, but normally it is done by interrupt. If 1 is set in the AIE bit in RCR1, an interrupt is generated when an alarm occurs.

The alarm flag is set when the clock and alarm times match. However, the alarm flag can be cleared by writing 0.

15.5 Usage Notes

15.5.1 Register Writing during RTC Count

The following RTC registers cannot be written to during an RTC count (while bit 0 = 1 in RCR2).

RSECCNT, RMINCNT, RHRCNT, RDAYCNT, RWKCNT, RMONCNT, RYRCNT

The RTC count must be stopped before writing to any of the above registers.

15.5.2 Use of Real-time Clock (RTC) Periodic Interrupts

The method of using the periodic interrupt function is shown in figure 15.5.

A periodic interrupt can be generated periodically at the interval set by bits PES2 to PES0 in RCR2. When the time set by bits PES2 to PES0 has elapsed, the PEF is set to 1.

The PEF is cleared to 0 upon periodic interrupt generation or when bits PES2 to PES0 are set. Periodic interrupt generation can be confirmed by reading this bit, but normally the interrupt function is used.

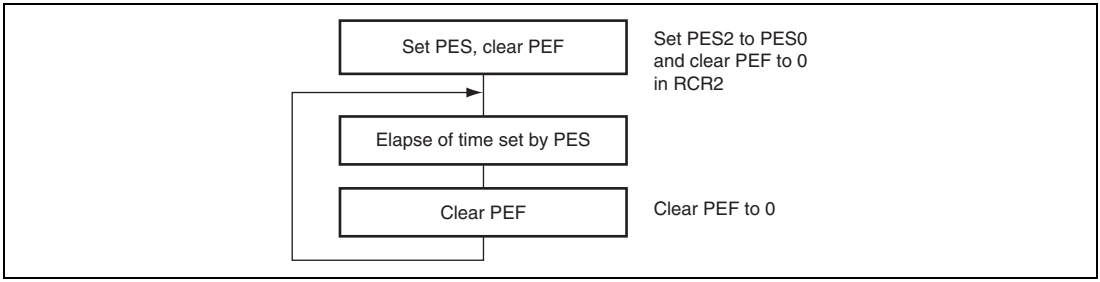


Figure 15.5 Using Periodic Interrupt Function

15.5.3 Transition to Standby Mode after Setting Register

When a transition to standby mode is made after registers in the RTC are set, sometimes counting is not performed correctly. In case the registers are set, be sure to make a transition to standby mode after performing one dummy read the register.

15.5.4 Usage Notes when Writing to and Reading the Register

- When reading a counter register such as the second counter after having written to the register, follow the procedure in section 15.4.2, Setting Time. In this case, it is necessary to write to all the counters, from second to year, in succession. Do not read the counter registers during the write processing shown as (2) in figure 15.2.
- When reading the RCR2 register after having written to it, read the register after having dummy-read it twice. The value read in both dummy-read operations will be the value before writing. The written value will be reflected in the value read the third time.
- For other registers, written values are immediately reflected in read values.

Section 16 Serial Communication Interface with FIFO (SCIF)

This LSI has a six-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.

16.1 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 serial port register when a framing error occurs.
- Clock synchronous serial communication (only channel 0, 1, 2, and 5):
  - Serial data communication is synchronized with a clock signal. The SCIF can communicate with other chips having a clock 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 interrupt, break interrupt, receive-FIFO-data-full interrupt, and receive-error interrupts are requested independently.

- When the SCIF is not in use, it can be stopped by halting the clock supplied to it, saving power.
- In asynchronous mode, on-chip modem control functions ($\overline{\text{RTS}}$ and $\overline{\text{CTS}}$) (only channel 0).
- The quantity of data in the transmit and receive FIFO data registers and the number of receive errors of the receive data in the receive FIFO data register can be ascertained.
- A time-out error (DR) can be detected when receiving in asynchronous mode.
- In asynchronous mode, the base clock frequency can be either 16 or 8 times the bit rate.
- When an internal clock is selected as a clock source and the SCK pin is used as an input pin in asynchronous mode, either normal mode or double-speed mode can be selected for the baud rate generator.

Figure 16.1 shows a block diagram of the SCIF (for a single channel). Some channels have no external pins.

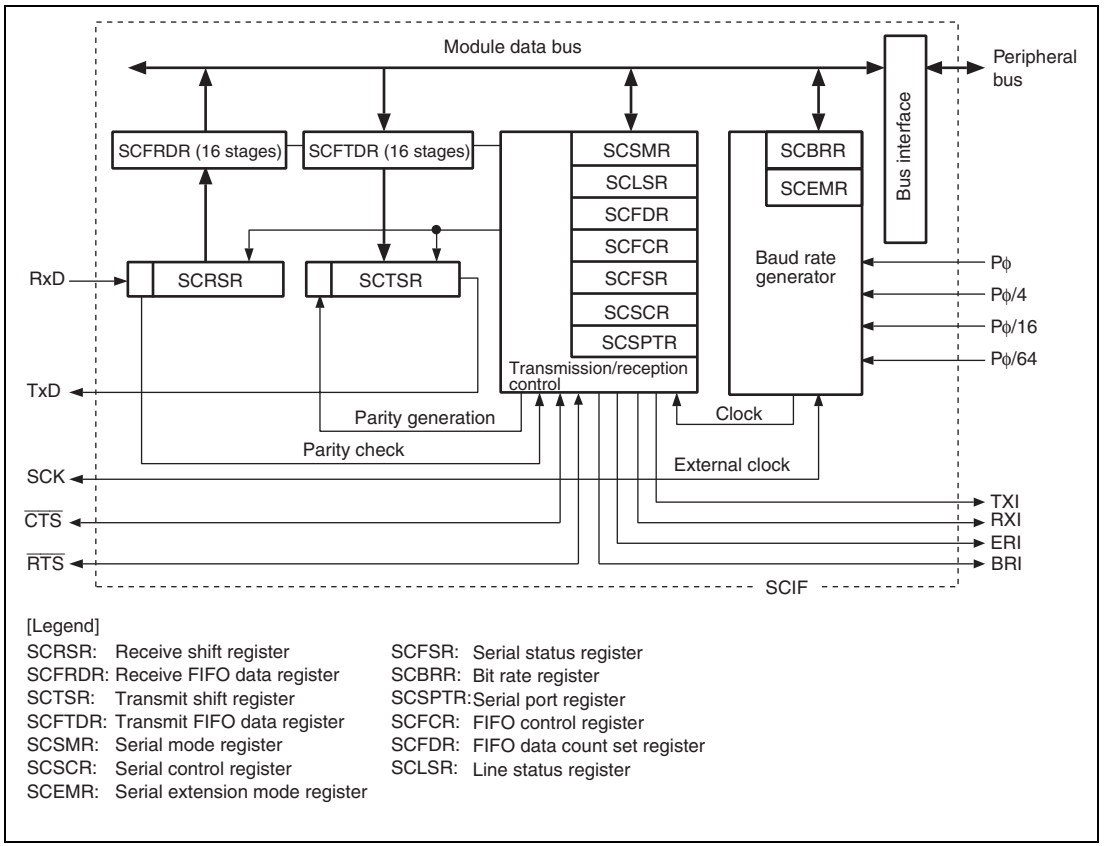


Figure 16.1 Block Diagram of SCIF

16.2 Input/Output Pins

Table 16.1 shows the pin configuration of the SCIF.

Table 16.1 Pin Configuration

Channel	Pin Name	Symbol	I/O	Function
0 to 5	Receive data pins	RxD0 to RxD5	Input	Receive data input
	Transmit data pins	TxD0 to TxD5	Output	Transmit data output
0, 1, 2, 5	Serial clock pins	SCK0, SCK1, SCK2, SCK5	I/O	Clock I/O
0	Request to send pin	$\overline{\text{RTS0}}$	I/O	Request to send
	Clear to send pin	$\overline{\text{CTS0}}$	I/O	Clear to send

16.3 Register Descriptions

The SCIF has the following registers.

Table 16.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Serial mode register_0	SCSMR_0	R/W	H'0000	H'FFFE8000	16
	Bit rate register_0	SCBRR_0	R/W	H'FF	H'FFFE8004	8
	Serial control register_0	SCSCR_0	R/W	H'0000	H'FFFE8008	16
	Transmit FIFO data register_0	SCFTDR_0	W	Undefined	H'FFFE800C	8
	Serial status register_0	SCFSR_0	R/(W)*[1]	H'0060	H'FFFE8010	16
	Receive FIFO data register_0	SCFRDR_0	R	Undefined	H'FFFE8014	8
	FIFO control register_0	SCFCR_0	R/W	H'0000	H'FFFE8018	16
	FIFO data count register_0	SCFDR_0	R	H'0000	H'FFFE801C	16
	Serial port register_0	SCSPTR_0	R/W	H'0050	H'FFFE8020	16
	Line status register_0	SCLSR_0	R/(W)*[2]	H'0000	H'FFFE8024	16
	Serial extension mode register_0	SCEMR_0	R/W	H'0000	H'FFFE8028	16
1	Serial mode register_1	SCSMR_1	R/W	H'0000	H'FFFE8800	16
	Bit rate register_1	SCBRR_1	R/W	H'FF	H'FFFE8804	8
	Serial control register_1	SCSCR_1	R/W	H'0000	H'FFFE8808	16
	Transmit FIFO data register_1	SCFTDR_1	W	Undefined	H'FFFE880C	8
	Serial status register_1	SCFSR_1	R/(W)*[1]	H'0060	H'FFFE8810	16
	Receive FIFO data register_1	SCFRDR_1	R	Undefined	H'FFFE8814	8
	FIFO control register_1	SCFCR_1	R/W	H'0000	H'FFFE8818	16
	FIFO data count register_1	SCFDR_1	R	H'0000	H'FFFE881C	16
	Serial port register_1	SCSPTR_1	R/W	H'0050	H'FFFE8820	16
	Line status register_1	SCLSR_1	R/(W)*[2]	H'0000	H'FFFE8824	16
	Serial extension mode register_1	SCEMR_1	R/W	H'0000	H'FFFE8828	16

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2	Serial mode register_2	SCSMR_2	R/W	H'0000	H'FFFE9000	16
	Bit rate register_2	SCBRR_2	R/W	H'FF	H'FFFE9004	8
	Serial control register_2	SCSCR_2	R/W	H'0000	H'FFFE9008	16
	Transmit FIFO data register_2	SCFTDR_2	W	Undefined	H'FFFE900C	8
	Serial status register_2	SCFSR_2	R/(W)*[1]	H'0060	H'FFFE9010	16
	Receive FIFO data register_2	SCFRDR_2	R	Undefined	H'FFFE9014	8
	FIFO control register_2	SCFCR_2	R/W	H'0000	H'FFFE9018	16
	FIFO data count register_2	SCFDR_2	R	H'0000	H'FFFE901C	16
	Serial port register_2	SCSPTR_2	R/W	H'0050	H'FFFE9020	16
	Line status register_2	SCLSR_2	R/(W)*[2]	H'0000	H'FFFE9024	16
	Serial extension mode register_2	SCEMR_2	R/W	H'0000	H'FFFE9028	16
3	Serial mode register_3	SCSMR_3	R/W	H'0000	H'FFFE9800	16
	Bit rate register_3	SCBRR_3	R/W	H'FF	H'FFFE9804	8
	Serial control register_3	SCSCR_3	R/W	H'0000	H'FFFE9808	16
	Transmit FIFO data register_3	SCFTDR_3	W	Undefined	H'FFFE980C	8
	Serial status register_3	SCFSR_3	R/(W)*[1]	H'0060	H'FFFE9810	16
	Receive FIFO data register_3	SCFRDR_3	R	Undefined	H'FFFE9814	8
	FIFO control register_3	SCFCR_3	R/W	H'0000	H'FFFE9818	16
	FIFO data count register_3	SCFDR_3	R	H'0000	H'FFFE981C	16
	Serial port register_3	SCSPTR_3	R/W	H'0050	H'FFFE9820	16
	Line status register_3	SCLSR_3	R/(W)*[2]	H'0000	H'FFFE9824	16
	Serial extension mode register_3	SCEMR_3	R/W	H'0000	H'FFFE9828	16

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
4	Serial mode register_4	SCSMR_4	R/W	H'0000	H'FFFEA000	16
	Bit rate register_4	SCBRR_4	R/W	H'FF	H'FFFEA004	8
	Serial control register_4	SCSCR_4	R/W	H'0000	H'FFFEA008	16
	Transmit FIFO data register_4	SCFTDR_4	W	Undefined	H'FFFEA00C	8
	Serial status register_4	SCFSR_4	R/(W)*[1]	H'0060	H'FFFEA010	16
	Receive FIFO data register_4	SCFRDR_4	R	Undefined	H'FFFEA014	8
	FIFO control register_4	SCFCR_4	R/W	H'0000	H'FFFEA018	16
	FIFO data count register_4	SCFDR_4	R	H'0000	H'FFFEA01C	16
	Serial port register_4	SCSPTR_4	R/W	H'0050	H'FFFEA020	16
	Line status register_4	SCLSR_4	R/(W)*[2]	H'0000	H'FFFEA024	16
	Serial extension mode register_4	SCEMR_4	R/W	H'0000	H'FFFEA028	16
5	Serial mode register_5	SCSMR_5	R/W	H'0000	H'FFFEA800	16
	Bit rate register_5	SCBRR_5	R/W	H'FF	H'FFFEA804	8
	Serial control register_5	SCSCR_5	R/W	H'0000	H'FFFEA808	16
	Transmit FIFO data register_5	SCFTDR_5	W	Undefined	H'FFFEA80C	8
	Serial status register_5	SCFSR_5	R/(W)*[1]	H'0060	H'FFFEA810	16
	Receive FIFO data register_5	SCFRDR_5	R	Undefined	H'FFFEA814	8
	FIFO control register_5	SCFCR_5	R/W	H'0000	H'FFFEA818	16
	FIFO data count register_5	SCFDR_5	R	H'0000	H'FFFEA81C	16
	Serial port register_5	SCSPTR_5	R/W	H'0050	H'FFFEA820	16
	Line status register_5	SCLSR_5	R/(W)*[2]	H'0000	H'FFFEA824	16
	Serial extension mode register_5	SCEMR_5	R/W	H'0000	H'FFFEA828	16

Notes: 1. Only 0 can be written to clear the flag. Bits 15 to 8, 3, and 2 are read-only bits that cannot be modified.

2. Only 0 can be written to clear the flag. Bits 15 to 1 are read-only bits that cannot be modified.

16.3.1 Receive Shift Register (SCRSR)

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 the receive FIFO data register (SCFRDR).

The CPU cannot read or write to SCRSR directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

16.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a 16-byte FIFO register that 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 SCFRDR is full of receive data, subsequent serial data is lost.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

16.3.3 Transmit Shift Register (SCTSR)

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 from or write to SCTSR directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

16.3.4 Transmit FIFO Data Register (SCFTDR)

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. The CPU can write to SCFTDR at all times.

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.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

16.3.5 Serial Mode Register (SCSMR)

SCSMR specifies the SCIF serial communication format and selects the clock source for the baud rate generator.

The CPU can always read from and write to SCSMR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

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 clock synchronous mode. On channels 3 and 4, this bit is reserved and always read as 0. The write value should always be 0. 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data length in asynchronous mode. In the clock synchronous mode, the data length is always 8 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.

Bit	Bit Name	Initial Value	R/W	Description
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 clock 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.
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 clock 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.

Bit	Bit Name	Initial Value	R/W	Description
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 clock 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.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	CKS[1:0]	00	R/W	Clock Select Select the internal clock source of the on-chip baud rate generator. For further information on the clock source, bit rate register settings, and baud rate, see section 16.3.8, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

16.3.6 Serial Control Register (SCSCR)

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.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TIE	RIE	TE	RE	REIE	-	CKE[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

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 FIFO 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 FIFO data full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are disabled 1: Receive FIFO 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 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 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 is detected in clock synchronous mode. Select the receive format in SCSMR and SCFCR and reset the receive FIFO before setting RE to 1.
3	REIE	0	R/W	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.

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, 0	CKE[1:0]	00	R/W	Clock Enable Select the SCIF clock source and enable or disable clock output from the SCK pin. Depending on CKE[1:0], the SCK pin can be used for serial clock output or serial clock input. If serial clock output is set in clock synchronous mode, set the C/$\overline{A}$ bit in SCSMR to 1, and then set CKE[1:0]. On channels 3 and 4, these bits are reserved and always read as 0. The write value should always be 0. • 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 either 16 or 8 times the bit rate) 10: External clock, SCK pin used for clock input (The input clock frequency is either 16 or 8 times the bit rate) 11: Setting prohibited • Clock 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

16.3.7 Serial Status Register (SCFSR)

SCFSR is a 16-bit register. The upper 8 bits indicate the number of receive errors in the receive FIFO data register, 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). The PER flag (bits 15 to 12 and bit 2) and the FER flag (bits 11 to 8 and bit 3) are read-only bits that cannot be written.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PER[3:0]				FER[3:0]				ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PER[3:0]	0000	R	Number of Parity Errors 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 after the ER bit in SCFSR is set, represents the number of parity errors in SCFRDR. When parity errors have occurred in all 16-byte receive data in SCFRDR, PER[3:0] shows 0000.
11 to 8	FER[3:0]	0000	R	Number of Framing Errors Indicate the quantity of data including a framing error in the receive data stored in SCFRDR. The value indicated by bits 11 to 8 after the ER bit in SCFSR is set, represents the number of framing errors in SCFRDR. When framing errors have occurred in all 16-byte receive data in SCFRDR, FER[3:0] shows 0000.

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 SCFRDR 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	1	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: * Do not use this bit as a transmit end flag when the DMAC writes data to SCFTDR due to a TXI interrupt request.

Bit	Bit Name	Initial Value	R/W	Description
5	TDFE	1	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 TTRG[1:0] 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 is activated by transmit FIFO data empty interrupt (TXI) and write data exceeding the specified transmission trigger number to SCFTDR 1: The quantity of transmit data in SCFTDR is less than or equal to 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 or equal to 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 Indication 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 Indication 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 next 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 RTRG[1:0] 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 is activated by receive FIFO data full interrupt (RXI) and 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: * As SCFTDR is a 16-byte FIFO register, the maximum quantity of data that can be read when RDF is 1 becomes the specified receive trigger number. 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 after DMAC is activated by receive FIFO data full interrupt (RXI). 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: * Only 0 can be written to clear the flag after 1 is read.

16.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that is used with the CKS1 and CKS0 bits in the serial mode register (SCSMR) and the BGDM and ABCS bits in the serial extension mode register (SCEMR) to determine 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.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The SCBRR setting is calculated as follows:

- Asynchronous mode:

When baud rate generator operates in normal mode (when the BGDM bit of SCEMR is 0):

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$N = \frac{P\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 8 times the bit rate)

When baud rate generator operates in double speed mode (when the BGDM bit of SCEMR is 1):

$$N = \frac{P\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$N = \frac{P\phi}{16 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 8 times the bit rate)

- Clock 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 ($0 \leq N \leq 255$)
(The setting must satisfy the electrical characteristics.)
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 16.3.)

Table 16.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS[1]**	**CKS[0]**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

The bit rate error in asynchronous mode is given by the following formula:

When baud rate generator operates in normal mode (the BGDM bit of SCEMR is 0):

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 8 times the bit rate)

When baud rate generator operates in double speed mode (the BGDM bit of SCEMR is 1):

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 16 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 8 times the bit rate)

Table 16.4 lists the sample SCBRR settings in asynchronous mode in which a base clock frequency is 16 times the bit rate (the ABCS bit in SCEMR is 0) and the baud rate generator operates in normal mode (the BGDM bit in SCEMR is 1), and table 16.5 lists the sample SCBRR settings in clock synchronous mode.

Table 16.4 Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0) (1)

Bit Rate (bit/s)	Pϕ (MHz)											
	8			9.8304			10			12		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00
38400	0	6	–6.99	0	7	0.00	0	7	1.73	0	9	–2.34

Table 16.4 Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0) (2)

	Pϕ (MHz)											
	12.288			14.7456			16			20		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	3	64	0.70	3	70	0.03	3	88	–0.25
150	2	159	0.00	2	191	0.00	2	207	0.16	3	64	0.16
300	2	79	0.00	2	95	0.00	2	103	0.16	2	129	0.16
600	1	159	0.00	1	191	0.00	1	207	0.16	2	64	0.16
1200	1	79	0.00	1	95	0.00	1	103	0.16	1	129	0.16
2400	0	159	0.00	0	191	0.00	0	207	0.16	1	64	0.16
4800	0	79	0.00	0	95	0.00	0	103	0.16	0	129	0.16
9600	0	39	0.00	0	47	0.00	0	51	0.16	0	64	0.16
19200	0	19	0.00	0	23	0.00	0	25	0.16	0	32	–1.36
31250	0	11	2.40	0	14	–1.70	0	15	0.00	0	19	0.00
38400	0	9	0.00	0	11	0.00	0	12	0.16	0	15	1.73

Table 16.4 Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0) (3)

	Pϕ (MHz)											
	24			24.576			28.7			30		
Bit Rate (bit/s)	n	N	Error %)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	106	–0.44	3	108	0.08	3	126	0.31	3	132	0.13
150	3	77	0.16	3	79	0.00	3	92	0.46	3	97	–0.35
300	2	155	0.16	2	159	0.00	2	186	–0.08	2	194	0.16
600	2	77	0.16	2	79	0.00	2	92	0.46	2	97	–0.35
1200	1	155	0.16	1	159	0.00	1	186	–0.08	1	194	0.16
2400	1	77	0.16	1	79	0.00	1	92	0.46	1	97	–0.35
4800	0	155	0.16	0	159	0.00	0	186	–0.08	0	194	0.16
9600	0	77	0.16	0	79	0.00	0	92	0.46	0	97	–0.35
19200	0	38	0.16	0	39	0.00	0	46	–0.61	0	48	–0.35
31250	0	23	0.00	0	24	–1.70	0	28	–1.03	0	29	0.00
38400	0	19	–2.34	0	19	0.10	0	22	1.55	0	23	1.73

Table 16.4 Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0) (4)

	Pϕ (MHz)		
	33		
Bit Rate (bit/s)	n	N	Error (%)
110	3	145	0.33
150	3	106	0.39
300	2	214	–0.07
600	2	106	0.39
1200	1	214	–0.07
2400	1	106	0.39
4800	0	214	–0.07
9600	0	106	0.39
19200	0	53	–0.54
31250	0	32	0.00
38400	0	26	–0.54

Note: Settings with an error of 1% or less are recommended.

Table 16.5 Bit Rates and SCBRR Settings (Clock Synchronous Mode)

Bit Rate (bit/s)	Pϕ (MHz)									
	8		16		28.7		30		33	
	n	N	n	N	n	N	n	N	n	N
250	3	124	3	249						
500	2	249	3	124	3	223	3	233	3	255
1 k	2	124	2	249	3	111	3	116	3	128
2.5 k	1	199	2	99	2	178	2	187	2	205
5 k	1	99	1	199	2	89	2	93	2	102
10 k	0	199	1	99	1	178	1	187	1	205
25 k	0	79	0	159	1	71	1	74	1	82
50 k	0	39	0	79	0	143	0	149	0	164
100 k	0	19	0	39	0	71	0	74	0	82
250 k	0	7	0	15	—	—	0	29	0	32
500 k	0	3	0	7	—	—	0	14	—	—
1 M			0	3	—	—	—	—	—	—
2 M					—	—	—	—	—	—

[Legend]

Blank: No setting possible, or the electrical characteristics of the SH7205 cannot be satisfied regardless of the device being communicated with.

—: Setting possible, but error occurs

Table 16.6 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Table 16.7 lists the maximum bit rates in asynchronous mode when the external clock input is used. Table 16.8 lists the maximum bit rates in clock synchronous mode when the external clock input is used (when $t_{Scyc} = 12t_{pcyc}$*).

Note: * Make sure that the electrical characteristics of this LSI and that of a connected LSI are satisfied.

Table 16.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

	Settings				
Pϕ (MHz)	BGDM	ABCS	n	N	Maximum Bit Rate (bits/s)
8	0	0	0	0	250000
		1	0	0	500000
	1	0	0	0	500000
		1	0	0	1000000
9.8304	0	0	0	0	307200
		1	0	0	614400
	1	0	0	0	614400
		1	0	0	1228800
12	0	0	0	0	375000
		1	0	0	750000
	1	0	0	0	750000
		1	0	0	1500000
14.7456	0	0	0	0	460800
		1	0	0	921600
	1	0	0	0	921600
		1	0	0	1843200
16	0	0	0	0	500000
		1	0	0	1000000
	1	0	0	0	1000000
		1	0	0	2000000

	Settings				Maximum Bit Rate
Pϕ (MHz)	BGDM	ABCS	n	N	(bits/s)
20	0	0	0	0	625000
		1	0	0	1250000
	1	0	0	0	1250000
		1	0	0	2500000
24	0	0	0	0	750000
		1	0	0	1500000
	1	0	0	0	1500000
		1	0	0	3000000
24.576	0	0	0	0	768000
		1	0	0	1536000
	1	0	0	0	1536000
		1	0	0	3072000
28.7	0	0	0	0	896875
		1	0	0	1793750
	1	0	0	0	1793750
		1	0	0	3587500
30	0	0	0	0	937500
		1	0	0	1875000
	1	0	0	0	1875000
		1	0	0	3750000
33	0	0	0	0	1031250
		1	0	0	2062500
	1	0	0	0	2062500
		1	0	0	4125000

Table 16.7 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

		Settings	
Pϕ (MHz)	External Input Clock (MHz)	ABCS	Maximum Bit Rate (bits/s)
8	2.0000	0	125000
		1	250000
9.8304	2.4576	0	153600
		1	307200
12	3.0000	0	187500
		1	375000
14.7456	3.6864	0	230400
		1	460800
16	4.0000	0	250000
		1	500000
20	5.0000	0	312500
		1	625000
24	6.0000	0	375000
		1	750000
24.576	6.1440	0	384000
		1	768000
28.7	4.9152	0	448436
		1	896872
30	7.5000	0	468750
		1	937500
33	8.2500	0	515625
		1	1031250

Table 16.8 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode, $t_{Scyc} = 12t_{pcyc}$)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
8	0.6666	666666.6
16	1.3333	1333333.3
24	2.0000	2000000.0
28.7	2.3916	2391666.6
30	2.5000	2500000.0
33	2.7500	2750000.0

16.3.9 FIFO Control Register (SCFCR)

SCFCR resets the quantity of data in the transmit and receive FIFO data 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.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	RSTRG[2:0]			RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
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 to 8	RSTRG[2:0]	000	R/W	$\overline{\text{RTS}}$ Output Active Trigger When the quantity of receive data in receive FIFO data register (SCFRDR) becomes more than the number shown below, $\overline{\text{RTS}}$ signal is set to high. 000: 15 001: 1 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14
7, 6	RTRG[1:0]	00	R/W	Receive FIFO Data Trigger 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 to 1 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 • Clock synchronous mode 00: 1 01: 2 10: 8 11: 14 Note: In clock synchronous mode, to transfer the receive data using DMAC, set the receive trigger number to 1. If set to other than 1, CPU must read the receive data left in SCFRDR.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	TTRG[1:0]	00	R/W	Transmit FIFO Data Trigger 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 to 1 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 $\overline{\text{RTS}}$. For channels 1 to 5 in clock synchronous mode, MCE bit should always be 0. 0: Modem signal disabled* 1: Modem signal enabled Note: * The $\overline{\text{CTS}}$ level has no effect on transmit operation, regardless of the input value, and the $\overline{\text{RTS}}$ level has no effect on receive operation.
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.

Bit	Bit Name	Initial Value	R/W	Description
0	LOOP	0	R/W	Loop-Back Test Internally connects the transmit output pin (TxD) and receive input pin (RxD) and internally connects the $\overline{\text{RTS}}$ pin and $\overline{\text{CTS}}$ pin and enables loop-back testing. 0: Loop back test disabled 1: Loop back test enabled

16.3.10 FIFO Data Count Set 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.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	T[4:0]					-	-	-	R[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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 to 8	T[4:0]	00000	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.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	R[4:0]	00000	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.

16.3.11 Serial Port Register (SCSPTR)

SCSPTR controls input/output and data of pins multiplexed to SCIF function. Bits 7 and 6 can control input/output data of $\overline{\text{RTS}}$ pin. Bits 5 and 4 can control input/output data of $\overline{\text{CTS}}$ pin. Bits 3 and 2 can control input/output data of SCK pin. Bits 1 and 0 can input data from RxD pin and output data to TxD pin, so they control break of serial transmitting/receiving.

The CPU can always read and write to SCSPTR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
Initial value:	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 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	$\overline{\text{RTS}}$ Port Input/Output Indicates input or output of the serial port $\overline{\text{RTS}}$ pin. When the $\overline{\text{RTS}}$ pin is actually used as a port outputting the RTSDT bit value, the MCE bit in SCFCR should be cleared to 0. 0: RTSDT bit value not output to $\overline{\text{RTS}}$ pin 1: RTSDT bit value output to $\overline{\text{RTS}}$ pin
6	RTSDT	1	R/W	$\overline{\text{RTS}}$ Port Data Indicates the input/output data of the serial port $\overline{\text{RTS}}$ pin. Input/output is specified by the RTSIO bit. For output, the RTSDT bit value is output to the $\overline{\text{RTS}}$ pin. The $\overline{\text{RTS}}$ pin status is read from the RTSDT bit regardless of the RTSIO bit setting. However, $\overline{\text{RTS}}$ input/output must be set in the PFC. 0: Input/output data is low level 1: Input/output data is high level

Bit	Bit Name	Initial Value	R/W	Description
5	CTSIO	0	R/W	$\overline{\text{CTS}}$ Port Input/Output Indicates input or output of the serial port $\overline{\text{CTS}}$ pin. When the $\overline{\text{CTS}}$ pin is actually used as a port outputting the CTSDT bit value, the MCE bit in SCFCR should be cleared to 0. 0: CTSDT bit value not output to $\overline{\text{CTS}}$ pin 1: CTSDT bit value output to $\overline{\text{CTS}}$ pin
4	CTSDT	1	R/W	$\overline{\text{CTS}}$ Port Data Indicates the input/output data of the serial port $\overline{\text{CTS}}$ pin. Input/output is specified by the CTSIO bit. For output, the CTSDT bit value is output to the $\overline{\text{CTS}}$ pin. The $\overline{\text{CTS}}$ pin status is read from the CTSDT bit regardless of the CTSIO bit setting. However, $\overline{\text{CTS}}$ input/output must be set in the PFC. 0: Input/output data is low level 1: Input/output data is high level
3	SCKIO	0	R/W	SCK Port Input/Output Indicates input or output of the serial port SCK pin. When the SCK pin is actually used as a port outputting the SCKDT bit value, the CKE[1:0] bits in SCSCR should be cleared to 0. 0: SCKDT bit value not output to SCK pin 1: SCKDT bit value output to SCK pin
2	SCKDT	0	R/W	SCK Port Data Indicates the input/output data of the serial port SCK pin. Input/output is specified by the SCKIO bit. For output, the SCKDT bit value is output to the SCK pin. The SCK pin status is read from the SCKDT bit regardless of the SCKIO bit setting. However, SCK input/output must be set in the PFC. 0: Input/output data is low level 1: Input/output data is high level

Bit	Bit Name	Initial Value	R/W	Description
1	SPB2IO	0	R/W	Serial Port Break Input/Output Indicates input or output of the serial port TxD pin. When the TxD pin is actually used as a port outputting the SPB2DT bit value, the TE bit in SCSCR should be cleared to 0. 0: SPB2DT bit value not output to TxD pin 1: SPB2DT bit value output to TxD pin
0	SPB2DT	0	R/W	Serial Port Break Data Indicates the input data of the RxD pin and the output data of the TxD pin used as serial ports. Input/output is specified by the SPB2IO bit. When the TxD pin is set to output, the SPB2DT bit value is output to the TxD pin. The RxD pin status is read from the SPB2DT bit regardless of the SPB2IO bit setting. However, RxD input and TxD output must be set in the PFC. 0: Input/output data is low level 1: Input/output data is high level

16.3.12 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).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ORER
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*

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) retains the data before an overrun error has occurred, and the next received data is discarded. When the ORER bit is set to 1, the SCIF cannot continue the next serial reception.

Note: * Only 0 can be written to clear the flag after 1 is read.

16.3.13 Serial Extension Mode Register (SCEMR)

The CPU can always read from or write to SCEMR. Setting the BGDM bit in this register to 1 allows the baud rate generator in the SCIF operates in double-speed mode when asynchronous mode is selected (by setting the C/$\overline{A}$ bit in SCSMR to 0) and an internal clock is selected as a clock source and the SCK pin is set as an input pin (by setting the CKE[1:0] bits in SCSCR to 00).

The base clock frequency in asynchronous mode can be selected by modifying the ABCS bit setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	BGDM	-	-	-	-	-	-	ABCS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W

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	BGDM	0	R/W	Baud Rate Generator Double-Speed Mode When the BGDM bit is set to 1, the baud rate generator in the SCIF operates in double-speed mode. This bit is valid only when asynchronous mode is selected by setting the C/$\overline{A}$ bit in SCSMR to 0 and an internal clock is selected as a clock source and the SCK pin is set as an input pin by setting the CKE[1:0] bits in SCSCR to 00. In other settings, this bit is invalid (the baud rate generator operates in normal mode regardless of the BGDM setting). 0: Normal mode 1: Double-speed mode
6 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ABCS	0	R/W	Base Clock Select in Asynchronous Mode This bit selects the base clock frequency in asynchronous mode. This bit is valid only in asynchronous mode (when the C/$\overline{A}$ bit in SCSMR is 0). 0: Base clock frequency is 16 times the bit rate 1: Base clock frequency is 8 times the bit rate

16.4 Operation

16.4.1 Overview

For serial communication, the SCIF has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses. However, clock synchronous mode is not available on channels 3 and 4.

The SCIF has a 16-stage FIFO buffer for both transmission and receptions, reducing the overhead of the CPU, and enabling continuous high-speed communication. Furthermore, channel 0 has $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ signals to be used as modem control signals.

The transmission format is selected in the serial mode register (SCSMR), as shown in table 16.9. The SCIF clock source is selected by the combination of the CKE1 and CKE0 bits in the serial control register (SCSCR), as shown in table 16.10.

(1) 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 clock of on-chip baud rate generator.
  - When an external clock is selected, the external clock input must have a frequency 16 or 8 times the bit rate. (The on-chip baud rate generator is not used.)

(2) Clock Synchronous Mode (Channels 0, 1, 2 and 5 only)

- 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 clock of the on-chip baud rate generator, and outputs this clock to external devices as the synchronous clock.
  - When an external clock is selected, the SCIF operates on the input external synchronous clock not using the on-chip baud rate generator.

Table 16.9 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	x	x	x	Clock synchronous	8 bits	Not set	None

[Legend]
x: Don't care

Table 16.10 SCSMR and SCSCR Settings and SCIF Clock Source Selection

SCSMR	SCSCR		SCIF Transmit/Receive Clock	
Bit 7 C/$\overline{A}$	Bit 1, 0 CKE[1:0]	Mode	Clock Source	SCK Pin Function
0	00	Asynchronous	Internal	SCIF does not use the SCK pin
	01			Outputs a clock with a frequency 16/8 times the bit rate
	10		External	Inputs a clock with frequency 16/8 times the bit rate
	11		Setting prohibited	
1	0x	Clock synchronous	Internal	Outputs the serial clock
	10		External	Inputs the serial clock
	11		Setting prohibited	

[Legend]

x: Don't care

Note: When using the baud rate generator in double-speed mode (BGMD = 1), select asynchronous mode by setting the C/A bit to 0, and select an internal clock as a clock source and the SCK pin is not used (the CKE[1:0] bits set to 00).

16.4.2 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 16.2 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 or fourth pulse of a clock with a frequency 16 or 8 times the bit rate. Receive data is latched at the center of each bit.

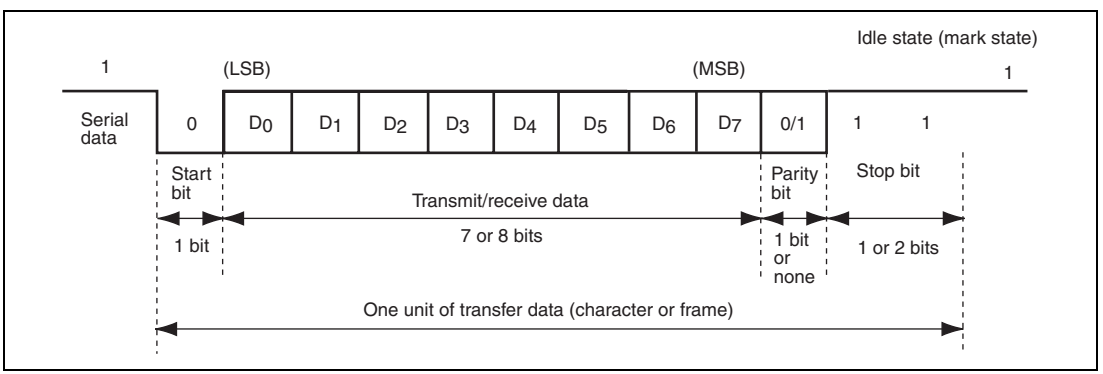


Figure 16.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

(1) Transmit/Receive Formats

Table 16.11 lists the eight communication formats that can be selected in asynchronous mode. The format is selected by settings in the serial mode register (SCSMR).

Table 16.11 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

(2) 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 the CKE1 and CKE0 bits in the serial control register (SCSCR). For clock source selection, refer to table 16.10, SCSMR and SCSCR Settings and SCIF Clock Source Selection.

When an external clock is input at the SCK pin, it must have a frequency equal to 16 or 8 times the desired bit rate.

When the SCIF operates on an internal clock, it can output a clock signal on the SCK pin. The frequency of this output clock is 16 or 8 times the desired bit rate.

(3) 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 16.3 shows a sample flowchart for initializing the SCIF.

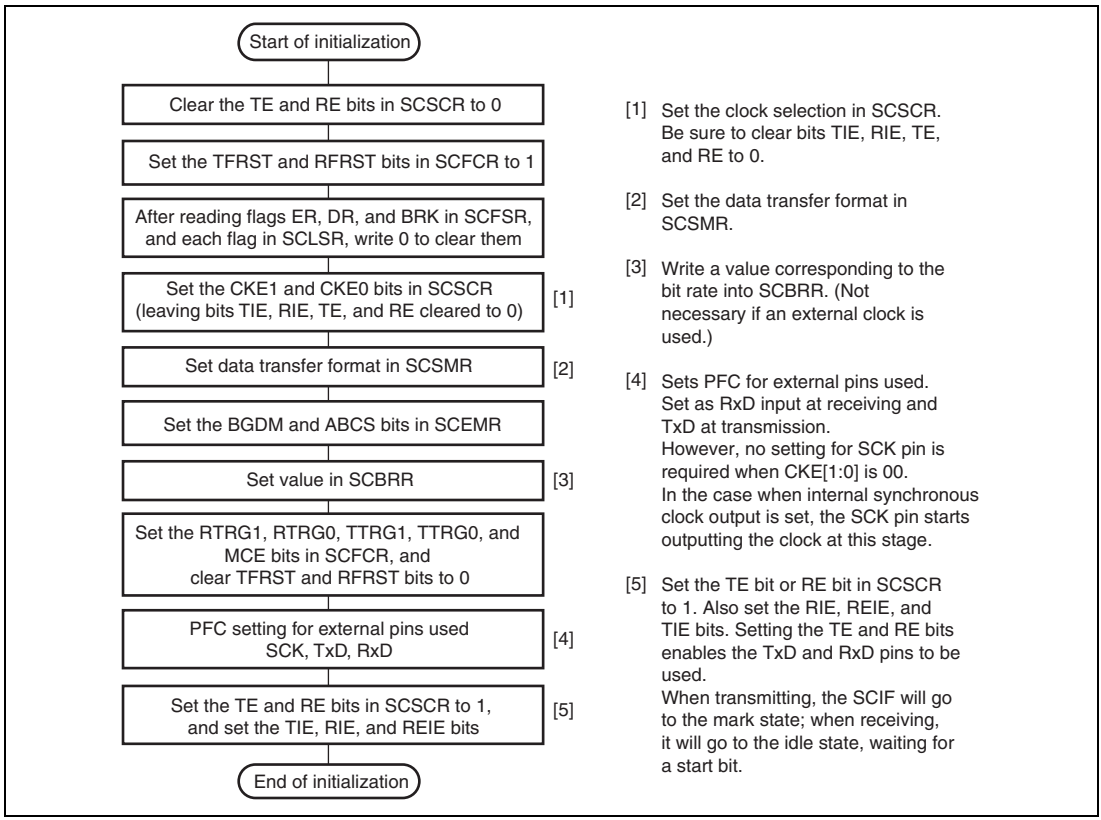
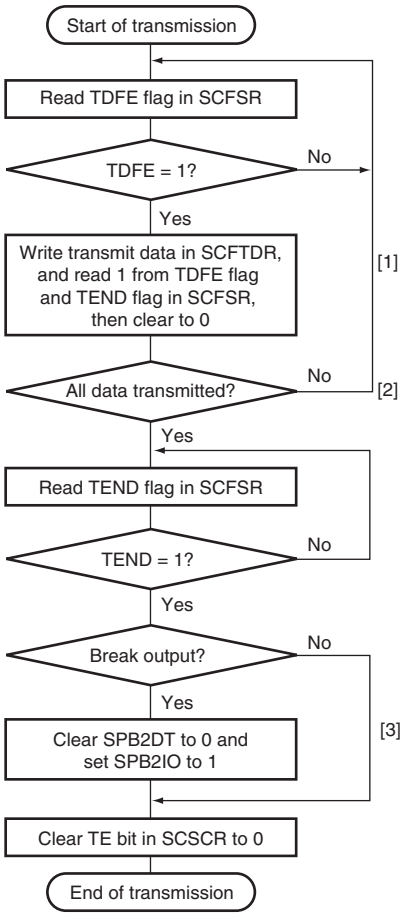


Figure 16.3 Sample Flowchart for SCIF Initialization

- Transmitting Serial Data (Asynchronous Mode)

  Figure 16.4 shows a sample flowchart for serial transmission.

  Use the following procedure for serial data transmission after enabling the SCIF for transmission.



[1] SCIF status check and transmit data write:
Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, and read 1 from the TDFE and TEND flags, then clear to 0.
The quantity of transmit data that can be written is 16 - (transmit trigger set number).

[2] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCFTDR, and then clear the TDFE flag to 0.

[3] Break output during serial transmission:
To output a break in serial transmission, clear the SPB2DT bit to 0 and set the SPB2IO bit to 1 in SCSPTR, then clear the TE bit in SCSCR to 0.

In [1] and [2], it is possible to ascertain the number of data bytes that can be written from the number of transmit data bytes in SCFTDR indicated by the upper 8 bits of SCFDR.

Figure 16.4 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.

Figure 16.5 shows an example of the operation for transmission.

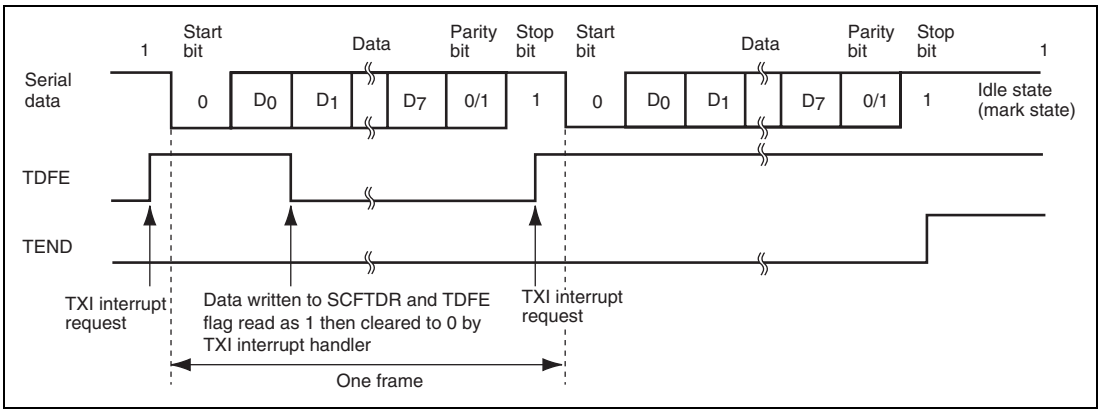


Figure 16.5 Example of Transmit Operation (8-Bit Data, Parity, 1 Stop Bit)

4. When modem control is enabled in channel 0, 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 16.6 shows an example of the operation when modem control is used.

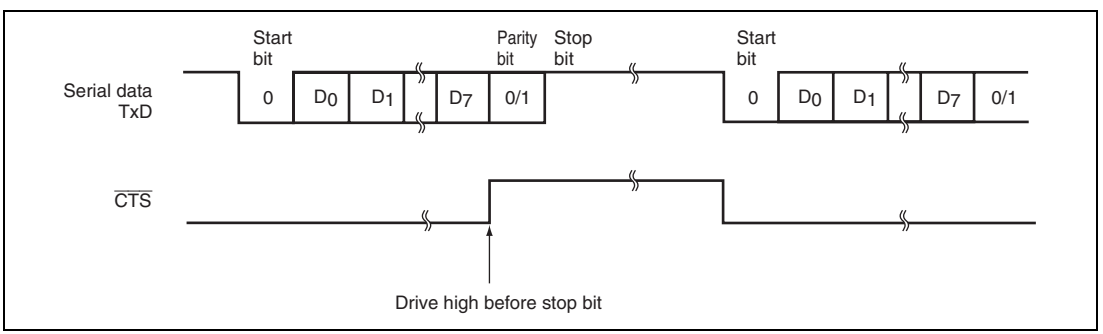
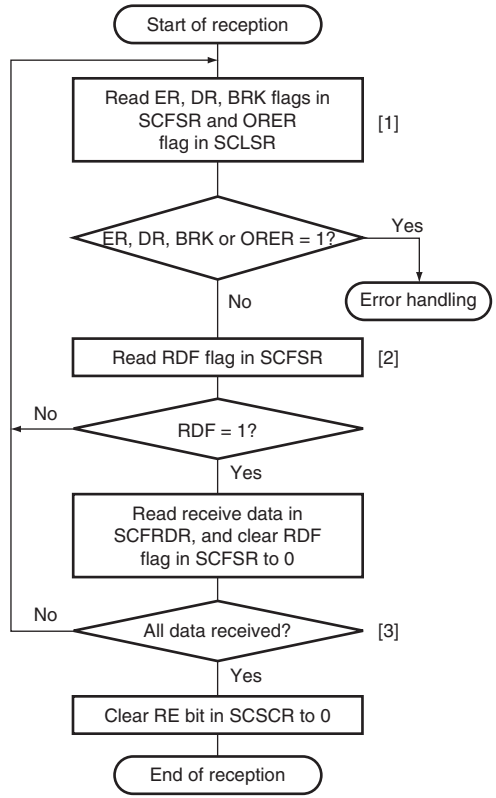


Figure 16.6 Example of Operation Using Modem Control ($\overline{\text{CTS}}$)

- Receiving Serial Data (Asynchronous Mode)

  Figures 16.7 and 16.8 show sample flowcharts for serial reception.

  Use the following procedure for serial data reception after enabling the SCIF for reception.



[1] Receive error handling and break detection:

Read the DR, ER, and BRK flags in SCFSR, and the ORER flag in SCLSR, to identify any error, perform the appropriate error handling, then clear the DR, ER, BRK, and ORER flags to 0. In the case of a framing error, a break can also be detected by reading the value of the RxD pin.

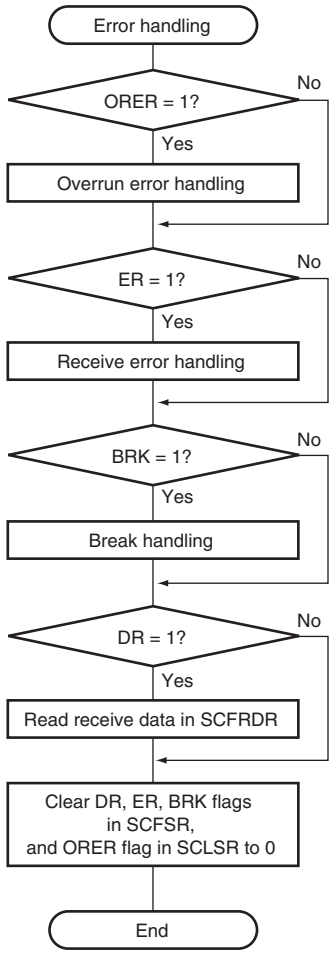
[2] SCIF status check and receive data read:

Read SCFSR and check that RDF flag = 1, then read the receive data in SCFRDR, read 1 from the RDF flag, and then clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by a receive FIFO data full interrupt (RXI).

[3] Serial reception continuation procedure:

To continue serial reception, read at least the receive trigger set number of receive data bytes from SCFRDR, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading from SCRFDR.

Figure 16.7 Sample Flowchart for Receiving Serial Data



- Whether a framing error or parity error has occurred in the receive data that is to be read from the receive FIFO data register (SCFRDR) can be ascertained from the FER and PER bits in the serial status register (SCFSR).
- 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 16.8 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 16.9 shows an example of the operation for reception.

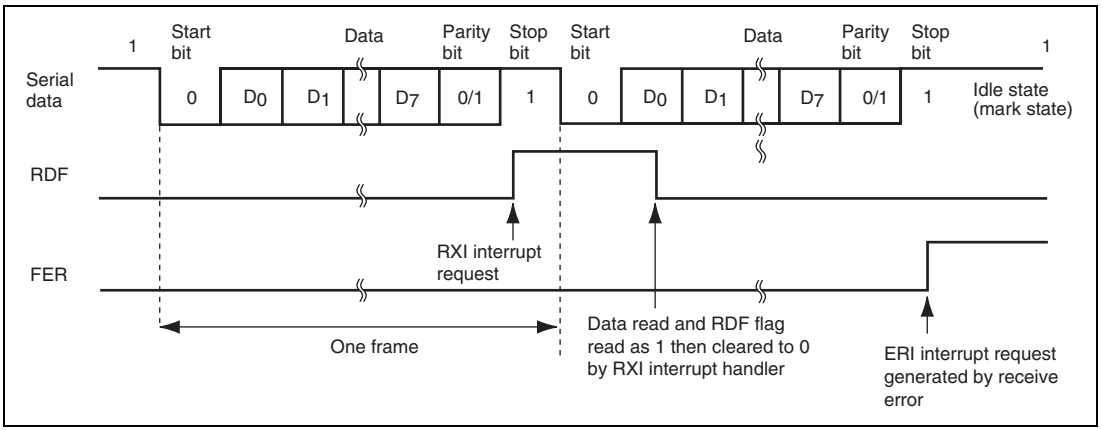


Figure 16.9 Example of SCIF Receive Operation (8-Bit Data, Parity, 1 Stop Bit)

5. When modem control is enabled in channel 0, the $\overline{\text{RTS}}$ signal is output when SCFRDR is empty. 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 16.10 shows an example of the operation when modem control is used.

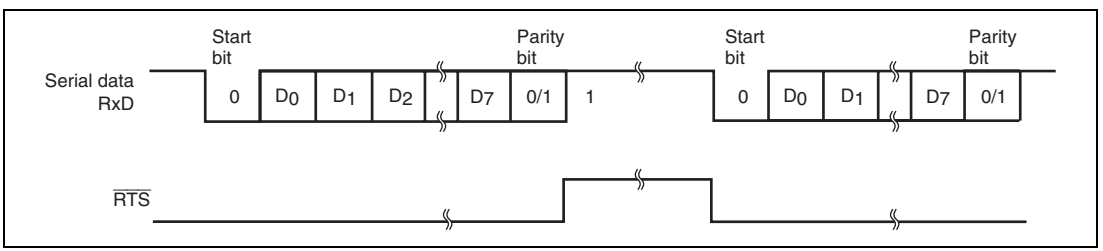


Figure 16.10 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

16.4.3 Operation in Clock Synchronous Mode

In clock 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 16.11 shows the general format in clock synchronous serial communication.

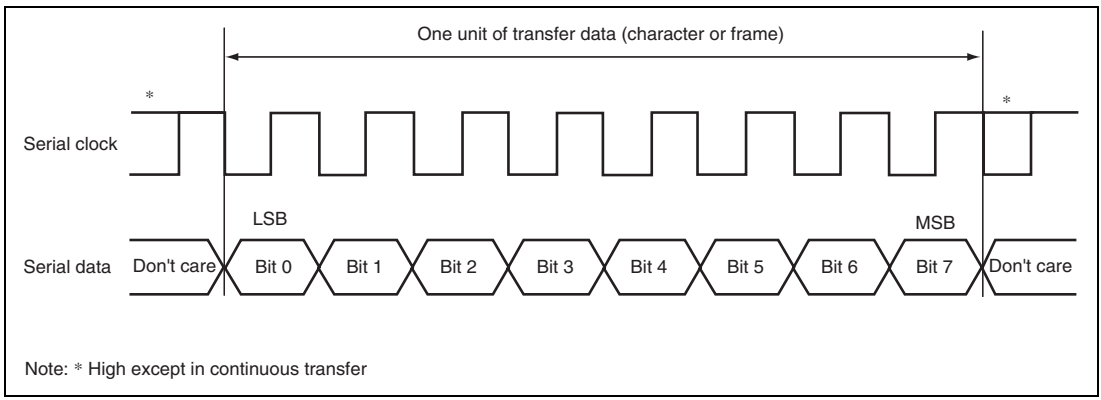


Figure 16.11 Data Format in Clock Synchronous Communication

In clock 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 clock synchronous mode, the SCIF receives data by synchronizing with the rising edge of the serial clock.

(1) Transmit/Receive Formats

The data length is fixed at eight bits. No parity bit can be added.

(2) Clock

An internal clock generated by the on-chip baud rate generator by the setting of the C/A bit in SCSMR and CKE[1:0] in SCSCR, 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 more than the receive FIFO data trigger number.

(3) Transmitting and Receiving Data

- SCIF Initialization (Clock 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 16.12 shows a sample flowchart for initializing the SCIF.

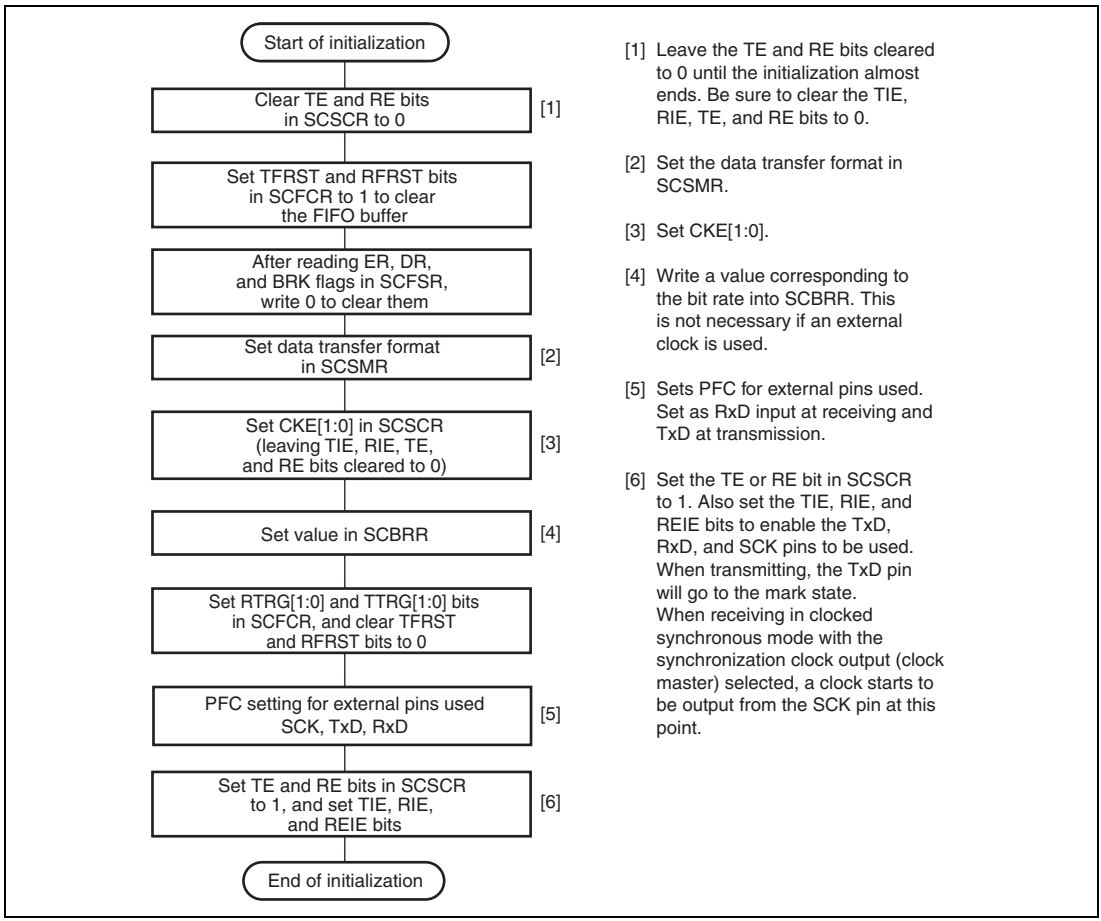


Figure 16.12 Sample Flowchart for SCIF Initialization

- Transmitting Serial Data (Clock Synchronous Mode)

  Figure 16.13 shows a sample flowchart for transmitting serial data.

  Use the following procedure for serial data transmission after enabling the SCIF for transmission.

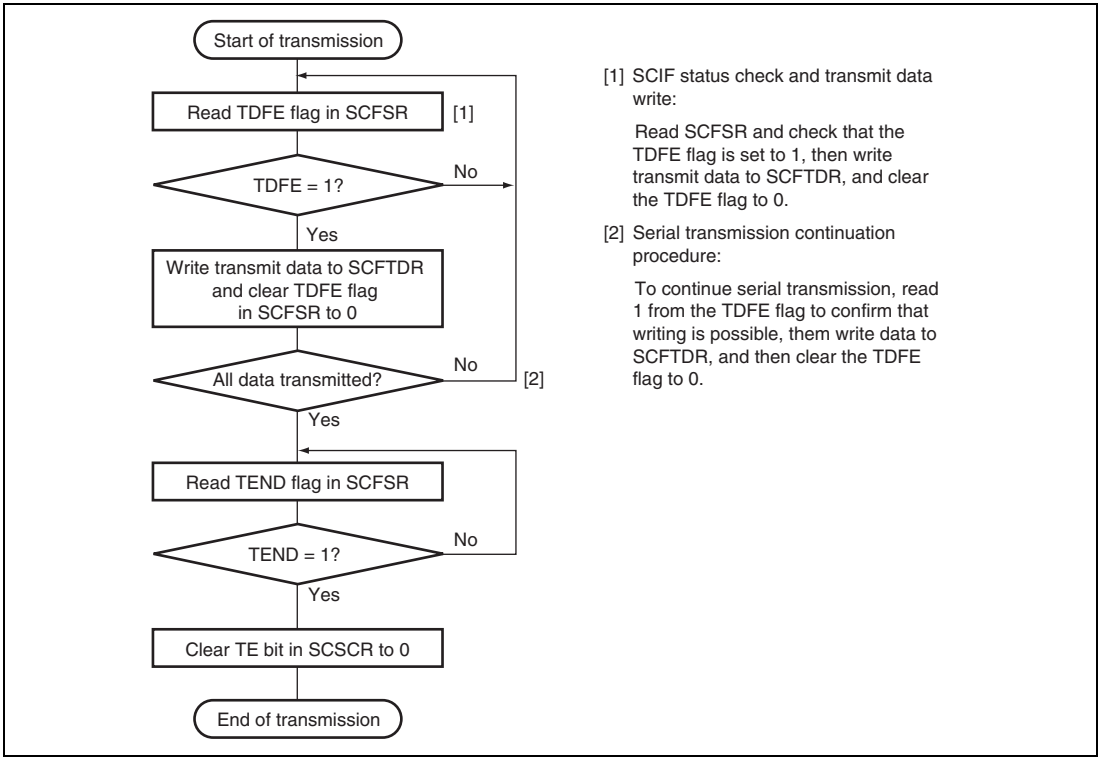


Figure 16.13 Sample Flowchart for Transmitting Serial Data

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, and then serial transmission of the next frame is started. If there is no data, the TxD pin holds the state after the TEND flag in SCFSR is set to 1 and the MSB (bit 7) is sent.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 16.14 shows an example of SCIF transmit operation.

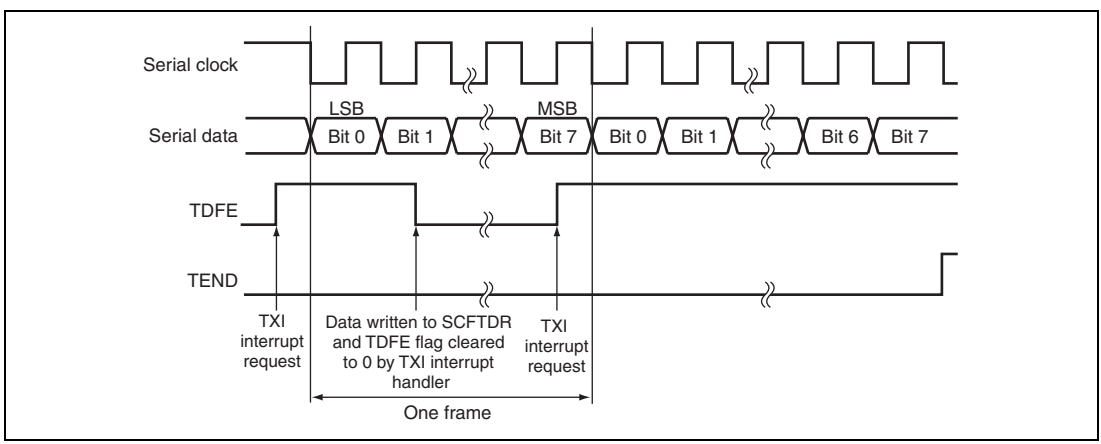


Figure 16.14 Example of SCIF Transmit Operation

- Receiving Serial Data (Clock Synchronous Mode)

  Figures 16.15 and 16.16 show sample flowcharts for receiving serial data. When switching from asynchronous mode to clock synchronous mode without SCIF initialization, make sure that ORER, PER, and FER are cleared to 0.

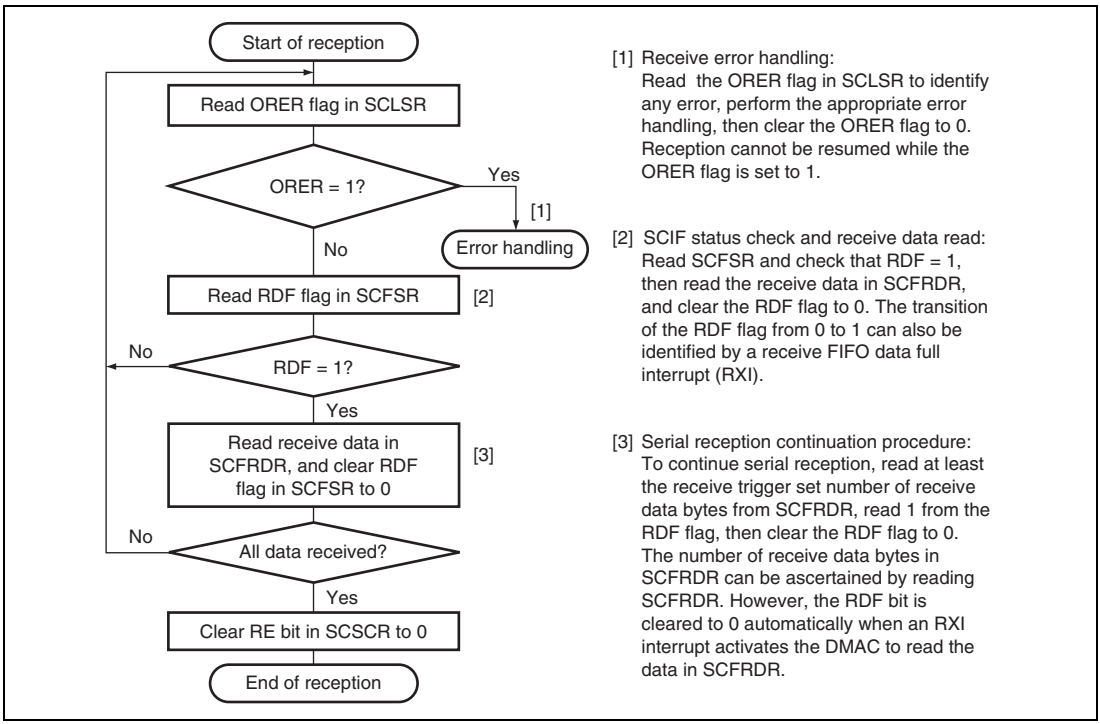


Figure 16.15 Sample Flowchart for Receiving Serial Data (1)

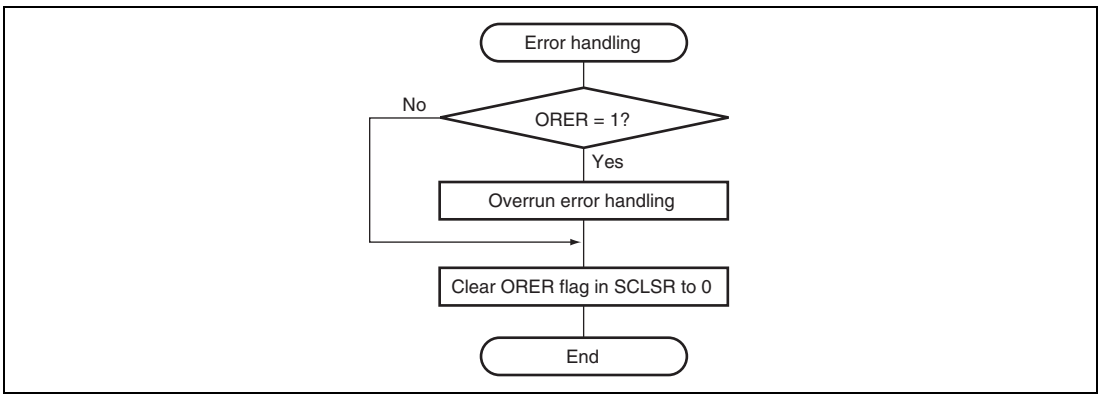


Figure 16.16 Sample Flowchart for Receiving Serial Data (2)

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 RDF flag is set to 1 and 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 FIFO 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 16.17 shows an example of SCIF receive operation.

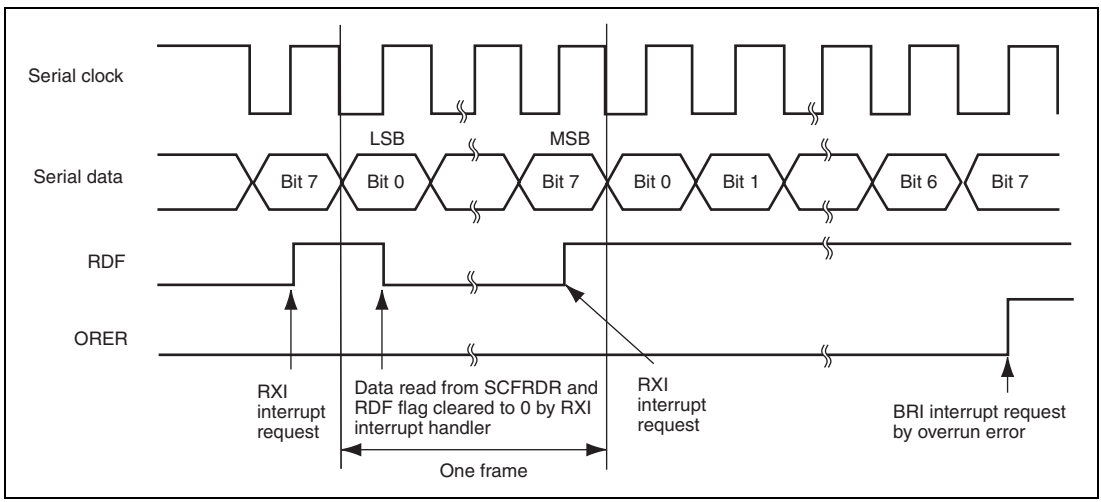
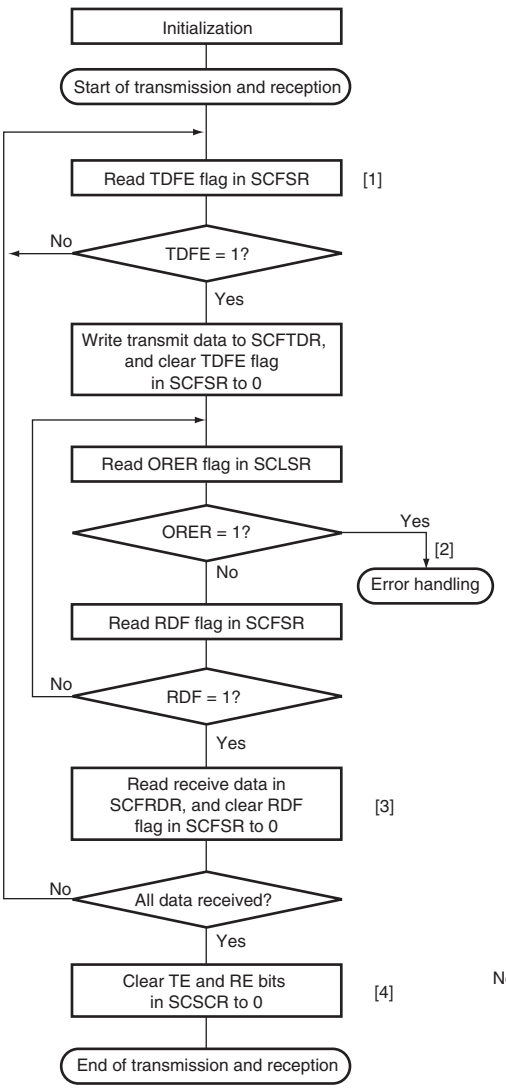


Figure 16.17 Example of SCIF Receive Operation

- Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode)

  Figure 16.18 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.



[1] SCIF status check and transmit data write:

Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, and clear the TDFE flag to 0. The transition of the TDFE flag from 0 to 1 can also be identified by a transmit FIFO data empty interrupt (TXI).

[2] Receive error handling:

Read the ORER flag in SCLSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0. Reception cannot be resumed while the ORER flag is set to 1.

[3] SCIF status check and receive data read:

Read SCFSR and check that RDF flag = 1, then read the receive data in SCFRDR, and clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by a receive FIFO data full interrupt (RXI).

[4] Serial transmission and reception continuation procedure:

To continue serial transmission and reception, read 1 from the RDF flag and the receive data in SCFRDR, and clear the RDF flag to 0 before receiving the MSB in the current frame. Similarly, read 1 from the TDFE flag to confirm that writing is possible before transmitting the MSB in the current frame. Then write data to SCFTDR and clear the TDFE flag to 0.

Note: When switching from a transmit operation or receive operation to simultaneous transmission and reception operations, clear the TE and RE bits to 0, and then set them simultaneously to 1.

Figure 16.18 Sample Flowchart for Transmitting/Receiving Serial Data

16.5 SCIF Interrupts

The SCIF has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive FIFO data full (RXI), and break (BRI).

Table 16.12 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 a TXI request is enabled by the TIE bit and the TDFE flag in the serial status register (SCFSR) is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DMA transfer or the CPU interrupt to perform data transfer. The DMA transfer or the CPU interrupt is selectable by the DMA transfer request enable register (DREQER) of the INTC.

When an RXI request is enabled by the RIE bit and the RDF flag or the DR flag in SCFSR is set to 1, an RXI interrupt request is generated. This RXI interrupt request activates the DMA transfer or the CPU interrupt to perform data transfer. The DMA transfer or the CPU interrupt is selectable by the DMA transfer request enable register (DREQER) of the INTC. The RXI interrupt request caused by the DR flag is generated only in asynchronous mode.

When the RIE bit is set to 0 and the REIE bit is set to 1, the SCIF requests an ERI or BRI interrupt without requesting an RXI interrupt.

The TXI indicates that transmit data can be written, and the RXI indicates that there is receive data in SCFRDR.

Table 16.12 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
BRI	Interrupt initiated by break (BRK) or overrun error (ORER)	Not possible	High
ERI	Interrupt initiated by receive error (ER)	Not possible	
RXI	Interrupt initiated by receive FIFO data full (RDF) or data ready (DR)	Possible	
TXI	Interrupt initiated by transmit FIFO data empty (TDFE)	Possible	Low

16.6 Usage Notes

Note the following when using the SCIF.

16.6.1 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 TTRG[1:0] in the FIFO control register (SCFCR). After the TDFE flag 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 flag 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).

16.6.2 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 RTRG[1:0] in the FIFO control register (SCFCR). After RDF flag 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 exceeds the trigger number, the RDF flag will be set to 1 again if it is cleared to 0. The RDF flag should therefore be cleared to 0 after being read as 1 after reading the number of the received data in the receive FIFO data register (SCFRDR) which is less than the trigger number.

The number of receive data bytes in SCFRDR can be found from the lower 8 bits of the FIFO data count register (SCFDR).

16.6.3 Restriction on DMAC Usage

When the DMAC writes data to SCFTDR due to a TXI interrupt request, the state of the TEND flag becomes undefined. Therefore, the TEND flag should not be used as the transfer end flag in such a case.

16.6.4 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.

16.6.5 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, the TxD pin does not work. During the period, mark status is performed by the 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.

16.6.6 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

The SCIF operates on a base clock with a frequency 16 or 8 times the bit 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 or fourth base clock pulse. When the SCIF operates on a base clock with a frequency 16 times the bit rate, the receive data is sampled at the timing shown in figure 16.19.

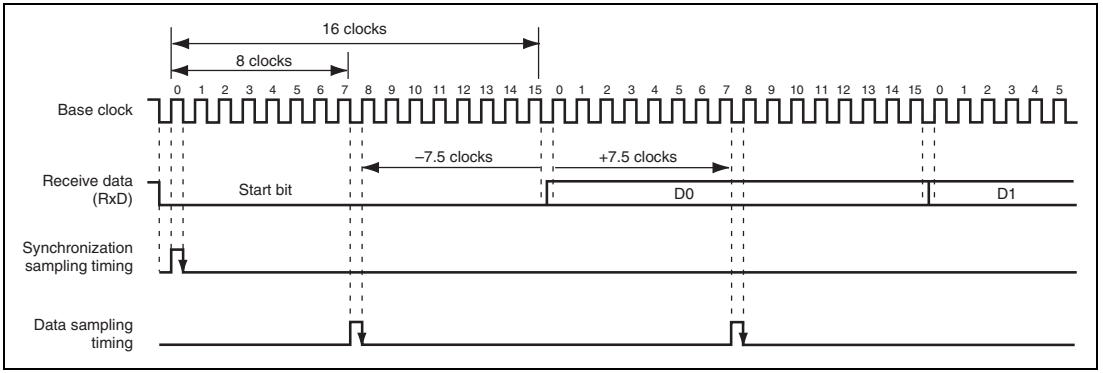


Figure 16.19 Receive Data Sampling Timing in Asynchronous Mode (Operation on a Base Clock with a Frequency 16 Times the Bit Rate)

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 or 8)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation 1, if F = 0, D = 0.5 and N = 16, 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%.

16.6.7 Selection of Base Clock in Asynchronous Mode

In this LSI, when asynchronous mode is selected, the base clock frequency within a bit period can be set to the frequency 16 or 8 times the bit rate by setting the ABCS bit in SCEMR.

Note that, however, if the base clock frequency 8 times the bit rate is used, receive margin is decreased as calculated using equation 1 in section 16.6.6, Receive Data Sampling Timing and Receive Margin (Asynchronous Mode).

If the desired bit rate can be set simply by setting SCBRR and the CKS1and CKS0 bits in SCSMR, it is recommended to use the base clock frequency within a bit period 16 times the bit rate (by setting the ABCS bit in SCEMR to 0). If an internal clock is selected as a clock source and the SCK pin is not used, the bit rate can be increased without decreasing receive margin by selecting double-speed mode for the baud rate generator (setting the BGDM bit in SCEMR to 1).

Section 17 Synchronous Serial Communication Unit (SSU)

This LSI has two synchronous serial communication unit (SSU) channels. The SSU has master mode in which this LSI outputs clocks as a master device for synchronous serial communication and slave mode in which clocks are input from an external device for synchronous serial communication. Synchronous serial communication can be performed with devices having different clock polarity and clock phase.

17.1 Features

- Choice of SSU mode and clock synchronous mode
- Choice of master mode and slave mode
- Choice of standard mode and bidirectional mode
- Synchronous serial communication with devices with different clock polarity and clock phase
- Choice of 8/16/32-bit width of transmit/receive data
- Full-duplex communication capability

  The shift register is incorporated, enabling transmission and reception to be executed simultaneously.
- Consecutive serial communication
- Choice of LSB-first or MSB-first transfer
- Choice of a clock source

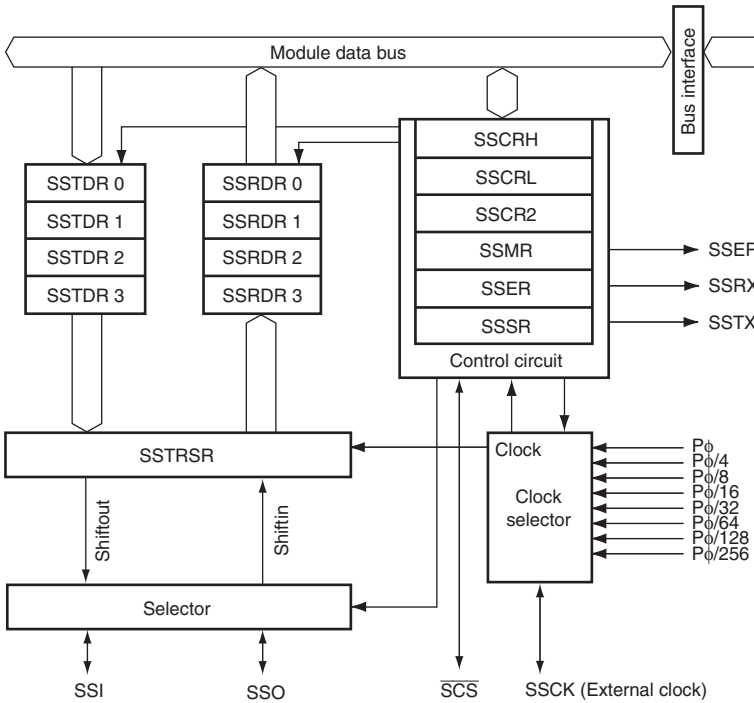
  Pϕ/4, Pϕ/8, Pϕ/16, Pϕ/32, Pϕ/64, Pϕ/128, Pϕ/256, or an external clock
- Five interrupt sources

  Transmit end, transmit data register empty, receive data full, overrun error, and conflict error.

  The direct memory access controller (DMAC) can be activated by a transmit data register empty request or a receive data full request to transfer data.
- Module standby mode can be set

  To reduce power consumption, the operation of the SSU can be suspended by stopping the clock supply to the SSU.

Figure 17.1 shows a block diagram of the SSU.



[Legend]

SSCRH:	SS control register H
SSCRL:	SS control register L
SSCR2:	SS control register 2
SSMR:	SS mode register
SSER:	SS enable register
SSSR:	SS status register
SSTDR0 to SSTDR3:	SS transmit data registers 0 to 3
SSRDR0 to SSRDR3:	SS receive data registers 0 to 3
SSTRSR:	SS shift register

Figure 17.1 Block Diagram of SSU

17.2 Input/Output Pins

Table 17.1 shows the SSU pin configuration.

Table 17.1 Pin Configuration

Channel	Symbol	I/O	Function
0, 1	SSCK0, SSCK1	I/O	SSU clock input/output
	SSI0, SSI1	I/O	SSU data input/output
	SSO0, SSO1	I/O	SSU data input/output
	$\overline{\text{SCS0}}$, $\overline{\text{SCS1}}$	I/O	SSU chip select input/output

17.3 Register Descriptions

The SSU has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 32, List of Registers.

Table 17.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access size
0	SS control register H_0	SSCRH_0	R/W	H'0D	H'FFFE7000	8, 16
	SS control register L_0	SSCRL_0	R/W	H'00	H'FFFE7001	8
	SS mode register_0	SSMR_0	R/W	H'00	H'FFFE7002	8, 16
	SS enable register_0	SSER_0	R/W	H'00	H'FFFE7003	8
	SS status register_0	SSSR_0	R/W	H'04	H'FFFE7004	8, 16
	SS control register 2_0	SSCR2_0	R/W	H'00	H'FFFE7005	8
	SS transmit data register 0_0	SSTDR0_0	R/W	H'00	H'FFFE7006	8, 16
	SS transmit data register 1_0	SSTDR1_0	R/W	H'00	H'FFFE7007	8
	SS transmit data register 2_0	SSTDR2_0	R/W	H'00	H'FFFE7008	8, 16
	SS transmit data register 3_0	SSTDR3_0	R/W	H'00	H'FFFE7009	8
	SS receive data register 0_0	SSRDR0_0	R	H'00	H'FFFE700A	8, 16
	SS receive data register 1_0	SSRDR1_0	R	H'00	H'FFFE700B	8
	SS receive data register 2_0	SSRDR2_0	R	H'00	H'FFFE700C	8, 16
	SS receive data register 3_0	SSRDR3_0	R	H'00	H'FFFE700D	8
1	SS control register H_1	SSCRH_1	R/W	H'0D	H'FFFE7800	8, 16
	SS control register L_1	SSCRL_1	R/W	H'00	H'FFFE7801	8
	SS mode register_1	SSMR_1	R/W	H'00	H'FFFE7802	8, 16
	SS enable register_1	SSER_1	R/W	H'00	H'FFFE7803	8
	SS status register_1	SSSR_1	R/W	H'04	H'FFFE7804	8, 16
	SS control register 2_1	SSCR2_1	R/W	H'00	H'FFFE7805	8
	SS transmit data register 0_1	SSTDR0_1	R/W	H'00	H'FFFE7806	8, 16
	SS transmit data register 1_1	SSTDR1_1	R/W	H'00	H'FFFE7807	8
	SS transmit data register 2_1	SSTDR2_1	R/W	H'00	H'FFFE7808	8, 16
	SS transmit data register 3_1	SSTDR3_1	R/W	H'00	H'FFFE7809	8
	SS receive data register 0_1	SSRDR0_1	R	H'00	H'FFFE780A	8, 16

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access size
1	SS receive data register 1_1	SSRDR1_1	R	H'00	H'FFFE780B	8
	SS receive data register 2_1	SSRDR2_1	R	H'00	H'FFFE780C	8, 16
	SS receive data register 3_1	SSRDR3_1	R	H'00	H'FFFE780D	8

17.3.1 SS Control Register H (SSCRH)

SSCRH specifies master/slave device selection, bidirectional mode enable, SSO pin output value selection, SSCK pin selection, and $\overline{SCS}$ pin selection.

Bit:	7	6	5	4	3	2	1	0
	MSS	BIDE	-	SOL	SOLP	-	CSS[1:0]	
Initial value:	0	0	0	0	1	1	0	1
R/W:	R/W	R/W	R	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MSS	0	R/W	Master/Slave Device Select Selects that this module is used in master mode or slave mode. When master mode is selected, transfer clocks are output from the SSCK pin. When the CE bit in SSSR is set, this bit is automatically cleared. 0: Slave mode is selected. 1: Master mode is selected.
6	BIDE	0	R/W	Bidirectional Mode Enable Selects that both serial data input pin and output pin are used or one of them is used. However, transmission and reception are not performed simultaneously when bidirectional mode is selected. For details, section 17.4.3, Relationship between Data Input/Output Pins and Shift Register. 0: Standard mode (two pins are used for data input and output) 1: Bidirectional mode (one pin is used for data input and output)

Bit	Bit Name	Initial Value	R/W	Description
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	SOL	0	R/W	Serial Data Output Value Select The serial data output retains its level of the last bit after completion of transmission. The output level before or after transmission can be specified by setting this bit. When specifying the output level, use the MOV instruction after clearing the SOLP bit to 0. Since writing to this bit during data transmission causes malfunctions, this bit should not be changed. 0: Serial data output is changed to low. 1: Serial data output is changed to high.
3	SOLP	1	R/W	SOL Bit Write Protect When changing the output level of serial data, set the SOL bit to 1 or clear the SOL bit to 0 after clearing the SOLP bit to 0 using the MOV instruction. Before writing 0 to this bit, read this bit and make sure that it is set to 1. 0: Output level can be changed by the SOL bit 1: Output level cannot be changed by the SOL bit.
2	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
1, 0	CSS[1:0]	01	R/W	$\overline{\text{SCS}}$ Pin Select Select that the $\overline{\text{SCS}}$ pin functions as $\overline{\text{SCS}}$ input or output. 00: Setting prohibited 01: Setting prohibited 10: Function as $\overline{\text{SCS}}$ automatic input/output (function as $\overline{\text{SCS}}$ input before and after transfer and output a low level during transfer) 11: Function as $\overline{\text{SCS}}$ automatic output (outputs a high level before and after transfer and outputs a low level during transfer)

17.3.2 SS Control Register L (SSCRL)

SSCRL selects operating mode, software reset, and transmit/receive data length.

Bit:	7	6	5	4	3	2	1	0
	-	SSUMS	SRES	-	-	-	DATS[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	SSUMS	0	R/W	Selects transfer mode from SSU mode and clock synchronous mode. 0: SSU mode 1: Clock synchronous mode
5	SRES	0	R/W	Software Reset Setting this bit to 1 forcibly resets the SSU internal sequencer. After that, this bit is automatically cleared. The ORER, TEND, TDRE, RDRF, and CE bits in SSSR and the TE and RE bits in SSER are also initialized. Values of other bits for SSU registers are held. To stop transfer, set this bit to 1 to reset the SSU internal sequencer.
4 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	DATS[1:0]	00	R/W	Transmit/Receive Data Length Select Select serial data length. 00: 8 bits 01: 16 bits 10: 32 bits 11: Setting prohibited

17.3.3 SS Mode Register (SSMR)

SSMR selects the MSB first/LSB first, clock polarity, clock phase, and clock rate of synchronous serial communication.

Bit:	7	6	5	4	3	2	1	0
	MLS	CPOS	CPHS	-	-	CKS[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB First/LSB First Select Selects that the serial data is transmitted in MSB first or LSB first. 0: LSB first 1: MSB first
6	CPOS	0	R/W	Clock Polarity Select Selects the SSCK clock polarity. 0: High output in idle mode, and low output in active mode 1: Low output in idle mode, and high output in active mode
5	CPHS	0	R/W	Clock Phase Select (Only for SSU Mode) Selects the SSCK clock phase. 0: Data changes at the first edge. 1: Data is latched at the first edge.
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	CKS[2:0]	000	R/W	Transfer Clock Rate Select Select the transfer clock rate (prescaler division rate) when an internal clock is selected. 000: Reserved 001: Pϕ/4 010: Pϕ/8 011: Pϕ/16 100: Pϕ/32 101: Pϕ/64 110: Pϕ/128 111: Pϕ/256

17.3.4 SS Enable Register (SSER)

SSER enables or disables transmission, reception, and interrupt requests.

Bit:	7	6	5	4	3	2	1	0
	TE	RE	-	-	TEIE	TIE	RIE	CEIE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled.
6	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, generation of an SSTXI interrupt request at the end of transmission is enabled.
2	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, generation of an SSTXI interrupt request at transmit data empty is enabled.

Bit	Bit Name	Initial Value	R/W	Description
1	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, generation of an SSRXI interrupt request and an SSERI interrupt request upon an overrun error are enabled.
0	CEIE	0	R/W	Conflict Error Interrupt Enable When this bit is set to 1, generation of an SSERI interrupt request upon a conflict error is enabled.

17.3.5 SS Status Register (SSSR)

SSSR is a status flag register for interrupts.

Bit:	7	6	5	4	3	2	1	0
	-	ORER	-	-	TEND	TDRE	RDRF	CE
Initial value:	0	0	0	0	0	1	0	0
R/W:	R	R/W	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	ORER	0	R/W	Overrun Error If the next data is received while RDRF = 1, an overrun error occurs, indicating abnormal termination. SSRDR stores 1-frame receive data before an overrun error occurs and loses data to be received later. While ORER = 1, consecutive serial reception cannot be continued. Serial transmission cannot be continued, either. However, this bit is invalid during the slave data receive operation (MSS = 0 in SSCRH, and TE = 0 and RE = 1 in SSER) in SSU mode (SSUMS = 0 in SSCRL). [Setting condition] • When one byte of the next reception is completed with RDRF = 1 (except the slave data receive operation in SSU mode) [Clearing condition] • When writing 0 after reading ORER = 1

Bit	Bit Name	Initial Value	R/W	Description
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	TEND	0	R/W	Transmit End [Setting conditions] • When the last bit of transmit data is transmitted while the TENDSTS bit in SSCR2 is cleared to 0 and the TDRE bit is set to 1 • After the last bit of transmit data is transmitted while the TENDSTS bit in SSCR2 is set to 1 and the TDRE bit is set to 1 [Clearing conditions] • When writing 0 after reading TEND = 1 • When writing data to SSTDR
2	TDRE	1	R/W	Transmit Data Empty Indicates whether or not SSTDR contains transmit data. [Setting conditions] • When the TE bit in SSER is 0 • When data is transferred from SSTDR to SSTRSR and SSTDR is ready to be written to [Clearing conditions] • When writing 0 after reading TDRE = 1 • When writing data to SSTDR with TE = 1 • When the DMAC is activated by an SSTXI interrupt and transmit data is written to SSTDR by the DMAC transfer
1	RDRF	0	R/W	Receive Data Full Indicates whether or not SSRDR contains receive data. [Setting condition] • When receive data is transferred from SSTRSR to SSRDR after successful serial data reception [Clearing conditions] • When writing 0 after reading RDRF = 1 • When reading receive data from SSRDR • When the DMAC is activated by an SSRXI interrupt and receive data is read from SSRDR by the DMAC transfer

Bit	Bit Name	Initial Value	R/W	Description
0	CE	0	R/W	Conflict/Incomplete Error Indicates that a conflict error has occurred when 0 is externally input to the $\overline{SCS}$ pin with SSUMS = 0 (SSU mode) and MSS = 1 (master mode). If the $\overline{SCS}$ pin level changes to 1 with SSUMS = 0 (SSU mode) and MSS = 0 (slave mode), an incomplete error occurs because it is determined that a master device has terminated the transfer. In reception as the slave device in SSU mode, received data (reading SSRDR) must be read out and RDRF in SSSR cleared before reception of the next frame starts. In transmission/reception as the slave device in SSU mode, the data for transmission must be written (writing to SSTDR) and TDRE in SSSR cleared before transmission of the next frame starts. If either condition is not met, an incomplete error will be generated at the end of that frame. Data reception does not continue while the CE bit is set to 1. Serial transmission also does not continue. Reset the SSU internal sequencer by setting the SRES bit in SSCRL to 1 before resuming transfer after incomplete error. [Setting conditions] • When a low level is input to the $\overline{SCS}$ pin in master mode (the MSS bit in SSCRH is set to 1) • When the $\overline{SCS}$ pin is changed to 1 during transfer in slave mode (the MSS bit in SSCRH is cleared to 0) • In reception as the slave device, following a frame in which reading of SSRDR and clearing of RDRF were not completed by time the next frame started, the end of the next frame. • In transmission as the slave device, following a frame in which writing to SSTDR and clearing of TDRE were not completed by time the next frame started, the end of the next frame [Clearing condition] • When writing 0 after reading CE = 1

17.3.6 SS Control Register 2 (SSCR2)

SSCR2 is a register that selects the assert timing of the $\overline{SCS}$ pin, data output timing of the SSO pin, and set timing of the TEND bit.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	TENDSTS	SCSATS	SSODTS	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R	R

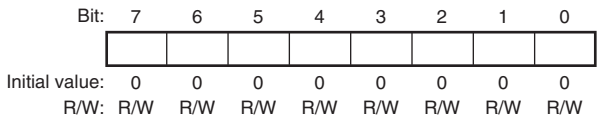
Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TENDSTS	0	R/W	Selects the timing of setting the TEND bit (valid in SSU and master mode). 0: Sets the TEND bit when the last bit is being transmitted 1: Sets the TEND bit after the last bit is transmitted
3	SCSATS	0	R/W	Selects the assertion timing of the $\overline{SCS}$ pin (valid in SSU and master mode). 0: Min. values of t_{LEAD} and t_{LAG} are $1/2 \times t_{SUcyc}$ 1: Min. values of t_{LEAD} and t_{LAG} are $3/2 \times t_{SUcyc}$
2	SSODTS	0	R/W	Selects the data output timing of the SSO pin (valid in SSU and master mode) 0: While BIDE = 0, MSS = 1, and TE = 1 or while BIDE = 1, TE = 1, and RE = 0, the SSO pin outputs data 1: While BIDE = 0, MSS = 1, and TE = 1 or while BIDE = 1, TE = 1, and RE = 0, the SSO pin outputs data while the $\overline{SCS}$ pin is driven low
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

17.3.7 SS Transmit Data Registers 0 to 3 (SSTDR0 to SSTDR3)

SSTDR is an 8-bit register that stores transmit data. When 8-bit data length is selected by bits DATS1 and DATS0 in SSCRL, SSTDR0 is valid. When 16-bit data length is selected, SSTDR0 and SSTDR1 are valid. When 32-bit data length is selected, SSTDR0 to SSTDR3 are valid. The SSTDR that has not been enabled must not be accessed.

When the SSU detects that SSTRSR is empty, it transfers the transmit data written in SSTDR to SSTRSR and starts serial transmission. If the next transmit data has already been written to SSTDR during serial transmission, the SSU performs consecutive serial transmission.

Although SSTDR can always be read from or written to by the CPU and DMAC, to achieve reliable serial transmission, write transmit data to SSTDR after confirming that the TDRE bit in SSSR is set to 1.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0		All 0	R/W	Serial transmit data

Table 17.3 Correspondence between the DATS Bit Setting and SSTDR

	DATS[1:0] (SSCRL[1:0])			
SSTDR	**00**	**01**	**10**	**11 (Setting Disabled)**
0	Valid	Valid	Valid	Invalid
1	Invalid	Valid	Valid	Invalid
2	Invalid	Invalid	Valid	Invalid
3	Invalid	Invalid	Valid	Invalid

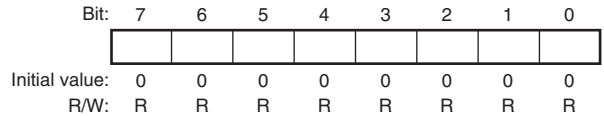
17.3.8 SS Receive Data Registers 0 to 3 (SSRDR0 to SSRDR3)

SSRDR is an 8-bit register that stores receive data. When 8-bit data length is selected by bits DATS1 and DATS0 in SSCRL, SSRDR0 is valid. When 16-bit data length is selected, SSRDR0 and SSRDR1 are valid. When 32-bit data length is selected, SSRDR0 to SSRDR3 are valid. The SSRDR that has not been enabled must not be accessed.

When the SSU has received 1-byte data, it transfers the received serial data from SSTRSR to SSRDR where it is stored. After this, SSTRSR is ready for reception. Since SSTRSR and SSRDR function as a double buffer in this way, consecutive receive operations can be performed.

Read SSRDR after confirming that the RDRF bit in SSSR is set to 1.

SSRDR is a read-only register, therefore, cannot be written to by the CPU.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0		All 0	R	Serial receive data

Table 17.4 Correspondence between DATS Bit Setting and SSRDR

	DATS[1:0] (SSCRL[1:0])			
SSRDR	**00**	**01**	**10**	**11 (Setting Disabled)**
0	Valid	Valid	Valid	Invalid
1	Invalid	Valid	Valid	Invalid
2	Invalid	Invalid	Valid	Invalid
3	Invalid	Invalid	Valid	Invalid

17.3.9 SS Shift Register (SSTRSR)

SSTRSR is a shift register that transmits and receives serial data.

When data is transferred from SSTDR to SSTRSR, bit 0 of transmit data is bit 0 in the SSTDR contents (MLS = 0: LSB first communication) and is bit 7 in the SSTDR contents (MLS = 1: MSB first communication). The SSU transfers data from the LSB (bit 0) in SSTRSR to the SSO pin to perform serial data transmission.

In reception, the SSU sets serial data that has been input via the SSI pin in SSTRSR from the LSB (bit 0). When 1-byte data has been received, the SSTRSR contents are automatically transferred to SSRDR. SSTRSR cannot be directly accessed by the CPU.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

17.4 Operation

17.4.1 Transfer Clock

A transfer clock can be selected from among seven internal clocks and an external clock. Before using this module, enable the SSCK pin function in the PFC. When the MSS bit in SSCRH is 1, an internal clock is selected and the SSCK pin is used as an output pin. When transfer is started, the clock with the transfer rate set by bits CKS2 to CKS0 in SSMR is output from the SSCK pin. When MSS = 0, an external clock is selected and the SSCK pin is used as an input pin.

17.4.2 Relationship of Clock Phase, Polarity, and Data

The relationship of clock phase, polarity, and transfer data depends on the combination of the CPOS and CPHS bits in SSMR when the value of the SSUMS bit in SSCRL is 0. Figure 17.2 shows the relationship. When SSUMS = 1, the CPHS setting is invalid although the CPOS setting is valid. Transmit data change timing and receive data fetch timing in SSUMS = 1 are the same timings shown in figure 17.2, (1) When CPHS = 0.

Setting the MLS bit in SSMR selects either MSB first or LSB first communication. When MLS = 0, data is transferred from the LSB to the MSB. When MLS = 1, data is transferred from the MSB to the LSB.

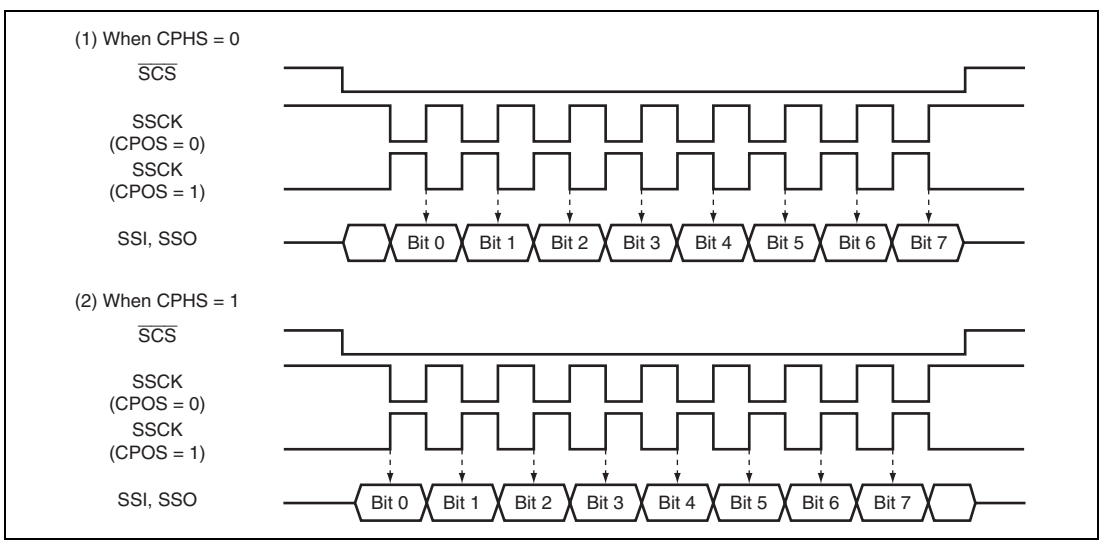


Figure 17.2 Relationship of Clock Phase, Polarity, and Data

17.4.3 Relationship between Data Input/Output Pins and Shift Register

The connection between data input/output pins and the SS shift register (SSTRSR) depends on the combination of the MSS and BIDE bits in SSCRH and the SSUMS bit in SSCRL. Figure 17.3 shows the relationship.

The SSU transmits serial data from the SSO pin and receives serial data from the SSI pin when operating with BIDE = 0 and MSS = 1 (standard, master mode) (see figure 17.3 (1)). The SSU transmits serial data from the SSI pin and receives serial data from the SSO pin when operating with BIDE = 0 and MSS = 0 (standard, slave mode) (see figure 17.3 (2)).

The SSU transmits and receives serial data from the SSO pin regardless of master or slave mode when operating with BIDE = 1 (bidirectional mode) (see figures 17.3 (3) and (4)).

However, even if both the TE and RE bits are set to 1, transmission and reception are not performed simultaneously. Either the TE or RE bit must be selected.

The SSU transmits serial data from the SSO pin and receives serial data from the SSI pin when operating with SSUMS = 1. The SSCK pin outputs the internal clock when MSS = 1 and function as an input pin when MSS = 0 (see figures 17.3 (5) and (6)).

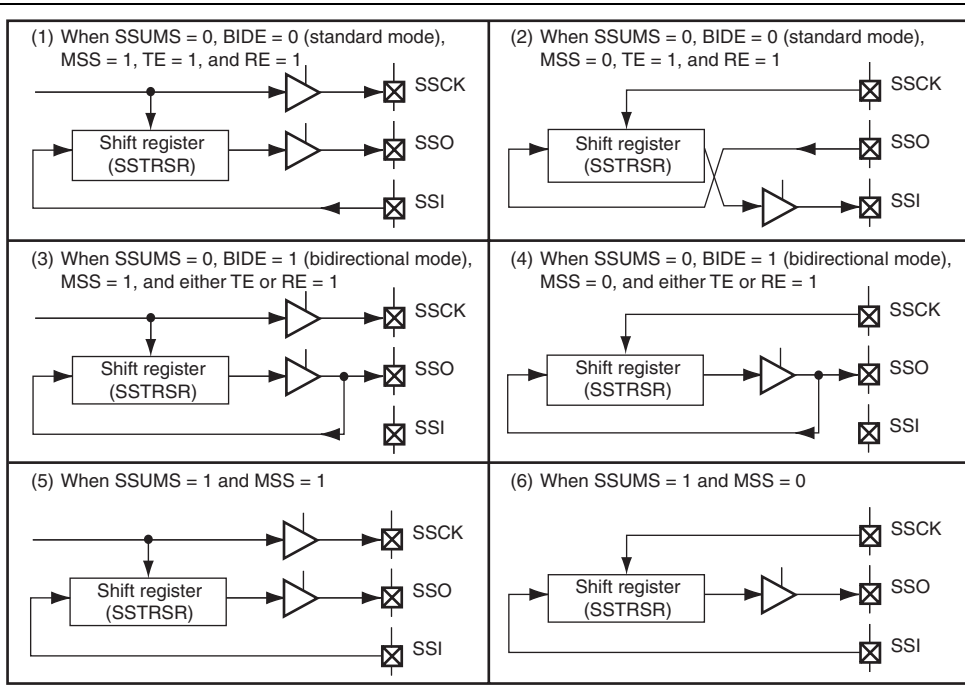


Figure 17.3 Relationship between Data Input/Output Pins and the Shift Register

17.4.4 Communication Modes and Pin Functions

The SSU switches the input/output pin (SSI, SSO, SSCK, and $\overline{SCS}$) functions according to the communication modes and register settings. The relationship of communication modes and input/output pin functions are shown in tables 17.5 to 17.7.

Table 17.5 Communication Modes and Pin States of SSI and SSO Pins

Communication Mode	Register Setting					Pin State	
	SSUMS	BIDE	MSS	TE	RE	SSI	SSO
SSU communication mode	0	0	0	0	1	—	Input
				1	0	Output	—
					1	Output	Input
			1	0	1	Input	—
				1	0	—	Output
					1	Input	Output
SSU (bidirectional) communication mode	0	1	0	0	1	—	Input
				1	0	—	Output
			1	0	1	—	Input
				1	0	—	Output
Clock synchronous communication mode	1	0	0	0	1	Input	—
				1	0	—	Output
					1	Input	Output
			1	0	1	Input	—
				1	0	—	Output
					1	Input	Output

[Legend]

—: Not used as SSU pin (but can be used as an I/O port)

Table 17.6 Communication Modes and Pin States of SSCK Pin

Communication Mode	Register Setting		Pin State
	SSUMS	MSS	SSCK
SSU communication mode	0	0	Input
		1	Output
Clock synchronous communication mode	1	0	Input
		1	Output

Table 17.7 Communication Modes and Pin States of $\overline{\text{SCS}}$ Pin

Communication Mode	Register Setting				Pin State
	SSUMS	MSS	CSS1	CSS0	$\overline{\text{SCS}}$
SSU communication mode	0	0	x	x	Input
		1	0	0	(Setting prohibited)
			0	1	(Setting prohibited)
			1	0	Automatic input/output
			1	1	Output
Clock synchronous communication mode	1	x	x	x	—

[Legend]

x: Don't care

—: Not used as SSU pin (but can be used as an I/O port)

17.4.5 SSU Mode

In SSU mode, data communications are performed via four lines: clock line (SSCK), data input line (SSI or SSO), data output line (SSI or SSO), and chip select line ($\overline{SCS}$).

In addition, the SSU supports bidirectional mode in which a single pin functions as data input and data output lines.

(1) Initial Settings in SSU Mode

Figure 17.4 shows an example of the initial settings in SSU mode. Before data transfer, clear both the TE and RE bits in SSER to 0 to set the initial values.

Note: Before changing operating modes and communications formats, clear both the TE and RE bits to 0. Although clearing the TE bit to 0 sets the TDRE bit to 1, clearing the RE bit to 0 does not change the values of the RDRF and ORER bits and SSRDR. Those bits retain the previous values.

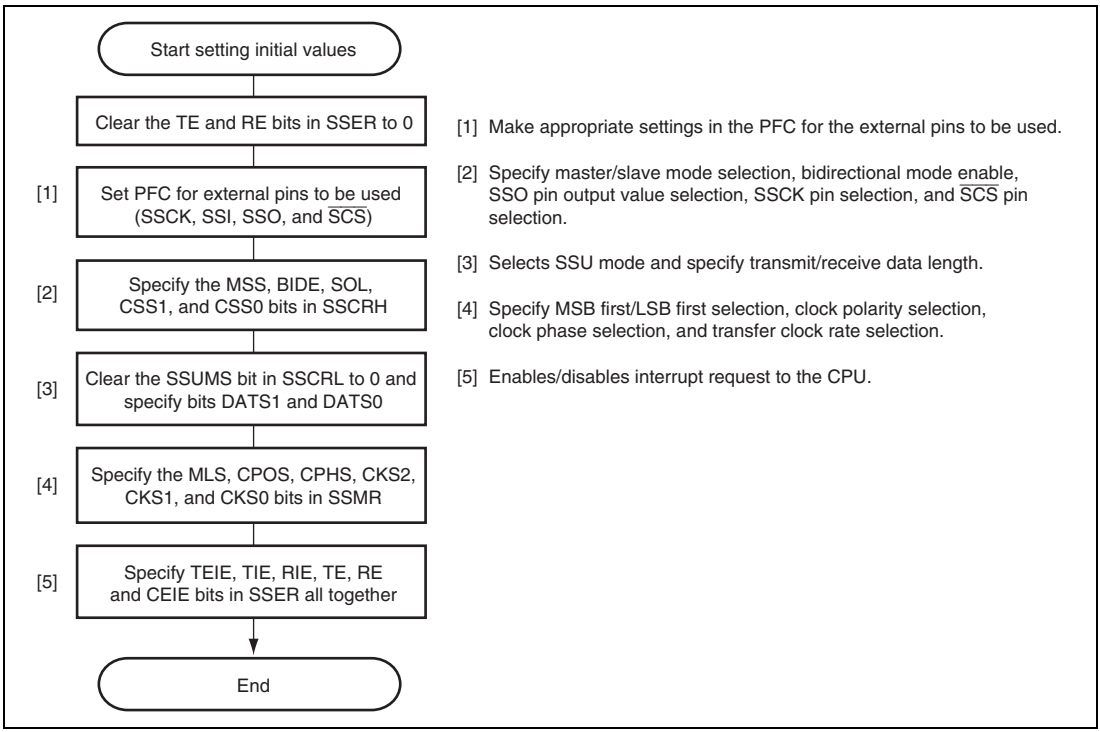


Figure 17.4 Example of Initial Settings in SSU Mode

(2) Data Transmission

Figure 17.5 shows an example of transmission operation, and figure 17.6 shows a flowchart example of data transmission.

When transmitting data, the SSU operates as shown below.

In master mode, the SSU outputs a transfer clock and data. In slave mode, when a low level signal is input to the $\overline{\text{SCS}}$ pin and a transfer clock is input to the SSCK pin, the SSU outputs data in synchronization with the transfer clock.

Writing transmit data to SSTDR after the TE bit is set to 1 clears the TDRE bit in SSSR to 0, and the SSTDR contents are transferred to SSTRSR. After that, the SSU sets the TDRE bit to 1 and starts transmission. At this time, if the TIE bit in SSER is set to 1, an SSTXI interrupt in the transmit data empty state is generated.

When 1-frame data has been transferred with TDRE = 0, the SSTDR contents are transferred to SSTRSR to start the next frame transmission. When the 8th bit of transmit data has been transferred with TDRE = 1, the TEND bit in SSSR is set to 1 and the state is retained. At this time, if the TEIE bit is set to 1, an SSTXI interrupt at the end of transmission is generated. After transmission, the output level of the SSCK pin is fixed high when CPOS = 0 and low when CPOS = 1.

While the ORER bit in SSSR is set to 1, transmission is not performed. Check that the ORER bit is cleared to 0 before transmission.

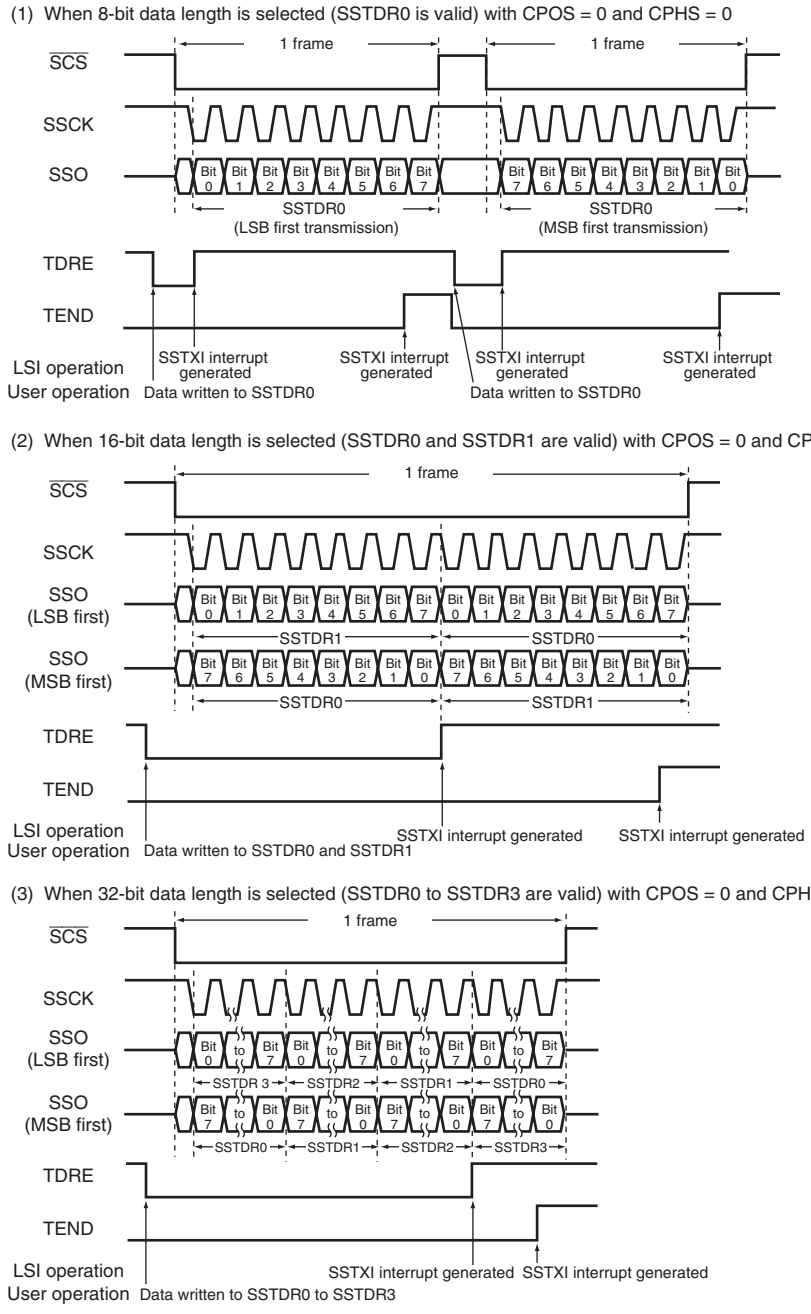
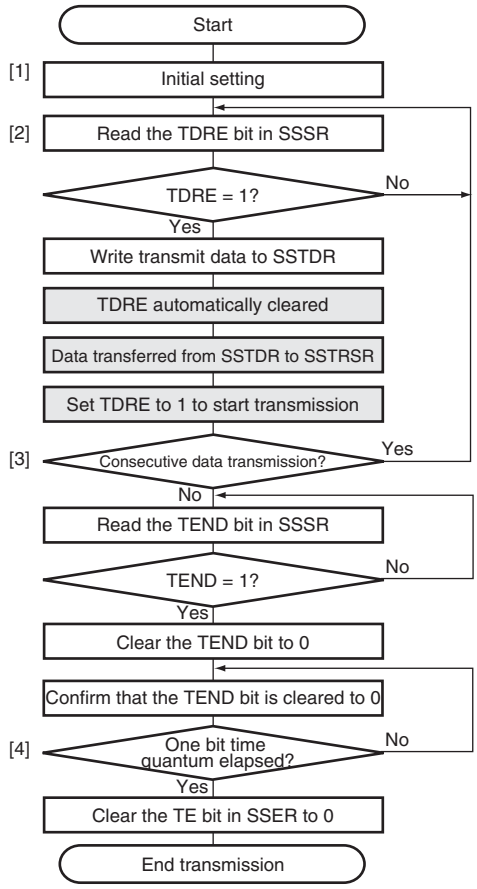


Figure 17.5 Example of Transmission Operation (SSU Mode)



[1] Initial setting:
Specify the transmit data format.

[2] Check that the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit is 1. The TDRE bit is automatically cleared to 0 and transmission is started by writing data to SSTDR.

[3] Procedure for consecutive data transmission:
To continue data transmission, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

[4] Procedure for data transmission end:
To end data transmission, confirm that the TEND bit is cleared to 0. After completion of transmitting the last bit, clear the TE bit to 0.

Note: Hatching boxes represent SSU internal operations.

Figure 17.6 Flowchart Example of Data Transmission (SSU Mode)

(3) Data Reception

Figure 17.7 shows an example of reception operation, and figure 17.8 shows a flowchart example of data reception. When receiving data, the SSU operates as shown below.

After setting the RE bit to 1 and dummy-reading SSRDR, the SSU starts data reception.

In master mode, the SSU outputs a transfer clock and receives data. In slave mode, when a low level signal is input to the $\overline{\text{SCS}}$ pin and a transfer clock is input to the SSCK pin, the SSU receives data in synchronization with the transfer clock.

When 1-frame data has been received, the RDRF bit in SSSR is set to 1 and the receive data is stored in SSRDR. At this time, if the RIE bit in SSER is set to 1, an SSRXI interrupt is generated. The RDRF bit is automatically cleared to 0 by reading SSRDR.

In continuous reception as the slave device in SSU mode, be sure to read the SS receive data register (SSRDR) before reception of the next frame starts (before the externally connected master device starts the next transmission). If reception of a next frame starts while the receive data full (RDRF) bit in the SS status register (SSSR) is set (to 1) because SSRDR has not yet been read, and SSRDR is then read before reception of the next frame is complete, the conflict/incomplete error (CE) bit in SSRDR will be set at the end of reception of the next frame. Furthermore, if reception of a next frame starts after RDRF has been set (to 1) but before SSRDR has been read, and SSRDR still has not been read by the end of reception of the next frame, neither the CE nor the overrun error (ORER) bit in SSSR will be set, but the received data will be discarded.

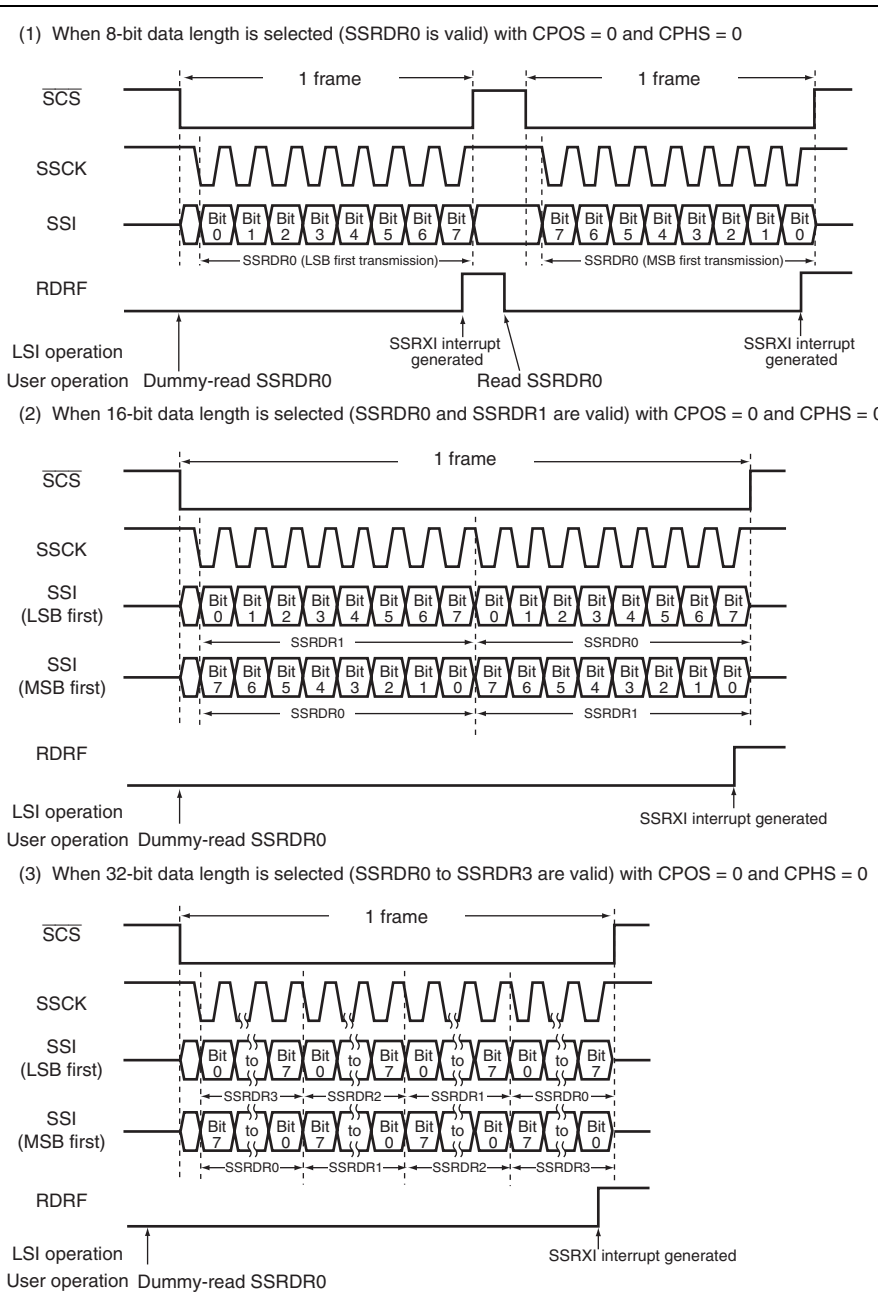
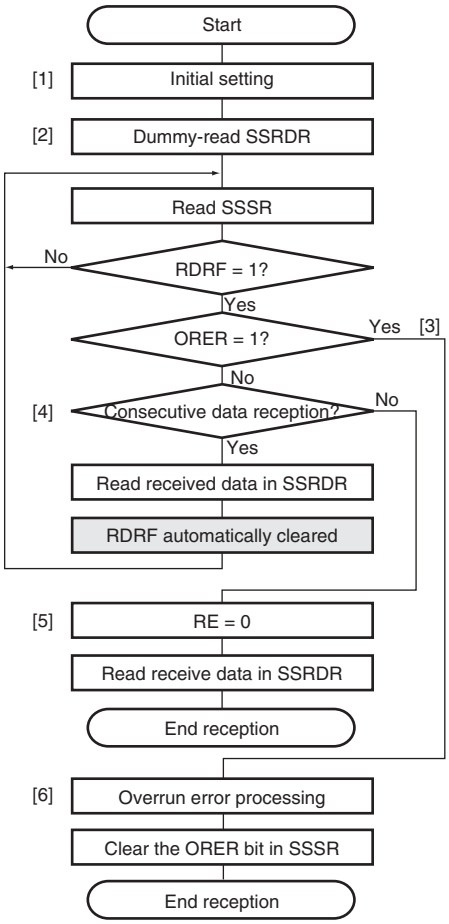


Figure 17.7 Example of Reception Operation (SSU Mode)



[1] Initial setting:
Specify the receive data format.

[2] Start reception:
When SSRDR is dummy-read with RE = 1, reception is started.

[3], [6] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[4] To continue single reception:
When continuing single reception, wait for time of t_{SUcyc} while the RDRF flag is set to 1 and then read receive data in SSRDR.
The next single reception starts after reading receive data in SSRDR.

[5] To complete reception:
To complete reception, read receive data after clearing the RE bit to 0. When reading SSRDR without clearing the RE bit, reception is resumed.

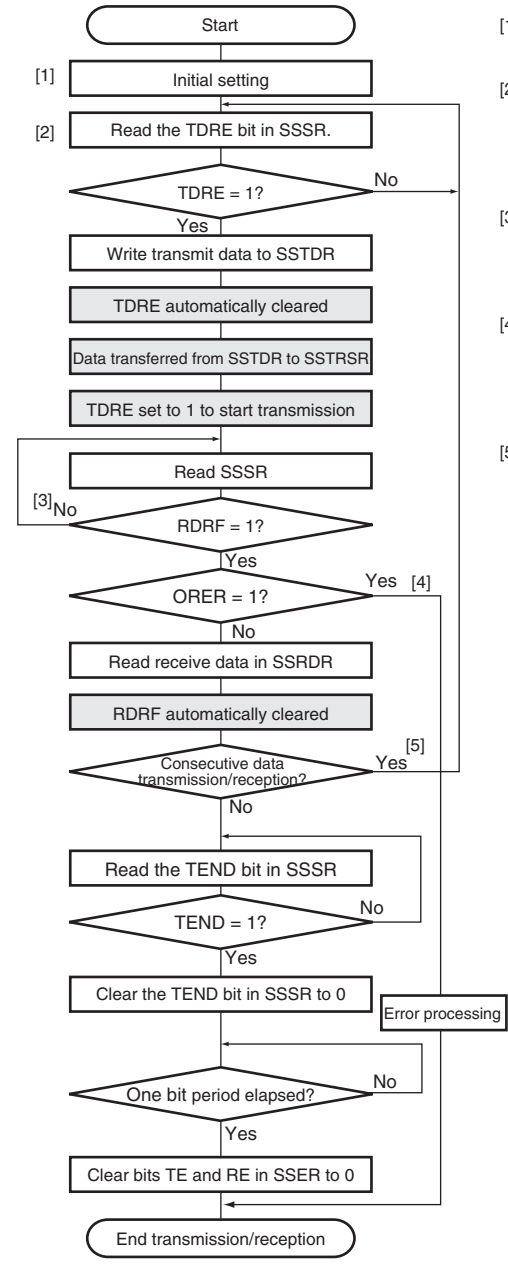
Note: Hatching boxes represent SSU internal operations.

Figure 17.8 Flowchart Example of Data Reception (SSU Mode)

(4) Data Transmission/Reception

Figure 17.9 shows a flowchart example of simultaneous transmission/reception. The data transmission/reception is performed combining the data transmission and data reception as mentioned above. The data transmission/reception is started by writing transmit data to SSTDR with TE = RE = 1. When the RDRF bit has been set to 1 at the 8th rising edge of the transfer clock, the ORER bit in SSSR is set to 1. This indicates that an overrun error (SSERI) has occurred, at this time, the data transmission/reception is stopped. While the ORER bit in SSSR is set to 1, transmission/reception is not performed. To resume the transmission/reception, clear the ORER bit to 0.

Before switching transmission mode (TE = 1) or reception mode (RE = 1) to transmission/reception mode (TE = RE = 1), clear the TE and RE bits to 0. When starting the transfer, confirm that the TEND, RDRF, and ORER bits are cleared to 0 before setting the TE or RE bit to 1.



Note: Hatching boxes represent SSU internal operations.

[1] Initial setting:
Specify the transmit/receive data format.

[2] Check the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit in SSSR is 1. The TDRE bit is automatically cleared to 0 and transmission/reception is started by writing data to SSTDR.

[3] Check the SSU state:
Read SSSR confirming that the RDRF bit is 1. A change of the RDRF bit (from 0 to 1) can be notified by RXI interrupt.

[4] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[5] Procedure for consecutive data transmission/reception:
To continue serial data transmission/reception, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

Figure 17.9 Flowchart Example of Simultaneous Transmission/Reception (SSU Mode)

17.4.6 $\overline{\text{SCS}}$ Pin Control and Conflict Error

When bits CSS1 and CSS0 in SSCRH are specified to B'10 and the SSUMS bit in SSCRL is cleared to 0, the $\overline{\text{SCS}}$ pin functions as an input (Hi-Z) to detect a conflict error. A conflict error detection period is from setting the MSS bit in SSCRH to 1 to starting serial transfer and after transfer ends. When a low level signal is input to the $\overline{\text{SCS}}$ pin within the period, a conflict error occurs. At this time, the CE bit in SSSR is set to 1 and the MSS bit is cleared to 0.

Note: While the CE bit is set to 1, transmission or reception is not resumed. Clear the CE bit to 0 before resuming the transmission or reception.

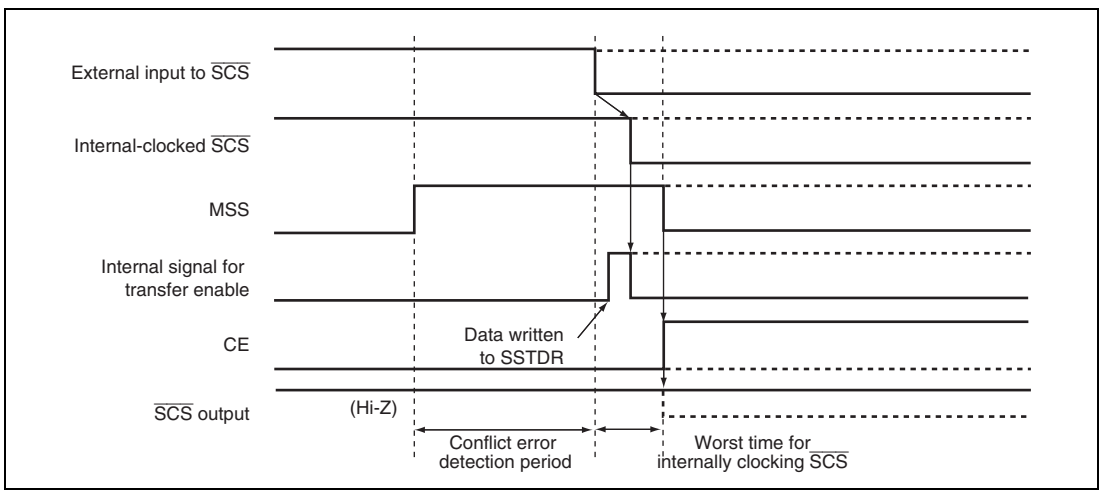


Figure 17.10 Conflict Error Detection Timing (Before Transfer)

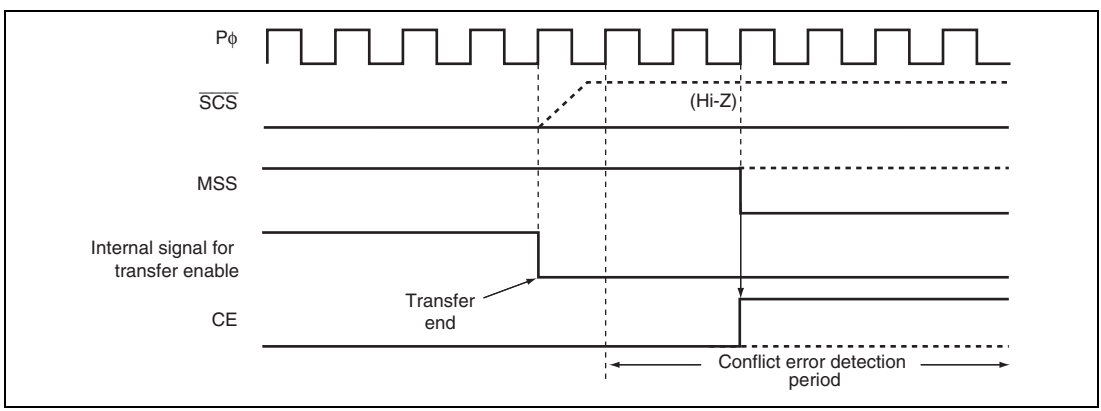


Figure 17.11 Conflict Error Detection Timing (After Transfer End)

17.4.7 Clock Synchronous Communication Mode

In clock synchronous communication mode, data communications are performed via three lines: clock line (SSCK), data input line (SSI), and data output line (SSO).

(1) Initial Settings in Clock Synchronous Communication Mode

Figure 17.12 shows an example of the initial settings in clock synchronous communication mode. Before data transfer, clear both the TE and RE bits in SSER to 0 to set the initial values.

Note: Before changing operating modes and communications formats, clear both the TE and RE bits to 0. Although clearing the TE bit to 0 sets the TDRE bit to 1, clearing the RE bit to 0 does not change the values of the RDRF and ORER bits and SSRDR. Those bits retain the previous values.

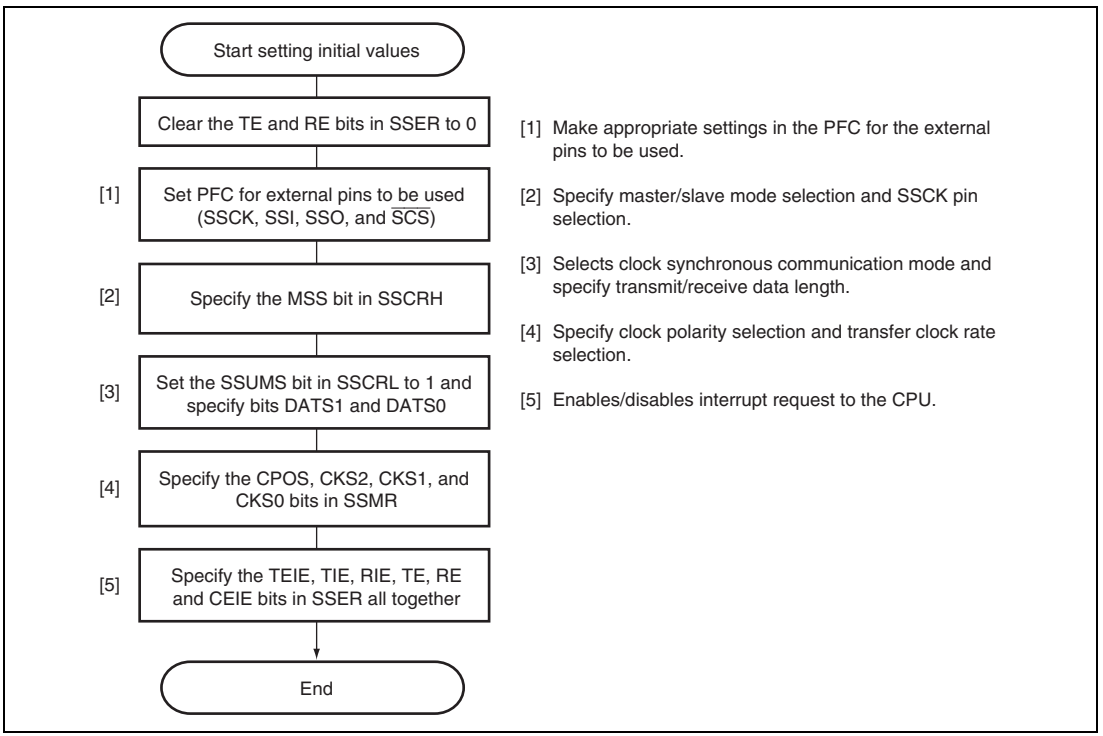


Figure 17.12 Example of Initial Settings in Clock Synchronous Communication Mode

(2) Data Transmission

Figure 17.13 shows an example of transmission operation, and figure 17.14 shows a flowchart example of data transmission. When transmitting data in clock synchronous communication mode, the SSU operates as shown below.

In master mode, the SSU outputs a transfer clock and data. In slave mode, when a transfer clock is input to the SSCK pin, the SSU outputs data in synchronization with the transfer clock.

Writing transmit data to SSTDR after the TE bit is set to 1 clears the TDRE bit in SSSR to 0, and the SSTDR contents are transferred to SSTRSR. After that, the SSU sets the TDRE bit to 1 and starts transmission. At this time, if the TIE bit in SSER is set to 1, an SSTXI interrupt in the transmit data empty state is generated.

When 1-frame data has been transferred with TDRE = 0, the SSTDR contents are transferred to SSTRSR to start the next frame transmission. When the 8th bit of transmit data has been transferred with TDRE = 1, the TEND bit in SSSR is set to 1 and the state is retained. At this time, if the TEIE bit is set to 1, an SSTXI interrupt at the end of transmission is generated.

While the ORER bit in SSSR is set to 1, transmission is not performed. Check that the ORER bit is cleared to 0 before transmission.

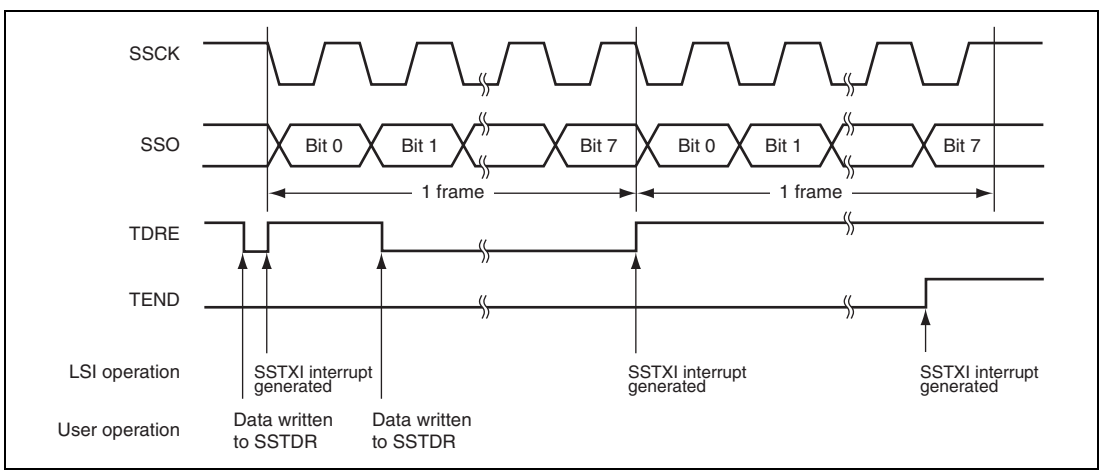
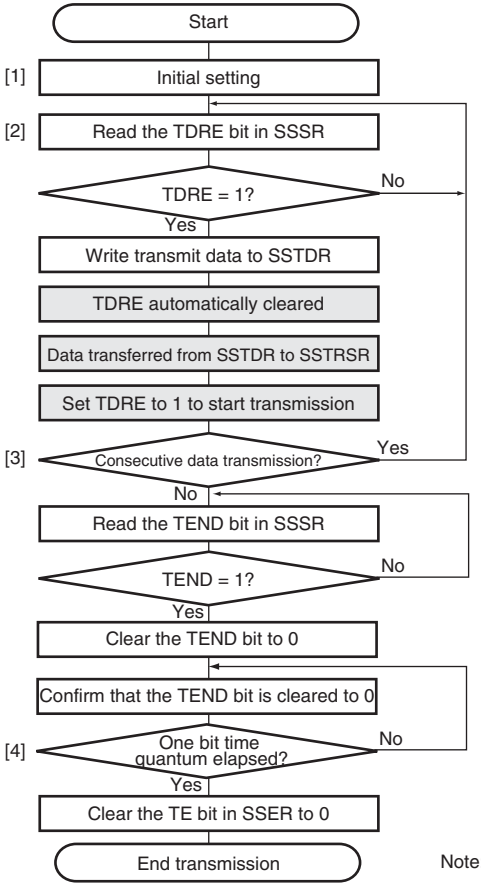


Figure 17.13 Example of Transmission Operation (Clock Synchronous Communication Mode)



[1] Initial setting:
Specify the transmit data format.

[2] Check that the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit is 1. The TDRE bit is automatically cleared to 0 and transmission is started by writing data to SSTDR.

[3] Procedure for consecutive data transmission:
To continue data transmission, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

[4] Procedure for data transmission end:
To end data transmission, confirm that the TEND bit is cleared to 0. After completion of transmitting the last bit, clear the TE bit to 0.

Note: Hatching boxes represent SSU internal operations.

Figure 17.14 Flowchart Example of Transmission Operation (Clock Synchronous Communication Mode)

(3) Data Reception

Figure 17.15 shows an example of reception operation, and figure 17.16 shows a flowchart example of data reception. When receiving data, the SSU operates as shown below.

After setting the RE bit in SSER to 1, the SSU starts data reception.

In master mode, the SSU outputs a transfer clock and receives data. In slave mode, when a transfer clock is input to the SSCK pin, the SSU receives data in synchronization with the transfer clock.

When 1-frame data has been received, the RDRF bit in SSSR is set to 1 and the receive data is stored in SSRDR. At this time, if the RIE bit is set to 1, an SSRXI interrupt is generated. The RDRF bit is automatically cleared to 0 by reading SSRDR.

When the RDRF bit has been set to 1 at the 8th rising edge of the transfer clock, the ORER bit in SSSR is set to 1. This indicates that an overrun error (SSERI) has occurred. At this time, data reception is stopped. While the ORER bit in SSSR is set to 1, reception is not performed. To resume the reception, clear the ORER bit to 0.

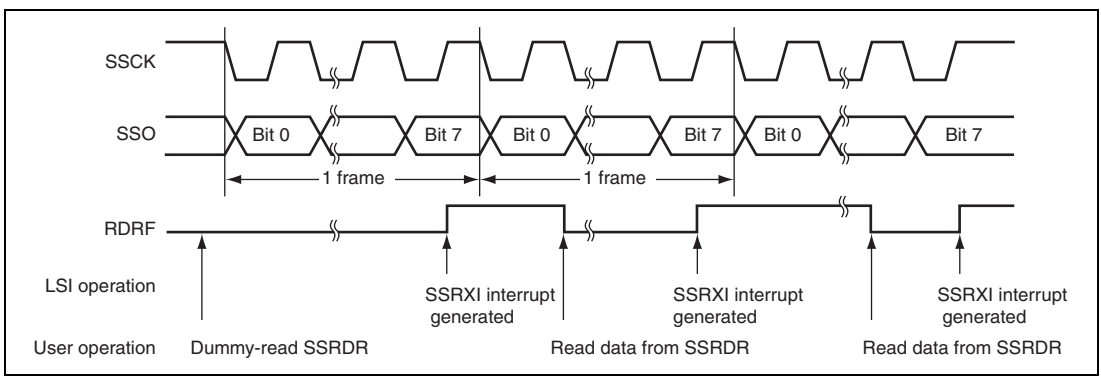
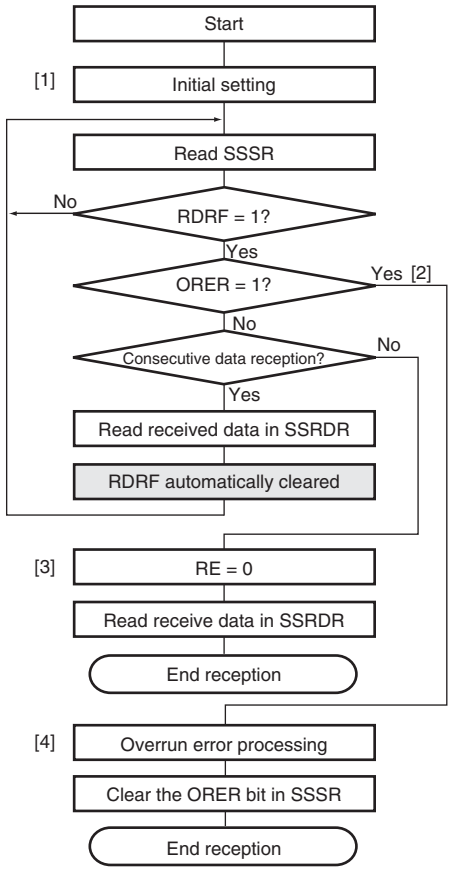


Figure 17.15 Example of Reception Operation (Clock Synchronous Communication Mode)



[1] Initial setting:
Specify the receive data format.

[2], [4] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[3] To complete reception:
To complete reception, read receive data after clearing the RE bit to 0. When reading SSRDR without clearing the RE bit, reception is resumed.

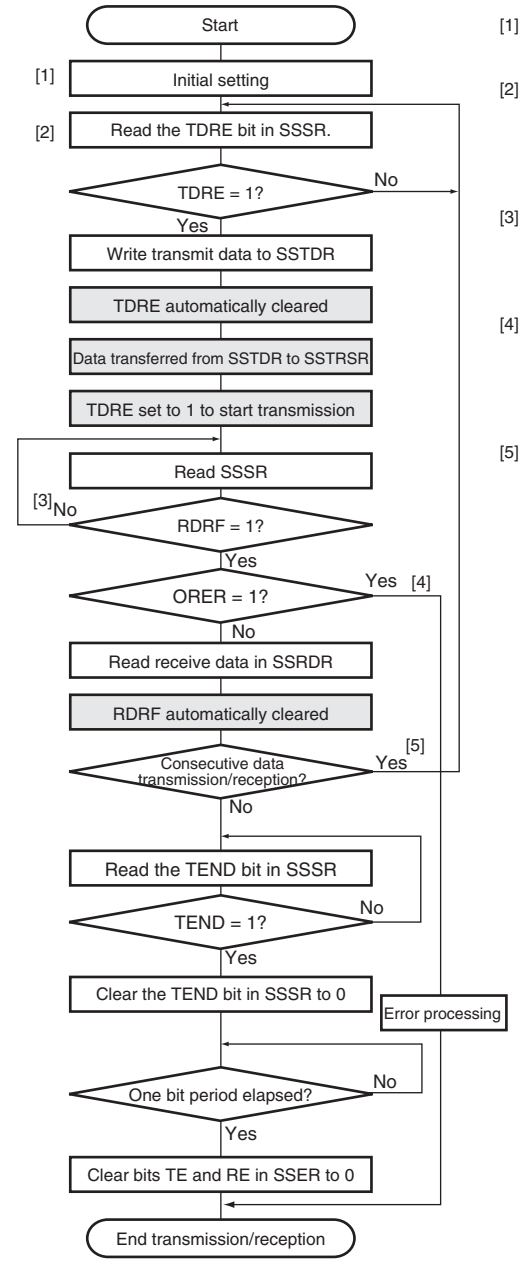
Note: Hatching boxes represent SSU internal operations.

Figure 17.16 Flowchart Example of Data Reception (Clock Synchronous Communication Mode)

(4) Data Transmission/Reception

Figure 17.17 shows a flowchart example of simultaneous transmission/reception. The data transmission/reception is performed combining the data transmission and data reception as mentioned above. The data transmission/reception is started by writing transmit data to SSTDR with TE = RE = 1. When the RDRF bit has been set to 1 at the 8th rising edge of the transfer clock, the ORER bit in SSSR is set to 1. This indicates that an overrun error (SSERI) has occurred. At this time, data transmission/reception is stopped. While the ORER bit in SSSR is set to 1, transmission/reception is not performed. To resume the transmission/reception, clear the ORER bit to 0.

Before switching transmission mode (TE = 1) or reception mode (RE = 1) to transmission/reception mode (TE = RE = 1), clear the TE and RE bits to 0. When starting the transfer, confirm that the TEND, RDRF, and ORER bits are cleared to 0 before setting the TE or RE bits to 1.



Note: Hatching boxes represent SSU internal operations.

[1] Initial setting:
Specify the transmit/receive data format.

[2] Check the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit in SSSR is 1. The TDRE bit is automatically cleared to 0 and transmission/reception is started by writing data to SSTDR.

[3] Check the SSU state:
Read SSSR confirming that the RDRF bit is 1. A change of the RDRF bit (from 0 to 1) can be notified by RXI interrupt.

[4] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[5] Procedure for consecutive data transmission/reception:
To continue serial data transmission/reception, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

Figure 17.17 Flowchart Example of Simultaneous Transmission/Reception (Clock Synchronous Communication Mode)

17.5 SSU Interrupt Sources and DMAC

The SSU interrupt requests are an overrun error, a conflict error, a receive data register full, transmit data register empty, and a transmit end interrupts. Of these interrupt sources, a receive data register full, and a transmit data register empty can activate the DMAC for data transfer.

Since both an overrun error and a conflict error interrupts are allocated to the SSERI vector address, and both a transmit data register empty and a transmit end interrupts are allocated to the SSTXI vector address, the interrupt source should be decided by their flags. Table 17.8 lists the interrupt sources.

When an interrupt condition shown in table 17.8 is satisfied, an interrupt is requested. Clear the interrupt source by CPU or DMAC data transfer.

Table 17.8 SSU Interrupt Sources

Abbreviation	Interrupt Source	Interrupt Condition	DMAC Activation
SSERI	Overrun error	(RIE = 1) • (ORER = 1) + (CEIE = 1) • (CE = 1)	—
	Conflict error		
SSRXI	Receive data register full	(RIE = 1) • (RDRF = 1)	Possible
SSTXI	Transmit data register empty	(TIE = 1) • (TDRE = 1) + (TEIE = 1) • (TEND = 1)	Possible
	Transmit end		

17.6 Usage Note

17.6.1 Module Standby Mode Setting

The SSU operation can be disabled or enabled using the standby control register. The initial setting is for SSU operation to be halted. Access to registers is enabled by clearing module standby mode. For details, refer to section 30, Power-Down Modes.

17.6.2 Consecutive Data Transmission/Reception in SSU Slave Mode

In the continuous reception or transmission of data in SSU slave mode, the $\overline{\text{SCS}}$ pin is negated (placed at the high level) after every frame. If the $\overline{\text{SCS}}$ pin remains asserted (low level) over a period longer than that for one frame, transmission or reception was not correct.

17.6.3 Note in the Master Transmission Operation or the Master Transmission/Reception Operation of SSU Mode

In the master transmission operation or the master transmission/reception operation of SSU mode, please operate one of the following three ways.

(1) Write the next transmission data to the SSTDR after the TDRE bit of the SSSR is set to 1 and before the transmission of one bit before the last bit starts.

(2) Write the next transmission data to the SSTDR after the TEND bit of the SSSR is set to 1.

(3) Set the SSCR2 as “TENDSTS = 0” or “TENDSTS = 1 and SCSATS = 1”.

Section 18 I²C Bus Interface 3 (IIC3)

The I²C bus interface 3 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.

The I²C bus interface 3 has four channels.

18.1 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 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
- The direct memory access controller (DMAC) can be activated by a transmit-data-empty request or receive-data-full request to transfer data.
- Direct bus drive

  Two pins, SCL and SDA pins, function as NMOS open-drain outputs when the bus drive function is selected.

Clocked synchronous serial format:

- Four interrupt sources

  Transmit-data-empty, transmit-end, receive-data-full, and overrun error
- The direct memory access controller (DMAC) can be activated by a transmit-data-empty request or receive-data-full request to transfer data.

Figure 18.1 shows a block diagram of the I²C bus interface 3.

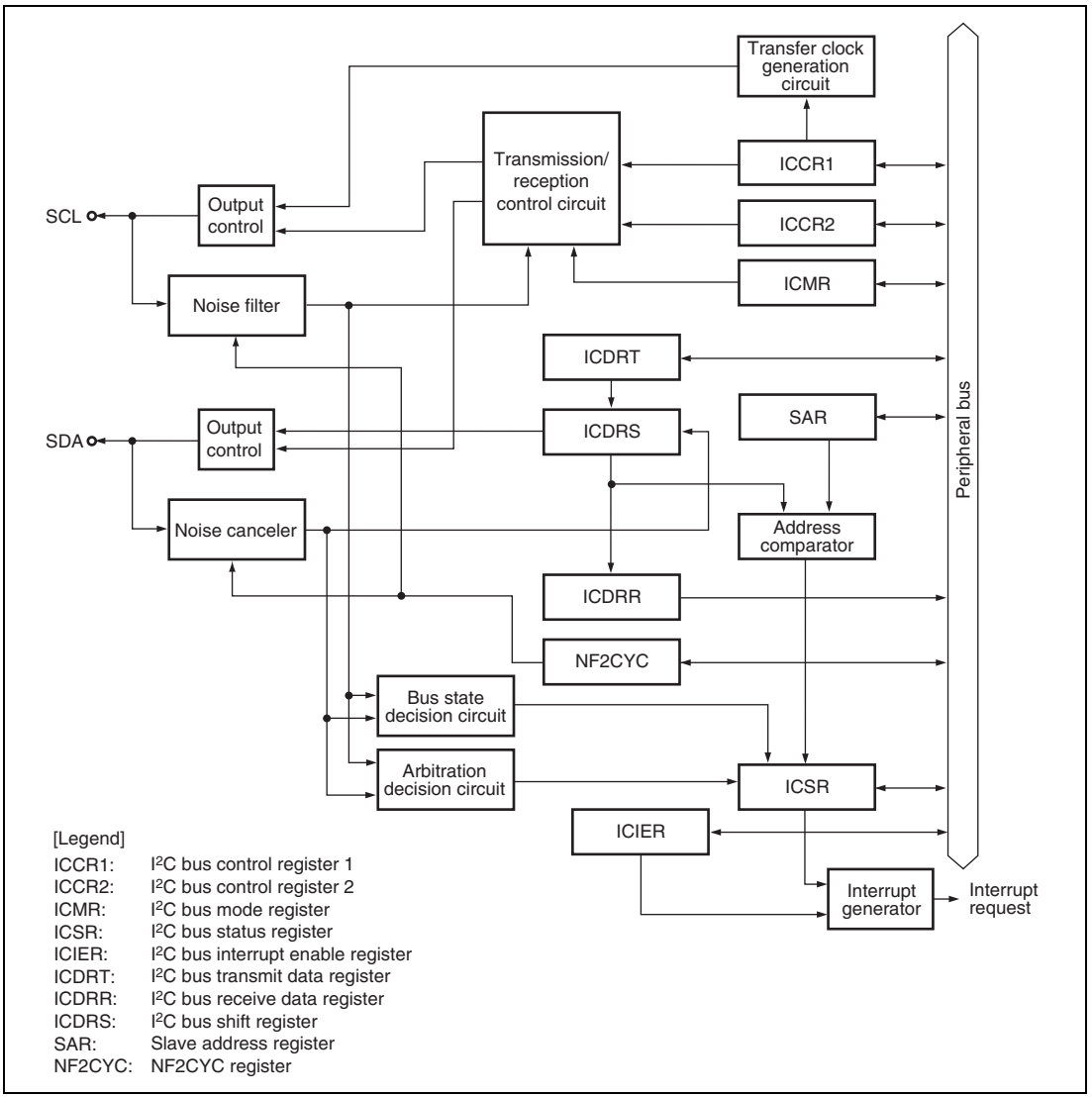


Figure 18.1 Block Diagram of I²C Bus Interface 3

18.2 Input/Output Pins

Table 18.1 shows the pin configuration of the I²C bus interface 3.

Table 18.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Serial clock	SCL0 to SCL3	I/O	I²C serial clock input/output
Serial data	SDA0 to SDA3	I/O	I²C serial data input/output

Figure 18.2 shows an example of I/O pin connections to external circuits.

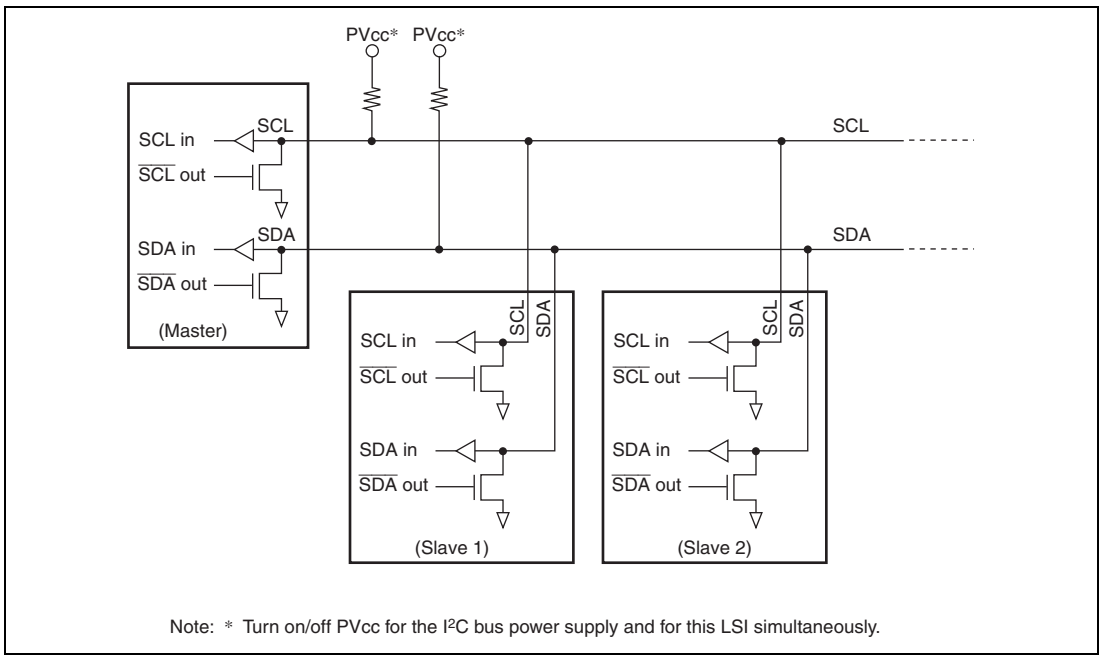


Figure 18.2 External Circuit Connections of I/O Pins

18.3 Register Descriptions

The I²C bus interface 3 has the following registers.

Table 18.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	I²C bus control register 1	ICCR1_0	R/W	H'00	H'FFFEE000	8
	I²C bus control register 2	ICCR2_0	R/W	H'7D	H'FFFEE001	8
	I²C bus mode register	ICMR_0	R/W	H'38	H'FFFEE002	8
	I²C bus interrupt enable register	ICIER_0	R/W	H'00	H'FFFEE003	8
	I²C bus status register	ICSR_0	R/W	H'00	H'FFFEE004	8
	Slave address register	SAR_0	R/W	H'00	H'FFFEE005	8
	I²C bus transmit data register	ICDRT_0	R/W	H'FF	H'FFFEE006	8
	I²C bus receive data register	ICDRR_0	R/W	H'FF	H'FFFEE007	8
	NF2CYC register	NF2CYC_0	R/W	H'00	H'FFFEE008	8
1	I²C bus control register 1	ICCR1_1	R/W	H'00	H'FFFEE400	8
	I²C bus control register 2	ICCR2_1	R/W	H'7D	H'FFFEE401	8
	I²C bus mode register	ICMR_1	R/W	H'38	H'FFFEE402	8
	I²C bus interrupt enable register	ICIER_1	R/W	H'00	H'FFFEE403	8
	I²C bus status register	ICSR_1	R/W	H'00	H'FFFEE404	8
	Slave address register	SAR_1	R/W	H'00	H'FFFEE405	8
	I²C bus transmit data register	ICDRT_1	R/W	H'FF	H'FFFEE406	8
	I²C bus receive data register	ICDRR_1	R/W	H'FF	H'FFFEE407	8
	NF2CYC register	NF2CYC_1	R/W	H'00	H'FFFEE408	8
2	I²C bus control register 1	ICCR1_2	R/W	H'00	H'FFFEE800	8
	I²C bus control register 2	ICCR2_2	R/W	H'7D	H'FFFEE801	8
	I²C bus mode register	ICMR_2	R/W	H'38	H'FFFEE802	8
	I²C bus interrupt enable register	ICIER_2	R/W	H'00	H'FFFEE803	8
	I²C bus status register	ICSR_2	R/W	H'00	H'FFFEE804	8
	Slave address register	SAR_2	R/W	H'00	H'FFFEE805	8
	I²C bus transmit data register	ICDRT_2	R/W	H'FF	H'FFFEE806	8
	I²C bus receive data register	ICDRR_2	R/W	H'FF	H'FFFEE807	8
	NF2CYC register	NF2CYC_2	R/W	H'00	H'FFFEE808	8

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
3	I²C bus control register 1	ICCR1_3	R/W	H'00	H'FFFEEC00	8
	I²C bus control register 2	ICCR2_3	R/W	H'7D	H'FFFEEC01	8
	I²C bus mode register	ICMR_3	R/W	H'38	H'FFFEEC02	8
	I²C bus interrupt enable register	ICIER_3	R/W	H'00	H'FFFEEC03	8
	I²C bus status register	ICSR_3	R/W	H'00	H'FFFEEC04	8
	Slave address register	SAR_3	R/W	H'00	H'FFFEEC05	8
	I²C bus transmit data register	ICDRT_3	R/W	H'FF	H'FFFEEC06	8
	I²C bus receive data register	ICDRR_3	R/W	H'FF	H'FFFEEC07	8
	NF2CYC register	NF2CYC_3	R/W	H'00	H'FFFEEC08	8

18.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 is an 8-bit readable/writable register that enables or disables the I²C bus interface 3, 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:	7	6	5	4	3	2	1	0
	ICE	RCVD	MST	TRS	CKS[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface 3 Enable 0: SCL and SDA output is disabled. (Input to SCL and SDA is enabled) 1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable 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 8th 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 to 0	CKS[3:0]	0000	R/W	Transfer Clock Select These bits should be set according to the necessary transfer rate (table 18.3) in master mode.

Table 18.3 Transfer Rate

Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate (kHz)				
CKS3	CKS2	CKS1	CKS0	Clock	Pϕ = 16.7 MHz	Pϕ = 20.0 MHz	Pϕ = 25.0 MHz	Pϕ = 26.7 MHz	Pϕ = 33.3 MHz
0	0	0	0	Pϕ/44	379 kHz	455 kHz	568 kHz	606 kHz	758 kHz
			1	Pϕ/52	321 kHz	385 kHz	481 kHz	513 kHz	641 kHz
		1	0	Pϕ/64	260 kHz	313 kHz	391 kHz	417 kHz	521 kHz
			1	Pϕ/72	231 kHz	278 kHz	347 kHz	370 kHz	463 kHz
	1	0	0	Pϕ/84	198 kHz	238 kHz	298 kHz	317 kHz	397 kHz
			1	Pϕ/92	181 kHz	217 kHz	272 kHz	290 kHz	362 kHz
		1	0	Pϕ/100	167 kHz	200 kHz	250 kHz	267 kHz	333 kHz
			1	Pϕ/108	154 kHz	185 kHz	231 kHz	247 kHz	309 kHz
1	0	0	0	Pϕ/176	94.7 kHz	114 kHz	142 kHz	152 kHz	189 kHz
			1	Pϕ/208	80.1 kHz	96.2 kHz	120 kHz	128 kHz	160 kHz
		1	0	Pϕ/256	65.1 kHz	78.1 kHz	97.7 kHz	104 kHz	130 kHz
			1	Pϕ/288	57.9 kHz	69.4 kHz	86.8 kHz	92.6 kHz	116 kHz
	1	0	0	Pϕ/336	49.6 kHz	59.5 kHz	74.4 kHz	79.4 kHz	99.2 kHz
			1	Pϕ/368	45.3 kHz	54.3 kHz	67.9 kHz	72.5 kHz	90.6 kHz
		1	0	Pϕ/400	41.7 kHz	50.0 kHz	62.5 kHz	66.7 kHz	83.3 kHz
			1	Pϕ/432	38.6 kHz	46.3 kHz	57.9 kHz	61.7 kHz	77.2 kHz

Note: The settings should satisfy external specifications.

18.3.2 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.

Bit:	7	6	5	4	3	2	1	0
	BBSY	SCP	SDAO	SDAOP	SCLO	-	IICRST	-
Initial value:	0	1	1	1	1	1	0	1
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy 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 is always read as 0. 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	R/W	Start/Stop Issue Condition Disable 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. Even if 1 is written to this bit, the data will not be stored.

Bit	Bit Name	Initial Value	R/W	Description
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).
4	SDAOP	1	R/W	SDAO Write Protect 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	SCL Output Level Monitors SCL output level. When SCLO is 1, SCL pin outputs high. When SCLO is 0, SCL pin outputs low.
2	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
1	IICRST	0	R/W	IIC Control Part Reset 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 bus operation, some IIC3 registers and the control part can be reset.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

18.3.3 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.

Bits BC[2:0] are initialized to H'0 by the IICRST bit in ICCR2.

Bit:	7	6	5	4	3	2	1	0
	MLS	-	-	-	BCWP	BC[2:0]		
Initial value:	0	0	1	1	1	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

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	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	BCWP	1	R/W	BC Write Protect Controls the BC[2:0] modifications. When modifying the BC[2:0] bits, this bit should be cleared to 0. In clocked synchronous serial mode, the BC[2:0] bits should not be modified. 0: When writing, values of the BC[2:0] bits are set. 1: When reading, 1 is always read. When writing, settings of the BC[2:0] bits are invalid.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	BC[2:0]	000	R/W	Bit Counter 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 these bits are set to a value other than B'000, the setting should be made while the SCL pin is low. The bit value returns to B'000 automatically at the end of a data transfer including the acknowledge bit. And the value becomes B'111 automatically after the stop condition detection. These bits are cleared by a power-on reset and in software standby mode and module standby mode. These bits are also cleared by setting the IICRST bit of ICCR2 to 1. With the clocked synchronous serial format, these bits should not be modified. I²C Bus Format — Clocked 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

18.3.4 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.

Bit:	7	6	5	4	3	2	1	0
	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

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 or 0, 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 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 Enables or disables the receive data full interrupt request (RXI) when 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.

Bit	Bit Name	Initial Value	R/W	Description
4	NAKIE	0	R/W	NACK Receive Interrupt Enable NAKIE enables or disables the NACK detection, arbitration lost, and overrun error interrupt request (NAKI) when the NACKF or AL/OVE bit in ICSR is set to 1. 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.
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable Enables or disables the stop condition detection interrupt request (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 setting 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.

18.3.5 I²C Bus Status Register (ICSR)

ICSR is an 8-bit readable/writable register that confirms interrupt request flags and their status.

Bit:	7	6	5	4	3	2	1	0
	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

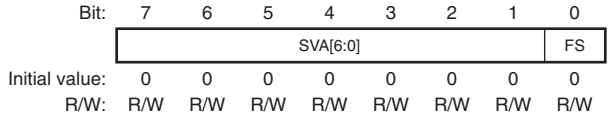
Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When data is written to ICDRT [Setting conditions] • When data is transferred from ICDRT to ICDRS and ICDRT becomes empty • When TRS is set • When the start condition (including retransmission) is issued • When slave mode is changed from receive mode to transmit mode
6	TEND	0	R/W	Transmit End [Clearing conditions] • When 0 is written in TEND after reading TEND = 1 • When data is written to ICDRT [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 clocked synchronous serial format

Bit	Bit Name	Initial Value	R/W	Description
5	RDRF	0	R/W	Receive Data Full [Clearing conditions] • When 0 is written in RDRF after reading RDRF = 1 • When ICDRR is read [Setting condition] • When a receive data is transferred from ICDRS to ICDRR
4	NACKF	0	R/W	No Acknowledge Detection Flag [Clearing condition] • When 0 is written in NACKF after reading NACKF = 1 [Setting condition] • When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1
3	STOP	0	R/W	Stop Condition Detection Flag [Clearing condition] • When 0 is written in STOP after reading STOP = 1 [Setting condition] • When a stop condition is detected after frame transfer is completed

Bit	Bit Name	Initial Value	R/W	Description
2	AL/OVE	0	R/W	Arbitration Lost Flag/Overrun Error 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 3 detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been occupied by another master. [Clearing condition] • When 0 is written in AL/OVE after reading AL/OVE = 1 [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
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 SVA[6:0] in SAR. [Clearing condition] • When 0 is written in AAS after reading AAS = 1 [Setting conditions] • When the slave address is detected in slave receive mode • When the general call address is detected in slave receive mode.
0	ADZ	0	R/W	General Call Address Recognition Flag This bit is valid in slave receive mode with the I²C bus format. [Clearing condition] • When 0 is written in ADZ after reading ADZ = 1 [Setting condition] • When the general call address is detected in slave receive mode

18.3.6 Slave Address Register (SAR)

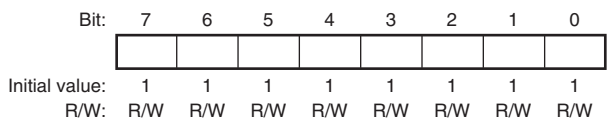
SAR is an 8-bit readable/writable register that selects the communications format and sets the slave address. In slave mode with the I²C bus format, if the upper seven bits of SAR match the upper seven bits of the first frame received after a start condition, this module operates as the slave device.



Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA[6:0]	0000000	R/W	Slave Address These bits set a unique address in these bits, 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

18.3.7 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 while transferring data of ICDRS, continuous transfer is possible.



18.3.8 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.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.3.9 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.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

18.3.10 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 18.4.7, Noise Filter.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PRS	NF2CYC
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PRS	0	R/W	Pulse Width Ratio Select Specifies the ratio of the high-level period to the low-level period for the SCL signal. 0: The ratio of high to low is 0.5 to 0.5. 1: The ratio of high to low is about 0.4 to 0.6.
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

18.4 Operation

The I²C bus interface 3 can communicate either in I²C bus mode or clocked synchronous serial mode by setting FS in SAR.

18.4.1 I²C Bus Format

Figure 18.3 shows the I²C bus formats. Figure 18.4 shows the I²C bus timing. The first frame following a start condition always consists of eight bits.

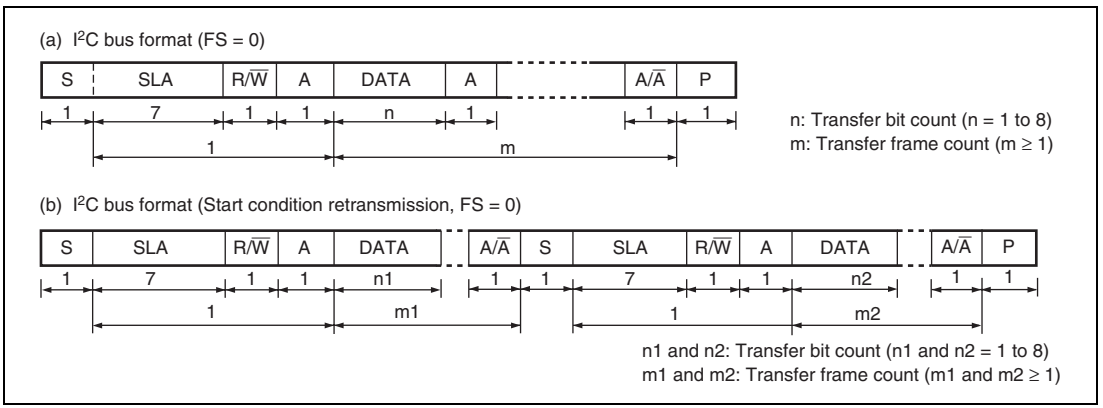


Figure 18.3 I²C Bus Formats

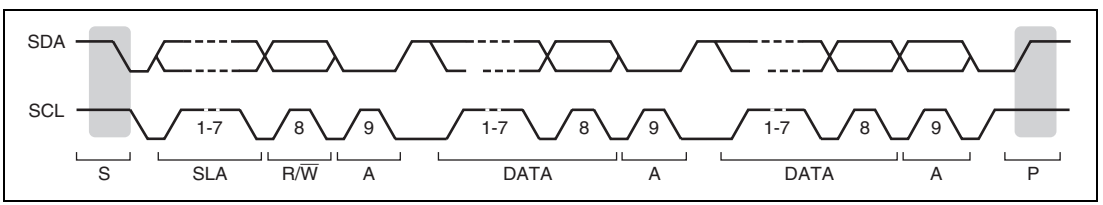


Figure 18.4 I²C Bus Timing

[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.

18.4.2 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 18.5 and 18.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Also, set bits CKS[3:0] in ICCR1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is released. 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 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT. 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.

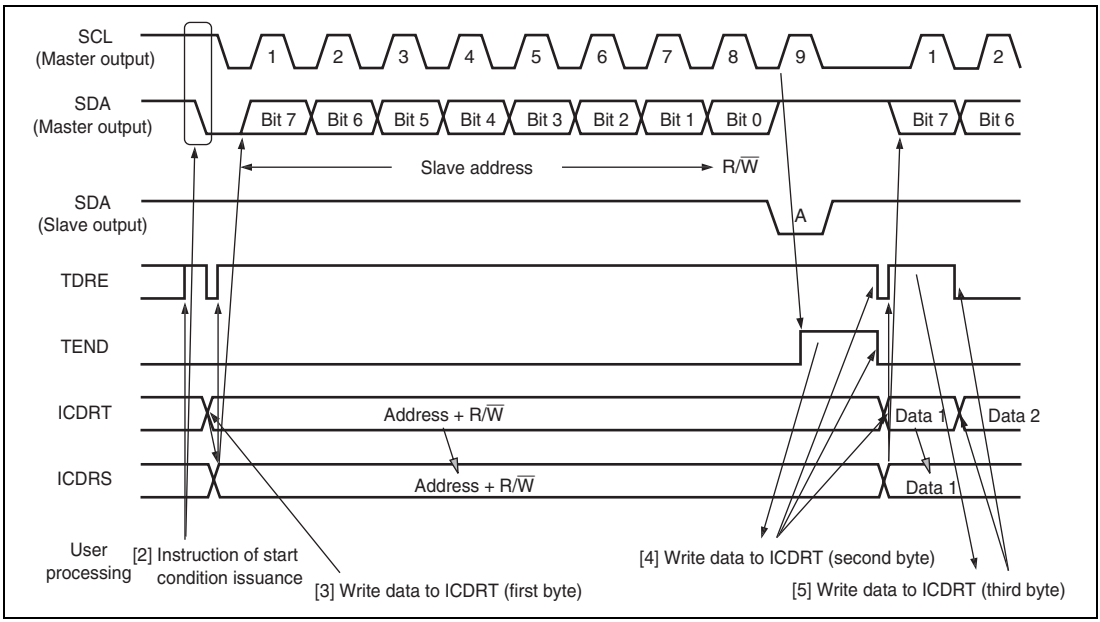


Figure 18.5 Master Transmit Mode Operation Timing (1)

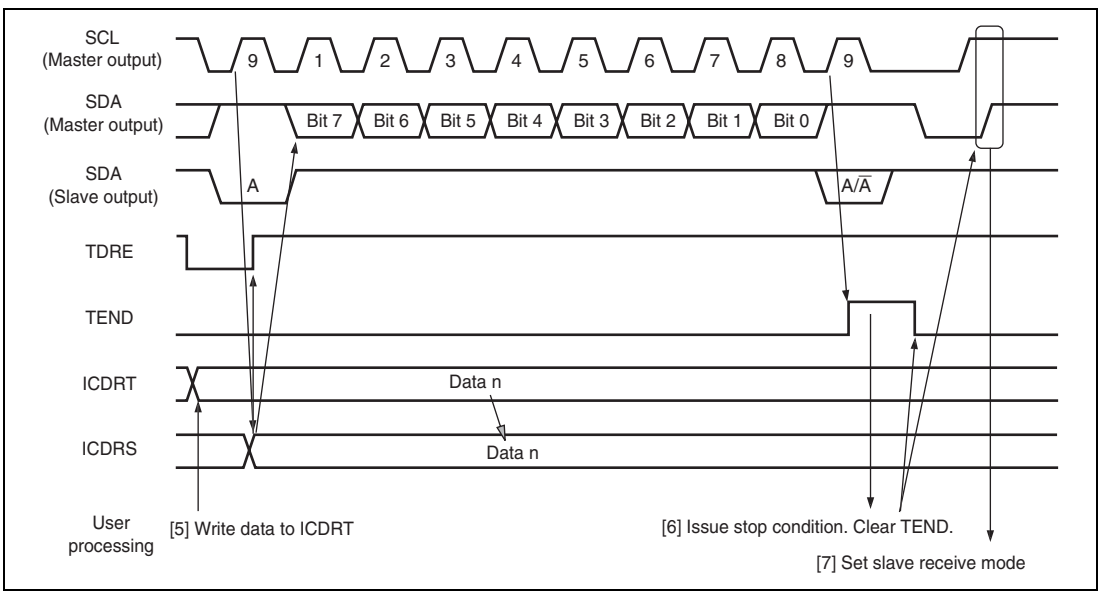


Figure 18.6 Master Transmit Mode Operation Timing (2)

18.4.3 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 18.7 and 18.8. 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 9th receive clock pulse.
3. After the reception of first frame data is completed, the RDRF bit in ICSR is set to 1 at the rise of 9th 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 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading ICDRR. This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, 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: If only one byte is received, read ICDRR (dummy-read) after the RCVD bit in ICCR1 is set.

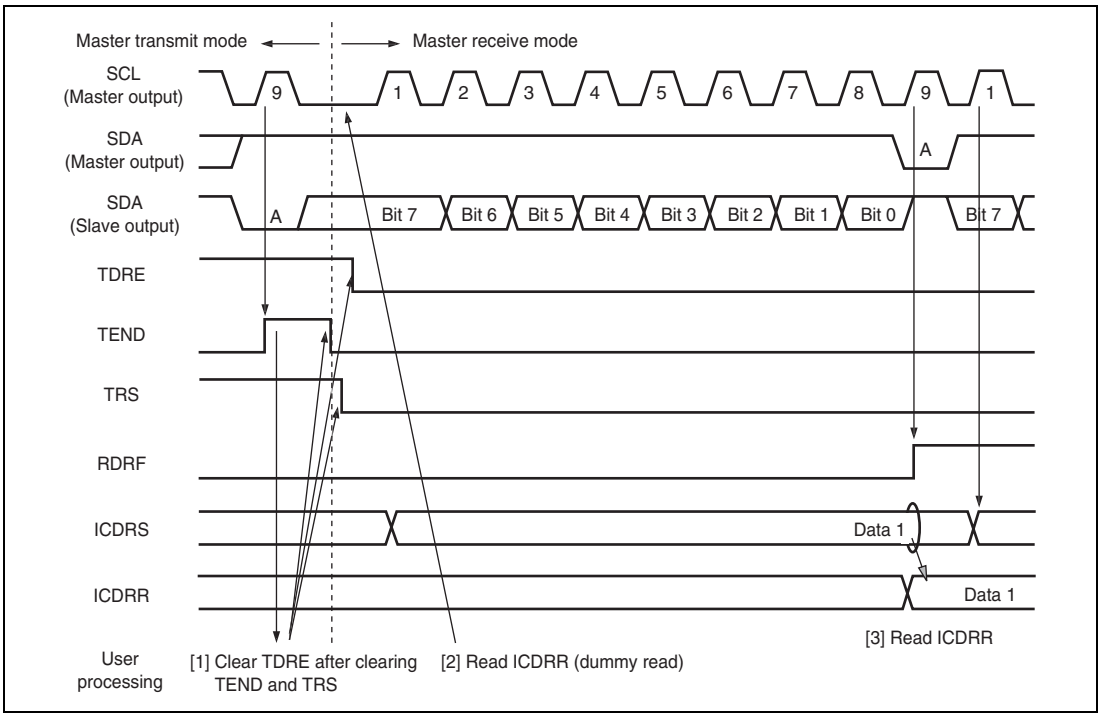


Figure 18.7 Master Receive Mode Operation Timing (1)

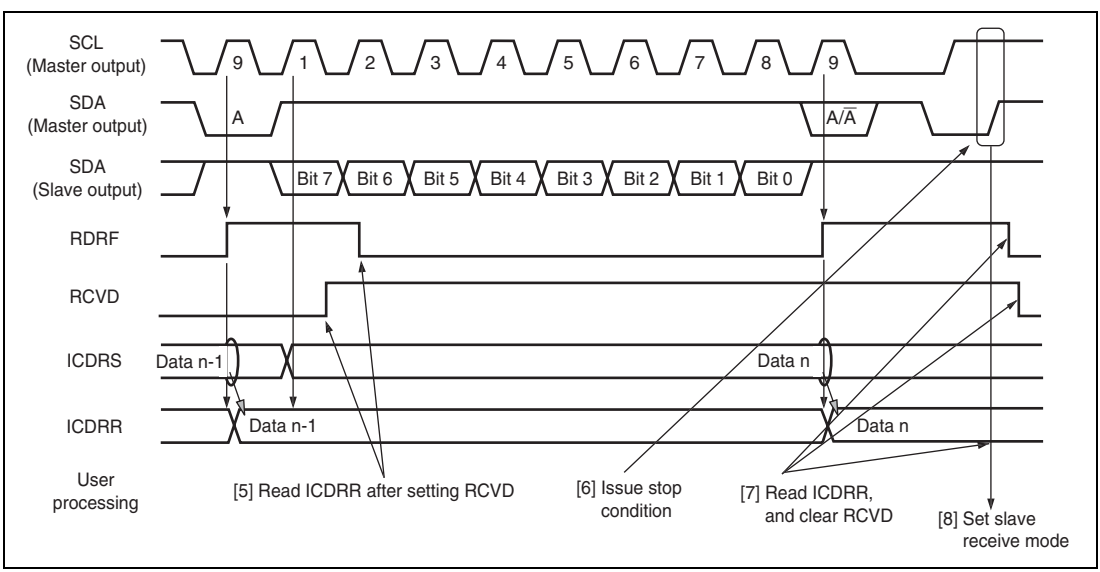


Figure 18.8 Master Receive Mode Operation Timing (2)

18.4.4 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 18.9 and 18.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (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 9th clock pulse. At this time, if the 8th bit data (R/W) is 1, the TRS bit in ICCR1 and the TDRE bit in ICSR 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 opened.
5. Clear TDRE.

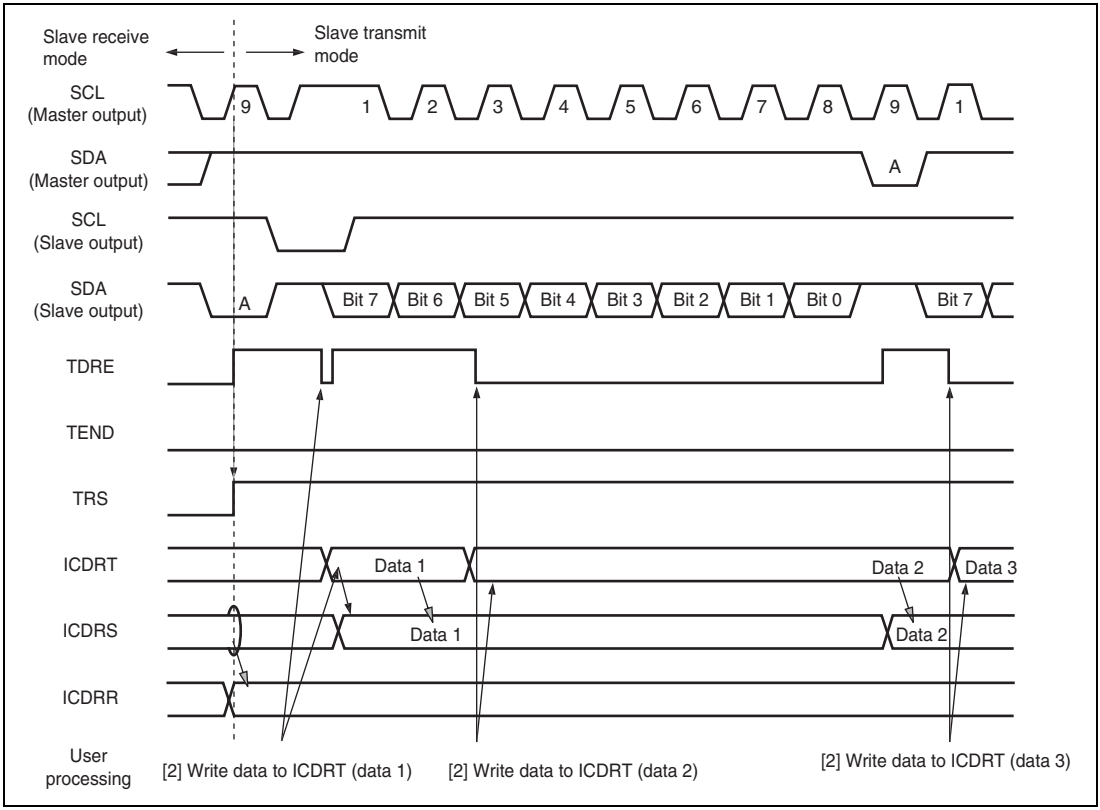


Figure 18.9 Slave Transmit Mode Operation Timing (1)

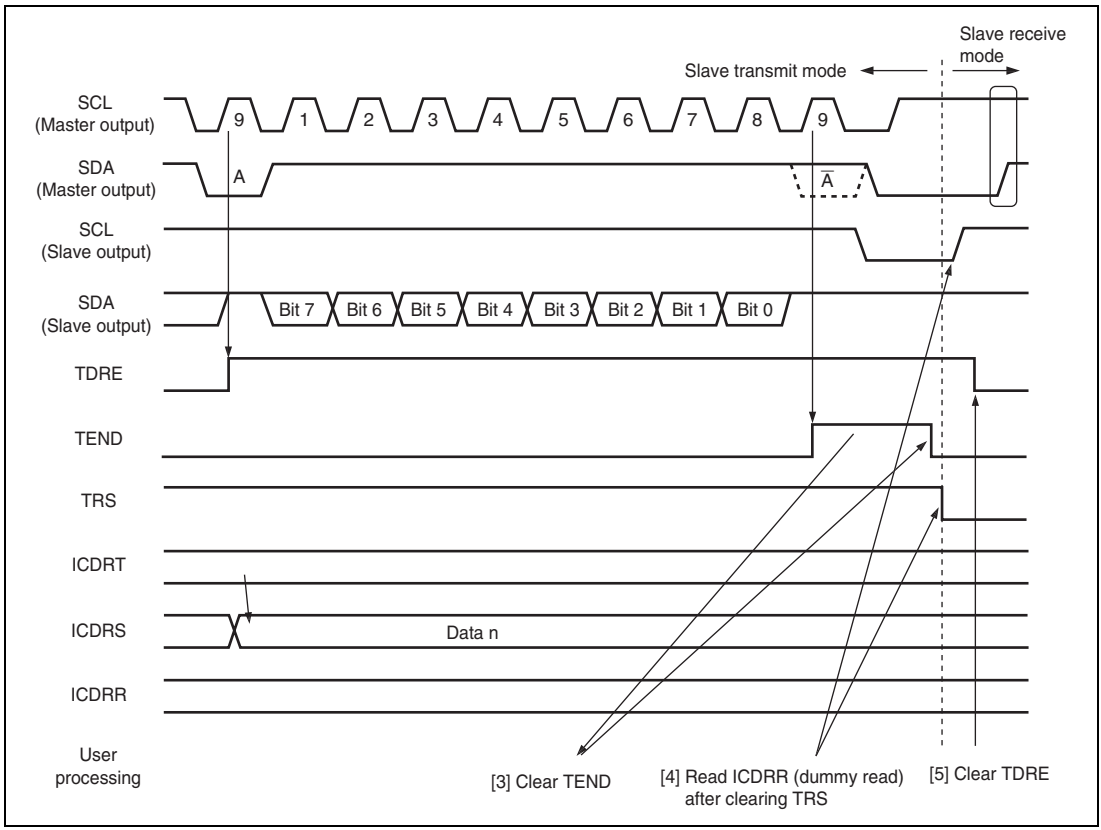


Figure 18.10 Slave Transmit Mode Operation Timing (2)

18.4.5 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 18.11 and 18.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (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 9th 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/W, it is not used.)
3. Read ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is 1, SCL is fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR, to be returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

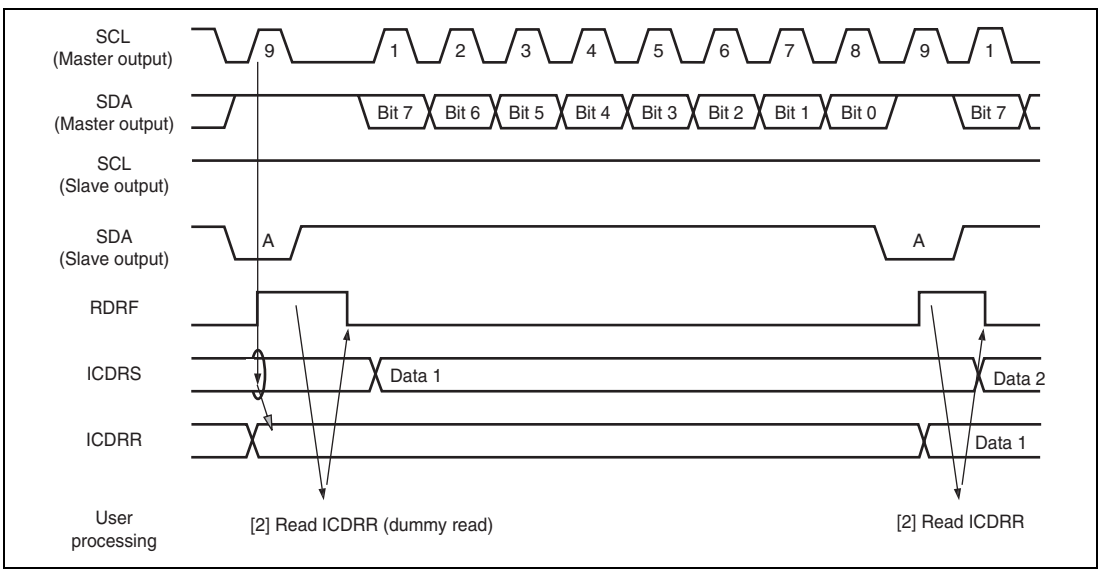


Figure 18.11 Slave Receive Mode Operation Timing (1)

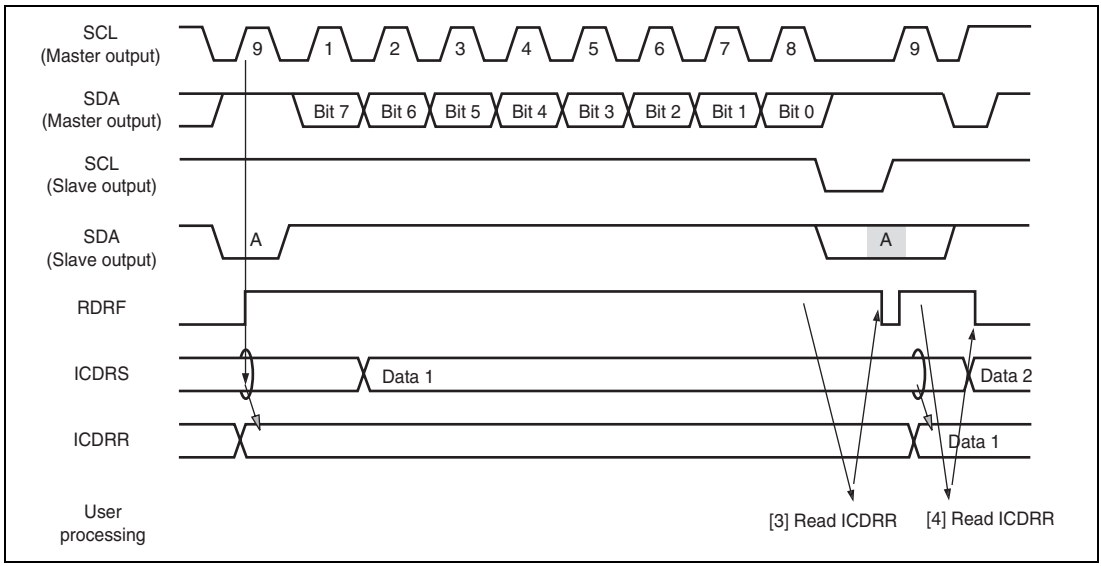


Figure 18.12 Slave Receive Mode Operation Timing (2)

18.4.6 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.

(1) Data Transfer Format

Figure 18.13 shows the clocked synchronous serial transfer format.

The transfer data is output from the fall 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.

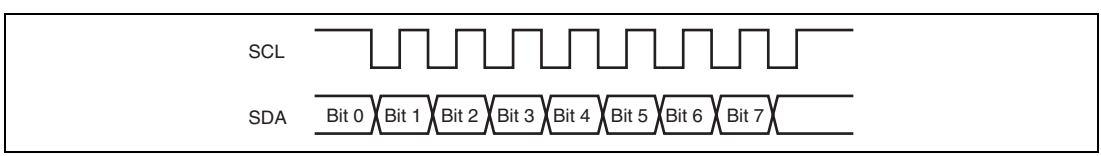


Figure 18.13 Clocked Synchronous Serial Transfer Format

(2) 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 18.14. The transmission procedure and operations in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS[3:0] bits in ICCR1. (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.

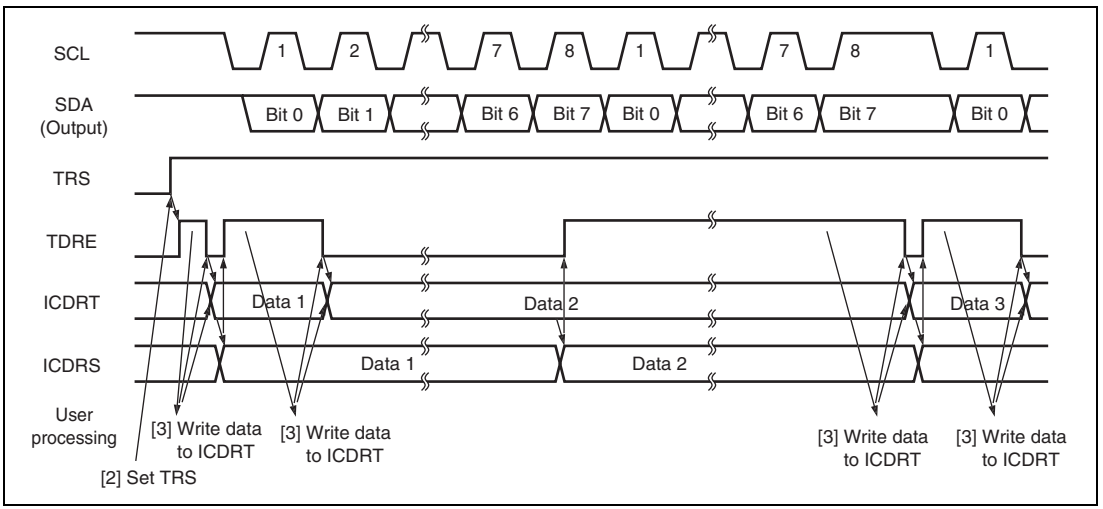


Figure 18.14 Transmit Mode Operation Timing

(3) 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 18.15. The reception procedure and operations in receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (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 8th clock is risen 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 18.16 for the operation timing.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] 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.

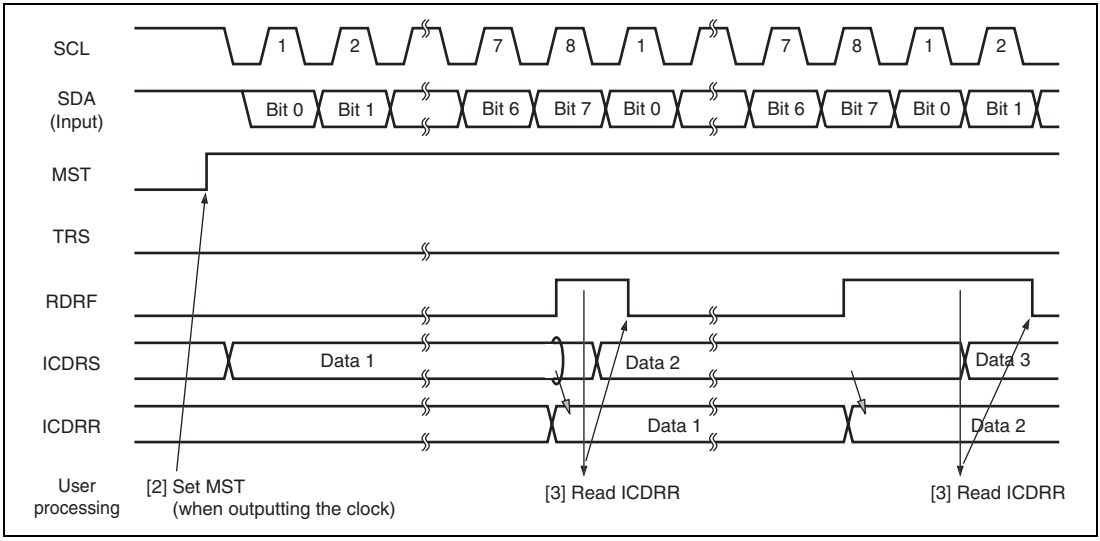


Figure 18.15 Receive Mode Operation Timing

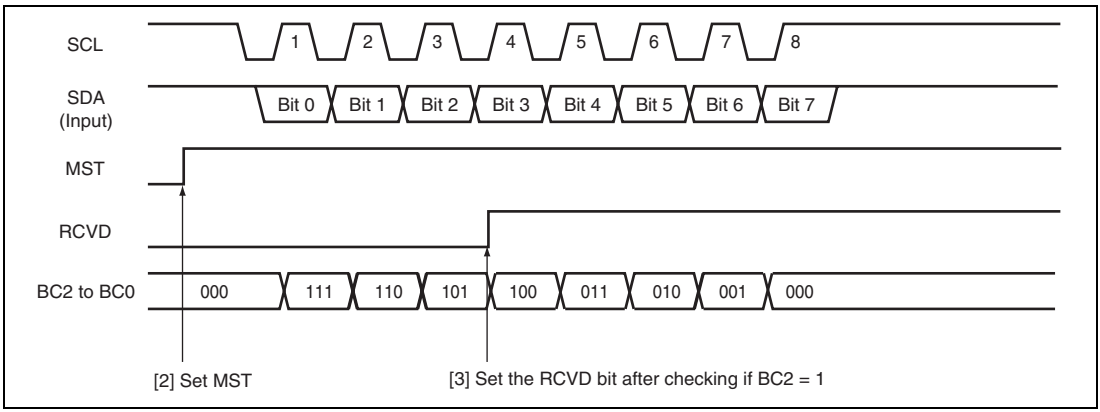


Figure 18.16 Operation Timing For Receiving One Byte (MST = 1)

18.4.7 Noise Filter

The logic levels at the SCL and SDA pins are routed through noise filters before being latched internally. Figure 18.17 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 peripheral 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.

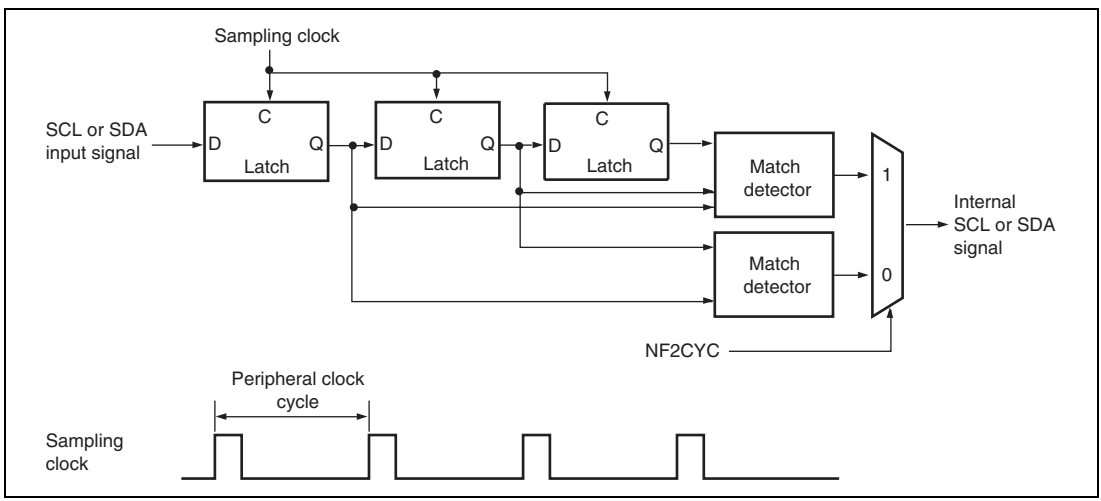


Figure 18.17 Block Diagram of Noise Filter

18.4.8 Example of Use

Flowcharts in respective modes that use the I²C bus interface 3 are shown in figures 18.18 to 18.21.

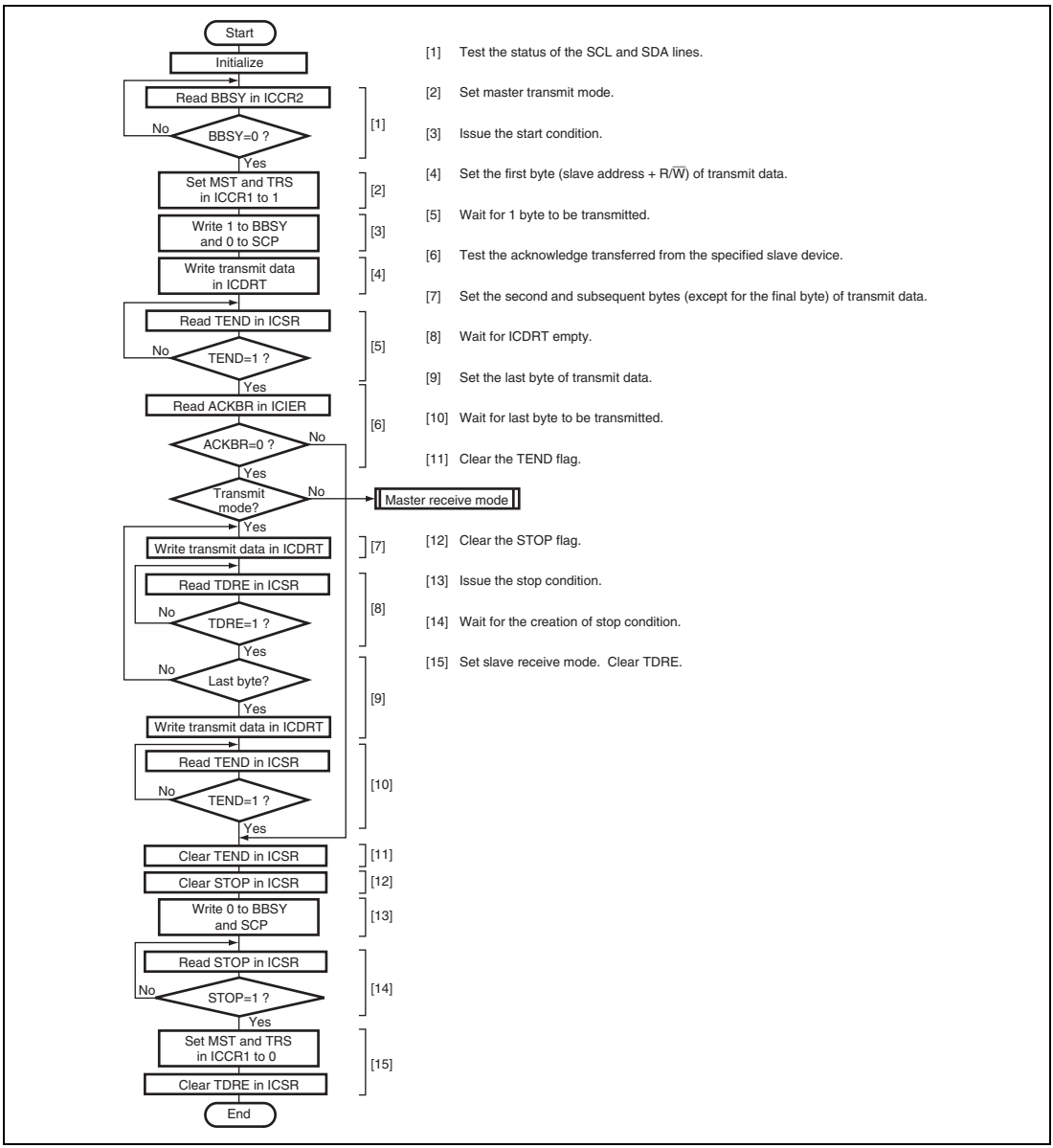
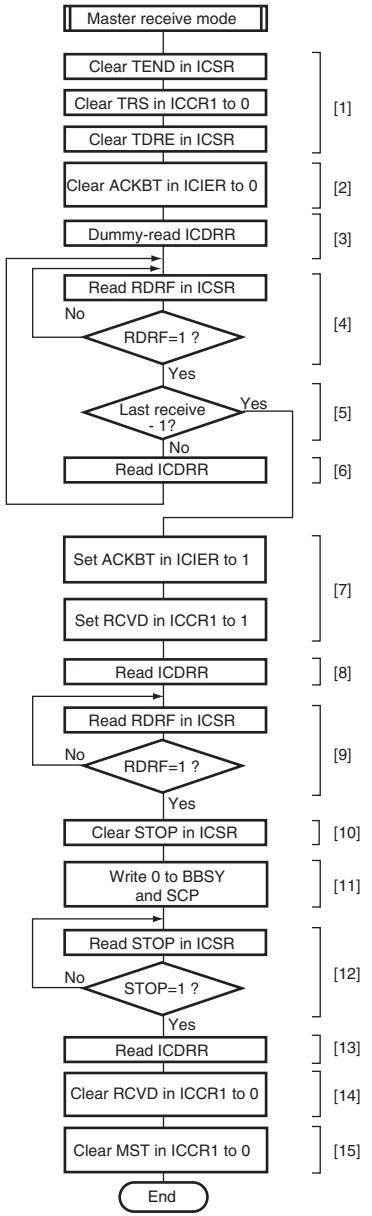


Figure 18.18 Sample Flowchart for Master Transmit Mode



[1] Clear TEND, select master receive mode, and then clear TDRE. *1

[2] Set acknowledge to the transmit device. *1

[3] Dummy-read ICDDR. *1

[4] Wait for 1 byte to be received*2

[5] Check whether it is the (last receive - 1).*2

[6] Read the receive data.

[7] Set acknowledge of the final byte. Disable continuous reception (RCVD = 1).*2

[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.

Notes: 1. Make sure that no interrupt will be generated during steps [1] to [3].
2. When the (last receive -1) is checked (when step [5] is approved), make sure that no interrupt will be generated during steps [4], [5], and [7].

[Complement] 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.

Figure 18.19 Sample Flowchart for Master Receive Mode

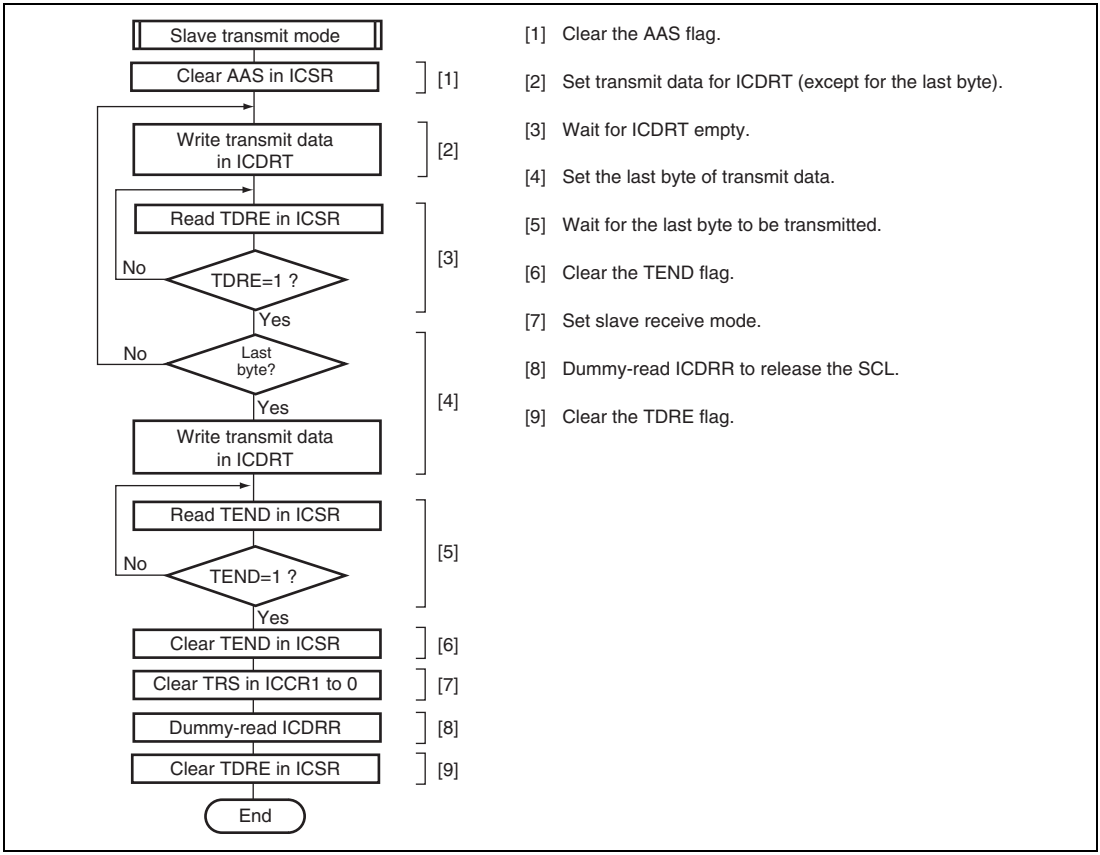


Figure 18.20 Sample Flowchart for Slave Transmit Mode

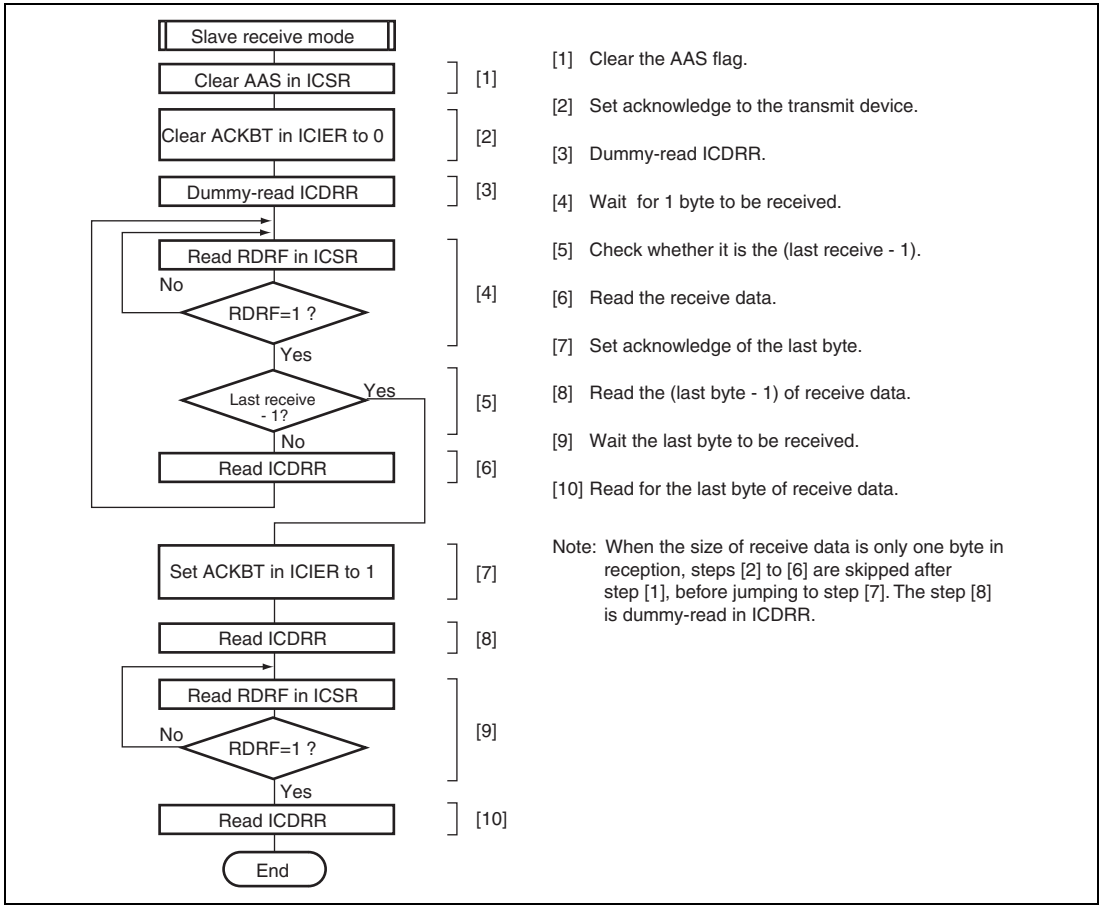


Figure 18.21 Sample Flowchart for Slave Receive Mode

18.5 Interrupt Requests

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK detection, STOP recognition, and arbitration lost/overrun error. Table 18.4 shows the contents of each interrupt request.

Table 18.4 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition	I²C Bus Format	Clocked Synchronous Serial Format
Transmit data Empty	TXI	(TDRE = 1) • (TIE = 1)	√	√
Transmit end	TEI	(TEND = 1) • (TEIE = 1)	√	√
Receive data full	RXI	(RDRF = 1) • (RIE = 1)	√	√
STOP recognition	STPI	(STOP = 1) • (STIE = 1)	√	—
NACK detection	NAKI	{(NACKF = 1) + (AL = 1)} • (NAKIE = 1)	√	—
Arbitration lost/ overrun error			√	√

When the interrupt condition described in table 18.4 is 1, the CPU executes an interrupt exception handling. Note that a TXI or RXI interrupt can activate the DMAC if the setting for DMAC activation has been made. In such a case, an interrupt request is not sent to the CPU. 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. Therefore, when the TDRE bit is cleared to 0, then an excessive data of one byte may be transmitted.

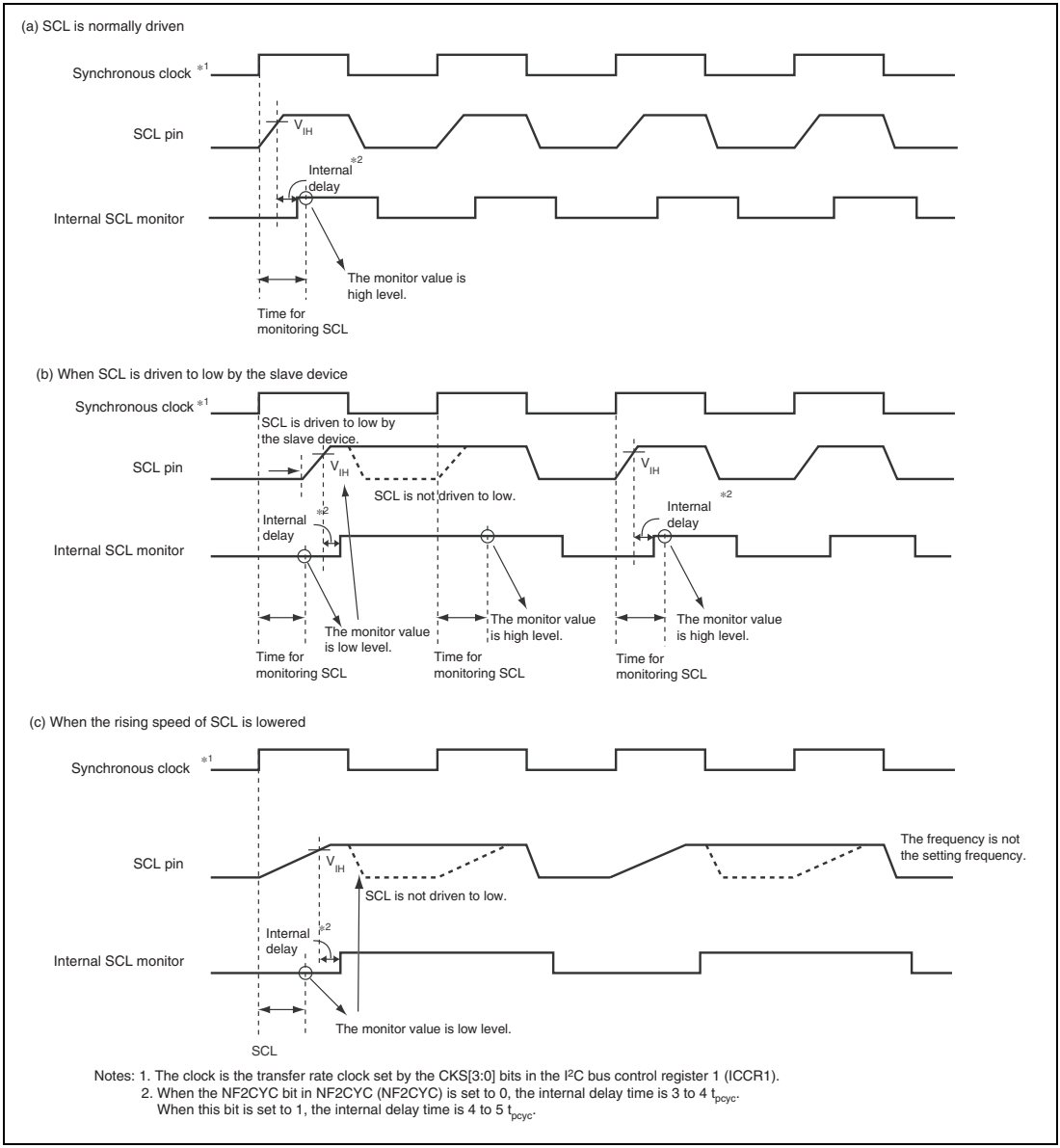
18.6 Bit Synchronous Circuit

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 18.22 shows the timing of the bit synchronous circuit and table 18.5 shows the time when the SCL output changes from low to Hi-Z then SCL is monitored.



Notes: 1. The clock is the transfer rate clock set by the CKS[3:0] bits in the I²C bus control register 1 (ICCR1).
2. When the NF2CYC bit in NF2CYC (NF2CYC) is set to 0, the internal delay time is 3 to 4 t_{pcyc}.
When this bit is set to 1, the internal delay time is 4 to 5 t_{pcyc}.

Figure 18.22 Bit Synchronous Circuit Timing

Table 18.5 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL
0	0	9 tpcyc*
	1	21 tpcyc*
1	0	39 tpcyc*
	1	87 tpcyc*

Note: * tpcyc indicates the frequency of the peripheral clock (Pϕ).

18.7 Usage Notes

18.7.1 Note on Setting for Multi-Master Operation

In multi-master operation, when the setting for IIC transfer rate (ICCR1.CKS[3:0]) makes this LSI slower than the other masters, pulse cycles with an unexpected length will infrequently be output on SCL.

Be sure to specify a transfer rate that is at least 1/1.8 of the fastest transfer rate among the other masters.

18.7.2 Note on Master Receive Mode

Reading ICDRR around the falling edge of the 8th clock might fail to fetch the receive data.

In addition, when RCVD is set to 1 around the falling edge of the 8th clock and the receive buffer full, a stop condition may not be issued.

Use either 1 or 2 below as a measure against the situations above.

1. In master receive mode, read ICDRR before the rising edge of the 8th clock.
2. In master receive mode, set the RCVD bit to 1 so that transfer proceeds in byte units.

18.7.3 Note on Setting ACKBT in Master Receive Mode

In master receive mode operation, set ACKBT before the falling edge of the 8th SCL cycle of the last data being continuously transferred. Not doing so can lead to an overrun for the slave transmission device.

18.7.4 Note on the States of Bits MST and TRN when Arbitration Is Lost

When sequential bit-manipulation instructions are used to set the MST and TRS bits to select master transmission in multi-master operation, a conflicting situation where AL in ICSR = 1 but the mode is master transmit mode (MST = 1 and TRS = 1) may arise; this depends on the timing of the loss of arbitration when the bit manipulation instruction for TRS is executed.

This can be avoided in either of the following ways.

- In multi-master operation, use the MOV instruction to set the MST and TRS bits.
- When arbitration is lost, check whether the MST and TRS bits are 0. If the MST and TRS bits have been set to a value other than 0, clear the bits to 0.

18.7.5 Note on I²C-bus Interface Master Receive Mode

After a master receive operation is completed, confirm the falling edge of the ninth clock cycle of the SCL signal and generate a stop condition or regenerate a start condition.

18.7.6 Note on IICRST and BBSY bits

When 1 is written to IICRST in ICCR2, this LSI release SCL and SDA pins. Then, if the SDA level changes from low to high under the condition of SCL = high, BBSY in ICCR2 is cleared to 0 assuming that the stop condition has been issued.

18.7.7 Note on Issuance of Stop Conditions in Master Transmit Mode while ACKE = 1

When a stop condition is issued in master transmit mode while the ACKE bit in the I²C bus interrupt enable register (ICIER) is 1, the stop condition may not be normally output depending on the issued timing. To avoid this, recognize the falling edge of the ninth clock before issuance of the stop condition.

The falling edge of the ninth clock can be recognized by checking the SCLO bit in the I²C control register 2 (ICCR2).

Section 19 Serial Sound Interface with FIFO (SSIF)

The serial sound interface with FIFO (SSIF) is a module designed to send or receive audio data interface with various devices offering I^2S bus compatibility. It also provides additional modes for other common formats, as well as support for multi-channel mode.

19.1 Features

- Number of channels: Six channels
- Operating mode: Non-compressed mode
  The non-compressed mode supports serial audio streams divided by channels.
- Serves as both a transmitter and a receiver
- Capable of using serial bus format
- Asynchronous transfer takes place between the data buffer and the shift register.
- It is possible to select a value as the dividing ratio for the clock used by the serial bus interface.
- It is possible to control data transmission or reception with DMAC and interrupt requests.
- Selects the oversampling clock input from among the following pins:
  EXTAL, XTAL (Clock operation modes 0 and 1: 10 to 33.33 MHz)
  CKIO (Clock operation mode 2: 40 to 50 MHz*)
  AUDIO_CLK (1 to 40 MHz)
  AUDIO_X1, AUDIO_X2 (when connecting a crystal oscillator: 10 to 40 MHz, when used to input external clock: 1 to 40 MHz)
  Note: * Do not select CKIO as the supply source for the oversampling clock if the frequency of CKIO is over 50 MHz and the mode is clock operating mode 2.
- Includes an 8-stage FIFO buffer for transmission and reception

Figure 19.1 shows a schematic diagram of the four channels in the SSIF module.

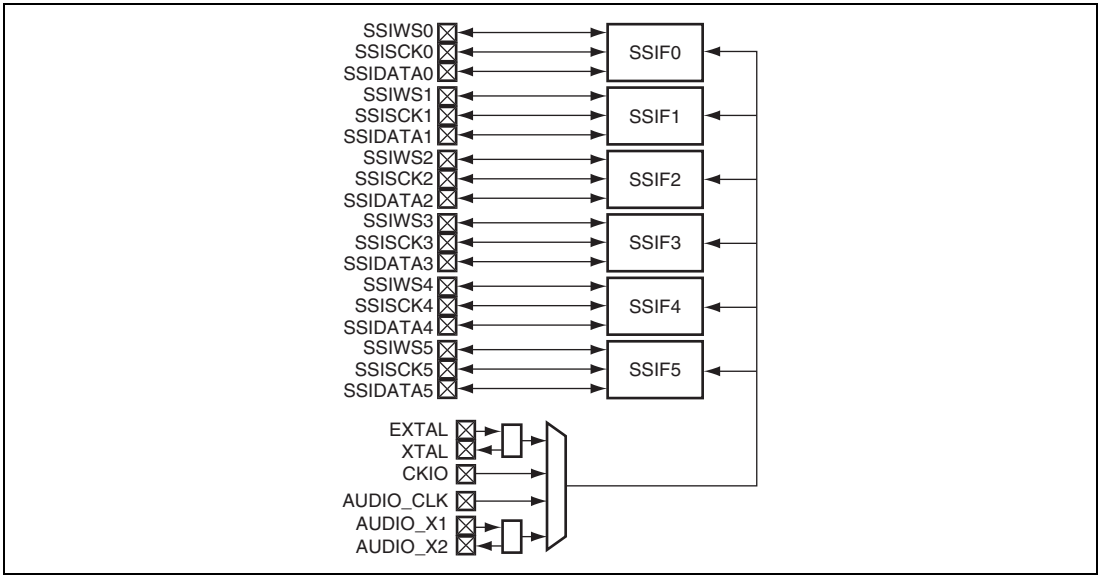


Figure 19.1 Schematic Diagram of SSIF Module

Figure 19.2 shows a block diagram of the SSIF module.

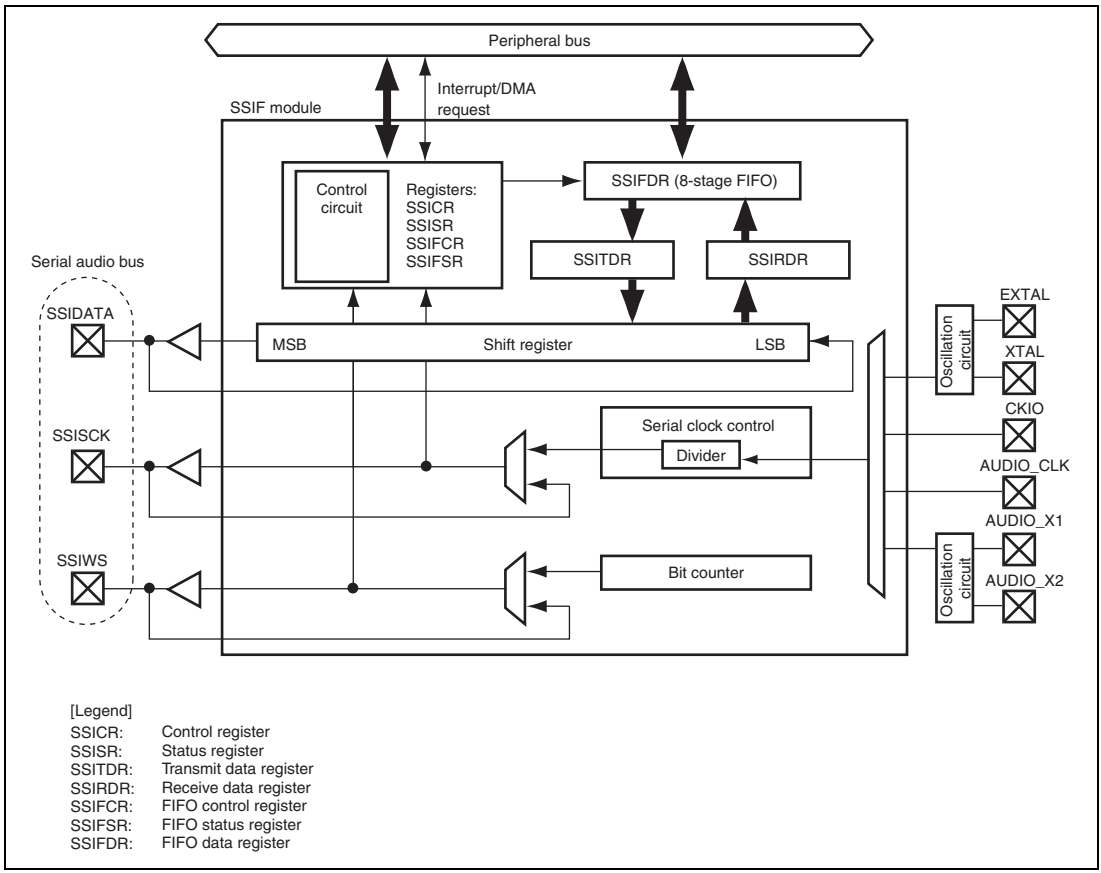


Figure 19.2 Block Diagram of SSIF

19.2 Input/Output Pins

Table 19.1 shows the pin assignments relating to the SSIF module.

Table 19.1 Pin Assignments

Channel	Pin Name	I/O	Description
0 to 5	SSISCK0 to SSISCK5	I/O	Serial bit clock
	SSIWS0 to SSIWS5	I/O	Word selection
	SSIDATA0 to SSIDATA5	I/O	Serial data input/output
Common	AUDIO_CLK	Input	External clock for audio (input oversampling clock)
	AUDIO_X1	Input	Crystal oscillator for audio (input oversampling clock)
	AUDIO_X2	Output	

19.3 Register Description

The SSIF has the following registers. Note that explanation in the text does not refer to the channels.

Table 19.2 Register Description

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Control register 0	SSICR_0	R/W	H'00000000	H'FFFEB000	8, 16, 32
	Status register 0	SSISR_0	R/W*1	H'02000003	H'FFFEB004	8, 16, 32
	FIFO control register 0	SSIFCR_0	R/W	H'00000000	H'FFFEB010	8, 16, 32
	FIFO status register 0	SSIFSR_0	R/(W)*2	H'00000002	H'FFFEB014	8, 16, 32
	FIFO data register 0	SSIFDR_0	R/W*3	Undefined	H'FFFEB018	32
1	Control register 1	SSICR_1	R/W	H'00000000	H'FFFEB400	8, 16, 32
	Status register 1	SSISR_1	R/W*1	H'02000003	H'FFFEB404	8, 16, 32
	FIFO control register 1	SSIFCR_1	R/W	H'00000000	H'FFFEB410	8, 16, 32
	FIFO status register 1	SSIFSR_1	R/(W)*2	H'00000002	H'FFFEB414	8, 16, 32
	FIFO data register 1	SSIFDR_1	R/W*3	Undefined	H'FFFEB418	32
2	Control register 2	SSICR_2	R/W	H'00000000	H'FFFEB800	8, 16, 32
	Status register 2	SSISR_2	R/W*1	H'02000003	H'FFFEB804	8, 16, 32
	FIFO control register 2	SSIFCR_2	R/W	H'00000000	H'FFFEB810	8, 16, 32
	FIFO status register 2	SSIFSR_2	R/(W)*2	H'00000002	H'FFFEB814	8, 16, 32
	FIFO data register 2	SSIFDR_2	R/W*3	Undefined	H'FFFEB818	32
3	Control register 3	SSICR_3	R/W	H'00000000	H'FFFEBC00	8, 16, 32
	Status register 3	SSISR_3	R/W*1	H'02000003	H'FFFEBC04	8, 16, 32
	FIFO control register 3	SSIFCR_3	R/W	H'00000000	H'FFFEBC10	8, 16, 32
	FIFO status register 3	SSIFSR_3	R/(W)*2	H'00000002	H'FFFEBC14	8, 16, 32
	FIFO data register 3	SSIFDR_3	R/W*3	Undefined	H'FFFEBC18	32
4	Control register 4	SSICR_4	R/W	H'00000000	H'FFFEC000	8, 16, 32
	Status register 4	SSISR_4	R/W*1	H'02000003	H'FFFEC004	8, 16, 32
	FIFO control register 4	SSIFCR_4	R/W	H'00000000	H'FFFEC010	8, 16, 32
	FIFO status register 4	SSIFSR_4	R/(W)*2	H'00000002	H'FFFEC014	8, 16, 32
	Receive data register 4	SSIFDR_4	R/W*3	Undefined	H'FFFEC018	32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
5	Control register 5	SSICR_5	R/W	H'00000000	H'FFFEC400	8, 16, 32
	Status register 5	SSISR_5	R/W*1	H'02000003	H'FFFEC404	8, 16, 32
	FIFO control register 5	SSIFCR_5	R/W	H'00000000	H'FFFEC410	8, 16, 32
	FIFO status register 5	SSIFSR_5	R/(W)*2	H'00000002	H'FFFEC414	8, 16, 32
	FIFO data register 5	SSIFDR_5	R/W*3	Undefined	H'FFFEC418	32

Notes: 1. Although bits 26 and 27 in these registers can be read from or written to, bits other than these are read-only. For details, refer to section 19.3.2, Status Register (SSISR).
2. To bits 1 and 0 in these registers, only 0 can be written to clear the flags. Other bits are read-only. For details, refer to section 19.3.6, FIFO Status Register (SSIFSR).
3. These registers cannot be written to during reception. For details, refer to section 19.3.7, FIFO Data Register (SSIFDR).

19.3.1 Control Register (SSICR)

SSICR is a readable/writable 32-bit register that controls the IRQ, selects the polarity status, and sets operating mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CKS[1:0]		-	-	UIEN	OIEN	IIEN	-	CHNL[1:0]		DWL[2:0]			SWL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	-	TRMD	EN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	CKS[1:0]	00	R/W	Oversampling Clock Select These bits select the clock source for oversampling.

CKS[1:0] Setting	Clock Operating Mode		
	0 or 1	2	3
00	AUDIO_X1 input		
01	AUDIO_CLK input*		
10	EXTAL input	CKIO input	Setting prohibited
11	Setting prohibited		

Note: * When using AUDIO_CLK, set the PH15MD0 bit in port control register L4 (PHCRL4) to 1.

29, 28	—	All 0	R	Reserved The read value is undefined. The write value should always be 0.
27	UIEN	0	R/W	Underflow Interrupt Enable 0: Underflow interrupt is disabled. 1: Underflow Interrupt is enabled.
26	OIEN	0	R/W	Overflow Interrupt Enable 0: Overflow interrupt is disabled. 1: Overflow interrupt is enabled.

Bit	Bit Name	Initial Value	R/W	Description
25	IIEN	0	R/W	Idle Mode Interrupt Enable 0: Idle mode interrupt is disabled. 1: Idle mode interrupt is enabled.
24	—	0	R	Reserved The read value is undefined. The write value should always be 0.
23, 22	CHNL[1:0]	00	R/W	Channels These bits show the number of channels in each system word. 00: Having one channel per system word 01: Having two channels per system word 10 Having three channels per system word 11: Having four channels per system word
21 to 19	DWL[2:0]	000	R/W	Data Word Length Indicates the number of bits in a data word. 000: 8 bits 001: 16 bits 010: 18 bits 011: 20 bits 100: 22 bits 101: 24 bits 110: 32 bits 111: Reserved
18 to 16	SWL[2:0]	000	R/W	System Word Length Indicates the number of bits in a system word. 000: 8 bits 001: 16 bits 010: 24 bits 011: 32 bits 100: 48 bits 101: 64 bits 110: 128 bits 111: 256 bits

Bit	Bit Name	Initial Value	R/W	Description
15	SCKD	0	R/W	Serial Bit Clock Direction 0: Serial bit clock is input, slave mode. 1: Serial bit clock is output, master mode. Note: Only the following settings are allowed: (SCKD, SWSD) = (0,0) and (1,1). Other settings are prohibited.
14	SWSD	0	R/W	Serial WS Direction 0: Serial word select is input, slave mode. 1: Serial word select is output, master mode. Note: Only the following settings are allowed: (SCKD, SWSD) = (0,0) and (1,1). Other settings are prohibited.
13	SCKP	0	R/W	Serial Bit Clock Polarity 0: SSIWS and SSIDATA change at the SSISCK falling edge (sampled at the SCK rising edge). 1: SSIWS and SSIDATA change at the SSISCK rising edge (sampled at the SCK falling edge).
12	SWSP	0	R/W	Serial WS Polarity 0: SSIWS is low for 1st channel, high for 2nd channel. 1: SSIWS is high for 1st channel, low for 2nd channel.

	SCKP = 0	SCKP = 1
SSIDATA input sampling timing at the time of reception (TRMD = 0)	SSISCK rising edge	SSISCK falling edge
SSIDATA output change timing at the time of transmission (TRMD = 1)	SSISCK falling edge	SSISCK rising edge
SSIWS input sampling timing at the time of slave mode (SWSD = 0)	SSISCK rising edge	SSISCK falling edge
SSIWS output change timing at the time of master mode (SWSD = 1)	SSISCK falling edge	SSISCK rising edge

Bit	Bit Name	Initial Value	R/W	Description
11	SPDP	0	R/W	Serial Padding Polarity 0: Padding bits are low. 1: Padding bits are high.
10	SDTA	0	R/W	Serial Data Alignment 0: Transmitting and receiving in the order of serial data and padding bits 1: Transmitting and receiving in the order of padding bits and serial data
9	PDTA	0	R/W	Parallel Data Alignment This bit is ignored if CPEN = 1. When the data word length is 32, 16 or 8 bit, this configuration field has no meaning. This bit applies to SSIRDR in receive mode and SSITDR in transmit mode. 0: Parallel data (SSITDR, SSIRDR) is left-aligned 1: Parallel data (SSITDR, SSIRDR) is right-aligned. • DWL = 000 (with a data word length of 8 bits), the PDTA setting is ignored. All data bits in SSIRDR or SSITDR are used on the audio serial bus. Four data words are transmitted or received at each 32-bit access. The first data word is derived from bits 7 to 0, the second from bits 15 to 8, the third from bits 23 to 16 and the last data word is derived from bits 31 to 24. • DWL = 001 (with a data word length of 16 bits), the PDTA setting is ignored. All data bits in SSIRDR or SSITDR are used on the audio serial bus. Two data words are transmitted or received at each 32-bit access. The first data word is derived from bits 15 to 0 and the second data word is derived from bits 31 to 16.

Bit	Bit Name	Initial Value	R/W	Description
9	PDTA	0	R/W	• DWL = 010, 011, 100, 101 (with a data word length of 18, 20, 22 or 24 bits), PDTA = 0 (left-aligned) The data bits used in SSIRDR or SSITDR are the following: Bits 31 down to (32 minus the number of bits in the data word length specified by DWL). That is, If DWL = 011, the data word length is 20 bits; therefore, bits 31 to 12 in either SSIRDR or SSITDR are used. All other bits are ignored or reserved. • DWL = 010, 011, 100, 101 (with a data word length of 18, 20, 22 or 24 bits), PDTA = 1 (right-aligned) The data bits used in SSIRDR or SSITDR are the following: Bits (the number of bits in the data word length specified by DWL minus 1) to 0 i.e. if DWL = 011, then DWL = 20 and bits 19 to 0 are used in either SSIRDR or SSITDR. All other bits are ignored or reserved. • DWL = 110 (with a data word length of 32 bits), the PDTA setting is ignored. All data bits in SSIRDR or SSITDR are used on the audio serial bus.
8	DEL	0	R/W	Serial Data Delay 0: 1 clock cycle delay between SSIWS and SSIDATA 1: No delay between SSIWS and SSIDATA

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	CKDV[3:0]	0000	R/W	Serial Oversampling Clock Division Ratio Sets the ratio between the oversampling clock (AUDIOϕ) and the serial bit clock. When the SCKD bit is 0, the setting of these bits is ignored. The serial bit clock is used in the shift register and is supplied from the SSISCK pin. 0000: AUDIOϕ 0001: AUDIOϕ/2 0010: AUDIOϕ/4 0011: AUDIOϕ/8 0100: AUDIOϕ/16 0101: AUDIOϕ/32 0110: AUDIOϕ/64 0111: AUDIOϕ/128 1000: AUDIOϕ/6 1001: AUDIOϕ/12 1010: AUDIOϕ/24 1011: AUDIOϕ/48 1100: AUDIOϕ/96 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
3	MUEN	0	R/W	Mute Enable 0: Module is not muted. 1: Module is muted. Note: When this module is muted, the value of outputting serial data is re-written to 0 but data transmission is not stopped. Write dummy data to the SSIFDR not to generate a transmit underflow because the number of data in the FIFO is decreasing.
2	—	0	R	Reserved The read value is undefined. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	TRMD	0	R/W	Transmit/Receive Mode Select 0: Module is in receive mode. 1: Module is in transmit mode.
0	EN	0	R/W	SSIF Module Enable 0: Module is disabled. 1: Module is enabled.

19.3.2 Status Register (SSISR)

SSISR consists of status flags indicating the operational status of the SSIF module and bits indicating the current channel numbers and word numbers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	UIRQ	OIRQ	IIRQ	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	0	0	1	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	CHNO[1:0]		SWNO	IDST
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * The bit can be read or written to. Writing 0 initializes the bit, but writing 1 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
27	UIRQ	0	R/(W)*	Underflow Error Interrupt Status Flag This status flag indicates that data was supplied at a lower rate than was required. In either case, this bit is set to 1 regardless of the value of the UIEN bit and can be cleared by writing 0 to this bit. If UIRQ = 1 and UIEN = 1, an interrupt occurs. • TRMD = 0 (Receive mode) If UIRQ = 1, SSIRDR was read while the FIFO is empty (DC = H'0). This can cause invalid receive data to be stored, which may lead to corruption of multi-channel data. • TRMD = 1 (Transmit mode) If UIRQ = 1, SSITDR did not have data written to it before it was required for transmission. This will lead to the same sample being transmitted once more and a potential corruption of multi-channel data. This error is more serious than an underflow in receive mode since the SSIF will output erroneous data. Note: When an underflow error occurs, the current data in the data buffer of this module is transmitted until the next data is written.
26	OIRQ	0	R/(W)*	Overflow Error Interrupt Status Flag This status flag indicates that data was supplied at a higher rate than was required. In either case this bit is set to 1 regardless of the value of the OIEN bit and can be cleared by writing 0 to this bit. If OIRQ = 1 and OIEN = 1, an interrupt occurs. • TRMD = 0 (Receive mode) If OIRQ = 1, SSIRDR was not read before there was new unread data written to it. This will lead to the loss of a sample and a potential corruption of multi-channel data. Note: When overflow error occurs, the current data in the data buffer of this module is overwritten by the next incoming data from the SSI interface. • TRMD = 1 (Transmit mode) If OIRQ = 1, SSITDR had data written to it while the FIFO is full (DC = H'8). This will lead to the loss of a sample and a potential corruption of multi-channel data.

Bit	Bit Name	Initial Value	R/W	Description
25	IIRQ	1	R	Idle Mode Interrupt Status Flag This interrupt status flag indicates whether the SSIF module is in idle state. This bit is set regardless of the value of the IIEN bit to allow polling. The interrupt can be masked by clearing IIEN, but cannot be cleared by writing to this bit. If IIRQ = 1 and IIEN = 1, an interrupt occurs. 0: The SSIF module is not in idle state. 1: The SSIF module is in idle state.
24 to 4	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
3, 2	CHNO [1:0]	00	R	Channel Number These bits show the current channel number. • TRMD = 0 (Receive mode) CHNO indicates which channel the data in SSIRDR currently represents. This value will change as the data in SSIRDR is updated from the shift register. • TRMD = 1 (Transmit mode) CHNO indicates which channel is required to be written to SSITDR. This value will change as the data is copied to the shift register, regardless of whether the data is written to SSITDR.
1	SWNO	1	R	System Word Number This status bit indicates the current word number. • TRMD = 0 (Receive mode) SWNO indicates which system word the data in SSIRDR currently represents. This value will change as the data in SSIRDR is updated from the shift register, regardless of whether SSIRDR has been read. • TRMD = 1 (Transmit mode) SWNO indicates which system word is required to be written to SSITDR. This value will change as the data is copied to the shift register, regardless of whether the data is written to SSITDR.

Bit	Bit Name	Initial Value	R/W	Description
0	IDST	1	R	Idle Mode Status Flag This status flag indicates that the serial bus activity has stopped. This bit is cleared if EN = 1 and the serial bus are currently active. This bit is automatically set to 1 under the following conditions. • SSIF = Master transmitter (SWSD = 1 and TRMD = 1) This bit is set to 1 if all the data in the system word to be transmitted has been written to SSITDR and if the EN bit is cleared to end the system word currently being output. • SSIF = Master receiver (SWSD = 1 and TRMD = 0) This bit is set to 1 if the EN bit is cleared and the current system word is completed. • SSIF = Slave transmitter/receiver (SWSD = 0) This bit is set to 1 if the EN bit is cleared and the current system word is completed. Note: If the external master stops the serial bus clock before the current system word is completed, this bit is not set.

Note: * The bit can be read or written to. Writing 0 initializes the bit, but writing 1 is ignored.

19.3.3 Transmit Data Register (SSITDR)

SSITDR is a 32-bit register that stores data to be transmitted. The data for transmission to be stored to SSITDR is automatically transferred from the FIFO data register.

Data written to this register is transferred to the shift register upon transmission request. If the data word length is less than 32 bits, the alignment is determined by the setting of the PDTA control bit in SSICR.

The CPU cannot read or write data from/to SSITDR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

19.3.4 Receive Data Register (SSIRDR)

SSIRDR is a 32-bit register that stores received data. The received data stored in SSIRDR is automatically transferred to the FIFO data register.

Data in this register is transferred from the shift register each time data word is received. If the data word length is less than 32 bits, the alignment is determined by the setting of the PDTA control bit in SSICR.

The CPU cannot read or write data from/to SSIRDR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

19.3.5 FIFO Control Register (SSIFCR)

SSIFCR is a readable/writable 32-bit register that specifies the data trigger numbers and selects between transmission and reception for the FIFO data register, and enables or disables FIFO data reset and interrupt requests.

SSIFCR can always be read or written by the CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TTRG[1:0]		RTRG[1:0]		-	TIE	RIE	FRST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7, 6	TTRG[1:0]	00	R/W	Transmit Data Trigger Number These bits specify the number of transmit data bytes in the FIFO (transmit trigger number) at which the TDE flag in the FIFO status register (SSIFSR) is set while the FIFO is operating for transmission. The TDE flag is set to 1 when the number of transmit data bytes in the FIFO data register (SSIFDR) has become equal to or less than the set trigger number shown below. 00: 7 (1)* 01: 6 (2)* 10: 4 (4)* 11: 2 (6)* Note: * The values in parenthesis are the number of empty stages in SSIFDR at which the TDE flag is set.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	RTRG[1:0]	00	R/W	Receive Data Trigger Number These bits specify the number of received data bytes in the FIFO (receive trigger number) at which the RDF flag in the FIFO status register (SSIFSR) is set while the FIFO is operating for reception. The RDF flag is set to 1 when the number of received data bytes in the FIFO data register (SSIFDR) has become equal to or greater than the set trigger number shown below. 00: 1 01: 2 10: 4 11: 6
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	TIE	0	R/W	Transmit Interrupt Enable Enables or disables generation of transmit data empty interrupt (TXI) requests in the following situation: when the FIFO is operating for transmission, the data for transmission in the FIFO data register (SSIFDR) is transferred to the transmit data register (SSITDR) and the number of data bytes in the FIFO data register has become less than the set transmit trigger number; and thus the TDE flag in the FIFO status register (SSIFSR) is set to 1. 0: Transmit data empty interrupt (TXI) request is disabled 1: Transmit data empty interrupt (TXI) request is enabled* Note: * TXI can be cleared by clearing either the TDE flag (see the description of the TDE bit for details) or TIE bit.

Bit	Bit Name	Initial Value	R/W	Description
1	RIE	0	R/W	Receive Interrupt Enable Enables or disables generation of receive data full interrupt (RXI) requests when the RDF flag in the FIFO status register (SSIFSR) is set to 1 while the FIFO is operating for reception. 0: Receive data full interrupt (RXI) request is disabled 1: Receive data full interrupt (RXI) request is enabled* Note: * RXI can be cleared by clearing either the RDF flag (see the description of the RDF bit for details) or RIE bit.
0	FRST	0	R/W	FIFO Data Register Reset Invalidates the data in the FIFO data register to reset the FIFO to an empty state. 0: Reset is disabled 1: Reset is enabled Note: FIFO is reset at a power-on reset.

19.3.6 FIFO Status Register (SSIFSR)

SSIFSR consists of status flags indicating the operating status of the FIFO data register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	DC[3:0]				-	-	-	-	-	-	TDE	RDF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*

Note: * The bit can be read or written to. Writing 0 initializes the bit, but writing 1 is ignored.

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 8	DC[3:0]	0000	R	Number of Data Bytes Stored in SSIFDR • Transmission DC[3:0] = H'0 indicates no data for transmission. DC[3:0] = H'8 indicates that 32 bytes of data for transmission is stored in SSIFDR. • Reception DC[3:0] = H'0 indicates no received data. DC[3:0] = H'8 indicates that 32 bytes of received data is stored in SSIFDR.
7 to 2	—	All 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
1	TDE	1	R(/W)*	Transmit Data Empty Indicates that, when the FIFO is operating for transmission, the data for transmission in the FIFO data register (SSIFDR) is transferred to the transmit data register (SSITDR), the number of data bytes in the FIFO data register has become less than the transmit trigger number specified by TTRG[1:0] in the FIFO control register (SSIFCR), and thus writing of data transmission to SSIFDR has been enabled. 0: Number of data bytes for transmission in SSIFDR is greater than the set transmit trigger number. [Clearing conditions] • "0" is written to TDE after data of the number of bytes larger than the set transmit trigger number is written to SSIFDR. • The DMAC is activated by transmit data empty (TXI) interrupt, and data of the number of bytes larger than the set transmit trigger number is written to SSIFDR. 1: Number of data bytes for transmission in SSIFDR is equal to or less than the set transmit trigger number. [Setting conditions] • Power-on reset • Number of bytes of data for transmission in SSIFDR has become equal to or less than the set transmit trigger number.*[1] Note: 1. Since SSIFDR is an 8-stage FIFO register, the amount of data that can be written to it while TDE = 1 is "8 – set transmit trigger number" bytes at maximum. Writing more data will be ignored. The number of data bytes in SSIFDR is indicated in the DC bits in SSIFSR.

Bit	Bit Name	Initial Value	R/W	Description
0	RDF	0	R(/W)*	Receive Data Full Indicates that, when the FIFO is operating for reception, the received data is transferred to the FIFO data register (SSIFDR) and the number of data bytes in the FIFO data register has become greater than the receive trigger number specified by RTRG[1:0] in the FIFO control register (SSIFCR). 0: Number of received data bytes in SSIFDR is less than the set receive trigger number. [Clearing conditions] • Power-on reset • “0” is written to RDF after the FIFO is empty with writing 1 to FRST. • “0” is written to RDF after data is read from SSIFDR until the number of data bytes in SSIFDR becomes less than the set receive trigger number. • The DMAC is activated by receive data full (RXI) interrupt, and data is read from SSIFDR until the number of data bytes in SSIFDR becomes less than the set receive trigger number. 1: Number of received data bytes in SSIFDR is equal to or greater than the set receive trigger number. [Setting condition] • Data of the number of bytes that is equal to or greater than the set receive trigger number is stored in SSIFDR.*[1] Note: 1. Since SSIFDR is an 8-stage FIFO register, the amount of data that can be read from it while RDF = 1 is the set receive trigger number of bytes at maximum. Continuing to read data from SSIFDR after reading all the data will result in lack of data. The number of data bytes in SSIFDR is indicated in the DC bits in SSIFSR.

Note: * The bit can be read or written to. Writing 0 initializes the bit, but writing 1 is ignored.

19.3.7 FIFO Data Register (SSIFDR)

In transmission, SSIFDR operates as a FIFO register consisting of eight stages of 32-bit registers for storing data to be serially transmitted. On detecting that the transmit data register (SSITDR) is empty, the SSIF transfers the data for transmission written to SSIFDR to SSITDR to start serial transmission, which can continue until SSIFDR becomes empty. SSIFDR can be written to by the CPU at any time. Note that when SSIFDR is full of data (32 bytes), the next data cannot be written to it and will be ignored if writing is attempted.

In reception, SSIFDR operates as a FIFO register consisting of eight stages of 32-bit registers for storing serially received data. When four bytes of data have been received, the SSIF transfers the received data in the receive data register (SSIRDR) to SSIFDR to complete reception operation. Reception can continue until 32 bytes of data have been stored to SSIFDR. SSIFDR can be read by the CPU but cannot be written to. Note that when SSIFDR is read when it stores no received data, undefined values will be read. After SSIFDR becomes full of received data, the data received thereafter will be lost.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Not writable during reception.

19.4 Operation Description

19.4.1 Bus Format

The SSIF module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode.

The bus format can be selected from one of the four major modes shown in table 19.3.

Table 19.3 Bus Format for SSIF Module

	Non-Compression Slave Receiver	Non-Compression Slave Transmitter	Non-Compression Master Receiver	Non-Compression Master Transmitter
TRMD	0	1	0	1
SCKD	0	0	1	1
SWSD	0	0	1	1
EN	Control Bits			
MUEN				
DIEN				
IIEN				
OIEN				
UIEN				
DEL	Configuration Bits			
PDTA				
SDTA				
SPDP				
SWSP				
SCKP				
SWL [2:0]				
DWL [2:0]				
CHNL [1:0]				

19.4.2 Non-Compressed Modes

The non-compressed modes support all serial audio streams split into channels. It supports the I^2S compatible format as well as many more variants on these modes.

(1) Slave Receiver

This mode allows the module to receive serial data from another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of the SSIF module, operation is not guaranteed.

(2) Slave Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of the SSIF module, operation is not guaranteed.

(3) Master Receiver

This mode allows the module to receive serial data from another device. The clock and word select signals are internally derived from the oversampling clock. The format of these signals is defined in the configuration fields of the SSIF module. If the incoming data does not follow the configured format, operation is not guaranteed.

(4) Master Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signals are internally derived from the oversampling clock. The format of these signals is defined in the configuration fields of the SSIF module.

(5) Operating Setting Related to Word Length

All bits related to the SSICR's word length are valid in non-compressed modes. There are many configurations the SSIF module supports, but some of the combinations are shown below for the I^2S compatible format, MSB-first and left-aligned format, and MSB-first and right-aligned format.

Figures 19.3 and 19.4 demonstrate the supported I²S compatible format both without and with padding. Padding occurs when the data word length is smaller than the system word length.

- I²S Compatible Format

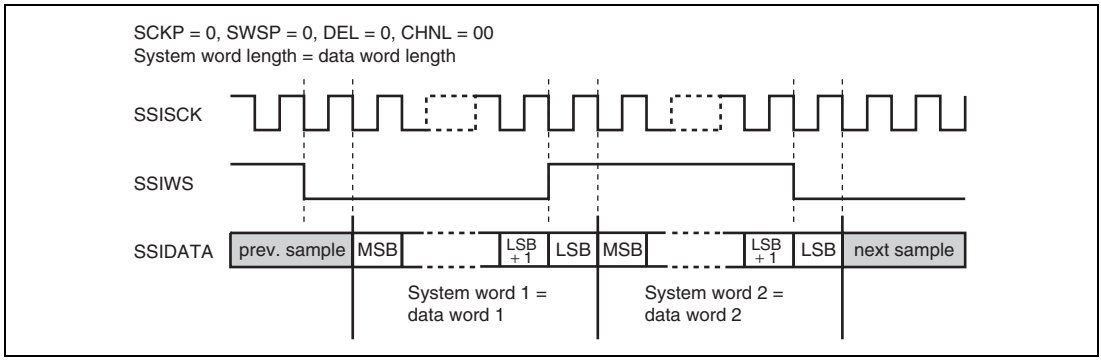


Figure 19.3 I²S Compatible Format (without Padding)

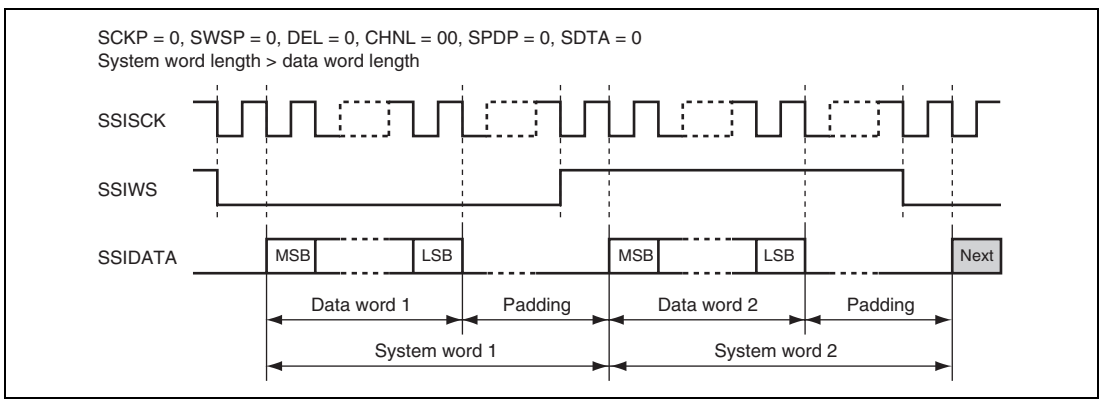


Figure 19.4 I²S Compatible Format (with Padding)

Figure 19.5 shows the MSB-first and left-aligned format and figure 19.6 shows the MSB-first and right-aligned format. Padding is assumed in both cases, but may not be present in a final implementation if the system word length equals the data word length.

- MSB-first and Left-Aligned Format

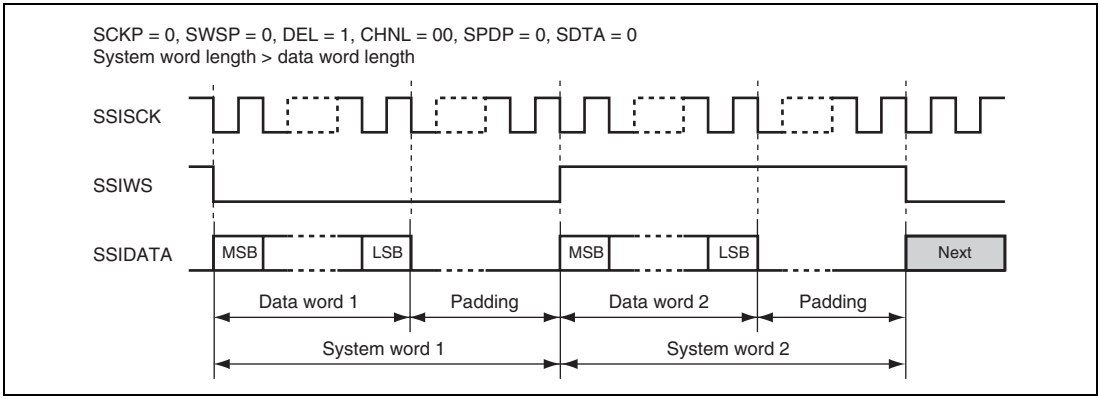


Figure 19.5 MSB-first and Left-Aligned Format
(Transmitted and received in the order of serial data and padding bits)

- MSB-first and Right-Aligned Format

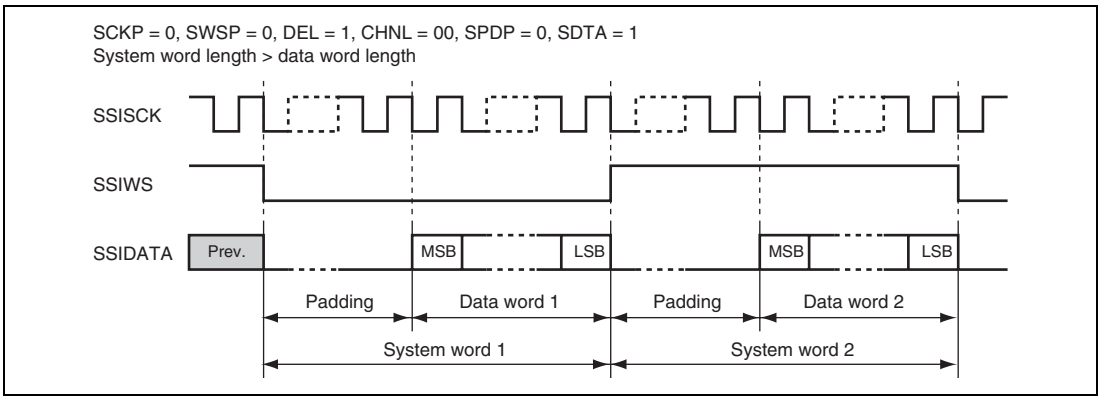


Figure 19.6 MSB-first and Right-Aligned Format
(Transmitted and received in the order of padding bits and serial data)

(6) Multi-channel Formats

Some devices extend the definition of the specification by I^2S bus and allow more than 2 channels to be transferred within two system words.

The SSIF module supports the transfer of 4, 6 and 8 channels by using the CHNL, SWL and DWL bits only when the system word length (SWL) is greater than or equal to the data word length (DWL) multiplied by channels (CHNL).

Table 19.4 shows the number of padding bits for each of the valid setting. If setting is not valid, "—" is indicated instead of a number.

Table 19.4 The Number of Padding Bits for Each Valid Setting

Padding Bits Per System Word			DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	Decoded Channels per System Word	SWL [2:0]	Decoded Word Length	8	16	18	20	22	24	32
00	1	000	8	0	—	—	—	—	—	—
		001	16	8	0	—	—	—	—	—
		010	24	16	8	6	4	2	0	—
		011	32	24	16	14	12	10	8	0
		100	48	40	32	30	28	26	24	16
		101	64	56	48	46	44	42	40	32
		110	128	120	112	110	108	106	104	96
		111	256	248	240	238	236	234	232	224
01	2	000	8	—	—	—	—	—	—	—
		001	16	0	—	—	—	—	—	—
		010	24	8	—	—	—	—	—	—
		011	32	16	0	—	—	—	—	—
		100	48	32	16	12	8	4	0	—
		101	64	48	32	28	24	20	16	0
		110	128	112	96	92	88	84	80	64
		111	256	240	224	220	216	212	208	192

Padding Bits Per System Word		DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	Decoded Channels per System Word	SWL [2:0]	Decoded Word Length	8	16	18	20	22	24	32
10	3	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	0	—	—	—	—	—	—
		011	32	8	—	—	—	—	—	—
		100	48	24	0	—	—	—	—	—
		101	64	40	16	10	4	—	—	—
		110	128	104	80	74	68	62	56	32
		111	256	232	208	202	196	190	184	160
11	4	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	—	—	—	—	—	—	—
		011	32	0	—	—	—	—	—	—
		100	48	16	—	—	—	—	—	—
		101	64	32	0	—	—	—	—	—
		110	128	96	64	56	48	40	32	0
		111	256	224	192	184	176	168	160	128

When the SSIF module acts as a transmitter, each word written to SSITDR is transmitted to the serial audio bus in the order they are written. When the SSIF module acts as a receiver, each word received by the serial audio bus is read in the order received from the SSIRDR register.

Figures 19.7 to 19.9 show how the data on 4, 6 and 8 channels are transferred to the serial audio bus. Note that there are no padding bits in the first example, the second example is left-aligned and the third is right-aligned. The other conditions in these examples have been selected arbitrarily.

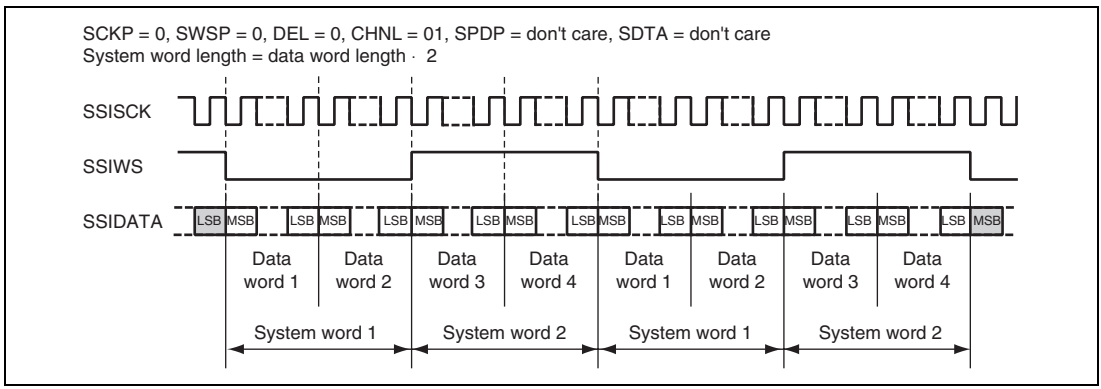


Figure 19.7 Multi-Channel Format (4 Channels Without Padding)

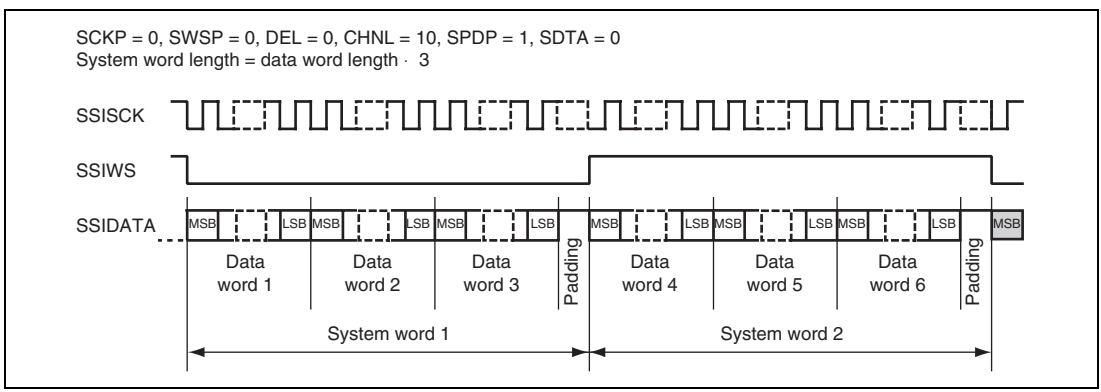


Figure 19.8 Multi-Channel Format (6 Channels with High Padding)

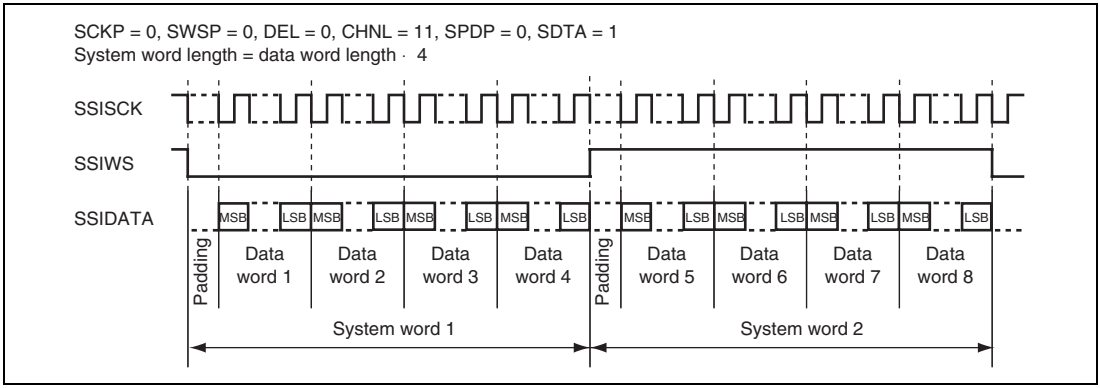


Figure 19.9 Multi-Channel Format (8 Channels; Transmitting and Receiving in the order of serial data and Padding Bits; with padding)

(7) Bit Setting Configuration Format

Several more configuration bits in non-compressed mode are shown below. These bits are not mutually exclusive, but some combinations may not be useful for any other device.

These configuration bits are described below with reference to figure 19.10.

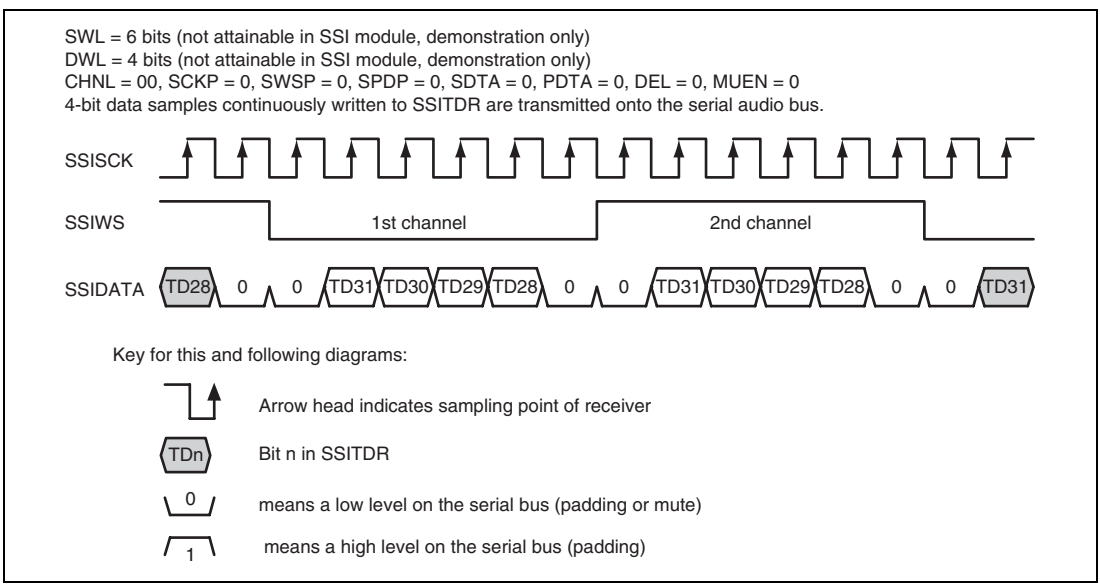


Figure 19.10 Basic Sample Format (Transmit Mode with Example System/Data Word Length)

Figure 19.10 uses a system word length of 6 bits and a data word length of 4 bits. These settings are not possible with the SSIF module but are used only for clarification of the other configuration bits.

- Inverted Clock

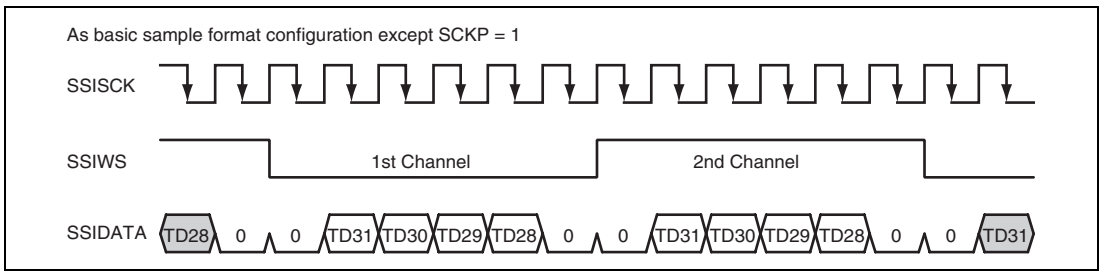


Figure 19.11 Inverted Clock

- Inverted Word Select

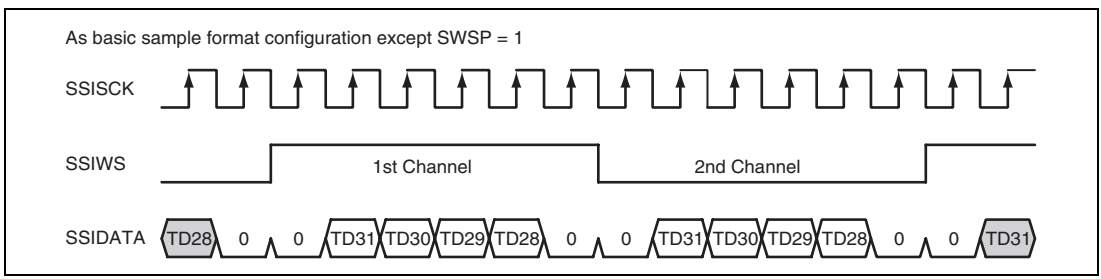


Figure 19.12 Inverted Word Select

- Inverted Padding Polarity

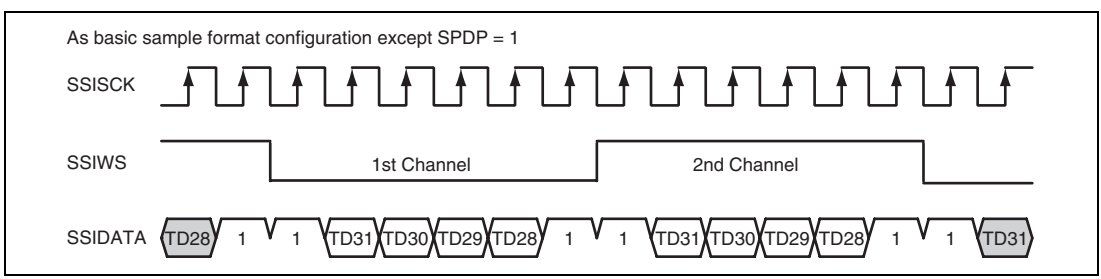


Figure 19.13 Inverted Padding Polarity

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

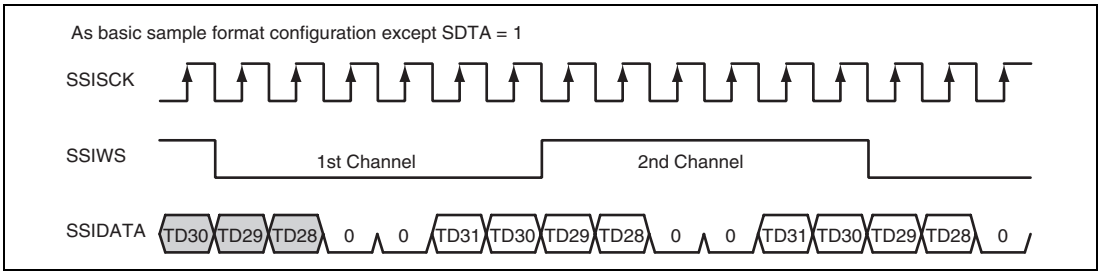


Figure 19.14 Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

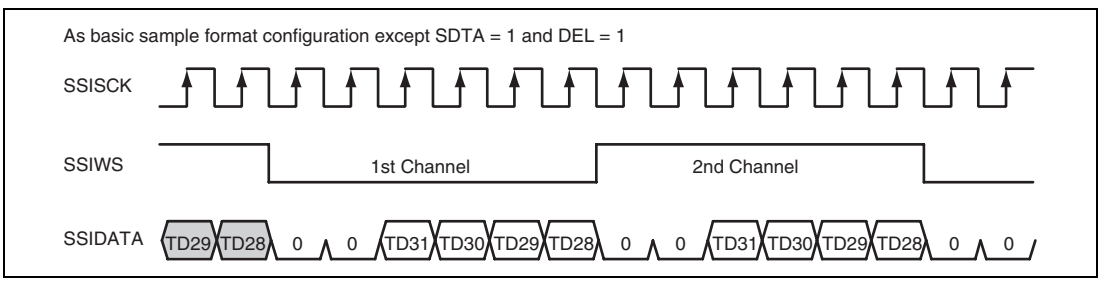


Figure 19.15 Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

- Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

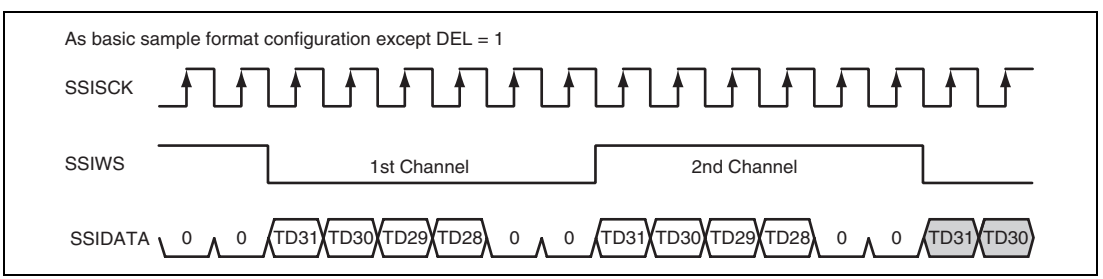


Figure 19.16 Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

- Parallel Right-Aligned with Delay

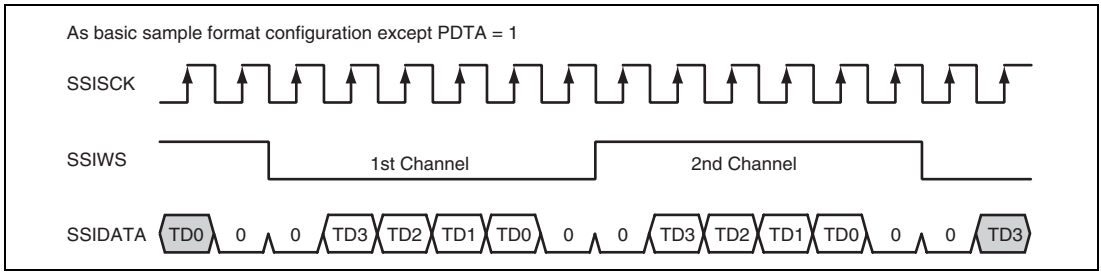


Figure 19.17 Parallel Right-Aligned with Delay

- Mute Enabled

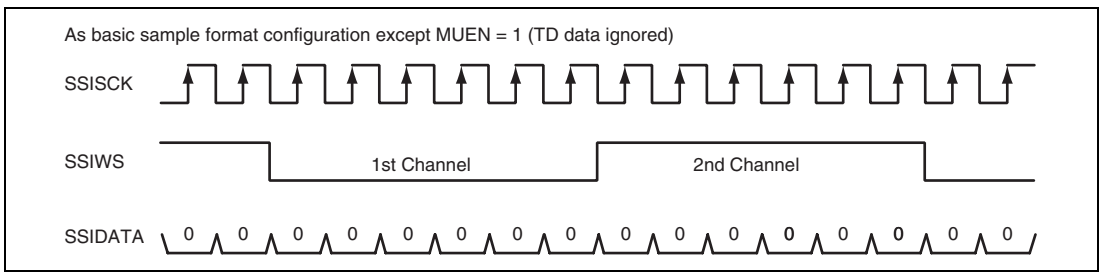


Figure 19.18 Mute Enabled

19.4.3 Operation Modes

There are three modes of operation: configuration, enabled and disabled. Figure 19.19 shows how the module enters each of these modes.

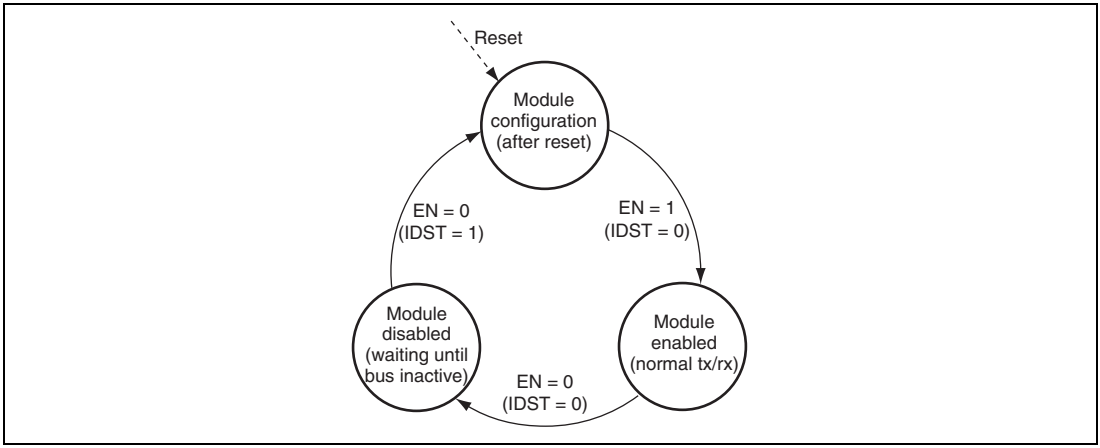


Figure 19.19 Operation Modes

(1) Configuration Mode

This mode is entered after the module is released from reset. All required configuration fields in the control register should be defined in this mode, before the SSIF module is enabled by setting the EN bit.

Setting the EN bit causes the module to enter the module enabled mode.

(2) Module Enabled Mode

Operation of the module in this mode is dependent on the operation mode selected. For details, refer to section 19.4.4, Transmit Operation and section 19.4.5, Receive Operation, below.

19.4.4 Transmit Operation

Transmission can be controlled either by DMA or interrupt.

DMA control is preferred to reduce the processor load. In DMA control mode the processor will only receive interrupts if there is an underflow or overflow of data or the DMAC has finished its transfer.

The alternative method is using the interrupts that the SSIF module generates to supply data as required.

When disabling the SSIF module, the clock* must be kept supplied to the SSIF until the IIRQ bit indicates that the module is in the idle state.

Figure 19.20 shows the transmit operation in DMA control mode, and figure 19.21 shows the transmit operation in interrupt control mode.

Note: * Input clock from the SSISCK pin when SCKD = 0.
Oversampling clock when SCKD = 1.

(1) Transmission Using DMA Controller

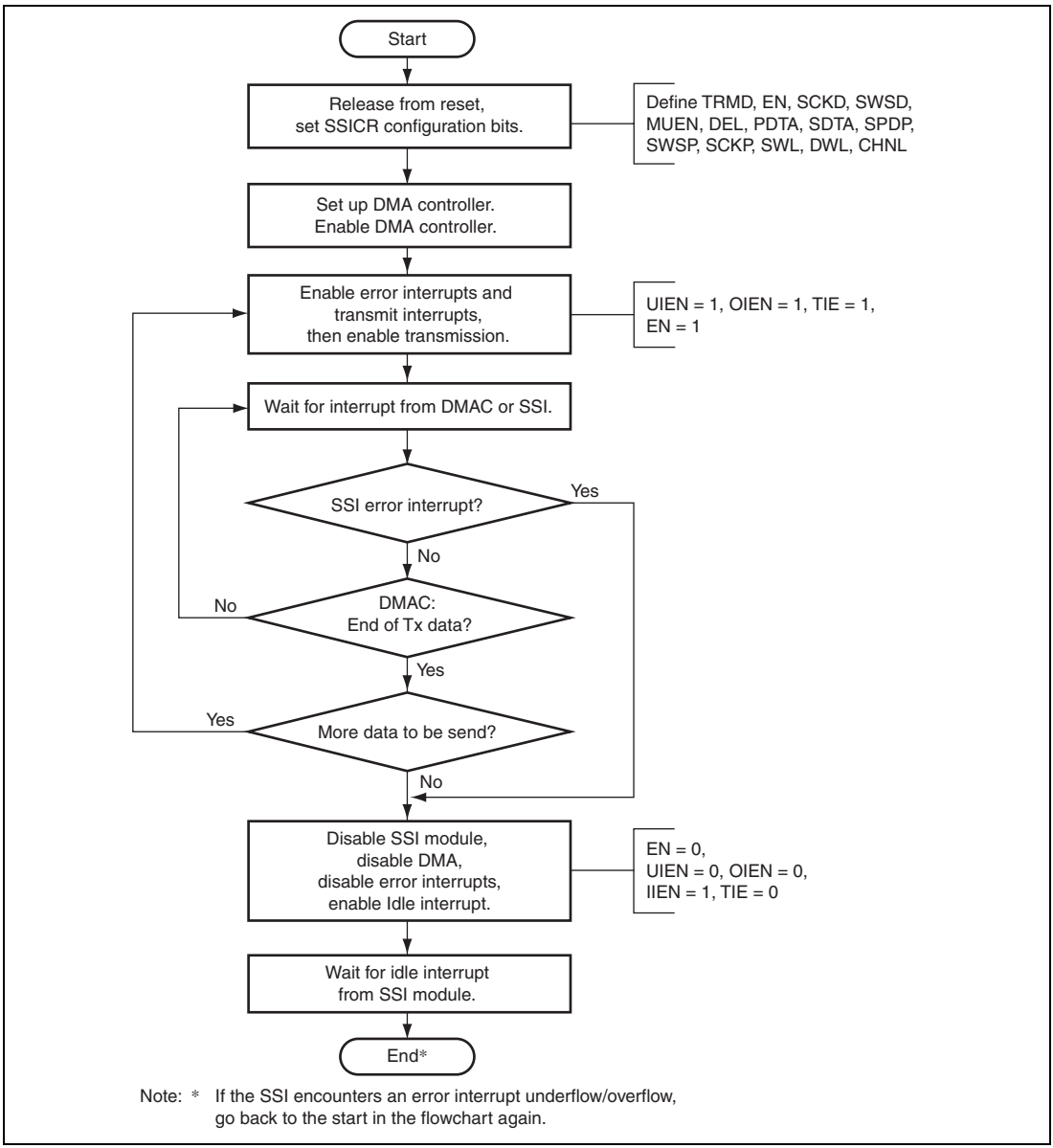


Figure 19.20 Transmission Using DMA Controller

(2) Transmission Using Interrupt Data Flow Control

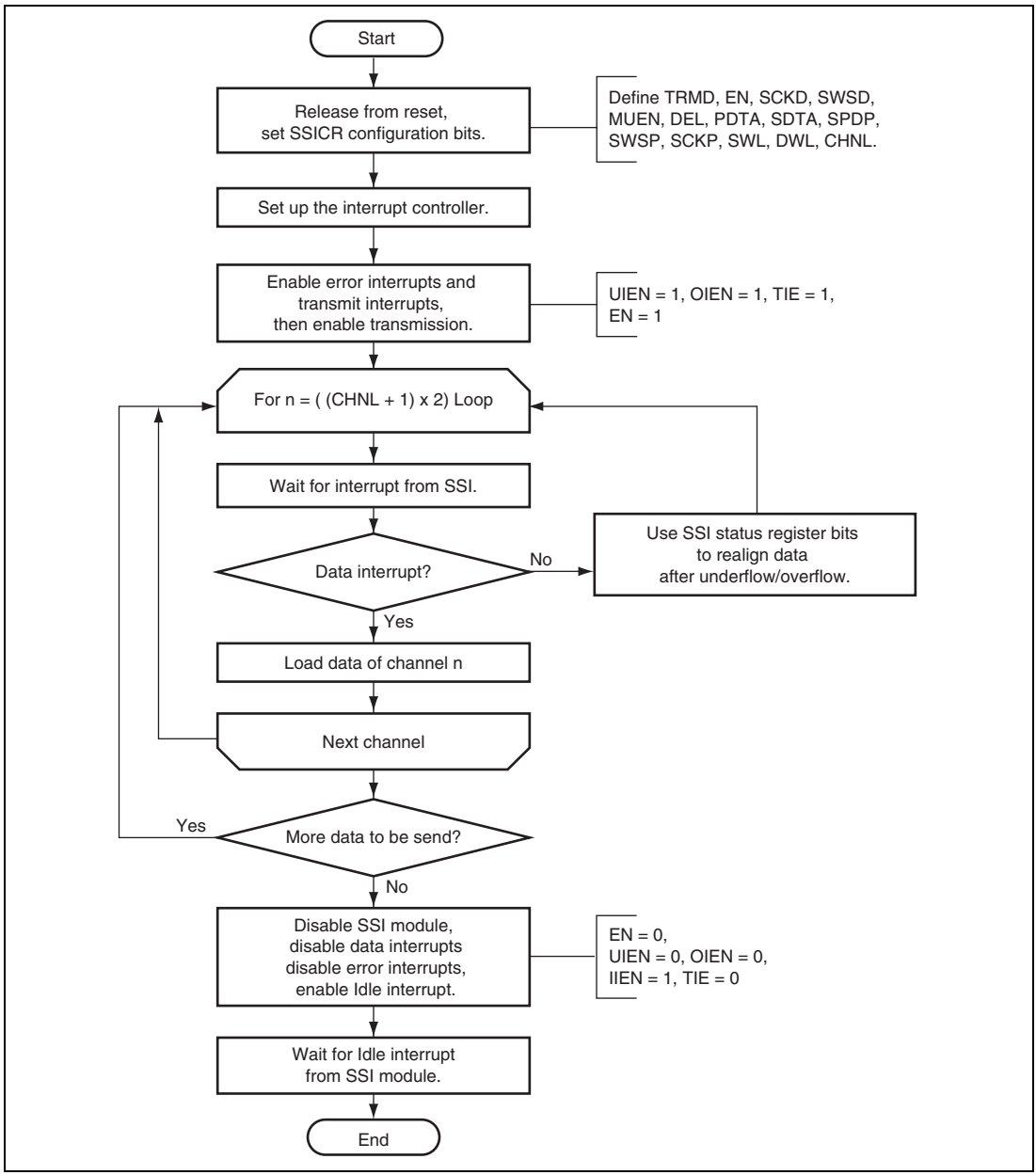


Figure 19.21 Transmission Using Interrupt Data Flow Control

19.4.5 Receive Operation

Like transmission, reception can be controlled either by DMA or interrupt.

Figures 19.22 and 19.23 show the flow of operation.

When disabling the SSIF module, the clock* must be kept supplied to the SSIF until the IIRQ bit indicates that the module is in the idle state.

Note: * Input clock from the SSISCK pin when SCKD = 0.
Oversampling clock when SCKD = 1.

(1) Reception Using DMA Controller

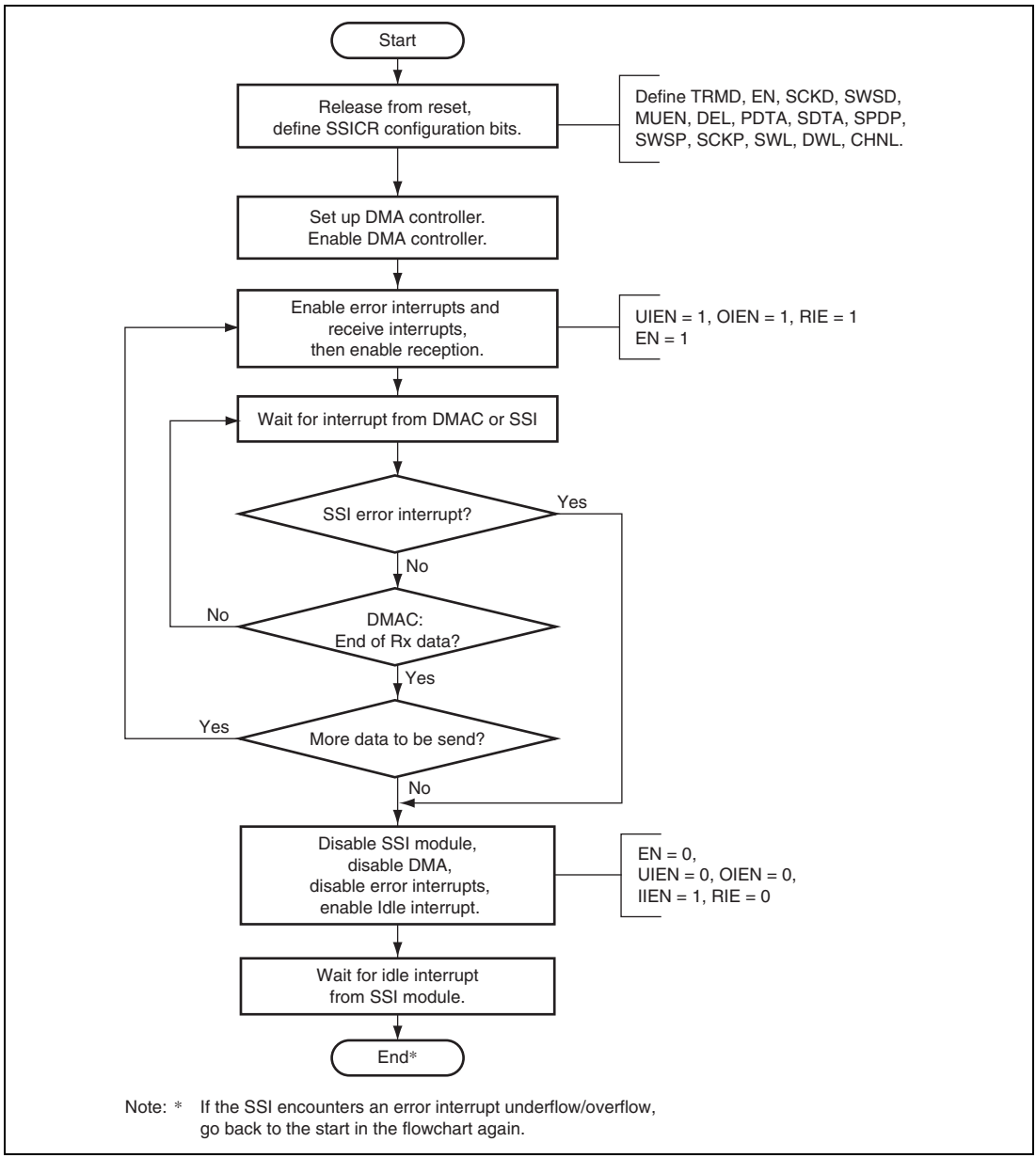


Figure 19.22 Reception Using DMA Controller

(2) Reception Using Interrupt Data Flow Control

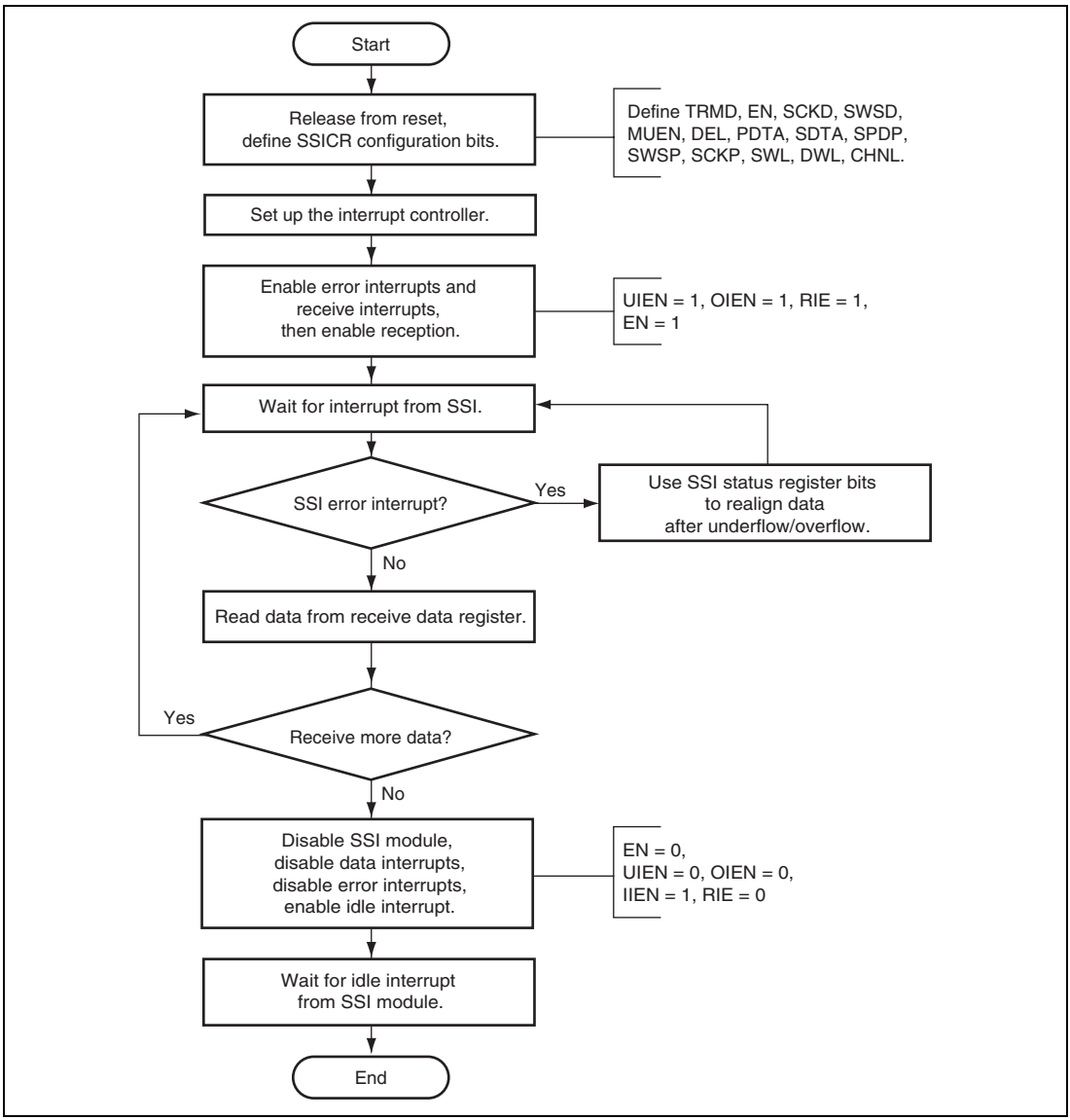


Figure 19.23 Reception Using Interrupt Data Flow Control

When an underflow or overflow error condition has matched, the CHNO [1:0] bit and the SWNO bit can be used to recover the SSIF module to a known status. When an underflow or overflow occurs, the host can read the channel number and system word number to determine what point the serial audio stream has reached. In the transmitter case, the host can skip forward through the data it wants to transmit until it finds the sample data that matches what the SSIF module is expecting to transmit next, and so resynchronize with the audio data stream. In the receiver case the host CPU can store null data to make the number of receive data items consistent until it is ready to store the sample data that the SSIF module is indicating will be received next, and so resynchronize with the audio data stream.

19.4.6 Serial Bit Clock Control

This function is used to control and select which clock is used for the serial bus interface.

If the serial clock direction is set to input (SCKD = 0), the SSIF module is in clock slave mode and the shift register uses the bit clock that was input to the SSISCK pin.

If the serial clock direction is set to output (SCKD = 1), the SSIF module is in clock master mode, and the shift register uses the oversampling clock or a divided oversampling clock as the bit clock. The oversampling clock is divided by the ratio specified by the serial oversampling clock division ratio bits (CKDV) in SSICR for use as the bit clock by the shift register.

In either case the module pin, SSISCK, is the same as the bit clock.

19.5 Usage Notes

19.5.1 Limitations from Underflow or Overflow during DMA Operation

If an underflow or overflow occurs while the DMA is in operation, the module should be restarted. The transmit and receive buffers in the SSI consists of 32-bit registers that share the L and R channels. Therefore, data to be transmitted and received at the L channel may sometimes be transmitted and received at the R channel if an underflow or overflow occurs, for example, under the following condition: the control register (SSICR) has a 32-bit setting for both data word length (DWL2 to DWL0) and system word length (SWL2 to SWL0).

If an error occurrence is confirmed with two types of error interrupts (underflow and overflow) or the corresponding error status flag (the bits UIRQ and OIRQ in SSISR), write 0 to the EN bit in SSICR to disable DMA transfer requests in this module, thus stopping the operation. (In this case, the direct memory access controller setting should also be stopped.) After this, write 0 to the error status flag bit to clear the error status, set the direct memory access controller again and restart the transfer.

Section 20 Controller Area Network (RCAN-TL1)

20.1 Summary

20.1.1 Overview

This document primarily describes the programming interface for the RCAN-TL1 (Renesas CAN Time Trigger Level 1) module. It serves to facilitate the hardware/software interface so that engineers involved in the RCAN-TL1 implementation can ensure the design is successful.

20.1.2 Scope

The CAN Data Link Controller function is not described in this document. It is the responsibility of the reader to investigate the CAN Specification Document (see references). The interfaces from the CAN Controller are described, in so far as they pertain to the connection with the User Interface.

The programming model is described in some detail. It is not the intention of this document to describe the implementation of the programming interface, but to simply present the interface to the underlying CAN functionality.

The document places no constraints upon the implementation of the RCAN-TL1 module in terms of process, packaging or power supply criteria. These issues are resolved where appropriate in implementation specifications.

20.1.3 Audience

In particular this document provides the design reference for software authors who are responsible for creating a CAN application using this module.

In the creation of the RCAN-TL1 user interface LSI engineers must use this document to understand the hardware requirements.

20.1.4 References

1. CAN Specification Version 2.0 part A, Robert Bosch GmbH, 1991
2. CAN Specification Version 2.0 part B, Robert Bosch GmbH, 1991
3. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany, 1997
4. Road vehicles - Controller area network (CAN): Part 1: Data link layer and physical signalling (ISO-11898-1, 2003)

5. Road vehicles - Controller area network (CAN): Part 4: Time triggered communication (ISO-11898-4, 2004)

20.1.5 Features

- Supports CAN specification 2.0B
- Bit timing compliant with ISO-11898-1
- 32 Mailbox version
- Clock 16 to 33 MHz
- 31 programmable Mailboxes for transmit / receive + 1 receive-only mailbox
- Sleep mode for low power consumption and automatic recovery from sleep mode by detecting CAN bus activity
- Programmable receive filter mask (standard and extended identifier) supported by all Mailboxes
- Programmable CAN data rate up to 1MBit/s
- Transmit message queuing with internal priority sorting mechanism against the problem of priority inversion for real-time applications
- Data buffer access without SW handshake requirement in reception
- Flexible micro-controller interface
- Flexible interrupt structure
- 16-bit free running timer with flexible clock sources and pre-scaler, 3 Timer Compare Match Registers
- 6-bit Basic Cycle Counter for Time Trigger Transmission
- Timer Compare Match Registers with interrupt generation
- Timer counter clear / set capability
- Registers for Time-Trigger: Local_Time, Cycle_time, Ref_Mark, Tx_Enable Window, Ref_Trigger_Offset
- Flexible TimeStamp at SOF for both transmission and reception supported
- Time-Trigger Transmission, Periodic Transmission supported (on top of Event Trigger Transmission)
- Basic Cycle value can be embedded into a CAN frame and transmitted

20.2 Architecture

The RCAN-TL1 device offers a flexible and sophisticated way to organise and control CAN frames, providing the compliance to CAN2.0B Active and ISO-11898-1. The module is formed from 5 different functional entities. These are the Micro Processor Interface (MPI), Mailbox, Mailbox Control, Timer, and CAN Interface. The figure below shows the block diagram of the RCAN-TL1 Module. The bus interface timing is designed according to the peripheral bus I/F required for each product.

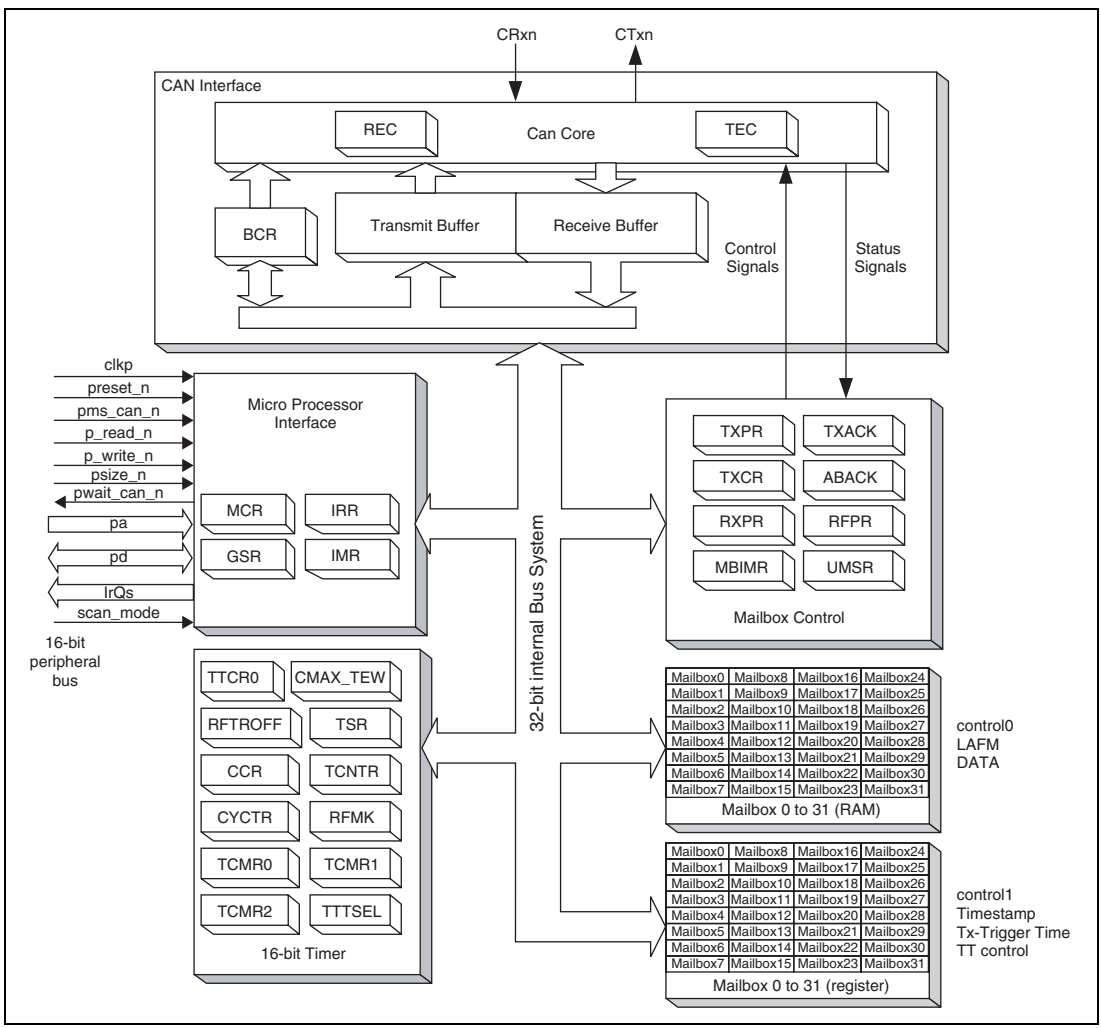


Figure 20.1 RCAN-TL1 architecture

Important: Although core of RCAN-TL1 is designed based on a 32-bit bus system, the whole RCAN-TL1 including MPI for the CPU has 16-bit bus interface to CPU. In that case, LongWord (32-bit) access must be implemented as 2 consecutive word (16-bit) accesses. In this manual, LongWord access means the two consecutive accesses.

- Micro Processor Interface (MPI)

  The MPI allows communication between the Renesas CPU and the RCAN-TL1's registers/mailboxes to control the memory interface. It also contains the Wakeup Control logic that detects the CAN bus activities and notifies the MPI and the other parts of RCAN-TL1 so that the RCAN-TL1 can automatically exit the Sleep mode.

  It contains registers such as MCR, IRR, GSR and IMR.

- Mailbox

  The Mailboxes consists of RAM configured as message buffers and registers. There are 32 Mailboxes, and each mailbox has the following information.

  <RAM>

  — CAN message control (identifier, rtr, ide,etc)

  — CAN message data (for CAN Data frames)

  — Local Acceptance Filter Mask for reception

  <Registers>

  — CAN message control (dlc)

  — Time Stamp for message reception/transmission

  — 3-bit wide Mailbox Configuration, Disable Automatic Re-Transmission bit, Auto-Transmission for Remote Request bit, New Message Control bit

  — Tx-Trigger Time

- Mailbox Control

  The Mailbox Control handles the following functions.

  — For received messages, compare the IDs and generate appropriate RAM addresses/data to store messages from the CAN Interface into the Mailbox and set/clear appropriate registers accordingly.

  — To transmit event-triggered messages, run the internal arbitration to pick the correct priority message, and load the message from the Mailbox into the Tx-buffer of the CAN Interface and set/clear appropriate registers accordingly. In the case of time-triggered transmission, compare match of Tx-Trigger time invoke loading the messages.

  — Arbitrates Mailbox accesses between the CPU and the Mailbox Control.

  — Contains registers such as TXPR, TXCR, TXACK, ABACK, RXPR, RFPR, UMSR and MBIMR.

- Timer

  The Timer function is the functional entity, which provides RCAN-TL1 with support for transmitting messages at a specific time frame and recording the result.

  The Timer is a 16-bit free running up counter which can be controlled by the CPU. It provides one 16-bit Compare Match Register to compare with Local Time and two 16-bit ones to compare with Cycle Time. The Compare Match Registers can generate interrupt signals and clear the Counter.

  The clock period of this Timer offers a wide selection derived from the system clock or can be programmed to be incremented with one nominal bit timing of CAN Bus.

  Contains registers such as TCNTR, TTCR0, CMAX_TEW, RFTROFF, TSR, CCR, CYCTR, RFMK, TCMR0, TCMR1, TCMR2 and TTTSEL.

- CAN Interface

  This block conforms to the requirements for a CAN Bus Data Link Controller which is specified in Ref. [2, 4]. It fulfils all the functions of a standard DLC as specified by the OSI 7 Layer Reference model. This functional entity also provides the registers and the logic which are specific to a given CAN bus, which includes the Receive Error Counter, Transmit Error Counter, the Bit Configuration Registers and various useful Test Modes. This block also contains functional entities to hold the data received and the data to be transmitted for the CAN Data Link Controller.

20.3 Programming Model—Overview

The purpose of this programming interface is to allow convenient, effective access to the CAN bus for efficient message transfer. Please bear in mind that the user manual reports all settings allowed by the RCAN-TL1 IP. Different use of RCAN-TL1 is not allowed.

20.3.1 Memory Map

The diagram of the memory map is shown below.

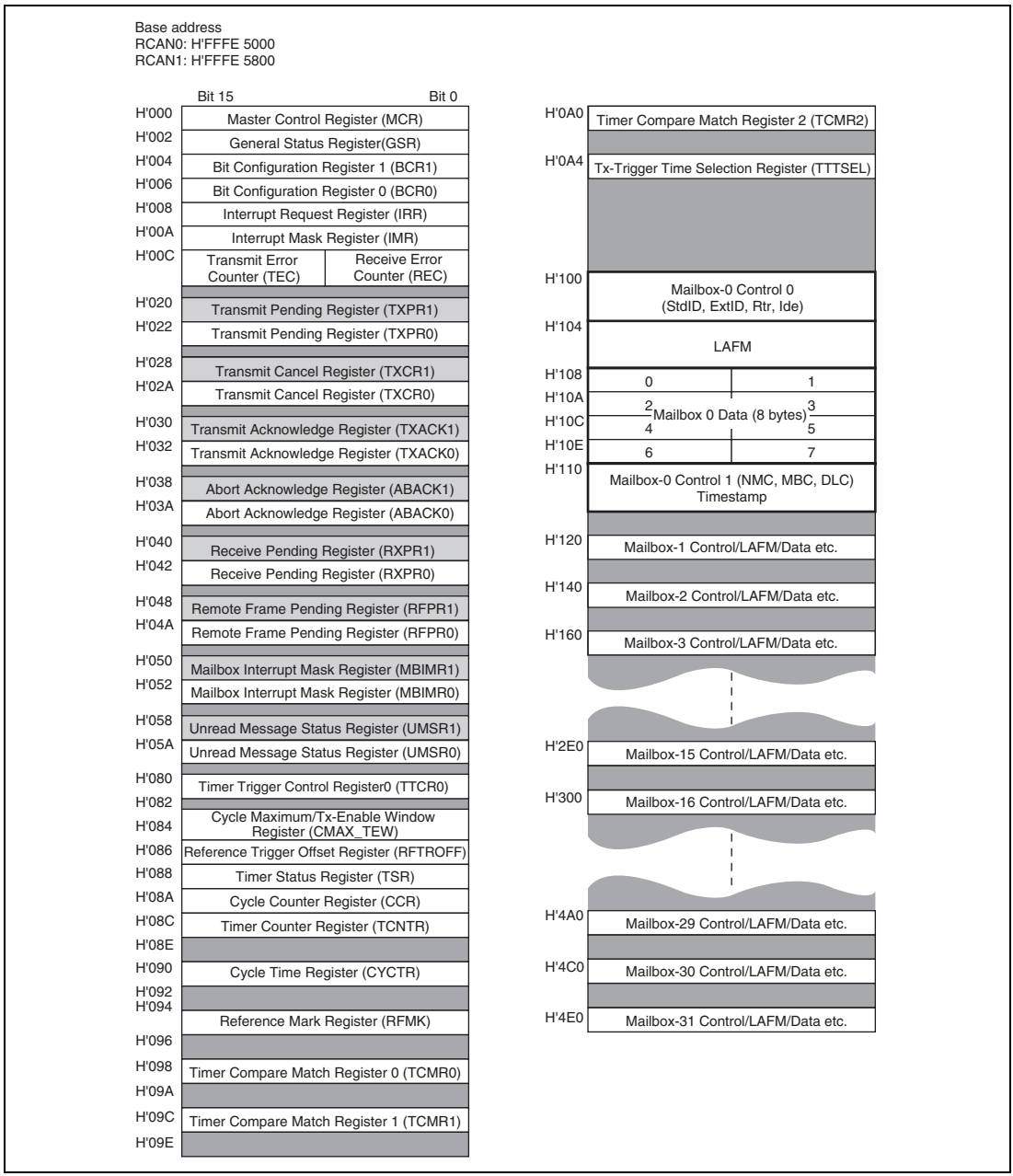


Figure 20.2 RCAN-TL1 Memory Map

The locations not used (between H'000 and H'4F3) are reserved and cannot be accessed.

20.3.2 Mailbox Structure

Mailboxes play a role as message buffers to transmit/receive CAN frames. Each Mailbox is comprised of 3 identical storage fields that are 1): Message Control, 2): Local Acceptance Filter Mask, 3): Message Data. In addition some Mailboxes contain the following extra Fields: 4): Time Stamp, 5): Time Trigger configuration and 6): Time Trigger Control. The following table shows the address map for the control, LAFM, data, timestamp, Transmission Trigger Time and Time Trigger Control addresses for each mailbox.

	Address						
	Control0	LAFM	Data	Control1	Time Stamp	Trigger Time	TT control
Mailbox	4 bytes	4 bytes	8 bytes	2 bytes	2 bytes	2 bytes	2 bytes
0 (Receive Only)	100 – 103	104– 107	108 – 10F	110 – 111	112 – 113	No	No
1	120 – 123	124 – 127	128 – 12F	130 – 131	132 – 133	No	No
2	140 – 143	144 – 147	148 – 14F	150 – 151	152 – 153	No	No
3	160 – 163	164 – 167	168 – 16F	170 – 171	172 – 173	No	No
4	180 – 183	184 – 187	188 – 18F	190 – 191	192 – 193	No	No
5	1A0 – 1A3	1A4 – 1A7	1A8 – 1AF	1B0 – 1B1	1B2 – 1B3	No	No
6	1C0 – 1C3	1C4 – 1C7	1C8 – 1CF	1D0 – 1D1	1D2 – 1D3	No	No
7	1E0 – 1E3	1E4 – 1E7	1E8 – 1EF	1F0 – 1F1	1F2 – 1F3	No	No
8	200 – 203	204 – 207	208 – 20F	210 – 211	212 – 213	No	No
9	220 – 223	224 – 227	228 – 22F	230 – 231	232 – 233	No	No
10	240 – 243	244 – 247	248 – 24F	250 – 251	252 – 253	No	No
11	260 – 263	264 – 267	268 – 26F	270 – 271	272 – 273	No	No
12	280 – 283	284 – 287	288 – 28F	290 – 291	292 – 293	No	No
13	2A0 – 2A3	2A4 – 2A7	2A8 – 2AF	2B0 – 2B1	2B2 – 2B3	No	No
14	2C0 – 2C3	2C4 – 2C7	2C8 – 2CF	2D0 – 2D1	2D2 – 2D3	No	No
15	2E0 – 2E3	2E4 – 2E7	2E8 – 2EF	2F0 – 2F1	2F2 – 2F3	No	No
16	300 – 303	304 – 307	308 – 30F	310 – 311	No	No	No
17	320 – 323	324 – 327	328 – 32F	330 – 331	No	No	No
18	340 – 343	344 – 347	348 – 34F	350 – 351	No	No	No

	Address						
	Control0	LAFM	Data	Control1	Time Stamp	Trigger Time	TT control
Mailbox	4 bytes	4 bytes	8 bytes	2 bytes	2 bytes	2 bytes	2 bytes
19	360 – 363	364 – 367	368 – 36F	370 – 371	No	No	No
20	380 – 383	384 – 387	388 – 38F	390 – 391	No	No	No
21	3A0 – 3A3	3A4 – 3A7	3A8 – 3AF	3B0 – 3B1	No	No	No
22	3C0 – 3C3	3C4 – 3C7	3C8 – 3CF	3D0 – 3D1	No	No	No
23	3E0 – 3E3	3E4 – 3E7	3E8 – 3EF	3F0 – 3F1	No	No	No
24	400 – 403	404 – 407	408 – 40F	410 – 411	No	414 – 415	416 – 417
25	420 – 423	424 – 427	428 – 42F	430 – 431	No	434 – 435	436 – 437
26	440 – 443	444 – 447	448 – 44F	450 – 451	No	454 – 455	456 – 457
27	460 – 463	464 – 467	468 – 46F	470 – 471	No	474 – 475	476 – 477
28	480 – 483	484 – 487	488 – 48F	490 – 491	No	494 – 495	496 – 497
29	4A0 – 4A3	4A4 – 4A7	4A8 – 4AF	4B0 – 4B1	No	4B4 – 4B5	4B6 – 4B7
30	4C0 – 4C3	4C4 – 4C7	4C8 – 4CF	4D0 – 4D1	4D2 – 4D3 (Local Time)	4D4 – 4D5	No
31	4E0 – 4E3	4E4 – 4E7	4E8 – 4EF	4F0 – 4F1	4F2 – 4F3 (Local Time)	No	No

Mailbox-0 is a receive-only box, and all the other Mailboxes can operate as both receive and transmit boxes, dependant upon the MBC (Mailbox Configuration) bits in the Message Control. The following diagram shows the structure of a Mailbox in detail.

Table 20.1 Roles of Mailboxes

	Event Trigger		Time Trigger		Remark	
	Tx	Rx	Tx	Rx	TimeStamp	Tx-Trigger Time
MB31	Settable	Settable	—	Time reference reception	Available	—
MB30	Settable	Settable	Time reference transmission in time master mode	Reception in time slave mode	Available	Available
MB29 - 24	Settable	Settable	Settable	Settable	—	Available
MB23 - 16	Settable	Settable	— (ET)	Settable	—	—
MB15 - 1	Settable	Settable	— (ET)	Settable	Available	—
MB0	—	Settable	—	Settable	Available	—

(ET) shows that it works during merged arbitrating window, after completion of time-triggered transmission.

MB0 (reception MB with timestamp)

Byte: 8-bit access, Word: 16-bit access, LW (LongWord) : 32-bit access

Address	Data Bus 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Access Size	Field Name
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE LAFM	0	0	STDID_LAFM[10:0]											EXTID LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	0	0	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + N*32	TimeStamp[15:0] (CYCTR[15:0] or CCR[5:0]/CYCTR[15:6] at SOF)																Word	TimeStamp

MBC[1] is fixed to "1"

MB15 to 1 (MB with timestamp)

Address	Data Bus 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Access Size	Field Name
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE LAFM	0	0	STDID_LAFM[10:0]											EXTID LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + N*32	TimeStamp[15:0] (CYCTR[15:0] or CCR[5:0]/CYCTR[15:6] at SOF)																Word	TimeStamp

Figure 20.3 Mailbox-N Structure

MB23 to 16 (MB without timestamp)

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1

MB29 to 24 (Time-Triggered Transmission in Time Trigger mode)

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + N*32	reserved																-	-
H'114 + N*32	Tx-Triggered Time (TTT)																Word	Trigger Time
H'116 + N*32	TTW[1:0]		offset						0	0	0	0	0	Rep_Factor			Word	TT control

Figure 20.3 Mailbox-N Structure (continued)

MB30 (Time Reference Transmitssion in Time Trigger mode)

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE LAFM	0	0	STDID_LAFM[10:0]											EXTID_ LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + N*32	TimeStamp[15:0] (TCNTR at SOF)																Word	TimeStamp
H'114 + N*32	Tx-Triggered Time (TTT) as Time Reference																Word	Trigger Time

MB31 (Time Reference Reception in Time Trigger mode)

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE LAFM	0	0	STDID_LAFM[10:0]											EXTID_ LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + N*32	TimeStamp[15:0] (TCNTR at SOF)																Word	TimeStamp

Figure 20.3 Mailbox-N Structure (continued)

Notes: 1. All bits shadowed in grey are reserved and must be written LOW. The value returned by a read may not always be '0' and should not be relied upon.

2. ATX and DART are not supported by Mailbox-0, and the MBC setting of Mailbox-0 is limited.

3. ID Reorder (MCR15) can change the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

(1) Message Control Field

STDID[10:0]: These bits set the identifier (standard identifier) of data frames and remote frames.

EXTID[17:0]: These bits set the identifier (extended identifier) of data frames and remote frames.

RTR (Remote Transmission Request bit): Used to distinguish between data frames and remote frames. This bit is overwritten by received CAN Frames depending on Data Frames or Remote Frames.

Important: Please note that, when ATX bit is set with the setting MBC = 001(bin), the RTR bit will never be set. When a Remote Frame is received, the CPU can be notified by the corresponding RFPR set or IRR[2] (Remote Frame Receive Interrupt), however, as RCAN-TL1 needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

Important: In order to support automatic answer to remote frame when MBC = 001 (bin) is used and ATX = 1 the RTR flag must be programmed to zero to allow data frame to be transmitted.

Note: when a Mailbox is configured to send a remote frame request the DLC used for transmission is the one stored into the Mailbox.

RTR	Description
0	Data frame
1	Remote frame

IDE (Identifier Extension bit): Used to distinguish between the standard format and extended format of CAN data frames and remote frames.

IDE	Description
0	Standard format
1	Extended format

- Mailbox-0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	NMC	0	0	MBC[2:0]			0	0	0	0	DLC[3:0]			
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: MBC[1] of MB0 is always "1".

- Mailbox-31 to 1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]			
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

NMC (New Message Control): When this bit is set to ‘0’, the Mailbox of which the RXPR or RFPR bit is already set does not store the new message but maintains the old one and sets the UMSR correspondent bit. When this bit is set to ‘1’, the Mailbox of which the RXPR or RFPR bit is already set overwrites with the new message and sets the UMSR correspondent bit.

Important: Please note that if a remote frame is overwritten with a data frame or vice versa could be that both RXPR and RFPR flags (together with UMSR) are set for the same Mailbox. In this case the RTR bit within the Mailbox Control Field should be relied upon.

Important: Please note that when the Time Triggered mode is used NMC needs to be set to ‘1’ for Mailbox 31 to allow synchronization with all incoming reference messages even when RXPR[31] is not cleared.

NMC	Description
0	Overrun mode (Initial value)
1	Overwrite mode

ATX (Automatic Transmission of Data Frame): When this bit is set to '1' and a Remote Frame is received into the Mailbox DLC is stored. Then, a Data Frame is transmitted from the same Mailbox using the current contents of the message data and updated DLC by setting the corresponding TXPR automatically. The scheduling of transmission is still governed by ID priority or Mailbox priority as configured with the Message Transmission Priority control bit (MCR.2). In order to use this function, MBC[2:0] needs to be programmed to be '001' (Bin). When a transmission is performed by this function, the DLC (Data Length Code) to be used is the one that has been received. Application needs to guarantee that the DLC of the remote frame correspond to the DLC of the data frame requested.

Important: When ATX is used and MBC = 001 (Bin) the filter for the IDE bit cannot be used since ID of remote frame has to be exactly the same as that of data frame as the reply message.

Important: Please note that, when this function is used, the RTR bit will never be set despite receiving a Remote Frame. When a Remote Frame is received, the CPU will be notified by the corresponding RFPR set, however, as RCAN-TL1 needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

Important: Please note that in case of overrun condition (UMSR flag set when the Mailbox has its NMC = 0) the message received is discarded. In case a remote frame is causing overrun into a Mailbox configured with ATX = 1, the transmission of the corresponding data frame may be triggered only if the related PFPR flag is cleared by the CPU when the UMSR flag is set. In such case PFPR flag would get set again.

ATX	Description
0	Automatic Transmission of Data Frame disabled (Initial value)
1	Automatic Transmission of Data Frame enabled

DART (Disable Automatic Re-Transmission): When this bit is set, it disables the automatic re-transmission of a message in the event of an error on the CAN bus or an arbitration lost on the CAN bus. In effect, when this function is used, the corresponding TXCR bit is automatically set at the start of transmission. When this bit is set to '0', RCAN-TL1 tries to transmit the message as many times as required until it is successfully transmitted or it is cancelled by the TXCR.

DART	Description
0	Re-transmission enabled (Initial value)
1	Re-Transmission disabled

MBC[2:0] (Mailbox Configuration): These bits configure the nature of each Mailbox as follows. When MBC = 111 (Bin), the Mailbox is inactive, i.e., it does not receive or transmit a message regardless of TXPR or other settings. The MBC = ’110’, ‘101’ and ‘100’ settings are prohibited. When the MBC is set to any other value, the LAFM field becomes available. Please don't set TXPR when MBC is set as reception as there is no hardware protection, and TXPR will remain set. MBC[1] of Mailbox-0 is fixed to "1" by hardware. This is to ensure that MB0 cannot be configured to transmit Messages.

MBC[2]	MBC[1]	MBC[0]	Data Frame Transmit	Remote Frame Transmit	Data Frame Receive	Remote Frame Receive	Remarks
0	0	0	Yes	Yes	No	No	• Not allowed for Mailbox-0 • Time-Triggered transmission can be used
0	0	1	Yes	Yes	No	Yes	• Can be used with ATX* • Not allowed for Mailbox-0 • LAFM can be used
0	1	0	No	No	Yes	Yes	• Allowed for Mailbox-0 • LAFM can be used
0	1	1	No	No	Yes	No	• Allowed for Mailbox-0 • LAFM can be used
1	0	0	Setting prohibited				
1	0	1	Setting prohibited				
1	1	0	Setting prohibited				
1	1	1	Mailbox inactive (Initial value)				

Notes: * In order to support automatic retransmission, RTR shall be "0" when MBC = 001(bin) and ATX = 1.
When ATX = 1 is used the filter for IDE must not be used.

DLC[3:0] (Data Length Code): These bits encode the number of data bytes from 0,1, 2, … 8 that will be transmitted in a data frame. Please note that when a remote frame request is transmitted the DLC value to be used must be the same as the DLC of the data frame that is requested.

DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
0	0	0	0	Data Length = 0 bytes (Initial value)
0	0	0	1	Data Length = 1 byte
0	0	1	0	Data Length = 2 bytes
0	0	1	1	Data Length = 3 bytes
0	1	0	0	Data Length = 4 bytes
0	1	0	1	Data Length = 5 bytes
0	1	1	0	Data Length = 6 bytes
0	1	1	1	Data Length = 7 bytes
1	x	x	x	Data Length = 8 bytes

(2) Local Acceptance Filter Mask (LAFM)

This area is used as Local Acceptance Filter Mask (LAFM) for receive boxes**.**

LAFM: When MBC is set to 001, 010, 011(Bin), this field is used as LAFM Field. It allows a Mailbox to accept more than one identifier. The LAFM is comprised of two 16-bit read/write areas as follows.

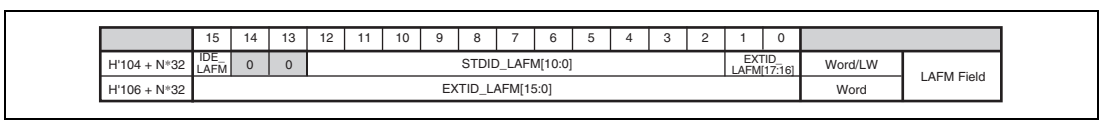


Figure 20.4 Acceptance filter

If a bit is set in the LAFM, then the corresponding bit of a received CAN identifier is ignored when the RCAN-TL1 searches a Mailbox with the matching CAN identifier. If the bit is cleared, then the corresponding bit of a received CAN identifier must match to the STDID/IDE/EXTID set in the mailbox to be stored. The structure of the LAFM is same as the message control in a Mailbox. If this function is not required, it must be filled with ‘0’.

Important: RCAN-TL1 starts to find a matching identifier from Mailbox-31 down to Mailbox-0. As soon as RCAN-TL1 finds one matching, it stops the search. The message will be stored or not depending on the NMC and RXPR/RFPR flags. This means that, even using LAFM, a received message can only be stored into 1 Mailbox.

Important: When a message is received and a matching Mailbox is found, the whole message is stored into the Mailbox. This means that, if the LAFM is used, the STDID, RTR, IDE and EXTID may differ to the ones originally set as they are updated with the STDID, RTR, IDE and EXTID of the received message.

STD_LAFM[10:0] — Filter mask bits for the CAN base identifier [10:0] bits.

STD_LAFM[10:0]	Description
0	Corresponding STD_ID bit is cared
1	Corresponding STD_ID bit is "don't cared"

EXT_LAFM[17:0] — Filter mask bits for the CAN Extended identifier [17:0] bits.

EXT_LAFM[17:0]	Description
0	Corresponding EXT_ID bit is cared
1	Corresponding EXT_ID bit is "don't cared"

IDE_LAFM — Filter mask bit for the CAN IDE bit.

IDE_LAFM	Description
0	Corresponding IDE bit is cared
1	Corresponding IDE bit is "don't cared"

(3) Message Data Fields

Storage for the CAN message data that is transmitted or received. MSG_DATA[0] corresponds to the first data byte that is transmitted or received. The bit order on the CAN bus is bit 7 through to bit 0.

When CMAX!= 3'b111/MBC[30] = 3'b000 and TXPR[30] is set, Mailbox-30 is configured as transmission of time reference. Its DLC must be greater than 0 and its RTR must be zero (as specified for TTCAN Level 1) so that the Cycle_count (CCR register) is embedded in the first byte of the data field instead of MSG_DATA_0[5:0] when this Mailbox starts transmission. This function shall be used when RCAN-TL1 is enabled to work in TTCAN mode to perform a Potential Time Master role to send the Time reference message. MSG_DATA_0[7:6] is still transmitted as stored in the Mailbox. User can set MSG_DATA_0[7] when a Next_is_Gap needs to be transmitted.

Please note that the CCR value is only embedded on the frame transmitted but not stored back into Mailbox 30.

When CMAX!= 3'b111, MBC[31] = 3'b011 and TXPR[31] is cleared, Mailbox-31 is configured as reception of time reference. When a valid reference message is received (DLC > 0) RCAN-TL1 performs internal synchronisation (modifying its RFMK and basic cycle CCR).

MB30 - 31

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'108 + N*32	Next_is_Gap/Cycle_Counter (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	

Figure 20.5 Message Data Field

(4) Timestamp

Storage for the Timestamp recorded on messages for transmit/receive. The Timestamp will be a useful function to monitor if messages are received/transmitted within expected schedule.

- Timestamp

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Message Receive: For received messages of Mailbox-15 to 0, Timestamp always captures the CYCTR (Cycle Time Register) value or Cycle_Counter CCR[5:0] + CYCTR[15:6] value, depending on the programmed value in the bit 14 of TTCR0 (Timer Trigger Control Register 0) at SOF.

For messages received into Mailboxes 30 and 31, Timestamp captures the TCNTR (Timer Counter Register) value at SOF.

Message Transmit: For transmitted messages of Mailbox-15 to 1, Timestamp always captures the CYCTR (Cycle Time Register) value or Cycle_Counter CCR[5:0] + CYCTR[15:6] value, depending on the programmed value in the bit 14 of TTCR0 (Timer Trigger Control Register 0), at SOF.

For messages transmitted from Mailboxes30 and 31, Timestamp captures the TCNTR (Timer Counter Register) value at SOF.

Important: Please note that the TimeStamp is stored in a temporary register. Only after a successful transmission or reception the value is then copied into the related Mailbox field. The TimeStamp may also be updated if the CPU clears RXPR[N]/RFPR[N] at the same time that UMSR[N] is set in overrun, however it can be read properly before clearing RXPR[N]/RFPR[N].

(5) Tx-Trigger Time (TTT) and Time Trigger control

For Mailbox-29 to 24, when MBC is set to 000 (Bin) in time trigger mode (CMAX!= 3'b111), Tx-Trigger Time works as Time_Mark to determine the boundary between time windows. The TTT and TT control are comprised of two 16-bit read/write areas as follows. Mailbox-30 doesn't have TT control and works as Time_Ref.

Mailbox 30 to 24 can be used for reception if not used for transmission in TT mode. However they cannot join the event trigger transmission queue when the TT mode is used.

- Tx-Trigger Time

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8	TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Time Trigger control

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTW[1:0]		Offset[5:0]						0	0	0	0	0	rep_factor[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

The following figure shows the differences between all Mailboxes supporting Time Triggered mode.

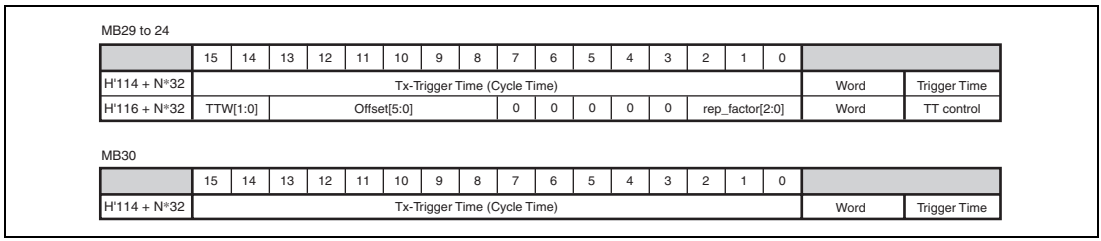


Figure 20.6 Tx-Trigger control field

- **TTW[1:0] (Time Trigger Window):** These bits show the attribute of time windows. Please note that once a merged arbitrating window is opened by TTW = 2'b10, the window must be closed by TTW = 2'b11. Several messages with TTW = 2'b10 may be used within the start and the end of a merged arbitrating window.

TTW[1]	TTW[0]	Description
0	0	Exclusive window (initial value)
0	1	Arbitrating window
1	0	Start of merged arbitrating window
1	1	End of merged arbitrating window

The first 16-bit area specifies the time that triggers the transmission of the message in cycle time. The second 16-bit area specifies the basic cycle in the system matrix where the transmission must start (Offset) and the frequency for periodic transmission. When the internal TTT register matches to the CYCTR value, and the internal Offset matches to CCR value transmission is attempted from the corresponding Mailbox. In order to enable this function, the CMAX (Cycle Maximum Register) must be set to a value different from 3'b111, the Timer (TCNTR) must be running (TTCR0 bit15 = 1), the corresponding MBC must be set to 3'b000 and the corresponding TXPR bit must be set. Once TXPR is set by S/W, RCAN-TL1 does not clear the corresponding TXPR bit (among Mailbox-30 to 24) to carry on performing the periodic transmission. In order to stop the periodic transmission, TXPR must be cleared by TXCR. Please note that in this case it is possible that both TXACK and ABACK are set for the same Mailbox if TXACK is not cleared right after completion of transmission. Please refer to figure 20.7.

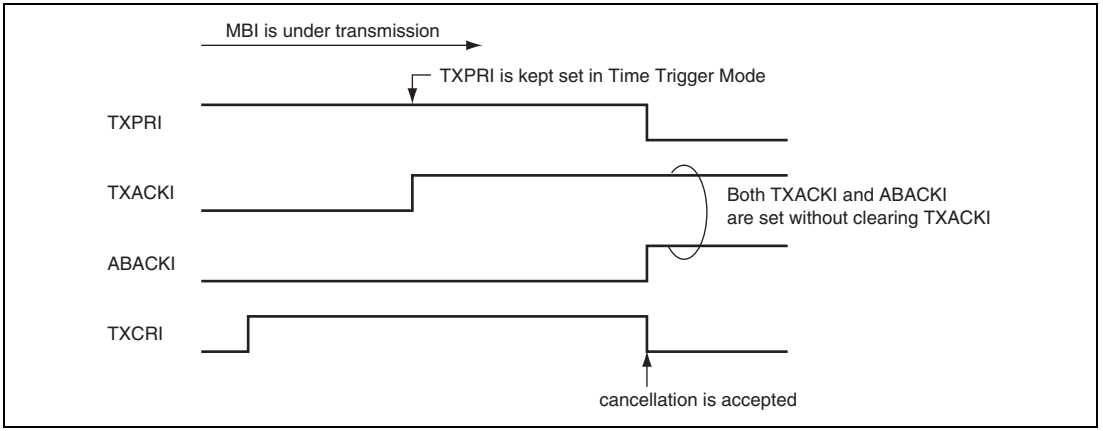


Figure 20.7 TXACK and ABACK in Time Trigger Transmission

Please note that for Mailbox 30 TTW is fixed to ‘01’, Offset to ‘00’ and rep_factor to ‘0’.The following tables report the combinations for the rep_factor and the offset.

Rep_factor	Description
3'b000	Every basic cycle (initial value)
3'b001	Every two basic cycle
3'b010	Every four basic cycle
3'b011	Every eight basic cycle
3'b100	Every sixteen basic cycle
3'b101	Every thirty two basic cycle
3'b110	Every sixty four basic cycle (once in system matrix)
3'b111	Reserved

The Offset Field determines the first cycle in which a Time Triggered Mailbox may start transmitting its Message.

Offset	Description
6'b000000	Initial Offset = 1st Basic Cycle (initial value)
6'b000001	Initial Offset = 2nd Basic Cycles
6'b000010	Initial Offset = 3rd Basic Cycles
6'b000011	Initial Offset = 4th Basic Cycles
6'b000100	Initial Offset = 5th Basic Cycles
…	
…	
6'b111110	Initial Offset = 63rd Basic Cycles
6'b111111	Initial Offset = 64th Basic Cycles

The following relation must be maintained:

Cycle_Count_Maximum + 1 >= Repeat_Factor > Offset

$$\text{Cycle\_Count\_Maximum} = 2^{CMAX} - 1$$

$$\text{Repeat\_Factor} = 2^{rep\_factor}$$

CMAX, Repeat_Factor, and Offset are register values

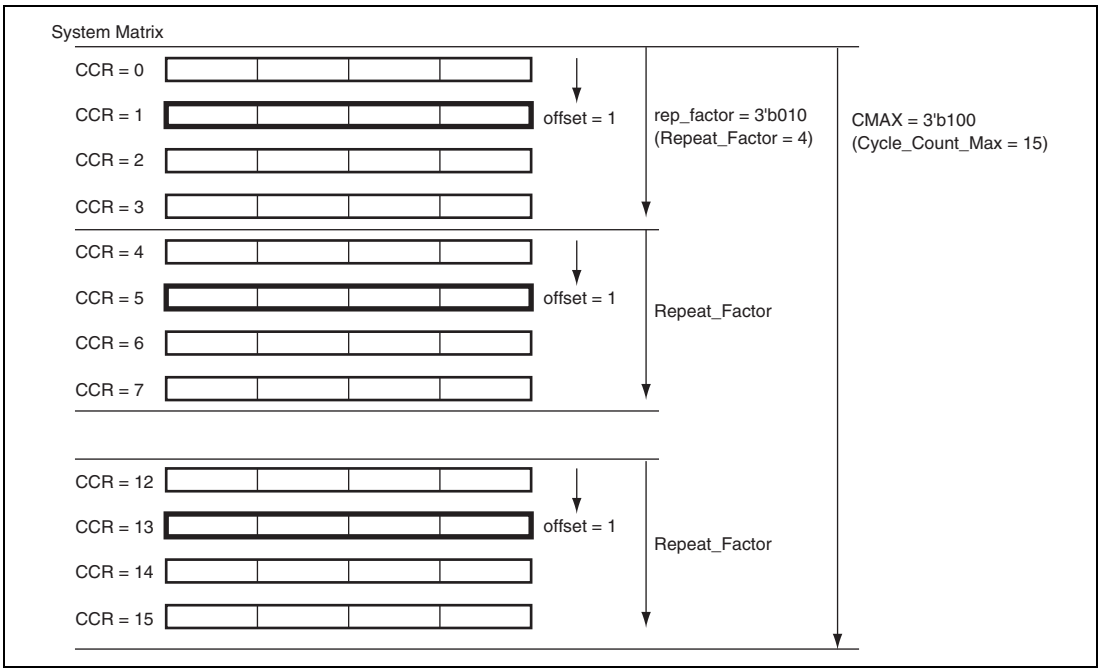


Figure 20.8 System Matrix

Tx-Trigger Times must be set in ascending order such that the difference between them satisfies the following condition.

TTT(mailbox i) –1 TTT(mailbox i-1) > TEW + Maximum frame length + 9

20.3.3 RCAN-TL1 Control Registers

The following sections describe RCAN-TL1 control registers. The address is mapped as follow.

Important: These registers can only be accessed in Word size (16-bit).

Register Name	Address	Abbreviation	Access Size (bits)
Master Control Register	000	MCR	16
General Status Register	002	GSR	16
Bit Configuration Register 1	004	BCR1	16
Bit Configuration Register 0	006	BCR0	16
Interrupt Register	008	IRR	16
Interrupt Mask Register	00A	IMR	16
Error Counter Register	00C	TEC/REC	16

Figure 20.9 RCAN-TL1 control registers

(1) Master Control Register (MCR)

The Master Control Register (MCR) is a 16-bit read/write register that controls RCAN-TL1.

- MCR (Address = H'000)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCR15	MCR14	-	-	-	TST[2:0]			MCR7	MCR6	MCR5	-	-	MCR2	MCR1	MCR0
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit 15 — ID Reorder (MCR15): This bit changes the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

Bit15: MCR15	Description
0	RCAN-TL1 is the same as HCAN2
1	RCAN-TL1 is not the same as HCAN2 (Initial value)

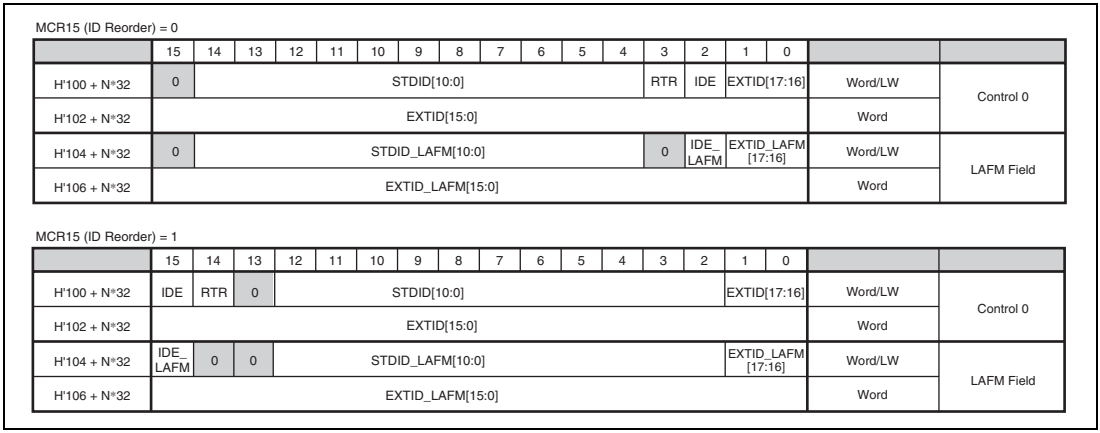


Figure 20.10 ID Reorder

This bit can be modified only in reset mode.

Bit 14 — Auto Halt Bus Off (MCR14): If both this bit and MCR6 are set, MCR1 is automatically set as soon as RCAN-TL1 enters BusOff.

Bit14: MCR14	Description
0	RCAN-TL1 remains in BusOff for normal recovery sequence (128 x 11 Recessive Bits) (Initial value)
1	RCAN-TL1 moves directly into Halt Mode after it enters BusOff if MCR6 is set.

This bit can be modified only in reset mode.

Bit 13 — Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 12 — Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 11 — Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 10 - 8 — Test Mode (TST[2:0]): This bit enables/disables the test modes. Please note that before activating the Test Mode it is requested to move RCAN-TL1 into Halt mode or Reset mode. This is to avoid that the transition to Test Mode could affect a transmission/reception in progress. For details, please refer to section 20.4.1, Test Mode Settings.

Please note that the test modes are allowed only for diagnosis and tests and not when RCAN-TL1 is used in normal operation.

Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	Normal Mode (initial value)
0	0	1	Listen-Only Mode (Receive-Only Mode)
0	1	0	Self Test Mode 1 (External)
0	1	1	Self Test Mode 2 (Internal)
1	0	0	Write Error Counter
1	0	1	Error Passive Mode
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Bit 7 — Auto-wake Mode (MCR7): MCR7 enables or disables the Auto-wake mode. If this bit is set, the RCAN-TL1 automatically cancels the sleep mode (MCR5) by detecting CAN bus activity (dominant bit). If MCR7 is cleared the RCAN-TL1 does not automatically cancel the sleep mode.

RCAN-TL1 cannot store the message that wakes it up.

Note: This bit can be modified only Reset or Halt mode.

Bit7: MCR7	Description
0	Auto-wake by CAN bus activity disabled (Initial value)
1	Auto-wake by CAN bus activity enabled

Bit 6 — Halt during Bus Off (MCR6): MCR6 enables or disables entering Halt mode immediately when MCR1 is set during Bus Off. This bit can be modified only in Reset or Halt mode. Please note that when Halt is entered in Bus Off the CAN engine is also recovering immediately to Error Active mode.

Bit6: MCR6	Description
0	If MCR[1] is set, RCAN-TL1 will not enter Halt mode during Bus Off but wait up to end of recovery sequence (Initial value)
1	Enter Halt mode immediately during Bus Off if MCR[1] or MCR[14] are asserted.

Bit 5 — Sleep Mode (MCR5): Enables or disables Sleep mode transition. If this bit is set, while RCAN-TL1 is in halt mode, the transition to sleep mode is enabled. Setting MCR5 is allowed after entering Halt mode. The two Error Counters (REC, TEC) will remain the same during Sleep mode. This mode will be exited in two ways:

1. by writing a '0' to this bit position,
2. or, if MCR[7] is enabled, after detecting a dominant bit on the CAN bus.

If Auto wake up mode is disabled, RCAN-TL1 will ignore all CAN bus activities until the sleep mode is terminated. When leaving this mode the RCAN-TL1 will synchronise to the CAN bus (by checking for 11 recessive bits) before joining CAN Bus activity. This means that, when the No.2 method is used, RCAN-TL1 will miss the first message to receive. CAN transceivers stand-by mode will also be unable to cope with the first message when exiting stand by mode, and the S/W needs to be designed in this manner.

In sleep mode only the following registers can be accessed: MCR, GSR, IRR and IMR.

Important: RCAN-TL1 is required to be in Halt mode before requesting to enter in Sleep mode. That allows the CPU to clear all pending interrupts before entering sleep mode. Once all interrupts are cleared RCAN-TL1 must leave the Halt mode and enter Sleep mode simultaneously (by writing MCR[5] = 1 and MCR[1] = 0 at the same time).

Bit 5: MCR5	Description
0	RCAN-TL1 sleep mode released (Initial value)
1	Transition to RCAN-TL1 sleep mode enabled

Bit 4 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 3 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 2 — Message Transmission Priority (MCR2): MCR2 selects the order of transmission for pending transmit data. If this bit is set, pending transmit data are sent in order of the bit position in the Transmission Pending Register (TXPR). The order of transmission starts from Mailbox-31 as the highest priority, and then down to Mailbox-1 (if those mailboxes are configured for transmission). Please note that this feature cannot be used for time trigger transmission of the Mailboxes 24 to 30.

If MCR2 is cleared, all messages for transmission are queued with respect to their priority (by running internal arbitration). The highest priority message has the Arbitration Field (STDID + IDE bit + EXTID (if IDE = 1) + RTR bit) with the lowest digital value and is transmitted first. The internal arbitration includes the RTR bit and the IDE bit (internal arbitration works in the same

way as the arbitration on the CAN Bus between two CAN nodes starting transmission at the same time).

This bit can be modified only in Reset or Halt mode.

Bit 2: MCR2	Description
0	Transmission order determined by message identifier priority (Initial value)
1	Transmission order determined by mailbox number priority (Mailbox-31 → Mailbox-1)

Bit 1—Halt Request (MCR1): Setting the MCR1 bit causes the CAN controller to complete its current operation and then enter Halt mode (where it is cut off from the CAN bus). The RCAN-TL1 remains in Halt Mode until the MCR1 is cleared. During the Halt mode, the CAN Interface does not join the CAN bus activity and does not store messages or transmit messages. All the user registers (including Mailbox contents and TEC/REC) remain unchanged with the exception of IRR0 and GSR4 which are used to notify the halt status itself. If the CAN bus is in idle or intermission state regardless of MCR6, RCAN-TL1 will enter Halt Mode within one Bit Time. If MCR6 is set, a halt request during Bus Off will be also processed within one Bit Time. Otherwise the full Bus Off recovery sequence will be performed beforehand. Entering the Halt Mode can be notified by IRR0 and GSR4.

If both MCR14 and MCR6 are set, MCR1 is automatically set as soon as RCAN-TL1 enters BusOff.

In the Halt mode, the RCAN-TL1 configuration can be modified with the exception of the Bit Timing setting, as it does not join the bus activity. MCR[1] has to be cleared by writing a ‘0’ in order to re-join the CAN bus. After this bit has been cleared, RCAN-TL1 waits until it detects 11 recessive bits, and then joins the CAN bus.

Notes: 1. After issuing a Halt request the CPU is not allowed to set TXPR or TXCR or clear MCR1 until the transition to Halt mode is completed (notified by IRR0 and GSR4). After MCR1 is set this can be cleared only after entering Halt mode or through a reset operation (SW or HW).

2. Transition into or recovery from HALT mode, is only possible if the BCR1 and BCR0 registers are configured to a proper Baud Rate.

Bit 1: MCR1	Description
0	Clear Halt request (Initial value)
1	Halt mode transition request

Bit 0 — Reset Request (MCR0): Controls resetting of the RCAN-TL1 module. When this bit is changed from ‘0’ to ‘1’ the RCAN-TL1 controller enters its reset routine, re-initialising the internal logic, which then sets GSR3 and IRR0 to notify the reset mode. During a re-initialisation, all user registers are initialised.

RCAN-TL1 can be re-configured while this bit is set. This bit has to be cleared by writing a ‘0’ to join the CAN bus. After this bit is cleared, the RCAN-TL1 module waits until it detects 11 recessive bits, and then joins the CAN bus. The Baud Rate needs to be set up to a proper value in order to sample the value on the CAN Bus.

After Power On Reset, this bit and GSR3 are always set. This means that a reset request has been made and RCAN-TL1 needs to be configured.

The Reset Request is equivalent to a Power On Reset but controlled by Software.

Bit 0: MCR0	Description
0	Clear Reset Request
1	CAN Interface reset mode transition request (Initial value)

(2) General Status Register (GSR)

The General Status Register (GSR) is a 16-bit read-only register that indicates the status of RCAN-TL1.

- GSR (Address = H'002)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bits 15 to 6: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 5 — Error Passive Status Bit (GSR5): Indicates whether the CAN Interface is in Error Passive or not. This bit will be set high as soon as the RCAN-TL1 enters the Error Passive state and is cleared when the module enters again the Error Active state (this means the GSR5 will stay high during Error Passive and during Bus Off). Consequently to find out the correct state both GSR5 and GSR0 must be considered.

Bit 5: GSR5	Description
0	RCAN-TL1 is not in Error Passive or in Bus Off status (Initial value) [Reset condition] RCAN-TL1 is in Error Active state
1	RCAN-TL1 is in Error Passive (if GSR0 = 0) or Bus Off (if GSR0 = 1) [Setting condition] When TEC ≥ 128 or REC ≥ 128 or if Error Passive Test Mode is selected

Bit 4 — Halt/Sleep Status Bit (GSR4): Indicates whether the CAN engine is in the halt/sleep state or not. Please note that the clearing time of this flag is not the same as the setting time of IRR12.

Please note that this flag reflects the status of the CAN engine and not of the full RCAN-TL1 IP. RCAN-TL1 exits sleep mode and can be accessed once MCR5 is cleared. The CAN engine exits sleep mode only after two additional transmission clocks on the CAN Bus.

Bit 4: GSR4	Description
0	RCAN-TL1 is not in the Halt state or Sleep state (Initial value)
1	Halt mode (if MCR1 = 1) or Sleep mode (if MCR5 = 1) [Setting condition] If MCR1 is set and the CAN bus is either in intermission or idle or MCR5 is set and RCAN-TL1 is in the halt mode or RCAN-TL1 is moving to Bus Off when MCR14 and MCR6 are both set

Bit 3 — Reset Status Bit (GSR3): Indicates whether the RCAN-TL1 is in the reset state or not.

Bit 3: GSR3	Description
0	RCAN-TL1 is not in the reset state
1	Reset state (Initial value) [Setting condition] After an RCAN-TL1 internal reset (due to SW or HW reset)

Bit 2 — Message Transmission in progress Flag (GSR2): Flag that indicates to the CPU if the RCAN-TL1 is in Bus Off or transmitting a message or an error/overload flag due to error detected during transmission. The timing to set TXACK is different from the time to clear GSR2. TXACK is set at the 7th bit of End Of Frame. GSR2 is set at the 3rd bit of intermission if there are no more messages ready to be transmitted. It is also set by arbitration lost, bus idle, reception, reset or halt transition.

Bit 2: GSR2	Description
0	RCAN-TL1 is in Bus Off or a transmission is in progress
1	[Setting condition] Not in Bus Off and no transmission in progress (Initial value)

Bit 1—Transmit/Receive Warning Flag (GSR1): Flag that indicates an error warning.

Bit 1: GSR1	Description
0	[Reset condition] When (TEC < 96 and REC < 96) or Bus Off (Initial value)
1	[Setting condition] When 96 ≤ TEC < 256 or 96 ≤ REC < 256

Note: REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence. However the flag GSR1 is not set in Bus Off.

Bit 0—Bus Off Flag (GSR0): Flag that indicates that RCAN-TL1 is in the bus off state.

Bit 0: GSR0	Description
0	[Reset condition] Recovery from bus off state or after a HW or SW reset (Initial value)
1	[Setting condition] When TEC ≥ 256 (bus off state)

Note: Only the lower 8 bits of TEC are accessible from the user interface. The 9th bit is equivalent to GSR0.

(3) Bit Configuration Register (BCR0, BCR1)

The bit configuration registers (BCR0 and BCR1) are 2 X 16-bit read/write register that are used to set CAN bit timing parameters and the baud rate pre-scaler for the CAN Interface.

The Time quanta is defined as:

$$Timequanta = \frac{2 * BRP}{f_{clk}}$$

Where: BRP (Baud Rate Pre-scaler) is the value stored in BCR0 incremented by 1 and fclk is the used peripheral clock frequency.

- BCR1 (Address = H'004)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG1[3:0]				-	TSG2[2:0]			-	-	SJW[1:0]		-	-	-	BSP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W

Bits 15 to 12 — Time Segment 1 (TSG1[3:0] = BCR1[15:12]): These bits are used to set the segment TSEG1 (= PRSEG + PHSEG1) to compensate for edges on the CAN Bus with a positive phase error. A value from 4 to 16 time quanta can be set.

Bit 15: TSG1[3]	Bit 14: TSG1[2]	Bit 13: TSG1[1]	Bit 12: TSG1[0]	Description
0	0	0	0	Setting prohibited (Initial value)
0	0	0	1	Setting prohibited
0	0	1	0	Setting prohibited
0	0	1	1	PRSEG + PHSEG1 = 4 time quanta
0	1	0	0	PRSEG + PHSEG1 = 5 time quanta
:	:	:	:	:
:	:	:	:	:
1	1	1	1	PRSEG + PHSEG1 = 16 time quanta

Bit 11: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bits 10 to 8 — Time Segment 2 (TSG2[2:0] = BCR1[10:8]): These bits are used to set the segment TSEG2 (= PHSEG2) to compensate for edges on the CAN Bus with a negative phase error. A value from 2 to 8 time quanta can be set as shown below.

Bit 10: TSG2[2]	Bit 9: TSG2[1]	Bit 8: TSG2[0]	Description
0	0	0	Setting prohibited (Initial value)
0	0	1	PHSEG2 = 2 time quanta (conditionally prohibited)
0	1	0	PHSEG2 = 3 time quanta
0	1	1	PHSEG2 = 4 time quanta
1	0	0	PHSEG2 = 5 time quanta
1	0	1	PHSEG2 = 6 time quanta
1	1	0	PHSEG2 = 7 time quanta
1	1	1	PHSEG2 = 8 time quanta

Bits 7 and 6: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bits 5 and 4 - ReSynchronisation Jump Width (SJW[1:0] = BCR0[5:4]): These bits set the synchronisation jump width.

Bit 5: SJW[1]	Bit 4: SJW[0]	Description
0	0	Synchronisation Jump width = 1 time quantum (Initial value)
0	1	Synchronisation Jump width = 2 time quanta
1	0	Synchronisation Jump width = 3 time quanta
1	1	Synchronisation Jump width = 4 time quanta

Bits 3 to 1: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 0 — Bit Sample Point (BSP = BCR1[0]): Sets the point at which data is sampled.

Bit 0 : BSP	Description
0	Bit sampling at one point (end of time segment 1) (Initial value)
1	Bit sampling at three points (rising edge of the last three clock cycles of PHSEG1)

- BCR0 (Address = H'006)

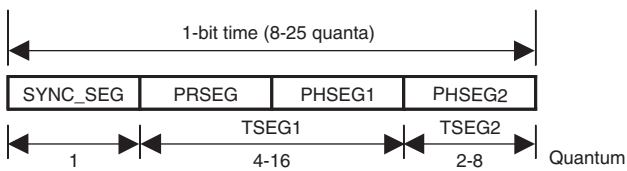
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	BRP[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 8 to 15: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bits 7 to 0—Baud Rate Pre-scale (BRP[7:0] = BCR0 [7:0]): These bits are used to define the peripheral clock periods contained in a Time Quantum.

Bit 7: BRP[7]	Bit 6: BRP[6]	Bit 5: BRP[5]	Bit 4: BRP[4]	Bit 3: BRP[3]	Bit 2: BRP[2]	Bit 1: BRP[1]	Bit 0: BRP[0]	Description
0	0	0	0	0	0	0	0	2 × peripheral clock (Initial value)
0	0	0	0	0	0	0	1	4 × peripheral clock
0	0	0	0	0	0	1	0	6 × peripheral clock
: :	: :	: :	: :	: :	: :	: :	: :	2 × (register value + 1) × peripheral clock
1	1	1	1	1	1	1	1	512 × peripheral clock

- Requirements of Bit Configuration Register



SYNC_SEG: Segment for establishing synchronisation of nodes on the CAN bus. (Normal bit edge transitions occur in this segment.)

PRSEG: Segment for compensating for physical delay between networks.

PHSEG1: Buffer segment for correcting phase drift (positive). (This segment is extended when synchronisation (resynchronisation) is established.)

PHSEG2: Buffer segment for correcting phase drift (negative). (This segment is shortened when synchronisation (resynchronisation) is established)

TSEG1: TSG1 + 1

TSEG2: TSG2 + 1

The RCAN-TL1 Bit Rate Calculation is:

$$\text{Bit Rate} = \frac{f_{clk}}{2 \times (BRP + 1) \times (TSEG1 + TSEG2 + 1)}$$

Where BRP is given by the register value and TSEG1 and TSEG2 are derived values from TSG1 and TSG2 register values. The '+1' in the above formula is for the Sync-Seg which duration is 1 time quanta.

f_{CLK} = Peripheral Clock

BCR Setting Constraints

TSEG1min > TSEG2 ≥ SJWmax (SJW = 1 to 4)

8 ≤ TSEG1 + TSEG2 + 1 ≤ 25 time quanta (TSEG1 + TSEG2 + 1 = 7 is not allowed)

TSEG2 ≥ 2

These constraints allow the setting range shown in the table below for TSEG1 and TSEG2 in the Bit Configuration Register. The number in the table shows possible setting of SJW. "No" shows that there is no allowed combination of TSEG1 and TSEG2.

		001	010	011	100	101	110	111	TSG2
		2	3	4	5	6	7	8	TSEG2
TSG1	TSEG1								
0011	4	No	1-3	No	No	No	No	No	
0100	5	1-2	1-3	1-4	No	No	No	No	
0101	6	1-2	1-3	1-4	1-4	No	No	No	
0110	7	1-2	1-3	1-4	1-4	1-4	No	No	
0111	8	1-2	1-3	1-4	1-4	1-4	1-4	No	
1000	9	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1001	10	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1010	11	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1011	12	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1100	13	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1101	14	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1110	15	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1111	16	1-2	1-3	1-4	1-4	1-4	1-4	1-4	

Example 1: To have a Bit rate of 500Kbps with a frequency of fclk = 32MHz it is possible to set: BRP = 1, TSEG1 = 11, TSEG2 = 4.

Then the configuration to write is BCR1 = H'A300 and BCR0 = H'0001.

Example 2: To have a Bit rate of 500Kbps with a frequency of fclk = 20MHz it is possible to set: BPR = 1, TSEG1 = 6, TSEG2 = 3.

Then the configuration to write is BCR1 = H'5200 and BCR0 = 0001.

(4) Interrupt Request Register (IRR)

The interrupt register (IRR) is a 16-bit read/write-clearable register containing status flags for the various interrupt sources.

- IRR (Address = H'008)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit 15 — Timer Compare Match Interrupt 1 (IRR15): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 1 (TCMR1). When the value set in the TCMR1 matches to Cycle Time (TCMR1 = CYCTR), this bit is set.

Bit 15: IRR15	Description
0	Timer Compare Match has not occurred to the TCMR1 (Initial value) [Clearing condition] Writing 1
1	Timer Compare Match has occurred to the TCMR1 [Setting condition] TCMR1 matches to Cycle Time (TCMR1 = CYCTR)

Bit 14 — Timer Compare Match Interrupt 0 (IRR14): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 0 (TCMR0). When the value set in the TCMR0 matches to Local Time (TCMR0 = TCNTR), this bit is set.

Bit 14: IRR14	Description
0	Timer Compare Match has not occurred to the TCMR0 (Initial value) [Clearing condition] Writing 1
1	Timer Compare Match has occurred to the TCMR0 [Setting condition] TCMR0 matches to the Timer value (TCMR0 = TCNTR)

Bit 13 - Timer Overrun Interrupt/Next_is_Gap Reception Interrupt/Message Error Interrupt (IRR13): This interrupt assumes a different meaning depending on the RCAN-TL1 mode. It indicates that:

- The Timer (TCNTR) has overrun when RCAN-TL1 is working in event-trigger mode (including test modes)

- Time reference message with Next_is_Gap set has been received when working in time-trigger mode. Please note that when a Next_is_Gap is received the application is responsible to stop all transmission at the end of the current basic cycle (including test modes)
- Message error has occurred when in test mode. Note: If a Message Overload condition occurs when in Test Mode, then this bit will not be set.

Bit 13: IRR13	Description
0	Timer (TCNTR) has not overrun in event-trigger mode (including test modes) (Initial value) Time reference message with Next_is_Gap has not been received in time-trigger mode (including test modes) Message error has not occurred in test mode [Clearing condition] Writing 1
1	[Setting condition] Timer (TCNTR) has overrun and changed from H'FFFF to H'0000 in event-trigger mode (including test modes) Time reference message with Next_is_Gap has been received in time-trigger mode (including test modes) Message error has occurred in test mode

Bit 12 – Bus activity while in sleep mode (IRR12): IRR12 indicates that a CAN bus activity is present. While the RCAN-TL1 is in sleep mode and a dominant bit is detected on the CAN bus, this bit is set. This interrupt is cleared by writing a '1' to this bit position. Writing a '0' has no effect. If auto wakeup is not used and this interrupt is not requested it needs to be disabled by the related interrupt mask register. If auto wake up is not used and this interrupt is requested it should be cleared only after recovering from sleep mode. This is to avoid that a new falling edge of the reception line causes the interrupt to get set again.

Please note that the setting time of this interrupt is different from the clearing time of GSR4.

Bit 12: IRR12	Description
0	Bus idle state (Initial value) [Clearing condition] Writing 1
1	CAN bus activity detected in RCAN-TL1 sleep mode [Setting condition] Dominant bit level detection on the Rx line while in sleep mode

Bit 11 — Timer Compare Match Interrupt 2 (IRR11): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 2 (TCMR2). When the value set in the TCMR2 matches to Cycle Time (TCMR2 = CYCTR), this bit is set.

Bit 11: IRR11	Description
0	Timer Compare Match has not occurred to the TCMR2 (initial value) [Clearing condition] Writing 1
1	Timer Compare Match has occurred to the TCMR2 [Setting condition] TCMR2 matches to Cycle Time (TCMR2 = CYCTR)

Bit 10 — Start of new system matrix Interrupt (IRR10): Indicates that a new system matrix is starting.

When CCR = 0, this bit is set at the successful completion of reception/transmission of time reference message. Please note that when CMAX = 0 this interrupt is set at every basic cycle.

Bit 10: IRR10	Description
0	A new system matrix is not starting (initial value) [Clearing condition] Writing 1
1	Cycle counter reached zero. [Setting condition] Reception/transmission of time reference message is successfully completed when CMAX!= 3'b111 and CCR = 0

Bit 9 – Message Overrun/Overwrite Interrupt Flag (IRR9): Flag indicating that a message has been received but the existing message in the matching Mailbox has not been read as the corresponding RXPR or RFPR is already set to ‘1’ and not yet cleared by the CPU. The received message is either abandoned (overrun) or overwritten dependant upon the NMC (New Message Control) bit. This bit is cleared when all bit in UMSR (Unread Message Status Register) are cleared (by writing ‘1’) or by setting MBIMR (MailBox interrupt Mast Register) for all UMSR flag set. It is also cleared by writing a ‘1’ to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

Bit 9: IRR9	Description
0	No pending notification of message overrun/overwrite [Clearing condition] Clearing of all bit in UMSR/setting MBIMR for all UMSR set (initial value)
1	A receive message has been discarded due to overrun condition or a message has been overwritten [Setting condition] Message is received while the corresponding RXPR and/or RFPR = 1 and MBIMR = 0

Bit 8 - Mailbox Empty Interrupt Flag (IRR8): This bit is set when one of the messages set for transmission has been successfully sent (corresponding TXACK flag is set) or has been successfully aborted (corresponding ABACK flag is set). In Event Triggered mode the related TXPR is also cleared and this mailbox is now ready to accept a new message data for the next transmission. In Time Trigger mode TXPR for the Mailboxes from 30 to 24 is not cleared after a successful transmission in order to keep transmitting at each programmed basic cycle. In effect, this bit is set by an OR'ed signal of the TXACK and ABACK bits not masked by the corresponding MBIMR flag. Therefore, this bit is automatically cleared when all the TXACK and ABACK bits are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

Bit 8: IRR8	Description
0	Messages set for transmission or transmission cancellation request NOT progressed. (Initial value) [Clearing Condition] All the TXACK and ABACK bits are cleared/setting MBIMR for all TXACK and ABACK set
1	Message has been transmitted or aborted, and new message can be stored (in TT mode Mailbox 24 to 30 can be programmed with a new message only in case of abortion) [Setting condition] When a TXACK or ABACK bit is set (if related MBIMR = 0).

Bit 7 - Overload Frame (IRR7): Flag indicating that the RCAN-TL1 has detected a condition that should initiate the transmission of an overload frame. Note that in the condition of transmission being prevented, such as listen only mode, an Overload Frame will NOT be transmitted, but IRR7 will still be set. IRR7 remains asserted until reset by writing a '1' to this bit position - writing a '0' has no effect.

Bit 7: IRR7	Description
0	[Clearing condition] Writing 1 (Initial value)
1	[Setting conditions] Overload condition detected

Bit 6 - Bus Off Interrupt Flag (IRR6): This bit is set when RCAN-TL1 enters the Bus-off state or when RCAN-TL1 leaves Bus-off and returns to Error-Active. The cause therefore is the existing condition TEC ≥ 256 at the node or the end of the Bus-off recovery sequence (128X11 consecutive recessive bits) or the transition from Bus Off to Halt (automatic or manual). This bit remains set even if the RCAN-TL1 node leaves the bus-off condition, and needs to be explicitly cleared by S/W. The S/W is expected to read the GSR0 to judge whether RCAN-TL1 is in the bus-off or error active status. It is cleared by writing a '1' to this bit position even if the node is still bus-off. Writing a '0' has no effect.

Bit 6: IRR6	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Enter Bus off state caused by transmit error or Error Active state returning from Bus-off [Setting condition] When TEC becomes ≥ 256 or End of Bus-off after 128X11 consecutive recessive bits or transition from Bus Off to Halt

Bit 5 - Error Passive Interrupt Flag (IRR5): Interrupt flag indicating the error passive state caused by the transmit or receive error counter or by Error Passive forced by test mode. This bit is reset by writing a '1' to this bit position, writing a '0' has no effect. If this bit is cleared the node may still be error passive. Please note that the SW needs to check GSR0 and GSR5 to judge whether RCAN-TL1 is in Error Passive or Bus Off status.

Bit 5: IRR5	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error passive state caused by transmit/receive error [Setting condition] When TEC ≥ 128 or REC ≥ 128 or Error Passive test mode is used

Bit 4 - Receive Error Counter Warning Interrupt Flag (IRR4): This bit becomes set if the receive error counter (REC) reaches a value greater than 95 when RCAN-TL1 is not in the Bus Off status. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

Bit 4: IRR4	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by receive error [Setting condition] When REC ≥ 96 and RCAN-TL1 is not in Bus Off

Bit 3 - Transmit Error Counter Warning Interrupt Flag (IRR3): This bit becomes set if the transmit error counter (TEC) reaches a value greater than 95. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

Bit 3: IRR3	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by transmit error [Setting condition] When TEC ≥ 96

Bit 2 - Remote Frame Receive Interrupt Flag (IRR2): Flag indicating that a remote frame has been received in a mailbox. This bit is set if at least one receive mailbox, with related MBIMR not set, contains a remote frame transmission request. This bit is automatically cleared when all bits in the Remote Frame Receive Pending Register (RFPR), are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

Bit 2: IRR2	Description
0	[Clearing condition] Clearing of all bits in RFPR (Initial value)
1	At least one remote request is pending [Setting condition] When remote frame is received and the corresponding MBIMR = 0

Bit 1 – Data Frame Received Interrupt Flag (IRR1): IRR1 indicates that there are pending Data Frames received. If this bit is set at least one receive mailbox contains a pending message. This bit is cleared when all bits in the Data Frame Receive Pending Register (RXPR) are cleared, i.e. there is no pending message in any receiving mailbox. It is in effect a logical OR of the RXPR flags from each configured receive mailbox with related MBIMR not set. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

Bit 1: IRR1	Description
0	[Clearing condition] Clearing of all bits in RXPR (Initial value)
1	Data frame received and stored in Mailbox [Setting condition] When data is received and the corresponding MBIMR = 0

Bit 0 – Reset/Halt/Sleep Interrupt Flag (IRR0): This flag can get set for three different reasons. It can indicate that:

1. Reset mode has been entered after a SW (MCR0) or HW reset
2. Halt mode has been entered after a Halt request (MCR1)
3. Sleep mode has been entered after a sleep request (MCR5) has been made while in Halt mode.

The GSR may be read after this bit is set to determine which state RCAN-TL1 is in.

Important: When a Sleep mode request needs to be made, the Halt mode must be used beforehand. Please refer to the MCR5 description and Figure 20.15 Halt Mode/Sleep Mode.

IRR0 is set by the transition from "0" to "1" of GSR3 or GSR4 or by transition from Halt mode to Sleep mode. So, IRR0 is not set if RCAN-TL1 enters Halt mode again right after exiting from Halt mode, without GSR4 being cleared. Similarly, IRR0 is not set by direct transition from Sleep mode to Halt Request. At the transition from Halt/Sleep mode to Transition/Reception, clearing GSR4 needs (one-bit time - TSEG2) to (one-bit time × 2 - TSEG2).

In the case of Reset mode, IRR0 is set, however, the interrupt to the CPU is not asserted since IMR0 is automatically set by initialisation.

Bit 0: IRR0	Description
0	[Clearing condition] Writing 1
1	Transition to S/W reset mode or transition to halt mode or transition to sleep mode (Initial value) [Setting condition] When reset/halt/sleep transition is completed after a reset (MCR0 or HW) or Halt mode (MCR1) or Sleep mode (MCR5) is requested

(5) Interrupt Mask Register (IMR)

The interrupt mask register is a 16 bit register that protects all corresponding interrupts in the Interrupt Request Register (IRR) from generating an output signal on the IRQ. An interrupt request is masked if the corresponding bit position is set to '1'. This register can be read or written at any time. The IMR directly controls the generation of IRQ, but does not prevent the setting of the corresponding bit in the IRR.

- IMR (Address = H'00A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0: Maskable interrupt sources corresponding to IRR[15:0] respectively. When a bit is set, the interrupt signal is not generated, although setting the corresponding IRR bit is still performed.

Bit[15:0]: IMRn	Description
0	Corresponding IRR is not masked (IRQ is generated for interrupt conditions)
1	Corresponding interrupt of IRR is masked (Initial value)

(6) Transmit Error Counter (TEC) and Receive Error Counter (REC)

The Transmit Error Counter (TEC) and Receive Error Counter (REC) is a 16-bit read/(write) register that functions as a counter indicating the number of transmit/receive message errors on the CAN Interface. The count value is stipulated in the CAN protocol specification Refs. [1], [2], [3] and [4]. When not in (Write Error Counter) test mode this register is read only, and can only be modified by the CAN Interface. This register can be cleared by a Reset request (MCR0) or entering to bus off.

In Write Error Counter test mode (i.e. TST[2:0] = 3'b100), it is possible to write to this register. The same value can only be written to TEC/REC, and the value written into TEC is set to TEC and REC. When writing to this register, RCAN-TL1 needs to be put into Halt Mode. This feature is only intended for test purposes.

- TEC/REC (Address = H'00C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * It is only possible to write the value in test mode when TST[2:0] in MCR is 3'b100.
REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence.

20.3.4 RCAN-TL1 Mailbox Registers

The following sections describe RCAN-TL1 Mailbox registers that control/flag individual Mailboxes. The address is mapped as follows.

Important: LongWord access is carried out as two consecutive Word accesses.

32-Mailboxes version

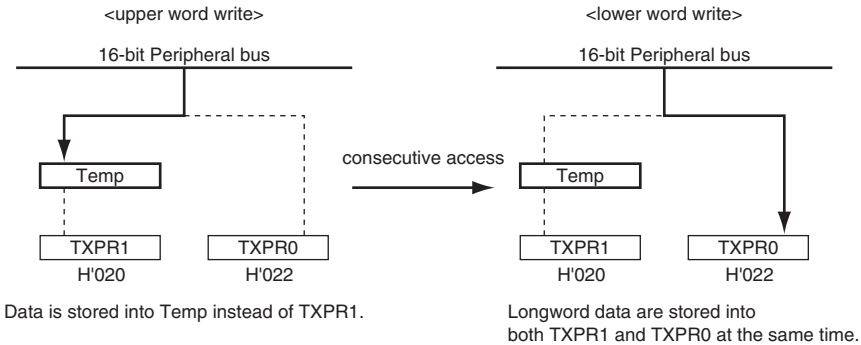
Description	Address	Name	Access Size (bits)
Transmit Pending 1	020	TXPR1	LW
Transmit Pending 0	022	TXPR0	—
	024		
	026		
Transmit Cancel 1	028	TXCR1	Word/LW
Transmit Cancel 0	02A	TXCR0	Word
	02C		
	02E		
Transmit Acknowledge 1	030	TXACK1	Word/LW
Transmit Acknowledge 0	032	TXACK0	Word
	034		
	036		
Abort Acknowledge 1	038	ABACK1	Word/LW
Abort Acknowledge 0	03A	ABACK0	Word
	03C		
	03E		
Data Frame Receive Pending 1	040	RXPR1	Word/LW
Data Frame Receive Pending 0	042	RXPR0	Word
	044		
	046		
Remote Frame Receive Pending 1	048	RFPR1	Word/LW
Remote Frame Receive Pending 0	04A	RFPR0	Word
	04C		
	04E		
Mailbox Interrupt Mask Register 1	050	MBIMR1	Word/LW
Mailbox Interrupt Mask Register 0	052	MBIMR0	Word
	054		
	056		
Unread message Status Register 1	058	UMSR1	Word/LW
Unread message Status Register 0	05A	UMSR0	Word
	05C		
	05E		

Figure 20.11 RCAN-TL1 Mailbox registers

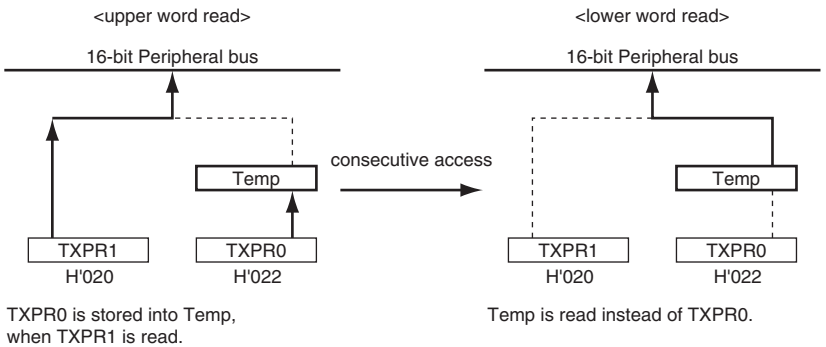
(1) Transmit Pending Register (TXPR1, TXPR0)

The concatenation of TXPR1 and TXPR0 is a 32-bit register that contains any transmit pending flags for the CAN module. In the case of 16-bit bus interface, Long Word access is carried out as two consecutive word accesses.

<Longword Write Operation>



<Longword Read Operation>



The TXPR1 controls Mailbox-31 to Mailbox-16, and the TXPR0 controls Mailbox-15 to Mailbox-1. The CPU may set the TXPR bits to affect any message being considered for transmission by writing a ‘1’ to the corresponding bit location. Writing a ‘0’ has no effect, and TXPR cannot be cleared by writing a ‘0’ and must be cleared by setting the corresponding TXCR bits. TXPR may be read by the CPU to determine which, if any, transmissions are pending or in progress. In effect there is a transmit pending bit for all Mailboxes except for the Mailbox-0. Writing a ‘1’ to a bit location when the mailbox is not configured to transmit is not allowed.

In Event Triggered Mode RCAN-TL1 will clear a transmit pending flag after successful transmission of its corresponding message or when a transmission abort is requested successfully from the TXCR. In Time Trigger Mode, TXPR for the Mailboxes from 30 to 24 is NOT cleared after a successful transmission, in order to keep transmitting at each programmed basic cycle. The TXPR flag is not cleared if the message is not transmitted due to the CAN node losing the arbitration process or due to errors on the CAN bus, and RCAN-TL1 automatically tries to transmit it again unless its DART bit (Disable Automatic Re-Transmission) is set in the Message-Control of the corresponding Mailbox. In such case (DART set), the transmission is cleared and notified through Mailbox Empty Interrupt Flag (IRR8) and the correspondent bit within the Abort Acknowledgement Register (ABACK).

If the status of the TXPR changes, the RCAN-TL1 shall ensure that in the identifier priority scheme (MCR2 = 0), the highest priority message is always presented for transmission in an intelligent way even under circumstances such as bus arbitration losses or errors on the CAN bus. Please refer to the Application Note for details.

When the RCAN-TL1 changes the state of any TXPR bit position to a '0', an empty slot interrupt (IRR8) may be generated. This indicates that either a successful or an aborted mailbox transmission has just been made. If a message transmission is successful it is signalled in the TXACK register, and if a message transmission abortion is successful it is signalled in the ABACK register. By checking these registers, the contents of the Message of the corresponding Mailbox may be modified to prepare for the next transmission.

- TXPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * It is possible only to write a ‘1’ for a Mailbox configured as transmitter.

Bit 15 to 0 — Requests the corresponding Mailbox to transmit a CAN Frame. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number.

Bit[15:0]: TXPR1	Description
0	Transmit message idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of message transmission (for Event Triggered Messages) or message transmission abortion (automatically cleared)
1	Transmission request made for corresponding mailbox

- TXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR0[15:1]															-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R

Note: * It is possible only to write a ‘1’ for a Mailbox configured as transmitter.

Bit 15 to 1 — Indicates that the corresponding Mailbox is requested to transmit a CAN Frame. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number.

Bit[15:1]: TXPR0	Description
0	Transmit message idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of message transmission (for Event Triggered Messages) or message transmission abortion (automatically cleared)
1	Transmission request made for corresponding mailbox

Bit 0— Reserved: This bit is always ‘0’ as this is a receive-only Mailbox. Writing a ‘1’ to this bit position has no effect. The returned value is ‘0’.

(2) Transmit Cancel Register (TXCR1, TXCR0)

The TXCR1 and TXCR0 are 16-bit read/conditionally-write registers. The TXCR1 controls Mailbox-31 to Mailbox-16, and the TXCR0 controls Mailbox-15 to Mailbox-1.This register is used by the CPU to request the pending transmission requests in the TXPR to be cancelled. To clear the corresponding bit in the TXPR the CPU must write a '1' to the bit position in the TXCR. Writing a '0' has no effect.

When an abort has succeeded the CAN controller clears the corresponding TXPR + TXCR bits, and sets the corresponding ABACK bit. However, once a Mailbox has started a transmission, it cannot be cancelled by this bit. In such a case, if the transmission finishes in success, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding TXACK bit, however, if the transmission fails due to a bus arbitration loss or an error on the bus, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding ABACK bit. If an attempt is made by the CPU to clear a mailbox transmission that is not transmit-pending it has no effect. In this case the CPU will be not able at all to set the TXCR flag.

- TXCR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only writing a '1' to a Mailbox that is requested for transmission and is configured as transmit.

Bit 15 to 0 — Requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 0 corresponds to Mailbox-31 to 16 (and TXPR1[15:0]) respectively.

Bit[15:0]:TXCR1	Description
0	Transmit message cancellation idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of transmit message cancellation (automatically cleared)
1	Transmission cancellation request made for corresponding mailbox

- TXCR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR0[15:1]															-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R

Note: * Only writing a ‘1’ to a Mailbox that is requested for transmission and is configured as transmit.

Bit 15 to 1 — Requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 1 corresponds to Mailbox-15 to 1 (and TXPR0[15:1]) respectively.

Bit[15:1]: TXCR0	Description
0	Transmit message cancellation idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of transmit message cancellation (automatically cleared)
1	Transmission cancellation request made for corresponding mailbox

Bit 0 — This bit is always ‘0’ as this is a receive-only mailbox. Writing a ‘1’ to this bit position has no effect and always read back as a ‘0’.

(3) Transmit Acknowledge Register (TXACK1, TXACK0)

The TXACK1 and TXACK0 are 16-bit read/conditionally-write registers. These registers are used to signal to the CPU that a mailbox transmission has been successfully made. When a transmission has succeeded the RCAN-TL1 sets the corresponding bit in the TXACK register. The CPU may clear a TXACK bit by writing a ‘1’ to the corresponding bit location. Writing a ‘0’ has no effect.

- TXACK1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a ‘1’ to clear.

Bit 15 to 0 — Notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively.

Bit[15:0]:TXACK1	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox has successfully transmitted message (Data or Remote Frame) [Setting Condition] Completion of message transmission for corresponding mailbox

- TXACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK0[15:1]															-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	-

Note: * Only when writing a ‘1’ to clear.

Bit 15 to 1 — Notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

Bit[15:1]:TXACK0	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox has successfully transmitted message (Data or Remote Frame) [Setting Condition] Completion of message transmission for corresponding mailbox

Bit 0 — This bit is always ‘0’ as this is a receive-only mailbox. Writing a ‘1’ to this bit position has no effect and always read back as a ‘0’.

(4) Abort Acknowledge Register (ABACK1, ABACK0)

The ABACK1 and ABACK0 are 16-bit read/conditionally-write registers. These registers are used to signal to the CPU that a mailbox transmission has been aborted as per its request. When an abort has succeeded the RCAN-TL1 sets the corresponding bit in the ABACK register. The CPU may clear the Abort Acknowledge bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect. An ABACK bit position is set by the RCAN-TL1 to acknowledge that a TXPR bit has been cleared by the corresponding TXCR bit.

- ABACK1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively.

Bit[15:0]:ABACK1	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame) [Setting Condition] Completion of transmission cancellation for corresponding mailbox

- ABACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK0[15:1]															-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R

Note: * Only when writing a '1' to clear.

Bit 15 to 1 — Notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

Bit[15:1]:ABACK0	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame) [Setting Condition] Completion of transmission cancellation for corresponding mailbox

Bit 0 — This bit is always ‘0’ as this is a receive-only mailbox. Writing a ‘1’ to this bit position has no effect and always read back as a ‘0’.

(5) Data Frame Receive Pending Register (RXPR1, RXPR0)

The RXPR1 and RXPR0 are 16-bit read/conditionally-write registers. The RXPR is a register that contains the received Data Frames pending flags associated with the configured Receive Mailboxes. When a CAN Data Frame is successfully stored in a receive mailbox the corresponding bit is set in the RXPR. The bit may be cleared by writing a ‘1’ to the corresponding bit position. Writing a ‘0’ has no effect. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Data Frames. When a RXPR bit is set, it also sets IRR1 (Data Frame Received Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR1 is not set. Please note that these bits are only set by receiving Data Frames and not by receiving Remote frames.

- RXPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note : * Only when writing a ‘1’ to clear.

Bit 15 to 0 — Configurable receive mailbox locations corresponding to each mailbox position from 31 to 16 respectively.

Bit[15:0]: RXPR1	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox received a CAN Data Frame [Setting Condition] Completion of Data Frame receive on corresponding mailbox

- RXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a ‘1’ to clear.

Bit 15 to 0 — Configurable receive mailbox locations corresponding to each mailbox position from 15 to 0 respectively.

Bit[15:0]: RXPR0	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox received a CAN Data Frame [Setting Condition] Completion of Data Frame receive on corresponding mailbox

(6) Remote Frame Receive Pending Register (RFPR1, RFPR0)

The RFPR1 and RFPR0 are 16-bit read/conditionally-write registers. The RFPR is a register that contains the received Remote Frame pending flags associated with the configured Receive Mailboxes. When a CAN Remote Frame is successfully stored in a receive mailbox the corresponding bit is set in the RFPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. In effect there is a bit position for all mailboxes. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Remote Frames. When a RFPR bit is set, it also sets IRR2 (Remote Frame Receive Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR2 is not set. Please note that these bits are only set by receiving Remote Frames and not by receiving Data frames.

- RFPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Remote Request pending flags for mailboxes 31 to 16 respectively.

Bit[15:0]: RFPR1	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox received Remote Frame [Setting Condition] Completion of remote frame receive in corresponding mailbox

- RFPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Remote Request pending flags for mailboxes 15 to 0 respectively.

Bit[15:0]: RFPR0	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox received Remote Frame [Setting Condition] Completion of remote frame receive in corresponding mailbox

(7) Mailbox Interrupt Mask Register (MBIMR)

The MBIMR1 and MBIMR0 are 16-bit read/write registers. The MBIMR only prevents the setting of IRR related to the Mailbox activities, that are IRR[1] – Data Frame Received Interrupt, IRR[2] – Remote Frame Receive Interrupt, IRR[8] – Mailbox Empty Interrupt, and IRR[9] – Message OverRun/OverWrite Interrupt. If a mailbox is configured as receive, a mask at the corresponding bit position prevents the generation of a receive interrupt (IRR[1] and IRR[2] and IRR[9]) but does not prevent the setting of the corresponding bit in the RXPR or RFPR or UMSR. Similarly when a mailbox has been configured for transmission, a mask prevents the generation of an Interrupt signal and setting of an Mailbox Empty Interrupt due to successful transmission or abortion of transmission (IRR[8]), however, it does not prevent the RCAN-TL1 from clearing the corresponding TXPR/TXCR bit + setting the TXACK bit for successful transmission, and it does not prevent the RCAN-TL1 from clearing the corresponding TXPR/TXCR bit + setting the ABACK bit for abortion of the transmission.

A mask is set by writing a ‘1’ to the corresponding bit position for the mailbox activity to be masked. At reset all mailbox interrupts are masked.

- MBIMR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR1[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0 — Enable or disable interrupt requests from individual Mailbox-31 to Mailbox-16 respectively.

Bit[15:0]: MBIMR1	Description
0	Interrupt Request from IRR1/IRR2/IRR8/IRR9 enabled
1	Interrupt Request from IRR1/IRR2/IRR8/IRR9 disabled (initial value)

- MBIMR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR0[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0 — Enable or disable interrupt requests from individual Mailbox-15 to Mailbox-0 respectively.

Bit[15:0]: MBIMR0	Description
0	Interrupt Request from IRR1/IRR2/IRR8/IRR9 enabled
1	Interrupt Request from IRR1/IRR2/IRR8/IRR9 disabled (initial value)

(8) Unread Message Status Register (UMSR)

This register is a 32-bit read/conditionally write register and it records the mailboxes whose contents have not been accessed by the CPU prior to a new message being received. If the CPU has not cleared the corresponding bit in the RXPR or RFPR when a new message for that mailbox is received, the corresponding UMSR bit is set to ‘1’. This bit may be cleared by writing a ‘1’ to the corresponding bit location in the UMSR. Writing a ‘0’ has no effect.

If a mailbox is configured as transmit box, the corresponding UMSR will not be set.

- UMSR1

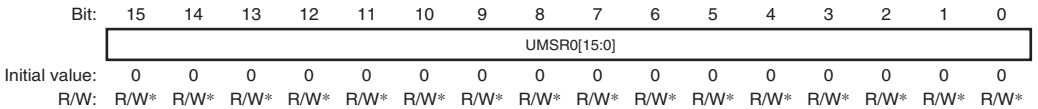
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a ‘1’ to clear.

Bit 15 to 0 — Indicate that an unread received message has been overwritten or overrun condition has occurred for Mailboxes 31 to 16.

Bit[15:0]: UMSR1	Description
0	[Clearing Condition] Writing ‘1’ (initial value)
1	Unread received message is overwritten by a new message or overrun condition [Setting Condition] When a new message is received before RXPR or RFPR is cleared

- UMSR0



Note: * Only when writing a ‘1’ to clear.

Bit 15 to 0 — Indicate that an unread received message has been overwritten or overrun condition has occurred for Mailboxes 15 to 0.

Bit[15:0]: UMSR0	Description
0	[Clearing Condition] Writing ‘1’ (initial value)
1	Unread received message is overwritten by a new message or overrun condition [Setting Condition] When a new message is received before RXPR or RFPR is cleared

20.3.5 Timer Registers

The Timer is 16 bits and supports several source clocks. A pre-scale counter can be used to reduce the speed of the clock. It also supports three Compare Match Registers (TCMR2, TCMR1, TCMR0). The address map is as follows.

Important: These registers can only be accessed in Word size (16-bit).

Description	Address	Name	Access Size (bits)
Timer Trigger Control Register 0	080	TTCR0	Word (16)
Cycle Maximum/Tx-Enable Window Register	084	CMAX_TEW	Word (16)
Reference Trigger Offset Register	086	RFTROFF	Word (16)
Timer Status Register	088	TSR	Word (16)
Cycle Counter Register	08A	CCR	Word (16)
Timer Counter Register	08C	TCNTR	Word (16)
Cycle Time Register	090	CYCTR	Word (16)
Reference Mark Register	094	RFMK	Word (16)
Timer Compare Match Register 0	098	TCMR0	Word (16)
Timer Compare Match Register 1	09C	TCMR1	Word (16)
Timer Compare Match Register 2	0A0	TCMR2	Word (16)
Tx-Trigger Time Selection Register	0A4	TTTSEL	Word (16)

Figure 20.12 RCAN-TL1 Timer registers

(1) Time Trigger Control Register0 (TTCR0)

The Time Trigger Control Register0 is a 16-bit read/write register and provides functions to control the operation of the Timer. When operating in Time Trigger Mode, please refer to section 20.4.3 (1), Time Triggered Transmission.

- TTCR0 (Address = H'080)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	-	-	-	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 — Enable Timer: When this bit is set, the timer TCNTR is running. When this bit is cleared, TCNTR and CCR are cleared.

Bit15: TTCR0 15	Description
0	Timer and CCR are cleared and disabled (initial value)
1	Timer is running

Bit 14 — TimeStamp value: Specifies if the Timestamp for transmission and reception in Mailboxes 15 to 0 must contain the Cycle Time (CYCTR) or the concatenation of CCR[5:0] + CYCTR[15:6]. This feature is very useful for time triggered transmission to monitor Rx_Trigger.

This register does not affect the TimeStamp for Mailboxes 30 and 31.

Bit14: TTCR0 14	Description
0	CYCTR[15:0] is used for the TimeStamp in Mailboxes 15 to 0 (initial value)
1	CCR[5:0] + CYCTR[15:6] is used for the TimeStamp in Mailboxes 15 to 0

Bit 13 — Cancellation by TCMR2: The messages in the transmission queue are cancelled by setting TXCR, when both this bit and bit12 are set and compare match occurs when RCAN-TL1 is not in the Halt status, causing the setting of all TXCR bits with the corresponding TXPR bits set.

Bit13: TTCR0 13	Description
0	Cancellation by TCMR2 compare match is disabled (initial value)
1	Cancellation by TCMR2 compare match is enabled

Bit 12 — TCMR2 compare match enable: When this bit is set, IRR11 is set by TCMR2 compare match.

Bit12 TTCR0 12	Description
0	IRR11 isn't set by TCMR2 compare match (initial value)
1	IRR11 is set by TCMR2 compare match

Bit 11 — TCMR1 compare match enable: When this bit is set, IRR15 is set by TCMR1 compare match.

Bit11 TTCR0 11	Description
0	IRR15 isn't set by TCMR1 compare match (initial value)
1	IRR15 is set by TCMR1 compare match

Bit 10 — TCMR0 compare match enable: When this bit is set, IRR14 is set by TCMR0 compare match.

Bit10 TTCR0 10	Description
0	IRR14 isn't set by TCMR0 compare match (initial value)
1	IRR14 is set by TCMR0 compare match

Bits 9 to 7: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 6 — Timer Clear-Set Control by TCMR0: Specifies if the Timer is to be cleared and set to H'0000 when the TCMR0 matches to the TCNTR. Please note that the TCMR0 is also capable to generate an interrupt signal to the CPU via IRR14.

Note: If RCAN-TL1 is working in TTCAN mode (CMAX isn't 3'b111), TTCR0 bit6 has to be ‘0’ to avoid clearing Local Time.

Bit6: TTCR0 6	Description
0	Timer is not cleared by the TCMR0 (initial value)
1	Timer is cleared by the TCMR0

Bit5 to 0 — RCAN-TL1 Timer Prescaler (TPSC[5:0]): This control field allows the timer source clock (4*[RCAN-TL1 system clock]) to be divided before it is used for the timer. This function is available only in event-trigger mode. In time trigger mode (CMAX is not 3'b111), one nominal Bit Timing (= one bit length of CAN bus) is automatically chosen as source clock of TCNTR.

The following relationship exists between source clock period and the timer period.

Bit[5:0]: TPSC[5:0]	Description
0 0 0 0 0 0	1 X Source Clock (initial value)
0 0 0 0 0 1	2 X Source Clock
0 0 0 0 1 0	3 X Source Clock
0 0 0 0 1 1	4 X Source Clock
0 0 0 1 0 0	5 X Source Clock
. . . . .	. . . . .
. . . . .	. . . . .
1 1 1 1 1 1	64 X Source Clock

(2) Cycle Maximum/Tx-Enable Window Register (CMAX_TEW)

This register is a 16-bit read/write register. CMAX specifies the maximum value for the cycle counter (CCR) for TT Transmissions to set the number of basic cycles in the matrix system. When the Cycle Counter reaches the maximum value (CCR = CMAX), after a full basic cycle, it is cleared to zero and an interrupt is generated on IRR.10.

TEW specifies the width of Tx-Enable window.

- CMAX_TEW (Address = H'084)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMAX[2:0]			-	-	-	-	TEW[3:0]			
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bits 15 to 11: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 10 to 8 — Cycle Count Maximum (CMAX): Indicates the maximum number of CCR. The number of basic cycles available in the matrix cycle for Timer Triggered transmission is (Cycle Count Maximum + 1).

Unless CMAX = 3'b111, RCAN-TL1 is in time-trigger mode and time trigger function is available. If CMAX = 3'b111, RCAN-TL1 is in event-trigger mode.

Bit[10:8]: CMAX[2:0]	Description
0 0 0	Cycle Count Maximum = 0
0 0 1	Cycle Count Maximum = 1
0 1 0	Cycle Count Maximum = 3
0 1 1	Cycle Count Maximum = 7
1 0 0	Cycle Count Maximum = 15
1 0 1	Cycle Count Maximum = 31
1 1 0	Cycle Count Maximum = 63
1 1 1	CCR is cleared and RCAN-TL1 is in event-trigger mode. (initial value)

Important: Please set CMAX = 3'b111 when event-trigger mode is used.

Bits 7 to 4: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 3 to 0 — Tx-Enable Window (TEW): Indicates the width of Tx-Enable Window. TEW = H'00 shows the width is one nominal Bit Timing. All values from 0 to 15 are allowed to be set.

Bit[3:0]: TEW[3:0]	Description
0 0 0 0	The width of Tx-Enable Window = 1 (initial value)
0 0 0 1	The width of Tx-Enable Window = 2
0 0 1 0	The width of Tx-Enable Window = 3
0 0 1 1	The width of Tx-Enable Window = 4
. . . .	. . . . .
. . . .	. . . . .
1 1 1 1	The width of Tx-Enable Window = 16

Note: The CAN core always needs a time between 1 to 2 bit timing to initiate transmission. The above values are not considering this accuracy.

(3) Reference Trigger Offset Register (RFTROFF)

This is a 8-bit read/write register that affects Tx-Trigger Time (TTT) of Mailbox-30. The TTT of Mailbox-30 is compared with CYCTR after RFTROFF extended with sign is added to the TTT. However, the value of TTT is not modified. The offset value doesn't affect others except Mailbox-30.

- RFTROFF (Address = H'086)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTROFF[7:0]								-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit 15 to 8 — Indicate the value of Reference Trigger Offset.

Bits 7 to 0: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Description
0	0	0	0	0	0	0	0	Ref_trigger_offset = +0 (initial value)
0	0	0	0	0	0	0	1	Ref_trigger_offset = +1
0	0	0	0	0	0	1	0	Ref_trigger_offset = +2
.	.	.	.	.	.	.	.	
0	1	1	1	1	1	1	1	Ref_trigger_offset = +127
.	.	.	.	.	.	.	.	
1	1	1	1	1	1	1	1	Ref_trigger_offset = –1
1	1	1	1	1	1	1	0	Ref_trigger_offset = –2
.	.	.	.	.	.	.	.	
1	0	0	0	0	0	0	1	Ref_trigger_offset = –127
1	0	0	0	0	0	0	0	Prohibited

(4) Timer Status Register (TSR)

This register is a 16-bit read-only register, and allows the CPU to monitor the Timer Compare Match status and the Timer Overrun Status.

- TSR (Address = H'088)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	TSR4	TSR3	TSR2	TSR1	TSR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bits 15 to 5: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 4 to 0 — RCAN-TL1 Timer Status (TSR[4:0]): This read-only field allows the CPU to monitor the status of the Cycle Counter, the Timer and the Compare Match registers. Writing to this field has no effect.

Bit 4 — Start of New System Matrix (TSR4): Indicates that a new system matrix is starting. When CCR = 0, this bit is set at the successful completion of reception/transmission of time reference message.

Bit4: TSR4	Description
0	A new system matrix is not starting (initial value) [Clearing condition] Writing ‘1’ to IRR10 (Cycle Counter Overflow Interrupt)
1	Cycle counter reached zero [Setting condition] When the Cycle Counter value changes from the maximum value (CMAX) to H'0. Reception/transmission of time reference message is successfully completed when CMAX!= 3'b111 and CCR = 0

Bit 3 — Timer Compare Match Flag 2 (TSR3): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 2 (TCMR2). When the value set in the TCMR2 matches to Cycle Time Register (TCMR2 = CYCTR), this bit is set if TTCR0 bit12 = 1. Please note that this bit is read-only and is cleared when IRR11 (Timer Compare Match Interrupt 2) is cleared.

Bit3: TSR3	Description
0	Timer Compare Match has not occurred to the TCMR2 (Initial value) [Clearing condition] Writing ‘1’ to IRR11 (Timer Compare Match Interrupt 1)
1	Timer Compare Match has occurred to the TCMR2 [Setting condition] TCMR2 matches to Cycle Time (TCMR2 = CYCTR), if TTCR0 bit12 = 1.

Bit 2 — Timer Compare Match Flag 1 (TSR2): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 1 (TCMR1). When the value set in the TCMR1 matches to Cycle Time Register (TCMR1 = CYCTR), this bit is set if TTCR0 bit11 = 1. Please note that this bit is read-only and is cleared when IRR15 (Timer Compare Match Interrupt 1) is cleared.

Bit2: TSR2	Description
0	Timer Compare Match has not occurred to the TCMR1 (Initial value) [Clearing condition] Writing ‘1’ to IRR15 (Timer Compare Match Interrupt 1)
1	Timer Compare Match has occurred to the TCMR1 [Setting condition] TCMR1 matches to Cycle Time (TCMR1 = CYCTR), if TTCR0 bit11 = 1.

Bit 1 — Timer Compare Match Flag 0 (TSR1): Indicates that a Compare-Match condition occurred to the Compare Match Register 0 (TCMR0). When the value set in the TCMR0 matches to the Timer value (TCMR0 = TCNTR), this bit is set if TTCR0 bit10 = 1. Please note that this bit is read-only and is cleared when IRR14 (Timer Compare Match Interrupt 0) is cleared.

Bit1: TSR1	Description
0	Compare Match has not occurred to the TCMR0 (Initial value) [Clearing condition] Writing ‘1’ to IRR14 (Timer Compare Match Interrupt 0)
1	Compare Match has occurred to the TCMR0 [Setting condition] TCMR0 matches to the Timer value (TCMR0 = TCNTR)

Bit 0 — Timer Overrun/Next_is_Gap Reception/Message Error (TSR0): This flag is assigned to three different functions. It indicates that the Timer has overrun when working in event-trigger mode, time reference message with Next_is_Gap set has been received in time-trigger mode, and error detected on the CAN bus has occurred in test mode, respectively. Test mode has higher priority with respect to the other settings.

Bit0: TSR0	Description
0	Timer (TCNTR) has not overrun in event-trigger mode (Initial value) Time reference message with Next_is_Gap has not been received in time-trigger mode message error has not occurred in test mode. [Clearing condition] Writing ‘1’ to IRR13
1	[Setting condition] Timer (TCNTR) has overrun and changed from H'FFFF to H'0000 in event-trigger mode.time reference message with Next_is_Gap has been received in time-trigger mode message error has occurred in test mode

(5) Cycle Counter Register (CCR)

This register is a 6-bit read/write register. Its purpose is to store the number of the basic cycle for Time -Triggered Transmissions. Its value is updated in different fashions depending if RCAN-TL1 is programmed to work as a potential time master or as a time slave. If RCAN-TL1 is working as (potential) time master, CCR is:

— Incremented by one every time the cycle time (CYCTR) matches to Tx-Trigger Time of Mailbox-30 or

— Overwritten with the value contained in MSG_DATA_0[5:0] of Mailbox 31 when a valid reference message is received.

If RCAN-TL1 is working as a time slave, CCR is only overwritten with the value of MSG_DATA_0[5:0] of Mailbox 31 when a valid reference message is received.

If CMAX = 3'111, CCR is always H'0000.

- CCR (Address = H'08A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	CCR[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 6: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 5 to 0 — Cycle Counter Register (CCR): Indicates the number of the current Base Cycle of the matrix cycle for Timer Triggered transmission.

(6) Timer Counter Register (TCNTR)

This is a 16-bit read/write register that allows the CPU to monitor and modify the value of the Free Running Timer Counter. When the Timer meets TCMR0 (Timer Compare Match Register 0) + TTCR0 [6] is set to '1', the TCNTR is cleared to H'0000 and starts running again. In Time-Trigger mode, this timer can be used as Local Time and TTCR0[6] has to be cleared to work as a free running timer.

Notes: 1. It is possible to write into this register only when it is enabled by the bit 15 in TTCR0. If TTCR0 bit15 = 0, TCNTR is always H'0000.
2. There could be a delay of a few clock cycles between the enabling of the timer and the moment where TCNTR starts incrementing. This is caused by the internal logic used for the pre-scaler.

- TCNTR (Address = H'08C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNTR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * The register can be written only when enabled in TTCR0[15]. Write operation is not allowed in Time Trigger mode (i.e. CMAX is not 3'b111).

Bit 15 to 0 — Indicate the value of the Free Running Timer.

(7) Cycle Time register (CYCTR)

This register is a 16-bit read-only register. This register shows Cycle Time = Local Time (TCNTR) - Reference_Mark (RFMK). In ET mode this register is the exact copy of TCNTR as RFMK is always fixed to zero.

- CYCTR (Address = H'090)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CYCTR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

(8) Reference Mark Register (RFMK)

This register is a 16-bit read-only register. The purpose of this register is to capture Local Time (TCNTR) at SOF of the reference message when the message is received or transmitted successfully. In ET mode this register is not used and it is always cleared to zero.

- RFMK (Address = H'094)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMK[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit 15 to 0 — Reference Mark Register (RFMK): Indicates the value of TCNTR at SOF of time reference message.

(9) Timer Compare Match Registers (TCMR0, TCMR1, TCMR2)

These three registers are 16-bit read/write registers and are capable of generating interrupt signals, clearing-setting the Timer value (only supported by TCMR0) or clear the transmission messages in the queue (only supported by TCMR2). TCMR0 is compared with TCNTR, however, TCMR1 and TCMR2 are compared with CYCTR.

The value used for the compare can be configured independently for each register. In order to set flags, TTCR0 bit 12-10 needs to be set.

In Time-Trigger mode, TTCR0 bit6 has to be cleared by software to prevent TCNTR from being cleared.

TMCR0 is for Init_Watch_Trigger, and TCMR2 is for Watch_Trigger.

Interrupt:

The interrupts are flagged by the Bit11, Bit15 and 14 in the IRR accordingly when a Compare Match occurs, and setting these bits can be enabled by Bit12, Bit11, Bit10 in TTCR0. The generation of interrupt signals itself can be prevented by the Bit11, Bit15 and Bit14 in the IMR. When a Compare Match occurs and the IRR11 (or IRR15 or IRR14) is set, the Bit3 or Bit2 or Bit1 in the TSR (Timer Status Register) is also set. Clearing the IRR bit also clears the corresponding bit of TSR.

Timer Clear-Set:

The Timer value can only be cleared when a Compare Match occurs if it is enabled by the Bit6 in the TTCR0. TCMR1 and TCMR2 do not have this function.

Cancellation of the messages in the transmission queue:

The messages in the transmission queue can only be cleared by the TCMR2 through setting TXCR when a Compare Match occurs while RCAN-TL1 is not in the halt status. TCMR1 and TCMR0 do not have this function.

- TCMR0 (Address = H'098)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR0[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0 — Timer Compare Match Register (TCMR0): Indicates the value of TCNTR when compare match occurs.

- TCMR1 (Address = H'09C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR1[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0 — Timer Compare Match Register (TCMR1): Indicates the value of CYCTR when compare match occurs.

- TCMR2 (Address = H'0A0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR2[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0 — Timer Compare Match Register (TCMR2): Indicates the value of CYCTR when compare match occurs.

(10) Tx-Trigger Time Selection Register (TTTSEL)

This register is a 16-bit read/write register and specifies the Tx-Trigger Time waiting for compare match with Cycle Time. Only one bit is allowed to be set. Please don't set more bits than one, or clear all bits.

This register may only be modified during configuration mode. The modification algorithm is shown in figure 20.13.

Please note that this register is only indented for test and diagnosis. When not in test mode, this register must not be written to and the returned value is not guaranteed.

- TTTSEL (Address = H'0A4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TTTSEL[14:8]							-	-	-	-	-	-	-	-
Initial value:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Note: Only one bit is allowed to be set.

Bit 15: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 14 to 8 — Specifies the Tx-Trigger Time waiting for compare match with CYCTR The bit 14 to 8 corresponds to Mailbox-30 to 24, respectively.

Bits 7 to 0: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

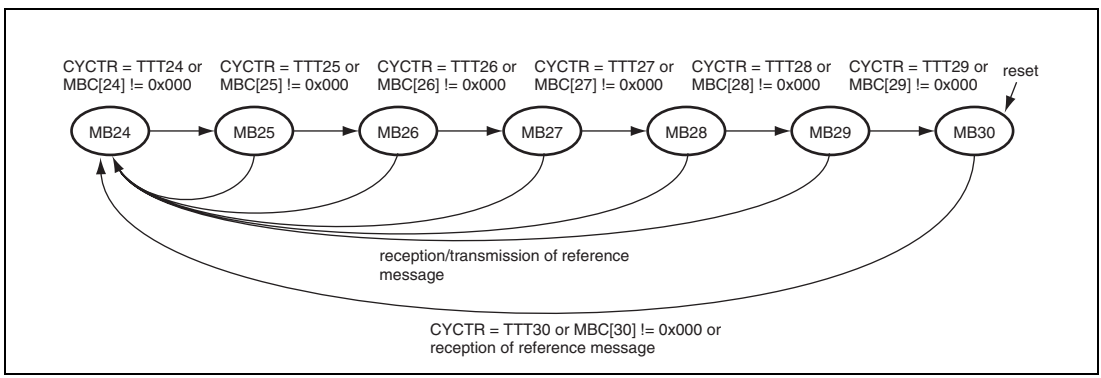


Figure 20.13 TTTSEL modification algorithm

20.4 Application Note

20.4.1 Test Mode Settings

The RCAN-TL1 has various test modes. The register TST[2:0] (MCR[10:8]) is used to select the RCAN-TL1 test mode. The default (initialised) settings allow RCAN-TL1 to operate in Normal mode. The following table is examples for test modes.

Test Mode can be selected only while in configuration mode. The user must then exit the configuration mode (ensuring BCR0/BCR1 is set) in order to run the selected test mode.

Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	Normal Mode (initial value)
0	0	1	Listen-Only Mode (Receive-Only Mode)
0	1	0	Self Test Mode 1 (External)
0	1	1	Self Test Mode 2 (Internal)
1	0	0	Write Error Counter
1	0	1	Error Passive Mode
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Normal Mode: RCAN-TL1 operates in the normal mode.

Listen-Only Mode: ISO-11898 requires this mode for baud rate detection. The Error Counters are cleared and disabled so that the TEC/REC does not increase the values, and the CTxn (n = 0, 1) Output is disabled so that RCAN-TL1 does not generate error frames or acknowledgment bits. IRR13 is set when a message error occurs.

Self Test Mode 1: RCAN-TL1 generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The CRxn/CTxn (n = 0, 1) pins must be connected to the CAN bus.

Self Test Mode 2: RCAN-TL1 generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The CRxn/CTxn (n = 0, 1) pins do not need to be connected to the CAN bus or any external devices, as the internal CTxn (n = 0, 1) is looped back to the internal CRxn (n = 0, 1). CTxn (n = 0, 1) pin outputs only recessive bits and CRxn (n = 0, 1) pin is disabled.

Write Error Counter: TEC/REC can be written in this mode. RCAN-TL1 can be forced to become an Error Passive mode by writing a value greater than 127 into the Error Counters. The value written into TEC is used to write into REC, so only the same value can be set to these registers. Similarly, RCAN-TL1 can be forced to become an Error Warning by writing a value greater than 95 into them.

RCAN-TL1 needs to be in Halt Mode when writing into TEC/REC (MCR1 must be "1" when writing to the Error Counter). Furthermore this test mode needs to be exited prior to leaving Halt mode.

Error Passive Mode: RCAN-TL1 can be forced to enter Error Passive mode.
Note: The REC will not be modified by implementing this Mode. However, once running in Error Passive Mode, the REC will increase normally should errors be received. In this Mode, RCAN-TL1 will enter BusOff if TEC reaches 256 (Dec). However when this mode is used RCAN-TL1 will not be able to become Error Active. Consequently, at the end of the Bus Off recovery sequence, RCAN-TL1 will move to Error Passive and not to Error Active.

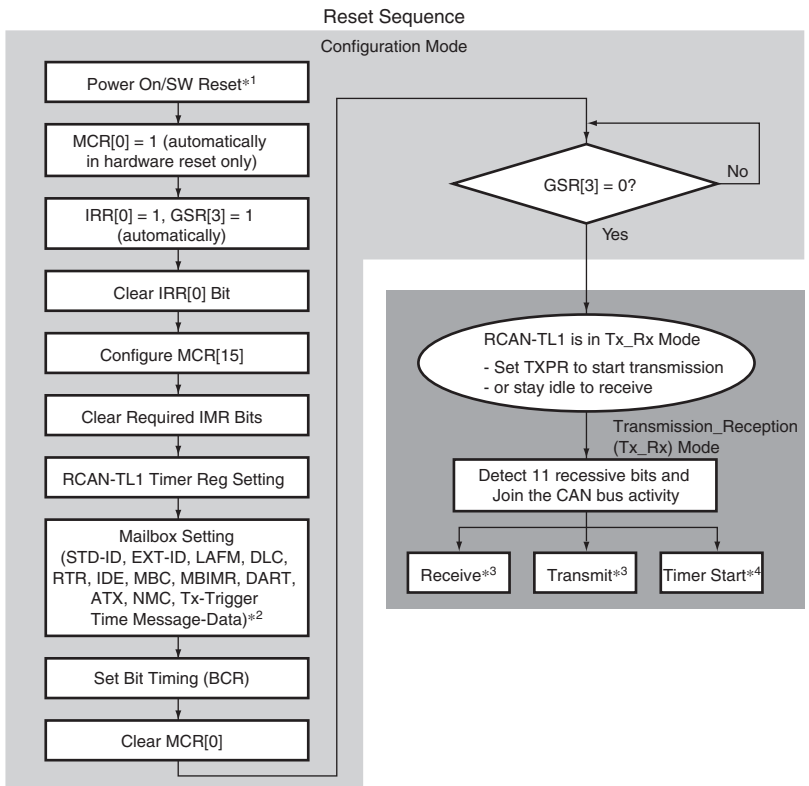
When message error occurs, IRR13 is set in all test modes.

20.4.2 Configuration of RCAN-TL1

RCAN-TL1 is considered in configuration mode or after a H/W (Power On Reset)/S/W (MCR[0]) reset or when in Halt mode. In both conditions RCAN-TL1 cannot join the CAN Bus activity and configuration changes have no impact on the traffic on the CAN Bus.

- After a Reset request

The following sequence must be implemented to configure the RCAN-TL1 after (S/W or H/W) reset. After reset, all the registers are initialised, therefore, RCAN-TL1 needs to be configured before joining the CAN bus activity. Please read the notes carefully.



Notes: 1. SW reset could be performed at any time by setting MCR[0] = 1.
2. Mailboxes are comprised of RAMs, therefore, please initialise all the mailboxes enabled by MBC.
3. If there is no TXPR set, RCAN-TL1 will receive the next incoming message. If there is a TXPR(s) set, RCAN-TL1 will start transmission of the message and will be arbitrated by the CAN bus. If it loses the arbitration, it will become a receiver.
4. Timer can be started at any time after the Timer Control regs and Tx-Trigger Time are set.

Figure 20.14 Reset Sequence

- Halt mode

When RCAN-TL1 is in Halt mode, it cannot take part to the CAN bus activity. Consequently the user can modify all the requested registers without influencing existing traffic on the CAN Bus. It is important for this that the user waits for the RCAN-TL1 to be in halt mode before to modify the requested registers - note that the transition to Halt Mode is not always immediate (transition will occurs when the CAN Bus is idle or in intermission). After RCAN-TL1 transit to Halt Mode, GSR4 is set.

Once the configuration is completed the Halt request needs to be released. RCAN-TL1 will join CAN Bus activity after the detection of 11 recessive bits on the CAN Bus.

- Sleep mode

When RCAN-TL1 is in sleep mode the clock for the main blocks of the IP is stopped in order to reduce power consumption. Only the following user registers are clocked and can be accessed: MCR, GSR, IRR and IMR. Interrupt related to transmission (TXACK and ABACK) and reception (RXPR and RFPR) cannot be cleared when in sleep mode (as TXACK, ABACK, RXPR and RFPR are not accessible) and must to be cleared beforehand.

The following diagram shows the flow to follow to move RCAN-TL1 into sleep mode.

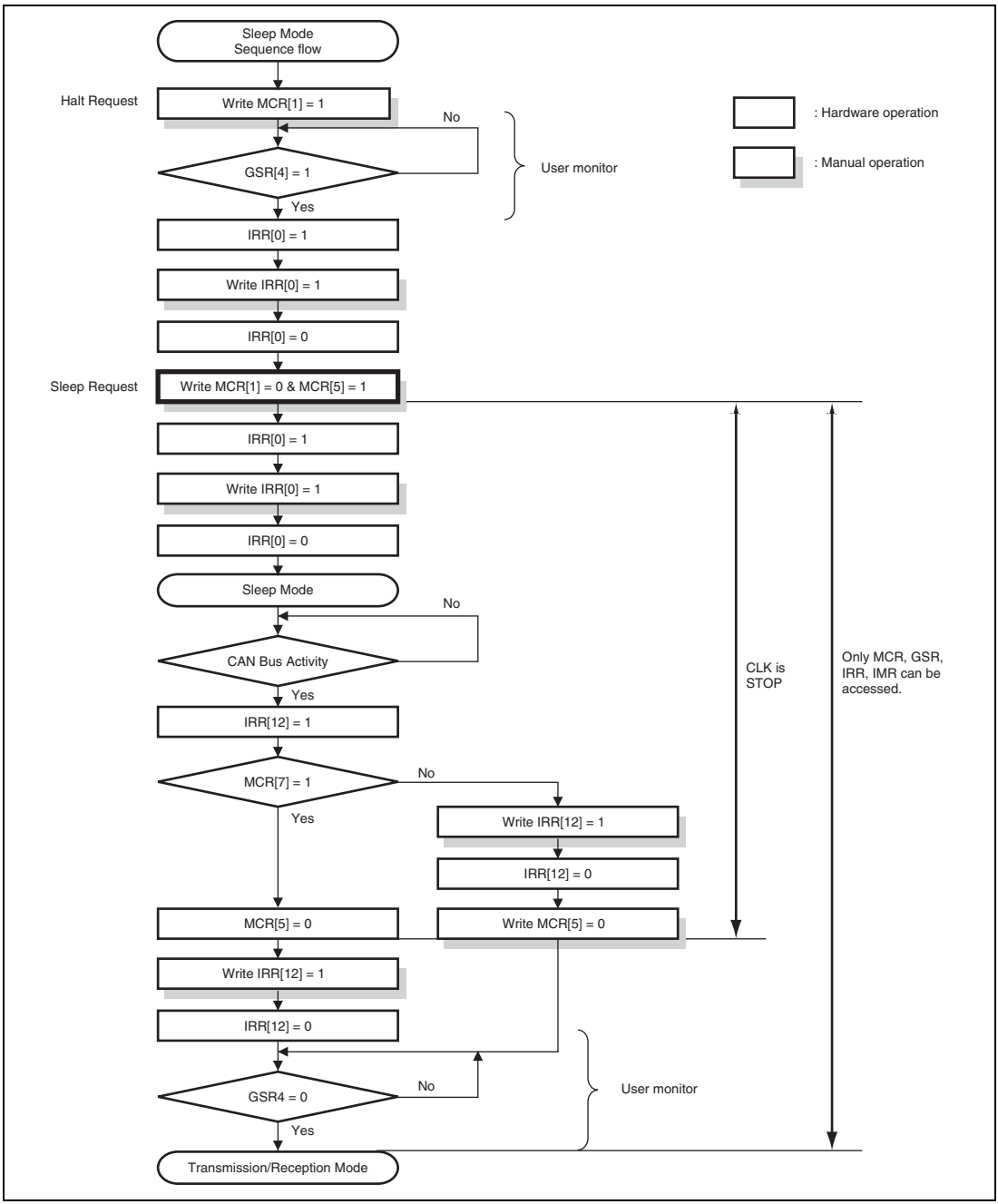


Figure 20.15 shows allowed state transitions.

— Please don't set MCR5 (Sleep Mode) without entering Halt Mode.

— After MCR1 is set, please don't clear it before GSR4 is set and RCAN-TL1 enters Halt Mode.

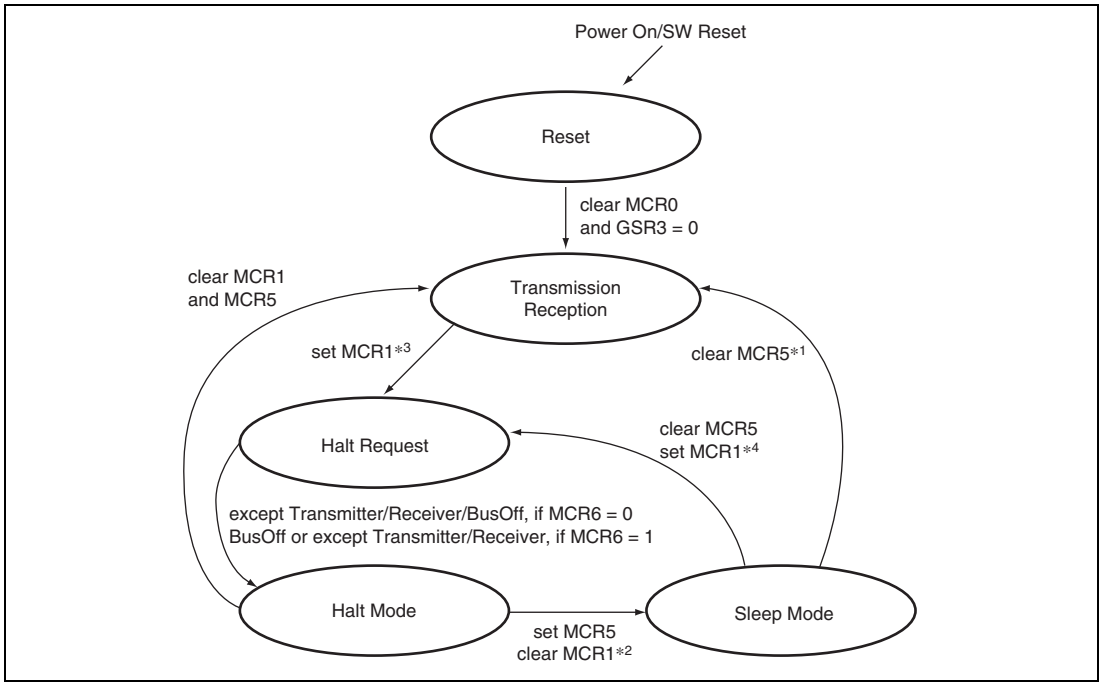


Figure 20.15 Halt Mode/Sleep Mode

Notes: 1. MCR5 can be cleared by automatically by detecting a dominant bit on the CAN Bus if MCR7 is set or by writing '0'.

2. MCR1 is cleared in SW. Clearing MCR1 and setting MCR5 have to be carried out by the same instruction.

3. MCR1 must not be cleared in SW, before GSR4 is set. MCR1 can be set automatically in HW when RCAN-TL1 moves to Bus Off and MCR14 and MCR6 are both set.

4. When MCR5 is cleared and MCR1 is set at the same time, RCAN-TL1 moves to Halt Request. Right after that, it moves to Halt Mode with no reception/transmission.

The following table shows conditions to access registers.

RCAN-TL1 Registers

Status Mode	MCR GSR	IRR IMR	BCR	MBIMR timer TT_register	Flag_ register	Mailbox (ctrl0, LAFM)		Mailbox (data)	Mailbox (ctrl1)		Mailbox Trigger Time TT control
Reset	yes	yes	yes	yes	yes	yes		yes	yes		yes
Transmission Reception Halt Request	yes	yes	no*1	yes	yes	no*1	yes*2	yes*2	no*1	yes*2	yes*2
Halt	yes	yes	no*1	yes	yes	yes		yes	yes		yes
Sleep	yes	yes	no	no	no	no		no	no		no

Notes: 1. No hardware protection.
2. When TXPR is not set.

20.4.3 Message Transmission Sequence

- Message Transmission Request

The following sequence is an example to transmit a CAN frame onto the bus. As described in the previous register section, please note that IRR8 is set when one of the TXACK or ABACK bits is set, meaning one of the Mailboxes has completed its transmission or transmission abortion and is now ready to be updated for the next transmission, whereas, the GSR2 means that there is currently no transmission request made (No TXPR flags set).

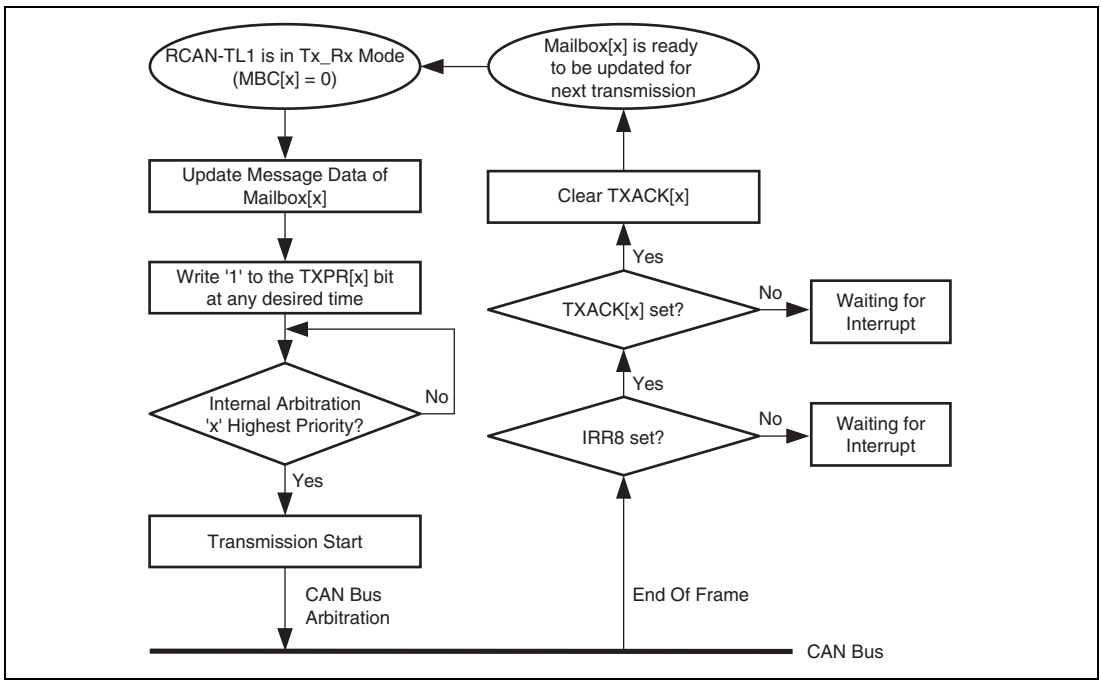


Figure 20.16 Transmission request

- Internal Arbitration for transmission

The following diagram explains how RCAN-TL1 manages to schedule transmission-requested messages in the correct order based on the CAN identifier. ‘Internal arbitration’ picks up the highest priority message amongst transmit-requested messages.

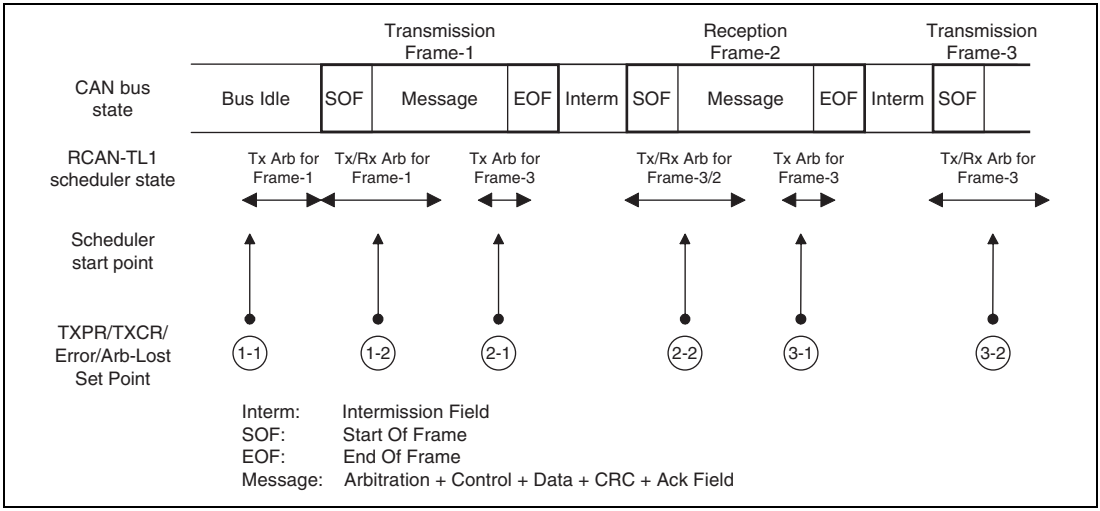


Figure 20.17 Internal Arbitration for transmission

The RCAN-TL1 has two state machines. One is for transmission, and the other is for reception.

1-1: When a TXPR bit(s) is set while the CAN bus is idle, the internal arbitration starts running immediately and the transmission is started.

1-2: Operations for both transmission and reception starts at SOF. Since there is no reception frame, RCAN-TL1 becomes transmitter.

2-1: At crc delimiter, internal arbitration to search next message transmitted starts.

2-2: Operations for both transmission and reception starts at SOF. Because of a reception frame with higher priority, RCAN-TL1 becomes receiver. Therefore, Reception is carried out instead of transmitting Frame-3.

3-1: At crc delimiter, internal arbitration to search next message transmitted starts.

3-2: Operations for both transmission and reception starts at SOF. Since a transmission frame has higher priority than reception one, RCAN-TL1 becomes transmitter.

Internal arbitration for the next transmission is also performed at the beginning of each error delimiter in case of an error is detected on the CAN Bus. It is also performed at the beginning of error delimiters following overload frame.

As the arbitration for transmission is performed at CRC delimiter, in case a remote frame request is received into a Mailbox with ATX = 1 the answer can join the arbitration for transmission only at the following Bus Idle, CRC delimiter or Error Delimiter.

Depending on the status of the CAN bus, following the assertion of the TXCR, the corresponding Message abortion can be handled with a delay of maximum 1 CAN Frame.

(1) Time Triggered Transmission

RCAN-TL1 offers a H/W support to perform communication in Time Trigger mode in line with the emerging ISO-11898-4 Level 1 Specification.

This section reports the basic procedures to use this mode.

- Setting Time Trigger Mode

In order to set up the time trigger mode the following settings need to be used.

— CMAX in CMAX_TEW must be programmed to a value different from 3'b111.
— Bit 15 in TTCR0 has to be set, to start TCNTR.
— Bit 6 in TTCR0 has to be cleared to prevent TCNTR from being cleared after a match.
— DART in Mailboxes used for time-triggered transmission cannot be used, since for Time Triggered Mailboxes, TXPR is not cleared to support periodic transmission.

- Roles of Registers

The user registers of RCAN-TL1 can be used to handle the main functions requested by the TTCAN standard.

TCNTR	Local Time
RFMK	Ref_Mark
CYCTR	Cycle Time = TCNTR - RFMK
RFTROFF	Ref_Trigger_Offset for Mailbox-30
Mailbox-31	Mailbox dedicated to the reception of time reference message
Mailbox-30	Mailbox dedicated to the transmission of time reference message when working as a potential time master
Mailbox-29 to 24	Mailboxes supporting time-triggered transmission
Mailbox-23 to 16	Mailboxes supporting reception without timestamp (may also be implemented as Mailboxes supporting Event Triggered transmission)
Mailbox-15 to 0	Mailboxes supporting reception with timestamp timestamp (may also be implemented as Mailboxes supporting Event Triggered transmission)
Tx-Trigger Time	Time_Mark to specify when a message should be transmitted

CMAX	Specifies the maximum number of basic cycles when working as potential time master
TEW	Specify the width of Tx_Enable
TCMR0	Init_Watch_Trigger (compare match with Local Time)
TCMR1	Compare match with Cycle Time to monitor users-specified events
TCMR2	Watch_Trigger (compare match with Cycle Time). This can be programmed to abort all pending transmissions
TTW	Specifies the attribute of a time window used for transmission
TTTSEL	Specifies the next Mailbox waiting for transmission

- Time Master/Time Slave

RCAN-TL1 can be programmed to work as a potential time master of the network or as a time slave. The following table shows the settings and the operation automatically performed by RCAN-TL1 in each mode.

mode	requested setting	function
Time Slave	TXPR[30] = 0 & MBC[30]!= 3'b000 & CMAX!= 3'b111 & MBC[31] = 3'b011	TCNTR is sampled at each SOF detected on the CAN Bus and stored into an internal register. When a valid Time Reference Message is received into Mailbox-31 the value of TCNTR (stored at the SOF) is copied into Ref_Mark. CCR embedded in the received Reference Message is copied to CCR. If Next_is_Gap = 1, IRR13 is set.
(Potential) Time Master	TXPR[30] = 1 & MBC[30] = 3'b000 & DLC[30] > 0 & CMAX!= 3'b111 & MBC[31] = 3'b011	Two cases are covered: (1) When a valid Time Reference message is received into Mailbox-31 the value of TCNTR stored into an internal register at the SOF is copied into Ref_Mark. CCR embedded in the received Reference Message is copied to CCR. If Next_is_Gap = 1, IRR13 is set. (2) When a Time Reference message is transmitted from Mailbox-30 the value of TCNTR stored into an internal register at the SOF is copied into Ref_Mark. CCR is incremented when TTT of Mailbox-30 matches with CYCTR . CCR is embedded into the first data byte of the time reference message { Data0[7:6], CCR[5:0] } .

- Setting Tx-Trigger Time

The Tx-Trigger Time(TTT) must be set in ascending order shown below, and the difference between them has to satisfy the following expressions. TEW in the following expressions is the register value.

TTT (Mailbox-24) < TTT (Mailbox-25) < TTT (Mailbox-26) < TTT (Mailbox-27) < TTT (Mailbox-28) < TTT (Mailbox-29) < TTT (Mailbox-30)

and

TTT (Mailbox-i) – TTT (Mailbox- i-1) > TEW + the maximum frame length + 9

TTT (Mailbox-24) to TTT (Mailbox-29) correspond to Time_Marks, and TTT (Mailbox-30) corresponds to Time_Ref showing the length of a basic cycle, respectively when working as potential time master.

The above limitation is not applied to mailboxes which are not set as time-triggered transmission.

Important: Because of limitation on setting Tx-Trigger Time, only one Mailbox can be assigned to one time window.

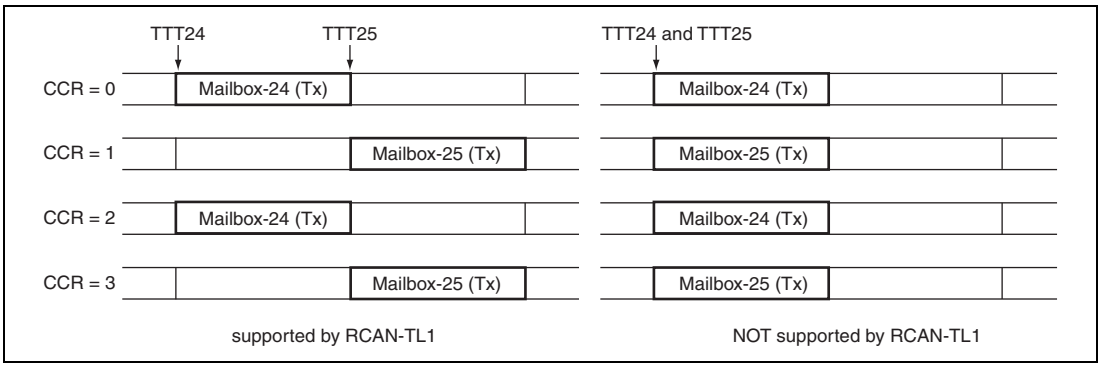


Figure 20.18 Limitation on Tx-Trigger Time

The value of TCMR2 as Watch_Trigger has to be larger than TTT(Mailbox-30), which shows the length of a basic cycle.

Figures 20.19 and 20.20 show examples of configurations for (Potential) Time Master and Time Slave. “L” in diagrams shows the length in time of the time reference messages.

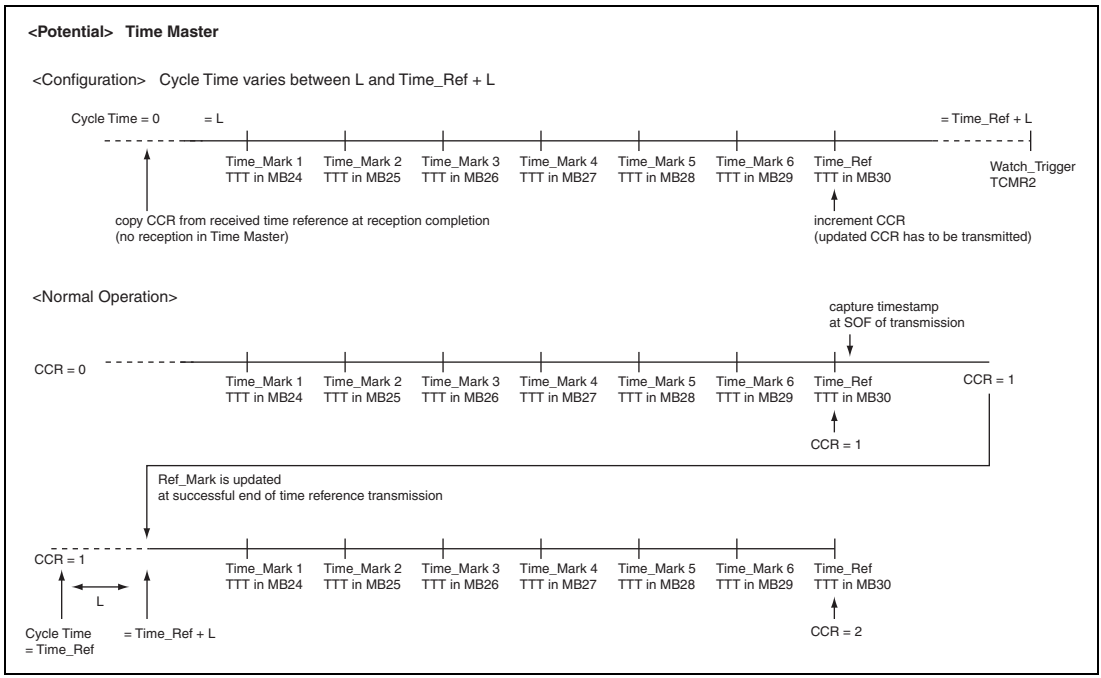


Figure 20.19 (Potential) Time Master

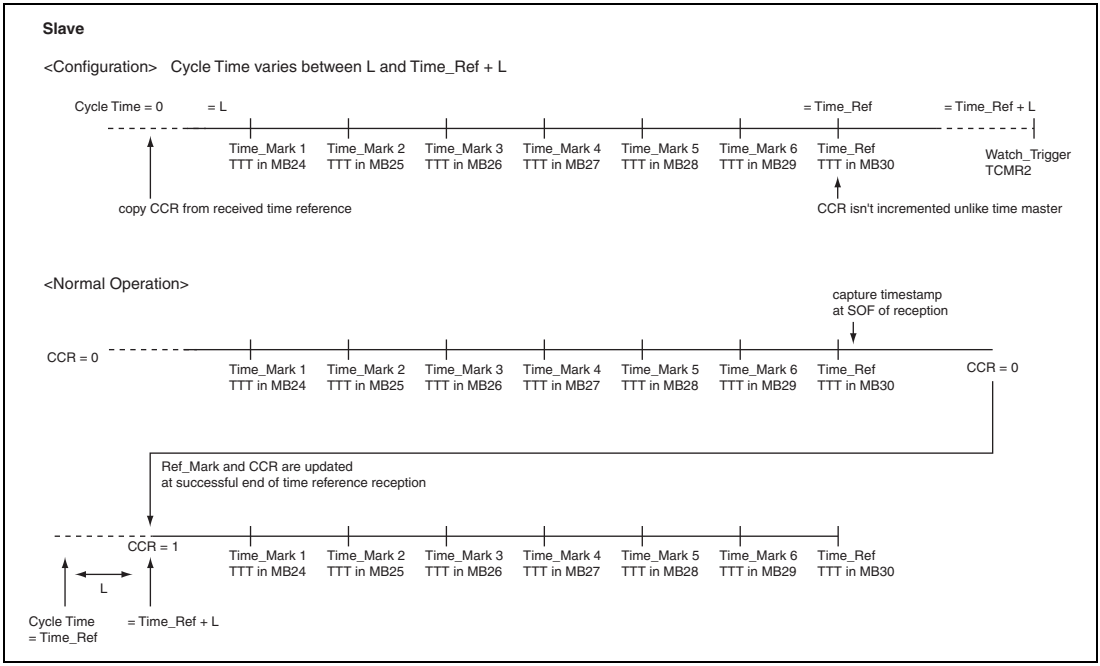


Figure 20.20 Time Slave

- Function to be implemented by software

  Some of the TTCAN functions need to be implemented in software. The main details are reported hereafter. Please refer to ISO-11898-4 for more details.

  — Change from Init_Watch_Trigger to Watch_Trigger

  RCAN-TL1 offers the two registers TCMR0 and TCMR2 as H/W support for Init_Watch_Trigger and Watch_Trigger respectively. The SW is requested to enable TCMR0 and disable TCMR2 up to the first reference message is detected on the CAN Bus and then disable TCMR0 and enable TCMR2.- Schedule Synchronization state machine.

  Only reception of Next_is_Gap interrupt is supported. The application needs to take care of stopping all transmission at the end of the current basic cycle by setting the related TXCR flags.Master-Slave Mode control.

  Only automatic cycle time synchronization and CCR increment is supported.

  — Message status count

  Software has to count scheduling errors for periodic messages in exclusive windows.

- Message Transmission Request for Time Triggered communication

When the Time Triggered mode is used communications must fulfils the ISO11898-4 requirements.

The following procedure should be used.

— Send RCAN-TL1 to reset or halt mode
— Set TCMR0 to the Init_Watch_Trigger (0xFFFF)
— Enable TCMR0 compare match setting bit 10 of TTCR0
— Set TCMR2 to the specified Watch_Trigger value
— Keep TCMR2 compare match disabled by keeping cleared the bit 12 of TTCR0
— Set CMAX to the requested value (different from 111 bin)
— Set TEW to the requested value
— Configure the necessary Mailboxes for Time Trigger transmission and reception
— Set LAFM for the 3 LSBs of Mailbox 31
— Configure MCR, BCR1 and BCR0 to the requested values
— If working as a potential time master:
  - Set RFTROFF to the requested Init_Ref_Offset value
  - Set TXPR for Mailbox 30
  - Write H'4000 into TTTSEL
— Enable the TCNTR timer through the bit 15 of TTCR0
— Move to Transmission_Reception mode
— Wait for the reception or transmission of a valid reference message or for TCMR0 match
— If the local time reaches the value of TCMR0 the Init_Watch_Trigger is reached and the application needs to set TXCR for Mailbox 30 and start again
— If the reference message is transmitted (TXACK[30] is set) set RFTROFF to zero
— If a valid reference message is received (RXPR[31] is set) then:
  - If 3 LSBs of ID of Mailbox 31 have high priority than the 3 LSBs of Mailbox 30 (if working as potential time master) keep RFTROFF to Init_Ref_Offset
  - If 3 LSBs of ID of Mailbox 31 have lower priority than the 3 LSBs of Mailbox 30 (if working as potential time master) decrement by 1 the value in RFTROFF
— Disable TCMR0 compare match by clearing bit 10 of TTCR0
— Enable TCMR2 compare match by setting bit 12 of TTCR0
— Only after two reference messages have been detected on the CAN Bus (transmitted or received) can the application set TXPR for the other Time Triggered Mailboxes.

If, at any time, a reference message cannot be detected on the CAN Bus, and the cycle time CYCTR reaches TCMR2, RCAN-TL1 automatically aborts all pending transmissions (including the Reference Message).

The following is the sequence to request further transmission in Time Triggered mode.

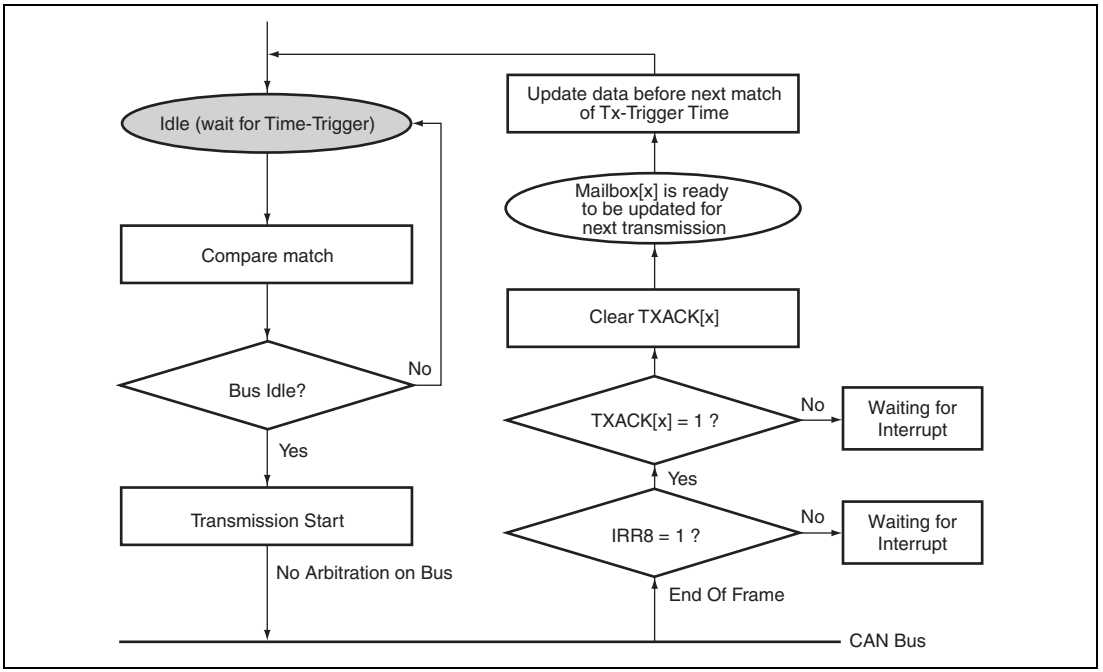


Figure 20.21 Message transmission request

S/W has to ensure that a message is updated before a Tx trigger for transmission occurs.

When the CYCTR reaches to TTT (Tx-Trigger Time) of a Mailbox and CCR matches with the programmed cycle for transmission, RCAN-TL1 immediately transfers the message into the Tx buffer. At this point, RCAN-TL1 will attempt a transmission within the specified Time Enable Window. If RCAN-TL1 misses this time slot, it will suspend the transmission request up to the next Tx Trigger, keeping the corresponding TXPR bit set to '1' if the transmission is periodic (Mailbox-24 to 30). There are three factors that may cause RCAN-TL1 to miss the time slot –

1. The CAN bus currently used
2. An error on the CAN bus during the time triggered message transmission
3. Arbitration loss during the time triggered message transmission

In case of Merged Arbitrating Window the slot for transmission goes from the Tx_Trig of the Mailbox opening the Window (TTW = 10 bin) to the end to the TEW of the Mailbox closing the Window (TTW = 11 bin).The TXPR can be modified at any time. RCAN-TL1 ensures the transmission of Time Triggered messages is always scheduled correctly. However, in order to guarantee the correct schedule, there are some important rules that are :

— TTT (Tx Trigger Time) can be modified during configuration mode.
— TTT cannot be set outside the range of Time_Ref, which specifies the length of basic cycle. This could cause a scheduling problem.
— TXPR is not automatically cleared for periodic transmission. If a periodic transmission needs to be cancelled, the corresponding TXCR bit needs to be set by the application.

- Example of Time Triggered System

The following diagram shows a simple example of how time trigger system works using RCAN-TL1 in time slave mode.

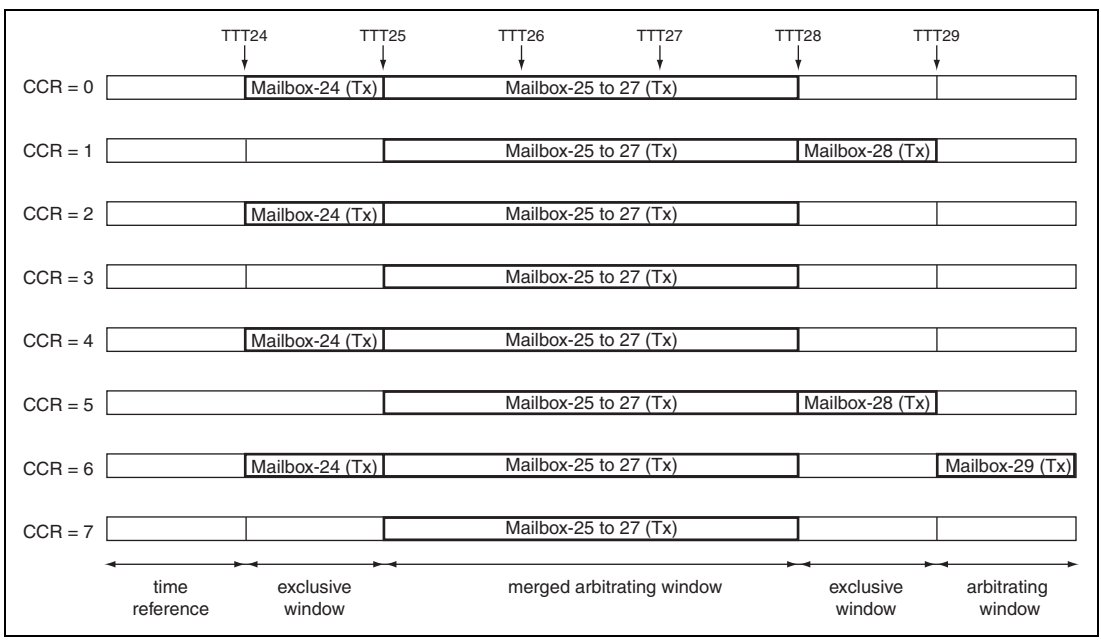


Figure 20.22 Example of Time trigger system as Time Slave

The following settings were used in the above example:

	rep_factor (register)	Offset	TTW[1:0]	MBC[2:0]
Mailbox-24	3'b001	6'b000000	2'b00	3'b000
Mailbox-25	3'b000	6'b000000	2'b10	3'b000
Mailbox-26	3'b000	6'b000000	2'b10	3'b000
Mailbox-27	3'b000	6'b000000	2'b11	3'b000
Mailbox-28	3'b010	6'b000001	2'b00	3'b000
Mailbox-29	3'b011	6'b000110	2'b01	3'b000
Mailbox-30	—	—	—	3'b111
Mailbox-31	—	—	—	3'b011

CMAX = 3'b011, TXPR[30] = 0

During merged arbitrating window, request by time-triggered transmission is served in the way of FCFS (First Come First Served). For example, if Mailbox-25 cannot be transmitted between Tx-Trigger Time 25 (TTT25) and TTT26, Mailbox-25 has higher priority than Mailbox-26 between TTT26 and 28.

MBC needs to be set into 3'b111, in order to disable time-triggered transmission. If RCAN-TL1 is Time Master, MBC[30] has to be 3'b000 and time reference window is automatically recognized as arbitrating window.

- Timer Operation

Figure 20.23 shows the timing diagram of the timer. By setting Tx-Trigger Time = n, time trigger transmission starts between CYCTR = n + 2 and
CYCTR = n + 3.

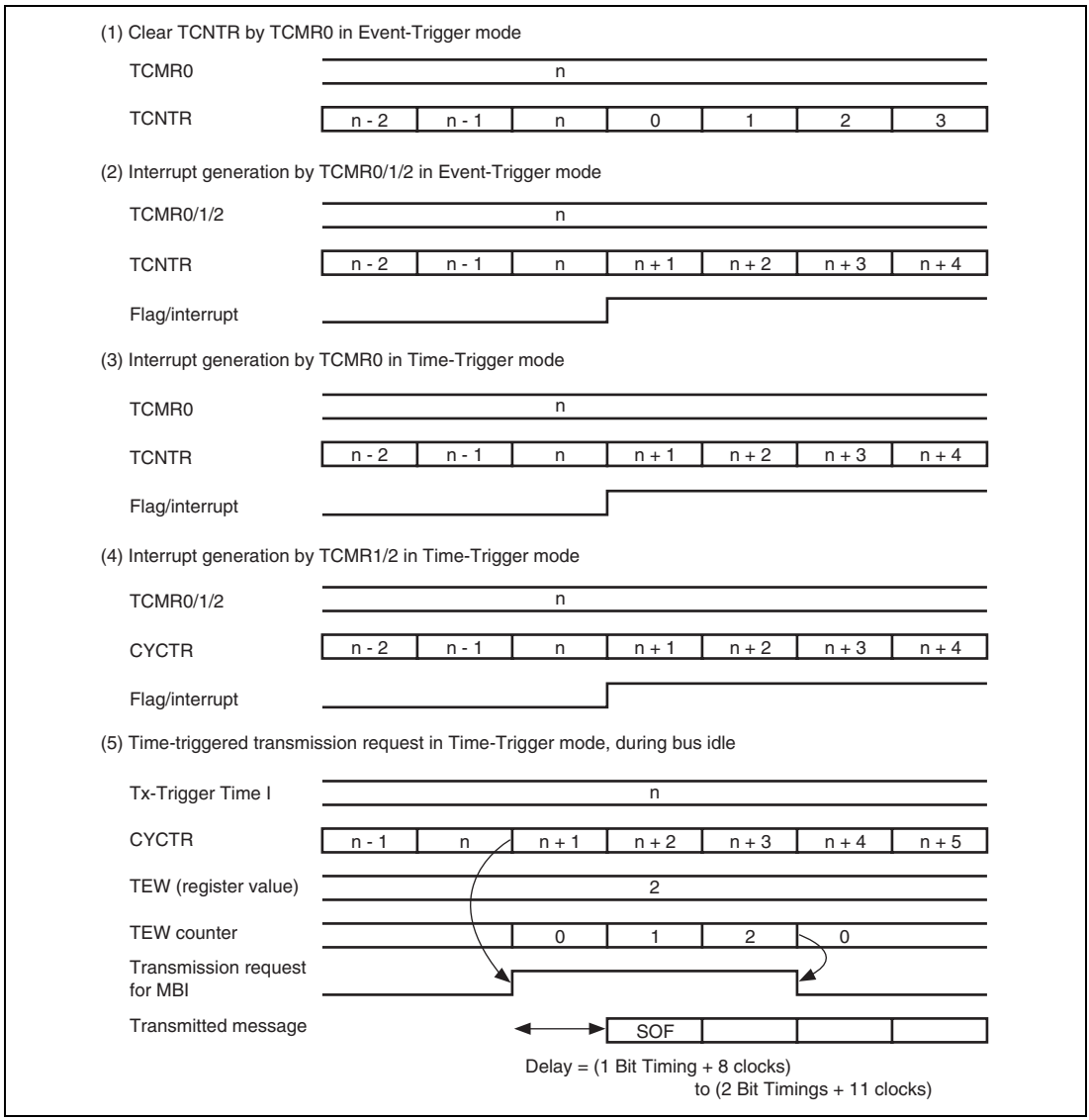


Figure 20.23 Timing Diagram of Timer

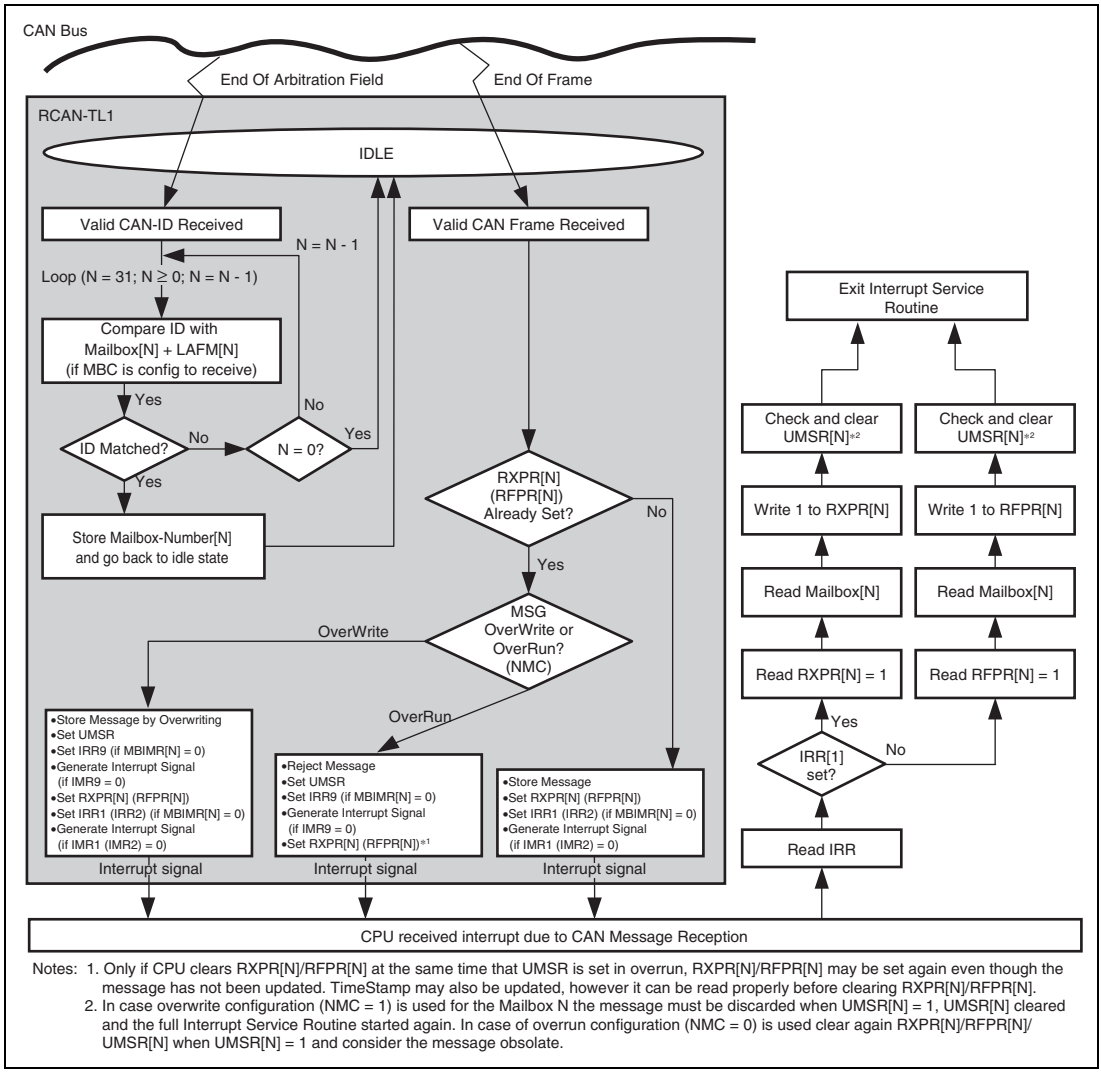
During merged arbitrating window, event-trigger transmission is served after completion of time-triggered transmission. For example, If transmission of Mailbox-25 is completed and CYCTR doesn't reach TTT26, event-trigger transmission starts based on message transmission priority specified by MCR2. TXPR of time-triggered transmission is not cleared after transmission completion, however, that of event-triggered transmission is cleared.

Note: that in the case that the TXPR is not set for the Mailbox which is assigned to close the Merged Arbitrating Window (MAW), then the MAW will still be closed (at the end of the TEW following the TTT of the assigned Mailbox.

Please refer to Table Roles of Mailboxes in section 20.3.2, Mailbox Structure.

20.4.4 Message Receive Sequence

The diagram below shows the message receive sequence.



Notes: 1. Only if CPU clears RXPR[N]/RFPR[N] at the same time that UMSR is set in overrun, RXPR[N]/RFPR[N] may be set again even though the message has not been updated. TimeStamp may also be updated, however it can be read properly before clearing RXPR[N]/RFPR[N].
2. In case overwrite configuration (NMC = 1) is used for the Mailbox N the message must be discarded when UMSR[N] = 1, UMSR[N] cleared and the full Interrupt Service Routine started again. In case of overrun configuration (NMC = 0) is used clear again RXPR[N]/RFPR[N]/ UMSR[N] when UMSR[N] = 1 and consider the message obsolate.

Figure 20.24 Message receive sequence

When RCAN-TL1 recognises the end of the Arbitration field while receiving a message, it starts comparing the received identifier to the identifiers set in the Mailboxes, starting from Mailbox-31 down to Mailbox-0. It first checks the MBC if it is configured as a receive box, and reads LAFM, and reads the CAN-ID of Mailbox-31 (if configured as receive) to finally compare them to the received ID. If it does not match, the same check takes place at Mailbox-30 (if configured as receive). Once RCAN-TL1 finds a matching identifier, it stores the number of Mailbox-[N] into an internal buffer, stops the search, and goes back to idle state, waiting for the EndOfFrame (EOF) to come. When the 6th bit of EOF is notified by the CAN Interface logic, the received message is written or abandoned, depending on the NMC bit. No modification of configuration during communication is allowed. Entering Halt Mode is one of ways to modify configuration. If it is written into the corresponding Mailbox, including the CAN-ID, i.e., there is a possibility that the CAN-ID is overwritten by a different CAN-ID of the received message due to the LAFM used. This also implies that, if the identifier of a received message matches to ID + LAFM of 2 or more Mailboxes, the higher numbered Mailbox will always store the relevant messages and the lower numbered Mailbox will never receive messages. Therefore, the settings of the identifiers and LAFMs need to be carefully selected.

With regards to the reception of data and remote frames described in the above flow diagram the clearing of the UMSR flag after the reading of IRR is to detect situations where a message is overwritten by a new incoming message stored in the same mailbox (if its NMC = 1) while the interrupt service routine is running. If during the final check of UMSR a overwrite condition is detected the message needs to be discarded and read again.

In case UMSR is set and the Mailbox is configured for overrun (NMC = 0) the message is still valid, however it is obsolete as it is not reflecting the latest message monitored on the CAN Bus.

Please access the full Mailbox content before clearing the related RXPR/RFPR flag.

Please note that in the case a received remote frame is overwritten by a data frame, both the remote frame receive interrupt (IRR2) and data frame received interrupt (IRR1) and also the Receive Flags (RXPR and RFPR) are set. In an analogous way, the overwriting of a data frame by a remote frame, leads to setting both IRR2 and IRR1.

When a message is received and stored into a Mailbox all the fields of the data not received are stored as zero. The same applies when a standard frame is received. The extended identifier part (EXTID[17:0]) is written as zero.

20.4.5 Reconfiguration of Mailbox

When re-configuration of Mailboxes is required, the following procedures should be taken.

- Change configuration of transmit box

  Two cases are possible.

  — Change of ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART

  This change is possible only when MBC = 3'b000. Confirm that the corresponding TXPR is not set. The configuration (except MBC bit) can be changed at any time.

  — Change from transmit to receive configuration (MBC)

  Confirm that the corresponding TXPR is not set. The configuration can be changed only in Halt or reset state. Please note that it might take longer for RCAN-TL1 to transit to halt state if it is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-TL1 will not be able to receive/transmit messages during the Halt state.

  In case RCAN-TL1 is in the Bus Off state the transition to halt state depends on the configuration of the bit 6 of MCR and also bit and 14 of MCR.

- Change configuration (ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART, MBC) of receiver box or Change receiver box to transmitter box

  The configuration can be changed only in Halt Mode.

  RCAN-TL1 will not lose a message if the message is currently on the CAN bus and RCAN-TL1 is a receiver. RCAN-TL1 will be moving into Halt Mode after completing the current reception. Please note that it might take longer if RCAN-TL1 is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-TL1 will not be able to receive/transmit messages during the Halt Mode.

  In case RCAN-TL1 is in the Bus Off state the transition to halt mode depends on the configuration of the bit 6 and 14 of MCR.

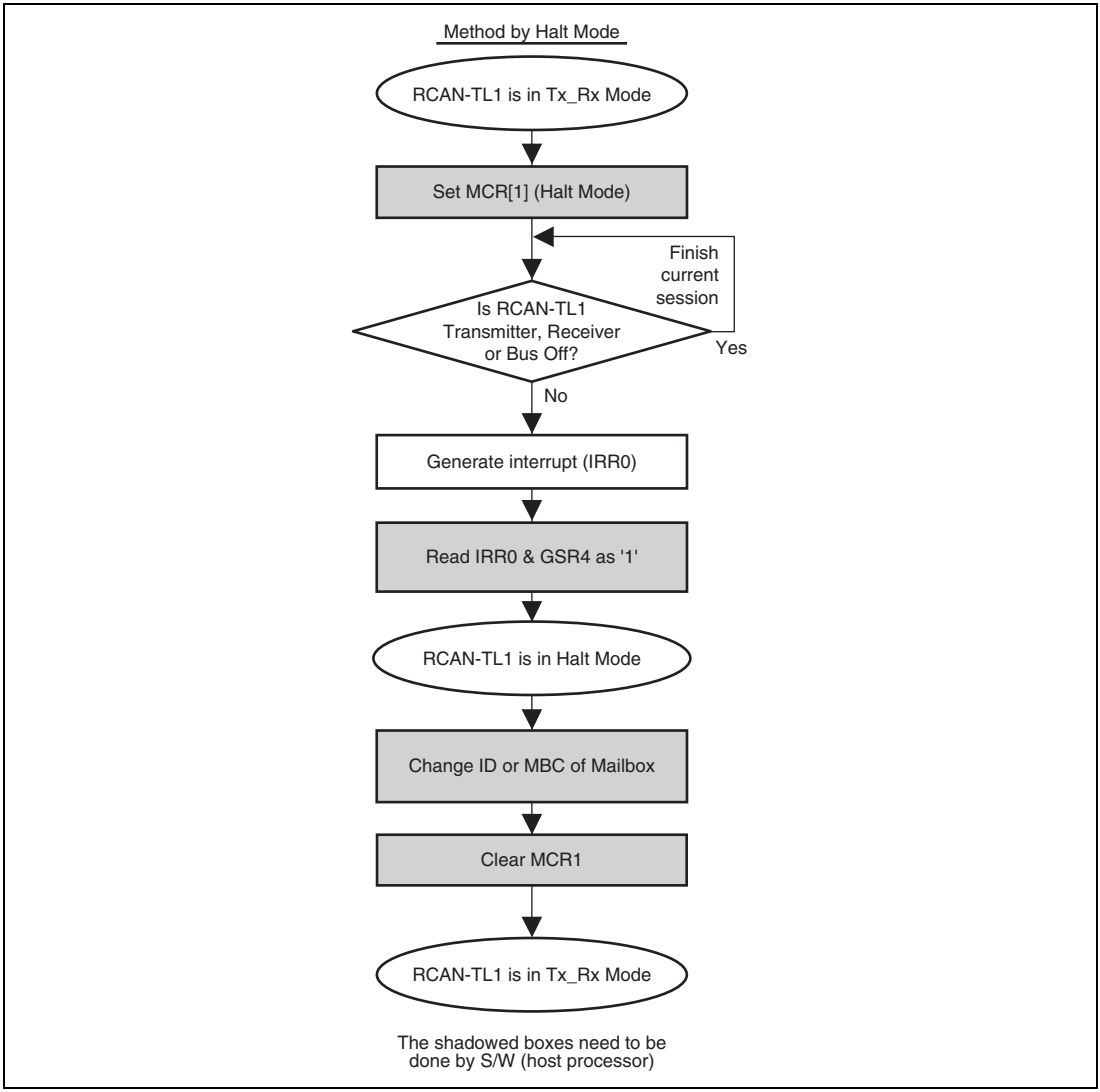


Figure 20.25 Change ID of receive box or Change receive box to transmit box

20.5 Interrupt Sources

Table 20.2 lists the RCAN-TL1 interrupt sources. These sources can be masked. Masking is implemented using the mailbox interrupt mask registers (MBIMR) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, see section 7, Interrupt Controller (INTC).

Table 20.2 RCAN-TL1-n*1 Interrupt Sources

Interrupt	Description	Interrupt Flag	DMAC Activation
ERSn*1	Error Passive Mode (TEC ≥ 128 or REC ≥ 128)	IRR5	Not possible
	Bus Off (TEC ≥ 256)/Bus Off recovery	IRR6	
	Error warning (TEC ≥ 96)	IRR3	
	Error warning (REC ≥ 96)	IRR4	
OVRn*1	Reset/halt/CAN sleep transition	IRR0	
	Overload frame transmission	IRR7	
	Unread message overwrite (overrun)	IRR9	
	Start of new system matrix	IRR10	
	TCMR2 compare match	IRR11	
	Bus activity while in sleep mode	IRR12	
	Timer overrun/Next_is_Gap reception/message error	IRR13	
	TCMR0 compare match	IRR14	
	TCMR1 compare match	IRR15	
RM0n*1*2, RM1n*1*2	Data frame reception	IRR1*3	Possible*4
	Remote frame reception	IRR2*3	
SLEn*1	Message transmission/transmission disabled (slot empty)	IRR8	Not possible

Notes: 1. n = 0, 1

2. RM0 is an interrupt generated by the remote request pending flag for mailbox 0 (RFPR0[0]) or the data frame receive flag for mailbox 0 (RXPR0[0]). RM1 is an interrupt generated by the remote request pending flag for mailbox n (RFPR0[n]) or the data frame receive flag for mailbox n (RXPR0[n]) (n = 1 to 31).
3. IRR1 is a data frame received interrupt flag for mailboxes 0 to 31, and IRR2 is a remote frame request interrupt flag for mailboxes 0 to 31.
4. The DMAC is activated only by an RM0n interrupt.

20.6 DMAC Interface

The DMAC can be activated by the reception of a message in RCAN-TL1 mailbox 0. When DMAC transfer ends after DMAC activation has been set, flags of RXPR0 and RFPR0 are cleared automatically. An interrupt request due to a receive interrupt from the RCAN-TL1 cannot be sent to the CPU in this case. Figure 20.26 shows a DMAC transfer flowchart.

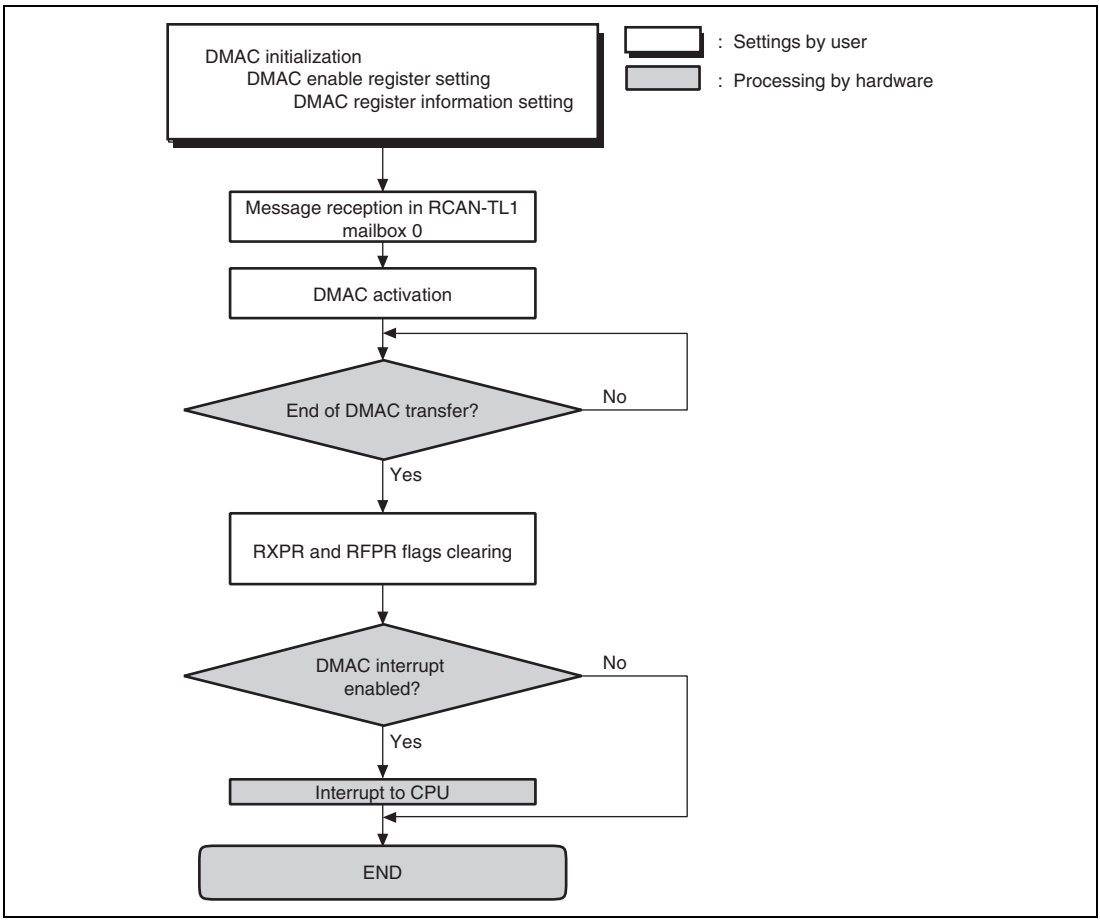


Figure 20.26 DMAC Transfer Flowchart

20.7 CAN Bus Interface

A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Renesas HA13721 transceiver IC and its compatible products are recommended. As the CRx and CTx pins use 3 V, an external level shifter is necessary. Figure 20.27 shows a sample connection diagram.

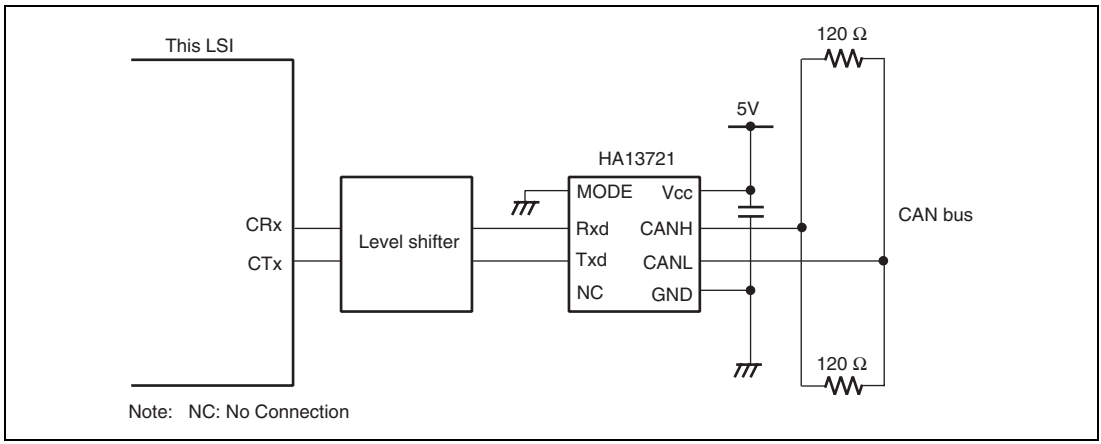


Figure 20.27 High-Speed CAN Interface Using HA13721

20.8 Setting I/O Ports for RCAN-TL1

The I/O ports for the RCAN-TL1 must be specified before or during the configuration mode. For details on the settings of I/O ports, see section 27, Pin Function Controller (PFC). Two methods are available using two channels of the RCAN-TL1 in this LSI.

- Using RCAN-TL1 as a 2-channel module (channels 0 and 1)
  Each channel has 32 Mailboxes.
- Using RCAN-TL1 as a 1-channel module (channels 0 and 1 functioning as a single channel)

When the second method is used, see section 20.9.1, Notes on Port Setting for Multiple Channels Used as Single Channel.

Figures 20.28 and 20.29 show connection examples for individual port settings.

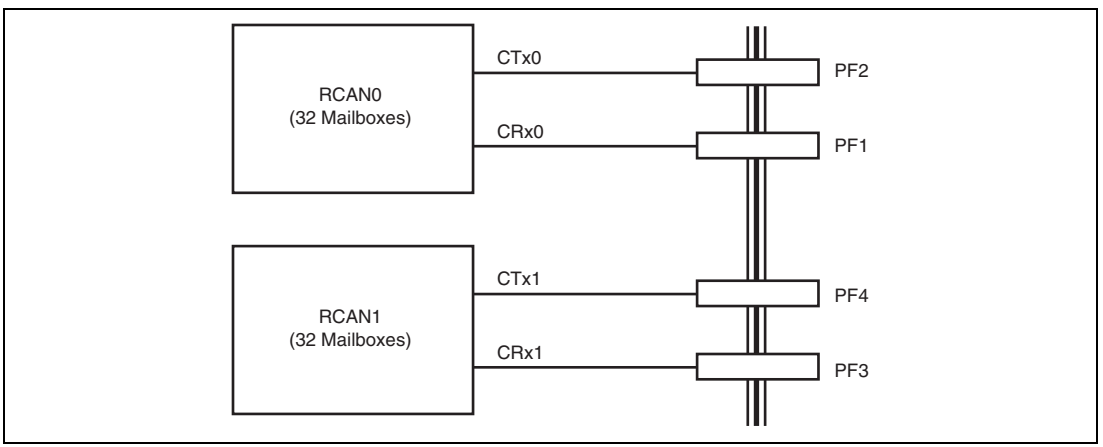


Figure 20.28 Connection Example when Using RCAN-TL1 as 2-Channel Module (32 Mailboxes × 2 Channels)

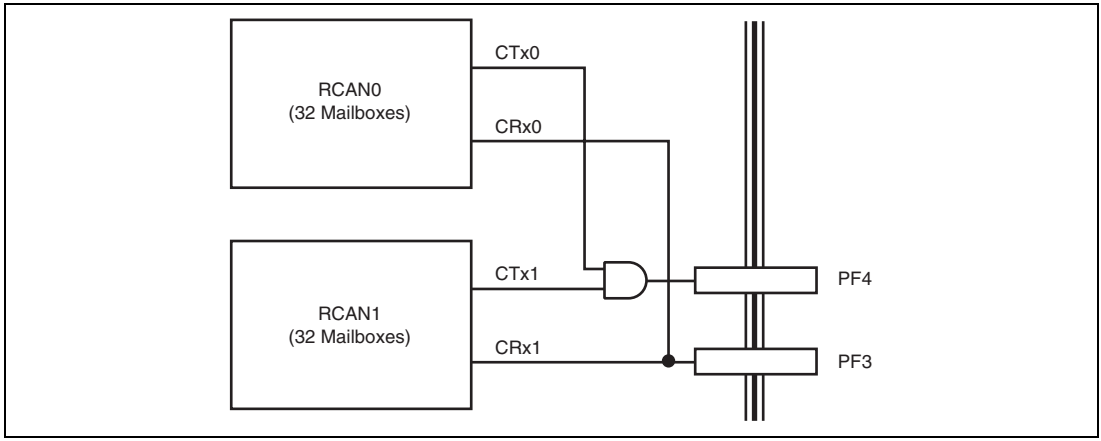


Figure 20.29 Connection Example when Using RCAN-TL1 as 1-Channel Module (64 Mailboxes × 1 Channel)

20.9 Usage Notes

20.9.1 Notes on Port Setting for Multiple Channels Used as Single Channel

The RCAN-TL1 in this LSI has two channels and some of these channels can be used as a single channel. When using multiple channels as a single channel, keep the following in mind.

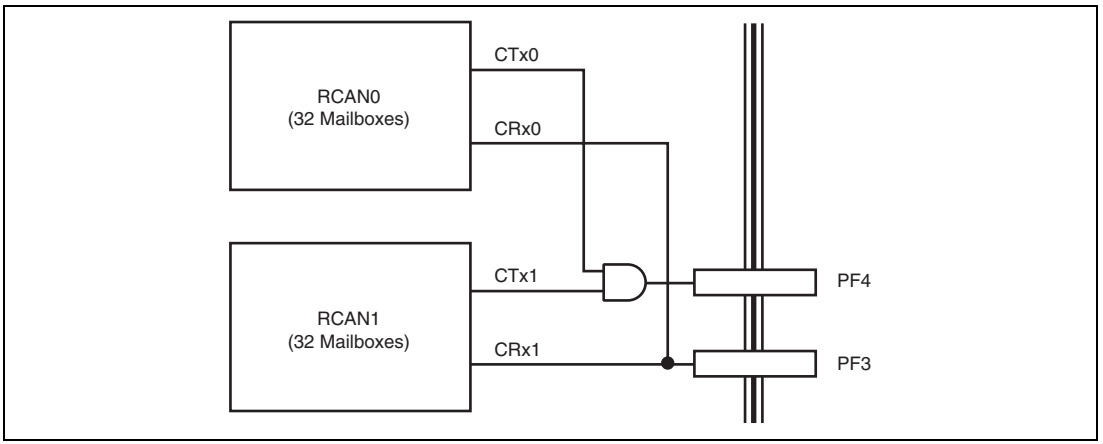


Figure 20.30 Connection Example when Using RCAN-TL1 as 1-Channel Module (64 Mailboxes × 1 Channel)

1. No ACK error is detected even when any other nodes are not connected to the CAN bus. This occurs when channel 1 transmits an ACK in the ACK field in response to a message channel 0 has transmitted.

   Channel 1 receives a message which channel 0 has transmitted on the CAN bus and then transmits an ACK in the ACK field. After that, channel 0 receives the ACK.

   To avoid this, make channel 1 which is not currently used for transmission the listen-only mode (TST[2:0] = B'001) or the reset state (MCR0 = 1). With this setting, only a channel which transmits a message transmits an ACK.
2. Internal arbitration for channels 0 and 1 is independently controlled to determine the order of transmission.

   Although the internal arbitration is performed on 31 Mailboxes at a time, it is not performed on 64 Mailboxes at a time even though multiple channels function as a single channel.
3. Do not set the same transmission message ID in both channels 0 and 1.

   Two messages may be transmitted from the two channels after arbitration on the CAN bus.

Section 21 A/D Converter (ADC)

This LSI includes a 10-bit successive-approximation A/D converter allowing selection of up to eight analog input channels.

21.1 Features

- Resolution: 10 bits
- Input channels: 8
- Minimum conversion time: 3.9 μs per channel
- Absolute accuracy: ±4 LSB
- Operating modes: 3
  - Single mode: A/D conversion on one channel
  - Multi mode: A/D conversion on one to four channels or on one to eight channels
  - Scan mode: Continuous A/D conversion on one to four channels or on one to eight channels
- Data registers: 8

  Conversion results are held in a 16-bit data register for each channel
- Sample-and-hold function
- A/D conversion start methods: 3
  - Software
  - Conversion start trigger from multi-function timer pulse unit 2 (MTU2)
  - External trigger signal
- Interrupt source

  An A/D conversion end interrupt (ADI) request can be generated on completion of A/D conversion.
- Module standby mode can be set

Figure 21.1 shows a block diagram of the A/D converter.

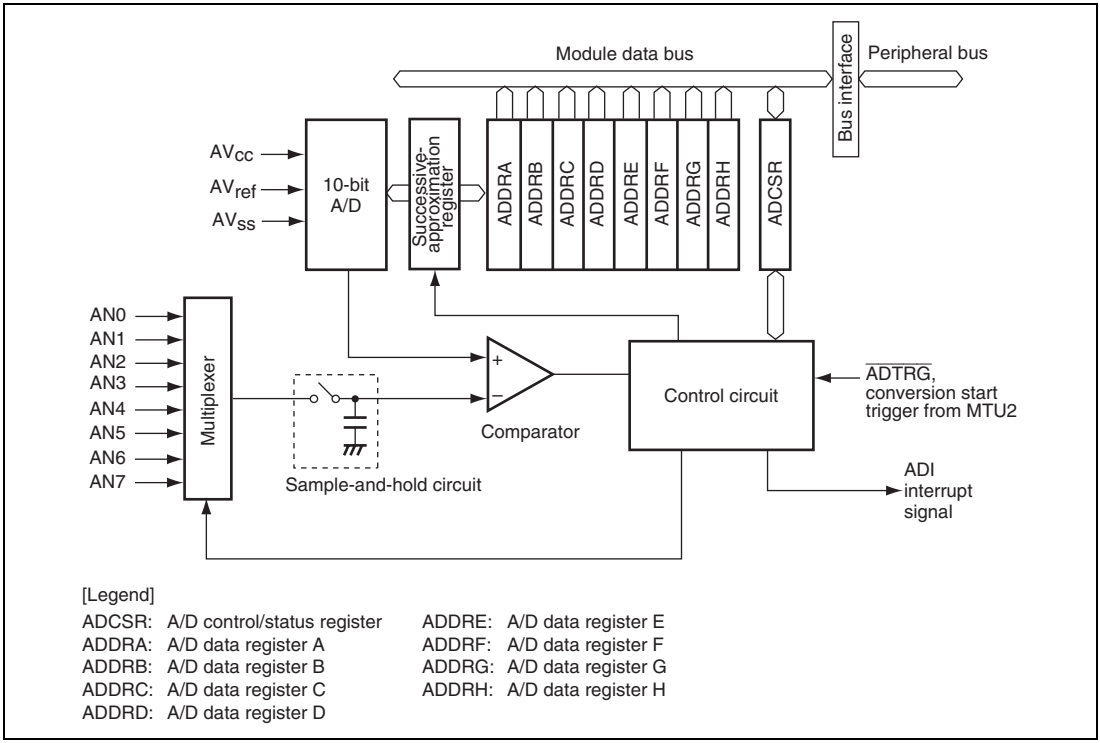


Figure 21.1 Block Diagram of A/D Converter

21.2 Input/Output Pins

Table 21.1 summarizes the A/D converter's input pins.

Table 21.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVcc	Input	Analog power supply pin
Analog ground pin	AVss	Input	Analog ground pin and A/D conversion reference ground
Analog reference voltage pin	AVref	Input	A/D converter reference voltage pin
Analog input pin 0	AN0	Input	Analog input
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input to start A/D conversion

21.3 Register Descriptions

The A/D converter has the following registers.

Table 21.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
A/D data register A	ADDRA	R	H'0000	H'FFFE4800	16
A/D data register B	ADDRB	R	H'0000	H'FFFE4802	16
A/D data register C	ADDRC	R	H'0000	H'FFFE4804	16
A/D data register D	ADDRD	R	H'0000	H'FFFE4806	16
A/D data register E	ADDRE	R	H'0000	H'FFFE4808	16
A/D data register F	ADDRF	R	H'0000	H'FFFE480A	16
A/D data register G	ADDRG	R	H'0000	H'FFFE480C	16
A/D data register H	ADDRH	R	H'0000	H'FFFE480E	16
A/D control/status register	ADCSR	R/W	H'0040	H'FFFE4820	16

21.3.1 A/D Data Registers A to H (ADDRA to ADDRH)

The sixteen A/D data registers, ADDRA to ADDRH, 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 ADDR corresponding to the selected channel. The 10 bits of the result are stored in the upper bits (bits 15 to 6) of ADDR. Bits 5 to 0 of ADDR are reserved bits that are always read as 0.

Access to ADDR in 8-bit units is prohibited. ADDR must always be accessed in 16-bit units.

Table 21.3 indicates the pairings of analog input channels and ADDR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 6		All 0	R	Bit data (10 bits)
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 21.3 Analog Input Channels and ADDR

Analog Input Channel	A/D Data Register where Conversion Result is Stored
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD
AN4	ADDRE
AN5	ADDRF
AN6	ADDRG
AN7	ADDRH

21.3.2 A/D Control/Status Register (ADCSR)

ADCSR is a 16-bit readable/writable register that selects the mode, controls the A/D converter, and enables or disables starting of A/D conversion by external trigger input.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	-	TRGS[3:0]				CKS[1:0]		MDS[2:0]			CH[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15	ADF	0	R/(W)*[1]	A/D End Flag Status flag indicating 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] • A/D conversion ends in single mode • A/D conversion ends for the selected channels in multi mode • A/D conversion ends for the selected channels in scan mode
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

Bit	Bit Name	Initial Value	R/W	Description
13	ADST	0	R/W	A/D Start Starts or stops A/D conversion. This bit remains set to 1 during A/D conversion. 0: A/D conversion is stopped 1: Single mode: A/D conversion starts. This bit is automatically cleared to 0 when A/D conversion ends on the selected channel. Multi mode: A/D conversion starts. This bit is automatically cleared to 0 when A/D conversion is completed cycling through the selected channels. Scan mode: A/D conversion starts. A/D conversion is continuously performed until this bit is cleared to 0 by software, by a power-on reset as well as by a transition to deep standby mode, software standby mode or module standby mode.
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11 to 8	TRGS[3:0]	0000	R/W	Timer Trigger Select These bits enable or disable starting of A/D conversion by a trigger signal. 0000: Start of A/D conversion by external trigger input is disabled 0001: A/D conversion is started by conversion trigger TRGAN from MTU2 0010: A/D conversion is started by conversion trigger TRG0N from MTU2 0011: A/D conversion is started by conversion trigger TRG4AN from MTU2 0100: A/D conversion is started by conversion trigger TRG4BN from MTU2 1001: A/D conversion is started by $\overline{\text{ADTRG}}$ Other than above: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CKS[1:0]	01	R/W	Clock Select These bits select the A/D conversion time*2. Set the A/D conversion time while A/D conversion is halted (ADST = 0). 00: Conversion time = 138 t_{pcyc}*3 (maximum) 01: Conversion time = 274 t_{pcyc}*3 (maximum) 10: Conversion time = 546 t_{pcyc}*3 (maximum) 11: Setting prohibited
5 to 3	MDS[2:0]	000	R/W	Multi-scan Mode These bits select the operating mode for A/D conversion. 0xx: Single mode 100: Multi mode: A/D conversion on 1 to 4 channels 101: Multi mode: A/D conversion on 1 to 8 channels 110: Scan mode: A/D conversion on 1 to 4 channels 111: Scan mode: A/D conversion on 1 to 8 channels
2 to 0	CH[2:0]	000	R/W	Channel Select These bits and the MDS bits in ADCSR select the analog input channels.

MDS[2:0] = 0xx	MDS[2:0] = 100 or MDS[2:0] = 110	MDS[2:0] = 101 or MDS[2:0] = 111
000: AN0	000: AN0	000: AN0
001: AN1	001: AN0, AN1	001: AN0, AN1
010: AN2	010: AN0 to AN2	010: AN0 to AN2
011: AN3	011: AN0 to AN3	011: AN0 to AN3
100: AN4	100: AN4	100: AN0 to AN4
101: AN5	101: AN4, AN5	101: AN0 to AN5
110: AN6	110: AN4 to AN6	110: AN0 to AN6
111: AN7	111: AN4 to AN7	111: AN0 to AN7

[Legend]

x: Don't care

Notes: 1. The flag can only be cleared by writing 0 to it after reading it as 1. However, in the following cases as well the flag is cleared by writing 0 to it:
(1) When the CPU reads the value of ADF as 1
(2) When ADF is cleared to 0 by the DMAC reading ADDR
(3) When the ADF flag is set to 1 at A/D conversion end
(4) When the CPU writes 0 to ADF

2. Set the A/D conversion time to minimum or more values to meet the absolute accuracy of the A/D conversion characteristics.

3. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.

21.4 Operation

The A/D converter uses the successive-approximation method, and the resolution is 10 bits. It has three operating modes: single mode, multi mode, and scan mode. Switching the operating mode or analog input channels must be done while the ADST bit in ADCSR is 0 to prevent incorrect operation. The ADST bit can be set at the same time as the operating mode or analog input channels are changed.

21.4.1 Single Mode

Single mode should be selected when only A/D conversion on one channel is required.

In single mode, A/D conversion is performed once for the specified one analog input channel, as follows:

1. A/D conversion for the selected channel starts when the ADST bit in ADCSR is set to 1 by software, MTU2, or external trigger input.
2. When A/D conversion is completed, the A/D conversion result is transferred to the A/D data register corresponding to the channel.
3. After A/D conversion has completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit that remains 1 during A/D conversion is automatically cleared to 0 when A/D conversion is completed, and the A/D converter becomes idle.

When the operating 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 to start A/D conversion again. The ADST bit can be set at the same time as the mode or channel selection is switched.

Typical operations when a single channel (AN1) is selected in single mode are described next. Figure 21.2 shows a timing diagram for this example (the bits which are set in this example belong to ADCSR).

1. Single mode is selected, input channel AN1 is selected (CH[2:0] = 001), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the A/D conversion result is transferred into ADDRB. 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 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADF = 1, and then writes 0 to the ADF flag.
6. The routine reads and processes the A/D conversion result (ADDRB).
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.

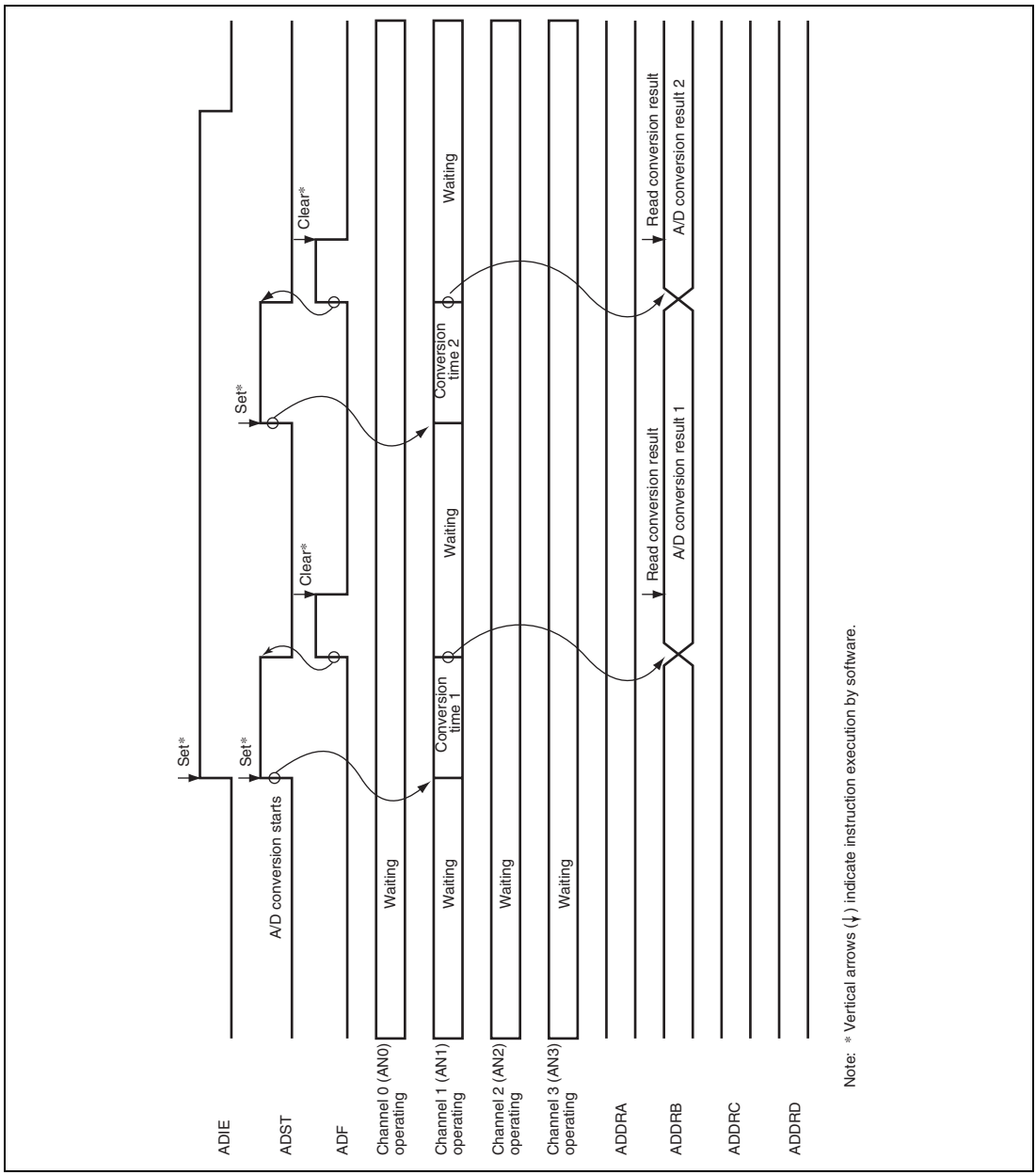


Figure 21.2 Example of A/D Converter Operation (Single Mode, One Channel (AN1) Selected)

21.4.2 Multi Mode

Multi mode should be selected when performing A/D conversion once on one or more channels.

In multi mode, A/D conversion is performed once for a maximum of eight specified analog input channels, as follows:

1. A/D conversion starts from the analog input channel with the lowest number (e.g. AN0, AN1, …, AN3) when the ADST bit in ADCSR is set to 1 by software, MTU2, or external trigger input.
2. When A/D conversion is completed on each channel, the A/D conversion result is sequentially transferred to the A/D data register corresponding to that channel.
3. After A/D conversion on all selected channels has completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit that remains 1 during A/D conversion is automatically cleared to 0 when A/D conversion is completed, and the A/D converter becomes idle. If the ADST bit is cleared to 0 during A/D conversion, A/D conversion is halted and the A/D converter becomes idle. The ADF bit is cleared by reading ADF while ADF = 1, then writing 0 to the ADF bit.

A/D conversion is to be performed once on all the specified channels. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the operating 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 (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 (MDS2 = 1, MDS1 = 0), analog input channels AN0 to AN2 are selected (CH[2:0] = 010), and A/D conversion is started (ADST = 1).
2. A/D conversion of the first channel (AN0) starts. When A/D conversion is completed, the A/D conversion result is transferred into ADDRA.
3. Next, the second channel (AN1) is selected automatically and A/D conversion starts.
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 the ADST bit cleared to 0.
6. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested.

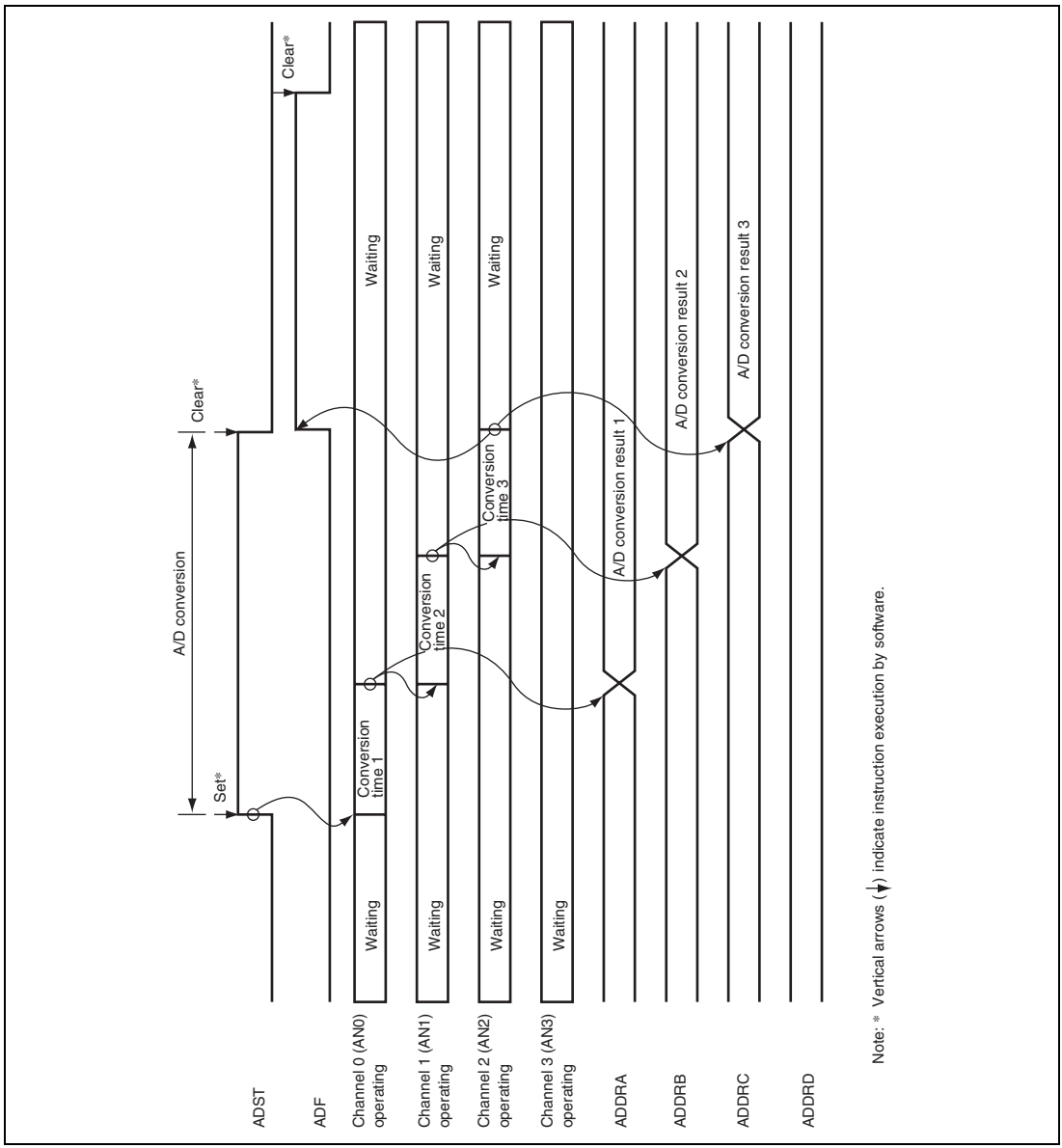


Figure 21.3 Example of A/D Converter Operation (Multi Mode, Three Channels (AN0 to AN2) Selected)

21.4.3 Scan Mode

Scan mode is useful for monitoring analog inputs in a group of one or more channels at all times. In scan mode, A/D conversion is performed sequentially for a maximum of eight specified analog input channels, as follows:

1. A/D conversion starts from the analog input channel with the lowest number (e.g. AN0, AN1, …, AN3) when the ADST bit in ADCSR is set to 1 by software, MTU2, or external trigger input.
2. When A/D conversion is completed on each channel, the A/D conversion result is sequentially transferred to the A/D data register corresponding to that channel.
3. After A/D conversion on all selected channels has completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. The A/D converter starts A/D conversion again from the channel with the lowest number.
4. The ADST bit is not cleared automatically, so steps 2. and 3. are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion halts and the A/D converter becomes idle.

   The ADF bit is cleared by reading ADF while ADF = 1, then writing 0 to the ADF bit.

When the operating 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 (AN0 to AN2) are selected in scan mode are described as follows. Figure 21.4 shows a timing diagram for this example.

1. Scan mode is selected (MDS2 = 1, MDS1 = 1), analog input channels AN0 to AN2 are selected (CH[2:0] = 010), and A/D conversion is started (ADST = 1).
2. A/D conversion of the first channel (AN0) starts. When A/D conversion is completed, the A/D conversion result is transferred into ADDRA.
3. Next, the second channel (AN1) is selected automatically and A/D conversion starts.
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 at this time, an ADI interrupt is requested.

6. The ADST bit is not cleared automatically, so 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 kept to 1. When the ADST bit is cleared to 0, A/D conversion stops. The ADF bit is cleared by reading ADF while ADF = 1, then writing 0 to the ADF bit.

If both the ADF flag and ADIE bit are set to 1 while steps 2. to 4. are repeated, an ADI interrupt is requested at all times. To generate an interrupt on completing conversion of the third channel, clear the ADF bit to 0 after an interrupt is requested.

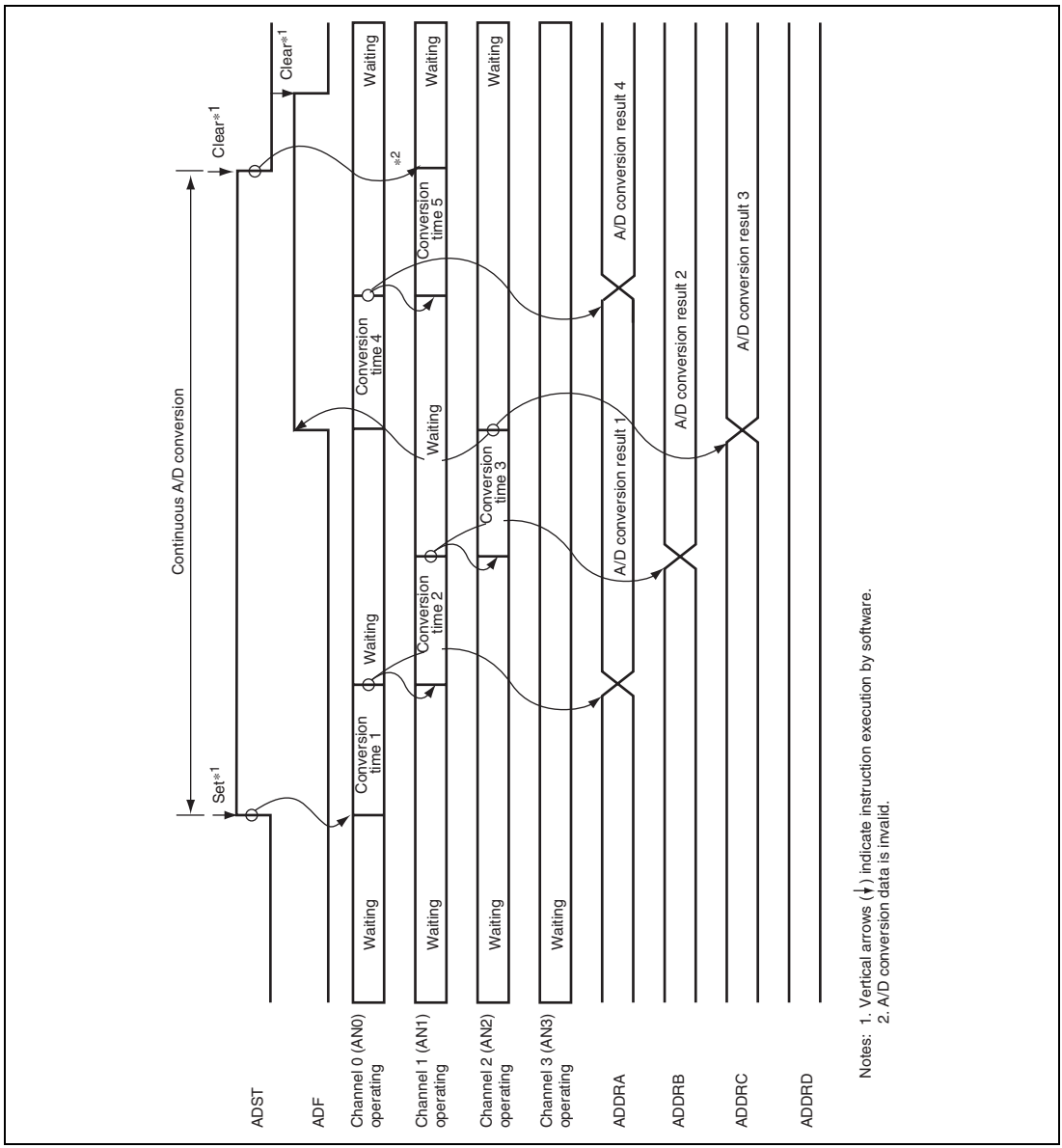


Figure 21.4 Example of A/D Converter Operation (Scan Mode, Three Channels (AN0 to AN2) Selected)

21.4.4 A/D Converter Activation by External Trigger or MTU2

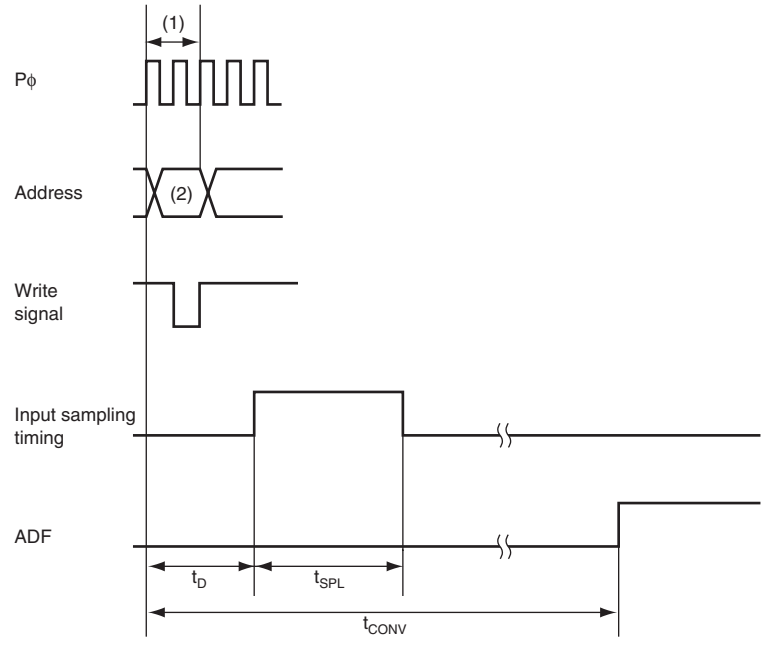
The A/D converter can be independently activated by an external trigger or an A/D conversion request from the MTU2. To activate the A/D converter by an external trigger or the MTU2, set the A/D trigger enable bits (TRGS[3:0]). When an external trigger or an A/D conversion request from the MTU2 is generated with this bit setting, the ADST bit is set to 1 to start A/D conversion. The channel combination is determined by bits CH2 to CH0 in ADCSR. 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.

21.4.5 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 the A/D conversion start delay time (t_D) after the ADST bit in ADCSR is set to 1, then starts conversion. Figure 21.5 shows the A/D conversion timing. Table 21.4 indicates the A/D conversion time.

As indicated in figure 21.5, the A/D conversion time (t_{CONV}) includes t_D and the input sampling time(t_{SPL}). The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 21.4.

In multi mode and scan mode, the values given in table 21.4 apply to the first conversion. In the second and subsequent conversions, time is the values given in table 21.5.



[Legend]
(1): ADCSR write cycle
(2): ADCSR address
t_D: A/D conversion start delay time
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 21.5 A/D Conversion Timing

Table 21.4 A/D Conversion Time (Single Mode)

		CKS1 = 0						CKS1 = 1		
		CKS0 = 0			CKS0 = 1			CKS0 = 0		
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**
A/D conversion start delay time	t_D	11	—	14	19	—	26	35	—	50
Input sampling time	t_{SPL}	—	33	—	—	65	—	—	129	—
A/D conversion time	t_{CONV}	135	—	138	267	—	274	531	—	546

Note: Values in the table are the numbers of t_{pcyc}. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.

Table 21.5 A/D Conversion Time (Multi Mode and Scan Mode)

CKS1	CKS0	Conversion Time (t_{Pcyc})
0	0	128 (constant)
	1	256 (constant)
1	0	512 (constant)

Note: Values in the table are the numbers of t_{pcyc}. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.

21.4.6 External Trigger Input Timing

A/D conversion can also be externally triggered. When the TRGS[3:0] bits in ADCSR are set to B'1001, an external trigger is input to the $\overline{\text{ADTRG}}$ pin. The ADST bit in ADCSR is set to 1 at the falling edge of the $\overline{\text{ADTRG}}$ pin, thus starting A/D conversion. Other operations, regardless of the operating mode, are the same as when the ADST bit has been set to 1 by software. Figure 21.6 shows the timing.

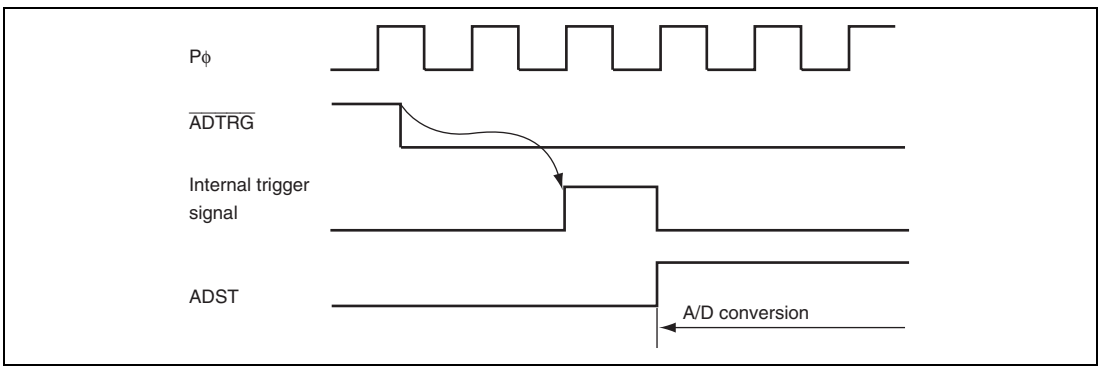


Figure 21.6 External Trigger Input Timing

21.5 Interrupt Sources and DMAC Transfer Request

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. An ADI interrupt request is generated if the ADIE bit is set to 1 when the ADF bit in ADCSR is set to 1 on completion of A/D conversion. Note that the interrupt controller (INTC) can be activated by an ADI interrupt depending on the DMAC setting. In this case, an interrupt is not issued to the CPU. If the setting to activate the DMAC has not been made, an interrupt request is sent to the CPU. Having the converted data read by the DMAC in response to an ADI interrupt enables continuous conversion to be achieved without imposing a load on software.

In single mode, set the DMAC so that DMA transfer initiated by an ADI interrupt is performed only once. In the case of A/D conversion on multiple channels in scan mode or multi mode, setting the DMA transfer count to one causes DMA transfer to finish after transferring only one channel of data. To make the DMAC transfer all conversion data, set the ADDR where A/D conversion data is stored as the transfer source address, the number of converted channels × 2 as the transfer byte count, and continuous operand transfer or non-stop transfer as the DMA transfer condition.

When the DMAC is activated by ADI, the ADF bit in ADCSR is automatically cleared to 0 when data is transferred by the DMAC.

Table 21.6 Relationship between Interrupt Sources and DMAC Transfer Request

Name	Interrupt Source	Interrupt Flag	DMAC Activation
ADI	A/D conversion end	ADF in ADCSR	Possible

21.6 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 21.7. In the figure, the 10-bit A/D converter is illustrated as the 3-bit A/D converter for explanation. Offset error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the minimum (zero voltage) B'0000000000 (000 in the figure) to B'000000001 (001 in the figure)(figure 21.7, item (1)). Full-scale error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from B'1111111110 (110 in the figure) to the maximum B'1111111111 (111 in the figure)(figure 21.7, item (2)). Quantization error is the intrinsic error of the A/D converter and is expressed as 1/2 LSB (figure 21.7, item (3)). Nonlinearity error is the deviation between actual and ideal A/D conversion characteristics between zero voltage and full-scale voltage (figure 21.7, item (4)). Note that it does not include offset, full-scale, or quantization error.

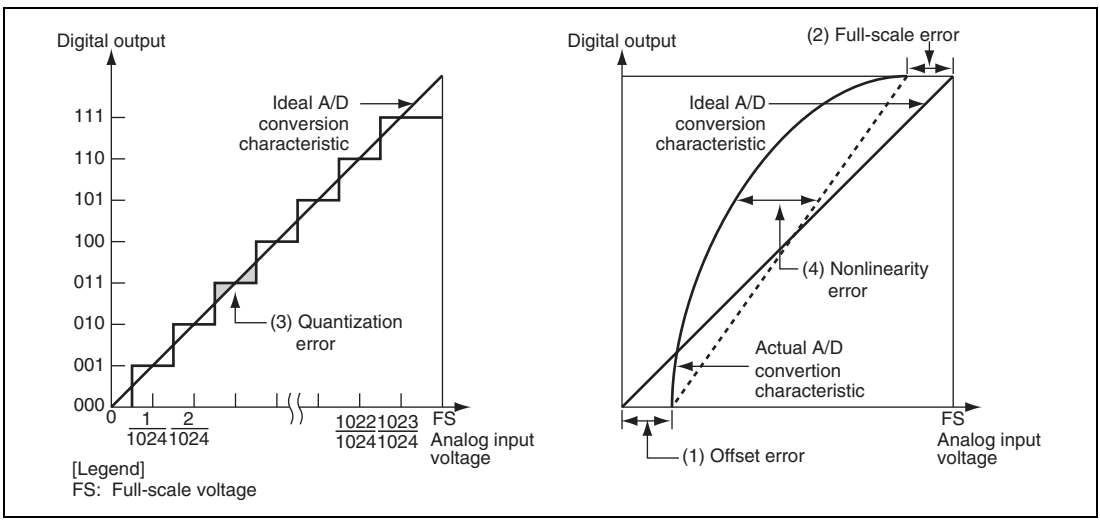


Figure 21.7 Definitions of A/D Conversion Accuracy

21.7 Usage Notes

When using the A/D converter, note the following points.

21.7.1 Module Standby Mode Setting

Operation of the A/D converter can be disabled or enabled using the standby control register. The initial setting is for operation of the A/D converter to be halted. Register access is enabled by clearing module standby mode. For details, see section 30, Power-Down Modes.

21.7.2 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 PVcc − 0.3 V ≤ AVcc ≤ PVcc and AVss = PVss. Do not leave the AVcc and AVss pins open when the A/D converter or D/A converter is not in use and in software standby mode. When not in use, connect AVcc to the power supply (PVcc) and AVss to the ground (PVss).
3. Setting range of AVref input voltage
   Set the reference voltage range of the AVref pin as 3.0 V ≤ AVref ≤ AVcc.

21.7.3 Notes on Board Design

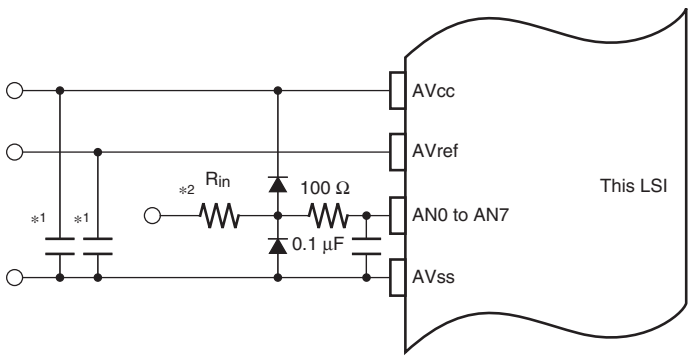
In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Digital circuitry must be isolated from the analog input signals (AN0 to AN7), analog reference voltage (AVref), and analog power supply (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable digital ground (PVss) on the board.

21.7.4 Processing of Analog Input Pins

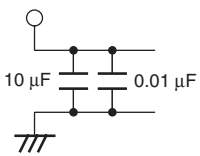
To prevent damage from voltage surges at the analog input pins (AN0 to AN7), connect an input protection circuit like the one shown in figure 21.8. The circuit shown also includes a CR filter to suppress noise. This circuit is shown as an example; the circuit constants should be selected according to actual application conditions.

Figure 21.9 shows an equivalent circuit diagram of the analog input ports and table 21.7 lists the analog input pin specifications.



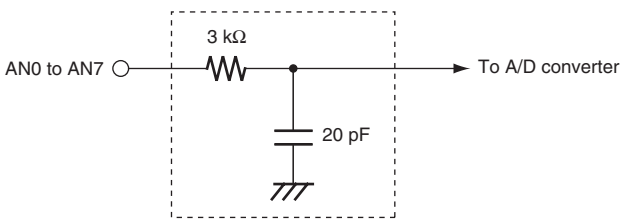
Notes: Values are reference values.

1.



2. R_{in}: Input impedance

Figure 21.8 Example of Analog Input Protection Circuit



Note: Values are reference values.

Figure 21.9 Analog Input Pin Equivalent Circuit

Table 21.7 Analog Input Pin Ratings

Item	Min.	Max.	Unit
Analog input capacitance	—	20	pF
Allowable signal-source impedance	—	5	kΩ

21.7.5 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 5 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficient and it may not be possible to guarantee 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 3 kΩ, 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., 5 mV/μs or greater) (see figure 21.10). When converting a high-speed analog signal, a low-impedance buffer should be inserted.

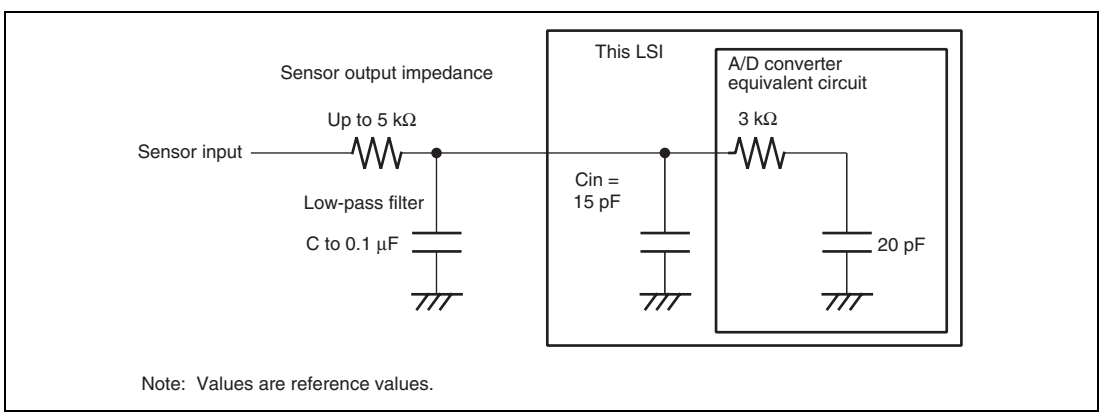


Figure 21.10 Example of Analog Input Circuit

21.7.6 Influences on Absolute Precision

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to connect AVss, etc. to an electrically stable GND.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board (i.e., acting as antennas).

21.7.7 Usage Notes in Scan Mode or Multi Mode

Starting conversion immediately after stopping scan mode or multi mode can cause incorrect conversion results. To continue with conversion in which cases, allow a duration equivalent to the A/D conversion time for one channel to elapse after clearing ADST to 0 before starting conversion (by setting ADST to 1). (The A/D conversion time for one channel differs depending on the peripheral register settings.)

Section 22 D/A Converter (DAC)

22.1 Features

- 8-bit resolution
- Two output channels
- Minimum conversion time of 10 μs (with 20 pF load)
- Output voltage of 0 V to AVref
- D/A output hold function in software standby mode
- Module standby mode can be set

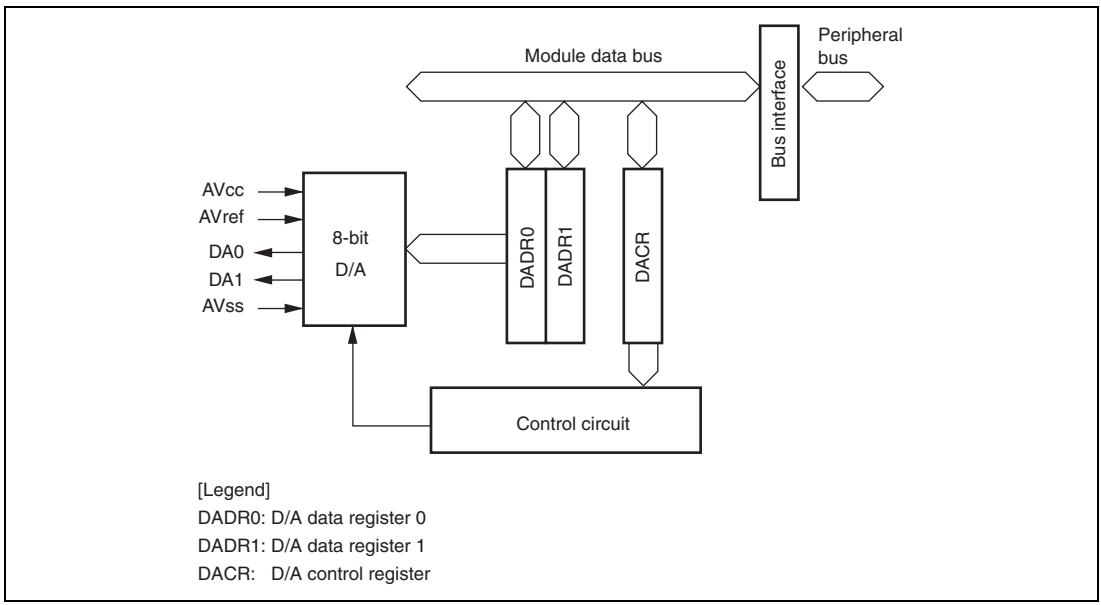


Figure 22.1 Block Diagram of D/A Converter

22.2 Input/Output Pins

Table 22.1 shows the pin configuration of the D/A converter.

Table 22.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVcc	Input	Analog block power supply
Analog ground pin	AVss	Input	Analog block ground
Analog reference voltage pin	AVref	Input	D/A conversion reference voltage
Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output

22.3 Register Descriptions

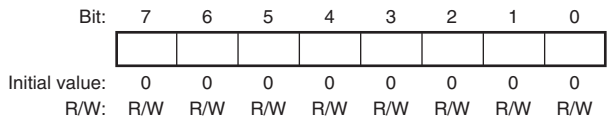
The D/A converter has the following registers.

Table 22.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
D/A data register 0	DADR0	R/W	H'00	H'FFFE4C00	8, 16
D/A data register 1	DADR1	R/W	H'00	H'FFFE4C01	8, 16
D/A control register	DACR	R/W	H'1F	H'FFFE4C02	8, 16

22.3.1 D/A Data Registers 0 and 1 (DADR0 and DADR1)

DADR is an 8-bit readable/writable register that stores data to which D/A conversion is to be performed. Whenever analog output is enabled, the values in DADR are converted and output to the analog output pins.



22.3.2 D/A Control Register (DACR)

DACR is an 8-bit readable/writable register that controls the operation of the D/A converter.

Bit:	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	-	-	-	-	-
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/W	R/W	R/W	-	-	-	-	-

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE1	0	R/W	D/A Output Enable 1 Controls D/A conversion and analog output for channel 1. 0: Analog output of channel 1 (DA1) is disabled 1: D/A conversion of channel 1 is enabled. Analog output of channel 1 (DA1) is enabled.
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output for channel 0. 0: Analog output of channel 0 (DA0) is disabled 1: D/A conversion of channel 0 is enabled. Analog output of channel 0 (DA0) is enabled.
5	DAE	0	R/W	D/A Enable Used together with the DAOE0 and DAOE1 bits to control D/A conversion. Output of conversion results is always controlled by the DAOE0 and DAOE1 bits. For details, see table 22.3. 0: D/A conversion for channels 0 and 1 is controlled independently 1: D/A conversion for channels 0 and 1 is controlled together
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Table 22.3 Control of D/A Conversion

Bit 5	Bit 7	Bit 6	
DAE	**DAOE1**	**DAOE0**	**Description**
0	0	0	D/A conversion is disabled.
		1	D/A conversion of channel 0 is enabled and D/A conversion of channel 1 is disabled.
	1	0	D/A conversion of channel 1 is enabled and D/A conversion of channel 0 is disabled.
		1	D/A conversion of channels 0 and 1 is enabled.
1	0	0	D/A conversion is disabled.
		1	D/A conversion of channels 0 and 1 is enabled.
	1	0	
		1	

22.4 Operation

The D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When the DAOE bit in DACR is set to 1, D/A conversion is enabled and the conversion result is output.

An operation example of D/A conversion on channel 0 is shown below. Figure 22.2 shows the timing of this operation.

1. Write the conversion data to DADR0.
2. Set the DAOE0 bit in DACR to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time t_{DCONV} has elapsed. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is cleared to 0. The output value is expressed by the following formula:

$$\frac{\text{Contents of DADR}}{256} \cdot \text{AVref}$$

3. If DADR0 is written to again, the conversion is immediately started. The conversion result is output after the conversion time t_{DCONV} has elapsed.
4. If the DAOE0 bit is cleared to 0, analog output is disabled.

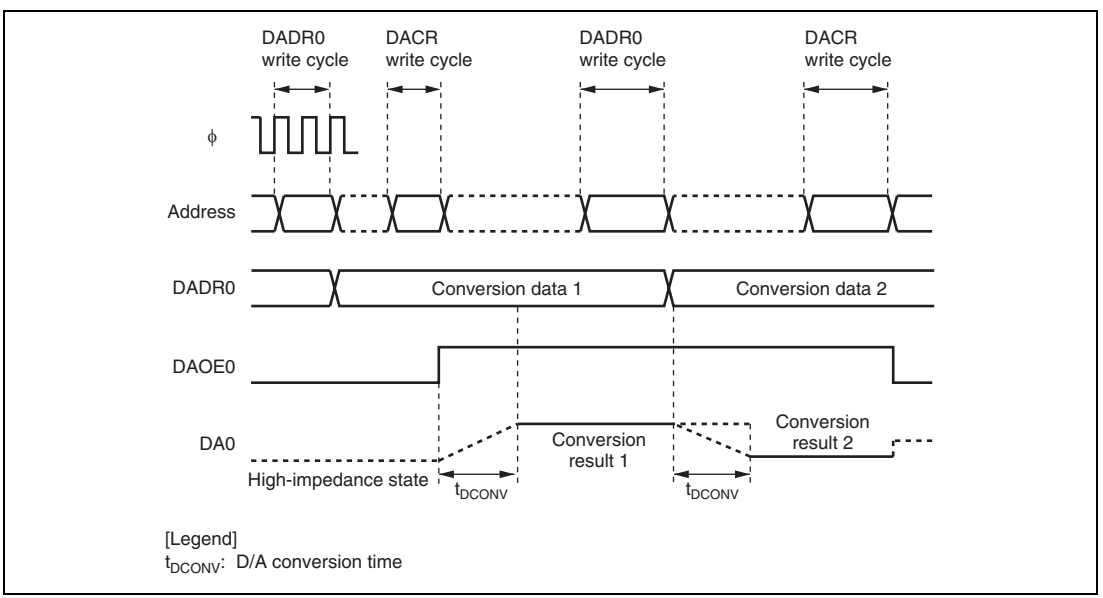


Figure 22.2 Example of D/A Converter Operation

22.5 Usage Notes

22.5.1 Module Standby Mode Setting

Operation of the D/A converter can be disabled or enabled using the standby control register. The initial setting is for operation of the D/A converter to be halted. Register access is enabled by canceling module standby mode. For details, see section 30, Power-Down Modes.

22.5.2 D/A Output Hold Function in Software Standby Mode

When this LSI enters software standby mode with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is equal to as during D/A conversion. If the analog power supply current needs to be reduced in software standby mode, clear the DAOE0, DAOE1, and DAE bits to 0 to disable the D/A outputs.

22.5.3 Setting Analog Input Voltage

The reliability of this LSI may be adversely affected if the following voltage ranges are exceeded.

1. AVcc and AVss input voltages

   Input voltages AVcc and AVss should be PVcc − 0.3 V ≤ AVcc ≤ PVcc and AVss = PVss. Do not leave the AVcc and AVss pins open when the A/D converter or D/A converter is not in use and in software standby mode. When not in use, connect AVcc to the power supply (PVcc) and AVss to the ground (PVss).
2. Setting range of AVref input voltage

   Set the reference voltage range of the AVref pin as 3.0 V ≤ AVref ≤ AVcc.

Section 23 AND/NAND Flash Memory Controller (FLCTL)

The AND/NAND flash memory controller (FLCTL) provides interfaces for an external AND-type flash memory and NAND-type flash memory. To take measures for errors specific to flash memory, the FLCTL supports the ECC generation and error detection functions.

Up to 4-symbol ECC generator, error detector, and hardware error pattern generator have been provided in addition to the 3-symbol ECC detector of the earlier products.

23.1 Features

(1) AND/NAND-Type Flash Memory Interface

- Interface directly connectable to AND/NAND-type flash memory
- Read or write in sector units (512 + 16 bytes) and ECC processing executed
- Read or write in byte units
- Supports large-block (2048 + 64 bytes) flash memory
- Supports addresses for 2 Gbits and more by extension to 5-byte addresses

Note: The FLCTL handles 512 + 16 bytes as a sector. For products with 2048 + 64 byte-pages, the FLCTL divide a page into 512 +16 bytes units (i.e. four sectors per page) for processing.

(2) Access Modes: The FLCTL can select one of the following two access modes.

- Command access mode: Performs an access by specifying a command to be issued from the FLCTL to flash memory, address, and data size to be input or output. Read, write, or erasure of data without ECC processing can be achieved.
- Sector access mode: Performs a read or write in sector units by specifying a sector address and controls ECC generation and check. By specifying the number of sectors, the continuous physical sectors can be read or written.

(3) Sectors and Control Codes

- A sector is the basic unit of access and comprised of 512-byte data and 16-byte control code fields. The control code field includes 8-byte ECC when the 3-symbol ECC circuit is used, and 10-byte ECC when the 4-symbol ECC circuit is used.
- The position of the ECC in the control code field can be specified in 4-byte units when the 3-symbol ECC circuit is used, and in 1-byte units when the 4-symbol ECC circuit is used.
- User information can be written to the part of the control code field where ECC is not placed.

(4) 3-Symbol ECC

- 64 bits (8 bytes) of ECC is added to a sector, which consists of 512-byte data + 0/4/8-byte control code.
- Error correction and detection is up to three errors (30 bits at maximum) at random positions.
- In a write operation, ECC is generated for the data and control code preceding the ECC. The control code following the ECC is not considered.
- In a read operation, an ECC error is checked for data and control code preceding the ECC. The ECC on the control code in the FIFO are the results of checking replaced by the ECC circuit, not the ECC read from flash memory.
- Error correction is not performed even when an ECC error occurs. Error corrections must be performed by software.

(5) 4-Symbol ECC

- 80 bits (10 bytes) of ECC is added to a sector, which consists of 512-byte data + 1-to 6-byte control code.
- Error correction and detection is up to four errors (40 bits at maximum) at random positions.
- In a write operation, ECC is generated for the data and control code preceding the ECC. The control code following the ECC is not considered.
- In a read operation, an ECC error is checked for data and control code preceding the ECC. The ECC on the control code in the FIFO are the results of checking replaced by the ECC circuit, not the ECC read from flash memory.
- The 4-symbol ECC circuit of the FLCTL has the capability of error correction pattern generation by hardware, which is executed on a sector-by-sector basis.
- In the error correction by hardware, addresses indicating the error positions and an error pattern for correcting the errors are output. Data replacement must be performed by software.

(6) Data Error

- When a program error or erase error occurs, the error is reflected on the error source flags. Interrupts for each source can be specified.
- When a read error occurs, an ECC in the control code is other than 0. This read error is reflected on the ECC error source flag.
- When an ECC error occurs, perform an error correction, specify another sector to be replaced, and copy the contents of the block to another sector as required.

(7) Data Transfer FIFO and Data Register

- The 224-byte data FIFO register (FLDTFIFO) is incorporated for data transfer of flash memory.
- The 32-byte control code FIFO register (FLECFIFO) is incorporated for data transfer of control code.

(8) DMA Transfer

- By individually specifying the destinations of data and control code of flash memory to the DMA controller, data and control code can be sent to different areas.

(9) Access Time

- The operating clock (FCLK) on the pins for the AND-/NAND-type flash memory is generated by dividing the peripheral clock (Pϕ). The division ratio can be specified by the FCKSEL and QTSEL bits in the common control register (FLCMNCR).
- Before changing the CPG configuration, the FLCTL must be placed in a module stop state.
- In NAND-type flash memory, the FSC and $\overline{\text{FWE}}$ pins operate at the frequency of FCLK. In AND-type flash memory, the FSC pin operates at the frequency of FCLK and the $\overline{\text{FWE}}$ pin operates at half the FCLK frequency. These operating frequencies must be specified within the maximum operating frequency of memory to be connected.

Figure 23.1 shows a block diagram of the FLCTL.

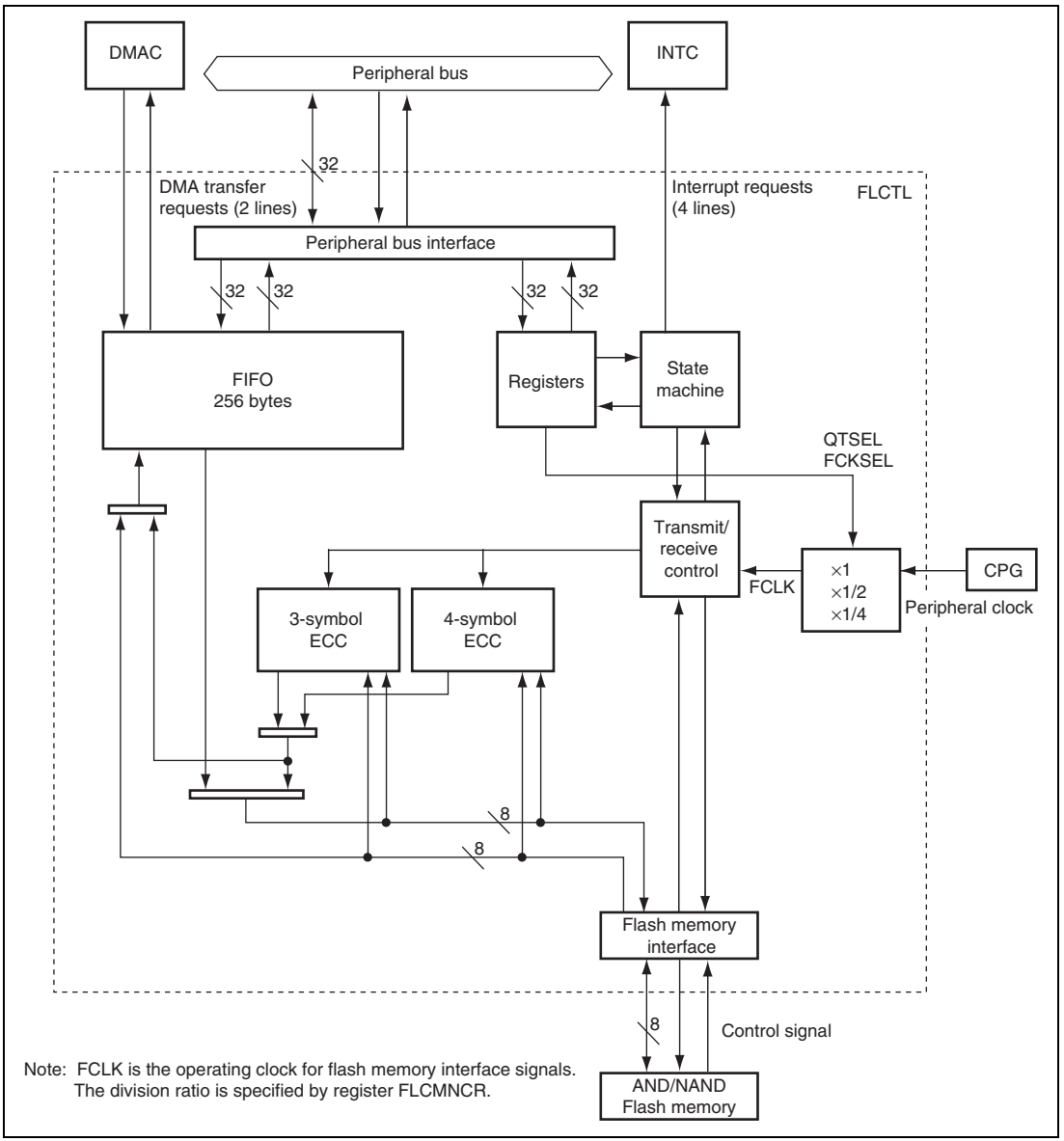


Figure 23.1 FLCTL Block Diagram

23.2 Input/Output Pins

The pin configuration of the FLCTL is listed in table 23.1.

Table 23.1 Pin Configuration

Pin Name	I/O	Corresponding Flash Memory Pin		Function
		NAND Type	AND Type	
$\overline{\text{FCE}}$	Output	$\overline{\text{CE}}$	$\overline{\text{CE}}$	Chip Enable Enables flash memory connected to this LSI.
NAF7 to NAF0	I/O	I/O7 to I/O0	I/O7 to I/O0	Data I/O I/O pins for command, address, and data.
FCDE	Output	CLE	$\overline{\text{CDE}}$	Command Latch Enable (CLE) Asserted when a command is output. Command Data Enable ($\overline{\text{CDE}}$) Asserted when a command is output.
FOE	Output	ALE	$\overline{\text{OE}}$	Address Latch Enable (ALE) Asserted when an address is output and negated when data is input or output. Output Enable ($\overline{\text{OE}}$) Asserted when data is input or when a status is read.
FSC	Output	$\overline{\text{RE}}$	SC	Read Enable ($\overline{\text{RE}}$) Reads data at the falling edge of $\overline{\text{RE}}$. Serial Clock (SC) Inputs or outputs data synchronously with the SC.
$\overline{\text{FWE}}$	Output	$\overline{\text{WE}}$	$\overline{\text{WE}}$	Write Enable Flash memory latches a command, address, and data at the rising edge of $\overline{\text{WE}}$.
FRB	Input	R/$\overline{\text{B}}$	R/$\overline{\text{B}}$	Ready/Busy Indicates ready state at high level; indicates busy state at low level.
—*	—	$\overline{\text{WP}}$	$\overline{\text{RES}}$	Write Protect/Reset When this pin goes low, erroneous erasure or programming at power on or off can be prevented.
—*	—	$\overline{\text{SE}}$	—	Spare Area Enable Used to access spare area. This pin must be fixed at low in sector access mode.

Note: * Not supported in this LSI.

23.3 Register Descriptions

Table 23.2 shows the FLCTL register configuration.

Table 23.2 Register Configuration of FLCTL

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common control register	FLCMNCR	R/W	H'00000000	H'FFFEC800	32
Command control register	FLCMDCR	R/W	H'00000000	H'FFFEC804	32
Command code register	FLCMCDR	R/W	H'00000000	H'FFFEC808	32
Address register	FLADR	R/W	H'00000000	H'FFFEC80C	32
Address register 2	FLADR2	R/W	H'00000000	H'FFFEC83C	32
Data register	FLDATAR	R/W	H'00000000	H'FFFEC810	32
Data counter register	FLDTCNTR	R/W	H'00000000	H'FFFEC814	32
Interrupt DMA control register	FLINTDMACR	R/W	H'00000000	H'FFFEC818	32
Ready busy timeout setting register	FLBSYTMR	R/W	H'00000000	H'FFFEC81C	32
Ready busy timeout counter	FLBSYCNT	R	H'00000000	H'FFFEC820	32
Data FIFO register	FLDTFIFO	R/W	H'xxxxxxxx	H'FFFEC850	32
Control code FIFO register	FLECFIFO	R/W	H'xxxxxxxx	H'FFFEC860	32
Transfer control register	FLTRCR	R/W	H'00	H'FFFEC82C	8
4-symbol ECC processing result register 1	FL4ECCRES1	R	H'03FF03FF	H'FFFEC880	32
4-symbol ECC processing result register 2	FL4ECCRES2	R	H'03FF03FF	H'FFFEC884	32
4-symbol ECC processing result register 3	FL4ECCRES3	R	H'03FF03FF	H'FFFEC888	32
4-symbol ECC processing result register 4	FL4ECCRES4	R	H'03FF03FF	H'FFFEC88C	32
4-symbol ECC control register	FL4ECCCR	R/W	H'00000000	H'FFFEC890	32
4-symbol ECC error count register	FL4ECCCNT	R/W	H'00000000	H'FFFEC894	32

23.3.1 Common Control Register (FLCMNCR)

FLCMNCR is a 32-bit readable/writable register that specifies the type (AND/NAND) of flash memory, access mode, and other items.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	ECCPOS[2]	4ECCCNTEN	4ECCEN	4ECCCORRECT	-	-	-	SNAND	QTSEL	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FCKSEL	-	ECCPOS[1:0]		ACM[1:0]		NANDWF	-	-	-	-	-	CE	-	-	TYPESEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	ECCPOS[2]	0	R/W	See the description of ECCPOS[1:0] at bits 13 and 12.
24	4ECCCNTEN	0	R/W	4-Symbol ECC Error Count Selects whether to output to the FL4ECCCNT register the total number of errors found in the sectors that have been read and the maximum number of errors found in a single sector. 0: Error counting is not performed. 1: When 4-symbol ECC circuit is used, the total number of errors found in the read sectors and the maximum number of errors in a sector are output to FL4ECCCNT. Note: When this bit is set to 1, the 4ECCCORRECT bit must be cleared to 0.
23	4ECCEN	0	R/W	4-Symbol ECC Circuit Enable Enables the 4-symbol ECC circuit by setting this bit to 1 in sector access mode. 0: 3-symbol ECC circuit is enabled. 1: 4-symbol ECC circuit is enabled. Note: When AND flash memory is used, this bit must be cleared to 0. For using 4-symbol ECC circuit, see section 23.7, Usage Notes.

Bit	Bit Name	Initial Value	R/W	Description
22	4ECCCORRECT	0	R/W	4-Symbol ECC Circuit Correction Execution Specifies to execute error correction for a single sector when the 4-symbol ECC circuit is used. The FLCTL suspends sector reading on detection of an ECC error and starts error pattern generation by the 4-symbol ECC circuit. 0: Error pattern is not output but only ECC is output. 1: Reading of sectors is suspended on detection of an ECC error.
21 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18	SNAND	0	R/W	Large-Capacity NAND Flash Memory Select This bit is used to specify 1-Gbit or larger NAND flash memory with the page configuration of 2048 + 64 bytes, and 1-Gbit or larger AG-AND flash memory. 0: When flash memory with the page configuration of 512 + 16 bytes, or AND flash memory is used. 1: When NAND flash memory with the page configuration of 2048 + 64 bytes, or 1-Gbit or larger AG-AND flash memory is used. Note: When TYPESEL = 0, this bit should not be set to 1.
17	QTSEL	0	R/W	Select Dividing Rates for Flash Clock Selects the dividing rate of clock FCLK in the flash memory. This bit is used together with FCKSEL. • QTSEL = 0, FCKSEL = 0: Divides a clock (Pϕ) provided from the CPG by two and uses it as FCLK. • QTSEL = 0, FCKSEL = 1: Uses a clock (Pϕ) provided from the CPG as FCLK. • QTSEL = 1, FCKSEL = 0: Divides a clock (Pϕ) provided from the CPG by four and uses it as FCLK. • QTSEL = 1, FCKSEL = 1: Setting prohibited
16	—	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
15	FCKSEL	0	R/W	Flash Clock Select Selects the dividing rate of clock FCLK in the flash memory. This bit is used together with QTSEL. Refer to the description of QTSEL.
14	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
13, 12	ECCPOS [1:0]	00	R/W	ECC Embedding Position Specification ECCPOS[2:0] (bits 25, 13, and 12 of this register) specifies the position to place ECC in the control code field when 3- or 4-symbol ECC circuit is used • When 4ECCEN = 0 (ECC is eight bytes) 000: Places ECC with offset of 512 bytes in a sector 001: Places ECC with offset of 516 bytes in a sector 010: Places ECC with offset of 520 bytes in a sector Other than above: Setting prohibited • When 4ECCEN = 1 (ECC is ten bytes) 000: Places ECC with offset of 518 bytes in a sector 001: Places ECC with offset of 517 bytes in a sector 010: Places ECC with offset of 516 bytes in a sector 011: Places ECC with offset of 515 bytes in a sector 100: Places ECC with offset of 514 bytes in a sector 101: Places ECC with offset of 513 bytes in a sector 110: Places ECC with offset of 512 bytes in a sector 111: Setting prohibited
11, 10	ACM[1:0]	00	R/W	Access Mode Specification 1 and 0 Specify access mode. 00: Command access mode 01: Sector access mode 10: Setting prohibited 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
9	NANDWF	0	R/W	NAND Wait Insertion Operation 0: Performs address or data input/output in one FCLK cycle 1: Performs address or data input/output in two FCLK cycles
8 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CE	0	R/W	Chip Enable 0: Disables the chip (Outputs high level to the $\overline{\text{FCE}}$ pin) 1: Enables the chip (Outputs low level to the $\overline{\text{FCE}}$ pin)
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TYPESEL	0	R/W	Memory Select 0: AND-type flash memory is selected 1: NAND-type flash memory or AG-AND is selected

23.3.2 Command Control Register (FLCMDCR)

FLCMDCR is a 32-bit readable/writable register that issues a command in command access mode, specifies address issue, and specifies source or destination of data transfer. In sector access mode, FLCMDCR specifies the number of sector transfers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR CNT2	SCTCNT[19:16]				ADR MD	CDS RC	DOSR	-	-	SEL RW	DOA DR	ADRCNT[1:0]		DOC MD2	DOC MD1
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCTCNT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	ADRCNT2	0	R	Address Issue Byte Count Specification 2 Specifies the number of bytes for the address data to be issued in address stage. This bit is used together with ADRCNT[1:0]. 0: Issue the address of byte count, specified by ADRCNT[1:0]. 1: Issue 5-byte address. ADRCNT[1:0] should be set to 00.
30 to 27	SCTCNT [19:16]	0000	R/W	Sector Transfer Count Specification [19:16] These bits are extended bits of the sector transfer count specification bits (SCTCNT) 15 to 0. SCTCNT[19:16] and SCTCNT[15:0] are used together to operate as SCTCNT[19:0], the 20-bit counter.
26	ADRMD	0	R/W	Sector Access Address Specification This bit is invalid in command access mode. This bit is valid only in sector access mode. (Set this bit to 1 when using AND-type flash memory.) 0: The value of the address register is handled as a sector address. Use this value usually in sector access. 1: The value of the address register is output as the address of flash memory. Note: Clear this bit to 0 in continuous sector access.

Bit	Bit Name	Initial Value	R/W	Description
25	CDSRC	0	R/W	Data Buffer Specification Specifies the data buffer to be read from or written to in the data stage in command access mode. 0: Specifies FLDATAR as the data buffer. 1: Specifies FLDTFIFO as the data buffer.
24	DOSR	0	R/W	Status Read Check Specifies whether or not the status read is performed after the second command has been issued in command access mode. 0: Performs no status read 1: Performs status read
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	SELRW	0	R/W	Data Read/Write Specification Specifies the direction of read or write in data stage. 0: Read 1: Write
20	DOADR	0	R/W	Address Stage Execution Specification Specifies whether or not the address stage is executed in command access mode. 0: Performs no address stage 1: Performs address stage
19, 18	ADRCNT [1:0]	00	R/W	Address Issue Byte Count Specification [1:0] Specify the number of bytes for the address data to be issued in address stage. 00: Issue 1-byte address 01: Issue 2-byte address 10: Issue 3-byte address 11: Issue 4-byte address
17	DOCMD2	0	R/W	Second Command Stage Execution Specification Specifies whether or not the second command stage is executed in command access mode. 0: Does not execute the second command stage 1: Executes the second command stage

Bit	Bit Name	Initial Value	R/W	Description
16	DOCMD1	0	R/W	First Command Stage Execution Specification Specifies whether or not the first command stage is executed in command access mode. 0: Does not execute the first command stage 1: Executes the first command stage
15 to 0	SCTCNT [15:0]	H'0000	R/W	Sector Transfer Count Specification [15:0] Specify the number of sectors to be read continuously in sector access mode. These bits are counted down for each sector transfer end and stop when they reach 0. These bits are used together with SCTCNT[19:16]. In command access mode, these bits are H'0 0001.

23.3.3 Command Code Register (FLCMCDR)

FLCMCDR is a 32-bit readable/writable register that specifies a command to be issued in command access or sector access.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMD2[7:0]								CMD1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 to 8	CMD2[7:0]	H'00	R/W	Second Command Data Specify a command code to be issued in the second command stage.
7 to 0	CMD1[7:0]	H'00	R/W	First Command Data Specify a command code to be issued in the first command stage.

23.3.4 Address Register (FLADR)

FLADR is a 32-bit readable/writable register that specifies the value to be output as an address.

The address of the size specified by ADRCNT[1:0] in the command control register is output sequentially from ADR1 in byte units. By the sector access address specification bit (ADRMD) of the command control register, it is possible to specify whether the sector number set in the address data bits is converted into an address to be output to the flash memory.

- When ADRMD = 1

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR4[7:0]								ADR3[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR2[7:0]								ADR1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	ADR4[7:0]	H'00	R/W	Fourth Address Data Specify 4th data to be output to flash memory as an address when ADRMD = 1.
23 to 16	ADR3[7:0]	H'00	R/W	Third Address Data Specify 3rd data to be output to flash memory as an address when ADRMD = 1.
15 to 8	ADR2[7:0]	H'00	R/W	Second Address Data Specify 2nd data to be output to flash memory as an address when ADRMD = 1.
7 to 0	ADR1[7:0]	H'00	R/W	First Address Data Specify 1st data to be output to flash memory as an address when ADRMD = 1.

- When ADRMD = 0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	ADR[25:16]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 0	ADR[25:0]	H'000 0000	R/W	Sector Address Specification Specify a sector number to be accessed when ADRMD = 0. The sector number is converted into an address and is output to flash memory. When the ADRCNT2 bit in FLCMDCR = 1, the ADR[25:0] bits are valid. When the ADRCNT2 bit in FLCMDCR = 0, the ADR[17:0] bits are valid. See figure 23.15 for details. • Large-block products (2048 + 64 bytes) ADR[25:2] specifies the page address and ADR[1:0] specifies the column address in sector units. ADR[1:0] = 00: 0th byte (sector 0) ADR[1:0] = 01: (512 + 16)th byte (sector 1) ADR[1:0] = 00: (1024 + 32)th byte (sector 2) ADR[1:0] = 00: (1536 + 48)th byte (sector 3) • Small-block products (512 + 16 bytes) Only the page address can be specified.

23.3.5 Address Register 2 (FLADR2)

FLADR2 is a 32-bit readable/writable register, and is valid when the ADRCNT2 bit in FLCMDCR is set to 1. FLADR2 specifies an address to be output in command access mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	ADR5[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	ADR5[7:0]	H'00	R/W	Fifth Address Data Specify 5th data to be output to flash memory as an address when ADRMD = 1.

23.3.6 Data Counter Register (FLDTCNTR)

FLDTCNTR is a 32-bit readable/writable register that specifies the number of bytes to be read or written in command access mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECFLW[7:0]								DTFLW[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	DTCNT[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	ECFLW[7:0]	H'00	R	FLECFIFO Access Count Specify the number of longwords in FLECFIFO to be read or written. These bit values are used when the CPU reads from or writes to FLECFIFO. In FLECFIFO read, these bits specify the number of longwords of the data that can be read from FLECFIFO. In FLECFIFO write, these bits specify the number of longwords of unoccupied area that can be written in FLECFIFO.
23 to 16	DTFLW[7:0]	H'00	R	FLDTFIFO Access Count Specify the number of longwords in FLDTFIFO to be read or written. These bit values are used when the CPU reads from or writes to FLDTFIFO. In FLDTFIFO read, these bits specify the number of longwords of the data that can be read from FLDTFIFO. In FLDTFIFO write, these bits specify the number of longwords of unoccupied area that can be written in FLDTFIFO.
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	DTCNT[11:0]	H'000	R/W	Data Count Specification Specify the number of bytes of data to be read or written in command access mode. (Up to 2048 + 64 bytes can be specified.)

23.3.7 Data Register (FLDATAR)

FLDATAR is a 32-bit readable/writable register. It stores input/output data used when 0 is written to the CDSRC bit in FLCMDCR in command access mode. FLDATAR cannot be used for reading or writing of five or more bytes of contiguous data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DT4[7:0]								DT3[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DT2[7:0]								DT1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DT4[7:0]	H'00	R/W	Fourth Data Specify the 4th data to be input or output via the NAF7 to NAF0 pins. In write: Specify write data In read: Store read data
23 to 16	DT3[7:0]	H'00	R/W	Third Data Specify the 3rd data to be input or output via the NAF7 to NAF0 pins. In write: Specify write data In read: Store read data
15 to 8	DT2[7:0]	H'00	R/W	Second Data Specify the 2nd data to be input or output via the NAF7 to NAF0 pins. In write: Specify write data In read: Store read data
7 to 0	DT1[7:0]	H'00	R/W	First Data Specify the 1st data to be input or output via the NAF7 to NAF0 pins. In write: Specify write data In read: Store read data

23.3.8 Interrupt DMA Control Register (FLINTDMACR)

FLINTDMACR is a 32-bit readable/writable register that enables or disables DMA transfer requests or interrupts. A transfer request from the FLCTL to the DMAC is issued after each access mode has been started.

Bits 9 to 5 are the flag bits that indicate various errors occurred in flash memory access and whether there is a transfer request from the FIFO. Only 0 can be written to these bits. To clear a flag, write 0 to the target flag bit and 1 to the other flag bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	4ECE INTE	ECER INTE	-	-	FIFOTRG [1:0]		AC1 CLR	AC0 CLR	DREQ1 EN	DREQ0 EN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	EC ERB	ST ERB	BTO ERB	TRR EQF1	TRR EQF0	STER INTE	RBER INTE	TE INTE	TR INTE1	TR INTE0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	4ECEINTE	0	R/W	4-Symbol ECC Pattern Generation End Interrupt Enable Enables or disables an interrupt to CPU by 4-symbol ECC pattern generation end. 0: Disables an interrupt to CPU by 4-symbol ECC pattern generation end. 1: Enables an interrupt to CPU by 4-symbol ECC pattern generation end.
24	ECERINTE	0	R/W	ECC Error Interrupt Enable Enables or disables an interrupt to CPU when ECC error occurs. 0: Disables an interrupt to CPU when an ECC error occurs 1: Enables an interrupt to CPU when an ECC error occurs
23, 22	—	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
21, 20	FIFOTRG [1:0]	00	R/W	FIFO Trigger Setting Specify the condition (the byte number) for generation of FLDTFIFO and FLECFIFO transfer requests. • In flash-memory read Issue an interrupt to the CPU or issue a DMA transfer request when FLDTFIFO (FLECFIFO) stores the following number of bytes or more: 00: 4 (4) 01: 16 (16) 10: 128 (4) 11: 128 (16) • In flash-memory programming Issue an interrupt to the CPU or issue a DMA transfer request when FLDTFIFO (FLECFIFO) has the following empty area of bytes or more: 00: 4 (4) 01: 16 (16) 10: 128 (4) 11: 128 (16) Note: For DMA transfer, set the same number as that of the single operand transfer data count (OPSEL).
19	AC1CLR	0	R/W	FLECFIFO Clear Clears FLECFIFO. 0: Retains the FLECFIFO value. In flash-memory access, this bit should be cleared to 0. 1: Clears FLECFIFO. After FLECFIFO has been cleared, this bit should be cleared to 0.
18	AC0CLR	0	R/W	FLDTFIFO Clear Clears FLDTFIFO. 0: Retains the FLDTFIFO value. In flash-memory access, this bit should be cleared to 0. 1: Clears FLDTFIFO. After FLDTFIFO has been cleared, this bit should be cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
17	DREQ1EN	0	R/W	FLECFIFODMA Request Enable Enables or disables the DMA transfer request issued from FLECFIFO. 0: Disables the DMA transfer request issued from FLECFIFO 1: Enables the DMA transfer request issued from FLECFIFO
16	DREQ0EN	0	R/W	FLDTFIFODMA Request Enable Enables or disables the DMA transfer request issued from FLDTFIFO. 0: Disables the DMA transfer request issued from the FLDTFIFO 1: Enables the DMA transfer request issued from the FLDTFIFO
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	ECERB	0	R/(W)*	ECC Error Indicates the result of ECC error detection. This bit is set to 1 if an ECC error occurs while flash memory is read in sector access mode. No interrupt occurs even if this bit is set to 1. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no ECC error occurs (Latched ECC is all 0.) 1: Indicates that an ECC error occurs
8	STERB	0	R/(W)*	Status Error Indicates the result of status read. This bit is set to 1 if the specific bit in the bits STAT[7:0] in FLBSYCNT is set to 1 in status read. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no status error occurs (the specific bit in the bits STAT[7:0] in FLBSYCNT is 0.) 1: Indicates that a status error occurs For details on the specific bit in STAT7 to STAT0 bits, see section 23.4.7, Status Read.

Bit	Bit Name	Initial Value	R/W	Description
7	BTOERB	0	R/(W)*	R/$\overline{B}$ Timeout Error This bit is set to 1 if an R/$\overline{B}$ timeout error occurs (the bits RBTIMCNT[19:0] in FLBSYCNT are decremented to 0). This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no R/$\overline{B}$ timeout error occurs 1: Indicates that an R/$\overline{B}$ timeout error occurs
6	TRREQF1	0	R/(W)*	FLECFIFO Transfer Request Flag Indicates that a transfer request is issued from FLECFIFO. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no transfer request is issued from FLECFIFO 1: Indicates that a transfer request is issued from FLECFIFO
5	TRREQF0	0	R/(W)*	FLDTFIFO Transfer Request Flag Indicates that a transfer request is issued from FLDTFIFO. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no transfer request is issued from FLDTFIFO 1: Indicates that a transfer request is issued from FLDTFIFO
4	STERINTE	0	R/W	Interrupt Enable at Status Error Enables or disables an interrupt request to the CPU when a status error has occurred. 0: Disables the interrupt request to the CPU by a status error 1: Enables the interrupt request to the CPU by a status error

Bit	Bit Name	Initial Value	R/W	Description
3	RBERINTE	0	RW	Interrupt Enable at R/$\overline{B}$ Timeout Error Enables or disables an interrupt request to the CPU when a timeout error has occurred. 0: Disables the interrupt request to the CPU by an R/$\overline{B}$ timeout error 1: Enables the interrupt request to the CPU by an R/$\overline{B}$ timeout error
2	TEINTE	0	R/W	Transfer End Interrupt Enable Enables or disables an interrupt request to the CPU when a transfer has been ended (TREND bit in FLTRCR). 0: Disables the transfer end interrupt request to the CPU 1: Enables the transfer end interrupt request to the CPU
1	TRINTE1	0	R/W	FLECFIFO Transfer Request Enable to CPU Enables or disables an interrupt request to the CPU by a transfer request issued from FLECFIFO. 0: Disables an interrupt request to the CPU by a transfer request from FLECFIFO. 1: Enables an interrupt request to the CPU by a transfer request from FLECFIFO. When the DMA transfer is enabled, this bit should be cleared to 0.
0	TRINTE0	0	R/W	FLDTFIFO Transfer Request Enable to CPU Enables or disables an interrupt request to the CPU by a transfer request issued from FLDTFIFO. 0: Disables an interrupt request to the CPU by a transfer request from FLDTFIFO 1: Enables an interrupt request to the CPU by a transfer request from FLDTFIFO When the DMA transfer is enabled, this bit should be cleared to 0.

Note: * Only 0 can be written to these bits.

23.3.9 Ready Busy Timeout Setting Register (FLBSYTMR)

FLBSYTMR is a 32-bit readable/writable register that specifies the timeout time when the FRB pin is busy.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	RBTMOUT[19:16]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBTMOUT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 0	RBTMOUT[19:0]	H'00000	R/W	Ready Busy Timeout Specify timeout time (the number of Pϕ clocks) in busy state. When these bits are set to 0, timeout is not generated.

23.3.10 Ready Busy Timeout Counter (FLBSYCNT)

FLBSYCNT is a 32-bit read-only register.

The status of flash memory obtained by the status read is stored in the bits STAT[7:0].

The timeout time set in the bits RBTMOUT[19:0] in FLBSYTMR is copied to the bits RBTIMCNT[19:0] and counting down is started when the FRB pin is placed in a busy state. When values in the RBTIMCNT[19:0] become 0, 1 is set to the BTOERB bit in FLINTDMACR, thus notifying that a timeout error has occurred. In this case, an FLSTE interrupt request can be issued if an interrupt is enabled by the RBERINTE bit in FLINTDMACR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STAT[7:0]								-	-	-	-	RBTIMCNT[19:16]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBTIMCNT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	STAT[7:0]	All 0	R	Indicate the flash memory status obtained by the status read.
23 to 20	—	All 0	R	Reserved These bits are always read as 0.
19 to 0	RBTIMCNT[19:0]	H'00000	R	Ready Busy Timeout Counter When the FRB pin is placed in a busy state, the values of the bits RBTMOUT[19:0] in FLBSYTMR are copied to these bits. These bits are counted down while the FRB pin is busy. A timeout error occurs when these bits are decremented to 0.

23.3.11 Data FIFO Register (FLDTFIFO)

FLDTFIFO is used to read or write the data FIFO area.

In DMA transfer, this register must be specified as the destination or source.

Note that the direction of read or write specified by the SELRW bit in FLCMDCR must match that specified in this register. When changing the read/write direction, FLDTFIFO should be cleared by setting the AC0CLR bit in FLINTDMACR before use.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTFO[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTFO[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DTFO[31:0]	H'xxxxxxxx	R/W	Data FIFO Area Read/Write Data In write: Data in this register is written to the data FIFO area. In read: Data read from the data FIFO area is stored in this register.

23.3.12 Control Code FIFO Register (FLECFIFO)

FLECFIFO is used to read or write the control code FIFO area.

In DMA transfer, data in this register must be specified as the destination (source).

Note that the direction of read or write specified by the SELRW bit in FLCMDCR must match that specified in this register. When changing the read/write direction, FLECFIFO should be cleared by setting the AC1CLR bit in FLINTDMACR before use.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECFO[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECFO[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ECFO[31:0]	H'xxxxxxxx	R/W	Control Code FIFO Area Read/Write Data In write: Data in this register is written to the control code FIFO area. In read: Data read from the control code FIFO area is stored in this register.

23.3.13 Transfer Control Register (FLTRCR)

Setting the TRSTRT bit to 1 initiates access to flash memory. Access completion can be checked by the TREND bit. During the transfer (from when the TRSTRT bit is set to 1 until the TREND bit is set to 1), the processing should not be forcibly ended (by setting the TRSTRT bit to 0).

When reading from flash memory, TREND is set when reading from flash memory have been finished. However, if there is any read data remaining in the FIFO, the processing should not be forcibly ended until all data has been read from the FIFO.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TREND	TRSTRT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TREND	0	R/W	Processing End Flag Bit Indicates that the processing performed in the specified access mode has been completed. The write value should always be 0.
0	TRSTRT	0	R/W	Transfer Start By setting this bit from 0 to 1 when the TREND bit is 0, processing in the access mode specified by the access mode specification bits ACM[1:0] is initiated. 0: Stops transfer 1: Starts transfer

23.3.14 4-Symbol ECC Processing Result Register n (FL4ECCRESn) (n = 1 to 4)

FL4ECCRESn is a 32-bit read-only register that stores the error correction pattern for the nth error generated by the 4-symbol ECC circuits and the address for the nth error. The contents of this register become valid when bits 23 (4EECEN) and 22 (4ECCCORRECT) are set to 1 and a correction pattern has been generated by the setting of the 4-symbol ECC control register (FL4ECCCR).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	LOCn[9:0]									
Initial value:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PATn[9:0]									
Initial value:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	LOCn[9:0]	All 1	R	nth Error Address Indication Indicates the address of the nth error of the four errors. Since one sector is handled as 528 bytes, the valid address range is from H'000 to H'20F. Addresses beyond the range from H'000 to H'20F are invalid (and indicate that generation of an error pattern was not possible or that there were no errors). The initial value is H’3FF. The values of these bits that are set after the 4ECCEND bit in the 4-symbol ECC control register is set to 1 are valid. Note that starting to read out the data for the next sector before reading these bits will destroy the data.
15 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 to 0	PATn[9:0]	All 1	R	nth Error Correction Pattern Indication Indicates the pattern for the correction of the nth error of the four errors. Patterns for which PAT[9:8] = B'11 and patterns for which all bits of PAT[9:0] are 0 are invalid (and indicate that generation of an error pattern was not possible or that there were no errors). The initial value is H'3FF. The values of these bits that are set after the 4ECCEND bit in the 4-symbol ECC control register is set to 1 are valid. Note that starting to read out the data for the next sector before reading these bits will destroy the data.

23.3.15 4-Symbol ECC Control Register (FL4ECCCR)

FL4ECCCR is a 32-bit readable register that indicates the processing states of the 4-symbol ECC circuit. This register consists of flag bits to which only 0 can be written. To clear a flag, write 0 to the target flag bit and 1 to the other flag bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	4ECCFA	4ECCEND	4ECCEXST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	4ECCFA	0	R/(W)*	4-Symbol ECC Uncorrectable Error Only 0 can be written to this bit. If five or more errors have been detected, it is regarded that the errors are uncorrectable and this bit is set to 1. 0: Indicates that the error is correctable. 1: Indicates that the error is uncorrectable
1	4ECCEND	0	R/(W)*	4-Symbol ECC Error Counting/Correction Pattern Generation End Only 0 can be written to this bit. When set, it indicates that counting of errors or generation of correction pattern has ended. If both of bits 4ECCFA and 4ECCEND are set to 1, it indicates that five or more errors were detected and the processing has therefore ended without generating a correction pattern.

Bit	Bit Name	Initial Value	R/W	Description
0	4ECCEXST	0	R/(W)*	4-Symbol ECC Correction Execution When an ECC error is detected, this bit is set and error counting or generation of correction pattern is executed. Generation of correction pattern is executed for a sector. 0: Error counting and correction pattern generation is stopped. 1: Error counting or correction pattern generation is executed. If the 4ECCCORRECT bit in FLCMNCR is set to 1, reading is stopped while 4ECCEXST is set to 1 and reading is restarted when 4ECCEND bit is set to 1.

Note: * Only 0 can be written to these bits.

23.3.16 4-Symbol ECC Error Count Register (FL4ECCCNT)

FL4ECCCNT is a 32-bit readable register that indicates the number of errors detected by the 4-symbol ECC circuit. Only 0 can be written to this register. To clear this register, write 0 to all bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	ERRCNT[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	ERRMAX[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	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
26 to 16	ERRCNT [10:0]	H'000	R/W	Error Counter Only 0 can be written to these bits. Indicates the total number of errors found in a series of sectors that have been read (for one block at maximum, which consists of 64 pages × 4 sectors). If a sector contains five or more errors, they are counted as five.
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	ERRMAX [2:0]	000	R/W	Maximum Number of Errors Only 0 can be written to these bits. Indicates the maximum number of errors found in a sector among the series of sectors that have been read. 000: Maximum number of errors was 0 001: Maximum number of errors was 1 010: Maximum number of errors was 2 011: Maximum number of errors was 3 100: Maximum number of errors was 4 101: Maximum number of errors was 5 or more 110: Not set 111: Not set

23.4 Operation

23.4.1 Access Sequence

The FLCTL performs accesses in several independent stages.

For example, AND-type flash memory programming consists of the following five stages.

- First command issue stage (program setup command)
- Address issue stage (program address)
- Data stage (output)
- Second command issue stage (program start command)
- Status read stage

AND-type flash memory programming access is achieved by executing these five stages sequentially. An access to flash memory is completed at the end of the final stage (status read stage).

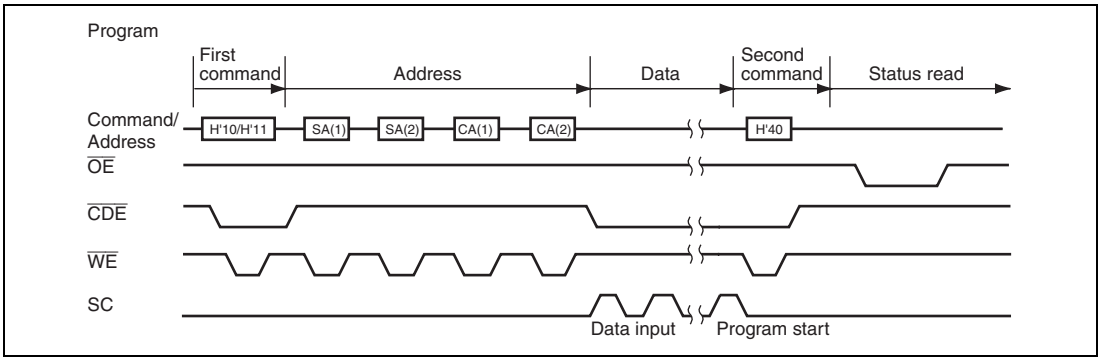


Figure 23.2 Programming Operation for AND-Type Flash Memory and Stages

For details on AND-type flash memory read and NAND-type flash memory read/program operation, see section 23.4.4, Command Access Mode.

23.4.2 Operating Modes

Two operating modes are supported.

- Command access mode
- Sector access mode

The ECC generation and error check are performed in sector access mode.

23.4.3 Register Setting Procedure

Figure 23.3 shows the register setting flow required for accessing the flash memory.

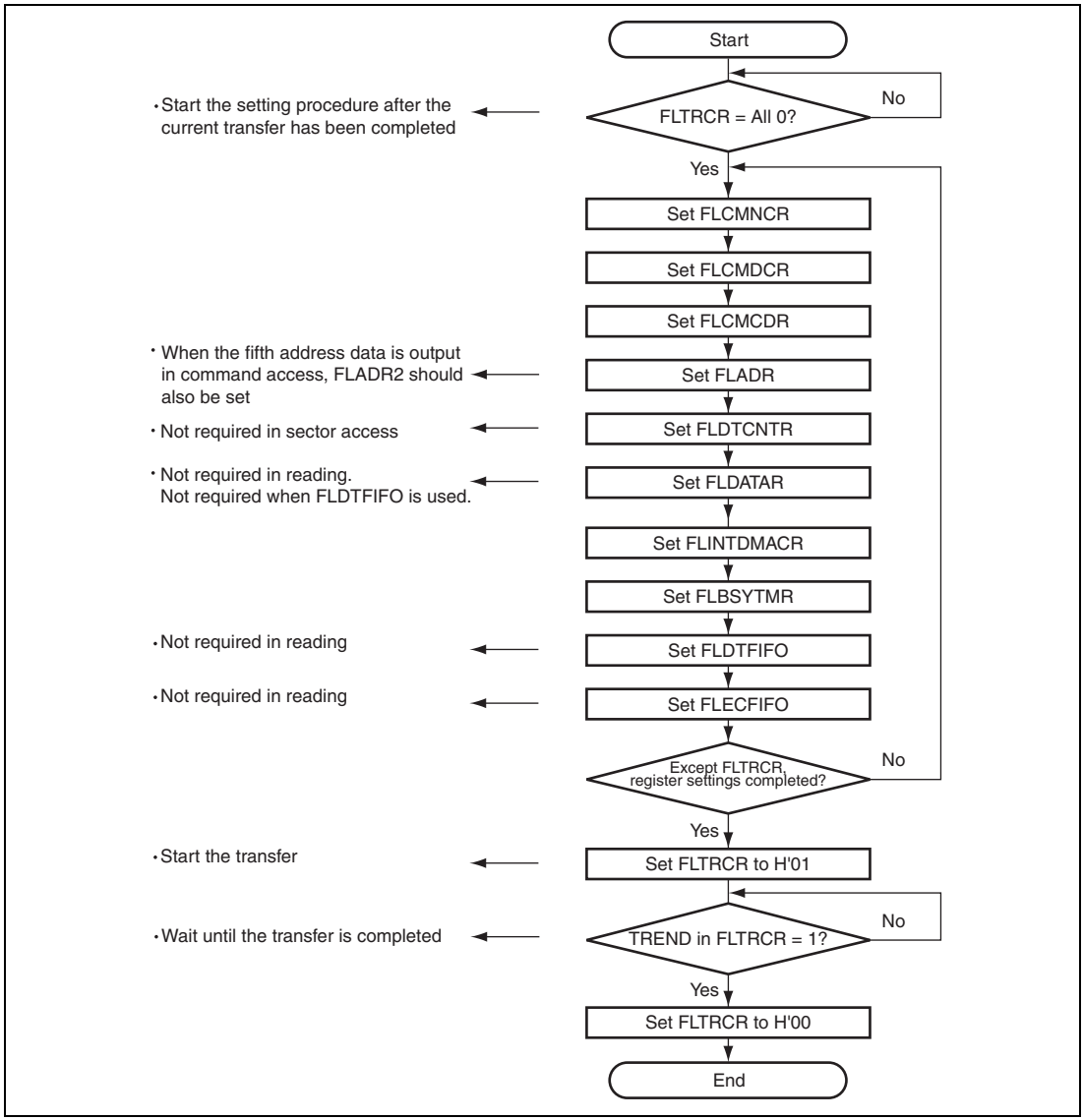


Figure 23.3 Register Setting Flow

23.4.4 Command Access Mode

Command access mode accesses flash memory by specifying a command to be issued to flash memory, address, data, read/write direction, and number of times to the registers. In this mode, I/O data can be transferred by the DMA via FLDTFIFO.

(1) AND-Type Flash Memory Access

Figures 23.4 and 23.5 show examples of read operation for AND-type flash memory. In these examples, the first command is specified as H'00 and address data length is specified as 2 bytes (SA1 and SA2). (Only SA1 and SA2 are specified, while CA1 and CA2 are not specified.). In addition, the number of read bytes is specified as 4 bytes in the data counter and H'FF is specified as the second command.

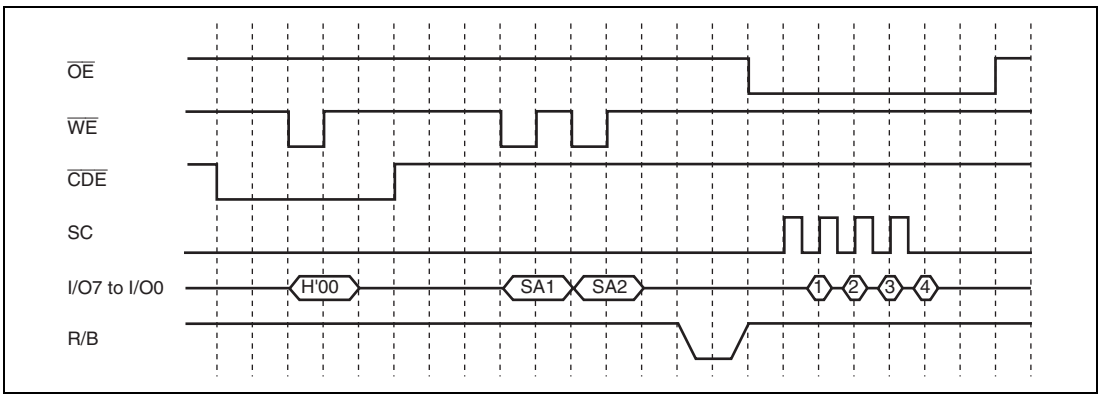


Figure 23.4 Read Operation Timing for AND-Type Flash Memory (1)

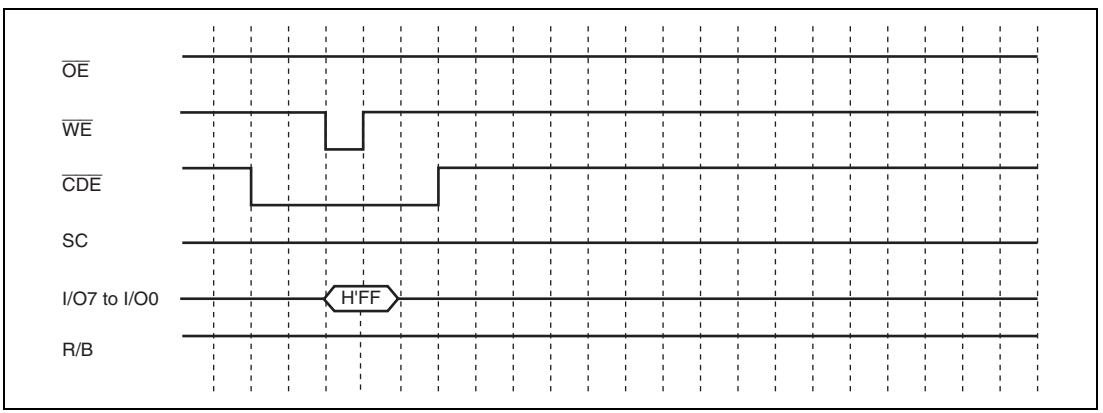


Figure 23.5 Read Operation Timing for AND-Type Flash Memory (2)

Figures 23.6 and 23.7 show examples of programming operation for AND-type flash memory.

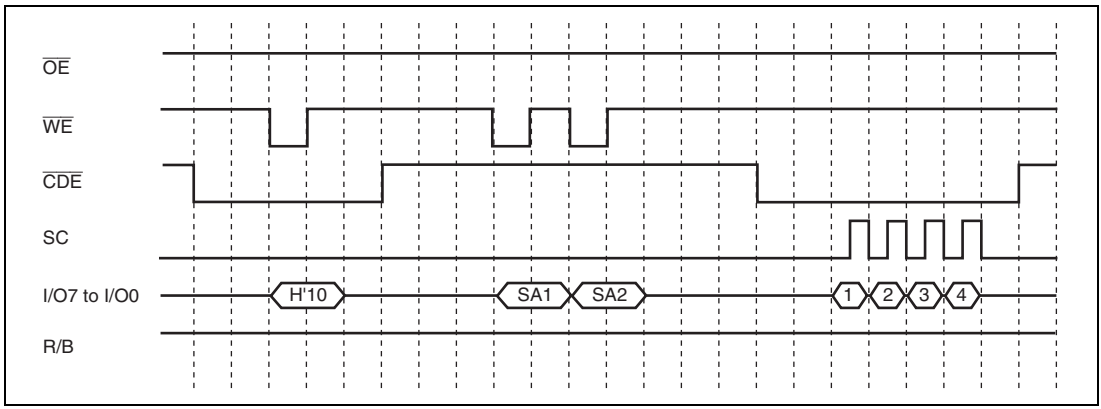


Figure 23.6 Programming Operation Timing for AND-Type Flash Memory (1)

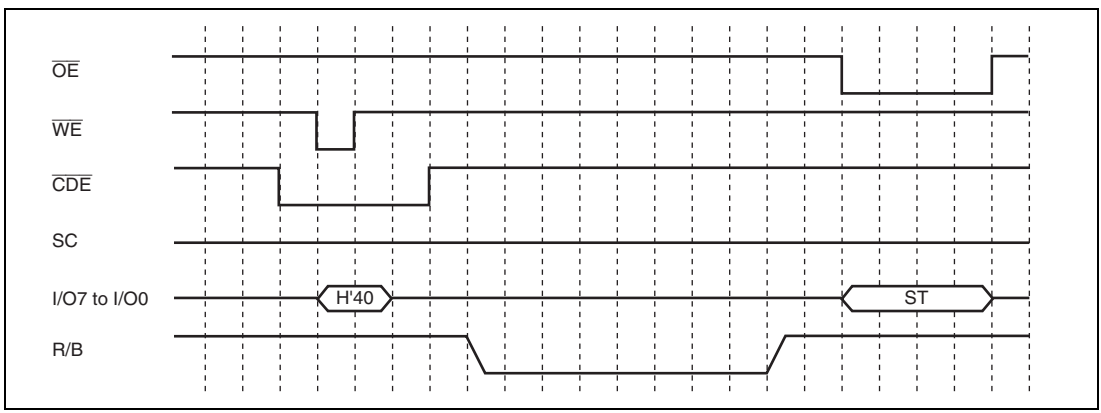


Figure 23.7 Programming Operation Timing for AND-Type Flash Memory (2)

(2) NAND-Type Flash Memory Access

Figure 23.8 shows an example of read operation for NAND-type flash memory. In this example, the first command is specified as H'00, address data length is specified as 3 bytes, and the number of read bytes is specified as 8 bytes in the data counter.

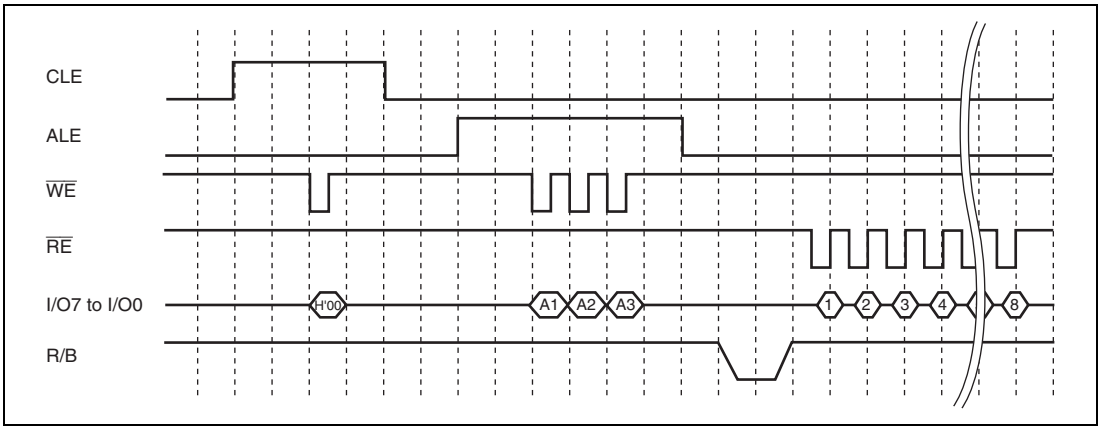


Figure 23.8 Read Operation Timing for NAND-Type Flash Memory

Figures 23.9 and 23.10 show examples of programming operation for NAND-type flash memory.

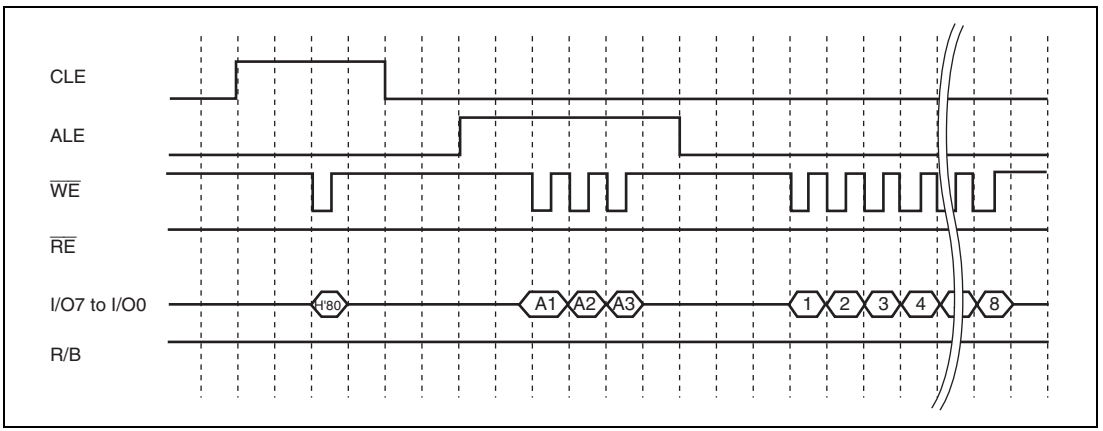


Figure 23.9 Programming Operation Timing for NAND-Type Flash Memory (1)

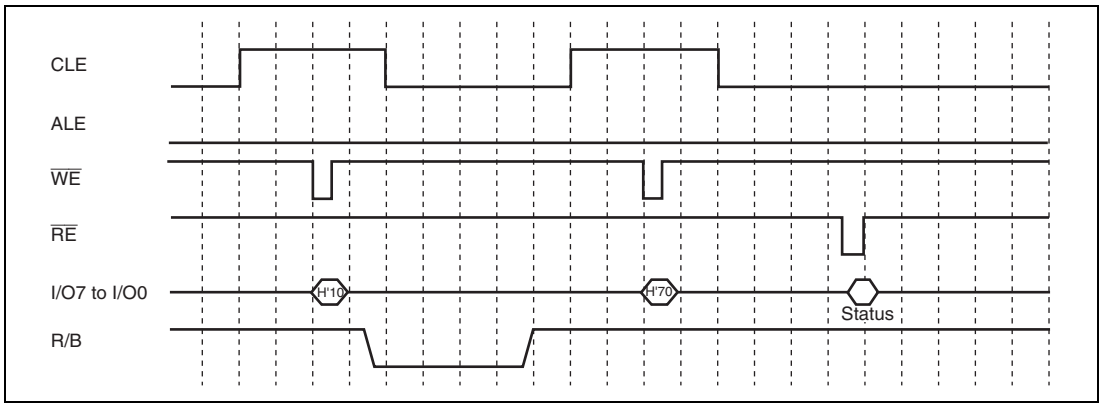


Figure 23.10 Programming Operation Timing for NAND-Type Flash Memory (2)

(3) NAND-Type Flash Memory (2048 + 64 Bytes) Access

Figure 23.11 shows an example of read operation for NAND-type flash memory (2048 + 64 bytes). In this example, the first command is specified as H'00, the second command is specified as H'30, and address data length is specified as 4 bytes. The number of read bytes is specified as 4 bytes in the data counter.

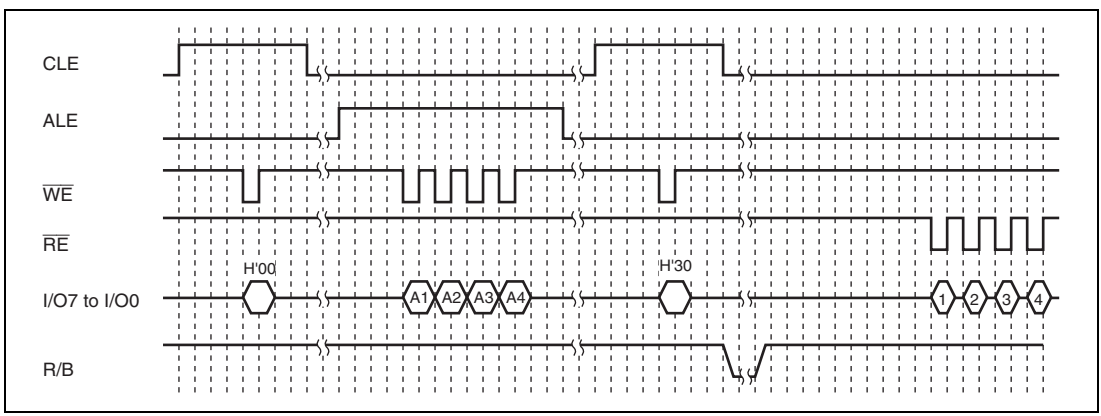


Figure 23.11 Read Operation Timing for NAND-Type Flash Memory

Figures 23.12 and 23.13 show examples of programming operation for NAND-type flash memory (2048 + 64 bytes).

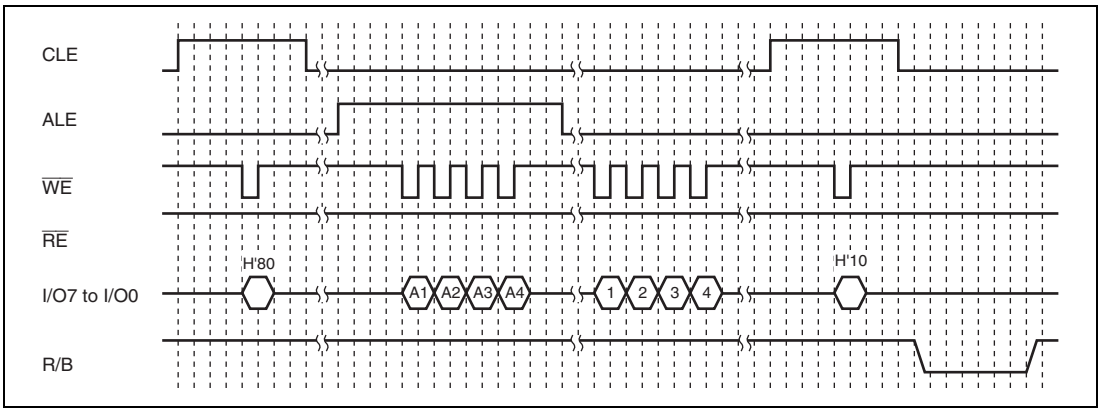


Figure 23.12 Programming Operation Timing for NAND-Type Flash Memory (1)

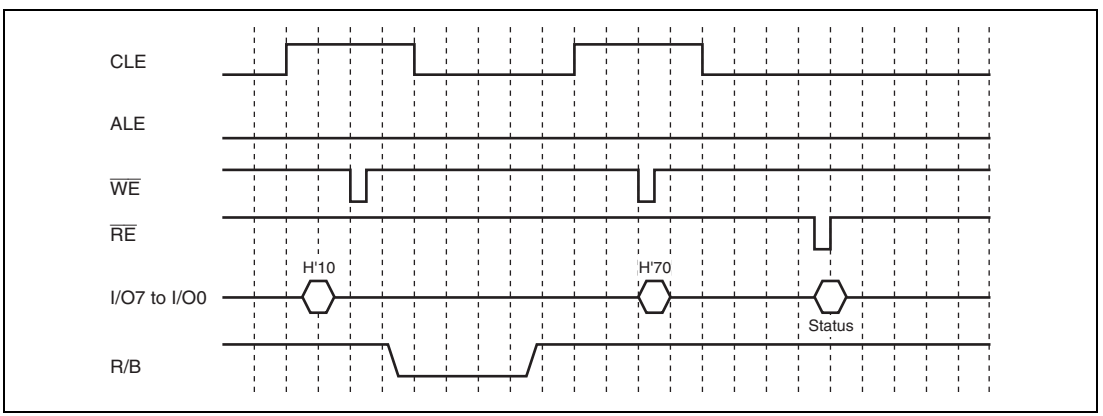


Figure 23.13 Programming Operation Timing for NAND-Type Flash Memory (2)

23.4.5 Sector Access Mode

In sector access mode, flash memory can be read or programmed in sector units by specifying the sector number of the sector to be accessed. In programming, an ECC is added. In read, an ECC error check (detection) is performed.

Since 512-byte data is stored in FLDTFIFO and 16-byte control code is stored in FLECFIFO, the DREQ1EN and DREQ0EN bits in FLINTDMACR can be set to transfer by the DMA.

Figure 23.14 shows the relationship of DMA transfer between sectors in flash memory (data and control code) and memory on the address space.

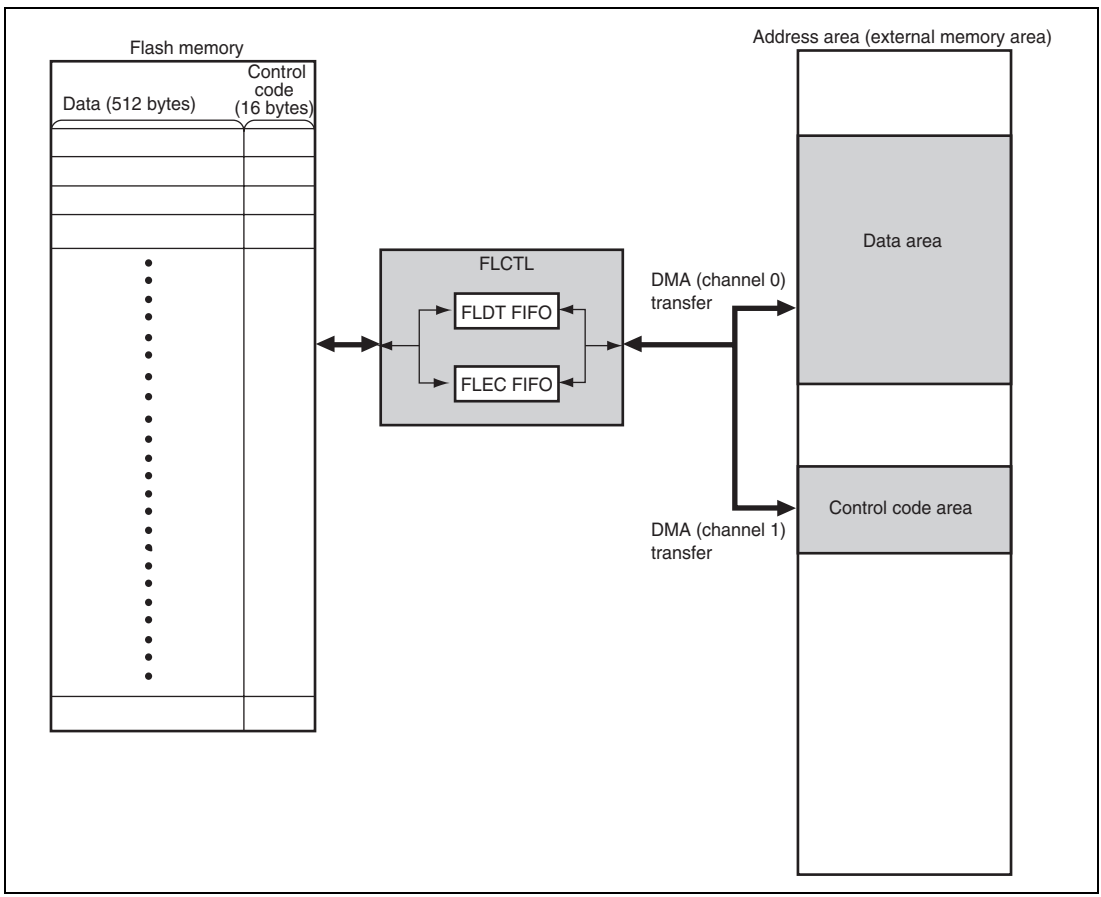


Figure 23.14 Relationship between DMA Transfer and Sector (Data and Control Code), and Memory and DMA Transfer

(1) Sector Address

Figure 23.15 shows the relationship between the physical sector address of AND/NAND-type flash memory and the address of flash memory.

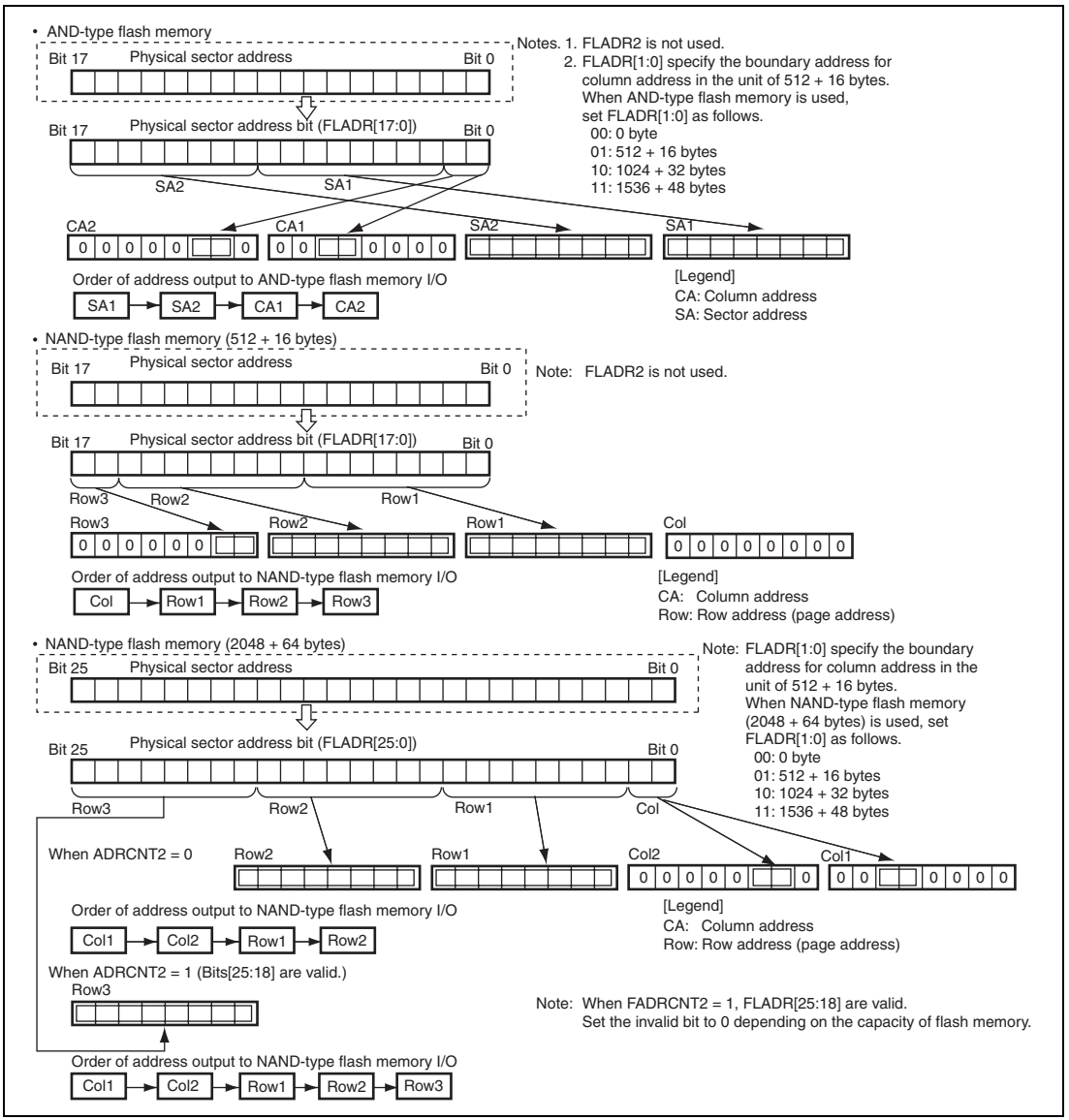


Figure 23.15 Relationship between Sector Number and Address Expansion of AND-/NAND-Type Flash Memory

(2) Continuous Sector Access

A series of sectors can be read or written by specifying the start sector address of NAND-type flash memory and the number of sectors to be transferred. Figure 23.16 shows an example of physical sector specification register and transfer count specification register settings when transferring logical sectors 0 to 40, which are not contiguous because of an unusable sector in NAND-type flash memory.

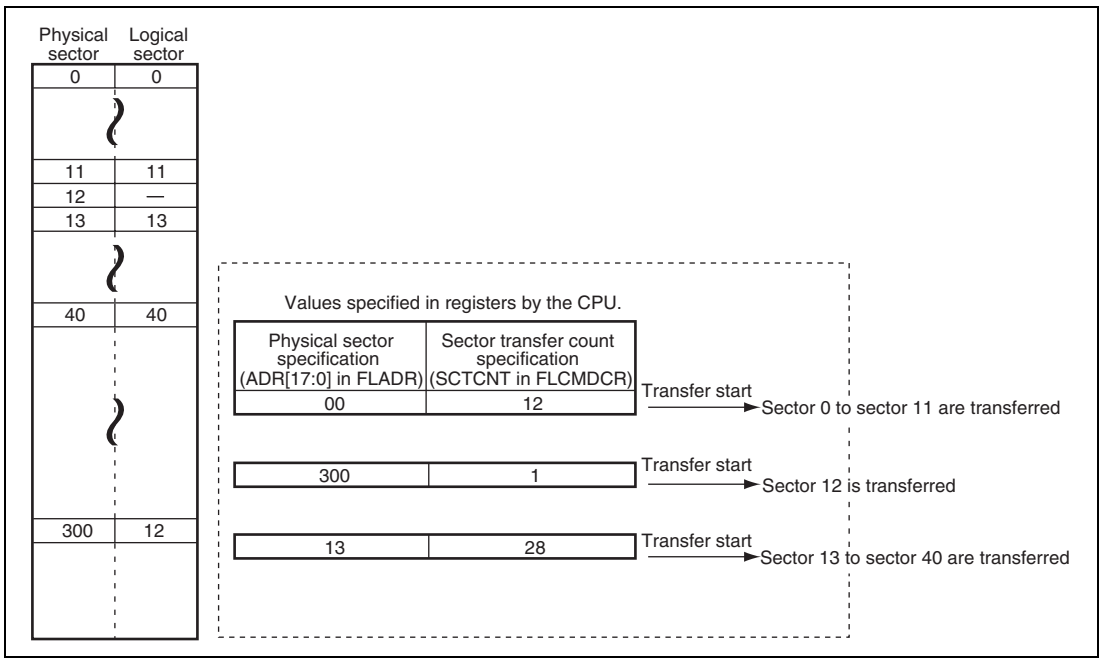


Figure 23.16 Sector Access when Unusable Sector Exists in Continuous Sectors

23.4.6 ECC Error Correction

The FLCTL generates and adds ECC during write operation in sector access mode and performs ECC error check during read operation in sector access mode. ECC processing is selectable between 3-symbol ECC, the function provided in the earlier FLCTL, and 4-symbol ECC.

With 3-symbol ECC, only ECC generation and error detection are performed and error correction is not performed. So, errors must be corrected by software. On the other hand, 4-symbol ECC is capable of ECC generation, error detection, and error correction pattern generation by hardware.

(1) Overview of 4-Symbol ECC Circuit

The 4-symbol ECC circuit in the FLCTL is capable of correcting up to10 bits per symbol, which makes a maximum of 40 bits for four symbols. However, the circuit corrects up to 32 bits because the data in the flash memory data area is counted as eight bits per symbol.

Error correction pattern generation means generation of information necessary for correcting errors, not execution of error correction. For details, see (3) 4-Symbol ECC Error Correction Pattern Generation.

The 4-symbol ECC circuit is roughly divided into three stages (figure 23.17).

1. ECC generator
2. Error count detector
3. Error correction pattern generator

ECC generation and error count detection can be executed continuously while error correction pattern generation is executed on a sector-by-sector basis.

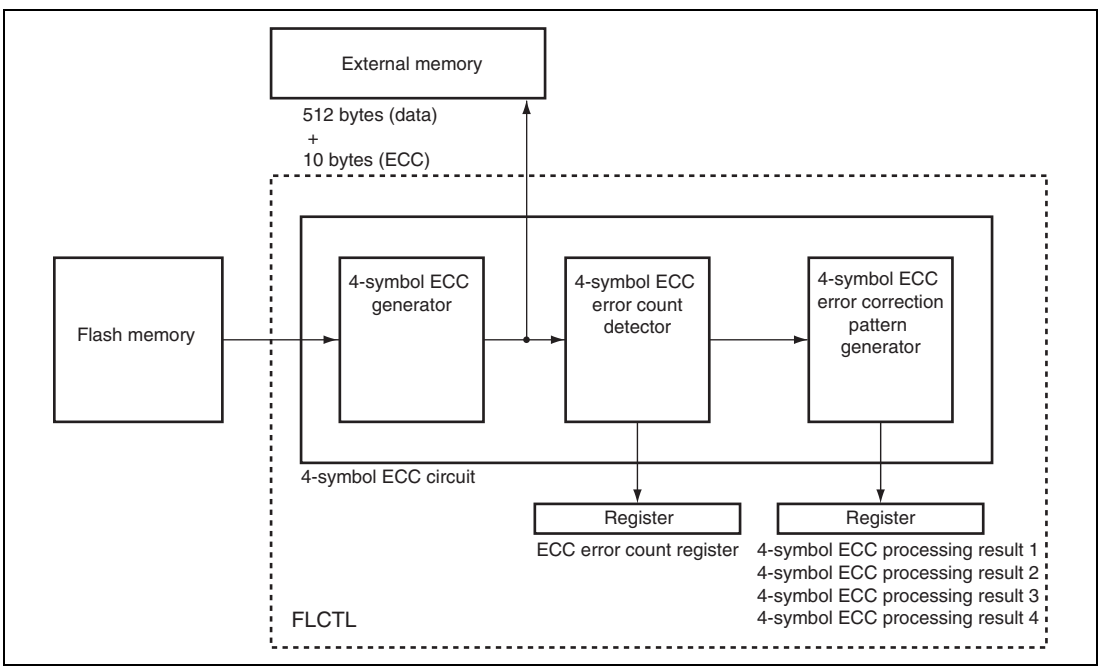


Figure 23.17 4-Symbol ECC Circuit

(2) 4-Symbol ECC Operation

Figure 23.18 shows a flowchart of the operation when the 4-symbol ECC circuit is used. Setting the 4ECCEN bit in FLCMNCR enables the 4-symbol ECC circuit and ECC is generated and output for each sector. If the 4ECCCORRECT bit in FLCMNCR is also set to 1, information necessary for correction pattern generation is accumulated in the 4-symbol ECC circuit.

In the case when the FLCTL is reading data from flash memory by continuous sector access, the reading operation stops when an error-containing sector has been read regardless of the number of remaining sectors. After reading of the error-containing sector has ended, generation of error correction pattern is started by setting the FL4ECCCR register. If the sector contains five or more errors, that sector is regarded as uncorrectable. Note that a sector may be uncorrectable for some error patterns even if it contains four or less errors. In such a case, invalid data are placed in the FL4ECCRES1 to FL4ECCRES4 registers.

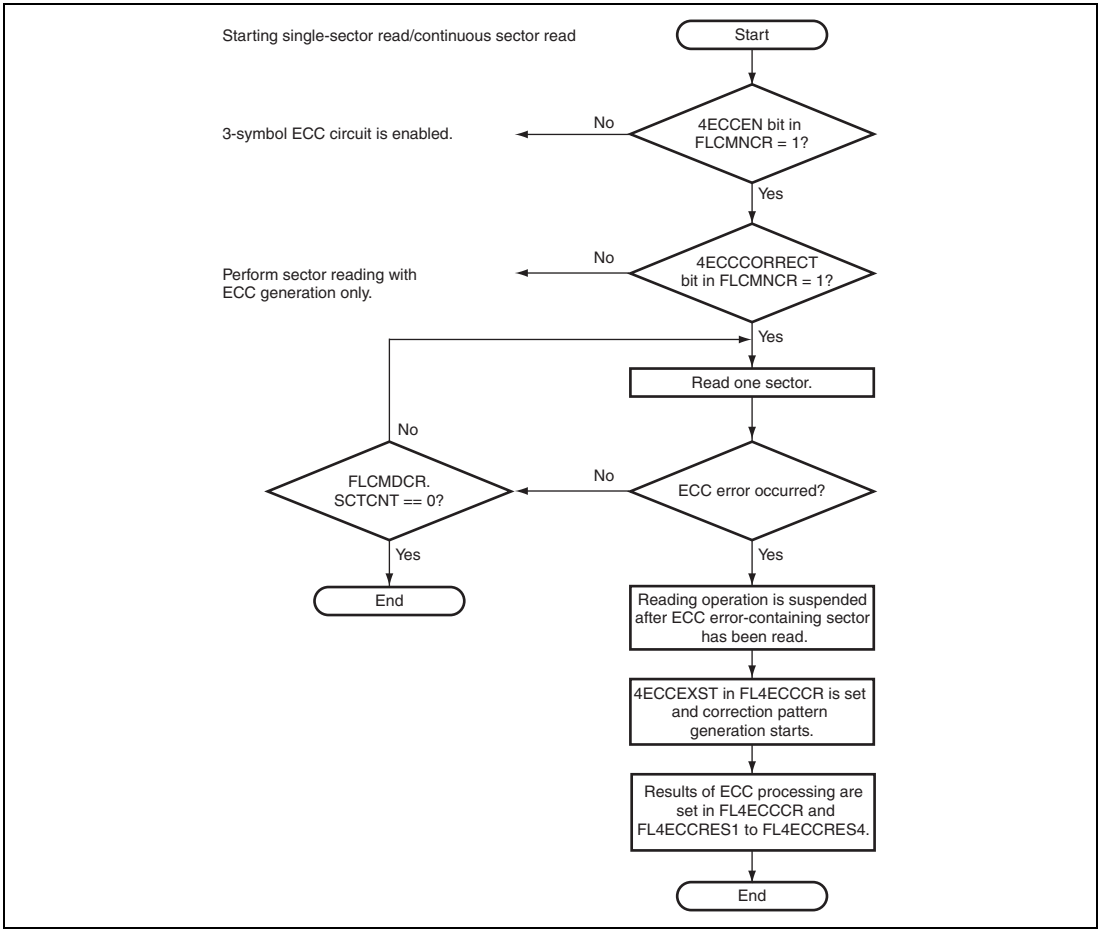


Figure 23.18 Flow of 4-Symbol ECC Operation

(3) 4-Symbol ECC Error Correction Pattern Generation

The 4-symbol ECC circuit of the FLCTL can generate error correction patterns by hardware. The original data can be restored by using the error correction patterns. Since the hardware processing only covers generation of error correction patterns, processing for data restoration must be provided by software.

The error correction patterns are output in the following format. The bits in a correction pattern at error bit positions are set to 1, so the original data is restored by taking EOR of error data and error correction pattern.

- Example 1
  Original data: B'00000000
  Erroneous data: B'11111111
  Correction pattern: B'0011111111 (higher two bits are unnecessary data)
  Recovered data: B'00000000
  (EOR of error pattern and correction pattern)

- Example 2
  Original data: B'10101010
  Erroneous data: B'01010101
  Correction pattern: B'0011111111 (higher two bits are unnecessary data)
  Recovered data: B'10101010
  (EOR of error pattern and correction pattern)

- Example 3
  Original data: B'11110000
  Erroneous data: B'00000000
  Correction pattern: B'0011110000 (higher two bits are unnecessary data)
  Recovered data: B'11110000
  (EOR of error pattern and correction pattern)

23.4.7 Status Read

The FLCTL can read the status register of an AND/NAND-type flash memory. The data in the status register is input through the I/O7 to I/O0 pins and stored in the bits STAT[7:0] in FLBSYCNT, which can be read by the CPU. If a program error or erase error is detected when the status register value is stored in the bits STAT[7:0] in FLBSYCNT, the STERB bit in FLINTDMACR is set to 1 and generates an interrupt to the CPU if the STERINTE bit in FLINTDMACR is enabled. If a status error is occurred during continuous sector access, the TREND bit in FLTRCR is set to 1 and the procedure stops.

(1) Status Read of AND-Type Flash Memory

The status register of AND-type flash memory can be read by asserting the output enable signal $\overline{OE}$ ($\overline{OE}$ = 0). If programming is executed in command access mode or sector access mode while the DOSR bit in FLCMDCR is set to 1, the FLCTL automatically asserts the $\overline{OE}$ signal and reads the status register of AND-type flash memory. When the status register of AND-type flash memory is read, the I/O7 to I/O0 pins indicate the following information as described in table 23.3.

Table 23.3 Status Read of AND-Type Flash Memory

I/O	Status (definition)	Description
I/O7	Ready/busy	0: Busy state 1: Ready state
I/O6	Reserved	—
I/O5	Erase check	0: Pass (erased) 1: Fail (erase failure)
I/O4	Program check	0: Pass (programmed) 1: Fail (program failure)
I/O3 to I/O0	Reserved	—

(2) Status Read of NAND-Type Flash Memory

The status register of NAND-type flash memory can be read by inputting command H'70 to NAND-type flash memory. If programming is executed in command access mode or sector access mode while the DOSR bit in FLCMDCR is set to 1, the FLCTL automatically inputs command H'70 to NAND-type flash memory and reads the status register of NAND-type flash memory. When the status register of NAND-type flash memory is read, the I/O7 to I/O0 pins indicate the following information as described in table 23.4.

Table 23.4 Status Read of NAND-Type Flash Memory

I/O	Status (definition)	Description
I/O7	Program protection	0: Cannot be programmed 1: Can be programmed
I/O6	Ready/busy	0: Busy state 1: Ready state
I/O5 to I/O1	Reserved	—
I/O0	Program/erase	0: Pass 1: Fail

23.5 Interrupt Sources

The FLCTL has seven interrupt sources: Status error, ready/busy timeout error, ECC error, 4-symbol ECC pattern generation end, transfer end, FIFO0 transfer request, and FIFO1 transfer request. Each of the interrupt sources has its corresponding interrupt flag and the interrupt can be requested independently to the CPU if the interrupt is enabled by the interrupt enable bit. Note that the status error, ready/busy timeout error, ECC error, and 4-symbol ECC pattern generation end, use the common FLSTE interrupt to the CPU.

Table 23.5 FLCTL Interrupt Requests

Interrupt Source	Interrupt Flag	Enable Bit	Description	Priority
FLSTE interrupt	STERB	STERINTE	Status error	Highest
	BTOERB	RBERINTE	Ready/busy timeout error	
	ECERB	ECERINTE	ECC error	
	4ECCEND	4ECEINTE	4-symbol ECC pattern generation end	
FLTEND interrupt	TREND	TEINTE	Transfer end	
FLTRQ0 interrupt	TRREQF0	TRINTE0	FIFO0 transfer request	
FLTRQ1 interrupt	TRREQF1	TRINTE1	FIFO1 transfer request	Lowest

Note: Flags for the FIFO0 overrun error/underrun error and FIFO1 overrun error/underrun error also exist. However, no interrupt is requested to the CPU.

23.6 DMA Transfer Specifications

The FLCTL can request DMA transfers separately to the data area FLDTFIFO and control code area FLECFIFO. Table 23.6 summarizes DMA transfer enable or disable states in each access mode.

Table 23.6 DMA Transfer Specifications

	Sector Access Mode	Command Access Mode
FLDTFIFO	DMA transfer enabled	DMA transfer enabled
FLECFIFO	DMA transfer enabled	DMA transfer disabled

For details on DMAC settings, see section 11, Direct Memory Access Controller (DMAC).

23.7 Usage Notes

23.7.1 Writing to the Control-Code Area when 4-Symbol ECC Circuit Is in Use

Follow the procedure given below to write to the control-code area when the 4-symbol ECC circuit is in use. If this procedure is not followed, correctly writing to the control-code area of the flash memory will not be possible.

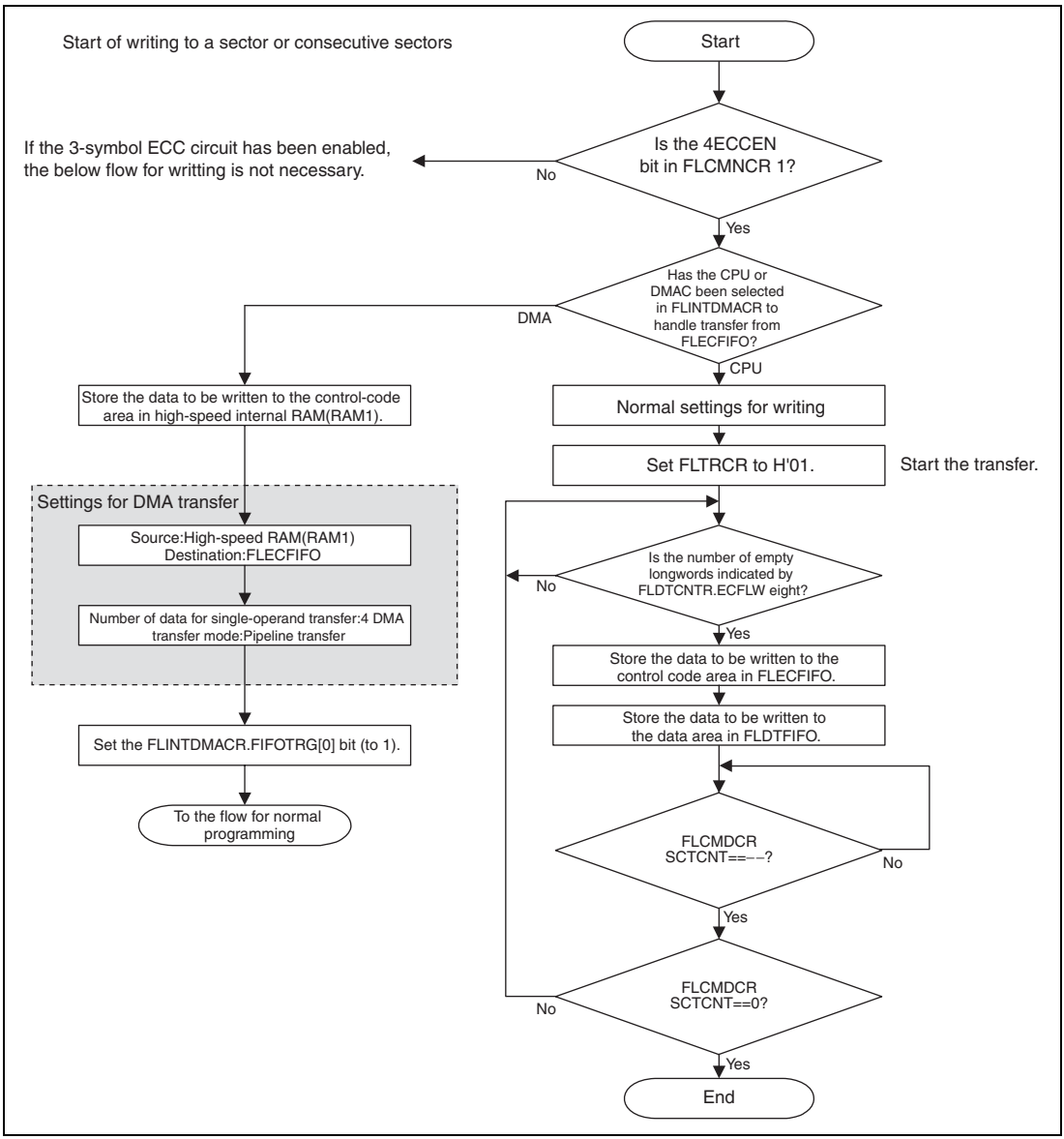


Figure 23.19 Writing Procedure to the Control-Code Area when 4-Symbol ECC is Used

23.7.2 Usage Notes for the SNAND bit

When using the SNAND bit in FLCMNCR, only the first command or the second command is corresponded in spite of the setting of the DOCMD1 or DOCMD2 bit in FLCMDCR.

When no command or only the first command is issued, 0 should be written in the SNAND bit.

Section 24 USB 2.0 Host/Function Module (USB)

The USB 2.0 host/function module (USB) is a USB controller, which provides capabilities as a USB host controller and USB function controller function.

When used as the host controller, this module supports high-speed transfer defined by USB (universal serial bus) Specification 2.0, full-speed transfer, and low-speed transfer, allowing the use of two USB ports.

When used as the function controller, this module supports high-speed transfer defined by USB Specification 2.0, and full-speed transfer, allowing the use of one USB port.

This module has a USB transceiver* and supports all of the transfer types defined by the USB specification.

This module has a 10-Kbyte buffer memory for data transfer, providing a maximum of ten pipes. Any endpoint numbers can be assigned to PIPE1 to PIPE9, based on the function devices or user system for communication.

Note: * When this module is to be used, start by making settings for the internal USB transceiver. For details, see section 24.5.1, Procedure for Setting the USB Transceiver.

24.1 Features

(1) Host Controller and Function Controller Supporting USB High-Speed Operation

- The USB host controller and USB function controller are incorporated.
- The USB host controller and USB function controller can be switched by register settings.
- Both high-speed transfer (480 Mbps) and full-speed transfer (12 Mbps) are supported.
- High-speed/full-speed/low-speed USB transceiver (shared by the USB host and USB function) is incorporated.

(2) Reduced Number of External Pins and Space-Saving Installation

- On-chip D+ pull-up resistor (during USB function operation)
- On-chip D+ and D- pull-down resistor (during USB host operation)
- On-chip D+ and D- terminal resistor (during high-speed operation)
- On-chip D+ and D- output resistor (during low-speed/full-speed operation)

(3) All Types of USB Transfers Supported

- Control transfer
- Bulk transfer
- Interrupt transfer (high bandwidth transfers not supported)
- Isochronous transfer (high bandwidth transfers not supported)

(4) Internal Bus Interfaces

- Two DMA interface channels are incorporated.

(5) Pipe Configuration

- On-chip 10-Kbyte buffer memory for USB communications
- Up to ten pipes can be selected (including the default control pipe)
- Programmable pipe configuration
- Endpoint numbers can be assigned flexibly to PIPE1 to PIPE9.
- Transfer conditions that can be set for each pipe:

  PIPE0: Control transfer, continuous transfer mode, 256-byte fixed-size single buffer

  PIPE1 and PIPE2: Bulk transfers/isochronous transfer, continuous transfer mode, programmable buffer size (up to 2-Kbytes: double buffer can be specified)

  PIPE3 to PIPE5: Bulk transfer, continuous transfer mode, programmable buffer size (up to 2-Kbytes: double buffer can be specified)

  PIPE6 and PIPE9: Interrupt transfer, 64-byte fixed single buffer

(6) Features of the USB Host Controller

- High-speed transfer (480 Mbps), full-speed transfer (12 Mbps), and low-speed transfer (1.5 Mbps) are supported.
- Communications with multiple peripheral devices connected via a single hub
- Auto response for reset handshake
- Automatic scheduling for SOF and packet transmissions
- Programmable intervals for isochronous and interrupt transfers

(7) Features of the USB Function Controller

- Both high-speed transfer (480 Mbps) and full-speed transfer (12 Mbps) are supported.
- Automatic recognition of high-speed operation or full-speed operation based on automatic response to the reset handshake
- Control transfer stage control function
- Device state control function
- Auto response function for SET_ADDRESS request
- NAK response interrupt function (NRDY)
- SOF interpolation function

(8) Other Features

- Byte endian swap function that allows handling of data in both big endian and little endian formats
- Transfer ending function using transaction count
- DMA transfer ending function
- BRDY interrupt event notification timing change function (BFRE)
- Function that automatically clears the buffer memory after the data for the pipe specified at the DnFIFO (n = 0 or 1) port has been read (DCLRM)
- NAK setting function for response PID generated by end of transfer (SHTNAK)

24.2 Input/Output Pins

Table 24.1 shows the pin configuration of the USB. If the module is not in use, handle the pins as indicated below.

- Be sure to apply the power supply voltage.
- Connect DP1, DP0, DM1, DM0, and VBUS to Vss.
- Connect REFRIN to USBAPVcc via a 5.6-kΩ resistor with tolerance of ±20%.
- Refer to section 5.3, Clock Operating Modes, for handling of ht USB_X1 and USB_X2 pins.

Table 24.1 USBH/F Pin Configuration

Category	Name	Pin Name	I/O	Function
USB bus interface	USB D+ data	DP1, DP0	I/O	D+ I/O of the USB on-chip transceiver This pin should be connected to the D+ pin of the USB bus.
	USB D- data	DM1, DM0	I/O	D− I/O of the USB on-chip transceiver This pin should be connected to the D- pin of the USB bus.
VBUS	VBUS input	VBUS	Input	USB cable connection monitor pin This pin should be connected directly to the Vbus of the USB bus. Whether the Vbus is connected or disconnected can be detected. If this pin is not connected with the Vbus of the USB bus, it should be supplied with 5 V. It should be supplied with 5 V also when the host controller function is selected. Note: Vbus is not provided to the connected device.
Reference resistance	Reference input	REFRIN	Input	Reference resistor connection pin This pin should be connected to USBAPVss through a 5.6-Ω ±1% resistor.
Clock	USB crystal oscillator/external clock	USB_X1	Input	These pins should be connected to crystal oscillators for the USB. The USB_X1 pin can be used for external clock input.
		USB_X2	Input	

Category	Name	Pin Name	I/O	Function
Power supply	Transceiver block analog pin power supply	USBAPVcc	Input	Power supply for pins
	Transceiver block analog pin ground	USBAPVss	Input	Ground for pins
	Transceiver block analog core power supply	USBAVcc	Input	Power supply for the core
	Transceiver block analog core ground	USBAVss	Input	Ground for the core
	Transceiver block digital core power supply	USBDVcc	Input	Power supply for the core

24.3 Register Description

Table 24.2 shows the register configuration of the USB.

Table 24.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
PORT0 system configuration control register	SYSCFG0	R/W	H'xx0x	H'FFFF0000	16
PORT1 system configuration control register	SYSCFG1	R/W	H'xxxF	H'FFFF0002	16
PORT0 system configuration status register	SYSSTS0	R	H'xxxx	H'FFFF0004	16
PORT1 system configuration status register	SYSSTS1	R	H'xxxx	H'FFFF0006	16
PORT0 device state control register	DVSTCTR0	R/W	H'xx0x	H'FFFF0008	16
PORT1 device state control register	DVSTCTR1	R/W	H'xx0x	H'FFFF000A	16
Test mode register	TESTMODE	R/W	H'xxx0	H'FFFF000C	16
DMA0 pin configuration register	D0FBCFG	R/W	H'xxxx	H'FFFF0010	16
DMA1 pin configuration register	D1FBCFG	R/W	H'xxxx	H'FFFF0012	16
CFIFO port register	CFIFO	R/W	H'00000000	H'FFFF0014	8, 16, 32
D0FIFO port register	D0FIFO	R/W	H'00000000	H'FFFF0018	8, 16, 32
D1FIFO port register	D1FIFO	R/W	H'00000000	H'FFFF001C	8, 16, 32
CFIFO port select register	CFIFOSEL	R/W	H'xxxx	H'FFFF0020	16
CFIFO port control register	CFIFOCTR	R/W	H'x000	H'FFFF0022	16
D0FIFO port select register	D0FIFOSEL	R/W	H'0xxx	H'FFFF0028	16
D0FIFO port control register	D0FIFOCTR	R/W	H'x000	H'FFFF002A	16
D1FIFO port select register	D1FIFOSEL	R/W	H'0xxx	H'FFFF002C	16
D1FIFO port control register	D1FIFOCTR	R/W	H'x000	H'FFFF002E	16
Interrupt enable register 0	INTENB0	R/W	H'00xx	H'FFFF0030	16
Interrupt enable register 1	INTENB1	R/W	H'xxxx	H'FFFF0032	16
Interrupt enable register 2	INTENB2	R/W	H'xxxx	H'FFFF0034	16
BRDY interrupt enable register	BRDYENB	R/W	H'xx00	H'FFFF0036	16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
NRDY interrupt enable register	NRDYENB	R/W	H'xx00	H'FFFF0038	16
BEMP interrupt enable register	BEMPENB	R/W	H'xx00	H'FFFF003A	16
SOF output configuration register	SOFCFG	R/W	H'xxxx	H'FFFF003C	16
Interrupt status register 0	INTSTS0	R/W	H'00x0	H'FFFF0040	16
Interrupt status register 1	INTSTS1	R/W	H'xxxx	H'FFFF0042	16
Interrupt status register 2	INTSTS2	R/W	H'xxxx	H'FFFF0044	16
BRDY interrupt status register	BRDYSTS	R/W	H'xx00	H'FFFF0046	16
NRDY interrupt status register	NRDYSTS	R/W	H'xx00	H'FFFF0048	16
BEMP interrupt status register	BEMPSTS	R/W	H'xx00	H'FFFF004A	16
Frame number register	FRMNUM	R/W	H'xx00	H'FFFF004C	16
μFrame number register	UFRMNUM	R	H'xxxx	H'FFFF004E	16
USB address register	USBADDR	R	H'xxx0	H'FFFF0050	16
USB request type register	USBREQ	R/W	H'0000	H'FFFF0054	16
USB request value register	USBVAL	R/W	H'0000	H'FFFF0056	16
USB request index register	USBINDX	R/W	H'0000	H'FFFF0058	16
USB request length register	USBLENG	R/W	H'0000	H'FFFF005A	16
DCP configuration register	DCPCFG	R/W	H'xxxx	H'FFFF005C	16
DCP maximum packet size register	DCPMAXP	R/W	H'0xx0	H'FFFF005E	16
DCP control register	DCPCTR	R/W	H'0x4x	H'FFFF0060	16
Pipe window select register	PIPESEL	R/W	H'xxx0	H'FFFF0064	16
Pipe configuration register	PIPECFG	R/W	H'xxx0	H'FFFF0068	16
Pipe buffer setting register	PIPEBUF	R/W	H'xx00	H'FFFF006A	16
Pipe maximum packet size register	PIPEMAXP	R/W	H'0x00	H'FFFF006C	16
Pipe cycle control register	PIPEPERI	R/W	H'xxxx	H'FFFF006E	16
Pipe 1 control register	PIPE1CTR	R/W	H'0xxx	H'FFFF0070	16
Pipe 2 control register	PIPE2CTR	R/W	H'0xxx	H'FFFF0072	16
Pipe 3 control register	PIPE3CTR	R/W	H'0xxx	H'FFFF0074	16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Pipe 4 control register	PIPE4CTR	R/W	H'0xxx	H'FFFF0076	16
Pipe 5 control register	PIPE5CTR	R/W	H'0xxx	H'FFFF0078	16
Pipe 6 control register	PIPE6CTR	R/W	H'xxxx	H'FFFF007A	16
Pipe 7 control register	PIPE7CTR	R/W	H'xxxx	H'FFFF007C	16
Pipe 8 control register	PIPE8CTR	R/W	H'xxxx	H'FFFF007E	16
Pipe 9 control register	PIPE9CTR	R/W	H'xxxx	H'FFFF0080	16
Pipe 1 transaction counter enable register	PIPE1TRE	R/W	H'xxxx	H'FFFF0090	16
Pipe 1 transaction counter register	PIPE1TRN	R/W	H'0000	H'FFFF0092	16
Pipe 2 transaction counter enable register	PIPE2TRE	R/W	H'xxxx	H'FFFF0094	16
Pipe 2 transaction counter register	PIPE2TRN	R/W	H'0000	H'FFFF0096	16
Pipe 3 transaction counter enable register	PIPE3TRE	R/W	H'xxxx	H'FFFF0098	16
Pipe 3 transaction counter register	PIPE3TRN	R/W	H'0000	H'FFFF009A	16
Pipe 4 transaction counter enable register	PIPE4TRE	R/W	H'xxxx	H'FFFF009C	16
Pipe 4 transaction counter register	PIPE4TRN	R/W	H'0000	H'FFFF009E	16
Pipe 5 transaction counter enable register	PIPE5TRE	R/W	H'xxxx	H'FFFF00A0	16
Pipe 5 transaction counter register	PIPE5TRN	R/W	H'0000	H'FFFF00A2	16
USB AC characteristics switching register 0	USBACSWR0	R/W	H'0000	H'FFFF00C0	16
USB AC characteristics switching register 1	USBACSWR1	R/W	H'0000	H'FFFF00C2	16
Device address 0 configuration register	DEVADD0	R/W	H'x0xx	H'FFFF00D0	16
Device address 1 configuration register	DEVADD1	R/W	H'x0xx	H'FFFF00D2	16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Device address 2 configuration register	DEVADD2	R/W	H'x0xx	H'FFFF00D4	16
Device address 3 configuration register	DEVADD3	R/W	H'x0xx	H'FFFF00D6	16
Device address 4 configuration register	DEVADD4	R/W	H'x0xx	H'FFFF00D8	16
Device address 5 configuration register	DEVADD5	R/W	H'x0xx	H'FFFF00DA	16
Device address 6 configuration register	DEVADD6	R/W	H'x0xx	H'FFFF00DC	16
Device address 7 configuration register	DEVADD7	R/W	H'x0xx	H'FFFF00DE	16
Device address 8 configuration register	DEVADD8	R/W	H'x0xx	H'FFFF00E0	16
Device address 9 configuration register	DEVADD9	R/W	H'x0xx	H'FFFF00E2	16
Device address A configuration register	DEVADDA	R/W	H'x0xx	H'FFFF00E4	16

24.3.1 System Configuration Control Register 0 (SYSCFG0)

SYSCFG0 is a register that enables supply of the USB clock to this module and high-speed operation on PORT0, selects the host controller function or function controller function, controls the DP and DM pins, and enables operation of the USB block of this module.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SCKE	—	—	HSE	DCFM	DRPD	DPRPU	—	—	—	USBE
Initial value:	-	-	-	-	-	0	-	-	0	0	0	0	-	-	-	0
R/W:	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
10	SCKE	0	R/W	USB Clock Enable Setting this bit to 1 enables supply of the USB clock to this module. To disable the USB clock supply, write 0 to this bit. While this bit is clear, only SYSCFG0 and SYSCFG1 can be written to and writing to other USB module registers is disabled. Even when this bit is 0, each register can be read. 0: Disables USB clock supply to this module. 1: Enables USB clock supply to this module.
9, 8	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	HSE	0	R/W	PORT0 High-Speed Operation Enable Enables or disables high-speed operation on PORT0. • When the host controller function is selected When HSE = 0, the USB PORT0 performs low-speed or full-speed operation. Set HSE to 0 when connection of a low-speed function device to the USB PORT0 has been detected. When HSE = 1, this module executes the reset handshake protocol, and automatically drives the USB PORT0 to perform high-speed or full-speed operation according to the protocol execution result. 0: High-speed operation is disabled (full-speed or low-speed) 1: High-speed operation is enabled (this module detects the communication rate) Note: This bit should be modified after detecting device connection (after detecting the ATTCH interrupt) and before executing a USB bus reset (before setting USBRESET to 1). • When the function controller function is selected When HSE = 0, this module performs full-speed operation. When HSE = 1, this module executes the reset handshake protocol, and automatically performs high-speed or full-speed operation according to the protocol execution result. 0: High-speed operation is disabled (full-speed) 1: High-speed operation is enabled (this module detects the communication rate) Note: This bit should be modified while DPRPU is 0.

Bit	Bit Name	Initial Value	R/W	Description
6	DCFM	0	R/W	Controller Function Select Selects the host controller function or function controller function. 0: Function controller function is selected. 1: Host controller function is selected. Note: This bit should be modified while DPRPU and DPRD are 0.
5	DRPD	0	R/W	PORT0 D+/D− Line Pull-Down Control When the host controller function is selected, setting this bit to 1 enables pulling down of the D+ and D- lines of PORT0. 0: Pulling down the lines is disabled. 1: Pulling down the lines is enabled.
4	DPRPU	0	R/W	PORT0 D+ Line Resistor Control When the function controller function is selected, setting this bit to 1 enables pulling up of the D+ and D- lines of PORT0. 0: Pulling up the line is disabled. 1: Pulling up the line is enabled. Note: Set this bit to 0 when the USB is disconnected. Include the following processing when this bit is changed from 1 to 0. (1) Set the DPRPU bit to 0. (2) Wait for at least 1 μs. (3) Set the DCFM bit to 1. (4) Wait for at least 200 ns. (5) Set the DCFM bit to 0.
3 to 1	—	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
0	USBE	0	R/W	USB Block Operation Enable Enables or disables operation of the USB block in this module. Modifying this bit from 1 to 0 initializes the register bits listed in tables 24.3 and 24.4. 0: USB block operation is disabled. 1: USB block operation is enabled. This bit should be modified while SCKE is 1. When the host controller function is selected, this bit should be set to 1 after setting DRPD to 1, eliminating LNST bit chattering, and checking that the USB bus has been settled.

Note: The DRPD bit should be cleared to 0 when the function controller function is selected.
The DPRPU bit should be cleared to 0 when the host controller function is selected.

Table 24.3 Register Bits Initialized by Clearing the USBE Bit to 0 (when Function Controller Function is Selected)

Register Name	Bit Name
SYSSTS0, SYSSTS1	LNST
DVSTCTR0, DVSTCTR1	RHST
INTSTS0	DVSQ
USBADDR	USBADDR
USEREQ	bRequest, bmRequestType
USBVAL	wValue
USBINDX	wIndex
USBLENG	wLength

Table 24.4 Register Bits Initialized by Clearing the USBE Bit to 0 (when Host Controller Function is Selected)

Register Name	Bit Name
DVSTCTR0, DVSTCTR1	RHST
FRMNUM	FRNM
UFRMNUM	UFRNM

24.3.2 System Configuration Control Register 1 (SYSCFG1)

SYSCFG1 is a register that enables high-speed operation on PORT1, controls the DP and DM pins and access cycles for access to this module.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	HSE	—	DRPD	—	BWAIT[3:0]			
Initial value:	-	-	-	-	-	-	-	-	0	-	0	-	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
7	HSE	0	R/W	PORT1 High-Speed Operation Enable Enables or disables high-speed operation on PORT1. When HSE = 0, the PORT1 performs low-speed or full-speed operation. Set HSE to 0 when connection of a low-speed function device to PORT1 has been detected. When HSE = 1, this module executes the reset handshake protocol, and automatically drives the PORT1 to perform high-speed or full-speed operation according to the protocol execution result. 0: High-speed operation is disabled (full-speed or low-speed) 1: High-speed operation is enabled (this module detects the communication rate) Note: This bit should be modified after detecting device connection (after detecting the ATTCH interrupt) and before executing a USB bus reset (before setting USBRESET to 1).
6	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	DRPD	0	R/W	PORT1 D+/D− Line Pull-Down Control When the host controller function is selected, setting this bit to 1 enables pulling down of the D+ and D- lines of PORT1. 0: Pulling down the lines is disabled. 1: Pulling down the lines is enabled.
4	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.
3 to 0	BWAIT[3:0]	1111	R/W	CPU Bus Access Wait Specification These bits specify the number of wait cycles inserted in register access cycles (this setting is also applied to the wait cycles in access to FIFO ports). For details, refer to section 24.4.1(4), Register Access Wait Control. 0000: 0 wait cycles (access cycle 2) : 0010: 2 wait cycles (access cycle 4) : 0100: 4 wait cycles (access cycle 6) : 1111: 15 wait cycles (access cycle 17)

Note: The DRPD bit should be cleared to 0 when the function controller function is selected.

24.3.3 System Configuration Status Register 0 (SYSSTS0)

SYSSTS0 is a register that monitors the line status (D+ and D– lines) of the USB data bus on PORT0.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LNST[1:0]	
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
1, 0	LNST[1:0]	*	R	PORT0 USB Data Line Status The statuses of the USB data bus lines (D+ and D-) of this module are listed in table 24.5. The USB data bus line status of PORT0 can be monitored by reading the value of these bits. Note: These bits should be read after setting DPRPU to 1 to notify connection when the function controller function is selected; whereas after setting DRPD to 1 to enable pulling down the lines when the host controller function is selected.

Note: * Depends on the states of the D+ and D- lines.

24.3.4 System Configuration Status Register 1 (SYSSTS1)

SYSSTS1 is a register that monitors the line status (D+ and D– lines) of the USB data bus on PORT1.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LNST[1:0]	
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
1, 0	LNST[1:0]	*	R	PORT1 USB Data Line Status The statuses of the USB data bus lines (D+ and D-) of this module are listed in table 24.5. The USB data bus line status of PORT1 can be monitored by reading the value of these bits. These bits are only valid when the host controller function is selected. Note: These bits should be read after setting DRPD to 1 to enable pulling down the lines.

Note: * Depends on the states of the D+ and D- lines.

Table 24.5 USB Data Bus Line Status

LNST[1]	LNST[0]	During Low-Speed Operation (only when Host Controller Function is Selected)	During Full-Speed Operation	During High-Speed Operation	During Chirp Operation
0	0	SE0	SE0	Squelch	Squelch
0	1	K state	J state	Not squelch	Chirp J
1	0	J state	K state	Invalid	Chirp K
1	1	SE1	SE1	Invalid	Invalid

[Legend]

Chirp: The reset handshake protocol is being executed in high-speed operation enabled state (HSE = 1).

Squelch: SE0 or idle state

Not squelch: High-speed J state or high-speed K state

Chirp J: Chirp J state

Chirp K: Chirp K state

24.3.5 Device State Control Register 0 (DVSTCTR0)

DVSTCTR0 is a register that controls and confirms the state of the USB data bus of PORT0.

This register is initialized by a power-on reset. Only the WKUP bit is initialized by a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	-	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
8	WKUP	0	R/W	Wakeup Output When the function controller function is selected, setting this bit to 1 enables output of the remote wakeup signal to the USB bus on PORT0. The module controls the output time of a remote wakeup signal. When this bit is set to 1, this module clears this bit to 0 after outputting the 10-ms K state. According to the USB specification, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is output. If this module writes 1 to this bit right after detection of suspended state, the K state will be output after 2 ms. 0: Remote wakeup signal is not output. 1: Remote wakeup signal is output. Note: Do not write 1 to this bit, unless the device state is in the suspended state (the DVSQ bit in the INTSTS0 register is set to 1xx) and the USB host enables the remote wakeup signal. When this bit is set to 1, the USB clock must not be stopped even in the suspended state (write 1 to this bit while SCKE is 1).

Bit	Bit Name	Initial Value	R/W	Description
7	RWUPE	0	R/W	Remote Wakeup Detection Enable Enables or disables the function device connected to PORT0 to use the remote wakeup function (resume signal output) when the host controller function is selected. With this bit set to 1, on detecting the remote wakeup signal (K-state for 2.5 μs) from the function device connected to PORT0, this module outputs the resume signal (drives the port to the K-state) and sets the RESUME bit to 1 at the same time. With this bit set to 0, this module ignores the detected remote wakeup signal (K-state) from the function device connected to PORT0. 0: Downstream port wakeup is disabled. 1: Downstream port wakeup is enabled. While this bit is 1, the USB clock should not be stopped even in the suspended state (SCKE should be set to 1). Also note that the USB bus should not be reset from the suspended state (USBRST should not be set to 1); it is prohibited by USB Specification 2.0.
6	USBRST	0	R/W	PORT0 USB Bus Reset Output When the host controller function is selected, setting this bit to 1 causes this module to drive PORT0 to SE0 to reset the USB bus. Here, this module performs the reset handshake protocol if the HSE bit for PORT0 is 1. This module continues outputting SE0 while USBRST is 1. Ensure that USBRST stays 1 (= USB bus reset period) for the time defined by USB Specification 2.0. 0: USB bus reset signal is not output. 1: USB bus reset signal is output. Note: Writing 1 to this bit during communication (UACT = 1) or during the resume process (RESUME = 1) prevents this module from starting the USB bus reset process until both UACT and RESUME become 0. When the USB bus reset processing has ended, write 0 to this bit and 1 to the UACT bit at the same time.

Bit	Bit Name	Initial Value	R/W	Description
5	RESUME	0	R/W	PORT0 Resume Output When the host controller function is selected, setting this bit to 1 causes this module to drive PORT0 to the K-state and perform the resume processing. This module continues outputting K-state while RESUME is 1. Ensure that RESUME stays 1 (= resume period) for the time defined by USB Specification 2.0. 0: Resume signal is not output. 1: Resume signal is output. Note: This bit should be set to 1 only in the suspended state. When the resume processing has ended, write 0 to this bit and 1 to the UACT bit at the same time.
4	UACT	0	R/W	PORT0 USB Bus Enable When the host controller function is selected, setting this bit to 1 causes this module to enable the USB bus on PORT0 and perform SOF output and data transmission and reception. After this bit is set to 1, this module starts outputting SOF/μSOF within 1 (μ) frame. When this bit is cleared to 0, this module enters the idle state after outputting SOF/μSOF. 0: Downstream port is disabled (SOF/μSOF transmission is disabled). This module clears this bit to 0 on any of the following conditions. • A DTCH interrupt is detected during communication (while UACT = 1). • An EOFERR interrupt is detected during communication (while UACT = 1). 1: Downstream port is enabled (SOF/μSOF transmission is enabled). Note: Writing 1 to this bit should be done at the end of the USB bus reset process (writing 0 to USBRST) or at the end of the resume process from the suspended state (writing 0 to RESUME).

Bit	Bit Name	Initial Value	R/W	Description
3	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.
2 to 0	RHST[2:0]	000	R	PORT0 Reset Handshake These bits indicate the state of reset handshake on PORT0. Table 24.6 lists the reset handshake statuses of PORT0. • When the host controller function is selected These bits indicate 100 after 1 is written to USBRST. If HSE has been set to 1 for PORT0, these bits indicate 111 as soon as this module detects Chirp K from the function device. This module fixes the value of the RHST bits when 0 is written to USBRST for PORT0 and this module completes SE0 driving. If 1xxxx is written to UTST (i.e., a parameter for host function testing is set), RHST indicates 011. • When the function controller function is selected If HSE has been set to 1 for PORT0, these bits indicate 100 as soon as this module detects the USB bus reset. Then, these bits indicate 011 as soon as this module outputs Chirp K and detects Chirp JK from the USB host three times. If the connection speed is not settled at a high speed within 2.5 ms after Chirp K output, these bits indicate 010. If HSE has been set to 0 for PORT0, these bits indicate 010 as soon as this module detects the USB bus reset. A DVST interrupt is generated as soon as this module detects the USB bus reset and then the value of the RHST bits is settled at 010 or 011.

Note: When the function controller function is selected, set bits RWUPE, AUSBRST, ARESUME, and AUACT to all 0s.
When the host controller function is selected, set bit WKUP to 0.

Table 24.6 PORT0 USB Data Bus Line Statuses

Bus Status	When Function Controller Function is Selected	When Host Controller Function is Selected
Powered or no connection	000	000
Reset handshake in progress	100	1xx
Low-speed connection	—	001
Full-speed connection	010	010
High-speed connection	011	011

24.3.6 Device State Control Register 1 (DVSTCTR1)

DVSTCTR1 is a register that controls and confirms the state of the USB data bus of PORT1.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Initial value:	-	-	-	-	-	-	-	-	0	0	0	0	-	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
7	RWUPE	0	R/W	Remote Wakeup Detection Enable Enables or disables the function device connected to PORT1 to use the remote wakeup function (resume signal output) when the host controller function is selected. With this bit set to 1, on detecting the remote wakeup signal (K-state for 2.5 μs) from the function device connected to PORT1, this module outputs the resume signal (drives the port to the K-state) and sets the RESUME bit to 1 at the same time. With this bit set to 0, this module ignores the detected remote wakeup signal (K-state) from the function device connected to PORT1. 0: Downstream port wakeup is disabled. 1: Downstream port wakeup is enabled. While this bit is 1, the USB clock should not be stopped even in the suspended state (SCKE should be set to 1). Also note that the USB bus should not be reset from the suspended state (USBRST should not be set to 1); it is prohibited by USB Specification 2.0.

Bit	Bit Name	Initial Value	R/W	Description
6	USBRST	0	R/W	PORT1 USB Bus Reset Output When the host controller function is selected, setting this bit to 1 causes this module to drive PORT1 to SE0 to reset the USB bus. Here, this module performs the reset handshake protocol if the HSE bit for PORT1 is 1. This module continues outputting SE0 while USBRST is 1. Ensure that USBRST stays 1 (= USB bus reset period) for the time defined by USB Specification 2.0. 0: USB bus reset signal is not output. 1: USB bus reset signal is output. Note: Writing 1 to this bit during communication (UACT = 1) or during the resume process (RESUME = 1) prevents this module from starting the USB bus reset process until both UACT and RESUME become 0. When the USB bus reset processing has ended, write 0 to this bit and 1 to the UACT bit at the same time.
5	RESUME	0	R/W	PORT1 Resume Output When the host controller function is selected, setting this bit to 1 causes this module to drive PORT1 to the K-state and perform the resume processing. This module continues outputting K-state while RESUME is 1. Ensure that RESUME stays 1 (= resume period) for the time defined by USB Specification 2.0. 0: Resume signal is not output. 1: Resume signal is output. Note: This bit should be set to 1 only in the suspended state. When the resume processing has ended, write 0 to this bit and 1 to the UACT bit at the same time.

Bit	Bit Name	Initial Value	R/W	Description
4	UACT	0	R/W	PORT1 USB Bus Enable When the host controller function is selected, setting this bit to 1 causes this module to enable the USB bus on PORT1 and perform SOF output and data transmission and reception. After this bit is set to 1, this module starts outputting SOF/μSOF within 1 (μ) frame. When this bit is cleared to 0, this module enters the idle state after outputting SOF/μSOF. 0: Downstream port is disabled (SOF/μSOF transmission is disabled). This module clears this bit to 0 on any of the following conditions. • A DTCH interrupt is detected during communication (while UACT = 1). • An EOFERR interrupt is detected during communication (while UACT = 1). 1: Downstream port is enabled (SOF/μSOF transmission is enabled). Note: Writing 1 to this bit should be done at the end of the USB bus reset process (writing 0 to USBRST) or at the end of the resume process from the suspended state (writing 0 to RESUME).
3	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	RHST[2:0]	000	R	PORT1 Reset Handshake These bits indicate the state of reset handshake on PORT1. Table 24.7 lists the reset handshake statuses of PORT1. These bits indicate 100 after 1 is written to USBRST. If HSE has been set to 1 for PORT1, these bits indicate 111 as soon as this module detects Chirp K from the function device. This module fixes the value of the RHST bits when 0 is written to USBRST for PORT1 and this module completes SE0 driving. If 1xxxx is written to UTST (i.e., a parameter for host function testing is set), RHST indicates 011.

Note: Clear the value of bits RWUPE, USBRST, RESUME, and UACT when the function controller function is selected.

Table 24.7 PORT1 USB Data Bus Line Statuses

Bus Status	When Function Controller Function is Selected	When Host Controller Function is Selected
Powered or no connection	—	000
Reset handshake in progress	—	1xx
Low-speed connection	—	001
Full-speed connection	—	010
High-speed connection	—	011

24.3.7 Test Mode Register (TESTMODE)

TESTMODE is a register that controls the USB test signal output during high-speed operation.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	UTST[3:0]			
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
3 to 0	UTST[3:0]	0000	R/W	Test Mode Controls the output of USB test signals in high-speed operation. Table 24.8 shows the test mode operation of this module. These bits are valid only during high-speed operation. RHST = 11 in DVSTCTR should be confirmed before use. UTST3 differs depending on the controller function select bit (DCFM) setting. UTST3 should be set based on the DCFM bit. After a test has been executed with these bit settings, this module should be returned from test mode by a system reset.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	UTST[3:0]	0000	R/W	• When the host controller function is selected These bits can be set after writing 1 to DRPD for the target port, either PORT0 or PORT1, to be tested. These bits are common to PORT0 and PORT1. This module outputs waveforms to the USB port for which both DPRD and UACT have been set to 1. This module also performs high-speed termination for PORT0 and PORT1 by writing to these bits. Procedure for setting the UTST bits: 1. Power-on reset the system. 2. Start the clock supply (set SCKE to 1). 3. Set DCFM and DRPD to 1 (setting HSE to 1 is not required). 4. Set USBE to 1. 5. Set the UTST bits to the appropriate value according to the test specifications. 6. Set the UACT bit for the target port to 1. Procedure for modifying the UTST bits: 1. In the state after step 6 above, clear UACT and USBE to 0. 2. Set USBE to 1. 3. Set the UTST bits to the appropriate value according to the test specifications. 4. Set the UACT bit for the target port to 1. Note: When these bits are set to Test_SE0_NAK (1011), this module does not output the SOF packet even for the port for which UACT is set to 1. When these bits are set to Test_Force_Enable (1101), this module outputs the SOF packet to the port for which UACT is set to 1. In this test mode, this module does not perform the pertinent control even when a high-speed disconnection is detected (detection of the DTCH interrupt). When setting the UTST bits, the PID bits for all the pipes should be set to NAK. To return to normal USB communication after a test mode has been set and executed, perform a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	UTST[3:0]	0000	R/W	• When the function controller function is selected The appropriate value should be set to these bits according to the SetFeature request from the USB host during high-speed communication. Note: This module does not enter a suspended state while these bits are 0001 to 0100.

Table 24.8 Test Mode Operation

	UTST Bit Setting	
Test Mode	**When Function Controller Function is Selected**	**When Host Controller Function is Selected**
Normal operation	0000	0000
Test_J	0001	1001
Test_K	0010	1010
Test_SE0_NAK	0011	1011
Test_Packet	0100	1100
Test_Force_Enable	—	1101
Reserved	0101 to 0111	1110 to 1111

24.3.8 DMA-FIFO Bus Configuration Registers (D0FBCFG, D1FBCFG)

D0FBCFG and D1FBCFG perform access control of the D0FIFO and D1FIFO ports.

These registers are initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	DFACC[1:0]		—	—	—	—	—	—	—	TENDE	—	—	—	—
Initial value:	-	-	0	0	-	-	-	-	-	-	-	0	-	-	-	-
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
13, 12	DFACC[1:0]	00	R/W	These bits specify an access mode of the corresponding FIFO port. Select the transfer data per one operand. For detail, see section 24.4.4 (4), DMA transfer (D0FIFO, D1FIFO Ports). 00: Single-data access mode (initial value) 01: 16-byte continuous access mode 10: 32-byte continuous access mode 11: Setting prohibited
11 to 5	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
4	TENDE	0	R/W	DMA Transfer End Sampling Enable Controls acceptance of DMA transfer end signal from the direct memory access controller (DMAC), which is output at the end of DMA transfer. For detail, see section 24.4.4 (4), DMA transfer (D0FIFO, D1FIFO Ports). 0: DMA transfer end signal is not sampled 1: DMA transfer end signal is sampled
3 to 0	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

24.3.9 FIFO Port Registers (CFIFO, D0FIFO, D1FIFO)

CFIFO, D0FIFO and D1FIFO are port registers that are used to read data from the FIFO buffer memory and writing data to the FIFO buffer memory.

The transmission/reception buffer memory of this module has a FIFO structure (FIFO buffer). Use the FIFO port registers to access the FIFO buffer. There are three FIFO ports: the CFIFO, D0FIFO and D1FIFO ports. Each FIFO port is configured of a port register (CFIFO, D0FIFO, D1FIFO) that handles reading of data from the buffer memory and writing of data to the FIFO buffer memory, a select register (CFIFOSEL, D0FIFOSEL, D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a control register (CFIFOCTR, D0FIFOCTR, D1FIFOCTR).

These registers are initialized by a power-on reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIFOPORT[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIFOPORT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FIFOPORT [31:0]	H'00000000	R/W	FIFO Port Accessing these bits allow reading the received data from the buffer memory or writing the transmit data to the buffer memory.

Notes:
1. The DCP FIFO buffer should be accessed through the CFIFO port.
2. Access to the buffer memory by DMA transfer is only possible through the D0FIFO or D1FIFO port.
3. The D1FIFO and D0FIFO ports can be accessed also by the CPU.
4. When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed.
5. Registers configuring a FIFO port do not affect other FIFO ports.
6. The same pipe should not be assigned to two or more FIFO ports.
7. There are two buffer memory states: the access right is on the CPU side and it is on the SIE side. When the access right is on the SIE side, the buffer memory cannot be accessed correctly from the CPU.
8. These registers can be accessed only while the FRDY bit in the corresponding control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

9. The valid bits in this register depend on the settings of the MBW bits (access bit width setting) and BIGEND bit (endian setting) as shown in tables 24.9 to 24.11.

Table 24.9 Endian Operation in 32-Bit Access (when MBW = 10)

BIGEND	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	N + 3 address	N + 2 address	N + 1 address	N + 0 address
1	N + 0 address	N + 1 address	N + 2 address	N + 3 address

Table 24.10 Endian Operation in 16-Bit Access (when MBW = 01)

BIGEND	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	Writing: invalid, reading: prohibited		Odd address	Even address
1	Even address	Odd address	Writing: invalid, reading: prohibited	

Table 24.11 Endian Operation in 8-Bit Access (when MBW = 00)

BIGEND	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	Writing: invalid, reading: prohibited			Writing: valid Reading: valid
1	Writing: valid Reading: valid	Writing: invalid, reading: prohibited		

24.3.10 FIFO Port Select Registers (CFIFOSEL, D0FIFOSEL, D1FIFOSEL)

CFIFOSEL, D0FIFOSEL and D1FIFOSEL are registers that select the pipes to be assigned to the FIFO ports, and control access to the corresponding ports.

The same pipe should not be specified in the CURPIPE bits of CFIFOSEL, D0FIFOSEL and D1FIFOSEL. When the CURPIPE bits in D0FIFOSEL and D1FIFOSEL are cleared to B'000, no pipe is selected.

The pipe number should not be changed while the DMA transfer is enabled.

These registers are initialized by a power-on reset.

(1) CFIFOSEL

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCNT	REW	—	—	MBW[1:0]		—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Initial value:	0	0	-	-	0	0	-	0	-	-	0	-	0	0	0	0
R/W:	R/W	R/W*	R	R	R/W	R/W	R	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	RCNT	0	R/W	Read Count Mode Specifies the received data reading mode for the DTLN bits in CFIFOCTR. If this bit is cleared to 0, the DTLN bits in CFIFOCTR are cleared when all the received data in the FIFO buffer that is assigned to the pipe specified by the CURPOPE bits have been read (in double buffer mode, the DTLN bit value is cleared when all the data in a single plane has been read). If this bit is set to 1, the value in the DTLN bits is decremented every time the received data is read from the FIFO buffer that is assigned to the specified pipe. 0: The DTLN bit is cleared when all of the received data has been read. 1: The DTLN bit is decremented every time the received data is read.

Bit	Bit Name	Initial Value	R/W	Description
14	REW	0	R/W*	Buffer Pointer Rewind Set this bit to 1 to rewind the buffer pointer. When the selected pipe is in the receiving direction, setting this bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed). 0: The buffer pointer is not rewound. 1: The buffer pointer is rewound. Note: Do not set REW to 1 simultaneously with modifying the CURPIPE bits. Before setting REW to 1, ensure that FRDY is 1. To re-write to the FIFO buffer from the first data for the pipe in the transmitting direction, use the BCLR bit.
13, 12	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
11, 10	MBW[1:0]	00	R/W	FIFO Port Access Bit Width Specifies the bit width for accessing the CFIFO port. 00: 8-bit width 01: 16-bit width 10: 32-bit width 11: Setting prohibited Once reading of data from the buffer memory is started, the FIFO port access bit width cannot be modified until all the data has been read. When the selected pipe is in the receiving direction, set this bit with one of the following procedures; • Set this bit and the CURPIPE bits simultaneously. • Set this bit and the ISEL bits simultaneously when the DCP is set (CURPIPE = 000). For details, refer to section 24.4.4, FIFO Buffer. The bit width cannot be changed from the 8-bit width to the 16-/32-bit width or from the 16-bit width to the 32-bit width while data is being written to the buffer memory.

Bit	Bit Name	Initial Value	R/W	Description
9	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.
8	BIGEND	0	R/W	FIFO Port Endian Control Specifies the byte endian for the CFIFO port. 0: Little endian 1: Big endian
7, 6	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
5	ISEL	0	R/W	FIFO Port Access Direction When DCP is Selected 0: Reading from the buffer memory is selected 1: Writing to the buffer memory is selected After writing to this bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process. Even if an attempt is made to modify the setting of this bit during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective thus enabling continuous access. Set this bit and the CURPIPE bits simultaneously.
4	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	CURPIPE[3:0]	0000	R/W	FIFO Port Access Pipe Specification Specifies the pipe number of the pipe for access to the CFIFO port. 0000: DCP 0001: Pipe 1 0010: Pipe 2 0011: Pipe 3 0100: Pipe 4 0101: Pipe 5 0110: Pipe 6 0111: Pipe 7 1000: Pipe 8 1001: Pipe 9 After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number in the CURPIPE bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL. Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained and access can be continued after the setting of these bits are written back.

Note: * Only 0 can be read and 1 can be written to.

(2) D0FIFOSEL, D1FIFOSEL

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCNT	REW	DCLRM	DREQE	MBW[1:0]		—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Initial value:	0	0	-	-	0	0	-	0	-	-	-	-	0	0	0	0
R/W:	R/W	R/W*	R	R	R/W	R/W	R	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	RCNT	0	R/W	Read Count Mode Specifies the received data reading mode for the DTLN bits in DnFIFOCTR. If this bit is cleared to 0, the DTLN bits in DnFIFOCTR are cleared when all the received data in the FIFO buffer that is assigned to the pipe specified by the CURPOPE bits have been read (in double buffer mode, the DTLN bit value is cleared when all the data in a single plane has been read). If this bit is set to 1, the value in the DTLN bits is decremented every time the received data is read from the FIFO buffer that is assigned to the specified pipe. 0: The DTLN bit is cleared when all of the received data has been read. 1: The DTLN bit is decremented every time the received data is read. Note: When accessing DnFIFO with the BFRE bit set to 1, set this bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
14	REW	0	R/W*	Buffer Pointer Rewind Set this bit to 1 to rewind the buffer pointer. When the selected pipe is in the receiving direction, setting this bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed). 0: The buffer pointer is not rewound. 1: The buffer pointer is rewound. Note: Do not set REW to 1 simultaneously with modifying the CURPIPE bits. Before setting REW to 1, ensure that FRDY is 1. When accessing DnFIFO with the BFRE bit set to 1, do not set this bit to 1 after the short packet data has been all read out. To re-write to the FIFO buffer from the first data for the pipe in the transmitting direction, use the BCLR bit.
13	DCLRM	0	R/W	Auto Buffer Memory Clear after Reading Data from Specified Pipe Enables or disables the buffer memory to be cleared automatically after data has been read out using the selected pipe. With this bit set to 1, this module sets BCLR to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while BFRE is 1. 0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled. Note: When the BRDYM bit set to 1, always set this bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
12	DREQE	0	R/W	DMA Transfer Request Enable Enables or disables generation of DMA transfer requests. 0: Request disabled 1: Request enabled Note: To enable the DMA transfer request, set the CURPIPE bits first, then set this bit to 1. Before modifying the CURPIPE bit setting, this bit must be cleared to 0.
11, 10	MBW[1:0]	00	R/W	FIFO Port Access Bit Width Specifies the bit width for accessing the DnFIFO port. 00: 8-bit width 01: 16-bit width 10: 32-bit width 11: Setting prohibited Once reading of data from the buffer memory is started, the FIFO port access bit width cannot be modified until all the data has been read. When the selected pipe is in the receiving direction, set the CURPIPE and MBW bits simultaneously. For details, refer to section 24.4.4, FIFO Buffer. The bit width cannot be changed from the 8-bit width to the 16-/32-bit width or from the 16-bit width to the 32-bit width while data is being written to the buffer memory.
9	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.
8	BIGEND	0	R/W	FIFO Port Endian Control Specifies the byte endian for the DnFIFO port. 0: Little endian 1: Big endian

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
3 to 0	CURPIPE[3:0]	0000	R/W	FIFO Port Access Pipe Specification Specifies the pipe number of the pipe for access to the DnFIFO port. 0000: Invalid 0001: Pipe 1 0010: Pipe 2 0011: Pipe 3 0100: Pipe 4 0101: Pipe 5 0110: Pipe 6 0111: Pipe 7 1000: Pipe 8 1001: Pipe 9 After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number in the CURPIPE bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL. Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained and access can be continued after the setting of these bits are written back.

Note: * Only 0 can be read and 1 can be written to.

24.3.11 FIFO Port Control Registers (CFIFOCTR, D0FIFOCTR, D1FIFOCTR)

These registers determine whether or not writing to the buffer memory has been finished, the buffer on the CPU side has been cleared, and the FIFO port is accessible. CFIFOCTR, D0FIFOCTR, and D1FIFOCTR are provided for the corresponding FIFO ports.

These registers are initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BVAL	BCLR	FRDY	—	DTLN[11:0]											
Initial value:	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*1	R/W*2	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	BVAL	0	R/W*1	Buffer Memory Valid Flag When the pipe selected by the CURPIPE bits is in the transmitting direction, set this bit to 1 in the following cases. Then, this module switches the FIFO buffer from the CPU side to the SIE side, enabling transmission. • When transmitting a short packet, set this bit to 1 after data has been written. • When transmitting a zero-length packet, set this bit to 1 before data is written to the FIFO buffer. • Set this bit to 1 after the number of data bytes has been written for the pipe in continuous transfer mode, where the number is a natural integer multiple of the maximum packet size and less than the buffer size. When the data of the maximum packet size has been written for the pipe in non-continuous transfer mode, this module sets this bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission. When the specified pipe is in the transmitting direction, if 1 is written to BVAL and BCLR bits simultaneously, this module clears the data that was written before, enabling transmission of a zero-length packet. 0: Invalid 1: Writing ended

Bit	Bit Name	Initial Value	R/W	Description
15	BVAL	0	R/W*[1]	Note: Writing 1 to this bit should be done while FRDY indicates 1 (set by this module). When the specified pipe is in the receiving direction, do not set this bit to 1.
14	BCLR	0	R/W*[2]	CPU Buffer Clear This bit should be set to 1 to clear the FIFO buffer on the CPU side for the specified pipe. Of the FIFO buffers assigned to the specified pipe, the one on the CPU side is cleared. When double buffer mode is set for the FIFO buffer assigned to the specified pipe, this module clears only one plane of the FIFO buffer even when both planes are read-enabled. 0: Invalid 1: Clears the buffer memory on the CPU side. Note: When the specified pipe is the DCP, setting BCLR to 1 allows this module to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the PID bits for the DCP to NAK before setting BCLR to 1. When the specified pipe is not the DCP, writing 1 to this bit should be done while FRDY indicates 1 (set by this module).
13	FRDY	0	R	FIFO Port Ready Indicates whether the FIFO port is accessible. In the following cases, FRDY is set to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set BCLR to 1 to clear the FIFO buffer to enable transmission and reception of the next data. • A zero-length packet is received when the FIFO buffer assigned to the specified pipe is empty. • A short packet is received and the data is completely read while BFRE is 1. 0: FIFO port access is disabled. 1: FIFO port access is enabled.

Bit	Bit Name	Initial Value	R/W	Description
12	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.
11 to 0	DTLN[11:0]	H'000	R	Receive Data Length Indicates the length of the received data. These bits indicate different values depending on the RCNT bit value as described below. • RCNT = 0: These bits retain the length of received data until all the received data are read from a single FIFO buffer plane. If BFRE is 1, however, these bits retain the length of the received data until BCLR is set to 1 even after all the data has been read. • RCNT = 1: The value of these bits is decremented each time data is read from the FIFO buffer. (The value is decremented by one when MBW = 00, by two when MBW = 01, and by four when MBW = 10.) DTLN becomes 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, this module sets these bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane. Note: When RCNT is 1, it takes 10 bus cycles until DTLN is updated after the FIFO port is read.

Notes: 1. Only 1 can be written to.
2. Only 0 can be read and 1 can be written to.

24.3.12 Interrupt Enable Register 0 (INTENB0)

INTENB0 enables or disables various interrupts. If an interrupt for which the corresponding bit in this register is set to 1 has occurred, an interrupt request is output to the interrupt controller.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	VBSE	0	R/W	VBUS Interrupts Enable Enables/disables the interrupt request when VBUS interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
14	RSME	0	R/W	Resume Interrupts Enable* Enables/disables the interrupt request when RESM interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
13	SOFE	0	R/W	Frame Number Update Interrupts Enable Enables/disables the interrupt request when SOF interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
12	DVSE	0	R/W	Device State Transition Interrupts Enable* Enables/disables the interrupt request when DVST interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled

Bit	Bit Name	Initial Value	R/W	Description
11	CTRE	0	R/W	Control Transfer Stage Transition Interrupts Enable*
				Enables/disables the interrupt request when CTRT interrupt is detected.
				0: Interrupt output disabled
				1: Interrupt output enabled
10	BEMPE	0	R/W	Buffer Empty Interrupts Enable
				Enables/disables the interrupt request when BEMP interrupt is detected.
				0: Interrupt output disabled
				1: Interrupt output enabled
9	NRDYE	0	R/W	Buffer Not Ready Response Interrupts Enable
				Enables/disables the interrupt request when NRDY interrupt is detected.
				0: Interrupt output disabled
				1: Interrupt output enabled
8	BRDYE	0	R/W	Buffer Ready Interrupts Enable
				Enables/disables the interrupt request when BRDY interrupt is detected.
				0: Interrupt output disabled
				1: Interrupt output enabled
7 to 0	—	Undefined	R	Reserved
				Undefined values are read from these bits. The write value should always be 0.

Note: * The RSME, DVSE, and CTRE bits are cleared to 0 when the host controller function is selected.

24.3.13 Interrupt Enable Register 1 (INTENB1)

INTENB1 enables or disables various interrupts. If an interrupt for which the corresponding bit in this register is set to 1 has occurred, an interrupt request is output to the interrupt controller.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	—
Initial value:	-	0	-	0	0	-	-	-	-	0	0	0	-	-	-	-
R/W:	R	R/W	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.
14	BCHGE	0	R/W	PORT0 USB Bus Change Interrupt Enable Enables/disables the interrupt request when PORT0 BCHG interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
13	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.
12	DTCHE	0	R/W	PORT0 Disconnection Detection Interrupt Enable Enables/disables the interrupt request when PORT0 DTCH interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
11	ATTCHE	0	R/W	PORT0 Connection Detection Interrupt Enable Enables/disables the interrupt request when PORT0 ATTCH interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
6	EOFERRE	0	R/W	PORT0 EOF Error Detection Interrupt Enable Enables/disables the interrupt request when PORT0 EOFERR interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
5	SIGNE	0	R/W	Setup Transaction Error Interrupt Enable Enables/disables the interrupt request when SIGN interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
4	SACKE	0	R/W	Setup Transaction Normal Response Interrupt Enable Enables/disables the interrupt request when SACK interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
3 to 0	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Note: Clear each bit in this register to 0 when the function controller function is selected.

24.3.14 Interrupt Enable Register 2 (INTENB2)

INTENB2 enables or disables various interrupts. If an interrupt for which the corresponding bit in this register is set to 1 has occurred, an interrupt request is output to the interrupt controller.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	—	—	—	—	—	—
Initial value:	-	0	-	0	0	-	-	-	-	0	-	-	-	-	-	-
R/W:	R	R/W	R	R/W	R/W	R	R	R	R	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.
14	BCHGE	0	R/W	PORT1 USB Bus Change Interrupt Enable Enables/disables the interrupt request when PORT1 BCHG interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
13	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.
12	DTCHE	0	R/W	PORT1 Disconnection Detection Interrupt Enable Enables/disables the interrupt request when PORT1 DTCH interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
11	ATTCHE	0	R/W	PORT1 Connection Detection Interrupt Enable Enables/disables the interrupt request when PORT1 ATTCH interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
6	EOFERRE	0	R/W	PORT1 EOF Error Detection Interrupt Enable Enables/disables the interrupt request when PORT1 EOFERR interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
5 to 0	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Note: Clear each bit in this register to 0 when the function controller function is selected.

24.3.15 BRDY Interrupt Enable Register (BRDYENB)

BRDYENB enables or disables the BRDY interrupts for individual pipes. If a BRDY interrupt for which the corresponding bit in this register is set to 1 has occurred, an interrupt request is output to the interrupt controller.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 BRDYE	PIPE8 BRDYE	PIPE7 BRDYE	PIPE6 BRDYE	PIPE5 BRDYE	PIPE4 BRDYE	PIPE3 BRDYE	PIPE2 BRDYE	PIPE1 BRDYE	PIPE0 BRDYE
Initial value:	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
9	PIPE9BRDYE	0	R/W	BRDY interrupt Enable for PIPE9 Enables/disables the interrupt request when PIPE9 BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
8	PIPE8BRDYE	0	R/W	BRDY interrupt Enable for PIPE8 Enables/disables the interrupt request when PIPE8 BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
7	PIPE7BRDYE	0	R/W	BRDY interrupt Enable for PIPE7 Enables/disables the interrupt request when PIPE7 BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
6	PIPE6BRDYE	0	R/W	BRDY interrupt Enable for PIPE6 Enables/disables the interrupt request when PIPE6 BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled

Bit	Bit Name	Initial Value	R/W	Description
5	PIPE5BRDYE	0	R/W	BRDY interrupt Enable for PIPE5 Enables/disables the interrupt request when PIPE5 BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
4	PIPE4BRDYE	0	R/W	BRDY interrupt Enable for PIPE4 Enables/disables the interrupt request when PIPE4 BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
3	PIPE3BRDYE	0	R/W	BRDY interrupt Enable for PIPE3 Enables/disables the interrupt request when PIPE3 BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
2	PIPE2BRDYE	0	R/W	BRDY interrupt Enable for PIPE2 Enables/disables the interrupt request when PIPE2 BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
1	PIPE1BRDYE	0	R/W	BRDY interrupt Enable for PIPE1 Enables/disables the interrupt request when PIPE1 BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
0	PIPE0BRDYE	0	R/W	BRDY interrupt Enable for PIPE0 Enables/disables the interrupt request when PIPE0 BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled

24.3.16 NRDY Interrupt Enable Register (NRDYENB)

NRDYENB enables or disables the NRDY interrupts for individual pipes. If a NRDY interrupt for which the corresponding bit in this register is set to 1 has occurred, an interrupt request is output to the interrupt controller.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 NRDYE	PIPE8 NRDYE	PIPE7 NRDYE	PIPE6 NRDYE	PIPE5 NRDYE	PIPE4 NRDYE	PIPE3 NRDYE	PIPE2 NRDYE	PIPE1 NRDYE	PIPE0 NRDYE
Initial value:	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
9	PIPE9NRDYE	0	R/W	NRDY Interrupt Enable for PIPE9 Enables/disables the interrupt request when PIPE9 NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
8	PIPE8NRDYE	0	R/W	NRDY Interrupt Enable for PIPE8 Enables/disables the interrupt request when PIPE8 NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
7	PIPE7NRDYE	0	R/W	NRDY Interrupt Enable for PIPE7 Enables/disables the interrupt request when PIPE7 NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
6	PIPE6NRDYE	0	R/W	NRDY Interrupt Enable for PIPE6 Enables/disables the interrupt request when PIPE6 NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled

Bit	Bit Name	Initial Value	R/W	Description
5	PIPE5NRDYE	0	R/W	NRDY Interrupt Enable for PIPE5 Enables/disables the interrupt request when PIPE5 NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
4	PIPE4NRDYE	0	R/W	NRDY Interrupt Enable for PIPE4 Enables/disables the interrupt request when PIPE4 NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
3	PIPE3NRDYE	0	R/W	NRDY Interrupt Enable for PIPE3 Enables/disables the interrupt request when PIPE3 NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
2	PIPE2NRDYE	0	R/W	NRDY Interrupt Enable for PIPE2 Enables/disables the interrupt request when PIPE2 NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
1	PIPE1NRDYE	0	R/W	NRDY Interrupt Enable for PIPE1 Enables/disables the interrupt request when PIPE1 NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
0	PIPE0NRDYE	0	R/W	NRDY Interrupt Enable for PIPE0 Enables/disables the interrupt request when PIPE0 NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled

24.3.17 BEMP Interrupt Enable Register (BEMPENB)

BEMPENB enables or disables the BEMP interrupts for individual pipes. If a NRDY interrupt for which the corresponding bit in this register is set to 1 has occurred, an interrupt request is output to the interrupt controller.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 BEMPE	PIPE8 BEMPE	PIPE7 BEMPE	PIPE6 BEMPE	PIPE5 BEMPE	PIPE4 BEMPE	PIPE3 BEMPE	PIPE2 BEMPE	PIPE1 BEMPE	PIPE0 BEMPE
Initial value:	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
9	PIPE9BEMPE	0	R/W	BEMP Interrupt Enable for PIPE9 Enables/disables the interrupt request when PIPE9 BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
8	PIPE8BEMPE	0	R/W	BEMP Interrupt Enable for PIPE8 Enables/disables the interrupt request when PIPE8 BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
7	PIPE7BEMPE	0	R/W	BEMP Interrupt Enable for PIPE7 Enables/disables the interrupt request when PIPE7 BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
6	PIPE6BEMPE	0	R/W	BEMP Interrupt Enable for PIPE6 Enables/disables the interrupt request when PIPE6 BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled

Bit	Bit Name	Initial Value	R/W	Description
5	PIPE5BEMPE	0	R/W	BEMP Interrupt Enable for PIPE5 Enables/disables the interrupt request when PIPE5 BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
4	PIPE4BEMPE	0	R/W	BEMP Interrupt Enable for PIPE4 Enables/disables the interrupt request when PIPE4 BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
3	PIPE3BEMPE	0	R/W	BEMP Interrupt Enable for PIPE3 Enables/disables the interrupt request when PIPE3 BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
2	PIPE2BEMPE	0	R/W	BEMP Interrupt Enable for PIPE2 Enables/disables the interrupt request when PIPE2 BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
1	PIPE1BEMPE	0	R/W	BEMP Interrupt Enable for PIPE1 Enables/disables the interrupt request when PIPE1 BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
0	PIPE0BEMPE	0	R/W	BEMP Interrupt Enable for PIPE0 Enables/disables the interrupt request when PIPE0 BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled

24.3.18 SOF Output Configuration Register (SOFCFG)

SOFCFG specifies the transaction-enabled time and PIPEBRDY interrupt status clear timing.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	—	—	—	—	—	—
Initial value:	-	-	-	-	-	-	-	0	-	0	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R/W	R	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
8	TRNENSEL	0	R/W	Transaction-Enabled Time Select Selects the transaction-enabled time either for full- or low-speed communication, where is the time in which this module issues tokens in a frame via the port. 0: For non-low-speed communication 1: For low-speed communication Note: This bit is only valid when the host controller function is selected. Even when the host controller function is selected, the setting of this bit has no effect on the transaction-enabled time during high-speed communication. This bit is used in common to the two ports.
7	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.
6	BRDYM	0	R/W	PIPEBRDY Interrupt Status Clear Timing Specifies the timing of clearing the BRDY interrupt status for each pipe. 0: Clearing by writing 0 1: Automatic clearing by data reading from the FIFO buffer or data writing to the FIFO buffer Note: This bit should be set when making initial settings to the module (before communication starts). Do not change the setting of this bit after communication starts.

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Note: Clear the value of the TRNENSEL bit to 0 when the function controller function is selected.

24.3.19 Interrupt Status Register 0 (INTSTS0)

INTSTS0 indicates the statuses of various interrupts.

This register is initialized by a power-on reset. The DVST and DVSQ[2:0] bits are initialized by a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Initial value:	0	0	0	*1	0	0	0	0	-	*1	*1	*1	0	0	0	0
R/W:	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R	R	R	R	R	R	R	R/W*2	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	VBINT	0	R/W*2	VBUS Transition Detection Interrupt Status*3*4 This bit is set to 1 on detecting a transition of the VBUS pin input. When a VBINT interrupt has occurred, read the VBSTS bit, which monitors the VBUS pin level, several times and confirm that the same value is read consecutively in order to prevent chattering. 0: VBUS interrupt has not occurred 1: VBUS interrupt has occurred
14	RESM	0	R/W*2	Resume Interrupt Status*3*4*5 When the function controller function is selected, this bit is set to 1 on detecting the falling edge of the signal on the DP pin in the suspended state (DVSQ = 1xx). 0: Resume interrupt has not occurred 1: Resume interrupt has occurred

Bit	Bit Name	Initial Value	R/W	Description
13	SOFR	0	R/W*2	Frame Number Refresh Interrupt Status*3 Indicates the frame number refresh interrupt status. This bit is set to 1 in the following conditions. • When the host controller function is selected The frame number is updated with the UACT bit corresponding to PRT0 or PORT1 set to 1. (detected every 1 ms) • When the function controller function is selected When the frame number is updated (detected every 1 ms) This bit is set by internal interpolation of the frame number even when a damaged SOF packet is received from the USB host. 0: SOF interrupt has not occurred 1: SOF interrupt has occurred
12	DVST	*1	R/W*2	Device State Transition Interrupt Status*3*5 When the function controller function is selected, this module updates the DVSQ value and sets this bit to 1 on detecting a transition in the device state. After this interrupt has occurred, clear the status before the next device state transition takes place. 0: Device state transition interrupt has not occurred 1: Device state transition interrupt has occurred
11	CTRT	0	R/W*2	Control Transfer Stage Transition Interrupt Status*3*6 When the function controller function is selected, this module updates the CTSQ value and sets this bit to 1 on detecting a change in the control transfer stage. After this interrupt has occurred, clear the status before the next control transfer stage transition takes place. 0: Control transfer stage transition interrupt has not occurred 1: Control transfer stage transition interrupt has occurred

Bit	Bit Name	Initial Value	R/W	Description
10	BEMP	0	R	Buffer Empty Interrupt Status This bit is set to 1 when at least one PIPEBEMP bit in BEMPSTS is set to 1 among the PIPEBEMP bits corresponding to the PIPEBEMPE bits in BEMPENB to which 1 has been set. This bit is cleared when all the bits in BEMPSTS have been cleared. 0: BEMP interrupt has not occurred 1: BEMP interrupt has occurred
9	NRDY	0	R	Buffer Not Ready Interrupt Status This bit is set to 1 when at least one PIPENRDY bit in NRDYSTS is set to 1 among the PIPENRDY bits corresponding to the PIPENRDYE bits in NRDYENB to which 1 has been set. This bit is cleared when all the bits in NRDYSTS have been cleared. 0: NRDY interrupt has not occurred 1: NRDY interrupt has occurred
8	BRDY	0	R	Buffer Ready Interrupt Status This bit is set to 1 when at least one PIPEBRDY bit in BRDYSTS is set to 1 among the PIPEBRDY bits corresponding to the PIPEBRDYE bits in BRDYENB to which 1 has been set. This bit is cleared when all the bits in BRDYSTS have been cleared. 0: BRDY interrupt has not occurred 1: BRDY interrupt has occurred
7	VBSTS	Undefined	R	VBUS Input Status This bit reflects the level of the signal input to the VBUS pin. The VBUS input status in this bit needs a control program to prevent chattering. 0: The VBUS pin is low level. 1: The VBUS pin is high level.
6 to 4	DVSQ[2:0]	*[1]	R	Device State*[6] 000: Powered state 001: Default state 010: Address state 011: Configured state 1xx: Suspended state

Bit	Bit Name	Initial Value	R/W	Description
3	VALID	0	R/W*2	USB Request Reception*6 This bit indicates whether reception of a USB request is detected or not. 0: Not detected 1: Setup packet received
2 to 0	CTSQ[2:0]	000	R	Control Transfer Stage These bits indicate the state of the control transfer stage. 000: Idle or setup stage 001: Control read data stage 010: Control read status stage 011: Control write data stage 100: Control write status stage 101: Control write (no data) status stage 110: Control transfer sequence error 111: Setting prohibited

Notes: 1. DVST is initialized to 0 and DVSQ[2:0] to 000 by a power-on reset.
DVST is initialized to 1 and DVSQ[2:0] to 001 by a USB bus reset.
2. Only 0 can be written to.
3. To clear the VBINT, RESM, SOFR, DVST, or CTRT bit, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.
4. This module detects the changes in the statuses indicated by the VBINT and RESM bits even while the clock supply is stopped (while SCKE is 0), and outputs the corresponding interrupt requests as long as they are enabled. Clearing the status should be done after enabling the clock supply.
5. Transitions in the status of the RESM, DVST, and CTRT bits only occur when the function controller function is selected; clear the corresponding interrupt enable bits to 0 (disable) when the host controller function is selected.
6. The DVSQ, VALID, and CTRQ bits are valid when the function controller function is selected

24.3.20 Interrupt Status Register 1 (INTSTS1)

INTSTS1 indicates the statuses of various interrupts.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BCHG	—	DTCH	ATTCH	—	—	—	—	EOF ERR	SIGN	SACK	—	—	—	—
Initial value:	-	0	-	0	0	-	-	-	-	0	0	0	-	-	-	-
R/W:	R	R/W*1	R	R/W*1	R/W*1	R	R	R	R	R/W*1	R/W*1	R/W*1	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.
14	BCHG	0	R/W*1	PORT0 USB Bus Change Interrupt Status*3 This bit is set to 1 when a transition in the full-speed or low-speed signal level occurs on PORT0 (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0). When a BCHG interrupt has occurred, read the LNST bit several times and confirm that the same value is read consecutively in order to prevent chattering. 0: BCHG interrupt has not occurred 1: BCHG interrupt has occurred
13	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12	DTCH	0	R/W*1	PORT0 USB Disconnection Detection Interrupt Status This bit is set to 1 on detecting USB bus disconnection on PORT0. This module detects bus disconnection based on USB Specification 2.0. After detecting a DTCH interrupt, this module performs control described below (irrespective of the setting of the corresponding interrupt enable bit). Terminate all the pipes in which communications on PORT0 are currently carried out and enter a state waiting for bus connection to PORT0 (wait for an ATTCH interrupt). (1) Modifies the UACT bit for PORT0 to 0. (2) Puts PORT0 into the idle state. 0: DTCH interrupt has not occurred 1: DTCH interrupt has occurred
11	ATTCH	0	R/W*1	PORT0 USB Connection Detection Interrupt Status When the host controller function is selected, this module detects a PORT0 ATTACH interrupt on generation of a J-state or K-state of the full-speed or low-speed level signal for 2.5 μs on PORT0, and sets this bit to 1. Detailed detection condition is as follows: (1) Change from a K-state, SEO or SE1 to a J-state, and continuation in the J-state for 2.5 μs (2) Change from a J-state, SEO or SE1 to a K-state, and continuation in the K-state for 2.5 μs 0: ATTCH interrupt has not occurred 1: ATTCH interrupt has occurred
10 to 7	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6	EOFERR	0	R/W*1	PORT0 EOF Error Detection Interrupt Status This bit is set to 1 when the communication on PORT0 does not end at the EOF2 timing defined by USB Specification 2.0. After detecting an EOFERR interrupt, this module performs control described below (irrespective of the setting of the corresponding interrupt enable bit). Terminate all the pipes in which communications on PORT0 are currently carried out and perform re-enumeration of PORT0. (1) Modifies the UACT bit for PORT0 to 0. (2) Puts PORT0 into the idle state. 0: EOFERR interrupt not generated 1: EOFERR interrupt generated
5	SIGN	0	R/W*1	Setup Transaction Error Interrupt Status This bit is set to 1 when ACK response is not returned from the function device three consecutive times during a setup transaction issued by this module. Detailed detection condition is as follows; (1) Timeout of this module before a response is received from the function device (2) An ACK packet was corrupted (3) Reception of an handshake other than an ACK (NAK, NTET, or STALL) 0: SIGN interrupt has not occurred 1: SIGN interrupt has occurred
4	SACK	0	R/W*1	Setup Transaction Normal Response Interrupt Status This bit is set to 1 when ACK response is received from the function device during a setup transaction issued by this module. 0: SACK interrupt has not occurred 1: SACK interrupt has occurred
3 to 0	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Notes: 1. Only 1 can be written to.
2. The interrupts generated by the status transitions indicated by each bit in this register should only be enabled when the host controller function is selected.

3. This module detects the change in the status indicated by the BCHG bit even while the clock supply is stopped (while SCKE is 0), and outputs the corresponding interrupt request as long as it is enabled. Clearing the status should be done after enabling the clock supply.

24.3.21 Interrupt Status Register 2 (INTSTS2)

INTSTS2 indicates the statuses of various interrupts.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BCHG	—	DTCH	ATTCH	—	—	—	—	EOF ERR	—	—	—	—	—	—
Initial value:	-	0	-	0	0	-	-	-	-	0	-	-	-	-	-	-
R/W:	R	R/W*1	R	R/W*1	R/W*1	R	R	R	R	R/W*1	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.
14	BCHG	0	R/W*1	PORT1 USB Bus Change Interrupt Status*3 This bit is set to 1 when a transition in the full-speed or low-speed signal level occurs on PORT1 (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0). When a BCHG interrupt has occurred, read the LNST bit several times and confirm that the same value is read consecutively in order to prevent chattering. 0: BCHG interrupt has not occurred 1: BCHG interrupt has occurred
13	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12	DTCH	0	R/W*1	PORT1 USB Disconnection Detection Interrupt Status This bit is set to 1 on detecting USB bus disconnection on PORT1. This module detects bus disconnection based on USB Specification 2.0. After detecting a DTCH interrupt, this module performs control described below (irrespective of the setting of the corresponding interrupt enable bit). Terminate all the pipes in which communications on PORT1 are currently carried out and enter a state waiting for bus connection to PORT1 (wait for an ATTCH interrupt). (1) Modifies the UACT bit for PORT1 to 0. (2) Puts PORT1 into the idle state. 0: DTCH interrupt has not occurred 1: DTCH interrupt has occurred
11	ATTCH	0	R/W*1	PORT1 USB Connection Detection Interrupt Status When the host controller function is selected, this module detects a PORT1 ATTACH interrupt on generation of a J-state or K-state of the full-speed or low-speed level signal for 2.5 μs on PORT1, and sets this bit to 1. Detailed detection condition is as follows; (1) Change from a K-state, SEO or SE1 to a J-state, and continuation in the J-state for 2.5 μs (2) Change from a J-state, SEO or SE1 to a K-state, and continuation in the K-state for 2.5 μs 0: ATTCH interrupt has not occurred 1: ATTCH interrupt has occurred
10 to 7	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6	EOFERR	0	R/W*1	PORT1 EOF Error Detection Interrupt Status This bit is set to 1 when the communication on PORT1 does not end at the EOF2 timing defined by USB Specification 2.0. After detecting an EOFERR interrupt, this module performs control described below (irrespective of the setting of the corresponding interrupt enable bit). Terminate all the pipes in which communications on PORT1 are currently carried out and perform re-enumeration of PORT1. (1) Modifies the UACT bit for PORT1 to 0. (2) Puts PORT1 into the idle state. 0: EOFERR interrupt not generated 1: EOFERR interrupt generated
5 to 0	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Notes: 1. Only 1 can be written to.
2. The interrupts generated by the status transitions indicated by each bit in this register should only be enabled when the host controller function is selected.
3. This module detects the change in the status indicated by the BCHG bit even while the clock supply is stopped (while SCKE is 0), and outputs the corresponding interrupt request as long as it is enabled. Clearing the status should be done after enabling the clock supply.

24.3.22 BRDY Interrupt Status Register (BRDYSTS)

BRDYSTS is used to confirm the BRDY interrupt status for each pipe. The conditions for generation and clearing of the BRDY interrupt status are different depending on the settings of the BRDYM bit and the BFRE bit for each pipe.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 BRDY	PIPE8 BRDY	PIPE7 BRDY	PIPE6 BRDY	PIPE5 BRDY	PIPE4 BRDY	PIPE3 BRDY	PIPE2 BRDY	PIPE1 BRDY	PIPE0 BRDY
Initial value:	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

(1) When BRDYM = 0 and BFRE = 0 are Set

With this setting, the BRDY interrupt indicates that the FIFO ports have become accessible.

In the event of the following conditions, an internal BRDY interrupt request trigger is generated and the bit corresponding to the pipe that caused the request trigger is set to 1.

(a) Conditions for Pipes in the Transmitting Direction

- The DIR bit is modified from 0 to 1.
- Packet transmission in the pertinent pipe has completed while the FIFO buffer assigned to the pipe is not writable by the CPU (i.e., when 0 is read from the BSTS bit).
  If continuous transmission/reception mode has been set, a request trigger is generated on completion of transmitting the data for one plane of FIFO buffer.
- When the FIFO buffer is configured as double buffers, writing to one buffer has completed while the other buffer is empty. Even if transmission from one FIFO buffer has completed while the other buffer is being written to, a request trigger is not generated until the writing to the buffer is completed.
- Buffer flush by this module has occurred in an isochronous transfer pipe.
- The FIFO buffer has become writable from a non-writable state by writing 1 to the ACLRM bit.

Request triggers are not generated for the DCP (that is, in data transfer by control transfer).

(b) Conditions for Pipes in the Receiving Direction

- Packet reception in the pertinent pipe has completed and the FIFO buffer has become ready for reading while the FIFO buffer assigned to the pipe is not writable by the CPU (i.e., when 0 is read from the BSTS bit).
  A request trigger is not generated for a transaction in which a data PID disagreement has occurred.
  In continuous transmission/reception mode, a request trigger is not generated if the size of data is MaxPacketSize and there is a free space remaining in the buffer.
  When a short packet is received, a request trigger is generated even if there is a free space in the FIFO buffer.
  When a transaction is used, a request trigger is generated when the set number of packets has been received. In this case, a request trigger is generated even when there is some free space in the FIFO buffer.
- When the FIFO buffer is configured as double buffers, reading from a FIFO buffer has completed while the other buffer is also ready to be read. Even if reception into one FIFO buffer has completed while the other buffer is being read, a request trigger is not generated until the reading from the buffer is completed.

The BRDY interrupt is not generated during communication in the status stage of control transfer when the function controller function is selected.

The BRDY interrupt status bit of the pertinent pipe can be cleared by writing 0 to the bit corresponding to the pipe in this register. This interrupt status clearing must be done before making access to the FIFO buffer.

(2) When BRDYM = 0 and BFRE = 1 are Set

With this setting, a BRDY interrupt is generated when all the data for one round of transfer have been read through a receiving pipe and the bit corresponding to that pipe is set.

This module determines that the last data for one round of transfer has been read in any of the following conditions.

(a) Short Packet, Including Zero-Length Packet, has been Received

(b) Packets for the Value Set in Bits TRNCNT have been Received When the Transaction Counter (Bits TRNCNT) is Used

When either of the above conditions is satisfied and reading of that data has been completed, this module determines that all the data for one round of transfer have been read out.

When a zero-length packet is received while the FIFO buffer is empty, the USB 2.0 host/function module determines that all the data for a single transfer has been read at the point at which the FRDY bit is set to 1 and the DTLN bit cleared to 0 in the FIFO port control register. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding FIFOCTR register.

With this setting, the BRDY interrupt is not detected for transmitting pipes.

The BRDY interrupt status bit of the pertinent pipe can be cleared by writing 0 to the bit corresponding to the pipe in this register.

When this mode is used, do not modify the BFRE bit setting until the processing for that round of transfer is completed. To modify the BFRE bit, use the ACLRM bit to clear all the contents of the FIFO buffer of the corresponding pipe.

(3) When BRDYM = 1 and BFRE = 0 are Set

With this setting, the values of the bits in this register reflect the value of the BSTS bit of the individual pipes. This means that the BRDY interrupt statuses are set according to the states of the FIFO buffers.

(a) Condition for Pipes in the Transmitting Direction

When data writing to the FIFO port is possible, the corresponding bit in this register is set to 1. When not possible, the bit is cleared to 0. For the transmitting DCP, however, the BRDY interrupt is not generated even if writing is possible.

(b) Condition for Pipes in the Receiving Direction

When data reading from the FIFO port is possible, the corresponding bit in this register is set to 1. After all the data have been read out (when reading has become not possible), the bit is cleared to 0. If a zero-length packet is received while the FIFO buffer is empty, the corresponding bit is set to 1 until 1 is written to BCLR, during which the BRDY interrupt is continuously generated.

With this setting, the status bits in this register cannot be cleared by writing 0.

When BRDYM = 1 is set, all the BFRE bits (for all pipes) must be cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
9	PIPE9BRDY	0	R/W*	BRDY Interrupt Status for PIPE9 0: Interrupt has not occurred 1: Interrupt has occurred
8	PIPE8BRDY	0	R/W*	BRDY Interrupt Status for PIPE8 0: Interrupt has not occurred 1: Interrupt has occurred
7	PIPE7BRDY	0	R/W*	BRDY Interrupt Status for PIPE7 0: Interrupt has not occurred 1: Interrupt has occurred
6	PIPE6BRDY	0	R/W*	BRDY Interrupt Status for PIPE6 0: Interrupt has not occurred 1: Interrupt has occurred
5	PIPE5BRDY	0	R/W*	BRDY Interrupt Status for PIPE5 0: Interrupt has not occurred 1: Interrupt has occurred
4	PIPE4BRDY	0	R/W*	BRDY Interrupt Status for PIPE4 0: Interrupt has not occurred 1: Interrupt has occurred
3	PIPE3BRDY	0	R/W*	BRDY Interrupt Status for PIPE3 0: Interrupt has not occurred 1: Interrupt has occurred
2	PIPE2BRDY	0	R/W*	BRDY Interrupt Status for PIPE2 0: Interrupt has not occurred 1: Interrupt has occurred
1	PIPE1BRDY	0	R/W*	BRDY Interrupt Status for PIPE1 0: Interrupt has not occurred 1: Interrupt has occurred
0	PIPE0BRDY	0	R/W*	BRDY Interrupt Status for PIPE0 0: Interrupt has not occurred 1: Interrupt has occurred

Note: * Only 0 can be written to these bits.

24.3.23 NRDY Interrupt Status Register (NRDYSTS)

NRDYSTS is used to confirm the NRDY interrupt status for each pipe

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 NRDY	PIPE8 NRDY	PIPE7 NRDY	PIPE6 NRDY	PIPE5 NRDY	PIPE4 NRDY	PIPE3 NRDY	PIPE2 NRDY	PIPE1 NRDY	PIPE0 NRDY
Initial value:	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

When an internal NRDY interrupt request has occurred in the pipe for which PID = BUF is set, the corresponding bit for that pipe is set to 1. An internal NRDY interrupt request is generated for a pipe in the conditions described below. Note, however, that these interrupt generating conditions do not apply to the cases when a setup transaction is being executed while the host controller function is selected. During a setup transaction with the host controller function selected, SACK interrupt or SIGN interrupt can occur. Also note that interrupt requests are not generated during execution of the status stage in control transfer when the function controller function is selected.

(1) Connection with which Split Transactions Do Not Occur when the Host Controller Function is Selected

(a) Conditions for Pipes in the Transmitting Direction

An NRDY interrupt is generated if any of the following conditions is met.

- In a pipe specified for isochronous transfer, when the time to issue an OUT token comes while the FIFO buffer contains no data for transmission:
  In this case, this module transmits a zero-length packet subsequent to the OUT token and sets the bit corresponding to the pipe and the OVRN bit to 1.
- In a pipe not specified for isochronous transfer and executing communication for other than setup transactions, when the function device returns no response (timeout is detected without detecting a handshake packet from the function device), or when any error has detected in the packet from the function device for consecutive three times:
  In this case, this module sets the bit corresponding to the pipe and modifies the PID bit setting for the pipe to STALL.
- When a STALL handshake (not only STALL for OUT token but also STALL for PING token apply) is received from the function device during communication for other than setup transactions:
  In this case, this module sets the bit corresponding to the pipe and modifies the PID bit setting for the pipe to STALL.

(b) Conditions for Pipes in the Receiving Direction

- In a pipe specified for isochronous transfer, when the time to issue an IN token comes while the FIFO buffer has no empty space:
  In this case, this module discards the data received in response to the IN token and sets the bit corresponding to the pipe and the OVRN bit to 1. Further, it also sets the CFRCDE bit to 1 if a packet error has been detected in the data received in response to the IN token.
- In a pipe not specified for isochronous transfer, when the function device returns no response (timeout is detected without detecting a data packet from the function device) to the IN token sent by this module, or when any error has detected in the packet from the function device for consecutive three times:
  In this case, this module sets the bit corresponding to the pipe and modifies the PID bit setting for the pipe to NAK.
- In a pipe specified for isochronous transfer, when the function device returns no response (timeout is detected without detecting a data packet from the function device) to the IN token, or when any error has detected in the packet from the function device:
  In this case, this module sets the bit corresponding to the pipe to 1 (it does not modify the PID bit setting for the pipe).
- In a pipe specified for isochronous transfer, when a CRC error or bit stuffing error has detected in the received data packet:
  In this case, this module sets the bit corresponding to the pipe and the CRCE bit to 1.
- When a STALL handshake is received:
  In this case, this module sets the bit corresponding to the pipe and modifies the PID bit setting for the pipe to STALL.

(2) Connection with which Split Transactions can Occur when the Host Controller Function is Selected

(a) Conditions for Pipes in the Transmitting Direction

- In a pipe specified for isochronous transfer, when the time to issue an OUT token comes while the FIFO buffer contains no data for transmission:
  In this case, this module sets the bit corresponding to the pipe to 1 when it issues a Start-Split transaction (S-Split) and sets the OVRN bit to 1. It also transmits a zero-length packet subsequent to the OUT token.
- In a pipe not specified for isochronous transfer, when the hub returns no response (timeout is detected without detecting a handshake packet from the hub) to the S-Split or Complete-Split transaction (C-Split), or when any error has detected in the packet from the hub for consecutive three times:
  In this case, this module sets the bit corresponding to the pipe and modifies the PID bit setting

for the pipe to NAK. If an NRDY interrupt is detected when a C-Split has been issued, the module clears the CSSTS bit to 0.

- When a STALL handshake is received in response to the C-Split:
  In this case, this module sets the bit corresponding to the pipe, modifies the PID bit setting for the pipe to STALL and clears the CSSTS bit to 0. Note that the NRDY interrupt is not detected in setup transactions.
- In a pipe specified for interrupt transfer, when NYET is received in response to the C-Split for the micro frame number of 4:
  In this case, this module sets the bit corresponding to the pipe to 1 and clears the CSSTS bit to 0 (it does not modify the PID bit setting for the pipe).

(b) Conditions for Pipes in the Receiving Direction

- In a pipe specified for isochronous transfer, when the time to issue an IN token comes while the FIFO buffer has no empty space:
  In this case, this module sets the bit corresponding to the pipe to 1 when it issues S-Split and sets the OVRN bit to 1. It also discards the data received in response to the IN token
- In a pipe specified for bulk transfer or during transfer for other than setup transactions in the DCP, when the hub returns no response (timeout is detected without detecting a data packet from the hub) to the IN token issued by this module at the issuance of S-Split or C-Split, or when any error has detected in the packet from the hub for consecutive three times:
  In this case, this module sets the bit corresponding to the pipe and modifies the PID bit setting for the pipe to NAK. If this condition arise with C-Split, the module also clears the CSSTS bit to 0.
- For C-Split in a pipe specified for isochronous transfer or interrupt transfer, when the hub returns no response (timeout is detected without detecting a data packet from the hub) to the IN token issued by this module, or when any error has detected in the packet from the hub for consecutive three times:
  If this condition arise with a pipe for interrupt transfer, this module sets the bit corresponding to the pipe, modifies the PID bit setting for the pipe to NAK, and clears the CSSTS bit to 0. If this condition arise with a pipe for isochronous transfer, this module sets the bit corresponding to the pipe and the CRCE bit to 1 and clears the CSSTS bit to 0 (it does not modify the PID bit setting for the pipe).
- For C-Split in a pipe not specified for isochronous transfer, when a STALL handshake is received:
  In this case, this module sets the bit corresponding to the pipe, modifies the PID bit setting for the pipe to STALL, and clears the CSSTS bit to 0.
- For C-Split in a pipe specified for isochronous transfer or interrupt transfer, when NYET handshake is received for the micro frame number of 4:

In this case, this module sets the bit corresponding to the pipe and the CRCE bit to 1 and clears the CSSTS bit to 0 (it does not modify the PID bit setting for the pipe).

(3) When the Function Controller Function is Selected

(a) Condition for Pipes in the Transmitting Direction

- When an IN token is received while the FIFO buffer contains no data for transmission:
  In this case, this module generates an NRDY interrupt when it has received the IN token and sets the bit corresponding to the pipe to 1. If the pipe where the interrupt has occurred is for isochronous transfer, this module transmits a zero-length packet and sets the OVRN bit to 1.

(b) Conditions for Pipes in the Receiving Direction

- When an OUT token is received while the FIFO buffer has no empty space:
  In the case of a pipe specified for isochronous transfer, this module generates an NRDY interrupt request when it has received the OUT token and sets the bit corresponding to the pipe and the OVRN bit to 1. In the case of a pipe not specified for isochronous transfer, this module generates an NRDY interrupt request when it issues a NAK handshake after receiving data that comes following the OUT token and sets the bit corresponding to the pipe to 1.
  Note that the NRDY interrupt request is not generated for re-transmission (on a mismatch of data PID). The interrupt request is also not generated when the data packet contains any error.
- When a PING token is received while the FIFO buffer has no empty space:
  In this case, this module generates an NRDY interrupt request when it has received the PING token and sets the bit corresponding to the pipe to 1.
- In a pipe specified for isochronous transfer, when the token is not received normally within the interval frame:
  In this case, this module generates an NRDY interrupt request when it has received SOF and sets the bit corresponding to the pipe to 1.

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
9	PIPE9NRDY	0	R/W*	NRDY Interrupt Status for PIPE9 0: Interrupt has not occurred 1: Interrupt has occurred

Bit	Bit Name	Initial Value	R/W	Description
8	PIPE8NRDY	0	R/W*	NRDY Interrupt Status for PIPE8 0: Interrupt has not occurred 1: Interrupt has occurred
7	PIPE7NRDY	0	R/W*	NRDY Interrupt Status for PIPE7 0: Interrupt has not occurred 1: Interrupt has occurred
6	PIPE6NRDY	0	R/W*	NRDY Interrupt Status for PIPE6 0: Interrupt has not occurred 1: Interrupt has occurred
5	PIPE5NRDY	0	R/W*	NRDY Interrupt Status for PIPE5 0: Interrupt has not occurred 1: Interrupt has occurred
4	PIPE4NRDY	0	R/W*	NRDY Interrupt Status for PIPE4 0: Interrupt has not occurred 1: Interrupt has occurred
3	PIPE3NRDY	0	R/W*	NRDY Interrupt Status for PIPE3 0: Interrupt has not occurred 1: Interrupt has occurred
2	PIPE2NRDY	0	R/W*	NRDY Interrupt Status for PIPE2 0: Interrupt has not occurred 1: Interrupt has occurred
1	PIPE1NRDY	0	R/W*	NRDY Interrupt Status for PIPE1 0: Interrupt has not occurred 1: Interrupt has occurred
0	PIPE0NRDY	0	R/W*	NRDY Interrupt Status for PIPE0 0: Interrupt has not occurred 1: Interrupt has occurred

Note: * Only 0 can be written to these bits.

24.3.24 BEMP Interrupt Status Register (BEMPSTS)

BEMPSTS is used to confirm the BEMP interrupt status for each pipe

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 BEMP	PIPE8 BEMP	PIPE7 BEMP	PIPE6 BEMP	PIPE5 BEMP	PIPE4 BEMP	PIPE3 BEMP	PIPE2 BEMP	PIPE1 BEMP	PIPE0 BEMP
Initial value:	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

When this module detects a BEMP interrupt in the pipe for which PID = BUF is set, the corresponding bit for that pipe is set to 1. This module generates an internal BEMP interrupt request in the following conditions.

(a) Pipe in the Transmitting Direction

An internal BEMP interrupt request is generated in a transmitting pipe when transmission has been completed (including zero-length packet transmission) while the FIFO buffer for the pipe is empty. With a single-buffer configuration, an internal BEMP interrupt request is generated at the same time as the BEMP interrupt for the pipes other than the DCP.

However, an internal BEMP interrupt request is not generated in the following cases.

- In a double-buffer configuration, writing to the FIFO buffer on the CPU side has been started when transmission for a single plane of data is completed.
- Buffer is cleared (has become empty) by writing 1 to the ACLRM or BCLR bit.
- When the function module function has been set, IN transfer (zero-length packet transmission) is executed for the status stage in control transfer.

(b) Pipe in the Receiving Direction

An internal BEMP interrupt request is generated when data of the amount greater than the size set by MaxPacketSize has been received normally. In this case, this module generates a BEMP interrupt request, sets the bit corresponding to the pipe to 1, discards the received data, and modifies the PID bit setting for the pipe to STALL. This module returns no response when the host controller function is selected, and carries out the STALL response when the function controller function is selected.

However, an internal BEMP interrupt request is not generated in the following cases.

- A CRC error or bit stuffing error is detected in the received data.
- A setup transaction is executed.

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
9	PIPE9BEMP	0	R/W*	BEMP Interrupts for PIPE9 0: Interrupt has not occurred 1: Interrupt has occurred
8	PIPE8BEMP	0	R/W*	BEMP Interrupts for PIPE8 0: Interrupt has not occurred 1: Interrupt has occurred
7	PIPE7BEMP	0	R/W*	BEMP Interrupts for PIPE7 0: Interrupt has not occurred 1: Interrupt has occurred
6	PIPE6BEMP	0	R/W*	BEMP Interrupts for PIPE6 0: Interrupt has not occurred 1: Interrupt has occurred
5	PIPE5BEMP	0	R/W*	BEMP Interrupts for PIPE5 0: Interrupt has not occurred 1: Interrupt has occurred
4	PIPE4BEMP	0	R/W*	BEMP Interrupts for PIPE4 0: Interrupt has not occurred 1: Interrupt has occurred

Bit	Bit Name	Initial Value	R/W	Description
3	PIPE3BEMP	0	R/W*	BEMP Interrupts for PIPE3 0: Interrupt has not occurred 1: Interrupt has occurred
2	PIPE2BEMP	0	R/W*	BEMP Interrupts for PIPE2 0: Interrupt has not occurred 1: Interrupt has occurred
1	PIPE1BEMP	0	R/W*	BEMP Interrupts for PIPE1 0: Interrupt has not occurred 1: Interrupt has occurred
0	PIPE0BEMP	0	R/W*	BEMP Interrupts for PIPE0 0: Interrupt has not occurred 1: Interrupt has occurred

Note: * Only 0 can be written to these bits.

24.3.25 Frame Number Register (FRMNUM)

FRMNUM determines the source of isochronous error notification and indicates the frame number.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OVRN	CRCE	—	—	—	FRNM[10:0]										
Initial value:	0	0	-	-	-	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	OVRN	0	R/W*	Overrun/Underrun Detection Status This bit is set when an overrun or underrun has been detected in the pipe during isochronous transfer. On detection of an overrun or underrun, an NRDY interrupt request is generated. For details, see section 24.3.23, NRDY Interrupt Status Register (NRDYSTS). [When the host controller function is selected] This module sets this bit to 1 on any of the following conditions. • For the isochronous transfer pipe in the transmitting direction, the time to issue an OUT token comes before all the transmit data has been written to the FIFO buffer. • For the isochronous transfer pipe in the receiving direction, the time to issue an IN token comes when no FIFO buffer planes are empty. [When the function controller function is selected] This module sets this bit to 1 on any of the following conditions. • For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer. • For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty. 0: No error 1: An error occurred Note: This bit is for debugging. The system should be designed such that overruns or underruns do not occur.

Bit	Bit Name	Initial Value	R/W	Description
14	CRCE	0	R/W*	Receive Data Error This bit is set when a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer. At the same time, an NRDY interrupt request is generated. For details, see section 24.3.23, NRDY Interrupt Status Register (NRDYSTS). 0: No error 1: An error occurred
13 to 11	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
10 to 0	FRNM[10:0]	H'000	R	Frame Number The latest frame number can be confirmed. The frame number is updated every time an SOF packet is issued or received (every 1 ms). Note: When reading these bits, read twice and make sure that the same value is read.

Note: * Only 0 can be written to.

24.3.26 μFrame Number Register (UFRMNUM)

UFRMNUM indicates the μframe number.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	UFRNM[2:0]		
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
2 to 0	UFRNM[2:0]	000	R	μFrame The μframe number can be confirmed. The latest μframe number is read from these bits when at least one of PORT0 and PORT1 is in high-speed operation. When neither of them is in high-speed operation, B'000 is read. Note: When reading these bits, read twice and make sure that the same value is read.

24.3.27 USB Address Register (USBADDR)

USBADDR indicates the USB address. This register is valid only when the function controller function is selected. When the host controller function is selected, function device addresses should be set using the DEVSEL bits in PIPEMAXP.

This register is initialized by a power-on reset or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	USBADDR[6:0]						
Initial value:	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
6 to 0	USBADDR [6:0]	H'00	R	USB Address When the function controller function is selected, the USB address assigned in the SetAddress request from the host is set in these bits.

24.3.28 USB Request Type Register (USBREQ)

USBREQ is a register for storing setup requests for control transfer.

When the function controller function is selected, the values of bRequest and bmRequestType that have been received are stored.

When the host controller function is selected, the values of bRequest and bmRequestType to be transmitted should be set. After setting SUREQ to 1, do not rewrite this register until 0 is read from SUREQ.

This register is initialized by a power-on reset or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	BREQUEST [7:0]	H'00	R/W*	Request These bits store the bRequest value of a USB request.
7 to 0	BMREQUEST-TYPE[7:0]	H'00	R/W*	Request Type These bits store the bmRequestType value of a USB request.

Note: * When the function controller function is selected, these bits are read-only. When the host controller function is selected, these bits can be read and written to.

24.3.29 USB Request Value Register (USBVAL)

USBVAL is a register for storing setup requests for control transfer.

When the function controller function is selected, the value of wValue that has been received is stored.

When the host controller function is selected, the value of wValue to be transmitted should be set. After setting SUREQ to 1, do not rewrite this register until 0 is read from SUREQ.

This register is initialized by a power-on reset or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WVALUE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	WVALUE[15:0]	H'0000	R/W*	Value These bits store the wValue value of a USB request.

Note: * When the function controller function is selected, these bits are read-only. When the host controller function is selected, these bits can be read and written to.

24.3.30 USB Request Index Register (USBINDX)

USBINDEX is a register for storing setup requests for control transfer.

When the function controller function is selected, the value of wIndex that has been received is stored.

When the host controller function is selected, the value of wIndex to be transmitted should be set. After setting SUREQ to 1, do not rewrite this register until 0 is read from SUREQ.

This register is initialized by a power-on reset or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WINDEX[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	WINDEX[15:0]	H'0000	R/W*	Index These bits store the wIndex value of a USB request.

Note: * When the function controller function is selected, these bits are read-only. When the host controller function is selected, these bits can be read and written to.

24.3.31 USB Request Length Register (USBLENG)

USBLENG is a register for storing setup requests for control transfer.

When the function controller function is selected, the value of wLength that has been received is stored.

When the host controller function is selected, the value of wLength to be transmitted should be set. After setting SUREQ to 1, do not rewrite this register until 0 is read from SUREQ.

This register is initialized by a power-on reset or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WLENGTH[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	WLENGTH [15:0]	H'0000	R/W*	Length These bits store the wLength value of a USB request.

Note: * When the function controller function is selected, these bits are read-only. When the host controller function is selected, these bits can be read and written to.

24.3.32 DCP Configuration Register (DCPCFG)

DCPCFG specifies the data transfer direction for the default control pipe (DCP).

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CNTMD	SHT NAK	—	—	DIR	—	—	—	—
Initial value:	-	-	-	-	-	-	-	0	0	-	-	0	-	-	-	-
R/W:	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
8	CNTMD	0	R/W	Continuous Transfer Mode* Specifies whether or not the DCP operates in continuous transfer mode. Based on the setting value of this bit, the USB module determines completion of transmit and receive to and from the FIFO buffer allocated to the DCP as described in table 24.12. 0: Non-continuous transfer mode 1: Continuous transfer mode
7	SHTNAK	0	R/W	Pipe Disable at Transfer-End* Specifies whether or not to change the PID bits to NAK at transfer-end when the DCP is operating in the receive direction. When this bit is set to 1, the USB module changes the PID bits to NAK when it determines that a transfer to the DCP has finished. The USB module determines that transfer-end has occurred when a short packet of data (or a zero-length packet) is received normally. 0: Continue pipe operation at transfer-end 1: Disable pipe operation at transfer-end Note: Clear this bit to 0 when the DCP is operating in the transmit direction.
6, 5	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4	DIR	0	R/W	Transfer Direction When the host controller function is selected, this bit sets the transfer direction of the data stage and status stage. 0: Data receiving direction 1: Data transmitting direction
3 to 0	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Notes: When the function controller function is selected, the DIR bit should be cleared to 0.

* When changing the setting of the DCP's PID bits after they have been changed from BUF to NAK, confirm that both CSSTS and PBUSY are cleared to 0 before changing the PID bits. However, it is not necessary to check the PBUSY bit if the PID bits were changed to NAK by the USB module.

Change the setting of the CNTMD bit when CSSTS = 0, PID = NAK, and the CURPIPE bits are unset. When changing the setting of the CNTMD bit after USB communication using the DCP, write 1 to BCLR and clear the FIFO buffer allocated to the DCP, in addition to ensuring that the above three registers are as described.

Change the setting of the SHTNAK bit when CSSTS = 0 and PID = NAK.

Table 24.12 Setting Value of CNTMD Bit and Method of Determining Completion of Transmit and Receive to and from FIFO Buffer

Setting of CNTMD Bit	Read Enabled State or Transmit Enabled State Determination Method
0	Condition for entering the FIFO buffer read enabled state when operating in the receive direction (DIR = 0): • The USB module has received one packet.
	Condition for entering the FIFO buffer transmit enabled state when operating in the transmit direction (DIR = 1): When either of the following conditions is met (1) Data equal to the maximum packet size has been written to the FIFO buffer. (2) A short packet of data (or a zero-byte packet) has been written to the FIFO buffer and 1 has been written to BVAL.
1	Condition for entering the FIFO buffer read enabled state when operating in the receive direction (DIR = 0): When one of (1) to (3) below is satisfied (1) The number of bytes of data received by the FIFO buffer allocated to the DCP is equal to the allocated byte count (fixed at 256 bytes). (2) A short packet other than a zero-byte packet has been received. (3) The USB module has received a zero-byte packet when there was already data stored in the FIFO buffer allocated to the DCP.
	Condition for entering the FIFO buffer transmit enabled state when operating in the transmit direction (DIR = 1): When either of the following conditions is met (1) The number of bytes of data written is equal to one plane of the FIFO buffer allocated to the DCP. (2) The number of bytes of data written is less than one plane of the FIFO buffer allocated to the DCP (or zero bytes were written) and 1 was written to BVAL.

24.3.33 DCP Maximum Packet Size Register (DCPMAXP)

DCPMAXP specifies the maximum packet size for the DCP.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
Initial value:	0	0	0	0	—	—	—	—	—	1	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	DEVSEL[3:0]	0000	R/W	Device Select*
				When the host controller function is selected, these bits specify the communication target device address. These bits should be set after setting the address in the DEVADDn register that corresponds to the value set in these bits. For example, before setting DEVSEL to 0010, the DEVADD2 register should be set.
				0000: Address 0000
				0001: Address 0001
				: :
				1001: Address 1001
				1010: Address 1010
				1011 to 1111: Setting prohibited
11 to 7	—	Undefined	R	Reserved
				Undefined values are read from these bits. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 0	MXPS[6:0]	H'40	R/W	Maximum Packet Size∗ Specify the maximum packet size for the DCP. Note: Do not set values that do not conform to the USB Specification. While MXPS is 0, do not write to the FIFO buffer or do not set PID to BUF.

Note: ∗ The DEVSEL bit should be set while CSSTS is 0, PID is NAK, and SUREQ is 0.
The MXPS bit should be set while CSSTS is 0 and PID is NAK.
Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY is not necessary.
When the function controller function is selected, the DEVSEL bit should be set to 0.

24.3.34 DCP Control Register (DCPCTR)

DCPCTR is a register that is used to confirm the buffer memory status, control setup transactions and split transactions, change and confirm the data PID sequence bit, and set the response PID for the DCP.

This register is initialized by a power-on reset. The CCPL and PID[2:0] bits are initialized by a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	SUREQ	CSCLR	CSSTS	SUREQ CLR	—	—	SQCLR	SQSET	SQMON	PBUSY	PINGE	—	CCPL	PID[1:0]	
Initial value:	0	0	0	0	0	-	-	0	0	1	0	0	-	0	0	0
R/W:	R	R/W∗2	R∗1/ W∗2	R	R∗1/ W∗2	R	R	R∗1/ W∗2	R∗1/ W∗2	R	R	R/W	R	R/W	R/W	R/W

Notes: 1. This bit is always read as 0.
2. Only 1 can be written to.

Bit	Bit Name	Initial Value	R/W	Description
15	BSTS	0	R	Buffer Status Indicates whether DCP FIFO buffer access is enabled or disabled. The direction of access, reading or writing, is determined by the ISEL bit in CFIFOSEL. 0: Buffer access is disabled. 1: Buffer access is enabled.
14	SUREQ	0	R/W*2	Setup Token Transmission Setting this bit to 1 transmits a setup packet. After completing the setup transaction process, this module generates either the SACK or SIGN interrupt and clears this bit to 0. Before setting this bit to 1, set the DEVSEL bits, USBREQ register, USBVAL register, USBINDX register, and USBLENG register appropriately to transmit the desired USB request in the setup transaction. Before setting this bit to 1, check that the PID bits for the DCP are set to NAK. 0: Invalid 1: Transmits the setup packet. Note: After setting this bit to 1, do not modify the DEVSEL bits, USBREQ register, USBVAL register, USBINDX register, or USBLENG register until the setup transaction is completed (SUREQ = 1). Write 1 to this bit only when transmitting the setup token; for the other purposes, write 0.

Bit	Bit Name	Initial Value	R/W	Description
13	CSCLR	0	R*1/W*2	C-Split Status Clear for Split Transaction When the host controller function is selected, setting this bit to 1 clears the CSSTS bit to 0. This bit should be set to 1 to restart the next transfer with S-Split forcibly in transfer using the split transaction. However, in normal split transactions, the CSSTS bit is automatically cleared to 0 upon completion of the C-Split; therefore, processing for clearing the CSSTS bit is not necessary. 0: No effect 1: Clears the CSSTS bit to 0. Note: Controlling the CSSTS bit through this bit must be done while UACT is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected. Setting this bit to 1 while CSSTS is 0 has no effect.
12	CSSTS	0	R	COMPLETE SPLIT (C-Split) Status of Split Transaction Indicates the C-Split status of the split transaction when the host controller function is selected. This bit is set to 1 upon start of the C-Split and cleared to 0 upon detection of C-Split completion. 0: START-SPLIT (S-Split) transaction being processed or processing of a device not using split transaction is in progress 1: C-Split transaction being processed

Bit	Bit Name	Initial Value	R/W	Description
11	SUREQCLR	0	R*[1]/W*[2]	SUREQ Bit Clear When the host controller function is selected, setting this bit to 1 clears the SUREQ bit to 0. Set this bit to 1 when communication has stopped with SUREQ being 1 during the setup transaction. However, in normal setup transactions, the SUREQ bit is automatically cleared to 0 upon completion of the transaction; therefore, processing for clearing the SUREQ bit is not necessary. 0: No effect 1: Clears the SUREQ bit to 0. Note: Controlling the SUREQ bit through this bit must be done while UACT is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected.
10, 9	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
8	SQCLR	0	R*[1]/W*[2]	Toggle Bit Clear*[3] Specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. 0: No effect 1: Specifies DATA0. Note: Do not set the SQCLR and SQSET bits to 1 simultaneously.

Bit	Bit Name	Initial Value	R/W	Description
7	SQSET	0	R*1/W*2	Toggle Bit Set*3 Specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. 0: No effect 1: Specifies DATA1. Note: Do not set the SQCLR and SQSET bits to 1 simultaneously.
6	SQMON	1	R	Sequence Toggle Bit Monitor Indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer. This bit toggles upon normal completion of the transaction. However, this bit does not toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction. When the function controller function is selected, this bit is set to 1 (specifies DATA1 as the expected value) upon normal reception of the setup packet. When the function controller function is selected, this module does not reference to this bit during the IN/OUT transaction of the status stage, and does not allow this bit to toggle upon normal completion. 0: DATA0 1: DATA1

Bit	Bit Name	Initial Value	R/W	Description
5	PBUSY	0	R	Pipe Busy This bit indicates whether the relevant pipe is used or not for the transaction. 0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction. The USB 2.0 host/function module switches the PBUSY bit from 0 to 1 when a USB transaction using the pertinent pipe starts. It clears the PBUSY bit from 1 to 0 when one transaction completes successfully. Reading this bit after PID has been set to NAK allows checking that modification of the pipe settings is possible. For details, see 25.4.3 (1), Pipe Control Register Modifying Procedures.
4	PINGE	0	R/W	PING Token Issue Enable*3 When the host controller function is selected, setting this bit to 1 allows this module to issue the PING token during transfers in the transmitting direction and start a transfer in the transmitting direction with the PING transaction. When having detected the ACK handshake during a PING transaction, this module performs an OUT transaction as the next transaction. When having detected the NAK handshake during a OUT transaction, this module performs a PING transaction as the next transaction. When the host controller function is selected, clearing this bit to 0 prevents this module from issuing the PING token during transfers in the transmitting direction and all transfers in the transmitting direction are performed as OUT transactions. 0: Disables issuing PING token. 1: Enables normal PING operation.

Bit	Bit Name	Initial Value	R/W	Description
3	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.
2	CCPL	0	R/W	Control Transfer End Enable When the function controller function is selected, setting this bit to 1 while the corresponding PID bits are set to BUF allows the control transfer stages to be completed. Specifically, during control read transfer, this module transmits the ACK handshake in response to the OUT transaction from the USB host, and outputs a zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, this module operates in auto response mode from the setup stage until completion of the status stage, irrespective of the setting of this bit. This bit is cleared from 1 to 0 on receiving the new setup packet. 0: No effect 1: Completion of control transfer is enabled. Note: While VALID is 1, writing of 1 to this bit is disabled.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	Response PID Controls the responses of this module during control transfer. These bits should be modified from NAK to BUF when the data stage or status stage is executed in control transfer. [When the host controller function is selected] Modify the setting of these bits from NAK to BUF using the following procedure. • When the transmitting direction is set Write all the transmit data to the FIFO buffer while UACT is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, this module executes the OUT transaction (or PING transaction). • When the receiving direction is set Check that the FIFO buffer is empty (or empty the buffer) while UACT is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, this module executes the IN transaction. This module modifies the setting of these bits in the following conditions. • This module sets PID to STALL if it receives data exceeding the maximum packet size when PID is set to BUF. • This module sets PID to NAK on detecting a receive error such as a CRC error three consecutive times. • This module also sets PID to STALL on receiving the STALL handshake.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	Even if the PID bits are modified to NAK after this module has issued S-Split of the split transaction for the specified pipe (while CSSTS indicates 1), this module continues the transaction until C-Split is completed. [When the function controller function is selected] This module modifies the setting of these bits in the following conditions. • This module modifies PID to NAK on receiving a setup packet. Here, this module sets VALID to 1, and the setting of PID cannot be changed until VALID is cleared to 0. • This module sets PID to STALL if it receives data exceeding the maximum packet size when PID is set to BUF. • This module sets PID to STALL on detecting a control transfer sequence error. • This module sets PID to NAK on detecting the USB bus reset. This module does not reference to the PID bits while it is processing a SET_ADDRESS request (auto processing). 00: NAK response 01: BUF response (depends on the buffer state) 10: STALL response 11: STALL response

Notes: When the function controller function is selected, bits SUREQ, CSCLR, CSSTS, SUREQCLR, and PINGE should be cleared to all 0s.

When the host controller function is selected, bit CCPL should be cleared to 0.

1. This bit is always read as 0.
2. Only 1 can be written to.
3. These bits should be modified while CSSTS is 0 and PID is NAK. Before modifying this bit after modifying the PID bits from BUF to NAK, make sure that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY is not necessary.

24.3.35 Pipe Window Select Register (PIPESEL)

PIPESEL selects the pipe for use from among PIPE1 to PIPE9. After a pipe has been selected, configure the functions of the individual pipe using the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers. The PIPEnCTR, PIPEnTRE, and PIPEnTRN registers can be set regardless of the pipe selection in PIPESEL.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
3 to 0	PIPESEL[3:0]	0000	R/W	Pipe Window Select When a value from 0001 to 1001 is set in these bits, the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers indicate the information and setting values of the selected pipe. 0000: No pipe selected 0001: PIPE1 0010: PIPE2 0011: PIPE3 0100: PIPE4 0101: PIPE5 0110: PIPE6 0111: PIPE7 1000: PIPE8 1001: PIPE9 Other than above: Setting prohibited Note: When PIPESEL = 0000, the PIPECFG, PIPEBUF, PIPEMAXP, PIPEERI and PIPEnCTR registers indicate all 0s and writing to these registers is ignored.

24.3.36 Pipe Configuration Register (PIPECFG)

PIPECFG is a register that specifies the transfer type, buffer memory access direction, and endpoint numbers for PIPE1 to PIPE9. It also selects continuous or non-continuous transfer mode, single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TYPE[1:0]		—	—	—	BFRE	DBLB	CNTMD	SHT NAK	—	—	DIR	EPNUM[3:0]			
Initial value:	0	0	-	-	-	0	0	0	0	-	-	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	TYPE[1:0]	00	R/W	Transfer Type* Selects the transfer type for the pipe selected by the PIPESEL bits (selected pipe) • PIPE1 and PIPE2 00: Pipe disabled 01: Bulk transfer 10: Setting prohibited 11: Isochronous transfer • PIPE3 to PIPE5 00: Pipe disabled 01: Bulk transfer 10: Setting prohibited 11: Setting prohibited • PIPE6 to PIPE9 00: Pipe disabled 01: Setting prohibited 10: Interrupt transfer 11: Setting prohibited Note: Before setting PID to BUF, be sure to set these bits to the value other than 00.

Bit	Bit Name	Initial Value	R/W	Description
13 to 11	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
10	BFRE	0	R/W	BRDY Interrupt Operation Specification* Specifies the BRDY interrupt generation timing for the selected pipe. This bit is valid when any one of PIPE1 to PIPE5 is selected. When this bit is set to 1 and the selected pipe is in the receiving direction (DIR = 0), this module detects the transfer completion and generates a BRDY interrupt on having read the pertinent packet. When the BRDY interrupt is generated with the above conditions, writing of 1 to BCLR is necessary. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to BCLR. When this bit is set to 1 and the selected pipe is in the transmitting direction (DIR = 1), this module does not generate the BRDY interrupt. 0: BRDY interrupt upon data transmission or reception 1: BRDY interrupt upon data reading
9	DBLB	0	R/W	Double Buffer Mode* Selects either single or double buffer mode for the FIFO buffer used by the selected pipe. This bit is valid when any one of PIPE1 to PIPE5 is selected. When this bit is set to 1, this module assigns two planes of the FIFO buffer size specified by the BUFSIZE bits in PIPEBUF to the selected pipe. Specifically, the FIFO buffer size assigned to the selected pipe is calculated by the following expression. (BUFSIZE + 1) × 64 × (DBLB + 1) [bytes] 0: Single buffer 1: Double buffer

Bit	Bit Name	Initial Value	R/W	Description
8	CNTMD	0	R/W	Continuous Transfer Mode* Specifies whether to use the selected pipe in continuous transfer mode. This bit is valid when any one of PIPE1 to PIPE5 is selected and also bulk transfer is selected. This module determines the completion of transmission or reception for the FIFO buffer assigned to the selected pipe as shown in table 24.13 according to the setting of this bit. 0: Non-continuous transfer mode 1: Continuous transfer mode
7	SHTNAK	0	R/W	Pipe Disable at End of Transfer* Specifies whether to modify PID to NAK upon the end of transfer when the selected pipe is in the receiving direction. This bit is valid when the selected pipe is one of PIPE1 to PIPE5 and is in the receiving direction. When this bit is set to 1 for the pipe in the receiving direction, this module modifies the PID bits corresponding to the selected pipe to NAK on determining the end of the transfer. This module determines that the transfer has ended in either of the following conditions. • A short packet (including a zero-length packet) is successfully received. • The transaction counter is used and the number of packets specified by the counter is successfully received. 0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer Note: This bit should be cleared to 0 for the pipe in the transmitting direction.
6, 5	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4	DIR	0	R/W	Transfer Direction*
				Specifies the transfer direction for the selected pipe.
				0: Receiving direction
				1: Sending direction
3 to 0	EPNUM[3:0]	0000	R/W	Endpoint Number*
				These bits specify the endpoint number for the selected pipe. Setting 0000 means unused pipe.
				Note: Do not make the settings such that the combination of the set values in the DIR and EPNUM bits are the same for two or more pipes (EPNUM = 0000 can be set for multiple pipes).

Notes: Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, make sure that CSSTS = 0 and PBUSY = 0. However, if the PID bits have been modified to NAK by this module, checking of PBUSY is not necessary.

* Modify the TYPE bit while the PID bit using the selected pipe is NAK.

  Modify the BFRE, DBLB, CNTMD, and DIR bits while CSSTS is 0, PID is NAK, and the CURPIPE bits has not been set. When modifying the settings of bits after USB communication is done using the selected pipe, successively write 1 and then 0 to ACLRM to clear the FIFO buffer assigned to the selected pipe, in addition to satisfying the conditions of the above mentioned three registers.

  Modify the SHTNAK bit while CSSTS is 0 and PID is NAK.

  Modify the EPNUM bit while CSSTS is 0, PID is NAK, and the CURPIPE bit has not been set.

Table 24.13 CNTMD Bit Setting and Method of Determining Completion of Transmission/Reception for the FIFO Buffer

CNTMD Bit Setting	Method of Determining Readiness for Reading and Transmission
0	Condition for the FIFO buffer being ready for reading when set for receiving direction (DIR = 0): • This controller receives one packet.
	Conditions for the FIFO buffer being ready for transmission when set for transmitting direction (DIR = 1): When either of the following conditions is met (1) Data equal to the maximum packet size is written to the FIFO buffer. (2) A short packet of data (or a zero-byte packet) is written to the FIFO buffer and 1 is written to BVAL.
1	Conditions for the FIFO buffer being ready for reading when set for receiving direction (DIR = 0): When one of (1) to (4) below is satisfied (1) The number of received data bytes in the FIFO buffer assigned to the selected pipe has become equal to the allocated number of bytes ([BUFSIZE + 1) × 64). (2) This controller receives a short packet other than a zero-length packet. (3) This controller receives a zero-length packet when data is already stored in the FIFO buffer assigned to the selected pipe. (4) This controller has received packets of the number of times set in the transaction counter for the selected pipe.
	Conditions for the FIFO buffer being ready for transmission when set for transmitting direction (DIR = 1): When one of (1) to (3) below is satisfied (1) The number of data bytes written to the FIFO buffer has become equal to the size of one plane of the FIFO buffer assigned to the selected pipe. (2) Data of the size smaller than the size of one plane of the FIFO buffer assigned to the selected pipe (including 0 bytes) is written to the FIFO buffer and 1 is written to BVAL. (3) During DMA transfer operation with the DMA transfer-end sampling enable bit (TENDE) set to 1, the number of bytes of data written is less than one plane of the FIFO buffer allocated to selected pipe (or zero bytes are written), and a DMA transfer-end signal is received while the last write is in progress.

24.3.37 Pipe Buffer Setting Register (PIPEBUF)

PIPEBUF is specifies the buffer size and buffer number for PIPE1 to PIPE9.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BUFSIZE[4:0]					—	—	BUFNMB[7:0]							
Initial value:	-	0	0	0	0	0	-	-	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
14 to 10	BUFSIZE[4:0]	H'00	R/W	Buffer Size* Specifies the size of the buffer for the selected pipe in number of blocks, where one block comprises 64 bytes. When the DBLB bit is set to 1, this module assigns two planes of the FIFO buffer size specified by the BUFSIZE bits to the selected pipe. Specifically, the FIFO buffer size assigned to the selected pipe is calculated by the following expression. (BUFSIZE + 1) × 64 × (DBLB + 1) [bytes] The settable values for these bits depends on the pipe selected by the PIPESEL bits of the PIPESEL register. • PIPE1 to PIPE5: Any value from H'00 to H'1F is valid. • PIPE6 to PIPE9: H'00 should be set. Note: When used with CNTMD = 1, set an integer multiple of the maximum packet size to the BUFSIZE bits.
9, 8	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BUFNMB[7:0]	H'00	R/W	Buffer Number*
				Specify the block number of the first block of the FIFO buffer to be assigned to the selected pipe. The blocks to be assigned to the selected buffer are shown below.
				Block number: BUFNMB + (BUFSIZE + 1) × (DBLB + 1) – 1
				A value from H'04 to H'9F can be set in these bits under the following conditions.
				BUFNMB = H'00 to H'03 are used exclusively by the DCP.
				BUFNMB = H'04 is used exclusively by PIPE6.
				When PIPE6 is not used, H'04 can be used for other pipes.
				When PIPE6 is selected, writing to these bits is ignored and H'04 is automatically assigned to PIPE6 by this module.
				BUFNMB = H'05 is used exclusively by PIPE7.
				When PIPE7 is not used, H'05 can be used for other pipes.
				When PIPE7 is selected, writing to these bits is ignored and H'05 is automatically assigned to PIPE7 by this module.
				BUFNMB = H'06 is used exclusively by PIPE8.
				When PIPE8 is not used, H'06 can be used for other pipes.
				When PIPE8 is selected, writing to these bits is ignored and H'06 is automatically assigned to PIPE8 by this module.
				BUFNMB = H'07is used exclusively by PIPE9.
				When PIPE9 is not used, H'07 can be used for other pipes.
				When PIPE9 is selected, writing to these bits is ignored and H'07 is automatically assigned to PIPE9 by this module.

Notes: Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, make sure that CSSTS = 0 and PBUSY = 0. However, if the PID bits have been modified to NAK by this module, checking of PBUSY is not necessary.

* Modify these bits while CSSTS is 0, PID is NAK, and the CURPIPE bit has not been set.

24.3.38 Pipe Maximum Packet Size Register (PIPEMAXP)

PIPEMAXP sets the function device address when the host controller function is selected and specifies the maximum packet size for PIPE1 to PIPE9.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEVSEL[3:0]				—	MXPS[10:0]										
Initial value:	0	0	0	0	-	*2	*2	*2	*2	*2	*2	*2	*2	*2	*2	*2
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	DEVSEL[3:0]	0000	R/W	Device Select*[1] When the host controller function is selected, these bits specify the device address of the function device. These bits should be set after setting the address to the DEVADDn register corresponding to the value to be set in these bits. For example, before setting DEVSEL to 0010, the address should be set to the DEVADD2 register. 0000: Address 0000 0001: Address 0001 : : 1001: Address 1001 1010: Address 1010 1011 to 1111: Setting prohibited
11	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 0	MXPS[10:0]	*2	R/W	Maximum Packet Size*1 Specifies the maximum data payload (maximum packet size) for the selected pipe. Set the value as follows. PIPE1, PIPE2: 1 byte (H'001) to 1,024 bytes (H'400) PIPE3 to PIPE5: 8 bytes (H'008), 16 bytes (H'010), 32 bytes (H'020), 64 bytes (H'040), and 512 bytes (H'200) PIPE6 to PIPE9: 1 byte (H'001) to 64 bytes (H'040) Note: Set the value for compliant with the USB standard every transfer type. When the communication performed on the selected pipe is isochronous transfer with split transactions, these bits should be set to 188 bytes or less. While MXPS is 0, do not write to the FIFO buffer or set PID to BUF.

Notes: When the function controller function is selected, the DEVSEL bit is cleared to 0.

1. The DEVSEL bit should be set while CSSTS = 0 and PID = NAK.
   Modify the MXPS bit while CSSTS is 0 and PID is NAK and before The CURPIPE bit is set.
   Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, make sure that CSSTS = 0 and PBUSY = 0. However, if the PID bits have been modified to NAK by this module, checking of PBUSY is not necessary.
2. The initial value of MXPS is H'000 when no pipe is selected with the PIPESEL bits in PIPESEL and H'040 when a pipe is selected with the PIPESEL bit in PIPESEL.

24.3.39 Pipe Timing Control Register (PIPEPERI)

PIPEPERI is a register that selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfer, and sets the interval error detection interval for PIPE1 to PIPE9.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Initial value:	-	-	-	0	-	-	-	-	-	-	-	-	-	0	0	0
R/W:	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
12	IFIS	0	R/W	Isochronous IN Buffer Flush When the function controller function is selected while Isochronous transfer and direction are IN transfer, this module automatically clears the FIFO buffer when this module fails to receive the IN token from the USB host within the interval set by the IITV bits in terms of (micro) frames. In double buffer mode (DBLB = 1), this module only clears the data in the plane used earlier. This module clears the FIFO buffer on receiving the SOF packet immediately after the (micro) frame in which this module has expected to receive the IN token. Even if the SOF packet is corrupted, this module also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation. 0: The buffer is not flushed. 1: The buffer is flushed. Note: When the selected pipe is not for the isochronous transfer, set this bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
11 to 3	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
2 to 0	IITV[2:0]	000	R/W	Interval Error Detection Interval* Specifies the interval error detection timing for the selected pipe in terms of frames, which is expressed as n-th power of 2 (n is the value to be set). As described later, the detailed functions are different in host controller mode and in function controller mode. Note: Before modifying these bits after USB communication has been completed with these bits set to a certain value, set PID to NAK and then set ACLRM to 1 to initialize the interval timer. The IITV bits are invalid for PIPE3 to PIPE5; set 0 in the bits corresponding to these pipes.

Notes: When the host controller function is selected, set the IFIS bit to 0.

* Modify these bits while CSSTS is 0 and PID is NAK and before the CURPIPE bit is selected.

Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, make sure that CSSTS = 0 and PBUSY = 0. However, if the PID bits have been modified to NAK by this module, checking of PBUSY is not necessary.

The interval error detection interval bits (IITV) are explained below.

(1) When the Host Controller Function is Selected

The IITV bits can be set when the selected pipe is specified for isochronous transfer or interrupt transfer. This module controls the intervals of token issuance according to the setting of this bit. It issues a token for the selected pipe once in 2^{IITV} (micro) frames.

This module counts 1-ms frames to measure the interval for the pipe used for communication with a full-speed/low-speed function device connected to a high-speed hub.

This module starts counting for token issuance intervals from the first (micro) frame that comes after the PID bits have been set to BUF.

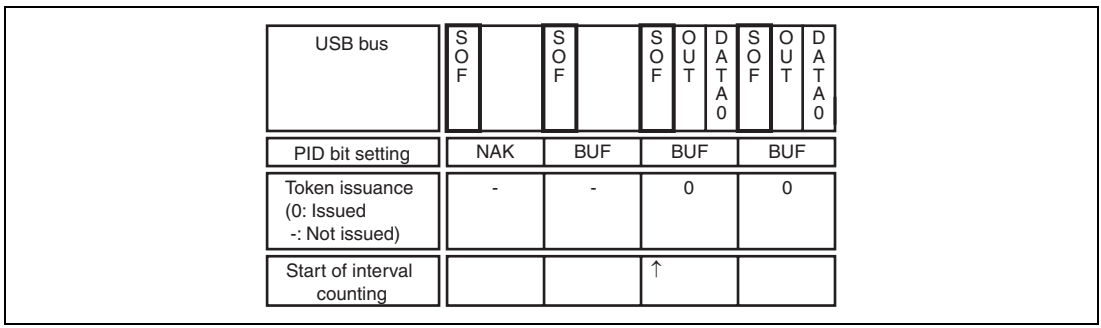


Figure 24.1 Whether a Token is Issued or Not when IITV = 0

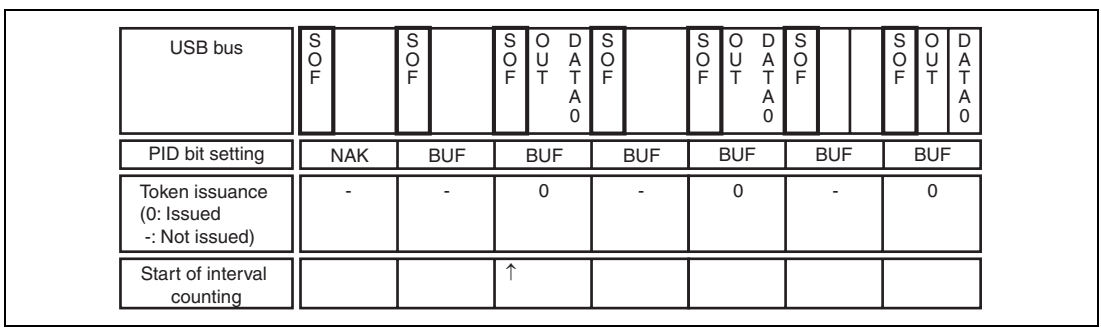


Figure 24.2 Whether a Token is Issued or Not when IITV = 1

When the selected pipe is for isochronous transfer, this module performs the following operations while it controls the token issuance intervals. The module issues tokens even when the condition for NRDY interrupt generation is met.

- When the selected pipe is for isochronous transfer and set in the IN direction:
  The module issues an IN token and, if it does not receive packets from the function device normally (i.e., no response or packet error), generates an NRDY interrupt.
  If the time to issue an IN token comes in a situation where the module is unable to receive data because of the full FIFO buffer, which may be caused by slow or delayed data reading from the buffer or some other reasons, the module sets the OVRN bit to 1 and generates an NRDY interrupt.

- When the selected pipe is for isochronous transfer and set in the OUT direction:
  If the time to issue an OUT token comes in a situation where the FIFO buffer does not contain any data ready for transmission because, for example, data writing to the FIFO buffer is slow or delayed, the module sets the OVRN bit to 1, generates an NRDY interrupt, and transmits a zero-length packet.

The interval of token issuance is reset by a power-on reset or when the ACLRM bit is set to 1.

(2) When the Function Controller Function is Selected

- When the selected pipe is for isochronous transfer and set in the OUT direction:
  If the module does not receive any data packets within the (micro) frames for the interval specified by the IITV bits, it generates an NRDY interrupt.
  The module also generates an NRDY interrupt when it cannot receive a data packet normally because any error, such as a CRC error, is found in the incoming data packet or when the module is unable to receive data because of the full FIFO buffer, which may be caused by slow or delayed data reading from the buffer or some other reasons.
  An NRDY interrupt is generated with the timing of receiving an SOF packet. Even if the SOF packet is corrupted, the interrupt is generated with the proper timing of SOF reception by the internal interpolating function. Note that in the cases other than IITV = 0, an NRDY interrupt is generated upon SOF packet reception at every interval after the counting of the interval is started.
  If the PID bits are set to NAK by software after the interval timer is started, this module does not generate an NRDY interrupt even when it receives an SOF packet.

The conditions for starting the interval timer differ according to the IITV bit setting.

(a) When IITV = 0

Counting of the interval is started from the first (micro) frame after the PID bit setting of the selected pipe is changed to BUF.

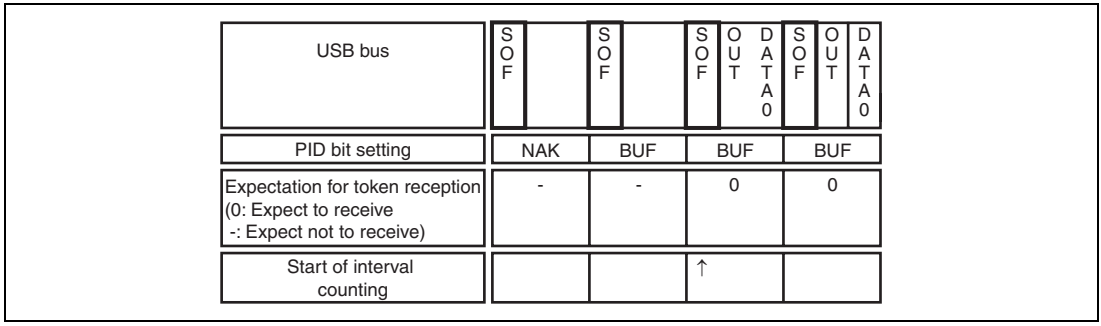


Figure 24.3 (Micro) Frames and Expectation for Token Reception when IITV = 0

(b) When Other than IITV = 0

Counting of the interval is started on completion of the first normal reception of data packet after the PID bit setting of the selected pipe is changed to BUF.

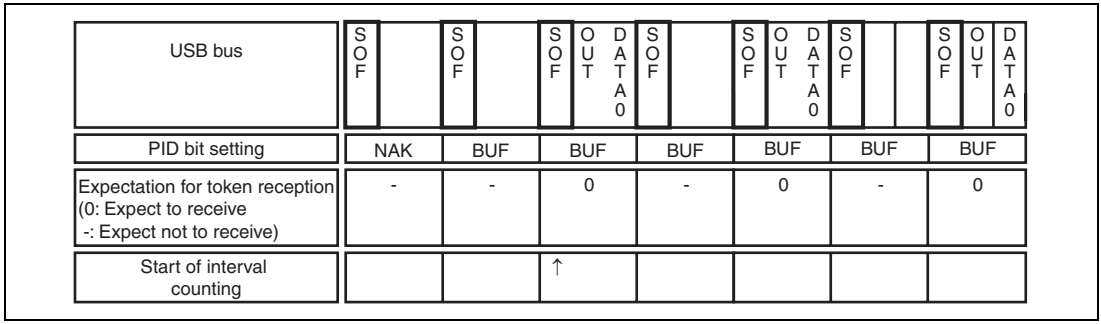


Figure 24.4 (Micro) Frames and Expectation for Token Reception when IITV = 1

- When the selected pipe is for isochronous transfer and set in the IN direction:

  The IITV bits should be used in combination with IFIS = 1. With IFIS = 0, the module transmits data packets in response to the received tokens regardless of the IITV bit setting. When IFIS = 1 is set, the module clears the FIFO buffer if it does not receive any IN token within the (micro) frames for the interval specified by the IITV bits.

  The module also clears the FIFO buffer when it cannot receive normally because a bus error, such as a CRC error, has occurred in the IN token.

  The FIFO buffer is cleared with the timing of receiving an SOF packet. Even if the SOF packet is corrupted, the FIFO buffer is cleared with the proper timing of SOF reception by the internal interpolating function.

  The clearing conditions for starting the interval timer differ according to the IITV bit setting (same as the case with the OUT direction).

  (a) Power-on reset

  (b) When ACLRM = 1 is set

  (c) When USB bus reset is detected

24.3.40 PIPEn Control Registers (PIPEnCTR) (n = 1 to 5)

The PIPEnCTR registers for PIPE1 to PIPE5 are used to confirm the buffer memory status, change and confirm the data PID sequence bit, determine whether auto response mode is set, determine whether auto buffer clear mode is set, and set a response PID for the corresponding pipe. These registers can be set regardless of the pipe selection in PIPESEL.

This register is initialized by a power-on reset. PID[1:0] are initialized by a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Initial value:	0	0	0	0	—	0	0	0	0	0	0	—	—	—	0	0
R/W:	R	R	R*1/W*2	R	R	R/W	R/W	R*1/W*2	R*1/W*2	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSTS	0	R	Buffer Status Indicates whether the FIFO buffer assigned to the corresponding pipe is accessible from the CPU. The meaning of this bit depends on the settings of the DIR, BFRE, and DCLRM bits as shown in table 24.13. 0: Buffer not accessible 1: Buffer accessible
14	INBUFM	0	R	IN Buffer Monitor When the pertinent pipe is in the transmitting direction (DIR = 1), this bit indicates 1 when writing of data to at least one FIFO buffer plane is completed. This bit indicates 0 when this module completes transmitting the data in the FIFO buffer plane to which all the data has been written. In double buffer mode (DBLB = 1), this bit indicates 0 when this module completes transmitting the data in the two FIFO buffer planes before writing of data to one FIFO buffer plane is completed. This bit indicates the same value as the BSTS bit when the pertinent pipe is in the receiving direction (DIR = 0). 0: There is no data to be transmitted in the buffer memory. 1: There is data to be transmitted in the buffer memory.

Bit	Bit Name	Initial Value	R/W	Description
13	CSCLR	0	R*[1]/W*[2]	C-Split Status Clear Bit Set this bit to 1 to clear the CSSTS bit of the pertinent pipe. When the host controller function is selected, setting this bit to 1 allows this module to clear the CSSTS bit to 0. For the transfer using the split transaction, to restart the next transfer with the S-Split forcibly, set this bit to 1. However, for the normal split transaction, this module automatically clears the CSSTS bit to 0 upon completion of the C-Split; therefore, processing for clearing the CSSTS bit is not necessary. Controlling the CSSTS bit through this bit must be done while UACT is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected. 0: Writing invalid 1: Clears the CSSTS bit to 0. Note: Setting this bit to 1 while CSSTS is 0 has no effect.
12	CSSTS	0	R	CSSTS Status Bit Indicates the C-Split status of the split transaction of the pertinent pipe when the host controller function is selected. This bit indicates 1 upon start of the C-Split and indicates 0 upon detection of C-Split completion. 0: S-Split transaction being processed or the transfer not using the split transaction in progress 1: C-Split transaction being processed
11	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10	ATREPM	0	R/W	Auto Response Mode*3 Enables or disables auto response mode for the pertinent pipe. When the function controller function is selected and the pertinent pipe is set for bulk transfer, this bit can be set to 1. When this bit is set to 1, this module responds to the token from the USB host as described below. (1) When the pertinent pipe is for bulk IN transfer (TYPE = 01 and DIR = 1) When ATREPM = 1 and PID = BUF, this module transmits a zero-length packet in response to the IN token. This module updates (allows toggling of) the sequence toggle bit (DATA-PID) each time this module receives the ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.). In this case, this module does not generate the BRDY or BEMP interrupt. (2) When the pertinent pipe is for bulk OUT transfer (TYPE = 01 and DIR = 0) When ATREPM = 1 and PID = BUF, this module returns NAK in response to the OUT (or PING) token and generates the NRDY interrupt. 0: Auto response disabled 1: Auto response enabled Note: For USB communication in auto response mode, set this bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode. When the pertinent pipe is for isochronous transfer, be sure to set this bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
9	ACLRM	0	R/W	Auto Buffer Clear Mode*3 Enables or disables automatic buffer clear mode for the pertinent pipe. To clear the information in the FIFO buffer assigned to the pertinent pipe completely, write 1 and then 0 to this bit successively. Table 24.14 (1) lists the information cleared in this module by successive writing of 0 and 1 to this bit, and table 24.14 (2) lists the cases in which clearing of the information is necessary. 0: Disabled 1: Enabled (all buffers are initialized)
8	SQCLR	0	R*1/W*2	Toggle Bit Clear*3 This bit should be set to 1 to clear the expected value of the sequence toggle bit for the next transaction of the pertinent pipe to DATA0. When the host controller function is selected, setting this bit to 1 for the pipe for bulk OUT transfer, this module starts the next transfer of the pertinent pipe with the PING token. 0: No effect 1: Specifies DATA0.
7	SQSET	0	R*1/W*2	Toggle Bit Set*3 This bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the pertinent pipe. 0: No effect 1: Specifies DATA1.

Bit	Bit Name	Initial Value	R/W	Description
6	SQMON	0	R	Toggle Bit Confirmation Indicates the expected value of the sequence toggle bit for the next transaction of the pertinent pipe. When the pertinent pipe is not for the isochronous transfer, this bit toggles upon normal completion of the transaction. However, this bit does not toggle when a DATA-PID disagreement occurs during the receiving transfer. 0: DATA0 1: DATA1
5	PBUSY	0	R	Pipe Busy Indicates whether the pertinent pipe is currently used for the transaction. This bit changes from 0 to 1 upon start of the USB transaction for the pertinent pipe, and changes from 1 to 0 upon normal completion of one transaction. Reading this bit after setting PID to NAK allows confirming that modification of the pipe settings has become possible. 0: The pertinent pipe is not currently used for the transaction. 1: The pertinent pipe is currently used for the transaction.
4 to 2	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
1, 0	PID[1:0]	00	R/W	Response PID Specifies the response type for the next transaction of the pertinent pipe. The default setting of these bits is NAK. Modify the setting to BUF to use the pertinent pipe for USB transfer. Tables 24.15 and 24.16 show the basic operation (operation when there are no errors in the transmitted and received packets) of this module for different PID bit settings. Even if the PID bits are modified to NAK after this module has issued S-Split of the split transaction for the specified pipe (while CSSTS indicates 1), this module continues the transaction until C-Split is completed.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	This module modifies the setting of these bits as follows. • Sets PID = NAK on recognizing the completion of the transfer when the pertinent pipe is in the receiving direction and software has set the SHTNAK bit for the selected pipe to 1. • Sets PID = STALL (11) on receiving the data packet with the payload exceeding the maximum packet size of the pertinent pipe. • Sets PID = NAK on detecting a USB bus reset when the function controller function is selected. • Sets PID = NAK on detecting a receive error such as a CRC error three consecutive times when the host controller function is selected. • Sets PID = STALL (11) on receiving the STALL handshake when the host controller function is selected. 00: NAK response 01: BUF response (depending on the buffer state) 10: STALL response 11: STALL response Note: After modifying the setting of these bits from BUF to NAK during USB communication using the pertinent pipe, check that PBUSY is 0 to see if USB communication using the pertinent pipe has actually entered the NAK state. However, if the PID bits have been modified to NAK by this module, checking of PBUSY is not necessary. To make a transition from PID = NAK (00) to STALL, set 10. To make a transition from PID = BUF (01) to STALL, set 11. To make a transition from PID = STALL (11) to NAK, set 10 and then 00. To make a transition from PID = STALL (11) to BUF, set 00 and then 01.

Notes: When the function controller function is selected, clear the CSCLR bit to 0. And when the host controller function is selected, clear the ATREPM bit to 0.

1. Only 0 can be read.
2. Only 1 can be written to.
3. Modify ATREPM, SZCLR, and SQSET bits while CSSTS is 0 and PID is NAK. Modify the ACLRM bit while CSSTS is 0, PID is NAK, and before the CURPIPE bit is selected.

   Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, make sure that CSSTS = 0 and PBUSY = 0. However, if the PID bits have been modified to NAK by this module, checking of PBUSY is not necessary.

Table 24.14 Meaning of BSTS Bit

DIR Bit	BFRE Bit	DCLRM Bit	Meaning of BSTS Bit
0	0	0	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer before setting BCLR to 1.
		1	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.
1	0	0	1: The transmit data can be written to the FIFO buffer. 0: The transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

Table 24.15 (1) Information Cleared by this Module by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation
1	All the information in the FIFO buffer assigned to the pertinent pipe (all the information in two FIFO buffer planes in double buffer mode)
2	The interval count value when the pertinent pipe is for isochronous transfer

Table 24.15 (2) Cases That Require Setting ACLRM to 1

No.	Cases in which Clearing the Information is Necessary
1	When it is necessary to clear all the information assigned to the pertinent pipe from the FIFO buffer
2	When the interval count value is to be reset
3	When the BFRE setting is modified
4	When the DBLB setting is modified
5	When the transaction count function is forcibly terminated

Table 24.16 Operation of This Module depending on PID Setting (when Host Controller Function is Selected)

PID Bits	Transfer Type (TYPE Bits)	Transfer Direction (DIR Bit)	Operation of This Module
00 (NAK)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.
01 (BUF)	Bulk or interrupt	Operation does not depend on the setting.	Issues tokens while UACT is 1 and the FIFO buffer corresponding to the pertinent pipe is ready for transmission and reception. Does not issue tokens while UACT is 0 or the FIFO buffer corresponding to the pertinent pipe is not ready for transmission or reception.
	Isochronous	Operation does not depend on the setting.	Issues tokens irrespective of the status of the FIFO buffer corresponding to the pertinent pipe.
10 (STALL) or 11 (STALL)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.

Table 24.17 Operation of This Module depending on PID Setting (when Function Controller Function is Selected)

PID Bits	Transfer Type (TYPE Bits)	Transfer Direction (DIR Bit)	Operation of This Module
00 (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host.
	Isochronous	Receiving direction (DIR = 0)	Returns nothing in response to a token from the USB host.
		Transmitting direction (DIR = 1)	Returns a zero-length packet in response to a token from the USB host.
01 (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the pertinent pipe is ready for reception. Returns NAK if not ready. Returns ACK in response to the PING token from the USB host if the FIFO buffer corresponding to the pertinent pipe is ready for reception. Returns NYET if not ready.
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the pertinent pipe is ready for reception. Returns NAK if not ready.

PID Bits	Transfer Type (TYPE Bits)	Transfer Direction (DIR Bit)	Operation of This Module
01 (BUF)	Bulk or interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
	Isochronous	Receiving direction (DIR = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer corresponding to the pertinent pipe is ready for reception. Discards data if not ready.
		Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Transmits the zero-length packet if not ready.
10 (STALL) or 11 (STALL)	Bulk or interrupt	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting	Returns nothing in response to the token from the USB host.

24.3.41 PIPEn Control Registers (PIPEnCTR) (n = 6 to 9)

The PIPEnCTR registers for PIPE6 to PIPE9 are used to confirm the buffer memory status, change and confirm the data PID sequence bit, determine whether auto buffer clear mode is set, and set a response PID for the corresponding pipe. These registers can be set regardless of the pipe selection in PIPESEL.

This register is initialized by a power-on reset. PID[1:0] are initialized by a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Initial value:	0	-	0	0	-	-	0	0	0	0	0	-	-	-	0	0
R/W:	R	R	R*1/ W*2	R	R	R	R/W	R*1/ W*2	R*1/ W*2	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSTS	0	R	Buffer Status Indicates whether the FIFO buffer assigned to the corresponding pipe is accessible from the CPU. The meaning of this bit depends on the settings of the DIR, BFRE, and DCLRM bits as shown in table 24.14. 0: Buffer not accessible 1: Buffer accessible
14	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
13	CSCLR	0	R*[1]/W*[2]	C-Split Status Clear Bit Set this bit to 1 to clear the CSSTS bit of the pertinent pipe. When the host controller function is selected, setting this bit to 1 allows this module to clear the CSSTS bit to 0. For the transfer using the split transaction, to restart the next transfer with the S-Split forcibly, set this bit to 1. However, for the normal split transaction, this module automatically clears the CSSTS bit to 0 upon completion of the C-Split; therefore, processing for clearing the CSSTS bit is not necessary. Controlling the CSSTS bit through this bit must be done while UACT is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected. 0: Writing invalid 1: Clears the CSSTS bit to 0. Note: Setting this bit to 1 while CSSTS is 0 has no effect.
12	CSSTS	0	R	CSSTS Status Bit Indicates the C-Split status of the split transaction of the pertinent pipe when the host controller function is selected. This bit indicates 1 upon start of the C-Split and indicates 0 upon detection of C-Split completion. If the USB device is detached when processing of a C-SPLIT transaction is in progress, the CSSTS bit may remain set to 1. In such a case (DTCH = 1 detected), use the CSCLR bit to clear the CSSTS bit to 0. 0: S-Split transaction being processed or the transfer not using the split transaction in progress 1: C-Split transaction being processed
11, 10	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	ACLRM	0	R/W	Auto Buffer Clear Mode*3 Enables or disables automatic buffer clear mode for the pertinent pipe. To clear the information in the FIFO buffer assigned to the pertinent pipe completely, write 1 and then 0 to this bit successively. Table 24.18 (1) lists the information cleared in this module by successive writing of 0 and 1 to this bit, and table 24.18 (2) lists the cases in which clearing of the information is necessary. 0: Disabled 1: Enabled (all buffers are initialized)
8	SQCLR	0	R*1/W*2	Toggle Bit Clear*3 This bit should be set to 1 to clear the expected value of the sequence toggle bit for the next transaction of the pertinent pipe to DATA0. When the host controller function is selected, setting this bit to 1 for the pipe for bulk OUT transfer, this module starts the next transfer of the pertinent pipe with the PING token. 0: No effect 1: Specifies DATA0.
7	SQSET	0	R*1/W*2	Toggle Bit Set*3 This bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the pertinent pipe. 0: No effect 1: Specifies DATA1.

Bit	Bit Name	Initial Value	R/W	Description
6	SQMON	0	R	Toggle Bit Confirmation Indicates the expected value of the sequence toggle bit for the next transaction of the pertinent pipe. When the pertinent pipe is not for the isochronous transfer, this bit toggles upon normal completion of the transaction. However, this bit does not toggle when a DATA-PID disagreement occurs during the receiving transfer. 0: DATA0 1: DATA1
5	PBUSY	0	R	Pipe Busy Indicates whether the pertinent pipe is currently used for the transaction. This bit changes from 0 to 1 upon start of the USB transaction for the pertinent pipe, and changes from 1 to 0 upon normal completion of one transaction. Reading this bit after setting PID to NAK allows confirming that modification of the pipe settings has become possible. 0: The pertinent pipe is not currently used for the transaction. 1: The pertinent pipe is currently used for the transaction.
4 to 2	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
1, 0	PID[1:0]	00	R/W	Response PID Specifies the response type for the next transaction of the pertinent pipe. The default setting of these bits is NAK. Modify the setting to BUF to use the pertinent pipe for USB transfer. Tables 24.15 and 24.16 show the basic operation (operation when there are no errors in the transmitted and received packets) of this module for different PID bit settings. Even if the PID bits are modified to NAK after this module has issued S-Split of the split transaction for the specified pipe (while CSSTS indicates 1), this module continues the transaction until C-Split is completed.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	This module modifies the setting of these bits as follows. • Sets PID = NAK on recognizing the completion of the transfer when the pertinent pipe is in the receiving direction and software has set the SHTNAK bit for the selected pipe to 1. • Sets PID = STALL (11) on receiving the data packet with the payload exceeding the maximum packet size of the pertinent pipe. • Sets PID = NAK on detecting a USB bus reset when the function controller function is selected. • Sets PID = NAK on detecting a receive error such as a CRC error three consecutive times when the host controller function is selected. • Sets PID = STALL (11) on receiving the STALL handshake when the host controller function is selected. 00: NAK response 01: BUF response (depending on the buffer state) 10: STALL response 11: STALL response Note: After modifying the setting of these bits from BUF to NAK during USB communication using the pertinent pipe, check that PBUSY is 0 to see if USB communication using the pertinent pipe has actually entered the NAK state. However, if the PID bits have been modified to NAK by this module, checking of PBUSY is not necessary. To make a transition from PID = NAK (00) to STALL, set 10. To make a transition from PID = BUF (01) to STALL, set 11. To make a transition from PID = STALL (11) to NAK, set 10 and then 00. To make a transition from PID = STALL (11) to BUF, set 00 and then 01.

Notes: When the function controller function is selected, clear the CSCLR bit to 0.

1. Only 0 can be read.
2. Only 1 can be written to.
3. Modify the ACLRM bit while CSSTS is 0, PID is NAK, and before the CURPIPE bit is selected. Modify the SQCLR and SQSET bits while CSSTS is 0 and PID is NAK.
   Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, make sure that CSSTS = 0 and PBUSY = 0. However, if the PID bits have been modified to NAK by this module, checking of PBUSY is not necessary.

Table 24.18 (1) Information Cleared by this Module by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation
1	All the information in the FIFO buffer assigned to the pertinent pipe
2	When the host controller function is selected, the interval count value when the pertinent pipe is for isochronous transfer

Table 24.18 (2) Cases That Require Setting ACLRM to 1

No.	Cases in which Clearing the Information is Necessary
1	When it is necessary to clear all the information assigned to the pertinent pipe from the FIFO buffer
2	When the interval count value is to be reset
3	When the BFRE setting is modified
4	When the transaction count function is forcibly terminated

24.3.42 Transaction Counter Enable Registers (PIPEnTRE) (n = 1 to 5)

The PIPEnTRE registers configure transaction counter operations for PIPE1 to PIPE5. These registers can be set regardless of the pipe selection in PIPESEL.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Initial value:	-	-	-	-	-	-	0	0	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R/W	R*1/ W*2	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
9	TRENB	0	R/W	Transaction Counter Enable*3 Enables or disables the transaction counter. For the pipe in the receiving direction, setting this bit to 1 after setting the total number of the packets to be received in the TRNCNT bits allows this module to control hardware as described below on having received the number of packets equal to the set value in the TRNCNT bits. • In continuous transmission/reception mode (CNTMD = 1), this module switches the FIFO buffer to the CPU side even if the FIFO buffer is not full on completion of reception. • While SHTNAK is 1, this module modifies the PID bits to NAK for the corresponding pipe on having received the number of packets equal to the set value in the TRNCNT bits. • While BFRE is 1, this module asserts the BRDY interrupt on having received the number of packets equal to the set value in the TRNCNT bits and then reading out the last received data. 0: The transaction counter is disabled. 1: The transaction counter is enabled. Note: For the pipe in the transmitting direction, set this bit to 0. When the transaction counter is not used, set this bit to 0. When the transaction counter is used, set the TRNCNT bits before setting this bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.

Bit	Bit Name	Initial Value	R/W	Description
8	TRCLR	0	R*[1]/W*[2]	Transaction Counter Clear*[3] Setting this bit to 1 clears the transaction counter to 0. 0: No effect 1: The current counter is cleared.
7 to 0	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Notes: 1. Only 0 can be read.
2. Only 1 can be written to.
3. Modify each bit when CSSTS = 0 and PID = NAK.
Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, make sure that CSSTS = 0 and PBUSY = 0. However, if the PID bits have been modified to NAK by this module, checking of PBUSY is not necessary.

24.3.43 Transaction Counter Registers (PIPEnTRN) (n = 1 to 5)

The PIPEnTRN registers are used to specify the number of transactions by DMA transfer for PIPE1 to PIPE5, and the current number of transactions can be read from them.

These registers are initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRNCNT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TRNCNT[15:0]	H'0000	R/W	Transaction Counter*
				After the total number of packets to be received is set in these bits for a receiving pipe, setting the TRENB bit to 1 allows this module to perform the control explained in the description of the TRENB bit.
				Indicates the specified number of transactions if TRENB = 0.
				Indicates the current value of the transaction counter if TRENB = 1.
				This module increments the value of TRNCNT by one when all of the following conditions (a) to (c) are satisfied on receiving the packet.
				(a) TRENB is 1.
				(b) (TRNCNT set value ≠ current counter value + 1) on receiving the packet.
				(c) The payload of the received packet agrees with the set value in the MXPS bits.
				This module clears the value of these bits to 0 when any of the following conditions are satisfied.
				1. All of conditions (a) to (c) are satisfied. (a) TRENB is 1. (b) (TRNCNT set value = current counter value + 1) on receiving the packet. (c) The payload of the received packet agrees with the set value in the MXPS bits.
				2. Both of conditions (a) and (b) are satisfied. (a) TRENB is 1. (b) This module has received a short packet.
				3. The TRCLR bit has been set to 1 when the following conditions are satisfied.
				When written: Specifies the number of transactions to be transferred by DMA.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TRNCNT[15:0]	H'0000	R/W	When read: Indicates the specified number of transactions if TRENB = 0. Indicates the current value of the transaction counter if TRENB = 1. Note: For the pipe in the transmitting direction, set these bits to 0. When the transaction counter is not used, set these bits to 0. To modify the value of these bits, set TRCLR to 1 before setting TRENB to 1.

Note: * Modify these bits while CSSTS = 0, PID = NAK, and TRENB = 0.
Before modifying these bits after modifying the PID bits for the corresponding pipe from BUF to NAK, make sure that CSSTS = 0 and PBUSY = 0. However, if the PID bits have been modified to NAK by this module, checking of PBUSY is not necessary.

24.3.44 Device Address Configuration Registers (DEVADDn) (n = 0 to 9, A)

When the host controller function is selected, the DEVADDn registers specify the address and port number of the hub to which the communication target function device is connected and also specifies the communication speed of the function device.

These registers are initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	UPPHUB[3:0]				HUBPORT[2:0]			USBSPD[1:0]		—	—	—	—	—	RTPORT
Initial value:	—	0	0	0	0	0	0	0	0	0	—	—	—	—	—	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	Undefined	R	Reserved Undefined value is read from this bit. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14 to 11	UPPHUB[3:0]	0000	R/W	Address of Hub to which Communication Target is Connected Specifies the USB address of the hub to which the communication target function device is connected. When the host controller function is selected, this module refers to the setting of these bits to generate packets for split transactions. 0000: The function device is directly connected to the port of this LSI. 0001 to 1010: USB address of the hub 1011 to 1111: Reserved
10 to 8	HUBPORT[2:0]	000	R/W	Port Number of Hub to which Communication Target is Connected Specifies the port number of the hub to which the communication target function device is connected. When the host controller function is selected, this module refers to the setting of these bits to generate packets for split transactions. 000: The function device is directly connected to the port of this LSI. 001 to 111: Port number of the hub
7, 6	USBSPD[1:0]	00	R/W	Transfer Speed of the Communication Target Device Specifies the USB transfer speed of the communication target function device. When the host controller function is selected, this module refers to the setting of these bits to generate packets. 00: DEVADDn is not used. 01: Low speed 10: Full speed 11: High speed
5 to 1	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	RTPORT	0	R/W	Root Hub Port Number of the Communication Target Tree Specifies the port number of the port of this module to which the communication target tree is connected (root hub port number). When the host controller function is selected, this module refers to the setting of this bit to generate packets. 0: PORT0 1: PORT1

Notes: 1. When the host controller function is selected, be sure to set the bits in this register before starting the communication on each pipe.

2. The settings of the bits in this register should be modified when there is no active pipe using the setting of these bits. The active pipe refers to a pipe that satisfies both of the following conditions:
(1) The DEVSEL bit setting is designating this register.
(2) The PID bits of the pipe are set to BUF, or the pipe is the DCP and SUREQ = 1.

3. When the function controller function is selected, all the bits in this register should be set to 0.

24.3.45 USB AC Characteristics Switching Register 0 (USBACSWR0)

The USBACSWR0 registers specify a USB transceiver that is stored in this module. This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	UACS14	—	—	—	—	—	—	—	—	UACS5	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R

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	UACS14	0	R/W	USB AC Characteristics Switching 14 This bit adjusts the cross point power supply voltage in low speed.*
13 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	UACS5	0	R/W	USB AC Characteristics Switching 5 This bit adjusts the disconnection power supply voltage.*
4 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * This bit should always be 1 when using this module.
For details, refer to section 24.5.1, Procedure for Setting the USB Transceiver.

24.3.46 USB AC Characteristics Switching Register 1 (USBACSWR1)

The USBACSWR1 registers specify a USB transceiver that is stored in this module. This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	UACS26	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R

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	UACS26	0	R/W	USB AC Characteristics Switching 26 This bit adjusts the cross point power supply voltage in full speed. *
9 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * This bit should always be 1 when using this module.
For details, refer to section 24.5.1, Procedure for Setting the USB Transceiver.

24.4 Operation

24.4.1 System Control

This section describes the register operations that are necessary to the initial settings of this module, and the registers necessary for power consumption control.

(1) Resets

Table 24.19 lists the types of controller resets. For the initialized states of the registers following the reset operations, see section 24.3, Register Description.

Table 24.19 Types of Reset

Name	Operation
Power-on reset	Low level input from the $\overline{\text{RES}}$ pin
USB bus reset	Automatically detected by this module from the D+ and D− lines when the function controller function is selected

(2) Controller Function Selection

This module can select the host controller function or function controller function using the DCFM bit in SYSCFG0.

This controller selects functions for each USB port as shown in table 24.19.

Table 24.20 Functions Selected for USB Ports

When Host Mode is Selected (DCFM = 1)		
PORT0	**PORT1**	**Remarks**
High speed	High speed	Transfer scheduling is common to ports 0 and 1. The outputs of ports 0 and 1 are driven separately.
Full speed, low speed	Full speed, low speed	
High speed	Full speed, low speed	Transfer scheduling runs separately for ports 0 and 1, irrespective of their transfer rates.
Full speed, low speed	High speed	
When Function Mode is Selected (DCFM = 0)		
PORT0	**PORT1**	**Remarks**
High speed, full speed	Not used	PORT1 is disabled, and low speed is not supported.

(3) USB Data Bus Resistor Control

This module controls switching of the pull-up resistor for the D+ signal and a pull-down resistor for the D+ and D- signals. These signals can be pulled up or down using the DPRPU and DRPD bits in SYSCFG0 (for PORT0) and DRPD bit in SYSCFG1 (for PORT1).

When the function controller function is selected, set the DPRPU bit in the SYSCFG register to 1 after the connection to the USB host is recognized to pull up the D+ signal. When disconnection from the USB host is recognized, manipulate the DPRPU and DCFM bits as follows:

1. Clear the DPRPU bit to 0.
2. Wait for at least 1 μs.
3. Set the DCFM bit to 1.
4. Wait for at least 200 ns.
5. Clear the DCFM bit to 0.

This module includes the terminal resistor for the D+ and D- signals during high-speed operation and the output resistor for the signals during full-speed operation. This module automatically switches the resistor after connection with the USB host or function device when reset handshake, suspended state or resume is detected.

When the function controller function is selected and the DPRPU bit in SYSCFG0 is cleared to 0 during communication with the USB host, the pull-up resistor (or the terminal resistor) of the USB data line is disabled. This allows notification of device disconnection to the USB host.

(4) Register Access Wait Control

The following restrictions apply to numbers of cycles for access to registers of this module below SYSSTS0.
The cycle time for consecutive access to registers of this module must be at least 4 USB clock (48 MHz) cycles (83.33 ns).

To comply with this constraint, the BWIT[3:0] bits of the SYSCFG1 register must be set to apply wait control for register access. Since the initial value is the largest value (17 clock cycles for a cycle of access), select the optimal value.

Example of settings (1): Consecutive access to registers of this module
Bus-clock frequency: 66 MHz
Calculation: (2 cycles (access cycle for registers of this module) + 1 cycle (interval between consecutive access operations) + BWAIT) × 1/66 MHz ≥ 83.33 ns
BWAIT = 3

Example of settings (2): Transfer of data from internal memory to the FIFO port registers
Bus-clock frequency: 66 MHz
Calculation: (2 cycles (access cycle for registers of this module) + 2 cycles (access cycle for internal memory) + BWAIT) × 1/66 MHz ≥ 83.33 ns
BWAIT = 2

24.4.2 Interrupt Functions

(1) Interrupt Control Overview

Table 24.21 lists the interrupt generation conditions for this module.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, this module outputs the USB interrupt request signal to the INTC.

Table 24.21 Interrupt Generation Conditions

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
VBINT	VBUS interrupt	When a change in the state of the VBUS input pin has been detected (low to high or high to low)	Host, function	VBSTS
RESM	Resume interrupt	When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)	Function	—
SOFR	Frame number update interrupt	When the host controller function is selected: • When an SOF packet with a different frame number has been transmitted When the function controller function is selected: • When an SOF packet with a different frame number is received	Host, function	—

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
DVST	Device state transition interrupt	When a device state transition is detected • A USB bus reset detected • The suspend state detected • Set address request received • Set configuration request received	Function	DVSQ
CTRT	Control transfer stage transition interrupt	When a stage transition is detected in control transfer • Setup stage completed • Control write transfer status stage transition • Control read transfer status stage transition • Control transfer completed • A control transfer sequence error occurred	Function	CTSQ
BEMP	Buffer empty interrupt	• When transmission of all of the data in the buffer memory has been completed • When an excessive maximum packet size error has been detected	Host, Function	PIPEBEMP

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
NRDY	Buffer not ready interrupt	When the host controller function is selected: • When STALL is received from the function side to the issued token • When the response from the function side to the issued token could not be received correctly (no response or three successive packet receive errors). • When an overrun/underrun occurred during isochronous transfer When the function controller function is selected: • When a token is received while the PID bits are set to BUF and transmission is not enabled for the buffer memory • When a CRC error or a bit stuffing error occurred during data reception in isochronous transfer • When an interval error occurred during data reception in isochronous transfer	Host, function	NRDYSTS. PIPENRDY
BRDY	Buffer ready interrupt	When the buffer is ready (reading or writing is enabled)	Host, function	BRDYSTS. PIPENRDY
BCHG	Bus change interrupt	When a change of USB bus state is detected	Host	—
DTCH	Device disconnection detection	When disconnection of a USB function is detected	Host	—
ATTCH	Device connection detection	When J-state or K-state is detected on the USB port for 2.5 μs. This interrupt can be used to detect whether a USB function is connected.	Host	—

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
EOFERR	EOF error detection	When an EOF error of a USB function is detected	Host	—
SACK	Normal setup operation	When the normal response (ACK) for the setup transaction is received	Host	—
SIGN	Setup error	When a setup transaction error (no response or ACK packet corruption) is detected consecutively three times	Host	—

Note: All the bits without register name indication are in INTSTS0.

Figure 24.5 shows a diagram relating to interrupts of this module.

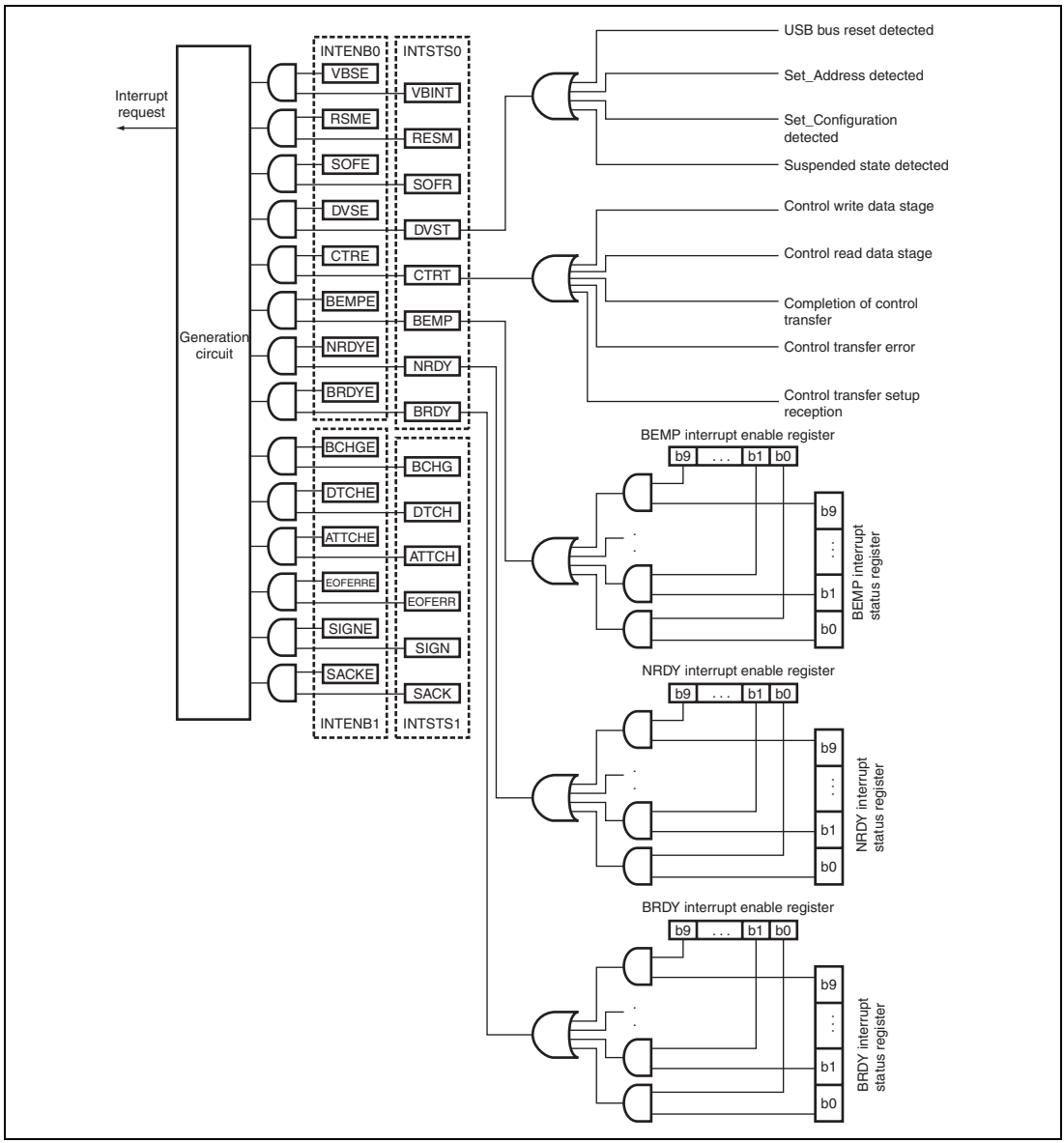


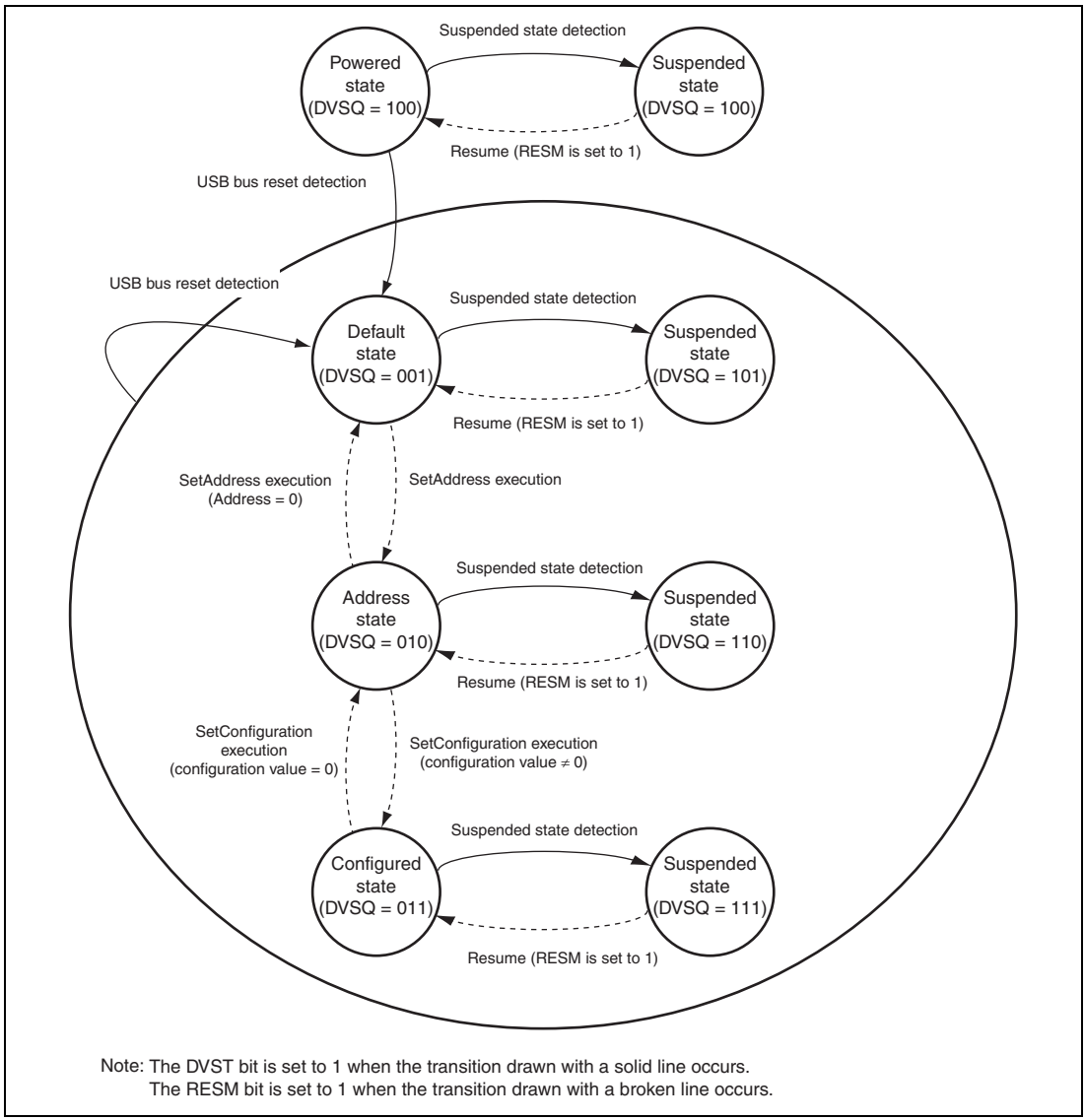
Figure 24.5 Items Relating to Interrupts

(2) Device State Transition Interrupt (Function Controller Function)

Figure 24.6 shows a diagram of this module device state transitions. This module controls device states and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually using INTENB0. The device state that made a transition can be confirmed using the DVSQ bit in INTSTS0.

To make a transition to the default state, the device state transition interrupt is generated after the reset handshake protocol has been completed.

Device state can be controlled only when the function controller function is selected. Also, the device state transition interrupts can be generated only when the function controller function is selected.



Note: The DVST bit is set to 1 when the transition drawn with a solid line occurs.
The RESM bit is set to 1 when the transition drawn with a broken line occurs.

Figure 24.6 Device State Transitions

(3) Control Transfer Stage Transition Interrupt (Function Controller Function)

Figure 24.7 shows a diagram of how this module handles the control transfer stage transition. This module controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled individually using INTENB0. The transfer stage that made a transition can be confirmed using the CTSQ bit in INTSTS0.

Control transfer stage transition interrupts are only generated when the function controller function is selected.

The control transfer sequence errors are described below. If an error occurs, the PID bit in DCPCTR is set to B'1x (STALL).

1. During control read transfers
   - At the IN token of the data stage, an OUT or PING token is received when there have been no data transfers at all.
   - An IN token is received at the status stage
   - A packet is received at the status stage for which the data packet is DATAPID = DATA0
2. During control write transfers
   - At the OUT token of the data stage, an IN token is received when there have been no ACK response at all
   - A packet is received at the data stage for which the first data packet is DATAPID = DATA0
   - At the status stage, an OUT or PING token is received
3. During no-data control transfers
   - At the status stage, an OUT or PING token is received

At the control write transfer stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (SERR = 1), the CTSQ = 110 value is retained until CTRT = 0 is written from the system (the interrupt status is cleared). Therefore, while CTSQ = 110 is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (This module retains the setup stage end, and after the interrupt status has been cleared by software, a CTRT interrupt is generated.)

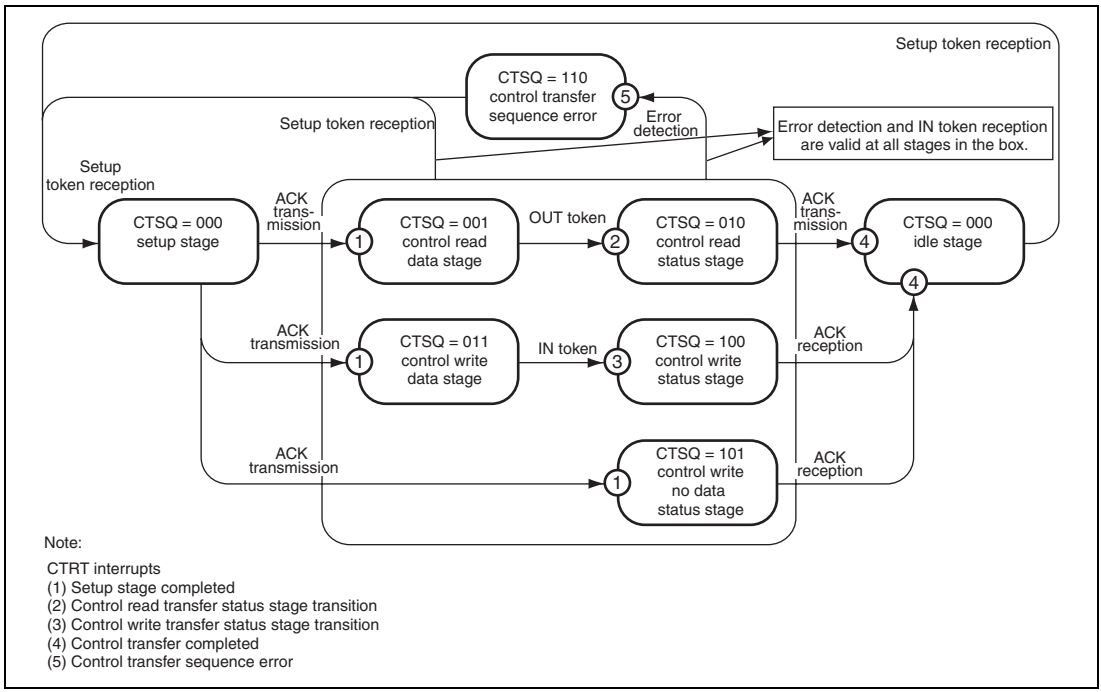


Figure 24.7 Control Transfer Stage Transitions

24.4.3 Pipe Control

Table 24.22 lists the pipe setting items of this module. With USB data transfer, data transmission has to be carried out using the logic pipe called the endpoint. This module has ten pipes that are used for data transfer.

Settings should be entered for each of the pipes in conjunction with the specifications of the system.

Table 24.22 Pipe Setting Items

Register Name	Bit Name	Setting Contents	Remarks
DCPCFG PIPECFG	TYPE	Specifies the transfer type	PIPE1 to PIPE9: Can be set
	BFRE	Selects the BRDY interrupt mode	PIPE1 to PIPE5: Can be set
	DBLB	Selects a single buffer or double buffer	PIPE1 to PIPE5: Can be set
	CNTMD	Selects continuous transfer or non-continuous transfer	DCP: Can be set PIPE1 and PIPE2: Can be set (only when bulk transfer has been selected). PIPE3 to PIPE5: Can be set With continuous transmission and reception, the buffer size should be set to an integer multiple of the payload.
	DIR	Selects transfer direction (reading or writing)	IN or OUT can be set
	EPNUM	Endpoint number	PIPE1 to PIPE9: Can be set Should be set to a non-zero value when using pipes.
	SHTNAK	Selects disabled state for pipe when transfer ends	DCP: Can be set PIPE1 and PIPE2: Can be set (only when bulk transfer has been selected) PIPE3 to PIPE5: Can be set

Register Name	Bit Name	Setting Contents	Remarks
PIPEBUF	BUFSIZE	Buffer memory size	DCP: Cannot be set (fixed at 256 bytes) PIPE1 to PIPE5: Can be set (a maximum of 2 Kbytes in 64-byte units can be specified) PIPE6 to PIPE9: Cannot be set (fixed at 64 bytes)
	BUFNMB	Buffer memory number	DCP: Cannot be set (areas fixed at H'0 to H'3) PIPE1 to PIPE5: Can be set (can be specified in areas H'6 to H'9F) PIPE6 to PIPE9: Cannot be set (areas fixed at H'4 to H'7)
DCPMAXP PIPEMAXP	DEVSEL	Device select	Referenced only when the host controller function is selected.
	MXPS	Maximum packet size	Compliant with the USB standard
PIPEPERI	IFIS	Buffer flush	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE9: Cannot be set
	IITV	Interval counter	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE5: Cannot be set PIPE6 and PIPE9: Can be set (only when host controller function has been selected)
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Only provided for PIPE3 to PIPE5.
	SUREQ	SETUP request	Can be set only for the DCP. Can be controlled only when the host controller function has been selected.
	SUREQCLR	SUREQ clear	Can be set only for the DCP. Can be controlled only when the host controller function has been selected.
	CSCLR	CSSTS clear	Can be controlled only when the host controller function has been selected.

Register Name	Bit Name	Setting Contents	Remarks
DCPCTR PIPEnCTR	CSSTS	SPLIT status indication	Can be referenced only when the host controller function has been selected.
	ATREPM	Auto response mode	PIPE1 to PIPE5: Can be set Can be set only when the function controller function has been selected.
	ACLRM	Auto buffer clear	PIPE1 to PIPE9: Can be set
	SQCLR	Sequence clear	Clears the data toggle bit
	SQSET	Sequence set	Sets the data toggle bit
	SQMON	Sequence monitor	Monitors the data toggle bit
	PBUSY	Pipe busy status	
	PID	Response PID	
PIPEnTRE	TRENB	Transaction counter enable	PIPE1 to PIPE5: Can be set
	TRCLR	Current transaction counter clear	PIPE1 to PIPE5: Can be set
PIPEnTRN	TRNCNT	Transaction counter	PIPE1 to PIPE5: Can be set

(1) Pipe Control Register Modifying Procedures

The following bits in the pipe control registers can be modified only when USB communication is disabled (PID = NAK). Figure 24.8 shows the procedure for rewriting the pipe control registers from a state in which USB communication is enabled (PID = BUF).

Registers that Should Not be Set when USB Communication is Enabled (PID = BUF):

- Bits in DCPMAXP
- The SQCLR, SQSET, and PINGE bits in DCPCTR
- Bits in PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI
- The ATREPM, ACLRM, SQCLR and SQSET bits in PIPEnCTR
- Bits in PIPEnTRE and PIPEnTRN
- Bits in DEVADDn

Note: In addition to the above, observe the setting procedures described in the register descriptions regarding the settings of the CSCLR bit and DEVADDn register.

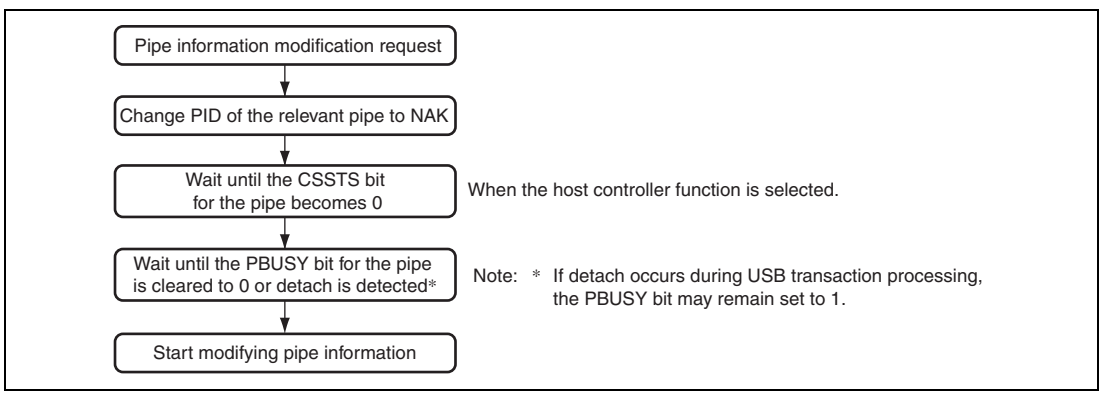


Figure 24.8 Procedure for Changing Pipe Information from a USB Communication Enabled State (PID = BUF)

The following bits in the pipe control registers can be modified only for the pipes that are not specified in the CURPIPE bits for any of the CPU/DMA0/DMA1-FIFO ports.

Registers that Should Not be Set When CURPIPE for FIFO Port is Set.

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI
- The ACLRM bit in PIPEnCTR

In order to modify pipe information, the CURPIPE bits should be set to the pipes other than the pipe to be modified. For the DCP, the buffer should be cleared using BCLR after the pipe information is modified.

(2) Maximum Packet Size Setting

The MXPS bit in DCPMAXP and PIPEMAXP is used to specify the maximum packet size for each pipe. DCP and PIPE1 to PIPE5 can be set to any of the maximum pipe sizes defined by the USB specification. For PIPE6 to PIPE9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID = BUF).

1. DCP: 64 should be set when using high-speed operation.
2. DCP: Select and set 8, 16, 32, or 64 when using full-speed operation.
3. PIPE1 to PIPE5: 512 should be set when using high-speed bulk transfer.
4. PIPE1 to PIPE5: Select and set 8, 16, 32, or 64 when using full-speed bulk transfer.
5. PIPE1 and PIPE2: Set a value between 1 and 1024 when using high-speed isochronous transfer.
6. PIPE1 and PIPE2: Set a value between 1 and 1023 when using full-speed isochronous transfer.
7. PIPE6 to PIPE9: Set a value between 1 and 64.

The high bandwidth transfers used with interrupt transfers and isochronous transfers are not supported.

(3) Response PID

The PID bits in DCPCTR and PIPEnCTR are used to set the response PID for each pipe.

The following shows this module operation with various response PID settings:

(a) Response PID settings when the host controller function is selected

The response PID is used to specify the execution of transactions.

- NAK setting: Using pipes is disabled. No transaction is executed.
- BUF setting: Transactions are executed based on the status of the buffer memory. For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued.
  For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.
- STALL setting: Using pipes is disabled. No transaction is executed.

Setup transactions for the DCP are set with the SUREQ bit.

(b) Response PID settings when the function controller function is selected

The response PID is used to specify the response to transactions from the host.

- NAK setting: The NAK response is always returned in response to the generated transaction.
- BUF setting: Responses are made to transactions based on the status of the buffer memory.
- STALL setting: The STALL response is always returned in response to the generated transaction.

For setup transactions, an ACK response is always returned, regardless of the PID setting, and the USB request is stored in the register.

This module may carry out writing to the PID bits, depending on the results of the transaction.

(a) When the host controller function has been selected and the response PID is set by hardware

- NAK setting: In the following cases, PID = NAK is set and issuing of tokens is automatically stopped:
  - When, during a transfer other than isochronous transfer, three receive errors such as no response, bit stuffing error, or CRC error are returned in succession after token transmission.
  - When, during isochronous transfer, three receive errors such bit stuffing error or CRC error are returned in succession after token transmission.
  - When a short packet is received in the data stage of control read transfer while the SHTNAK bit in the DCPCFG register is set to 1.
  - If a short packet is received when the SHTNAK bit in PIPECFG has been set to 1 for bulk transfer.
  - If the transaction counter ended when the SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by this module.
- STALL setting: In the following cases, PID = STALL is set and issuing of tokens is automatically stopped:
  - When STALL is received in response to the transmitted token.
  - When the size of the receive data packet exceeds the maximum packet size.

(b) When the function controller function has been selected and the response PID is set by hardware

- NAK setting: In the following cases, PID = NAK is set and NAK is always returned in response to transactions:
  - When the SETUP token is received normally (DCP only).
  - If the transaction counter ended or a short packet is received when the SHTNAK bit in PIPECFG has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by this module.
- STALL setting: In the following cases, PID = STALL is set and STALL is always returned in response to transactions:
  - When the size of the receive data packet exceeds the maximum packet size.
  - When a control transfer sequence error has been detected (DCP only).

(4) Data PID Sequence Bit

This module automatically toggles the sequence bit in the data PID when data is transferred normally in the control transfer data stage, bulk transfer and interrupt transfer. The sequence bit of the data PID that was transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit switches at the timing at which the ACK handshake is received. When data is received, the sequence bit switches at the timing at which the ACK handshake is transmitted. The SQCLR bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.

When the function controller function has been selected and control transfer is used, this module automatically sets the sequence bit when a stage transition is made. The bit is set to DATA1 when the setup stage ends. In the status stage, DATA1 is returned without referencing the sequence bit. Therefore, settings are not required. However, when the host controller function has been selected and control transfer is used, the sequence bit should be set at the stage transition.

For the Clearfeature request transmission or reception, the data PID sequence bit should be set, regardless of whether the host controller function or function controller function is selected.

With pipes for which isochronous transfer has been set, sequence bit operation cannot be carried out using the SQSET bit.

24.4.4 FIFO Buffer

This section describes the operation related to the FIFO buffer in the USB module. Unless otherwise noted, the operation is the same regardless of whether the host controller or function controller function is selected.

(1) FIFO Buffer Allocation

Figure 24.9 shows an example of FIFO buffer mapping of this module. The FIFO buffer is an area shared by the CPU and this module. As the FIFO buffer statuses, there is a state in which the access right to the FIFO buffer is held by the system (CPU side), and a state in which the access right is held by this module (SIE side).

Areas for the FIFO buffer are set independently for each pipe. A memory area is set by specifying the first block number and the number of blocks, where one block consists of 64 bytes (the settings are made through the BUFNMB and BUFSIZE bits in PIPEBUF). When continuous transfer mode has been selected using the CNTMD bit in PIPECFG, the BUFSIZE bits should be set so that the buffer memory size should be an integral multiple of the maximum packet size. When double buffer mode has been selected using the DBLB bit in PIPECFG, two planes of the memory area specified using the BUFSIZE bits in PIPEBUF can be assigned to a single pipe.

Three FIFO ports are used for access to the FIFO buffer (reading and writing data). A pipe is assigned to the FIFO port by specifying the pipe number using the CURPIPE bit in C/DnFIFOSEL.

The FIFO buffer statuses of the various pipes can be confirmed using the BSTS bit in DCPCTR and the INBUFM bit in PIPEnCTR. Also, the access right of the FIFO port can be confirmed using the FRDY bit in C/DnFIFOCTR.

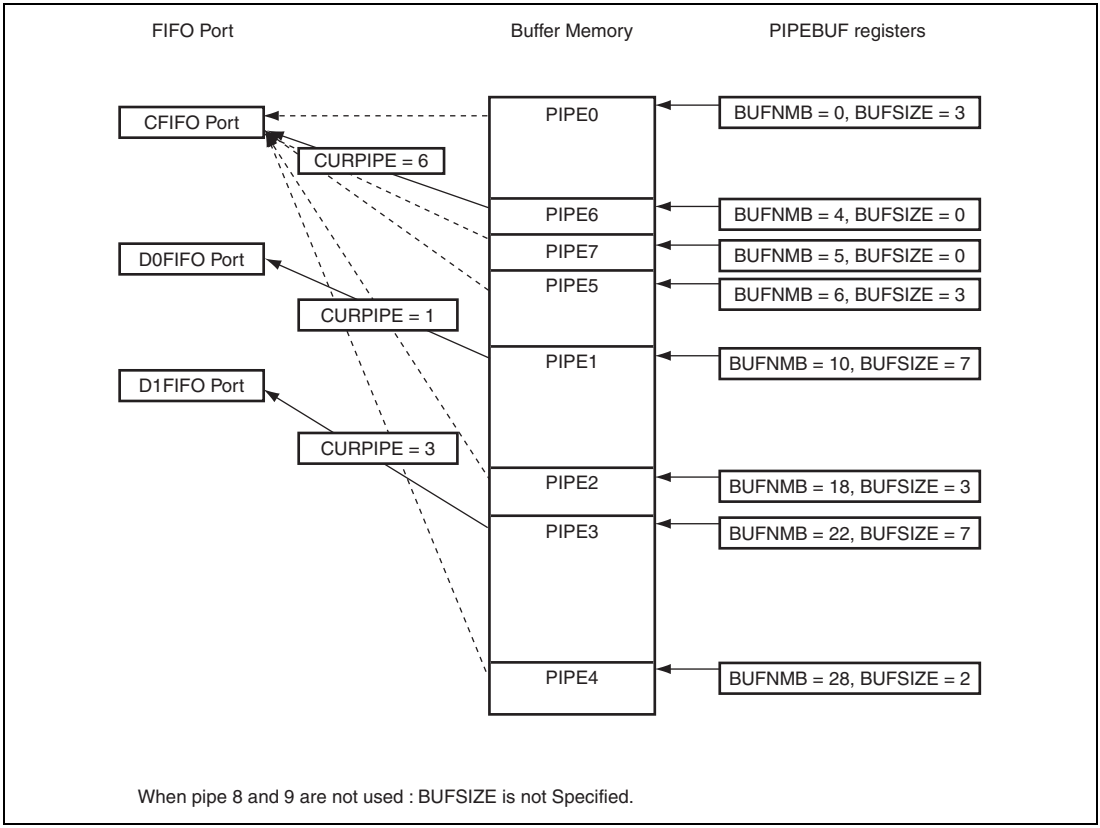


Figure 24.9 Example of FIFO Buffer Memory Mapping

(2) FIFO Buffer Clearing

Table 24.23 summarizes the clearing of the FIFO buffer by this module. The FIFO buffer can be cleared using the three bits shown below.

Table 24.23 List of Buffer Clearing Methods

Bit Name	BCLR	DCLRM	ACLRM
Register	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Function	Clears the FIFO buffer on the CPU side	In this mode, the FIFO buffer is cleared automatically after the data of the specified pipe has been read.	This is the auto buffer clear mode, in which all of the received packets are discarded.
Clearing method	Clear by writing 1 to this bit	1: Mode enabled 0: Mode disabled	1: Mode enabled 0: Mode disabled

(3) FIFO Port Functions

Table 24.24 shows the settings for the FIFO port functions of this module. In write access, writing data until the buffer is full (or the maximum packet size for non-continuous transfers) automatically enables sending of the data to the USB bus. To enable sending of data before the buffer is full (or before the maximum packet size for non-continuous transfers), the BVAL bit in C/DnFIFOCTR must be set to signal that the writing has ended. Also, to send a zero-length packet, the BCLR bit in the same register must be used to clear the buffer and then the BVAL bit set in order to signal the end of writing.

In read access, reception of new packets is automatically enabled if all of the data has been read. Data cannot be read when a zero-length packet is being received (DTLN = 0), so the BCLR bit in the register must be used to release the buffer. The length of the data being received can be confirmed using the DTLN bit in C/DnFIFOCTR.

Table 24.24 FIFO Port Function Settings

Register Name	Bit Name	Function	Note
C/DnFIFOSEL	RCNT	Selects DTLN read mode	
	REW	Buffer memory rewind (re-read, rewrite)	
	DCLRM	Automatically clears data received for a specified pipe after the data has been read	For DnFIFO only
	DREQE	Enable DMA transfer	For DnFIFO only
	MBW	FIFO port access bit width	
	BIGEND	Selects endian for FIFO ports	
C/DnFIFOCTR	BVAL	Indicates writing to the buffer memory has ended	
	BCLR	Clears the buffer memory on the CPU side	
	FRDY	FIFO port ready monitor	
	DTLN	Read to confirm the length of received data	

(a) FIFO Port Selection

Table 24.25 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed is selected using the CURPIPE bit in C/DnFIFOSEL. After the pipe has been selected, check to see whether the CURPIPE value for the last-written pipe has been read correctly (if the previous pipe number is read, it indicates that the pipe changing processing is being done by this module). When a correct value has been read from CURPIPE, confirm FRDY = 1 and then access the FIFO port.

The bus width to be accessed should be selected using the MBW bit. The buffer memory access direction is in accord with the ISEL bit setting for the DCP, and in accord with the DIR bit in PIPECFG for other pipes.

Table 24.25 FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register
PIPE1 to PIPE7	CPU access	CFIFO port register
	DMA access	D0FIFO/D1FIFO port register

(b) Method of Reading Partial Data from FIFO ports

In reading data form an FIFO port when the width of the data to be read is shorter than the bit width set by the MBW bits in the FIFO port select register, read the bit width specified by the MBW bits and use software to discard the unnecessary portion of the data.

In writing data form an FIFO port when the width of the data to be written is shorter than the bit width set by the MBW bits in the FIFO port select register, proceed with access as shown in the examples below. These examples show ways of writing 24-bit data when the width for access to the FIFO port has been set to 32 bits (MBW = 10).

Example 1 of writing partial data: Writing data with 16-and 8-bit widths once each

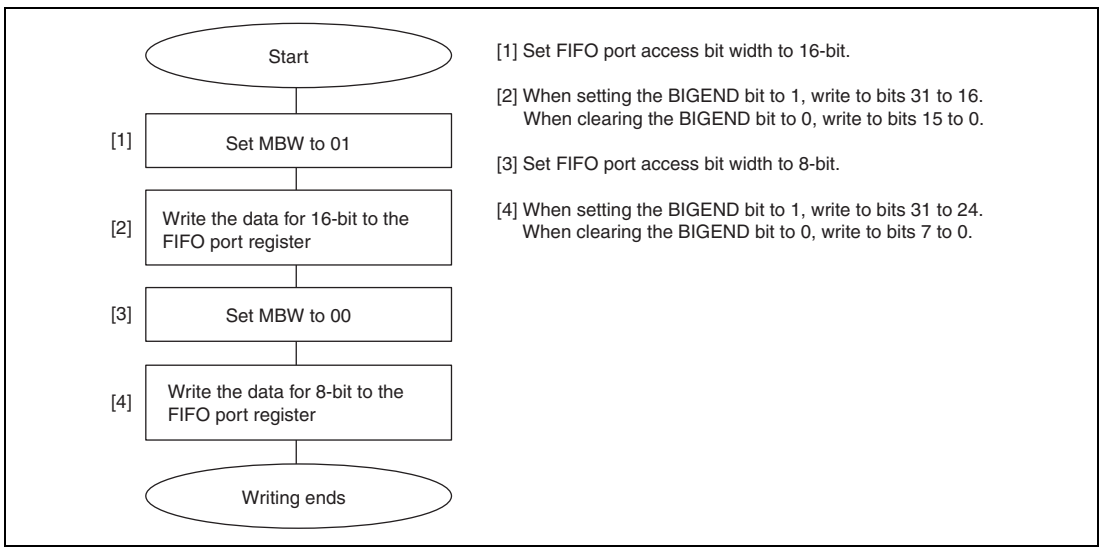


Figure 24.10 Example 1 of Writing Partial Data to an FIFO Port

Example 2 of writing partial data: Write three times by 8-bit width

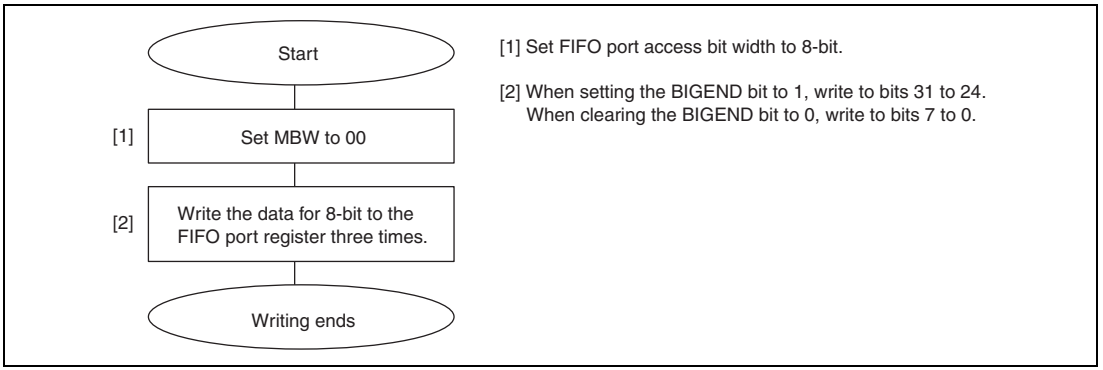


Figure 24.11 Example 2 of Writing Partial Data to an FIFO Port

(c) Changing the MBW Bits when the Direction for the Specified Pipe is Reception

When the direction for the specified pipe is reception, write to the MBW bits of the FIFO port selection register (CFIFOSEL, D0FIFOSEL, and D1FIFOSEL) at the same time as the CURPIPE setting is made. When the CFIFO register has the DCP setting (CURPIPE = 0), write to the MBW bits at the same time as the settings for CURPIPE or ISEL are made.

The procedure for changing the setting of only the MBW bits for a given current pipe setting is shown below. However, once processing to read from the buffer memory has started, do not change the setting of the MBW bits until reading out of all data is complete.

- This is applicable other than when the CURPIPE bits for DFIFO0, DFIFO1, or CFIFO have the DCP setting.

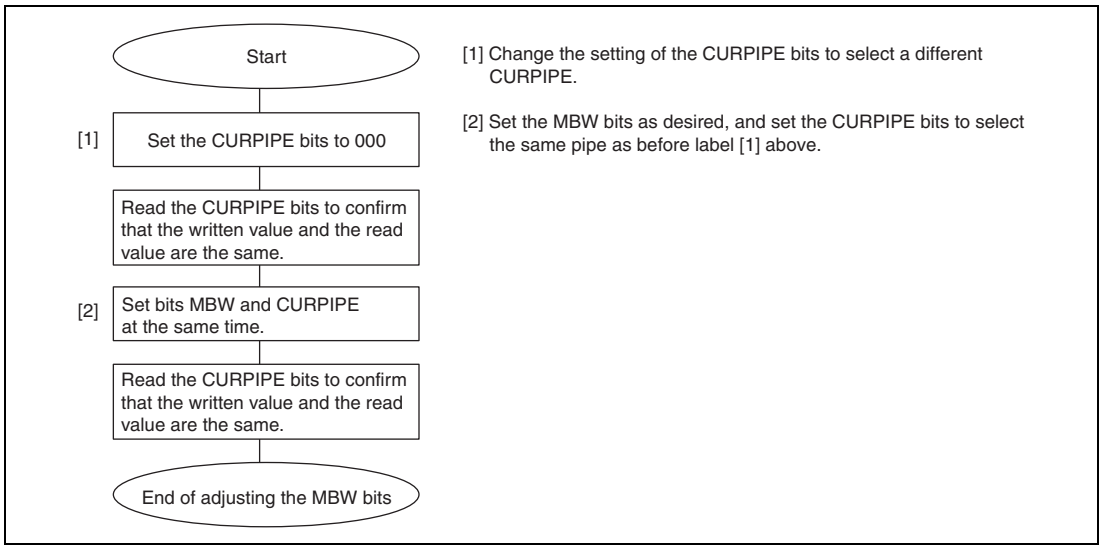


Figure 24.12 Example of Adjusting the MBW Bits when the CURPIPE Bits of DFIFO0, DFIFO1, or CFIFO Have a Setting Other than DCP (000)

- This is applicable when the CURPIPE bits for CFIFO have the DCP setting (000).

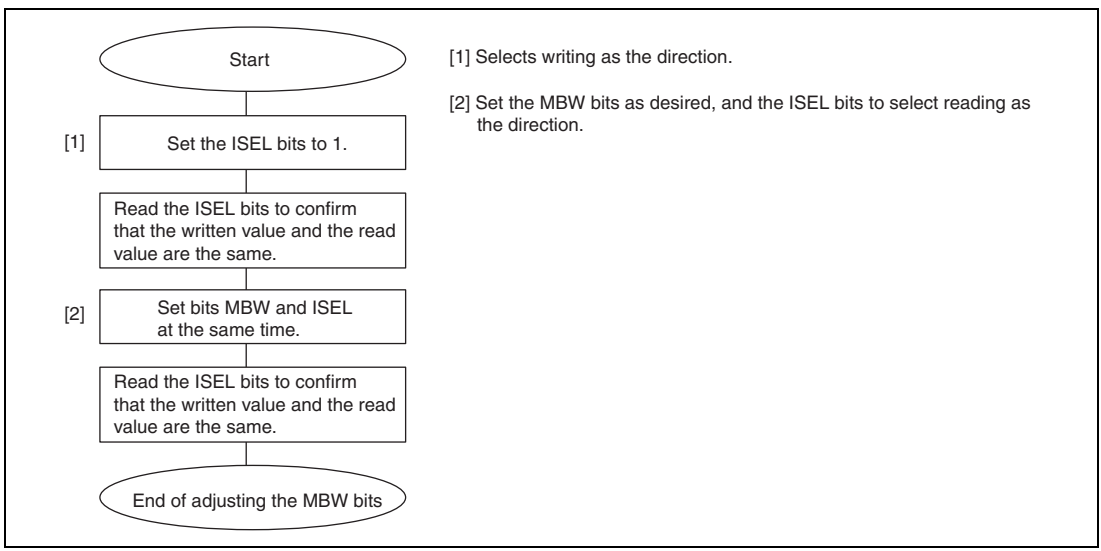


Figure 24.13 Example of Adjusting the MBW Bits when the CURPIPE Bits of CFIFO have a DCP Setting (000)

(4) DMA Transfer (D0FIFO, D1FIFO Ports)

(a) Overview of DMA Transfer

For pipes 1 to 9, the FIFO port is accessible by the DMAC. When the buffer to the pipe set for DMA becomes accessible, a DMA transfer request is output.

In the DnFIFOSEL register, use the MBW bits to set the unit of transfer to the FIFO port for the pipe set for DMA transfer by the CURPIPE bits. Do not change the pipe number while DMA transfer is enabled.

(b) Automatic Recognition of Completion of DMA Transfer

In this module, completion of the writing of FIFO data by DMA transfer can be under the control of the input of a DMA transfer end signal. Then DMA transfer end signal causes the DMAC to proceed with DMA transfer the number of times that corresponds to the setting of the DMA current byte count register (DMCBCT) in the DMAC. Transfer to the buffer memory is enabled (the same as setting BVAL = 1) when the DMA transfer end signal is sampled. The setting for sampling or non-sampling of the DMA transfer end signal can be made in the TENDE bit of the DnFBCFG register. Furthermore, be sure to set the DMA transfer end signal output control bits (DTCM) in the DMA mode register (DMMOD) to 10 (output of the DMA transfer end signal on the final write cycle) whenever this function is used.

(c) Data Transferred for One Operand

In this module, the data to be transferred for each operand is selectable as one unit of data, 16 bytes, or 32 bytes by the DFACC bits in the DMA-FIFO bus-configuration registers (DnFBCFG).

- With the DFACC = 00 (access to one unit of data) setting, set the operand size for the DMAC's method of transfer to 1 and the data size to the size selected by the MBW bits.
- With the DFACC = 01 (consecutive access to 16 bytes) setting, set the operand size for the DMAC's method of transfer and the data size (the size selected by the MBW bits) such that the multiple of the two is 16 bytes.
- With the DFACC = 10 (consecutive access to 32 bytes) setting, set the operand size for the DMAC's method of transfer and the data size (the size selected by the MBW bits) such that the multiple of the two is 32 bytes.

(d) DnFIFO Auto Clear Mode (D0FIFO/D1FIFO Port Reading Direction)

If 1 is set for the DCLRM bit in DnFIFOSEL, the module automatically clears the buffer memory of the corresponding pipe when reading of the data from the buffer memory has been completed.

Table 24.26 shows the packet reception and buffer memory clearing processing for each of the various settings. As shown, the buffer clear conditions depend on the value set to the BFRE bit. Using the DCLRM bit eliminates the need for the buffer to be cleared by software even if a situation occurs that necessitates clearing of the buffer. This makes it possible to carry out DMA transfers without involving software.

This function can be set only in the buffer memory reading direction.

Table 24.26 Packet Reception and Buffer Memory Clearing Processing

Register Setting / Buffer Status When Packet is Received	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Doesn't need to be cleared	Doesn't need to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Zero-length packet reception	Needs to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Normal short packet reception	Doesn't need to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Transaction count ended	Doesn't need to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared

(e) BRDY Interrupt Timing Selection Function

By setting the BFRE bit setting in PIPECFG, it is possible to keep the BRDY interrupt from being generated when a data packet consisting of the maximum packet size is received.

When using DMA transfers, this function can be used to generate an interrupt only when the last data item has been received. The last data item refers to the reception of a short packet, or the ending of the transaction counter. When the BFRE bit is set to 1, the BRDY interrupt is generated after the received data has been read. When the DTLN bit in DnFIFOCTR is read, the length of the data received in the last data packet to have been received can be confirmed.

Table 24.27 shows the timing at which the BRDY interrupts are generated by this module.

Table 24.27 Timing at which BRDY Interrupts are Generated

Register setting / Buffer State When Packet is Received	BFRE = 0	BFRE = 1
Buffer full (normal packet received)	When packet is received	Not generated
Zero-length packet received	When packet is received	When packet is received
Normal short packet received	When packet is received	When reading of the received data from the buffer memory has been completed
Transaction count ended	When packet is received	When reading of the received data from the buffer memory has been completed

Note: This function is valid only in the reading direction of reading from the buffer memory. In the writing direction, the BFRE bit should be fixed at 0.

24.4.5 Control Transfers (DCP)

Data transfers of the data stage of control transfers are done using the default control pipe (DCP). The DCP buffer memory is a 256-byte single buffer, and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed through the CFIFO port.

(1) Control Transfers when the Host Controller Function is Selected

(a) Setup Stage

USQREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the SUREQ bit in DCPCTR transmits the specified data for setup transactions. Upon completion of transactions, the SUREQ bit is cleared to 0. The above USB request registers should not be modified while SUREQ = 1. The device address for setup transactions is specified using the DEVSEL bits in DCPMAXP.

When the data for setup transactions has been sent, a SIGN or SACK interrupt request is generated according to the response received from the peripheral side (SIGN1 or SACK bits in INTSTS1), by means of which the result of the setup transactions can be confirmed.

A data packet of DATA0 (USB request) is transmitted as the data packet for the setup transactions regardless of the setting of the SQMON bit in DCPCTR.

(b) Data Stage

Data transfers are done using the DCP buffer memory.

The access direction of the DCP buffer memory should be specified using the ISEL bit in CFIFOSEL.

For the first data packet of the data stage, the data PID must be transferred as DATA1. Transaction is done by setting the data PID = DATA1 and the PID bit = BUF using the SQSET bit in DCPCTR. Completion of data transfer is detected using the BRDY or BEMP interrupts.

For control write transfers, when the number of data bytes to be sent is the integer multiple of the maximum packet size, control to send a zero-length packet at the end.

(c) Status Stage

Zero-length packet data transfers are done in the direction opposite to that in the data stage. As with the data stage, data transfers are done using the DCP buffer memory. Transactions are done in the same manner as the data stage.

For the data packets of the status stage, the data PID must be transferred as DATA1. The data PID should be set to DATA1 using the SQSET bit in DCPCFG.

For reception of a zero-length packet, the received data length must be confirmed using the DTLN bits in CFIFOCTR after the BRDY interrupt is generated, and the buffer memory must then be cleared using the BCLR bit in C/DnFIFOCTR.

(2) Control Transfers when the Function Controller Function is Selected

(a) Setup Stage

This module always sends an ACK response in response to a setup packet that is normal with respect to this module. The operation of this module operates in the setup stage is noted below.

1. When a new USB request is received, this module sets the following registers:
   — Set the VALID bit in INTSTS0 to 1.
   — Set the PID bit in DCPCTR to NAK.
   — Set the CCPL bit in DCPCTR to 0.
2. When a data packet is received right after the SETUP packet, the USB request parameters are stored in USBREQ, USBVAL, USBINDX, and USBLENG.

Response processing with respect to the control transfer should always be carried out after first setting VALID = 0. In the VALID = 1 state, PID = BUF cannot be set, and the data stage cannot be terminated.

Using the function of the VALID bit, this module is able to interrupt the processing of a request currently being processed if a new USB request is received during a control transfer, and can send a response in response to the newest request.

Also, this module automatically judges the direction bit (bit 8 of the bmRequestType) and the request data length (wLength) of the USB request that was received, and then distinguishes between control read transfers, control write transfers, and no-data control transfers, and controls the stage transition. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified. For information on the stage control of this module, see figure 24.7.

(b) Data Stage

Data transfers corresponding to USB requests that have been received should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the ISEL bit in CFIFOSEL.

A transaction is executed by setting the PID bits in the DCPCTR register to BUF. The BRDY interrupt or the BEMP interrupt can be used to detect the end of data transfer. Use the BRDY interrupt to detect the end of control write transfers and the BEMP interrupt to detect the end of control read transfers.

With control write transfers during high-speed operation, the NYET handshake response is carried out in accordance with the state of the buffer memory.

(c) Status Stage

Control transfers are terminated by setting the CCPL bit to 1 with the PID bit in DCPCTR set to PID = BUF.

After the above settings have been entered, this module automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

1. For control read transfers:
   The zero-length packet is received from the USB host, and this module sends an ACK response.
2. For control write transfers and no-data control transfers:
   This module sends a zero-length packet and receives an ACK response from the USB host.

(d) Control Transfer Auto Response Function

This module automatically responds to a normal SET_ADDRESS request. If any of the following errors occur in the SET_ADDRESS request, a response from the software is necessary.

1. bmRequestType ≠ H'00
2. wIndex ≠ H'00
3. wLength ≠ H'00
4. wValue > H'7F
5. DVSQ = 011 (Configured)

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

24.4.6 Bulk Transfers (PIPE1 to PIPE5)

The buffer memory specifications for bulk transfers (single/double buffer setting, or continuous/non-continuous transfer mode setting) can be selected. The maximum size that can be set for the buffer memory is 2 Kbytes. The buffer memory state is controlled by this module, with a response sent automatically for a PING packet/NYET handshake.

(1) PING Packet Control when the Host Controller Function is Selected

This module automatically sends a PING packet in the OUT direction.

On receiving an ACK handshake in the initial state in which PING packet sending mode is set, this module sends an OUT packet as noted below. The module returns to PING packet sending mode when a NAK or NYET is received during an OUT transaction.

1. Sets OUT data sending mode.
2. Sends a PING packet.
3. Receives an ACK handshake.
4. Sends an OUT data packet.
5. Receives an ACK handshake.

(Repeats steps 4 and 5.)

6. Sends an OUT data packet.
7. Receives an NAK/NYET handshake.
8. Sends a PING packet.

This module is returned to PING packet sending mode by a power-on reset, receiving a NYET/NAK handshake, clearing the sequence toggle bits (SQCLR), and setting the buffer clear bit (ACLRM) in PIPEnCTR.

(2) NYET Handshake Control when the Function Controller Function is Selected

Table 24.28 lists the responses to received tokens during bulk transmission and control transmission. The USB 2.0 host/function module returns a NYET when an OUT token is received while there is only one packet of empty space in the buffer memory, during both bulk transmission and control transmission. However, an ACK is returned instead of a NYET even under the above conditions when a short packet is received.

Table 24.28 List of Responses to Received Tokens

Value Set for PID Bit in DCPCTR	Buffer Memory State	Token	Response	Note
NAK/STALL	—	SETUP	ACK	—
	—	IN/OUT/ PING	NAK/STALL	—
BUF	—	SETUP	ACK	—
	RCV-BRDY1	OUT/PING	ACK	If an OUT token is received, a data packet is received.
	RCV-BRDY2	OUT	NYET	Notifies whether a data packet can be received
	RCV-BRDY2	OUT (Short)	ACK	Notifies whether a data packet can be received
	RCV-BRDY2	PING	ACK	Notifies that a data packet can be received
	RCV-NRDY	OUT/PING	NAK	Notifies that a data packet cannot be received
	TRN-BRDY	IN	DATA0/DATA1	A data packet is transmitted
	TRN-NRDY	IN	NAK	Notifies that a data packet cannot be transmitted

[Legend]

RCV-BRDY1: When an OUT/PING token is received, there is space in the buffer memory for two or more packets.

RCV-BRDY2: When an OUT token is received, there is only enough space in the buffer memory for one packet.

RCV-NRDY: When a PING token is received, there is no space in the buffer memory.

TRN-BRDY: When an IN token is received, there is data to be sent in the buffer memory.

TRN-NRDY: When an IN token is received, there is no data to be sent in the buffer memory.

24.4.7 Interrupt Transfers (PIPE6 to PIPE9)

When the function controller function is selected, this module carries out interrupt transfers in accordance with the timing controlled by the host controller. In interrupt transfers, PING packets are ignored (no responses are sent), and the ACK, NAK, and STALL responses are carried out without an NYET handshake response being made.

When the host controller function is selected, this module can set the timing of issuing a token using the interval timer. This module issues an OUT token without issuing a PING token even in the OUT direction.

This module does not support high bandwidth transfers of interrupt transfers.

(1) Interval Counter during Interrupt Transfers when the Host Controller Function is Selected

(a) Outline of Operation

For interrupt transfers, intervals between transactions are set in the IITV bits in PIPEPERI. This controller issues an interrupt transfer token based on the specified intervals.

(b) Counter Initialization

This controller initializes the interval counter under the following conditions.

- Power-on reset:
  The IITV bits are initialized.
- Buffer memory initialization using the ACLRM bit:
  The IITV bits are not initialized but the count value is. Setting the ACLRM bit to 0 starts counting from the value set in the IITV bits.

Note that the interval counter is not initialized in the following case.

- USB bus reset, USB suspended:
  The IITV bits are not initialized. Setting 1 to the UACT bit starts counting from the value before entering the USB bus reset state or USB suspended state.

(c) Operation when Transmission/Reception is Impossible at Token Issuance Timing

This module cannot issue tokens even at token issuance timing in the following cases. In such a case, this module attempts transactions at the subsequent interval.

- When the PID is set to NAK or STALL.
- When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- When there is no data to be sent in the buffer memory at the token sending timing in the sending (OUT) direction.

24.4.8 Isochronous Transfers (PIPE1 and PIPE2)

This module has the following functions pertaining to isochronous transfers.

- Notification of isochronous transfer error information
- Interval counter (specified by the IITV bit)
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function (specified by the IFIS bit)

This module does not support the High Bandwidth transfers of isochronous transfers.

When operation as a host controller is selected and two pipes are used for isochronous transfer at the same time, observe the restrictions on packets stated in the USB 2.0 Specification, section 5.6.3, Isochronous Transfer Packet Size Constraints.

(1) Error Detection with Isochronous Transfers

This module has a function for detecting the error information noted below, so that when errors occur in isochronous transfers, software can control them. Tables 24.28 and 24.29 show the priority in which errors are confirmed and the interrupts that are generated.

(a) PID errors

If the PID of the packet being received is illegal

(b) CRC errors and bit stuffing errors

If an error occurs in the CRC of the packet being received, or the bit stuffing is illegal

(c) Maximum packet size exceeded

The maximum packet size exceeded the set value.

(d) Overrun and underrun errors

- When host controller function is selected:
  - When using isochronous IN transfers (reception), the IN token was received but the buffer memory is not empty.
  - When using isochronous OUT transfers (transmission), the OUT token was transmitted, but the data was not in the buffer memory.

- When function controller function is selected:
  - When using isochronous IN transfers (transmission), the IN token was received but the data was not in the buffer memory.
  - When using isochronous OUT transfers (reception), the OUT token was received, but the buffer memory was not empty.

(e) Interval errors

When function controller function is selected, the following cases are considered as interval errors:

- During an isochronous IN transfer, the token could not be received during the interval frame.
- During an isochronous OUT transfer, the OUT token was received during frames other than the interval frame.

Table 24.29 Error Detection when a Token is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated in both cases when the host controller function is selected and the function controller function is selected (ignored as a corrupted packet).
2	CRC error and bit stuffing errors	No interrupts generated in both cases when the host controller function is selected and the function controller function is selected (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated to set the OVRN bit in both cases when host controller function is selected and function controller function is selected. When the host controller function is selected, no tokens are transmitted. When the function controller function is selected, a zero-length packet is transmitted in response to IN token. However, no data packets are received in response to OUT token.
4	Interval errors	An NRDY interrupt is generated when the function controller function is selected. It is not generated when the host controller function is selected.

Table 24.30 Error Detection when a Data Packet is Received

Detection Priority Order	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated (ignored as a corrupted packet)
2	CRC error and bit stuffing errors	An NRDY interrupt is generated to set the CRCE bit in both cases when the host controller function is selected and the function controller function is selected.
3	Maximum packet size exceeded error	A BEMP interrupt is generated to set the PID bits to STALL in both cases when the host controller function is selected and the function controller function is selected.

(2) DATA-PID

This module does not support High Bandwidth transfers. When the function controller function is selected, this module operates as follows in response to the received PID.

(a) IN direction

- DATA0: Sent as data packet PID
- DATA1: Not sent
- DATA2: Not sent
- mDATA: Not sent

(b) OUT direction (when using full-speed operation)

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets are ignored
- mDATA: Packets are ignored

(c) OUT direction (when using high-speed operation)

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Received normally as data packet PID
- mDATA: Received normally as data packet PID

(3) Interval Counter

(a) Outline of Operation

The isochronous interval can be set using the IITV bits in PIPEPERI. The interval counter enables the functions shown in table 24.31 when the function controller function is selected. When the host controller function is selected, this module generates the token issuance timing. When the host controller function is selected, the interval counter operation is the same as the interrupt transfer operation.

Table 24.31 Functions of the Interval Counter when the Function Controller Function is Selected

Transfer Direction	Function	Conditions for Detection
IN	IN buffer flush function	When a token cannot be normally received in the interval frame during an isochronous IN transfer
OUT	Notifies that a token not being received	When a token cannot be normally received in the interval frame during an isochronous OUT transfer

The interval count is carried out when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is damaged. The frame interval that can be set is the 2^{IITV} frame or $2^{IITV}\mu$ frames.

(b) Interval Counter Initialization when the Function Controller Function is Selected

This module initializes the interval counter under the following conditions.

- Power-on reset
  The IITV bit is initialized.
- Buffer memory initialization using the ACLRM bit
  The IITV bits are not initialized but the count value is.
- USB bus reset

After the interval counter has been initialized, the counter is started under the following conditions 1 or 2 when a packet has been transferred normally.

1. An SOF is received following transmission of data in response to an IN token, in the PID = BUF state.
2. An SOF is received after data following an OUT token is received in the PID = BUF state.

The interval counter is not initialized under the conditions noted below.

1. When the PID bit is set to NAK or STALL

   The interval timer does not stop. This module attempts the transactions at the subsequent interval.
2. The USB bus reset or the USB is suspended

   The IITV bit is not initialized. When the SOF has been received, the counter is restarted from the value prior to the reception of the SOF.

(4) Setup of Data to be Transmitted using Isochronous Transfer when the Function Controller Function is Selected

With isochronous data transmission using this module in function controller function, after data has been written to the buffer memory, a data packet can be sent with the next frame in which an SOF packet is detected. This function is called the isochronous transfer transmission data setup function, and it makes it possible to designate the frame from which transmission began.

If a double buffer is used for the buffer memory, transmission will be enabled for only one of the two buffers even after the writing of data to both buffers has been completed, that buffer memory being the one to which the data writing was completed first. For this reason, even if multiple IN tokens are received, the only buffer memory that can be sent is one packet's worth of data.

When an IN token is received, if the buffer memory is in the transmission enabled state, this module transmits the data. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 24.14 shows an example of transmission using the isochronous transfer transmission data setup function with this module, when IITV = 0 (every frame) has been set.

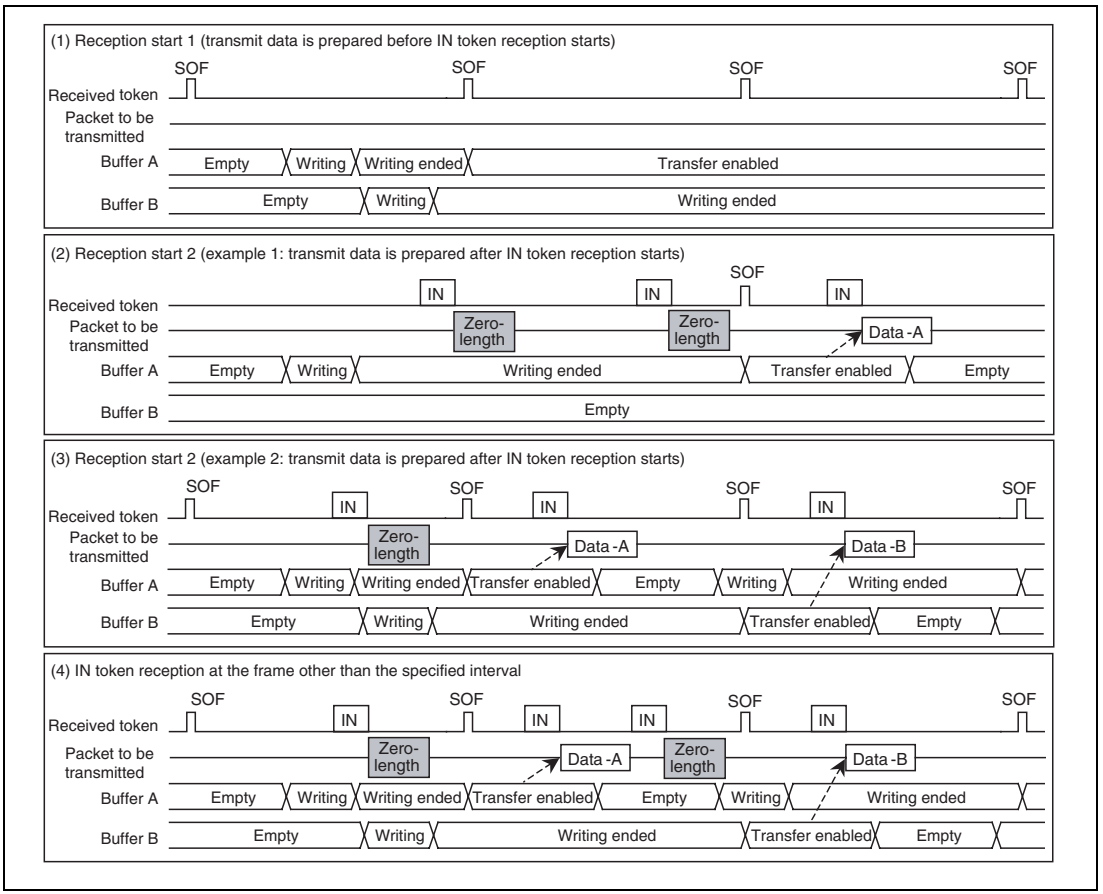


Figure 24.14 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmission Buffer Flush when the Function Controller Function is Selected

If an SOF packet or a μSOF packet is received without receiving an IN token in the interval frame during isochronous data transmission, this module operates as if an IN token had been corrupted, and clears the buffer for which transmission is enabled, putting that buffer in the writing enabled state.

If a double buffer is being used and writing to both buffers has been completed, the buffer memory that was cleared is seen as the data having been sent at the same interval frame, and transmission is enabled for the buffer memory that is not discarded with SOF or μSOF packets reception.

The timing at which the operation of the buffer flush function varies depending on the value set for the IITV bit.

- If IITV = 0
  The buffer flush operation starts from the next frame after the pipe becomes valid.
- In any cases other than IITV = 0
  The buffer flush operation is carried out subsequent to the first normal transaction.

Figure 24.15 shows an example of the buffer flush function of this module. When an unanticipated token is received prior to the interval frame, this module sends the written data or a zero-length packet according to the buffer state.

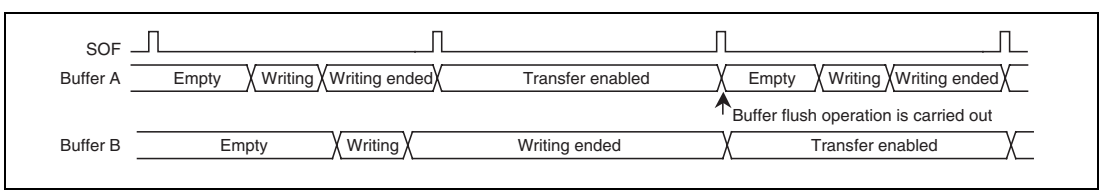


Figure 24.15 Example of Buffer Flush Function Operation

Figure 24.16 shows an example of this module generating an interval error. There are five types of interval errors, as shown below. The interval error is generated at the timing indicated by (1) in the figure, and the IN buffer flush function is activated.

If an interval error occurs during an IN transfers, the buffer flush function is activated; and if it occurs during an OUT transfer, an NRDY interrupt is generated.

The OVRN bit should be used to distinguish between NRDY interrupts such as received packet errors and overrun errors.

In response to tokens that are shaded in the figure, responses occur based on the buffer memory status.

1. IN direction:
   — If the buffer is in the transmission enabled state, the data is transferred as a normal response.
   — If the buffer is in the transmission disabled state, a zero-length packet is sent and an underrun error occurs.
2. OUT direction:
   — If the buffer is in the reception enabled state, the data is received as a normal response.
   — If the buffer is in the reception disabled state, the data is discarded and an overrun error occurs.

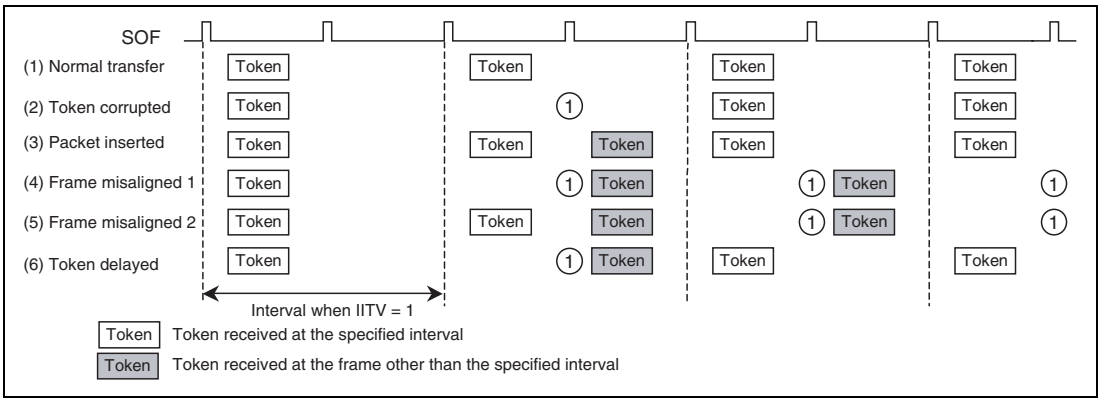


Figure 24.16 Example of an Interval Error Being Generated when IITV = 1

24.4.9 SOF Interpolation Function

When the function controller function is selected and if data could not be received at intervals of 1 ms (when using full-speed operation) or 125 μs (when using high-speed operation) because an SOF packet was corrupted or missing, this module interpolates the SOF. The SOF interpolation operation begins when USBE = 1, SCKE = 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- Power-on reset
- USB bus reset
- Suspended state detected

Also, the SOF interpolation operates under the following specifications.

- 125 μs/1 ms conforms to the results of the reset handshake protocol.
- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, either 125 μs or 1 ms is counted with an internal clock of 48 MHz, and interpolation is carried out.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received. (With suspended transitions in high-speed operation, interpolation continues for 3 ms after the last packet is received.)

This module supports the following functions based on the SOF detection. These functions also operate normally with SOF interpolation, if the SOF packet was corrupted.

- Refreshing of the frame number and the micro-frame number
- SOFR interrupt and μSOF lock
- Isochronous transfer interval count

If an SOF packet is missing when full-speed operation is being used, the FRNM bit in FRMNUM0 is not refreshed.

If a μSOF packet is missing during high-speed operation, the UFRNM bit in FRMNUM1 is refreshed.

However, if a μSOF packet for which the μFRNM = 000 is missing, the FRNM bit is not refreshed. In this case, the FRNM bit is not refreshed even if successive μSOF packets other than μFRNM = 000 are received normally.

24.5 Usage Notes

24.5.1 Procedure for Setting the USB Transceiver

When this module is to be used, start by making settings for the internal USB transceiver.

The method for the settings is described below. Figure 24.17 also gives an example of program code to implement the procedure.

(1) Write 1 to bits UACS14 and UACS5 in USBAC characteristics switching register 0 (USBACSWR0).

(2) Write 1 to the UACS26 bit in USBAC characteristics switching register 1 (USBACSWR1).

```
Initialization routine
(1) Set 1 to UACS14 and UACS5.
        MOVI20      #H'FFFF00C0, R0
        MOV.W       #H'4020, R1
        MOV.W       R1, @R0

(2) Set 1 to UACS26.
        MOVI20      #H'FFFF00C2, R0
        MOV.W       #H'0400, R1
        MOV.W       R1, @R0
```

Figure 24.17 Procedure for Setting the USB Transceiver

Section 25 AT Attachment Packet Interface (ATAPI)

The ATAPI interface device provides both the ATA and ATAPI physical interfaces. This device also supports both the ATA task and ATAPI packet commands.

25.1 Features

- Supporting primary channel
- Supporting master/slave
- Supporting 3.3V I/O interface
- Supporting PIO modes 0 to 4, multiword DMA modes 0 to 2, and ultra DMA modes 0 to 2

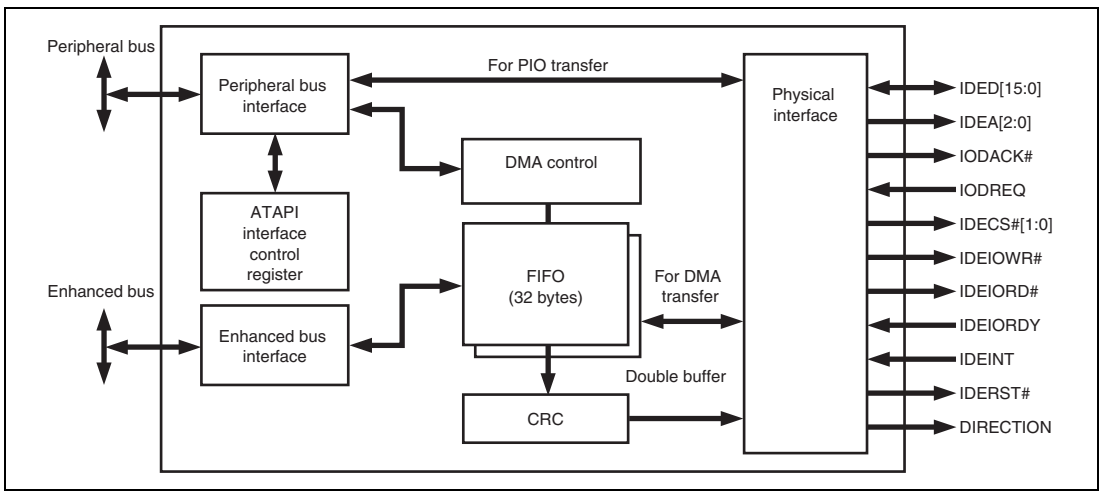


Figure 25.1 Block Diagram of ATAPI

25.2 Input/Output Pins

Table 25.1 Pin Configuration

Signal	ATAPI Specification	I/O	Function
IDED[15:0]	DD[15:0]	I/O	Bidirectional data bus
IDEA[2:0]	DA[2:0]	Output	Address bus
IODACK#	DMACK#	Output	Primary channel DMA acknowledge (active low)
IODREQ	DMARQ	Input	Primary channel DMA request (active high)
IDECS#[1:0]	CS0#, CS1#	Output	Primary channel chip select (active low)
IDEIOWR#	DIOW#, STOP	Output	Primary channel disk write (active low)
IDEIORD#	DIOR#, HDMARDY#, HSTROBE	Output	Primary channel disk read (active low)
IDEIORDY	IORDY, DDMARDY#, DSTROBE	Input	Primary channel ready signal (active high)
IDEINT	INTRQ	Input	Primary channel interrupt request* (active high)
IDERST#	RESET#	Output	Primary channel ATAPI device reset (active low)
DIRECTION	—	Output	External level shifter direction signal (0 when writing to the device)

Note: * The ATAPI interface treats the interrupt signal from the ATAPI device as a level-triggered input.

25.3 Register Description

The following register set is allocated in the on-chip peripheral module space of this LSI device.

25.3.1 ATAPI Interface Registers

Table 25.2 ATA Task File Register Map

(These resisters are allocated to the ATAPI or ATA device, but not to this module.)

Address	Read Register	Write Register	Pin Address (IDECS[1:0]#, IDEA[2:0]) H: High Level L: Low Level@3.3V I/O	Access Size*1 (Available Bit Size)	Register Location
H'FFFECC00	Data	Data	HL-LLL/HH-XXX (X: Don't care)	32 (16)*2	Drive
H'FFFECC04	Error	Function	HL-LLH	32 (8)*3	Drive
H'FFFECC08	Sector count	Sector count	HL-LHL	32 (8)*3	Drive
H'FFFECC0C	Sector number	Sector number	HL-LHH	32 (8)*3	Drive
H'FFFECC10	Cylinder low	Cylinder low	HL-HLL	32 (8)*3	Drive
H'FFFECC14	Cylinder high	Cylinder high	HL-HLH	32 (8)*3	Drive
H'FFFECC18	Device/head	Device/head	HL-HHL	32 (8)*3	Drive
H'FFFECC1C	Status	Command	HL-HHH	32 (8)*3	Drive
H'FFFECC38	Alternate status	Device control	LH-HHL	32 (8)*3	Drive

Notes: 1. The CPU must access these registers in longword (32-bit) units. Byte and word accesses are prohibited.
2. Bits 15 to 0 of the data bus are used.
3. Bits 7 to 0 of the data bus are used.

Table 25.3 ATAPI Packet Command Task File Register Map

(These resisters are allocated to the ATAPI or ATA device, but not to this module.)

Address	Read Register	Write Register	Pin Address (IDECS[1:0]#, IDEA[2:0])	Access Size*[1] (Available Bit Size)	Register Location
H'FFFECC00	Data	Data	HL-LLL	32 (16)*[2]	Drive
H'FFFECC04	Error	Function	HL-LLH	32 (8)*[3]	Drive
H'FFFECC08	Interrupt source	—	HL-LHL	32 (8)*[3]	Drive
H'FFFECC0C	—	—	HL-LHH	32 (8)*[3]	Drive
H'FFFECC10	Byte count low	Byte count low	HL-HLL	32 (8)*[3]	Drive
H'FFFECC14	Byte count high	Byte count high	HL-HLH	32 (8)*[3]	Drive
H'FFFECC18	Device select	Device select	HL-HHL	32 (8)*[3]	Drive
H'FFFECC1C	Status	Command	HL-HHH	32 (8)*[3]	Drive
H'FFFECC38	Alternate status	Device control	LH-HHL	32 (8)*[3]	Drive

Notes: 1. The CPU must access these registers in longword (32-bit) units. Byte and word accesses are prohibited.
2. Bits 15 to 0 of the data bus are used.
3. Bits 7 to 0 of the data bus are used.

Table 25.4 ATAPI Interface Control Register Map

(These resisters are allocated to this module.)

Address	Register Name	Abbreviation	Access Type	Access Size*
H'FFFECC80	ATAPI control	ATAPI_CONTROL	R/W	32
H'FFFECC84	ATAPI status	ATAPI_STATUS	R/W	32
H'FFFECC88	Interrupt enable	ATAPI_INT_ENABLE	R/W	32
H'FFFECC8C	PIO timing	ATAPI_PIO_TIMING	R/W	32
H'FFFECC90	Multiword DMA timing	ATAPI_MULTI_TIMING	R/W	32
H'FFFECC94	Ultra DMA timing	ATAPI_ULTRA_TIMING	R/W	32
H'FFFECC9C	DMA start address	ATAPI_DMA_START_ADR	R/W	32
H'FFFECCA0	DMA transfer count	ATAPI_DMA_TRANS_CNT	R/W	32
H'FFFECCA4	ATAPI control 2	ATAPI_CONTROL2	R/W	32
H'FFFECCB0	ATAPI signal status	ATAPI_SIG_ST	R	32
H'FFFECCBC	Byte swap	ATAPI_BYTE_SWAP	R/W	32

Note: * These registers must be accessed in longword (32-bit) units. Byte and word accesses are prohibited.

25.3.2 ATAPI Interface Control Register Map

[Legend]

Initial value: Register value after power-on reset

—: Undefined value

R/W: Readable and writable bit. The write value can be read.

R/WC0: Readable and writable bit. If 0 is written, the bit is initialized. If 1 is written, it is ignored.

R: Read-only register; only 0 should be written unless otherwise stated.

—/W: Write-only bit. The read value is undefined.

All control/status registers are active high.

(1) ATAPI control register (ATAPI_CONTROL)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	DTCD	-	RESET	M/S	-	UDMAEN	-	R/W	STOP	START
Initial value:	-	-	-	-	-	-	0	-	0	0	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	—	R	Reserved

Bit	Bit Name	Initial Value	R/W	Description
9	DTCD	0	R/W	This bit controls operating mode for device terminations that occur continuously during ultra DMA. No abnormal termination occurs if the specified number of transfers has not been reached after device termination acceptance. Transfer will restart at a DMARQ from the next device. Some of the existing ATA devices handle device terminations in the same way as pauses. Therefore, if the specified number of transfers has not been reached after device termination acceptance, no abnormal termination occurs, and it is necessary to restart transfer at a DMARQ from the next device. This operating mode is called “device termination continuation mode.” 1: Device termination continuation mode prohibited 0: Device termination continuation mode
8	—	—	R	Reserved
7	RESET	0	R/W	This bit controls resetting the ATAPI device. Setting the bit to 1 causes the ATAPI reset signal to be asserted. The IDERST# signal is an active-low signal. If the bit is set to 1, the IDERST# signal goes low. If the bit is cleared to 0, the IDERST# signal goes high.
6	M/S	0	R/W	This bit selects an ATAPI device as master or slave. 1: The ATAPI device is selected as master. 0: The ATAPI device is selected as slave.
5	—	1	R	Reserved The write value should always be 1.
4	UDMAEN	0	R/W	This bit is an ultra DMA enable bit. To use ultra DMA mode, set the bit to 1. To use multiword DMA or PIO mode, clear the bit to 0.
3	—	0	R	Reserved

Bit	Bit Name	Initial Value	R/W	Description
2	R/W	0	R/W	This bit controls which to perform, FIFO read or write. 1: FIFO read (DMA transfer data-in operation) 0: FIFO write (DMA transfer data-out operation) To read data from an ATAPI device, set the bit to 1. To write data to an ATAPI device, clear the bit to 0.
1	STOP	0	R/W	This bit forcibly terminates a DMA transfer. [Write] 0: Ignored. 1: A data transfer is forcibly terminated. [Read] 0: No forced termination command is issued. 1: A forced data transfer termination command is issued. The bit is cleared to 0 when the next DMA transfer starts. Note: No transfer can be restarted from the address at which a DMA transfer was forcibly terminated.
0	START	0	R/W	This bit causes a DMA transfer to start. If the bit is set to 1, a DMA transfer is started. If it is cleared to 0, it is ignored. [Write] 0: Ignored. 1: A DMA transfer is started. [Read] 0: DMA transfer inactive. 1: Busy, performing a DMA transfer. Note: The task file register must not be accessed while DMA is active.

(2) ATAPI status register (ATAPI_STATUS)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	SWERR	IFERR	-	DEVTRM	DEVINT	TOUT	ERR	NEND	ACT
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/WC0	R/WC0	R	R/WC0	R	R/WC0	R/WC0	R/WC0	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	—	R	Reserved
8	SWERR	0	R/WC0	This bit is a software error bit. If the bit is set to 1, it indicates that an attempt was made to access the task file register when DMA is active. The task register must not be accessed when DMA is active. The bit is set to 1, for example, if an attempt is made to perform a PIO transfer during an ultra or multiword DMA transfer. No error is reported to the outside of the LSI device; the attempt is merely ignored. Writing 0 results in the bit being reset.
7	IFERR	0	R/WC0	This bit indicates that an ATAPI interface protocol error was detected. • (IDEDREQ = 1) or (IDEIORDY = 0) if ultra DMA data-in burst was terminated by the host • IDEIORDY = 0 if ultra DMA data-in burst was terminated by the device • IDEIORDY = 0 if ultra DMA data-out burst was started • (IDEDREQ = 1) or (IDEIORDY = 0) if ultra DMA data-out burst was terminated by the host Writing 0 results in the bit being reset.
6	—	0	R	Reserved
5	DEVTRM	0	R/WC0	This bit is set to 1 if an ATAPI device exits ultra DMA mode before the number of transfer bytes specified by the ATAPI module is reached. Writing 0 results in the bit being reset.

Bit	Bit Name	Initial Value	R/W	Description
4	DEVINT	0	R	This bit indicates the status of the ATAPI device interrupt IDEINT. The bit is a read-only bit. Because the bit does not hold its status in this LSI device, it is cleared to 0 if the IDEINT bit is 0. The ATAPI interface treats the interrupt signal from the ATAPI device as a level-triggered input. According to the ATAPI standard, the ATAPI device negates IDEINT, in order to clear any interrupt pending condition, within 400 ns after IDEIORD# used to read from the status register is negated.
3	TOUT	0	R/WC0	This bit indicates that an IORDY time-out was detected. This time-out is detected if no response is made for 150 or more enhanced bus clock cycles (IDEIORDY pin is low). Writing 0 results in the bit being reset.
2	ERR	0	R/WC0	This bit is set to 1 if a DMA abort is detected. ERR = 1 if: • The host forcibly terminates a DMA transfer. • DTCD = 1 and a device termination occurs, leading to ACT = 0. Writing 0 results in the bit being reset.
1	NEND	0	R/WC0	This bit indicates that a DMA ended normally. Writing 0 results in the bit being reset.
0	ACT	0	R	This bit indicates that DMA is active. The bit is a read-only bit. It is cleared to 0 when a DMA transfer is completed. It is not recommended to use the bit as an interrupt source.

(3) Interrupt enable (ATAPI_INT_ENABLE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	iSWERR	iIFERR	-	iDEVTRM	iDEVINT	iTOUT	iERR	iNEND	iACT
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	—	R	Reserved
8	iSWERR	0	R/W	This bit is an SWERR interrupt enable bit.
7	iIFERR	0	R/W	This bit is an IFERR interrupt enable bit.
6	—	—	R	Reserved
5	iDEVTRM	0	R/W	This bit is a DEVTRM interrupt enable bit.
4	iDEVINT	0	R/W	This bit is a DEVINT interrupt enable bit.
3	iTOUT	0	R/W	This bit is a TOUT interrupt enable bit.
2	iERR	0	R/W	This bit is an ERR interrupt enable bit.
1	iNEND	0	R/W	This bit is an NEND interrupt enable bit.
0	iACT	0	R/W	This bit is an ACT interrupt enable bit. It is not recommended to set the bit to 1, because ACT is cleared automatically when a DMA transfer is completed.

Note: Writing 1 to each bit enables the interrupt signal corresponding to each bit in the ATAPI status register.

(4) PIO timing register (ATAPI_PIO_TIMING)

Before accessing an ATAPI device, set the number of machine cycles in the following bits in this register.

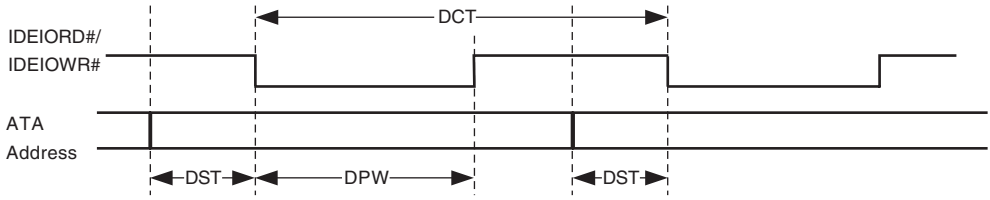
A machine cycle is equal to an enhanced bus clock cycle. Its frequency is the same as of the bus clock.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	pSDCT						pSDPW					pSDST		
Initial value:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	pMDCT						pMDPW					pMDST		
Initial value:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	R	Reserved
29 to 24	pSDCT	000000	R/W	These bits specify the cycle time for a slave ATAPI device.
23 to 19	pSDPW	00000	R/W	These bits specify the width of the IDEIORD#/IDEIOWR# pulse for a slave ATAPI device.
18 to 16	pSDST	000	R/W	These bits specify the address setup time for IDEIORD#/IDEIOWR# for a slave ATAPI device in PIO mode.
15, 14	—	—	R	Reserved
13 to 8	pMDCT	000000	R/W	These bits specify the cycle time for the master ATAPI device.
7 to 3	pMDPW	00000	R/W	These bits specify the width of the IDEIORD#/IDEIOWR# pulse for the master ATAPI device.
2 to 0	pMDST	000	R/W	These bits specify the address setup time for IDEIORD#/IDEIOWR# for the master ATAPI device in PIO mode.

Note: The prefix pS pertains to slaves, and pM, to the master.



DCT: Period setting
DPW: Low-level width setting for the IDEIORD#/IDEIOWR# pulse
DST: Setup time setting for address and IDEIORD#/IDEIOWR#
Note: The DCT, DPW, and DST are determined by their respective register settings · enhanced bus clock period.

Figure 25.2 PIO Timing Register

- PIO timing register value table (master/slave)

Enhanced Bus Clock	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4
66 MHz	H'29A5	H'1BA4	H'11A3	H'0D3B	H'0933

(5) Multiword DMA timing register (ATAPI_MULTI_TIMING)

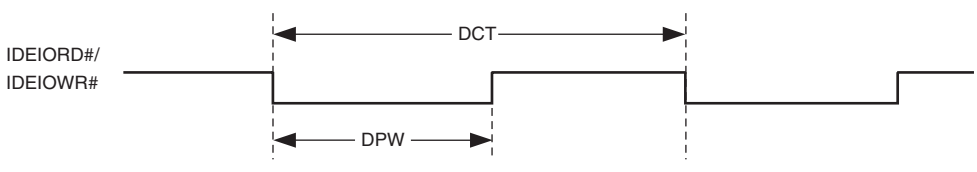
Before accessing an ATAPI device, set the number of machine cycles in the following bits in this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	mSDCT						mSDPW				
Initial value:	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	mMDCT						mMDPW				
Initial value:	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	—	R	Reserved
26 to 21	mSDCT	000000	R/W	These bits specify the cycle time for a slave ATAPI device.
20 to 16	mSDPW	00000	R/W	These bits specify the width of the IDEIORD#/IDEIOWR# pulse for a slave ATAPI device.
15 to 11	—	—	R	Reserved
10 to 5	mMDCT	000000	R/W	These bits specify the cycle time for the master ATAPI device.
4 to 0	mMDPW	00000	R/W	These bits specify the width of the IDEIORD#/IDEIOWR# pulse for the master ATAPI device.

Note: The prefix mS pertains to slaves, and mM, to the master.



DCT: Period setting
DPW: Low-level width setting for the IDEIORD#/IDEIOWR# pulse
Note: The DCT and DPW are determined by their respective register settings × enhanced bus clock period.

Figure 25.3 Multiword DMA Timing Register

- Multiword DMA timing register value table

Enhanced Bus Clock	Mode 0	Mode 1	Mode 2
66 MHz	H'042F	H'0166	H'0126

(6) Ultra DMA timing register (ATAPI_ULTRA_TIMING)

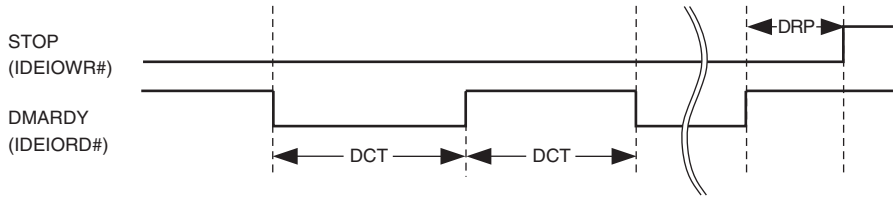
Before accessing an ATAPI device, set the number of machine cycles in the following bits in this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	uSDCT				uSDRP				
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	uMDCT				uMDRP				
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	—	R	Reserved
24 to 21	uSDCT	0000	R/W	These bits specify the cycle time for a slave ATAPI device.
20 to 16	uSDRP	00000	R/W	These bits specify the time from the negation of DMARDY (not IDEIORDY) until this module is suspended by a slave ATAPI device.
15 to 9	—	—	R	Reserved
8 to 5	uMDCT	0000	R/W	These bits specify the cycle time for the master ATAPI device.
4 to 0	uMDRP	00000	R/W	These bits specify the time from the negation of DMARDY (not IDEIORDY) until this module is suspended by the master ATAPI device.

Note: The prefix uS pertains to slaves, and uM, to the master.



DCT: Period setting
DRP: Set period from the negation of DMARDY (IDEIORD#) until the STOP (IDEIOWR#) signal is issued (used during data-in burst)

Note: The DCT and DRP are determined by their respective register settings · enhanced bus clock period.

Figure 25.4 Ultra DMA Timing Register

- Ultra DMA timing register value table

Enhanced Bus Clock	Mode 0	Mode 1	Mode 2
66 MHz	H'010C	H'00C9	H'00A8

(7) DMA start address register (ATAPI_DMA_START_ADR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	DSTA[28:16]												
Initial value:	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSTA[15:2]														-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	—	R	Reserved
28 to 2	DSTA[28:2]	H'x000000	R/W	These bits specify a DMA start address, which is a transfer start address for data in memory. Bits 28 to 0 are used to specify a DMA start address on a byte basis. Bits 1 and 0 are ignored because it is necessary to secure a 32-bit address boundary for the DMA start address. [Write] Write 1 to bits 28 and 27.
1, 0	—	—	R	Reserved

Notes: 1. This address will not change even after DMA becomes active; it will retain its setting.
2. The access destination is in SDRAM.

(8) DMA transfer count register (ATAPI_DMA_TRANS_CNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	DTRC[28:16]												
Initial value:	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTRC[15:1]															-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	—	R	Reserved
28 to 1	DTRC[28:1]	H'0000000	R/W	These bits specify a DMA transfer count. Bits 28 to 0 are used to set a DMA transfer count on a byte basis. Bit 0 is ignored because the ATAPI data bus operates in16-bit (1-word) units.
0	—	—	R	Reserved

Note: This count value will not change even after DMA becomes active; it will retain its setting.

(9) ATAPI control 2 register (ATAPI_CONTROL2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	WORD SWAP	IFEN
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	—	R	Reserved
1	WORDSWAP	0	R/W	This bit controls whether to exchange the upper 16-bit data for the lower 16-bit data if the 32-bit enhanced bus is enabled. 0: No word swap is executed. The 32-bit data on the enhanced bus is represented in big endian format. 1: Word swap is executed between the ATAPI interface and the register/enhanced bus. The 32-bit data on the enhanced bus is represented in little endian format. Word swap in data transfer is enabled only if bit 0 in the ATAPI control register is set to 1 to initiate DMA mode. All types of register access are made in longword units in any mode other than DMA mode.
0	IFEN	0	R/W	This bit controls whether to enable the ATAPI interface. 0: ATAPI interface is disabled. 1: ATAPI interface is enabled. Note: When this bit is 0, the I/O pins of the ATAPI interface work as input pins, and the output pins are in high-impedance state.

(10) ATAPI signal status register (ATAPI_SIG_ST)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DDMARDY	DMARQ
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	—	R	Reserved
1	DDMARDY	—	R	This bit indicates the state of the ATAPIDDMARDY (inversion of IDEIORDY) signal.
0	DMARQ	—	R	This bit indicates the state of the ATAPIDMARQ (IDEDREQ) signal.

(11) Byte swap register (ATAPI_BYTE_SWAP)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BYTE SWAP
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	—	R	Reserved
0	BYTESWAP	0	R/W	This bit controls whether to swap the upper eight bits with the lower eight bits on the ATAPI interface. 1: Byte swap is executed between the APAPI interface and enhanced bus. Byte swap is enabled only if bit 0 in the ATAPI control register is 1 and DMA mode has initiated.

25.4 Operation

The ATAPI interface supports the primary channel as a host. It also supports a master/slave configuration as stipulated in the ATAPI interface specification. The FIFO read/write buffer of the ATAPI interface is designed to transfer data at up to 16 Mbytes/s in multiword DMA mode and up to 33 Mbytes/s in ultra DMA mode. The ATAPI interface supports the 3.3V I/O interface.

The ATA task file register and ATAPI packet command file register are allocated in the on-chip peripheral module space of this LSI device. Therefore, accessing these registers from the LSI device can be made by addressing the on-chip register of the DVDROM drive or the like with the DCS1, DCS0, and DSA2 to DSA0 pins.

25.4.1 Data Transfer Modes

The ATAPI interface control register supports PIO transfer modes, multiword DMA transfer modes, and ultra DMA transfer modes. The ATAPI interface control register is used to initiate each transfer mode and set the ATAPI interface timing that varies from one transfer mode to another.

The transfer modes supported with the ATAPI interface include PIO modes 0 to 4, multiword DMA modes 0 to 2, and ultra DMA modes 0 to 2.

The enhanced bus is used for multiword and ultra DMA data transfers, and the peripheral bus, for PIO transfers.

Table 25.5 Data Transfer Modes

Internal Operation and Internal Register	Data Transfer Mode		
	PIO Data Transfer	DMA Data Transfer between ATA Device and Enhanced Bus	
		Multiword DMA	Ultra DMA
FIFO operation	Bypass*	Used	Used
UDMAEN bit in control register	Don't Care	0	1
START/STOP bit in control register	Not Used	Used	Used

Note: * The CPU accesses the ATA device in PIO mode.
In enhanced bus DMA transfers, data is transferred between the ATAPI device and memory.

25.4.2 Initialization Procedure

(1) Setting the interface enable bit

Set the IFEN bit in ATAPI control 2 register to 1.

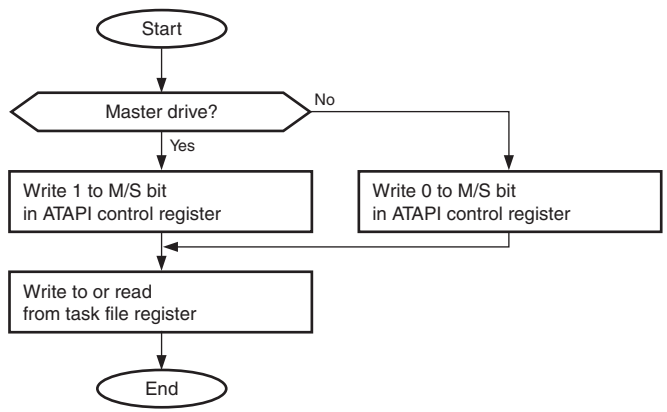
(2) Setting the timing register

Write an appropriate value to the following registers.

See the respective descriptions for what value is appropriate for each register.

- PIO timing register
- Multiword DMA timing register
- Ultra DMA timing register

25.4.3 PIO Transfer Mode Operation Procedure



Note: Never execute a PIO transfer while the ACT bit in the ATAPI status register is 1.

Figure 25.5 PIO Transfer Mode Operation Procedure

25.4.4 Multiword DMA Transfer Mode Operation Procedure

(1) Transfer to and from memory via enhanced bus by polling

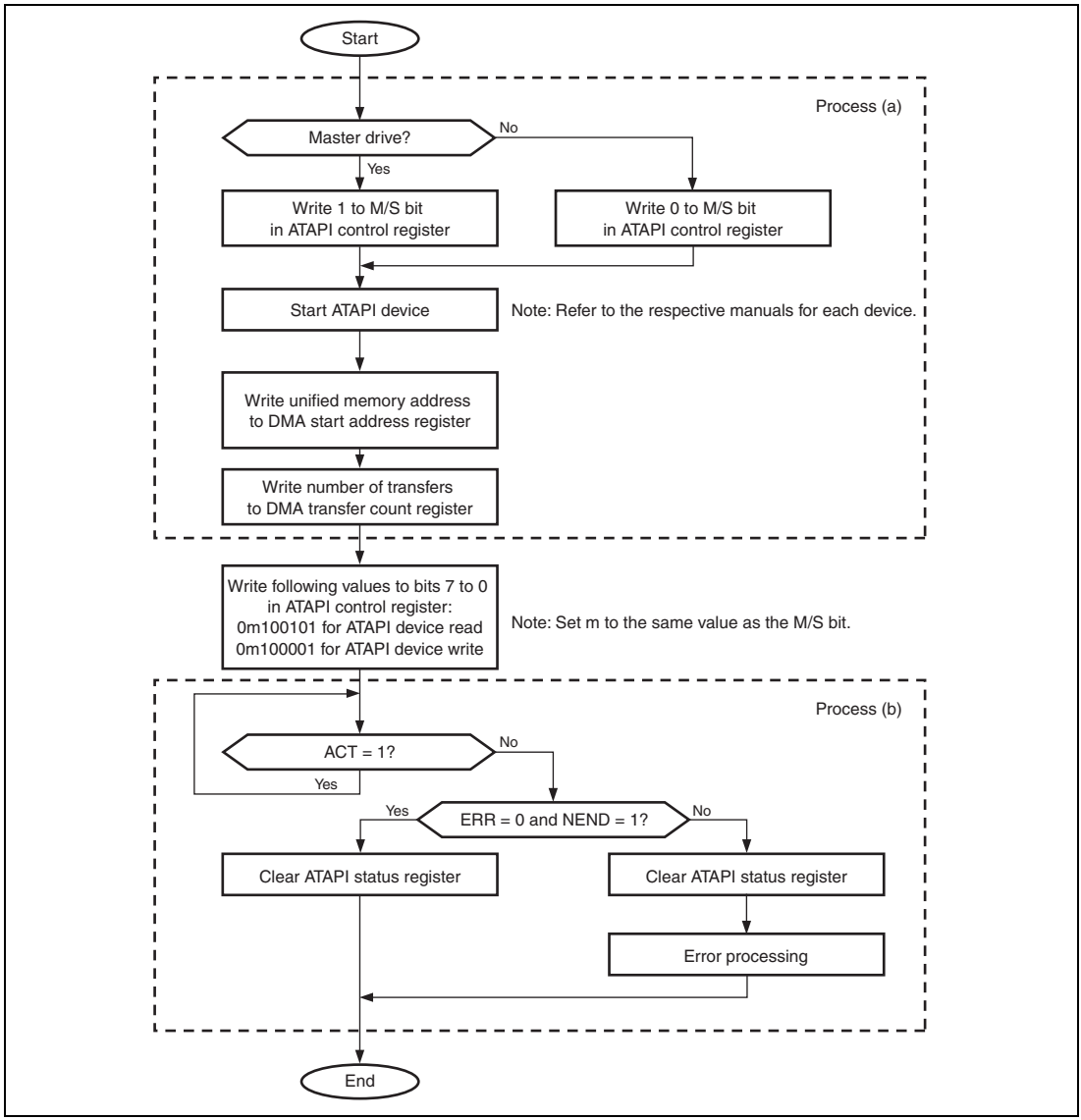


Figure 25.6 Transfer to and from Memory via Enhanced Bus by Polling

(2) Transfer to and from memory via enhanced bus by interrupt

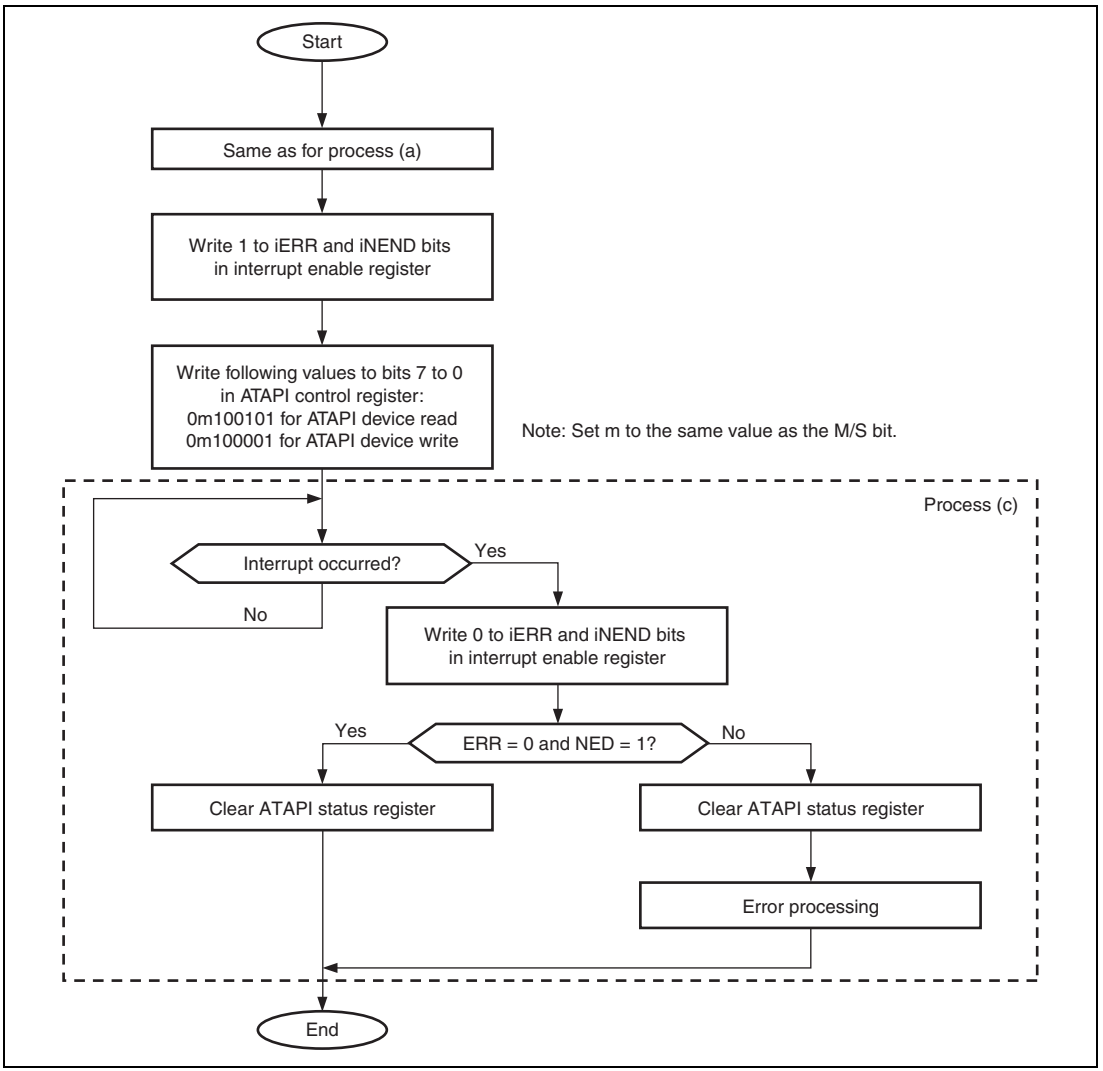


Figure 25.7 Transfer to and from Memory via Enhanced Bus by Interrupt

25.4.5 Ultra DMA Transfer Mode Operation Procedure

(1) Transfer to and from memory via enhanced bus by polling

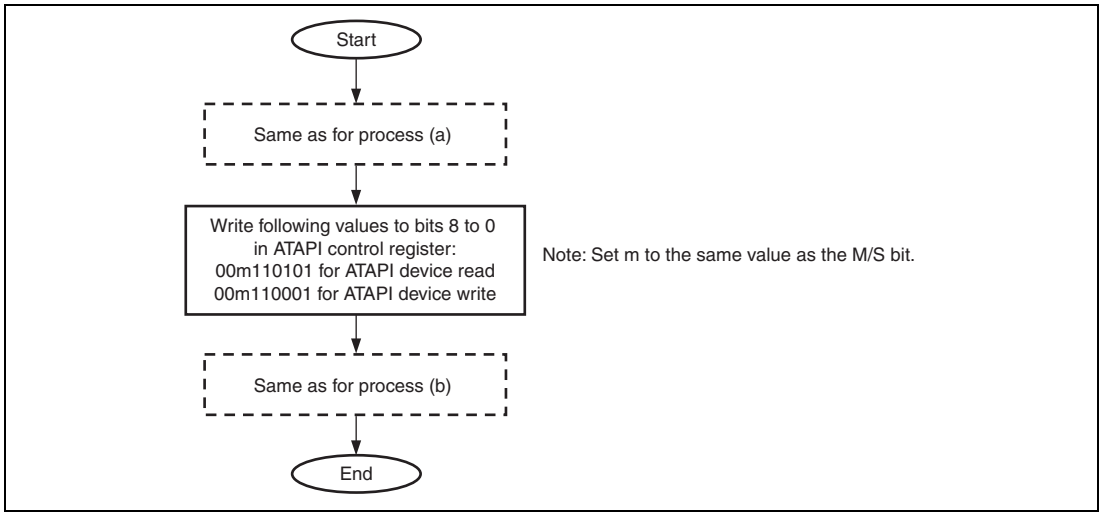


Figure 25.8 Transfer to and from Memory via Enhanced Bus by Polling

(2) Transfer to and from memory via enhanced bus by interrupt

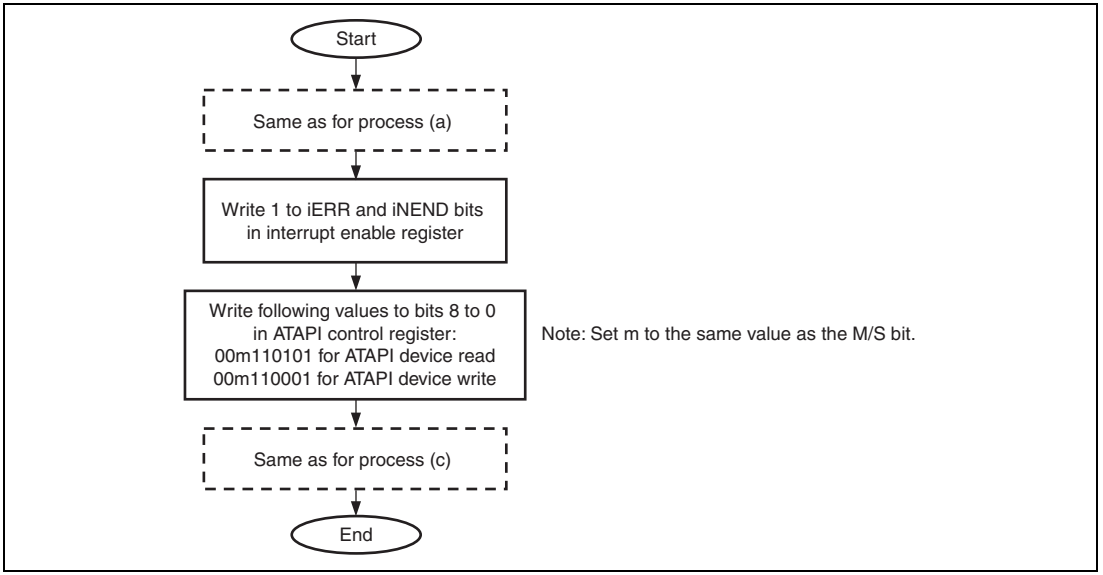


Figure 25.9 Transfer to and from Memory via Enhanced Bus by Interrupt

25.4.6 ATAPI Device Hardware Reset Procedure

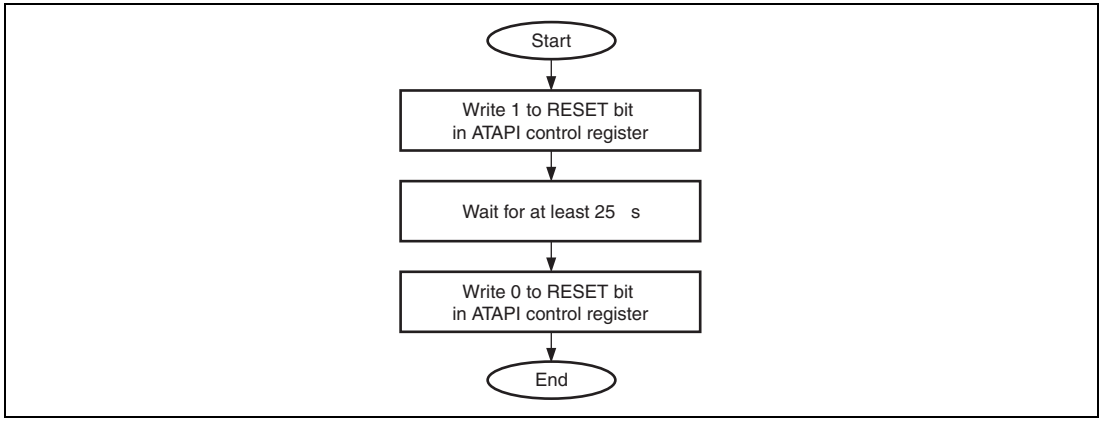


Figure 25.10 ATAPI Device Hardware Reset Procedure

25.5 DIRECTION Pin

The DIRECTION pin outputs a low level while data is being written to an external ATA device by this LSI.

To be specific, the DIRECTION pin goes low at:

- PIO data transfer to an ATA device
- Multiword DMA transfer (data-out)
- Ultra DMA data-in CRC transmission
- Ultra DMA transfer data-out

See section 33, Electrical Characteristics for descriptions about the timing of each transfer mode.

25.6 Usage Note

When using the ATAPI module, set the frequencies of the bus clock and the peripheral clock so that their ratio is 2:1.

Section 26 2D Graphics Engine (2DG)

The 2DG engine (below referred to as the 2DG) is provided as a two-dimensional graphics accelerator that has the following functions:
alpha blending of two areas specified as rectangles followed by resizing and the output of enlarged or reduced rectangular areas: and
the resizing of externally supplied moving pictures to the size of the display panel, followed by composition with the output graphics plane, and outputting to VIDEO OUT (D/A converter) at a constant rate.

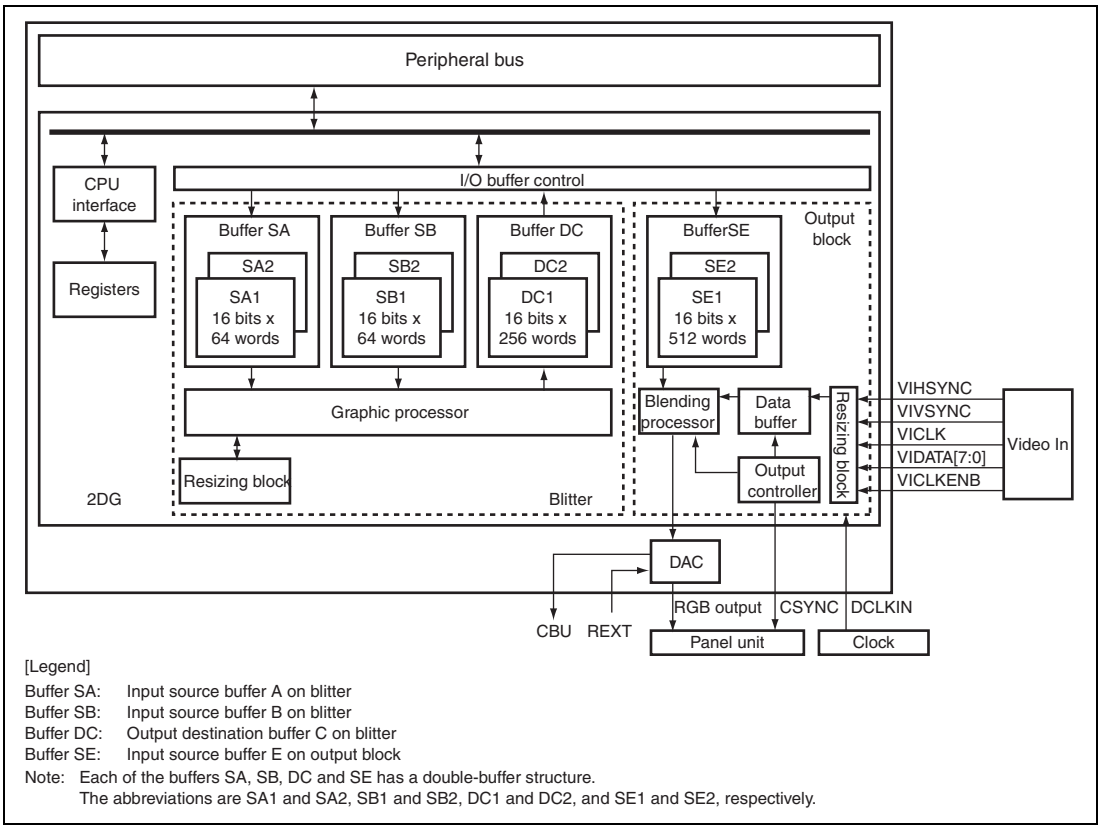
The graphics data is transferred between the SDRAM and the 2DG at high speed by the DMAC under CPU control. All of the required data, including the source and destination addresses for the SDRAM area and the source and destination addresses for the 2DG are issued by the DMAC. The 2DG blit operations are the specified processing for the data supplied to the source buffers and the output to the destination buffer.

26.1 Features

- Planes: Major examples (characters, graphics plane, and output plane, totally two planes)
- Acceleration: blitting with two inputs and one output, filling, bit-blitting, chromakey, logical operations, color gradation handling, variable blending operations
- Resizing,
  - Blitter: Bilinear and nearest-neighbor method are independently selectable for the horizontal and vertical directions (conversion ratios from 1/2 to 2)
    Selection of pre-filtering (as a measure against the Moire effect) on or off.
  - Output block: Bi-cubic algorithm in the horizontal direction (conversion ratios: from 1/3 to 1)
- Moving picture input: BT656 format (NTSC/PAL system) input (requiring both the VIHSYNC and VIVSYNC signal inputs.)
- Superimposition on moving pictures: Alpha blending of the graphics plane and moving picture, followed by constant-rate output in RGB666 format
- Input pixel formats for blit blocks: αRGB444 (16 bits), αRGB555 (16 bits), or α(4 bits)
- Output pixel formats for blit blocks: αRGB444 (16 bits), or αRGB555 (16 bits)

- Final picture resolution: WQVGA (480 x 234) or QVGA (320 x 240)
- Capacity of input/output buffers for graphics (each is in a double-buffer configuration)
  - — Input buffer E for the output block: 16bits x 512 words x 2 planes
  - — Input buffers A and B for the blitter: 16 bits x 64 words x 2 planes, each
  - — Output buffer C for the blitter: 16 bits x 256 words x 2 planes

Figure 26.1 is a block diagram of the 2DG.



[Legend]
Buffer SA: Input source buffer A on blitter
Buffer SB: Input source buffer B on blitter
Buffer DC: Output destination buffer C on blitter
Buffer SE: Input source buffer E on output block
Note: Each of the buffers SA, SB, DC and SE has a double-buffer structure.
The abbreviations are SA1 and SA2, SB1 and SB2, DC1 and DC2, and SE1 and SE2, respectively.

Figure 26.1 Block Diagram of the 2DG

26.2 Input/Output Pins

Table 26.1 Pin Configuration

Name	Pin	In/Out	Description
External HSYNC input	VIHSYNC	Input	HSYNC signal input from a video decoder
External VSYNC input	VIVSYNC	Input	VSYNC signal input from a video decoder
External clock input	VICLK	Input	Clock signal input from a video decoder
External data input	VIDATA[7:0]	Input	Data signal input from a video decoder
External enable input	VICLKENB	Input	Enable signal from a video decoder
External clock input	DCLKIN	Input	Clock signal input for RGB display
CSYNC output pin	CSYNC	Output	Composite SYNC output signal for RGB display
Display data R output	R	Output	Analog R signal output for RGB display
Display data G output	G	Output	Analog G signal output for RGB display
Display data B output	B	Output	Analog B signal output for RGB display
External reference	REXT	Input	Control signal for analog output amplitude
External capacitor	CBU	Output	Phase compensation signal for the internal amplifier
2DG D/A converter analog power supply	2DGAPVcc0, 2DGAPVcc1	Input	Power supply pin of 2DG D/A converter
2DG D/A converter analog ground	2DGAPVss0, 2DGAPVss1	Input	Ground pin of 2DG D/A converter

26.3 Register Descriptions

The 2DG has the following registers. During operation in synchronization with the VSYNC signal, register values are applied to the 2DG when the VSYNC signals are low pulse. However, the read/write to the applicable register is irrelevant to the VSYNC synchronization.

Table 26.2 Configuration of Registers

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Blit function setting register for graphics	GR_BLTPLY	R/W	H'00000000	H'E8000000	16, 32
Mixing function setting register for graphics (synchronized with VSYNC)	GR_MIXPLY	R/W	H'00000000	H'E8000004	16, 32
Operation status register for graphics	GR_DOSTAT	R	H'00000000	H'E8000008	16, 32
Interrupt status register for graphics	GR_IRSTAT	R	H'00000000	H'E800000C	16, 32
Interrupt mask control register for graphics	GR_INTMSK	R/W	H'00007171	H'E8000010	16, 32
Interrupt reset control register for graphics	GR_INTDIS	W	H'00000000	H'E8000014	16, 32
DMAC-request control register for graphics	GR_DMAC	R/W	H'30000010	H'E8000020	16, 32
Source A&B read-in-area setting register for blitter	GR_SABSET	R/W	H'00000000	H'E8000030	16, 32
Destination C write area setting register for blitter	GR_DCSET	R/W	H'00000001	H'E8000038	16, 32
Source E read-in area setting register for output block (synchronized with VSYNC)	MGR_SESET	R/W	H'00000000	H'E8000040	16, 32
Pixel format setting register for graphics (only one bit, SE_FMT, is synchronized with VSYNC)	GR_PIXLFMT	R/W	H'00000000	H'E8000048	16, 32
Operation mode setting register for blitter	GR_BLTMODE	R/W	H'00000000	H'E8000050	16, 32
Resize display setting register for graphics	GR_RISZSET	R/W	H'00010300	H'E8000060	16, 32

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Resize mode select register for blitter	GR_RISZMOD	R/W	H'00000404	H'E8000064	16, 32
Resize delta setting register for blitter	GR_DELT	R/W	H'00000000	H'E8000068	16, 32
Resize horizontal starting phase register for blitter	GR_HSPHAS	R/W	H'00000000	H'E800006C	16, 32
Resize vertical starting phase register for blitter	GR_VSPHAS	R/W	H'00000000	H'E8000070	16, 32
Resize horizontal delta setting register for output block (synchronized with VSYNC)	MGR_HDELT	R/W	H'00001800	H'E8000074	16, 32
Resize horizontal starting phase register for output block (synchronized with VSYNC)	MGR_HPHAS	R/W	H'00000000	H'E8000078	16, 32
Logical operation input data register for blitter	GR_LGDAT	R/W	H'00000000	H'E8000080	16, 32
Chromakey target color data register for blitter	GR_DETCOL	R/W	H'00000000	H'E8000084	16, 32
Replacement color data register for blitter blending	GR_BRDCOL	R/W	H'00000000	H'E8000088	16, 32
Blend 1 control register for blitter	GR_BRD1CNT	R/W	H'00000000	H'E800008C	16, 32
Mixing mode setting register for output block (synchronized with VSYNC)	MGR_MIXMODE	R/W	H'00000000	H'E8000098	16, 32
Panel-output horizontal timing setting register for output block (synchronized with VSYNC)	MGR_MIXHTMG	R/W	H'0005000F	H'E80000A0	16, 32
Panel-output mixing horizontal valid area setting register for output block (synchronized with VSYNC)	MGR_MIXHS	R/W	H'00370020	H'E80000A4	16, 32
Panel-output vertical timing setting register for output block (synchronized with VSYNC)	MGR_MIXVTMG	R/W	H'00003004	H'E80000A8	16, 32
Panel-output mixing vertical valid area setting register for output block (synchronized with VSYNC)	MGR_MIXVS	R/W	H'000D0007	H'E80000AC	16, 32

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Output SYNC position setting register for graphics	GR_VSDLY	R/W	H'00000160	H'E80000C4	16, 32
Video DAC timing setting register	VDAC_TMC	R/W	H'00000000	H'EA000000	32

26.3.1 Blit Function Setting Register for Graphics (GR_BLTPLY)

Register GR_BLTPLY is used to enable blitting.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SB_STEN	SA_STEN
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 2	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
1	SB_STEN	0	R/W	Enable Blitting from Source B. This bit enables blitting from source B or makes source B blitting wait. 0: Waiting mode 1: Execution is enabled or in progress. On completion of the blit operation, the hardware automatically clears the bit to 0.
0	SA_STEN	0	R/W	Enable Blitting from Source A This bit enables blitting from source A or makes source A blotting wait 0: Waiting mode 1: Execution is enabled or in progress. On completion of the blit operation, the hardware automatically clears the bit to 0.

Setting values of the SB_STEN and SA_STEN bits for blit operations are shown below.

Table 26.3 Settings of the SB_STEN and SA_STEN Bits and Blit Operations

SB_STEN	SA_STEN	
0	0	Waiting mode
	1	Setting is prohibited. Blitting does not proceed.
1	0	Blitting is enabled but only for Source B.
	1	Blitting is enabled for both Source A and Source B.

- Blit operations only for source SA alone are not possible. If only a single source signal is supplied, use source SB.
- If the value 0 is written to during blitting, the operation is ended forcibly.
- The value "1" should never be written to these bits when neither buffer SA nor buffer SB is empty. Whether the buffers are empty or not is checked by the state of the GR_DOSTAT register.
- When both of the bits SB_STEN and SA_STEN are set to 1, blit operation only proceeds when the same amounts of data are to be transferred from buffers SA and SB (for details, see section 26.4.3 (2), Summary of Operations between the Blitter and External Memory.)

26.3.2 Mixing Function Setting Register for Graphics (GR_MIXPLY)

The register GR_MIXPLY specifies the display of externally input pictures and graphics. The register value is applied to the 2DG in synchronization with the VSYNC signal.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	EXTEN	-	-	-	OUTEN
Initial value:	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 5	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
4	EXTEN	0	R/W	Display Enable Bit for Externally Input Pictures This bit enables or disables the display of externally input pictures. 0: Disabled 1: Enabled
3 to 1	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
0	OUTEN	0	R/W	Graphics Display Enable This bit enables or disables the graphics display. 0: Disabled 1: Enabled

Setting values of the EXTEN and OUTEN bits for the graphics display are shown below.

Table 26.4 Setting Value of the EXTEN and OUTEN bits for Blit Operation

OUTEN	EXTEN	
0	0	Display of both externally input pictures and graphics is disabled. The result is a black display.
	1	Only display of externally input pictures is enabled.
1	0	Only graphics display is enabled.
	1	Display of images composed from externally input pictures and graphics is enabled.

26.3.3 Operation Status Register for Graphics (GR_DOSTAT)

The register GR_DOSTAT indicates the operating status of the 2DG.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DISP_STAT		SEHF_STAT		-	-	DCHF_STAT		SBHF_STAT		SAHF_STAT		-	-	SB_REND	SA_REND
Initial value:	0	0	0	0	-	-	0	0	0	0	0	0	-	-	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit name	Initial Value	R/W	Description
31 to 16	—	Undefined	R	Reserved The read value is undefined.
15, 14	DISP_STAT	00	R	Display State for Output Block These bits indicate the display being fed to the output block. 00: No image 01: Externally input pictures only 10: Graphics only 11: Images composed from externally input pictures and graphics.

Bit	Bit name	Initial Value	R/W	Description
13, 12	SEHF_STAT	00	R	Input Buffer E Half-Control for the Output Block These bits indicate the states of buffers SE1 (bit 12) and SE2 (bit 13). 00: Both SE1 and SE2 are empty. 01: The size of SE1 coincides but buffer SE2 is empty. 10: The size of SE2 coincides but buffer SE1 is empty. 11: The sizes of buffers SE1 and SE2 coincide.
11, 10	—	Undefined	R	Reserved The read value is undefined.
9, 8	DCHF_STAT	00	R	Output Buffer C Half-Control of Destination Data from the Blitter These bits indicate the states of buffers DC1 (bit 8) and DC2 (bit 9). 00: Both DC1 and DC2 are empty. 01: The size of DC1 coincides but buffer DC2 is empty. 10: The size of DC2 coincides but buffer DC1 is empty 11: The sizes of buffers DC1 and DC2 coincide.
7, 6	SBHF_STAT	00	R	Input Buffer B Half-Control of Data Source for the Blitter These bits indicate the state of the buffers SB1 (bit 6) and SB2 (bit 7). 00: Both SB1 and SB2 are empty. 01: The size of SB1 coincides but buffer SB2 is empty. 10: The size of SB2 coincides but buffer SB1 is empty. 11: The sizes of both buffers SB1 and SB2 coincide.

Bit	Bit name	Initial Value	R/W	Description
5, 4	SAHF_STAT	00	R	Input Buffer A Half-control of Data Source for the Blitter These bits indicate the states of the buffers SA1 (bit 4) and SA2 (bit 5). 00: Both SA1 and SA2 are empty. 01: The size of SA1 coincides but buffer SA2 is empty. 10: The size of SA2 coincides but buffer SA1 is empty. 11: The size of both buffers SA1 and SA2 coincide.
3, 2	—	Undefined	R	Reserved The read value is undefined.
1	SB_REND	0	R	Access Mode of Input Buffer B, Source for the Blitter This bit indicates the access mode for buffer B. 0: On completion of access for blitting or on standby mode 1: Access to buffer SB is completed and access to buffer DC is underway.
0	SA_REND	0	R	Access mode of Input Buffer A, Source for the Blitter This bit indicates the access mode for buffer A. 0: On completion of access for blitting or on standby mode 1: Access to buffer SA is completed and access to buffer DC is underway.

- Each SEHF_STAT bit changes from 0 to 1 when the corresponding buffer SE1 or SE2 is full or when the amount of pixel data in the SE buffer coincides with the specified number of pixels.
  Furthermore, the SEHF_STAT bit changes from 1 to 0 on completion of reading the data in the corresponding half of SE buffer (or the data remaining herein).
- Each DCHF_STAT bit changes from 0 to 1 when the corresponding buffer DC1 or DC2 is full or when the amount of pixel data in the DC buffer coincides with the specified number of pixels.
  Furthermore, the DCHF_STAT bit changes from 1 to 0 on completion of DMA transfer of the data in the corresponding half of DC buffer (or the data remaining herein).

- Each SBHF_STAT bit changes from 0 to 1 when the corresponding buffer SB1 or SB2 is full or when the amount of pixel data in the SB buffer coincides with the specified number of pixels.
  Furthermore, the SBHF_STAT bit changes from 1 to 0 on completion of reading the data in the corresponding half of SB buffer (or the data remaining herein).
- Each SAHF_STAT bit changes from 0 to 1 when the corresponding buffer SA1 or SA2 is full or when the amount of pixel data in the SA buffer coincides with the specified number of pixels.
  Furthermore, the SAHF_STAT bit changes from 1 to 0 on completion of reading the data in the corresponding half of SA buffer (or the data remaining herein).
- "Coincide" in the above list means that the width setting in register GR_SABSET, GR_DCSET, or MGR_SESET matches the number of data transferred to or from the relevant buffer or that the corresponding half of the double-buffer is full.
- If an abnormal state arises, such as stopping of graphics operations before they are completed, use this register identify the cause of the problem.
  - If the SB_REND or SA_REND bit is being held at 1, check the settings of registers related to buffer DC (for example, the value of GR_DCSET).
  - If the SB_REND or SA_REND bit is being held at 0, and one buffer of the SB and SA double-buffers remains full, check the settings of registers related to buffers SB and SA (for example, the value of GR_SABSET).
  - If the blitter is reactivated, write 1 to bits SB_STEN and SA_STEN in register GR_BLTPLY.
  - If buffer SE is empty, write 0 to bits OUTEN and EXTEN in register GR_MIXPLY. Output can then be restarted.

26.3.4 Interrupt Status Register for Graphics (GR_IRSTAT)

The register GR_IRSTAT indicates the interrupt state of the 2DG. When an interrupt event corresponding to the IRQ_DEMPT, IRQ_ASHFUL, IRQ_DHFUL, or IRQ_SHFUL bit in this register occurs, the given bit will be set as long as the event has not been masked by the MSK_DEMPT, MSK_ASHFUL, MSK_DHFUL, or MSK_SHFUL bit in the GR_INTMSK register. For the other bits in register GR_IRSTAT, the bit will be set to "1" when the corresponding event occurs, regardless of the setting in the interrupt mask control register for graphics (GR_INTMSK). For details on interrupts, see section 26.4.5, Interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	IRQ_DEMPT	-	IRQ_ASHFUL	IRQ_DHFUL	IRQ_SHFUL	-	-	-	-
Initial value:	-	-	-	-	-	-	-	0	-	0	0	0	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	INT_VSYC	INT_UDFL	INT_FILD	-	-	-	INT_DEMPT	-	INT_ASHFUL	INT_DHFUL	INT_SHFUL	-	-	-	INT_GR
Initial value:	-	0	0	0	-	-	-	0	-	0	0	0	-	-	-	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit name	Initial Value	R/W	Description
31 to 25	—	Undefined	R	Reserved The read value is undefined.
24	IRQ_DEMPT	0	R	Input Buffer E Full Flag for the Output Block This bit indicates that the input buffer E for the output block is full. 0: Input buffer E for the output block is not full. 1: Input buffer E for the output block is full. [Clearing condition] • Writing 1 to the DIS_DEMPT bit of register GR_INTDIS. [Setting condition] • Input buffer E for the output block is full.
23	—	Undefined	R	Reserved The read value is undefined.

Bit	Bit name	Initial Value	R/W	Description
22	IRQ_ASHFUL	0	R	Blitter Input Buffer A Full Flag This bit indicates that the input buffer A for the blitter is full. 0: Input buffer A for the blitter is not full. 1: Input buffer A for the blitter is full. [Clearing condition] • Writing 1 to the DIS_ASHFUL bit of register GR_INTDIS. [Setting condition] • Input buffer A for the output block is full.
21	IRQ_DHFUL	0	R	Blitter Output Buffer C Full Flag This bit indicates that the output buffer C for the blitter is full. 0: Output buffer C for the blitter is not full. 1: Output buffer C for the blitter is full. [Clearing condition] • Writing 1 to the DIS_DHFUL bit of register GR_INTDIS. [Setting condition] • Output buffer C for the blitter is full.
20	IRQ_SHFUL	0	R	Blitter Input Buffer B Full Flag This bit indicates that the input buffer B for the blitter is full. 0: Input buffer B for the blitter is not full. 1: Input buffer B for the blitter is full. [Clearing condition] • Writing 1 to the DIS_SHFUL bit of register GR_INTDIS. [Setting condition] • Input buffer B for the blitter is full.
19 to 15	—	Undefined	R	Reserved The read value is undefined.

Bit	Bit name	Initial Value	R/W	Description
14	INT_VSYC	0	R	VSYNC Input for the Output Block This bit indicates the state of VSYNC input. 0: VSYNC input signal is not being supplied 1: VSYNC input signal is being supplied. (This bit is only effective when display is enabled by GR_MISPLY.) [Clearing condition] • Writing 1 to the DIS_VSYC bit of register GR_INTDIS. [Setting condition] • The VSYNC input is supplied.
13	INT_UDFL	0	R	Output Underflow for the Output Block This bit indicates underflow of the output from the output block. 0: Output from the output block is normal. 1: Output from the output block underflowed. [Clearing condition] • Writing 1 to the DIS_UDFL bit of register GR_INTDIS. [Setting condition] • Underflow of output from the output block.
12	INT_FILD	0	R	Last Line Captured by Output Block This bit indicates that the output block has finished capturing the last line in the SE buffer. 0: Last line is not in the output buffer E. 1: Last line has captured in output buffer E. [Clearing condition] • Writing 1 to the DIS_FILD bit of register GR_INTDIS. [Setting condition] • Last line being captured in input buffer E for the output block.

Bit	Bit name	Initial Value	R/W	Description
11 to 9	—	Undefined	R	Reserved The read value is undefined.
8	INT_DEMPT	0	R	Output Block Input Buffer E Full This bit indicates that the input buffer E for the output block is full. 0: Input buffer E for the output block is not full. 1: Input buffer E for the output block is full. [Clearing condition] • The hardware automatically clears the bit, when either buffer SE1 or SE2 is empty. [Setting condition] • Input buffer E for the output block is full.
7	—	Undefined	R	Reserved The read value is undefined.
6	INT_ASHFUL	0	R	Blitter Input Buffer A Full This bit indicates that the input buffer A for the blitter is full. 0: Input buffer A for the blitter is not full. 1: Input buffer A for the blitter is full. [Clearing condition] • The hardware automatically clears the bit, when either buffer SA1 or SA2 is empty. [Setting condition] • Input buffer A for the output block is full.
5	INT_DHFUL	0	R	Blitter Output Buffer C Full This bit indicates that the output buffer C for the blitter is full. 0: Output buffer C for the blitter is not full. 1: Output buffer C for the blitter is full. [Clearing condition] • The hardware automatically clears the bit, when either buffer DC1 or DC2 is empty. [Setting condition] • Output buffer C for the blitter is full.

Bit	Bit name	Initial Value	R/W	Description
4	INT_SHFUL	0	R	Blitter Input Buffer B Full This bit indicates that the input buffer B for the blitter is full. 0: Input buffer B for the blitter is not full. 1: Input buffer B for the blitter is full. [Clearing condition] • The hardware automatically clears the bit, when either buffer SB1 or SB2 is empty. [Setting condition] • Input buffer B for the blitter is full.
3 to 1	—	Undefined	R	Reserved The read value is undefined.
0	INT_GR	0	R	Blitter Operation Completion This bit indicates whether the blitter operation has or has not been completed. 0: The blitter operation is in progress or no blitter operation has not been set up. 1: The blitter operation has been completed. [Clearing condition] • Writing 1 to the DIS_GR bit in GR_INTDIS. [Setting condition] • Completion of the blitter operation

Note: The INT_UDFL and INT_VSYC bits may be set even when the output block has not been started. So, be sure to clear the INT_UDFL and INT_VSYC bits in the GR_INTDIS register before starting up the output block.

26.3.5 Interrupt Mask Control Register for Graphics (GR_INTMSK)

The register GR_INTMSK masks 2DG interrupts. When an interrupt event occurs, the interrupt status register for graphics (GR_IRSTAT) will be set even if the corresponding interrupt is not enabled (masked). For details on interrupts, see section 26.4.5, Interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	MSK_VSYC	MSK_UDFL	MSK_FILD	-	-	-	MSK_DEMPT	-	MSK_ASHFUL	MSK_DHFUL	MSK_SHFUL	-	-	-	MSK_GR
Initial value:	-	1	1	1	-	-	-	1	-	1	1	1	-	-	-	1
R/W:	R	R/W	R/W	R/W	R	R	R	R/W	R	R/W	R/W	R/W	R	R	R	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 15	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
14	MSK_VSYC	1	R/W	Output Block VSYNC Input Interrupt Mask This bit masks a VSYNC input interrupt for the output block. 0: Enables a VSYNC input interrupt for the output block. 1: Masks a VSYNC input interrupt for the output block.
13	MSK_UDFL	1	R/W	Output Block Output Underflow Interrupt Mask This bit masks an output underflow interrupt for the output block. 0: Enables an output underflow interrupt for the output block. 1: Masks an output underflow interrupt for the output block.

Bit	Bit name	Initial Value	R/W	Description
12	MSK_FILD	1	R/W	Output Block Last Line Capture Completion Interrupt Mask This bit masks a last line capture completion interrupt for the output block. 0: Enables a last line capture completion interrupt for the output block. 1: Masks a last line capture completion interrupt for the output block.
11 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8	MSK_DEMPT	1	R/W	Output Block Input Buffer E Full Interrupt Mask This bit masks an input buffer E full interrupt for the output block. 0: Enables an input buffer E full interrupt for the output block. 1: Masks an input buffer E full interrupt for the output block.
7	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
6	MSK_ASHFUL	1	R/W	Blitter Input Buffer A Full Interrupt Mask This bit masks an input buffer A full interrupt for the blitter. 0: Enables an input buffer A full interrupt for the blitter. 1: Masks an input buffer A full interrupt for the blitter.
5	MSK_DHFUL	1	R/W	Blitter Output Buffer C Full Interrupt Mask This bit masks an output buffer C full interrupt for the blitter. 0: Enables an output buffer C full interrupt for the blitter. 1: Masks an output buffer C full interrupt for the blitter.

Bit	Bit name	Initial Value	R/W	Description
4	MSK_SHFUL	1	R/W	Blitter Input Buffer B Full Interrupt Mask This bit masks an input buffer B full interrupt for the blitter. 0: Enables an input buffer B full interrupt for the blitter. 1: Masks an input buffer B full interrupt for the blitter.
3 to 1	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
0	MSK_GR	1	R/W	Blitter Operation Completion Interrupt Mask This bit masks a blitter operation completion interrupt. 0: Enables a blitter operation completion interrupt. 1: Masks a blitter operation completion interrupt.

- Note that the MGR_MIXHTMG, MGR_MIXHS, and MGR_MIXVTMG registers should be set according to the display panel to be used before the interrupts masked by the MSK_UDFL and MSK_FILD bits are cancelled.

26.3.6 Interrupt Reset Control Register for Graphics (GR_INTDIS)

The register GR_INTDIS cancels 2DG interrupts. Interrupt signals are deasserted by writing 1 to the corresponding bits in this register. Furthermore, the IRQ_DEMPT, IRQ_ASHFUL, IRQ_DHFUL, IRQ_SHFUL, INT_VSYC, INT_UDFL, INT_ FILD, and INT_GR bits in GR_IRSTAT are cleared by writing 1 to the corresponding bits in this register. Note, however, that the INT_DEMP, INT_ASHFUL, INT_DHFUL, and INT_SHFUL bits in GR_IRSTAT are not cleared even if 1 is written to the corresponding bits in this register (the hardware automatically handles clearing of these bits). When a 1 is written to any of these bits, the hardware automatically sets the bit to its initial value. For details on interrupts, see section 26.4.5, Interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	DIS_ VSYC	DIS_ UDFL	DIS_ FILD	-	-	-	DIS_ DEMPT	-	DIS_ ASHFUL	DIS_ DHFUL	DIS_ SHFUL	-	-	-	DIS_ GR
Initial value:	-	0	0	0	-	-	-	0	-	0	0	0	-	-	-	0
R/W:	R	W	W	W	R	R	R	W	R	W	W	W	R	R	R	W

Bit	Bit name	Initial Value	R/W	Description
31 to 15	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
14	DIS_VSYC	0	W	Output Block VSYNC Input Interrupt Cancellation This bit cancels a VSYNC input interrupt for the output block. 0: Retains the current status. 1: Cancels a VSYNC input interrupt for the output block.
13	DIS_UDFL	0	W	Output Block Output Underflow Interrupt Cancellation This bit cancels an output underflow interrupt for the output block. 0: Retains the current status. 1: Cancels an output underflow interrupt for the output block.

Bit	Bit name	Initial Value	R/W	Description
12	DIS_FILD	0	W	Output Block Last Line Capture Completion Interrupt Cancellation This bit cancels a last line capture completion interrupt for the output block. 0: Retains the current status. 1: Cancels a last line capture completion interrupt for the output block
11 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8	DIS_DEMPT	0	W	Output Block Input Buffer E Full Interrupt Cancellation This bit cancels an input buffer E full interrupt for the output block. 0: Retains the current status. 1: Cancels an input buffer E full interrupt for the output block.
7	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
6	DIS_ASHFUL	0	W	Blitter Input Buffer A Full Interrupt Cancellation This bit cancels an input buffer A full interrupt for the blitter. 0: Retains the current status. 1: Cancels an input buffer A full interrupt for the blitter.
5	DIS_DHFUL	0	W	Blitter Output Buffer C Full Interrupt Cancellation This bit cancels an output buffer C full interrupt for the blitter. 0: Retains the current status. 1: Cancels an output buffer C full interrupt for the blitter.

Bit	Bit name	Initial Value	R/W	Description
4	DIS_SHFUL	0	W	Blitter Input Buffer B Full Interrupt Cancellation This bit cancels an input buffer B full interrupt for the blitter. 0: Retains the current status. 1: Cancels an input buffer B full interrupt for the blitter.
3 to 1	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
0	DIS_GR	0	W	Blitter Operation Completion Interrupt Cancellation This bit cancels a blitter operation completion interrupt. 0: Retains the current status. 1: Cancels a blitter operation completion interrupt.

26.3.7 DMAC-Request Control Register for Graphics (GR_DMAC)

The register GR_DMAC sets DMA transfer and CPU transfer control methods for SA, SB, DC and SE buffers. Note that the settings for this register should be the same as the corresponding settings in the DMAC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	SZSEL2	SZSEL1	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	1	1	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	DM1_DSEL		DM2_DSEL		DM34_DSEL		-	-	DM1_MSEL		DM2_MSEL		DM34_MSEL	
Initial value:	-	-	0	0	0	0	0	0	-	-	0	1	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31, 30	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
29	SZSEL2	1	R/W	Output Block DMA Transfer Data Size This bit sets the size (number of bits) of data used for DMA transfer in the output block. 0: 16 bits 1: 32 bits
28	SZSEL1	1	R/W	Blitter DMA Transfer Data Size This bit sets the size (number of bits) of data used for DMA transfer in the blitter. 0: 16 bits 1: 32 bits
27 to 14	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

Bit	Bit name	Initial Value	R/W	Description
13, 12	DM1_DSEL	00	R/W	SE Buffer DMA Transfer Condition These bits set the conditions for DMA transfer to the SE buffer. 00: Single operand transfer 01: Continuous operand transfer 10: Reserved 11: Reserved
11, 10	DM2_DSEL	00	R/W	DC Buffer DMA Transfer Condition These bits set the conditions for DMA transfer from the DC buffer. 00: Single operand transfer 01: Continuous operand transfer 10: Reserved 11: Reserved
9, 8	DM34_DSEL	00	R/W	SA/SB Buffer DMA Transfer Condition These bits set the conditions for DMA transfer to the SA/SB buffer. 00: Single operand transfer 01: Continuous operand transfer 10: Reserved 11: Reserved
7, 6	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
5, 4	DM1_MSEL	01	R/W	SE Buffer DMA Transfer Mode These bits set the transfer mode for DMA transfer to the SE buffer. 00: Cycle stealing transfer 01: Pipeline transfer 10: Reserved 11: CPU transfer

Bit	Bit name	Initial Value	R/W	Description
3, 2	DM2_MSEL	00	R/W	DC Buffer DMA Transfer Mode These bits set the transfer mode for DMA transfer from the DC buffer. 00: Cycle stealing transfer 01: Pipeline transfer 10: Reserved 11: CPU transfer
1, 0	DM34_MSEL	00	R/W	SA/SB Buffer DMA Transfer Mode These bits set the transfer mode for DMA transfer to the SA/SB buffer. 00: Cycle stealing transfer 01: Pipeline transfer 10: Reserved 11: CPU transfer

Note: When the H1PHS_INTGR bits in GR_HSPHAS are odd (H1PS_INTGR [0] = 1), be sure to set the SZEL1 bit to 0 (16 bits).

26.3.8 Source A&B Read-In-Area Setting Register for Blitter (GR_SABSET)

The register GR_SABSET sets the SA and SB areas. In a DMA transfer, the total number of pixels to be transferred from the external memory space is obtained by SSWIDH × SSHIGH.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	SSHIGH								
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	SSWIDH								
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 25	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
24 to 16	SSHIGH	H'000	R/W	SA/SB Area Vertical Setting These bits set the vertical height (number of lines) of the rectangular area (SA or SB area) to be transferred. Valid range: 1 to 288 lines
15 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8 to 0	SSWIDH	H'000	R/W	SA/SB Area Horizontal Setting These bits set the horizontal width (number of pixels) of the rectangular area (SA or SB area) to be transferred. Valid range: 1 to 511 pixels

- When both SA and SB areas are used, it is assumed that SB (foreground) > SA (background). For example, if two-plane blending is performed while SB and SA are assumed to be a character and background, respectively, blending is performed so that the character will always be foreground.
- If the pixel format for SB area is set to α (4 bits), the SSWIDH bits in GR_SABSET should be set as follows.

  In 16-bit access: For the SSWIDH bits in GR_SABSET, the minimum number of pixels to be transferred and the pixel transfer unit should be set to 4 and 4 × n (n: arbitrary integer), respectively. Setting 2 or 3 pixels to these bits is prohibited.

  In 32-bit access: For the SSWIDH bits in GR_SABSET, the minimum number of pixels to be transferred and the pixel transfer unit should be set to 8 and 8 × n (n: arbitrary integer), respectively.
- If the pixel format is set to αRGB444 or αRGB555, only 16-bit access is enabled when the total number of pixels to be transferred (SSWIDH × SSHIGH) is odd; both 32-bit and 16-bit accesses are enabled when the value of SSWIDH × SSHIGH is even. If duplicate-lines are set for enlargement resizing, either 32-bit access or 16-bit access should be selected according to the number of pixels on a line (SSWIDH), not according to the total number of pixels to be transferred (SSWIDH × SSHIGH).

26.3.9 Destination C Write Area Setting Register for Blitter (GR_DCSET)

The register GR_DCSET sets the DC area. In a DMA transfer, the total number of pixels to be transferred to the external memory space is obtained by DCWIDH × DCHIGH.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	DCHIGH								
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	DCWIDH								
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 25	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
24 to 16	DCHIGH	H'000	R/W	DC Area Vertical Setting These bits set the vertical height (number of lines) of the rectangular area (DC area) to be transferred. Valid range: 1 to 288 lines
15 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8 to 0	DCWIDH	H'001	R/W	DC Area Horizontal Setting These bits set the horizontal width (number of pixels) of the rectangular area (DC area) to be transferred. Valid range: 1 to 511 pixels

- If the pixel format is set to αRGB444 or αRGB555, only 16-bit access is enabled when the total number of pixels to be transferred (DCWIDH × DCHIGH) is odd; both 32-bit and 16-bit accesses are enabled when the value of DCWIDH × DCHIGH is even.

26.3.10 Source E Read-In Area Setting Register for Output Block (MGR_SESET)

The register MGR_SESET sets the SE area. In a DMA transfer, the total number of pixels to be transferred from the external memory space is obtained by SEWIDH × SEHIGH. The register value is applied to the 2DG in synchronization with the VSYNC signal.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	SEHIGH								
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	SEWIDH								
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 25	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
24 to 16	SEHIGH	H'000	R/W	SE Area Vertical Setting These bits set the vertical height (number of lines) of the rectangular area (SE area) to be transferred. Valid range: 2 to 288 lines
15 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8 to 0	SEWIDH	H'000	R/W	SE Area Horizontal Setting These bits set the horizontal width (number of pixels) of the rectangular area (SE area) to be transferred. Valid range: 2 to 511 pixels

- Only 16-bit access is enabled when the total number of pixels to be transferred (SEWIDH × SEHIGH) is odd; both 32-bit and 16-bit accesses are enabled when the value of (SEWIDH × SEHIGH) is even.
- The settings of this register should be the same as those of the MGR_MIXHS and MGR_MIXVS registers.

  SEWIDH bits = VLDPH bits in MGR_MIXHS
  SEHIGH bits = VLDPV bits in MGR_MIXVS

- So that the display does not break up, make settings such that DMA transfer to the SE buffer is efficient. Some recommendations follow as examples.
  - SEWIDH bits: Set these bits for a number of pixels that is a multiple of eight (320, 480, and so on).
  - Output block DMA transfer size bit: Set this to one (32 bits).
  - Amount transferred per operand: Set this to a large value.
  - DMAC transfer mode: Select pipeline transfer.

26.3.11 Pixel Format Setting Register for Graphics (GR_PIXLFMT)

The register GR_PIXLFMT sets the pixel formats which are used for the input and output buffers. The SE_FMT bit is applied to the 2DG in synchronization with the VSYNC signal. Bits other than the SE_FMT bit are applied to the 2DG immediately.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SE_FMT
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	DC_FMT	-	-	SB_FMT		-	-	-	SA_FMT
Initial value:	-	-	-	-	-	-	-	0	-	-	0	0	-	-	-	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R	R	R	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 17	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
16	SE_FMT	0	R/W	SE Image Format This bit specifies the format of an image data sent to SE. 0: αRGB444 (16 bits) 1: αRGB555 (16 bits)
15 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8	DC_FMT	0	R/W	DC Image Format This bit specifies the format of an image data sent from DC. 0: αRGB444 (16 bits) 1: αRGB555 (16 bits)
7, 6	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

Bit	Bit name	Initial Value	R/W	Description
5, 4	SB_FMT	00	R/W	SB Image Format These bits specify the format of an image sent to SB. 00: αRGB444 (16 bits) 01: αRGB555 (16 bits) 10: α (4 bits) 11: Reserved
3 to 1	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
0	SA_FMT	0	R/W	SA Image Format This bit specifies the format of an image data sent to SA. 0: αRGB444 (16 bits) 1: αRGB555 (16 bits)

- If the SB_FMT bits are set to 10, the SSWIDH bits in GR_SABSET should be set as follows:
  In 16-bit access: For the SSWIDH bits in GR_SABSET, the minimum number of pixels to be transferred and the pixel transfer unit should be set to 4 and 4 × n (n: arbitrary integer), respectively. Setting 2 or 3 pixels to these bits is prohibited.
  In 32-bit access: For the SSWIDH bits in GR_SABSET, the minimum number of pixels to be transferred and the pixel transfer unit should be set to 8 and 8 × n (n: arbitrary integer), respectively.

26.3.12 Operation Mode Setting Register for Blitter (GR_BLTMODE)

The register GR_BLTMODE sets the blitter operation mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	CRKEY		-	-	LGTYPE		SBSEL		BTYPE	
Initial value:	-	-	-	-	-	-	0	0	-	-	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 10	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
9, 8	CRKEY	00	R/W	Chromakey Type Select These bits select the type of chromakey. 00: No chromakey processing. The settings for chromakey target color (in GR_DETCOL) and replacement color (in GR_BRDCOL) are ineffective. 01: Replaces the chromakey target color specified in GR_DETCOL with the replacement color specified in GR_BRDCOL. 10: Blends the chromakey target color specified in GR_DETCOL with the replacement color specified in GR_BRDCOL. 11: Setting prohibited
7, 6	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

Bit	Bit name	Initial Value	R/W	Description
5, 4	LGTYPE	00	R/W	Logical Operation Type Select These bits select the type of logical operation. 00: Setting prohibited 01: XORs SB data and GR_LGDAT register. 10: ORs SB data and GR_LGDAT regsister. 11: Inverts the result obtained by XORing SB data and GR_LGDAT register.
3, 2	SBSEL	00	R/W	SB Output Data Select These bits select data output from SB after various processings. 00: SB data 01: Data after chromakey processing 10: Data after logical operation 11: Data after color gradation processing
1, 0	BTYPE	00	R/W	Blitting Mode These bits set the blitting modes. 00: Blit operation (Data is input to SA or SB and output to DC.) 01: Setting prohibited 10: Filling operation (The blending is performed using data input to SB and register values. The result is output to DC.) 11: Setting prohibited

- The CRKEY bits are valid only when the SBSEL bits are set to 01 and the BTYPE bits are set to 10.
- When the chromakey is enabled (when the SBSEL bits are set to 01 and the BTYPE bits are set to 10), operation is the same as when the GCOLR bit in GR_BRD1CNT is set to 1. However, the value of the GCOLR bit does not change.
- The LGTYPE bit is valid only when the SBSEL bits are set to 10.
- The SBSEL bits can be set to 00, 01, or 10 when the filling operation is selected (BTYPE = 10), and they can be set to 00, 10, or 11 when the blitting operation is selected (BTYPE = 00).
- If the BTYPE bits are set to 00, setting chromakey is prohibited.

26.3.13 Resize Display Setting Register for Graphics (GR_RISZSET)

The register GR_RISZSET sets the resizing function.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PREON	-	-	EDGE		-	-	-	-	-	-	-	BRSIZ
Initial value:	-	-	-	0	-	-	1	1	-	-	-	-	-	-	-	0
R/W:	R	R	R	R/W	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 13	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
12	PREON	0	R/W	Blitter Pre-Filtering This bit sets whether or not to perform pre-filtering for the blitter. 0: Does not perform pre-filtering 1: Performs pre-filtering
11, 10	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

Bit	Bit name	Initial Value	R/W	Description
9, 8	EDGE	11	R/W	Blitter Edge Processing These bits set processings of the right edge in the horizontal direction and lower edge in the vertical direction for the blitter. When resizing is not performed, these bits should be set to 11. When resizing is performed, these bits can be set as follows. 00: The vertical edge is not the lower edge. The horizontal edge is not the right edge. 01: The vertical edge is not the lower edge. The horizontal edge is the right edge. 10: The vertical edge is the lower edge. The horizontal edge is not the right edge. 11: The vertical edge is the lower edge. The horizontal edge is the right edge.
7 to 1	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
0	BRSIZ	0	R/W	Blitter Resizing Function This bit sets whether or not to use the resizing function for the blitter. 0: Does not use the resizing function. 1: Uses the resizing function.

- When the resizing function is used for the blitter, the EDGE bits should be set as follows:

  The EDGE bits should be set to 11 if full resizing which resizes the entire source area is performed.

  The EDGE bits should be set according to the bit description above if partial resizing which resizes a part of the source area is performed.
- The interference stripes can be reduced by setting the PREON bit to 1 and performing pre-filtering.

26.3.14 Resize Mode Select Register for Blitter (GR_RISZMOD)

The register GR_RISZMOD sets the resizing function for the blitter.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	A1_H	-	H1_ MTHD	-	-	-	-	-	A1_V	-	V1_ MTHD
Initial value:	-	-	-	-	-	1	-	0	-	-	-	-	-	1	-	0
R/W:	R	R	R	R	R	R/W	R	R/W	R	R	R	R	R	R/W	R	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 11	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
10	A1_H	1	R/W	Horizontal α Resizing Method This bit selects α resizing method in the horizontal direction. 0: Bilinear method 1: Nearest-neighbor method
9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8	H1_MTHD	0	R/W	Horizontal Resizing Method This bit selects a resizing method in the horizontal direction. 0: Bilinear method 1: Nearest-neighbor method
7 to 3	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

Bit	Bit name	Initial Value	R/W	Description
2	A1_V	1	R/W	Vertical α Resizing Method This bit selects α resizing method in the vertical direction. 0: Bilinear method 1: Nearest-neighbor method
1	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
0	V1_MTHD	0	R/W	Vertical Resizing Method This bit selects a resizing method in the vertical direction. 0: Bilinear method 1: Nearest-neighbor method

Note: The bits in GR_RISZMOD must be set when resizing is performed. These bits need not to be set when resizing is not performed.

26.3.15 Resize Delta Setting Register for Blitter (GR_DELT)

The register GR_DELT sets delta computation results for resizing on the blitter.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	VDLT_INTGR		VDLT_DCML											
Initial value:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	HDLT_INTGR		HDLT_DCML											
Initial value:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31, 30	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

Bit	Bit name	Initial Value	R/W	Description
29, 28	VDLT_INTGR	00	R/W	Vertical-Direction Delta Computation Result Integer Part These bits set the integer part of the delta computation result in the vertical direction.
27 to 16	VDLT_DCML	H'000	R/W	Vertical-Direction Delta Computation Result Fractional Part These bits set the fractional part of the delta computation result in the vertical direction.
15, 14	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
13, 12	HDLT_INTGR	00	R/W	Horizontal-Direction Delta Computation Result Integer Part These bits set the integer part of the delta computation result in the horizontal direction.
11 to 0	HDLT_DCML	H'000	R/W	Horizontal-Direction Delta Computation Result Fractional Part These bits set the fractional part of the delta computation result in the horizontal direction.

Note: This register must be set before resizing. If resizing is not performed, this register need not be set.

- If one line of data to be transferred by the CPU is 65 pixels or more when an enlargement resizing in the vertical direction is performed, there may be a case that the pixels at the same lines should be transferred twice by the CPU. For details, see section 26.4.3 (6), Duplicate-line Setting of Enlargement Resizing.

26.3.16 Resize Horizontal Starting Phase Register for Blitter (GR_HSPHAS)

The register GR_HSPHAS sets results of the starting position phase computation in the horizontal direction for the blitter resizing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	H1PHS_DCML											
Initial value:	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	H1PHS_INTGR									
Initial value:	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 28	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
27 to 16	H1PHS_DCML	H'000	R/W	Horizontal Starting Position Phase Computation Result Fractional Part These bits set the fractional part of the starting-position phase computation result in the horizontal direction on the source side.
15 to 10	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
9 to 0	H1PHS_INTGR	H'000	R/W	Horizontal Starting Position Phase Computation Result Integer Part These bits set the integer part of the starting-position phase computation result in the horizontal direction on the source side.

Notes: 1. This register must be set before resizing. All bits should be cleared to 0 when resizing is not performed.

2. When the H1PHS_INTGR bits are odd (H1PHS_INTGR [0] = 1), be sure to set the SZEL1 bit in GR_DMAC to 0 (16 bits).

26.3.17 Resize Vertical Starting Phase Register for Blitter (GR_VSPHAS)

The register GR_VSPHAS sets results of the starting position phase computation in the vertical direction for the blitter resizing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	V1PHS_DCML											
Initial value:	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	V1PHS_INTGR								
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 28	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
27 to 16	V1PHS_DCML	H'000	R/W	Vertical Starting Position Phase Computation Result Fractional Part These bits set the fractional part of the starting-position phase computation result in the vertical direction on the source side.
15 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8 to 0	V1PHS_INTGR	H'000	R/W	Vertical Starting Position Phase Computation Result Integer Part These bits set the integer part of the starting-position phase computation result in the vertical direction on the source side.

Note: This register must be set before resizing. All bits should be cleared to 0 when resizing is not performed.

26.3.18 Resize Horizontal Delta Setting Register for Output Block (MGR_HDELT)

The register MGR_HDELT sets the delta computation results in the horizontal direction for the output block resizing. The register value is applied to the output block in synchronization with the VSYNC signal.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MHDLT_INTGR				MHDLT_DCML											
Initial value:	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 16	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
15 to 12	MHDLT_INTGR	0001	R/W	Horizontal Delta Computation Result Integer Part These bits set the integer part of the delta computation result in the horizontal direction.
11 to 0	MHDLT_DCML	H'800	R/W	Horizontal Delta Computation Result Fractional Part These bits set the fractional part of delta computation result in the horizontal direction.

26.3.19 Resize Horizontal Starting Phase Register for Output Block (MGR_HPHAS)

The register MGR_HPHAS sets results of the starting position phase computation in the horizontal direction for the output block resizing. The register value is applied to the output block in synchronization with the VSYNC signal.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	MH1PHS_DCML											
Initial value:	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 12	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
11 to 0	MH1PHS_DCML	H'000	R/W	Horizontal Starting Position Phase Computation Result Fractional Part These bits set the fractional part of the starting-position phase computation result in the horizontal direction on the source side.

26.3.20 Logical Operation Input Data Register for Blitter (GR_LGDAT)

The register GR_LGDAT sets data for logical operation to be performed on the blitter. Since logical operation is performed after format conversion, data should be specified as 5 bits of LGDAT_R (R data), 5 bits of LGDAT_G (G data) and 5 bits of LGDAT_B (B data).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	LGDAT_A				-	-	-	LGDAT_R				
Initial value:	-	-	-	-	0	0	0	0	-	-	-	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	LGDAT_G					-	-	-	LGDAT_B				
Initial value:	-	-	-	0	0	0	0	0	-	-	-	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 28	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
27 to 24	LGDAT_A	0000	R/W	Logical Operation Data (α) These bits set logical operation data (α) = α log.
23 to 21	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
20 to 16	LGDAT_R	00000	R/W	Logical Operation Data (R) These bits set logical operation data (R) = Clog_r.
15 to 13	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
12 to 8	LGDAT_G	00000	R/W	Logical Operation Data (G) These bits set logical operation data (G) = Clog_g.
7 to 5	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
4 to 0	LGDAT_B	00000	R/W	Logical Operation Data (B) These bits set logical operation data (B) = Clog_b.

Note: This register is valid only when the logical operation function has been selected (SBSEL in GR_BLTMODE = 10)

26.3.21 Chromakey Target Color Data Register for Blitter (GR_DETCOL)

The register GR_DETCOL sets the target color for chromakey (α data is not set). Since chromakey processing is performed after format conversion, data should be specified as 5 bits of DETC_R (R data), 5 bits of DETC_G (G data) and 5 bits of DETC_B (B data).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	DETC_R				
Initial value:	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	DETC_G					-	-	-	DETC_B				
Initial value:	-	-	-	0	0	0	0	0	-	-	-	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 21	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
20 to 16	DETC_R	00000	R/W	Chromakey Target Color Data (R) These bits set chromakey target color data (R) = Cdasg_r
15 to 13	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
12 to 8	DETC_G	00000	R/W	Chromakey Target Color Data (G) These bits set chromakey target color data (G) = Cdasg_g.
7 to 5	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
4 to 0	DETC_B	00000	R/W	Chromakey Target Color Data (B) These bits set chromakey target color data (B) = Cdasg_b

- This register is valid only when the chromakey function has been selected (SBSEL in GR_BLTMODE = 01)
- This register value is used differently depending on the CRKEY bits in GR_BLTMODE:

  When CRKEY = 01: The target color specified in this register is replaced with the replacement color specified in GR_BRDCOL for the SB input data.

  When CRKEY = 10: The target color specified in this register is blended with the replacement color specified in GR_BRDCOL for the SB input data.

26.3.22 Replacement Color Data Register for Blitter Blending (GR_BRDCOL)

The register GR_BRDCOL is used for chromakey processing and color gradation processing. In chromakey processing, the color specified in this register replaces or is blended with the target color. In color gradation processing, the color specified in this register is used as the replacement color. Since chromakey processing and color gradation processing are performed after format conversion, the data should be specified as 5 bits of BRDC_R (R data), 5 bits of BRDC_G (G data) and 5 bits of BRDC_B (B data).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	BRDC_A				-	-	-	BRDC_R				
Initial value:	-	-	-	-	0	0	0	0	-	-	-	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	BRDC_G					-	-	-	BRDC_B				
Initial value:	-	-	-	0	0	0	0	0	-	-	-	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 28	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
27 to 24	BRDC_A	0000	R/W	Blending Replacement Color Data (α) These bits set the replacement color (α) = αsasg for blending
23 to 21	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

Bit	Bit name	Initial Value	R/W	Description
20 to 16	BRDC_R	00000	R/W	Blending Replacement Color Data (R) These bits set the replacement color (R) = Csasg_r for blending
15 to 13	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
12 to 8	BRDC_G	00000	R/W	Blending Replacement Color Data (G) These bits set the replacement color (G) = Csasg_g for blending
7 to 5	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
4 to 0	BRDC_B	00000	R/W	Blending Replacement Color Data (B) These bits set the replacement color (B) = Csasg_b for blending

26.3.23 Blend 1 Control Register for Blitter (GR_BRD1CNT)

The register GR_BRD1CNT specifies settings for blending on the blitter. For details, see section 26.4.3 (3) (a), Blending.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	AFTER_A			
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	FBFA		-	-	-	GALFA	-	-	-	GCOLR
Initial value:	-	-	-	-	-	-	0	0	-	-	-	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 20	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

Bit	Bit name	Initial Value	R/W	Description
19 to 16	AFTER_A	0000	R/W	Alpha-Value Replacement Data Used after Blending (α) These bits set alpha-value replacement data used after blending (α) = α after.
15 to 10	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
9, 8	FBFA	00	R/W	Blending Coefficient Select These bits set a value for blending coefficient. For the combination, see table 26.5.
7 to 5	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
4	GALFA	0	R/W	Global ALPHA This bit sets to turn on or off the global ALPHA. Only the α value after blending can be changed to the AFTER_A bit. 0: Turns off (α dc = α out) 1: Turns on (α dc = AFTER_A bit)
3 to 1	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
0	GCOLR	0	R/W	Global Color This bit sets to turn on or off the global color. 0: Turns off (Ca1 = Ca) 1: Turns on (Ca1 = GR_BRDCOL)

Note: For symbols above, see figure 26.23.

Table 26.5 FBFA Bit Details

FBFA (Register Value)	Fb	Fa	Remarks
00 (Initial value)	αb	(1-αb)αa	2-input plane processing is performed. αout = Fb+Fa
01	1	0	1-input plane processing is performed. αout = αb
10	αb	(1-αb)	Blending without using αa is performed. αout = αb
11	0	1	Only the SA or GR_BRDCOL register value is output. αout = αa

- When the fill operation has been selected (BTYPE bits in BR_BLTMODE = 10), the GCOLR bit should be set to 1 and the FBFA bits are arbitrary (basically cleared to 00 because fill operation with blending is performed.)
- The AFTER_A bits are valid only when GALFA = 1.
- FBFA should be set to 01 when chromakey processing has been selected.
- GCOLR should be set to 1 to perform fill operation (BTYPE in GR_BLTMODE = 10).

26.3.24 Mixing Mode Setting Register for Output Block (MGR_MIXMODE)

The register MGR_MIXMODE sets mixing mode for the output block. The register value is applied to the output block in synchronization with the VSYNC signal. For details, see section 26.4.4 (5), Blending in Output Block.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	NTSC	-	-	-	-	CHG_A				-	FCFD		
Initial value:	-	-	-	0	-	-	-	-	0	0	0	0	-	0	0	0
R/W:	R	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	MVON	-	-	-	CBCR	-	-	-	-	-	-	-	VLD_N
Initial value:	-	-	-	0	-	-	-	0	-	-	-	-	-	-	-	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 29	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
28	NTSC	0	R/W	Output Block NTSC/PAL This bit sets NTSC/PAL for the output block. 0: NTSC 1: PAL
27 to 24	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
23 to 20	CHG_A	0000	R/W	Output Block α Value Replacement Data These bits set data to be replaced with α value in the αRGB555 format for the output block.
19	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
18 to 16	FCFD	000	R/W	Output Block α Blending Value These bits set an α-blending value for output block. Cp = (Fc $\times$ Cdc) + (Fd $\times$ Cv) is output from the blending section on the output block. For the combination, see table 26.6.

Bit	Bit name	Initial Value	R/W	Description
15 to 13	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
12	MVON	0	R/W	External Moving Picture Input Specification This bit sets to turn on or off the external moving picture input. 0: Turns off the external moving picture input for the system (synchronization with VSYNC generated internally) 1: Turns on the external moving picture input for the system with external moving picture input (pseudo-synchronization with external VIVSYNC)
11 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8	CBCR	0	R/W	CbCr Bit Position Swapping This bit selects whether CbCr position is swapped or not when the YCbCr422 →YCbCr444 conversion is performed. 0: Not swapped 1: Swapped
7 to 1	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
0	VLD_N	0	R/W	VICLKENB Polarity Select This bit selects the polarity of VICLKENB which is the VALID signal for external moving picture input. 0: L polarity during VALID 1: H polarity during VALID

Notes: 1. For symbols, see figure 26.55.
2. For details on the CBCR bit, see figure 26.15.
3. When a system with the low quality external moving picture input has been selected and displays only the graphic image, the MVON bit should be cleared to 0. By this setting, synchronization errors can be prevented by using internal SYNC signal, and this improves the quality of display. When MVON is set to 1, VIHSYNC or VIVSYNC which is synchronous with external moving picture input should be input.
4. When αRGB555 is selected for the pixel format (SE_FMT in GR_PIXFMT = 1) and α data of the pixel data is 1, four bits of α data are replaced with CHG_A.

5. MVON should be set to 1 before EXTEN in GR_MIXPLY is set to 1. When MVON = 0, setting EXTEN in GR_MIXPLY to 1 is prohibited.
6. Follow the procedure below when using the MVON bit to switch VSYNC between external and internal synchronization.
   (1) Set the GR_MIXPLY register to disable display.
   (2) Change the MVON bit to change the synchronization of VSYNC.
   (3) At least twice, check that VSYNC is being generated after the change.
   (4) Set the GR_MIXPLY register to enable display.

Table 26.6 FCFD Bit Details

FCFD (Register Value)	Fc	Fd	Remarks
000 (Initial value)	1	1 - αdc	SE buffer input image is premultiplied
001	αdc	1 - αdc	SE buffer input image is non-premultiplied
010	1	0	Only the graphics are output
011	0	1	Only the moving pictures are output.
100	0	0	Nothing is output (Black screen output)
Others	—	—	Reserved

Note: The FCFD bits are set automatically by the hardware according to the GR_MIXPLY settings as follows, however, the FCFD bit is not changed:
When only the externally supplied moving picture is selected (OUTEN = 0, EXTEN = 1): FCFD = 011
When only the graphic image is selected (OUTEN = 1, EXTEN = 0): FCFD = 010
When display is prohibited (OUTEN = 0, EXTEN = 0): FCFD = 100

26.3.25 Panel-Output Horizontal Timing Setting Register for Output-Block (MGR_MIXHTMG)

The register MGR_MIXHTMG sets the timing of signal output to the panel in the horizontal direction. The register value is applied in synchronization with the VSYNC signal. For details, see section 26.4.1 (5), Setting of Panel Output.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	WPH					
Initial value:	-	-	-	-	-	-	-	-	-	-	0	0	0	1	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PDPH								
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	1	1	1	1
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 22	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
21 to 16	WPH	H'05	R/W	Panel Output HSYNC Pulse Width These bits set the HSYNC pulse width for panel output using the number of DCLKIN from the falling edge of HSync_out. Valid range: 1 to 63 pixels
15 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8 to 0	PDPH	H'00F	R/W	Panel Output Image Horizontal Reading Start Timing These bits set the start timing to read image to be output to the panel in the horizontal direction using the number of DCLKIN from WPH. Valid range: 0 to 511 pixels

26.3.26 Panel-Output Mixing Horizontal Valid Area Setting Register for Output Block (MGR_MIXHS)

The register MGR_MIXHS sets a valid area for signal output to the panel in the horizontal direction. The register value is applied in synchronization with the VSYNC signal. For details, see section 26.4.1 (5), Setting of Panel Output.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	ALLPH									
Initial value:	-	-	-	-	-	-	0	0	0	0	1	1	0	1	1	1
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	VLDPH									
Initial value:	-	-	-	-	-	-	0	0	0	0	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 26	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
25 to 16	ALLPH	H'037	R/W	Panel Output Image Horizontal Width These bits set the horizontal width of panel output image using the number of DCLKIN from the rising edge of HSync_out. Valid range: 0 to 1023 pixels
15 to 10	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
9 to 0	VLDPH	H'020	R/W	Panel Output Image Horizontal Valid Width These bits set the valid width of panel output image in the horizontal direction using the number of DCLKIN from PDPH. Valid range: 0 to 511 pixels

Note: The settings in this register and in the source E read-in area for the output block (the MGR_SESET register) must be the same
The VLDPH bits are equivalent to the SEWIDH bits in MGR_SESET.

26.3.27 Panel-Output Vertical Timing Setting Register for Output-Block (MGR_MIXVTMG)

The register MGR_MIXVTMG sets the vertical timing of signal output to the panel. The register value is applied in synchronization with the VSYNC signal. For details, see section 26.4.1 (5), Setting of Panel Output.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WPV				-	-	-	PDPV								
Initial value:	0	0	1	1	-	-	-	0	0	0	0	0	0	1	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 16	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
15 to 12	WPV	0011	R/W	Panel Output VSYNC Pulse Width These bits set the pulse width of VSYNC for panel output using the number of lines from the falling edge of VSync_out. Valid range: 1 to 15 lines
11 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8 to 0	PDPV	H'004	R/W	Panel Output Image Vertical Reading Start Timing These bits set the start timing to read image to be output to the panel in the vertical direction using the number of lines from the rising edge of VSync_out. Valid range: 0 to 511 lines

26.3.28 Panel-Output Mixing Vertical Valid Area Setting Register for Output Block (MGR_MIXVS)

The register MGR_MIXVS sets the vertical area for signal output to the panel. The register value is applied in synchronization with the VSYNC signal. For details, see section 26.4.1 (5), Setting of Panel Output.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	ALLPV								
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	1	1	0	1
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	VLDPV								
Initial value:	-	-	-	-	-	-	-	0	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 25	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
24 to 16	ALLPV	H'00D	R/W	Panel Output VSYNCH Period Width These bits set high-level period of VSYNC for panel output using the number of lines. Valid range: 0 to 511 lines
15 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8 to 0	VLDPV	H'007	R/W	Panel Output Image Vertical Valid Width These bits set the valid width of panel output image in the vertical direction using the number of lines from PDPV. Valid range: 0 to 511 lines

Note: The settings in this register and in the source E read-in area for the output block (the MGR_SESET register) must be the same

The VLDPV bits are equivalent to the SEHIGH bits in MGR_SESET.

26.3.29 Graphics Block Output SYNC Position Setting Register (GR_VSDLY)

This register specifies the position of the output VSYNC signal. The vertical direction for the moving pictures can vary with the monitor in use. In situations where this is the case, this register can be adjusted to eliminate fluctuations in the vertical direction.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	VSDLY									
Initial value:	-	-	-	-	-	-	0	1	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 10	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
9 to 0	VSDLY	H'160	R/W	Output VSYNC Position Setting These bits adjust, in units of DCLKIN cycles, the amount of delay to the position for output of the VSYNC signal.

Notes: 1. When the external image is NTSC (when the NTSC bit in the MGR_MIXMODE register has been set to 0), use the initial value (H'160). When PAL is in use, after setting this register to H'100, set the NTSC bit in the MGR_MIXMODE register to 1.
2. Setting all bits of this register to 0 is prohibited.

26.3.30 Video DAC Timing Setting Register (VDAC_TMC)

The setting in this register is for the timing of output to a monitor.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	edgesel
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit name	Initial Value	R/W	Description
31 to 1	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
0	edgesel	0	R/W	Output timing setting This bit controls the timing of output to a monitor. 0: Output of analog RGB data is synchronized with rising edges of DLCKIN. 1: Output of analog RGB data is synchronized with falling edges of DLCKIN.

Note: When writing to this register, stop the 2DG module beforehand.

26.4 Operation

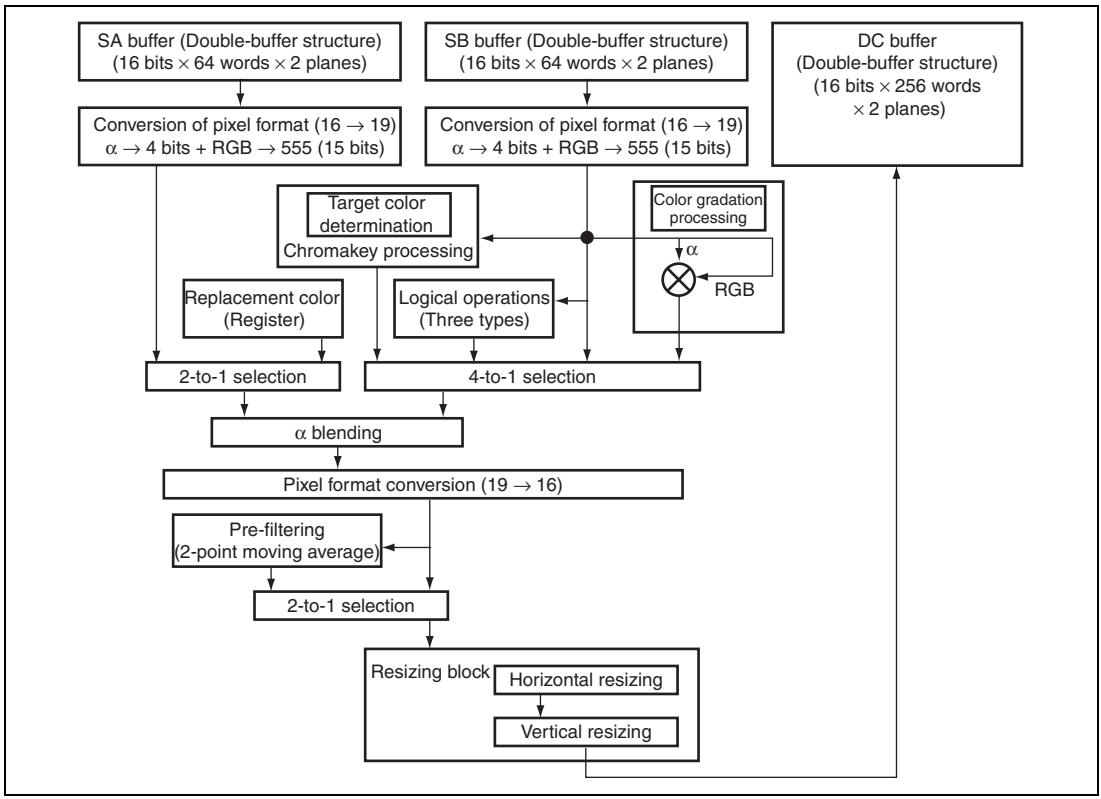


Figure 26.2 Block Diagram of the Blitter

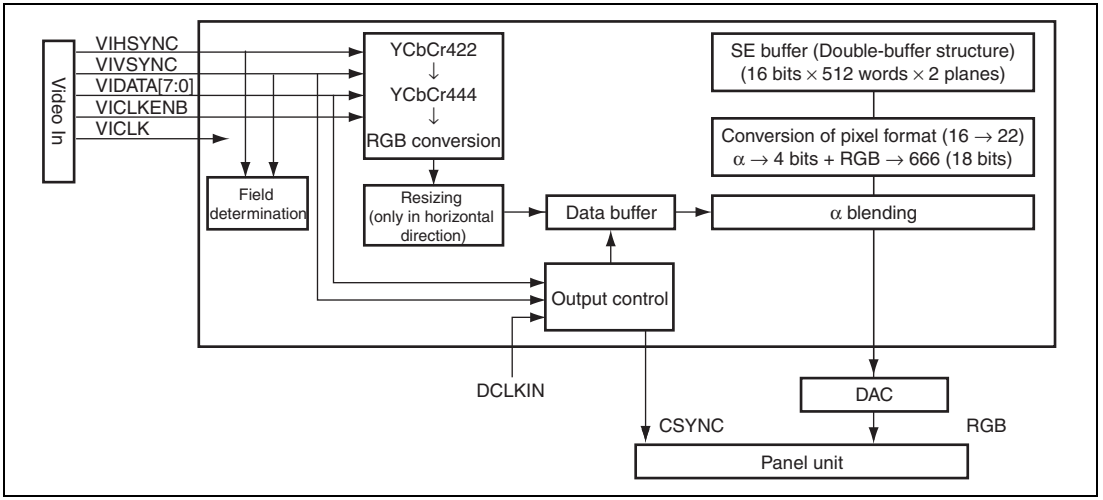


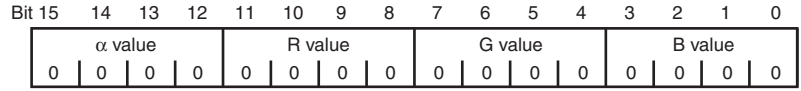
Figure 26.3 Block Diagram of the Output Block

26.4.1 Input and Output Operations

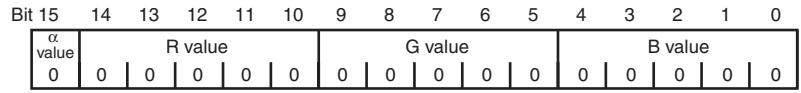
(1) Data Bit Map for the Pixel Format

Data in the αRGB444 (16 bits), αRGB555 (16 bits), and α (4 bits) pixel formats are applied to the inputs and output of the 2DG. The formats are shown below.

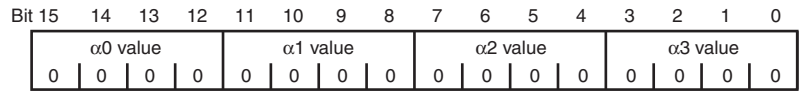
- αRGB444



- αRGB555



- α (4 bits)



Note: Transfer the α data from memory in 4 blocks unit.
Transfer of the data from α2 or α3 block is prohibited.

(2) Assignment of Pixels to the Memory Space

Input and output data are mapped on the memory space as shown in figure 26.4. This example shows the data mapping in the case where the selected format is αRGB444 (16 bits).

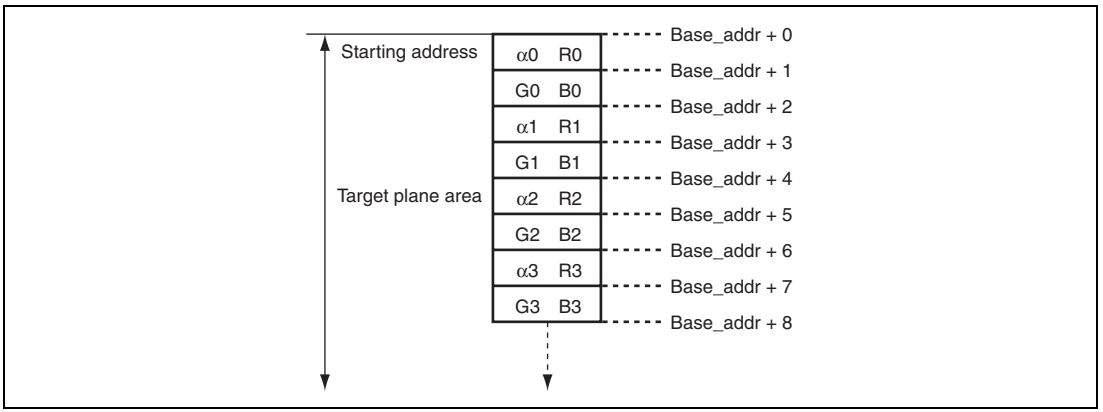


Figure 26.4 Example of Data Mapping (for the αRGB444 (16 bit) Pixel Format)

(3) Relations between Line Pitches and Memory Plane

The target display panels for the 2DG are QVGA (320 pixels × 240 lines) and WQVGA (480 pixels × 234 lines). The line pitch that determines the relations between memory space and each work screen (such as the character plane and graphics plane) of the SDRAM must be placed on the 64-byte boundaries. Thus, the start addresses of each planes become the following.

XXXX_XX [4n] [0] (H) (X = arbitrary number, n = integer)

So, the starting address must be one of these: XXXX_XX00, XXXX_XX40, XXXX_XX80, or XXXX_XXC0.

Figure 26.5 shows an example of the relations between the arrangement of planes A, B, and C, (WQVGA size) and the pitch of display lines in memory for the αRGB444 (16-bits) pixel format.

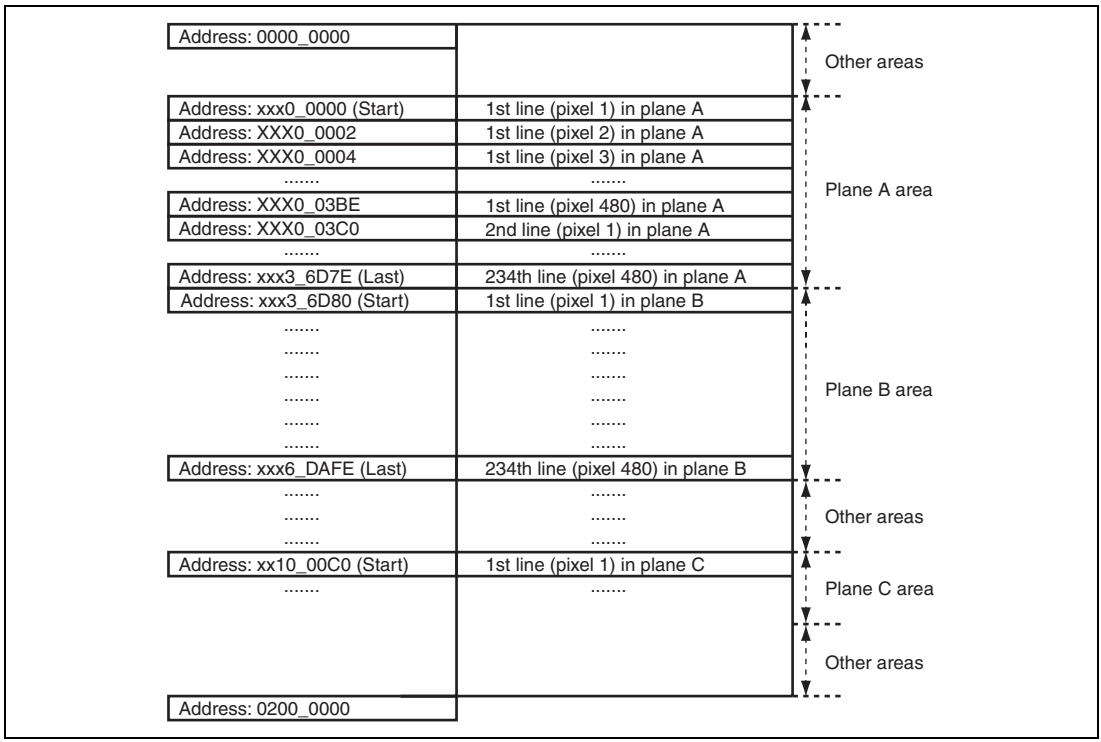


Figure 26.5 Relations between the Planes and Display-Line Pitch in Memory for the αRGB444 (16 Bits) Pixel Format

As shown in figure 26.5, the start address for each of planes A, B, and C is a 64-byte boundary. The readout area from each planes can be set to desired start address and area. However, the access unit is restricted by the byte number per pixel. For example, when the αRGB444 format has been set, access is in 2-byte units and only access to even addresses is allowed, access to odd addresses is prohibited.

(4) Input and Output Buffers

Since transfer to and from buffers SA, SB, DC, and SE in the 2DG must be handled by the DMAC under CPU control, these buffers are mapped on memory spaces (SRAM spaces) of the CPU. Table 26.7 shows the address map of the input and output buffers.

Table 26.7 Address Map of the Input and Output Buffers

Buffer Name	Abbreviation	Address
Input buffer E for the output block (276 Kbytes)	SE buffer	H'E8010000 to H'E8054FFC
Input buffer A for the blitter (276 Kbytes)	SA buffer	H'E8060000 to H'E80A4FFC
Input buffer B for the blitter (276 Kbytes)	SB buffer	H'E80B0000 to H'E80F4FFC
Output buffer C for the blitter (276 Kbytes)	DC buffer	H'E8100000 to H'E8144FFC

Since the 2DG has fixed-size input and output buffers, an image processing is performed with repeated DMA data transfer from these areas. Table 26.8 shows the specifications of the buffers. Each buffer is configured as the double buffer structure.

Table 26.8 Specifications of Input and Output Buffers

Buffer Name	Size
SE buffer	16 bits × 512 words × two planes
SA buffer	16 bits × 64 words × two planes
SB buffer	16 bits × 64 words × two planes
DC buffer	16 bits × 256 words × two planes

(5) Setting of Panel Output

Figure 26.6 shows the relations between the sync signals and the registers setting for display-panel output. These panel-output settings are made in the following registers.

- MGR_MIXHTMG register: WPH bits and PDPH bits
- MGR_MIXHS register: ALLPH bits and VLDPH bits
- MGR_MIXVTMG register: WPV bits and PDPV bits
- MGR_MIXVS register: ALLPV bits and VLDPV bits

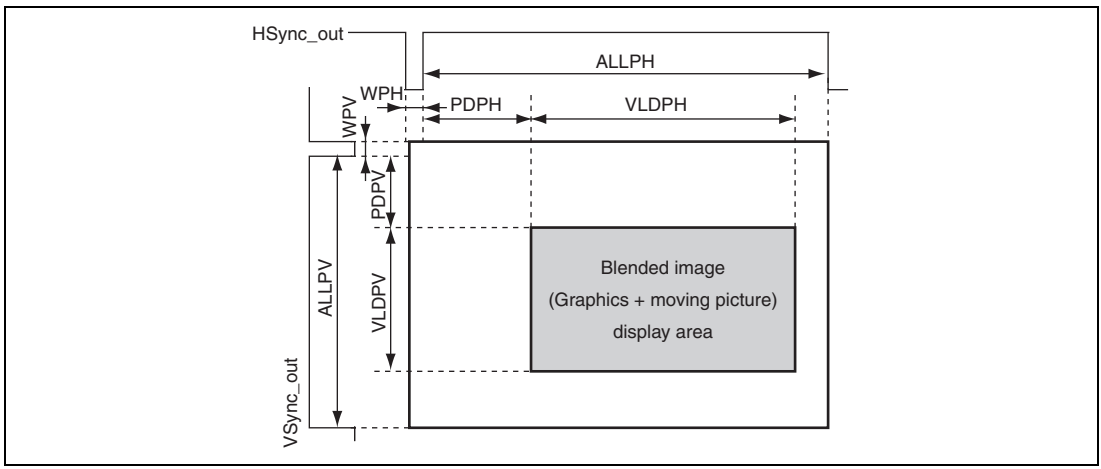


Figure 26.6 Relationship between Panel Output and Register Settings

(6) Relations between the Sync Signals for the Output Block and the Individual Clock Signals

(a) With an externally mounted video-decoder LSI

In cases where a moving-picture input is being supplied to the system, set the MVON bit in the MGR_MIXMODE register to 1. Then, the free-running HSYNC_dck (internal signal) is generated from DCLKIN and VSYNC_dck (internal signal) is created from the VIVSYNC signal. After that, the CSYNC signal is composed from the two signals, i.e. HSYNC_dck and VSYNC_dck.

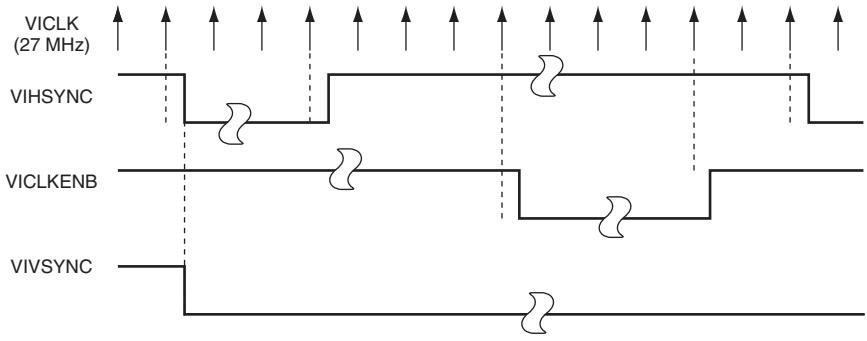
When the system includes an external video decoder LSI but this is not supplying the moving-picture input with a signal, set the MVON bit in MGR_MIXMODE register to 0. In this case, the free-running HSYNC_dck generated from DCLKIN is counted and the hardware itself produces the VSYNC_dck: the CSYNC signal is again composed from the HSYNC_dck and the VSYNC_dck.

(b) Without externally mounted video decoder LSI

In cases where there is no external video-decoder LSI to supply the system with a moving picture input, set the MVON bit in MGR_MIXMODE to 0. In this case, since only the free-running HSYNC_dck generated from DCLKIN is available, the hardware automatically produces the VSYNC_dck signal by counting cycles of HSYNC_dck, and then composes the CSYNC signal from HSYNC_dck and VSYNC_dck.

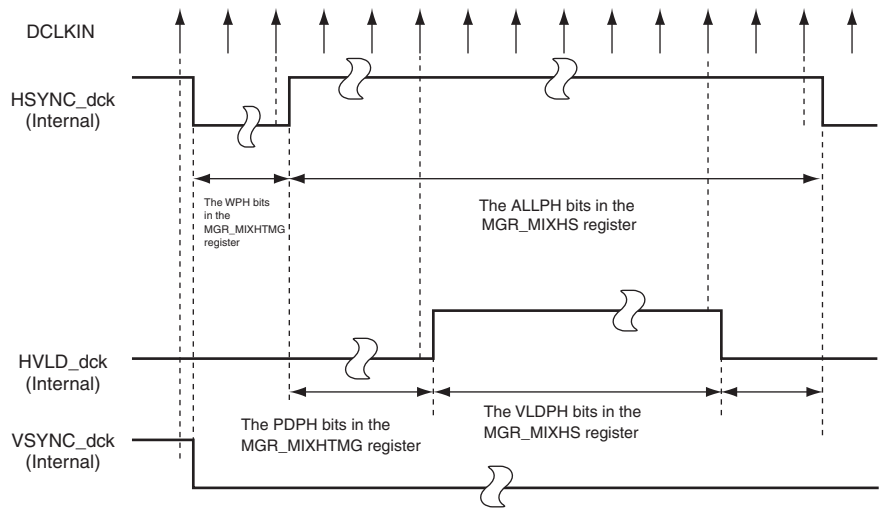
References:

- The image data from the output block, (the image data composed from the moving-picture and graphics data), and the CSYNC signal are output in synchronization with rising edges of the externally input DCLKIN signal.
- Timing with which the graphics data are read from the SE buffer is controlled by the HSYNC_dck and VSYNC_dck signals, and the MGR_SESET register (timing is not controlled by the VICLK system).
- For externally input moving pictures, the valid number of pixels horizontally is controlled by the VICLKENB signal and the valid number of lines is set by the VLDPV bits in the MGR_MIXVS register.
- Externally input moving pictures specified valid area are resized and then written to the data buffer. Reading of the data from the buffer is controlled by signals HSYNC_dck, and VSYNC_dck, and register MGR_MIXxx.



Each signal input from the video decoder is synchronous with the rising edge of VICLK (for both NTSC/PAL).

Figure 26.7 Relations between Externally Input Sync Signal and VICLK



Each internally generated sync signal is synchronous with the rising edge of DCLKIN (for both NTSC/PAL).

Figure 26.8 Relations between Internal Generated Sync Signals and DCLKIN

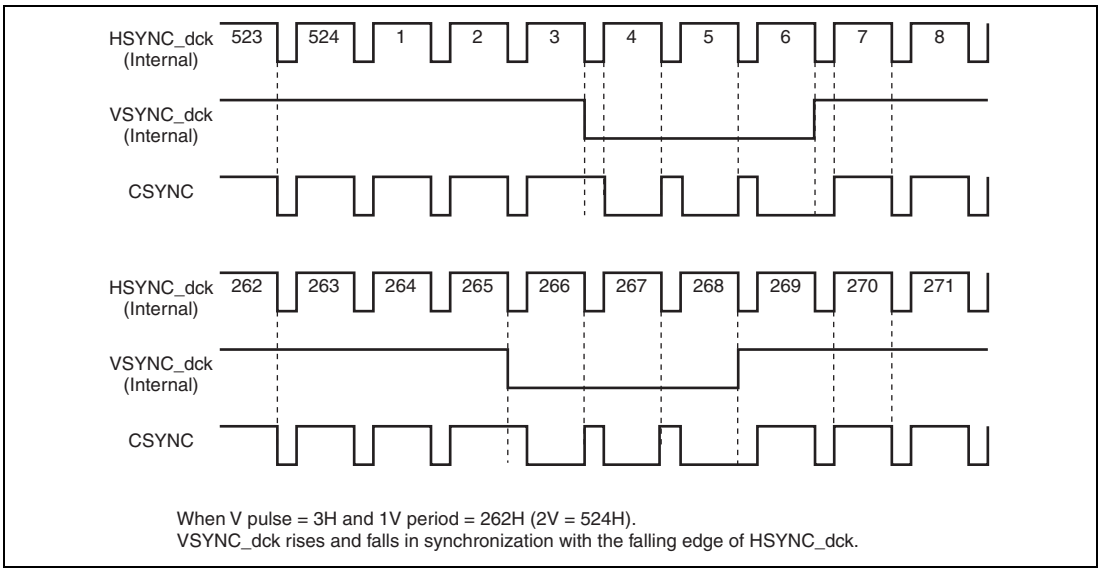


Figure 26.9 Timing of Internally Generated Sync Signals without VIVSYNC input (1) (NTSC System)

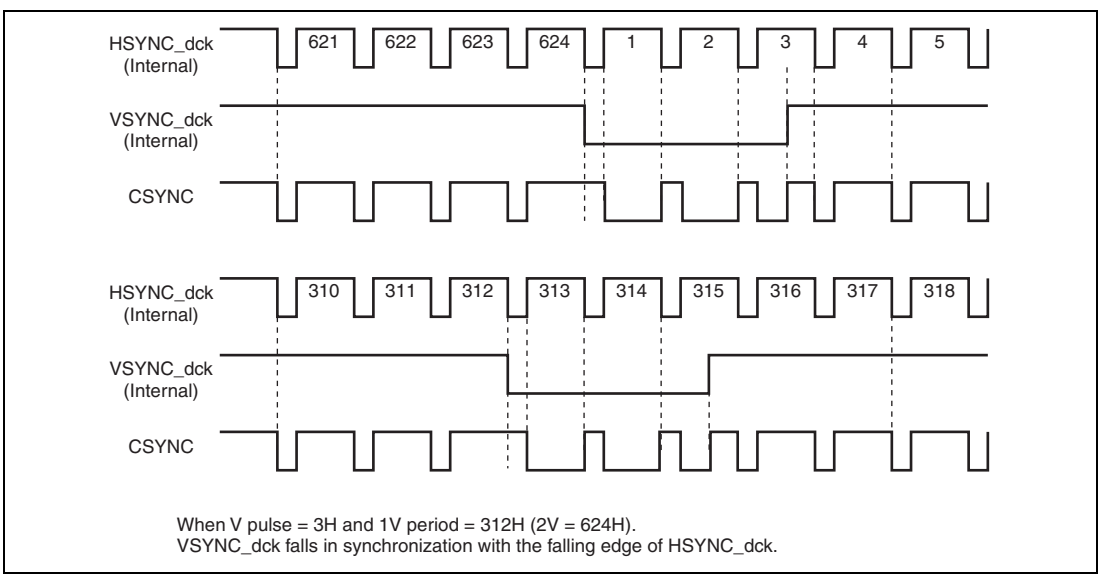


Figure 26.10 Timing of Internally Generated Sync Signals without VIVSYNC input (2) (PAL System)

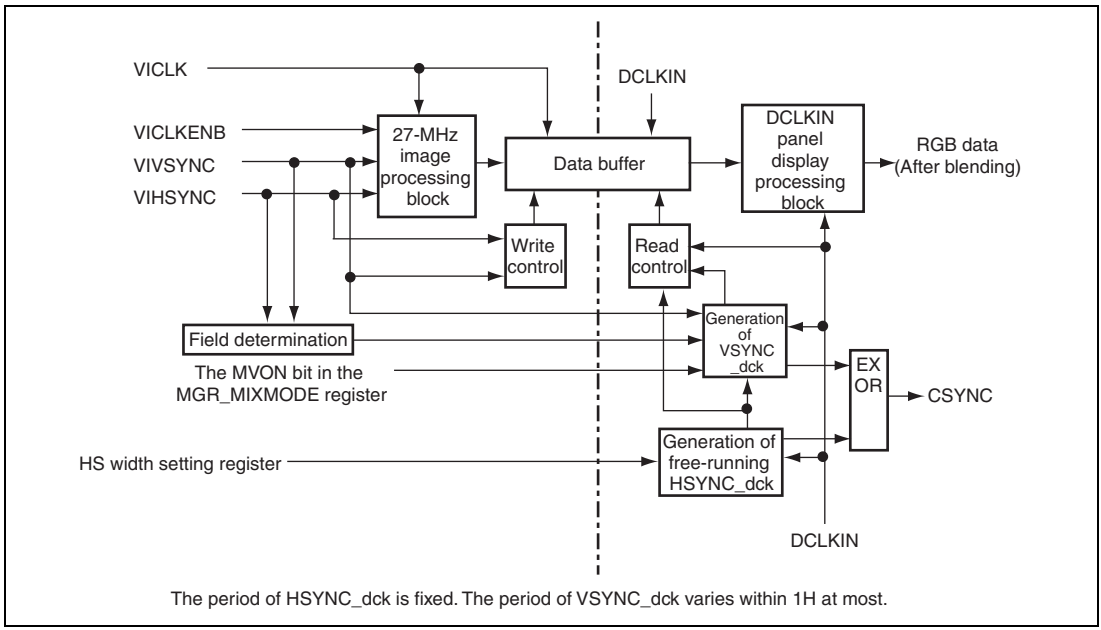
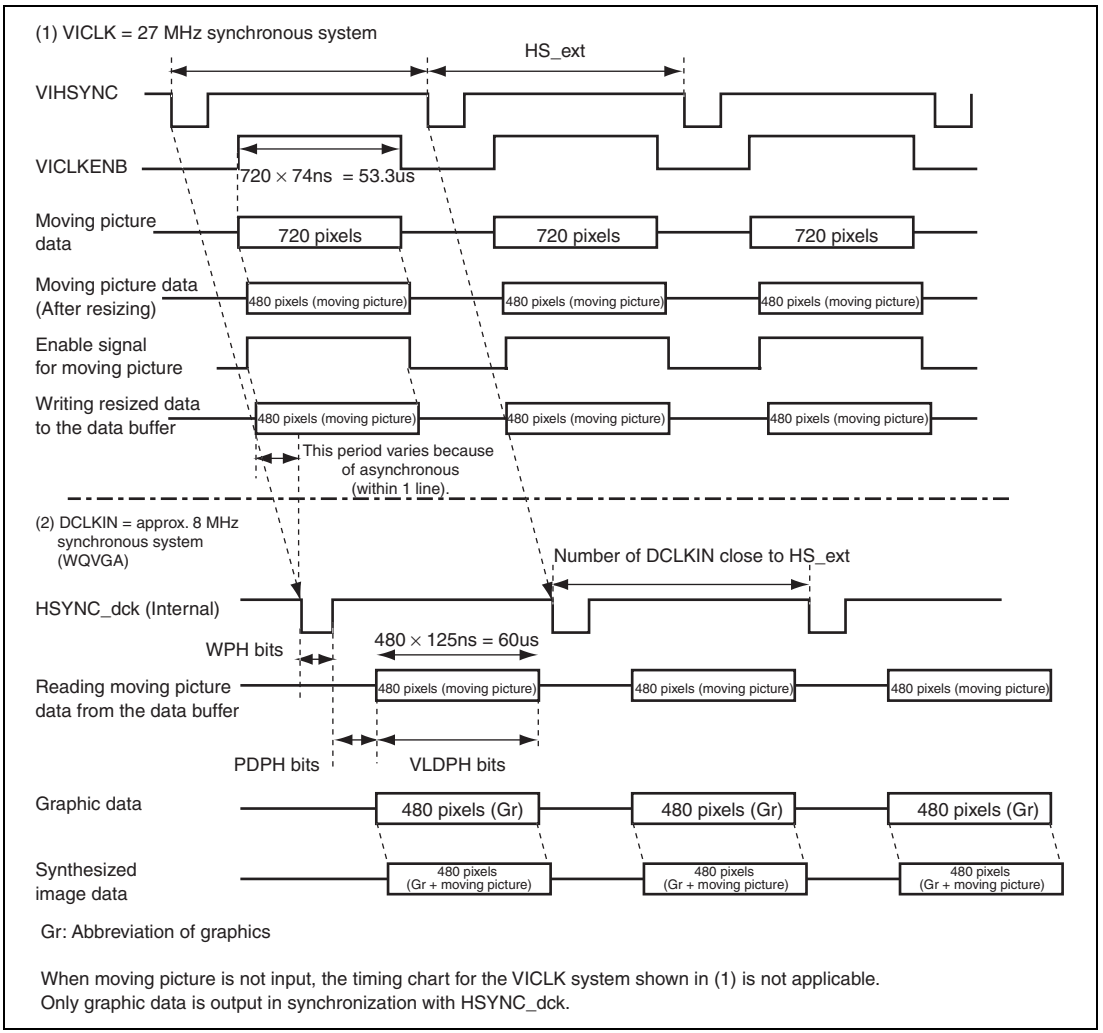


Figure 26.11 SYNC Signal Generating Unit (VICLK and DCLKIN Subsystems)

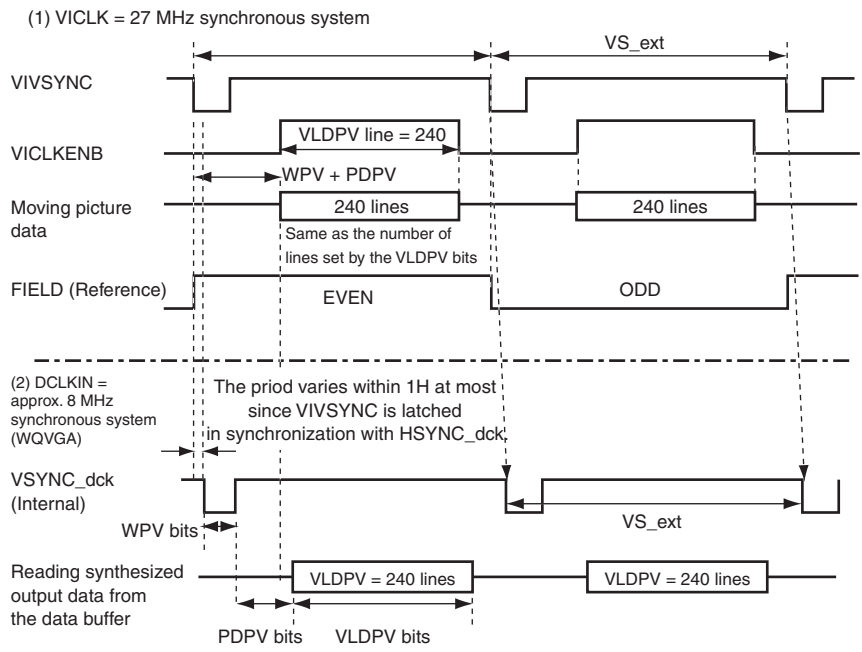
(7) Timing for Composition of Moving Pictures and Graphics by the Output Block



Gr: Abbreviation of graphics

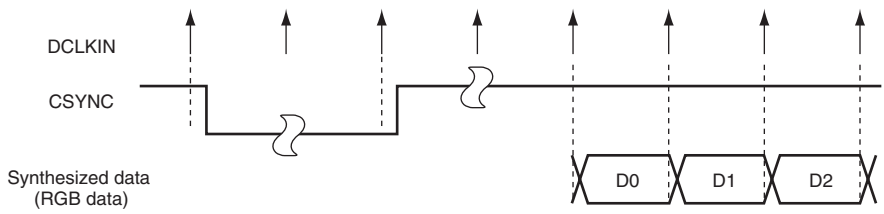
When moving picture is not input, the timing chart for the VICLK system shown in (1) is not applicable.
Only graphic data is output in synchronization with HSYNC_dck.

Figure 26.12 Timing in the Horizontal Direction (with Moving Pictures Supplied)



When moving picture is not input, the timing chart for the VICLK system shown in (1) is not applicable. Only graphic data is output in synchronization with VSYNC_dck set in the register.

Figure 26.13 Timing in the Vertical Direction (with Moving Pictures Supplied)



Synthesized data and the CSYNC signal are synchronous with the rising edge of DCLKIN (for both NTSC/PAL).

Figure 26.14 2DG Output Signals and DCLKIN

(8) RGB Conversion of External Input Moving Picture

The timing relations between the output data and clock signal from the external video decoder, i.e. the VICLK, VICLKENB, and VIDATA[7:0] are shown in figure 26.15.

Since the data is in YCbCr422 pixel format and the output data is to be composed with graphics (RGB) data, conversion is initially from YCbCr422 to YCbCr444, and then from YCbCr to RGB.

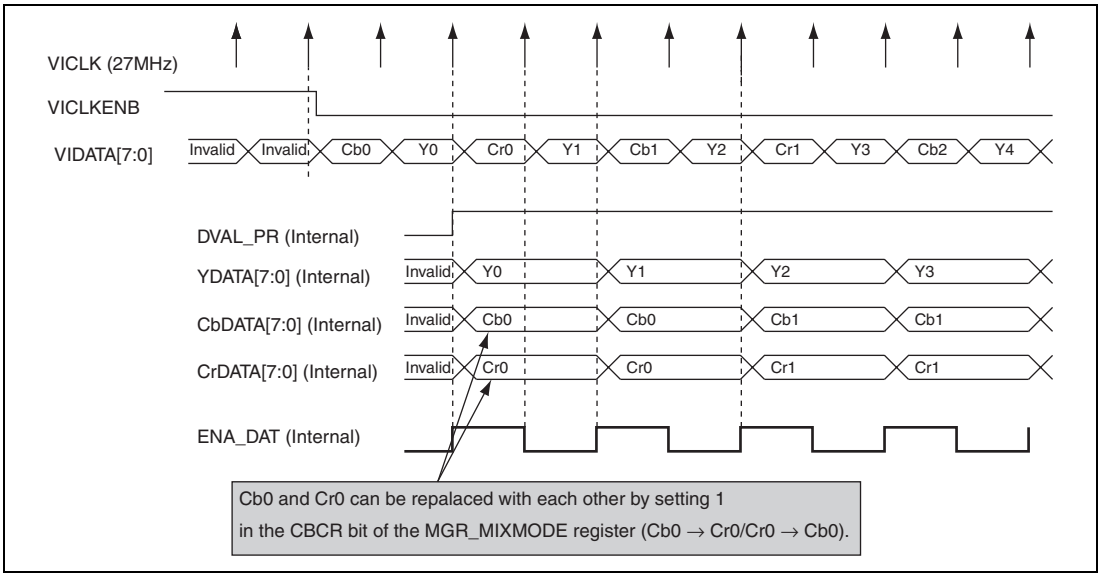


Figure 26.15 Timing of Conversion from YCbCr422 to YCbCr444

The formulae for converting from YCbCr to RGB are given below.

$$R = 1.164(Y-16) + 1.596\ (Cr - 128)$$

$$G = 1.164\ (Y-16) - 0.391\ (Cb - 128) - 0.813\ (Cr - 128)$$

$$B = 1.164\ (Y-16) + 2.018\ (Cb-128)$$

26.4.2 How to Use the DMA

- The 2DG DMAC has four channels. DMAC allocate the channels according to the priority level shown below:

  SE buffer DMA (2DG output) > DC buffer DMA (2DG BLT output C) > SB buffer DMA (2DG BLT input B) > SA buffer DMA (2DG BLT input A)

  Examples of allocation are as follows:

  Channel 1: SE buffer DMA

  Channel 2: DC buffer DMA

  Channel 3: SB buffer DMA

  Channel 4: SA buffer DMA
- When performing a data transfer using the normal CPU transfer instead of the DMAC, set the applicable DMx_MSEL bits of the GR_DMAC register to 11. Note that changing the transfer method to a CPU transfer during a DMA transfer is prohibited.
- For the SA/SB areas, the same DMA setting can be effected by setting the DM34_DSEL and DM34_MSEL bits of the GR_DMAC register.
- The number of bits (32 or 16) per data item is specified in terms of SZSEL1 (blitter)/SZSEL2 (output block). When the lower 2-bit addresses of the starting pixel for the DMA transfer are not "00", clear SZSEL1 and SZSEL2 to 0 (16 bits).
- When SZSEL1 = 32 bits, SSWIDH is always set to an even number; for SZSEL1 = 16 bits, SSWIDH can be either odd or even. Similarly, for SZSEL2 = 32 bits, SEWIDH is always even, and when SZSEL2 = 16 bits, SEWIDH can be either odd or even. Even when the data width of the SDRAM, which is external memory, is 16 bits, it is recommended to set the size to 32 bits when the efficiency of data transfer is important. Basically, use 32 bits if one transfer line = even pixels, and use 16 bits if one transfer line = odd pixels.
- Set the number of data items per operand so that the size of the buffer to be accessed is evenly divisible.

  For example, if data transfer size = 16 or 32 bits and one operand = 1, 2, 4, or 16 data items, full image transfer (480 pixels) can be made to all buffers. However, if data transfer size = 32 bits and one operand = 32 data items, a full pixel (480 pixels) transfer to the SE buffer results in $480/(32 \times 2) = 7.5$, the DMA transfer cannot be stopped at 240th word after performing operand transfers seven times (access up to $32 \times 7 = 224$ words), which means that transfers must be made up to the 256th word. In this case, if one operand = 16 data items, a full image transfer can be made by performing operand transfers 15 times.
- When performing DMA access to the 2DG, set the first address of the buffer as the access-start address. Starting DMA access beginning with a middle address is prohibited.

- In a DMA transfer to a buffer, if a continuous operand transfer is specified, the maximum transfer size per operation should be set equal to the buffer size to be accessed (for example, 128 bytes for an SB buffer, and 960 bytes for an SE buffer).

  If a continuous operand setting is selected, the transfer is continued until all transfer data that is specified is transferred; therefore, the DMA transfer cannot be suspended on account of SB, SA, DC, or SE buffer being full. For this reason, if a continuous operand transfer is to be executed to the SB or SA buffer, line-by-line transfers should be performed solely on the basis of one horizontal pixel setting = 32 or 64 pixels. In the case of a DC buffer, line-by-line transfers should be performed solely on the basis of one horizontal pixel setting = 32, 64, 128, or 256 pixels. In the case of an SE buffer, line-by-line transfers should be performed based on the number of pixels that is set on the one horizontal pixel setting.
- If a single-operand transfer is specified in a DMA-based data transfer to a buffer, and if another access equal to the buffer size is possible after completion of access equal to the buffer size, the hardware automatically continues issuing DMA requests in order to align buffer boundaries with operand boundaries. If another access equal to the buffer size is not possible, the DMAC is held in standby until a condition that permits the issuance of a DMA request is satisfied after the execution of internal processing.
- Examples of DMA transfer settings are given below:

  The number of pixels transferred to the 2DG is set as horizontal width 64 pixels/3 vertical lines, and each data item transferred consists of 1 pixel = 16 bits. In this case, the total number of pixels transferred will be 16 bits × 64 pixels × 3 lines = 384 bytes. If such data is transferred by a DMA transfer, the following two settings are possible:

  Approach 1: If one data item = 16 bits/operand = eight data items/transfer mode = pipeline (or cycle stealing)/transfer condition = single-operand transfer is set, and if the number of bytes transferred per DMA transfer is 384 bytes, for each buffer capacity the 2DG performs DMA request control to request a resumption of DMA transfer each time the buffer becomes empty and performs DMA transfers.

  Approach 2: If one data item = 16 bits/operand = eight data items/transfer mode = pipeline (or cycle stealing)/transfer condition = continuous-operand transfer is set, and if the number of bytes transferred per DMA transfer is 128 bytes and DMA is executed three times by reloading, because the buffer becomes full in each DMA transfer, for each buffer capacity the 2DG performs DMA request control to request a resumption of DMA transfer each time the buffer becomes empty and performs DMA transfers.

- The following are examples of CPU-side settings for using a DMA transfer:

  The number of pixels transferred to the 2DG is 52 pixels (horizontal width) × 20 vertical lines, and the size to be transferred is 32 bits.

  During normal DMA use: The amount of DMA transfer per operation is 52 pixels; if transfers are made by reloading 20 times, the OPSEL bits of the DMA mode register must be set to an integer multiple of 52 pixels. For this purpose, a transfer of two data items/operand must be set.

  During use of a two-dimensional DMAC: The following settings must be specified: number of blocks per line DBN = 1; number of lines per block, DRN = 20; number of column data items per block, DCDN = 26 data items; OPSEL = 2 data items/operand.

- The number of data items transferred per operand, which is a transfer parameter on the DMA side, can be any value; however, on some transfer areas the situation can arise where one line of data transfer cannot be finished within an HSYNC period (causing an underflow). If this problem occurs, it can be verified by checking the INT_UDRFL bit of the GR_IRSTAT register. If an underflow occurs, increase the number of data items transferred per operand on the DMA side.
- During a data write transfer to the SE, SB, or SA buffer or during a data read transfer from the DC buffer, in either case, DMA transfer or CPU transfer, internally the 2DG processes data by re-assigning internally-generated address. For this reason, even when the CPU side performs a data transfer from any address in the applicable memory space, the 2DG side performs access beginning with the first address in the memory space.
- The direction control bit of the DMAC during a destination transfer from the DMAC to the SE, SB, or SA buffer of the 2DG or during a source transfer from the DC buffer, targets on increment (memory-to-memory image) (memory-to-I/O transfers are prohibited).
- Operation during a CPU transfer

  If data access to the 2DG is to be made using the CPU transfer instead of a DMA transfer, the transfer must be performed by taking the maximum capacity of the applicable buffer into consideration (during a DMA transfer, the buffer size is controlled by internal hardware and, therefore, it need not be considered).

  For example, if an image consisting of 60 pixels × 4 lines (for a total of 240 pixels) is to be transferred to the SB buffer by means of CPU transfer,

1. Write access to the SB buffer is performed for 128 (= 64 × 2) pixels (generating an INT_SHFUL)
2. If the GR_DOSTAT register is read and the SB buffer is not in a two-banks-full condition, the data transfer is resumed in units of 64 pixels.
3. If an INT_SHFUL is not generated, the system continues to transfer the remaining 48 pixels, and completes the data transfer process.

In the case of the DC buffer, there will be 256 pixels per bank (or 512 pixels for the two banks). Since access to the SE buffer can adversely affect display, access by means of a CPU transfer is not recommended.

Figure 26.16 shows an example of data transfer to the SB buffer using the DMAC.

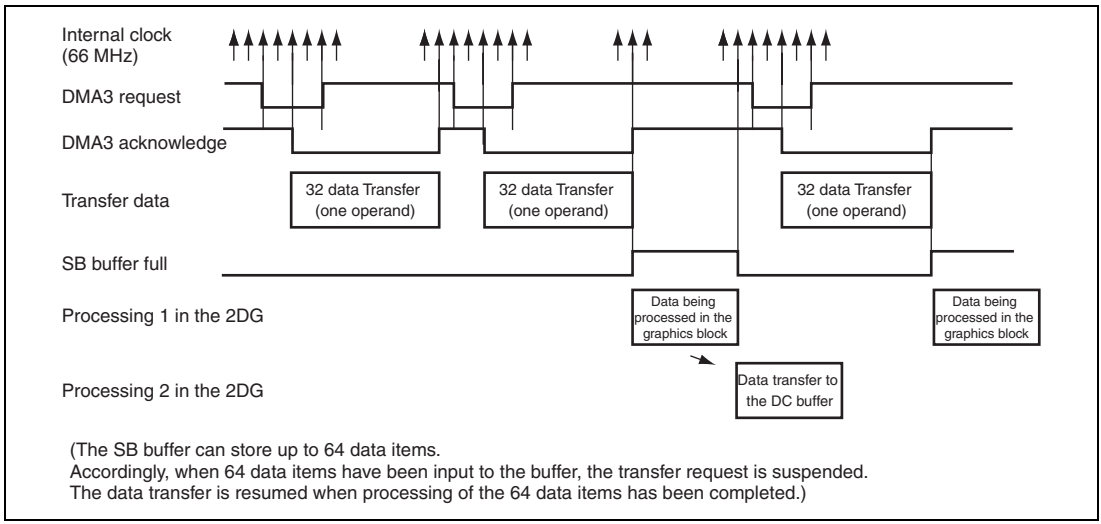


Figure 26.16 96-item Data Transfer Timing in Dual/Single Address Transfer Mode (Single-Operand Transfer)

26.4.3 Blitter Operation

(1) Pixel Format Conversion in Blitter

On the blitter, as I/O pixel formats three formats can be set: αRGB444, αRGB555, and α (4 bits). For this reason, internally in the blitter, various operations are performed after converting a given format into a standard format. The size of the standard format is α (4 bits) + RGB (5 bits each) for a total of 19 bits. The formula below shows rules for the conversion of a given format to the standard format:

- αRGB444 (AF83(H)) converted into a standard format
  α: A(H) → A(H) R:F(H) → 1E(H) G:8(H) → 10(H) B:3(H) → 06(H)

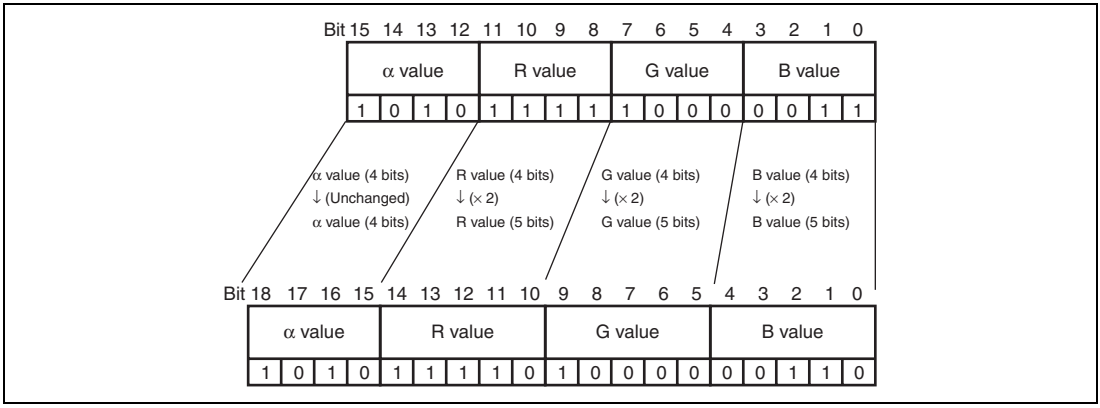


Figure 26.17 Pixel Format Conversion 1 in Blitter

- αRGB555 (F599(H)) converted into a standard format
  α:1(H) → F(H) R:1D(H) → 1D(H) G:0C(H) → 0C(H) B:19(H) → 19(H)

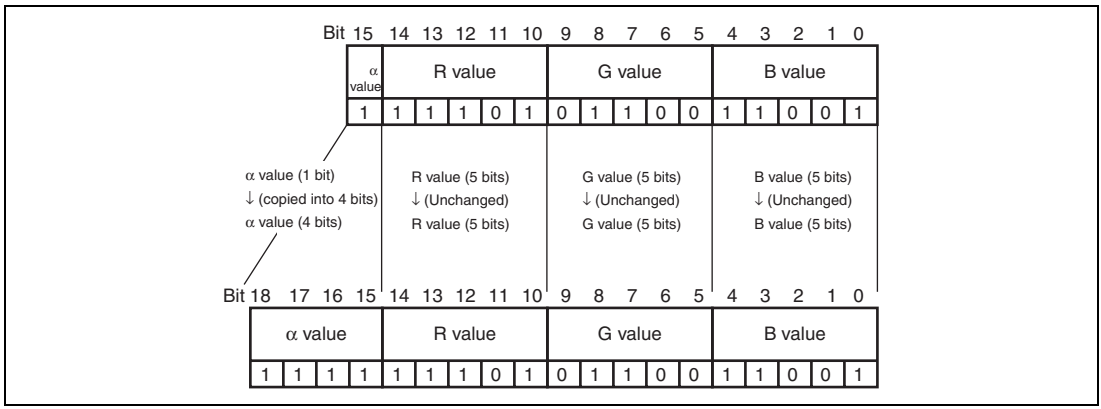


Figure 26.18 Pixel Format Conversion 2 in Blitter

- α (4 bits) (B(H)) converted into a standard format
  α:B(H) → B(H) R:Csasg_r G:Csasg_g B:Csasg_b

The native value is assigned to the α value. To the RGB value, the BRDC_R, BRDC_G, and BRDC_B bits of the GR_BRDCOL register are assigned.

(2) Summary of Operations between the Blitter and External Memory

The following is a summary of operations between the blitter and external memory:

1. The blitter negates the DMA request signal and accepts a DMA transfer from the external memory.
2. The SA/SB buffers ((SA1, SA2), (SB1, SB2)), alternately buffer the data received through the DMA transfer.
3. When the INT_SHFUL and INT_ASHFUL are asserted (if there is source buffer half agreement), blitter operations are started.
4. Upon completion of blitter operations, a DMA transfer from the DC buffer is performed.
5. The above steps 1 to 4 are repeated until all data processing is completed.

As described above, processing such as image synthesis is possible by reading the external memory area first, performing blitter processing on the area that has been read next, and writing back to the same memory area last.

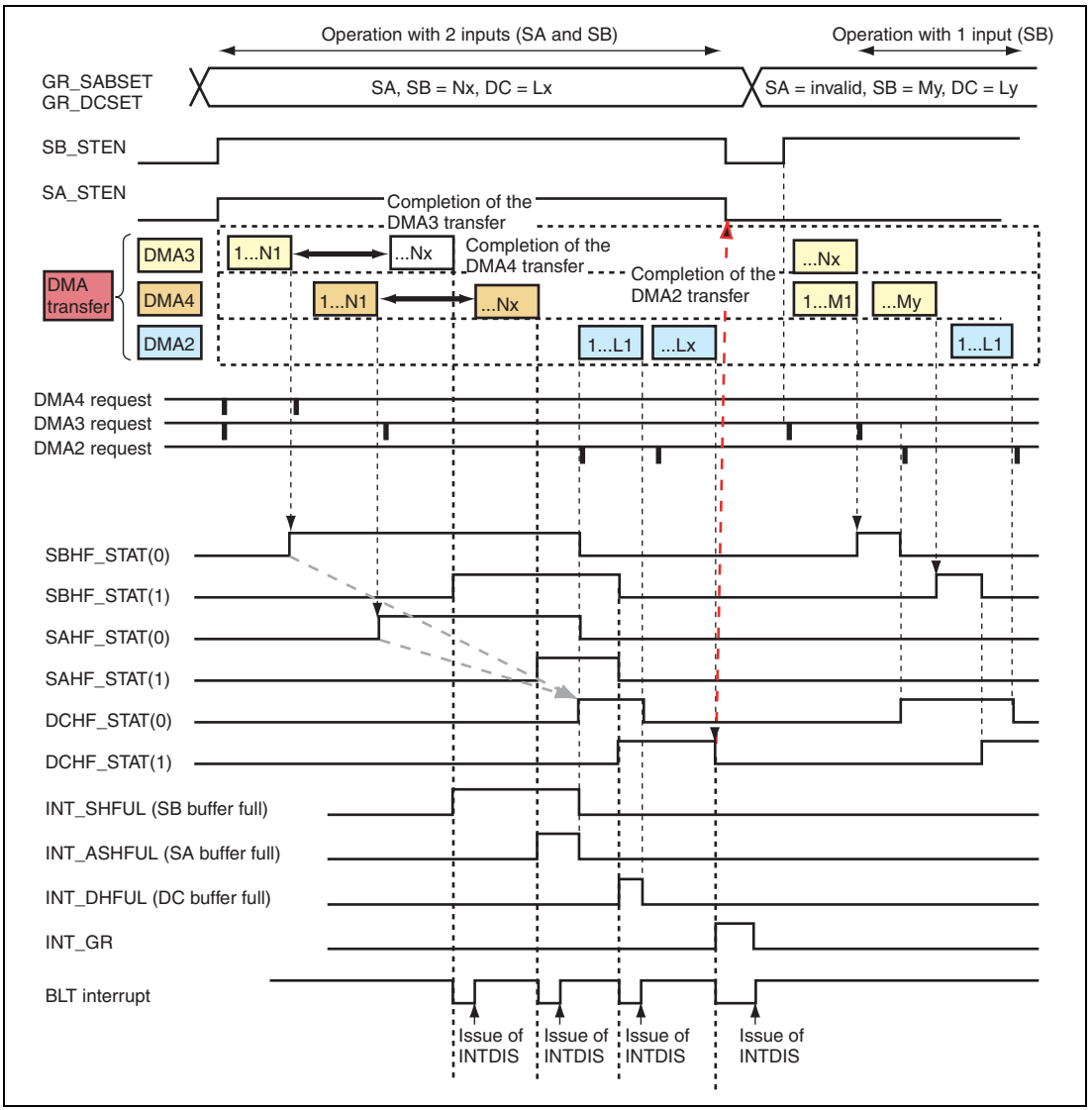


Figure 26.19 Summary of Operations between Blitter and External Memory

(3) Summary of Blitter Operations

The text below provides a summary of blitter operations. Table 26.9 shows allowable combinations of operations. Chromakey processing, color gradation processing, and logical operations cannot be performed simultaneously. Chromakey processing is allowed only during a fill operation. Color gradation processing can be performed only during blitting. Logical operations, blending, and resizing can be performed irrespective of blit/filling (subject to some restrictions).

Table 26.9 Allowable Combinations of Blitter Operations

Item No.	Operation (the BTYPE bits of the GR_BLTMODE register)	SB Path Operations (the SBSEL bits of the GR_BLTMODE register)			Blending (the FBFA bits of the GR_BRD1CNT register)	Reference Sections
		Chromakey Processing	Color Gradation Processing	Logical Operations		
1	Blitting	×	×	×	×	26.4.3 (3) (c)
2		×	×	×	O	
3		×	×	O	×	–
4		×	×	O	O	–
5		×	O	×	×	26.4.3 (3) (f)
6*[1]		×	O	×	O*[2]	
7	Filling	×	×	×	×	–
8		×	×	O	×	–
9		×	×	×	O	26.4.3 (3) (b)
10		×	×	O	O	26.4.3 (3) (e)
11		O	×	×	×	26.4.3 (3) (d)
12		O	×	×	O	

Notes: Resizing function: During a blitting operation, resizing can be turned on/off on each items; it cannot be used during a filling operation.

1. The resizing function can be applied only to full resizing; it cannot be used for partial resizing.
2. Only SA-input data is eligible for blending; blending with register-stored data cannot be performed.

As a summary of blitter operations, the text below provides an example where target planes P1 and P2 are blended for full-plane synthesis, and the results are written back to an arbitrary memory space PX on the SDRAM.

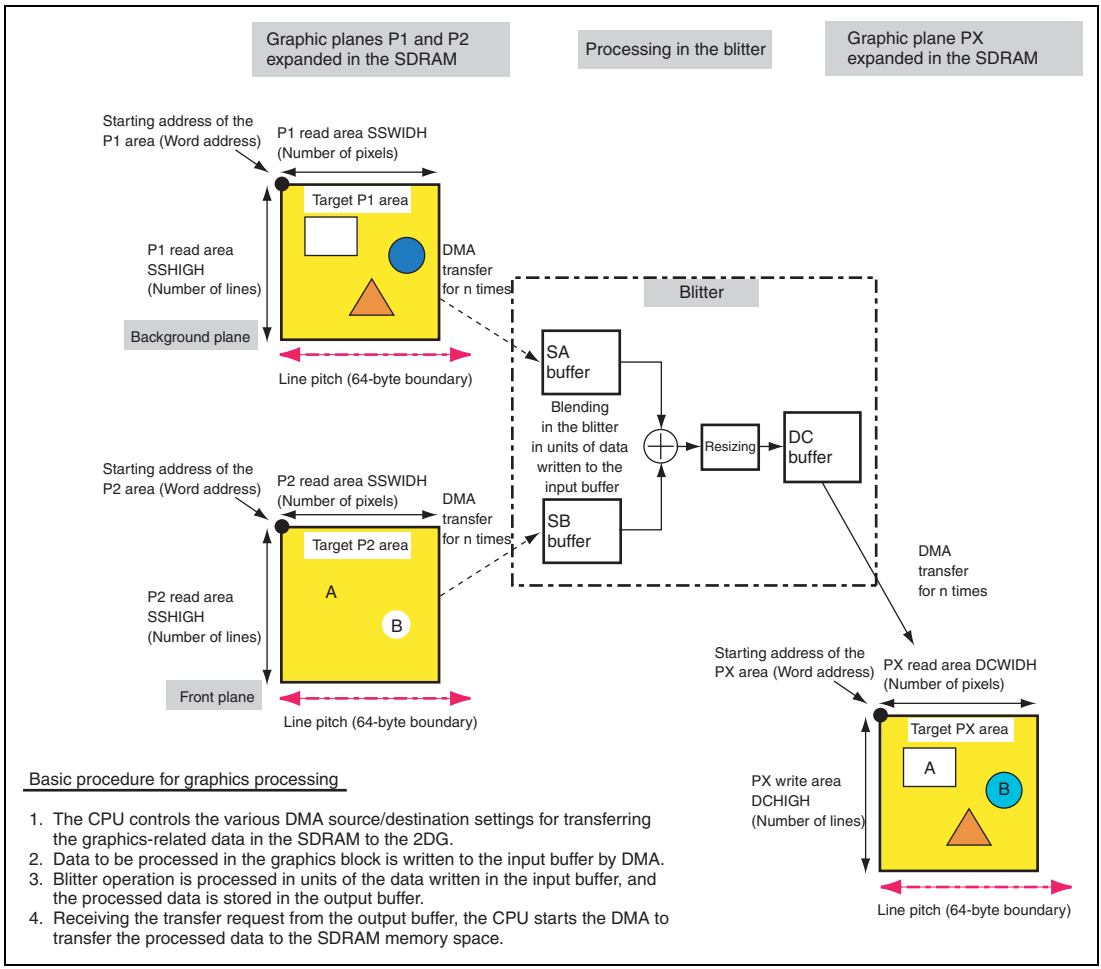
The areas for planes P1 and P2 are set as follows: the number of lines in the SSHIGH bits of the GR_SABSET register, and the number of pixels in the SSWIDH bits of the GR_SABSET register. For the PX, the number of lines is set in the DCHIGH bits of the GR_DCSET register, and the number of pixels in the DCWIDH bits of the GR_DCSET register.

The SA and SB buffers each have a 128-byte double-buffer structure ((SA1, SA2), (SB1, SB2)). For example, if one input is selected (i.e., SB_STEN bit = 1 and SA_STEN bit = 0 in the GR_BLTPLY register), and if the following values are assigned: SSWIDH bits = 40 (pixels) in the GR_SABSET register and SSHIGH bits = 4 (lines) in the GR_SABSET register, blitter operations work as follows:

1. Transfers the first 64 pixels to SB1 (SBHF_STAT (0) = 1), followed by blitter processing and output.
2. Transfers the next 64 pixels to SB2 (SBHF_STAT (1) = 1), followed by blitter processing and output.
3. Transfers the remaining 32 pixels to SB1 (SBHF_STAT (0) = 1), followed by blitter processing and output.
4. This completes the blitter operation (SB_STEN=0).

The DC buffer has a 256-byte double-buffer structure (DC1, DC2). For example, if DCWIDH bits = 60 (pixels) in the GR_DCSET register and DCHIGH bits = 5 (lines) in the GR_DCSET register, blitter operations work as follows:

1. Transfers the first 128 pixels to DC1 (DCHF_STAT (0) = 1), followed by a DMA transfer.
2. Transfers the next 128 pixels to DC2 (DCHF_STAT (1) = 1), followed by a DMA transfer.
3. Transfers the remaining 44 pixels to DC1 (DCHF_STAT (0) = 1), followed by a DMA transfer.



Basic procedure for graphics processing

1. The CPU controls the various DMA source/destination settings for transferring the graphics-related data in the SDRAM to the 2DG.
2. Data to be processed in the graphics block is written to the input buffer by DMA.
3. Blitter operation is processed in units of the data written in the input buffer, and the processed data is stored in the output buffer.
4. Receiving the transfer request from the output buffer, the CPU starts the DMA to transfer the processed data to the SDRAM memory space.

Figure 26.20 Summary of Blitter Operations

(a) Blending

The blending for the synthesis using two planes is performed according to a general formula, based on the following algorithm irrespective of the order in which the color value Cb and α value α b of the background plane and the color value Cf and α value α f of the front plane: (C(i): an offset value if Cpout: color value after blending, and Apout: α value after blending):

$$Cpout = Mf * (Cf - C(i)) + (1 - Af) * Mb * (Cb - C(i)) + C(i)$$

$$Apout = 1 - (1 - Af) * (1 - Ab)$$

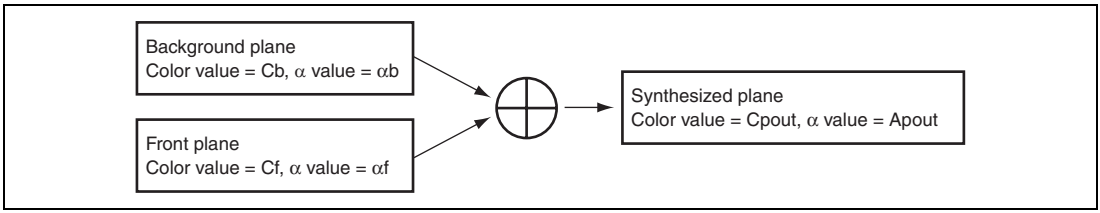


Figure 26.21 Blending for Synthesis using Two Planes

provided

if the front plane is premultiplied, Mf = 1, Cf = Cpf
if the front plane is non-premultiplied, Mf = Af, Cf = Cf
if the background plane is premultiplied, Mb = 1, Cb = Cpb
if the background plane is non-premultiplied, Mb = Ab, Cb = Cb

The diagram below shows the relationship between graphic synthesis planes:

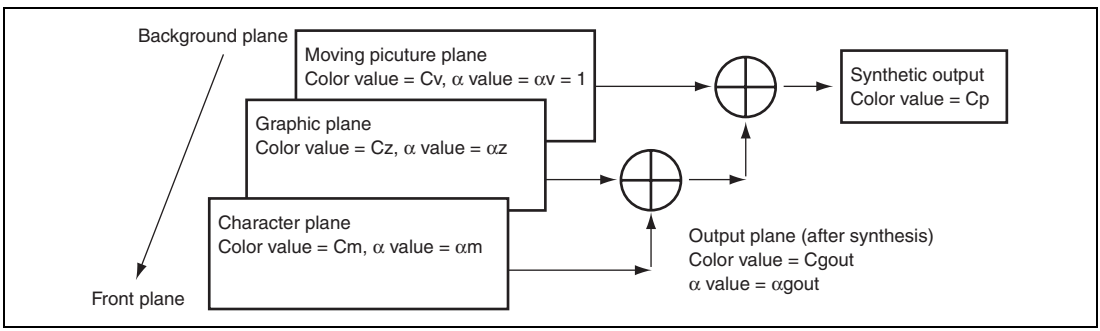


Figure 26.22 Relationship between Graphic Synthesis Planes

For the output plane, since its source character and graphics planes are both non-premultiplied, it follows that Mf = αm, Cf = Cm, Mb = αz, and Cb = Cz. Given αRGB444 as the image format, it follows that C(i) = 0. Therefore, by applying the algorithms for Cpout and Apout to the output plane, we obtain:

$$Cgout = \alpha m * Cm + (1 - \alpha m) * \alpha z * Cz$$

$$\alpha gout = 1 - (1 - \alpha m) * (1 - \alpha z)$$

Also, for the synthetic output, which is the final output, given that the source output image plane is premultiplied and the moving picture plane is non-premultiplied, it follows that Mb = αv = 1. Therefore, by applying the algorithm for Cpout to the synthetic output, we obtain:

$$Cp = 1 * Cgout + (1 - \alpha gout) * 1 * Cv$$

The following is an example where priority is given to character display (the case where the α value assigned to the character = 1). In this case, since αm = 1, the blending of the color value of the character part and the α value will result in the following:

$$Cgout = Cm, \alpha gout = 1$$

In the final output, the character part will be:

$$Cp = Cm \text{ (no moving pictures blended)}$$

Consequently, only characters are displayed in the results of output synthesis.

If 1 is assigned as α value (αz = 1) of the graphics plane to give priority to graphics plane display, since the front most plane is a character plane, the results of the blending will be as follows, consisting of a blending between characters and graphics, unless the α value of the character plane = 0 (αm = 0):

$$Cgout = \alpha mCm + (1 - \alpha m)\, Cz, \alpha gout = 1$$

The final output will be:

$$Cp = \alpha mCm + (1 - \alpha m)\, Cz \text{ (no moving pictures blended)}$$

The blending processor in the blitter, which performs pseudo-anti-aliasing with characters, blends characters having an arbitrary α value with an α value = 1 graphic in a rectangular region, and it can transfer the results to the output plane by converting the blending results to an arbitrary α value (register value) for blending with the moving pictures. In this case, the blending results between characters and graphics will be:

$$Cgout = \alpha mCm + (1 - \alpha m)\, Cz,\ \alpha gout = 1$$

Further, if global α (the BRDC_A bit of the GR_BRDCOL register) is selected using the subsequent stage global α selection bit (the GALFA bit of the GR_BRD1CNT register), the results that are output from the blending processor in the blitter will be:

$$Cgout = \alpha mCm + (1 - \alpha m)\, Cz,\ \alpha gout = \alpha dc = \text{BRDC\_A bit of GR\_BRDCOL register}$$

Thus, pixels having αdc values are created in the output plane.

In terms of the 2DG, if synthesis using two planes is performed by blending in the blitter and all color values of the output plane are α-value-computed (i.e., weighted) data, the blending between the moving picture and the output plane in the output block will be performed by selecting premultiplied in the blending processor in the output block. On the other hand, if the color values on the output plane are not α-weighted, the blending between the moving picture and the output plane in the output block will be performed by selecting non-premultiplied in the blending processor in the output block.

Figure 26.23 shows a summary of operations during blending, as follows:

(1) If GCOLR bit in GR_BRD1CNT register = 0, selects data from the SA buffer.
If GCOLR bit in GR_BRD1CNT register = 1, selects the value of the GR_BRDCOL register.
In fill mode, the value in the GR_BRDCOL register is selected in the same way as if the GCOLR bit of the GR_BRD1CNT register was set (1). However, the value of the GCOLR bit does not change at this time.

(2) Blends the data selected in Step 1 above with the data from the SB buffer selected on the basis of the SBSEL bit of the GR_BLTMODE register.

(3) Of the data blended in Step 2 above, the color value Cout is output to the DC buffer as is. With regard to the α value α out, the blended α value is output to the DC buffer as is, provided the GALFA bit of the GR_BRD1CNT register = 0. If the GALFA bit of the GR_BRD1CNT register = 1, the AFTER_A bit value of the GR_BRD1CNT register is output to the DC buffer.

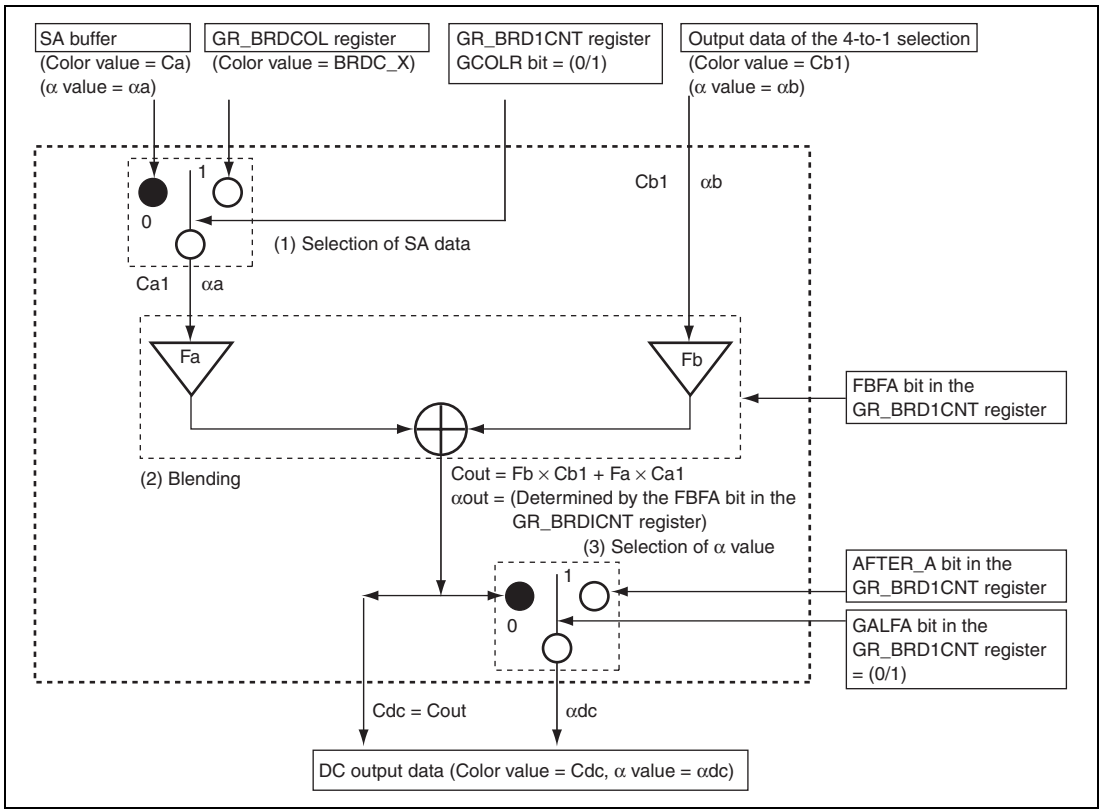


Figure 26.23 Summary of Blitter Operations during Blending

(b) Summary of Fill Function

This section shows examples of using the fill function. Set the registers as follows:

- Set the GR_BRDCOL register to specify a fill color.
- Set the GR_SABSET register to specify the number of pixels to be transferred to the SB buffer.
- To specify the number of pixels to be transferred to the DC buffer, set the GR_DCSET register.
- Set the BTYPE bits of the GR_BLTMODE register to 10 in order to specify filling operation for blitter operating mode.
- To specify the blitter function, set the SB_SETEN bit of the GR_BLTPLY register to 1, and the SA_SETEN bit to 0.

With the above settings, the fill function operates as follows:

1. By the GR_BLTPLY register settings, image data equal to the pixel count set in the GR_SABSET register is transferred to the SB buffer.
2. The pixel data transferred to the SB buffer is blended with the pixel data set in the GR_BRDCOL register.
3. Image data equal to the number of pixels set in the GR_DCSET register is output to the DC buffer.

The DMAC first transfers, memory-to-memory, pixels equal to the pixel count set in the register from an SDRAM area specified by the CPU to the SB buffer. After that, the DMAC transfers, memory-to-memory, the pixel data processed in the 2DG from the DC buffer to the SDRAM area specified by the CPU (the original image area whose contents were transferred to the SB buffer). As a result, the specified area on the SDRAM can be replaced with blended pixel data. In this case, the 2DG performs input to the SB buffer and output from the DC buffer.

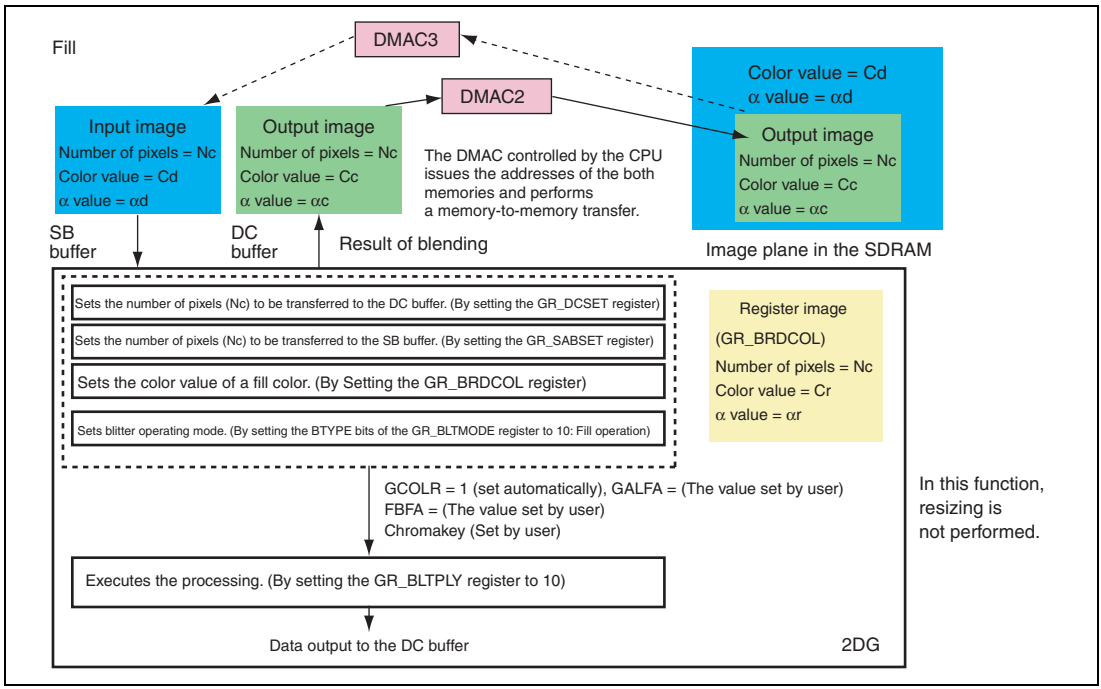


Figure 26.24 Example of Fill Function

(c) Summary of Blit Function

- Blit function without blending

This section shows an example of using the blit function without blending. Set the registers as follows:

- Set the GR_SABSET register to specify the number of pixels to be transferred to the SB buffer.
- Set the GR_DCSET register to specify the number of pixels to be transferred to the DC buffer.
- Set the BTYPE bits of the GR_BLTMODE register to 00 in order to specify blit operation for blitter operating mode.
- Set the FBFA bits of the GR_BRD1CNT register to 01 in order to specify 1-input processing.
- To enable the blit function, set the SB_SETEN bit of the GR_BLTPLY register to 1, and the SA_SETEN bit to 0.

With the above settings, the blit function operates as follows:

1. By the GR_BLTPLY register settings, image data equal to the pixel count set in the GR_SABSET register is transferred to the SB buffer.
2. The pixel data transferred to the SB buffer undergoes various processings set by the relevant registers.
3. Image data equal to the number of pixels set in the GR_DCSET register is output to the DC buffer.

The DMAC first transfers, memory-to-memory, pixels equal to the pixel count set in the register from an SDRAM area specified by the CPU to the SB buffer. After that, the DMAC transfers, memory-to-memory, the pixel data processed in the 2DG from the DC buffer to the SDRAM area specified by the CPU. As a result, the image on any area on the SDRAM can be replaced with pixel data having undergone various image processings by the 2DG. In this case, the 2DG performs input to the SB buffer and output from the DC buffer.

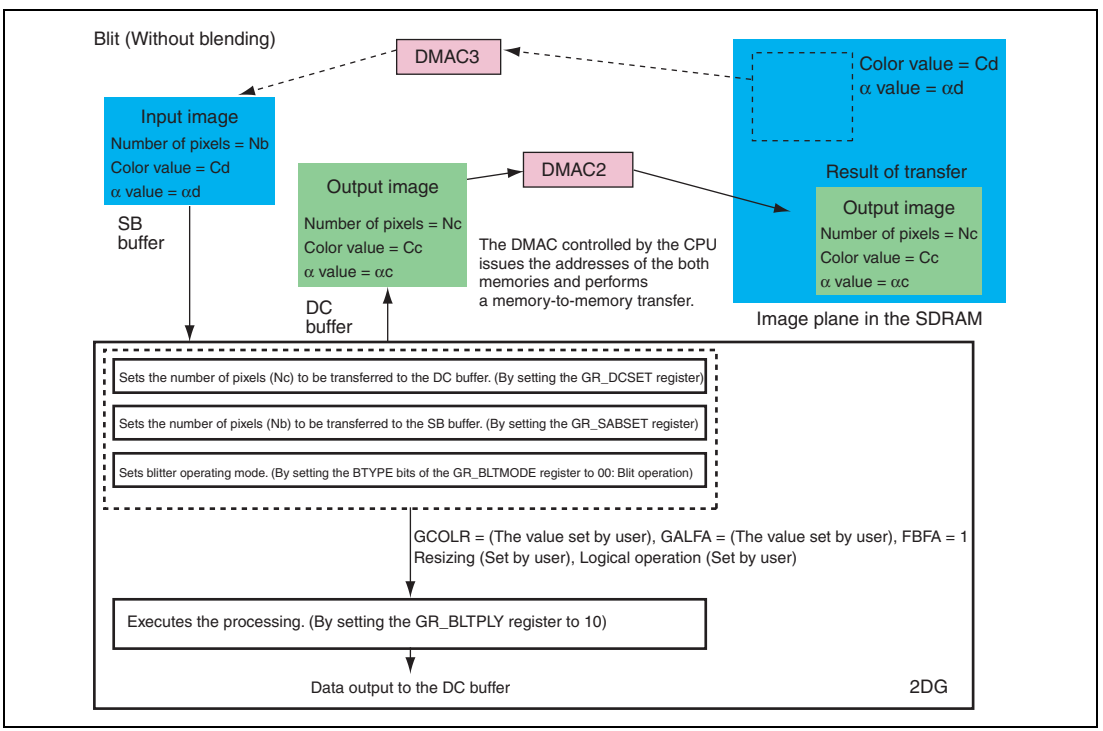


Figure 26.25 Example of Blit Function without Blending

- Blit function with blending (1)

This section shows an example of blending graphic images. Set the registers as follows:

- Set the GR_SABSET register to specify the number of pixels to be transferred to the SA/SB buffers.
- Set the GR_DCSET register to specify the number of pixels to be transferred to the DC buffer.
- Set the BTYPE bits of the GR_BLTMODE register to 00 in order to specify blit operation for blitter operating mode.
- To enable the blit function, set the SB_SETEN and SA_SETEN bits of the GR_BLTPLY register to 1.

With the above settings, the blit function operates as follows:

1. By the GR_BLTPLY register settings, image data equal to the pixel count set in the GR_SABSET register is transferred to the SA/SB buffers.
2. The pixel data transferred to the SA/SB buffers undergoes various processings set by the relevant registers.
3. Image data equal to the number of pixels set in the GR_DCSET register is output to the DC buffer.

The DMAC first transfers, memory-to-memory, pixels equal to the pixel count set in the register from an SDRAM area specified by the CPU to the SA/SB buffers. After that, the DMAC transfers, memory-to-memory, the pixel data processed in the 2DG from the DC buffer to the SDRAM area specified by the CPU. As a result, the image on any area on the SDRAM can be replaced with pixel data having undergone various image processings by the 2DG. In this case, the 2DG performs input to the SA/SB buffers and output from the DC buffer.

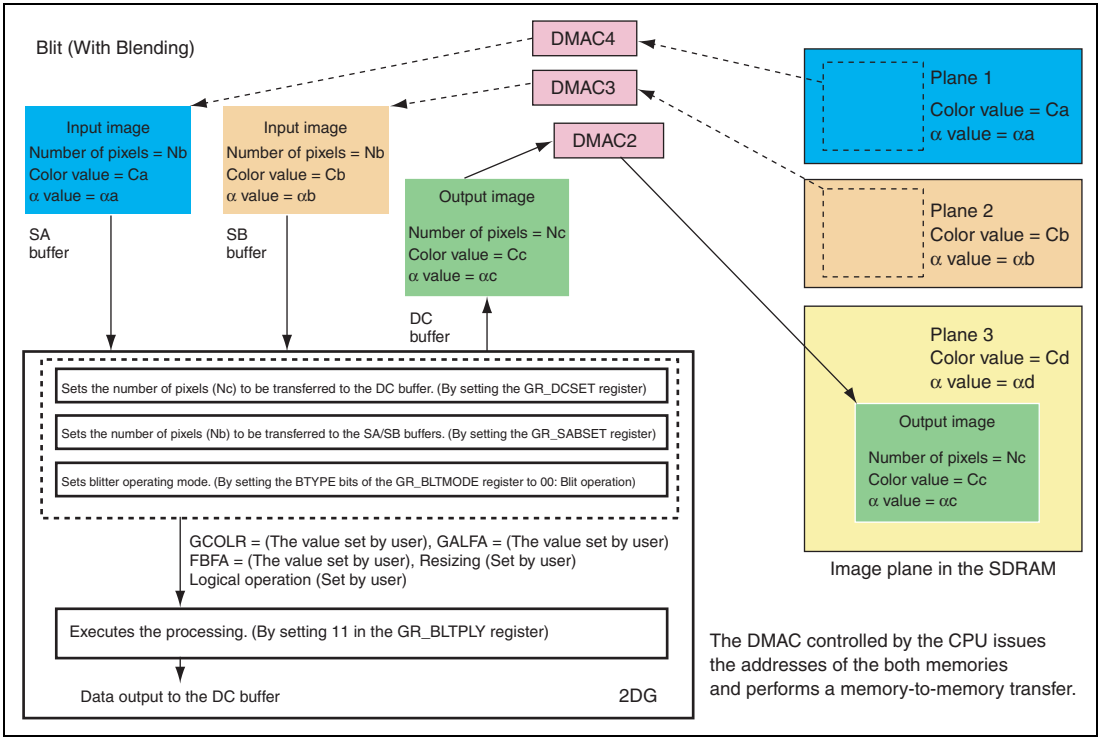


Figure 26.26 Example of Blit Function with Blending (1)

- Blit function with blending (2)

This section shows an example of blending graphic and character images. Set the registers as follows:

- Set the GR_SABSET register to specify the number of pixels to be transferred to the SA/SB buffers.
- Set the GR_DCSET register to specify the number of pixels to be transferred to the DC buffer.
- Set the BTYPE bits of the GR_BLTMODE register to 00 in order to specify blit operation for blitter operating mode.
- To enable the blit function, set the SB_SETEN and SA_SETEN bits to 1.

In this case, if the α – value-weighted character pixels set in a rectangular area are input into the SB buffer and graphic pixels (α value = F(H)=1) set in the same rectangular area as SB are input into the SA buffer, the output image will be an α-blending of the graphics and characters in the rectangular area followed by resizing. In this case, in order to place character information in the foremost part of the image, it should be input into the SB buffer.

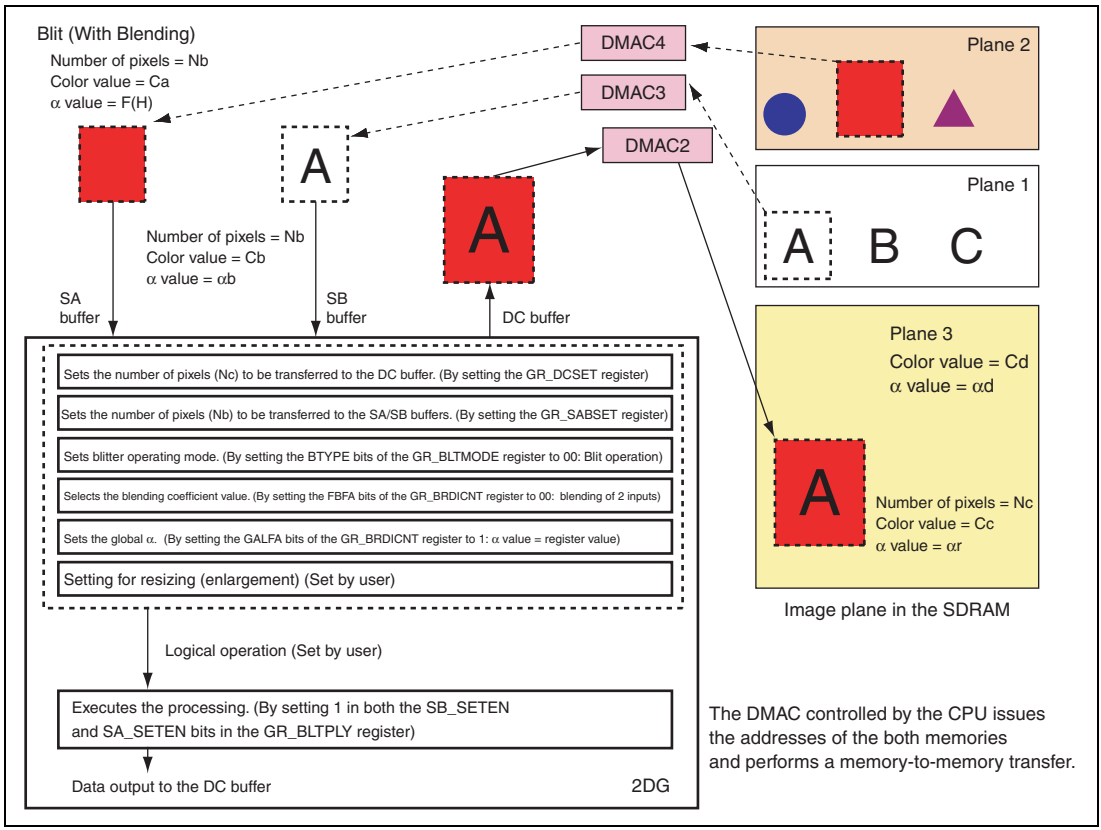


Figure 26.27 Example of Blit Function with Blending (2)

(d) Summary of Chromakey Processing

Since the chromakey processing function is installed only on the input B path, chromakey processing cannot be applied to the input A path. Furthermore, neither logical operations nor color gradation processing can be executed at the same time as chromakey processing. Part of the chromakey specification is that chromakey settings are only effective when fill operations are running. For this reason, select fill mode by setting the BTYPE bits of the GR_BLTMODE register to 10 if chromakey processing is to be used. Additionally, since chromakey processing only operates with fill mode, it is not available in conjunction with the resizing function.

Examples of the effects of executing chromakey processing are as follows:

- Chromakey processing example (1)

  Blit mode setting: Fill mode (GR_BLTMODE register, BTYPE bits = 10)

  Chromakey type selection: Chromakey replacement (GR_BLTMODE register, CRKEY bits = 01) Operation is the same as when the FBFA bit in the GR_BRD1CNT register (blending coefficient selection) had been set to 01. However, the value of the FBFA bits in the GR_BRD1CNT register does not change.

  SB output data selection: SB data selection = (GR_BLTMODE register, SBSEL bits = 01)

  Target color setting: Setting for green as the target color in the GR_DETCOL register

  Replacement color setting: Setting for blue as the replacement color in the GR_BRDCOL register

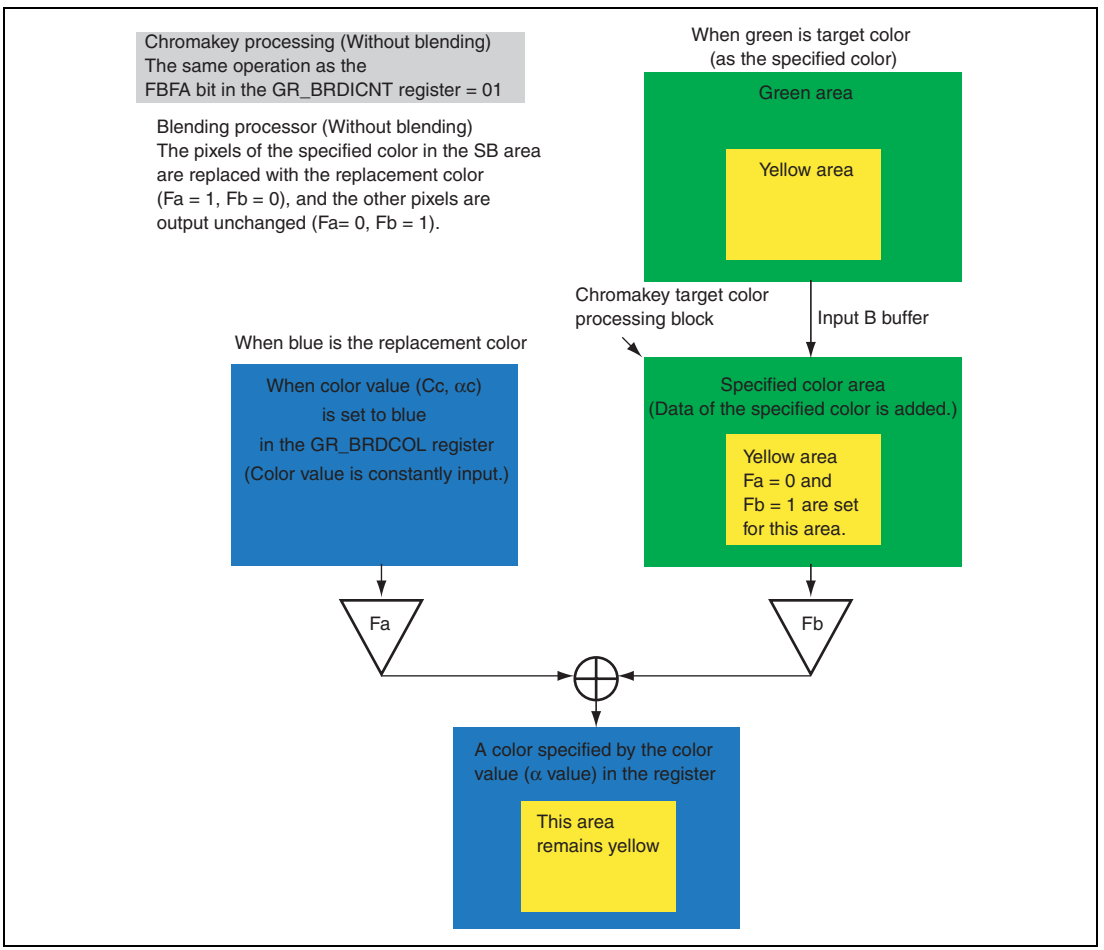


Figure 26.28 Example of Chromakey Processing (1)

- Chromakey processing example (2)

  Blit mode setting: Fill mode (GR_BLTMODE register, BTYPE bits = 10)

  Chromakey type selection: Chromakey blending (GR_BLTMODE register, CRKEY bits = 10)

  Blending coefficient selection:
  Blending coefficient = (GR_BRD1CNT register, FBFA bits = 00)

  SB output data selection: SB data selection = (GR_BLTMODE register, SBSEL bits = 01)

  Target color setting: Setting for green as the target color in the GR_DETCOL register

  Replacement color setting: Setting for blue as the replacement color in the GR_BRDCOL register

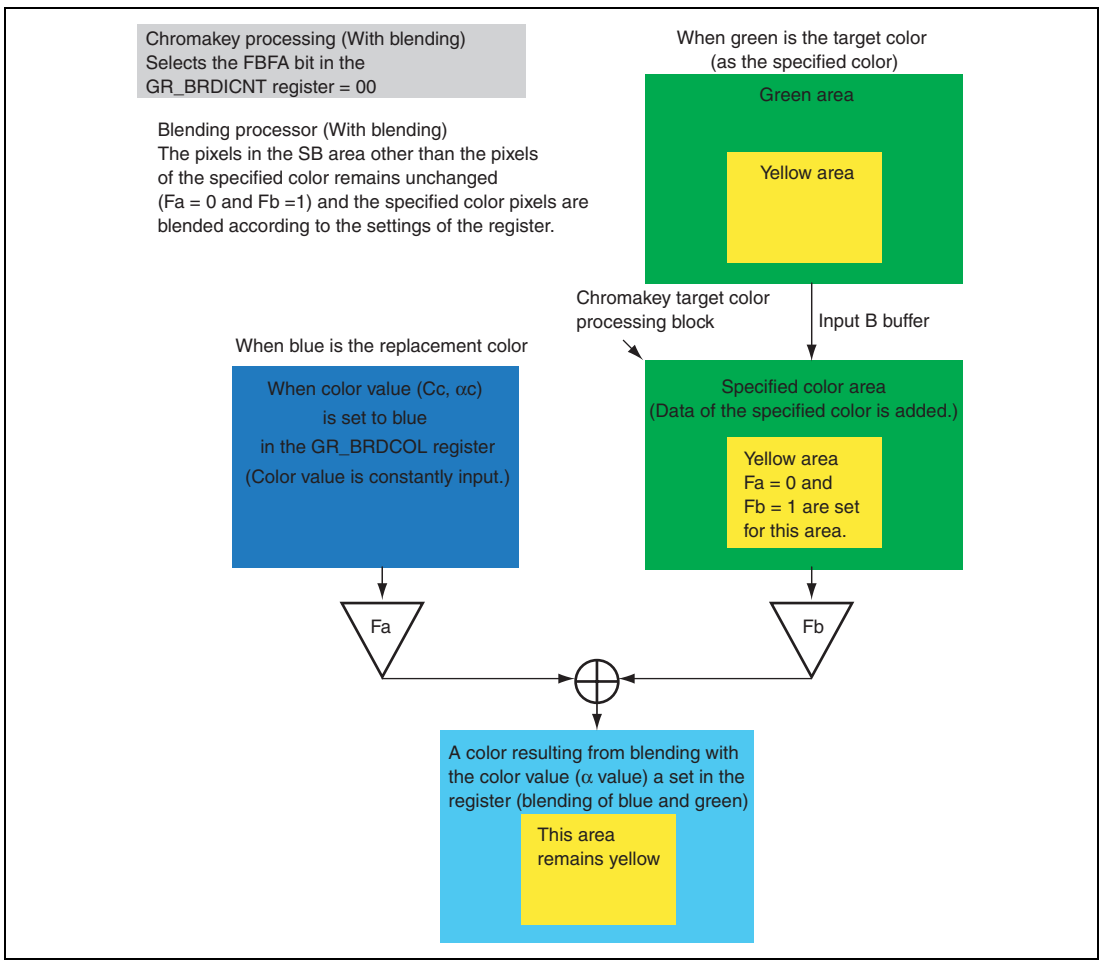


Figure 26.29 Example of Chromakey Processing (2)

(e) Summary of Logical Operations

Since the logical operation function is installed only on the input B path, it cannot be performed on the input A path. Furthermore, chromakey processing and color gradation processing cannot proceed simultaneously. As logical operation specifications, logical operation settings are enabled on all operating modes that are set by the BTYPE bits of the GR_BLTMODE register.

Examples of the effects of using logical operations are given below:

Blit mode setting:	Fill mode (GR_BLTMODE register, BTYPE bits = 10)
Blending coefficient selection:	Blending coefficient = (GR_BRD1CNT register, FBFA bits = 00)
Logical operation type selection:	Type of Logical operation (GR_BRD1CNT register, LGTYPE bits = 01)
Logical operation color setting:	Setting for white as the color for logical operations in the GR_LGDAT register
Replacement color setting:	Setting for gray as the replacement color in the GR_BRDCOL register

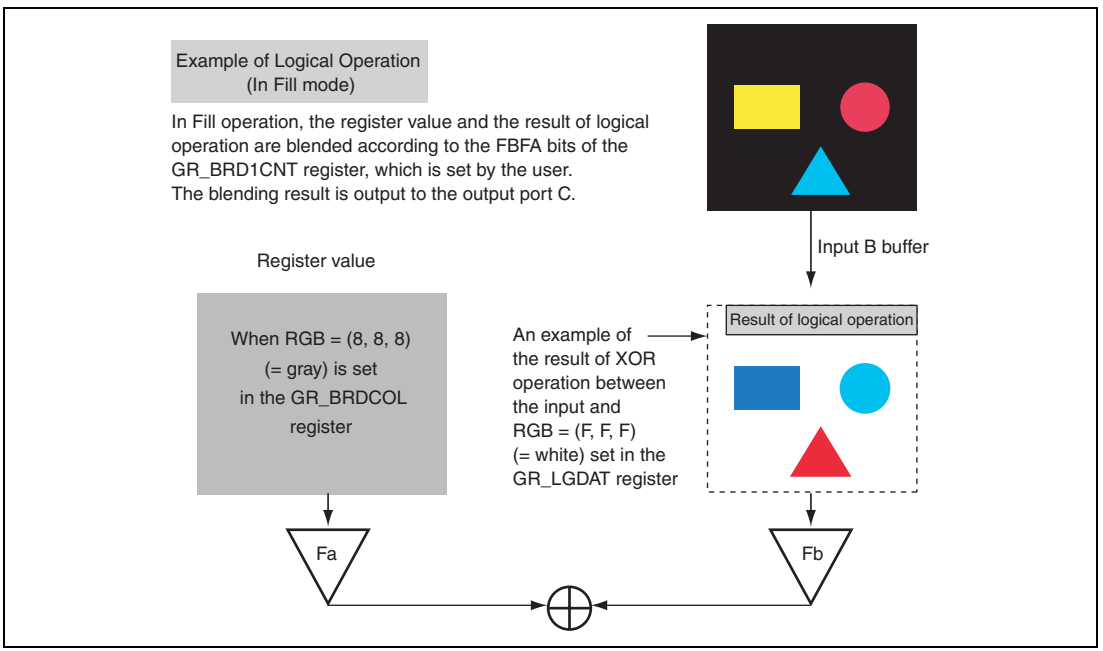


Figure 26.30 Example of Logical Operations

(f) Summary of Color Gradation Processing

Since the color gradation processing function is installed only on the input B path, it cannot be performed on the input A path. Irrespective of blending, color gradation processing is available only during blit operations. During such operations, chromakey processing or logical operations cannot be executed at the same time. Although resizing is possible, the use of it is not recommended in situations where character quality is important. Where possible, resizing should be avoided. The color gradation processing is not designed to accommodate partial resizing.

The basic operations of the color gradation processing are as follows:

1. DMA-transfers only a specified area from any font area, created with α (4 bits) on the SDRAM, to the 2DG.
2. Expands the color that is set in the GR_BRDCOL register.
3. Creates a color gradation according to the α value by weighting (multiplying) with the transferred α value and a specified color.

Depending on how a font is created using an α value, pseudo-anti-alias effects can also be produced.

- Color gradation processing example (1)
  Blit mode setting: Blit mode (GR_BLTMODE register, BTYPE bits = 00)
  Blending operation: Without blending (GR_BLTPLY register, SB_STEN bit = 1 and SA_STEN bit = 0)

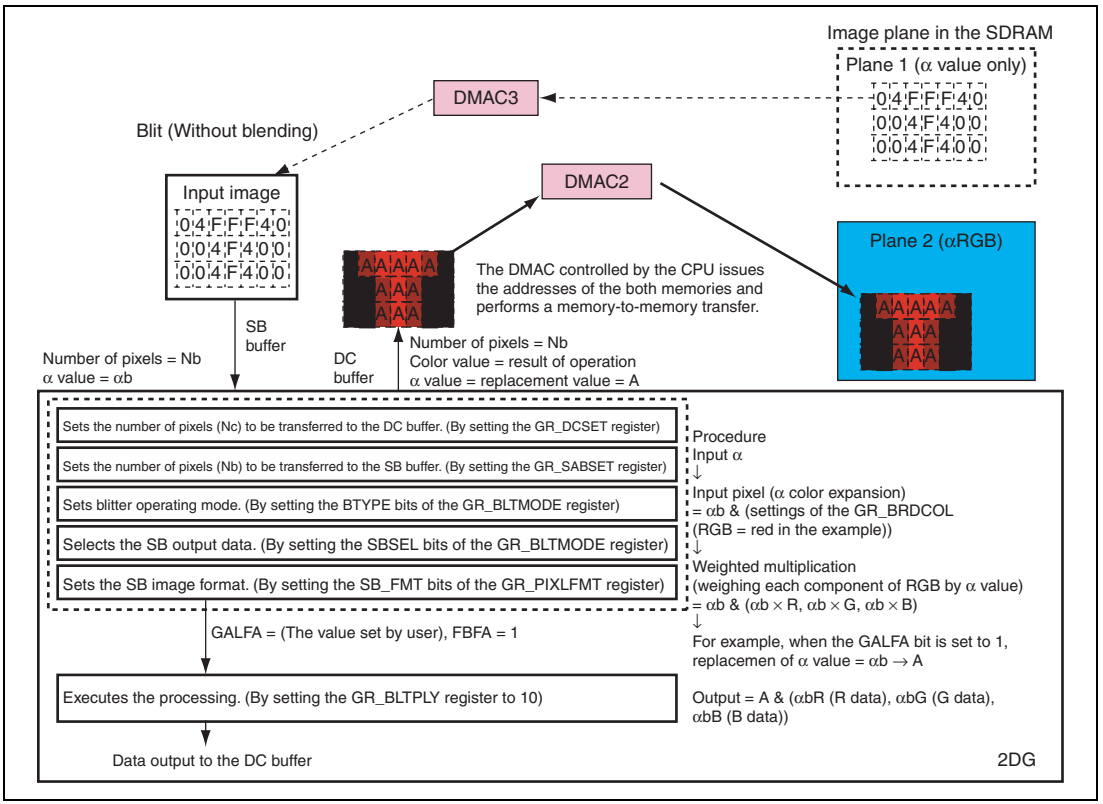


Figure 26.31 Example of Color Gradation Processing (1)

- Color gradation processing example (2)

  The text below describes the processing where data in a specified area from the α font region and data from a specified area on the output plane is entered into the 2DG, a color gradation processing is performed, and the results are written back to the specified area on the output plane:

  Blit mode setting: Blit mode (GR_BLTMODE register, BTYPE bits = 00)

  Blending operation: With blending (GR_BLTPLY register, SB_STEN = 1 and SA_STEN bit = 1)

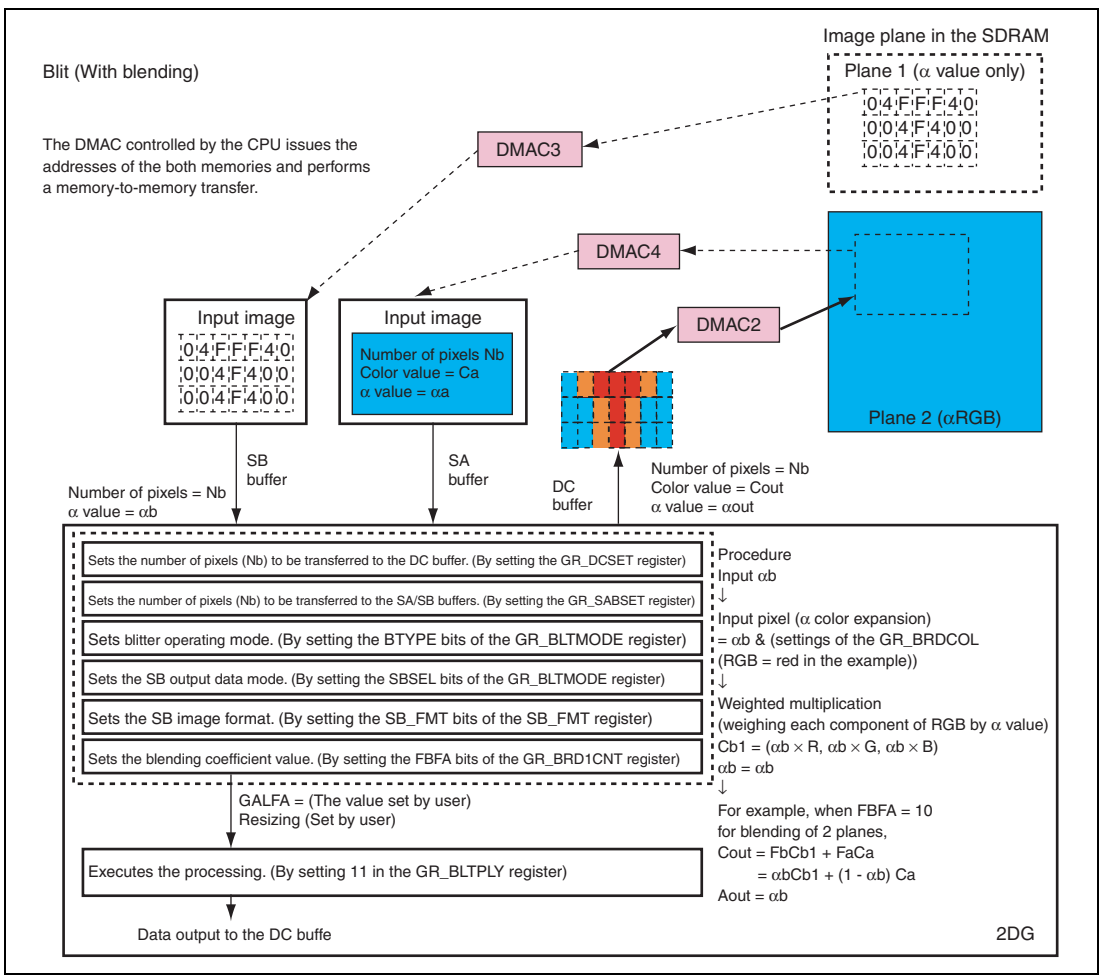


Figure 26.32 Example of Color Gradation Processing (2)

(4) Resizing

(a) Resize mode setting

Use of the resizing function in the blitter can be specified by the BRISZ bit of the GR_RISZSET register. Resize mode can be specified with respect to the following four factors using the relevant bits of the GR_RISZMOD register.

- A1_H bit: Selects the horizontal α-resizing method.
  0: Bilinear method, 1: nearest-neighbor method.
- H1_MTHD bit: Selects the horizontal resizing method.
  0: Bilinear method, 1: nearest-neighbor method.
- A1_V bit: Selects the vertical α-resizing method.
  0: Bilinear method, 1: nearest-neighbor method.
- V1_MTHD bit: Selects the vertical resizing method.
  0: Bilinear method, 1: nearest-neighbor method.

(b) Setting EDGE processing

When full or partial resizing is to be performed, it is necessary to determine whether the right or lower edge of the area to be resized coincides with the right or lower edge of the full resize plane. Therefore, when performing full or partial resizing, the EDGE bits of the GR_RISZSET register must be set accordingly.

An example of a setting for the EDGE bits of the GR_RISZSET register shows the case of the horizontal direction (right edge). In figure 26.33, the decision regarding the EDGE bits of the GR_RISZSET register is depicted in terms of the whole source area, the source update area, and the source-setting area. The mass with an attached color is the region to be updated within the whole source area (the updated source area). Furthermore, when the region indicated by the bold arrow is partially resized, this region is set in registers as the source area (source-setting area). For details, see section 26.4.3 (5), Partial Resizing.

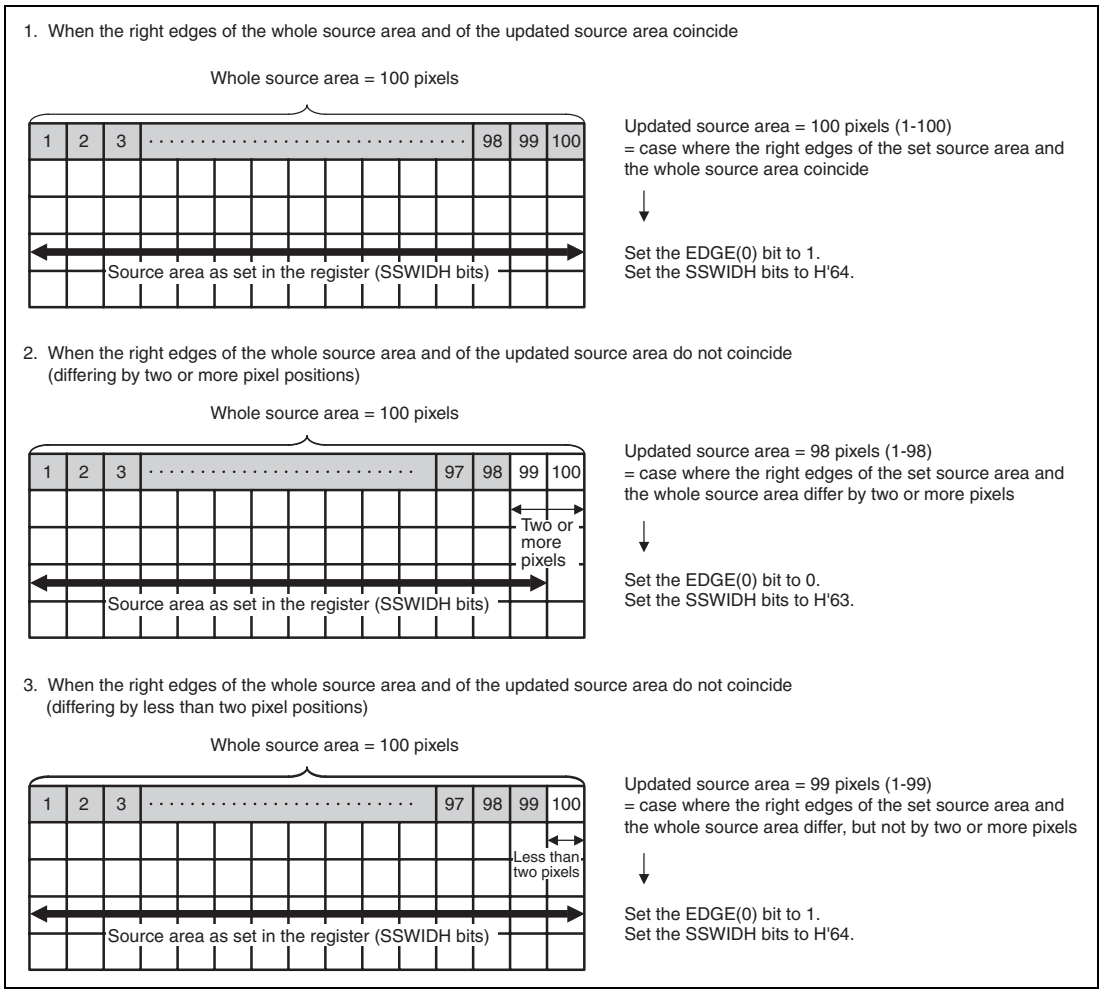


Figure 26.33 Settings of EDGE Bits of GR_RISZSET Register (only for horizontal direction)

(c) Setting the delta value for use in resizing

The delta value for use in resizing (Ch) can be calculated by using the following equation:

$$Ch = (\text{source resolution} / \text{destination resolution}) \times 4096$$

Note: Ch = 1/resizing ratio must be calculated from (number of source pixels/number of destination pixels). Also, truncate any fractional part of the result.

For example, if the number of source pixels is 479 and the number of destination pixels is 240, Ch will be:

$$Ch = (479 / 240) \times 4096 \cong 8174.933 = 1FEE\ (H)$$

Based on these results, the delta value for resizing will be as follows:

Since HDLT_INTGR is the integer part of Ch, HDLT_INTGR = 1 (H)

Since HDLT_DCML is the fractional part of Ch, HDLT_DCML = FEE (H)

If the BRSIZ bit of the GR_RISZSET register is 0, i.e., the resizing function is not to be used, the operation will be the same as if the HDLT_INTGR and HDLT_DCML bits were set to H'1 and H'000, respectively. However, the values of the HDLT_INTGR and HDLT_DCML bits do not change.

The delta value for use in resizing (Cv) value can be calculated by using the following equation:

$$Cv = (\text{source resolution} / \text{destination resolution}) \times 4096$$

In this case, through calculation similar to that for Ch, VDLT_INTGR and VDLT_DCML are derived as the integer and fractional parts of Cv (2 and 12 bits, respectively). If the BRSIZ bit of the GR_RISZSET register is 0, i.e., the resizing function is not to be used, the operation will be the same as if the VDLT_INTGR and VDLT_DCML bits were set to H'1 and H'000, respectively. However, the values of the VDLT_INTGR and VDLT_DCML bits do not change.

- Range of settings for full resizing

  Since this is full resizing, the integer component (GR_HSPHAS register, H1PHS_INTGR bit) is always 0.

  Enlargement: H1PHS_INTGR = H'000, H1PHS_DCML = H'000 to H'(HDLT_DCML − 1)

  Reduction: H1PHS_INTGR = H'000, H1PHS_DCML = H'000 to H'FFF
- Range of settings for partial resizing

  Set the integer and fractional parts according to the left-edge pixels of the area for partial resizing.

(d) Setting the source-side starting phase

The source-side starting phase (Psh) can be calculated by using the following equation:

$$Psh = Ch \times (\text{number of pixels to that where resizing starts}) + (\text{initial phase at starting pixel} \times 4096)$$

In this case, the integer part (10 bits) of Psh will be in H1PHS_INTGR and the fractional part (12 bits) of Psh will be in H1PHS_DCML. If there is to be no resizing or the magnification is to remain the same (resizing by a factor of one), set Psh = 0.

The source-side starting phase (Psv) can be calculated by using the following equation:

Psv = Cv × (number of pixels to that where resizing starts) + (initial phase at starting pixel × 4096)

In this case, the integer part (9 bits) of Psv will be in V1PHS_INTGR and the fractional part (12 bits) of Psv will be in V1PHS_DCML. If there is to be no resizing or the magnification is to remain the same (resizing by a factor of one), set Psv = 0.

- Range of settings for full resizing
  Since this is full resizing, the integer part (V1PHS_INTGR) is always 0.
  Enlargement: V1PHS_INTGR = H'000, V1PHS_DCML = H'000 to H'(VDLT_DCML – 1)
  Reduction: V1PHS_INTGR = H'000, V1PHS_DCML = H'000 to H'FFF
- Range of settings for partial resizing
  Set the integer and fractional parts for correspondence with the pixels at the top-edge of the area for partial resizing.

The starting phase is used to vary the proportion of mixing for the two source pixels used as a reference for bilinear filtering in resizing. This can be used to eliminate the omission of pixels in halving of the size (reduction by a factor of two) and so on. However, too large an initial phase can lead to a mismatch of colors at the left edge of the destination. For these reasons, the following limitations apply to settings for the starting phase in resizing.

Limitations:

Always set the integer components of the starting phase (the H1PHS_INTGR and V1PHS_INTGR bits) to zero.

Set the fractional components of the starting phase (the H1PHS_DCML and V1PHS_DCML bits) within the ranges given below:

— Resizing for enlargement: H1PHS_DCML and V1PHS_DCML bits = H'000 to H'(VDLT_DCML – 1)

— Resizing for reduction: H1PHS_DCML and V1PHS_DCML bits = H'000 to H'FFF

Examples of the settings for starting phase in resizing for reduction (magnification by half) are given below (the explanation only applied to the horizontal direction). In figure 26.34, S0 to S4 are source pixels and D0 to D2 are the destination pixels interpolated for the given magnification factor. Destination pixels (1) and (2) indicate the phases for interpolation of the destination pixels for starting phase settings of H'0 and H'800, respectively.

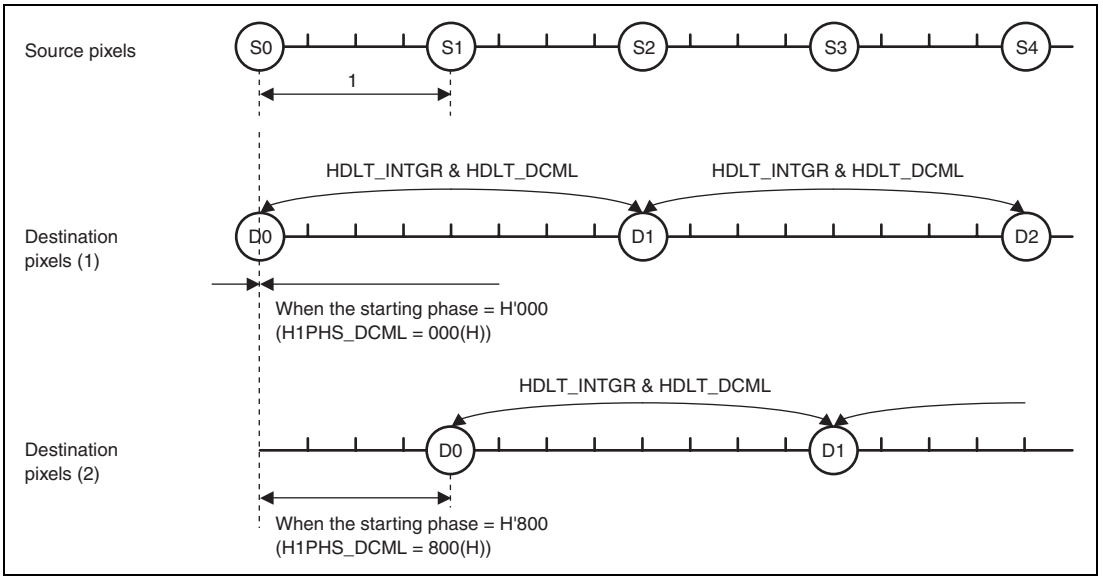


Figure 26.34 Settings for Starting Phase in Resizing for Reduction (Magnification by Half)

The fractional components of the starting phase (the H1PHS_DCML and V1PHS_DCML bits) are asset for an interval of one between source pixels. In figure 26.34, destination pixels (1) shows the positions of interpolation for the destination pixels when the starting phase (H1PHS_DCML) = H'000, where the position of the leading pixel (D0) is generated with a phase of zero. Destination pixels (2) shows the positions of interpolation for the destination pixels when the starting phase (H1PHS_DCML) = H'800, where the position of the leading pixel (D0) is advanced by only the starting phase (H1PHS_DCML), and the starting phase is maintained in subsequent interpolation. The proportions of the two source pixels for reference can be varied by attaching or not attaching a starting phase in this way.

The method can thus be used to alleviate the effects of the omission of pixels. Figure 26.35 shows examples of the results when a starting phase is and is not attached in resizing for reduction (magnification by half).

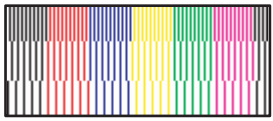
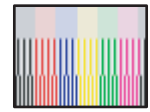
Figure 25.34: Source pixels, original image	Figure 25.34: Destination pixels (1) Through the bilinear method, reduction by 1/2 in the horizontal direction only Starting phase (H1PHS_DCML) = H'000	Figure 25.34: Destination pixels (2) Through the bilinear method, reduction by 1/2 in the horizontal direction only Starting phase (H1PHS_DCML) = H'800
		
The image consists of colored vertical bars, with the color omitted for one dot, two dots, and three dots between bars for the upper third, central third, and lower third, respectively. Image size: 130 × 60 pixels	Since the contribution of the colored pixels is cut from the upper third, where the source image had colored bars separated by one dot, the corresponding area is white (the proportion of the color value to the white value is 1:0).	In the case of the upper third, where the source image had colored bars separated by one dot, since the color value and white value are blended by averaging, the color value is taken into account even though the lines disappear (the proportion of the color value to the white value is 1:1).

Figure 26.35 Examples of the Results when Settings for Starting Phase in Resizing for Reduction (Magnification by Half)

As shown in figure 26.35, when resizing for reduction of the upper third with one dot omitted is obtained by the bilinear method with a starting phase of H'000, the proportion between the color values of the white and colored pixels becomes 1:0, so the color value of the colored pixels is simply eliminated. When the starting phase is H'800, the proportion between the color values of the white and colored pixels becomes 1:1 and, since the color values of the white and colored pixels are blended in equal portions, the color of the colored pixels is reproduced although the vertical lines disappear.

Since pixels are not omitted in the case of resizing for enlargement, the kind of strong effect seen in the case of resizing for reduction is absent. However, the setting for starting phase can still be used to control the proportions of the two source pixels used as reference in resizing for enlargement (through the bilinear method).

(5) Partial Resizing

When modifying a part of the area on a resized source plane, a partial extraction should be performed on the source area before resizing instead of after. The reason is that the correct boundaries can be maintained by partially extracting the source area before resizing and by pasting the area to the part of the area on the resized plane. An example of this is shown in figure 26.36. Partial resizing, however, cannot be performed when color gradation processing is selected.

The partially modified area (the updated area in figure 26.36) is defined as follows:

Vertical offset = Va, horizontal offset = Ha, height = Vb, width = Hb

The register settings for the horizontal width and vertical height of the source area are specified by determining a setting area appropriate for partial resizing, instead of specifying the updated area itself, according to the following equations, and by assigning the results to the GR_HSPHAS and GR_VSPHAS registers:

Vertical offset = Va1 = Va – Vx1, horizontal offset: Ha1 = Ha – Hx1

Vertical height: Vb1 = Vb + Vx1 + Vx2, horizontal width: Hb1 = Hb1 + Hx1 + Hx2

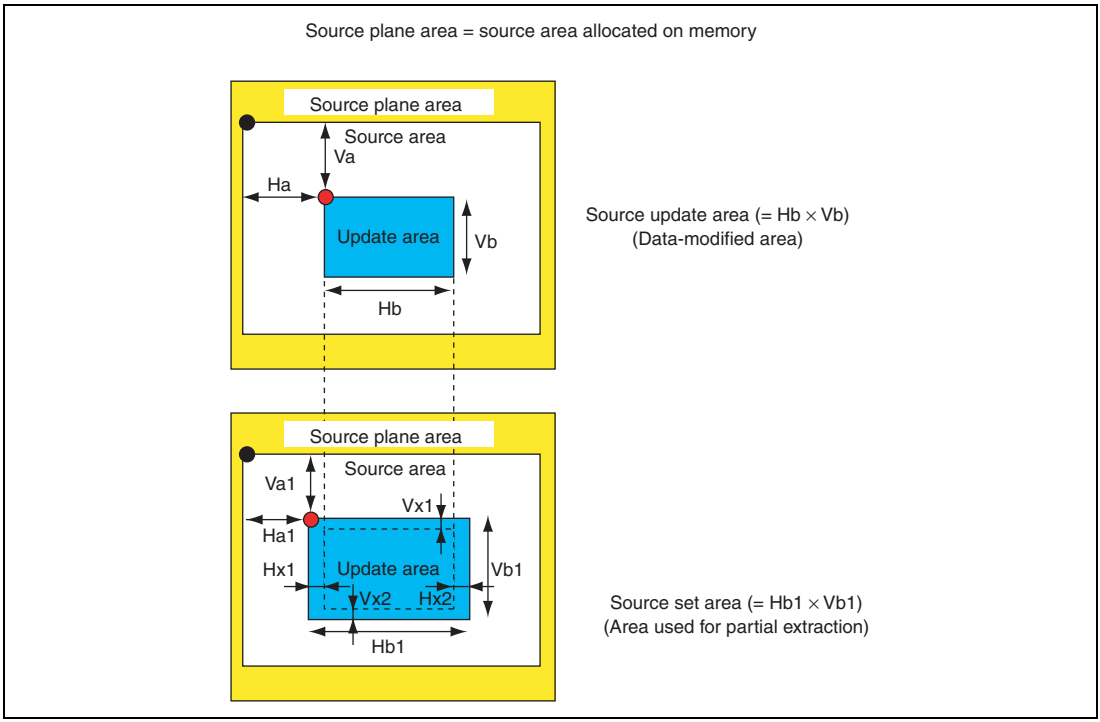


Figure 26.36 Setting Areas for Partial Extraction Resizing

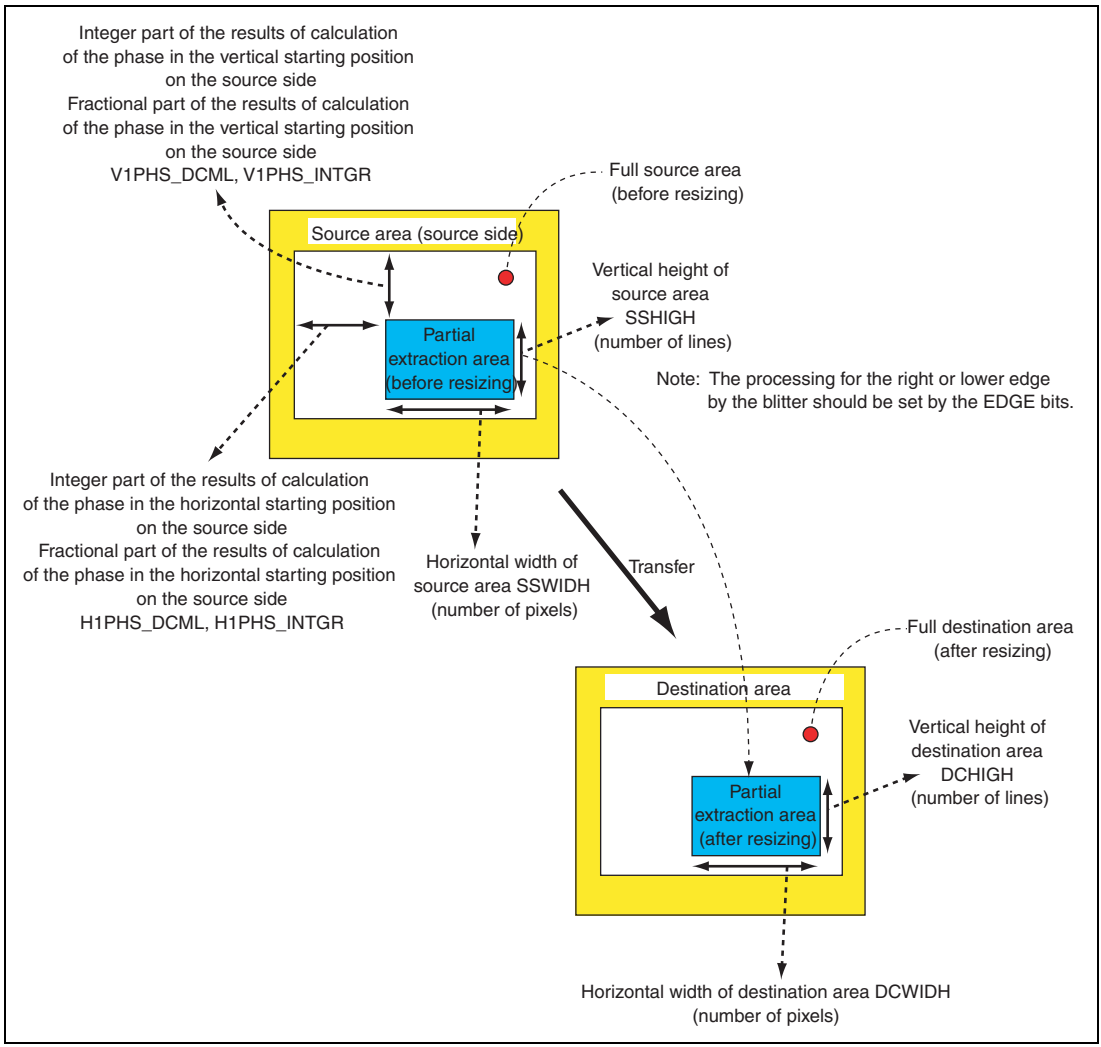


Figure 26.37 Summary of Partial Extraction Resizing

(a) What is partial resizing?

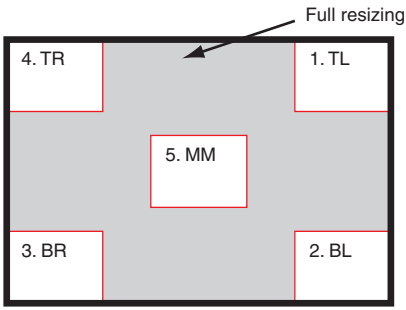
Given an original source image area (for example, a 500 × 300 pixel area) for a fully resized destination image, if only a part (for example, 50 × 50 pixels) of the source image area is modified, partial resizing refers to the method by which only the modified area (approximately 50 × 50 pixels) is resized, instead of resizing the entire source image area (500 × 300).

(b) Defining the area to be partially resized

- EDGE bits of GR_RISZSET register

When performing full or partial resizing, it is necessary to determine whether the right edge or the lower edge of the source area to be resized coincides with the right or lower edge of the full-resize area. For this reason, when performing a full or partial resizing, the EDGE bits of the GR_RISZSET register must be set.

Figure 26.38 explains the values of the EDGE bits to be set in the GR_RISZSET register for areas 1 (TL) to 5 (MM) for partial resizing relative to the full resizing area (area in the bold frame). In reference to the pointers given below, set the EDGE bits according to the position of the partial resizing area.



1. TL: Not on the lower edge in the vertical direction, but on the right edge in the horizontal direction. EDGE = 01
2. BL: On the lower edge in the vertical direction, and on the right edge in the horizontal direction. EDGE = 11
3. BR: On the lower edge in the vertical direction, but not on the right edge in the horizontal direction. EDGE = 10
4. TR: Not on the lower edge in the vertical direction, and not on the right edge in the horizontal direction. EDGE = 00
5. MM: Not on the lower edge of the vertical direction, and not on the right edge in the horizontal direction. EDGE = 00

Note: For full resizing, EDGE should be set to 11 (Initial value).

Figure 26.38 Examples of Partial Resizing Areas

- Partial resizing source set area (SSWIDH and SSHIGH bits of GR_SABSET register)

For partial resizing, an area a little larger than the actually updated area is assigned to the register as a partial resizing source area. In this manner, the boundary between the partially updated area and the area that is not updated is eliminated. An example of this given shown in figure 26.39. The area to be set can vary according to the particular resizing ratio (enlargement/reduction) employed.

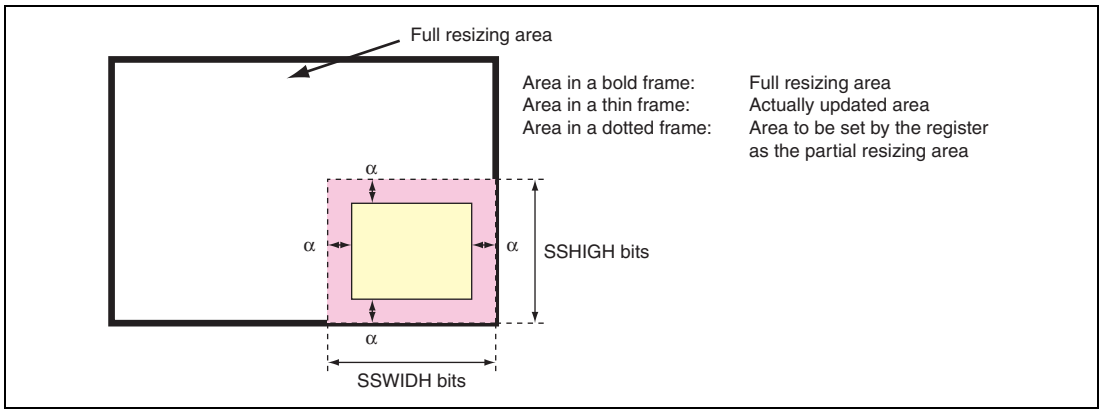


Figure 26.39 Partial Resizing Set Area

As shown in figure 26.39, an area expanded by +α vertically and horizontally from the actually updated area (yellow) is assigned to the partial resizing set area (SSWIDH bits/SSHIGH bits). Also, as shown in figure 26.39, if the partially updated area (yellow) is close to the right or lower edge of the fully resized area, in some cases the partially updated area is defined on the right or lower edge as a set area, different from the actual updated area. In such a case, the right and lower edges must be specified to the EDGE bits. (The factor α varies with the resizing ratio (enlargement/reduction)).

(c) Horizontal/vertical starting phase for the source (H1PHS_DCML bits of GR_HSPHAS register /V1PHS_DCML bits of GR_VSPHAS register)

When performing partial resizing, it is necessary to consider the starting phase of the starting pixel for the full resizing in addition to the set area for the partial resizing, in order to ensure that the complete match of the boundary between the output image that is fully resized in advance and the output image that is partially resized later. For example, if the horizontal starting phase (H1PHS_DCML) for the source for full resizing is defined as H'FFF, the same value as full resizing (H'FFF) must also be added to the starting phase (H1PHS_DCML) when partial resizing is performed.

If the parameter H1PHS_DCML or V1PHS_DCML is to be used during the full resizing process, constraint conditions vary between enlargement resizing and reduction resizing; therefore, each parameter should be set in the following ranges:

[Enlargement]
H1PHS_DCML or V1PHS_DCML = H'000 to H'(HDLT_DCML − 1)
[Reduction]
H1PHS_DCML or V1PHS_DCML = H'000 to H'FFF

(d) Determining the area to set up for partial resizing

In figure 26.40, ST, SL, SSHIGH, and SSWIDH are values to be obtained by the user. The other values are known.

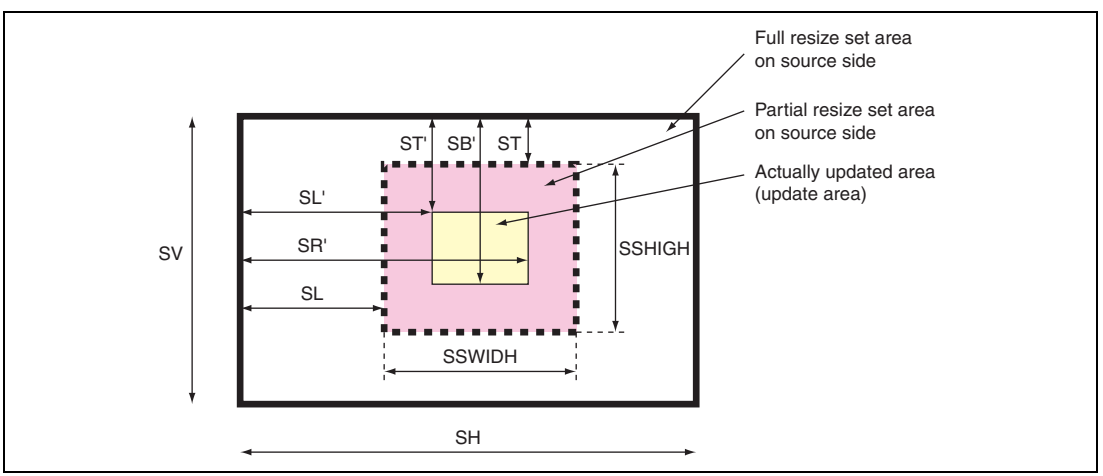


Figure 26.40 Partial Resizing Set Area in Source Data Area

In figure 26.41, DT, DL, DCHIGH, and DCWIDH are values to be obtained by the user. The other values are known.

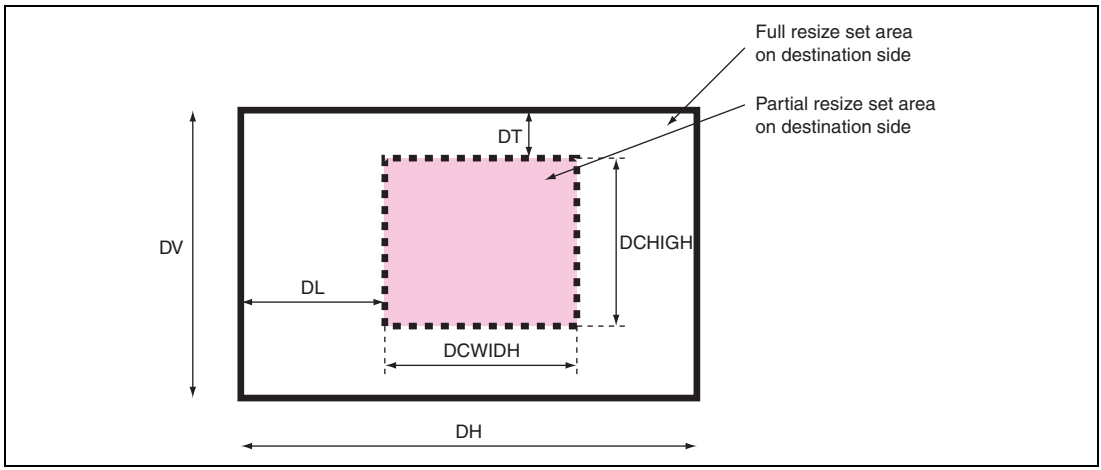


Figure 26.41 Partial Resizing Set Area in Destination Data Area

- Definitions: Explanation of codes provided in figures 26.40 and 26.41

  SH: Width of the horizontal resize set area on source side when the data is fully resized

  SV: Height of the vertical resize set area on source side when the data is fully resized

  SSWIDH: Width of the horizontal resize set area on source side when the data is partially resized

  SSHIGH: Height of the vertical resize set area on source side when the data is partially resized

  DH: Width of the horizontal resize set area on destination side when the data is fully resized

  DV: Height of the vertical resize set area on destination side when the data is fully resized

  DCWIDH: Width of the horizontal resize set area on destination side when the data is partially resized

  DCHIGH: Height of the vertical resize set area on destination side when the data is partially resized

ST': The number of lines from the starting line for the vertical source read-out area for full resizing to the top-edge pixel in the vertical source read-out area in the updated area

SB': The number of lines from the starting line for the vertical source read-out area for full resizing to the lower-edge pixel in the vertical source read-out area in the updated area

SL': The number of pixels from the left-edge pixel for the horizontal source read-out area for full resizing to the left-edge (starting) pixel in the horizontal source read-out area in the updated area

SR': The number of pixels from the left-edge pixel for the horizontal source read-out area for full resizing to the right-edge (end) pixel in the horizontal source read-out area in the updated area

ST: The number of lines from the starting line for the vertical source read-out area for full resizing to the top-edge pixel in the vertical source read-out area in the partial resize set area

SL: The number of pixels from the left-edge pixel for the horizontal source read-out area for full resizing to the left-edge (starting) pixel in the horizontal source read-out area in the partial resize set area

DT: The number of lines from the starting line for the vertical destination area for full resizing to the top-edge pixel in the vertical destination area in the partial resize set area

DL: The number of pixels from the left-edge pixel for the horizontal destination area for full resizing to the left-edge (starting) pixel in the horizontal destination area in the partial resize set area

- Definitions: Explanation of codes not provided in figures 26.40 or 26.41

  GR_HSPHAS register, H1PHS_DCML bit:

  For full resizing, the fractional part of the results of calculation of the phase in the horizontal starting position on the source side

  GR_VSPHAS register, V1PHS_DCML bit:

  For full resizing, the fractional part of the results of calculation of the phase in the vertical starting position on the source side

1. Formula for calculating the number of left-edge pixels (DL) and the number of top-edge lines (DT) in the destination set area

   Formula for judging DL:

   When SL' > 1, INT (DH/ SH × (SL' − 1)) × INT (SH/ DH × 4096) + H1PHS_DCML ≥ (SL' − 1) × 4096

   When SL' ≤ 1, DL = 0 for exception handling, regardless of formula for calculation

Formula for calculating DL:

When the result of formula for judging is true: DL = INT (DH/SH × (SL' − 1))

When the result of formula for judging is false: DL = INT (DH/SH × (SL' − 1)) + 1

Formula for judging DT:

When ST' > 1, INT (DV/ SV × (ST' − 1)) × INT (SV/ DV × 4096) + V1PHS_DCML ≥ (ST' − 1) × 4096

When ST' ≤ 1, DT = 0 for exception handling, regardless of formula for calculation

Formula for calculating DT:

When the result of formula for judging is true: DT = INT (DV/ SV × (ST' − 1))

When the result of formula for judging is false: DT = INT (DV/ SV × (ST' − 1)) + 1

2. Formula for calculating the number of left-edge pixels (SL) and the number of top-edge lines (ST) in the source set area

Formula for calculating SL:

When SL' > 1, SL = INT ((DL × INT (SH/ DH × 4096) + H1PHS_DCML/ 4096) = (H1PHS_INTGR)

When SL' ≤ 1, SL = 0 as exception handling

Formula for calculating ST:

When ST' > 1, ST = INT ((DT × INT (SV/ DV × 4096) + V1PHS_DCML/ 4096) = (V1PHS_INTGR)

When ST' ≤ 1, ST = 0 as exception handling

3. Formula for calculating the horizontal width (DCWIDH) and the vertical height (DCHIGH) in the destination set area

Formula for judging DCWIDH:

When SR' < SH – 2, INT ((INT (DH/ SH × (SR' + 1)) × INT (SH/ DH × 4096) + H1PHS_DCML ≥ (SR' + 1) × 4096

When SR' ≥ SH – 2, DCWIDH = DH – DL for exception handling, regardless of formula for calculation

Formula for calculating DCWIDH:

When the result of formula for judging is true: DCWIDH = INT (DH/ SH × (SR' + 1)) – DL

When the result of formula for judging is false: DCWIDH = INT (DH/ SH × (SR' + 1)) – DL + 1

Formula for judging DCHIGH:

When SB' < SV – 2, INT ((INT (DV/ SV × (SB' + 1)) × INT (SV/ DV × 4096) + V1PHS_DCML ≥ (SB' + 1) × 4096

When SB' ≥ SV – 2, DCHIGH = DV – DT for exception handling, regardless of formula for calculation

Formula for calculating DCHIGH:

When the result of formula for judging is true: DCHIGH = INT (DV/ SV × (SB' + 1)) – DT

When the result of formula for judging is false: DCHIGH = INT (DV/ SV × (SB' + 1)) – DT + 1

4. Formula for calculating the horizontal width (SSWIDH) and the vertical height (SSHIGH) in the source set area

Formula for judging SSWIDH:

When SR' < SH– 2, INT ((INT × (DH/ SH (SR' + 1)) × INT (SH/ DH × 4096) + H1PHS_DCML ≥ (SR' + 1) × 4096

When SR' ≥ SH – 2, SSWIDH = SH – SL for exception handling, regardless of formula for calculation

Formula for calculating SSWIDH:

When the result of formula for judging is true: SSWIDH = INT (((INT (DH/ SH × (SR' + 1)) – 1) × INT (SH/ DH × 4096) + H1PHS_DCML)/ 4096) – SL + 2

When the result of formula for judging is false: SSWIDH = INT (((INT (DH/ SH × (SR' + 1)) – 0) × INT (SH/ DH × 4096) + H1PHS_DCML)/ 4096) – SL + 2

Formula for judging SSHIGH:

When SB' < SV– 2, INT (DV/ SV × (SB' + 1)) × INT (SV/ DV × 4096) + V1PHS_DCML ≥ (SB' + 1) × 4096

When SB' ≥ SV – 2, SSHIGH = SV – ST for exception handling, regardless of formula for calculation

Formula for calculating SSHIGH:

When the result of formula for judging is true: SSHIGH = INT (((INT (DV/ SV × (SB' + 1)) – 1) × INT (SV/ DV × 4096) + V1PHS_DCML)/ 4096) – ST + 2

When the result of formula for judging is false: SSHIGH = INT (((INT (DH/ SH × (SR' + 1)) – 0) × INT (SV/ DV × 4096) + V1PHS_DCML)/ 4096) – ST + 2

5. Calculation of the phase in the starting position on the source side (PHS_H/ PHS_V)

PHS_H = INT (SH/ DH × 4096) × DL + H1PHS_DCML

Integer part of the results of calculation of the phase in the starting position on the source side (H1PHS_INTGR) = PHS_H (upper 10 bits) = H'

Fractional part of the results of calculation of the phase in the starting position on the source side (H1PHS_DCML) = PHS_V (lower 12 bits) = H'

PHS_V = INT (SV/ DV × 4096) × DT + V1PHS_DCML

Integer part of the results of calculation of the phase in the starting position on the source side (V1PHS_INTGR) = PHS_H (upper 9 bits) = H'

Fractional part of the results of calculation of the phase in the starting position on the source side (V1PHS_DCML) = PHS_V (lower 12 bits) = H'

Note: In the above formulas, the letters INT mean that the fractional part is rounded to an integer.

(e) Calculation examples of the values in the register setting (only for horizontal direction)

- Conditions

  Width of the source area (SH): 280 pixels

  Width of the destination area (DH): 350 pixels

  Number of the left-edge pixel of the actually updated partial area of the source area (SL'): 55

  Number of the right-edge pixel of the actually updated partial area of the source area (SR'): 133

  Fractional part of the results of calculation of the phase for the starting position on the source side (H1PHS_DCML (D)): 819 = H'333

- Parameters to be calculated by the user

  In the case of partial resizing, the number of the left-edge pixel of the area set as the destination (DL)

  In the case of partial resizing, the number of the left-edge pixel of the area set as the source (SL)

  In the case of partial resizing, the width of the area set as the destination (DCWIDH)

  In the case of partial resizing, the width of the area set as the source (SSWIDH)

  Integer part of the result of calculating the phase of the starting position on the source side (H1PHS_INTGR)

  Fractional part of the result of calculating the phase of the starting position on the source side (H1PHS_DCML)

1. Calculating DL, the left-edge pixel of the area set as the destination

   Since SL' = 55 pixels, judgment proceeds as follows.

   Conditional expression:

   INT (DH/SH × (SL' − 1)) × INT (SH/DH × 4096) + H1PHS_DCML ≥ (SL' − 1) × 4096

   INT (350/280 × (55 − 1)) × INT (280/350 × 4096) + 819 ≥ (55 − 1) × 4096

   Since the result of the conditional expression is "false", the value is obtained as shown below.

   False: DL = INT (DH/SH × (SL' − 1)) + 1 = INT (350/280 × (55 − 1)) + 1 = 68

2. Calculating SL, the left-edge pixel of the area set as the source

   Since SL' = 55 pixels, the value is obtained as follows.

   SL = INT ((DL × INT (SH/DH × 4096) + H1PHS_DCML)/ 4096

   = INT ((68 × INT (280/350 × 4096) + 819)/ 4096) = 54

   From the above, the result for the integer part of the phase for the source-side starting position is obtained as H1PHS_INTGR = 54.

3. Calculating DCWIDH, the width of the destination area

   Since SR' = 133 and SH = 280 pixels, judgment proceeds as follows.

   Conditional expression:

   INT (DH/SH × (SR' + 1)) × INT (SH/DH × 4096) + H1PHS_DCML ≥ (SR' + 1) × 4096

   INT (350/280 × (133 + 1)) × INT (280/350 × 4096) + 819 ≥ (133 + 1) × 4096

   Since the result of the conditional expression is "false", the value is obtained as shown below.

   False: DCWIDH

   = INT (DH/SH × (SR' + 1)) − DL + 1 = INT (350/280 × (133 + 1)) − 68 + 1 = 100

4. Calculating SSWIDH, the width of the source area

   Since SR' = 133 and SH = 280 pixels, judgment proceeds as follows.

   Conditional expression:

   INT (DH/SH × (SR' + 1)) × INT (SH/DH × 4096) + H1PHS_DCML ≥ (SR' + 1) × 4096

   INT (350/280 × (133 + 1)) × INT (280/350 × 4096) + 819 ≥ (133 + 1) × 4096

   Since the result of the conditional expression is "false", the value is obtained as shown below.

   False: SSWIDH

   = INT (((INT (DH/SH × (SR' + 1)) − 0) × INT (SH/DH × 4096) + H1PHS_DCML)/ 4096) − SL+ 2

   = INT (((INT (350/280 × (133 + 1)) − 0) × INT (280/350 × 4096) + 819)/ 4096) − 54 + 2 = 81

Note: Since SL + SSWIDH ≠ SH at this time, the horizontal edge is not at the right edge. Accordingly, set the EDGE (0) bit to 0.

5. Calculating PHS_H, the result of phase calculation for the source-side starting position

   PHS = INT (INT (SH/DH × 4096) × DL + H1PHS_DCML) = INT (INT (280/350 × 4096) × 68 + 819) = 223587 (H'036963)

   From the above result of phase calculation for the source-side starting position, the following result is obtained.

   H1PHS_INTGR, integer part of the result of phase calculation for the source-side starting position = PHS_H (upper 10 bits) = H'036

   H1PHS_DCML, fractional part of the result of phase calculation for the source-side starting position = PHS_H (lower 12 bits) = H'963

(f) Determining the starting address of the partial resize area to be assigned to the DMAC by the CPU

- Determining the starting address (Sa) of the source set area

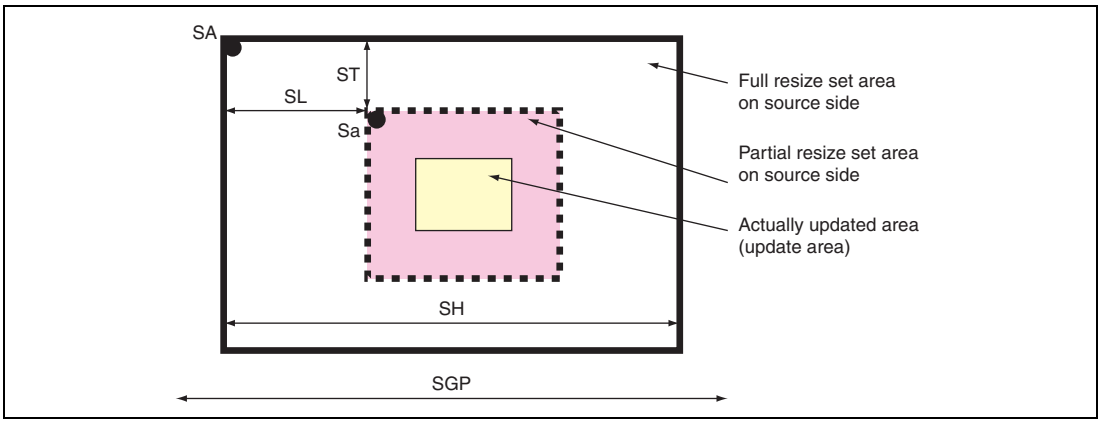


Figure 26.42 Determination of Starting Address (Sa) of Source Set Area

[Definition]

Sa: Starting address for reading the source data for partial resizing

SA: Starting address for reading the source data for full resizing

SGP: Line pitch in the source plane (a 64-byte boundary)

SGP = ROUNDUP (byte count per pixel) × SH/64 bytes) × 64 bytes

Note: ROUNDUP in the formula indicates rounding up, i.e. taking the nearest integer above the fractional component.

ST: The number of lines from the starting line for the vertical source read-out area for full resizing to the top-edge pixel in the vertical source read-out area for partial resizing

SL: The number of pixels from the left-edge pixel for the horizontal source read-out area for full resizing to the left-edge (starting) pixel in the horizontal source read-out area for partial resizing

Formula for calculating Sa:

Sa = SA + SGP × ST + SL × (byte count per pixel)

- Determining the starting address (Da) of the destination set area

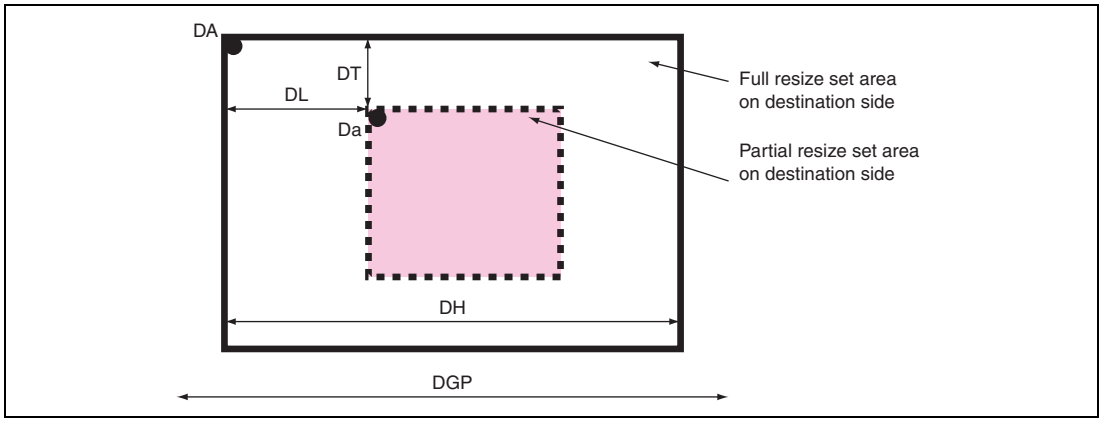


Figure 26.43 Determination of Starting Address (Da) of Destination Set Area

[Definition]

Da: Starting address for writing to the destination area for partial resizing

DA: Starting address for writing to the destination area for full resizing

DGP: Line pitch in the destination area (a 64-byte boundary)

DGP = ROUNDUP (byte count per pixel) × DH/64 bytes) × 64 bytes

Note: ROUNDUP in the formula indicates rounding up, i.e. taking the nearest integer above the fractional component.

DT: The number of lines from the starting line for the vertical destination area for full resizing, to the top-edge pixel in the vertical destination area for partial resizing

DL: The number of pixels from the left-edge pixel for the horizontal destination area for full resizing, to the left-edge (starting) pixel in the horizontal destination area for partial resizing

Formula for calculating Da:

Da = DA + DGP × DT + DL × (byte count per pixel)

- Allocation within memory space

Figure 26.44 depicts the allocation within memory space of a source image that is 100 pixels wide and three lines high.

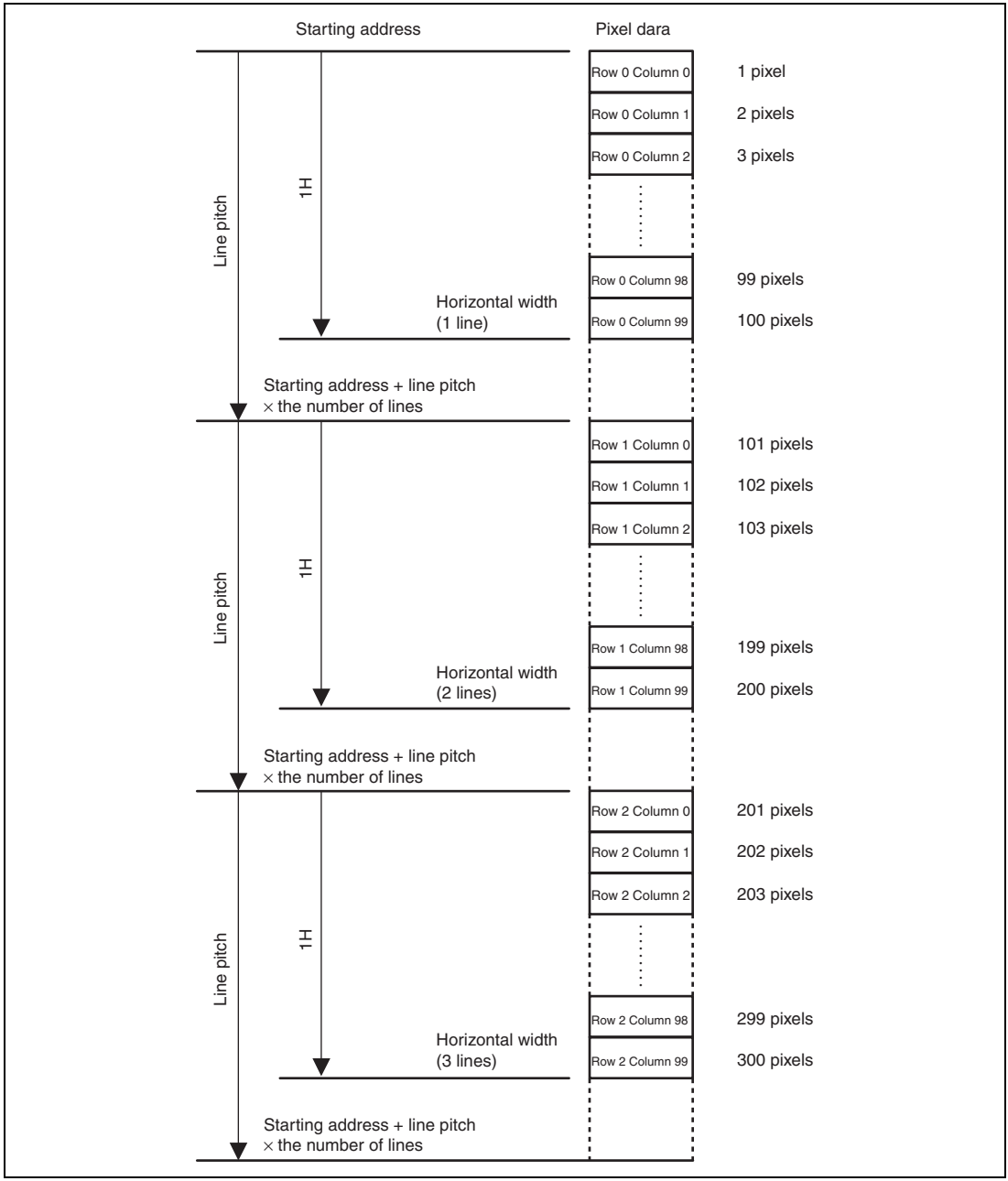


Figure 26.44 Allocation of images within memory space

- Examples of calculation

(1) Obtaining Sa, the first address of the area set as the source.

(a) Conditions

SA, the first address of the source area in the ca+se of full resizing: H'10000 (= 65536)

SH, the number of pixels across the source area in the case of full resizing: 120

ST, the number of lines to the top of the source area to be read out in the case of partial resizing: 15

SL, the number of pixels horizontally to the left edge of the source area (first point) to be read out in the case of partial resizing: 5

Number of bytes per pixel: 2

(b) Obtaining the pitch of rows in the source plane

SGP = ROUNDUP ((number of bytes for one pixel) × SH/64 bytes) × 64 bytes
= ROUNDUP ((2 bytes) × 120/64 bytes) × 64 bytes = 256 bytes

(c) Obtaining Sa, the first address of the area set as the source

Sa = SA + SGP × ST + SL × (number of bytes for one pixel)
=65536 + 256 bytes × 15 + 5 × 2 bytes = 69386 (= H'10F0A)

(2) Obtaining Da, the first address of the area set as the destination.

(a) Conditions

DA, the first address of the destination area in the case of full resizing: H'11000 (= 69632)

DH, the number of pixels across the destination area in the case of full resizing: 277

DT, the number of lines to the top of the destination area in the case of partial resizing: 33

DL, the number of pixels horizontally to the left edge of the destination area (first point) to be read out in the case of partial resizing: 15

Number of bytes per pixel: 2

(b) Obtaining the pitch of rows in the source plane

DGP = ROUNDUP ((number of bytes for one pixel) × DH/64 bytes) × 64 bytes
= ROUNDUP ((2 bytes) × 277/64 bytes) × 64 bytes = 576 bytes

(c) Obtaining Da, the first address of the area set as the destination

Da = DA + DGP × DT + DL × (number of bytes for one pixel)
=69632 + 576 bytes × 33 + 15 × 2 bytes = 888670 (= H'15A5E)

(6) Duplicate-Line Setting of Enlargement Resizing

When performing an enlargement resizing in the vertical direction, in some cases image data of the source image on the memory is used twice in succession by line. For this reason, the pixel data (in units of lines) of the lines used twice needs to be transferred by the CPU to the 2DG twice in succession.

(a) Subject conditions

Duplicate setting applies to the case where a source image, whose horizontal pixel count is larger than the bank size (64 pixels) of the buffer, is to be resized vertically for enlargement (the horizontal resizing ratio is not applicable). In other words, duplicate setting applies to the cases where

horizontal pixel count of source image ≥ 65 and vertical enlargement resizing.

Figures 26.45 and 26.46 illustrate examples.

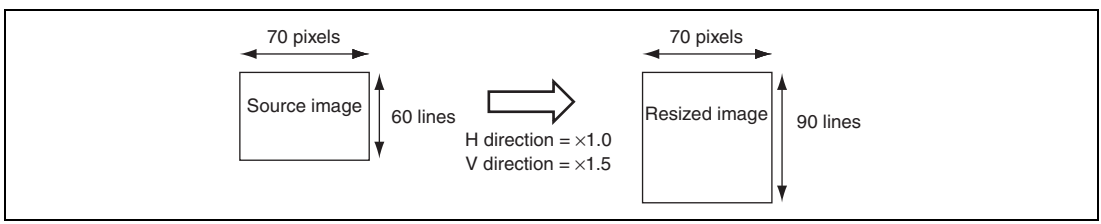


Figure 26.45 Example of Duplicate-line Setting of Enlargement Resizing (1)

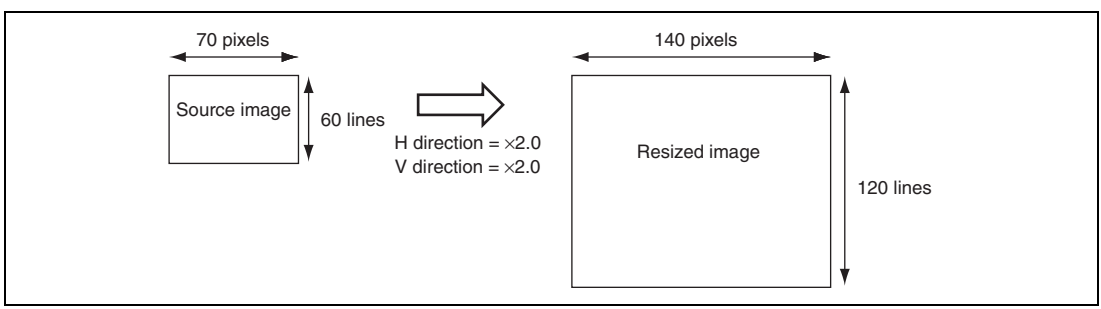


Figure 26.46 Example of Duplicate-line Setting of Enlargement Resizing (2)

(b) Determining a subject line (duplicate-line)

The line to be used twice in succession can be determined according to the following test formula:

[Test formulae]

b0 = INT ((VDLT_DCML × INT (SL × DV/SL) + V1PHS_DCML + (VDLT_DCML × (– 2)))/4096)

b1 = INT ((VDLT_DCML × INT (SL × DV/SL) + V1PHS_DCML + (VDLT_DCML × (– 1)))/4096)

b2 = INT ((VDLT_DCML × INT (SL × DV/SL) + V1PHS_DCML + (VDLT_DCML × (0)))/4096)

b3 = INT ((VDLT_DCML × INT (SL × DV/SL) + V1PHS_DCML + (VDLT_DCML × (1)))/4096)

IF (OR (AND (b1 = (SL – 1), b1 = b0),

AND (b1 = (SL – 1), b1 = b2),

AND (b2 = (SL – 1), b2 = b3)), 2, 1)

[Legend]

SV: Vertical line count of the source image (before resizing)

DV: Vertical line count of the image after resizing

SL: Line number of the source image (before resizing) to be determined

VIPHS_DCML: Fractional part of the initial phase in the vertical direction (allows for settings less than delta)

VDLT_DCML: Fractional part of delta in the vertical direction

Note: The line is transferred twice in succession if the computational result of the test formula is true (2); if it is false (1), the line is transferred once.

Table 26.10 Examples of Determining Subject Lines (duplicate-lines)

Example 1: Vertical enlargement ratio = 3/2 SV = 100, DV = 150, OFS = 0		Example 2: Vertical enlargement ratio = 5/4 SV = 80, DV = 100, OFS = 0		Example 3: Vertical enlargement ratio = 2/1 SV = 100, DV = 200, OFS = 0	
SL	Computational Result	SL	Computational Result	SL	Computational Result
0	1	0	1	0	1
1	2	1	2	1	2
2	2	2	1	2	2
3	1	3	1	3	2
4	2	4	2	4	2
5	1	5	1	5	2
6	2	6	1	6	2
:		:		:	:
:		:		:	:
78	2	:		:	:
:		79	1	:	:
:				:	:
99	1			99	2

Note: Since SL = 0 lines leads to exception handling, always set the transfer of a line at least once regardless of the multiplier.

An example shows to judge whether or not duplicate-line settings are required in the following case.

- Condition

  Number of source pixels (SV): 100

  Number of destination pixels (DV): 150

  Fractional part of the result of calculating phase of vertical starting position (V1PHS_DCML): H'500 = 1365

  Line number (SL): 78

- Items calculated by the user
  Fractional part of the result of calculating the vertical delta (VDLT_DCML)
  Cv is calculated for use in obtaining VDLT_DCML.
  Cv = INT (SV/DV) × 4096 = INT (100/150) × 4096) = 2730.
  Since VDLT_DCML becomes the lower-order 12 bits of Cv, VDLT_DCML = 2730.

When the line number (SL) is 78, use the formula given earlier to discern whether duplicate-line settings are or are not necessary.

b0 = INT (2730 × INT (78 × 150/100) + 1365 + (2730 × (–2)))/4096 = 76

b1 = INT (2730 × INT (78 × 150/100) + 1365 + (2730 × (–1)))/4096 = 77

b2 = INT (2730 × INT (78 × 150/100) + 1365 + (2730 × (0)))/4096 = 78

b3 = INT (2730 × INT (78 × 150/100) + 1365 + (2730 × (1)))/4096 = 78

```
IF (OR (AND (77 = (78 – 1), 77 = 76),
        AND (77 = (78 – 1), 77 = 78),
        AND (78 = (78 – 1), 78 = 78)), 2, 1)
```

Since all terms of the conditional expression are false, the transfer of line 78 is only to proceed once.

[Example of settings for pixel-data transfer]

As an example, the text below explains Example 1 in table 26.10.

In this case, make settings such that pixel data (in line units) is transferred from the CPU to the 2DG module in the following order.

Data for transfer (in line units) = 0, 1, 1, 2, 3, 3, 4, 5, 5, 6, 7, 7, 8, …

In this way, lines 1, 3, 5, 7 etc. of pixel data (in line units) must each be transferred twice consecutively.

[Obtaining the number of lines for transfer in the case of duplicate-line settings]

Regardless of whether resizing is full or partial, the number of lines for transfer in the case of duplicate-line settings is always the total obtained from the conditional expression for duplicate-lines. Examples for the cases of full or partial resizing are explained below. Furthermore, duplicate-line settings are made in the examples for the number of all horizontal pixels being 65 or more.

Example 1: Duplicate-line settings in the case of full resizing

- Conditions
  Number of source lines: 20
  Number of destination lines: 40
  Multiplication: 2 times

In this case, when the conditional expression given earlier is applied to all lines, the results are as shown in figure 26.47. The number of lines for transfer as the total obtained from the conditional expression for line duplication is thus 39.

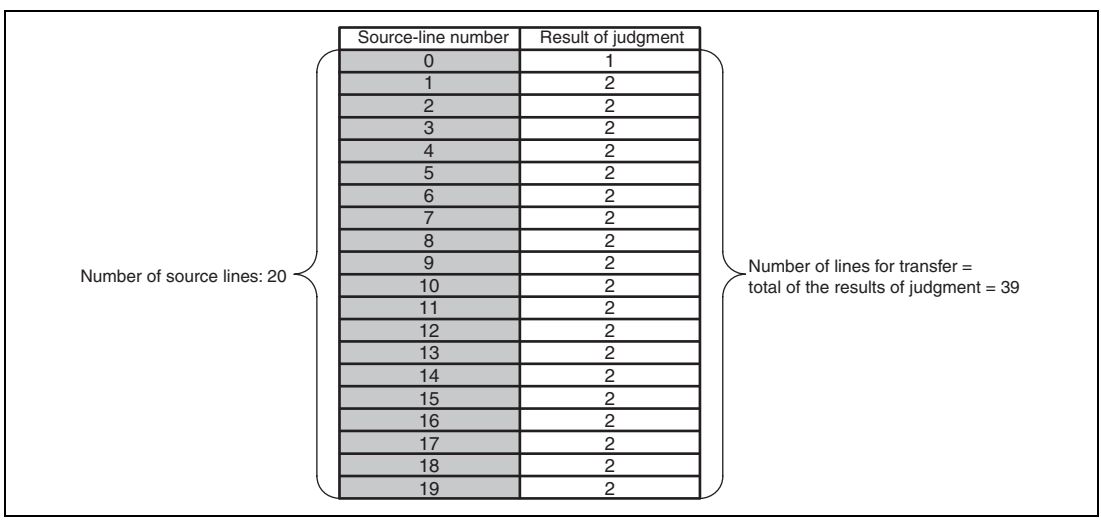


Figure 26.47 Duplicate-Line Settings for Full Resizing

Example 2: Duplicate-line settings in the case of partial resizing

- Conditions
  Number of source lines: 9
  Number of destination lines: 18
  Multiplication: 2 times
  Source starting position: Line 5

In this case, when the conditional expression given earlier is applied to all lines, the results are as shown in figure 26.48. The number of lines for transfer as the total obtained from the conditional expression for line duplication is thus 17.

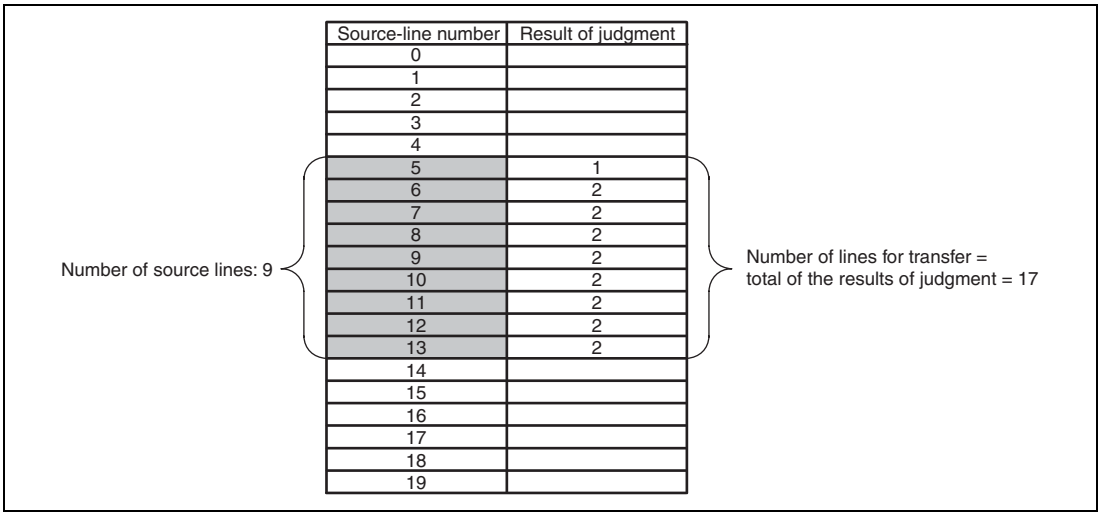
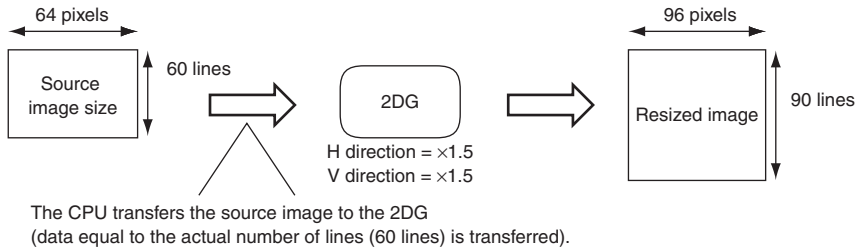


Figure 26.48 Duplicate-line Setting for partial resizing

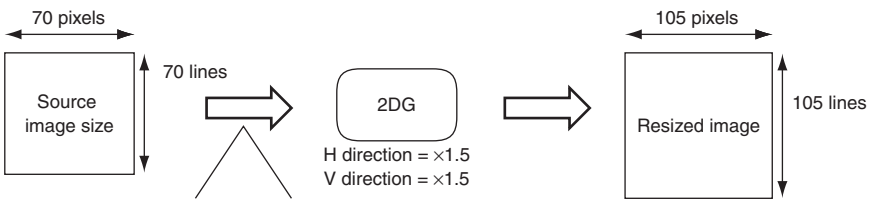
(c) How to set the register for duplicate-line setting

When lines are to be set in duplicate, the resize register in the 2DG must be set in the same manner as normal resizing settings. The source image size should be set in terms of the number of horizontal pixels and the number of lines that are actually transferred by the CPU to the 2DG. For the resize image size, set the number of horizontal pixels and the number of lines that are actually written to the memory by the 2DG. Examples are shown in figures 26.49 and 26.50.



- The size of image which is transferred to the 2DG by the CPU:
  64 pixels in the horizontal direction and 60 lines in the vertical direction
- The size of image which is written to memory by the 2DG:
  96 pixels in the horizontal direction and 90 lines in the vertical direction

Figure 26.49 Enlarging an image, both horizontally and vertically, when the horizontal image has 64 pixels or less (normal setting (not duplicate setting))



- The image size which is transferred to the 2DG by the CPU:
  70 pixels in the horizontal direction and 105 lines (line duplicate setting) in the vertical direction
- The image size which is written to memory by the 2DG:
  105 pixels in the horizontal direction and 105 lines in the vertical direction

Figure 26.50 Enlarging an image, both horizontally and vertically, when the horizontal image has 65 pixels or more (duplicate-line setting)

26.4.4 Output Operations

(1) Summary of Operations between the Output Block and External Memory

The following is a summary of operations between the output block and external memory.

1. The output block negates the DMA request signal and accepts a DMA transfer from the external memory.
2. The SE buffers (SE1, SE2), alternately buffer the data received through the DMA transfer.
3. Triggered by the VSYNC signal, the data undergoes various processings and then output to the panel unit.
4. Steps 1 to 3 are repeated until all data processing is completed.

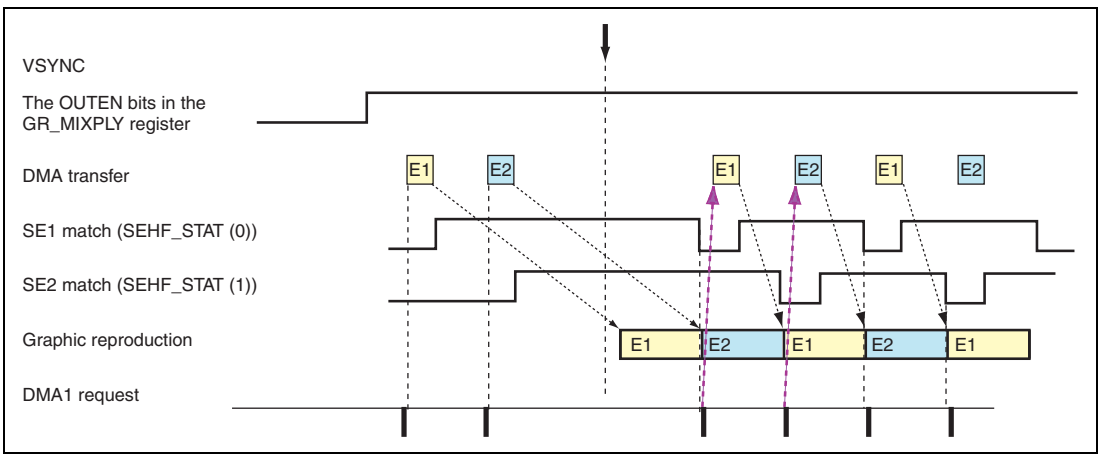


Figure 26.51 Summary of Operations between Output Block and External Memory

(2) Pixel Format Conversion in Output Block

For the output block, either αRGB444 or αRGB555 can be set as the pixel format. The output block uses 6 bits for each color in blending involving the moving pictures. For this reason, a given format is converted into a standard format: α (4 bits) + RGB (6 bits each) for a total of 22 bits. The formula below shows rules for the conversion of a given format to the standard format:

- αRGB444 (AF83 (H)) converted into a standard format
  α: A (H) → A (H) R: F (H) → 3C (H) G: 8 (H) → 20 (H) B: 3 (H) → 0C (H)

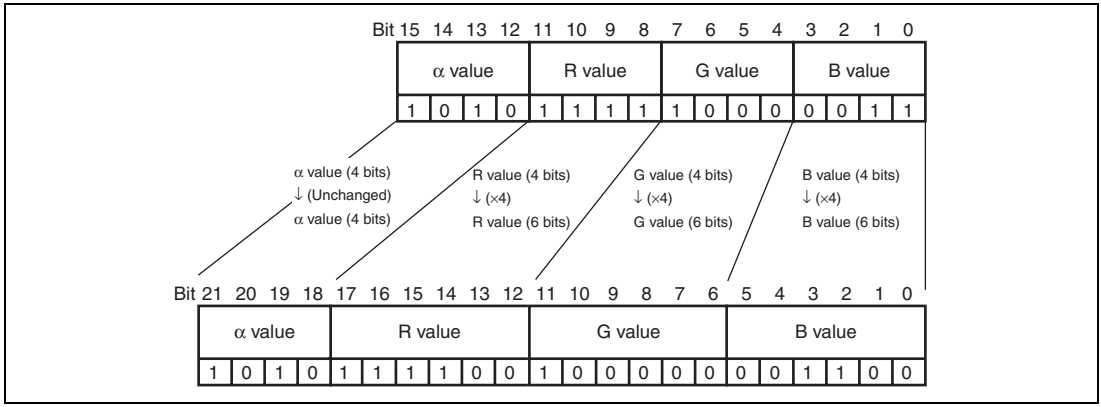


Figure 26.52 Pixel Format Conversion in Output Block (1)

- αRGB555 (F599 (H)) converted into a standard format
  α: 1 (H) → CHG_A bit in register MGR_MIXMODE
  R: 1D (H) → 3A (H) G: 0C (H) → 18 (H) B: 19 (H) → 32 (H)

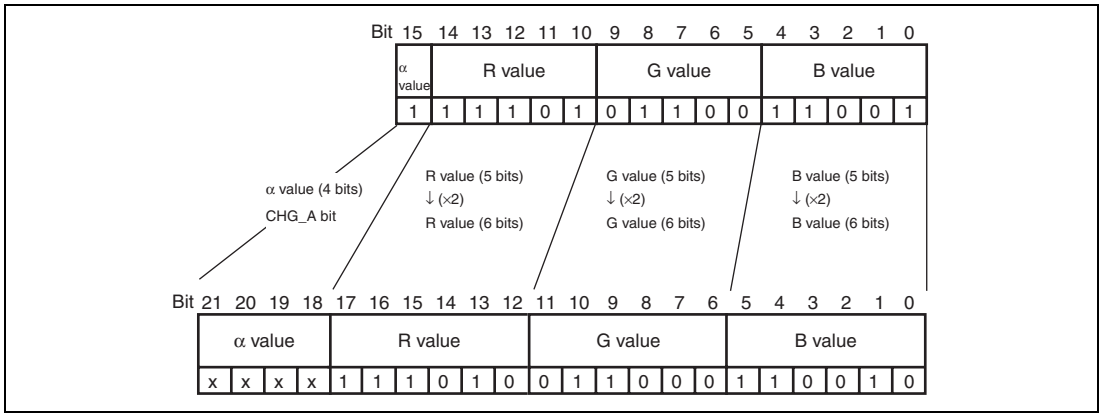


Figure 26.53 Pixel Format Conversion in Output Block (2)

(3) Summary of Output Block Operations

As a summary of output block operations, the text below provides an example where data equal to the specified number of pixels is transferred from output plane PX, which is written in an arbitrary memory space on the SDRAM, to the SE buffer using the DMAC, and the data is blended with moving pictures before being output.

The area for memory plane PX is set as follows: the number of lines in the SEHIGH bits of the MGR_SESET register, and the number of pixels in the SEWIDH bits of the GR_SESET register.

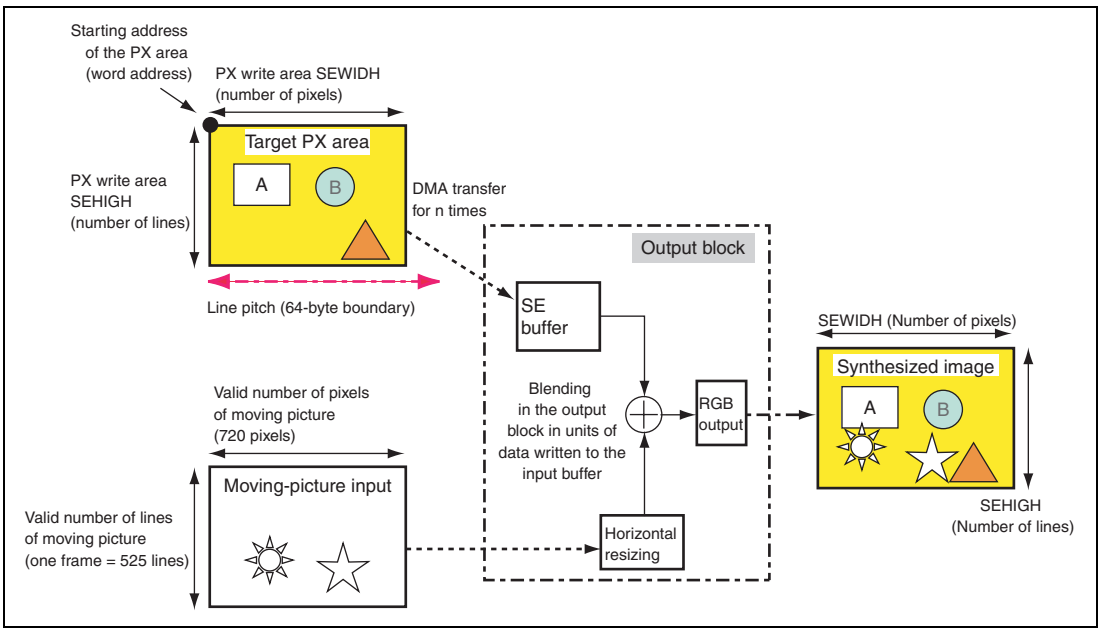


Figure 26.54 Summary of Output Block Operations

The SE buffer has a 960-byte double-buffer structure (SE1, SE2). For example, if the following values are assigned: SEWIDH bits = 480 (pixels) in the MGR_SESET register and SEHIGH bits = 240 (lines) in the MGR_SESET register, and if the OUTEN bit = 1 in the GR_MIXPLY register to enable graphics display, output block operations work as follows:

1. Transfers the first 480 pixels to SE1 (SEHF_STAT (0) = 1), followed by VIVSYNC input, constant-rate output processing, and output to the panel unit.
2. Transfers the next 480 pixels to SE2 (SEHF_STAT (1) = 1), followed by VIVSYNC input, constant-rate output processing, and output to the panel unit.

3. Repeats transferring data to SE1 and SE2 up to the 240th line, and outputs interrupt signal INT_FILD at the 240th line.
4. The above steps 1 to 3 are repeated until the OUTEN bit of the GR_MIXPLY register is updated to 0.

(4) Resizing

(a) How to set the delta value for use in resizing

The value of a resize delta (Ch) can be determined according to the following equation:

$$Ch = (\text{source resolution}/\text{destination resolution}) \times 4096$$

Note: Ch = 1/resizing ratio must be calculated from source pixel count/destination pixel count.

For example, if the source pixel count is 720 and the destination pixel count is 480, Ch will be:

$$Ch = (720 / 480) \times 4096 = 6144 = 1800\ (H)$$

Based on these results, the resize delta value will be as follows:

Since MHDLT_INTGR is the integer part of Ch (4 bits), MHDLT_INTGR = 1 (H)
Since MHDLT_DCML is the fractional part of Ch (12 bits), MHDLT_DCML = 800 (H)
If the resizing function is not to be used, set the MHDLT_INTGR bit and MHDLT_DCML bit to H'1 and H'000, respectively.

(b) Howe to set the source-side starting phase for use in resizing

The source-side starting phase (Psh) can be determined according to the following equation:

$$Psh = Ch \times (\text{starting pixel count}) + (\text{starting initial phase} \times 4096)$$

In this case, MH1PHS_DCML will be the fractional part (12 bits) of Psh. If resizing is not performed or a resizing ratio = 1, Psh = 0 should be set. Note that the starting pixel count is always 0 (first pixel) for the output block, since the output block exclusively resizes moving pictures. Therefore, only the fractional part is necessary for Psh unlike for the blitter.

(5) Blending in Output Block

The following describes the blending in the output block. The blending processor in the output block supplies the output according to the following formula.

$$Cp = (Fc \times Cdc) + (Fd \times Cv)$$

Here, Fc and Fd are set using the FCFD bits of the MGR_MIXMODE register. Table 26.11 shows the FCFD bit settings and the corresponding Fc and Fd values.

Table 26.11 Details of FCFD Bits of MGR_MIXMODE Register

FCFD (Bit Values)	Fc	Fd	Remarks
000 (Initial value)	1	1 – αdc	When SE input image is premultiplied.
001	αdc	1 – αdc	When SE input image is non-premultiplied.
010	1	0	Only graphics are output.
011	0	1	Only moving pictures are output.
100	0	0	Nothing is output. (Black screen)
Other than the above	—	—	Reserved

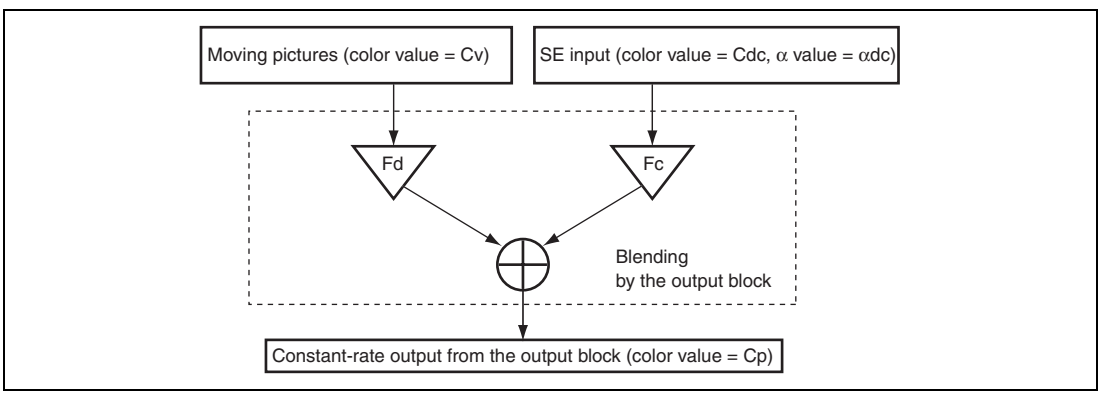


Figure 26.55 Summary of Output Block Operations during Blending

26.4.5 Interrupts

2DG interrupt signals are classified into two types: the interrupts related to the blitter (BLT interrupts) and the interrupts related to the output block (output interrupts). Table 26.12 shows the interrupt sources and the conditions on which each source is generated and can be cleared. Figure 26.56 shows the structure of the interrupts.

Table 26.12 Interrupt Sources and Corresponding Generation/Clearing Conditions

Interrupt Sources	Status Bit	Generation Conditions	Clearing Conditions
BLT interrupts	DC buffer full flag (IRQ_DHFUL bit)	The DC buffer becomes full.	This bit is cleared when 1 is written to the DIS_DHFUL bit in GR_INTDIS.
	SA buffer full flag (IRQ_ASHFUL bit)	The SA buffer becomes full.	This bit is cleared when 1 is written to the DIS_ASHFUL bit in GR_INTDIS.
	SB buffer full flag (IRQ_SHFUL bit)	The SB buffer becomes full.	This bit is cleared when 1 is written to the DIS_SHFUL bit in GR_INTDIS.
	Blit operation completed (INT_GR bit)	Blitter operation is completed.	This bit is cleared when 1 is written to the DIS_GR bit in GR_INTDIS.
Output interrupts	SE buffer full flag (IRQ_DEMPT bit)	The SE buffer becomes full.	This bit is cleared when 1 is written to the DIS_DEMPT bit in GR_INTDIS.
	VSYNC input for output block (INT_VSYC bit)	VSYNC input is supplied. (when display is on.)	This bit is cleared when 1 is written to the DIS_VSYC bit in GR_INTDIS.
	Output underflow for output block (INT_UDFL bit)	Underflow of output from the output block occurs.	This bit is cleared when 1 is written to the DIS_UDFL bit in GR_INTDIS.
	Last line captured by output block (INT_FILD bit)	The last line is captured in the SE buffer.	This bit is cleared when 1 is written to the DIS_FILD bit in GR_INTDIS.

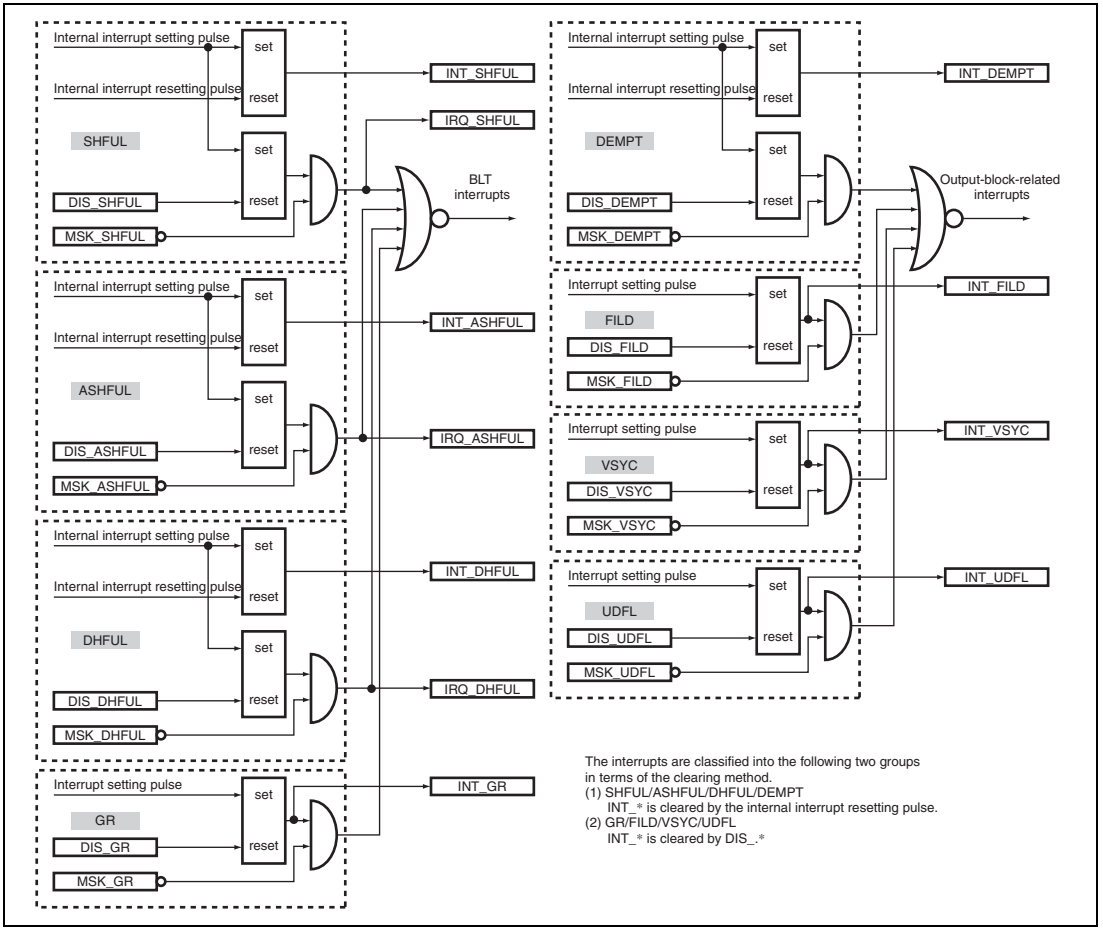


Figure 26.56 Structure of Interrupts

The 2DG handles the interrupts differently for the following two cases.

(1) When the interrupt source is a DC buffer full, an SA buffer full, an SB buffer full, or an SE buffer full

(1-1) An interrupt event occurs in the 2DG.

(1-2) The INT_*** and IRQ_*** bits in the interrupt status register for graphics (GR_IRSTAT) are set accordingly (interrupt signal = negative logic).

(1-3) The CPU recognizes the interrupt and reads the interrupt status register for graphics.

(1-4) The CPU writes 1 to the interrupt reset control register for graphics (GR_INTDIS).

(1-5) On reception of the value written in the above step, the IRQ_*** bit in the GR_IRSTAT register is cleared (thus deasserting the interrupt signal).

(1-6) Resetting of the response to the interrupt event proceeds within the 2DG.

(1-7) The INT_*** bit in the GR_IRSTAT register is cleared in response to the above step.

Figure 26.57 shows the flow of processing when SB buffer full.

Note that, within the interrupt status register for graphics (GR_IRSTAT), the CPU writing a 1 to a bit of the interrupt reset control register for graphics (in step (1-4) above) clears the bit in the case of IRQ_*** bits but does not clear the bit in the case of INT_*** bits.

If an interrupt event occurs and is the response within the 2DG is cleared (step (1-6) above) before the CPU has read the GR_ISTAT register (step (1-3) above), reading the value of the GR_ISTAT register will return the value of the register with the corresponding INT_*** bit cleared.

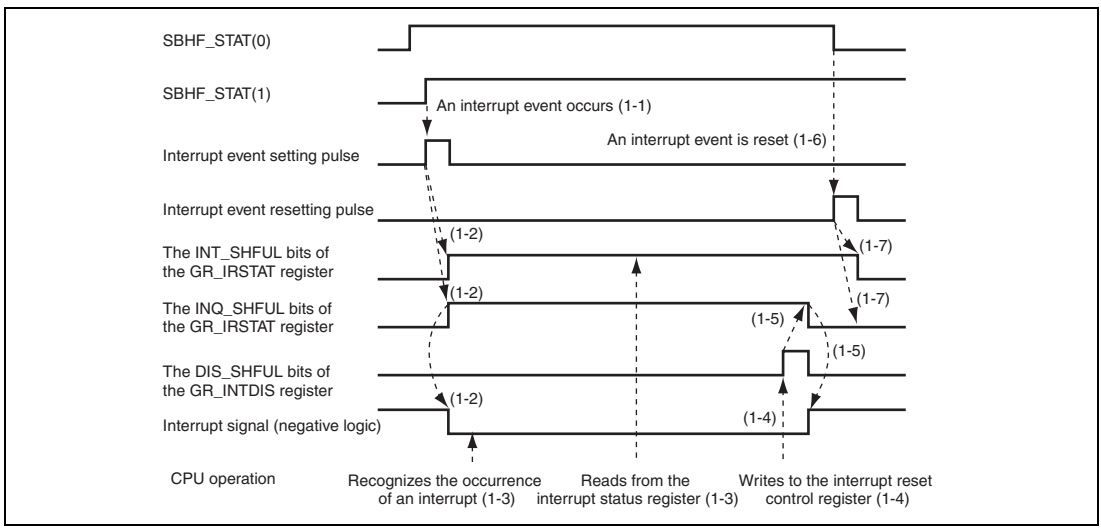


Figure 26.57 Interrupt Handling (1)

(2) When the interrupt source is a completion of a blit operation, input of a VSYNC signal for the output block, and output underflow for the output block, or capture of a last line by the output block

(2-1) An interrupt event occurs in the 2DG.

(2-2) The INT_*** bit in the interrupt status register for graphics (GR_IRSTST) is set accordingly (interrupt signal = negative logic).

(2-3) The CPU recognizes the interrupt and reads the GR_IRSTAT register.

(2-4) The CPU writes 1 to the interrupt reset control register for graphics (GR_INTDIS).

(2-5) The IRQ_*** bit in the GR_IRSTAT register is cleared in response to the above step (thus deasserting the interrupt signal).

Figure 26.58 shows the interrupt processing flow.

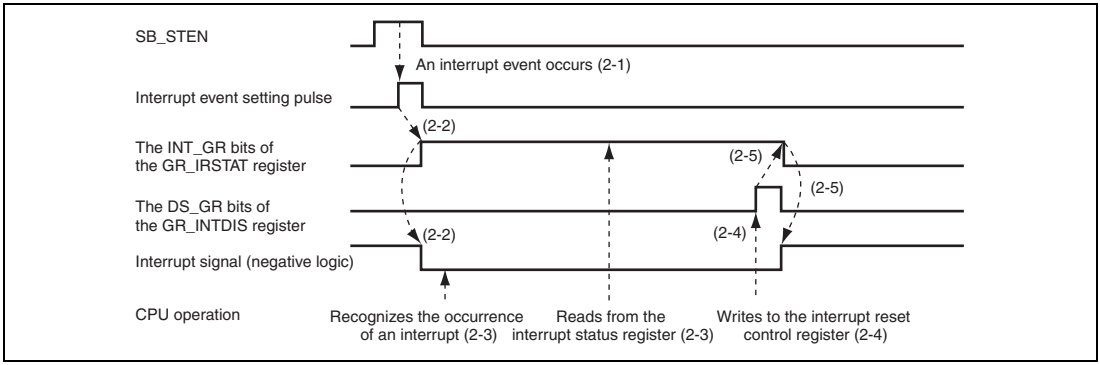


Figure 26.58 Interrupt Handling (2)

The 2DG interrupt signals are level-sensitive and more than one status bit is assigned to each interrupt signal. Because of this, it is necessary for the CPU to read the interrupt status register until the relevant interrupt signal is reset so that the CPU should recognize all the corresponding status bits and handle them according to the specified priority.

26.5 Appendix. VIDEO OUT (D/A Converter)

VIDEO OUT contains a built-in D/A converter (DAC) of the current output method using a current cell matrix D/A converter; it outputs a current corresponding to 6-bit digital input signals to analog output pins R, G, and B.

26.5.1 Analog Output Current

(1) Determination of the DAC Output Resistor (RL)

Since this DAC is a current output type unit, in order to convert a current into voltage, it requires a resistor (RL) external to the LSI. The RL can be set between 75 and 180 Ω based on an allowable current level, the external load capacity (CL), and a desired settling time (tset). The formula for calculating RL is given below:

Here, we define the time required from the beginning of a change in output to the time it converges to the level within ±1.0% of the final attainable level as settling time.

$tset = 4.6 \times RL \times C$ tset: ±1% settling time [ns], C: total load capacity [pF]

$C = Cin + CL$ Cin: internal capacity of module (approximately 5 pF) [pF],
CL: I/O + PKG + on-board capacity [pF]

The condition that must be satisfied by RL in order to obtain the desired settling time will be:

$RL < tset / \{4.6 \times (Cin + CL)\}$

Example: (if tset = 18 [ns], CL = 15 [pF])

$RL < 18ns / \{4.6 \times (5\ pF + 15\ pF)\} = 195.6\ \Omega$

Therefore, to settle it at 18 ns, set RL to 195.6 Ω or less (for example, 180 Ω).

(2) Determination of the DAC Output Current (I/O Max.)

This DAC is designed for a maximum output voltage of 1.0 V. Therefore, I/O max. can be calculated once RL is determined.

I/O max. = 1.0 / RL (if RL = 180 Ω, I/O max. = 5.6 mA)

Note: I/O max. should be designed for 13.4 mA or less. Exceeding this limit can cause decreases in performance and reliability.

(3) Determination of the Rext Resistance

As illustrated in figure 26.59, internally in this module the current cells are driven by means of a circuit that uses op amp.

When VCCA = 3.3 V, the "+" pin of the op amp is set so that it will be approximately 0.91 V. Since negative feedback is applied to the op amp, approximately 0.91 V also appears on the REXT pin.

Since each current cell is designed to act as a current mirror with respect to a load circuit for the op amp, reducing the external resistance Rext increases the output current from each current cell, and increases the output current from analog outputs R, G, and B. For full-scale output, the relationship between the output current I/O max. and the Rext resistance is given by the following equation:

Rext = {VCCA × (4 / 15) + 0.03} × (1023 / 32) / I/O max.

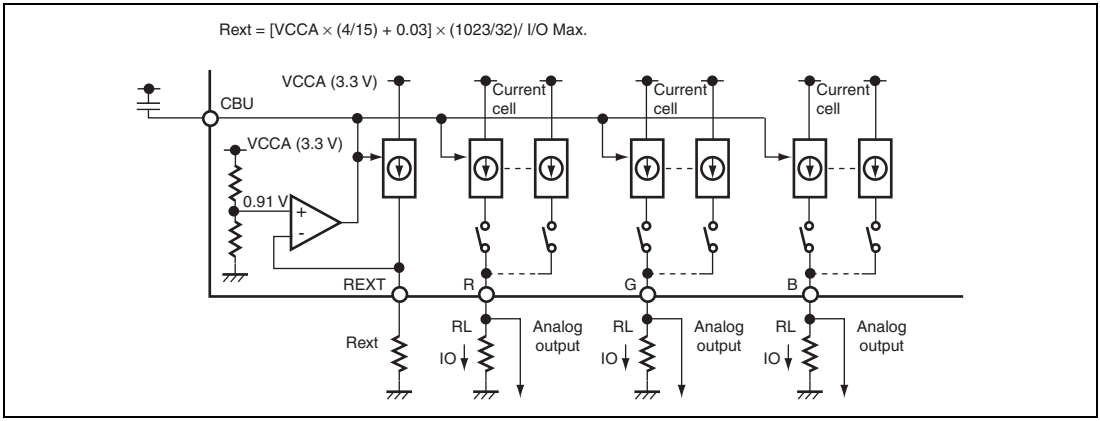


Figure 26.59 Current Cells and Analog Outputs

This module has a maximum analog output voltage of 1.0 V, as shown in figure 26.60.

The analog output voltage should be adjusted in terms of the value of Rext, to a level that fits in the range shown in figure 26.60. Exceeding this range can cause decreases in precision and reliability.

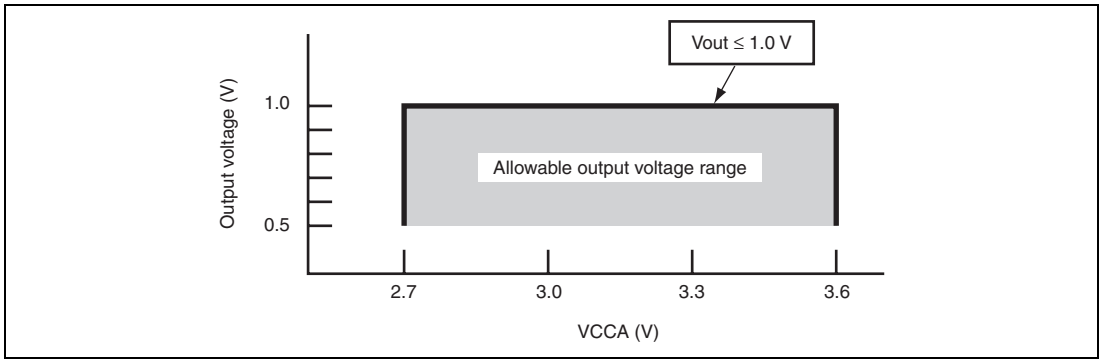


Figure 26.60 Allowable Analog Ouptut Voltage Range

26.5.2 Notes on Usage

(1) Power Supply Pins

Because the D/A output voltage is a micro-voltage with approximately 1 mV per step, any noise penetrating from the outside of the LSI must be minimized. For this purpose, a ceramic capacitor approximately 0.01 to 0.1 μF must be placed between each power supply pin and the VSS as close to the LSI pins as possible. It is recommended to place at least one 10-μF capacitor between each power supply and VSSA (figure 26.61).

(2) Power Supply Separation

In order to supply as a noise-free voltage as possible to the analog power supply, the system's analog power should be supplied to this DAC's power supply (figure 26.61). Also, VSSA should be connected to the system's analog ground.

(3) CBU Pin

The CBU pin is used to connect the phase compensation capacitor for op amps. A capacitor should be connected between the CBU pin and VCCA. No other elements or circuits should be connected to CBU pin.

(4) REXT Pin

The REXT pin serves to connect an external resistance element that determines the module's reference current. A resistance element appropriate for the RL should be connected between the REXT pin and VSSA. Notice that any noise on this pin can significantly affect D/A conversion results.

When reviewing pin layout or designing the board, exercise care that the REXT pin does not cross, or run parallel to, DAC output or other digital signals.

26.5.3 Application Example

Figure 26.61 shows a DAC application example.

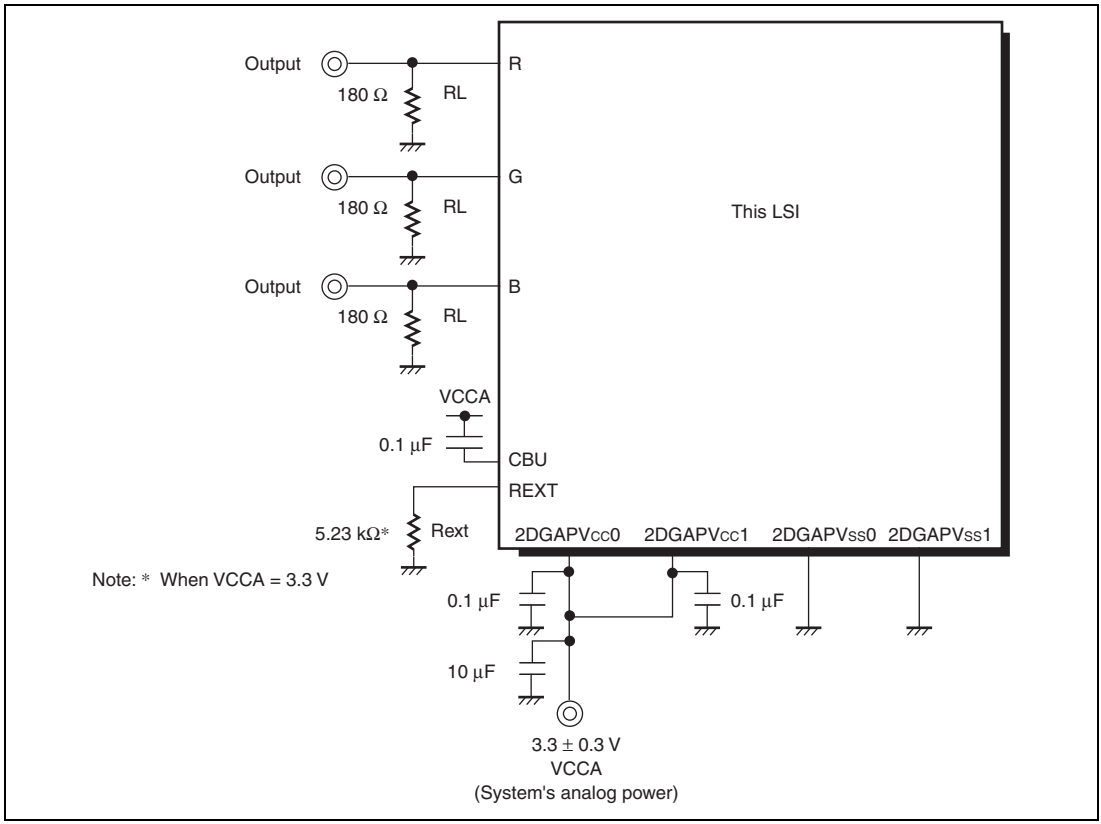


Figure 26.61 Application Example of DAC

Section 27 Pin Function Controller (PFC)

The pin function controller (PFC) is composed of registers that are used to select the functions of multiplexed pins and assign pins to be inputs or outputs. Tables 27.1 to 27.10 list the multiplexed pins of this LSI.

Table 27.1 Multiplexed Pins (Port A)

	Setting of Mode Bits (PAnMD[3:0])			
	0000	**0001**	**0010**	**0011**
Setting Register	**Function 1 (General I/O)**	**Function 2 (Related Module)**	**Function 3 (Related Module)**	**Function 4 (Related Module)**
PACRL4	PA15 I/O (port)	D31 I/O (data)	IDED15 I/O (ATAPI)	$\overline{\text{ADTRG}}$ input (analog)
	PA14 I/O (port)	D30 I/O (data)	IDED14 I/O (ATAPI)	—
	PA13 I/O (port)	D29 I/O (data)	IDED13 I/O (ATAPI)	$\overline{\text{TEND1}}$ output (DMAC)
	PA12 I/O (port)	D28 I/O (data)	IDED12 I/O (ATAPI)	$\overline{\text{DACK1}}$ output (DMAC)
PACRL3	PA11 I/O (port)	D27 I/O (data)	IDED11 I/O (ATAPI)	DREQ1 input (DMAC)
	PA10 I/O (port)	D26 I/O (data)	IDED10 I/O (ATAPI)	$\overline{\text{TEND0}}$ output (DMAC)
	PA9 I/O (port)	D25 I/O (data)	IDED9 I/O (ATAPI)	$\overline{\text{DACK0}}$ output (DMAC)
	PA8 I/O (port)	D24 I/O (data)	IDED8 I/O (ATAPI)	DREQ0 input (DMAC)
PACRL2	PA7 I/O (port)	D23 I/O (data)	IDED7 I/O (ATAPI)	TCLKD input (MTU2)
	PA6 I/O (port)	D22 I/O (data)	IDED6 I/O (ATAPI)	TCLKC input (MTU2)
	PA5 I/O (port)	D21 I/O (data)	IDED5 I/O (ATAPI)	TCLKB input (MTU2)
	PA4 I/O (port)	D20 I/O (data)	IDED4 I/O (ATAPI)	TCLKA input (MTU2)
PACRL1	PA3 I/O (port)	D19 I/O (data)	IDED3 I/O (ATAPI)	—
	PA2 I/O (port)	D18 I/O (data)	IDED2 I/O (ATAPI)	—
	PA1 I/O (port)	D17 I/O (data)	IDED1 I/O (ATAPI)	$\overline{\text{DACK2}}$ output (DMAC)
	PA0 I/O (port)	D16 I/O (data)	IDED0 I/O (ATAPI)	DREQ2 input (DMAC)

	Setting of Mode Bits (PAnMD[3:0])			
	0100	**0101**	**0110**	**0111**
Setting Register	**Function 5 (Related Module)**	**Function 6 (Related Module)**	**Function 7 (Related Module)**	**Function 8 (Related Module)**
PACRL4	TIOC4D I/O (MTU2)	PINT7 input (INTC)	—	—
	TIOC4C I/O (MTU2)	PINT6 input (INTC)	—	—
	TIOC4B I/O (MTU2)	PINT5 input (INTC)	—	—
	TIOC4A I/O (MTU2)	PINT4 input (INTC)	—	$\overline{\text{DACT1}}$ output (DMAC)
PACRL3	TIOC3D I/O (MTU2)	PINT3 input (INTC)	—	—
	TIOC3C I/O (MTU2)	PINT2 input (INTC)	—	—
	TIOC3B I/O (MTU2)	PINT1 input (INTC)	—	$\overline{\text{DACT0}}$ output (DMAC)
	TIOC3A I/O (MTU2)	PINT0 input (INTC)	—	—
PACRL2	TIOC2D I/O (MTU2)	IRQ7 input (INTC)	$\overline{\text{SCS1}}$ I/O (SSU)	—
	TIOC2C I/O (MTU2)	IRQ6 input (INTC)	SSO1 I/O (SSU)	—
	TIOC1B I/O (MTU2)	IRQ5 input (INTC)	SSI1 I/O (SSU)	—
	TIOC1A I/O (MTU2)	IRQ4 input (INTC)	SSCK1 I/O (SSU)	—
PACRL1	TIOC0D I/O (MTU2)	IRQ3 input (INTC)	$\overline{\text{SCS0}}$ I/O (SSU)	—
	TIOC0C I/O (MTU2)	IRQ2 input (INTC)	SSO0 I/O (SSU)	—
	TIOC0B I/O (MTU2)	IRQ1 input (INTC)	SSI0 I/O (SSU)	$\overline{\text{DACK2}}$ output (DMAC)
	TIOC0A I/O (MTU2)	IRQ0 input (INTC)	SSCK0 I/O (SSU)	—

Table 27.2 Multiplexed Pins (Port B)

	Setting of Mode Bits (PBnMD[3:0])			
	0000	0001	0010	0011
Setting Register	Function 1 (General I/O)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
PBCRH2	PB18 I/O (port)	$\overline{\text{WDTOVF}}$ output (WDT)	$\overline{\text{UBCTRG}}$ output (UBC)	—
PBCRH1	PB17 I/O (port)	$\overline{\text{WAIT}}$ input (BSC)	—	—
	PB16 I/O (port)	$\overline{\text{SDWE}}$ output (BSC)	—	—
PBCRL4	PB15 I/O (port)	CKE output (BSC)	—	—
	PB14 I/O (port)	$\overline{\text{CAS}}$ output (BSC)	—	—
	PB13 I/O (port)	$\overline{\text{RAS}}$ output (BSC)	—	—
	PB12 I/O (port)	$\overline{\text{WE3}}$/$\overline{\text{BC3}}$/DQM3 output (BSC)	IDECS#1 output (ATAPI)	FCDE output (FLCTL)
PBCRL3	PB11 I/O (port)	$\overline{\text{WE2}}$/$\overline{\text{BC2}}$/DQM2 output (BSC)	IDECS#0 output (ATAPI)	$\overline{\text{FWE}}$ output (FLCTL)
	PB10 I/O (port)	$\overline{\text{WE1}}$/$\overline{\text{BC1}}$/DQM1 output (BSC)	—	—
	PB9 I/O (port)	$\overline{\text{WE0}}$/$\overline{\text{BC0}}$/DQM0 output (BSC)	—	—
	PB8 I/O (port)	$\overline{\text{CS5}}$ output (BSC)	$\overline{\text{SDCS1}}$ output (BSC)	$\overline{\text{MRES}}$ input (system control)
PBCRL2	PB7 I/O (port)	$\overline{\text{CS4}}$ output (BSC)	$\overline{\text{SDCS0}}$ output (BSC)	—
	PB6 I/O (port)	$\overline{\text{CS3}}$ output (BSC)	RD_$\overline{\text{WR}}$/$\overline{\text{WE}}$ output (BSC)	—
	PB5 I/O (port)	$\overline{\text{CS2}}$ output (BSC)	—	—
	PB4 I/O (port)	$\overline{\text{CS1}}$ output (BSC)	—	—
PBCRL1	PB3 I/O (port)	A22 output (address)	TxD2 output (SCIF)	—
	PB2 I/O (port)	A21 output (address)	RxD2 input (SCIF)	—
	PB1 I/O (port)	A1 output (address)	—	—
	PB0 I/O (port)	A0 output (address)	RD_$\overline{\text{WR}}$/$\overline{\text{WE}}$ output (BSC)	—

Table 27.3 Multiplexed Pins (Port C)

	Setting of Mode Bits (PCnMD[3:0])					
	0000	0001	0010	0011	0100	0101
Setting Register	Function 1 (General I/O)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)
PCCRL3	PC10 I/O (port)	DIRECTION output (ATAPI)	$\overline{\text{TEND1}}$ output (DMAC)	$\overline{\text{FCE}}$ output (FLCTL)	PINT6 input (INTC)	—
	PC9 I/O (port)	IDERST# output (ATAPI)	$\overline{\text{DACK1}}$ output (DMAC)	NAF7 I/O (FLCTL)	PINT5 input (INTC)	$\overline{\text{DACT1}}$ output (DMAC)
	PC8 I/O (port)	IDEINT input (ATAPI)	DREQ1 input (DMAC)	NAF6 I/O (FLCTL)	PINT4 input (INTC)	—
PCCRL2	PC7 I/O (port)	IDEIORDY input (ATAPI)	TIOC4D I/O (MTU2)	NAF5 I/O (FLCTL)	PINT3 input (INTC)	—
	PC6 I/O (port)	IDEIORD# output (ATAPI)	TIOC4C I/O (MTU2)	NAF4 I/O (FLCTL)	PINT2 input (INTC)	—
	PC5 I/O (port)	IDEIOWR# output (ATAPI)	TIOC4B I/O (MTU2)	NAF3 I/O (FLCTL)	PINT1 input (INTC)	—
	PC4 I/O (port)	IODREQ input (ATAPI)	TIOC4A I/O (MTU2)	NAF2 I/O (FLCTL)	PINT0 input (INTC)	—
PCCRL1	PC3 I/O (port)	IODACK# output (ATAPI)	TCLKD input (MTU2)	NAF1 I/O (FLCTL)	IRQ3 input (INTC)	—
	PC2 I/O (port)	IDEA2 output (ATAPI)	TCLKC input (MTU2)	NAF0 I/O (FLCTL)	IRQ2 input (INTC)	—
	PC1 I/O (port)	IDEA1 output (ATAPI)	TCLKB input (MTU2)	FSC output (FLCTL)	IRQ1 input (INTC)	—
	PC0 I/O (port)	IDEA0 output (ATAPI)	TCLKA input (MTU2)	FOE output (FLCTL)	IRQ0 input (INTC)	—

Table 27.4 Multiplexed Pins (Port D)

	Setting of Mode Bits (PDnMD[3:0])				
	0000	0001	0010	0011	0100
Setting Register	Function 1 (General I/O)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)
PDCRL1	PD2 I/O (port)	$\overline{\text{TEND0}}$ output (DMAC)	A23 output (address)	SCK2 I/O (SCIF)	IRQ6 input (INTC)
	PD1 I/O (port)	$\overline{\text{DACK0}}$ output (DMAC)	A24 output (address)	$\overline{\text{DACT0}}$ output (DMAC)	IRQ5 input (INTC)
	PD0 /O (port)	DREQ0 input (DMAC)	A25 output (address)	$\overline{\text{ADTRG}}$ input (ADC)	IRQ4 input (INTC)

Table 27.5 Multiplexed Pins (Port E)

	Setting of Mode Bits (PEnMD[3:0])				
	0000	0001	0010	0011	0100
Setting Register	Function 1 (General I/O)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)
PECRL4	PE13 I/O (port)	TxD4 output (SCIF)	SDA2 I/O (IIC3)	—	—
	PE12 input (port)	RxD4 input (SCIF)	SCL2 I/O (IIC3)	—	—
PECRL3	PE11 I/O (port)	TxD3 output (SCIF)	SDA1 I/O (IIC3)	—	—
	PE10 input (port)	RxD3 input (SCIF)	SCL1 I/O (IIC3)	—	—
	PE9 I/O (port)	TxD2 output (SCIF)	SDA0 I/O (IIC3)	—	—
	PE8 input (port)	RxD2 input (SCIF)	SCL0 I/O (IIC3)	—	—
PECRL2	PE7 I/O (port)	SCK1 I/O (SCIF)	$\overline{\text{SCS1}}$ I/O (SSU)	—	SSIDATA5 output (SSIF)
	PE6 I/O (port)	TxD1 output (SCIF)	SSO1 I/O (SSU)	—	SSIWS5 I/O (SSIF)
	PE5 I/O (port)	RxD1 input (SCIF)	SSI1 I/O (SSU)	—	SSISCK5 I/O (SSIF)
	PE4 I/O (port)	SCK0 I/O (SCIF)	SSCK1 I/O (SSU)	—	—
PECRL1	PE3 I/O (port)	$\overline{\text{RTS0}}$ I/O (SCIF)	$\overline{\text{SCS0}}$ I/O (SSU)	—	TIOC2B I/O (MTU2)
	PE2 I/O (port)	$\overline{\text{CTS0}}$ I/O (SCIF)	SSO0 I/O (SSU)	—	TIOC2A I/O (MTU2)
	PE1 I/O (port)	TxD0 output (SCIF)	SSI0 I/O (SSU)	—	—
	PE0 I/O (port)	RxD0 input (SCIF)	SSCK0 I/O (SSU)	—	—

Table 27.6 Multiplexed Pins (Port F)

	Setting of Mode Bits (PFnMD[3:0])					
	0000	0001	0010	0011	0100	0101
Setting Register	Function 1 (General I/O)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)
PFCRL2	PF4 I/O (port)	—	$\overline{\text{DACK3}}$ output (DMAC)	CTx1 output (RCAN-TL1)	CTx0&CTx1 output (RCAN-TL1)	$\overline{\text{DACT3}}$ output (DMAC)
PFCRL1	PF3 I/O (port)	—	DREQ3 input (DMAC)	CRx1 input (RCAN-TL1)	CRx0/CRx1 input (RCAN-TL1)	—
	PF2 I/O (port)	—	—	CTx0 output (RCAN-TL1)	—	—
	PF1 I/O (port)	—	SDA3 I/O (IIC3)	CRx0 input (RCAN-TL1)	—	—
	PF0 I/O (port)	—	SCL3 I/O (IIC3)	—	—	—

Table 27.7 Multiplexed Pins (Port G)

	Setting of Mode Bits (PGnMD[3:0])				
	0000	0001	0010	0011	0100
Setting Register	Function 1 (General I/O)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)
PGCRL2	PG7 input (port)	VIHSYNC input (Video-In)	AN7 input (ADC)	DA1 output (DAC)	—
	PG6 input (port)	VIVSYNC input (Video-In)	AN6 input (ADC)	DA0 output (DAC)	—
	PG5 input (port)	—	AN5 input (ADC)	—	—
	PG4 input (port)	VICLKENB input (Video-In)	AN4 input (ADC)	—	—
PGCRL1	PG3 input (port)	IRQ3 input (INTC)	AN3 input (ADC)	—	TCLKD input (MTU2)
	PG2 input (port)	IRQ2 input (INTC)	AN2 input (ADC)	—	TCLKC input (MTU2)
	PG1 input (port)	IRQ1 input (INTC)	AN1 input (ADC)	—	TCLKB input (MTU2)
	PG0 input (port)	IRQ0 input (INTC)	AN0 input (ADC)	FRB input (FLCTL)	TCLKA input (MTU2)

Table 27.8 Multiplexed Pins (Port H)

	Setting of Mode Bits (PHnMD[3:0])				
	0000	0001	0010	0011	0100
Setting Register	Function 1 (General I/O)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)
PHCRL4	PH15 I/O (port)	AUDIO_CLK input (SSIF)	—	—	—
	PH14 I/O (port)	SSIDATA4 I/O (SSIF)	SCK5 I/O (SCIF)	—	NAF5 I/O (FLCTL)
	PH13 I/O (port)	SSIWS4 I/O (SSIF)	TxD5 output (SCIF)	—	NAF4 I/O (FLCTL)
	PH12 I/O (port)	SSISCK4 I/O (SSIF)	RxD5 input (SCIF)	—	NAF3 I/O (FLCTL)
PHCRL3	PH11 I/O (port)	SSIDATA3 I/O (SSIF)	—	—	NAF2 I/O (FLCTL)
	PH10 I/O (port)	SSIWS3 I/O (SSIF)	—	—	NAF1 I/O (FLCTL)
	PH9 I/O (port)	SSISCK3 I/O (SSIF)	—	—	NAF0 I/O (FLCTL)
	PH8 I/O (port)	SSIDATA2 I/O (SSIF)	—	—	—
PHCRL2	PH7 I/O (port)	SSIWS2 I/O (SSIF)	—	—	—
	PH6 I/O (port)	SSISCK2 I/O (SSIF)	—	—	—
	PH5 I/O (port)	SSIDATA1 I/O (SSIF)	$\overline{\text{TEND2}}$ output (DMAC)	—	—
	PH4 I/O (port)	SSIWS1 I/O (SSIF)	$\overline{\text{DACK2}}$ output (DMAC)	$\overline{\text{DACT2}}$ output (DMAC)	—
PHCRL1	PH3 I/O (port)	SSISCK1 I/O (SSIF)	DREQ2 input (DMAC)	—	—
	PH2 I/O (port)	SSIDATA0 I/O (SSIF)	—	—	—
	PH1 I/O (port)	SSIWS0 I/O (SSIF)	—	—	—
	PH0 I/O (port)	SSISCK0 I/O (SSIF)	—	—	—

Table 27.9 Multiplexed Pins (Port J)

	Setting of Mode Bits (PJnMD[3:0])					
	0000	0001	0010	0011	0100	0101
Setting Register	Function 1 (General I/O)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)
PJCRL4	PJ12 output (port)	VIDATA7 input (Video-In)	$\overline{\text{SCS1}}$ I/O (SSU)	—	FRB input (FLCTL)	—
PJCRL3	PJ11 I/O (port)	VIDATA6 input (Video-In)	SSO1 I/O (SSU)	—	—	—
	PJ10 I/O (port)	VIDATA5 input (Video-In)	SSI1 I/O (SSU)	—	—	—
	PJ9 I/O (port)	VIDATA4 input (Video-In)	SSCK1 I/O (SSU)	—	—	—
	PJ8 I/O (port)	VIDATA3 input (Video-In)	TIOC1B I/O (MTU2)	—	NAF7 I/O (FLCTL)	—
PJCRL2	PJ7 I/O (port)	VIDATA2 input (Video-In)	TIOC1A I/O (MTU2)	—	NAF6 I/O (FLCTL)	—
	PJ6 I/O (port)	VIDATA1 input (Video-In)	$\overline{\text{TEND3}}$ output (DMAC)	—	$\overline{\text{FCE}}$ output (FLCTL)	—
	PJ5 I/O (port)	VIDATA0 input (Video-In)	$\overline{\text{DACK3}}$ output (DMAC)	$\overline{\text{DACT3}}$ output (DMAC)	FSC output (FLCTL)	TxD4 output (SCIF)
	PJ4 I/O (port)	VICLK input (Video-In)	DREQ3 input (DMAC)	—	FOE output (FLCTL)	RxD4 input (SCIF)
PJCRL1	PJ3 I/O (port)	IRQ7 input (INTC)	TIOC0D I/O (MTU2)	—	—	TxD3 output (SCIF)
	PJ2 I/O (port)	IRQ6 input (INTC)	TIOC0C I/O (MTU2)	—	—	RxD3 input (SCIF)
	PJ1 I/O (port)	IRQ5 input (INTC)	TIOC0B I/O (MTU2)	—	—	—
	PJ0 I/O (port)	IRQ4 input (INTC)	TIOC0A I/O (MTU2)	—	—	—

Table 27.10 Multiplexed Pins (Port K)

	Setting of Mode Bits (PKnMD[3:0])				
	0000	0001	0010	0011	0100
Setting Register	Function 1 (General I/O)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)
PKCRL1	PK1 I/O (port)	DCLKIN input (Video-Out)	—	—	FCDE output (FLCTL)
	PK0 I/O (port)	CSYNC output (Video-Out)	—	—	$\overline{\text{FWE}}$ output (FLCTL)

27.1 Features

- Functions for the multiplexed pins can be selected by setting the control registers.
- When the general I/O function or TIOC I/O function of the MTU2 is specified, the I/O direction should be selected by setting the corresponding I/O register.

27.2 Register Descriptions

The PFC has the following registers.

Table 27.11 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A I/O register L	PAIORL	R/W	H'0000	H'FFFE3802	8, 16
Port A control register L4	PACRL4	R/W	H'0000/ H'1111*2	H'FFFE380C	8, 16, 32
Port A control register L3	PACRL3	R/W	H'0000/ H'1111*2	H'FFFE380E	8, 16
Port A control register L2	PACRL2	R/W	H'0000/ H'1111*2	H'FFFE3810	8, 16, 32
Port A control register L1	PACRL1	R/W	H'0000/ H'1111*2	H'FFFE3812	8, 16
Port B I/O register H	PBIORH	R/W	H'0000	H'FFFE3820	8, 16, 32
Port B I/O register L	PBIORL	R/W	H'0000	H'FFFE3822	8, 16
Port B control register H2	PBCRH2	R/W	H'0000	H'FFFE3828	8*1, 16, 32
Port B control register H1	PBCRH1	R/W	H'0000	H'FFFE382A	8, 16
Port B control register L4	PBCRL4	R/W	H'0000	H'FFFE382C	8, 16, 32
Port B control register L3	PBCRL3	R/W	H'0000	H'FFFE382E	8, 16
Port B control register L2	PBCRL2	R/W	H'0000	H'FFFE3830	8, 16, 32
Port B control register L1	PBCRL1	R/W	H'0000/ H'0010*2	H'FFFE3832	8, 16
Port C I/O register L	PCIORL	R/W	H'0000	H'FFFE3842	8, 16
Port C control register L3	PCCRL3	R/W	H'0000	H'FFFE384E	8, 16
Port C control register L2	PCCRL2	R/W	H'0000	H'FFFE3850	8, 16, 32
Port C control register L1	PCCRL1	R/W	H'0000	H'FFFE3852	8, 16
Port D I/O register L	PDIORL	R/W	H'0000	H'FFFE3862	8, 16
Port D control register L1	PDCRL1	R/W	H'0000	H'FFFE3872	8, 16
Port E I/O register L	PEIORL	R/W	H'0000	H'FFFE3882	8, 16
Port E control register L4	PECRL4	R/W	H'0000	H'FFFE388C	8, 16, 32
Port E control register L3	PECRL3	R/W	H'0000	H'FFFE388E	8, 16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port E control register L2	PECRL2	R/W	H'0000	H'FFFE3890	8, 16, 32
Port E control register L1	PECRL1	R/W	H'0000	H'FFFE3892	8, 16
Port F I/O register L	PFIORL	R/W	H'0000	H'FFFE38A2	8, 16
Port F control register L2	PFCRL2	R/W	H'0000	H'FFFE38B0	8, 16, 32
Port F control register L1	PFCRL1	R/W	H'0000	H'FFFE38B2	8, 16
Port G control register L2	PGCRL2	R/W	H'0000	H'FFFE38D0	8, 16, 32
Port G control register L1	PGCRL1	R/W	H'0000	H'FFFE38D2	8, 16
Port H I/O register L	PHIORL	R/W	H'0000	H'FFFE38E2	8, 16
Port H control register L4	PHCRL4	R/W	H'0000	H'FFFE38EC	8, 16, 32
Port H control register L3	PHCRL3	R/W	H'0000	H'FFFE38EE	8, 16
Port H control register L2	PHCRL2	R/W	H'0000	H'FFFE38F0	8, 16, 32
Port H control register L1	PHCRL1	R/W	H'0000	H'FFFE38F2	8, 16
Port J I/O register L	PJIORL	R/W	H'0000	H'FFFE3902	8, 16
Port J control register L4	PJCRL4	R/W	H'0000	H'FFFE390C	8, 16, 32
Port J control register L3	PJCRL3	R/W	H'0000	H'FFFE390E	8, 16
Port J control register L2	PJCRL2	R/W	H'0000	H'FFFE3910	8, 16, 32
Port J control register L1	PJCRL1	R/W	H'0000	H'FFFE3912	8, 16
Port K I/O register L	PKIORL	R/W	H'0000	H'FFFE3922	8, 16
Port K control register L1	PKCRL1	R/W	H'0000	H'FFFE3932	8, 16

Notes: 1. In 8-bit access, the register can be read but cannot be written to.
2. The initial value depends on the operating mode of the LSI.

27.2.1 Port A I/O Register L (PAIORL)

PAIORL is a 16-bit readable/writable register that is used to set the pins on port A as inputs or outputs. The PA15IOR to PA0IOR bits correspond to the PA15 to PA0 pins, respectively. The setting of PAIORL is valid for the pins for which general I/O function or TIOC I/O function of the MTU2 is selected. PAIORL has no effect on the pins for which other function is selected. If a bit in PAIORL is set to 1, the corresponding pin on port A functions as output. If it is cleared to 0, the corresponding pin functions as input.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 IOR	PA14 IOR	PA13 IOR	PA12 IOR	PA11 IOR	PA10 IOR	PA9 IOR	PA8 IOR	PA7 IOR	PA6 IOR	PA5 IOR	PA4 IOR	PA3 IOR	PA2 IOR	PA1 IOR	PA0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

27.2.2 Port A Control Registers L1 to L4 (PACRL1 to PACRL4)

PACRL1 to PACRL4 are 16-bit readable/writable registers that are used to select the function of the multiplexed pins on port A. See table 27.1 for the multiplexed functions.

(1) Port A Control Register L4 (PACRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15MD[3:0]				PA14MD[3:0]				PA13MD[3:0]				PA12MD[3:0]			
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PA15MD[3:0]	0000/ 0001*	R/W	PA15 Mode Select the function of the PA15.
11 to 8	PA14MD[3:0]	0000/ 0001*	R/W	PA14 Mode Select the function of the PA14.
7 to 4	PA13MD[3:0]	0000/ 0001*	R/W	PA13 Mode Select the function of the PA13.
3 to 0	PA12MD[3:0]	0000/ 0001*	R/W	PA12 Mode Select the function of the PA12.

Note: * The initial value is 0000 in 16-bit mode (MD = 0), and 0001 in 32-bit mode (MD= 1).

(2) Port A Control Register L3 (PACRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA11MD[3:0]				PA10MD[3:0]				PA9MD[3:0]				PA8MD[3:0]			
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PA11MD[3:0]	0000/ 0001*	R/W	PA11 Mode Select the function of the PA11.
11 to 8	PA10MD[3:0]	0000/ 0001*	R/W	PA10 Mode Select the function of the PA10.
7 to 4	PA9MD[3:0]	0000/ 0001*	R/W	PA9 Mode Select the function of the PA9.
3 to 0	PA8MD[3:0]	0000/ 0001*	R/W	PA8 Mode Select the function of the PA8.

Note: * The initial value is 0000 in 16-bit mode (MD = 0), and 0001 in 32-bit mode (MD= 1).

(3) Port A Control Register L2 (PACRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA7MD[3:0]				PA6MD[3:0]				PA5MD[3:0]				PA4MD[3:0]			
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PA7MD[3:0]	0000/ 0001*	R/W	PA7 Mode Select the function of the PA7.
11 to 8	PA6MD[3:0]	0000/ 0001*	R/W	PA6 Mode Select the function of the PA6.
7 to 4	PA5MD[3:0]	0000/ 0001*	R/W	PA5 Mode Select the function of the PA5.
3 to 0	PA4MD[3:0]	0000/ 0001*	R/W	PA4 Mode Select the function of the PA4.

Note: * The initial value is 0000 in 16-bit mode (MD = 0), and 0001 in 32-bit mode (MD= 1).

(4) Port A Control Register L1 (PACRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA3MD[3:0]				PA2MD[3:0]				PA1MD[3:0]				PA0MD[3:0]			
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PA3MD[3:0]	0000/ 0001*	R/W	PA3 Mode Select the function of the PA3.
11 to 8	PA2MD[3:0]	0000/ 0001*	R/W	PA2 Mode Select the function of the PA2.
7 to 4	PA1MD[3:0]	0000/ 0001*	R/W	PA1 Mode Select the function of the PA1.
3 to 0	PA0MD[3:0]	0000/ 0001*	R/W	PA0 Mode Select the function of the PA0.

Note: * The initial value is 0000 in 16-bit mode (MD = 0), and 0001 in 32-bit mode (MD= 1).

27.2.3 Port B I/O Register H (PBIORH)

PBIORH is a 16-bit readable/writable register that is used to set the pins on port B as inputs or outputs. The PB18IOR to PB16IOR bits correspond to the PB18 to PB16 pins, respectively. The setting of PBIORH is valid for the pins for which general I/O function is selected and has no effect on the pins for which other function is selected. If a bit in PBIORH is set to 1, the corresponding pin on port B functions as output. If it is cleared to 0, the corresponding pin functions as input.

Bits 15 to 3 in PBIORH are reserved. These bits are always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	PB18 IOR	PB17 IOR	PB16 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

27.2.4 Port B I/O Register L (PBIORL)

PBIORL is a 16-bit readable/writable register that is used to set the pins on port B as inputs or outputs. The PB15IOR to PB0IOR bits correspond to the PB15 to PB0 pins, respectively. The setting of PBIORL is valid for the pins for which general I/O function is selected and has no effect on the pins for which other function is selected. If a bit in PBIORL is set to 1, the corresponding pin on port B functions as output. If it is cleared to 0, the corresponding pin functions as input.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB15IOR	PB14IOR	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IOR	PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

27.2.5 Port B Control Registers H1, H2, L1 to L4 (PBCRH1, PBCRH2, PBCRL1 to PBCRL4)

PBCRH1, PBCRH2, and PBCRL1 to PBCRL4 are 16-bit readable/writable registers that are used to select the function of the multiplexed pins on port B. See table 27.2 for the multiplexed functions.

(1) Port B Control Register H2 (PBCRH2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	PB18MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0*1	0*1	0*1	0*1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W*2	R/W*2	R/W*2	R/W*2

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0.
3 to 0	PB18MD[3:0]	0000*1	R/W*2	PB18 Mode Select the function of the PB18.

Notes: 1. Not initialized by a reset triggered by WDT overflow.
2. To write to PBCRH2, write by 16-bit or 32-bit access such that the write value for bits 15 to 8 is H'A5 and that for bits 7 to 4 is 0.

(2) Port B Control Register H1 (PBCRH1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PB17MD[3:0]				PB16MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 to 4	PB17MD[3:0]	0000	R/W	PB17 Mode Select the function of the PB17.
3 to 0	PB16MD[3:0]	0000	R/W	PB16 Mode Select the function of the PB16.

(3) Port B Control Register L4 (PBCRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB15MD[3:0]				PB14MD[3:0]				PB13MD[3:0]				PB12MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PB15MD[3:0]	0000	R/W	PB15 Mode Select the function of the PB15.
11 to 8	PB14MD[3:0]	0000	R/W	PB14 Mode Select the function of the PB14.
7 to 4	PB13MD[3:0]	0000	R/W	PB13 Mode Select the function of the PB13.
3 to 0	PB12MD[3:0]	0000	R/W	PB12 Mode Select the function of the PB12.

(4) Port B Control Register L3 (PBCRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB11MD[3:0]				PB10MD[3:0]				PB9MD[3:0]				PB8MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PB11MD[3:0]	0000	R/W	PB11 Mode Select the function of the PB11.
11 to 8	PB10MD[3:0]	0000	R/W	PB10 Mode Select the function of the PB10.
7 to 4	PB9MD[3:0]	0000	R/W	PB9 Mode Select the function of the PB9.
3 to 0	PB8MD[3:0]	0000	R/W	PB8 Mode Select the function of the PB8.

(5) Port B Control Register L2 (PBCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB7MD[3:0]				PB6MD[3:0]				PB5MD[3:0]				PB4MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PB7MD[3:0]	0000	R/W	PB7 Mode Select the function of the PB7.
11 to 8	PB6MD[3:0]	0000	R/W	PB6 Mode Select the function of the PB6.
7 to 4	PB5MD[3:0]	0000	R/W	PB5 Mode Select the function of the PB5.
3 to 0	PB4MD[3:0]	0000	R/W	PB4 Mode Select the function of the PB4.

(6) Port B Control Register L1 (PBCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB3MD[3:0]				PB2MD[3:0]				PB1MD[3:0]				PB0MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0/1*	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PB3MD[3:0]	0000	R/W	PB3 Mode Select the function of the PB3.
11 to 8	PB2MD[3:0]	0000	R/W	PB2 Mode Select the function of the PB2.
7 to 4	PB1MD[3:0]	0001/ 0000*	R/W	PB1 Mode Select the function of the PB1.
3 to 0	PB0MD[3:0]	0000	R/W	PB0 Mode Select the function of the PB0.

Note: * The initial value is 0001 in 16-bit mode (MD = 0), and 0000 in 32-bit mode (MD= 1).

27.2.6 Port C I/O Register L (PCIORL)

PCIORL is a 16-bit readable/writable register that is used to set the pins on port C as inputs or outputs. The PC10IOR to PC0IOR bits correspond to the PC10 to PC0 pins, respectively. The setting of PCIORL is valid for the pins for which general I/O function or TIOC I/O function of the MTU2 is selected. PCIORL has no effect on the pins for which other function is selected. If a bit in PCIORL is set to 1, the corresponding pin on port C functions as an output pin. If it is cleared to 0, the corresponding pin functions as an input pin.

Bits 15 to 11 in PCIORL are reserved. These bits are always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PC10 IOR	PC9 IOR	PC8 IOR	PC7 IOR	PC6 IOR	PC5 IOR	PC4 IOR	PC3 IOR	PC2 IOR	PC1 IOR	PC0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

27.2.7 Port C Control Registers L1 to L3 (PCCRL1 to PCCRL3)

PCCRL1 to PCCRL3 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port C. See table 27.3 for the multiplexed functions.

(1) Port C Control Register L3 (PCCRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PC10MD[3:0]				PC9MD[3:0]				PC8MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	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 8	PC10MD[3:0]	0000	R/W	PC10 Mode Select the function of the PC10.
7 to 4	PC9MD[3:0]	0000	R/W	PC9 Mode Select the function of the PC9.
3 to 0	PC8MD[3:0]	0000	R/W	PC8 Mode Select the function of the PC8.

(2) Port C Control Register L2 (PCCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC7MD[3:0]				PC6MD[3:0]				PC5MD[3:0]				PC4MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PC7MD[3:0]	0000	R/W	PC7 Mode Select the function of the PC7.
11 to 8	PC6MD[3:0]	0000	R/W	PC6 Mode Select the function of the PC6.
7 to 4	PC5MD[3:0]	0000	R/W	PC5 Mode Select the function of the PC5.
3 to 0	PC4MD[3:0]	0000	R/W	PC4 Mode Select the function of the PC4.

(3) Port C Control Register L1 (PCCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC3MD[3:0]				PC2MD[3:0]				PC1MD[3:0]				PC0MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PC3MD[3:0]	0000	R/W	PC3 Mode Select the function of the PC3.
11 to 8	PC2MD[3:0]	0000	R/W	PC2 Mode Select the function of the PC2.
7 to 4	PC1MD[3:0]	0000	R/W	PC1 Mode Select the function of the PC1.
3 to 0	PC0MD[3:0]	0000	R/W	PC0 Mode Select the function of the PC0.

27.2.8 Port D I/O Register L (PDIORL)

PDIORL is a 16-bit readable/writable register that is used to set the pins on port D as inputs or outputs. The PD2IOR to PD0IOR bits correspond to the PD2 to PD0 pins, respectively. The setting of PDIORL is valid for the pins for which general I/O function is selected and has no effect on the pins for which other function is selected. If a bit in PDIORL is set to 1, the corresponding pin on port D functions as an output. If it is cleared to 0, the corresponding pin functions as an input.

Bits 15 to 3 in PDIORL are reserved. These bits are always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	PD2 IOR	PD1 IOR	PD0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

27.2.9 Port D Control Register L1 (PDCRL1)

PDCRL1 is 16-bit readable/writable register that is used to select the functions of the multiplexed pins on port D. See table 27.4 for the multiplexed functions.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PD2MD[3:0]				PD1MD[3:0]				PD0MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	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 8	PD2MD[3:0]	0000	R/W	PD2 Mode Select the function of the PD2.
7 to 4	PD1MD[3:0]	0000	R/W	PD1 Mode Select the function of the PD1.
3 to 0	PD0MD[3:0]	0000	R/W	PD0 Mode Select the function of the PD0.

27.2.10 Port E I/O Register L (PEIORL)

PEIORL is 16-bit readable/writable register that is used to set the pins on port E as inputs or outputs. The PE13IOR, PE11IOR, PE9IOR, and PE7IOR to PE0IOR bits correspond to the PE13, PE11, PE9, and PE7 to PE0 pins respectively. The setting of PEIORL is valid for the pins for which general I/O function or TIOC I/O function of the MTU2 is selected. PEIORL has no effect on the pins for which other function is selected. If a bit in PEIORL is set to 1, the corresponding pin on port E functions as an output pin. If it is cleared to 0, the corresponding pin functions as an input pin.

Bits 15, 14, 12, 10, and 8 in PEIORL are reserved. These bits are always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE13 IOR	-	PE11 IOR	-	PE9 IOR	-	PE7 IOR	PE6 IOR	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR	PE0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

27.2.11 Port E Control Registers L1 to L4 (PECRL1 to PECRL4)

PECRL1 to PECRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port E. See table 27.5 for the multiplexed functions.

(1) Port E Control Register L4 (PECRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PE13MD[3:0]				PE12MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 to 4	PE13MD[3:0]	0000	R/W	PE13 Mode Select the function of the PE13.
3 to 0	PE12MD[3:0]	0000	R/W	PE12 Mode Select the function of the PE12.

(2) Port E Control Register L3 (PECRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE11MD[3:0]				PE10MD[3:0]				PE9MD[3:0]				PE8MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PE11MD[3:0]	0000	R/W	PE11 Mode Select the function of the PE11.
11 to 8	PE10MD[3:0]	0000	R/W	PE10 Mode Select the function of the PE10.
7 to 4	PE9MD[3:0]	0000	R/W	PE9 Mode Select the function of the PE9.
3 to 0	PE8MD[3:0]	0000	R/W	PE8 Mode Select the function of the PE8.

(3) Port E Control Register L2 (PECRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE7MD[3:0]				PE6MD[3:0]				PE5MD[3:0]				PE4MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PE7MD[3:0]	0000	R/W	PE7 Mode Select the function of the PE7.
11 to 8	PE6MD[3:0]	0000	R/W	PE6 Mode Select the function of the PE6.
7 to 4	PE5MD[3:0]	0000	R/W	PE5 Mode Select the function of the PE5.
3 to 0	PE4MD[3:0]	0000	R/W	PE4 Mode Select the function of the PE4.

(4) Port E Control Register L1 (PECRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE3MD[3:0]				PE2MD[3:0]				PE1MD[3:0]				PE0MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PE3MD[3:0]	0000	R/W	PE3 Mode Select the function of the PE3.
11 to 8	PE2MD[3:0]	0000	R/W	PE2 Mode Select the function of the PE2.
7 to 4	PE1MD[3:0]	0000	R/W	PE1 Mode Select the function of the PE1.
3 to 0	PE0MD[3:0]	0000	R/W	PE0 Mode Select the function of the PE0.

27.2.12 Port F I/O Register L (PFIORL)

PFIORL is a 16-bit readable/writable register that is used to set the pins on port F as inputs or outputs. The PF4IOR to PF0IOR bits correspond to the PF4 to PF0 pins, respectively. PFIORL is enabled when the port F pins are functioning as general-purpose inputs/outputs. In other states, PFIORL is disabled. If a bit in PFIORL is set to 1, the corresponding pin on port F functions as an output. If it is cleared to 0, the corresponding pin functions as an input.

Bits 15 to 5 in PFIORL are reserved. These bits are always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	PF4IOR	PF3IOR	PF2IOR	PF1IOR	PF0IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

27.2.13 Port F Control Registers L1, L2 (PFCRL1, PFCRL2)

PFCRL1 and PFCRL2 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port F. See table 27.6 for the multiplexed functions.

(1) Port F Control Register L2 (PFCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	PF4MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

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 to 0	PF4MD[3:0]	0000	R/W	PF4 Mode Select the function of the PF4.

(2) Port F Control Register L1 (PFCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF3MD[3:0]				PF2MD[3:0]				PF1MD[3:0]				PF0MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PF3MD[3:0]	0000	R/W	PF3 Mode Select the function of the PF3.
11 to 8	PF2MD[3:0]	0000	R/W	PF2 Mode Select the function of the PF2.
7 to 4	PF1MD[3:0]	0000	R/W	PF1 Mode Select the function of the PF1.
3 to 0	PF0MD[3:0]	0000	R/W	PF0 Mode Select the function of the PF0.

27.2.14 Port G Control Registers L1, L2 (PGCRL1, PGCRL2)

PGCRL1 and PGCRL2 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port G. See table 27.7 for the multiplexed functions.

(1) Port G Control Register L2 (PGCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG7MD[3:0]				PG6MD[3:0]				PG5MD[3:0]				PG4MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PG7MD[3:0]	0000	R/W	PG7 Mode Select the function of the PG7.
11 to 8	PG6MD[3:0]	0000	R/W	PG6 Mode Select the function of the PG6.
7 to 4	PG5MD[3:0]	0000	R/W	PG5 Mode Select the function of the PG5.
3 to 0	PG4MD[3:0]	0000	R/W	PG4 Mode Select the function of the PG4.

(2) Port G Control Register L1 (PGCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG3MD[3:0]				PG2MD[3:0]				PG1MD[3:0]				PG0MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PG3MD[3:0]	0000	R/W	PG3 Mode Select the function of the PG3.
11 to 8	PG2MD[3:0]	0000	R/W	PG2 Mode Select the function of the PG2.
7 to 4	PG1MD[3:0]	0000	R/W	PG1 Mode Select the function of the PG1.
3 to 0	PG0MD[3:0]	0000	R/W	PG0 Mode Select the function of the PG0.

27.2.15 Port H I/O Register L (PHIORL)

PHIORL is 16-bit readable/writable register that is used to set the pins on port H as inputs or outputs. The PH15IOR to PH0IOR bits correspond to the PH15 to PH0 pins respectively. The setting of PHIORL is valid for the pins for which general I/O function is selected and has no effect on the pins for which other function is selected. If a bit in PHIORL is set to 1, the corresponding pin on port H functions as an output pin. If it is cleared to 0, the corresponding pin functions as an input pin.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH15 IOR	PH14 IOR	PH13 IOR	PH12 IOR	PH11 IOR	PH10 IOR	PH9 IOR	PH8 IOR	PH7 IOR	PH6 IOR	PH5 IOR	PH4 IOR	PH3 IOR	PH2 IOR	PH1 IOR	PH0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

27.2.16 Port H Control Registers L1 to L4 (PHCRL1 to PHCRL4)

PHCRL1 to PHCRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port H. See table 27.8 for the multiplexed functions.

(1) Port H Control Register L4 (PHCRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH15MD[3:0]				PH14MD[3:0]				PH13MD[3:0]				PH12MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PH15MD[3:0]	0000	R/W	PH15 Mode Select the function of the PH15.
11 to 8	PH14MD[3:0]	0000	R/W	PH14 Mode Select the function of the PH14.
7 to 4	PH13MD[3:0]	0000	R/W	PH13 Mode Select the function of the PH13.
3 to 0	PH12MD[3:0]	0000	R/W	PH12 Mode Select the function of the PH12.

(2) Port H Control Register L3 (PHCRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH11MD[3:0]				PH10MD[3:0]				PH9MD[3:0]				PH8MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PH11MD[3:0]	0000	R/W	PH11 Mode Select the function of the PH11.
11 to 8	PH10MD[3:0]	0000	R/W	PH10 Mode Select the function of the PH10.
7 to 4	PH9MD[3:0]	0000	R/W	PH9 Mode Select the function of the PH9.
3 to 0	PH8MD[3:0]	0000	R/W	PH8 Mode Select the function of the PH8.

(3) Port H Control Register L2 (PHCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH7MD[3:0]				PH6MD[3:0]				PH5MD[3:0]				PH4MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PH7MD[3:0]	0000	R/W	PH7 Mode Select the function of the PH7.
11 to 8	PH6MD[3:0]	0000	R/W	PH6 Mode Select the function of the PH6.
7 to 4	PH5MD[3:0]	0000	R/W	PH5 Mode Select the function of the PH5.
3 to 0	PH4MD[3:0]	0000	R/W	PH4 Mode Select the function of the PH4.

(4) Port H Control Register L1 (PHCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH3MD[3:0]				PH2MD[3:0]				PH1MD[3:0]				PH0MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PH3MD[3:0]	0000	R/W	PH3 Mode Select the function of the PH3.
11 to 8	PH2MD[3:0]	0000	R/W	PH2 Mode Select the function of the PH2.
7 to 4	PH1MD[3:0]	0000	R/W	PH1 Mode Select the function of the PH1.
3 to 0	PH0MD[3:0]	0000	R/W	PH0 Mode Select the function of the PH0.

27.2.17 Port J I/O Register L (PJIORL)

PJIORL is 16-bit readable/writable register that is used to set the pins on port J as inputs or outputs. The PJ12IOR to PJ0IOR bits correspond to the PJ12 to PJ0 pins respectively. The setting of PJIORL is valid for the pins for which general I/O function or TIOC I/O function of the MTU2 is selected. PJIORL has no effect on the pins for which other function is selected. If a bit in PJIORL is set to 1, the corresponding pin on port J functions as an output pin. If it is cleared to 0, the corresponding pin functions as an input pin.

Bits 15 to 13 in PJIORL are reserved. These bits are always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PJ12 IOR	PJ11 IOR	PJ10 IOR	PJ9 IOR	PJ8 IOR	PJ7 IOR	PJ6 IOR	PJ5 IOR	PJ4 IOR	PJ3 IOR	PJ2 IOR	PJ1 IOR	PJ0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

27.2.18 Port J Control Registers L1 to L4 (PJCRL1 to PJCRL4)

PJCRL1 to PJCRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port J. See table 27.9 for the multiplexed functions.

(1) Port J Control Register L4 (PJCRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	PJ12MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

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 to 0	PJ12MD[3:0]	0000	R/W	PJ12 Mode Select the function of the PJ12.

(2) Port J Control Register L3 (PJCRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ11MD[3:0]				PJ10MD[3:0]				PJ9MD[3:0]				PJ8MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PJ11MD[3:0]	0000	R/W	PJ11 Mode Select the function of the PJ11.
11 to 8	PJ10MD[3:0]	0000	R/W	PJ10 Mode Select the function of the PJ10.
7 to 4	PJ9MD[3:0]	0000	R/W	PJ9 Mode Select the function of the PJ9.
3 to 0	PJ8MD[3:0]	0000	R/W	PJ8 Mode Select the function of the PJ8.

(3) Port J Control Register L2 (PJCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ7MD[3:0]				PJ6MD[3:0]				PJ5MD[3:0]				PJ4MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PJ7MD[3:0]	0000	R/W	PJ7 Mode Select the function of the PJ7.
11 to 8	PJ6MD[3:0]	0000	R/W	PJ6 Mode Select the function of the PJ6.
7 to 4	PJ5MD[3:0]	0000	R/W	PJ5 Mode Select the function of the PJ5.
3 to 0	PJ4MD[3:0]	0000	R/W	PJ4 Mode Select the function of the PJ4.

(4) Port J Control Register L1 (PJCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ3MD[3:0]				PJ2MD[3:0]				PJ1MD[3:0]				PJ0MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PJ3MD[3:0]	0000	R/W	PJ3 Mode Select the function of the PJ3.
11 to 8	PJ2MD[3:0]	0000	R/W	PJ2 Mode Select the function of the PJ2.
7 to 4	PJ1MD[3:0]	0000	R/W	PJ1 Mode Select the function of the PJ1.
3 to 0	PJ0MD[3:0]	0000	R/W	PJ0 Mode Select the function of the PJ0.

27.2.19 Port K I/O Register L (PKIORL)

PKIORL is 16-bit readable/writable register that is used to set the pins on port K as inputs or outputs. The PK11IOR to PK0IOR bits correspond to the PK11 to PK0 pins respectively. The setting of PKIORL is valid for the pins for which general I/O function is selected and has no effect on the pins for which other function is selected. If a bit in PKIORL is set to 1, the corresponding pin on port K functions as an output pin. If it is cleared to 0, the corresponding pin functions as an input pin.

Bits 15 to 2 in PKIORL are reserved. These bits are always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PK1IOR	PK0IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

27.2.20 Port K Control Register L1 (PKCRL1)

PKCRL1 is 16-bit readable/writable register that is used to select the functions of the multiplexed pins on port K. See table 27.10 for the multiplexed functions.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PK1MD[3:0]				PK0MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 to 4	PK1MD[3:0]	0000	R/W	PK1 Mode Select the function of the PK1.
3 to 0	PK0MD[3:0]	0000	R/W	PK0 Mode Select the function of the PK0.

27.3 Usage Notes

The multiplexed pins listed in tables 27.1 to 27.10 except for pins PE8 to PE13, PF0, PF1, and PG0 to PG7 are provided with weak keepers or pull-up circuit (PB18) in their I/O buffers to prevent the pins from floating into intermediate voltage levels. However, note that the voltage retained in the high-impedance state may fluctuate due to noise.

Section 28 I/O Ports

This LSI has ten ports: A to H, J, and K.

All port pins are multiplexed with other pin functions. The functions of the multiplex pins are selected by means of the pin function controller (PFC).

Each port is provided with data registers for storing the pin data and port registers for reading the states of the pins.

28.1 Features

1. Total number of port pins: 107 pins (I/O: 96 pins, Input: 11 pins)
   - Port A: (Input: 16 pins)
   - Port B: (I/O: 19 pins)
   - Port C: (I/O: 11 pins)
   - Port D: (I/O: 3 pins)
   - Port E: (I/O: 11 pins, Input: 3 pins)
   - Port F: (I/O: 5 pins)
   - Port G: (Input: 8 pins)
   - Port H: (I/O: 16 pins)
   - Port J: (I/O: 13 pins)
   - Port K: (I/O: 2 pins)

2. Pins with a weak keeper

   The following pins of this LSI have a weak keeper circuit or pull-up circuit (PB18) that prevents the pin from floating into intermediate voltage levels.
   - Port A: PA0 to PA15
   - Port B: PB0 to PB18
   - Port C: PC0 to PC10
   - Port D: PD0 to PD2
   - Port E: PE0 to PE7
   - Port F: PF2 to PF4
   - Port H: PH0 to PH15
   - Port J: PJ0 to PJ12
   - Port K: PK0 and PK1

The I/O pins include a weak keeper circuit or pull-up circuit that fixes the input level high or low when the I/O pins are not driven from outside. Generally in the CMOS products, input levels on unused input pins must be fixed by way of external pull-up or pull-down resistors. However, the I/O pins having a weak keeper circuit or pull-up circuit of this LSI can eliminate these outer circuits and reduce the number of parts in the system. If the pull-up or pull-down resistors is necessary to fix the pin level, use the resistor of 10 kΩ or smaller.

28.2 Register Descriptions

The port has the following registers.

Table 28.1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A data register L	PADRL	R/W	H'0000	H'FFFE3816	8, 16
Port A port register L	PAPRL	R	H'xxxx	H'FFFE381A	8, 16
Port B data register H	PBDRH	R/W	H'0000	H'FFFE3834	8, 16, 32
Port B data register L	PBDRL	R/W	H'0000	H'FFFE3836	8, 16
Port B port register H	PBPRH	R	H'000x	H'FFFE3838	8, 16, 32
Port B port register L	PBPRL	R	H'xxxx	H'FFFE383A	8, 16
Port C data register L	PCDRL	R/W	H'0000	H'FFFE3856	8, 16
Port C port register L	PCPRL	R	H'0xxx	H'FFFE385A	8, 16
Port D data register L	PDDRL	R/W	H'0000	H'FFFE3876	8, 16
Port D port register L	PDPRL	R	H'000x	H'FFFE387A	8, 16
Port E data register L	PEDRL	R/W	H'xx00	H'FFFE3896	8, 16
Port E port register L	PEPRL	R	H'xxxx	H'FFFE389A	8, 16
Port F data register L	PFDRL	R/W	H'0000	H'FFFE38B6	8, 16
Port F port register L	PFPRL	R	H'00xx	H'FFFE38BA	8, 16
Port G data register L	PGDRL	R/W	H'00xx	H'FFFE38D6	8, 16
Port H data register L	PHDRL	R/W	H'0000	H'FFFE38F6	8, 16
Port H port register L	PHPRL	R	H'xxxx	H'FFFE38FA	8, 16
Port J data register L	PJDRL	R/W	H'0000	H'FFFE3916	8, 16
Port J port register L	PJPRL	R	H'xxxx	H'FFFE391A	8, 16
Port K data register L	PKDRL	R/W	H'0000	H'FFFE3936	8, 16
Port K port register L	PKPRL	R	H'xxxx	H'FFFE393A	8, 16

28.2.1 Port A Data Register L (PADRL)

PADRL is a 16-bit readable/writable register that stores port A data. The PA15DR to PA0DR bits correspond to the PA15 to PA0 pins, respectively.

When a pin function is general output, if a value is written to PADRL, that value is output from the pin, and if PADRL is read, the register value is returned regardless of the pin state.

When a pin function is general input, if PADRL is read, the pin state, not the register value, is returned directly. If a value is written to PADRL, although that value is written into PADRL, it does not affect the pin state. Table 28.2 summarizes PADRL read/write operations.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PA15DR	0	R/W	See table 28.2.
14	PA14DR	0	R/W	
13	PA13DR	0	R/W	
12	PA12DR	0	R/W	
11	PA11DR	0	R/W	
10	PA10DR	0	R/W	
9	PA9DR	0	R/W	
8	PA8DR	0	R/W	
7	PA7DR	0	R/W	
6	PA6DR	0	R/W	
5	PA5DR	0	R/W	
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

Table 28.2 Port A Data Register L (PADRL) Read/Write Operation

- Bits 15 to 0 of PADRL

PAIORL	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PADRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PADRL, but it has no effect on pin state
1	General output	PADRL value	Value written is output from pin
	Other than general output	PADRL value	Can write to PADRL, but it has no effect on pin state

28.2.2 Port A Port Register L (PAPRL)

PAPRL is a 16-bit read-only register, in which the PA15PR to PA0PR bits correspond to the PA15 to PA0 pins, respectively. PAPRL always returns the states of the pins regardless of the PFC setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15PR	PA14PR	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR	PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR
Initial value:	PA15	PA14	PA13	PA12	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PA15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PA14PR	Pin state	R	
13	PA13PR	Pin state	R	
12	PA12PR	Pin state	R	
11	PA11PR	Pin state	R	
10	PA10PR	Pin state	R	
9	PA9PR	Pin state	R	
8	PA8PR	Pin state	R	
7	PA7PR	Pin state	R	
6	PA6PR	Pin state	R	
5	PA5PR	Pin state	R	
4	PA4PR	Pin state	R	
3	PA3PR	Pin state	R	
2	PA2PR	Pin state	R	
1	PA1PR	Pin state	R	
0	PA0PR	Pin state	R	

28.2.3 Port B Data Registers H, L (PBDRH, PBDRL)

PBDRH and PBDRL are 16-bit readable/writable registers that store port B data. The PB18DR to PB0DR bits correspond to the PB18 to PB0 pins, respectively.

When a pin function is general output, if a value is written to PBDRH or PBDRL, that value is output from the pin, and if PBDRH or PBDRL is read, the register value is returned regardless of the pin state.

When a pin function is general input, if PBDRH or PBDRL is read, the pin state, not the register value, is returned directly. If a value is written to PBDRH or PBDRL, although that value is written into PBDRH or PBDRL, it does not affect the pin state. Table 28.3 summarizes PBDRH/PBDRL read/write operation.

(1) Port B Data Register H (PBDRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	PB18DR	PB17DR	PB16DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	PB18DR	0	R/W	See table 28.3.
1	PB17DR	0	R/W	
0	PB16DR	0	R/W	

(2) Port B Data Register L (PBDRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB15DR	PB14DR	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PB15DR	0	R/W	See table 28.3.
14	PB14DR	0	R/W	
13	PB13DR	0	R/W	
12	PB12DR	0	R/W	
11	PB11DR	0	R/W	
10	PB10DR	0	R/W	
9	PB9DR	0	R/W	
8	PB8DR	0	R/W	
7	PB7DR	0	R/W	
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

Table 28.3 Port B Data Registers H, L (PBDRH, PBDRL) Read/Write Operations

- Bits 2 to 0 of PBDRH and Bits 15 to 0 of PBDRL

PBIORH, L	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PBDRH/PBDRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PBDRH/PBDRL, but it has no effect on pin state
1	General output	PBDRH/PBDRL value	Value written is output from pin
	Other than general output	PBDRH/PBDRL value	Can write to PBDRH/PBDRL, but it has no effect on pin state

28.2.4 Port B Port Registers H, L (PBPRH, PBPRL)

PBPRH and PBPRL are 16-bit read-only registers, in which the PB18PR to PB0PR bits correspond to the PB18 to PB0 pins, respectively. PBPRH and PBPRL always return the states of the pins regardless of the PFC setting.

(1) Port B Port Register H (PBPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	PB18PR	PB17PR	PB16PR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	PB18	PB17	PB16
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
2	PB18PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
1	PB17PR	Pin state	R	
0	PB16PR	Pin state	R	

(2) Port B Port Register L (PBPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB15PR	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR	PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
Initial value:	PB15	PB14	PB13	PB12	PB11	PB10	PB9	PB8	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PB15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PB14PR	Pin state	R	
13	PB13PR	Pin state	R	
12	PB12PR	Pin state	R	
11	PB11PR	Pin state	R	
10	PB10PR	Pin state	R	
9	PB9PR	Pin state	R	
8	PB8PR	Pin state	R	
7	PB7PR	Pin state	R	
6	PB6PR	Pin state	R	
5	PB5PR	Pin state	R	
4	PB4PR	Pin state	R	
3	PB3PR	Pin state	R	
2	PB2PR	Pin state	R	
1	PB1PR	Pin state	R	
0	PB0PR	Pin state	R	

28.2.5 Port C Data Register L (PCDRL)

PCDRL is a 16-bit readable/writable register that stores port C data. The PC10DR to PC0DR bits correspond to the PC10 to PC0 pins, respectively.

When a pin function is general output, if a value is written to PCDRL, that value is output from the pin, and if PCDRL is read, the register value is returned regardless of the pin state.

When a pin function is general input, if PCDRL is read, the pin state, not the register value, is returned directly. If a value is written to PCDRL, although that value is written into PCDRL, it does not affect the pin state. Table 28.4 summarizes PCDRL read/write operations.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PC10DR	PC9DR	PC8DR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
10	PC10DR	0	R/W	See table 28.4.
9	PC9DR	0	R/W	
8	PC8DR	0	R/W	
7	PC7DR	0	R/W	
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

Table 28.4 Port C Data Register L (PCDRL) Read/Write Operations

- Bits 10 to 0 of PCDRL

PCIORL	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PCDRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PCDRL, but it has no effect on pin state
1	General output	PCDRL value	Value written is output from pin
	Other than general output	PCDRL value	Can write to PCDRL, but it has no effect on pin state

28.2.6 Port C Port Register L (PCPRL)

PCPRL is a 16-bit read-only register, in which the PC10PR to PC0PR bits correspond to the PC10 to PC0 pins, respectively. PCPRL always returns the states of the pins regardless of the PFC setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PC10PR	PC9PR	PC8PR	PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
Initial value:	0	0	0	0	0	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
10	PC10PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
9	PC9PR	Pin state	R	
8	PC8PR	Pin state	R	
7	PC7PR	Pin state	R	
6	PC6PR	Pin state	R	
5	PC5PR	Pin state	R	
4	PC4PR	Pin state	R	
3	PC3PR	Pin state	R	
2	PC2PR	Pin state	R	
1	PC1PR	Pin state	R	
0	PC0PR	Pin state	R	

28.2.7 Port D Data Register L (PDDRL)

PDDRL is a 16-bit readable/writable register that stores port D data. The PD2DR to PD0DR bits correspond to the PD2 to PD0 pins, respectively.

When a pin function is general output, if a value is written to PDDRL, that value is output from the pin, and if PDDRL is read, the register value is returned regardless of the pin state.

When a pin function is general input, if PDDRL is read, the pin state, not the register value, is returned directly. If a value is written to PDDRL, although that value is written into PDDRL, it does not affect the pin state. Table 28.5 summarizes PDDRL read/write operation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	PD2DR	PD1DR	PD0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	PD2DR	0	R/W	See table 28.5.
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

Table 28.5 Port D Data Register L (PDDRL) Read/Write Operation

- Bits 2 to 0 of PDDRL

PDIORL	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PDDRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PDDRL, but it has no effect on pin state
1	General output	PDDRL value	Value written is output from pin
	Other than general output	PDDRL value	Can write to PDDRL, but it has no effect on pin state

28.2.8 Port D Port Register L (PDPRL)

PDPRL is a 16-bit read-only register, in which the PD2PR to PD0PR bits correspond to the PD2 to PD0 pins, respectively. PDPRL always returns the states of the pins regardless of the PFC setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	PD2PR	PD1PR	PD0PR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	PD2	PD1	PD0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
2	PD2PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
1	PD1PR	Pin state	R	
0	PD0PR	Pin state	R	

28.2.9 Port E Data Register L (PEDRL)

PEDRL is a 16-bit readable/writable register that stores port E data. The PE13DR to PE0DR bits correspond to the PE13 to PE0 pins, respectively.

When a pin function is general output, if a value is written to PEDRL, that value is output from the pin, and if PEDRL is read, the register value is returned regardless of the pin state.

When a pin function is general input, if PEDRL is read, the pin state, not the register value, is returned directly. If a value is written to PEDRL, although that value is written into PEDRL, it does not affect the pin state. Table 28.6 summarizes PEDRL read/write operation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value:	0	0	0	*	0	*	0	*	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Depends on the state of the external pin.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
13	PE13DR	0	R/W	See table 28.6.
12	PE12DR	Pin state	R	
11	PE11DR	0	R/W	
10	PE10DR	Pin state	R	
9	PE9DR	0	R/W	
8	PE8DR	Pin state	R	
7	PE7DR	0	R/W	
6	PE6DR	0	R/W	
5	PE5DR	0	R/W	
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

Table 28.6 Port E Data Register L (PEDRL) Read/Write Operation

- Bits 13, 11, 9, 7 to 0 of PEDRL

PEIORL	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PEDRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PEDRL, but it has no effect on pin state
1	General output	PEDRL value	Value written is output from pin
	Other than general output	PEDRL value	Can write to PEDRL, but it has no effect on pin state

- Bits 12, 10, 8 of PEDRL

Pin Function	Read Operation	Write Operation
General input	Pin state	Ignored
Other than general output	Pin state	Ignored

28.2.10 Port E Port Register L (PEPRL)

PEPRL is a 16-bit read-only register, in which the PE13PR to PE0PR bits correspond to the PE13 to PE0 pins, respectively. PEPRL always returns the states of the pins regardless of the PFC setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR	PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR
Initial value:	0	0	PE13	PE12	PE11	PE10	PE9	PE8	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
13	PE13PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
12	PE12PR	Pin state	R	
11	PE11PR	Pin state	R	
10	PE10PR	Pin state	R	
9	PE9PR	Pin state	R	
8	PE8PR	Pin state	R	
7	PE7PR	Pin state	R	
6	PE6PR	Pin state	R	
5	PE5PR	Pin state	R	
4	PE4PR	Pin state	R	
3	PE3PR	Pin state	R	
2	PE2PR	Pin state	R	
1	PE1PR	Pin state	R	
0	PE0PR	Pin state	R	

28.2.11 Port F Data Register L (PFDRL)

PFDRL is a 16-bit readable/writable register that stores port F data. The PF4DR to PF0DR bits correspond to the PF4 to PF0 pins, respectively.

When a pin function is general output, if a value is written to PFDRL, that value is output from the pin, and if PFDRL is read, the register value is returned regardless of the pin state.

When a pin function is general input, if PFDRL is read, the pin state, not the register value, is returned directly. If a value is written to PFDRL, although that value is written into PFDRL, it does not affect the pin state. Table 28.7 summarizes PFDRL read/write operations.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
4	PF4DR	0	R/W	See table 28.7.
3	PF3DR	0	R/W	
2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

Table 28.7 Port F Data Register L (PFDRL) Read/Write Operation

- Bits 4 to 0 of PFDRL

PFIORL	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PFDRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PFDRL, but it has no effect on pin state
1	General output	PFDRL value	Value written is output from pin
	Other than general output	PFDRL value	Can write to PFDRL, but it has no effect on pin state

28.2.12 Port F Port Register L (PFPRL)

PFPRL is a 16-bit read-only register, in which the PF4PR to PF0PR bits correspond to the PF4 to PF0 pins, respectively. PFPRL always returns the states of the pins regardless of the PFC setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	PF4PR	PF3PR	PF2PR	PF1PR	PF0PR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	PF4	PF3	PF2	PF1	PF0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PF4PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
3	PF3PR	Pin state	R	
2	PF2PR	Pin state	R	
1	PF1PR	Pin state	R	
0	PF0PR	Pin state	R	

28.2.13 Port G Data Register L (PGDRL)

PGDRL is a 16-bit read-only register that stores port G data. The PG7DR to PG0DR bits correspond to the PG7 to PG0 pins, respectively. The general input function of the PG7 to PG0 pins is enabled only when the A/D and D/A converters are halted.

Writing to these bits is ignored, and therefore does not affect the pin state. If these bits are read, the pin state, not the bit value, is directly returned. However, a fixed value is returned for pins for which the A/D or D/A converter function is selected. Table 28.8 summarizes PGDRL read/write operation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PG7DR	PG6DR	PG5DR	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
Initial value:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Depends on the state of the external pin.

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	PG7DR	Pin state	R/W	See table 28.8.
6	PG6DR	Pin state	R/W	
5	PG5DR	Pin state	R/W	
4	PG4DR	Pin state	R/W	
3	PG3DR	Pin state	R/W	
2	PG2DR	Pin state	R/W	
1	PG1DR	Pin state	R/W	
0	PG0DR	Pin state	R/W	

Table 28.8 Port G Data Register L (PGDRL) Read/Write Operation

- Bits 7 to 0 of PGDRL

Pin Function	Read Operation	Write Operation
General input or function input other than ANn and DAm	Pin state	Ignored (no effect on pin state)
ANn input or DAm output	Fixed value	Ignored (no effect on pin state)

[Legend]

n = 7 to 0

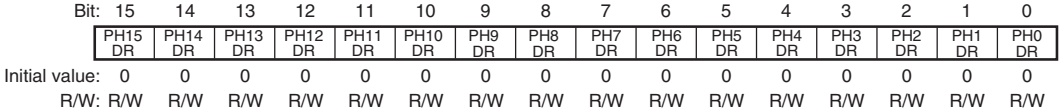
m = 1, 0

28.2.14 Port H Data Register L (PHDRL)

PHDRL is a 16-bit readable/writable register that stores port H data. The PH15DR to PH0DR bits correspond to the PH15 to PH0 pins, respectively.

When a pin function is general output, if a value is written to PHDRL, that value is output from the pin, and if PHDRL is read, the register value is returned regardless of the pin state.

When a pin function is general input, if PHDRL is read, the pin state, not the register value, is returned directly. If a value is written to PHDRL, although that value is written into PHDRL, it does not affect the pin state. Table 28.9 summarizes PHDRL read/write operations.



Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH15 DR	PH14 DR	PH13 DR	PH12 DR	PH11 DR	PH10 DR	PH9 DR	PH8 DR	PH7 DR	PH6 DR	PH5 DR	PH4 DR	PH3 DR	PH2 DR	PH1 DR	PH0 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PH15DR	0	R/W	See table 28.9.
14	PH14DR	0	R/W	
13	PH13DR	0	R/W	
12	PH12DR	0	R/W	
11	PH11DR	0	R/W	
10	PH10DR	0	R/W	
9	PH9DR	0	R/W	
8	PH8DR	0	R/W	
7	PH7DR	0	R/W	
6	PH6DR	0	R/W	
5	PH5DR	0	R/W	
4	PH4DR	0	R/W	
3	PH3DR	0	R/W	
2	PH2DR	0	R/W	
1	PH1DR	0	R/W	
0	PH0DR	0	R/W	

Table 28.9 Port H Data Register L (PHDRL) Read/Write Operation

- Bits 15 to 0 of PHDRL

PHIORL	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PHDRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PHDRL, but it has no effect on pin state
1	General output	PHDRL value	Value written is output from pin
	Other than general output	PHDRL value	Can write to PHDRL, but it has no effect on pin state

28.2.15 Port H Port Register L (PHPRL)

PHPRL is a 16-bit read-only register, in which the PH15PR to PH0PR bits correspond to the PH15 to PH0 pins, respectively. PHPRL always returns the states of the pins regardless of the PFC setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH15PR	PH14PR	PH13PR	PH12PR	PH11PR	PH10PR	PH9PR	PH8PR	PH7PR	PH6PR	PH5PR	PH4PR	PH3PR	PH2PR	PH1PR	PH0PR
Initial value:	PH15	PH14	PH13	PH12	PH11	PH10	PH9	PH8	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PH15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PH14PR	Pin state	R	
13	PH13PR	Pin state	R	
12	PH12PR	Pin state	R	
11	PH11PR	Pin state	R	
10	PH10PR	Pin state	R	
9	PH9PR	Pin state	R	
8	PH8PR	Pin state	R	
7	PH7PR	Pin state	R	
6	PH6PR	Pin state	R	
5	PH5PR	Pin state	R	
4	PH4PR	Pin state	R	
3	PH3PR	Pin state	R	
2	PH2PR	Pin state	R	
1	PH1PR	Pin state	R	
0	PH0PR	Pin state	R	

28.2.16 Port J Data Register L (PJDRL)

PJDRL is a 16-bit readable/writable register that stores port J data. The PJ12DR to PJ0DR bits correspond to the PJ12 to PJ0 pins, respectively.

When a pin function is general output, if a value is written to PJDRL, that value is output from the pin, and if PJDRL is read, the register value is returned regardless of the pin state.

When a pin function is general input, if PJDRL is read, the pin state, not the register value, is returned directly. If a value is written to PJDRL, although that value is written into PJDRL, it does not affect the pin state. Table 28.10 summarizes PJDRL read/write operations.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PJ12DR	PJ11DR	PJ10DR	PJ9DR	PJ8DR	PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	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	PJ12DR	0	R/W	See table 28.10.
11	PJ11DR	0	R/W	
10	PJ10DR	0	R/W	
9	PJ9DR	0	R/W	
8	PJ8DR	0	R/W	
7	PJ7DR	0	R/W	
6	PJ6DR	0	R/W	
5	PJ5DR	0	R/W	
4	PJ4DR	0	R/W	
3	PJ3DR	0	R/W	
2	PJ2DR	0	R/W	
1	PJ1DR	0	R/W	
0	PJ0DR	0	R/W	

Table 28.10 Port J Data Register L (PJDRL) Read/Write Operations

- Bits 12 to 0 of PJDRL

PJIORL	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PJDRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PJDRL, but it has no effect on pin state
1	General output	PJDRL value	Value written is output from pin
	Other than general output	PJDRL value	Can write to PJDRL, but it has no effect on pin state

28.2.17 Port J Port Register L (PJPRL)

PJPRL is a 16-bit read-only register, in which the PJ12PR to PJ0PR bits correspond to the PJ12 to PJ0 pins, respectively. PJPRL always returns the states of the pins regardless of the PFC setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PJ12PR	PJ11PR	PJ10PR	PJ9PR	PJ8PR	PJ7PR	PJ6PR	PJ5PR	PJ4PR	PJ3PR	PJ2PR	PJ1PR	PJ0PR
Initial value:	0	0	0	PJ12	PJ11	PJ10	PJ9	PJ8	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
12	PJ12PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
11	PJ11PR	Pin state	R	
10	PJ10PR	Pin state	R	
9	PJ9PR	Pin state	R	
8	PJ8PR	Pin state	R	
7	PJ7PR	Pin state	R	
6	PJ6PR	Pin state	R	
5	PJ5PR	Pin state	R	
4	PJ4PR	Pin state	R	
3	PJ3PR	Pin state	R	
2	PJ2PR	Pin state	R	
1	PJ1PR	Pin state	R	
0	PJ0PR	Pin state	R	

28.2.18 Port K Data Register L (PKDRL)

PKDRL is a 16-bit readable/writable register that stores port K data. The PK1DR and PK0DR bits correspond to the PK1 and PK0 pins, respectively.

When a pin function is general output, if a value is written to PKDRL, that value is output from the pin, and if PKDRL is read, the register value is returned regardless of the pin state.

When a pin function is general input, if PKDRL is read, the pin state, not the register value, is returned directly. If a value is written to PKDRL, although that value is written into PKDRL, it does not affect the pin state. Table 28.11 summarizes PKDRL read/write operations.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PK1DR	PK0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PK1DR	0	R/W	See table 28.11.
0	PK0DR	0	R/W	

Table 28.11 Port K Data Register L (PKDRL) Read/Write Operations

- Bits 1 and 0 of PKDRL

PKIORL	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PKDRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PKDRL, but it has no effect on pin state
1	General output	PKDRL value	Value written is output from pin
	Other than general output	PKDRL value	Can write to PKDRL, but it has no effect on pin state

28.2.19 Port K Port Register L (PKPRL)

PKPRL is a 16-bit read-only register, in which the PK1PR and PK0PR bits correspond to the PK1 and PK0 pins, respectively. PKPRL always returns the states of the pins regardless of the PFC setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PK1PR	PK0PR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PK1	PK0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
1	PK1PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
0	PK0PR	Pin state	R	

28.3 Usage Notes

When the PFC has been configured to select the following pin functions, the pin state cannot be read from the data registers or port registers.

- D31 to D16 (data bus)

Section 29 On-Chip RAM

This LSI has an on-chip high-speed RAM, which achieves fast access, and an on-chip RAM for data retention, which can retain data even in deep standby mode. These memory units can store instructions or data.

The memory operation and writing operation of the on-chip high-speed RAM can be enabled or disabled through the RAM enable bits and RAM write enable bits.

For the on-chip RAM for data retention, it is possible to specify whether to retain data in deep standby mode on a page-by-page basis.

29.1 Features

- Pages

  On-chip high-speed RAM0 consists of four pages (pages 0, 1, 2, and 3) and on-chip high-speed RAM1 consists of two pages (pages 0 and 1) and each of these pages has a size of 16 Kbytes.

  The on-chip RAM for data retention consists of four pages and each of these pages has a size of 4 Kbytes.

- Memory map

  The on-chip high-speed RAMs are allocated in the address space shown in tables 29.1 and 29.2. The on-chip RAM for data retention is allocated in the address space shown in table 29.3.

  When a common area for CPU0 and CPU1 is placed on the on-chip high-speed RAMs and the area is exclusively accessed by the TAS.B instruction, the on-chip high-speed RAMs should be accessed through the address space shown in table 29.2.

Table 29.1 Address Spaces of On-Chip High-Speed RAM

Page	Address
Page 0 of on-chip high-speed RAM0	H'FFF80000 to H'FFF83FFF
Page 1 of on-chip high-speed RAM0	H'FFF84000 to H'FFF87FFF
Page 2 of on-chip high-speed RAM0	H'FFF88000 to H'FFF8BFFF
Page 3 of on-chip high-speed RAM0	H'FFF8C000 to H'FFF8FFFF
Page 0 of on-chip high-speed RAM1	H'FFFA0000 to H'FFFA3FFF
Page 1 of on-chip high-speed RAM1	H'FFFA4000 to H'FFFA7FFF

Table 29.2 Address Spaces of On-Chip High-Speed RAM (Shadow Spaces)

Page	Address
Page 0 of on-chip high-speed RAM0	H'FFD80000 to H'FFD83FFF
Page 1 of on-chip high-speed RAM0	H'FFD84000 to H'FFD87FFF
Page 2 of on-chip high-speed RAM0	H'FFD88000 to H'FFD8BFFF
Page 3 of on-chip high-speed RAM0	H'FFD8C000 to H'FFD8FFFF
Page 0 of on-chip high-speed RAM1	H'FFDA0000 to H'FFDA3FFF
Page 1 of on-chip high-speed RAM1	H'FFDA4000 to H'FFDA7FFF

Table 29.3 Address Spaces of On-Chip RAM for Data Retention

Page	Address
Page 0	H'FF800000 to H'FF800FFF
Page 1	H'FF801000 to H'FF801FFF
Page 2	H'FF802000 to H'FF802FFF
Page 3	H'FF803000 to H'FF803FFF

- Port

  On-chip high-speed RAM0 is connected to the CPU0 instruction fetch bus, CPU0 memory access bus, and on-chip high-speed RAM0 access bus. When CPU0 accesses on-chip high-speed RAM0 through the address spaces shown in table 29.1, the CPU0 instruction fetch bus or CPU0 memory access bus is used. When CPU0 accesses on-chip high-speed RAM0 through the address spaces shown in table 29.2, the on-chip high-speed RAM0 access bus is used. When CPU1 or DMAC accesses on-chip high-speed RAM0, the on-chip high-speed RAM0 access bus is used in access through the both address spaces shown in table 29.1 and table 29.2.

  On-chip high-speed RAM1 is connected to the CPU1 instruction fetch bus, CPU1 memory access bus, and on-chip high-speed RAM1 access bus. When CPU1 accesses on-chip high-speed RAM1 through the address space shown in table 29.1, the CPU1 instruction fetch bus or CPU1 memory access bus is used. When CPU1 accesses on-chip high-speed RAM1 through the address space shown in table 29.2, the on-chip high-speed RAM1 access bus is used. When CPU0 or DMAC accesses on-chip high-speed RAM1, the on-chip high-speed RAM1 access bus is used in access through the both address spaces shown in table 29.1 and table 29.2.

  Each page of the on-chip RAM for data retention has one read and write port and is connected to the peripheral bus.

- Priority

  When the same page of the on-chip high-speed RAM is simultaneously accessed from different buses, the access is controlled based on the priority. The priority order is on-chip high-speed RAM access bus > memory access bus > instruction fetch bus.

29.2 Usage Notes

29.2.1 Page Conflict

When the same page of the on-chip high-speed RAM is simultaneously accessed from different buses, a page conflict occurs. Although each access is completed correctly, such a conflict degrades the memory access speed. Therefore, it is advisable to provide software measures to prevent such a conflict. Different pages, instead of the same page, can be simultaneously accessed from different buses.

29.2.2 RAME Bit and RAMWE Bit

Before clearing the RAME bits of SYSCR1, SYSCR3, SYSCR5, SYSCR7, SYSCR9, and SYSCR11 to 0 and the RAMWE bits of SYSCR2, SYSCR4, SYSCR6, SYSCR8, SYSCR10, and SYSCR12 to 0, be sure to read and write the arbitrarily-selected same address in each page. Otherwise, the data last written to the corresponding page may not be actually written to the RAM.

```
//For page 0 of on-chip RAM0
  MOV.L   #H'FFF80000, R0
  MOV.L   @R0, R1
  MOV.L   R1, @R0

//For page 1 of on-chip RAM0
  MOV.L   #H'FFF84000, R0
  MOV.L   @R0, R1
  MOV.L   R1, @R0

//For page 2 of on-chip RAM0
  MOV.L   #H'FFF88000, R0
  MOV.L   @R0, R1
  MOV.L   R1, @R0

//For page 3 of on-chip RAM0
  MOV.L   #H'FFF8C000, R0
  MOV.L   @R0, R1
  MOV.L   R1, @R0

//For page 0 of on-chip RAM1
  MOV.L   #H'FFFA0000, R0
  MOV.L   @R0, R1
  MOV.L   R1, @R0

//For page 1 of on-chip RAM1
  MOV.L   #H'FFFA4000, R0
  MOV.L   @R0, R1
  MOV.L   R1, @R0
```

Figure 29.1 Examples of Read/Write

29.2.3 Data Retention

Data in the on-chip high-speed RAM and including on-chip data retention RAM are retained in the states other than power-on reset and deep standby mode. In power-on reset and deep standby mode, these RAMs operate as described below.

(1) Power-on Reset

(a) On-Chip High-Speed RAM

Data are retained on a power-on reset by disabling the setting of either the RAME or RAMWE bit.

Data are not retained when the setting of the RAME and RAMWE bits are both enabled.

(b) On-Chip Data Retention RAM

Data are not retained.

(2) Deep Standby Mode

(a) On-Chip High-Speed RAM

Data are not retained.

(b) On-Chip Data Retention RAM

Data are retained in deep standby mode by enabling the setting of the RRAMKP bit, excluding the case that deep standby mode is canceled by power-on reset. In the case that deep standby mode is canceled by interrupt or manual reset pins, power-on reset exception handling is executed, but the data are retained.

Section 30 Power-Down Modes

This LSI supports single-processor mode, dual-sleep mode, software standby mode, deep standby mode, and module standby function as power-down modes. In power-down modes, functions of the CPU, clocks, on-chip memory, or part of on-chip peripheral modules are halted or the power supply is turned off, through which power consumption is reduced. These modes can be exited by a reset or interrupt.

30.1 Power-Down Modes

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

1. Dual-processor mode
2. Single-processor mode (single-processor 0 mode, single-processor 1 mode)
3. Dual-sleep mode
4. Software standby mode
5. Deep standby mode
6. Module standby function

Table 30.1 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 30.1 States of Power-Down Modes

Power-Down Mode	Transition Conditions	State*1												Means of Exit
		CPG	CPU0	CPU0 Register	High-Speed On-Chip RAM0 Cache Memory 0	CPU1	CPU1 Register	High-Speed On-Chip RAM0 Cache Memory 1	On-Chip RAM (for Data Retention)	On-Chip Peripheral Modules	RTC	Power Supply	External Memory	
Dual-processor	—	Runs	Runs	Held	Runs	Runs	Held	Runs	Runs	Selective*2	Selective*2*3	Runs	Auto-refreshing	—
Single-processor 0	CPU1 executes SLEEP instruction in dual-processor mode	Runs	Runs	Held	Runs	Halts	Held	Runs	Runs	Selective*2	Selective*2*3	Runs	Auto-refreshing	• Interrupt • Manual reset • Power-on reset • CPU address error
Single-processor 1	CPU0 executes SLEEP instruction with STBY bit in STBCR1 cleared to 0 in dual-processor mode	Runs	Halts	Held	Runs	Runs	Held	Runs	Runs	Selected*2	Selective*2*3	Runs	Auto-refreshing	• Interrupt • Manual reset • Power-on reset • CPU address error
Dual-sleep	• CPU0 executes SLEEP instruction with STBY bit in STBCR1 cleared to 0 in single-processor 0 mode • CPU1 executes SLEEP instruction in single-processor 1 mode	Runs	Halts	Held	Runs	Halts	Held	Runs	Runs	Selective*2	Selective*2*3	Runs	Auto-refreshing	• Interrupt • Manual reset • Power-on reset • CPU address error
Software standby	CPU0 executes SLEEP instruction with STBY bit in STBCR1 set to 1 and DEEP bit to 0 in single-processor 0 mode	Halts	Halts	Held	Halts (contents are held*6*7)	Halts	Held	Halts (contents are held*6*8)	Halts (contents are held*8)	Halts	Runs*3	Runs	Should be put into self-refreshing mode	• NMI interrupt • IRQ interrupt • Manual reset • Power-on reset
Deep standby	CPU0 executes SLEEP instruction with STBY bit in STBCR1 set to 1 and DEEP bit to 1 in single-processor 0 mode	Halts	Halts	Halts	Halts (contents are not held)	Halts	Halts	Halts (contents are not held)	Halts (contents are held*4)	Halts	Runs*3	Halts	Should be put into self-refreshing mode	• NMI interrupt*5 • IRQ interrupt*5 • Manual reset*5 • Power-on reset*5

Notes: 1. The pin state is retained or set to high impedance. For details, see appendix A, Pin States.

2. By specifying the module standby function, individual on-chip peripheral modules (including the RTC) can be halted. To specify the module standby function, set the corresponding MSTP bit in STBCR2 to STBCR7 to 1. To cancel the module standby function, clear the MSTP bit to 0. For the H-UDI and UBC, the module standby function can also be canceled by power-on reset.

3. RTC operates when the START bit in the RCR2 register is set to 1. For details, see section 15, Realtime Clock (RTC). When deep standby mode is canceled by a power-on reset, the running state cannot be retained. Make the initial setting for the realtime clock again.

4. Setting the bits RRAMKP3 to RRAMKP0 in the RRAMKP register to 1 enables to retain the data in the corresponding area on the on-chip RAM (for data retention) during the transition to deep standby mode. However, the stored contents are initialized when deep standby mode is canceled by a power-on reset.

5. Deep standby mode can be canceled by an interrupt (NMI or IRQ) or a reset (manual reset or power-on reset). However, when deep standby mode is canceled by the NMI interrupt or IRQ interrupt, power-on reset exception handling is executed instead of interrupt exception handling. The power-on reset exception handling is executed also in the cancellation of deep standby mode by manual reset.

6. The stored contents are initialized when software standby mode is canceled by a power-on reset.
7. Clearing the RAME bits in registers SYSCR1, SYSCR3, and SYSCR5 or the RAMWE bits in registers SYSCR2, SYSCR4, and SYSCR6 to 0 (disabled) enables retention of the data in high-speed on-chip RAM0 when software standby mode is canceled by a power-on reset.
8. Clearing the RAME bits in registers SYSCR7, SYSCR9, and SYSCR11 or the RAMWE bits in registers SYSCR8, SYSCR10, and SYSCR12 to 0 (disabled) enables retention of the data in high-speed on-chip RAM1 when software standby mode is canceled by a power-on reset.

30.2 Register Descriptions

The following registers are used in power-down modes.

Table 30.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Standby control register 1	STBCR1	R/W	H'00	H'FFFE0014	8
Standby control register 2	STBCR2	R/W	H'00	H'FFFE0018	8
Standby control register 3	STBCR3	R/W	H'FE	H'FFFE0400	8
Standby control register 4	STBCR4	R/W	H'FF	H'FFFE0402	8
Standby control register 5	STBCR5	R/W	H'FF	H'FFFE0404	8
Standby control register 6	STBCR6	R/W	H'FF	H'FFFE0406	8
Standby control register 7	STBCR7	R/W	H'FF	H'FFFE0408	8
System control register 1	SYSCR1	R/W	H'FF	H'FFFE0480	8
System control register 2	SYSCR2	R/W	H'FF	H'FFFE0482	8
System control register 3	SYSCR3	R/W	H'FF	H'FFFE0484	8
System control register 4	SYSCR4	R/W	H'FF	H'FFFE0486	8
System control register 5	SYSCR5	R/W	H'FF	H'FFFE0488	8
System control register 6	SYSCR6	R/W	H'FF	H'FFFE048A	8
System control register 7	SYSCR7	R/W	H'FF	H'FFFE04A0	8
System control register 8	SYSCR8	R/W	H'FF	H'FFFE04A2	8
System control register 9	SYSCR9	R/W	H'FF	H'FFFE04A4	8
System control register 10	SYSCR10	R/W	H'FF	H'FFFE04A6	8
System control register 11	SYSCR11	R/W	H'FF	H'FFFE04A8	8
System control register 12	SYSCR12	R/W	H'FF	H'FFFE04AA	8
Software reset control register	SWRSTCR	R/W	H'00	H'FFFE0440	8
High-impedance control register	HIZCR	R/W	H'00	H'FFFE0442	8
CPU0 mode status register	C0MSR	R	H'00	H'FFFE0040	8
CPU1 mode status register	C1MSR	R	H'00	H'FFFE0042	8
Data retention on-chip RAM area specification register	RRAMKP	R/W	H'00	H'FFFE0C00	8
Deep standby control register	DSCTR	R/W	H'00	H'FFFE0C02	8
Deep standby cancel source select register	DSSSR	R/W	H'0000	H'FFFE0C04	16
Deep standby cancel source flag register	DSFR	R/W	H'0000	H'FFFE0C08	16

30.2.1 Standby Control Register 1 (STBCR1)

STBCR1 is an 8-bit readable/writable register that specifies the states of power-down modes.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	STBY	DEEP	SLP ERE	AXTALE	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Software Standby, Deep Standby
6	DEEP	0	R/W	Specifies transition to software standby mode or deep standby mode. 0x: Execution of SLEEP instruction by CPU0 puts CPU0 into sleep mode. 10: Execution of SLEEP instruction by CPU0 puts the chip in software standby mode. 11: Execution of SLEEP instruction by CPU0 puts the chip in deep standby mode.
5	SLPERE	0	R/W	Sleep Error Enable Enables or disables to generate a sleep error exception. After a sleep error exception is generated with this bit set to 1, be sure to clear this bit to 0 in the interrupt exception handling routine. 0: Sleep error exception is disabled. 1: Sleep error exception is enabled.
4	AXTALE	0	R/W	AUDIO_X1 Clock Control Controls the function of AUDIO_X1 pin. 0: Runs the on-chip crystal oscillator/enables the external clock input. 1: Halts the on-chip crystal oscillator/disables the external clock input.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

[Legend]
x: Don't care

30.2.2 Standby Control Register 2 (STBCR2)

STBCR2 is an 8-bit readable/writable register that controls the operation of individual modules.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP27	-	-	MSTP24	MSTP23	MSTP22	MSTP21	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP27	0	R/W	Module Stop 27 When set to 1, the clock supply to the H-UDI is halted. 0: H-UDI runs. 1: Clock supply to H-UDI is halted.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	MSTP24	0	R/W	Module Stop 24 When set to 1, the clock supply to FPU0 is halted. After being set to 1, this bit cannot be cleared by writing 0. This means that the clock supply the FPU0 cannot be restarted by clearing this bit to 0. To restart the clock supply to the FPU0, reset the LSI by a power-on reset. 0: FPU0 runs. 1: Clock supply to FPU0 is halted.
3	MSTP23	0	R/W	Module Stop 23 When set to 1, the clock supply to FPU1 is halted. After being set to 1, this bit cannot be cleared by writing 0. This means that the clock supply the FPU1 cannot be restarted by clearing this bit to 0. To restart the clock supply to the FPU1, reset the LSI by a power-on reset. 0: FPU1 runs. 1: Clock supply to FPU1 is halted.

Bit	Bit Name	Initial Value	R/W	Description
2	MSTP22	0	R/W	Module Stop 22 When set to 1, the clock supply to UBC0 is halted. 0: UBC0 runs. 1: Clock supply to UBC0 is halted.
1	MSTP21	0	R/W	Module Stop 21 When set to 1, the clock supply to UBC1 is halted. 0: UBC1 runs. 1: Clock supply to UBC1 is halted.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

30.2.3 Standby Control Register 3 (STBCR3)

STBCR3 is an 8-bit readable/writable register that controls the operation of modules.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP37	-	MSTP35	-	-	MSTP32	MSTP31	MSTP30
Initial value:	1	1	1	1	1	1	1	0
R/W:	R/W	R	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP37	1	R/W	Module Stop 37 When set to 1, the clock supply to the ATAPI is halted. 0: ATAPI runs. 1: Clock supply to ATAPI is halted.
6	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

Bit	Bit Name	Initial Value	R/W	Description
5	MSTP35	1	R/W	Module Stop 35 When set to 1, the clock supply to the MTU2 is halted. 0: MTU2 runs. 1: Clock supply to MTU2 is halted.
4, 3	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
2	MSTP32	1	R/W	Module Stop 32 When set to 1, the clock supply to the ADC is halted. 0: ADC runs. 1: Clock supply to ADC is halted.
1	MSTP31	1	R/W	Module Stop 31 When set to 1, the clock supply to the DAC is halted. 0: DAC runs. 1: Clock supply to DAC is halted.
0	MSTP30	0	R/W	Module Stop 30 When set to 1, the clock supply to the RTC is halted. 0: RTC runs. 1: Clock supply to RTC is halted.

30.2.4 Standby Control Register 4 (STBCR4)

STBCR4 is an 8-bit readable/writable register that controls the operation of modules.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP47	MSTP46	MSTP45	MSTP44	MSTP43	MSTP42	-	-
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP47	1	R/W	Module Stop 47 When set to 1, the clock supply to the SCIF0 is halted. 0: SCIF0 runs. 1: Clock supply to SCIF0 is halted.
6	MSTP46	1	R/W	Module Stop 46 When set to 1, the clock supply to the SCIF1 is halted. 0: SCIF1 runs. 1: Clock supply to SCIF1 is halted.
5	MSTP45	1	R/W	Module Stop 45 When set to 1, the clock supply to the SCIF2 is halted. 0: SCIF2 runs. 1: Clock supply to SCIF2 is halted.
4	MSTP44	1	R/W	Module Stop 44 When set to 1, the clock supply to the SCIF3 is halted. 0: SCIF3 runs. 1: Clock supply to SCIF3 is halted.
3	MSTP43	1	R/W	Module Stop 43 When set to 1, the clock supply to the SCIF4 is halted. 0: SCIF4 runs. 1: Clock supply to SCIF4 is halted.

Bit	Bit Name	Initial Value	R/W	Description
2	MSTP42	1	R/W	Module Stop 42 When set to 1, the clock supply to the SCIF5 is halted. 0: SCIF5 runs. 1: Clock supply to SCIF5 is halted.
1, 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

30.2.5 Standby Control Register 5 (STBCR5)

STBCR5 is an 8-bit readable/writable register that controls the operation of modules.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP57	MSTP56	MSTP55	MSTP54	MSTP53	MSTP52	-	-
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP57	1	R/W	Module Stop 57 When set to 1, the clock supply to the IIC3_0 is halted. 0: IIC3_0 runs. 1: Clock supply to IIC3_0 is halted.
6	MSTP56	1	R/W	Module Stop 56 When set to 1, the clock supply to the IIC3_1 is halted. 0: IIC3_1 runs. 1: Clock supply to IIC3_1 is halted.
5	MSTP55	1	R/W	Module Stop 55 When set to 1, the clock supply to the IIC3_2 is halted. 0: IIC3_2 runs. 1: Clock supply to IIC3_2 is halted.

Bit	Bit Name	Initial Value	R/W	Description
4	MSTP54	1	R/W	Module Stop 54 When set to 1, the clock supply to the IIC3_3 is halted. 0: IIC3_3 runs. 1: Clock supply to IIC3_3 is halted.
3	MSTP53	1	R/W	Module Stop 53 When set to 1, the clock supply to the RCAN0 is halted. 0: RCAN0 runs. 1: Clock supply to RCAN0 is halted.
2	MSTP52	1	R/W	Module Stop 52 When set to 1, the clock supply to the RCAN1 is halted. 0: RCAN1 runs. 1: Clock supply to RCAN1 is halted.
1, 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

30.2.6 Standby Control Register 6 (STBCR6)

STBCR6 is an 8-bit readable/writable register that controls the operation of modules.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP67	MSTP66	MSTP65	MSTP64	MSTP63	MSTP62	-	-
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP67	1	R/W	Module Stop 67 When set to 1, the clock supply to the SSIF0 is halted. 0: SSIF0 runs. 1: Clock supply to SSIF0 is halted.

Bit	Bit Name	Initial Value	R/W	Description
6	MSTP66	1	R/W	Module Stop 66 When set to 1, the clock supply to the SSIF1 is halted. 0: SSIF1 runs. 1: Clock supply to SSIF1 is halted.
5	MSTP65	1	R/W	Module Stop 65 When set to 1, the clock supply to the SSIF2 is halted. 0: SSIF2 runs. 1: Clock supply to SSIF2 is halted.
4	MSTP64	1	R/W	Module Stop 64 When set to 1, the clock supply to the SSIF3 is halted. 0: SSIF3 runs. 1: Clock supply to SSIF3 is halted.
3	MSTP63	1	R/W	Module Stop 63 When set to 1, the clock supply to the SSIF4 is halted. 0: SSIF4 runs. 1: Clock supply to SSIF4 is halted.
2	MSTP62	1	R/W	Module Stop 62 When set to 1, the clock supply to the SSIF5 is halted. 0: SSIF5 runs. 1: Clock supply to SSIF5 is halted.
1, 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

30.2.7 Standby Control Register 7 (STBCR7)

STBCR7 is an 8-bit readable/writable register that controls the operation of modules.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP77	MSTP76	-	MSTP74	MSTP73	MSTP72	MSTP71	MSTP70
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP77	1	R/W	Module Stop 77 When set to 1, the clock supply to the CMT0/CMT1 is halted. 0: CMT0/CMT1 run. 1: Clock supply to CMT0/CMT1 is halted.
6	MSTP76	1	R/W	Module Stop 76 When set to 1, the clock supply to the CMT2/CMT3 is halted. 0: CMT2/CMT3 run. 1: Clock supply to CMT2/CMT3 is halted.
5	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
4	MSTP74	1	R/W	Module Stop 74 When set to 1, the clock supply to the FLCTL is halted. 0: FLCTL runs. 1: Clock supply to FLCTL is halted.
3	MSTP73	1	R/W	Module Stop 73 When set to 1, the clock supply to the SSU0 is halted. 0: SSU0 runs. 1: Clock supply to SSU0 is halted.

Bit	Bit Name	Initial Value	R/W	Description
2	MSTP72	1	R/W	Module Stop 72 When set to 1, the clock supply to the SSU1 is halted. 0: SSU1 runs. 1: Clock supply to SSU1 is halted.
1	MSTP71	1	R/W	Module Stop 71 When set to 1, the clock supply to the Video IN/2DG/Video OUT is halted. 0: Video IN/2DG/Video OUT run. 1: Clock supply to Video IN/2DG/Video OUT is halted.
0	MSTP70	1	R/W	Module Stop 70 When set to 1, the clock supply to the USB is halted. 0: USB runs. 1: Clock supply to USB is halted.

30.2.8 System Control Register 1 (SYSCR1)

SYSCR1 is an 8-bit readable/writable register that enables or disables access (read/write) from CPU0 to each page of the high-speed on-chip RAM0.

Setting the RAMEn (n = 0 to 3) bit in SYSCR1 to 1 enables access to page n. Clearing the RAMEn bit to 0 disables access to page n. In this case, an undefined value is returned when reading data or fetching an instruction from page n, and writing to page n is ignored. The initial value of the RAMEn bit is 1.

When clearing the RAMEn bit to 0, be sure to execute instructions to read from and write to the same arbitrary address in page n before clearing the RAMEn bit. If not executed, the data last written to page n may not be actually written to the high-speed on-chip RAM.

SYSCR1 should be set by the program that is placed in a space other than the high-speed on-chip RAM space. Furthermore, an instruction to read SYSCR1 should be located immediately after the instruction to write to SYSCR1. Otherwise, normal access to the high-speed on-chip RAM is not guaranteed.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RAME3	RAME2	RAME1	RAME0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	RAME3	1	R/W	RAM Enable 3 (page 3 of high-speed on-chip RAM0*) 0: Access to page 3 is disabled. 1: Access to page 3 is enabled.
2	RAME2	1	R/W	RAM Enable 2 (page 2 of high-speed on-chip RAM0*) 0: Access to page 2 is disabled. 1: Access to page 2 is enabled.
1	RAME1	1	R/W	RAM Enable 1 (page 1 of high-speed on-chip RAM0*) 0: Access to page 1 is disabled. 1: Access to page 1 is enabled.
0	RAME0	1	R/W	RAM Enable 0 (page 0 of high-speed on-chip RAM0*) 0: Access to page 0 is disabled. 1: Access to page 0 is enabled.

Note: * For the addresses of each page, see section 29, On-Chip RAM.

30.2.9 System Control Register 2 (SYSCR2)

SYSCR2 is an 8-bit readable/writable register that enables or disables writing from CPU0 to each page of the high-speed on-chip RAM0.

Setting the RAMWEn (n = 0 to 3) bit in SYSCR2 to 1 enables writing to page n. When the RAMWEn bit is cleared to 0, writing to page n is ignored. The initial value of the RAMWEn bit is 1.

When clearing the RAMWEn bit to 0, be sure to execute an instruction to read from and write to the same arbitrary address in page n before setting the RAMWEn bit. If not executed, the data last written to page n may not be written to the high-speed on-chip RAM.

Set SYSCR2 using the program that is placed in a space other than the high-speed on-chip RAM space. Furthermore, an instruction to read SYSCR2 should be located immediately after the instruction to write to SYSCR2. Otherwise, normal access to the high-speed on-chip RAM is not guaranteed.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RAM WE3	RAM WE2	RAM WE1	RAM WE0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	RAMWE3	1	R/W	RAM Write Enable 3 (page 3 of high-speed on-chip RAM0*) 0: Write to page 3 is disabled 1: Write to page 3 is enabled
2	RAMWE2	1	R/W	RAM Write Enable 2 (page 2 of high-speed on-chip RAM0*) 0: Write to page 2 is disabled 1: Write to page 2 is enabled
1	RAMWE1	1	R/W	RAM Write Enable 1 (page 1 of high-speed on-chip RAM0*) 0: Write to page 1 is disabled 1: Write to page 1 is enabled
0	RAMWE0	1	R/W	RAM Write Enable 0 (page 0 of high-speed on-chip RAM0*) 0: Write to page 0 is disabled 1: Write to page 0 is enabled

Note: * For the addresses of each page, see section 29, On-Chip RAM.

30.2.10 System Control Register 3 (SYSCR3)

SYSCR3 is an 8-bit readable/writable register that enables or disables access (read/write) from CPU1 to each page of the high-speed on-chip RAM0. Other descriptions on this register are the same as SYSCR1.

Note: When writing to this register, see section 30.4, Usage Notes.

30.2.11 System Control Register 4 (SYSCR4)

SYSCR4 is an 8-bit readable/writable register that enables or disables writing from CPU1 to each page of the high-speed on-chip RAM0. Other descriptions on this register are the same as SYSCR2.

Note: When writing to this register, see section 30.4, Usage Notes.

30.2.12 System Control Register 5 (SYSCR5)

SYSCR5 is an 8-bit readable/writable register that enables or disables access (read/write) from the DMAC to each page of the high-speed on-chip RAM0. Other descriptions on this register are the same as SYSCR1.

Note: When writing to this register, see section 30.4, Usage Notes.

30.2.13 System Control Register 6 (SYSCR6)

SYSCR6 is an 8-bit readable/writable register that enables or disables writing from the DMAC to each page of the high-speed on-chip RAM0. Other descriptions on this register are the same as SYSCR2.

Note: When writing to this register, see section 30.4, Usage Notes.

30.2.14 System Control Register 7 (SYSCR7)

SYSCR7 is an 8-bit readable/writable register that enables or disables access (read/write) from CPU0 to each page of the high-speed on-chip RAM1. Other descriptions on this register are the same as SYSCR1.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	RAME1	RAME0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
1	RAME1	1	R/W	RAM Enable 1 (page 1 of high-speed on-chip RAM1*) 0: Access to page 1 is disabled. 1: Access to page 1 is enabled.
0	RAME0	1	R/W	RAM Enable 0 (page 0 of high-speed on-chip RAM1*) 0: Access to page 0 is disabled. 1: Access to page 0 is enabled.

Note: * For the addresses of each page, see section 29, On-Chip RAM.

30.2.15 System Control Register 8 (SYSCR8)

SYSCR8 is an 8-bit readable/writable register that enables or disables writing from CPU0 to each page of the high-speed on-chip RAM1. Other descriptions on this register are the same as SYSCR2.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	RAM WE1	RAM WE0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
1	RAMWE1	1	R/W	RAM Write Enable 1 (page 1 of high-speed on-chip RAM1*) 0: Write to page 1 is disabled. 1: Write to page 1 is enabled.
0	RAMWE0	1	R/W	RAM Write Enable 0 (page 0 of high-speed on-chip RAM1*) 0: Write to page 0 is disabled. 1: Write to page 0 is enabled.

Note: * For the addresses of each page, see section 29, On-Chip RAM.

30.2.16 System Control Register 9 (SYSCR9)

SYSCR9 is an 8-bit readable/writable register that enables or disables access (read/write) from CPU1 to each page of the high-speed on-chip RAM1. Other descriptions on this register are the same as SYSCR1.

Note: When writing to this register, see section 30.4, Usage Notes.

30.2.17 System Control Register 10 (SYSCR10)

SYSCR10 is an 8-bit readable/writable register that enables or disables writing from CPU1 to each page of the high-speed on-chip RAM1. Other descriptions on this register are the same as SYSCR2.

Note: When writing to this register, see section 30.4, Usage Notes.

30.2.18 System Control Register 11 (SYSCR11)

SYSCR11 is an 8-bit readable/writable register that enables or disables access (read/write) from the DMAC to each page of the high-speed on-chip RAM1. Other descriptions on this register are the same as SYSCR1.

Note: When writing to this register, see section 30.4, Usage Notes.

30.2.19 System Control Register 12 (SYSCR12)

SYSCR12 is an 8-bit readable/writable register that enables or disables writing from the DMAC to each page of the high-speed on-chip RAM1. Other descriptions on this register are the same as SYSCR2.

Note: When writing to this register, see section 30.4, Usage Notes.

30.2.20 Software Reset Control Register (SWRSTCR)

SWRSTCR is an 8-bit readable/writable register that controls the software reset of the SSIF0 to SSIF5 and the IEB.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	SSIF5 SRST	SSIF4 SRST	SSIF3 SRST	SSIF2 SRST	SSIF1 SRST	SSIF0 SRST
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	SSIF5SRST	0	R/W	SSIF5 Software Reset Controls the SSIF5 reset by software. 0: Cancels the SSIF5 reset. 1: Puts the SSIF5 in the reset state.
4	SSIF4SRST	0	R/W	SSIF4 Software Reset Controls the SSIF4 reset by software. 0: Cancels the SSIF4 reset. 1: Puts the SSIF4 in the reset state.
3	SSIF3SRST	0	R/W	SSIF3 Software Reset Controls the SSIF3 reset by software. 0: Cancels the SSIF3 reset. 1: Puts the SSIF3 in the reset state.
2	SSIF2SRST	0	R/W	SSIF2 Software Reset Controls the SSIF2 reset by software. 0: Cancels the SSIF2 reset. 1: Puts the SSIF2 in the reset state.

Bit	Bit Name	Initial Value	R/W	Description
1	SSIF1SRST	0	R/W	SSIF1 Software Reset Controls the SSIF1 reset by software. 0: Cancels the SSIF1 reset. 1: Puts the SSIF1 in the reset state.
0	SSIF0SRST	0	R/W	SSIF0 Software Reset Controls the SSIF0 reset by software. 0: Cancels the SSIF0 reset. 1: Puts the SSIF0 in the reset state.

30.2.21 High-Impedance Control Register (HIZCR)

HIZCR is an 8-bit readable/writable register that selects whether to retain the pin state or set it to high impedance in software standby mode or deep standby mode.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	HIZ	HIZ BSC
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HIZ	0	R/W	High Impedance on Ports Other Than External Bus Control Selects whether to retain the pin state or set it to high impedance for pins other than external bus control pins in software standby mode or deep standby mode. For pins to be controlled, see appendix A, Pin States. Do not set this bit when the TME bit in WTSCR0 of the WDT is 1. To set the output pin state to high impedance, be sure to set this bit while the TME bit is 0. 0: Pin state is retained in software standby mode or deep standby mode 1: Pin state is set to high impedance in software standby mode or deep standby mode.
0	HIZBSC	0	R/W	High Impedance on External Bus Control Ports Selects whether to retain the pin state or set it to high impedance for external bus control pins in software standby mode or deep standby mode. For pins to be controlled, see appendix A, Pin States. Do not set this bit when the TME bit in WTSCR0 of the WDT is 1. To set the output pin state to high impedance, be sure to set this bit while the TME bit is 0. While SDRAM self-refresh operation is enabled, clear the HIZBSC bit to 0 before transition to software standby mode or deep standby mode. For details, refer to section 10, Bus State Controller (BSC). 0: Pin state is retained in software standby mode or deep standby mode. 1: Pin state is set to high impedance in software standby mode or deep standby mode.

30.2.22 CPU0/CPU1 Mode Status Registers (C0MSR, C1MSR)

C0MSR and C1MSR are 8-bit read-only registers that indicate current operating mode of respective processors. Writing to these registers is ignored.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	SLEEP
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

- C0MSR

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0.
0	SLEEP	0	R	CPU0 Status 0: CPU0 is in normal operating mode (CPU0 clock supplied). 1: CPU0 is in sleep mode (CPU0 clock halted).

- C1MSR

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0.
0	SLEEP	0	R	CPU1 Status 0: CPU1 is in normal operating mode (CPU1 clock supplied). 1: CPU1 is in sleep mode (CPU1 clock halted).

30.2.23 Data Retention On-Chip RAM Area Specification Register (RRAMKP)

RRAMKP is an 8-bit readable/writable register specifies whether to retain the contents of the corresponding area of the on-chip RAM (for data retention) in deep standby mode.

When the RRAMKP bit is set to 1, the contents of the corresponding area of the on-chip RAM are retained in deep standby mode. When the bit is cleared to 0, the contents of the corresponding area of the on-chip RAM are not retained in deep standby mode.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RRAMKP3	RRAMKP2	RRAMKP1	RRAMKP0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	RRAMKP3	0	R/W	RRAM Storage Area 3 (page 3* of on-chip RAM for data retention) 0: The contents of the corresponding on-chip RAM area are not retained in deep standby mode. 1: The contents of the corresponding on-chip RAM area are retained in deep standby mode.
2	RRAMKP2	0	R/W	RRAM Storage Area 2 (page 2* of on-chip RAM for data retention) 0: The contents of the corresponding on-chip RAM area are not retained in deep standby mode. 1: The contents of the corresponding on-chip RAM area are retained in deep standby mode.
1	RRAMKP1	0	R/W	RRAM Storage Area 1 (page 1* of on-chip RAM for data retention) 0: The contents of the corresponding on-chip RAM area are not retained in deep standby mode. 1: The contents of the corresponding on-chip RAM area are retained in deep standby mode.

Bit	Bit Name	Initial Value	R/W	Description
0	RRAMKP0	0	R/W	RRAM Storage Area 0 (page 0* of on-chip RAM for data retention) 0: The contents of the corresponding on-chip RAM area are not retained in deep standby mode. 1: The contents of the corresponding on-chip RAM area are retained in deep standby mode.

Note: * For the addresses of each page, see section 29, On-Chip RAM.

30.2.24 Deep Standby Control Register (DSCTR)

DSCTR is an 8-bit readable/writable register that control the states of the external bus control pins and specifies the startup method when deep standby mode is canceled.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	CS0 KEEPE	RAM BOOT	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	CS0KEEPE	0	R/W	Retention of External Bus Control Pin State 0: The state of the external bus control pins is not retained when deep standby mode is canceled. 1: The state of the external bus control pins is retained when deep standby mode is canceled.
6	RAMBOOT	0	R/W	Selection of Startup Method after Return from Deep Standby Mode In the power-on reset exception handling executed when deep standby mode is canceled by the $\overline{\text{MRES}}$, NMI, or IRQ, the program counter (PC) and the stack pointer (SP) are retrieved from the following addresses, respectively. 0: Addresses H'00000000 and H'00000004 1: Addresses H'FF800000 and H'FF800004
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

30.2.25 Deep Standby Cancel Source Select Register (DSSSR)

DSSSR is a 16-bit readable/writable register that consists of the bits for selecting the interrupt to cancel deep standby mode. IRQ7 to IRQ0 bits are valid only for pins allocated to PJ3 to PJ0 and PC3 to PC0.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	MRES	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	MRES	0	R/W	Cancellation of Deep Standby Mode by Manual Reset 0: Deep standby mode is not canceled by a manual reset. 1: Deep standby mode is canceled by a manual reset.
7	IRQ7	0	R/W	Cancellation of Deep Standby Mode by IRQ7 (PJ3 only) 0: Deep standby mode is not canceled by an IRQ7 interrupt. 1: Deep standby mode is canceled by an IRQ7 interrupt.
6	IRQ6	0	R/W	Cancellation of Deep Standby Mode by IRQ6 (PJ2 only) 0: Deep standby mode is not canceled by an IRQ6 interrupt. 1: Deep standby mode is canceled by an IRQ6 interrupt.
5	IRQ5	0	R/W	Cancellation of Deep Standby Mode by IRQ5 (PJ1 only) 0: Deep standby mode is not canceled by an IRQ5 interrupt. 1: Deep standby mode is canceled by an IRQ5 interrupt.
4	IRQ4	0	R/W	Cancellation of Deep Standby Mode by IRQ4 (PJ0 only) 0: Deep standby mode is not canceled by an IRQ4 interrupt. 1: Deep standby mode is canceled by an IRQ4 interrupt.
3	IRQ3	0	R/W	Cancellation of Deep Standby Mode by IRQ3 (PC3 only) 0: Deep standby mode is not canceled by an IRQ3 interrupt. 1: Deep standby mode is canceled by an IRQ3 interrupt.

Bit	Bit Name	Initial Value	R/W	Description
2	IRQ2	0	R/W	Cancellation of Deep Standby Mode by IRQ2 (PC2 only) 0: Deep standby mode is not canceled by an IRQ2 interrupt. 1: Deep standby mode is canceled by an IRQ2 interrupt.
1	IRQ1	0	R/W	Cancellation of Deep Standby Mode by IRQ1 (PC1 only) 0: Deep standby mode is not canceled by an IRQ1 interrupt. 1: Deep standby mode is canceled by an IRQ1 interrupt.
0	IRQ0	0	R/W	Cancellation of Deep Standby Mode by IRQ0 (PC0 only) 0: Deep standby mode is not canceled by an IRQ0 interrupt. 1: Deep standby mode is canceled by an IRQ0 interrupt.

30.2.26 Deep Standby Cancel Source Flag Register (DSFR)

DSFR is a 16-bit readable/writable register composed of two types of bits. One is the flags that are used to confirm which interrupt has canceled deep standby mode. The other is the bit that releases the retention of pin state after deep standby mode is canceled. When deep standby mode is canceled by an interrupt (NMI or IRQ) or a manual reset, this register retains the value before the cancellation although power-on reset exception handling is executed. When deep standby mode is canceled by a power-on reset, this register is initialized to H'0000.

All flags must be cleared immediately before transition to deep standby mode.

Note: When writing to this register, see section 30.4, Usage Notes.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IO KEEP	-	-	-	-	-	MRESF	NMIF	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	IOKEEP	0	R/W	Pin State Retention Releases the retention of the pin states after deep standby mode is canceled. 0: Pin states are not retained. [Clearing condition] Writing 0 1: Pin states are retained. [Setting condition] When deep standby mode is entered
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	MRESF	0	R/W	MRES Flag 0: No interrupt on $\overline{\text{MRES}}$ pin 1: Interrupt on $\overline{\text{MRES}}$ pin

Bit	Bit Name	Initial Value	R/W	Description
8	NMIF	0	R/W	NMI Flag 0: No interrupt on NMI pin 1: Interrupt on NMI pin
7	IRQ7F	0	R/W	IRQ7 Flag (PJ3 only) 0: No interrupt on IRQ7 pin 1: Interrupt on IRQ7 pin
6	IRQ6F	0	R/W	IRQ6 Flag (PJ2 only) 0: No interrupt on IRQ6 pin 1: Interrupt on IRQ6 pin
5	IRQ5F	0	R/W	IRQ5 Flag (PJ1 only) 0: No interrupt on IRQ5 pin 1: Interrupt on IRQ5 pin
4	IRQ4F	0	R/W	IRQ4 Flag (PJ0 only) 0: No interrupt on IRQ4 pin 1: Interrupt on IRQ4 pin
3	IRQ3F	0	R/W	IRQ3 Flag (PC3 only) 0: No interrupt on IRQ3 pin 1: Interrupt on IRQ3 pin
2	IRQ2F	0	R/W	IRQ2 Flag (PC2 only) 0: No interrupt on IRQ2 pin 1: Interrupt on IRQ2 pin
1	IRQ1F	0	R/W	IRQ1 Flag (PC1 only) 0: No interrupt on IRQ1 pin 1: Interrupt on IRQ1 pin
0	IRQ0F	0	R/W	IRQ0 Flag (PC0 only) 0: No interrupt on IRQ0 pin 1: Interrupt on IRQ0 pin

30.3 Operation

30.3.1 Transitions in Power-Down Modes

Figure 30.1 illustrates the state transitions between power-down modes.

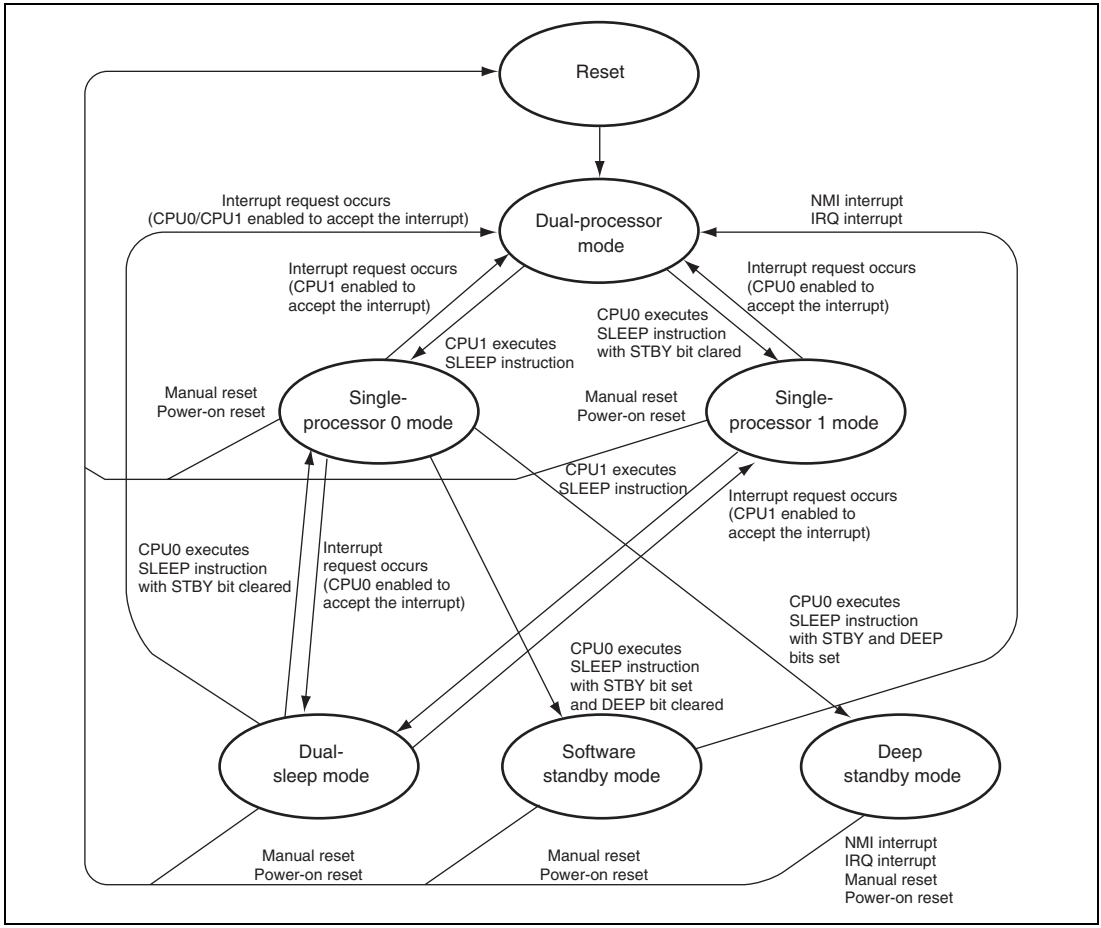


Figure 30.1 Transitions of States in Power-Down Modes

30.3.2 Dual-Processor Mode

After the reset exception handling, both CPU0 and CPU1 are activated and the LSI enters dual-processor mode. For details of the sequence after canceling reset, see section 4, Multi-Core Processor.

30.3.3 Single-Processor Mode

(1) Transition to Single-Processor Mode

In dual-processor mode where CPU0 and CPU1 are running, the LSI can enter single-processor mode where either CPU0 or CPU1 works.

When CPU1 executes the SLEEP instruction in dual-processor mode, CPU1 switches from a program execution state to a sleep state, and the LSI enters single-processor 0 mode irrespective of the value of the STBY bit in STBCR1.

On the other hand, when CPU0 executes the SLEEP instruction while the STBY bit in STBCR1 is 0 in dual-processor mode, CPU0 switches from a program execution state to a sleep state, and the LSI enters single-processor 1 mode. However, if CPU0 executes the SLEEP instruction when the SLPERE bit in STBCR1 is 1, the LSI does not enter single-processor 1 mode and a sleep error exception occurs.

The CPU halts after executing the SLEEP instruction, but the contents of its registers remain unchanged. The on-chip peripheral modules continue to operate. The CKIO pin continues to output the clock.

(2) Canceling Single-Processor Mode

Single-processor mode is canceled by an interrupt (NMI, IRQ, PINT, on-chip peripheral module interrupt, and inter-processor interrupt) or a reset (manual reset or power-on reset).

(a) Canceling by an interrupt

When an NMI, IRQ, on-chip peripheral module interrupt, or inter-processor interrupt occurs, single-processor mode is canceled and interrupt exception handling is executed, and then the LSI enters dual-processor mode. At this time, set the interrupt enable bit for the occurring interrupt so that the CPU in the sleep state is enabled to accept the interrupt. For details of the interrupt enable bit, see section 7, Interrupt Controller (INTC). If the priority level of the interrupt occurred is not higher than the interrupt mask level that is set in SR of the CPU, the interrupt request is not accepted and single-processor mode is not canceled.

(b) Canceling by a reset

Single-processor mode is canceled and the reset exception handling is executed by a power-on reset or manual reset, and then the LSI enters dual-processor mode.

30.3.4 Dual-Sleep Mode

(1) Transition to Dual-Sleep Mode

In single-processor mode where CPU0 or CPU1 is running, the LSI can enter dual-sleep mode.

Executing the SLEEP instruction by CPU0 when the STBY bit in STBCR1 is 0 in single-processor 0 mode causes a transition from the program execution state to dual-sleep mode. However, if CPU0 executes the SLEEP instruction when the SLPERE bit in STBCR1 is 1, the LSI does not enter dual-sleep mode and a sleep error exception occurs.

On the other hand, when CPU1 executes the SLEEP instruction in single-processor 1 mode, CPU1 switches from the program execution state to dual-sleep mode. The LSI enters dual-sleep mode irrespective of the value of the STBY bit in STBCR1.

The running CPU halts after executing the SLEEP instruction, but the contents of its registers remain unchanged. The on-chip peripheral modules continue to operate. The CKIO pin continues to output the clock.

(2) Canceling Dual-Sleep Mode

Dual-sleep mode is canceled by an interrupt (NMI, IRQ, PINT, on-chip peripheral module interrupt, and inter-processor interrupt) or a reset (manual reset or power-on reset).

(a) Canceling by an interrupt

When an NMI, IRQ, on-chip peripheral module interrupt, or inter-processor interrupt occurs, single-processor mode is canceled and interrupt exception handling is executed. The following transition depends on the setting of the interrupt enable bit that controls enabling/disabling of acceptance of interrupts by each CPU. When both CPUs are enabled to accept interrupts, the LSI enters dual-processor mode. When only CPU0 (CPU1) is enabled to accept interrupts, the LSI enters single-processor 0 (single-processor 1) mode. For details of the interrupt enable bit, see section 7, Interrupt Controller (INTC). If the priority level of the interrupt occurred is not higher than the interrupt mask level that is set in SR of the CPU, the interrupt request is not accepted and single-processor mode is not canceled.

(b) Canceling by a reset

Dual-sleep mode is canceled and the reset exception handling is executed by a power-on reset or manual reset, and then the LSI enters dual-processor mode.

30.3.5 Software Standby Mode

(1) Transition to Software Standby Mode

In single-processor 0 mode where only CPU0 is running, the LSI can enter software standby mode.

After setting to disable an interrupt to CPU1 and then confirming that the SLEEP bit in C1MSR is 1, when CPU0 executes the SLEEP instruction with the STBY bit set to 1 and the DEEP bit cleared to 0 in STBCR1, the LSI switches from a program execution state to software standby mode. However, if CPU0 executes the SLEEP instruction when the SLPERE bit in STBCR1 is 1, the LSI does not enter software standby mode and a sleep error exception occurs.

In software standby mode, not only CPU0 and CPU1 but also the clock and on-chip peripheral modules halt. The clock output from the CKIO pin also stops.

The contents of the CPU0/CPU1 registers and the cache registers remain unchanged. Some registers of on-chip peripheral modules are, however, initialized. As for the states of on-chip peripheral module registers in software standby mode, see section 32.3, Register States in Each Operating Mode.

The CPU takes one cycle to finish writing to STBCR1, and then executes processing for the next instruction. However, it takes one or more cycles to actually write. Therefore, execute a SLEEP instruction after reading STBCR1 to have the values written to STBCR1 by the CPU to be definitely reflected in the SLEEP instruction.

The procedure for switching to software standby mode is as follows:

1. Clear the TME bit in the WDT0 timer control/status register (WTCSR0) to 0 to stop the WDT.
2. Set the WDT0 timer counter (WTCNT0) to 0 and set the clock select bits CKS[2:0] in WTCSR0 to appropriate values to secure the specified oscillation settling time.
3. After setting the STBY bit to 1 and DEEP bit to 0 in STBCR1, read STBCR1.
4. Setting to disable an interrupt to CPU1 and confirming that the SLEEP bit in C1MSR is 1, and then make CPU0 to execute the SLEEP instruction.

(2) Canceling Software Standby Mode

Software standby mode is canceled by an interrupt (NMI and IRQ) or a reset (power-on reset or manual reset).

(a) Canceling by an interrupt

When the falling edge or rising edge of the NMI pin (selected by the NMI edge select bit (NMIE) in interrupt control register 0 (C0ICR0, C1ICR0) of the interrupt controller (INTC)) or the falling edge or rising edge of an IRQ pin (IRQ7 to IRQ0) (selected by the IRQn sense select bits (IRQn1S and IRQn0S) in interrupt control register 1 (C0ICR1, C1ICR1) of INTC) is detected, clock oscillation is started. This clock pulse is supplied only to the oscillation settling counter (WDT) used to count the oscillation settling time.

After the elapse of the time set in the clock select bits (CKS[2:0]) in the watchdog timer control/status register (WTCSR0) of WDT0 before the transition to software standby mode, the WDT overflow occurs. Since this overflow indicates that the clock has been stabilized, the clock pulse will be supplied to the entire chip after this overflow. After software standby mode is thus canceled and NMI interrupt exception handling (IRQ interrupt exception handling in case of IRQ) is executed, the LSI enters dual-processor mode. If the priority level of the generated interrupt is equal to or lower than the interrupt mask level specified in the status register (SR) of the CPU, the interrupt request is not accepted and software standby mode is not canceled.

When canceling software standby mode by the NMI interrupt or IRQ interrupt, set the CKS[2:0] bits so that the WDT overflow period will be equal to or longer than the oscillation settling time.

The clock output phase of the CKIO pin may be unstable or fixed to low level immediately after an interrupt is detected and until software standby mode in canceled. When software standby mode is canceled by the falling edge of the NMI pin, the NMI pin should be high when the CPU enters software standby mode (when the clock pulse stops) and should be low when software standby mode is canceled (when the clock is initiated after the oscillation settling). When software standby mode is canceled by the rising edge of the NMI pin, the NMI pin should be low when the CPU enters software standby mode (when the clock pulse stops) and should be high when software standby mode is canceled (when the clock is initiated after the oscillation settling). (This is the same with the IRQ pin.)

(b) Canceling by a reset

When the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin is driven low, the LSI enters the power-on reset or manual reset state and software standby mode is canceled. After that, the reset exception handling is executed and then the LSI enters dual-processor mode.

Keep the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin low until the clock oscillation settles.

The CKIO pin continues to output the internal clock.

(3) Note on Release from Software Standby Mode

Release from software standby mode is triggered by interrupts (NMI and IRQ) or resets (manual reset and power-on reset). If, however, a SLEEP instruction and an interrupt other than NMI and IRQ are generated at the same time, cancellation of software standby mode may occur due to acceptance of the interrupt.

When initiating a transition to software standby mode, make settings so that interrupts are not generated before execution of the SLEEP instruction.

30.3.6 Software Standby Mode Application Example

This example describes a transition to software standby mode on the falling edge of the NMI signal and the cancellation of software standby mode on the rising edge of the NMI signal. The timing is shown in figure 30.2.

When the NMI pin is changed from high to low level while the NMI edge select bit (NMIE) in the interrupt control register (ICR) is set to 0 (falling edge detection), the NMI interrupt is accepted. When the NMIE bit is set to 1 (rising edge detection) by the NMI exception service routine, and then the SLEEP instruction is executed with the STBY bit set to 1 and the DEEP bit set to 0 in STBCR1, the LSI enters software standby mode. Thereafter, when the NMI pin is changed from low to high level, software standby mode is canceled.

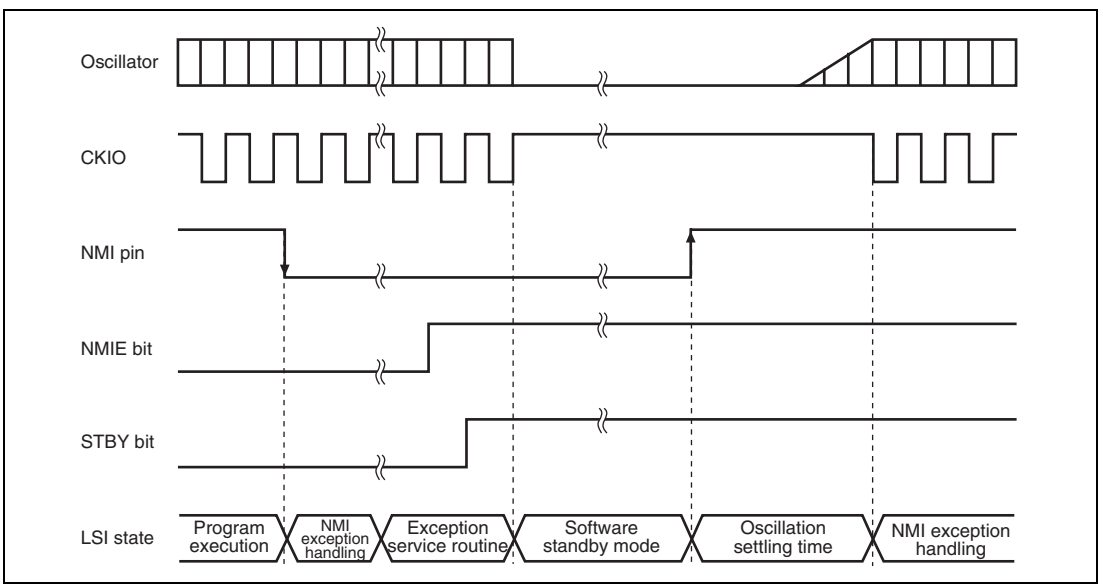


Figure 30.2 NMI Timing in Software Standby Mode (Application Example)

30.3.7 Deep Standby Mode

(1) Transition to Deep Standby Mode

In single-processor 0 mode where only CPU0 is running, the LSI can enter deep standby mode.

After setting to disable an interrupt to CPU1 and confirming that the SLEEP bit in C1MSR is 1, when CPU0 executes the SLEEP instruction with the STBY and DEEP bits in STBCR1 set to 1, the LSI switches from a program execution state to deep standby mode. However, if CPU0 executes the SLEEP instruction when the SLPERE bit in STBCR1 is 1, the LSI does not enter deep standby mode and a sleep error exception occurs.

In deep standby mode, not only CPU0, CPU1, the clock, and on-chip peripheral modules halt, but also power supply is turned off except for that supplied to the RTC and the on-chip RAM (for data retention) area specified by the RRAMKP3 to RRAMKP0 bits in RRAMKP, which can significantly reduce power consumption. Therefore, data in the registers of the CPU0, CPU1, cache, and on-chip peripheral modules are not retained. However, the pin state values immediately before the transition to deep standby mode are retained.

The CPU takes one cycle to finish writing to DSFR, and then executes processing for the next instruction. However, it actually takes one or more cycles to write. Therefore, execute a SLEEP instruction after reading DSFR to definitely reflect the values written to DSFR by the CPU in the SLEEP instruction.

The procedure for switching to deep standby mode is shown below. Figure 30.3 shows its flowchart.

1. Set the RRAMKP3 to RRAMKP0 bits in RRAMKP for the corresponding on-chip RAM (for data retention) area that must be retained. Transfer the programs to be retained to the specified areas of the on-chip RAM (for data retention).
2. To cancel deep standby mode by means of an interrupt or manual reset, set the appropriate bits in DSSSR to select the pins to be used to cancel deep standby mode. In addition, an appropriate input signal detection mode must be specified (using interrupt control registers 0 and 1 (C0ICR0, C1ICR0, C0ICR1, and C1ICR1) of the interrupt controller (INTC)) for the pins what will be used to cancel deep standby mode. Only rising- or falling-edge detection is effective for canceling deep standby mode. (Deep standby mode cannot be canceled when low-level or both-edge IRQ detection is selected.)
3. Execute read and write accesses to an arbitrary but the same address for each page in the on-chip RAM (for data retention) area. If this is not executed, data last written may not be written to the on-chip RAM. If there is any write to the on-chip RAM (for data retention) hereafter, execute this processing after the last write to the on-chip RAM.

4. Set the STBY and DEEP bits in STBCR1 to 1.
5. Clear the flag in DSFR and then read DSFR.
6. Set to disable an interrupt to CPU1 and confirm that the SLEEP bit in C1MSR is 1, and then execute the SLEEP instruction by CPU0.

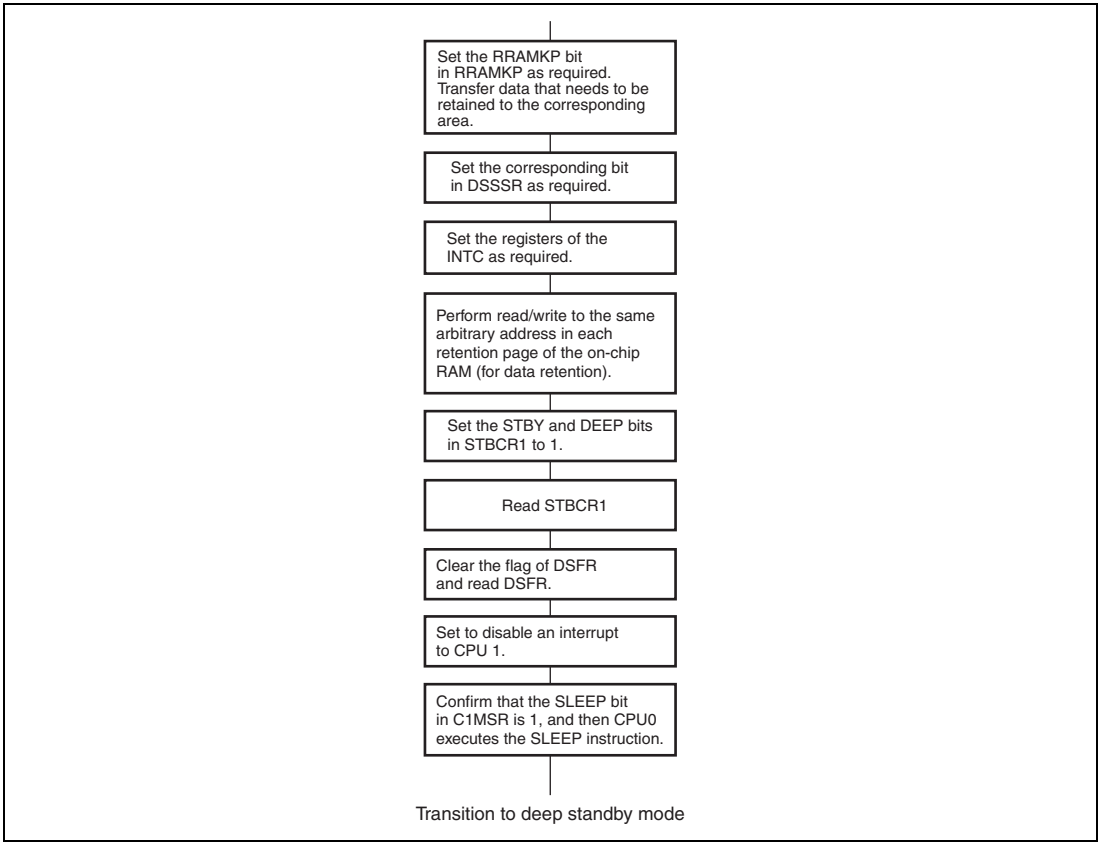
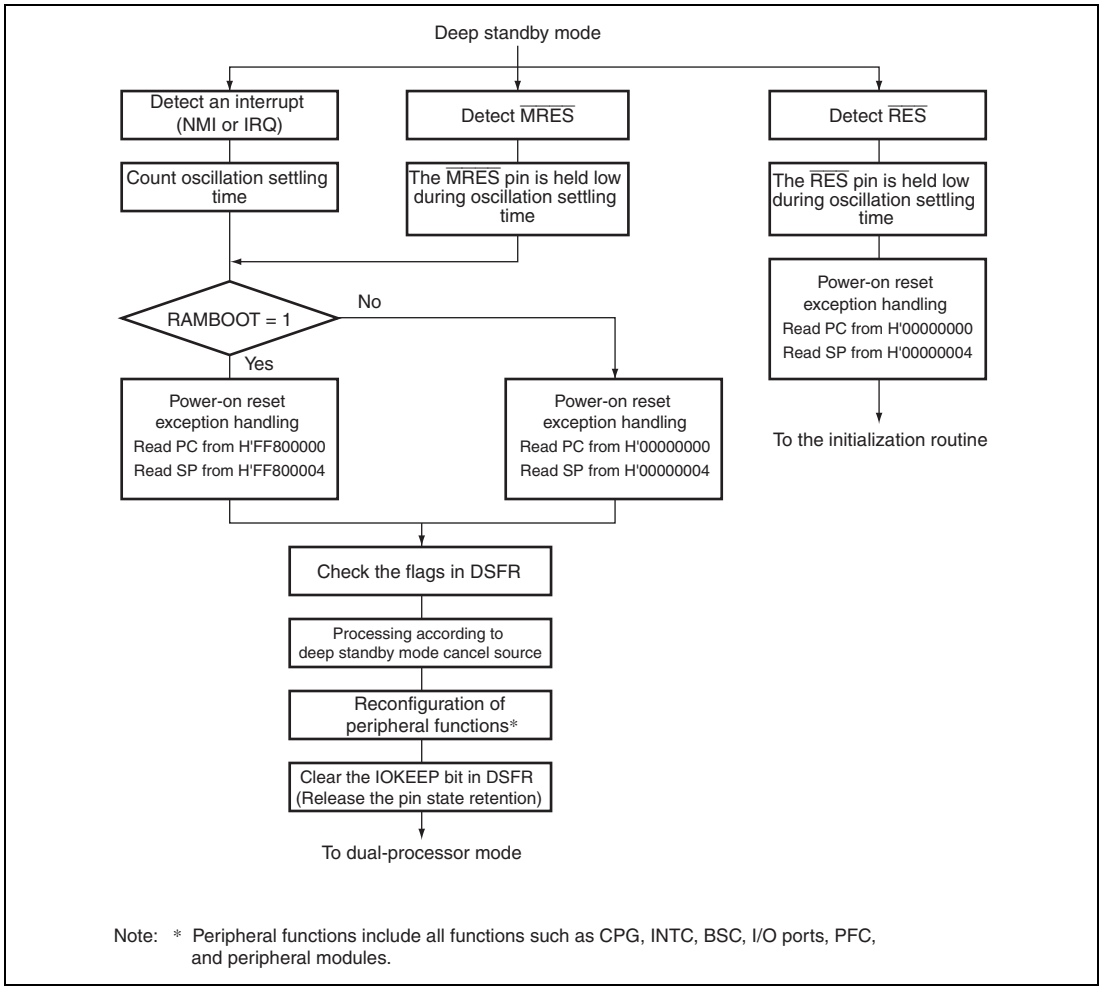


Figure 30.3 Flowchart of Transition to Deep Standby Mode

(2) Canceling Deep Standby Mode

Deep standby mode is canceled by interrupts (NMI or IRQ allocated to PJ3 to PJ0 and PC3 to PC0) or a reset (manual reset or power-on reset). When canceling deep standby mode by the NMI or IRQ interrupt, a power-on reset exception handling is executed instead of an interrupt exception handling. When canceling deep standby mode by manual reset, a power-on reset exception handling is also executed. After executing the power-on reset exception handling, the LSI enters dual-processor mode. Figure 30.4 shows the flowchart of canceling deep standby mode.



Note: * Peripheral functions include all functions such as CPG, INTC, BSC, I/O ports, PFC, and peripheral modules.

Figure 30.4 Flowchart of Canceling Deep Standby Mode

(a) Canceling by an interrupt

When the falling edge or rising edge of the NMI pin (selected by the NMI edge select bit (NMIE) in interrupt control register 0 (C0ICR0, C1ICR0) of the interrupt controller (INTC)) or the falling edge or rising edge of an IRQ pin (IRQ7 to IRQ0 allocated to PJ3 to PJ0 and PC3 to PC0) (selected by the IRQn sense select bits (IRQn1S and IRQn0S) in interrupt control register 1 (C0ICR1, C1ICR1) of INTC) is detected, clock oscillation is started after waiting for the power supply stabilization time. After the oscillation settling time has elapsed, deep standby mode is cancelled and the power-on reset exception handling is executed. If the priority level of the generated interrupt is equal to or lower than the interrupt mask level specified in the status register (SR) of the CPU, the interrupt request is not accepted and deep standby mode is not canceled.

The clock output phase of the CKIO pin may be unstable immediately after an interrupt is detected and until deep standby mode is canceled. When deep standby mode is canceled on the falling edge of the NMI pin, the NMI pin should be high when the CPU enters deep standby mode (when the clock pulse stops) and should be low when deep standby mode is canceled (when the clock is initiated after the oscillation settling). When deep standby mode is canceled by the rising edge of the NMI pin, the NMI pin should be low when the CPU enters deep standby mode (when the clock pulse stops) and should be high when deep standby mode is canceled (when the clock is initiated after the oscillation settling). (This is the same with the IRQ pin.)

In addition, the pin levels of the NMI pin and all interrupt pins (IRQ) selected to cancel deep standby mode (by settings in the deep standby mode cancellation source select register) should be as follows during the transition to deep standby mode, regardless of whether or not those pins are actually used to cancel deep standby mode:

- Pins set to cancel deep standby mode at their rising edge should be low during the transition.
- Pins set to cancel deep standby mode at their falling edge should be high during the transition.

(b) Canceling by a reset

When the RES pin is driven low, this LSI enters the power-on reset state and deep standby mode is canceled. Then, the RES pin is driven high and the power-on reset exception handling is executed. When the RES pin is driven low in clock mode 0, 1, or 3, the internal clock output from the CKIO is executed.

When the MRES pin is driven low, this LSI enters the power-on reset state and deep standby mode is canceled. Then, the MRES pin is driven high and the power-on reset exception handling is executed. When the MRES pin is driven high in clock mode 0, 1, or 3, the internal clock output from the CKIO is executed.

Keep the RES or MRES pin low until the clock oscillation settles.

(3) Operation after Canceling Deep Standby Mode

After exiting from deep standby mode, the LSI can be booted up either through the external bus or from the on-chip RAM (for data retention), which can be selected by setting the RAMBOOT bit in DSCTR. By setting the CS0KEEPE bit, the states of the external bus control pins can be retained even after cancellation of deep standby mode. Table 30.3 shows the pin states after cancellation of deep standby mode according to the setting of each bit. Table 30.4 lists the external bus control pins.

Table 30.3 Pin States and Boot-Up Method after Exit form Deep Standby Mode according to DSCTR Register Setting

CS0KEEPE bit	RAMBOOT bit	Boot-up Method	Pin States after Exit from Deep Standby Mode
0	0	External bus	The states of the external bus control pins are not retained. For other pins, the retention of their states is canceled when the IOKEEP bit is cleared.
	1	On-chip RAM (for data retention)	The states of the external bus control pins are not retained. After cancellation of deep standby mode, the retention of the external bus control pin states is canceled. For other pins, the retention of their states is canceled when the IOKEEP bit is cleared.
1	0	—	Setting prohibited
	1	On-chip RAM (for data retention)	The states of the external bus control pins are retained. The retention of the states of the external bus control pins and other pins is canceled when the IOKEEP bit is cleared.

Table 30.4 External Bus Control Pins in Different Modes

Operating mode 0 (Activation through external 16-bit bus)	Operating mode 1 (Activation through external 32-bit bus)
A[20:1]	A[20:2]
D[15:0]	D[31:0]
$\overline{CS0}$, $\overline{RD}$, CKIO	$\overline{CS0}$, $\overline{RD}$, CKIO

When deep standby mode is canceled by interrupts (NMI or IRQ) or a manual reset, the deep standby cancel source flag register (DSFR) can be used to confirm which interrupt has canceled the mode.

Pins retain the state immediately before the transition to deep standby mode. However, in system activation through the external bus, the retention of the states of the external bus control pins is canceled so that programs can be fetched after cancellation of deep standby mode. The pin states are retained after cancellation of deep standby mode until writing 0 to the IOKEEP bit in DSFR from the same bit. Reconfiguration of peripheral functions is required to return to the previous state of deep standby mode. Peripheral functions include all functions such as CPG, INTC, BSC, I/O ports, PFC, and peripheral modules. After the reconfiguration, the retention of the pin state can be canceled by writing 0 to the IOKEEP bit in DSFR.

(4) Note on Transition to Deep Standby Mode

When multiple cancellation sources have been set and more than one cancellation source is input, multiple cancellation source flags are set.

30.3.8 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. This function can be used to reduce the power consumption in dual-processor mode, single-processor mode, and dual-sleep mode. Disable a module before placing it in module standby mode. In addition, do not access the module's registers while it is in the module standby state.

For details on the states of registers, see section 32.3, Register States in Each Operating Mode.

(2) Canceling Module Standby Function

The module standby function can be canceled by clearing each MSTP bit to 0, or by a power-on reset (only possible for RTC, H-UDI, and UBC). When canceling the module standby function by clearing the corresponding MSTP bit to 0, read the MSTP bit to confirm that it has been cleared to 0.

30.4 Usage Notes

30.4.1 Notes on Writing to Registers

When writing to the registers related to power-down modes, note the following.

When writing to the register related to power-down modes, the CPU, after executing a write instruction, executes the next instruction without waiting for the write operation to complete.

Therefore, to reflect the change specified by writing to the register while the next instruction is executed, insert a dummy read of the same register between the register write instruction and the next instruction.

30.4.2 Notice about Deep Standby Control Register (DSCTR)

After (1) power-on reset by $\overline{\text{RES}}$ pin is released, and (2) the LSI transits to deep standby mode in case that bit 7 (CS0KEEPE) and bit 6 (RAMBOOT) of deep standby control register (DSCTR) are set to "1", these bits become unable to be written as "0" since then.

To write these as “0”, it is necessary to assert $\overline{\text{RES}}$ pin to low.

30.4.3 Notice about Power-On Reset Exception Handling

After (1) power-on reset by $\overline{\text{RES}}$ pin is released, (2) the LSI transit to deep standby mode in case that bit 6 (RAMBOOT) of deep standby control register (DSCTR) is set to "1", (3) the deep standby mode is cancelled, and (4) power-on reset by WDT or H-UDI reset is occurred before power-on reset by $\overline{\text{RES}}$ pin is executed again, then the behavior of the power-on reset exception handling is as table 30.5. So if applicable as above case, PC and SP are necessary to be retained in the area of on-chip RAM for data retention.

Table 30.5 Power-On Reset Exception Handling

Address where the program counter (PC) is fetched	Address where the stack pointer (SP) is fetched
H'FF800000	H'FF800004

After (1) power-on reset by $\overline{\text{RES}}$ pin is released, (2) the LSI transit to deep standby mode, and (3) the deep standby mode is cancelled, if there is a possibility that power-on reset by WDT or H-UDI reset is occurred before power-on reset by $\overline{\text{RES}}$ pin is executed again, the settings of WDT or H-UDI should be done in the condition that bit 15 (IOKEEP) and bits 9 to 0 of deep standby cancel source flag register (DSFR) are all cleared after canceling deep standby mode (if some bits are 1, please write these as “0”).

If (1) the setting of WDT or H-UDI is done in the condition that IOKEEP bit is not 0, and (2) power-on reset by WDT or H-UDI reset is occurred before power-on reset by $\overline{\text{RES}}$ pin is executed again, the pin status of the pins, whose pin status are retained in deep standby mode and which are not in table 30.4, are kept retained. Additionally, in the case that bit 7 (CS0KEEPE) of deep standby control register (DSCTR) are set to “1”, the pin status of the pins in table 30.4 are also keep retained.

If (1) the settings of WDT or H-UDI is done in the condition that bits 9 to 0 are not all 0, and (2) power-on reset by WDT or H-UDI reset is occurred before power-on reset by $\overline{\text{RES}}$ pin is executed again, the internal information about the deep standby canceling source is not cleared, and deep standby mode are cancelled by the wrong canceling source when the LSI attempt to transit to deep standby mode since then.

Section 31 User Debugging Interface (H-UDI)

This LSI incorporates a user debugging interface (H-UDI) for the boundary scan function and emulator support.

31.1 Features

The user debugging interface (H-UDI) is a serial input/output interface that supports JTAG (Joint Test Action Group, IEEE Std.1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture).

The H-UDI of this LSI incorporates a boundary scan TAP controller and an emulation TAP controller for controlling the H-UDI interrupt function. When the $\overline{\text{TRST}}$ pin is asserted, including the case of power-on, the boundary scan TAP controller is selected. By inputting the emulation TAP controller switching command, the emulation TAP controller is selected. To switch from the emulation TAP controller to the boundary scan TAP controller, assert the $\overline{\text{TRST}}$ pin.

In ASE mode, the emulation TAP controller is selected. For connection with the emulator, see the manual for the emulator.

Figure 31.1 shows a block diagram of the H-UDI.

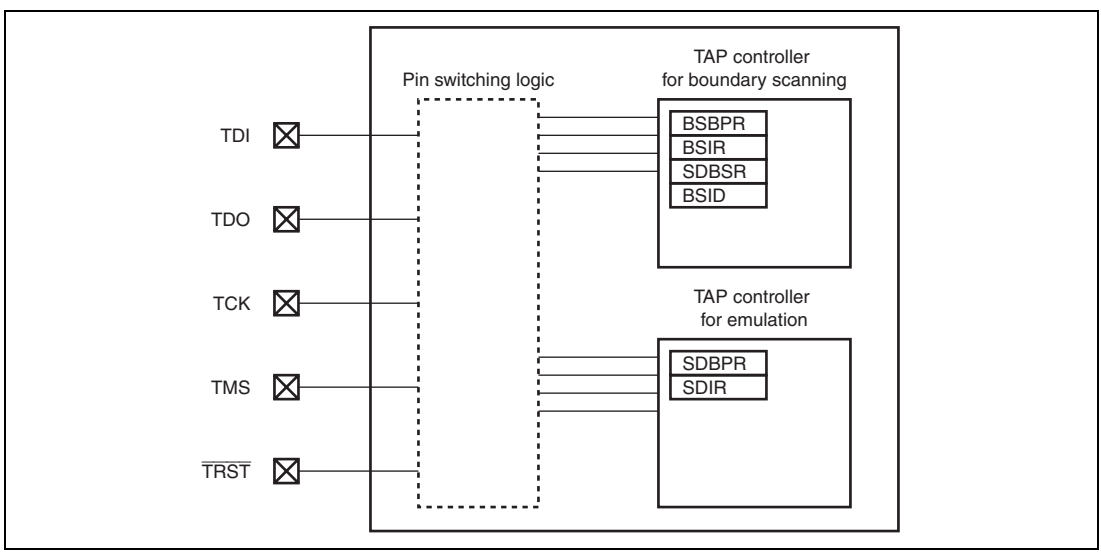


Figure 31.1 Block Diagram of H-UDI

31.2 Input/Output Pins

Table 31.1 Pin Configuration

Pin Name	Symbol	I/O	Function
H-UDI serial data input/output clock pin	TCK	Input	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.
Mode select input pin	TMS	Input	The state of the TAP control circuit is determined by changing this signal in synchronization with TCK. The protocol complies with the JTAG standard (IEEE Std.1149.1).
H-UDI reset input pin	$\overline{\text{TRST}}$	Input	Input is accepted asynchronously with respect to TCK, and when low, the H-UDI is reset. $\overline{\text{TRST}}$ must be low for a period when power is turned on regardless of using the H-UDI function. See section 31.5.2, Reset Configuration, for more information.
H-UDI serial data input pin	TDI	Input	Data is transferred to the H-UDI by changing this signal in synchronization with TCK.
H-UDI serial data output pin	TDO	Output	Data is read from the H-UDI by reading this pin in synchronization with TCK. The initial value of the data output timing is the TCK falling edge, but this initial value can be changed to the TCK rising edge by inputting the TDO transition timing switching command to SDIR. See section 31.5.3, TDO Output Timing, for more information.
ASE mode select pin	$\overline{\text{ASEMD}}$*	Input	If a low level is input at the $\overline{\text{ASEMD}}$ pin while the $\overline{\text{RES}}$ pin is asserted, ASE mode is entered; if a high level is input, product chip mode is entered. In ASE mode, dedicated emulator function can be used. The input level at the $\overline{\text{ASEMD}}$ pin should be held for at least one cycle after $\overline{\text{RES}}$ negation.

Note: * When the emulator is not in use, fix this pin to the high level.

31.3 Description of the Boundary Scan TAP Controller

The boundary scan TAP controller has the following registers.

Table 31.2 Register Configuration of the Boundary Scan TAP Controller

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Bypass register	BSBPR	—	—	—	—
Instruction register	BSIR	—	H'4	—	—
Boundary scan register	SDBSR	—	—	—	—
ID register	BSID	—	H'08057447	—	—

31.3.1 Bypass Register (BSBPR)

BSBPR is a 1-bit register that cannot be accessed by the CPU. When BSIR is set to BYPASS mode, BSBPR is connected between H-UDI pins TDI and TDO. The initial value is undefined.

31.3.2 Instruction Register (BSIR)

BSIR is a 4-bit register and initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state. This register cannot be accessed by the CPU.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	TI[3:0]	0100	—	Test Instruction The H-UDI instruction is transferred to BSIR as a serial input from TDI. For commands, see table 31.3.

Table 31.3 Supported Commands for Boundary Scan TAP Controller

Bits 3 to 0				
TI3	**TI2**	**TI1**	**TI0**	**Description**
0	0	0	0	EXTEST
0	0	0	1	SAMPLE/PRELOAD
0	0	1	1	Emulation TAP controller switching command
0	1	0	0	IDCODE (initial value)
0	1	1	0	CLAMP
0	1	1	1	HIGHZ
1	1	1	1	BYPASS
Other than the above				Reserved

31.3.3 Boundary Scan Register (SDBSR)

SDBSR is a shift register located on the PAD to control input/output pins of this LSI. This register cannot be accessed by the CPU. The initial value is undefined.

The EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ commands can be used to perform the boundary scan test that conforms to the JTAG standard. Table 31.4 shows the correspondence between the LSI pins and the bits of the boundary scan register.

Table 31.4 Correspondence between the LSI Pins and the Bits of the Boundary Scan Register

Bit Number	Pin Name*1	Type	Bit Number	Pin Name*1	Type	Bit Number	Pin Name*1	Type
	From TDI		370	PE1	CONTROL	341	PC10	INPUT
398	NMI	INPUT	369	PE1	OUTPUT	340	PC9	CONTROL
397	PE10	OUTPUT*2	368	PE1	INPUT	339	PC9	OUTPUT
396	PE10	INPUT	367	PE2	CONTROL	338	PC9	INPUT
395	PE11	CONTROL	366	PE2	OUTPUT	337	PC8	CONTROL
394	PE11	OUTPUT	365	PE2	INPUT	336	PC8	OUTPUT
393	PE11	INPUT	364	PE0	CONTROL	335	PC8	INPUT
392	PE7	CONTROL	363	PE0	OUTPUT	334	PC7	CONTROL
391	PE7	OUTPUT	362	PE0	INPUT	333	PC7	OUTPUT
390	PE7	INPUT	361	PE5	CONTROL	332	PC7	INPUT
389	PE12	OUTPUT*2	360	PE5	OUTPUT	331	PC6	CONTROL
388	PE12	INPUT	359	PE5	INPUT	330	PC6	OUTPUT
387	PE13	CONTROL	358	PF4	CONTROL	329	PC6	INPUT
386	PE13	OUTPUT	357	PF4	OUTPUT	328	PC5	CONTROL
385	PE13	INPUT	356	PF4	INPUT	327	PC5	OUTPUT
384	PE3	CONTROL	355	PF3	CONTROL	326	PC5	INPUT
383	PE3	OUTPUT	354	PF3	OUTPUT	325	PC4	CONTROL
382	PE3	INPUT	353	PF3	INPUT	324	PC4	OUTPUT
381	PE8	OUTPUT*2	352	PF2	CONTROL	323	PC4	INPUT
380	PE8	INPUT	351	PF2	OUTPUT	322	PC3	CONTROL
379	PE9	CONTROL	350	PF2	INPUT	321	PC3	OUTPUT
378	PE9	OUTPUT	349	PF0	CONTROL	320	PC3	INPUT
377	PE9	INPUT	348	PF0	OUTPUT	319	PC2	CONTROL
376	PE6	CONTROL	347	PF0	INPUT	318	PC2	OUTPUT
375	PE6	OUTPUT	346	PF1	CONTROL	317	PC2	INPUT
374	PE6	INPUT	345	PF1	OUTPUT	316	PC1	CONTROL
373	PE4	CONTROL	344	PF1	INPUT	315	PC1	OUTPUT
372	PE4	OUTPUT	343	PC10	CONTROL	314	PC1	INPUT
371	PE4	INPUT	342	PC10	OUTPUT	313	PC0	CONTROL

Bit Number	Pin Name*1	Type	Bit Number	Pin Name*1	Type	Bit Number	Pin Name*1	Type
312	PC0	OUTPUT	282	PA9	CONTROL	252	D3	CONTROL
311	PC0	INPUT	281	PA9	OUTPUT	251	D3	OUTPUT
310	MD0	INPUT	280	PA9	INPUT	250	D3	INPUT
309	PA0	CONTROL	279	PA10	CONTROL	249	D4	CONTROL
308	PA0	OUTPUT	278	PA10	OUTPUT	248	D4	OUTPUT
307	PA0	INPUT	277	PA10	INPUT	247	D4	INPUT
306	PA1	CONTROL	276	PA11	CONTROL	246	D5	CONTROL
305	PA1	OUTPUT	275	PA11	OUTPUT	245	D5	OUTPUT
304	PA1	INPUT	274	PA11	INPUT	244	D5	INPUT
303	PA2	CONTROL	273	PA12	CONTROL	243	D6	CONTROL
302	PA2	OUTPUT	272	PA12	OUTPUT	242	D6	OUTPUT
301	PA2	INPUT	271	PA12	INPUT	241	D6	INPUT
300	PA3	CONTROL	270	PA13	CONTROL	240	D7	CONTROL
299	PA3	OUTPUT	269	PA13	OUTPUT	239	D7	OUTPUT
298	PA3	INPUT	268	PA13	INPUT	238	D7	INPUT
297	PA4	CONTROL	267	PA14	CONTROL	237	D8	CONTROL
296	PA4	OUTPUT	266	PA14	OUTPUT	236	D8	OUTPUT
295	PA4	INPUT	265	PA14	INPUT	235	D8	INPUT
294	PA5	CONTROL	264	PA15	CONTROL	234	D9	CONTROL
293	PA5	OUTPUT	263	PA15	OUTPUT	233	D9	OUTPUT
292	PA5	INPUT	262	PA15	INPUT	232	D9	INPUT
291	PA6	CONTROL	261	D0	CONTROL	231	D10	CONTROL
290	PA6	OUTPUT	260	D0	OUTPUT	230	D10	OUTPUT
289	PA6	INPUT	259	D0	INPUT	229	D10	INPUT
288	PA7	CONTROL	258	D1	CONTROL	228	D11	CONTROL
287	PA7	OUTPUT	257	D1	OUTPUT	227	D11	OUTPUT
286	PA7	INPUT	256	D1	INPUT	226	D11	INPUT
285	PA8	CONTROL	255	D2	CONTROL	225	D12	CONTROL
284	PA8	OUTPUT	254	D2	OUTPUT	224	D12	OUTPUT
283	PA8	INPUT	253	D2	INPUT	223	D12	INPUT

Bit Number	Pin Name*1	Type	Bit Number	Pin Name*1	Type	Bit Number	Pin Name*1	Type
222	D13	CONTROL	192	PB4	OUTPUT	162	PB1	INPUT
221	D13	OUTPUT	191	PB4	INPUT	161	A2	CONTROL
220	D13	INPUT	190	PB5	CONTROL	160	A2	OUTPUT
219	D14	CONTROL	189	PB5	OUTPUT	159	A3	CONTROL
218	D14	OUTPUT	188	PB5	INPUT	158	A3	OUTPUT
217	D14	INPUT	187	PB6	CONTROL	157	A4	CONTROL
216	D15	CONTROL	186	PB6	OUTPUT	156	A4	OUTPUT
215	D15	OUTPUT	185	PB6	INPUT	155	A5	CONTROL
214	D15	INPUT	184	PB7	CONTROL	154	A5	OUTPUT
213	PA9	CONTROL	183	PB7	OUTPUT	153	A6	CONTROL
212	PA9	OUTPUT	182	PB7	INPUT	152	A6	OUTPUT
211	PA9	INPUT	181	PB16	CONTROL	151	A7	CONTROL
210	PB10	CONTROL	180	PB16	OUTPUT	150	A7	OUTPUT
209	PB10	OUTPUT	179	PB16	INPUT	149	A8	CONTROL
208	PB10	INPUT	178	PB15	CONTROL	148	A8	OUTPUT
207	PB11	CONTROL	177	PB15	OUTPUT	147	A9	CONTROL
206	PB11	OUTPUT	176	PB15	INPUT	146	A9	OUTPUT
205	PB11	INPUT	175	PB8	CONTROL	145	A10	CONTROL
204	PB12	CONTROL	174	PB8	OUTPUT	144	A10	OUTPUT
203	PB12	OUTPUT	173	PB8	INPUT	143	A11	CONTROL
202	PB12	INPUT	172	RD	CONTROL	142	A11	OUTPUT
201	PB13	CONTROL	171	RD	OUTPUT	141	A12	CONTROL
200	PB13	OUTPUT	170	PB17	CONTROL	140	A12	OUTPUT
199	PB13	INPUT	169	PB17	OUTPUT	139	A13	CONTROL
198	PB14	CONTROL	168	PB17	INPUT	138	A13	OUTPUT
197	PB14	OUTPUT	167	PB0	CONTROL	137	A14	CONTROL
196	PB14	INPUT	166	PB0	OUTPUT	136	A14	OUTPUT
195	CS0	CONTROL	165	PB0	INPUT	135	A15	CONTROL
194	CS0	OUTPUT	164	PB1	CONTROL	134	A15	OUTPUT
193	PB4	CONTROL	163	PB1	OUTPUT	133	A16	CONTROL

Bit Number	Pin Name*1	Type	Bit Number	Pin Name*1	Type	Bit Number	Pin Name*1	Type
132	A16	OUTPUT	102	PH2	CONTROL	72	PH13	OUTPUT
131	A17	CONTROL	101	PH2	OUTPUT	71	PH13	INPUT
130	A17	OUTPUT	100	PH2	INPUT	70	PH6	CONTROL
129	A18	CONTROL	99	PH3	CONTROL	69	PH6	OUTPUT
128	A18	OUTPUT	98	PH3	OUTPUT	68	PH6	INPUT
127	A19	CONTROL	97	PH3	INPUT	67	PH7	CONTROL
126	A19	OUTPUT	96	PH15	CONTROL	66	PH7	OUTPUT
125	A20	CONTROL	95	PH15	OUTPUT	65	PH7	INPUT
124	A20	OUTPUT	94	PH15	INPUT	64	PH12	CONTROL
123	PB2	CONTROL	93	PH4	CONTROL	63	PH12	OUTPUT
122	PB2	OUTPUT	92	PH4	OUTPUT	62	PH12	INPUT
121	PB2	INPUT	91	PH4	INPUT	61	PH11	CONTROL
120	PB3	CONTROL	90	PB18	CONTROL	60	PH11	OUTPUT
119	PB3	OUTPUT	89	PB18	OUTPUT	59	PH11	INPUT
118	PB3	INPUT	88	PB18	INPUT	58	PH14	CONTROL
117	PD2	CONTROL	87	MD_CLK1	INPUT	57	PH14	OUTPUT
116	PD2	OUTPUT	86	MD_CLK0	INPUT	56	PH14	INPUT
115	PD2	INPUT	85	PH9	CONTROL	55	PJ2	CONTROL
114	PD1	CONTROL	84	PH9	OUTPUT	54	PJ2	OUTPUT
113	PD1	OUTPUT	83	PH9	INPUT	53	PJ2	INPUT
112	PD1	INPUT	82	PH10	CONTROL	52	PJ3	CONTROL
111	PD0	CONTROL	81	PH10	OUTPUT	51	PJ3	OUTPUT
110	PD0	OUTPUT	80	PH10	INPUT	50	PJ3	INPUT
109	PD0	INPUT	79	PH5	CONTROL	49	PJ1	CONTROL
108	PH0	CONTROL	78	PH5	OUTPUT	48	PJ1	OUTPUT
107	PH0	OUTPUT	77	PH5	INPUT	47	PJ1	INPUT
106	PH0	INPUT	76	PH8	CONTROL	46	PJ0	CONTROL
105	PH1	CONTROL	75	PH8	OUTPUT	45	PJ0	OUTPUT
104	PH1	OUTPUT	74	PH8	INPUT	44	PJ0	INPUT
103	PH1	INPUT	73	PH13	CONTROL	43	PJ4	CONTROL

Bit Number	Pin Name*1	Type	Bit Number	Pin Name*1	Type	Bit Number	Pin Name*1	Type
42	PJ4	OUTPUT	27	PJ9	OUTPUT	12	PK0	OUTPUT
41	PJ4	INPUT	26	PJ9	INPUT	11	PK0	INPUT
40	PJ5	CONTROL	25	PJ10	CONTROL	10	PG0	INPUT
39	PJ5	OUTPUT	24	PJ10	OUTPUT	9	PG1	INPUT
38	PJ5	INPUT	23	PJ10	INPUT	8	PG2	INPUT
37	PJ7	CONTROL	22	PJ11	CONTROL	7	PG3	INPUT
36	PJ7	OUTPUT	21	PJ11	OUTPUT	6	PG4	INPUT
35	PJ7	INPUT	20	PJ11	INPUT	5	PG5	INPUT
34	PJ6	CONTROL	19	PJ12	CONTROL	4	PG6	INPUT
33	PJ6	OUTPUT	18	PJ12	OUTPUT	3	PG7	INPUT
32	PJ6	INPUT	17	PJ12	INPUT	2	$\overline{\text{ASEBRKAK}}$ /$\overline{\text{ASEBRK}}$	CONTROL
31	PJ8	CONTROL	16	PK1	CONTROL	1	$\overline{\text{ASEBRKAK}}$ /$\overline{\text{ASEBRK}}$	OUTPUT
30	PJ8	OUTPUT	15	PK1	OUTPUT	0	$\overline{\text{ASEBRKAK}}$ /$\overline{\text{ASEBRK}}$	INPUT
29	PJ8	INPUT	14	PK1	INPUT	To TDO		
28	PJ9	CONTROL	13	PK0	CONTROL			

Notes: 1. The pin name used for function 1

2. The pin is open-drain. The pin state is low when driven low, whereas high impedance (Hi-Z) when driven high.

3. The pin of CONTROL is active-low. When this pin is driven low, the state of the corresponding pin is output.

31.3.4 ID Register (BSID)

BSID is a 32-bit register that cannot be accessed by the CPU. The register can be read from H-UDI pins when the IDCODE command is set, but is not writable.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DID[31:16]															
Initial value:	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DID[15:0]															
Initial value:	0	1	1	1	0	1	0	0	0	1	0	0	0	1	1	1
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DID[31:0]	H'08057447	—	Device This is an ID register defined by JTAG. The value in this LSI is H'08057447. The upper four bits may be changed for different chip versions.

31.4 Description of the Emulation TAP Controller

To use the emulation TAP controller, enter the emulation TAP controller switching command in the BSIR register of the boundary scan TAP controller. The emulation TAP controller has the following registers.

Table 31.5 Register Configuration of the Emulation TAP Controller

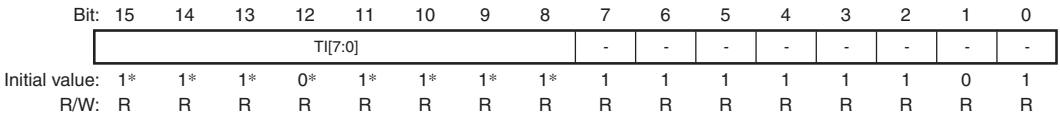
Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Bypass register	SDBPR	—	—	—	—
Instruction register	SDIR	R	H'EFFD	H'FFFD9000	16

31.4.1 Bypass Register (SDBPR)

SDBPR is a 1-bit register that cannot be accessed by the CPU. When SDIR is set to BYPASS mode, SDBPR is connected between H-UDI pins TDI and TDO. The initial value is undefined.

31.4.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register and initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state. H-UDI can write to this register regardless of the CPU mode. When a reserved command is set in this register, the operation is not guaranteed. The initial value is H'EFFD.



Note: * The initial value of TI[7:0] is a reserved value, but replace it with a non-reserved value when setting a command.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	TI[7:0]	11101111*	R	Test Instruction Instruction for the H-UDI is transferred to SDIR as a serial input from TDI. For commands, see table 31.6.
7 to 2	—	All 1	R	Reserved These bits are always read as 1.
1	—	0	R	Reserved These bits are always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
0	—	1	R	Reserved These bits are always read as 1.

Table 31.6 Supported Commands for Emulation TAP Controller

Bits 15 to 8								
TI7	**TI6**	**TI5**	**TI4**	**TI3**	**TI2**	**TI1**	**TI0**	**Description**
0	1	1	0	—	—	—	—	H-UDI reset negation
0	1	1	1	—	—	—	—	H-UDI reset assertion
1	0	0	1	1	1	0	0	TDO transition timing switch
1	0	1	1	—	—	—	—	H-UDI interrupt
1	1	1	1	—	—	—	—	BYPASS
Other than the above								Reserved

31.5 Operation

31.5.1 TAP Controller

Figure 31.2 shows the internal states of the TAP controller. This state machine conforms to the state transitions defined by JTAG.

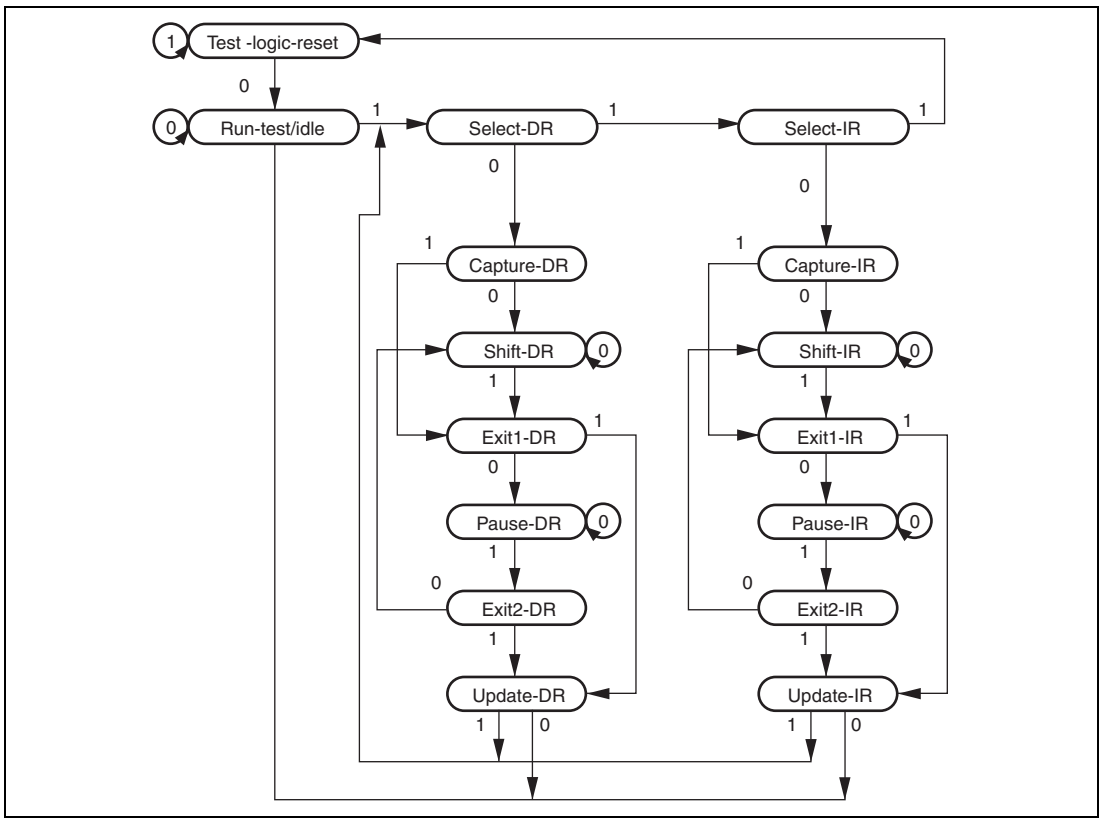


Figure 31.2 TAP Controller State Transitions

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 transition timing of the TDO value, see section 31.5.3, 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.

31.5.2 Reset Configuration

Table 31.7 Reset Configuration

$\overline{\text{ASEMD}}$*1	$\overline{\text{RES}}$	$\overline{\text{TRST}}$	Chip State
H	L	L	Power-on reset and H-UDI reset
		H	Power-on reset
	H	L	H-UDI reset only
		H	Normal operation
L	L	L	Reset hold*2
		H	Power-on reset
	H	L	H-UDI reset only
		H	Normal operation

Notes: 1. Performs product chip mode and ASE mode settings
$\overline{\text{ASEMD}}$ = H, product chip mode
$\overline{\text{ASEMD}}$ = L, ASE mode

2. In ASE mode, reset hold is entered if the $\overline{\text{TRST}}$ pin is driven low while the $\overline{\text{RES}}$ 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 a power-on reset.

31.5.3 TDO Output Timing

When the emulation TAP controller is selected, a transition on the TDO pin is output on the falling edge of TCK with the initial value. However, setting a TDO transition timing switching command in SDIR via the H-UDI pin and passing the Update-IR state synchronizes the TDO transition with the rising edge of TCK. This command does not affect the output timing of the boundary scan TAP controller.

To synchronize the transition of TDO with the falling edge of TCK after setting the TDO transition timing switching command, the $\overline{\text{TRST}}$ pin must be asserted simultaneously with the power-on reset. In the case of power-on reset by the $\overline{\text{RES}}$ pin, the sync reset is still in operation for a certain period in the LSI even after the $\overline{\text{RES}}$ pin is negated. Thus, if the $\overline{\text{TRST}}$ pin is asserted immediately after the negation of the $\overline{\text{RES}}$ pin, the TDO transition timing switching command is cleared, resulting in TDO transitions synchronized with the falling edges of TCK. To prevent this, make sure to allow a period of 20 tcyc or longer between the signal transitions of the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ pins.

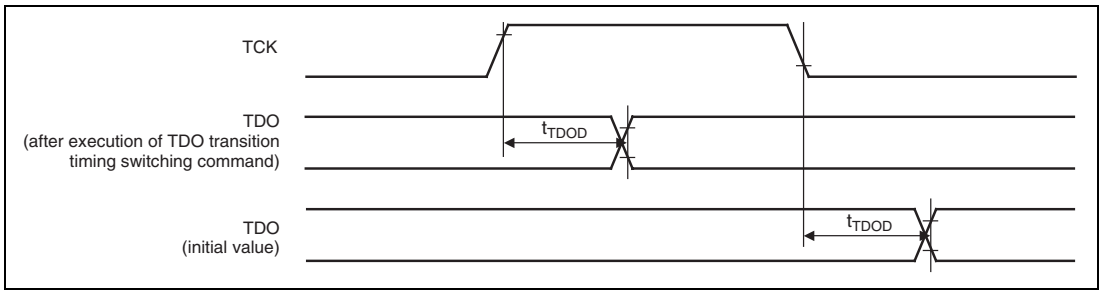


Figure 31.3 H-UDI Data Transfer Timing

31.5.4 H-UDI Reset

An H-UDI reset occurs when an H-UDI reset assert command is set in SDIR. An H-UDI reset is of the same kind as a power-on reset. An H-UDI reset is cleared by setting 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 $\overline{\text{RES}}$ pin low to apply a power-on reset.

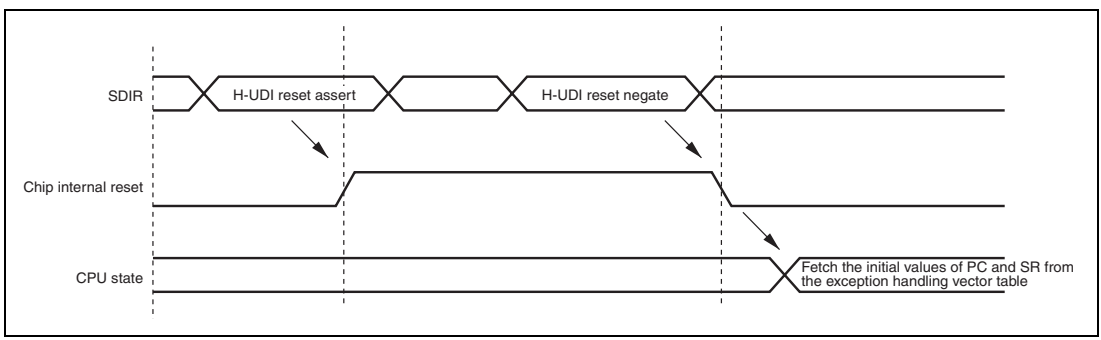


Figure 31.4 H-UDI Reset

31.5.5 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting a command from the H-UDI in SDIR. An H-UDI interrupt is a general exception/interrupt operation, resulting in fetching the exception service routine start address from the exception handling vector table, jumping to that address, and starting program execution from that address. This interrupt request has a fixed priority level of 15.

H-UDI interrupts are accepted in sleep mode, but not in software standby mode.

31.6 Boundary Scan

By setting the commands in BSIR by the H-UDI, the H-UDI pins can be configured for boundary scan mode defined by JTAG.

31.6.1 Supported Instructions

This LSI supports three required instructions (BYPASS, SAMPLE/PRELOAD, and EXTEST) and three optional instructions (IDCODE, CLAMP, and HIGHZ) defined by JTAG.

(1) BYPASS

The BYPASS instruction is a required standard instruction to operate the bypass register. This instruction is used to increase the transfer speed of serial data of other LSIs on the printed circuit board by reducing the shift path. During execution of this instruction, the test circuit does not affect the system circuit.

(2) SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction inputs a value from the internal circuit of the LSI to the boundary scan register, and output the data from scan path or load the data to the scan path. During execution of the instruction, the value on the input pin of the LSI is transferred to the internal circuit and the value of the internal circuit is output externally from the output pin. Execution of the instruction does not affect the system circuit of the LSI.

In SAMPLE operation, the snapshots of the value transferred from the input pin to the internal circuit and the value transferred from the internal circuit to the output pin are captured in the boundary scan register and then read from the scan path. Capturing of the snapshots is performed in synchronization with the rising edge of TCK in the capture-DR state. The capturing is performed without interfering with normal operation of the LSI.

In PRELOAD operation, an initial value is set in the output latch of the boundary scan register from the scan path before execution of the EXTEST instruction. Without PRELOAD operation, an undefined value is output from the output pin until the first scan sequence is completed (transferred to the output latch) during execution of the EXTEST instruction (the parallel output latch is always output to the output pin with the EXTEST instruction).

(3) EXTEST

The EXTEST instruction tests the external circuit when this LSI is mounted on the printed circuit board. During execution of this instruction, the output pin is used to output the test data (set in advance by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed circuit board and the input pin is used to capture the test result from the printed circuit board to the boundary scan register. When a test is performed using the EXTEST instruction N times, the N-th test data is scanned-in during (N-1)-th scan-out.

The data loaded in the boundary scan register of the output pin in the capture-DR state of this instruction is not used in testing of the external circuit (an exchange is made in shift operation).

(4) IDCODE

To set the H-UDI pin to IDCODE mode defined by JTAG, set the command in SDIR from the H-UDI pin. When H-UDI is initialized ($\overline{\text{TRST}}$ is asserted or TAP is placed in the test-logic-reset state), IDCODE mode is entered.

(5) CLAMP and HIGHZ

To set the H-UDI pin to CLAMP or HIGHZ mode defined by JTAG, set the command in SDIR from the H-UDI pin.

31.6.2 Notes

1. The clock related signals (EXTAL, XTAL, CKIO, AUDIO_X1, AUDIO_X2, USB_X1, USB_X2, RTC_X1, and RTC_X2) are inapplicable to the boundary scan.
2. The reset-related signal ($\overline{\text{RES}}$) is inapplicable to the boundary scan.
3. The H-UDI related signals (TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$, and $\overline{\text{ASEMD}}$) are inapplicable to the boundary scan.
4. The USB related signals (DP0, DM0, DP1, DM1, VBUS, and REFRIN) are inapplicable to the boundary scan.
5. 2DG VIDEO OUT related signals (R, G, B, REXT, CBU) are inapplicable to the boundary scan.
6. Execute the boundary scan in product chip mode and input the $\overline{\text{ASEMD}}$ pin to high during the $\overline{\text{RES}}$ pin assertion period. And make sure to fix the $\overline{\text{ASEMD}}$ pin at high while executing the boundary scan.

31.7 Usage Notes

1. An H-UDI command, once set, will not be modified as long as another command is not set again from the H-UDI. If the same command is to be set continuously, the command must be set after a command (BYPASS mode, etc.) that does not affect chip operations is once set.
2. In software standby mode and H-UDI module standby state, none of the functions in the H-UDI can be used. To retain the TAP status before and after standby mode, keep TCK high before entering standby mode.
3. Regardless of whether the H-UDI is used, make sure to keep the $\overline{\text{TRST}}$ pin low to initialize the H-UDI at power-on or in recovery from deep standby by the $\overline{\text{RES}}$ pin assertion.
4. If the TRST pin is asserted immediately after the setting of the TDO transition timing switching command and the negation of the RES pin, the TDO transition timing switching command is cleared. To avoid this case, make sure to put 20 tcyc or longer between the signal transition timing of the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ pins. For details, see section 31.5.3, TDO Output Timing.
5. When starting the TAP controller after the negation of the $\overline{\text{TRST}}$ pin, make sure to allow 200 ns or longer after the negation.
6. Please keep TMS pin high for 200 ns from $\overline{\text{TRST}}$ pin negation.

Section 32 List of Registers

This section gives information on the on-chip I/O registers of this LSI in the following structure.

1. Register Addresses (by functional module, in order of the corresponding section numbers)
- Registers are described by functional module, in order of the corresponding section numbers.
- Access to reserved addresses which are not described in this register address list is prohibited.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given assuming big endian.

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.

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 peripheral module, refer to the section on that on-chip peripheral module.

4. Notes when Writing to the On-Chip Peripheral Modules
- To access an on-chip module register, two or more peripheral module clock (Pϕ) cycles are required. Care must be taken in system design. When the CPU writes data to an on-chip peripheral register, the CPU executes the succeeding instruction without waiting for the completion of writing to the register. For example, a case in which the system enters software standby mode for power saving is described here. To make this transition, the SLEEP instruction must be performed after setting the STBY bit in the STBCR register to 1. However, a dummy read of the STBCR register is required before executing the SLEEP instruction. If a dummy read is omitted, the CPU executes the SLEEP instruction before the STBY bit is set to 1, thus the system enters sleep mode not software standby mode. A dummy read of the STBCR register is indispensable to complete writing to the STBY bit. To reflect the change made to an on-chip peripheral register while performing the succeeding instruction, dummy-read the register to which write instruction is applied and then execute the succeeding instruction.

32.1 Register Addresses (by functional module, in order of the corresponding section numbers)

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Multi-core proces-sor	CPU ID register	CPUIDR	32	H'FFFC1404	32
	Semaphore register 0	SEMR0	8	H'FFFC1E00	8
	Semaphore register 1	SEMR1	8	H'FFFC1E04	8
	Semaphore register 2	SEMR2	8	H'FFFC1E08	8
	Semaphore register 3	SEMR3	8	H'FFFC1E0C	8
	Semaphore register 4	SEMR4	8	H'FFFC1E10	8
	Semaphore register 5	SEMR5	8	H'FFFC1E14	8
	Semaphore register 6	SEMR6	8	H'FFFC1E18	8
	Semaphore register 7	SEMR7	8	H'FFFC1E1C	8
	Semaphore register 8	SEMR8	8	H'FFFC1E20	8
	Semaphore register 9	SEMR9	8	H'FFFC1E24	8
	Semaphore register 10	SEMR10	8	H'FFFC1E28	8
	Semaphore register 11	SEMR11	8	H'FFFC1E2C	8
	Semaphore register 12	SEMR12	8	H'FFFC1E30	8
	Semaphore register 13	SEMR13	8	H'FFFC1E34	8
	Semaphore register 14	SEMR14	8	H'FFFC1E38	8
	Semaphore register 15	SEMR15	8	H'FFFC1E3C	8
	Semaphore register 16	SEMR16	8	H'FFFC1E40	8
	Semaphore register 17	SEMR17	8	H'FFFC1E44	8
	Semaphore register 18	SEMR18	8	H'FFFC1E48	8
	Semaphore register 19	SEMR19	8	H'FFFC1E4C	8
	Semaphore register 20	SEMR20	8	H'FFFC1E50	8
	Semaphore register 21	SEMR21	8	H'FFFC1E54	8
	Semaphore register 22	SEMR22	8	H'FFFC1E58	8
	Semaphore register 23	SEMR23	8	H'FFFC1E5C	8
	Semaphore register 24	SEMR24	8	H'FFFC1E60	8
	Semaphore register 25	SEMR25	8	H'FFFC1E64	8
	Semaphore register 26	SEMR26	8	H'FFFC1E68	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Multi-core proce-ssor	Semaphore register 27	SEMR27	8	H'FFFC1E6C	8
	Semaphore register 28	SEMR28	8	H'FFFC1E70	8
	Semaphore register 29	SEMR29	8	H'FFFC1E74	8
	Semaphore register 30	SEMR30	8	H'FFFC1E78	8
	Semaphore register 31	SEMR31	8	H'FFFC1E7C	8
CPG	Frequency control register 0	FRQCR0	16	H'FFFE0010	16
	Frequency control register 1	FRQCR1	16	H'FFFE0012	16
INTC	Interrupt control register 0	C0ICR0	16	H'FFFD9400	16, 32
	Interrupt control register 1	C0ICR1	16	H'FFFD9402	16, 32
	Interrupt control register 2	C0ICR2	16	H'FFFD9404	16, 32
	IRQ interrupt request register	C0IRQRR	16	H'FFFD9406	16, 32
	PINT interrupt enable register	C0PINTER	16	H'FFFD9408	16, 32
	PINT interrupt request register	C0PIRR	16	H'FFFD940A	16, 32
	Bank control register	C0IBCR	16	H'FFFD940C	16, 32
	Bank number register	C0IBNR	16	H'FFFD940E	16, 32
	Interrupt priority register 01	C0IPR01	16	H'FFFD9418	16, 32
	Interrupt priority register 02	C0IPR02	16	H'FFFD941A	16, 32
	Interrupt priority register 05	C0IPR05	16	H'FFFD9420	16, 32
	Interrupt enable control register	C0INTER	16	H'FFFD9428	16, 32
	IRQ interrupt enable control register	C0IRQER	16	H'FFFD942A	16, 32
	Interrupt control register 0	C1ICR0	16	H'FFFD9500	16, 32
	Interrupt control register 1	C1ICR1	16	H'FFFD9502	16, 32
	Interrupt control register 2	C1ICR2	16	H'FFFD9504	16, 32
	IRQ interrupt request register	C1IRQRR	16	H'FFFD9506	16, 32
	PINT interrupt enable register	C1PINTER	16	H'FFFD9508	16, 32
	PINT interrupt request register	C1PIRR	16	H'FFFD950A	16, 32
	Bank control register	C1IBCR	16	H'FFFD950C	16, 32
	Bank number register	C1IBNR	16	H'FFFD950E	16, 32
	Interrupt priority register 01	C1IPR01	16	H'FFFD9518	16, 32
	Interrupt priority register 02	C1IPR02	16	H'FFFD951A	16, 32
	Interrupt priority register 05	C1IPR05	16	H'FFFD9520	16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
INTC	Interrupt enable control register	C1INTER	16	H'FFFD9528	16, 32
	IRQ interrupt enable control register	C1IRQER	16	H'FFFD952A	16, 32
	Inter-processor interrupt control register 15	C0IPCR15	16	H'FFFC1C00	16
	Inter-processor interrupt control register 14	C0IPCR14	16	H'FFFC1C02	16
	Inter-processor interrupt control register 13	C0IPCR13	16	H'FFFC1C04	16
	Inter-processor interrupt control register 12	C0IPCR12	16	H'FFFC1C06	16
	Inter-processor interrupt control register 11	C0IPCR11	16	H'FFFC1C08	16
	Inter-processor interrupt control register 10	C0IPCR10	16	H'FFFC1C0A	16
	Inter-processor interrupt control register 9	C0IPCR09	16	H'FFFC1C0C	16
	Inter-processor interrupt control register 8	C0IPCR08	16	H'FFFC1C0E	16
	Inter-processor interrupt enable register	C0IPER	16	H'FFFC1C10	16
	Inter-processor interrupt control register 15	C1IPCR15	16	H'FFFC1C20	16
	Inter-processor interrupt control register 14	C1IPCR14	16	H'FFFC1C22	16
	Inter-processor interrupt control register 13	C1IPCR13	16	H'FFFC1C24	16
	Inter-processor interrupt control register 12	C1IPCR12	16	H'FFFC1C26	16
	Inter-processor interrupt control register 11	C1IPCR11	16	H'FFFC1C28	16
	Inter-processor interrupt control register 10	C1IPCR10	16	H'FFFC1C2A	16
	Inter-processor interrupt control register 9	C1IPCR09	16	H'FFFC1C2C	16
	Inter-processor interrupt control register 8	C1IPCR08	16	H'FFFC1C2E	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
INTC	Inter-processor interrupt enable register	C1IPER	16	H'FFFC1C30	16
	Interrupt priority register 06	C0IPR06	16	H'FFFD9800	16, 32
	Interrupt priority register 07	C0IPR07	16	H'FFFD9802	16, 32
	Interrupt priority register 08	C0IPR08	16	H'FFFD9804	16, 32
	Interrupt priority register 09	C0IPR09	16	H'FFFD9806	16, 32
	Interrupt priority register 10	C0IPR10	16	H'FFFD9808	16, 32
	Interrupt priority register 11	C0IPR11	16	H'FFFD980A	16, 32
	Interrupt priority register 12	C0IPR12	16	H'FFFD980C	16, 32
	Interrupt priority register 13	C0IPR13	16	H'FFFD980E	16, 32
	Interrupt priority register 14	C0IPR14	16	H'FFFD9810	16, 32
	Interrupt priority register 15	C0IPR15	16	H'FFFD9812	16, 32
	Interrupt priority register 16	C0IPR16	16	H'FFFD9814	16, 32
	Interrupt priority register 17	C0IPR17	16	H'FFFD9816	16, 32
	Interrupt priority register 18	C0IPR18	16	H'FFFD9818	16, 32
	Interrupt priority register 19	C0IPR19	16	H'FFFD981A	16, 32
	Interrupt priority register 20	C0IPR20	16	H'FFFD981C	16, 32
	Interrupt priority register 21	C0IPR21	16	H'FFFD981E	16, 32
	Interrupt priority register 06	C1IPR06	16	H'FFFD9900	16, 32
	Interrupt priority register 07	C1IPR07	16	H'FFFD9902	16, 32
	Interrupt priority register 08	C1IPR08	16	H'FFFD9904	16, 32
	Interrupt priority register 09	C1IPR09	16	H'FFFD9906	16, 32
	Interrupt priority register 10	C1IPR10	16	H'FFFD9908	16, 32
	Interrupt priority register 11	C1IPR11	16	H'FFFD990A	16, 32
	Interrupt priority register 12	C1IPR12	16	H'FFFD990C	16, 32
	Interrupt priority register 13	C1IPR13	16	H'FFFD990E	16, 32
	Interrupt priority register 14	C1IPR14	16	H'FFFD9910	16, 32
	Interrupt priority register 15	C1IPR15	16	H'FFFD9912	16, 32
	Interrupt priority register 16	C1IPR16	16	H'FFFD9914	16, 32
	Interrupt priority register 17	C1IPR17	16	H'FFFD9916	16, 32
	Interrupt priority register 18	C1IPR18	16	H'FFFD9918	16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
INTC	Interrupt priority register 19	C1IPR19	16	H'FFFD991A	16, 32
	Interrupt priority register 20	C1IPR20	16	H'FFFD991C	16, 32
	Interrupt priority register 21	C1IPR21	16	H'FFFD991E	16, 32
	Interrupt detect control register 6	IDCNT6	16	H'FFFD9C0C	8, 16
	Interrupt detect control register 7	IDCNT7	16	H'FFFD9C0E	8, 16
	Interrupt detect control register 8	IDCNT8	16	H'FFFD9C10	8, 16
	Interrupt detect control register 9	IDCNT9	16	H'FFFD9C12	8, 16
	Interrupt detect control register 10	IDCNT10	16	H'FFFD9C14	8, 16
	Interrupt detect control register 11	IDCNT11	16	H'FFFD9C16	8, 16
	Interrupt detect control register 12	IDCNT12	16	H'FFFD9C18	8, 16
	Interrupt detect control register 13	IDCNT13	16	H'FFFD9C1A	8, 16
	Interrupt detect control register 14	IDCNT14	16	H'FFFD9C1C	8, 16
	Interrupt detect control register 15	IDCNT15	16	H'FFFD9C1E	8, 16
	Interrupt detect control register 16	IDCNT16	16	H'FFFD9C20	8, 16
	Interrupt detect control register 17	IDCNT17	16	H'FFFD9C22	8, 16
	Interrupt detect control register 18	IDCNT18	16	H'FFFD9C24	8, 16
	Interrupt detect control register 19	IDCNT19	16	H'FFFD9C26	8, 16
	Interrupt detect control register 20	IDCNT20	16	H'FFFD9C28	8, 16
	Interrupt detect control register 21	IDCNT21	16	H'FFFD9C2A	8, 16
	Interrupt detect control register 22	IDCNT22	16	H'FFFD9C2C	8, 16
	Interrupt detect control register 23	IDCNT23	16	H'FFFD9C2E	8, 16
	Interrupt detect control register 24	IDCNT24	16	H'FFFD9C30	8, 16
	Interrupt detect control register 25	IDCNT25	16	H'FFFD9C32	8, 16
	Interrupt detect control register 26	IDCNT26	16	H'FFFD9C34	8, 16
	Interrupt detect control register 27	IDCNT27	16	H'FFFD9C36	8, 16
	Interrupt detect control register 28	IDCNT28	16	H'FFFD9C38	8, 16
	Interrupt detect control register 29	IDCNT29	16	H'FFFD9C3A	8, 16
	Interrupt detect control register 30	IDCNT30	16	H'FFFD9C3C	8, 16
	Interrupt detect control register 31	IDCNT31	16	H'FFFD9C3E	8, 16
	Interrupt detect control register 32	IDCNT32	16	H'FFFD9C40	8, 16
	Interrupt detect control register 33	IDCNT33	16	H'FFFD9C42	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
INTC	Interrupt detect control register 34	IDCNT34	16	H'FFFD9C44	8, 16
	Interrupt detect control register 35	IDCNT35	16	H'FFFD9C46	8, 16
	Interrupt detect control register 36	IDCNT36	16	H'FFFD9C48	8, 16
	Interrupt detect control register 37	IDCNT37	16	H'FFFD9C4A	8, 16
	Interrupt detect control register 38	IDCNT38	16	H'FFFD9C4C	8, 16
	Interrupt detect control register 39	IDCNT39	16	H'FFFD9C4E	8, 16
	Interrupt detect control register 40	IDCNT40	16	H'FFFD9C50	8, 16
	Interrupt detect control register 41	IDCNT41	16	H'FFFD9C52	8, 16
	Interrupt detect control register 42	IDCNT42	16	H'FFFD9C54	8, 16
	Interrupt detect control register 43	IDCNT43	16	H'FFFD9C56	8, 16
	Interrupt detect control register 44	IDCNT44	16	H'FFFD9C58	8, 16
	Interrupt detect control register 45	IDCNT45	16	H'FFFD9C5A	8, 16
	Interrupt detect control register 46	IDCNT46	16	H'FFFD9C5C	8, 16
	Interrupt detect control register 47	IDCNT47	16	H'FFFD9C5E	8, 16
	Interrupt detect control register 48	IDCNT48	16	H'FFFD9C60	8, 16
	Interrupt detect control register 49	IDCNT49	16	H'FFFD9C62	8, 16
	Interrupt detect control register 50	IDCNT50	16	H'FFFD9C64	8, 16
	Interrupt detect control register 51	IDCNT51	16	H'FFFD9C66	8, 16
	Interrupt detect control register 52	IDCNT52	16	H'FFFD9C68	8, 16
	Interrupt detect control register 53	IDCNT53	16	H'FFFD9C6A	8, 16
	Interrupt detect control register 54	IDCNT54	16	H'FFFD9C6C	8, 16
	Interrupt detect control register 55	IDCNT55	16	H'FFFD9C6E	8, 16
	Interrupt detect control register 56	IDCNT56	16	H'FFFD9C70	8, 16
	Interrupt detect control register 57	IDCNT57	16	H'FFFD9C72	8, 16
	Interrupt detect control register 58	IDCNT58	16	H'FFFD9C74	8, 16
	Interrupt detect control register 59	IDCNT59	16	H'FFFD9C76	8, 16
	Interrupt detect control register 60	IDCNT60	16	H'FFFD9C78	8, 16
	Interrupt detect control register 61	IDCNT61	16	H'FFFD9C7A	8, 16
	Interrupt detect control register 62	IDCNT62	16	H'FFFD9C7C	8, 16
	Interrupt detect control register 63	IDCNT63	16	H'FFFD9C7E	8, 16
	Interrupt detect control register 64	IDCNT64	16	H'FFFD9C80	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
INTC	Interrupt detect control register 66	IDCNT66	16	H'FFFD9C84	8, 16
	Interrupt detect control register 67	IDCNT67	16	H'FFFD9C86	8, 16
	Interrupt detect control register 68	IDCNT68	16	H'FFFD9C88	8, 16
	Interrupt detect control register 69	IDCNT69	16	H'FFFD9C8A	8, 16
	Interrupt detect control register 70	IDCNT70	16	H'FFFD9C8C	8, 16
	Interrupt detect control register 71	IDCNT71	16	H'FFFD9C8E	8, 16
	Interrupt detect control register 72	IDCNT72	16	H'FFFD9C90	8, 16
	Interrupt detect control register 73	IDCNT73	16	H'FFFD9C92	8, 16
	Interrupt detect control register 74	IDCNT74	16	H'FFFD9C94	8, 16
	Interrupt detect control register 75	IDCNT75	16	H'FFFD9C96	8, 16
	Interrupt detect control register 76	IDCNT76	16	H'FFFD9C98	8, 16
	Interrupt detect control register 77	IDCNT77	16	H'FFFD9C9A	8, 16
	Interrupt detect control register 78	IDCNT78	16	H'FFFD9C9C	8, 16
	Interrupt detect control register 79	IDCNT79	16	H'FFFD9C9E	8, 16
	Interrupt detect control register 80	IDCNT80	16	H'FFFD9CA0	8, 16
	Interrupt detect control register 81	IDCNT81	16	H'FFFD9CA2	8, 16
	Interrupt detect control register 82	IDCNT82	16	H'FFFD9CA4	8, 16
	Interrupt detect control register 83	IDCNT83	16	H'FFFD9CA6	8, 16
	Interrupt detect control register 84	IDCNT84	16	H'FFFD9CA8	8, 16
	Interrupt detect control register 85	IDCNT85	16	H'FFFD9CAA	8, 16
	Interrupt detect control register 86	IDCNT86	16	H'FFFD9CAC	8, 16
	Interrupt detect control register 87	IDCNT87	16	H'FFFD9CAE	8, 16
	Interrupt detect control register 88	IDCNT88	16	H'FFFD9CB0	8, 16
	Interrupt detect control register 89	IDCNT89	16	H'FFFD9CB2	8, 16
	Interrupt detect control register 90	IDCNT90	16	H'FFFD9CB4	8, 16
	Interrupt detect control register 91	IDCNT91	16	H'FFFD9CB6	8, 16
	Interrupt detect control register 92	IDCNT92	16	H'FFFD9CB8	8, 16
	Interrupt detect control register 93	IDCNT93	16	H'FFFD9CBA	8, 16
	Interrupt detect control register 94	IDCNT94	16	H'FFFD9CBC	8, 16
	Interrupt detect control register 95	IDCNT95	16	H'FFFD9CBE	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
INTC	Interrupt detect control register 96	IDCNT96	16	H'FFFD9CC0	8, 16
	Interrupt detect control register 97	IDCNT97	16	H'FFFD9CC2	8, 16
	Interrupt detect control register 98	IDCNT98	16	H'FFFD9CC4	8, 16
	Interrupt detect control register 99	IDCNT99	16	H'FFFD9CC6	8, 16
	Interrupt detect control register 100	IDCNT100	16	H'FFFD9CC8	8, 16
	Interrupt detect control register 101	IDCNT101	16	H'FFFD9CCA	8, 16
	Interrupt detect control register 102	IDCNT102	16	H'FFFD9CCC	8, 16
	Interrupt detect control register 103	IDCNT103	16	H'FFFD9CCE	8, 16
	Interrupt detect control register 104	IDCNT104	16	H'FFFD9CD0	8, 16
	Interrupt detect control register 105	IDCNT105	16	H'FFFD9CD2	8, 16
	Interrupt detect control register 106	IDCNT106	16	H'FFFD9CD4	8, 16
	Interrupt detect control register 107	IDCNT107	16	H'FFFD9CD6	8, 16
	Interrupt detect control register 108	IDCNT108	16	H'FFFD9CD8	8, 16
	Interrupt detect control register 109	IDCNT109	16	H'FFFD9CDA	8, 16
	Interrupt detect control register 110	IDCNT110	16	H'FFFD9CDC	8, 16
	Interrupt detect control register 111	IDCNT111	16	H'FFFD9CDE	8, 16
	Interrupt detect control register 112	IDCNT112	16	H'FFFD9CE0	8, 16
	Interrupt detect control register 113	IDCNT113	16	H'FFFD9CE2	8, 16
	Interrupt detect control register 114	IDCNT114	16	H'FFFD9CE4	8, 16
	Interrupt detect control register 115	IDCNT115	16	H'FFFD9CE6	8, 16
	Interrupt detect control register 116	IDCNT116	16	H'FFFD9CE8	8, 16
	Interrupt detect control register 117	IDCNT117	16	H'FFFD9CEA	8, 16
	Interrupt detect control register 118	IDCNT118	16	H'FFFD9CEC	8, 16
	Interrupt detect control register 119	IDCNT119	16	H'FFFD9CEE	8, 16
	Interrupt detect control register 120	IDCNT120	16	H'FFFD9CF0	8, 16
	Interrupt detect control register 121	IDCNT121	16	H'FFFD9CF2	8, 16
	Interrupt detect control register 122	IDCNT122	16	H'FFFD9CF4	8, 16
	Interrupt detect control register 123	IDCNT123	16	H'FFFD9CF6	8, 16
	Interrupt detect control register 124	IDCNT124	16	H'FFFD9CF8	8, 16
	Interrupt detect control register 125	IDCNT125	16	H'FFFD9CFA	8, 16
	Interrupt detect control register 126	IDCNT126	16	H'FFFD9CFC	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
INTC	Interrupt detect control register 130	IDCNT130	16	H'FFFD9D04	8, 16
	Interrupt detect control register 131	IDCNT131	16	H'FFFD9D06	8, 16
	Interrupt detect control register 132	IDCNT132	16	H'FFFD9D08	8, 16
	Interrupt detect control register 133	IDCNT133	16	H'FFFD9D0A	8, 16
	Interrupt detect control register 134	IDCNT134	16	H'FFFD9D0C	8, 16
	Interrupt detect control register 135	IDCNT135	16	H'FFFD9D0E	8, 16
	Interrupt detect control register 136	IDCNT136	16	H'FFFD9D10	8, 16
	Interrupt detect control register 137	IDCNT137	16	H'FFFD9D12	8, 16
	Interrupt detect control register 138	IDCNT138	16	H'FFFD9D14	8, 16
	Interrupt detect control register 139	IDCNT139	16	H'FFFD9D16	8, 16
	DMA transfer request enable register 0	DREQER0	8	H'FFFE0800	8, 16, 32
	DMA transfer request enable register 1	DREQER1	8	H'FFFE0801	8, 16, 32
	DMA transfer request enable register 2	DREQER2	8	H'FFFE0802	8, 16, 32
	DMA transfer request enable register 3	DREQER3	8	H'FFFE0803	8, 16, 32
	DMA transfer request enable register 4	DREQER4	8	H'FFFE0804	8, 16, 32
	DMA transfer request enable register 5	DREQER5	8	H'FFFE0805	8, 16, 32
	DMA transfer request enable register 6	DREQER6	8	H'FFFE0806	8, 16, 32
	DMA transfer request enable register 7	DREQER7	8	H'FFFE0807	8, 16, 32
	DMA transfer request enable register 8	DREQER8	8	H'FFFE0808	8, 16, 32
UBC	Break address register_0	BAR_0	32	H'FFFC0400	32
	Break address mask register_0	BAMR_0	32	H'FFFC0404	32
	Break bus cycle register_0	BBR_0	16	H'FFFC04A0	16
	Break data register_0	BDR_0	32	H'FFFC0408	32
	Break data mask register_0	BDMR_0	32	H'FFFC040C	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
UBC	Break address register_1	BAR_1	32	H'FFFC0410	32
	Break address mask register_1	BAMR_1	32	H'FFFC0414	32
	Break bus cycle register_1	BBR_1	16	H'FFFC04B0	16
	Break data register_1	BDR_1	32	H'FFFC0418	32
	Break data mask register_1	BDMR_1	32	H'FFFC041C	32
	Break control register	BRCR	32	H'FFFC04C0	32
Cache	Cache control register 1	CCR1	32	H'FFFC1000	32
	Cache control register 2	CCR2	32	H'FFFC1004	32
BSC	CS0 control register	CS0CNT	32	H'FF420000	8, 16, 32
	CS0 recovery cycle setting register	CS0REC	32	H'FF420008	8, 16, 32
	CS1 control register	CS1CNT	32	H'FF420010	8, 16, 32
	CS1 recovery cycle setting register	CS1REC	32	H'FF420018	8, 16, 32
	CS2 control register	CS2CNT	32	H'FF420020	8, 16, 32
	CS2 recovery cycle setting register	CS2REC	32	H'FF420028	8, 16, 32
	CS3 control register	CS3CNT	32	H'FF420030	8, 16, 32
	CS3 recovery cycle setting register	CS3REC	32	H'FF420038	8, 16, 32
	CS4 control register	CS4CNT	32	H'FF420040	8, 16, 32
	CS4 recovery cycle setting register	CS4REC	32	H'FF420048	8, 16, 32
	CS5 control register	CS5CNT	32	H'FF420050	8, 16, 32
	CS5 recovery cycle setting register	CS5REC	32	H'FF420058	8, 16, 32
	SDRAMC0 control register	SDC0CNT	32	H'FF420100	8, 16, 32
	SDRAMC1 control register	SDC1CNT	32	H'FF420110	8, 16, 32
	CS0 mode register	CSMOD0	32	H'FF421000	8, 16, 32
	CS0 wait control register 1	CS1WCNT0	32	H'FF421004	8, 16, 32
	CS0 wait control register 2	CS2WCNT0	32	H'FF421008	8, 16, 32
	CS1 mode register	CSMOD1	32	H'FF421010	8, 16, 32
	CS1 wait control register 1	CS1WCNT1	32	H'FF421014	8, 16, 32
	CS1 wait control register 2	CS2WCNT1	32	H'FF421018	8, 16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
BSC	CS2 mode register	CSMOD2	32	H'FF421020	8, 16, 32
	CS2 wait control register 1	CS1WCNT2	32	H'FF421024	8, 16, 32
	CS2 wait control register 2	CS2WCNT2	32	H'FF421028	8, 16, 32
	CS3 mode register	CSMOD3	32	H'FF421030	8, 16, 32
	CS3 wait control register 1	CS1WCNT3	32	H'FF421034	8, 16, 32
	CS3 wait control register 2	CS2WCNT3	32	H'FF421038	8, 16, 32
	CS4 mode register	CSMOD4	32	H'FF421040	8, 16, 32
	CS4 wait control register 1	CS1WCNT4	32	H'FF421044	8, 16, 32
	CS4 wait control register 2	CS2WCNT4	32	H'FF421048	8, 16, 32
	CS5 mode register	CSMOD5	32	H'FF421050	8, 16, 32
	CS5 wait control register 1	CS1WCNT5	32	H'FF421054	8, 16, 32
	CS5 wait control register 2	CS2WCNT5	32	H'FF421058	8, 16, 32
	SDRAM refresh control register 0	SDRFCNT0	32	H'FF422000	8, 16, 32
	SDRAM refresh control register 1	SDRFCNT1	32	H'FF422004	16, 32
	SDRAM initialization register 0	SDIR0	32	H'FF422008	8, 16, 32
	SDRAM initialization register 1	SDIR1	32	H'FF42200C	8, 16, 32
	SDRAM power-down control register	SDPWDCNT	32	H'FF422010	8, 16, 32
	SDRAM deep-power-down control register	SDDPWDCNT	32	H'FF422014	8, 16, 32
	SDRAM0 address register	SD0ADR	32	H'FF422020	8, 16, 32
	SDRAM0 timing register	SD0TR	32	H'FF422024	8, 16, 32
	SDRAM0 mode register	SD0MOD	32	H'FF422028	16, 32
	SDRAM1 address register	SD1ADR	32	H'FF422040	8, 16, 32
	SDRAM1 timing register	SD1TR	32	H'FF422044	8, 16, 32
	SDRAM1 mode register	SD1MOD	32	H'FF422048	16, 32
	SDRAM status register	SDSTR	32	H'FF4220E4	8, 16, 32
	SDRAM clock stop control signal setting register	SDCKSCNT	32	H'FF4220E8	8, 16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA current source address register 0	DMCSADR0	32	H'FF460000	32
	DMA current destination address register 0	DMCDADR0	32	H'FF460004	32
	DMA current byte count register 0	DMCBCT0	32	H'FF460008	32
	DMA mode register 0	DMMOD0	32	H'FF46000C	32
	DMA reload source address register 0	DMRSADR0	32	H'FF460200	32
	DMA reload destination address register 0	DMRDADR0	32	H'FF460204	32
	DMA reload byte count register 0	DMRBCT0	32	H'FF460208	32
	DMA control register A0	DMACNTA0	32	H'FF460400	8, 16, 32
	DMA control register B0	DMACNTB0	32	H'FF460404	8, 16, 32
	DMA current source address register 1	DMCSADR1	32	H'FF460010	32
	DMA current destination address register 1	DMCDADR1	32	H'FF460014	32
	DMA current byte count register 1	DMCBCT1	32	H'FF460018	32
	DMA mode register 1	DMMOD1	32	H'FF46001C	32
	DMA reload source address register 1	DMRSADR1	32	H'FF460210	32
	DMA reload destination address register 1	DMRDADR1	32	H'FF460214	32
	DMA reload byte count register 1	DMRBCT1	32	H'FF460218	32
	DMA control register A1	DMACNTA1	32	H'FF460408	8, 16, 32
	DMA control register B1	DMACNTB1	32	H'FF46040C	8, 16, 32
	DMA current source address register 2	DMCSADR2	32	H'FF460020	32
	DMA current destination address register 2	DMCDADR2	32	H'FF460024	32
	DMA current byte count register 2	DMCBCT2	32	H'FF460028	32
	DMA mode register 2	DMMOD2	32	H'FF46002C	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA reload source address register 2	DMRSADR2	32	H'FF460220	32
	DMA reload destination address register 2	DMRDADR2	32	H'FF460224	32
	DMA reload byte count register 2	DMRBCT2	32	H'FF460228	32
	DMA control register A2	DMACNTA2	32	H'FF460410	8, 16, 32
	DMA control register B2	DMACNTB2	32	H'FF460414	8, 16, 32
	DMA current source address register 3	DMCSADR3	32	H'FF460030	32
	DMA current destination address register 3	DMCDADR3	32	H'FF460034	32
	DMA current byte count register 3	DMCBCT3	32	H'FF460038	32
	DMA mode register 3	DMMOD3	32	H'FF46003C	32
	DMA reload source address register 3	DMRSADR3	32	H'FF460230	32
	DMA reload destination address register 3	DMRDADR3	32	H'FF460234	32
	DMA reload byte count register 3	DMRBCT3	32	H'FF460238	32
	DMA control register A3	DMACNTA3	32	H'FF460418	8, 16, 32
	DMA control register B3	DMACNTB3	32	H'FF46041C	8, 16, 32
	DMA current source address register 4	DMCSADR4	32	H'FF460040	32
	DMA current destination address register 4	DMCDADR4	32	H'FF460044	32
	DMA current byte count register 4	DMCBCT4	32	H'FF460048	32
	DMA mode register 4	DMMOD4	32	H'FF46004C	32
	DMA reload source address register 4	DMRSADR4	32	H'FF460240	32
	DMA reload destination address register 4	DMRDADR4	32	H'FF460244	32
	DMA reload byte count register 4	DMRBCT4	32	H'FF460248	32
	DMA control register A4	DMACNTA4	32	H'FF460420	8, 16, 32
	DMA control register B4	DMACNTB4	32	H'FF460424	8, 16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA current source address register 5	DMCSADR5	32	H'FF460050	32
	DMA current destination address register 5	DMCDADR5	32	H'FF460054	32
	DMA current byte count register 5	DMCBCT5	32	H'FF460058	32
	DMA mode register 5	DMMOD5	32	H'FF46005C	32
	DMA reload source address register 5	DMRSADR5	32	H'FF460250	32
	DMA reload destination address register 5	DMRDADR5	32	H'FF460254	32
	DMA reload byte count register 5	DMRBCT5	32	H'FF460258	32
	DMA control register A5	DMACNTA5	32	H'FF460428	8, 16, 32
	DMA control register B5	DMACNTB5	32	H'FF46042C	8, 16, 32
	DMA current source address register 6	DMCSADR6	32	H'FF460060	32
	DMA current destination address register 6	DMCDADR6	32	H'FF460064	32
	DMA current byte count register 6	DMCBCT6	32	H'FF460068	32
	DMA mode register 6	DMMOD6	32	H'FF46006C	32
	DMA reload source address register 6	DMRSADR6	32	H'FF460260	32
	DMA reload destination address register 6	DMRDADR6	32	H'FF460264	32
	DMA reload byte count register 6	DMRBCT6	32	H'FF460268	32
	DMA control register A6	DMACNTA6	32	H'FF460430	8, 16, 32
	DMA control register B6	DMACNTB6	32	H'FF460434	8, 16, 32
	DMA current source address register 7	DMCSADR7	32	H'FF460070	32
	DMA current destination address register 7	DMCDADR7	32	H'FF460074	32
	DMA current byte count register 7	DMCBCT7	32	H'FF460078	32
	DMA mode register 7	DMMOD7	32	H'FF46007C	32
	DMA reload source address register 7	DMRSADR7	32	H'FF460270	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA reload destination address register 7	DMRDADR7	32	H'FF460274	32
	DMA reload byte count register 7	DMRBCT7	32	H'FF460278	32
	DMA control register A7	DMACNTA7	32	H'FF460438	8, 16, 32
	DMA control register B7	DMACNTB7	32	H'FF46043C	8, 16, 32
	DMA current source address register 8	DMCSADR8	32	H'FF460080	32
	DMA current destination address register 8	DMCDADR8	32	H'FF460084	32
	DMA current byte count register 8	DMCBCT8	32	H'FF460088	32
	DMA mode register 8	DMMOD8	32	H'FF46008C	32
	DMA reload source address register 8	DMRSADR8	32	H'FF460280	32
	DMA reload destination address register 8	DMRDADR8	32	H'FF460284	32
	DMA reload byte count register 8	DMRBCT8	32	H'FF460288	32
	DMA control register A8	DMACNTA8	32	H'FF460440	8, 16, 32
	DMA control register B8	DMACNTB8	32	H'FF460444	8, 16, 32
	DMA current source address register 9	DMCSADR9	32	H'FF460090	32
	DMA current destination address register 9	DMCDADR9	32	H'FF460094	32
	DMA current byte count register 9	DMCBCT9	32	H'FF460098	32
	DMA mode register 9	DMMOD9	32	H'FF46009C	32
	DMA reload source address register 9	DMRSADR9	32	H'FF460290	32
	DMA reload destination address register 9	DMRDADR9	32	H'FF460294	32
	DMA reload byte count register 9	DMRBCT9	32	H'FF460298	32
	DMA control register A9	DMACNTA9	32	H'FF460448	8, 16, 32
	DMA control register B9	DMACNTB9	32	H'FF46044C	8, 16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA current source address register 10	DMCSADR10	32	H'FF4600A0	32
	DMA current destination address register 10	DMCDADR10	32	H'FF4600A4	32
	DMA current byte count register 10	DMCBCT10	32	H'FF4600A8	32
	DMA mode register 10	DMMOD10	32	H'FF4600AC	32
	DMA reload source address register 10	DMRSADR10	32	H'FF4602A0	32
	DMA reload destination address register 10	DMRDADR10	32	H'FF4602A4	32
	DMA reload byte count register 10	DMRBCT10	32	H'FF4602A8	32
	DMA control register A10	DMACNTA10	32	H'FF460450	8, 16, 32
	DMA control register B10	DMACNTB10	32	H'FF460454	8, 16, 32
	DMA current source address register 11	DMCSADR11	32	H'FF4600B0	32
	DMA current destination address register 11	DMCDADR11	32	H'FF4600B4	32
	DMA current byte count register 11	DMCBCT11	32	H'FF4600B8	32
	DMA mode register 11	DMMOD11	32	H'FF4600BC	32
	DMA reload source address register 11	DMRSADR11	32	H'FF4602B0	32
	DMA reload destination address register 11	DMRDADR11	32	H'FF4602B4	32
	DMA reload byte count register 11	DMRBCT11	32	H'FF4602B8	32
	DMA control register A11	DMACNTA11	32	H'FF460458	8, 16, 32
	DMA control register B11	DMACNTB11	32	H'FF46045C	8, 16, 32
	DMA current source address register 12	DMCSADR12	32	H'FF4600C0	32
	DMA current destination address register 12	DMCDADR12	32	H'FF4600C4	32
	DMA current byte count register 12	DMCBCT12	32	H'FF4600C8	32
	DMA mode register 12	DMMOD12	32	H'FF4600CC	32
	DMA reload source address register 12	DMRSADR12	32	H'FF4602C0	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA reload destination address register 12	DMRDADR12	32	H'FF4602C4	32
	DMA reload byte count register 12	DMRBCT12	32	H'FF4602C8	32
	DMA control register A12	DMACNTA12	32	H'FF460460	8, 16, 32
	DMA control register B12	DMACNTB12	32	H'FF460464	8, 16, 32
	DMA current source address register 13	DMCSADR13	32	H'FF4600D0	32
	DMA current destination address register 13	DMCDADR13	32	H'FF4600D4	32
	DMA current byte count register 13	DMCBCT13	32	H'FF4600D8	32
	DMA mode register 13	DMMOD13	32	H'FF4600DC	32
	DMA reload source address register 13	DMRSADR13	32	H'FF4602D0	32
	DMA reload destination address register 13	DMRDADR13	32	H'FF4602D4	32
	DMA reload byte count register 13	DMRBCT13	32	H'FF4602D8	32
	DMA control register A13	DMACNTA13	32	H'FF460468	8, 16, 32
	DMA control register B13	DMACNTB13	32	H'FF46046C	8, 16, 32
	DMA activation control register	DMSCNT	32	H'FF460500	8, 16, 32
	DMA interrupt control register	DMICNT	32	H'FF460508	8, 16, 32
	DMA common interrupt control register	DMICNTA	32	H'FF46050C	8, 16, 32
	DMA interrupt status register	DMISTS	32	H'FF460510	8, 16, 32
	DMA transfer end detection register	DMEDET	32	H'FF460514	8, 16, 32
	DMA arbitration status register	DMASTS	32	H'FF460518	8, 16, 32
	DMA two-dimensional addressing column setting register 0	DM2DCLM0	32	H'FF460600	32
	DMA two-dimensional addressing row setting register 0	DM2DROW0	32	H'FF460604	32
	DMA two-dimensional addressing block setting register 0	DM2DBLK0	32	H'FF460608	32
	DMA two-dimensional addressing next row offset register 0	DM2DNROST0	32	H'FF46060C	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA two-dimensional addressing next block offset register 0	DM2DNBOST0	32	H'FF460610	32
	DMA two-dimensional addressing next line offset register 0	DM2DNLOST0	32	H'FF460614	32
	DMA reload two-dimensional addressing column setting register 0	DMR2DCLM0	32	H'FF460A00	32
	DMA reload two-dimensional addressing row setting register 0	DMR2DROW0	32	H'FF460A04	32
	DMA reload two-dimensional addressing block setting register 0	DMR2DBLK0	32	H'FF460A08	32
	DMA reload two-dimensional addressing next row offset register 0	DMR2DNROST0	32	H'FF460A0C	32
	DMA reload two-dimensional addressing next block offset register 0	DMR2DNBOST0	32	H'FF460A10	32
	DMA reload two-dimensional addressing next line offset register 0	DMR2DNLOST0	32	H'FF460A14	32
	DMA two-dimensional addressing column setting register 1	DM2DCLM1	32	H'FF460620	32
	DMA two-dimensional addressing row setting register 1	DM2DROW1	32	H'FF460624	32
	DMA two-dimensional addressing block setting register 1	DM2DBLK1	32	H'FF460628	32
	DMA two-dimensional addressing next row offset register 1	DM2DNROST1	32	H'FF46062C	32
	DMA two-dimensional addressing next block offset register 1	DM2DNBOST1	32	H'FF460630	32
	DMA two-dimensional addressing next line offset register 1	DM2DNLOST1	32	H'FF460634	32
	DMA reload two-dimensional addressing column setting register 1	DMR2DCLM1	32	H'FF460A20	32
	DMA reload two-dimensional addressing row setting register 1	DMR2DROW1	32	H'FF460A24	32
	DMA reload two-dimensional addressing block setting register 1	DMR2DBLK1	32	H'FF460A28	32
	DMA reload two-dimensional addressing next row offset register 1	DMR2DNROST1	32	H'FF460A2C	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA reload two-dimensional addressing next block offset register 1	DMR2DNBOST1	32	H'FF460A30	32
	DMA reload two-dimensional addressing next line offset register 1	DMR2DNLOST1	32	H'FF460A34	32
	DMA two-dimensional addressing column setting register 2	DM2DCLM2	32	H'FF460640	32
	DMA two-dimensional addressing row setting register 2	DM2DROW2	32	H'FF460644	32
	DMA two-dimensional addressing block setting register 2	DM2DBLK2	32	H'FF460648	32
	DMA two-dimensional addressing next row offset register 2	DM2DNROST2	32	H'FF46064C	32
	DMA two-dimensional addressing next block offset register 2	DM2DNBOST2	32	H'FF460650	32
	DMA two-dimensional addressing next line offset register 2	DM2DNLOST2	32	H'FF460654	32
	DMA reload two-dimensional addressing column setting register 2	DMR2DCLM2	32	H'FF460A40	32
	DMA reload two-dimensional addressing row setting register 2	DMR2DROW2	32	H'FF460A44	32
	DMA reload two-dimensional addressing block setting register 2	DMR2DBLK2	32	H'FF460A48	32
	DMA reload two-dimensional addressing next row offset register 2	DMR2DNROST2	32	H'FF460A4C	32
	DMA reload two-dimensional addressing next block offset register 2	DMR2DNBOST2	32	H'FF460A50	32
	DMA reload two-dimensional addressing next line offset register 2	DMR2DNLOST2	32	H'FF460A54	32
	DMA two-dimensional addressing column setting register 3	DM2DCLM3	32	H'FF460660	32
	DMA two-dimensional addressing row setting register 3	DM2DROW3	32	H'FF460664	32
	DMA two-dimensional addressing block setting register 3	DM2DBLK3	32	H'FF460668	32
	DMA two-dimensional addressing next row offset register 3	DM2DNROST3	32	H'FF46066C	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA two-dimensional addressing next block offset register 3	DM2DNBOST3	32	H'FF460670	32
	DMA two-dimensional addressing next line offset register 3	DM2DNLOST3	32	H'FF460674	32
	DMA reload two-dimensional addressing column setting register 3	DMR2DCLM3	32	H'FF460A60	32
	DMA reload two-dimensional addressing row setting register 3	DMR2DROW3	32	H'FF460A64	32
	DMA reload two-dimensional addressing block setting register 3	DMR2DBLK3	32	H'FF460A68	32
	DMA reload two-dimensional addressing next row offset register 3	DMR2DNROST3	32	H'FF460A6C	32
	DMA reload two-dimensional addressing next block offset register 3	DMR2DNBOST3	32	H'FF460A70	32
	DMA reload two-dimensional addressing next line offset register 3	DMR2DNLOST3	32	H'FF460A74	32
	DMA two-dimensional addressing column setting register 4	DM2DCLM4	32	H'FF460680	32
	DMA two-dimensional addressing row setting register 4	DM2DROW4	32	H'FF460684	32
	DMA two-dimensional addressing block setting register 4	DM2DBLK4	32	H'FF460688	32
	DMA two-dimensional addressing next row offset register 4	DM2DNROST4	32	H'FF46068C	32
	DMA two-dimensional addressing next block offset register 4	DM2DNBOST4	32	H'FF460690	32
	DMA two-dimensional addressing next line offset register 4	DM2DNLOST4	32	H'FF460694	32
	DMA reload two-dimensional addressing column setting register 4	DMR2DCLM4	32	H'FF460A80	32
	DMA reload two-dimensional addressing row setting register 4	DMR2DROW4	32	H'FF460A84	32
	DMA reload two-dimensional block setting register 4	DMR2DBLK4	32	H'FF460A88	32
	DMA reload two-dimensional addressing next row offset register 4	DMR2DNROST4	32	H'FF460A8C	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA reload two-dimensional addressing next block offset register 4	DMR2DNBOST4	32	H'FF460A90	32
	DMA reload two-dimensional addressing next line offset register 4	DMR2DNLOST4	32	H'FF460A94	32
	DMA two-dimensional addressing column setting register 5	DM2DCLM5	32	H'FF4606A0	32
	DMA two-dimensional addressing row setting register 5	DM2DROW5	32	H'FF4606A4	32
	DMA two-dimensional addressing block setting register 5	DM2DBLK5	32	H'FF4606A8	32
	DMA two-dimensional addressing next row offset register 5	DM2DNROST5	32	H'FF4606AC	32
	DMA two-dimensional addressing next block offset register 5	DM2DNBOST5	32	H'FF4606B0	32
	DMA two-dimensional addressing next line offset register 5	DM2DNLOST5	32	H'FF4606B4	32
	DMA reload two-dimensional addressing column setting register 5	DMR2DCLM5	32	H'FF460AA0	32
	DMA reload two-dimensional addressing row setting register 5	DMR2DROW5	32	H'FF460AA4	32
	DMA reload two-dimensional block setting register 5	DMR2DBLK5	32	H'FF460AA8	32
	DMA reload two-dimensional addressing next row offset register 5	DMR2DNROST5	32	H'FF460AAC	32
	DMA reload two-dimensional addressing next block offset register 5	DMR2DNBOST5	32	H'FF460AB0	32
	DMA reload two-dimensional addressing next line offset register 5	DMR2DNLOST5	32	H'FF460AB4	32
	DMA two-dimensional addressing column setting register 6	DM2DCLM6	32	H'FF4606C0	32
	DMA two-dimensional addressing row setting register 6	DM2DROW6	32	H'FF4606C4	32
	DMA two-dimensional addressing block setting register 6	DM2DBLK6	32	H'FF4606C8	32
	DMA two-dimensional addressing next row offset register 6	DM2DNROST6	32	H'FF4606CC	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA two-dimensional addressing next block offset register 6	DM2DNBOST6	32	H'FF4606D0	32
	DMA two-dimensional addressing next line offset register 6	DM2DNLOST6	32	H'FF4606D4	32
	DMA reload two-dimensional addressing column setting register 6	DMR2DCLM6	32	H'FF460AC0	32
	DMA reload two-dimensional addressing row setting register 6	DMR2DROW6	32	H'FF460AC4	32
	DMA reload two-dimensional block setting register 6	DMR2DBLK6	32	H'FF460AC8	32
	DMA reload two-dimensional addressing next row offset register 6	DMR2DNROST6	32	H'FF460ACC	32
	DMA reload two-dimensional addressing next block offset register 6	DMR2DNBOST6	32	H'FF460AD0	32
	DMA reload two-dimensional addressing next line offset register 6	DMR2DNLOST6	32	H'FF460AD4	32
	DMA two-dimensional addressing column setting register 7	DM2DCLM7	32	H'FF4606E0	32
	DMA two-dimensional addressing row setting register 7	DM2DROW7	32	H'FF4606E4	32
	DMA two-dimensional addressing block setting register 7	DM2DBLK7	32	H'FF4606E8	32
	DMA two-dimensional addressing row setting register 7	DM2DNROST7	32	H'FF4606EC	32
	DMA two-dimensional addressing next block offset register 7	DM2DNBOST7	32	H'FF4606F0	32
	DMA two-dimensional addressing next line offset register 7	DM2DNLOST7	32	H'FF4606F4	32
	DMA reload two-dimensional addressing column setting register 7	DMR2DCLM7	32	H'FF460AE0	32
	DMA reload two-dimensional addressing row setting register 7	DMR2DROW7	32	H'FF460AE4	32
	DMA reload two-dimensional addressing block setting register 7	DMR2DBLK7	32	H'FF460AE8	32
	DMA reload two-dimensional addressing next row offset register 7	DMR2DNROST7	32	H'FF460AEC	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA reload two-dimensional addressing next block offset register 7	DMR2DNBOST7	32	H'FF460AF0	32
	DMA reload two-dimensional addressing next line offset register 7	DMR2DNLOST7	32	H'FF460AF4	32
MTU2	Timer control register_0	TCR_0	8	H'FFFE2300	8
	Timer mode register_0	TMDR_0	8	H'FFFE2301	8
	Timer I/O control register H_0	TIORH_0	8	H'FFFE2302	8
	Timer I/O control register L_0	TIORL_0	8	H'FFFE2303	8
	Timer interrupt enable register_0	TIER_0	8	H'FFFE2304	8
	Timer status register_0	TSR_0	8	H'FFFE2305	8
	Timer counter_0	TCNT_0	16	H'FFFE2306	16
	Timer general register A_0	TGRA_0	16	H'FFFE2308	16
	Timer general register B_0	TGRB_0	16	H'FFFE230A	16
	Timer general register C_0	TGRC_0	16	H'FFFE230C	16
	Timer general register D_0	TGRD_0	16	H'FFFE230E	16
	Timer general register E_0	TGRE_0	16	H'FFFE2320	16
	Timer general register F_0	TGRF_0	16	H'FFFE2322	16
	Timer interrupt enable register 2_0	TIER2_0	8	H'FFFE2324	8
	Timer status register 2_0	TSR2_0	8	H'FFFE2325	8
	Timer buffer operation transfer mode register_0	TBTM_0	8	H'FFFE2326	8
	Timer control register_1	TCR_1	8	H'FFFE2380	8
	Timer mode register_1	TMDR_1	8	H'FFFE2381	8
	Timer I/O control register_1	TIOR_1	8	H'FFFE2382	8
	Timer interrupt enable register_1	TIER_1	8	H'FFFE2384	8
	Timer status register_1	TSR_1	8	H'FFFE2385	8
	Timer counter_1	TCNT_1	16	H'FFFE2386	16
	Timer general register A_1	TGRA_1	16	H'FFFE2388	16
	Timer general register B_1	TGRB_1	16	H'FFFE238A	16
	Timer input capture control register	TICCR	8	H'FFFE2390	8
	Timer control register_2	TCR_2	8	H'FFFE2000	8
	Timer mode register_2	TMDR_2	8	H'FFFE2001	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer I/O control register_2	TIOR_2	8	H'FFFE2002	8
	Timer interrupt enable register_2	TIER_2	8	H'FFFE2004	8
	Timer status register_2	TSR_2	8	H'FFFE2005	8
	Timer counter_2	TCNT_2	16	H'FFFE2006	16
	Timer general register A_2	TGRA_2	16	H'FFFE2008	16
	Timer general register B_2	TGRB_2	16	H'FFFE200A	16
	Timer control register_3	TCR_3	8	H'FFFE2200	8
	Timer mode register_3	TMDR_3	8	H'FFFE2202	8
	Timer I/O control register H_3	TIORH_3	8	H'FFFE2204	8
	Timer I/O control register L_3	TIORL_3	8	H'FFFE2205	8
	Timer interrupt enable register_3	TIER_3	8	H'FFFE2208	8
	Timer status register_3	TSR_3	8	H'FFFE222C	8
	Timer counter_3	TCNT_3	16	H'FFFE2210	16
	Timer general register A_3	TGRA_3	16	H'FFFE2218	16
	Timer general register B_3	TGRB_3	16	H'FFFE221A	16
	Timer general register C_3	TGRC_3	16	H'FFFE2224	16
	Timer general register D_3	TGRD_3	16	H'FFFE2226	16
	Timer buffer operation transfer mode register_3	TBTM_3	8	H'FFFE2238	8
	Timer control register_4	TCR_4	8	H'FFFE2201	8
	Timer mode register_4	TMDR_4	8	H'FFFE2203	8
	Timer I/O control register H_4	TIORH_4	8	H'FFFE2206	8
	Timer I/O control register L_4	TIORL_4	8	H'FFFE2207	8
	Timer interrupt enable register_4	TIER_4	8	H'FFFE2209	8
	Timer status register_4	TSR_4	8	H'FFFE222D	8
	Timer counter_4	TCNT_4	16	H'FFFE2212	16
	Timer general register A_4	TGRA_4	16	H'FFFE221C	16
	Timer general register B_4	TGRB_4	16	H'FFFE221E	16
	Timer general register C_4	TGRC_4	16	H'FFFE2228	16
	Timer general register D_4	TGRD_4	16	H'FFFE222A	16
	Timer buffer operation transfer mode register_4	TBTM_4	8	H'FFFE2239	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer A/D converter start request control register	TADCR	16	H'FFFE2240	16
	Timer A/D converter start request cycle set register A_4	TADCORA_4	16	H'FFFE2244	16
	Timer A/D converter start request cycle set register B_4	TADCORB_4	16	H'FFFE2246	16
	Timer A/D converter start request cycle set buffer register A_4	TADCOBRA_4	16	H'FFFE2248	16
	Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	16	H'FFFE224A	16
	Timer start register	TSTR	8	H'FFFE2280	8
	Timer synchronous register	TSYR	8	H'FFFE2281	8
	Timer read/write enable register	TRWER	8	H'FFFE2284	8
	Timer output master enable register	TOER	8	H'FFFE220A	8
	Timer output control register 1	TOCR1	8	H'FFFE220E	8
	Timer output control register 2	TOCR2	8	H'FFFE220F	8
	Timer gate control register	TGCR	8	H'FFFE220D	8
	Timer cycle data register	TCDR	16	H'FFFE2214	16
	Timer dead time data register	TDDR	16	H'FFFE2216	16
	Timer subcounter	TCNTS	16	H'FFFE2220	16
	Timer cycle buffer register	TCBR	16	H'FFFE2222	16
	Timer interrupt skipping set register	TITCR	8	H'FFFE2230	8
	Timer interrupt skipping counter	TITCNT	8	H'FFFE2231	8
	Timer buffer transfer set register	TBTER	8	H'FFFE2232	8
	Timer dead time enable register	TDER	8	H'FFFE2234	8
	Timer waveform control register	TWCR	8	H'FFFE2260	8
	Timer output level buffer register	TOLBR	8	H'FFFE2236	8
CMT	Compare match timer start register 01	CMSTR01	16	H'FFFE3000	16
	Compare match timer control/ status register_0	CMCSR0	16	H'FFFE3002	16
	Compare match counter_0	CMCNT0	16	H'FFFE3004	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
CMT	Compare match constant register_0	CMCOR0	16	H'FFFE3006	8, 16
	Compare match timer control/ status register_1	CMCSR1	16	H'FFFE3008	16
	Compare match counter_1	CMCNT1	16	H'FFFE300A	8, 16
	Compare match constant register_1	CMCOR1	16	H'FFFE300C	8, 16
	Compare match timer start register 23	CMSTR23	16	H'FFFE3400	16
	Compare match timer control/ status register_2	CMCSR2	16	H'FFFE3402	16
	Compare match counter_2	CMCNT2	16	H'FFFE3404	8, 16
	Compare match constant register_2	CMCOR2	16	H'FFFE3406	8, 16
	Compare match timer control/ status register_3	CMCSR3	16	H'FFFE3408	16
	Compare match counter_3	CMCNT3	16	H'FFFE340A	8, 16
	Compare match constant register_3	CMCOR3	16	H'FFFE340C	8, 16
WDT	Watchdog timer control/status register 0	WTCSR0	16	H'FFFE0000	16
	Watchdog timer counter 0	WTCNT0	16	H'FFFE0002	16
	Watchdog reset control/status register 0	WRCSR0	16	H'FFFE0004	16
	Watchdog timer control/status register 1	WTCSR1	16	H'FFFE0008	16
	Watchdog timer counter 1	WTCNT1	16	H'FFFE000A	16
	Watchdog reset control/status register 1	WRCSR1	16	H'FFFE000C	16
RTC	64-Hz counter	R64CNT	8	H'FFFE1000	8
	Second counter	RSECCNT	8	H'FFFE1002	8
	Minute counter	RMINCNT	8	H'FFFE1004	8
	Hour counter	RHRCNT	8	H'FFFE1006	8
	Day of week counter	RWKCNT	8	H'FFFE1008	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
RTC	Date counter	RDAYCNT	8	H'FFFE100A	8
	Month counter	RMONCNT	8	H'FFFE100C	8
	Year counter	RYRCNT	16	H'FFFE100E	16
	Second alarm register	RSECAR	8	H'FFFE1010	8
	Minute alarm register	RMINAR	8	H'FFFE1012	8
	Hour alarm register	RHRAR	8	H'FFFE1014	8
	Day of week alarm register	RWKAR	8	H'FFFE1016	8
	Date alarm register	RDAYAR	8	H'FFFE1018	8
	Month alarm register	RMONAR	8	H'FFFE101A	8
	Year alarm register	RYRAR	16	H'FFFE1020	16
	RTC control register 1	RCR1	8	H'FFFE101C	8
	RTC control register 2	RCR2	8	H'FFFE101E	8
	RTC control register 3	RCR3	8	H'FFFE1024	8
SCIF	Serial mode register_0	SCSMR_0	16	H'FFFE8000	16
	Bit rate register_0	SCBRR_0	8	H'FFFE8004	8
	Serial control register_0	SCSCR_0	16	H'FFFE8008	16
	Transmit FIFO data register_0	SCFTDR_0	8	H'FFFE800C	8
	Serial status register_0	SCFSR_0	16	H'FFFE8010	16
	Receive FIFO data register_0	SCFRDR_0	8	H'FFFE8014	8
	FIFO control register_0	SCFCR_0	16	H'FFFE8018	16
	FIFO data count set register_0	SCFDR_0	16	H'FFFE801C	16
	Serial port register_0	SCSPTR_0	16	H'FFFE8020	16
	Line status register_0	SCLSR_0	16	H'FFFE8024	16
	Serial extension mode register_0	SCEMR_0	16	H'FFFE8028	16
	Serial mode register_1	SCSMR_1	16	H'FFFE8800	16
	Bit rate register_1	SCBRR_1	8	H'FFFE8804	8
	Serial control register_1	SCSCR_1	16	H'FFFE8808	16
	Transmit FIFO data register_1	SCFTDR_1	8	H'FFFE880C	8
	Serial status register_1	SCFSR_1	16	H'FFFE8810	16
	Receive FIFO data register_1	SCFRDR_1	8	H'FFFE8814	8
	FIFO control register_1	SCFCR_1	16	H'FFFE8818	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCIF	FIFO data count register_1	SCFDR_1	16	H'FFFE881C	16
	Serial port register_1	SCSPTR_1	16	H'FFFE8820	16
	Line status register_1	SCLSR_1	16	H'FFFE8824	16
	Serial extension mode register_1	SCEMR_1	16	H'FFFE8828	16
	Serial mode register_2	SCSMR_2	16	H'FFFE9000	16
	Bit rate register_2	SCBRR_2	8	H'FFFE9004	8
	Serial control register_2	SCSCR_2	16	H'FFFE9008	16
	Transmit FIFO data register_2	SCFTDR_2	8	H'FFFE900C	8
	Serial status register_2	SCFSR_2	16	H'FFFE9010	16
	Receive FIFO data register_2	SCFRDR_2	8	H'FFFE9014	8
	FIFO control register_2	SCFCR_2	16	H'FFFE9018	16
	FIFO data count register_2	SCFDR_2	16	H'FFFE901C	16
	Serial port register_2	SCSPTR_2	16	H'FFFE9020	16
	Line status register_2	SCLSR_2	16	H'FFFE9024	16
	Serial extension mode register_2	SCEMR_2	16	H'FFFE9028	16
	Serial mode register_3	SCSMR_3	16	H'FFFE9800	16
	Bit rate register_3	SCBRR_3	8	H'FFFE9804	8
	Serial control register_3	SCSCR_3	16	H'FFFE9808	16
	Transmit FIFO data register_3	SCFTDR_3	8	H'FFFE980C	8
	Serial status register_3	SCFSR_3	16	H'FFFE9810	16
	Receive FIFO data register_3	SCFRDR_3	8	H'FFFE9814	8
	FIFO control register_3	SCFCR_3	16	H'FFFE9818	16
	FIFO data count register_3	SCFDR_3	16	H'FFFE981C	16
	Serial port register_3	SCSPTR_3	16	H'FFFE9820	16
	Line status register_3	SCLSR_3	16	H'FFFE9824	16
	Serial extension mode register_3	SCEMR_3	16	H'FFFE9828	16
	Serial mode register_4	SCSMR_4	16	H'FFFEA000	16
	Bit rate register_4	SCBRR_4	8	H'FFFEA004	8
	Serial control register_4	SCSCR_4	16	H'FFFEA008	16
	Transmit FIFO data register_4	SCFTDR_4	8	H'FFFEA00C	8
	Serial status register_4	SCFSR_4	16	H'FFFEA010	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCIF	Receive FIFO data register_4	SCFRDR_4	8	H'FFFEA014	8
	FIFO control register_4	SCFCR_4	16	H'FFFEA018	16
	FIFO data count register_4	SCFDR_4	16	H'FFFEA01C	16
	Serial port register_4	SCSPTR_4	16	H'FFFEA020	16
	Line status register_4	SCLSR_4	16	H'FFFEA024	16
	Serial extension mode register_4	SCEMR_4	16	H'FFFEA028	16
	Serial mode register_5	SCSMR_5	16	H'FFFEA800	16
	Bit rate register_5	SCBRR_5	8	H'FFFEA804	8
	Serial control register_5	SCSCR_5	16	H'FFFEA808	16
	Transmit FIFO data register_5	SCFTDR_5	8	H'FFFEA80C	8
	Serial status register_5	SCFSR_5	16	H'FFFEA810	16
	Receive FIFO data register_5	SCFRDR_5	8	H'FFFEA814	8
	FIFO control register_5	SCFCR_5	16	H'FFFEA818	16
	FIFO data count register_5	SCFDR_5	16	H'FFFEA81C	16
	Serial port register_5	SCSPTR_5	16	H'FFFEA820	16
	Line status register_5	SCLSR_5	16	H'FFFEA824	16
	Serial extension mode register_5	SCEMR_5	16	H'FFFEA828	16
SSU	SS control register H_0	SSCRH_0	8	H'FFFE7000	8, 16
	SS control register L_0	SSCRL_0	8	H'FFFE7001	8
	SS mode register_0	SSMR_0	8	H'FFFE7002	8, 16
	SS enable register_0	SSER_0	8	H'FFFE7003	8
	SS status register_0	SSSR_0	8	H'FFFE7004	8, 16
	SS control register 2_0	SSCR2_0	8	H'FFFE7005	8
	SS transmit data register 0_0	SSTDR0_0	8	H'FFFE7006	8, 16
	SS transmit data register 1_0	SSTDR1_0	8	H'FFFE7007	8
	SS transmit data register 2_0	SSTDR2_0	8	H'FFFE7008	8, 16
	SS transmit data register 3_0	SSTDR3_0	8	H'FFFE7009	8
	SS receive data register 0_0	SSRDR0_0	8	H'FFFE700A	8, 16
	SS receive data register 1_0	SSRDR1_0	8	H'FFFE700B	8
	SS receive data register 2_0	SSRDR2_0	8	H'FFFE700C	8, 16
	SS receive data register 3_0	SSRDR3_0	8	H'FFFE700D	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
SSU	SS control register H_1	SSCRH_1	8	H'FFFE7800	8, 16
	SS control register L_1	SSCRL_1	8	H'FFFE7801	8
	SS mode register_1	SSMR_1	8	H'FFFE7802	8, 16
	SS enable register_1	SSER_1	8	H'FFFE7803	8
	SS status register_1	SSSR_1	8	H'FFFE7804	8, 16
	SS control register 2_1	SSCR2_1	8	H'FFFE7805	8
	SS transmit data register 0_1	SSTDR0_1	8	H'FFFE7806	8, 16
	SS transmit data register 1_1	SSTDR1_1	8	H'FFFE7807	8
	SS transmit data register 2_1	SSTDR2_1	8	H'FFFE7808	8, 16
	SS transmit data register 3_1	SSTDR3_1	8	H'FFFE7809	8
	SS receive data register 0_1	SSRDR0_1	8	H'FFFE780A	8, 16
	SS receive data register 1_1	SSRDR1_1	8	H'FFFE780B	8
	SS receive data register 2_1	SSRDR2_1	8	H'FFFE780C	8, 16
	SS receive data register 3_1	SSRDR3_1	8	H'FFFE780D	8
IIC3	I^2C bus control register 1_0	ICCR1_0	8	H'FFFEE000	8
	I^2C bus control register 2_0	ICCR2_0	8	H'FFFEE001	8
	I^2C bus mode register_0	ICMR_0	8	H'FFFEE002	8
	I^2C bus interrupt enable register_0	ICIER_0	8	H'FFFEE003	8
	I^2C bus status register_0	ICSR_0	8	H'FFFEE004	8
	Slave address register_0	SAR_0	8	H'FFFEE005	8
	I^2C bus transmit data register_0	ICDRT_0	8	H'FFFEE006	8
	I^2C bus receive data register_0	ICDRR_0	8	H'FFFEE007	8
	NF2CYC register_0	NF2CYC_0	8	H'FFFEE008	8
	I^2C bus control register 1_1	ICCR1_1	8	H'FFFEE400	8
	I^2C bus control register 2_1	ICCR2_1	8	H'FFFEE401	8
	I^2C bus mode register_1	ICMR_1	8	H'FFFEE402	8
	I^2C bus interrupt enable register_1	ICIER_1	8	H'FFFEE403	8
	I^2C bus status register_1	ICSR_1	8	H'FFFEE404	8
	Slave address register_1	SAR_1	8	H'FFFEE405	8
	I^2C bus transmit data register_1	ICDRT_1	8	H'FFFEE406	8
	I^2C bus receive data register_1	ICDRR_1	8	H'FFFEE407	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
IIC3	NF2CYC register_1	NF2CYC_1	8	H'FFFEE408	8
	I²C bus control register 1_2	ICCR1_2	8	H'FFFEE800	8
	I²C bus control register 2_2	ICCR2_2	8	H'FFFEE801	8
	I²C bus mode register_2	ICMR_2	8	H'FFFEE802	8
	I²C bus interrupt enable register_2	ICIER_2	8	H'FFFEE803	8
	I²C bus status register_2	ICSR_2	8	H'FFFEE804	8
	Slave address register_2	SAR_2	8	H'FFFEE805	8
	I²C bus transmit data register_2	ICDRT_2	8	H'FFFEE806	8
	I²C bus receive data register_2	ICDRR_2	8	H'FFFEE807	8
	NF2CYC register_2	NF2CYC_2	8	H'FFFEE808	8
	I²C bus control register 1_3	ICCR1_3	8	H'FFFEEC00	8
	I²C bus control register 2_3	ICCR2_3	8	H'FFFEEC01	8
	I²C bus mode register_3	ICMR_3	8	H'FFFEEC02	8
	I²C bus interrupt enable register_3	ICIER_3	8	H'FFFEEC03	8
	I²C bus status register_3	ICSR_3	8	H'FFFEEC04	8
	Slave address register_3	SAR_3	8	H'FFFEEC05	8
	I²C bus transmit data register_3	ICDRT_3	8	H'FFFEEC06	8
	I²C bus receive data register_3	ICDRR_3	8	H'FFFEEC07	8
	NF2CYC register_3	NF2CYC_3	8	H'FFFEEC08	8
SSIF	Control register_0	SSICR_0	32	H'FFFEB000	8, 16, 32
	Status register_0	SSISR_0	32	H'FFFEB004	8, 16, 32
	FIFO control register_0	SSIFCR_0	32	H'FFFEB010	8, 16, 32
	FIFO status register_0	SSIFSR_0	32	H'FFFEB014	8, 16, 32
	FIFO data register_0	SSIFDR_0	32	H'FFFEB018	32
	Control register_1	SSICR_1	32	H'FFFEB400	8, 16, 32
	Status register_1	SSISR_1	32	H'FFFEB404	8, 16, 32
	FIFO control register_1	SSIFCR_1	32	H'FFFEB410	8, 16, 32
	FIFO status register_1	SSIFSR_1	32	H'FFFEB414	8, 16, 32
	FIFO data register_1	SSIFDR_1	32	H'FFFEB418	32
	Control register_2	SSICR_2	32	H'FFFEB800	8, 16, 32
	Status register_2	SSISR_2	32	H'FFFEB804	8, 16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
SSIF	FIFO control register_2	SSIFCR_2	32	H'FFFEB810	8, 16, 32
	FIFO status register_2	SSIFSR_2	32	H'FFFEB814	8, 16, 32
	FIFO data register_2	SSIFDR_2	32	H'FFFEB818	32
	Control register_3	SSICR_3	32	H'FFFEBC00	8, 16, 32
	Status register_3	SSISR_3	32	H'FFFEBC04	8, 16, 32
	FIFO control register_3	SSIFCR_3	32	H'FFFEBC10	8, 16, 32
	FIFO status register_3	SSIFSR_3	32	H'FFFEBC14	8, 16, 32
	FIFO data register_3	SSIFDR_3	32	H'FFFEBC18	32
	Control register_4	SSICR_4	32	H'FFFEC000	8, 16, 32
	Status register_4	SSISR_4	32	H'FFFEC004	8, 16, 32
	FIFO control register_4	SSIFCR_4	32	H'FFFEC010	8, 16, 32
	FIFO status register_4	SSIFSR_4	32	H'FFFEC014	8, 16, 32
	FIFO data register_4	SSIFDR_4	32	H'FFFEC018	32
	Control register_5	SSICR_5	32	H'FFFEC400	8, 16, 32
	Status register_5	SSISR_5	32	H'FFFEC404	8, 16, 32
	FIFO control register_5	SSIFCR_5	32	H'FFFEC410	8, 16, 32
	FIFO status register_5	SSIFSR_5	32	H'FFFEC414	8, 16, 32
	FIFO data register_5	SSIFDR_5	32	H'FFFEC418	32
RCAN-TL1	Master Control Register_0	MCR_0	16	H'FFFE5000	16
	General Status Register_0	GSR_0	16	H'FFFE5002	16
	Bit Configuration Register 1_0	BCR1_0	16	H'FFFE5004	16
	Bit Configuration Register 0_0	BCR0_0	16	H'FFFE5006	16
	Interrupt Request Register_0	IRR_0	16	H'FFFE5008	16
	Interrupt Mask Register_0	IMR_0	16	H'FFFE500A	16
	Error Counter Register_0	TEC_REC_0	16	H'FFFE500C	8, 16
	Transmit Pending Register 1_0	TXPR1_0	16	H'FFFE5020	32
	Transmit Pending Register 0_0	TXPR0_0	16	H'FFFE5022	16
	Transmit Cancel Register 1_0	TXCR1_0	16	H'FFFE5028	16
	Transmit Cancel Register 0_0	TXCR0_0	16	H'FFFE502A	16
	Transmit Acknowledge Register 1_0	TXACK1_0	16	H'FFFE5030	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1	Transmit Acknowledge Register 0_0	TXACK0_0	16	H'FFFE5032	16
	Abort Acknowledge Register 1_0	ABACK1_0	16	H'FFFE5038	16
	Abort Acknowledge Register 0_0	ABACK0_0	16	H'FFFE503A	16
	Data Frame Receive Pending Register 1_0	RXPR1_0	16	H'FFFE5040	16
	Data Frame Receive Pending Register 0_0	RXPR0_0	16	H'FFFE5042	16
	Remote Frame Receive Pending Register 1_0	RFPR1_0	16	H'FFFE5048	16
	Remote Frame Receive Pending Register 0_0	RFPR0_0	16	H'FFFE504A	16
	Mailbox Interrupt Mask Register 1_0	MBIMR1_0	16	H'FFFE5050	16
	Mailbox Interrupt Mask Register 0_0	MBIMR0_0	16	H'FFFE5052	16
	Unread Message Status Register 1_0	UMSR1_0	16	H'FFFE5058	16
	Unread Message Status Register 0_0	UMSR0_0	16	H'FFFE505A	16
	Timer Trigger Control Register 0_0	TTCR0_0	16	H'FFFE5080	16
	Cycle Maximum/Tx-Enable Window Register_0	CMAX_TEW_0	16	H'FFFE5084	16
	Reference Trigger Offset Register_0	RFTROFF_0	16	H'FFFE5086	16
	Timer Status Register_0	TSR_0	16	H'FFFE5088	16
	Cycle Counter Register_0	CCR_0	16	H'FFFE508A	16
	Timer Counter Register_0	TCNTR_0	16	H'FFFE508C	16
	Cycle Time Register_0	CYCTR_0	16	H'FFFE5090	16
	Reference Mark Register_0	RFMK_0	16	H'FFFE5094	16
	Timer Compare Match Register 0_0	TCMR0_0	16	H'FFFE5098	16
	Timer Compare Match Register 1_0	TCMR1_0	16	H'FFFE509C	16
	Timer Compare Match Register 2_0	TCMR2_0	16	H'FFFE50A0	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1	Tx-Trigger Time Selection Register_0	TTTSEL_0	16	H'FFFE50A4	16
	Mailbox n Control 0_H_0 (n = 0 to 31)	MBn_CONTROL0_H_0 (n = 0 to 31)	16	H'FFFE5100 + n × 32	16, 32
	Mailbox n Control 0_L_0 (n = 0 to 31)	MBn_CONTROL0_L_0 (n = 0 to 31)	16	H'FFFE5102 + n × 32	16
	Mailbox n Local Acceptance Filter Mask 0_0 (n = 0 to 31)	MBn_LAFM0_0 (n = 0 to 31)	16	H'FFFE5104 + n × 32	16, 32
	Mailbox n Local Acceptance Filter Mask 1_0 (n = 0 to 31)	MBn_LAFM1_0 (n = 0 to 31)	16	H'FFFE5106 + n × 32	16
	Mailbox n Data 01_0 (n = 0 to 31)	MBn_DATA_01_0 (n = 0 to 31)	16	H'FFFE5108 + n × 32	8, 16, 32
	Mailbox n Data 23_0 (n = 0 to 31)	MBn_DATA_23_0 (n = 0 to 31)	16	H'FFFE510A + n × 32	8, 16
	Mailbox n Data 45_0 (n = 0 to 31)	MBn_DATA_45_0 (n = 0 to 31)	16	H'FFFE510C + n × 32	8, 16, 32
	Mailbox n Data 67_0 (n = 0 to 31)	MBn_DATA_67_0 (n = 0 to 31)	16	H'FFFE510E + n × 32	8, 16
	Mailbox n Control 1_0 (n = 0 to 31)	MBn_CONTROL1_0 (n = 0 to 31)	16	H'FFFE5110 + n × 32	8, 16
	Mailbox n Time Stamp_0 (n = 0 to 15, 30, 31)	MBn_TIMESTAMP_0 (n = 0 to 15, 30, 31)	16	H'FFFE5112 + n × 32	16
	Mailbox n Trigger Time_0 (n = 24 to 30)	MBn_TTT_0 (n = 24 to 30)	16	H'FFFE5114 + n × 32	16
	Mailbox n TT Control_0 (n = 24 to 29)	MBn_TTCONTROL_0 (n = 24 to 29)	16	H'FFFE5116 + n × 32	16
	Master Control Register_1	MCR_1	16	H'FFFE5800	16
	General Status Register_1	GSR_1	16	H'FFFE5802	16
	Bit Configuration Register 1_1	BCR1_1	16	H'FFFE5804	16
	Bit Configuration Register 0_1	BCR0_1	16	H'FFFE5806	16
	Interrupt Register_1	IRR_1	16	H'FFFE5808	16
	Interrupt Mask Register_1	IMR_1	16	H'FFFE580A	16
	Error Counter Register_1	TEC_REC_1	16	H'FFFE580C	8, 16
	Transmit Pending Register 1_1	TXPR1_1	16	H'FFFE5820	32
	Transmit Pending Register 0_1	TXPR0_1	16	H'FFFE5822	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1	Transmit Cancel Register 1_1	TXCR1_1	16	H'FFFE5828	16
	Transmit Cancel Register 0_1	TXCR0_1	16	H'FFFE582A	16
	Transmit Acknowledge Register 1_1	TXACK1_1	16	H'FFFE5830	16
	Transmit Acknowledge Register 0_1	TXACK0_1	16	H'FFFE5832	16
	Abort Acknowledge Register 1_1	ABACK1_1	16	H'FFFE5838	16
	Abort Acknowledge Register 0_1	ABACK0_1	16	H'FFFE583A	16
	Data Frame Receive Pending Register 1_1	RXPR1_1	16	H'FFFE5840	16
	Data Frame Receive Pending Register 0_1	RXPR0_1	16	H'FFFE5842	16
	Remote Frame Receive Pending Register 1_1	RFPR1_1	16	H'FFFE5848	16
	Remote Frame Receive Pending Register 0_1	RFPR0_1	16	H'FFFE584A	16
	Mailbox Interrupt Mask Register 1_1	MBIMR1_1	16	H'FFFE5850	16
	Mailbox Interrupt Mask Register 0_1	MBIMR0_1	16	H'FFFE5852	16
	Unread Message Status Register 1_1	UMSR1_1	16	H'FFFE5858	16
	Unread Message Status Register 0_1	UMSR0_1	16	H'FFFE585A	16
	Timer Trigger Control Register 0_1	TTCR0_1	16	H'FFFE5880	16
	Cycle Maximum/Tx-Enable Window Register_1	CMAX_TEW_1	16	H'FFFE5884	16
	Reference Trigger Offset Register_1	RFTROFF_1	16	H'FFFE5886	16
	Timer Status Register_1	TSR_1	16	H'FFFE5888	16
	Cycle Counter Register_1	CCR_1	16	H'FFFE588A	16
	Timer Counter Register_1	TCNTR_1	16	H'FFFE588C	16
	Cycle Time Register_1	CYCTR_1	16	H'FFFE5890	16
	Reference Mark Register_1	RFMK_1	16	H'FFFE5894	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1	Timer Compare Match Register 0_1	TCMR0_1	16	H'FFFE5898	16
	Timer Compare Match Register 1_1	TCMR1_1	16	H'FFFE589C	16
	Timer Compare Match Register 2_1	TCMR2_1	16	H'FFFE58A0	16
	Tx-Trigger Time Selection Register_1	TTTSEL_1	16	H'FFFE58A4	16
	Mailbox n Control 0_H_1 (n = 0 to 31)	MBn_CONTROL0_H_1 (n = 0 to 31)	16	H'FFFE5900 + n × 32	16, 32
	Mailbox n Control 0_L_1 (n = 0 to 31)	MBn_CONTROL0_L_1 (n = 0 to 31)	16	H'FFFE5902 + n × 32	16
	Mailbox n Local Acceptance Filter Mask 0_1 (n = 0 to 31)	MBn_LAFM0_1 (n = 0 to 31)	16	H'FFFE5904 + n × 32	16, 32
	Mailbox n Local Acceptance Filter Mask 1_1 (n = 0 to 31)	MBn_LAFM1_1 (n = 0 to 31)	16	H'FFFE5906 + n × 32	16
	Mailbox n Data 01_1 (n = 0 to 31)	MBn_DATA_01_1 (n = 0 to 31)	16	H'FFFE5908 + n × 32	8, 16, 32
	Mailbox n Data 23_1 (n = 0 to 31)	MBn_DATA_23_1 (n = 0 to 31)	16	H'FFFE590A + n × 32	8, 16
	Mailbox n Data 45_1 (n = 0 to 31)	MBn_DATA_45_1 (n = 0 to 31)	16	H'FFFE590C + n × 32	8, 16, 32
	Mailbox n Data 67_1 (n = 0 to 31)	MBn_DATA_67_1 (n = 0 to 31)	16	H'FFFE590E + n × 32	8, 16
	Mailbox n Control 1_1 (n = 0 to 31)	MBn_CONTROL1_1 (n = 0 to 31)	16	H'FFFE5910 + n × 32	8, 16
	Mailbox n Time Stamp_1 (n = 0 to 15, 30, 31)	MBn_TIMESTAMP_1 (n = 0 to 15, 30, 31)	16	H'FFFE5912 + n × 32	16
	Mailbox n Trigger Time_1 (n = 24 to 30)	MBn_TTT_1 (n = 24 to 30)	16	H'FFFE5914 + n × 32	16
	Mailbox n TT Control_1 (n = 24 to 29)	MBn_TTCONTROL_1 (n = 24 to 29)	16	H'FFFE5916 + n × 32	16
ADC	A/D data register A	ADDRA	16	H'FFFE4800	16
	A/D data register B	ADDRB	16	H'FFFE4802	16
	A/D data register C	ADDRC	16	H'FFFE4804	16
	A/D data register D	ADDRD	16	H'FFFE4806	16
	A/D data register E	ADDRE	16	H'FFFE4808	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
ADC	A/D data register F	ADDRF	16	H'FFFE480A	16
	A/D data register G	ADDRG	16	H'FFFE480C	16
	A/D data register H	ADDRH	16	H'FFFE480E	16
	A/D control/status register	ADCSR	16	H'FFFE4820	16
DAC	D/A data register 0	DADR0	8	H'FFFE4C00	8, 16
	D/A data register 1	DADR1	8	H'FFFE4C0 1	8, 16
	D/A control register	DACR	8	H'FFFE4C0 2	8, 16
FLCTL	Common control register	FLCMNCR	32	H'FFFEC800	32
	Command control register	FLCMDCR	32	H'FFFEC804	32
	Command code register	FLCMCDR	32	H'FFFEC808	32
	Address register	FLADR	32	H'FFFEC80C	32
	Address register 2	FLADR2	32	H'FFFEC83C	32
	Data register	FLDATAR	32	H'FFFEC810	32
	Data counter register	FLDTCNTR	32	H'FFFEC814	32
	Interrupt DMA control register	FLINTDMACR	32	H'FFFEC818	32
	Ready busy timeout setting register	FLBSYTMR	32	H'FFFEC81C	32
	Ready busy timeout counter	FLBSYCNT	32	H'FFFEC820	32
	Data FIFO register	FLDTFIFO	32	H'FFFEC850	32
	Control code FIFO register	FLECFIFO	32	H'FFFEC860	32
	Transfer control register	FLTRCR	8	H'FFFEC82C	8
	4-symbol ECC processing result register 1	FL4ECCRES1	32	H'FFFEC880	32
	4-symbol ECC processing result register 2	FL4ECCRES2	32	H'FFFEC884	32
	4-symbol ECC processing result register 3	FL4ECCRES3	32	H'FFFEC888	32
	4-symbol ECC processing result register 4	FL4ECCRES4	32	H'FFFEC88C	32
	4-symbol ECC control register	FL4ECCCR	32	H'FFFEC890	32
	4-symbol ECC error count register	FL4ECCCNT	32	H'FFFEC894	32
USB	Port0 system configuration control register	SYSCFG0	16	H'FFFF0000	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB	Port1 system configuration control register	SYSCFG1	16	H'FFFF0002	16
	Port0 system configuration status register	SYSSTS0	16	H'FFFF0004	16
	Port1 system configuration status register	SYSSTS1	16	H'FFFF0006	16
	Port0 device state control register	DVSTCTR0	16	H'FFFF0008	16
	Port1 device state control register	DVSTCTR1	16	H'FFFF000A	16
	Test mode register	TESTMODE	16	H'FFFF000C	16
	DMA0 pin configuration register	D0FBCFG	16	H'FFFF0010	16
	DMA1 pin configuration register	D1FBCFG	16	H'FFFF0012	16
	CFIFO port register	CFIFO	32	H'FFFF0014	8, 16, 32
	D0FIFO port register	D0FIFO	32	H'FFFF0018	8, 16, 32
	D1FIFO port register	D1FIFO	32	H'FFFF001C	8, 16, 32
	CFIFO port select register	CFIFOSEL	16	H'FFFF0020	16
	CFIFO port control register	CFIFOCTR	16	H'FFFF0022	16
	D0FIFO port select register	D0FIFOSEL	16	H'FFFF0028	16
	D0FIFO port control register	D0FIFOCTR	16	H'FFFF002A	16
	D1FIFO port select register	D1FIFOSEL	16	H'FFFF002C	16
	D1FIFO port control register	D1FIFOCTR	16	H'FFFF002E	16
	Interrupt enable register 0	INTENB0	16	H'FFFF0030	16
	Interrupt enable register 1	INTENB1	16	H'FFFF0032	16
	Interrupt enable register 2	INTENB2	16	H'FFFF0034	16
	BRDY interrupt enable register	BRDYENB	16	H'FFFF0036	16
	NRDY interrupt enable register	NRDYENB	16	H'FFFF0038	16
	BEMP interrupt enable register	BEMPENB	16	H'FFFF003A	16
	SOF output configuration register	SOFCFG	16	H'FFFF003C	16
	Interrupt status register 0	INTSTS0	16	H'FFFF0040	16
	Interrupt status register 1	INTSTS1	16	H'FFFF0042	16
	Interrupt status register 2	INTSTS2	16	H'FFFF0044	16
	BRDY interrupt status register	BRDYSTS	16	H'FFFF0046	16
	NRDY interrupt status register	NRDYSTS	16	H'FFFF0048	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB	BEMP interrupt status register	BEMPSTS	16	H'FFFF004A	16
	Frame number register	FRMNUM	16	H'FFFF004C	16
	μFrame number register	UFRMNUM	16	H'FFFF004E	16
	USB address register	USBADDR	16	H'FFFF0050	16
	USB request type register	USBREQ	16	H'FFFF0054	16
	USB request value register	USBVAL	16	H'FFFF0056	16
	USB request index register	USBINDX	16	H'FFFF0058	16
	USB request length register	USBLENG	16	H'FFFF005A	16
	DCP configuration register	DCPCFG	16	H'FFFF005C	16
	DCP maximum packet size register	DCPMAXP	16	H'FFFF005E	16
	DCP control register	DCPCTR	16	H'FFFF0060	16
	Pipe window select register	PIPESEL	16	H'FFFF0064	16
	Pipe configuration register	PIPECFG	16	H'FFFF0068	16
	Pipe buffer setting register	PIPEBUF	16	H'FFFF006A	16
	Pipe maximum packet size register	PIPEMAXP	16	H'FFFF006C	16
	Pipe cycle control register	PIPEPERI	16	H'FFFF006E	16
	Pipe 1 control register	PIPE1CTR	16	H'FFFF0070	16
	Pipe 2 control register	PIPE2CTR	16	H'FFFF0072	16
	Pipe 3 control register	PIPE3CTR	16	H'FFFF0074	16
	Pipe 4 control register	PIPE4CTR	16	H'FFFF0076	16
	Pipe 5 control register	PIPE5CTR	16	H'FFFF0078	16
	Pipe 6 control register	PIPE6CTR	16	H'FFFF007A	16
	Pipe 7 control register	PIPE7CTR	16	H'FFFF007C	16
	Pipe 8 control register	PIPE8CTR	16	H'FFFF007E	16
	Pipe 9 control register	PIPE9CTR	16	H'FFFF0080	16
	Pipe 1 transaction counter enable register	PIPE1TRE	16	H'FFFF0090	16
	Pipe 1 transaction counter register	PIPE1TRN	16	H'FFFF0092	16
	Pipe 2 transaction counter enable register	PIPE2TRE	16	H'FFFF0094	16
	Pipe 2 transaction counter register	PIPE2TRN	16	H'FFFF0096	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB	Pipe 3 transaction counter enable register	PIPE3TRE	16	H'FFFF0098	16
	Pipe 3 transaction counter register	PIPE3TRN	16	H'FFFF009A	16
	Pipe 4 transaction counter enable register	PIPE4TRE	16	H'FFFF009C	16
	Pipe 4 transaction counter register	PIPE4TRN	16	H'FFFF009E	16
	Pipe 5 transaction counter enable register	PIPE5TRE	16	H'FFFF00A0	16
	Pipe 5 transaction counter register	PIPE5TRN	16	H'FFFF00A2	16
	USB AC characteristics switching register 0	USBACSWR0	16	H'FFFF00C0	16
	USB AC characteristics switching register 1	USBACSWR1	16	H'FFFF00C2	16
	Device address 0 configuration register	DEVADD0	16	H'FFFF00D0	16
	Device address 1 configuration register	DEVADD1	16	H'FFFF00D2	16
	Device address 2 configuration register	DEVADD2	16	H'FFFF00D4	16
	Device address 3 configuration register	DEVADD3	16	H'FFFF00D6	16
	Device address 4 configuration register	DEVADD4	16	H'FFFF00D8	16
	Device address 5 configuration register	DEVADD5	16	H'FFFF00DA	16
	Device address 6 configuration register	DEVADD6	16	H'FFFF00DC	16
	Device address 7 configuration register	DEVADD7	16	H'FFFF00DE	16
	Device address 8 configuration register	DEVADD8	16	H'FFFF00E0	16
	Device address 9 configuration register	DEVADD9	16	H'FFFF00E2	16
	Device address A configuration register	DEVADDA	16	H'FFFF00E4	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATAPI	ATAPI control register	ATAPI_CONTROL	32	H'FFFECC80	32
	ATAPI status register	ATAPI_STATUS	32	H'FFFECC84	32
	Interrupt enable register	ATAPI_INT_ENABLE	32	H'FFFECC88	32
	PIO timing register	ATAPI_PIO_TIMING	32	H'FFFECC8C	32
	Multiword DMA timing register	ATAPI_MULTI_TIMING	32	H'FFFECC90	32
	Ultra DMA timing register	ATAPI_ULTRA_ TIMING	32	H'FFFECC94	32
	DMA start address register	ATAPI_DMA_START_ ADR	32	H'FFFECC9C	32
	DMA transfer count register	ATAPI_DMA_TRANS_ CNT	32	H'FFFECCA0	32
	ATAPI control 2 register	ATAPI_CONTROL2	32	H'FFFECCA4	32
	ATAPI signal status register	ATAPI_SIG_ST	32	H'FFFECCB0	32
	Byte swap register	ATAPI_BYTE_SWAP	32	H'FFFECCBC	32
2DG	Blit function setting register for graphics	GR_BLTPLY	32	H'E8000000	16, 32
	Mixing function setting register for graphics (synchronized with VSYNC)	GR_MIXPLY	32	H'E8000004	16, 32
	Operation status register for graphics	GR_DOSTAT	32	H'E8000008	16, 32
	Interrupt status register for graphics	GR_IRSTAT	32	H'E800000C	16, 32
	Interrupt mask control register for graphics	GR_INTMSK	32	H'E8000010	16, 32
	Interrupt reset control register for graphics	GR_INTDIS	32	H'E8000014	16, 32
	DMAC-request control register for graphics	GR_DMAC	32	H'E8000020	16, 32
	Source A&B read-in-area setting register for blitter	GR_SABSET	32	H'E8000030	16, 32
	Destination C write area setting register for blitter	GR_DCSET	32	H'E8000038	16, 32
	Source E read in-area setting register for output block (synchronized with VSYNC)	MGR_SESET	32	H'E8000040	16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
2DG	Pixel format setting register for graphics (only one bit, SF_FMT, is synchronized with VSYNC)	GR_PIXLFMT	32	H'E8000048	16, 32
	Operation mode setting register for blitter	GR_BLTMODE	32	H'E8000050	16, 32
	Resize display setting register for graphics	GR_RISZSET	32	H'E8000060	16, 32
	Resize mode select register for blitter	GR_RISZMOD	32	H'E8000064	16, 32
	Resize delta setting register for blitter	GR_DELT	32	H'E8000068	16, 32
	Resize horizontal starting phase register for blitter	GR_HSPHAS	32	H'E800006C	16, 32
	Resize vertical starting phase register for blitter	GR_VSPHAS	32	H'E8000070	16, 32
	Resize horizontal delta setting register for output block (synchronized with VSYNC)	MGR_HDELT	32	H'E8000074	16, 32
	Resize horizontal starting phase register for output block (synchronized with VSYNC)	MGR_HPHAS	32	H'E8000078	16, 32
	Logical operation input data register for blitter	GR_LGDAT	32	H'E8000080	16, 32
	Chromakey target color data register for blitter	GR_DETCOL	32	H'E8000084	16, 32
	Replacement color data register for blitter blending	GR_BRDCOL	32	H'E8000088	16, 32
	Blend 1 control register for blitter	GR_BRD1CNT	32	H'E800008C	16, 32
	Mixing mode setting register for output block (synchronized with VSYNC)	MGR_MIXMODE	32	H'E8000098	16, 32
	Panel-output horizontal timing setting register for output block (synchronized with VSYNC)	MGR_MIXHTMG	32	H'E80000A0	16, 32
	Panel-output mixing horizontal valid area setting register for output block (synchronized with VSYNC)	MGR_MIXHS	32	H'E80000A4	16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
2DG	Panel-output vertical timing setting register for output block (synchronized with VSYNC)	MGR_MIXVTMG	32	H'E80000A8	16, 32
	Panel-output mixing vertical valid area setting register for output block (synchronized with VSYNC)	MGR_MIXVS	32	H'E80000AC	16, 32
	Output SYNC setting register for graphics	GR_VSDLY	32	H'E80000C4	16, 32
	Video DAC timing setting register	VDAC_TMC	32	H'EA000000	32
PFC	Port A I/O register L	PAIORL	16	H'FFFE3802	8, 16
	Port A control register L4	PACRL4	16	H'FFFE380C	8, 16, 32
	Port A control register L3	PACRL3	16	H'FFFE380E	8, 16
	Port A control register L2	PACRL2	16	H'FFFE3810	8, 16, 32
	Port A control register L1	PACRL1	16	H'FFFE3812	8, 16
	Port B I/O register H	PBIORH	16	H'FFFE3820	8, 16, 32
	Port B I/O register L	PBIORL	16	H'FFFE3822	8, 16
	Port B control register H2	PBCRH2	16	H'FFFE3828	8, 16, 32
	Port B control register H1	PBCRH1	16	H'FFFE382A	8, 16
	Port B control register L4	PBCRL4	16	H'FFFE382C	8, 16, 32
	Port B control register L3	PBCRL3	16	H'FFFE382E	8, 16
	Port B control register L2	PBCRL2	16	H'FFFE3830	8, 16, 32
	Port B control register L1	PBCRL1	16	H'FFFE3832	8, 16
	Port C I/O register L	PCIORL	16	H'FFFE3842	8, 16
	Port C control register L3	PCCRL3	16	H'FFFE384E	8, 16
	Port C control register L2	PCCRL2	16	H'FFFE3850	8, 16, 32
	Port C control register L1	PCCRL1	16	H'FFFE3852	8, 16
	Port D I/O register L	PDIORL	16	H'FFFE3862	8, 16
	Port D control register L1	PDCRL1	16	H'FFFE3872	8, 16
	Port E I/O register L	PEIORL	16	H'FFFE3882	8, 16
	Port E control register L4	PECRL4	16	H'FFFE388C	8, 16, 32
	Port E control register L3	PECRL3	16	H'FFFE388E	8, 16
	Port E control register L2	PECRL2	16	H'FFFE3890	8, 16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
PFC	Port E control register L1	PECRL1	16	H'FFFE3892	8, 16
	Port F I/O register L	PFIORL	16	H'FFFE38A2	8, 16
	Port F control register L2	PFCRL2	16	H'FFFE38B0	8, 16, 32
	Port F control register L1	PFCRL1	16	H'FFFE38B2	8, 16
	Port G control register L2	PGCRL2	16	H'FFFE38D0	8, 16, 32
	Port G control register L1	PGCRL1	16	H'FFFE38D2	8, 16
	Port H I/O register L	PHIORL	16	H'FFFE38E2	8, 16
	Port H control register L4	PHCRL4	16	H'FFFE38EC	8, 16, 32
	Port H control register L3	PHCRL3	16	H'FFFE38EE	8, 16
	Port H control register L2	PHCRL2	16	H'FFFE38F0	8, 16, 32
	Port H control register L1	PHCRL1	16	H'FFFE38F2	8, 16
	Port J I/O register L	PJIORL	16	H'FFFE3902	8, 16
	Port J control register L4	PJCRL4	16	H'FFFE390C	8, 16, 32
	Port J control register L3	PJCRL3	16	H'FFFE390E	8, 16
	Port J control register L2	PJCRL2	16	H'FFFE3910	8, 16, 32
	Port J control register L1	PJCRL1	16	H'FFFE3912	8, 16
	Port K I/O register L	PKIORL	16	H'FFFE3922	8, 16
	Port K control register L1	PKCRL1	16	H'FFFE3932	8, 16
I/O Port	Port A data register L	PADRL	16	H'FFFE3816	8, 16
	Port A port register L	PAPRL	16	H'FFFE381A	8, 16
	Port B data register H	PBDRH	16	H'FFFE3834	8, 16, 32
	Port B data register L	PBDRL	16	H'FFFE3836	8, 16
	Port B port register H	PBPRH	16	H'FFFE3838	8, 16, 32
	Port B port register L	PBPRL	16	H'FFFE383A	8, 16
	Port C data register L	PCDRL	16	H'FFFE3856	8, 16
	Port C port register L	PCPRL	16	H'FFFE385A	8, 16
	Port D data register L	PDDRL	16	H'FFFE3876	8, 16
	Port D port register L	PDPRL	16	H'FFFE387A	8, 16
	Port E data register L	PEDRL	16	H'FFFE3896	8, 16
	Port E port register L	PEPRL	16	H'FFFE389A	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
I/O Port	Port F data register L	PFDRL	16	H'FFFE38B6	8, 16
	Port F port register L	PFPRL	16	H'FFFE38BA	8, 16
	Port G data register L	PGDRL	16	H'FFFE38D6	8, 16
	Port H data register L	PHDRL	16	H'FFFE38F6	8, 16
	Port H port register L	PHPRL	16	H'FFFE38FA	8, 16
	Port J data register L	PJDRL	16	H'FFFE3916	8, 16
	Port J port register L	PJPRL	16	H'FFFE391A	8, 16
	Port K data register L	PKDRL	16	H'FFFE3936	8, 16
	Port K port register L	PKPRL	16	H'FFFE393A	8, 16
Power-Down Modes	Standby control register 1	STBCR1	8	H'FFFE0014	8
	Standby control register 2	STBCR2	8	H'FFFE0018	8
	Standby control register 3	STBCR3	8	H'FFFE0400	8
	Standby control register 4	STBCR4	8	H'FFFE0402	8
	Standby control register 5	STBCR5	8	H'FFFE0404	8
	Standby control register 6	STBCR6	8	H'FFFE0406	8
	Standby control register 7	STBCR7	8	H'FFFE0408	8
	System control register 1	SYSCR1	8	H'FFFE0480	8
	System control register 2	SYSCR2	8	H'FFFE0482	8
	System control register 3	SYSCR3	8	H'FFFE0484	8
	System control register 4	SYSCR4	8	H'FFFE0486	8
	System control register 5	SYSCR5	8	H'FFFE0488	8
	System control register 6	SYSCR6	8	H'FFFE048A	8
	System control register 7	SYSCR7	8	H'FFFE04A0	8
	System control register 8	SYSCR8	8	H'FFFE04A2	8
	System control register 9	SYSCR9	8	H'FFFE04A4	8
	System control register 10	SYSCR10	8	H'FFFE04A6	8
	System control register 11	SYSCR11	8	H'FFFE04A8	8
	System control register 12	SYSCR12	8	H'FFFE04AA	8
	Software reset control register	SWRSTCR	8	H'FFFE0440	8
	High-impedance control register	HIZCR	8	H'FFFE0442	8
	CPU0 mode status register	C0MSR	8	H'FFFE0040	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Power-Down Modes	CPU1 mode status register	C1MSR	8	H'FFFE0042	8
	Data retention on-chip RAM area specification register	RRAMKP	8	H'FFFE0C00	8
	Deep standby control register	DSCTR	8	H'FFFE0C02	8
	Deep standby cancel source select register	DSSSR	16	H'FFFE0C04	16
	Deep standby cancel source flag register	DSFR	16	H'FFFE0C08	16
H-UDI	Instruction register	SDIR	16	H'FFFD9000	16

32.2 Register Bits

Module Name	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
Multi-core processor	CPUIDR	—	ID	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	SEMR0	—	—	—	—	—	—	—	SEMF
	SEMR1	—	—	—	—	—	—	—	SEMF
	SEMR2	—	—	—	—	—	—	—	SEMF
	SEMR3	—	—	—	—	—	—	—	SEMF
	SEMR4	—	—	—	—	—	—	—	SEMF
	SEMR5	—	—	—	—	—	—	—	SEMF
	SEMR6	—	—	—	—	—	—	—	SEMF
	SEMR7	—	—	—	—	—	—	—	SEMF
	SEMR8	—	—	—	—	—	—	—	SEMF
	SEMR9	—	—	—	—	—	—	—	SEMF
	SEMR10	—	—	—	—	—	—	—	SEMF
	SEMR11	—	—	—	—	—	—	—	SEMF
	SEMR12	—	—	—	—	—	—	—	SEMF
	SEMR13	—	—	—	—	—	—	—	SEMF
	SEMR14	—	—	—	—	—	—	—	SEMF
	SEMR15	—	—	—	—	—	—	—	SEMF
	SEMR16	—	—	—	—	—	—	—	SEMF
	SEMR17	—	—	—	—	—	—	—	SEMF
	SEMR18	—	—	—	—	—	—	—	SEMF
	SEMR19	—	—	—	—	—	—	—	SEMF
	SEMR20	—	—	—	—	—	—	—	SEMF
	SEMR21	—	—	—	—	—	—	—	SEMF
	SEMR22	—	—	—	—	—	—	—	SEMF
	SEMR23	—	—	—	—	—	—	—	SEMF
	SEMR24	—	—	—	—	—	—	—	SEMF

Module Name	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
Multi-core processor	SEMR25	—	—	—	—	—	—	—	SEMF
	SEMR26	—	—	—	—	—	—	—	SEMF
	SEMR27	—	—	—	—	—	—	—	SEMF
	SEMR28	—	—	—	—	—	—	—	SEMF
	SEMR29	—	—	—	—	—	—	—	SEMF
	SEMR30	—	—	—	—	—	—	—	SEMF
	SEMR31	—	—	—	—	—	—	—	SEMF
CPG	FRQCR0	—	CKOEN2	CKOEN[1:0]		—	—	STC[1:0]	
		—	—	IFC[1:0]		—	PFC[2:0]		
	FRQCR1	—	—	—	—	—	—	—	—
		—	—	IFC[1:0]		—	—	—	—
INTC	C0ICR0	NMIL	—	—	—	—	—	—	NMIS
		—	—	—	—	—	—	—	—
	C0ICR1	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S
		IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
	C0ICR2	—	—	—	—	—	—	—	—
		PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S
	C0IRQRR	—	—	—	—	—	—	—	—
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
	C0PINTER	—	—	—	—	—	—	—	—
		PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E
	C0PIRR	—	—	—	—	—	—	—	—
		PINT7R	PINT6R	PINT5R	PINT4R	PINT3R	PINT2R	PINT1R	PINT0R
	C0IBCR	E15	E14	E13	E12	E11	E10	E9	E8
		E7	E6	E5	E4	E3	E2	E1	—
	C0IBNR	BE[1:0]		BOVE	—	—	—	—	—
		—	—	—	—	BN[3:0]			
	C0IPR01								
	C0IPR02								

Module Name	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
INTC	C0IPR05								
	C0INTER	NMIE	UDIE	SLPEE	—	—	—	—	—
		—	—	—	—	—	—	—	—
	C0IRQER	—	—	—	—	—	—	—	—
		IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
	C1ICR0	NMIL	—	—	—	—	—	—	NMIS
		—	—	—	—	—	—	—	—
	C1ICR1	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S
		IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
	C1ICR2	—	—	—	—	—	—	—	—
		PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S
	C1IRQRR	—	—	—	—	—	—	—	—
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
	C1PINTER	—	—	—	—	—	—	—	—
		PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E
	C1PIRR	—	—	—	—	—	—	—	—
		PINT7R	PINT6R	PINT5R	PINT4R	PINT3R	PINT2R	PINT1R	PINT0R
	C1IBCR	E15	E14	E13	E12	E11	E10	E9	E8
		E7	E6	E5	E4	E3	E2	E1	—
	C1IBNR	BE[1:0]		BOVE	—	—	—	—	—
		—	—	—	—	BN[3:0]			
	C1IPR01								
	C1IPR02								
	C1IPR05								
	C1INTER	NMIE	UDIE	SLPEE	—	—	—	—	—
		—	—	—	—	—	—	—	—

Module Name	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
INTC	C1IRQER	—	—	—	—	—	—	—	—
		IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
	C0IPCR15	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C0IPCR14	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C0IPCR13	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C0IPCR12	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C0IPCR11	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C0IPCR10	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C0IPCR09	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C0IPCR08	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C0IPER	CIPE15	CIPE14	CIPE13	CIPE12	CIPE11	CIPE10	CIPE9	CIPE8
		—	—	—	—	—	—	—	—
	C1IPCR15	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C1IPCR14	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C1IPCR13	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C1IPCR12	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C1IPCR11	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI

Module Name	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
INTC	C1IPCR10	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C1IPCR09	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C1IPCR08	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CI
	C1IPER	CIPE15	CIPE14	CIPE13	CIPE12	CIPE11	CIPE10	CIPE9	CIPE8
		—	—	—	—	—	—	—	—
	C0IPR06								
	C0IPR07								
	C0IPR08								
	C0IPR09								
	C0IPR10								
	C0IPR11								
	C0IPR12								
	C0IPR13								
	C0IPR14								
	C0IPR15								
	C0IPR16								

Module Name	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
INTC	C0IPR17								
	C0IPR18								
	C0IPR19								
	C0IPR20								
	C0IPR21								
	C1IPR06								
	C1IPR07								
	C1IPR08								
	C1IPR09								
	C1IPR10								
	C1IPR11								
	C1IPR12								
	C1IPR13								
	C1IPR14								
	C1IPR15								

Module Name	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
INTC	C1IPR16								
	C1IPR17								
	C1IPR18								
	C1IPR19								
	C1IPR20								
	C1IPR21								
	IDCNT6	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT7	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT8	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT9	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT10	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT11	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT12	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT13	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT14	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—

Module Name	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
INTC	IDCNT15	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT16	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT17	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT18	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT19	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT20	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT21	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT22	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT23	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT24	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT25	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT26	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT27	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT28	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT29	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—

Module Name	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
INTC	IDCNT30	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT31	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT32	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT33	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT34	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT35	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT36	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT37	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT38	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT39	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT40	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT41	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT42	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT43	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT44	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—

Module Name	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
INTC	IDCNT45	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT46	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT47	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT48	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT49	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT50	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT51	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT52	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT53	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT54	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT55	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT56	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT57	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT58	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT59	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—

Module Name	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
INTC	IDCNT60	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT61	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT62	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT63	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT64	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT66	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT67	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT68	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT69	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT70	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT71	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT72	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT73	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT74	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—

Module Name	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
INTC	IDCNT75	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT76	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT77	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT78	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT79	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT80	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT81	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT82	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT83	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT84	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT85	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT86	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT87	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT88	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT89	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—

Module Name	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
INTC	IDCNT90	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT91	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT92	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT93	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT94	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT95	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT96	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT97	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT98	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT99	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT100	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT101	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT102	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT103	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT104	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—

Module Name	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
INTC	IDCNT105	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT106	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT107	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT108	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT109	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT110	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT111	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT112	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT113	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT114	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT115	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT116	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT117	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT118	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT119	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—

Module Name	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
INTC	IDCNT120	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT121	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT122	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT123	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT124	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT125	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT126	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT130	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT131	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT132	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT133	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT134	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT135	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT136	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT137	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—

Module Name	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
INTC	IDCNT138	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	IDCNT139	—	—	—	—	—	—	CPUN	INTEN
		—	—	—	MON	—	—	—	—
	DREQER0	—	—	—	—	CMI3	CMI2	CMI1	CMI0
	DREQER1	—	—	—	TGI4A	TGI3A	TGI2A	TGI1A	TGI0A
	DREQER2	IIC TXI3	IIC RXI3	IIC TXI2	IIC RXI2	IIC TXI1	IIC RXI1	IIC TXI0	IIC RXI0
	DREQER3	—	—	—	—	SCIF TXI5	SCIF RXI5	SCIF TXI4	SCIF RXI4
	DREQER4	SCIF TXI3	SCIF RXI3	SCIF TXI2	SCIF RXI2	SCIF TXI1	SCIF RXI1	SCIF TXI0	SCIF RXI0
	DREQER5	—	—	SSIF5	SSIF4	SSIF3	SSIF2	SSIF1	SSIF0
	DREQER6	—	—	—	—	SSTXI1	SSRXI1	SSTXI0	SSRXI0
	DREQER7	—	—	—	—	—	—	—	ADC ADI
	DREQER8	—	—	—	—	—	—	RM01	RM00
UBC	BAR_0	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24
		BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
		BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
		BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
	BAMR_0	BAM31	BAM30	BAM29	BAM28	BAM27	BAM26	BAM25	BAM24
		BAM23	BAM22	BAM21	BAM20	BAM19	BAM18	BAM17	BAM16
		BAM15	BAM14	BAM13	BAM12	BAM11	BAM10	BAM9	BAM8
		BAM7	BAM6	BAM5	BAM4	BAM3	BAM2	BAM1	BAM0
	BBR_0	—	—	UBID	DBE	—	—	—	CP
		CD[1:0]		ID[1:0]		RW[1:0]		SZ[1:0]	
	BDR_0	BD31	BD30	BD29	BD28	BD27	BD26	BD25	BD24
		BD23	BD22	BD21	BD20	BD19	BD18	BD17	BD16
		BD15	BD14	BD13	BD12	BD11	BD10	BD9	BD8
		BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
	BDMR_0	BDM31	BDM30	BDM29	BDM28	BDM27	BDM26	BDM25	BDM24
		BDM23	BDM22	BDM21	BDM20	BDM19	BDM18	BDM17	BDM16
		BDM15	BDM14	BDM13	BDM12	BDM11	BDM10	BDM9	BDM8
		BDM7	BDM6	BDM5	BDM4	BDM3	BDM2	BDM1	BDM0

Module Name	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
UBC	BAR_1	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24
		BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
		BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
		BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
	BAMR_1	BAM31	BAM30	BAM29	BAM28	BAM27	BAM26	BAM25	BAM24
		BAM23	BAM22	BAM21	BAM20	BAM19	BAM18	BAM17	BAM16
		BAM15	BAM14	BAM13	BAM12	BAM11	BAM10	BAM9	BAM8
		BAM7	BAM6	BAM5	BAM4	BAM3	BAM2	BAM1	BAM0
	BBR_1	—	—	UBID	DBE	—	—	—	CP
		CD[1:0]		ID[1:0]		RW[1:0]		SZ[1:0]	
	BDR_1	BD31	BD30	BD29	BD28	BD27	BD26	BD25	BD24
		BD23	BD22	BD21	BD20	BD19	BD18	BD17	BD16
		BD15	BD14	BD13	BD12	BD11	BD10	BD9	BD8
		BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
	BDMR_1	BDM31	BDM30	BDM29	BDM28	BDM27	BDM26	BDM25	BDM24
		BDM23	BDM22	BDM21	BDM20	BDM19	BDM18	BDM17	BDM16
		BDM15	BDM14	BDM13	BDM12	BDM11	BDM10	BDM9	BDM8
		BDM7	BDM6	BDM5	BDM4	BDM3	BDM2	BDM1	BDM0
	BRCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	CKS[1:0]	
		SCMFC0	SCMFC1	SCMFD0	SCMFD1	—	—	—	—
		—	PCB1	PCB0	—	—	—	—	—
Cache	CCR1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	ICF	—	—	ICE
		—	—	—	—	OCF	—	WT	OCE
	CCR2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	LE
		—	—	—	—	—	—	W3LOAD	W3LOCK
		—	—	—	—	—	—	W2LOAD	W2LOCK

Module Name	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
BSC	CS0CNT	—	—	—	—	—	—	—	—
		—	—	BSIZE[1:0]		—	—	—	EXENB
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS0REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS1CNT	—	—	—	—	—	—	—	—
		—	—	BSIZE[1:0]		—	—	—	EXENB
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS1REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS2CNT	—	—	—	—	—	—	—	—
		—	—	BSIZE[1:0]		—	—	—	EXENB
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS2REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS3CNT	—	—	—	—	—	—	—	—
		—	—	BSIZE[1:0]		—	—	—	EXENB
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—

Module Name	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
BSC	CS3REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS4CNT	—	—	—	—	—	—	—	—
		—	—	BSIZE[1:0]		—	—	—	EXENB
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS4REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS5CNT	—	—	—	—	—	—	—	—
		—	—	BSIZE[1:0]		—	—	—	EXENB
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS5REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	SDC0CNT	—	—	—	—	—	—	—	—
		—	—	BSIZE[1:0]		—	—	—	EXENB
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	SDC1CNT	—	—	—	—	—	—	—	—
		—	—	BSIZE[1:0]		—	—	—	EXENB
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—

Module Name	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
BSC	CSMOD0	PRMOD	—	PBCNT[1:0]		—	—	PWENB	PRENB
		—	—	—	—	EWENB	—	—	WRMOD
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS1WCNT0	—	—	—	CSRWAIT[4:0]				
		—	—	—	CSWWAIT[4:0]				
		—	—	—	—	—	CSPRWAIT[2:0]		
		—	—	—	—	—	CSPWWAIT[2:0]		
	CS2WCNT0	—	CSON[2:0]			—	WDON[2:0]		
		—	WRON[2:0]			—	RDON[2:0]		
		—	—	—	—	—	WDOFF[2:0]		
		—	CSWOFF[2:0]			—	CSROFF[2:0]		
	CSMOD1	PRMOD	—	PBCNT[1:0]		—	—	PWENB	PRENB
		—	—	—	—	EWENB	—	—	WRMOD
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS1WCNT1	—	—	—	CSRWAIT[4:0]				
		—	—	—	CSWWAIT[4:0]				
		—	—	—	—	—	CSPRWAIT[2:0]		
		—	—	—	—	—	CSPWWAIT[2:0]		
	CS2WCNT1	—	CSON[2:0]			—	WDON[2:0]		
		—	WRON[2:0]			—	RDON[2:0]		
		—	—	—	—	—	WDOFF[2:0]		
		—	CSWOFF[2:0]			—	CSROFF[2:0]		
	CSMOD2	PRMOD	—	PBCNT[1:0]		—	—	PWENB	PRENB
		—	—	—	—	EWENB	—	—	WRMOD
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—

Module Name	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
BSC	CS1WCNT2	—	—	—	CSRWAIT[4:0]				
		—	—	—	CSWWAIT[4:0]				
		—	—	—	—	—	CSPRWAIT[2:0]		
		—	—	—	—	—	CSPWWAIT[2:0]		
	CS2WCNT2	—	CSON[2:0]			—	WDON[2:0]		
		—	WRON[2:0]			—	RDON[2:0]		
		—	—	—	—	—	WDOFF[2:0]		
		—	CSWOFF[2:0]			—	CSROFF[2:0]		
	CSMOD3	PRMOD	—	PBCNT[1:0]		—	—	PWENB	PRENB
		—	—	—	—	EWENB	—	—	WRMOD
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS1WCNT3	—	—	—	CSRWAIT[4:0]				
		—	—	—	CSWWAIT[4:0]				
		—	—	—	—	—	CSPRWAIT[2:0]		
		—	—	—	—	—	CSPWWAIT[2:0]		
	CS2WCNT3	—	CSON[2:0]			—	WDON[2:0]		
		—	WRON[2:0]			—	RDON[2:0]		
		—	—	—	—	—	WDOFF[2:0]		
		—	CSWOFF[2:0]			—	CSROFF[2:0]		
	CSMOD4	PRMOD	—	PBCNT[1:0]		—	—	PWENB	PRENB
		—	—	—	—	EWENB	—	—	WRMOD
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS1WCNT4	—	—	—	CSRWAIT[4:0]				
		—	—	—	CSWWAIT[4:0]				
		—	—	—	—	—	CSPRWAIT[2:0]		
		—	—	—	—	—	CSPWWAIT[2:0]		

Module Name	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
BSC	CS2WCNT4	—	CSON[2:0]			—	WDON[2:0]		
		—	WRON[2:0]			—	RDON[2:0]		
		—	—	—	—	—	WDOFF[2:0]		
		—	CSWOFF[2:0]			—	CSROFF[2:0]		
	CSMOD5	PRMOD	—	PBCNT[1:0]		—	—	PWENB	PRENB
		—	—	—	—	EWENB	—	—	WRMOD
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CS1WCNT5	—	—	—	CSRWAIT[4:0]				
		—	—	—	CSWWAIT[4:0]				
		—	—	—	—	—	CSPRWAIT[2:0]		
		—	—	—	—	—	CSPWWAIT[2:0]		
	CS2WCNT5	—	CSON[2:0]			—	WDON[2:0]		
		—	WRON[2:0]			—	RDON[2:0]		
		—	—	—	—	—	WDOFF[2:0]		
		—	CSWOFF[2:0]			—	CSROFF[2:0]		
	SDRFCNT0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	DSFEN
	SDRFCNT1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	DRFEN
		DREFW[3:0]				DRFC[11:8]			
		DRFC[7:0]							
	SDIR0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	DPC[2:0]		
		DARFC[3:0]				DARFI[3:0]			

Module Name	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
BSC	SDIR1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	DINIST
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	DINIRQ
	SDPWDCNT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	DPWD
	SDDPWDCNT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	DDPD
	SD0ADR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	DDBW[1:0]	
		—	—	—	—	—	DSZ[2:0]		
	SD0TR	—	—	—	—	—	—	—	—
		—	—	—	—	—	DRAS[2:0]		
		—	—	DRCD[1:0]		DPCG[2:0]			DWR
		—	—	—	—	—	DCL[2:0]		
	SD0MOD	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	DMR[14:8]						
		DMR[7:0]							
	SD1ADR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	DDBW[1:0]	
		—	—	—	—	—	DSZ[2:0]		

Module Name	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
BSC	SD1TR	—	—	—	—	—	—	—	—
		—	—	—	—	—	DRAS[2:0]		
		—	—	DRCD[1:0]		DPCG[2:0]			DWR
		—	—	—	—	—	DCL[2:0]		
	SD1MOD	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	DMR[14:8]						
		DMR[7:0]							
	SDSTR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	DSRFST	DINIST	DPWDST	DDPDST	DMRSST
	SDCKSCNT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	DCKSEN
		—	—	—	—	—	—	—	—
		DCKSC[7:0]							
DMAC	DMCSADR0	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							

Module Name	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
DMAC	DMCDADR0	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							
	DMCBCT0	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:16]							
		CBC[15:8]							
		CBC[7:0]							
	DMMOD0	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	
	DMRSADR0	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							
	DMRDADR0	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							
	DMRBCT0	—	—	—	—	—	—	RBC[25:24]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							

Module Name	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
DMAC	DMACNTA0	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					
	DMACNTB0	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR
	DMCSADR1	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							
	DMCDADR1	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							
	DMCBCT1	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:16]							
		CBC[15:8]							
		CBC[7:0]							
	DMMOD1	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	
	DMRSADR1	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							

Module Name	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
DMAC	DMRDADR1	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							
	DMRBCT1	—	—	—	—	—	—	RBC[25:24]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							
	DMACNTA1	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					
	DMACNTB1	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR
	DMCSADR2	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							
	DMCDADR2	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							
	DMCBCT2	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:16]							
		CBC[15:8]							
		CBC[7:0]							

Module Name	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
DMAC	DMMOD2	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	
	DMRSADR2	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							
	DMRDADR2	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							
	DMRBCT2	—	—	—	—	—	—	RBC[25:24]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							
	DMACNTA2	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					
	DMACNTB2	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR
	DMCSADR3	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							

Module Name	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
DMAC	DMCDADR3	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							
	DMCBCT3	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:16]							
		CBC[15:8]							
		CBC[7:0]							
	DMMOD3	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	
	DMRSADR3	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							
	DMRDADR3	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							
	DMRBCT3	—	—	—	—	—	—	RBC[25:24]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							
	DMACNTA3	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					

Module Name	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
DMAC	DMACNTB3	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR
	DMCSADR4	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							
	DMCDADR4	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							
	DMCBCT4	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:16]							
		CBC[15:8]							
		CBC[7:0]							
	DMMOD4	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	
	DMRSADR4	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							
	DMRDADR4	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							

Module Name	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
DMAC	DMRBCT4	—	—	—	—	—	—	RBC[25:24]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							
	DMACNTA4	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					
	DMACNTB4	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR
	DMCSADR5	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							
	DMCDADR5	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							
	DMCBCT5	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:24]							
		CBC[15:8]							
		CBC[7:0]							
	DMMOD5	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	

Module Name	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
DMAC	DMRSADR5	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							
	DMRDADR5	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							
	DMRBCT5	—	—	—	—	—	—	RBC[25:0]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							
	DMACNTA5	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					
	DMACNTB5	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR
	DMCSADR6	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							
	DMCDADR6	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							

Module Name	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
DMAC	DMCBCT6	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:16]							
		CBC[15:8]							
		CBC[7:0]							
	DMMOD6	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	
	DMRSADR6	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							
	DMRDADR6	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							
	DMRBCT6	—	—	—	—	—	—	RBC[25:24]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							
	DMACNTA6	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					
	DMACNTB6	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR

Module Name	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
DMAC	DMCSADR7	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							
	DMCDADR7	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							
	DMCBCT7	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:16]							
		CBC[15:8]							
		CBC[7:0]							
	DMMOD7	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	
	DMRSADR7	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							
	DMRDADR7	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							
	DMRBCT7	—	—	—	—	—	—	RBC[25:24]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							

Module Name	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
DMAC	DMACNTA7	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					
	DMACNTB7	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR
	DMCSADR8	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							
	DMCDADR8	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							
	DMCBCT8	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:16]							
		CBC[15:8]							
		CBC[7:0]							
	DMMOD8	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	
	DMRSADR8	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							

Module Name	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
DMAC	DMRDADR8	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							
	DMRBCT8	—	—	—	—	—	—	RBC[25:24]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							
	DMACNTA8	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					
	DMACNTB8	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR
	DMCSADR9	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							
	DMCDADR9	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							
	DMCBCT9	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:16]							
		CBC[15:8]							
		CBC[7:0]							

Module Name	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
DMAC	DMMOD9	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	
	DMRSADR9	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							
	DMRDADR9	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							
	DMRBCT9	—	—	—	—	—	—	RBC[25:24]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							
	DMACNTA9	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					
	DMACNTB9	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR
	DMCSADR10	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							

Module Name	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
DMAC	DMCDADR10	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							
	DMCBCT10	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:16]							
		CBC[15:8]							
		CBC[7:0]							
	DMMOD10	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	
	DMRSADR10	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							
	DMRDADR10	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							
	DMRBCT10	—	—	—	—	—	—	RBC[25:24]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							
	DMACNTA10	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					

Module Name	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
DMAC	DMACNTB10	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR
	DMCSADR11	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							
	DMCDADR11	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							
	DMCBCT11	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:16]							
		CBC[15:8]							
		CBC[7:0]							
	DMMOD11	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	
	DMRSADR11	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							
	DMRDADR11	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							

Module Name	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
DMAC	DMRBCT11	—	—	—	—	—	—	RBC[25:24]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							
	DMACNTA11	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					
	DMACNTB11	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR
	DMCSADR12	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							
	DMCDADR12	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							
	DMCBCT12	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:16]							
		CBC[15:8]							
		CBC[7:0]							
	DMMOD12	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	

Module Name	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
DMAC	DMRSADR12	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							
	DMRDADR12	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							
	DMRBCT12	—	—	—	—	—	—	RBC[25:24]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							
	DMACNTA12	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					
	DMACNTB12	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR
	DMCSADR13	CSA[31:24]							
		CSA[23:16]							
		CSA[15:8]							
		CSA[7:0]							
	DMCDADR13	CDA[31:24]							
		CDA[23:16]							
		CDA[15:8]							
		CDA[7:0]							

Module Name	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
DMAC	DMCBCT13	—	—	—	—	—	—	CBC[25:24]	
		CBC[23:16]							
		CBC[15:8]							
		CBC[7:0]							
	DMMOD13	—	—	—	—	OPSEL[3:0]			
		—	—	—	—	—	SZSEL[2:0]		
		—	SAMOD[2:0]			—	DAMOD[2:0]		
		—	—	—	—	SACT	DACT	DTCM[1:0]	
	DMRSADR13	RSA[31:24]							
		RSA[23:16]							
		RSA[15:8]							
		RSA[7:0]							
	DMRDADR13	RDA[31:24]							
		RDA[23:16]							
		RDA[15:8]							
		RDA[7:0]							
	DMRBCT13	—	—	—	—	—	—	RBC[25:24]	
		RBC[23:16]							
		RBC[15:8]							
		RBC[7:0]							
	DMACNTA13	—	—	MDSEL[1:0]		—	—	DSEL[1:0]	
		—	—	—	—	—	—	STRG[1:0]	
		—	—	—	—	2DRLOD	BRLOD	SRLOD	DRLOD
		—	—	DCTG[5:0]					
	DMACNTB13	—	—	—	—	—	—	—	DEN
		—	—	—	—	—	—	—	DREQ
		—	—	—	—	—	—	—	ECLR
		—	—	—	—	—	—	—	DSCLR

Module Name	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
DMAC	DMSCNT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	DMST
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	DMICNT	DINTM0	DINTM1	DINTM2	DINTM3	DINTM4	DINTM5	DINTM6	DINTM7
		DINTM8	DINTM9	DINTM10	DINTM11	DINTM12	DINTM13	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	DMICNTA	DINTA0	DINTA1	DINTA2	DINTA3	DINTA4	DINTA5	DINTA6	DINTA7
		DINTA8	DINTA9	DINTA10	DINTA11	DINTA12	DINTA13	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	DMISTS	DISTS0	DISTS1	DISTS2	DISTS3	DISTS4	DISTS5	DISTS6	DISTS7
		DISTS8	DISTS9	DISTS10	DISTS11	DISTS12	DISTS13	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	DMEDET	DEDET0	DEDET1	DEDET2	DEDET3	DEDET4	DEDET5	DEDET6	DEDET7
		DEDET8	DEDET9	DEDET10	DEDET11	DEDET12	DEDET13	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	DMASTS	DASTS0	DASTS1	DASTS2	DASTS3	DASTS4	DASTS5	DASTS6	DASTS7
		DASTS8	DASTS9	DASTS10	DASTS11	DASTS12	DASTS13	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	DM2DCLM0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DCDN[15:8]							
		DCDN[7:0]							

Module Name	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
DMAC	DM2DROW0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRN[15:8]							
		DRN[7:0]							
	DM2DBLK0	—	—	—	—	—	—	—	—
		DBN[23:16]							
		DBN[15:8]							
		DBN[7:0]							
	DM2DNROST0	DNROST[31:24]							
		DNROST[23:16]							
		DNROST[15:8]							
		DNROST[7:0]							
	DM2DNBOST0	DNBOST[31:24]							
		DNBOST[23:16]							
		DNBOST[15:8]							
		DNBOST[7:0]							
	DM2DNLOST0	DNLOST[31:24]							
		DNLOST[23:16]							
		DNLOST[15:8]							
		DNLOST[7:0]							
	DMR2DCLM0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRCDN[15:8]							
		DRCDN[7:0]							
	DMR2DROW0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRRN[15:8]							
		DRRN[7:0]							

Module Name	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
DMAC	DMR2DBLK0	—	—	—	—	—	—	—	—
		DRBN[23:16]							
		DRBN[15:8]							
		DRBN[7:0]							
	DMR2DNROST0	DRNROST[31:24]							
		DRNROST[23:16]							
		DRNROST[15:8]							
		DRNROST[7:0]							
	DMR2DNBOST0	DRNBOST[31:24]							
		DRNBOST[23:16]							
		DRNBOST[15:8]							
		DRNBOST[7:0]							
	DMR2DNLOST0	DRNLOST[31:24]							
		DRNLOST[23:16]							
		DRNLOST[15:8]							
		DRNLOST[7:0]							
	DM2DCLM1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DCDN[15:8]							
		DCDN[7:0]							
	DM2DROW1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRN[15:8]							
		DRN[7:0]							
	DM2DBLK1	—	—	—	—	—	—	—	—
		DBN[23:16]							
		DBN[15:8]							
		DBN[7:0]							

Module Name	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
DMAC	DM2DNROST1	DNROST[31:24]							
		DNROST[23:16]							
		DNROST[15:8]							
		DNROST[7:0]							
	DM2DNBOST1	DNBOST[31:24]							
		DNBOST[23:16]							
		DNBOST[15:8]							
		DNBOST[7:0]							
	DM2DNLOST1	DNLOST[31:24]							
		DNLOST[23:16]							
		DNLOST[15:8]							
		DNLOST[7:0]							
	DMR2DCLM1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRCDN[15:8]							
		DRCDN[7:0]							
	DMR2DROW1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRRN[15:8]							
		DRRN[7:0]							
	DMR2DBLK1	—	—	—	—	—	—	—	—
		DRBN[23:16]							
		DRBN[15:8]							
		DRBN[7:0]							
	DMR2DNROST1	DRNROST[31:24]							
		DRNROST[23:16]							
		DRNROST[15:8]							
		DRNROST[7:0]							

Module Name	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
DMAC	DMR2DNBOST1	DRNBOST[31:24]							
		DRNBOST[23:16]							
		DRNBOST[15:8]							
		DRNBOST[7:0]							
	DMR2DNLOST1	DRNLOST[31:24]							
		DRNLOST[23:16]							
		DRNLOST[15:8]							
		DRNLOST[7:0]							
	DM2DCLM2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DCDN[15:8]							
		DCDN[7:0]							
	DM2DROW2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRN[15:8]							
		DRN[7:0]							
	DM2DBLK2	—	—	—	—	—	—	—	—
		DBN[23:16]							
		DBN[15:8]							
		DBN[7:0]							
	DM2DNROST2	DNROST[31:24]							
		DNROST[23:16]							
		DNROST[15:8]							
		DNROST[7:0]							
	DM2DNBOST2	DNBOST[31:24]							
		DNBOST[23:16]							
		DNBOST[15:8]							
		DNBOST[7:0]							

Module Name	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
DMAC	DM2DNLOST2	DNLOST[31:24]							
		DNLOST[23:16]							
		DNLOST[15:8]							
		DNLOST[7:0]							
	DMR2DCLM2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRCDN[15:8]							
		DRCDN[7:0]							
	DMR2DROW2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRRN[15:8]							
		DRRN[7:0]							
	DMR2DBLK2	—	—	—	—	—	—	—	—
		DRBN[23:16]							
		DRBN[15:8]							
		DRBN[7:0]							
	DMR2DNROST2	DRNROST[31:24]							
		DRNROST[23:16]							
		DRNROST[15:8]							
		DRNROST[7:0]							
	DMR2DNBOST2	DRNBOST[31:24]							
		DRNBOST[23:16]							
		DRNBOST[15:8]							
		DRNBOST[7:0]							
	DMR2DNLOST2	DRNLOST[31:24]							
		DRNLOST[23:16]							
		DRNLOST[15:8]							
		DRNLOST[7:0]							

Module Name	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
DMAC	DM2DCLM3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DCDN[15:8]							
		DCDN[7:0]							
	DM2DROW3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRN[15:8]							
		DRN[7:0]							
	DM2DBLK3	—	—	—	—	—	—	—	—
		DBN[23:16]							
		DBN[15:8]							
		DBN[7:0]							
	DM2DNROST3	DNROST[31:24]							
		DNROST[23:16]							
		DNROST[15:8]							
		DNROST[7:0]							
	DM2DNBOST3	DNBOST[31:24]							
		DNBOST[23:16]							
		DNBOST[15:8]							
		DNBOST[7:0]							
	DM2DNLOST3	DNLOST[31:24]							
		DNLOST[23:16]							
		DNLOST[15:8]							
		DNLOST[7:0]							
	DMR2DCLM3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRCDN[15:8]							
		DRCDN[7:0]							

Module Name	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
DMAC	DMR2DROW3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRRN[15:8]							
		DRRN[7:0]							
	DMR2DBLK3	—	—	—	—	—	—	—	—
		DRBN[23:16]							
		DRBN[15:8]							
		DRBN[7:0]							
	DMR2DNROST3	DRNROST[31:24]							
		DRNROST[23:16]							
		DRNROST[15:8]							
		DRNROST[7:0]							
	DMR2DNBOST3	DRNBOST[31:24]							
		DRNBOST[23:16]							
		DRNBOST[15:8]							
		DRNBOST[7:0]							
	DMR2DNLOST3	DRNLOST[31:24]							
		DRNLOST[23:16]							
		DRNLOST[15:8]							
		DRNLOST[7:0]							
	DM2DCLM4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DCDN[15:8]							
		DCDN[7:0]							
	DM2DROW4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRN[15::8]							
		DRN[7:0]							

Module Name	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
DMAC	DM2DBLK4	—	—	—	—	—	—	—	—
		DBN[23:16]							
		DBN[15:8]							
		DBN[7:0]							
	DM2DNROST4	DNROST[31:24]							
		DNROST[23:16]							
		DNROST[15:8]							
		DNROST[7:0]							
	DM2DNBOST4	DNBOST[31:24]							
		DNBOST[23:16]							
		DNBOST[15:8]							
		DNBOST[7:0]							
	DM2DNLOST4	DNLOST[31:24]							
		DNLOST[23:16]							
		DNLOST[15:8]							
		DNLOST[7:0]							
	DMR2DCLM4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRCDN[15:8]							
		DRCDN[7:0]							
	DMR2DROW4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRRN[15:8]							
		DRRN[7:0]							
	DMR2DBLK4	—	—	—	—	—	—	—	—
		DRBN[23:16]							
		DRBN[15:8]							
		DRBN[7:0]							

Module Name	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
DMAC	DMR2DNROST4	DRNROST[31:24]							
		DRNROST[23:16]							
		DRNROST[15:8]							
		DRNROST[7:0]							
	DMR2DNBOST4	DRNBOST[31:24]							
		DRNBOST[23:16]							
		DRNBOST[15:8]							
		DRNBOST[7:0]							
	DMR2DNLOST4	DRNLOST[31:24]							
		DRNLOST[23:16]							
		DRNLOST[15:8]							
		DRNLOST[7:0]							
	DM2DCLM5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DCDN[15:8]							
		DCDN[7:0]							
	DM2DROW5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRN[15:8]							
		DRN[7:0]							
	DM2DBLK5	—	—	—	—	—	—	—	—
		DBN[23:16]							
		DBN[15:8]							
		DBN[7:0]							
	DM2DNROST5	DNROST[31:24]							
		DNROST[23:16]							
		DNROST[15:8]							
		DNROST[7:0]							

Module Name	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
DMAC	DM2DNBOST5	DNBOST[31:24]							
		DNBOST[23:16]							
		DNBOST[15:8]							
		DNBOST[7:0]							
	DM2DNLOST5	DNLOST[31:24]							
		DNLOST[23:16]							
		DNLOST[15:8]							
		DNLOST[7:0]							
	DMR2DCLM5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRCDN[15:8]							
		DRCDN[7:0]							
	DMR2DROW5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRRN[15:8]							
		DRRN[7:0]							
	DMR2DBLK5	—	—	—	—	—	—	—	—
		DRBN[23:16]							
		DRBN[15:8]							
		DRBN[7:0]							
	DMR2DNROST5	DRNROST[31:24]							
		DRNROST[23:16]							
		DRNROST[15:8]							
		DRNROST[7:0]							
	DMR2DNBOST5	DRNBOST[31:24]							
		DRNBOST[23:16]							
		DRNBOST[15:8]							
		DRNBOST[7:0]							

Module Name	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
DMAC	DMR2DNLOST5	DRNLOST[31:24]							
		DRNLOST[23:16]							
		DRNLOST[15:8]							
		DRNLOST[7:0]							
	DM2DCLM6	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DCDN[15:8]							
		DCDN[7:0]							
	DM2DROW6	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRN[15:8]							
		DRN[7:0]							
	DM2DBLK6	—	—	—	—	—	—	—	—
		DRN[23:16]							
		DRN[15:8]							
		DRN[7:0]							
	DM2DNROST6	DNROST[31:24]							
		DNROST[23:16]							
		DNROST[15:8]							
		DNROST[7:0]							
	DM2DNBOST6	DNBOST[31:24]							
		DNBOST[23:16]							
		DNBOST[15:8]							
		DNBOST[7:0]							
	DM2DNLOST6	DNLOST[31:24]							
		DNLOST[23:16]							
		DNLOST[15:8]							
		DNLOST[7:0]							

Module Name	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
DMAC	DMR2DCLM6	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRCDN[15:8]							
		DRCDN[7:0]							
	DMR2DROW6	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRRN[15:8]							
		DRRN[7:0]							
	DMR2DBLK6	—	—	—	—	—	—	—	—
		DRBN[23:16]							
		DRBN[15:8]							
		DRBN[7:0]							
	DMR2DNROST6	DRNROST[31:24]							
		DRNROST[23:16]							
		DRNROST[15:8]							
		DRNROST[7:0]							
	DMR2DNBOST6	DRNBOST[31:24]							
		DRNBOST[23:16]							
		DRNBOST[15:8]							
		DRNBOST[7:0]							
	DMR2DNLOST6	DRNLOST[31:24]							
		DRNLOST[23:16]							
		DRNLOST[15:8]							
		DRNLOST[7:0]							
	DM2DCLM7	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DCDN[15:8]							
		DCDN[7:0]							

Module Name	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
DMAC	DM2DROW7	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRN[15:8]							
		DRN[7:0]							
	DM2DBLK7	—	—	—	—	—	—	—	—
		DBN[23:16]							
		DBN[15:8]							
		DBN[7:0]							
	DM2DNROST7	DNROST[31:24]							
		DNROST[23:16]							
		DNROST[15:8]							
		DNROST[7:0]							
	DM2DNBOST7	DNBOST[31:24]							
		DNBOST[23:16]							
		DNBOST[15:8]							
		DNBOST[7:0]							
	DM2DNLOST7	DNLOST[31:24]							
		DNLOST[23:16]							
		DNLOST[15:8]							
		DNLOST[7:0]							
	DMR2DCLM7	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRCDN[15:8]							
		DRCDN[7:0]							
	DMR2DROW7	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DRRN[15:8]							
		DRRN[7:0]							

Module Name	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
DMAC	DMR2DBLK7	—	—	—	—	—	—	—	—
		DRBN[23:16]							
		DRBN[15:8]							
		DRBN[7:0]							
	DMR2DNROST7	DRNROST[31:24]							
		DRNROST[23:16]							
		DRNROST[15:8]							
		DRNROST[7:0]							
	DMR2DNBOST7	DRNBOST[31:24]							
		DRNBOST[23:16]							
		DRNBOST[15:8]							
		DRNBOST[7:0]							
	DMR2DNLOST7	DRNLOST[31:24]							
		DRNLOST[23:16]							
		DRNLOST[15:8]							
		DRNLOST[7:0]							
MTU2	TCR_0	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
	TMDR_0	—	BFE	BFB	BFA	MD[3:0]			
	TIORH_0	IOB[3:0]				IOA[3:0]			
	TIORL_0	IOD[3:0]				IOC[3:0]			
	TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_0	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_0								
	TGRA_0								

Module Name	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
MTU2	TGRB_0								
	TGRC_0								
	TGRD_0								
	TGRE_0								
	TGRF_0								
	TIER2_0	TTGE2	—	—	—	—	—	TGIEF	TGIEE
	TSR2_0	—	—	—	—	—	—	TGFF	TGFE
	TBTM_0	—	—	—	—	—	TTSE	TTSB	TTSA
	TCR_1	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]		
	TMDR_1	—	—	—	—	MD[3:0]			
	TIOR_1	IOB[3:0]				IOA[3:0]			
	TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
	TSR_1	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_1								
	TGRA_1								
	TGRB_1								
	TICCR	—	—	—	—	I2BE	I2AE	I1BE	I1AE
	TCR_2	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]		
	TMDR_2	—	—	—	—	MD[3:0]			
	TIOR_2	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
	TSR_2	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA

Module Name	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
MTU2	TCNT_2								
	TGRA_2								
	TGRB_2								
	TCR_3	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
	TMDR_3	—	—	BFB	BFA	MD[3:0]			
	TIORH_3	IOB[3:0]				IOA[3:0]			
	TIORL_3	IOD[3:0]				IOC[3:0]			
	TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_3								
	TGRA_3								
	TGRB_3								
	TGRC_3								
	TGRD_3								
	TBTM_3	—	—	—	—	—	—	TTSB	TTSA
	TCR_4	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
	TMDR_4	—	—	BFB	BFA	MD[3:0]			
	TIORH_4	IOB[3:0]				IOA[3:0]			
	TIORL_4	IOD[3:0]				IOC[3:0]			
	TIER_4	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_4	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_4								

Module Name	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
MTU2	TGRA_4								
	TGRB_4								
	TGRC_4								
	TGRD_4								
	TBTM_4	—	—	—	—	—	—	TTSB	TTSA
	TADCR	BF[1:0]		—	—	—	—	—	—
		UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
	TADCORA_4								
	TADCORB_4								
	TADCOBRA_4								
	TADCOBRB_4								
	TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0
	TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
	TRWER	—	—	—	—	—	—	—	RWE
	TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
	TOCR1	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
	TOCR2	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
	TGCR	—	BDC	N	P	FB	WF	VF	UF
	TCDR								
	TDDR								

Module Name	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
MTU2	TCNTS								
	TCBR								
	TITCR	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]		
	TITCNT	—	3ACNT[2:0]			—	4VCNT[2:0]		
	TBTER	—	—	—	—	—	—	BTE[1:0]	
	TDER	—	—	—	—	—	—	—	TDER
	TWCR	CCE	—	—	—	—	—	—	WRE
	TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
CMT	CMSTR01	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	STR1	STR0
	CMCSR0	—	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CKS[1:0]	
	CMCNT0								
	CMCOR0								
	CMCSR1	—	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CKS[1:0]	
	CMCNT1								
	CMCOR1								
	CMSTR23	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	STR3	STR2
	CMCSR2	—	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CKS[1:0]	
	CMCNT2								

Module Name	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
CMT	CMCOR2								
	CMCSR3	—	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CKS[1:0]	
	CMCNT3								
	CMCOR3								
WDT	WTCSR0	IOVF	WT/IT	TME	—	—	CKS[2:0]		
	WTCNT0	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
	WRCSR0	WOVF	RSTE	RSTS	—	—	—	—	—
	WTCSR1	IOVF	WT/IT	TME	—	—	CKS[2:0]		
	WTCNT1	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
	WRCSR1	WOVF	RSTE	—	—	—	—	—	—
RTC	R64CNT	—	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz
	RSECCNT	—	10 seconds			1 second			
	RMINCNT	—	10 minutes			1 minute			
	RHRCNT	—	—	10 hours		1 hour			
	RWKCNT	—	—	—	—	—	Day		
	RDAYCNT	—	—	10 days		1 day			
	RMONCNT	—	—	—	10 months	1 month			
	RYRCNT	1000 years				100 years			
		10 years				1 year			
	RSECAR	ENB	10 seconds			1 second			
	RMINAR	ENB	10 minutes			1 minute			
	RHRAR	ENB	—	10 hours		1 hour			
	RWKAR	ENB	—	—	—	—	Day		
	RDAYAR	ENB	—	10 days		1 day			
	RMONAR	ENB	—	—	10 months	1 month			
	RYRAR	1000 years				100 years			
		10 years				1 year			

Module Name	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
RTC	RCR1	CF	—	—	CIE	AIE	—	—	AF
	RCR2	PEF	PES[2:0]			RTCEN	ADJ	RESET	START
	RCR3	ENB	—	—	—	—	—	—	—
SCIF	SCSMR_0	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1:0]	
	SCBRR_0								
	SCSCR_0	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1:0]	
	SCFTDR_0								
	SCFSR_0	PER[3:0]				FER[3:0]			
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_0								
	SCFCR_0	—	—	—	—	—	RSTRG[2:0]		
		RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP
	SCFDR_0	—	—	—	T[4:0]				
		—	—	—	R[4:0]				
	SCSPTR_0	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_0	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_1	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1:0]	
	SCBRR_1								
	SCSCR_1	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1:0]	
	SCFTDR_1								
	SCFSR_1	PER[3:0]				FER[3:0]			
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_1								

Module Name	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
SCIF	SCFCR_1	—	—	—	—	—	RSTRG[2:0]		
		RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP
	SCFDR_1	—	—	—	T[4:0]				
		—	—	—	R[4:0]				
	SCSPTR_1	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_1	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_2	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1:0]	
	SCBRR_2								
	SCSCR_2	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1:0]	
	SCFTDR_2								
	SCFSR_2	PER[3:0]				FER[3:0]			
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_2								
	SCFCR_2	—	—	—	—	—	RSTRG[2:0]		
		RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP
	SCFDR_2	—	—	—	T[4:0]				
		—	—	—	R[4:0]				
	SCSPTR_2	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_2	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS

Module Name	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
SCIF	SCSMR_3	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1:0]	
	SCBRR_3								
	SCSCR_3	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1:0]	
	SCFTDR_3								
	SCFSR_3	PER[3:0]				FER[3:0]			
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_3								
	SCFCR_3	—	—	—	—	—	RSTRG[2:0]		
		RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP
	SCFDR_3	—	—	—	T[4:0]				
		—	—	—	R[4:0]				
	SCSPTR_3	—	—	—	—	—	—	—	—
		RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_3	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_4	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1:0]	
	SCBRR_4								
	SCSCR_4	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1:0]	
	SCFTDR_4								
	SCFSR_4	PER[3:0]				FER[3:0]			
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_4								
	SCFCR_4	—	—	—	—	—	RSTRG[2:0]		
		RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP

Module Name	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
SCIF	SCFDR_4	—	—	—	T[4:0]				
		—	—	—	R[4:0]				
	SCSPTR_4	—	—	—	—	—	—	—	—
		RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_4	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_5	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1:0]	
	SCBRR_5								
	SCSCR_5	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1:0]	
	SCFTDR_5								
	SCFSR_5	PER[3:0]				FER[3:0]			
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_5								
	SCFCR_5	—	—	—	—	—	RSTRG[2:0]		
		RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP
	SCFDR_5	—	—	—	T[4:0]				
		—	—	—	R[4:0]				
	SCSPTR_5	—	—	—	—	—	—	—	—
		RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_5	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS

Module Name	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
SSU	SSCRH_0	MSS	BIDE	—	SOL	SOLP	—	CSS[1:0]	
	SSCRL_0	—	SSUMS	SRES	—	—	—	DATS[1:0]	
	SSMR_0	MLS	CPOS	CPHS	—	—	CKS[2:0]		
	SSER_0	TE	RE	—	—	TEIE	TIE	RIE	CEIE
	SSSR_0	—	ORER	—	—	TEND	TDRE	RDRF	CE
	SSCR2_0	—	—	—	TENDSTS	SCSATS	SSODTS	—	—
	SSTDR0_0								
	SSTDR1_0								
	SSTDR2_0								
	SSTDR3_0								
	SSRDR0_0								
	SSRDR1_0								
	SSRDR2_0								
	SSRDR3_0								
	SSCRH_1	MSS	BIDE	—	SOL	SOLP	—	CSS[1:0]	
	SSCRL_1	—	SSUMS	SRES	—	—	—	DATS[1:0]	
	SSMR_1	MLS	CPOS	CPHS	—	—	CKS[2:0]		
	SSER_1	TE	RE	—	—	TEIE	TIE	RIE	CEIE
	SSSR_1	—	ORER	—	—	TEND	TDRE	RDRF	CE
	SSCR2_1	—	—	—	TENDSTS	SCSATS	SSODTS	—	—
	SSTDR0_1								
	SSTDR1_1								
	SSTDR2_1								
	SSTDR3_1								
	SSRDR0_1								
	SSRDR1_1								
	SSRDR2_1								
	SSRDR3_1								

Module Name	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
IIC3	ICCR1_0	ICE	RCVD	MST	TRS	CKS[3:0]			
	ICCR2_0	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—
	ICMR_0	MLS	—	—	—	BCWP	BC[2:0]		
	ICIER_0	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_0	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_0	SVA[6:0]							FS
	ICDRT_0								
	ICDRR_0								
	NF2CYC_0	—	—	—	—	—	—	PRS	NF2CYC
	ICCR1_1	ICE	RCVD	MST	TRS	CKS[3:0]			
	ICCR2_1	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—
	ICMR_1	MLS	—	—	—	BCWP	BC[2:0]		
	ICIER_1	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_1	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_1	SVA[6:0]							FS
	ICDRT_1								
	ICDRR_1								
	NF2CYC_1	—	—	—	—	—	—	PRS	NF2CYC
	ICCR1_2	ICE	RCVD	MST	TRS	CKS[3:0]			
	ICCR2_2	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—
	ICMR_2	MLS	—	—	—	BCWP	BC[2:0]		
	ICIER_2	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_2	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_2	SVA[6:0]							FS
	ICDRT_2								
	ICDRR_2								
	NF2CYC_2	—	—	—	—	—	—	PRS	NF2CYC
	ICCR1_3	ICE	RCVD	MST	TRS	CKS[3:0]			
	ICCR2_3	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—
	ICMR_3	MLS	—	—	—	BCWP	BC[2:0]		
	ICIER_3	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT

Module Name	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
IIC3	ICSR_3	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_3	SVA[6:0]							FS
	ICDRT_3								
	ICDRR_3								
	NF2CYC_3	—	—	—	—	—	—	PRS	NF2CYC
SSIF	SSICR_0	CKS[1:0]		—	—	UIEN	OIEN	IIEN	—
		CHNL[1:0]		DWL[2:0]		SWL[2:0]			
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		CKDV[3:0]				MUEN	—	TRMD	EN
	SSISR_0	—	—	—	—	UIRQ	OIRQ	IIRQ	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO[1:0]		SWNO	IDST
	SSIFCR_0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		TTRG[1:0]		RTRG[1:0]		—	TIE	RIE	FRST
	SSIFSR_0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DC[3:0]				—	—	TDE	RDF
	SSIFDR_0								
	SSICR_1	CKS[1:0]		—	—	UIEN	OIEN	IIEN	—
		CHNL[1:0]		DWL[2:0]		SWL[2:0]			
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		CKDV[3:0]				MUEN	—	TRMD	EN

Module Name	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
SSIF	SSISR_1	—	—	—	—	UIRQ	OIRQ	IIRQ	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO[1:0]		SWNO	IDST
	SSIFCR_1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		TTRG[1:0]		RTRG[1:0]		—	TIE	RIE	FRST
	SSIFSR_1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DC[3:0]				—	—	TDE	RDF
	SSIFDR_1								
	SSICR_2	CKS[1:0]		—	—	UIEN	OIEN	IIEN	—
		CHNL[1:0]		DWL[2:0]			SWL[2:0]		
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		CKDV[3:0]				MUEN	—	TRMD	EN
	SSISR_2	—	—	—	—	UIRQ	OIRQ	IIRQ	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO[1:0]		SWNO	IDST
	SSIFCR_2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		TTRG[1:0]		RTRG[1:0]		—	TIE	RIE	FRST

Module Name	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
SSIF	SSIFSR_2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DC[3:0]				—	—	TDE	RDF
	SSIFDR_2								
	SSICR_3	CKS[1:0]		—	—	UIEN	OIEN	IIEN	—
		CHNL[1:0]		DWL[2:0]			SWL[2:0]		
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		CKDV[3:0]				MUEN	—	TRMD	EN
	SSISR_3	—	—	—	—	UIRQ	OIRQ	IIRQ	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO[1:0]		SWNO	IDST
	SSIFCR_3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		TTRG[1:0]		RTRG[1:0]		—	TIE	RIE	FRST
	SSIFSR_3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DC[3:0]				—	—	TDE	RDF
	SSIFDR_3								

Module Name	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
SSIF	SSICR_4	CKS[1:0]		—	—	UIEN	OIEN	IIEN	—
		CHNL[1:0]		DWL[2:0]			SWL[2:0]		
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		CKDV[3:0]				MUEN	—	TRMD	EN
	SSISR_4	—	—	—	—	UIRQ	OIRQ	IIRQ	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO[1:0]		SWNO	IDST
	SSIFCR_4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		TTRG[1:0]		RTRG[1:0]		—	TIE	RIE	FRST
	SSIFSR_4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DC[3:0]				—	—	TDE	RDF
	SSIFDR_4								
	SSICR_5	CKS[1:0]		—	—	UIEN	OIEN	IIEN	—
		CHNL[1:0]		DWL[2:0]			SWL[2:0]		
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		CKDV[3:0]				MUEN	—	TRMD	EN
	SSISR_5	—	—	—	—	UIRQ	OIRQ	IIRQ	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO[1:0]		SWNO	IDST

Module Name	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
SSIF	SSIFCR_5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		TTRG[1:0]		RTRG[1:0]		—	TIE	RIE	FRST
	SSIFSR_5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DC[3:0]				—	—	TDE	RDF
	SSIFDR_5								
RCAN-TL1	MCR_0	MCR15	MCR14	—	—	—	TST[2:0]		
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	GSR_0	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1_0	TSG1[3:0]				—	TSG2[2:0]		
		—	—	SJW[1]	SJW[0]	—	—	—	BSP
	BCR0_0	—	—	—	—	—	—	—	—
		BRP[7:0]							
	IRR_0	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	IMR_0	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC_REC_0	TEC[7:0]							
		REC[7:0]							
	TXPR1_0	TXPR1[15:8]							
		TXPR1[7:0]							
	TXPR0_0	TXPR0[15:8]							
		TXPR0[7:1]							—

Module Name	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
RCAN-TL1	TXCR1_0	TXCR1[15:8]							
		TXCR1[7:0]							
	TXCR0_0	TXCR0[15:8]							
		TXCR0[7:1]							—
	TXACK1_0	TXACK1[15:8]							
		TXACK1[7:0]							
	TXACK0_0	TXACK0[15:8]							
		TXACK0[7:1]							—
	ABACK1_0	ABACK1[15:8]							
		ABACK1[7:0]							
	ABACK0_0	ABACK0[15:8]							
		ABACK0[7:1]							—
	RXPR1_0	RXPR1[15:8]							
		RXPR1[7:0]							
	RXPR0_0	RXPR0[15:8]							
		RXPR0[7:0]							
	RFPR1_0	RFPR1[15:8]							
		RFPR1[7:0]							
	RFPR0_0	RFPR0[15:8]							
		RFPR0[7:0]							
	MBIMR1_0	MBIMR1[15:8]							
		MBIMR1[7:0]							
	MBIMR0_0	MBIMR0[15:8]							
		MBIMR0[7:0]							
	UMSR1_0	UMSR1[15:8]							
		UMSR1[7:0]							
	UMSR0_0	UMSR0[15:8]							
		UMSR0[7:0]							
	TTCR0_0	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—
		—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0

Module Name	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
RCAN-TL1	CMAX_TEW_0	—	—	—	—	—	CMAX[2:0]		
		—	—	—	—	TEW[3:0]			
	RFTROFF_0	RFTROFF[7:0]							
		—	—	—	—	—	—	—	—
	TSR_0	—	—	—	—	—	—	—	—
		—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0
	CCR_0	—	—	—	—	—	—	—	—
		—	—	CCR[5:0]					
	TCNTR_0	TCNTR[15:8]							
		TCNTR[7:0]							
	CYCTR_0	CYCTR[15:8]							
		CYCTR[7:0]							
	RFMK_0	RFMK[15:8]							
		RFMK[7:0]							
	TCMR0_0	TCMR0[15:8]							
		TCMR0[7:0]							
	TCMR1_0	TCMR1[15:8]							
		TCMR1[7:0]							
	TCMR2_0	TCMR2[15:8]							
		TCMR2[7:0]							
	TTTSEL_0	—	TTTSEL[14:8]						
		—	—	—	—	—	—	—	—
	MBn_CONTROL0_H_0 (n = 0 to 31)	—	STDID[10:4]						
		STDID[3:0]				RTR	IDE	EXTID[17:16]	
	MBn_CONTROL0_H_0 (n = 0 to 31)	IDE	RTR	—	STDID[10:6]				
		STDID[5:0]						EXTID[17:16]	
	MBn_CONTROL0_L_0 (n = 0 to 31)	EXTID[15:8]							
		EXTID[7:0]							

Module Name	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
RCAN-TL1	MBn_LAFM0_0 (n = 0 to 31)	—	STDID_LAFM[10:4]						
		STDID_LAFM[3:0]				—	IDE	EXTID_LAFM[17:16]	
	MBn_LAFM0_0 (n = 0 to 31)	IDE	—	—	STDID_LAFM[10:6]				
		STDID_LAFM[5:0]						EXTID_LAFM[17:16]	
	MBn_LAFM1_0 (n = 0 to 31)	EXTID_LAFM[15:8]							
		EXTID_LAFM[7:0]							
	MBn_DATA_01_0 (n = 0 to 31)	MSG_DATA_0							
		MSG_DATA_1							
	MBn_DATA_23_0 (n = 0 to 31)	MSG_DATA_2							
		MSG_DATA_3							
	MBn_DATA_45_0 (n = 0 to 31)	MSG_DATA_4							
		MSG_DATA_5							
	MBn_DATA_67_0 (n = 0 to 31)	MSG_DATA_6							
		MSG_DATA_7							
	MBn_CONTROL1_0 (n = 0)	—	—	NMC	—	—	MBC[2:0]		
		—	—	—	—	DLC[3:0]			
	MBn_CONTROL1_0 (n = 0 to 31)	—	—	NMC	ATX	DART	MBC[2:0]		
		—	—	—	—	DLC[3:0]			
	MBn_TIMESTAMP_0 (n = 0 to 15, 30, 31)	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8
		TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
	MBn_TTT_0 (n = 24 to 30)	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8
		TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0
	MBn_TTCONTROL_0 (n = 24 to 29)	TTW[1:0]		Offset[5:0]					
		—	—	—	—	—	rep_factor[2:0]		
	MCR_1	MCR15	MCR14	—	—	—	TST[2:0]		
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0

Module Name	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
RCAN-TL1	GSR_1	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1_1	TSG1[3:0]				—	TSG2[2:0]		
		—	—	SJW[1:0]		—	—	—	BSP
	BCR0_1	—	—	—	—	—	—	—	—
		BRP[7:0]							
	IRR_1	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	IMR_1	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC_REC_1	TEC[7:0]							
		REC[7:0]							
	TXPR1_1	TXPR1[15:8]							
		TXPR1[7:0]							
	TXPR0_1	TXPR0[15:8]							
		TXPR0[7:1]							—
	TXCR1_1	TXCR1[15:8]							
		TXCR1[7:0]							
	TXCR0_1	TXCR0[15:8]							
		TXCR0[7:1]							—
	TXACK1_1	TXACK1[15:8]							
		TXACK1[7:0]							
	TXACK0_1	TXACK0[15:8]							
		TXACK0[7:1]							—
	ABACK1_1	ABACK1[15:8]							
		ABACK1[7:0]							
	ABACK0_1	ABACK0[15:8]							
		ABACK0[7:1]							—

Module Name	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
RCAN-TL1	RXPR1_1	RXPR1[15:8]							
		RXPR1[7:0]							
	RXPR0_1	RXPR0[15:8]							
		RXPR0[7:0]							
	RFPR1_1	RFPR1[15:8]							
		RFPR1[7:0]							
	RFPR0_1	RFPR0[15:8]							
		RFPR0[7:0]							
	MBIMR1_1	MBIMR1[15:8]							
		MBIMR1[7:0]							
	MBIMR0_1	MBIMR0[15:8]							
		MBIMR0[7:0]							
	UMSR1_1	UMSR1[15:8]							
		UMSR1[7:0]							
	UMSR0_1	UMSR0[15:8]							
		UMSR0[7:0]							
	TTCR0_1	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—
		—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
	CMAX_TEW_1	—	—	—	—	—	CMAX[2:0]		
		—	—	—	—	TEW[3:0]			
	RFTROFF_1	RFTROFF[7:0]							
		—	—	—	—	—	—	—	—
	TSR_1	—	—	—	—	—	—	—	—
		—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0
	CCR_1	—	—	—	—	—	—	—	—
		—	—	CCR[5:0]					
	TCNTR_1	TCNTR[15:8]							
		TCNTR[7:0]							
	CYCTR_1	CYCTR[15:8]							
		CYCTR[7:0]							

Module Name	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
RCAN-TL1	RFMK_1	RFMK[15:8]							
		RFMK[7:0]							
	TCMR0_1	TCMR0[15:8]							
		TCMR0[7:0]							
	TCMR1_1	TCMR1[15:8]							
		TCMR1[7:0]							
	TCMR2_1	TCMR2[15:8]							
		TCMR2[7:0]							
	TTTSEL_1	—	TTTSEL[14:8]						
		—	—	—	—	—	—	—	—
	MBn_CONTROL0_H_1 (n = 0 to 31)	—	STDID[10:4]						
		STDID[3:0]				RTR	IDE	EXTID[17:16]	
	MBn_CONTROL0_H_1 (n = 0 to 31)	IDE	RTR	—	STDID[10:6]				
		STDID[5:0]						EXTID[17:16]	
	MBn_CONTROL0_L_1 (n = 0 to 31)	EXTID[15:8]							
		EXTID[7:0]							
	MBn_LAFM0_1 (n = 0 to 31)	—	STDID_LAFM[10:4]						
		STDID_LAFM[3:0]				—	IDE	EXTID_LAFM[17:16]	
	MBn_LAFM0_1 (n = 0 to 31)	IDE	—	—	STDID_LAFM[10:6]				
		STDID_LAFM[5:0]						EXTID_LAFM[17:16]	
	MBn_LAFM1_1 (n = 0 to 31)	EXTID_LAFM[15:8]							
		EXTID_LAFM[7:0]							
	MBn_DATA_01_1 (n = 0 to 31)	MSG_DATA_0							
		MSG_DATA_1							
	MBn_DATA_23_1 (n = 0 to 31)	MSG_DATA_2							
		MSG_DATA_3							
	MBn_DATA_45_1 (n = 0 to 31)	MSG_DATA_4							
		MSG_DATA_5							

Module Name	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
RCAN-TL1	MBn_DATA_67_1 (n = 0 to 31)	MSG_DATA_6							
		MSG_DATA_7							
	MBn_CONTROL 1_1 (n = 0)	—	—	NMC	—	—	MBC[2:0]		
		—	—	—	—	DLC[3:0]			
	MBn_CONTROL 1_1 (n = 1 to 31)	—	—	NMC	ATX	DART	MBC[2:0]		
		—	—	—	—	DLC[3:0]			
	MBn_TIMESTAMP_1 (n = 0 to 15, 30, 31)	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8
		TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
	MBn_TTT_1 (n = 24 to 30)	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8
		TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0
	MBn_TTCONTROL_1 (n = 24 to 29)	TTW[1:0]		Offset[5:0]					
		—	—	—	—	—	rep_factor[2:0]		
ADC	ADDRA								
				—	—	—	—	—	—
	ADDRB								
				—	—	—	—	—	—
	ADDRC								
				—	—	—	—	—	—
	ADDRD								
				—	—	—	—	—	—
	ADDRE								
				—	—	—	—	—	—
	ADDRF								
				—	—	—	—	—	—
	ADDRG								
				—	—	—	—	—	—
	ADDRH								
				—	—	—	—	—	—
	ADCSR	ADF	ADIE	ADST	—	TRGS[3:0]			
		CKS[1:0]		MDS[2:0]		CH[2:0]			

Module Name	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
DAC	DADR0								
	DADR1								
	DACR	DAOE1	DAOE0	DAE	—	—	—	—	—
FLCTL	FLCMNCR	—	—	—	—	—	—	ECCPOS[2]	4ECCCNTEN
		4ECCEN	4ECCCORRECT	—	—	—	SNAND	QTSEL	—
		FCKSEL	—	ECCPOS[1:0]		ACM[1:0]		NANDWF	—
		—	—	—	—	CE	—	—	TYPESEL
	FLCMDCR	ADRCNT2	SCTCNT[19:16]				ADRMD	CDSRC	DOSR
		—	—	SELRW	DOADR	ADRCNT[1:0]		DOCMD2	DOCMD1
		SCTCNT[15:8]							
		SCTCNT[7:0]							
	FLCMCDR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		CMD2[7:0]							
		CMD1[7:0]							
	FLADR (ADRMD = 1)	ADR4[7:0]							
		ADR3[7:0]							
		ADR2[7:0]							
		ADR1[7:0]							
	FLADR (ADRMD=0)	—	—	—	—	—	—	ADR[25:24]	
		ADR[23:16]							
		ADR[15:8]							
		ADR[7:0]							
	FLADR2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		ADR5[7:0]							

Module Name	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
FLCTL	FLDTCNTR	ECFLW[7:0]							
		DTFLW[7:0]							
		—	—	—	—	DTCNT[11:8]			
		DTCNT[7:0]							
	FLDATAR	DT4[7:0]							
		DT3[7:0]							
		DT2[7:0]							
		DT1[7:0]							
	FLINTDMACR	—	—	—	—	—	—	4ECEINTE	ECERINTE
		—	—	FIFOTRG[1:0]		AC1CLR	AC0CLR	DREQ1EN	DREQ0EN
		—	—	—	—	—	—	ECERB	STERB
		BTOERB	TRREQF1	TRREQF0	STERINTE	RBERINTE	TEINTE	TRINTE1	TRINTE0
	FLBSYTMR	—	—	—	—	—	—	—	—
		—	—	—	—	RBTMOUT[19:16]			
		RBTMOUT[15:8]							
		RBTMOUT[7:0]							
	FLBSYCNT	STAT[7:0]							
		—	—	—	—	RBTIMCNT[19:16]			
		RBTIMCNT[15:8]							
		RBTIMCNT[7:0]							
	FLDTFIFO	DTFO[31:24]							
		DTFO[23:16]							
		DTFO[15:8]							
		DTFO[7:0]							
	FLECFIFO	ECFO[31:24]							
		ECFO[23:16]							
		ECFO[15:8]							
		ECFO[7:0]							
	FLTRCR	—	—	—	—	—	—	TREND	TRSTRT

Module Name	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
FLCTL	FL4ECCRES1	—	—	—	—	—	—	LOC1[9:8]	
		LOC1[7:0]							
		—	—	—	—	—	—	PAT1[9:8]	
		PAT1[7:0]							
	FL4ECCRES2	—	—	—	—	—	—	LOC2[9:8]	
		LOC2[7:0]							
		—	—	—	—	—	—	PAT2[9:8]	
		PAT2[7:0]							
	FL4ECCRES3	—	—	—	—	—	—	LOC3[9:8]	
		LOC3[7:0]							
		—	—	—	—	—	—	PAT3[9:8]	
		PAT3[7:0]							
	FL4ECCRES4	—	—	—	—	—	—	LOC4[9:8]	
		LOC4[7:0]							
		—	—	—	—	—	—	PAT4[9:8]	
		PAT4[7:0]							
	FL4ECCCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	4ECCFA	4ECCEND	4ECCEXST
	FL4ECCCNT	—	—	—	—	—	ERRCNT[10:8]		
		ERRCNT[7:0]							
		—	—	—	—	—	—	—	—
		—	—	—	—	—	ERRMAX[2:0]		
USB	SYSCFG0	—	—	—	—	—	SCKE	—	—
		HSE	DCFM	DRPD	DPRPU	—	—	—	USBE
	SYSCFG1	—	—	—	—	—	—	—	—
		HSE	—	DRPD	—	BWAIT[3:0]			
	SYSSTS0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	LNST[1:0]	

Module Name	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
USB	SYSSTS1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	LNST[1:0]	
	DVSTCTR0	—	—	—	—	—	—	—	WKUP
		RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
	DVSTCTR1	—	—	—	—	—	—	—	—
		RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
	TESTMODE	—	—	—	—	—	—	—	—
		—	—	—	—	UTST[3:0]			
	D0FBCFG	—	—	DFACC[1:0]		—	—	—	—
		—	—	—	TENDE	—	—	—	—
	D1FBCFG	—	—	DFACC[1:0]		—	—	—	—
		—	—	—	TENDE	—	—	—	—
	CFIFO	FIFOPORT[31:24]							
		FIFOPORT[23:16]							
		FIFOPORT[15:8]							
		FIFOPORT[7:0]							
	D0FIFO	FIFOPORT[31:24]							
		FIFOPORT[23:16]							
		FIFOPORT[15:8]							
		FIFOPORT[7:0]							
	D1FIFO	FIFOPORT[31:0]							
		FIFOPORT[23:16]							
		FIFOPORT[15:8]							
		FIFOPORT[7:0]							
	CFIFOSEL	RCNT	REW	—	—	MBW[1:0]		—	BIGEND
		—	—	ISEL	—	CURPIPE[3:0]			
	CFIFOCTR	BVAL	BCLR	FRDY	—	DTLN[11:8]			
		DTLN[7:0]							
	D0FIFOSEL	RCNT	REW	DCLRM	DREQE	MBW[1:0]		—	BIGEND
		—	—	—	—	CURPIPE[3:0]			

Module Name	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
USB	D0FIFOCTR	BVAL	BCLR	FRDY	—	DTLN[11:8]			
		DTLN[7:0]							
	D1FIFOSEL	RCNT	REW	DCLRM	DREQE	MBW[1:0]		—	BIGEND
		—	—	—	—	CURPIPE[3:0]			
	D1FIFOCTR	BVAL	BCLR	FRDY	—	DTLN[11:8]			
		DTLN[7:0]							
	INTENB0	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE
		—	—	—	—	—	—	—	—
	INTENB1	—	BCHGE	—	DTCHE	ATTCHE	—	—	—
		—	EOFERRE	SIGNE	SACKE	—	—	—	—
	INTENB2	—	BCHGE	—	DTCHE	ATTCHE	—	—	—
		—	EOFERRE	—	—	—	—	—	—
	BRDYENB	—	—	—	—	—	—	PIPE9BRDYE	PIPE8BRDYE
		PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	PIPE3BRDYE	PIPE2BRDYE	PIPE1BRDYE	PIPE0BRDYE
	NRDYENB	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE
		PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
	BEMPENB	—	—	—	—	—	—	PIPE9BEMPE	PIPE8BEMPE
		PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	PIPE3BEMPE	PIPE2BEMPE	PIPE1BEMPE	PIPE0BEMPE
	SOFCFG	—	—	—	—	—	—	—	TRNENSEL
		—	BRDYM	—	—	—	—	—	—
	INTSTS0	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY
		VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
	INTSTS1	—	BCHG	—	DTCH	ATTCH	—	—	—
		—	EOFERR	SIGN	SACK	—	—	—	—
	INTSTS2	—	BCHG	—	DTCH	ATTCH	—	—	—
		—	EOFERR	—	—	—	—	—	—
	BRDYSTS	—	—	—	—	—	—	PIPE9BRDY	PIPE8BRDY
		PIPE7BRDY	PIPE6BRDY	PIPE5BRDY	PIPE4BRDY	PIPE3BRDY	PIPE2BRDY	PIPE1BRDY	PIPE0BRDY

Module Name	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
USB	NRDYSTS	—	—	—	—	—	—	PIPE9NRDY	PIPE8NRDY
		PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	PIPE3NRDY	PIPE2NRDY	PIPE1NRDY	PIPE0NRDY
	BEMPSTS	—	—	—	—	—	—	PIPE9BEMP	PIPE8BEMP
		PIPE7BEMP	PIPE6BEMP	PIPE5BEMP	PIPE4BEMP	PIPE3BEMP	PIPE2BEMP	PIPE1BEMP	PIPE0BEMP
	FRMNUM	OVRN	CRCE	—	—	—	FRNM[10:8]		
		FRNM[7:0]							
	UFRMNUM	—	—	—	—	—	—	—	—
		—	—	—	—	—	UFRNM[2:0]		
	USBADDR	—	—	—	—	—	—	—	—
		—	USBADDR[6:0]						
	USBREQ	BREQUEST[7:0]							
		BMREQUESTTYPE[7:0]							
	USBVAL	WVALUE[15:8]							
		WVALUE[7:0]							
	USBINDX	WINDEX[15:8]							
		WINDEX[7:0]							
	USBLENG	WLENGTH[15:8]							
		WLENGTH[7:0]							
	DCPCFG	—	—	—	—	—	—	—	CNTMD
		SHTNAK	—	—	DIR	—	—	—	—
	DCPMAXP	DEVSEL[3:0]				—	—	—	—
		—	MXPS[6:0]						
	DCPCTR	BSTS	SUREQ	CSCLR	CSSTS	SUREQCLR	—	—	SQCLR
		SQSET	SQMON	PBUSY	PINGE	—	CCPL	PID[1]	PID[0]
	PIPESEL	—	—	—	—	—	—	—	—
		—	—	—	—	PIPESEL[3:0]			
	PIPECFG	TYPE[1:0]		—	—	—	BFRE	DBLB	CNTMD
		SHTNAK	—	—	DIR	EPNUM[3:0]			
	PIPEBUF	—	BUFSIZE[4:0]					—	—
		BUFNMB[7:0]							

Module Name	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
USB	PIPEMAXP	DEVSEL[3:0]				—	MXPS[10:8]		
		MXPS[7:0]							
	PIPEPERI	—	—	—	IFIS	—	—	—	—
		—	—	—	—	—	IITV[2:0]		
	PIPE1CTR	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
	PIPE2CTR	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
	PIPE3CTR	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
	PIPE4CTR	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
	PIPE5CTR	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
	PIPE6CTR	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
	PIPE7CTR	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
	PIPE8CTR	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
	PIPE9CTR	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
	PIPE1TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	PIPE1TRN	TRNCNT[15:8]							
		TRNCNT[7:0]							
	PIPE2TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	PIPE2TRN	TRNCNT[15:8]							
		TRNCNT[7:0]							

Module Name	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
USB	PIPE3TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	PIPE3TRN	TRNCNT[15:8]							
		TRNCNT[7:0]							
	PIPE4TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	PIPE4TRN	TRNCNT[15:8]							
		TRNCNT[7:0]							
	PIPE5TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	PIPE5TRN	TRNCNT[15:8]							
		TRNCNT[7:0]							
	USBACSWR0	—	UACS14	—	—	—	—	—	—
		—	—	UACS5	—	—	—	—	—
	USBACSWR1	—	—	—	—	—	UACS26	—	—
		—	—	—	—	—	—	—	—
	DEVADD0	—	UPPHUB[3:0]				HUBPORT[2:0]		
		USBSPD[1:0]		—	—	—	—	—	RTPORT
	DEVADD1	—	UPPHUB[3:0]				HUBPORT[2:0]		
		USBSPD[1:0]		—	—	—	—	—	RTPORT
	DEVADD2	—	UPPHUB[3:0]				HUBPORT[2:0]		
		USBSPD[1:0]		—	—	—	—	—	RTPORT
	DEVADD3	—	UPPHUB[3:0]				HUBPORT[2:0]		
		USBSPD[1:0]		—	—	—	—	—	RTPORT
	DEVADD4	—	UPPHUB[3:0]				HUBPORT[2:0]		
		USBSPD[1:0]		—	—	—	—	—	RTPORT
	DEVADD5	—	UPPHUB[3:0]				HUBPORT[2:0]		
		USBSPD[1:0]		—	—	—	—	—	RTPORT
	DEVADD6	—	UPPHUB[3:0]				HUBPORT[2:0]		
		USBSPD[1:0]		—	—	—	—	—	RTPORT

Module Name	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
USB	DEVADD7	—	UPPHUB[3:0]				HUBPORT[2:0]		
		USBSPD[1:0]		—	—	—	—	—	RTPORT
	DEVADD8	—	UPPHUB[3:0]				HUBPORT[2:0]		
		USBSPD[1:0]		—	—	—	—	—	RTPORT
	DEVADD9	—	UPPHUB[3:0]				HUBPORT[2:0]		
		USBSPD[1:0]		—	—	—	—	—	RTPORT
	DEVADDA	—	UPPHUB[3:0]				HUBPORT[2:0]		
		USBSPD[1:0]		—	—	—	—	—	RTPORT
ATAPI	ATAPI_ CONTROL	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	DTCD	—
		RESET	M/S	—	UDMAEN	—	R/W	STOP	START
	ATAPI_STATUS	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	SWERR
		IFERR	—	DEVTRM	DEVINT	TOUT	ERR	NEND	ACT
	ATAPI_INT_ ENABLE	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	iSWERR
		iIFERR	—	iDEVTRM	iDEVINT	iTOUT	iERR	iNEND	iACT
	ATAPI_PIO_ TIMING	—	—	pSDCT					
		pSDPW					pSDST		
		—	—	pMDCT					
		pMDPW					pMDST		
	ATAPI_MULTI_ TIMING	—	—	—	—	—	mSDCT		
		mSDCT			mSDPW				
		—	—	—	—	—	mMDCT		
		mMDCT			mMDPW				

Module Name	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
ATAPI	ATAPI_ULTRA_TIMING	—	—	—	—	—	—	—	uSDCT
		uSDCT			uSDRP				
		—	—	—	—	—	—	—	uMDCT
		uMDCT			uMDRP				
	ATAPI_DMA_START_ADR	—	—	—	DSTA[28:24]				
		DSTA[23:16]							
		DSTA[15:8]							
		DSTA[7:2]						—	—
	ATAPI_DMA_TRANS_CNT	—	—	—	DTRC[28:24]				
		DTRC[23:16]							
		DTRC[15:8]							
		DTRC[7:1]							—
	ATAPI_CONTROL2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	WORDSWAP	IFEN
	ATAPI_SIG_ST	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	DDMARDY	DMARQ
	ATAPI_BYTE_SWAP	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	BYTESWAP
2DG	GR_BLTPLY	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	SB_STEN	SA_STEN

Module Name	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
2DG	GR_MIXPLY	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	EXTEN	—	—	—	OUTEN
	GR_DOSTAT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		DISP_STAT[1:0]		SEHF_STAT[1:0]		—	—	DCHF_STAT[1:0]	
		SBHF_STAT[1:0]		SAHF_STAT[1:0]		—	—	SB_REND	SA_REND
	GR_IRSTAT	—	—	—	—	—	—	—	IRQ_DEMPT
		—	IRQ_ASHFUL	IRQ_DHFUL	IRQ_SHFUL	—	—	—	—
		—	INT_VSYC	INT_UDFL	INT_FILD	—	—	—	INT_DEMPT
		—	INT_ASHFUL	INT_DHFUL	INT_SHFUL	—	—	—	INT_GR
	GR_INTMSK	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	MSK_VSYC	MSK_UDFL	MSK_FILD	—	—	—	MSK_DEMPT
		—	MSK_ASHFUL	MSK_DHFUL	MSK_SHFUL	—	—	—	MSK_GR
	GR_INTDIS	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	DIS_VSYC	DIS_UDFL	DIS_FILD	—	—	—	DIS_DEMPT
		—	DIS_ASHFUL	DIS_DHFUL	DIS_SHFUL	—	—	—	DIS_GR
	GR_DMAC	—	—	SZSEL2	SZSEL1	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	DM1_DSEL[1:0]		DM2_DSEL[1:0]		DM34_DSEL[1:0]	
		—	—	DM1_MSEL[1:0]		DM2_MSEL[1:0]		DM34_MSEL[1:0]	
	GR_SABSET	—	—	—	—	—	—	—	SSHIGH[8]
		SSHIGH[7:0]							
		—	—	—	—	—	—	—	SSWIDH[8]
		SSWIDH[7:0]							

Module Name	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
2DG	GR_DCSET	—	—	—	—	—	—	—	DCHIGH[8]
		DCHIGH[7:0]							
		—	—	—	—	—	—	—	DCWIDH[8]
		DCWIDH[7:0]							
	MGR_SESET	—	—	—	—	—	—	—	SEHIGH[8]
		SEHIGH[7:0]							
		—	—	—	—	—	—	—	SEWIDH[8]
		SEWIDH[7:0]							
	GR_PIXLFMT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	SE_FMT
		—	—	—	—	—	—	—	DC_FMT
		—	—	SB_FMT[1:0]		—	—	—	SA_FMT
	GR_BLTMODE	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	CRKEY[1:0]	
		—	—	LGTYPE[1:0]		SBSEL[1:0]		BTYPE[1:0]	
	GR_RISZSET	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	PREON	—	—	EDGE[1:0]	
		—	—	—	—	—	—	—	BRSIZ
	GR_RISZMOD	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	A1_H	—	H1_MTHD
		—	—	—	—	—	A1_V	—	V1_MTHD
	GR_DELT	—	—	VDLT_INTGR[1:0]		VDLT_DCML[11:8]			
		VDLT_DCML[7:0]							
		—	—	HDLT_INTGR[1:0]		HDLT_DCML[11:8]			
		HDLT_DCML[7:0]							

Module Name	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
2DG	GR_HSPHAS	—	—	—	—	H1PHS_DCML[11:8]			
		H1PHS_DCML[7:0]							
		—	—	—	—	—	—	H1PHS_INTGR[9:8]	
		H1PHS_INTGR[7:0]							
	GR_VSPHAS	—	—	—	—	V1PHS_DCML[11:8]			
		V1PHS_DCML[7:0]							
		—	—	—	—	—	—	—	V1PHS_ INTGR[8]
		V1PHS_INTGR[7:0]							
	MGR_HDELT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		MHDLT_INTGR[3:0]				MHDLT_DCML[11:8]			
		MHDLT_DCML[7:0]							
	MGR_HPHAS	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	MH1PHS_DCML[11:8]			
		MH1PHS_DCML[7:0]							
	GR_LGDAT	—	—	—	—	LGDAT_A[3:0]			
		—	—	—	LGDAT_R[4:0]				
		—	—	—	LGDAT_G[4:0]				
		—	—	—	LGDAT_B[4:0]				
	GR_DETCOL	—	—	—	—	—	—	—	—
		—	—	—	DETC_R[4:0]				
		—	—	—	DETC_G[4:0]				
		—	—	—	DETC_B[4:0]				
	GR_BRDCOL	—	—	—	—	BRDC_A[3:0]			
		—	—	—	BRDC_R[4:0]				
		—	—	—	BRDC_G[4:0]				
		—	—	—	BRDC_B[4:0]				

Module Name	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
2DG	GR_BRD1CNT	—	—	—	—	—	—	—	—
		—	—	—	—	AFTER_A[3:0]			
		—	—	—	—	—	—	FBFA[1:0]	
		—	—	—	GALFA	—	—	—	GCOLR
	MGR_MIXMODE	—	—	—	NTSC	—	—	—	—
		CHG_A[3:0]				—	FCFD[2:0]		
		—	—	—	MVON	—	—	—	CBCR
		—	—	—	—	—	—	—	VLD_N
	MGR_MIXHTMG	—	—	—	—	—	—	—	—
		—	—	WPH[5:0]					
		—	—	—	—	—	—	—	PDPH[8]
		PDPH[7:0]							
	MGR_MIXHS	—	—	—	—	—	—	ALLPH[9:8]	
		ALLPH[7:0]							
		—	—	—	—	—	—	VLDPH[9:8]	
		VLDPH[7:0]							
	MGR_MIXVTMG	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		WPV[3:0]				—	—	—	PDPV[8]
		PDPV[7:0]							
	MGR_MIXVS	—	—	—	—	—	—	—	ALLPV[8]
		ALLPV[7:0]							
		—	—	—	—	—	—	—	VLDPV[8]
		VLDPV[7:0]							
	GR_VSDLY	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	VSDLY[9:8]	
		VSDLY[7:0]							

Module Name	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
2DG	VDAC_TMG	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	edgesel
PFC	PAIORL	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR
		PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR
	PACRL4	PA15MD[3:0]				PA14MD[3:0]			
		PA13MD[3:0]				PA12MD[3:0]			
	PACRL3	PA11MD[3:0]				PA10MD[3:0]			
		PA9MD[3:0]				PA8MD[3:0]			
	PACRL2	PA7MD[3:0]				PA6MD[3:0]			
		PA5MD[3:0]				PA4MD[3:0]			
	PACRL1	PA3MD[3:0]				PA2MD[3:0]			
		PA1MD[3:0]				PA0MD[3:0]			
	PBIORH	—	—	—	—	—	—	—	—
		—	—	—	—	—	PB18IOR	PB17IOR	PB16IOR
	PBIORL	PB15IOR	PB14IOR	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IOR
		PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR
	PBCRH2	—	—	—	—	—	—	—	—
		—	—	—	—	PB18MD[3:0]			
	PBCRH1	—	—	—	—	—	—	—	—
		PB17MD[3:0]				PB16MD[3:0]			
	PBCRL4	PB15MD[3:0]				PB14MD[3:0]			
		PB13MD[3:0]				PB12MD[3:0]			
	PBCRL3	PB11MD[3:0]				PB10MD[3:0]			
		PB9MD[3:0]				PB8MD[3:0]			
	PBCRL2	PB7MD[3:0]				PB6MD[3:0]			
		PB5MD[3:0]				PB4MD[3:0]			
	PBCRL1	PB3MD[3:0]				PB2MD[3:0]			
		PB1MD[3:0]				PB0MD[3:0]			

Module Name	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
PFC	PCIORL	—	—	—	—	—	PC10IOR	PC9IOR	PC8IOR
		PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR
	PCCRL3	—	—	—	—	PC10MD[3:0]			
		PC9MD[3:0]				PC8MD[3:0]			
	PCCRL2	PC7MD[3:0]				PC6MD[3:0]			
		PC5MD[3:0]				PC4MD[3:0]			
	PCCRL1	PC3MD[3:0]				PC2MD[3:0]			
		PC1MD[3:0]				PC0MD[3:0]			
	PDIORL	—	—	—	—	—	—	—	—
		—	—	—	—	—	PD2IOR	PD1IOR	PD0IOR
	PDCRL1	—	—	—	—	PD2MD[3:0]			
		PD1MD[3:0]				PD0MD[3:0]			
	PEIORL	—	—	PE13IOR	—	PE11IOR	—	PE9IOR	—
		PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR
	PECRL4	—	—	—	—	—	—	—	—
		PE13MD[3:0]				PE12MD[3:0]			
	PECRL3	PE11MD[3:0]				PE10MD[3:0]			
		PE9MD[3:0]				PE8MD[3:0]			
	PECRL2	PE7MD[3:0]				PE6MD[3:0]			
		PE5MD[3:0]				PE4MD[3:0]			
	PECRL1	PE3MD[3:0]				PE2MD[3:0]			
		PE1MD[3:0]				PE0MD[3:0]			
	PFIORL	—	—	—	—	—	—	—	—
		—	—	—	PF4IOR	PF3IOR	PF2IOR	PF1IOR	PF0IOR
	PFCRL2	—	—	—	—	—	—	—	—
		—	—	—	—	PF4MD[3:0]			
	PFCRL1	PF3MD[3:0]				PF2MD[3:0]			
		PF1MD[3:0]				PF0MD[3:0]			
	PGCRL2	PG7MD[3:0]				PG6MD[3:0]			
		PG5MD[3:0]				PG4MD[3:0]			

Module Name	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
PFC	PGCRL1	PG3MD[3:0]				PG2MD[3:0]			
		PG1MD[3:0]				PG0MD[3:0]			
	PHIORL	PH15IOR	PH14IOR	PH13IOR	PH12IOR	PH11IOR	PH10IOR	PH9IOR	PH8IOR
		PH7IOR	PH6IOR	PH5IOR	PH4IOR	PH3IOR	PH2IOR	PH1IOR	PH0IOR
	PHCRL4	PH15MD[3:0]				PH14MD[3:0]			
		PH13MD[3:0]				PH12MD[3:0]			
	PHCRL3	PH11MD[3:0]				PH10MD[3:0]			
		PH9MD[3:0]				PH8MD[3:0]			
	PHCRL2	PH7MD[3:0]				PH6MD[3:0]			
		PH5MD[3:0]				PH4MD[3:0]			
	PHCRL1	PH3MD[3:0]				PH2MD[3:0]			
		PH1MD[3:0]				PH0MD[3:0]			
	PJIORL	—	—	—	PJ12IOR	PJ11IOR	PJ10IOR	PJ9IOR	PJ8IOR
		PJ7IOR	PJ6IOR	PJ5IOR	PJ4IOR	PJ3IOR	PJ2IOR	PJ1IOR	PJ0IOR
	PJCRL4	—	—	—	—	—	—	—	—
		—	—	—	—	PJ12MD[3:0]			
	PJCRL3	PJ11MD[3:0]				PJ10MD[3:0]			
		PJ9MD[3:0]				PJ8MD[3:0]			
	PJCRL2	PJ7MD[3:0]				PJ6MD[3:0]			
		PJ5MD[3:0]				PJ4MD[3:0]			
	PJCRL1	PJ3MD[3:0]				PJ2MD[3:0]			
		PJ1MD[3:0]				PJ0MD[3:0]			
	PKIORL	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	PK1IOR	PK0IOR
	PKCRL1	—	—	—	—	—	—	—	—
		PK1MD[3:0]				PK0MD[3:0]			
I/O Ports	PADRL	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR
		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
	PAPRL	PA15PR	PA14PR	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR
		PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR

Module Name	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
I/O Ports	PBDRH	—	—	—	—	—	—	—	—
		—	—	—	—	—	PB18DR	PB17DR	PB16DR
	PBDRL	PB15DR	PB14DR	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
	PBPRH	—	—	—	—	—	—	—	—
		—	—	—	—	—	PB18PR	PB17PR	PB16PR
	PBPRL	PB15PR	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR
		PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
	PCDRL	—	—	—	—	—	PC10DR	PC9DR	PC8DR
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
	PCPRL	—	—	—	—	—	PC10PR	PC9PR	PC8PR
		PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
	PDDRL	—	—	—	—	—	—	—	—
		—	—	—	—	—	PD2DR	PD1DR	PD0DR
	PDPRL	—	—	—	—	—	—	—	—
		—	—	—	—	—	PD2PR	PD1PR	PD0PR
	PEDRL	—	—	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
	PEPRL	—	—	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR
		PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR
	PFDRL	—	—	—	—	—	—	—	—
		—	—	—	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
	PFPRL	—	—	—	—	—	—	—	—
		—	—	—	PF4PR	PF3PR	PF2PR	PF1PR	PF0PR
	PGDRL	—	—	—	—	—	—	—	—
		PG7DR	PG6DR	PG5DR	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
	PHDRL	PH15DR	PH14DR	PH13DR	PH12DR	PH11DR	PH10DR	PH9DR	PH8DR
		PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR
	PHPRL	PH15PR	PH14PR	PH13PR	PH12PR	PH11PR	PH10PR	PH9PR	PH8PR
		PH7PR	PH6PR	PH5PR	PH4PR	PH3PR	PH2PR	PH1PR	PH0PR

Module Name	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
I/O Ports	PJDRL	—	—	—	PJ12DR	PJ11DR	PJ10DR	PJ9DR	PJ8DR
		PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR
	PJPRL	—	—	—	PJ12PR	PJ11PR	PJ10PR	PJ9PR	PJ8PR
		PJ7PR	PJ6PR	PJ5PR	PJ4PR	PJ3PR	PJ2PR	PJ1PR	PJ0PR
	PKDRL	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	PK1DR	PK0DR
	PKPRL	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	PK1PR	PK0PR
Power-Down Modes	STBCR1	STBY	DEEP	SLPERE	AXTALE	—	—	—	—
	STBCR2	MSTP27	—	—	MSTP24	MSTP23	MSTP22	MSTP21	—
	STBCR3	MSTP37	—	MSTP35	—	—	MSTP32	MSTP31	MSTP30
	STBCR4	MSTP47	MSTP46	MSTP45	MSTP44	MSTP43	MSTP42	—	—
	STBCR5	MSTP57	MSTP56	MSTP55	MSTP54	MSTP53	MSTP52	—	—
	STBCR6	MSTP67	MSTP66	MSTP65	MSTP64	MSTP63	MSTP62	—	—
	STBCR7	MSTP77	MSTP76	—	MSTP74	MSTP73	MSTP72	MSTP71	MSTP70
	SYSCR1	—	—	—	—	RAME3	RAME2	RAME1	RAME0
	SYSCR2	—	—	—	—	RAMWE3	RAMWE2	RAMWE1	RAMWE0
	SYSCR3	—	—	—	—	RAME3	RAME2	RAME1	RAME0
	SYSCR4	—	—	—	—	RAMWE3	RAMWE2	RAMWE1	RAMWE0
	SYSCR5	—	—	—	—	RAME3	RAME2	RAME1	RAME0
	SYSCR6	—	—	—	—	RAMWE3	RAMWE2	RAMWE1	RAMWE0
	SYSCR7	—	—	—	—	—	—	RAME1	RAME0
	SYSCR8	—	—	—	—	—	—	RAMWE1	RAMWE0
	SYSCR9	—	—	—	—	—	—	RAME1	RAME0
	SYSCR10	—	—	—	—	—	—	RAMWE1	RAMWE0
	SYSCR11	—	—	—	—	—	—	RAME1	RAME0
	SYSCR12	—	—	—	—	—	—	RAMWE1	RAMWE0
	SWRSTCR	—	—	SSIF5SRST	SSIF4SRST	SSIF3SRST	SSIF2SRST	SSIF1SRST	SSIF0SRST
	HIZCR	—	—	—	—	—	—	HIZ	HIZBSC
	C0MSR	—	—	—	—	—	—	—	SLEEP

Module Name	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
Power-Down Modes	C1MSR	—	—	—	—	—	—	—	SLEEP
	RRAMKP	—	—	—	—	RRAMKP3	RRAMKP2	RRAMKP1	RRAMKP0
	DSCTR	CS0KEEPE	RAMBOOT	—	—	—	—	—	—
	DSSSR	—	—	—	—	—	—	—	MRES
		IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
	DSFR	IOKEEP	—	—	—	—	—	MRESF	NMIF
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
H-UDI	SDIR	TI[7:0]							
		—	—	—	—	—	—	—	—

32.3 Register States in Each Operating Mode

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Deep Standby	Software Standby	Module Standby	Sleep
Multi-core processor	All registers	Initialized	Retained	Initialized	Retained	—	Retained
CPG	FRQCR0	Initialized*1	Retained	Initialized	Retained	—	Retained
	FRQCR1	Initialized*1	Retained	Initialized	Retained	—	Retained
INTC	IBNR	Initialized	Retained*2	Initialized	Retained	—	Retained
	Other than above	Initialized	Retained	Initialized	Retained	—	Retained
UBC	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
Cache	All registers	Initialized	Retained	Initialized	Retained	—	Retained
BSC	All registers	Initialized	Retained	Initialized	Retained	—	Retained
DMAC	All registers	Initialized	Retained	Initialized	Retained	—	Retained
MTU2	All registers	Initialized	Retained	Initialized	Retained	Initialized	Retained
CMT	All registers	Initialized	Retained	Initialized	Initialized	Retained	Retained
WDT	WTCSR0	Initialized	Initialized	Initialized	Retained	—	Retained
	WTCNT0	Initialized	Initialized	Initialized	Retained	—	Retained
	WRCSR0	Initialized*1	Retained	Initialized	Retained	—	Retained
	WTCSR1	Initialized	Initialized	Initialized	Retained	—	Retained
	WTCNT1	Initialized	Initialized	Initialized	Retained	—	Retained
	WRCSR1	Initialized*1	Retained	Initialized	Retained	—	Retained
RTC	R64CNT	Retained*3	Retained*3	Retained*3	Retained*3	Retained	Retained*3
	RSECCNT						
	RMINCNT						
	RHRCNT						
	RWKCNT						
	RDAYCNT						
	RMONCNT						
	RYRCNT						
	RSECAR	Initialized	Retained	Initialized	Retained	Retained	Retained
	RMINAR						

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Deep Standby	Software Standby	Module Standby	Sleep
RTC	RHRAR						
	RWKAR						
	RDAYAR						
	RMONAR						
	RYRAR						
	RCR1	Initialized	Initialized	Initialized	Retained	Retained	Retained
	RCR2	Initialized	Initialized*4	Initialized	Retained	Retained	Retained
	RCR3	Initialized	Retained	Initialized	Retained	Retained	Retained
SCIF	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
SSU	All registers	Initialized	Retained	Initialized	Initialized	Initialized	Retained
IIC3	ICMR_0 to 3	Initialized	Retained	Initialized	Retained*5	Retained*5	Retained
	Other than above	Initialized	Retained	Initialized	Retained	Retained	Retained
SSIF	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
RCAN-TL1	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
ADC	All registers	Initialized	Retained	Initialized	Initialized	Initialized	Retained
DAC	All registers	Initialized	Retained	Initialized	Retained	Initialized	Retained
FLCTL	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
USB	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
ATAPI	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
2DG	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
PFC	PBCRH2	Initialized*1	Retained	Initialized	Retained	—	Retained
	All registers	Initialized	Retained	Initialized	Retained	—	Retained
I/O Ports	All registers	Initialized*6	Retained	Initialized	Retained	—	Retained
Power-Down Modes	STBCR1	Initialized	Retained	Initialized	Retained	—	Retained
	STBCR2	Initialized	Retained	Initialized	Retained	—	Retained
	STBCR3	Initialized	Retained	Initialized	Retained	—	Retained
	STBCR4	Initialized	Retained	Initialized	Retained	—	Retained
	STBCR5	Initialized	Retained	Initialized	Retained	—	Retained
	STBCR6	Initialized	Retained	Initialized	Retained	—	Retained

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Deep Standby	Software Standby	Module Standby	Sleep
Power-Down Modes	STBCR7	Initialized	Retained	Initialized	Retained	—	Retained
	CSTBCR1	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR1	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR2	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR3	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR4	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR5	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR6	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR7	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR8	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR9	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR10	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR11	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR12	Initialized	Retained	Initialized	Retained	—	Retained
	SWRSTCR	Initialized	Retained	Initialized	Retained	—	Retained
	HIZCR	Initialized	Retained	Initialized	Retained	—	Retained
	C0MSR	Initialized	Initialized	Initialized	Initialized	—	Initialized
	C1MSR	Initialized	Initialized	Initialized	Initialized	—	Initialized
	RRAMKP	Initialized	Retained	Initialized	Retained	—	Retained
	DSCTR	Initialized	Retained	Initialized	Retained	—	Retained
	DSSSR	Initialized	Retained	Initialized	Retained	—	Retained
	DSFR	Initialized	Retained	Retained	Retained	—	Retained
H-UDI*7	SDIR	Retained	Retained	Initialized	Retained	Retained	Retained

Notes: 1. Retains the previous value after an internal power-on reset by means of the WDT.
2. The BN3 to BN0 bits are initialized.
3. Counting up continues.
4. Bits RTCEN and START are retained.
5. Bits BC2 to BC0 are initialized.
6. Since pin states are read out on the port G data register (PGDRL) and the port registers, values in these registers are neither retained nor initialized.
7. Initialized by $\overline{\text{TRST}}$ assertion or in the Test-Logic-Reset state of the TAP controller.

Section 33 Electrical Characteristics

33.1 Absolute Maximum Ratings

Table 33.1 Absolute Maximum Ratings

Item		Symbol	Value	Unit
Power supply voltage (I/O)		PV_{CC}	–0.3 to 4.6	V
Power supply voltage (Internal)		V_{CC}	–0.3 to 1.7	V
PLL power supply voltage		$PLLV_{CC}$	–0.3 to 1.7	V
Analog power supply voltage		AV_{CC}	–0.3 to 4.6	V
Analog reference voltage		AV_{ref}	–0.3 to AV_{CC} +0.3	V
USB transceiver analog power supply voltage (I/O)		$USBAPV_{CC}$	–0.3 to 4.6	V
USB transceiver analog power supply voltage (internal)		$USBAV_{CC}$	–0.3 to 1.7	V
USB transceiver digital power supply voltage (internal)		$USBDV_{CC}$	–0.3 to 1.7	V
2DG DAC analog power supply voltage 0		$2DGAPV_{CC}0$	–0.3 to 4.6	V
2DG DAC analog power supply voltage 1		$2DGAPV_{CC}1$	–0.3 to 4.6	V
Input voltage	Analog input pin	V_{AN}	–0.3 to AV_{CC} +0.3	V
	VBUS	V_{in}	–0.3 to 5.5	
	Other input pins	V_{in}	–0.3 to PV_{CC} +0.3	V
Operating temperature		T_{opr}	–20 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.

33.2 Power-on/Power-off Sequence

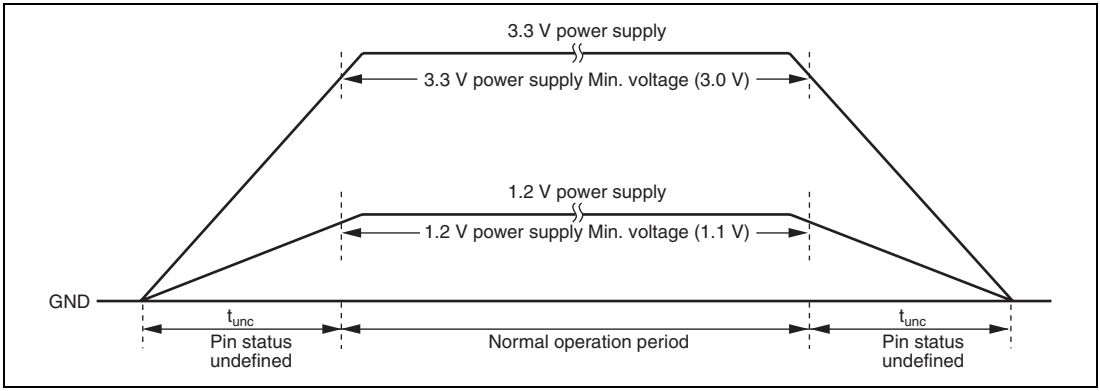


Figure 33.1 Power-on/Power-off Sequence

Table 33.2 Time for Power-on/Power-off Sequence

Item	Symbol	Min.	Max.	Unit
State undefined time	t_{unc}	—	100	ms

Note: It is recommended that the 1.2-V power supply (V_{CC}, $PLLV_{CC}$, $USBAV_{CC}$, and $USBDV_{CC}$) and the 3.3-V power supply (PV_{CC}, AV_{CC}, $USBAPV_{CC}$, $2DGAPV_{CC}0$, and $2DGAPV_{CC}1$) are turned on and off nearly simultaneously.

An indefinite period of time appears, from the time that power is turned on to the time that both of the 1.2-V power supply and the 3.3-V power supply rise to the Min. voltage (1.1 V for 1.2-V power supply and 3.0 V for 3.3-V power supply), or from the time that either of the 1.2-V power supply or the 3.3-V power supply is turned off and passes the Min. voltage (1.1 V for 1.2-V power supply and 3.0 V for 3.3-V power supply) to the time that both of the 1.2-V power supply and the 3.3-V power supply fall to GND.

During these periods, states of output pins and in-out pins and internal states become undefined. So it should be as short as possible. Also design the system so that these undefined states do not cause an overall malfunction.

33.3 DC Characteristics

Table 33.3 DC Characteristics (1) [Common Items]

Conditions: V_{CC} = $PLLV_{CC}$ = 1.1 to 1.3 V, $USBDV_{CC}$ = 1.1 to 1.3 V, $USBAV_{CC}$ = 1.1 to 1.3 V, PV_{CC} = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, $USBAPV_{CC}$ = 3.0 to 3.6 V, $2DGAPV_{CC}0$ = 3.0 to 3.6 V, $2DGAPV_{CC}1$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $USBAV_{SS}$ = AV_{SS} = $USBAPV_{SS}$ = $2DGAPV_{SS}0$ = $2DGAPV_{SS}1$ = 0 V, T_a = –20 to 85 °C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage			PV_{CC}	3.0	3.3	3.6	V	
			V_{CC}	1.1	1.2	1.3	V	
PLL power supply voltage			$PLLV_{CC}$	1.1	1.2	1.3	V	
Analog power supply voltage			AV_{CC}	3.0	3.3	3.6	V	
USB power supply voltage			$USBAPV_{CC}$	3.0	3.3	3.6	V	
			$USBAV_{CC}$ $USBDV_{CC}$	1.1	1.2	1.3	V	
2DG DAC power supply voltage			$2DGAPV_{CC}0$	3.0	3.3	3.6	V	
			$2DGAPV_{CC}1$	3.0	3.3	3.6	V	
Supply current*1	Normal operation		I_{CC}*2	—	380	500	mA	V_{CC} = 1.2 V $I_0\phi$ = 200.000 MHz $I_1\phi$ = 200.000 MHz $B\phi$ = 66.66 MHz $P\phi$ = 33.33 MHz
		Without operating USB high speed	PI_{CC}*3	—	110	—	mA	
		During USB high speed operation		—	210	—	mA	
	Dual sleep mode		I_{sleep}*2	—	270	450	mA	V_{CC} = 1.2 V $I_0\phi$ = 200.000 MHz $I_1\phi$ = 200.000 MHz $B\phi$ = 66.66 MHz $P\phi$ = 33.33 MHz
	Software standby mode		I_{sstby}*2	—	30	150	mA	T_a > 50°C V_{CC} = 1.2 V
				—	10	50	mA	$T_a \leq$ 50°C V_{CC} = 1.2 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Supply current*1	Deep standby mode	I_{dstby}*2	—	5	30	µA	$T_a > 50°C$ 1.2 V-power supply*4 = 1.2 V RAM: 0 Kbyte retained
			—	23	130	µA	$T_a > 50°C$ 1.2 V-power supply*4 = 1.2 V RAM: 4 Kbytes retained
			—	41	230	µA	$T_a > 50°C$ 1.2 V-power supply*4 = 1.2 V RAM: 8 Kbytes retained
			—	59	330	µA	$T_a > 50°C$ 1.2 V-power supply*4 = 1.2 V RAM: 12 Kbytes retained
			—	77	430	µA	$T_a > 50°C$ 1.2 V-power supply*4 = 1.2 V RAM: 16 Kbytes retained
			—	9	58	µA	$T_a > 50°C$ 3.3 V-power supply*5 = 3.3 V
			—	11	12	µA	Ta > 50°C VBUS = 5.0 V
			—	2	10	µA	$T_a \leq 50°C$ 1.2 V-power supply*4 = 1.2 V RAM: 0 Kbyte retained
			—	12	32	µA	$T_a \leq 50°C$ 1.2 V-power supply*4 = 1.2 V RAM: 4 Kbytes retained
			—	22	54	µA	$T_a \leq 50°C$ 1.2 V-power supply*4 = 1.2 V RAM: 8 Kbytes retained

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Supply current*1	Deep standby mode	I_{dstby}*2	—	32	76	µA	Ta ≤ 50°C 1.2 V-power supply*4 = 1.2 V RAM: 12 Kbytes retained
			—	42	98	µA	Ta ≤ 50°C 1.2 V-power supply*4 = 1.2 V RAM: 16 Kbytes retained
			—	5	26	µA	Ta ≤ 50°C 3.3 V-power supply*5 = 3.3 V
			—	11	12	µA	Ta ≤ 50°C VBUS = 5.0 V
Input leakage current	All input pins	$\|I_{in}\|$	—	—	1.0	µA	V_{in} = 0.5 to PV_{CC} – 0.5 V
Three-state leakage current	All input/output pins, all output pins (except PE13 to PE8, PF1, PF0, and pins with weak keeper) (off state)	$\|I_{STI}\|$	—	—	1.0	µA	V_{in} = 0.5 to PV_{CC} – 0.5 V
	PE13 to PE8, PF1, PF0		—	—	10	µA	
Input capacitance	All pins	C_{in}	—	—	20	pF	
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	0.7	4	mA	
	Waiting for A/D and D/A conversion		—	0.8	3	µA	
Analog reference voltage current	During A/D and D/A conversion	A_{iref}	—	1.2	4	mA	
	Waiting for A/D and D/A conversion		—	0.5	2	mA	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
USB power supply current	$USBAV_{CC}$ + $USBDV_{CC}$	I_{USBCC}	—	15	20	mA	$USBAV_{CC}$ = $USBDV_{CC}$ = 1.2 V
	$USBAPV_{CC}$	I_{USBPCC}	—	3	7	mA	$USBAPV_{CC}$ = 3.3 V
2DG DAC power supply current	$2DGAPV_{CC}0$ + $2DGAPV_{CC}1$	I_{2DGPCC}	—	20	—	mA	$2DGAPV_{CC}0$ + $2DGAPV_{CC}1$ = 3.3 V RL = 180Ω

Caution: When the A/D converter or D/A converter is not in use, the AV_{CC} and AV_{SS} pins should not be open.

Notes:
1. The supply current values are when all output pins and pins with the pull-up function are unloaded.
2. I_{CC}, I_{sleep}, I_{sstby}, and I_{dstby} represent the total currents supplied in the V_{CC} and $PLLV_{CC}$ systems.
3. PI_{CC} is the current through PV_{CC} (reference value) when there is no load on any output pin and the input pins are fixed. Since actual currents in operation are strongly dependent on the system (waveforms determined by IO, frequency of toggling, and so on), be sure to measure actual values for systems.
4. I_{dstby} for the 1.2-V power supply is the total current drawn through V_{CC}, PLV_{CC}, $USBAV_{CC}$, and $USBDV_{CC}$.
5. I_{dstby} for the 3.3-V power supply is the total current drawn through PV_{CC}, AV_{CC}, $USBAPV_{CC}$, $2DGAPV_{CC}0$, and $2DGAPV_{CC}1$.

Table 33.3 DC Characteristics (2) [Except I²C and USB-Related Pins]

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	$\overline{RES}$, $\overline{MRES}$, NMI, MD, MD_CLK1, MD_CLK0, $\overline{ASEMD}$, $\overline{TRST}$, EXTAL, CKIO, AUDIO_X1, RTC_X1	V_{IH}	PV_{CC} – 0.5	—	PV_{CC} + 0.3	V	
	PG7 to PG0		2.2	—	AV_{CC} + 0.3	V	
	Input pins other than above (except Schmitt pins)		2.2	—	PV_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{MRES}$, NMI, MD, MD_CLK1, MD_CLK0, $\overline{ASEMD}$, $\overline{TRST}$, EXTAL, CKIO, AUDIO_X1, RTC_X1	V_{IL}	–0.3	—	0.5	V	
	Input pins other than above (except Schmitt pins)		–0.3	—	0.8	V	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input characteristics	IRQ7 to IRQ0, PINT7 to PINT0, DREQ3 to DREQ0, TIOC0A to TIOC0D, TIOC1A, TIOC1B, TIOC2A, TIOC2B, TIOC3A to TIOC3D, TIOC4A to TIOC4D, TCLKA to TCLKD, SCK5, SCK2, SCK1, SCK0, RxD5 to RxD0, $\overline{\text{CTS0}}$, $\overline{\text{RTS0}}$, SSCK1, SSCK0, SSI1, SSI0, $\overline{\text{SSO1}}$, $\overline{\text{SSO0}}$, $\overline{\text{SCS1}}$, $\overline{\text{SCS0}}$, SSIDATA5 to SSIDATA0, SSISCK5 to SSISCK0, SSIWS5 to SSIWS0, AUDIO_CLK, CRx1, CRx0, $\overline{\text{ADTRG}}$, FRB, NAF7 to NAF0, PB2, PB8, PC15 to PC0, PD2 to PD0, PE7 to PE0, PG3 to PG0, PJ3 to PJ0, PH15 to PH0	V_T^+	PV_{CC} – 0.5	—	—	V	
		V_T^-	—	—	0.5	V	
		$V_T^+ - V_T^-$	0.2	—	—	V	
Output high voltage		V_{OH}	PV_{CC} – 0.5	—	—	V	I_{OH} = –2 mA
Output low voltage		V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RAM standby voltage	Software standby mode (high-speed on-chip RAM or on-chip RAM for data retention)	V_{RAMS}	0.75	—	—	V	Measured with V_{CC} (= $PLLV_{CC}$) as parameter
	Deep standby mode (only for on-chip RAM for data retention)	V_{RAMD}	1.1	—	—	V	

Table 33.3 DC Characteristics (3) [I^2C-Related Pins*]

Conditions: V_{CC} = $PLLV_{CC}$ = 1.1 to 1.3 V, $USBDV_{CC}$ = 1.1 to 1.3 V, $USBAV_{CC}$ = 1.1 to 1.3 V, PV_{CC} = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, $USBAPV_{CC}$ = 3.0 to 3.6 V, $2DGAPV_{CC}0$ = 3.0 to 3.6 V, $2DGAPV_{CC}1$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $USBAV_{SS}$ = AV_{SS} = $USBAPV_{SS}$ = $2DGAPV_{SS}0$ = $2DGAPV_{SS}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	V_{IH}	$PV_{CC} \times 0.7$	—	PV_{CC} + 0.3	V	
Input low voltage	V_{IL}	–0.3	—	$PV_{CC} \times 0.3$	V	
Schmitt trigger input characteristics	$V_{IH} - V_{IL}$	$PV_{CC} \times 0.05$	—	—	V	
Output low voltage	V_{OL}	—	—	0.4	V	I_{OL} = 3.0 mA

Note: * The PE13/TxD4/SDA2 to PE8/RxD2/SCL0, PF1/SCL3/CRx0, and PF0/SDA3 pins are open-drain pins.

Table 33.3 DC Characteristics (4) [USB-Related Pins*]

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reference resistance	R_{REF}	5.6kΩ ± 1%	5.6kΩ ± 1%	5.6kΩ ± 1%		
Input high voltage (VBUS)	V_{IH}	4.02	—	5.25	V	
Input low voltage (VBUS)	V_{IL}	–0.3	—	0.5	V	
Input high voltage (USB_X1)	V_{IH}	PV_{cc} – 0.5	—	PV_{cc} + 0.3	V	
Input low voltage (USB_X1)	V_{IL}	–0.3	—	0.5	V	

Note: * REFRIN, VBUS, USB_X1, and USB_X2 pins

Table 33.3 DC Characteristics (5) [USB-Related Pins* (Low-Speed, Full-Speed, and High-Speed Common Items)]

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
DP pull-up resistance (when function is selected)	R_{pu}	0.900	—	1.575	kΩ	In idle mode
		1.425	—	3.090	kΩ	In transmit/ receive mode
DP and DM pull-down resistance (when host is selected)	R_{pd}	14.25	—	24.80	kΩ	

Note: * DP1, DP0, DM1 and DM0 pins

Table 33.3 DC Characteristics (6) [USB-Related Pins* (Low-Speed/Full-Speed)]

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	V_{IH}	2.0	—	—	V	
Input low voltage	V_{IL}	—	—	0.8	V	
Differential input sensitivity	V_{DI}	0.2	—	—	V	| (DP) – (DM) |
Differential common mode range	V_{CM}	0.8	—	2.5	V	
Output high voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 μA
Output low voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
Output signal crossover voltage	V_{CRS}	1.3	—	2.0	V	C_L = 50 pF (full-speed) C_L = 200 pF to 600 pF (low-speed)

Note: * DP1, DP0, DM1 and DM0 pins

Table 33.3 DC Characteristics (7) [USB-Related Pins* (High-Speed)]

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V,
PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V,
$2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V,
V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V,
T_a = –20 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Squelch detection threshold voltage (differential voltage)	V_{HSSQ}	100	—	150	mV	
Common mode voltage range	V_{HSCM}	–50	—	500	mV	
Idle state	V_{HSOI}	–10.0	—	10.0	mV	
Output high voltage	V_{HSOH}	360	—	440	mV	
Output low voltage	V_{HSOL}	–10.0	—	10.0	mV	
Chirp J output voltage (difference)	V_{CHIRPJ}	700	—	1100	mV	
Chirp K output voltage (difference)	V_{CHIRPK}	–900	—	–500	mV	

Note: * DP1, DP0, DM1 and DM0 pins

Table 33.4 Permissible Output Currents

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V,
PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V,
$2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V,
V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V,
T_a = –20 to 85 °C

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	PE13 to PE8, PF1, PF0	I_{OL}	—	—	10	mA
	Output pins other than above				2	mA
Permissible output low current (total)		ΣI_{OL}	—	—	150	mA
Permissible output high current (per pin)		$-I_{OH}$	—	—	2	mA
Permissible output high current (total)		$\Sigma -I_{OH}$	—	—	50	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in table 33.4.

33.4 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 33.5 Operating Frequency

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = −20 to 85 °C

Item		Symbol	Min.	Max.	Unit	Remarks
Operating frequency	CPU0 clock($I_0\phi$)	f	40.00	200.00	MHz	
	CPU1 clock($I_1\phi$)		40.00	200.00	MHz	
	Bus clock(Bϕ)		40.00	66.66	MHz	
	Peripheral clock (Pϕ)		10.00	33.33	MHz	

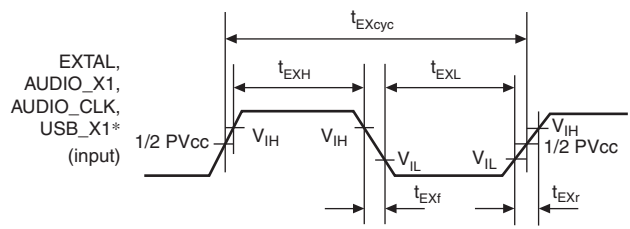
33.4.1 Clock Timing

Table 33.6 Clock Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	f_{EX}	10.00	33.33	MHz	Figure 33.2
EXTAL clock input cycle time	t_{EXcyc}	30	100	ns	
AUDIO_X1, AUDIO_CLK clock input frequency	f_{EX}	10	40	MHz	
AUDIO_X1, AUDIO_CLK clock input cycle time	t_{EXcyc}	25	100	ns	
USB_X1 clock input frequency (when high-speed transfer function is used)	f_{EX}	48 MHz ±100 ppm			
USB_X1 clock input frequency (when high-speed transfer function is not used and host controller function is used)		48 MHz ±500 ppm			
USB_X1 clock input frequency (when neither high-speed transfer function nor host controller function is used)		48 MHz ±2500 ppm			
EXTAL, AUDIO_X1, AUDIO_CLK, USB_X1 clock input low pulse width	t_{EXL}	0.4	0.6	t_{EXcyc}	
EXTAL, AUDIO_X1, AUDIO_CLK, USB_X1 clock input high pulse width	t_{EXH}	0.4	0.6	t_{EXcyc}	
EXTAL, AUDIO_X1, AUDIO_CLK, USB_ X1 clock input rise time	t_{EXr}	—	4	ns	
EXTAL, AUDIO_X1, AUDIO_CLK, USB_ X1 clock input fall time	t_{EXf}	—	4	ns	
CKIO clock input frequency	f_{CK}	40.00	66.66	MHz	Figure 33.3
CKIO clock input cycle time	t_{CKIcyc}	15	25	ns	
CKIO clock input low pulse width	t_{CKIL}	0.4	0.6	t_{CKIcyc}	
CKIO clock input high pulse width	t_{CKIH}	0.4	0.6	t_{CKIcyc}	
CKIO clock input rise time	t_{CKIr}	—	3	ns	
CKIO clock input fall time	t_{CKIf}	—	3	ns	

Item	Symbol	Min.	Max.	Unit	Figure
CKIO clock output frequency	f_{OP}	40.00	66.66	MHz	Figure 33.4
CKIO clock output cycle time	t_{cyc}	15	25	ns	
CKIO clock output low pulse width	t_{CKOL}	$t_{cyc}/2 - t_{CKOr}$	—	ns	
CKIO clock output high pulse width	t_{CKOH}	$t_{cyc}/2 - t_{CKOf}$	—	ns	
CKIO clock output rise time	t_{CKOr}	—	3	ns	
CKIO clock output fall time	t_{CKOf}	—	3	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	Figure 33.5
Oscillation settling time 1 on return from standby	t_{OSC2}	10	—	ms	Figure 33.6
Oscillation settling time 2 on return from standby	t_{OSC3}	10	—	ms	Figure 33.7
RTC clock oscillation settling time	t_{ROSC}	3	—	s	Figure 33.8



Note: * When the clock is input on the EXTAL, AUDIO_X1, AUDIO_CLK, or USB_X1 pin.

Figure 33.2 EXTAL, AUDIO_X1, AUDIO_CLK, and USB_X1 Clock Input Timing

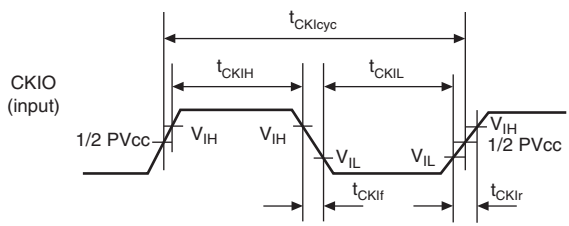


Figure 33.3 CKIO Clock Input Timing

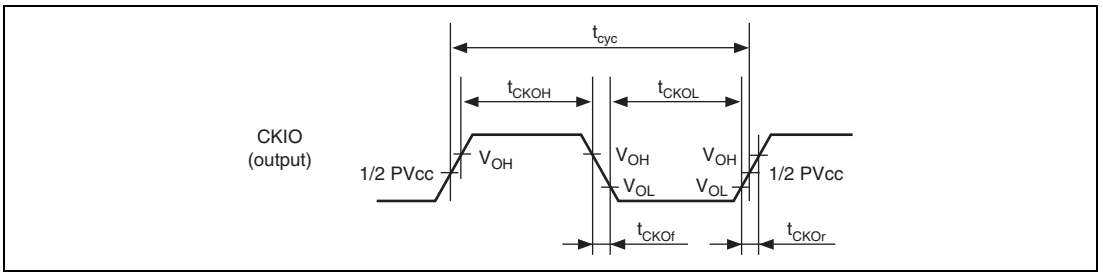


Figure 33.4 CKIO Clock Output Timing

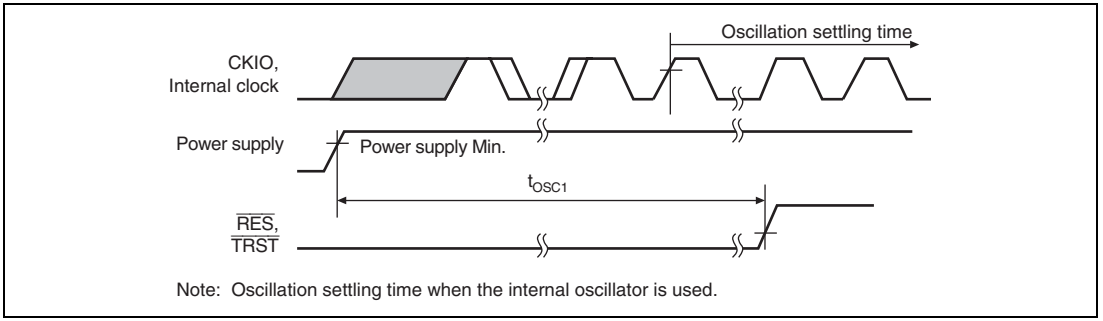


Figure 33.5 Power-On Oscillation Settling Time

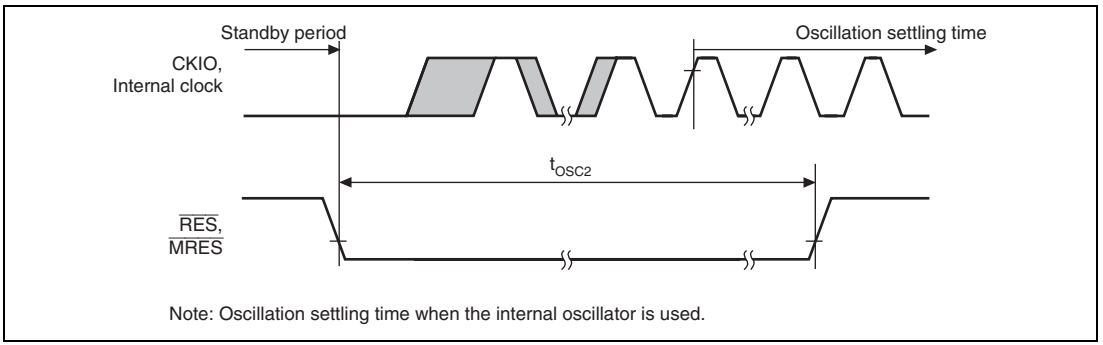
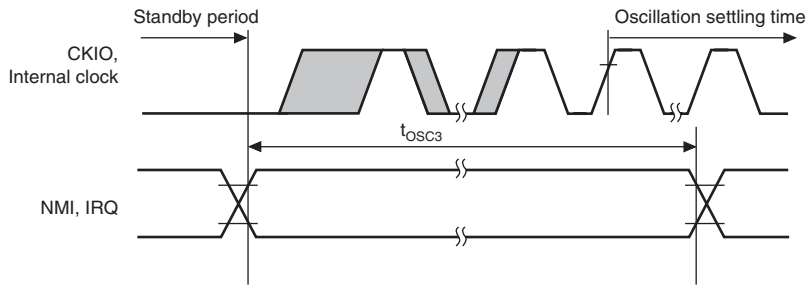


Figure 33.6 Oscillation Settling Time on Return from Standby (Return by Reset)



Note: Oscillation settling time when the internal oscillator is used.

Figure 33.7 Oscillation Settling Time on Return from Standby (Return by NMI or IRQ)

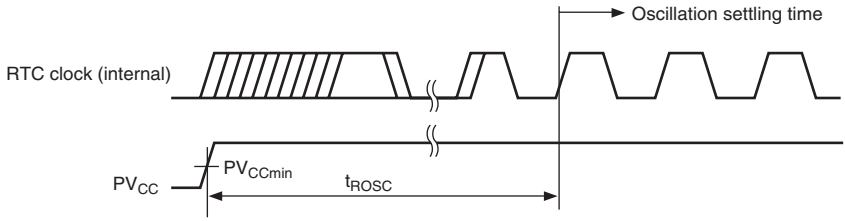


Figure 33.8 RTC Clock Oscillation Settling Time

33.4.2 Control Signal Timing

Table 33.7 Control Signal Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = −20 to 85 °C

			Bϕ = 66.66 MHz			
Item		**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
$\overline{RES}$ pulse width	Exit from standby mode or change the multiplication ratio of the PLL circuit	t_{RESW}	10	—	ms	Figure 33.9
	Other than above		20	—	t_{cyc}	
$\overline{MRES}$ pulse width	Exit from standby mode	t_{MRESW}	10	—	ms	
	Other than above		20	—	t_{cyc}	
$\overline{TRST}$ pulse width		t_{TRSW}	20	—	t_{cyc}	
NMI pulse width	Exit from standby mode	t_{NMIW}	10	—	ms	Figure 33.10
	Other than above		20	—	t_{cyc}	
IRQ pulse width	Exit from standby mode	t_{IRQW}	10	—	ms	
	Other than above		20	—	t_{cyc}	
PINT pulse width		t_{PINTW}	20	—	t_{cyc}	

Figure 33.9 Reset Input Timing

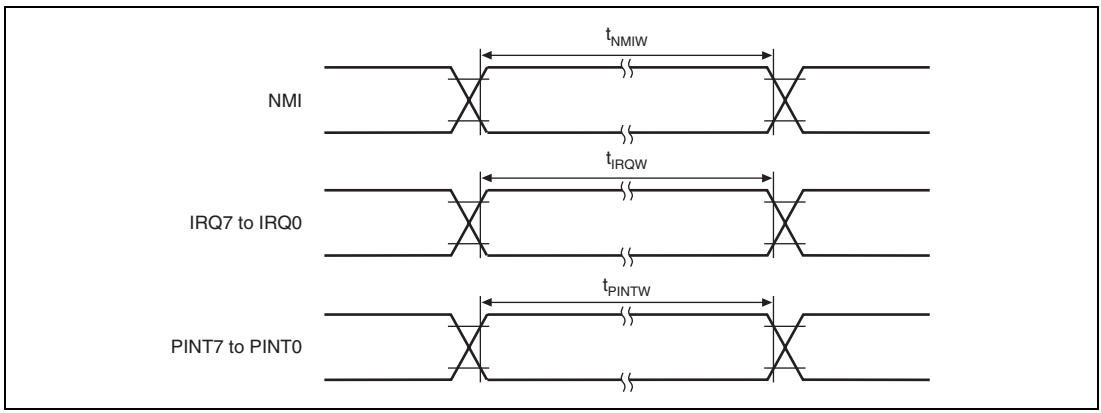


Figure 33.10 Interrupt Signal Input Timing

33.4.3 Bus Timing

Table 33.8 Bus Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = −20 to 85 °C

		Bϕ = 66.66 MHz*			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
Address delay time 1 (external space)	t_{AD1}	1	13	ns	Figures 33.11 to 33.15
Address delay time 2 (SDRAM space)	t_{AD2}	1	13	ns	Figures 33.16 to 33.22
Byte control delay time	t_{BCD}	—	13	ns	Figures 33.11 to 33.15
Chip select delay time 1 (external space)	t_{CSD1}	1	13	ns	Figures 33.11 to 33.15
Chip select delay time 2 (SDRAM space)	t_{CSD2}	1	13	ns	Figures 33.16 to 33.22
Read strobe delay time	t_{RSD}	—	13	ns	Figures 33.11 to 33.13, 33.15
Read data setup time 1 (external space)	t_{RDS1}	7	—	ns	Figures 33.11 to 33.13, 33.15
Read data setup time 2 (SDRAM space)	t_{RDS2}	7	—	ns	Figures 33.16, 33.18, 33.20
Read data hold time 1 (external space)	t_{RDH1}	2	—	ns	Figures 33.11 to 33.13, 33.15
Read data hold time 2 (SDRAM space)	t_{RDH2}	2	—	ns	Figures 33.16, 33.18, 33.20
Read data hold time 3 (external space)	t_{RDH3}	0	—	ns	Figures 33.11 to 33.13, 33.15
Read/write mode delay time	t_{RWM}	1	13	ns	Figures 33.11 to 33.15
Write enable delay time 1 (external space)	t_{WED1}	—	13	ns	Figures 33.11, 33.14
Write enable delay time 2 (SDRAM space)	t_{WED2}	1	13	ns	Figures 33.17, 33.19, 33.21

Item	Symbol	Bϕ = 66.66 MHz* Min.	Max.	Unit	Figure
Write data delay time 1 (external space)	t_{WDD1}	—	13	ns	Figures 33.11, 33.14
Write data delay time 2 (SDRAM space)	t_{WDD2}	—	13	ns	Figures 33.17, 33.19
Write data hold time 1 (external space)	t_{WDH1}	1	—	ns	Figures 33.11, 33.14
Write data hold time 2 (SDRAM space)	t_{WDH2}	1	—	ns	Figures 33.17, 33.19
External wait setup time	t_{WTS}	7	—	ns	Figure 33.15
External wait hold time	t_{WTH}	2	—	ns	Figure 33.15
$\overline{RAS}$ delay time	t_{RASD}	1	13	ns	Figures 33.16 to 33.22
$\overline{CAS}$ delay time	t_{CASD}	1	13	ns	Figures 33.16 to 33.22
DQM delay time	t_{DQMD}	1	13	ns	Figures 33.16 to 33.22
CKE delay time	t_{CKED}	1	13	ns	Figure 33.22

Note: * The maximum value (f_{max}) of Bϕ (bus clock) depends on the number of wait cycles and the system configuration of your board.

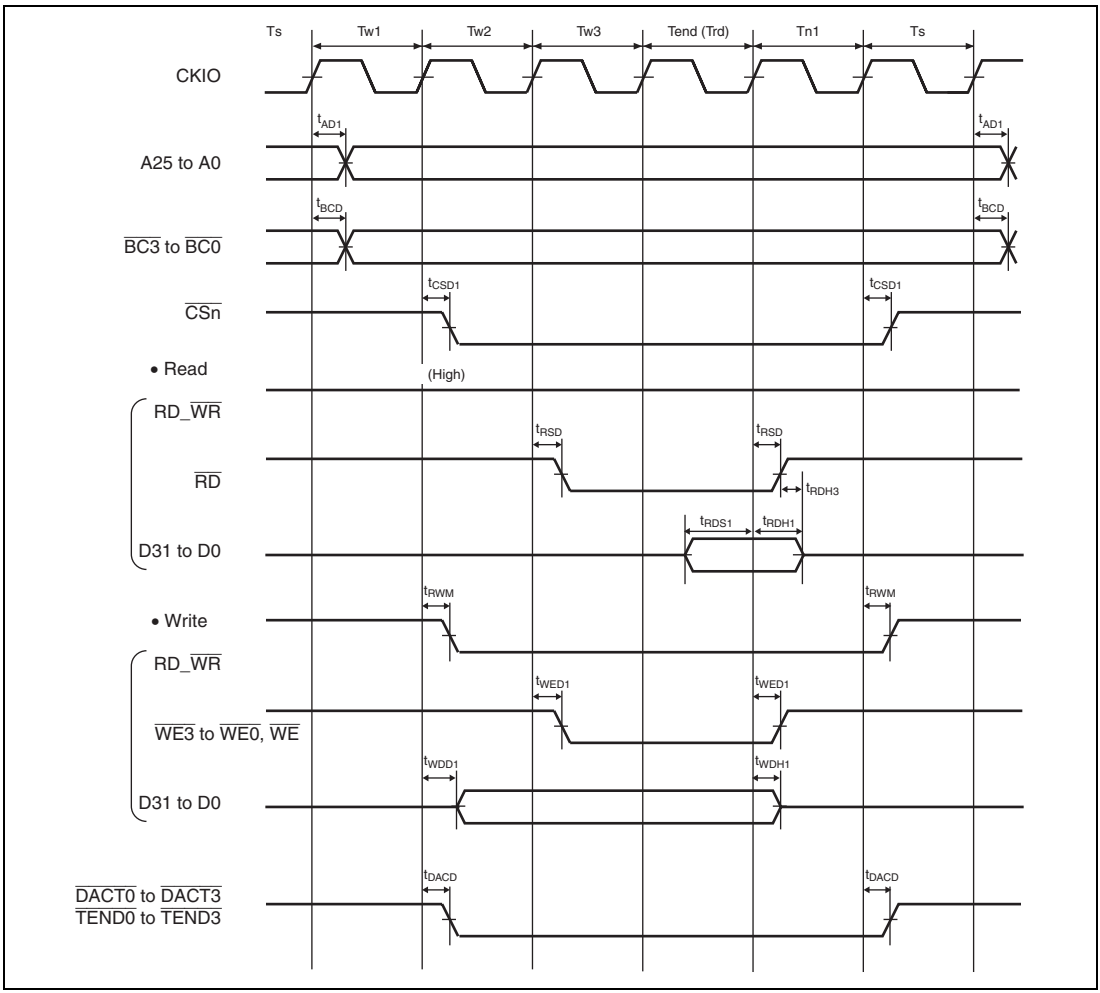


Figure 33.11 (1) External Address Space: Basic Bus Timing (Normal Access, Read/Write Cycle Wait = 3, CS Assert Wait = 1, Write Data Output Wait = 1, WR/RD Assert Wait = 2, Write Data Output Delay Cycles = 0, Read/Write CS Delay Cycles = 1)

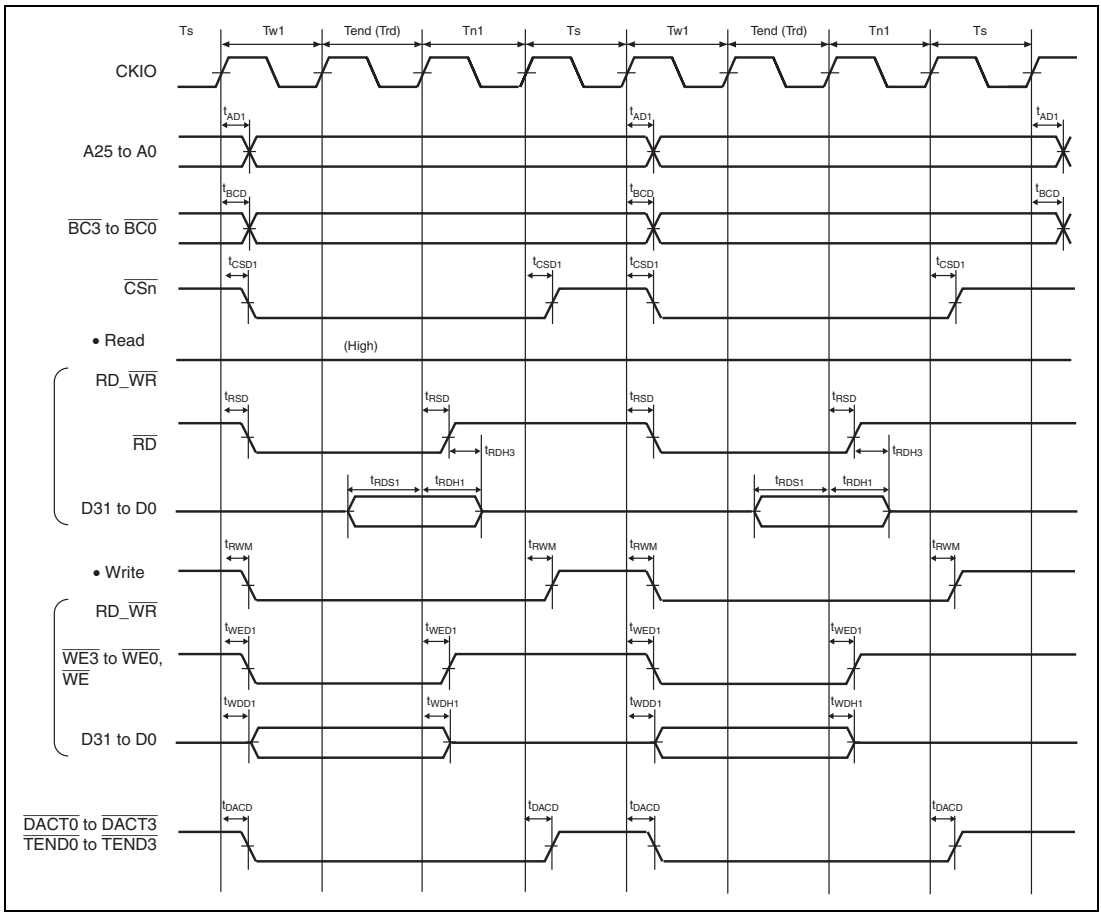


Figure 33.11 (2) External Address Space: Basic Bus Timing (Normal Access, Data Recovery Cycles = 0, Read/Write Cycle Wait = 1, Read/Write CS Delay Cycles = 1, Other Wait Settings = 0)

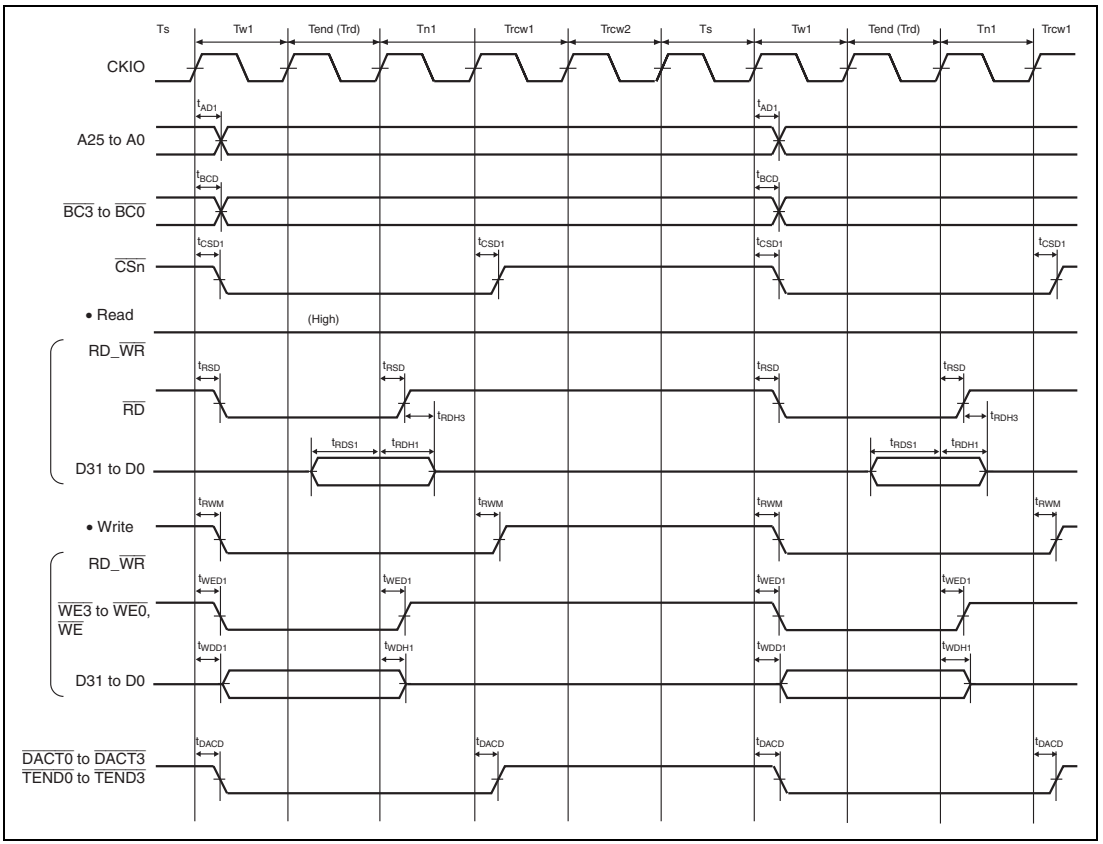


Figure 33.11 (3) External Address Space: Basic Bus Timing (Normal Access, Data Recovery Cycles = 2, Read/Write Cycle Wait = 1, Read/Write CS Delay Cycles = 1, Other Wait Settings = 0)

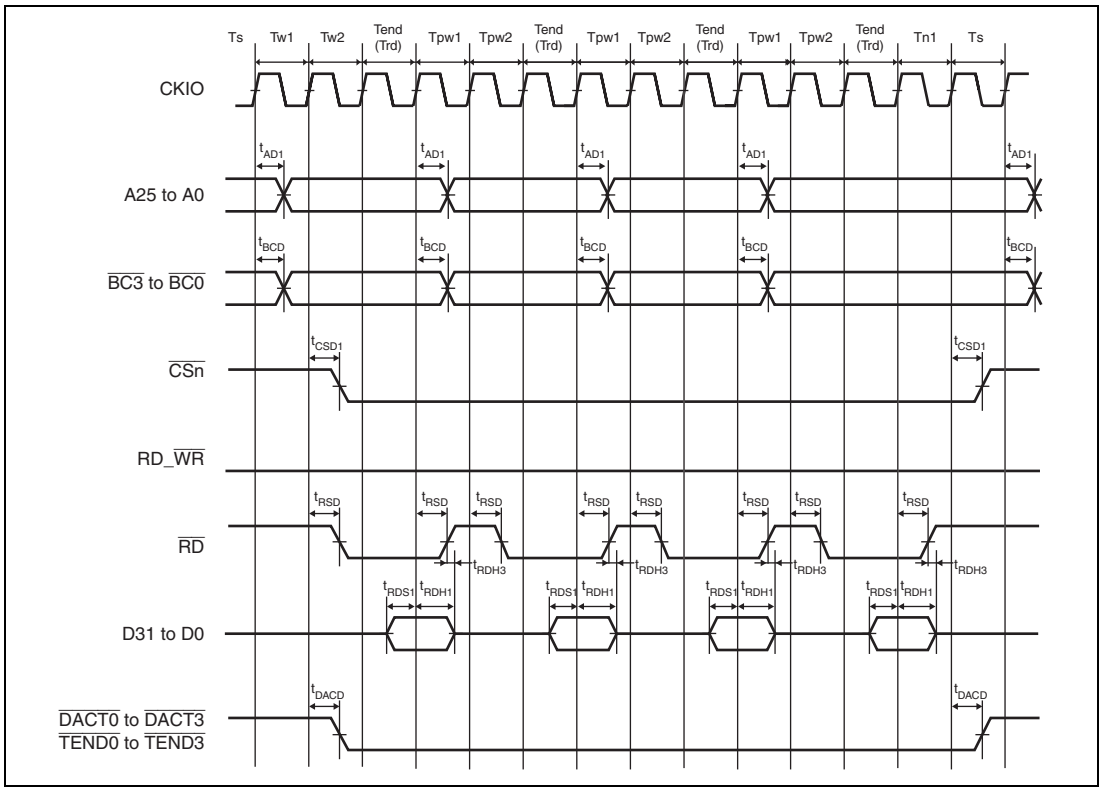


Figure 33.12 External Address Space: Basic Bus Timing
(Page Read Access, Normal Access Compatible Mode , Read Cycle Wait = 2, Page Read Cycle Wait = 2, CS Assert Wait = 1, RD Assert Wait = 1, Read CS Delay Cycles = 1)

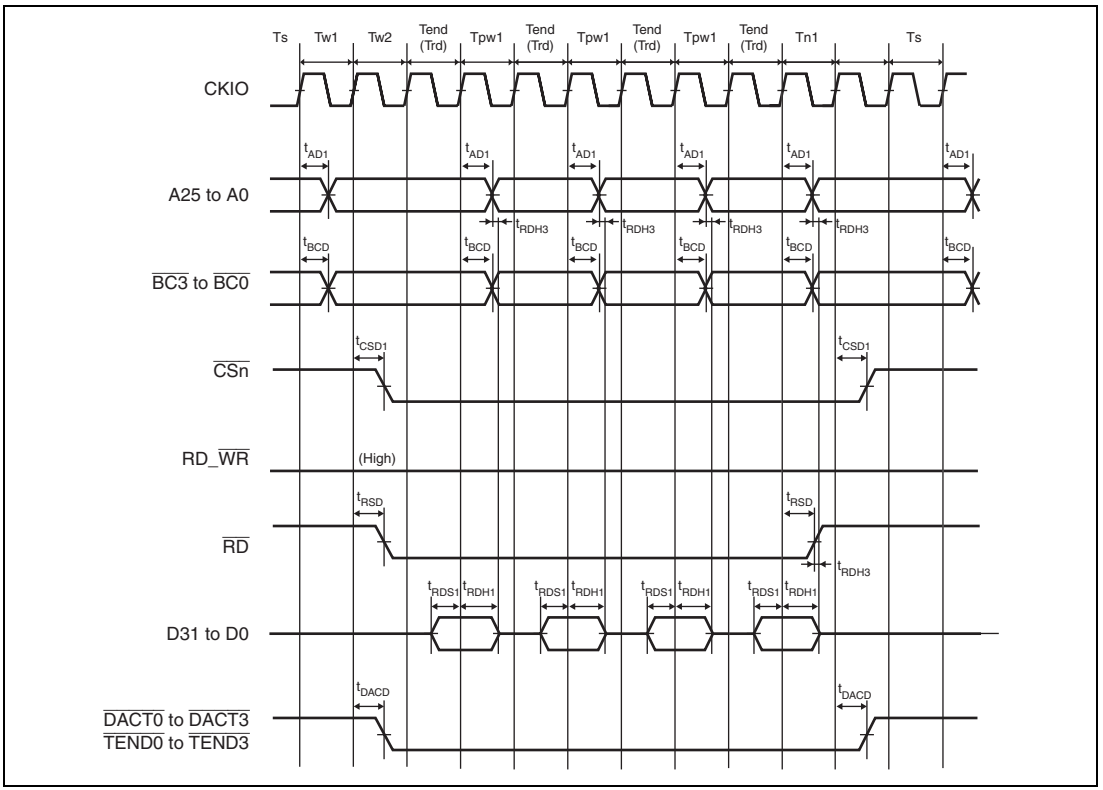


Figure 33.13 External Address Space: Basic Bus Timing (Page Read Access, External Read Data Continuous Assert Mode, Read Cycle Wait = 2, Page Read Cycle Wait = 1, CS Assert Wait = 1, RD Assert Wait = 1, Read CS Delay Cycles = 1)

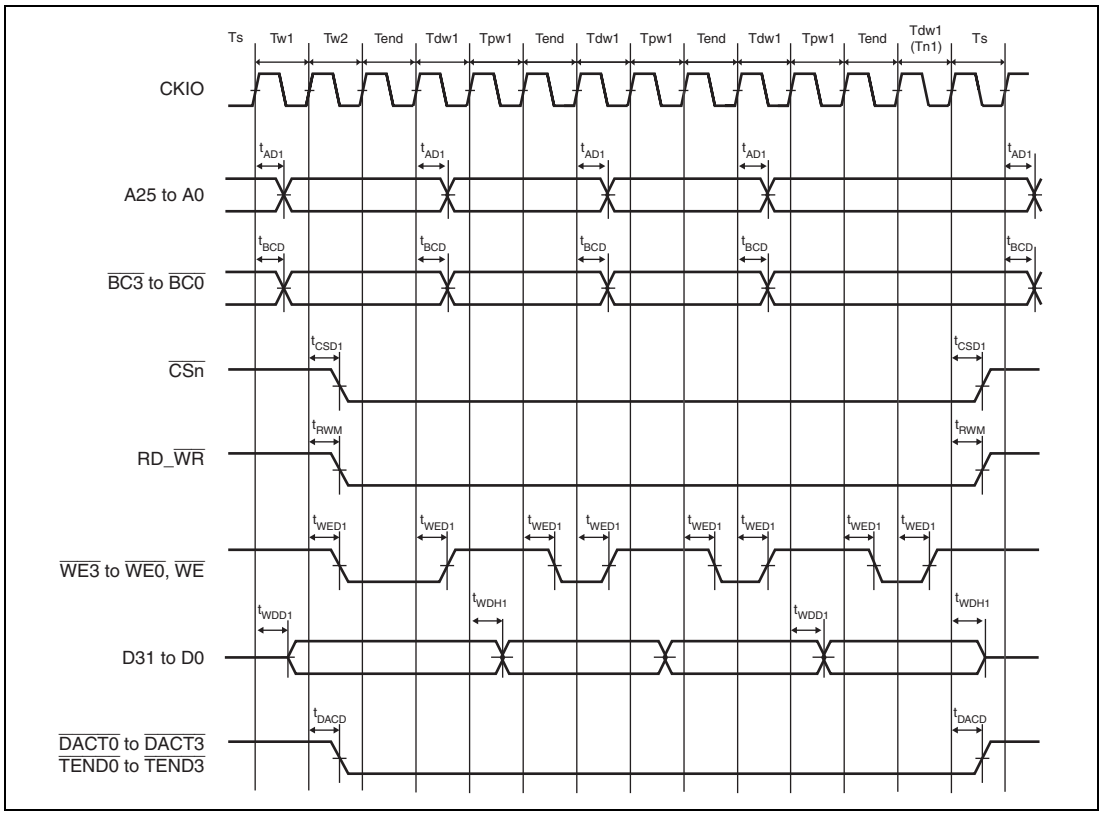


Figure 33.14 External Address Space: Basic Bus Timing (Page Write Access, Write Cycle Wait = 2, CS Assert Wait = 1, WR Assert Wait = 1, Write Data Output Delay Cycles = 1, Other Wait Settings = 0)

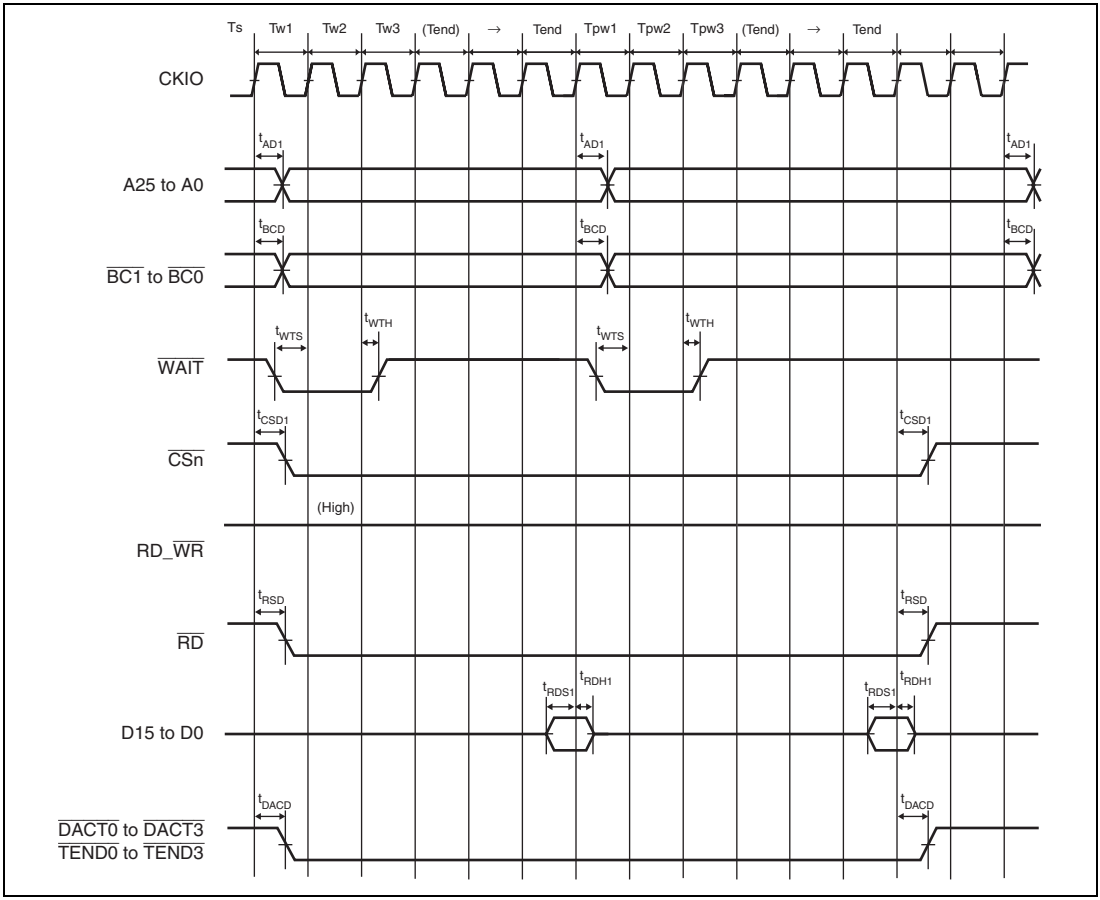


Figure 33.15 External Address Space: Timing with External Wait (Page Read Access to 16-Bit Width Channel, External Read Data Continuous Assert Mode, Read Cycle Wait = 3, Page Read Cycle Wait = 3, Other Wait Settings = 0, External Wait Cycles = 2)

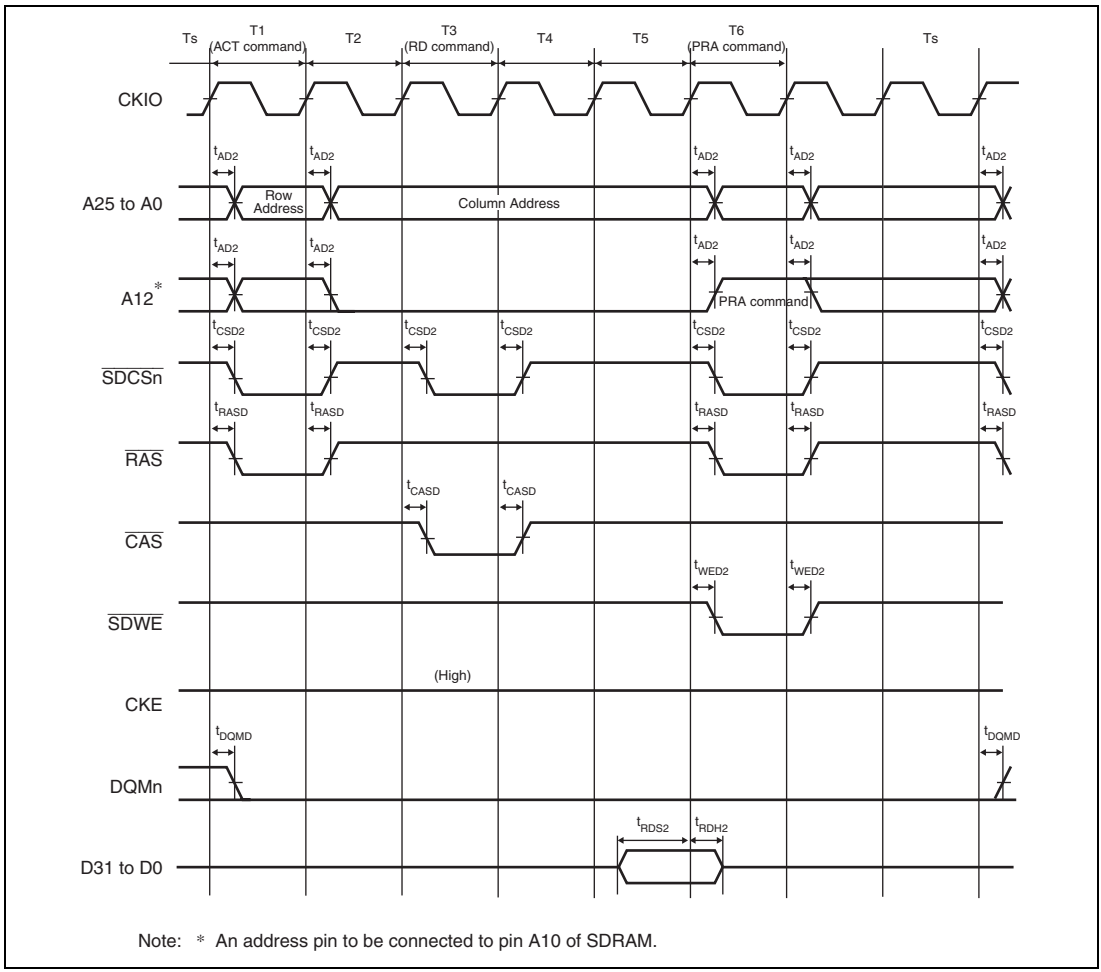
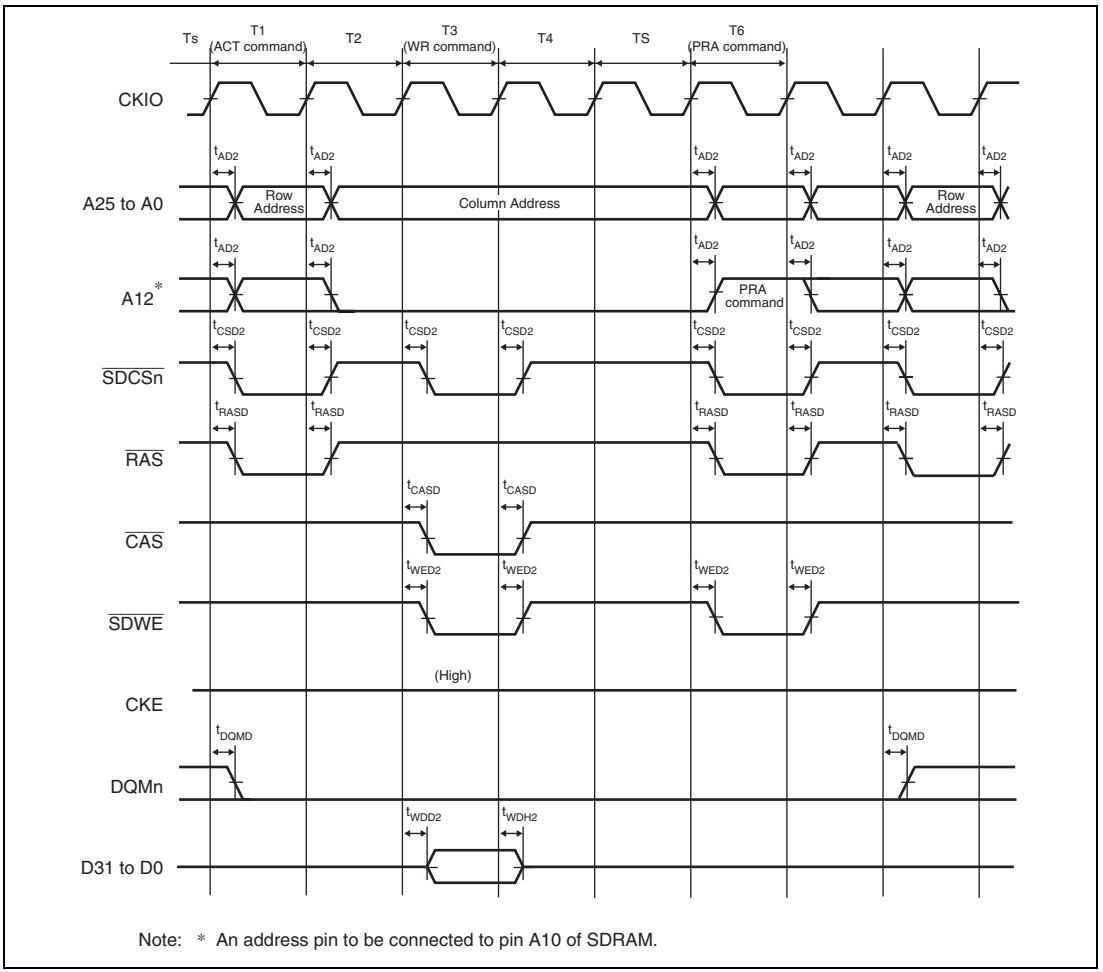


Figure 33.16 SDRAM Space: Single-Read Bus Timing (DCL = 2 (2 Cycles), DRCD = 1 (2 Cycles), DPCG = 1 (2 Cycles))



Note: * An address pin to be connected to pin A10 of SDRAM.

Figure 33.17 SDRAM Space: Single-Write Bus Timing (DCL = 2 (2 Cycles), DRCD = 1 (2 Cycles), DPCG = 1 (2 Cycles))

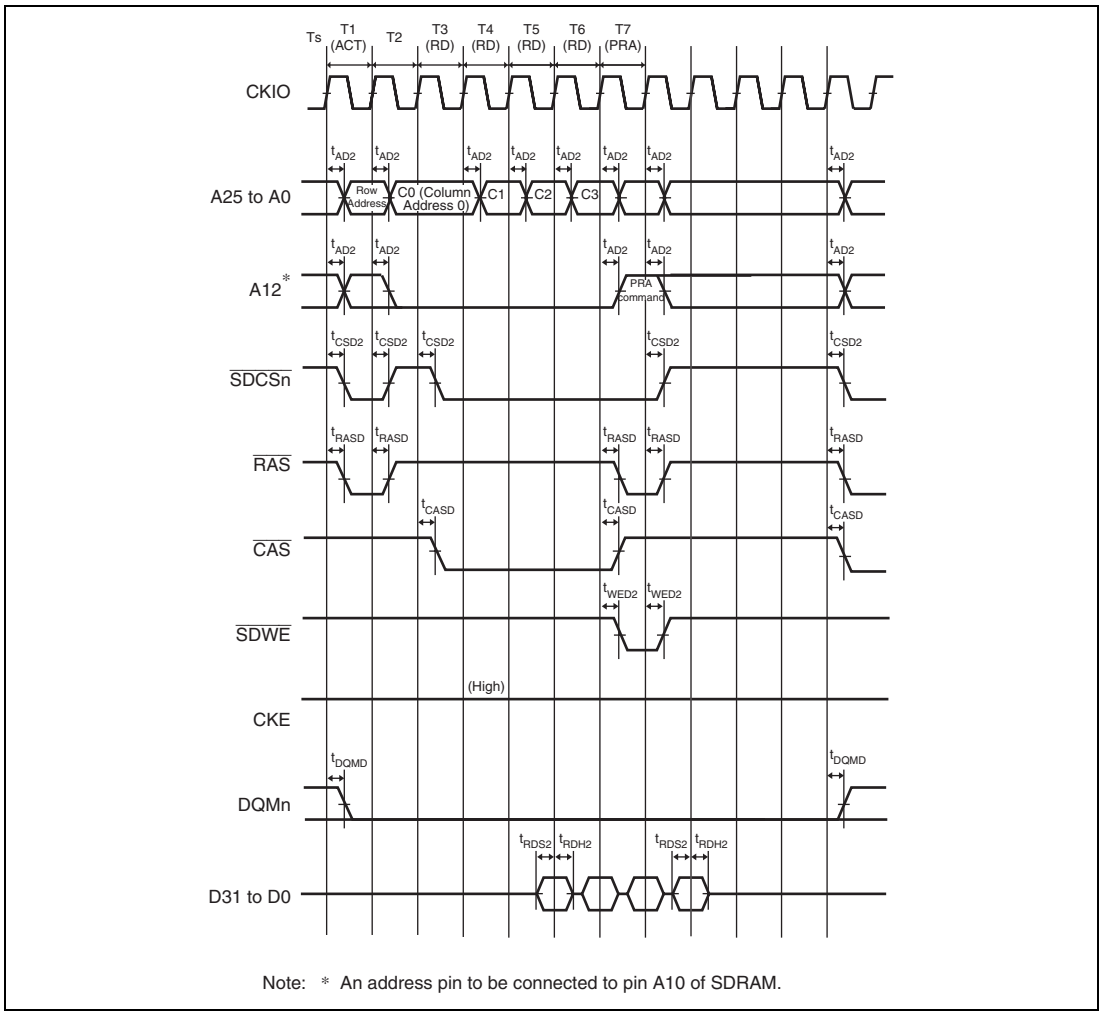
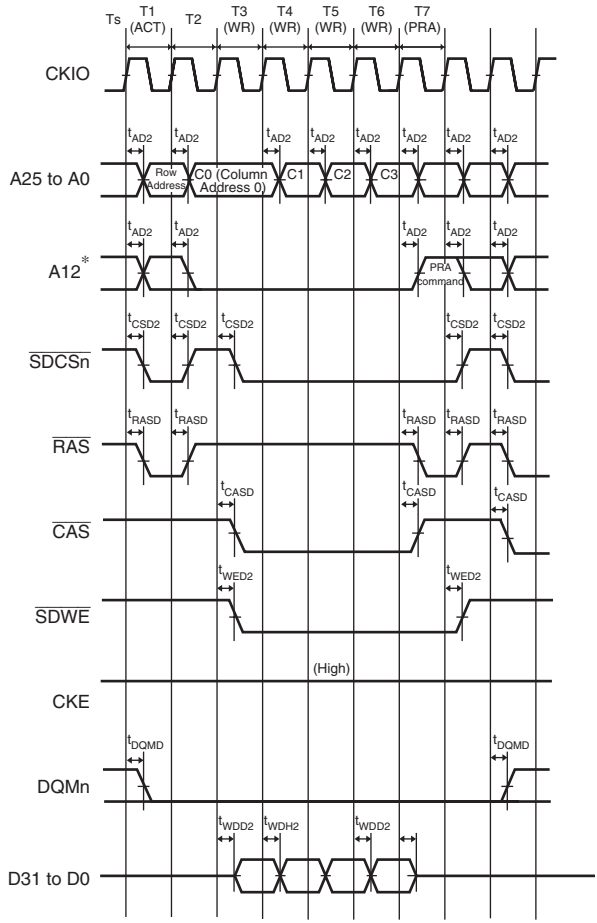
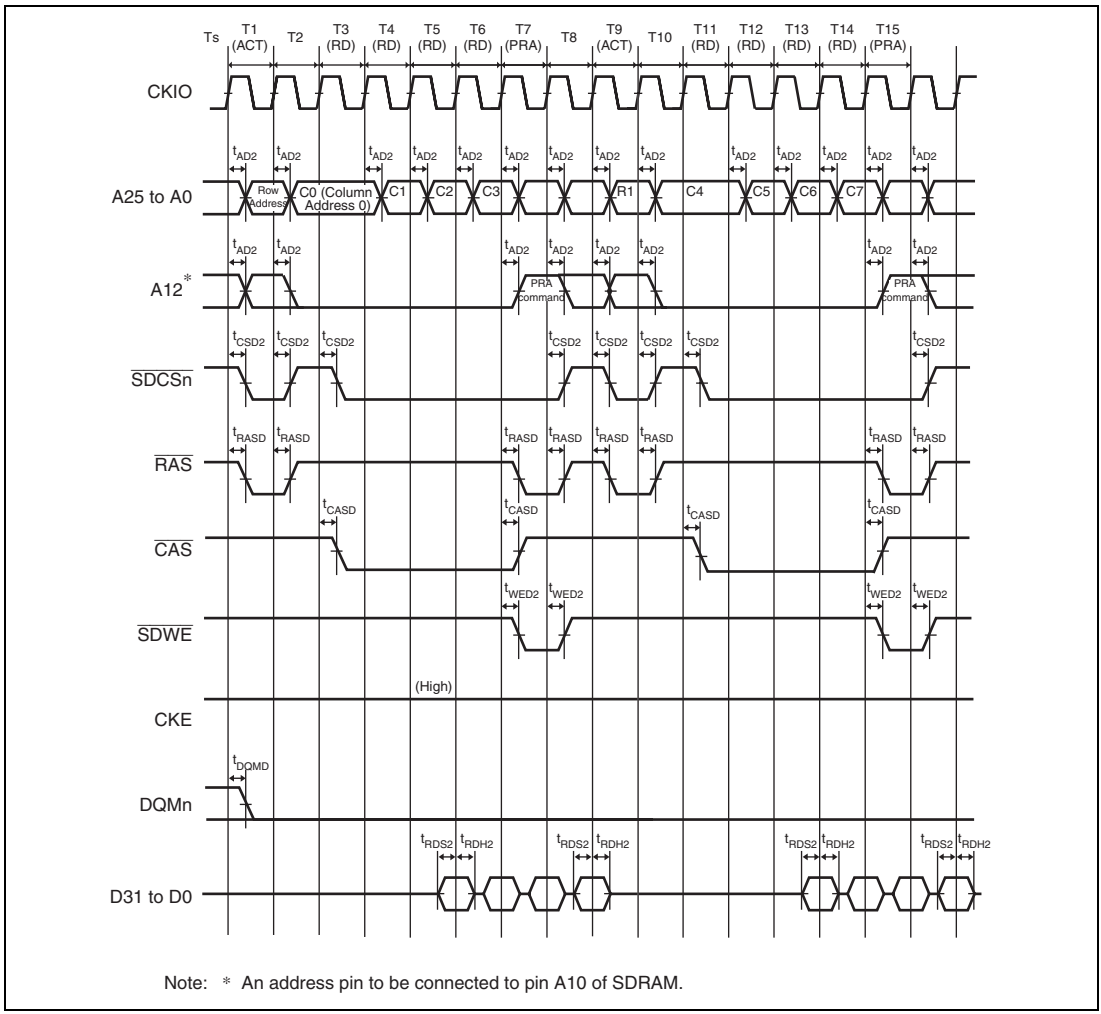


Figure 33.18 SDRAM Space: Multiple-Read Bus Timing
(4 Data Accesses, DCL = 2 (2 Cycles), DRCD = 1 (2 Cycles), DPCG = 1 (2 Cycles))



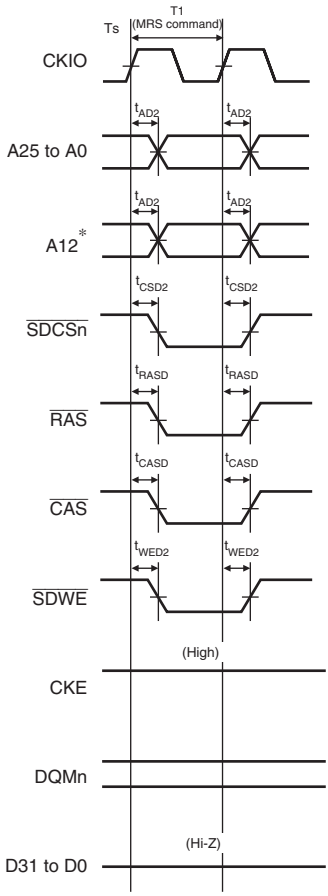
Note: * An address pin to be connected to pin A10 of SDRAM.

Figure 33.19 SDRAM Space: Multiple-Write Bus Timing (4 Data Accesses, DCL = 2 (2 Cycles), DRCD = 1 (2 Cycles), DPCG = 1 (2 Cycles))



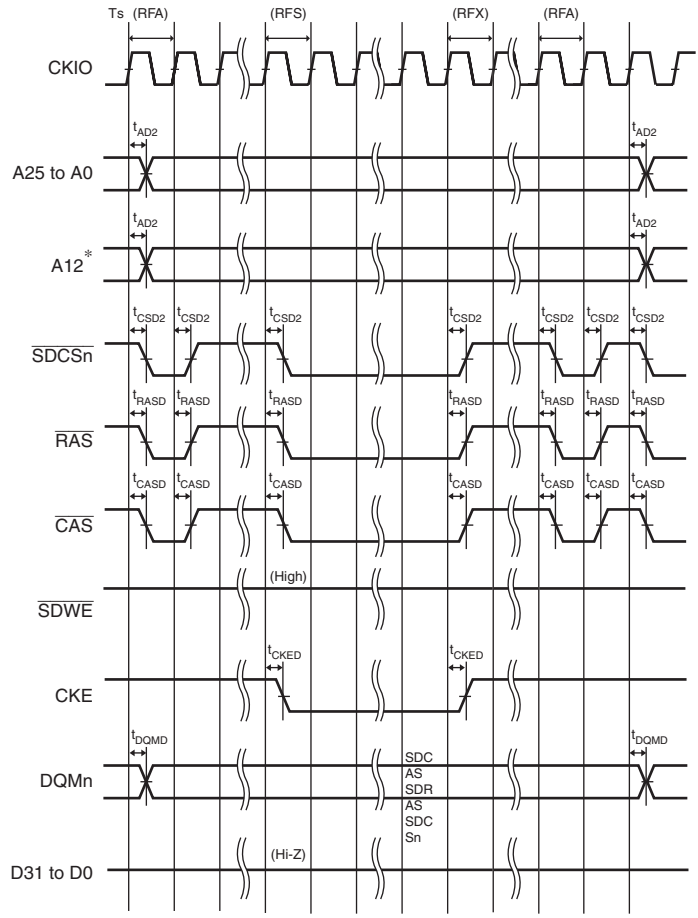
Note: * An address pin to be connected to pin A10 of SDRAM.

Figure 33.20 SDRAM Space: Bus Timing of Multiple-Read Across Rows (8 Data Accesses, DCL = 2 (2 Cycles), DRCD = 1 (2 Cycles), DPCG = 1 (2 Cycles))



Note: * An address pin to be connected to pin A10 of SDRAM.

Figure 33.21 SDRAM Space: Mode Register Set Bus Timing



Note: * An address pin to be connected to pin A10 of SDRAM.

Figure 33.22 SDRAM Space: Self-Refresh Bus Timing

33.4.4 UBC Timing

Table 33.9 UBC Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = −20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{UBCTRG}}$ delay time	t_{UBCTGD}	—	14	ns	Figure 33.23

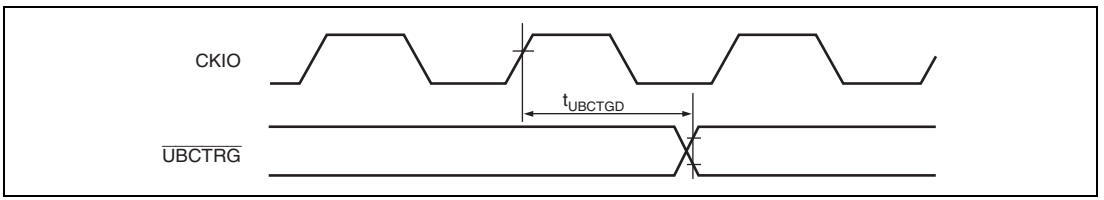


Figure 33.23 UBC Timing

33.4.5 DMAC Timing

Table 33.10 DMAC Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
DREQ setup time	t_{DRQS}	15	—	ns	Figure 33.24
DREQ hold time	t_{DRQH}	15	—	ns	Figure 33.24
DACK delay time	t_{DACK}	0	13	ns	Figure 33.25
DACT, TEND delay time	t_{DACD}	0	13	ns	Figures 33.25, 33.11 to 33.15

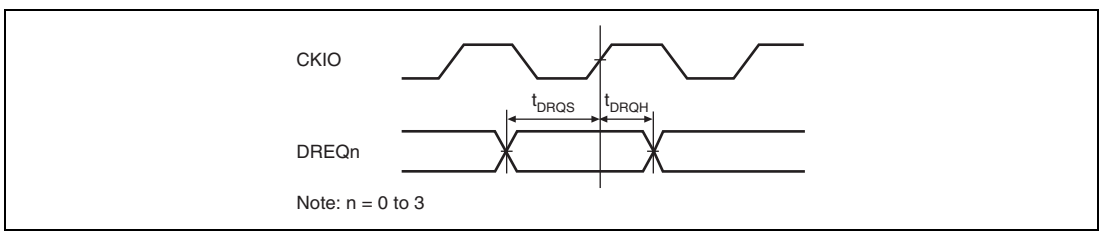


Figure 33.24 DREQ Input Timing

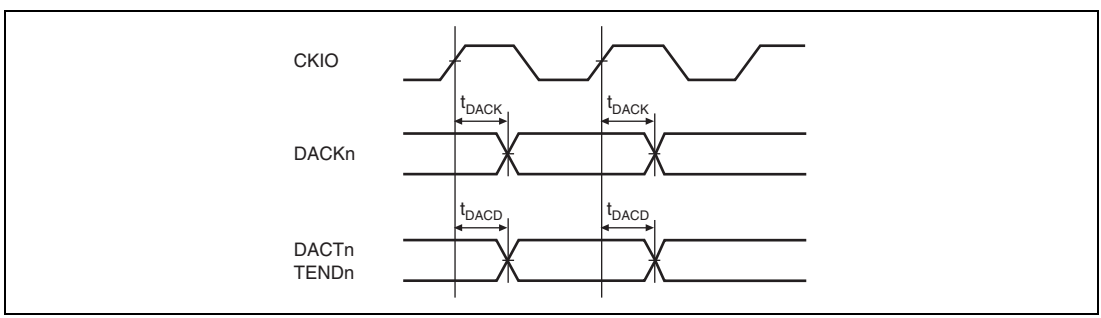


Figure 33.25 DACK, DACT, TEND Output Timing

33.4.6 MTU2 Timing

Table 33.11 MTU2 Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
Output compare output delay time	t_{TOCD}	—	100	ns	Figure 33.26
Input capture input setup time	t_{TICS}	20	—	ns	
Timer input setup time	t_{TCKS}	20	—	ns	Figure 33.27
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}	

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

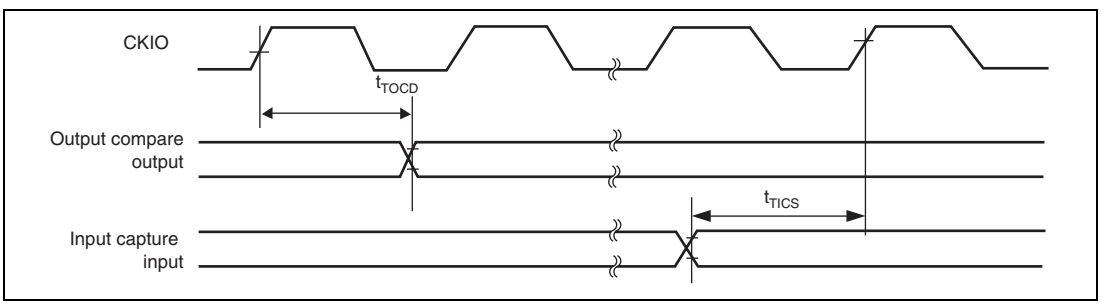


Figure 33.26 MTU2 Input/Output Timing

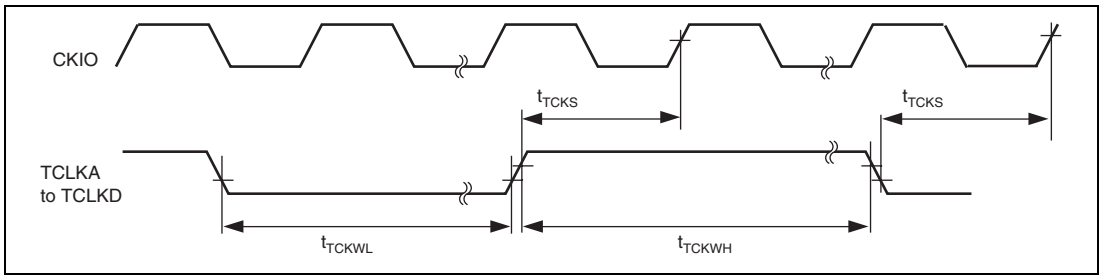


Figure 33.27 MTU2 Clock Input Timing

33.4.7 WDT Timing

Table 33.12 WDT Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	100	ns	Figure 33.28

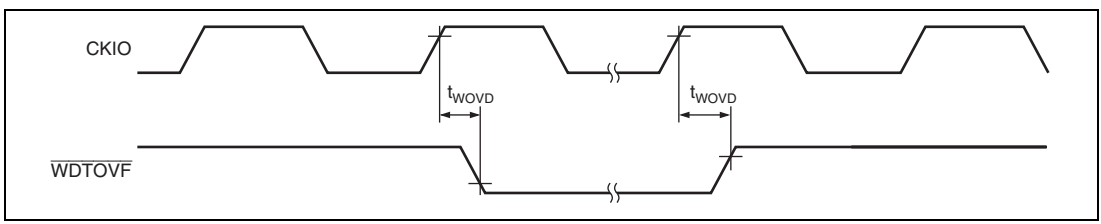


Figure 33.28 WDT Timing

33.4.8 SCIF Timing

Table 33.13 SCIF Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = −20 to 85 °C

Item		Symbol	Min.	Max.	Unit	Figure
Input clock cycle	(clocked synchronous)	t_{Scyc}	12	—	t_{pcyc}	Figure 33.29
	(asynchronous)		4	—	t_{pcyc}	Figure 33.29
Input clock rise time		t_{SCKr}	—	1.5	t_{pcyc}	Figure 33.29
Input clock fall time		t_{SCKf}	—	1.5	t_{pcyc}	Figure 33.29
Input clock width		t_{SCKW}	0.4	0.6	t_{Scyc}	Figure 33.29
Transmit data delay time (clocked synchronous)		t_{TXD}	—	3 t_{pcyc} + 15	ns	Figure 33.30
Receive data setup time (clocked synchronous)		t_{RXS}	4 t_{pcyc} + 15	—	ns	Figure 33.30
Receive data hold time (clocked synchronous)		t_{RXH}	1 t_{pcyc} + 15	—	ns	Figure 33.30

Note: t_{pcyc} indicates the peripheral clock (Pϕ) cycle.

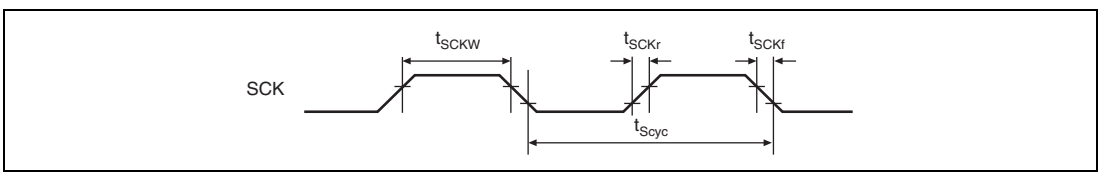


Figure 33.29 SCK Input Clock Timing

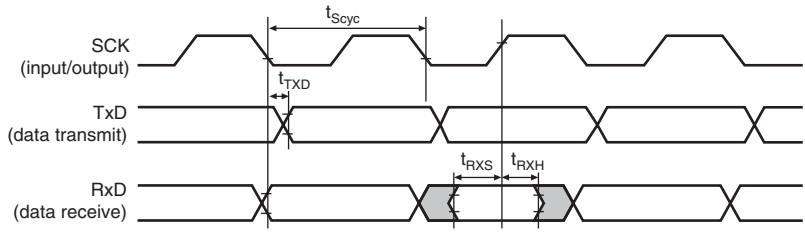


Figure 33.30 SCIF Input/Output Timing in Clocked Synchronous Mode

33.4.9 SSU Timing

Table 33.14 SSU Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item		Symbol	Min.	Max.	Unit	Figure
Clock cycle	Master	t_{SUcyc}	4	256	t_{pcyc}	Figures 33.31, 33.32, 33.33, 33.34
	Slave		4	256		
Clock high pulse width	Master	t_{HI}	48	—	ns	
	Slave		48	—		
Clock low pulse width	Master	t_{LO}	48	—	ns	
	Slave		48	—		
Clock rise time		t_{RISE}	—	12	ns	
Clock fall time		t_{FALL}	—	12	ns	
Data input setup time	Master	t_{SU}	30	—	ns	
	Slave		20	—		
Data input hold time	Master	t_H	0	—	ns	
	Slave		20	—		
$\overline{SCS}$ setup time	Master	t_{LEAD}	1.5	—	t_{pcyc}	
	Slave		1.5	—		
$\overline{SCS}$ hold time	Master	t_{LAG}	1.5	—	t_{pcyc}	
	Slave		1.5	—		
Data output delay time	Master	t_{OD}	—	50	ns	
	Slave		—	50		
Data output hold time	Master	t_{OH}	0	—	ns	
	Slave		0	—		
Continuous transmission delay time	Master	t_{TD}	1.5	—	t_{pcyc}	
	Slave		1.5	—		
Slave access time		t_{SA}	—	1	t_{pcyc}	Figures 33.33, 33.34
Slave out release time		t_{REL}	—	1	t_{pcyc}	

Note: t_{pcyc} indicates the peripheral clock (Pϕ) cycle.

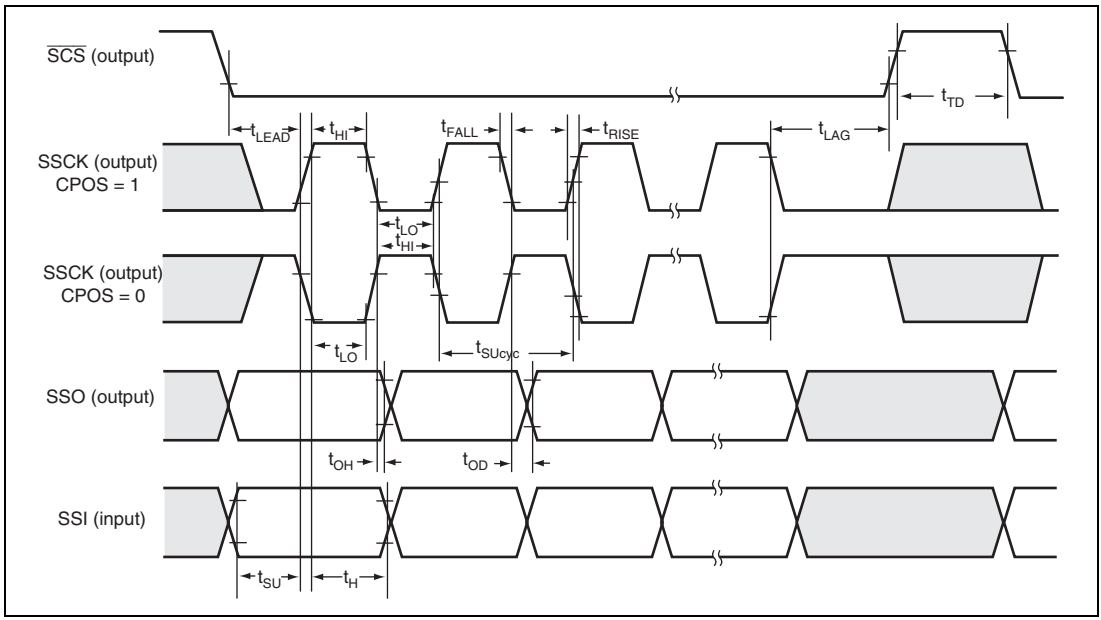


Figure 33.31 SSU Timing (Master, CPHS = 1)

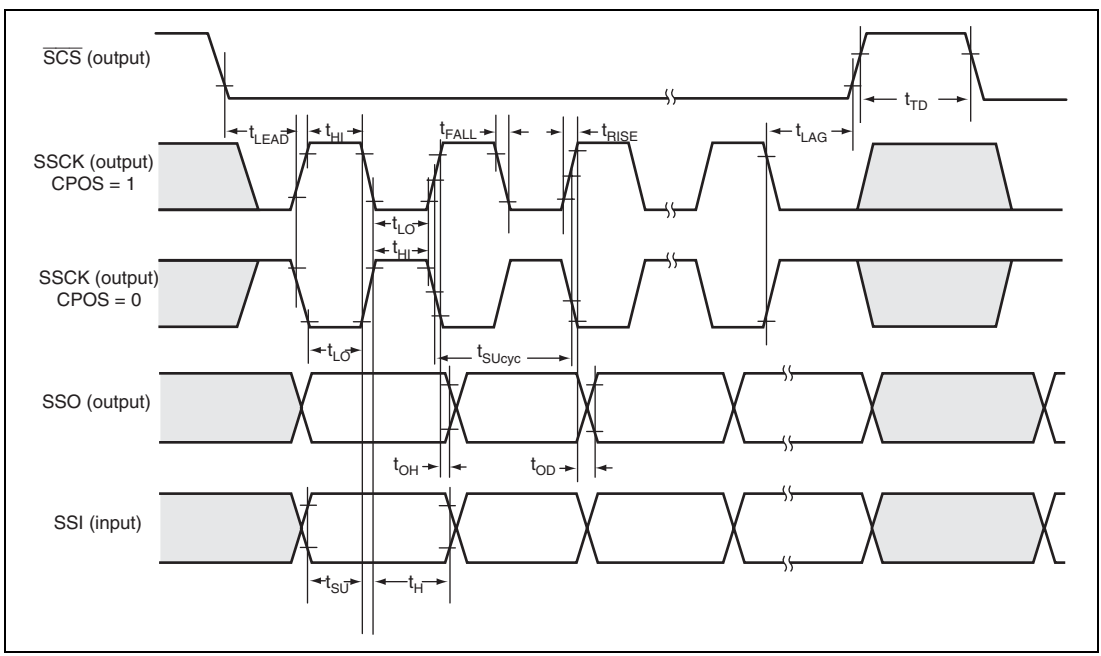


Figure 33.32 SSU Timing (Master, CPHS = 0)

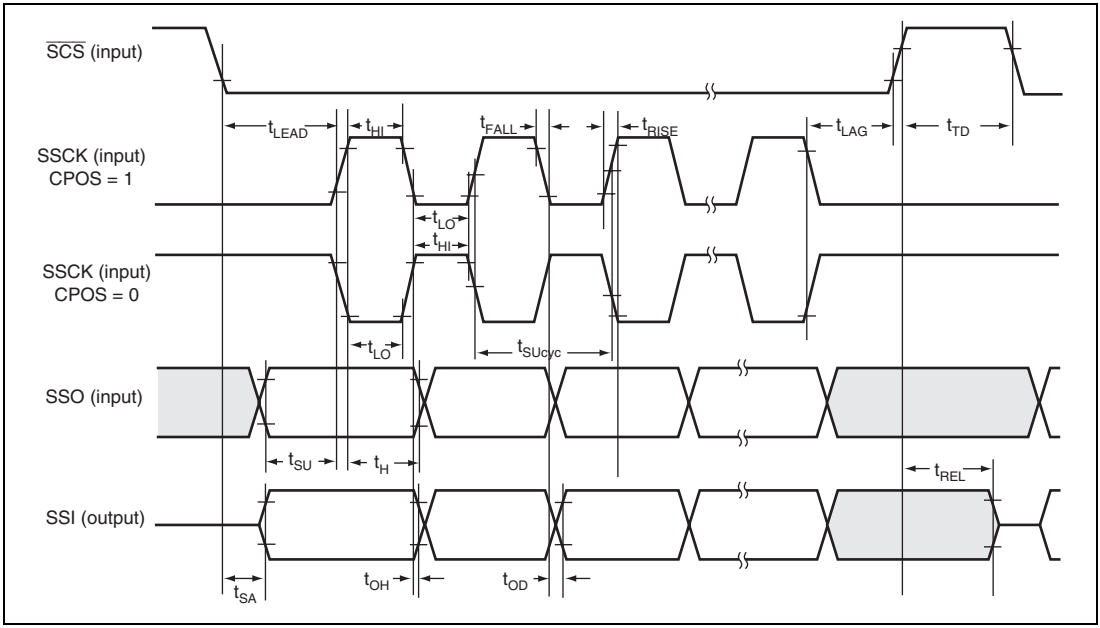


Figure 33.33 SSU Timing (Slave, CPHS = 1)

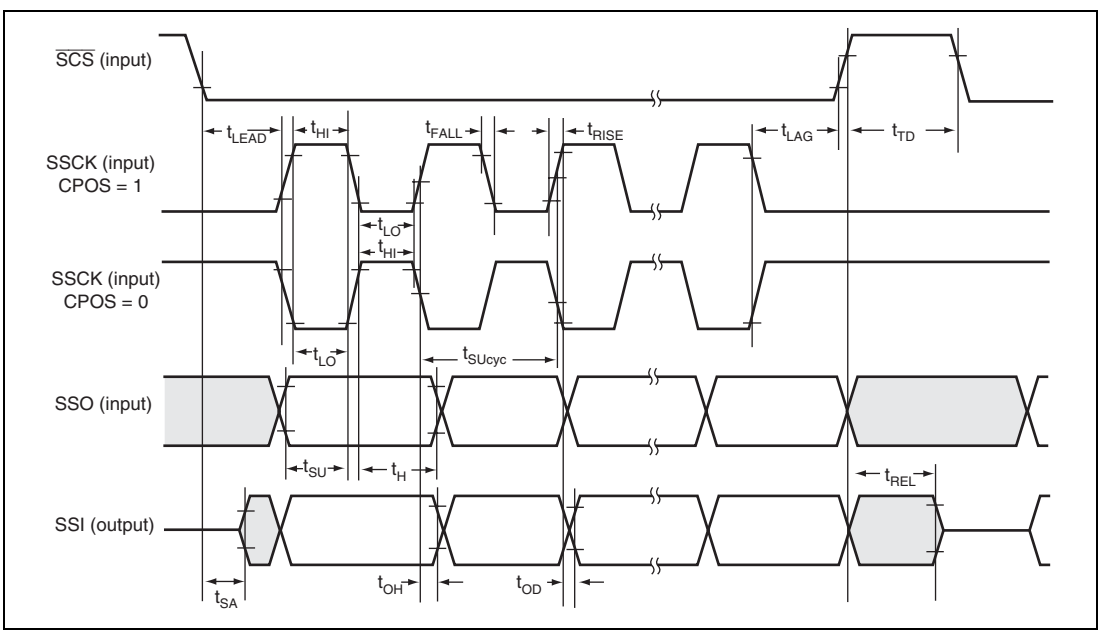


Figure 33.34 SSU Timing (Slave, CPHS = 0)

33.4.10 IIC3 Timing

Table 33.15 (1) IIC3 Timing I²C Bus Format

Conditions: V_{CC} = $PLLV_{CC}$ = 1.1 to 1.3 V, $USBDV_{CC}$ = 1.1 to 1.3 V, $USBAV_{CC}$ = 1.1 to 1.3 V, PV_{CC} = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, $USBAPV_{CC}$ = 3.0 to 3.6 V, $2DGAPV_{CC}0$ = 3.0 to 3.6 V, $2DGAPV_{CC}1$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $USBAV_{SS}$ = AV_{SS} = $USBAPV_{SS}$ = $2DGAPV_{SS}0$ = $2DGAPV_{SS}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
SCL input cycle time	t_{SCL}	12 t_{pcyc}*1 + 600	—	ns	Figure 33.35 (1)
SCL input high pulse width	t_{SCLH}	3 t_{pcyc}*1 + 300	—	ns	
SCL input low pulse width	t_{SCLL}	5 t_{pcyc}*1 + 300	—	ns	
SCL, SDA input rise time	t_{Sr}	—	300	ns	
SCL, SDA input fall 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}*1	
Start condition input hold time	t_{STAH}	3	—	t_{pcyc}*1	
Retransmit start condition input setup time	t_{STAS}	3	—	t_{pcyc}*1	
Stop condition input setup time	t_{STOS}	3	—	t_{pcyc}*1	
Data input setup time	t_{SDAS}	1 t_{pcyc}*1 + 20	—	ns	
Data input hold time	t_{SDAH}	0	—	ns	
SCL, SDA capacitive load	Cb	0	400	pF	
SCL, SDA output fall time*3	t_{Sf}	—	250	ns	

Notes: 1. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.
2. Depends on the value of NF2CYC.
3. Indicates the I/O buffer characteristic.

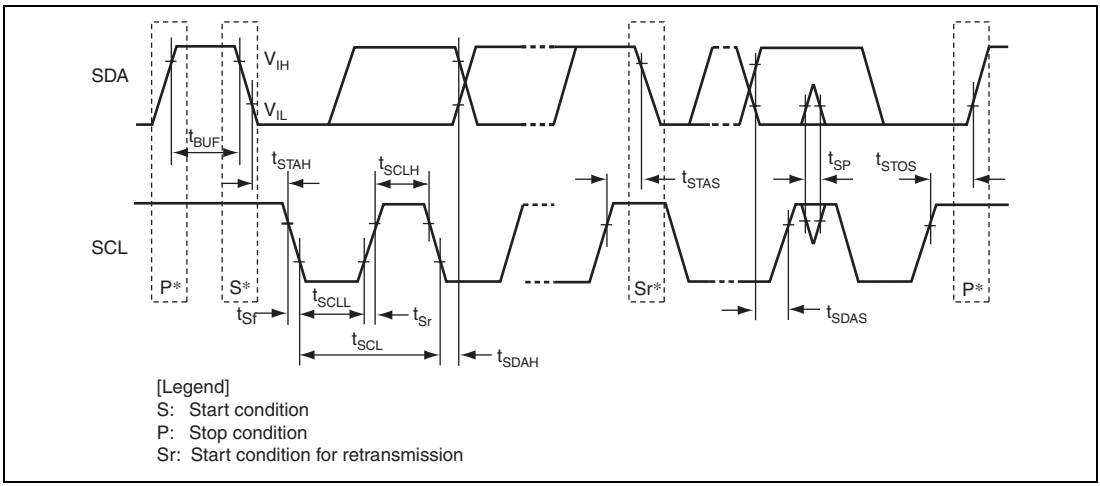


Figure 33.35 (1) IIC3 Input/Output Timing

Table 33.15 (2) IIC3 Timing Clock Synchronized Serial Format

Item	Symbol	Min.	Max.	Unit	Figure
SCL input cycle time	t_{SCL}	$12t_{pcyc}$*1 +600	—	ns	Figure 33.35 (2)
SCL input high pulse width	t_{SCLH}	$3t_{pcyc}$*1 +300	—	ns	
SCL input high low width	t_{SCLL}	$5t_{pcyc}$*1 +300	—	ns	
SCL, SDA input rise time	t_{Sr}	—	300	ns	
SCL, SDA input fall time	t_{Sf}	—	300	ns	
SCL, SDA input spike pulse removal time*2	t_{SP}	—	1, 2	t_{pcyc}*1	Figure 33.35 (3)
Data output delay time	t_{HD}	0	900	ns	
Data input setup time	t_{SDAS}	$1t_{pcyc}$*1 +20	—	ns	
Data input hold time	t_{SDAH}	0	—	ns	
SCL, SDA capacitive load	Cb	0	400	pF	Figures 33.35 (2) and 33.35 (3)
SCL, SDA output fall time*3	f_{Sf}	—	250	ns	

Notes: 1. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.
2. Depends on the value of NF2CYC.
3. Indicates the I/O buffer characteristic.

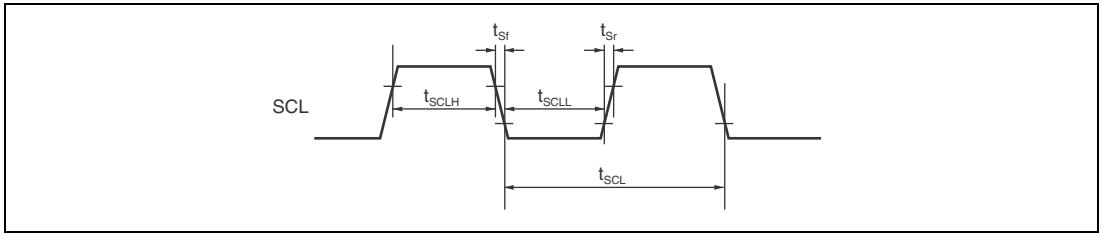


Figure 33.35 (2) Clock Input/Output Timing

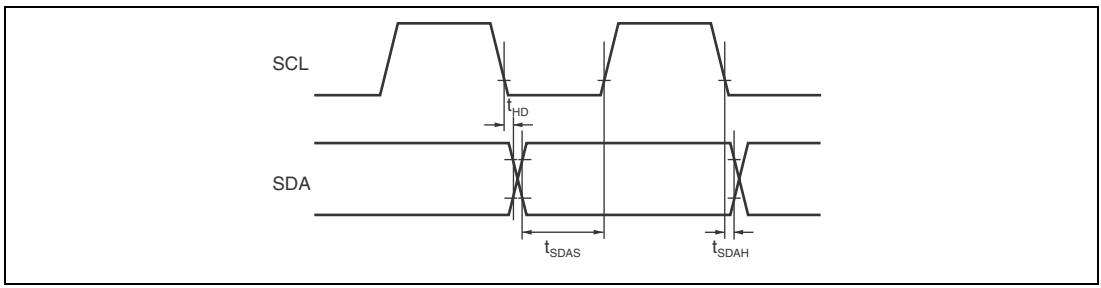


Figure 33.35 (3) Transmission and Reception Timing

33.4.11 SSIF Timing

Table 33.16 SSIF Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Remarks	Figure
Output clock cycle	t_O	80	64000	ns	Output	Figure 33.36
Input clock cycle	t_I	80	64000	ns	Input	
Clock high	t_{HC}	32	—	ns	Bidirectional	
Clock low	t_{LC}	32	—	ns		
Clock rise time	t_{RC}	—	25	ns	Output	
Delay	t_{DTR}	–5	25	ns	Transmit	Figures 33.37, 33.38
Setup time	t_{SR}	25	—	ns	Receive	Figures 33.39, 33.40
Hold time	t_{HTR}	5	—	ns	Receive, transmit	Figures 33.37 to 33.40
AUDIO_CLK input frequency	f_{AUDIO}	1	40	MHz		Figure 33.41

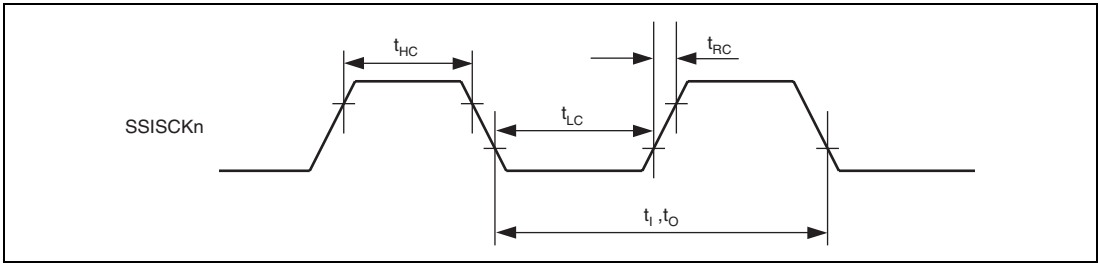


Figure 33.36 Clock Input/Output Timing

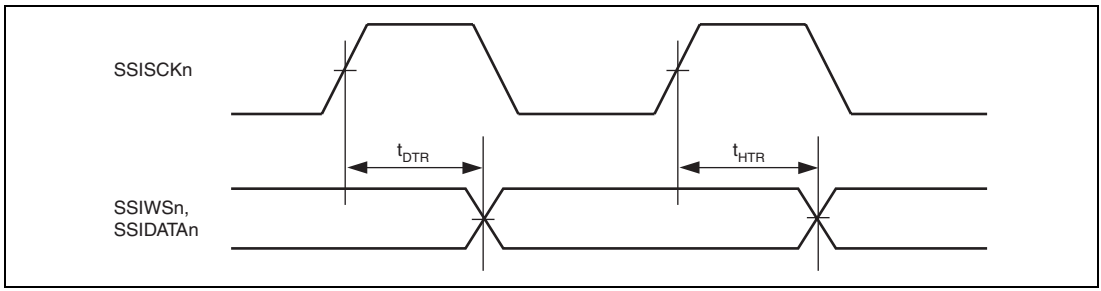


Figure 33.37 SSIF Transmit Timing (1)

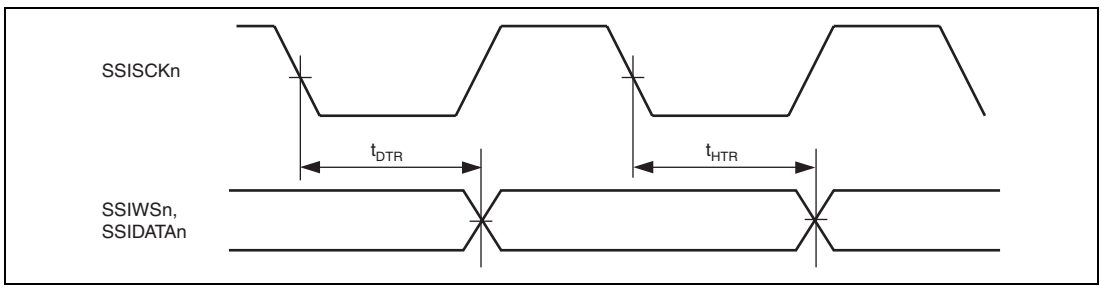


Figure 33.38 SSIF Transmit Timing (2)

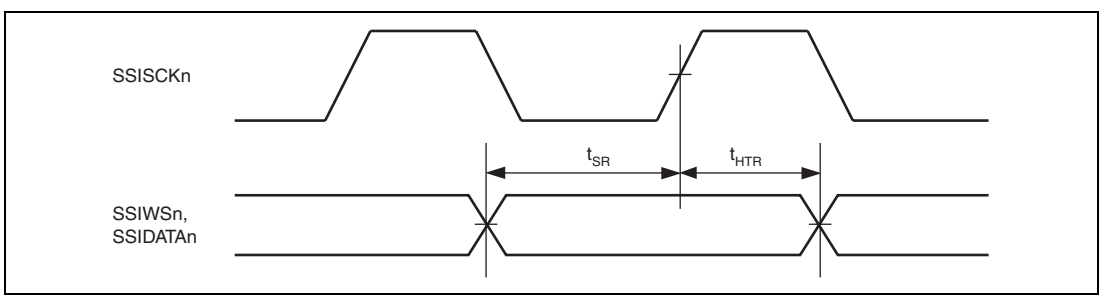


Figure 33.39 SSIF Receive Timing (1)

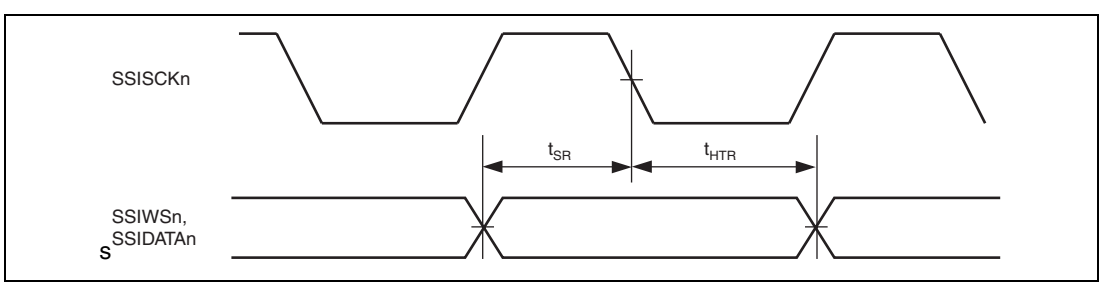


Figure 33.40 SSIF Receive Timing (2)

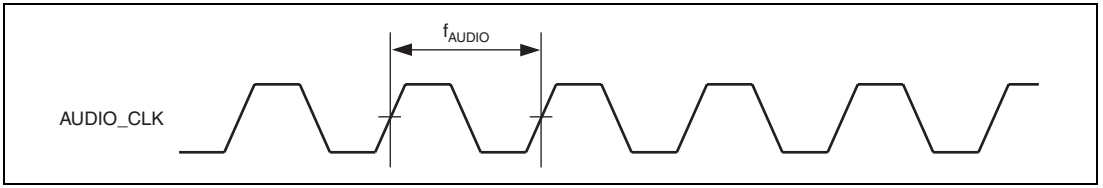


Figure 33.41 AUDIO_CLK Input Timing

33.4.12 RCAN-TL1 Timing

Table 33.17 RCAN-TL1 Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = −20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
Transmit data delay time	t_{CTXD}	—	100	ns	Figure 33.42
Receive data setup time	t_{CRXS}	100	—		
Receive data hold time	t_{CRXH}	100	—		

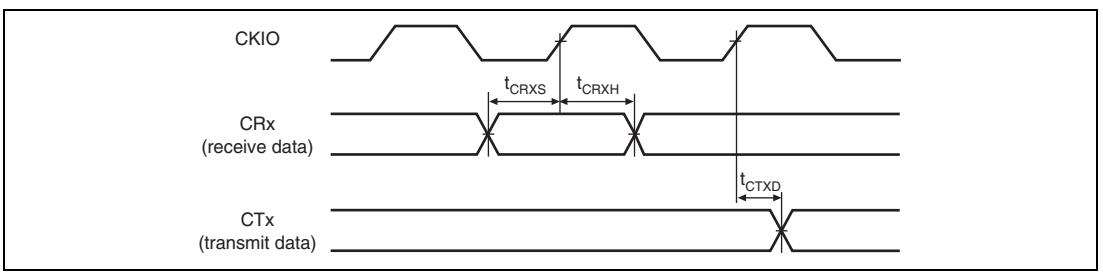


Figure 33.42 RCAN-TL1 Input/Output Timing

33.4.13 ADC Timing

Table 33.18 ADC Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Module	Item		Symbol	Min.	Max.	Unit	Figure
A/D converter	Trigger input setup time	B:P clock ratio = 1:1	t_{TRGS}	17	—	ns	Figure 33.43
		B:P clock ratio = 2:1		t_{cyc} + 17	—		
		B:P clock ratio = 4:1		3 × t_{cyc} + 17	—		

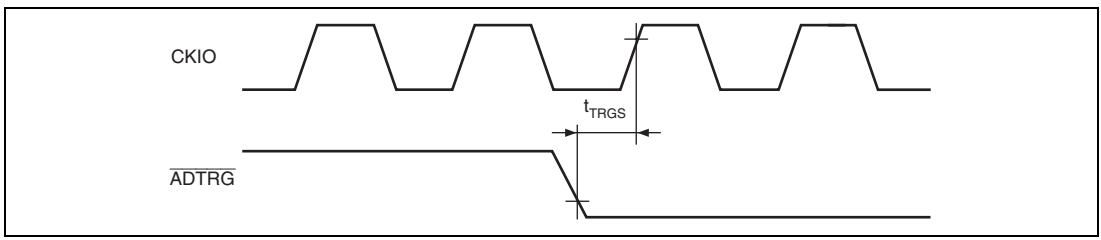


Figure 33.43 A/D Converter External Trigger Input Timing

33.4.14 FLCTL Timing

Table 33.19 AND Type Flash Memory Interface Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
Command issue setup time	t_{ACDS}	$2 \times t_{fcyc} - 10$	—	ns	Figures 33.44, 33.48
Command issue hold time	t_{ACDH}	$2 \times t_{fcyc} - 10$	—	ns	
Data output setup time	t_{ADOS}	$t_{fcyc} - 10$	—	ns	Figures 33.44, 33.45, 33.48
Data output hold time	t_{ADOH}	$t_{fcyc} - 10$	—	ns	
Data output setup time 2	t_{ADOS2}	$0.5 \times t_{fcyc} - 10$	—	ns	Figure 33.47
Data output hold time 2	t_{ADOH2}	$0.5 \times t_{fcyc} - 10$	—	ns	
$\overline{FWE}$ cycle time	t_{ACWC}	$2 \times t_{fcyc} - 5$	—	ns	Figure 33.45
$\overline{FWE}$ low pulse width	t_{AWP}	$t_{fcyc} - 5$	—	ns	Figures 33.44, 33.45, 33.48
$\overline{FWE}$ high pulse width	t_{AWPH}	$t_{fcyc} - 5$	—	ns	Figure 33.45
Command to address transition time	t_{ACAS}	$4 \times t_{fcyc}$	—	ns	
Address to data read transition time	t_{AADDR}	$32 \times t_{pcyc}$	—	ns	Figure 33.46
Address to ready/busy transition time	t_{AADRB}	—	$35 \times t_{pcyc}$	ns	
Ready/busy to data read transition time	t_{ARBDR}	$3 \times t_{fcyc}$	—	ns	
Data read setup time	t_{ADRS}	$t_{fcyc} - 10$	—	ns	Figure 33.46
FSC cycle time	t_{ASCC}	$t_{fcyc} - 5$	—	ns	Figures 33.46, 33.47
FSC high pulse width	t_{ASP}	$0.5 \times t_{fcyc} - 5$	—	ns	
FSC low pulse width	t_{ASPL}	$0.5 \times t_{fcyc} - 5$	—	ns	
Read data setup time	t_{ARDS}	24	—	ns	Figures 33.46, 33.48
Read data hold time	t_{ARDH}	5	—	ns	
Status read setup time	t_{ASRDS}	$2 \times t_{pcyc} + 24$	—	ns	Figure 33.48

Item	Symbol	Min.	Max.	Unit	Figure
Address to data write transition time	t_{AADDW}	$4 \times t_{pcyc}$	—	ns	Figure 33.47
Data write setup time	t_{ADWS}	$50 \times t_{pcyc}$	—	ns	
FSC to FOE hold time	t_{ASOH}	$2 \times t_{fcyc} - 10$	—	ns	Figure 33.46

Note: t_{fcyc} indicates the period of one cycle of the FLCTL clock.
t_{pcyc} indicates the period of one cycle of the peripheral clock (Pϕ).

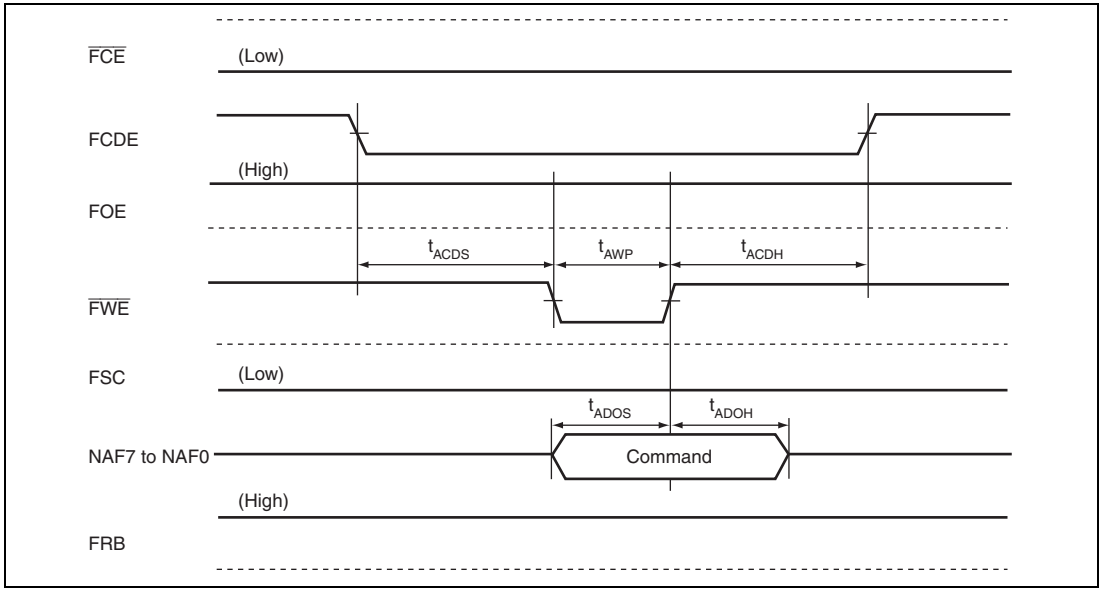


Figure 33.44 AND Type Flash Memory Command Issuance Timing

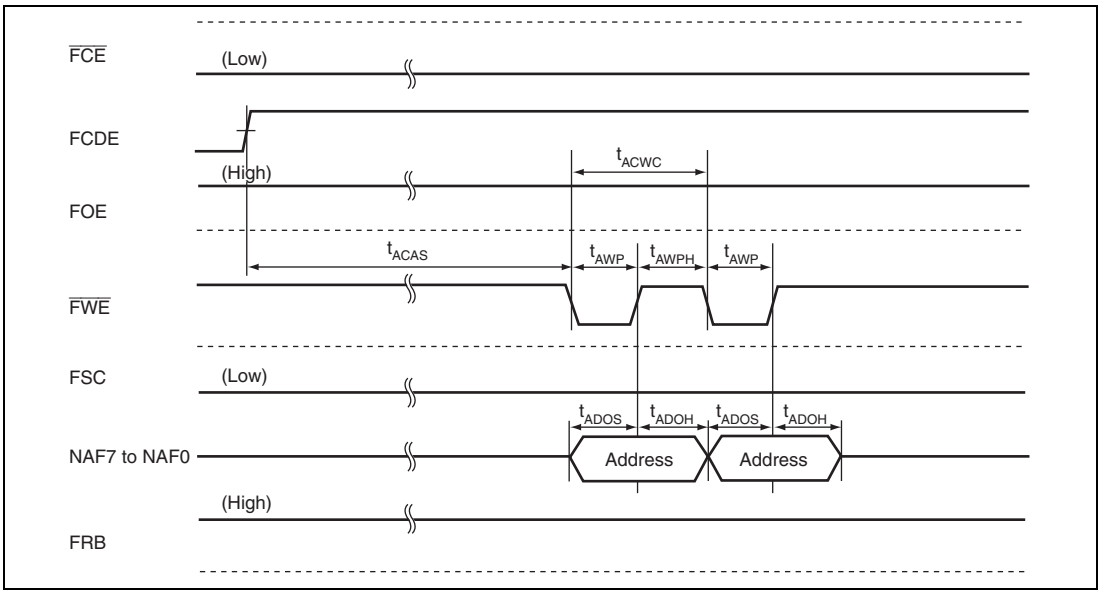


Figure 33.45 AND Type Flash Memory Address Issuance Timing

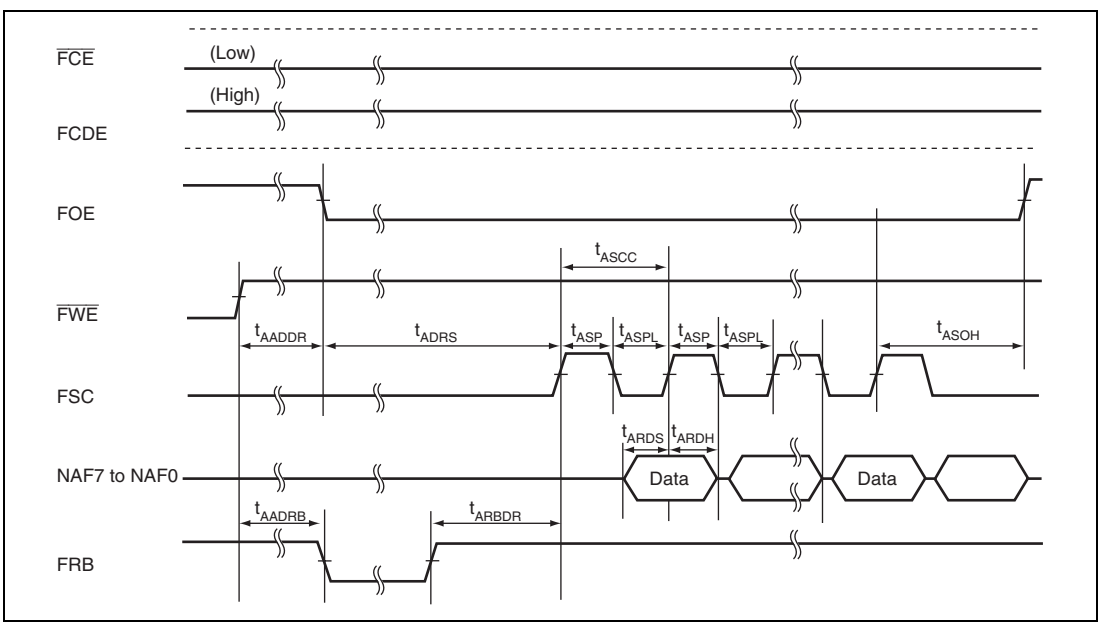


Figure 33.46 AND Type Flash Memory Data Read Timing

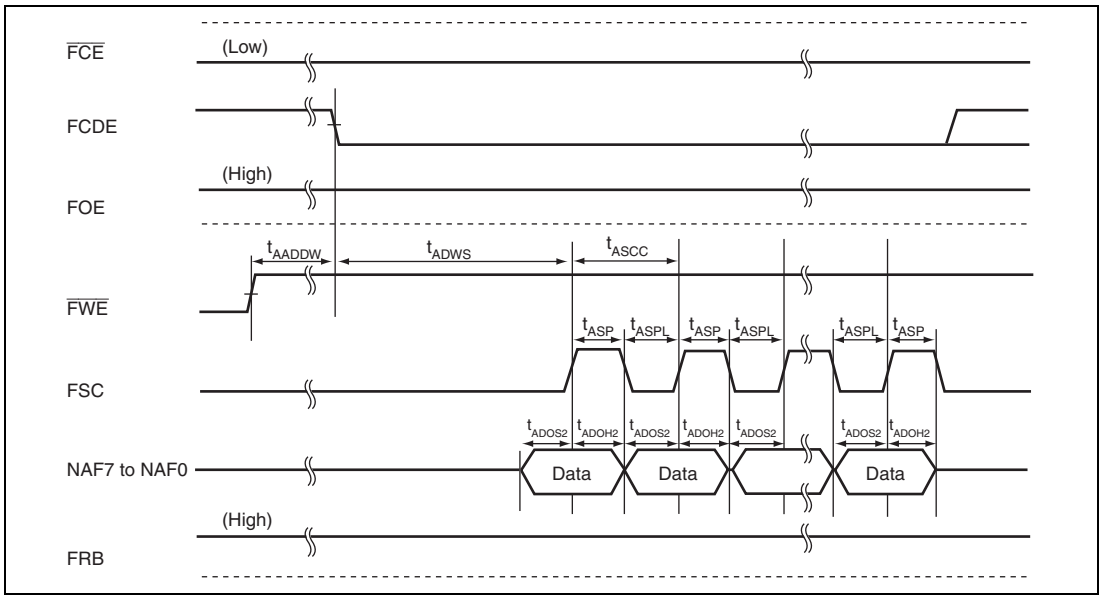


Figure 33.47 AND Type Flash Memory Data Write Timing

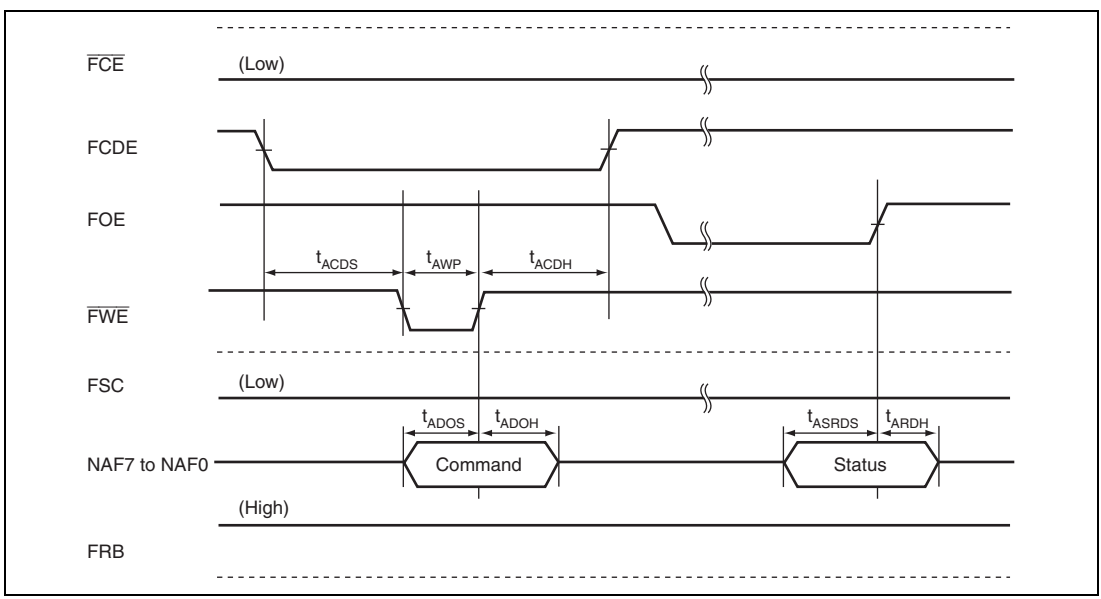


Figure 33.48 AND Type Flash Memory Status Read Timing

Table 33.20 NAND Type Flash Memory Interface Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
Command output setup time	t_{NCDS}	$2 \times t_{fcyc} - 10$	—	ns	Figures 33.49, 33.53
Command output hold time	t_{NCDH}	$1.5 \times t_{fcyc} - 5$	—	ns	
Data output setup time	t_{NDOS}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figures 33.49, 33.50, 33.52, 33.53
Data output hold time	t_{NDOH}	$0.5 \times t_{wfcyc} - 10$	—	ns	
Command to address transition time 1	t_{NCDAD1}	$1.5 \times t_{fcyc} - 10$	—	ns	Figures 33.49, 33.50
Command to address transition time 2	t_{NCDAD2}	$2 \times t_{fcyc} - 10$	—	ns	Figure 33.50
$\overline{FWE}$ cycle time	t_{NWC}	$t_{wfcyc} - 5$	—	ns	Figures 33.50, 33.52
$\overline{FWE}$ low pulse width	t_{NWP}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figures 33.49, 33.50, 33.52, 33.53
$\overline{FWE}$ high pulse width	t_{NWH}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figures 33.50, 33.52
Address to ready/busy transition time	t_{NADRB}	—	$32 \times t_{pcyc}$	ns	Figures 33.50, 33.51
Command to ready/busy transition time	t_{NCDRB}	—	$10 \times t_{pcyc}$	ns	
Ready/busy to data read transition time 1	t_{NRBDR1}	$1.5 \times t_{fcyc}$	—	ns	Figure 33.51
Ready/busy to data read transition time 2	t_{NRBDR2}	$32 \times t_{pcyc}$	—	ns	
FSC cycle time	t_{NSCC}	$t_{wfcyc} - 5$	—	ns	
FSC low pulse width	t_{NSP}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figures 33.51, 33.53
FSC high pulse width	t_{NSPH}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figure 33.51

Item	Symbol	Min.	Max.	Unit	Figure
Read data setup time	t_{NRDS}	24	—	ns	Figures 33.51, 33.53
Read data hold time	t_{NRDH}	5	—	ns	Figures 33.51, 33.53
Data write setup time	t_{NDWS}	32 × t_{pcyc}	—	ns	Figure 33.52
Command to status read transition time	t_{NCDSR}	4 × t_{fcyc}	—	ns	Figure 33.53
Command output off to status read transition time	t_{NCDFSR}	3.5 × t_{fcyc}	—	ns	
Status read setup time	t_{NSTS}	2.5 × t_{fcyc}	—	ns	

Note: t_{fcyc} indicates the period of one cycle of the FLCTL clock.
t_{wfcyc} indicates the period of one cycle of the FLCTL clock when the NANDWF bit is cleared to 0, and indicates the period of two cycles of the FLCTL clock when the NANDWF bit is set to 1.
t_{pcyc} indicates the period of one cycle of the peripheral clock (Pϕ).

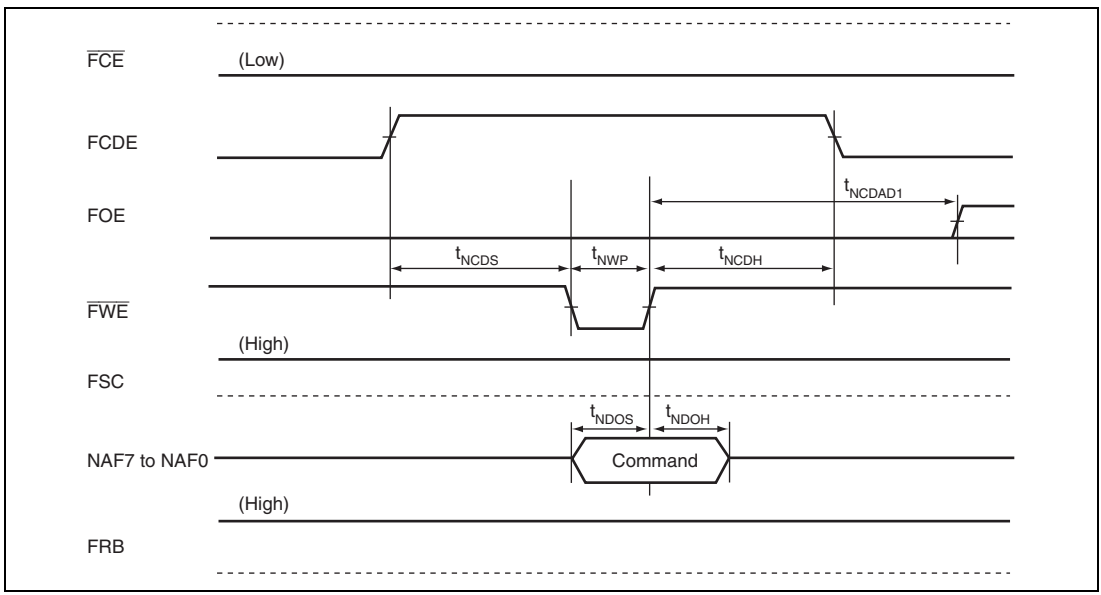


Figure 33.49 NAND Type Flash Memory Command Issuance Timing

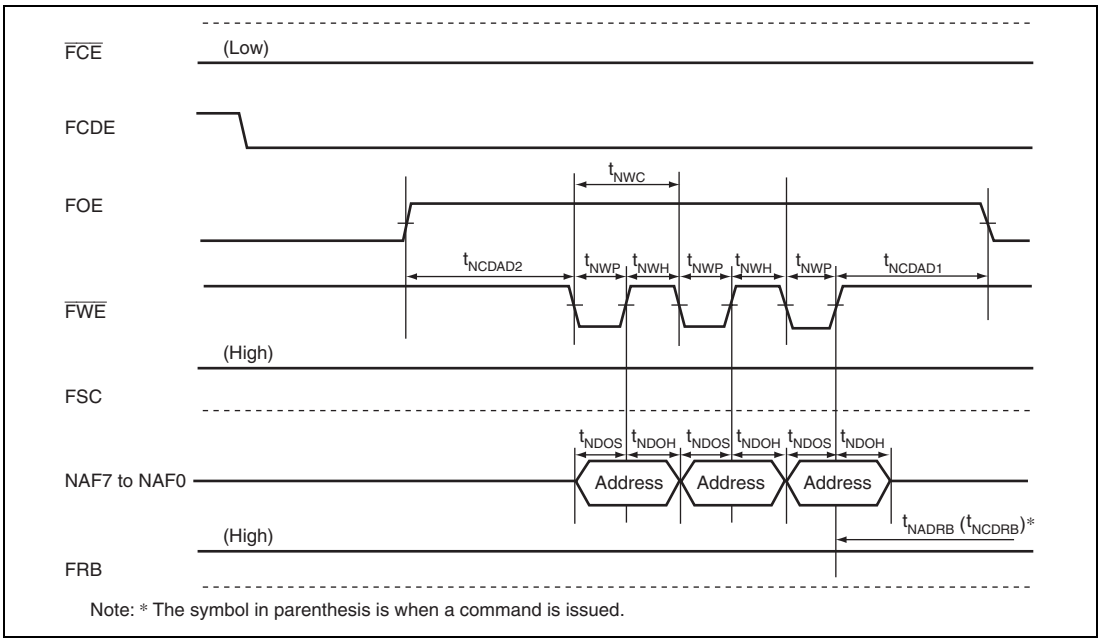


Figure 33.50 NAND Type Flash Memory Address Issuance Timing

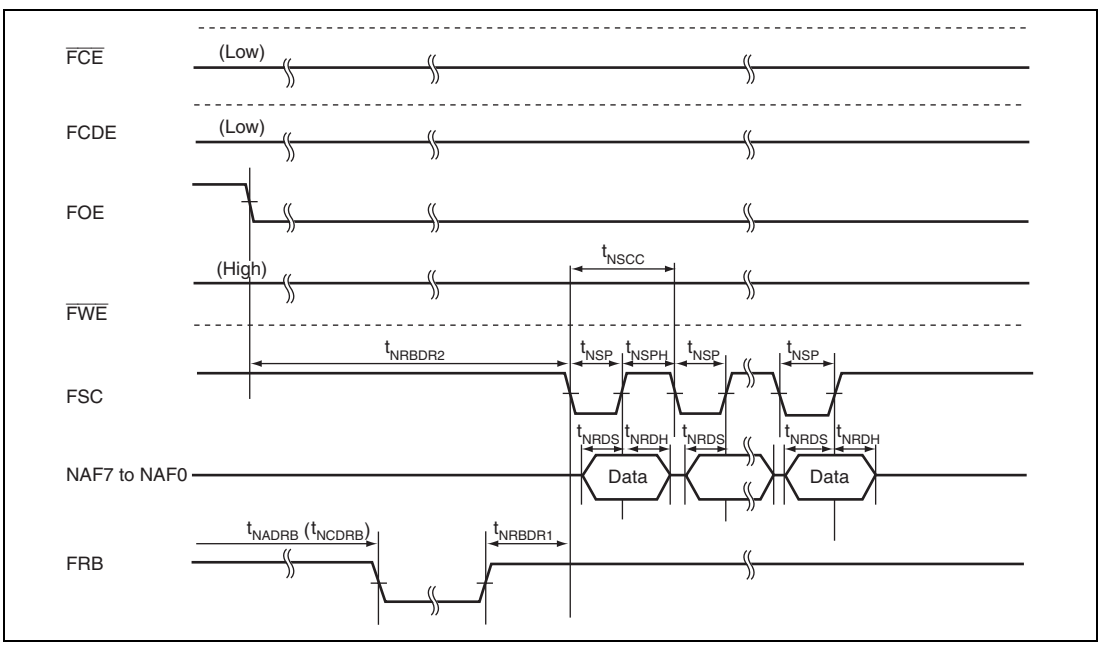


Figure 33.51 NAND Type Flash Memory Data Read Timing

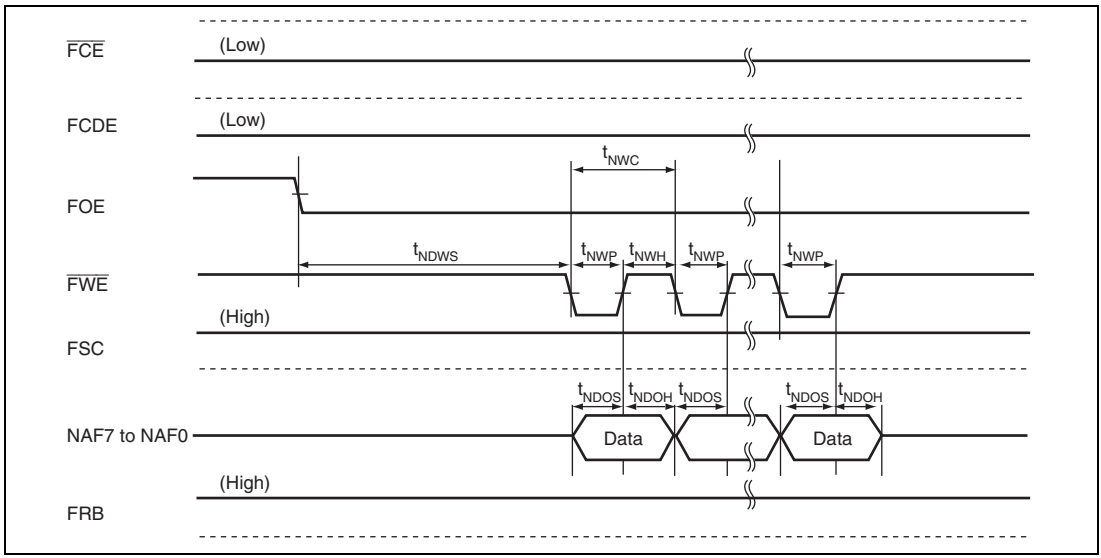


Figure 33.52 NAND Type Flash Memory Data Write Timing

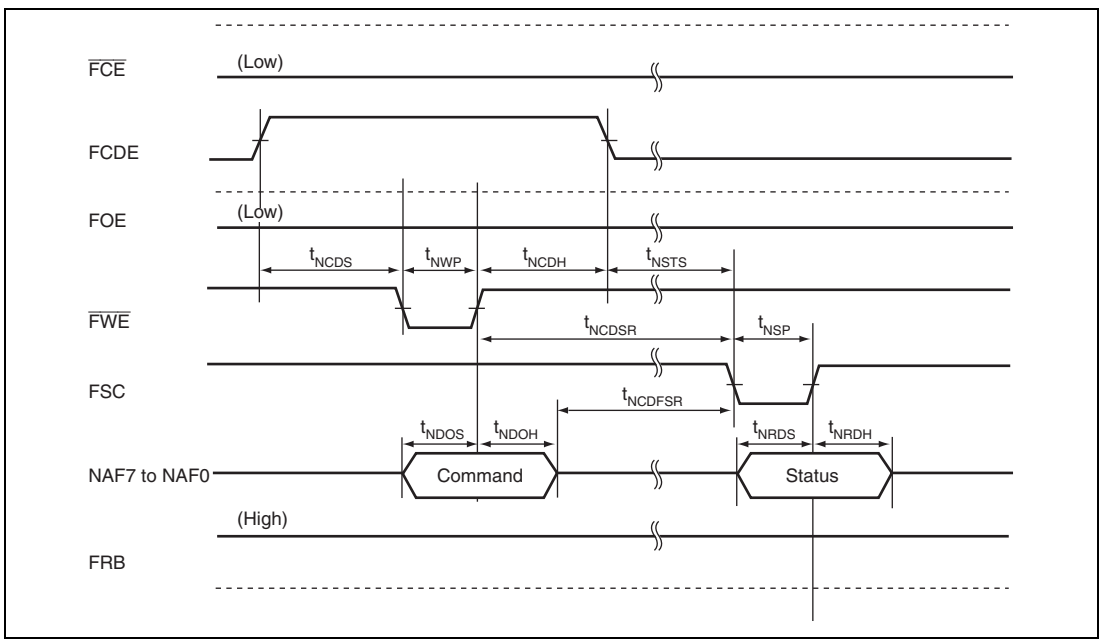


Figure 33.53 NAND Type Flash Memory Status Read Timing

33.4.15 USB Timing

Table 33.21 USB Transceiver Timing (Low-Speed)

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
Rise time	t_{LR}	75	—	300	ns	Figure 33.54
Fall time	t_{LF}	75	—	300	ns	
Rise/fall time lag	t_{LR}/t_{LF}	80	—	125	%	

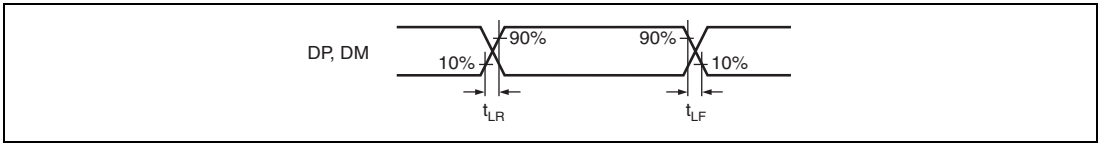


Figure 33.54 DP and DM Output Timing (Low-Speed)

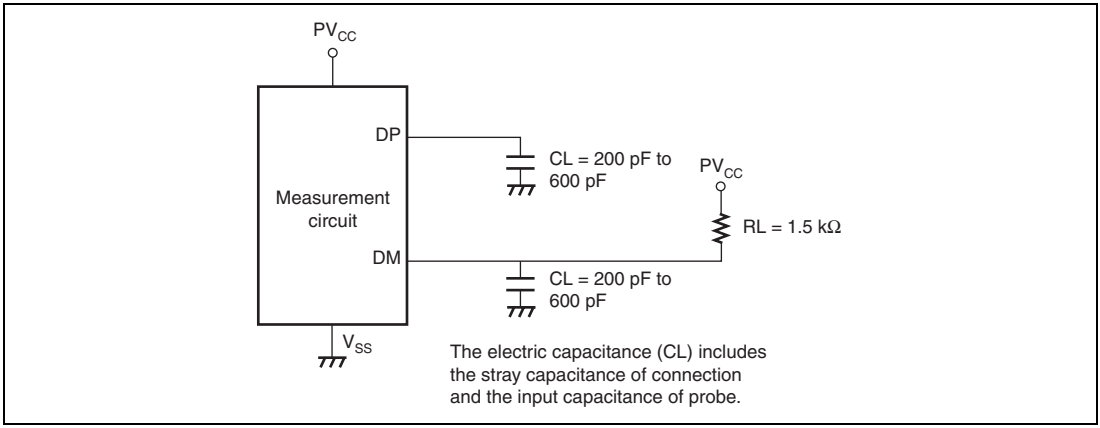


Figure 33.55 Measurement Circuit (Low-Speed)

Table 33.22 USB Transceiver Timing (Full-Speed)

Conditions: V_{CC} = $PLLV_{CC}$ = 1.1 to 1.3 V, $USBDV_{CC}$ = 1.1 to 1.3 V, $USBAV_{CC}$ = 1.1 to 1.3 V, PV_{CC} = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, $USBAPV_{CC}$ = 3.0 to 3.6 V, $2DGAPV_{CC}0$ = 3.0 to 3.6 V, $2DGAPV_{CC}1$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $USBAV_{SS}$ = AV_{SS} = $USBAPV_{SS}$ = $2DGAPV_{SS}0$ = $2DGAPV_{SS}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
Rise time	t_{FR}	4	—	20	ns	Figure 33.56
Fall time	t_{FF}	4	—	20	ns	
Rise/fall time lag	t_{FR}/t_{FF}	90	—	111.11	%	

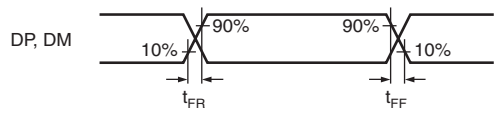


Figure 33.56 DP and DM Output Timing (Full-Speed)

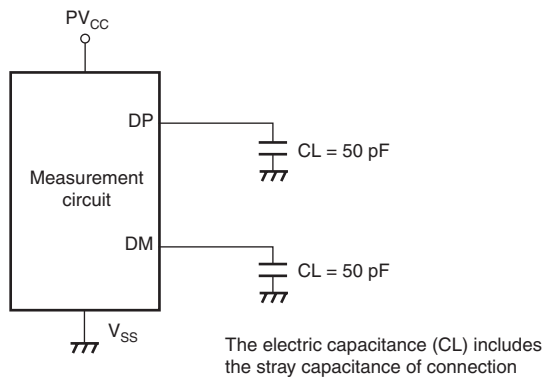


Figure 33.57 Measurement Circuit (Full-Speed)

Table 33.23 USB Transceiver Timing (High-Speed)

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = −20 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
Rise time	t_{HSR}	500	—	—	ps	Figure 33.58
Fall time	t_{HSF}	500	—	—	ps	
Output driver resistance	Z_{HSDRV}	40.5	—	49.5	Ω	

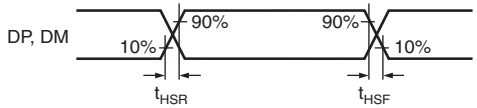


Figure 33.58 DP and DM Output Timing (High-Speed)

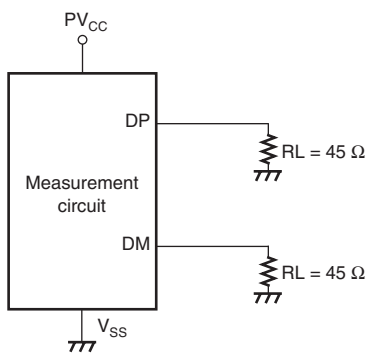


Figure 33.59 Measurement Circuit (High-Speed)

33.4.16 ATAPI Timing

Table 33.24 Timing of Register Access by PIO Transfer by ATAPI Interface

Conditions: PV_{CC} = 3.3 ± 0.3 V, T_a = –20 to 85 °C, PV_{SS} = 0 V

			Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	
Item	**Symbol**	**Conditions**	**ns**	**ns**	**ns**	**ns**	**ns**	**Figure**
Cycle time	t_0	Min.	600	383	330	180	120	Figure 33.60
Address setup time	t_1	Min.	70	50	30	30	25	
IDEIORD#/IDEIOWR# pulse width (8 bits)	t_2	Min.	290	290	290	80	70	
IDEIORD#/IDEIOWR# recovery time	t_{2i}	Min.	—	—	—	70	25	
IDEIOWR# data setup time	t_3	Min.	60	45	30	30	20	
IDEIOWR# data hold time	t_4	Min.	30	20	15	10	10	
IDEIORD# data setup time	t_5	Min.	50	35	20	20	20	
IDEIORD# data hold time	t_6	Min.	5	5	5	5	5	
IDEIORD# 3-state delay time	t_{6Z}	Max.	30	30	30	30	30	
Address hold time	t_9	Min.	20	15	10	10	10	
IDEIORDY read data valid time	t_{RD}	Min.	0	0	0	0	0	
Response time from rise of IDEIORDY to rise of IDEIORD#	t_{RR}	Min.	30	30	30	30	30	
		Max.	90	90	90	90	90	
IDEIORDY setup time	t_A	Min.	35	35	35	35	35	
IDEIORDY pulse time	t_B	Max.	1250	1250	1250	1250	1250	
Time from IDEIORDY negation to Hi-Z	t_C	Max.	5	5	5	5	5	

Table 33.25 ATAPI Interface PIO Data Transfer Timing

Conditions: PV_{CC} = 3.3 ± 0.3 V, T_a = –20 to 85 °C, PV_{SS} = 0 V

Item	Symbol	Conditions	Mode 0 ns	Mode 1 ns	Mode 2 ns	Mode 3 ns	Mode 4 ns	Figure
Cycle time	t_0	Min.	600	383	240	180	120	Figure 33.60
Address setup time	t_1	Min.	70	50	30	30	25	
IDEIORD#/IDEIOWR# pulse width (8 bits)	t_2	Min.	165	125	100	80	70	
IDEIORD#/IDEIOWR# recovery time	t_{2i}	Min.	—	—	—	70	25	
IDEIOWR# data setup time	t_3	Min.	60	45	30	30	20	
IDEIOWR# data hold time	t_4	Min.	30	20	15	10	10	
IDEIORD# data setup time	t_5	Min.	50	35	20	20	20	
IDEIORD# data hold time	t_6	Min.	5	5	5	5	5	
IDEIORD# 3-state delay time	t_{6Z}	Max.	30	30	30	30	30	
Address hold time	t_9	Min.	20	15	10	10	10	
IDEIORDY read data valid time	t_{RD}	Min.	0	0	0	0	0	
Response time from rise of IDEIORDY to rise of IDEIORD#	t_{RR}	Min.	30	30	30	30	30	
		Max.	90	90	90	90	90	
IDEIORDY setup time	t_A	Min.	35	35	35	35	35	
IDEIORDY pulse time	t_B	Max.	1250	1250	1250	1250	1250	
Time from IDEIORDY negation to Hi-Z	t_C	Max.	5	5	5	5	5	

Table 33.26 ATAPI Interface Multi-Word Transfer Timing

Conditions: PV_{CC} = 3.3 ± 0.3 V, T_a = –20 to 85 °C, PV_{SS} = 0 V

			Mode 0	Mode 1	Mode 2	
Item	**Symbol**	**Conditions**	**ns**	**ns**	**ns**	**Figure**
Cycle time	t_0	Min.	480	150	120	Figures 33.61 to 33.64
IDEIORD#/IDEIOWR# pulse width	t_D	Min.	215	80	70	
IDEIORD# data access time	t_E	Max.	150	60	50	
IDEIORD# data hold time	t_F	Min.	5	5	5	
IDEIORD# data setup time	t_G	Min.	100	30	20	
IDEIOWR# data setup time		Min.	100	30	20	
IDEIOWR# data hold time	t_H	Min.	20	15	10	
IODACK# data setup time	t_I	Min.	0	0	0	
IODACK# data hold time	t_J	Min.	20	5	5	
IDEIORD# negate pulse width	t_{KR}	Min.	50	50	25	
IDEIOWR# negate pulse width	t_{KW}	Min.	215	50	25	
IDEIORD# IODREQ delay time	t_{LR}	Max.	120	40	35	
IDEIOWR# IODREQ delay time	t_{LW}	Max.	40	40	35	
IDECS#[1:0] setup time	t_M	Min.	50	30	25	
IDECS#[1:0] hold time	t_N	Min.	15	10	10	
IODACK# 3-state delay time	t_Z	Max.	20	25	25	
DREQ negate detection time (from rise of IDEIORD#/IDEIOWR# to fall of IODREQ)	t_S	Min.	20	20	20	
		Max.	45	45	45	

Table 33.27 ATAPI Interface Ultrta DMA Transfer Timing

Conditions: PV_{CC} = 3.3 ± 0.3 V, T_a = –20 to 85 °C, PV_{SS} = 0 V

Symbol for Ultra DMA Transfer	Mode 0		Mode 1		Mode 2		
	ns		ns		ns		
	Min.	Max.	Min.	Max.	Min.	Max.	Figure
$t_{2CYCTYP}$	240	—	160	—	120	—	Figures 33.65 to 33.74
t_{CYC}	112	—	73	—	54	—	
t_{2CYC}	230	—	154	—	115	—	
t_{DS}	15	—	10	—	7	—	
t_{DH}	5	—	5	—	5	—	
t_{DVS}	70	—	48	—	30	—	
t_{DVH}	6.2	—	6.2	—	6.2	—	
t_{CS}	15	—	10	—	7	—	
t_{CH}	5	—	5	—	5	—	
t_{CVS}	70	—	48	—	31	—	
t_{CVH}	6.2	—	6.2	—	6.2	—	
t_{ZFS}	0	—	0	—	0	—	
t_{DZFS}	70	—	48	—	31	—	
t_{FS}	—	230	—	200	—	170	
t_{LI}	0	150	0	150	0	150	
t_{MLI}	20	—	20	—	20	—	
t_{UI}	0	—	0	—	0	—	
t_{AZ}	—	10	—	10	—	10	
t_{ZAH}	20	—	20	—	20	—	
t_{ZAD}	0	—	0	—	0	—	
t_{ENV}	20	70	20	70	20	70	
t_{RFS}	—	75	—	70	—	60	
t_{RP}	160	—	125	—	100	—	
t_{IORDYZ}	—	20	—	20	—	20	
t_{ZIORDY}	0	—	0	—	0	—	
t_{ACK}	20	—	20	—	20	—	
t_{SS}	50	—	50	—	50	—	

Table 33.28 Symbols for ATAPI Interface Ultrta DMA Transfer Timing

Symbol	Meaning
$t_{2CYCTYP}$	Typical average two-cycle time
t_{CYC}	Cycle time
t_{2CYC}	Minimum two-cycle time
t_{DS}	Data setup time (receiver)
t_{DH}	Data hold time (receiver)
t_{DVS}	Data setup time (transmitter)
t_{DVH}	Data hold time (transmitter)
t_{CS}	CRC data setup time (receiver)
t_{CH}	CRC data hold time (receiver)
t_{CVS}	CRC data setup time (transmitter)
t_{CVH}	CRC data hold time (transmitter)
t_{ZFS}	Setup time (from the point at which the strobe signal is driven to the first assertion of the strobe signal) (transmitter)
t_{DZFS}	Setup time (from the point at which the data signal is driven to the first assertion of the strobe signal) (transmitter)
t_{FS}	First strobe time
t_{LI}	Limited interlock time
t_{MLI}	Minimum limited interlock time
t_{UI}	Unlimited interlock time
t_{AZ}	Output release time
t_{ZAH}	Output delay time
t_{ZAD}	Output assert/negate time (from release timing)
t_{ENV}	Envelope time
t_{RFS}	Final strobe time
t_{RP}	STOP assertion or DMARQ negation time
t_{IORDYZ}	IORDY release time
t_{ZIORDY}	Strobe driven time
t_{ACK}	DMACK# setup/hold time
t_{SS}	Strobe stop time

Table 33.29 ATAPI Interface DIRECTION Timing

Conditions: PV_{CC} = 3.3 ± 0.3 V, T_a = –20 to 85 °C, PV_{SS} = 0 V

		Mode 0		Mode 1		Mode 2		Mode 3		Mode 4		
		ns		ns		ns		ns		ns		
Item	Symbol	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Figure
DIRECTION fall delay time in PIO write	$t_{DIRECTION\_WF}$	53	64	38	49	23	34	23	34	23	34	Figure 33.75
DIRECTION rise delay time in PIO write	$t_{DIRECTION\_WR}$	71	80	71	80	71	80	71	80	71	80	
DIRECTION fall delay time in multiword DMA data-out	$t_{MDIRECTION\_F}$	–18	–9	–18	–9	–18	–9	—	—	—	—	Figure 33.77
DIRECTION rise delay time in multiword DMA data-out	$t_{MDIRECTION\_R}$	11	20	11	20	11	20	—	—	—	—	
DIRECTION fall delay time on CRC transmission in ultra DMA data-in	$t_{UDIRECTION\_F}$ (CRC)	130	139	100	109	85	94	—	—	—	—	Figures 33.79, 33.80
DIRECTION rise delay time on CRC transmission in ultra DMA data-in	$t_{UDIRECTION\_R}$ (CRC)	26	35	26	35	26	35	—	—	—	—	
DIRECTION fall delay time in ultra DMA data-out	$t_{UDIRECTION\_F}$	54	64	54	64	54	64	—	—	—	—	Figure 33.81
DIRECTION rise delay time in ultra DMA data-out	$t_{UDIRECTION\_R}$	71	80	71	80	71	80	—	—	—	—	Figures 33.82, 33.83
Time until IDED data bus is turned on after fall of DIRECTION	t_{DON}	24	34	24	34	24	34	24	34	24	34	Figures 33.75, 33.77, 33.79 to 33.83
Time until rise of DIRECTION after IDED data bus is turned off	t_{DOFF}	11	19	11	19	11	19	11	19	11	19	

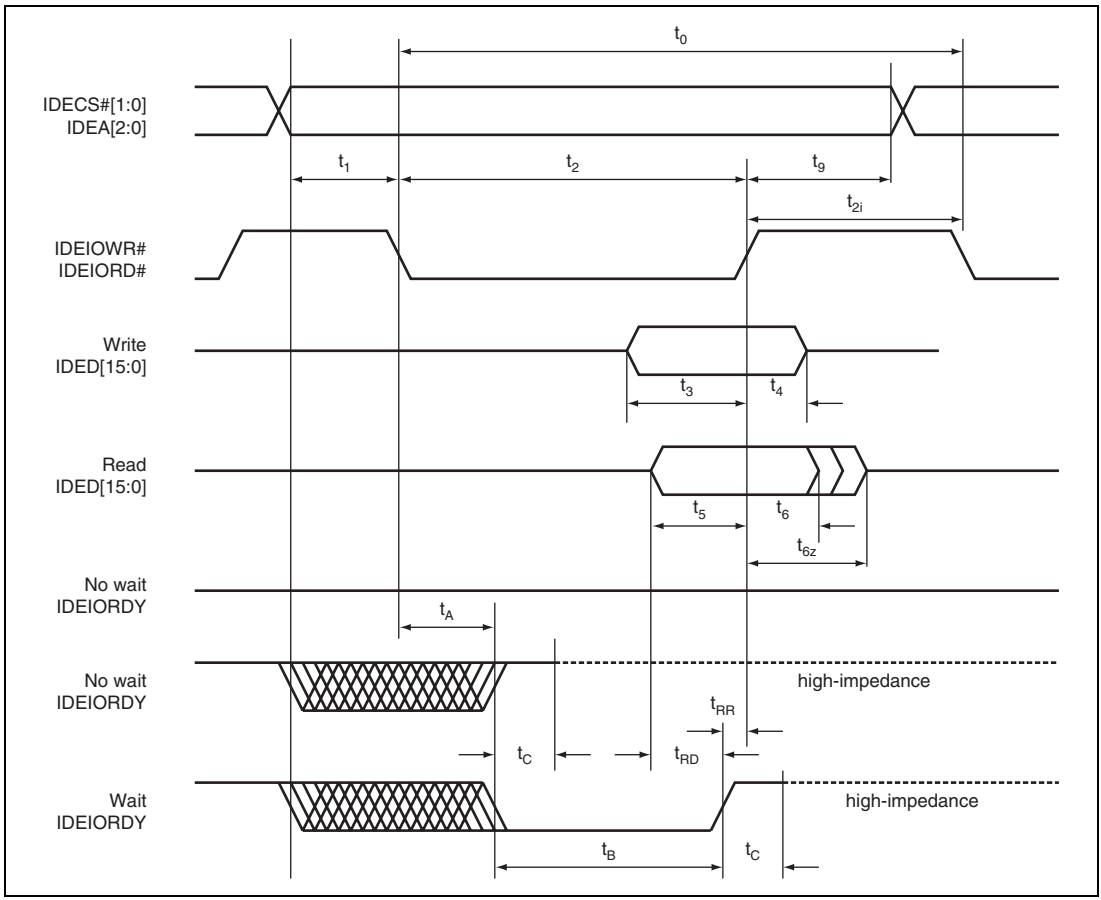


Figure 33.60 Register Transfer and PIO Data Transfer to/from Device

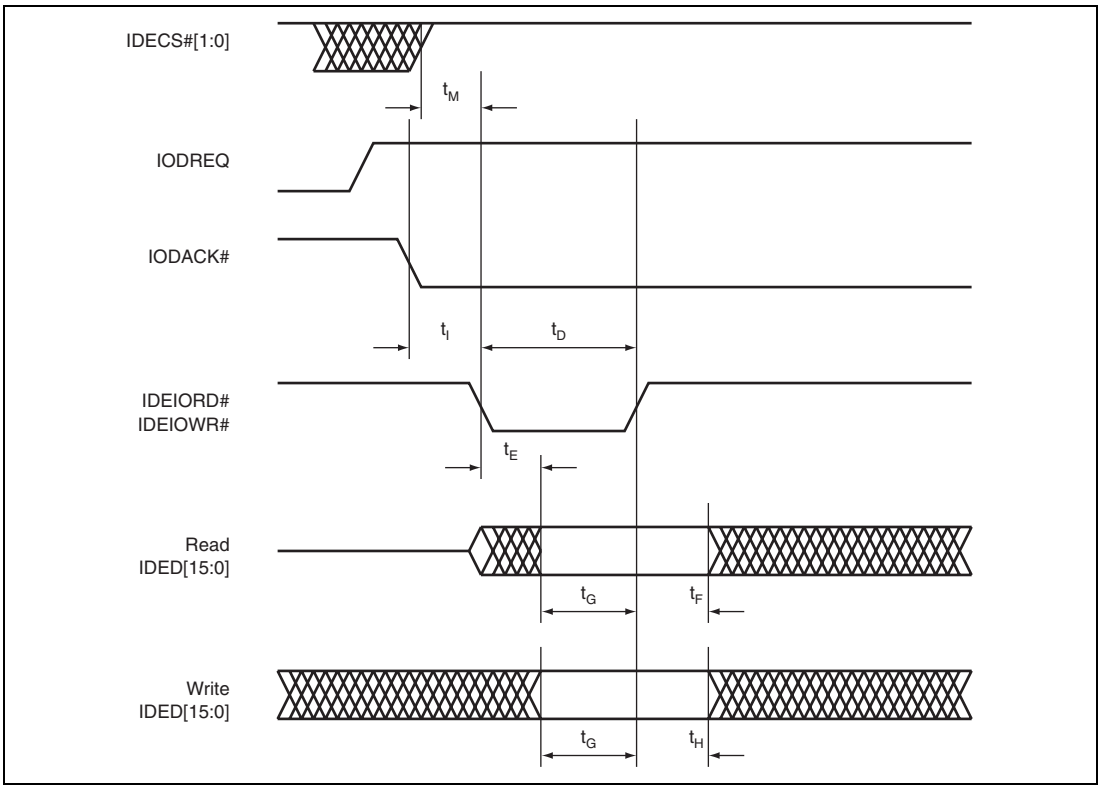


Figure 33.61 Initiating Multiword DMA Data Burst

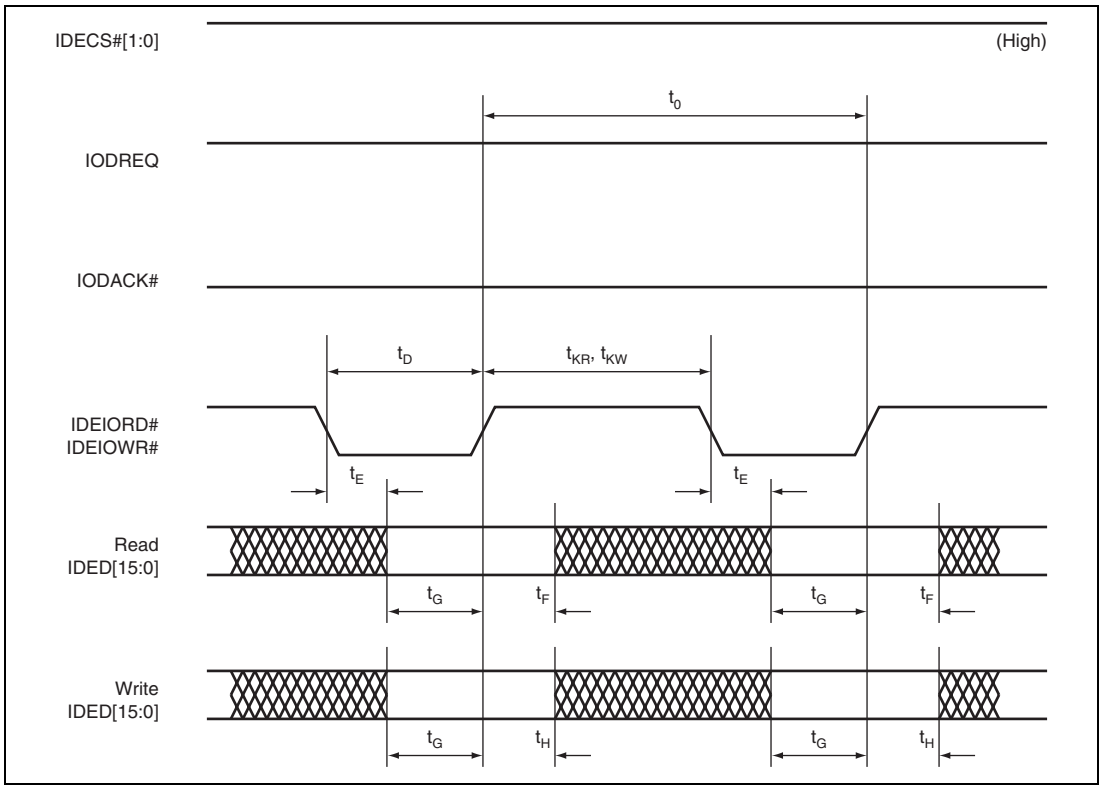


Figure 33.62 Sustaining Multiword DMA Data Burst

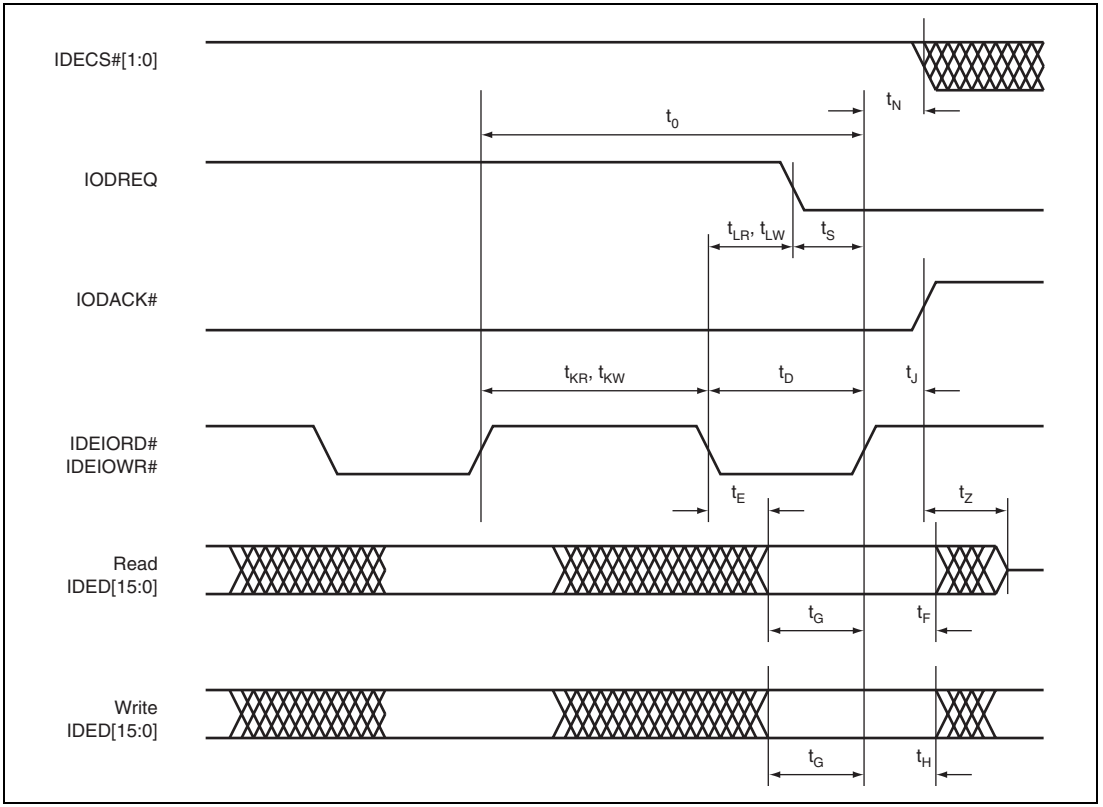


Figure 33.63 Device Terminating Multiword DMA Data Burst

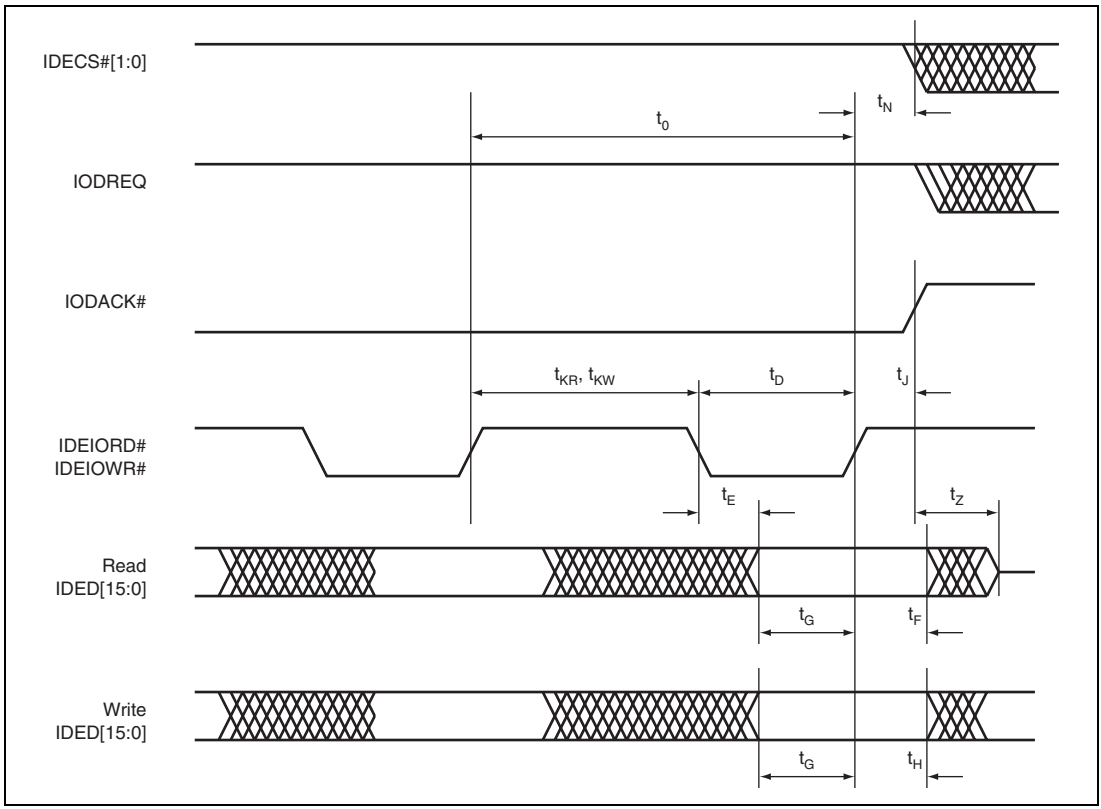


Figure 33.64 Host Terminating Multiword DMA Data Burst

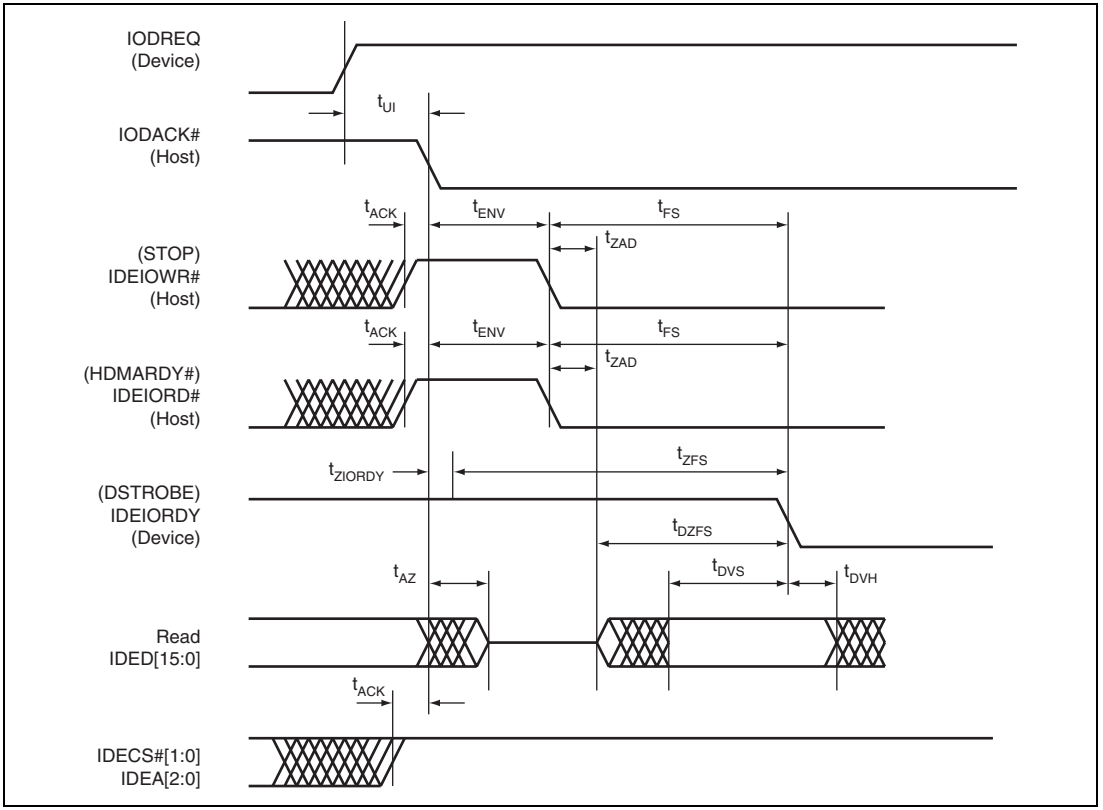


Figure 33.65 Initiating Ultra DMA Data-In Burst

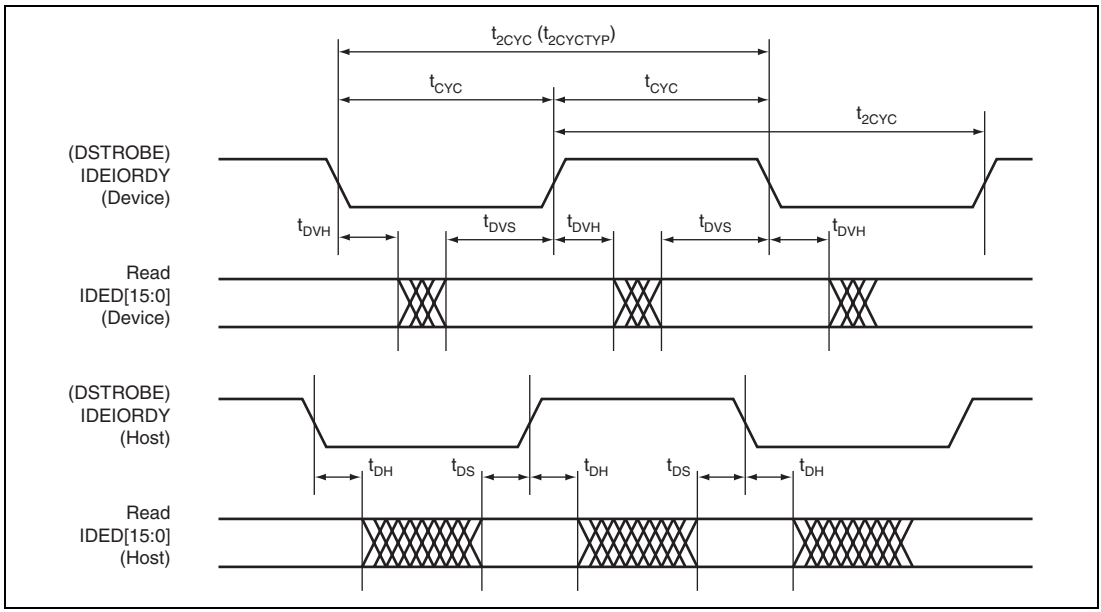


Figure 33.66 Sustained Ultra DMA Data-In Burst

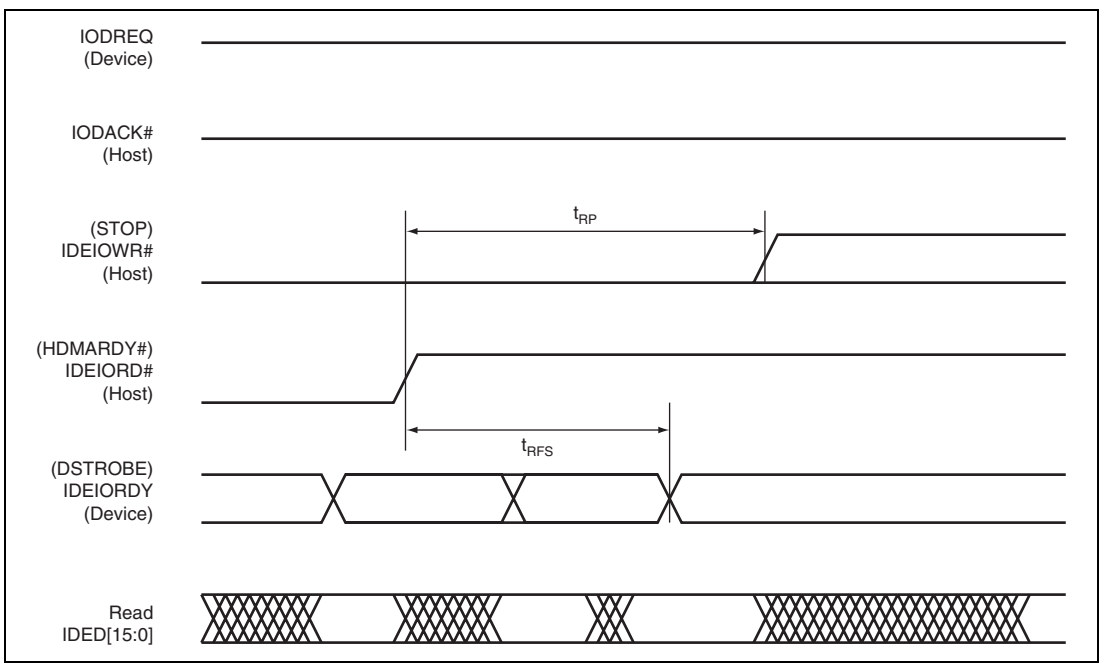


Figure 33.67 Host Pausing Ultra DMA Data-In Burst

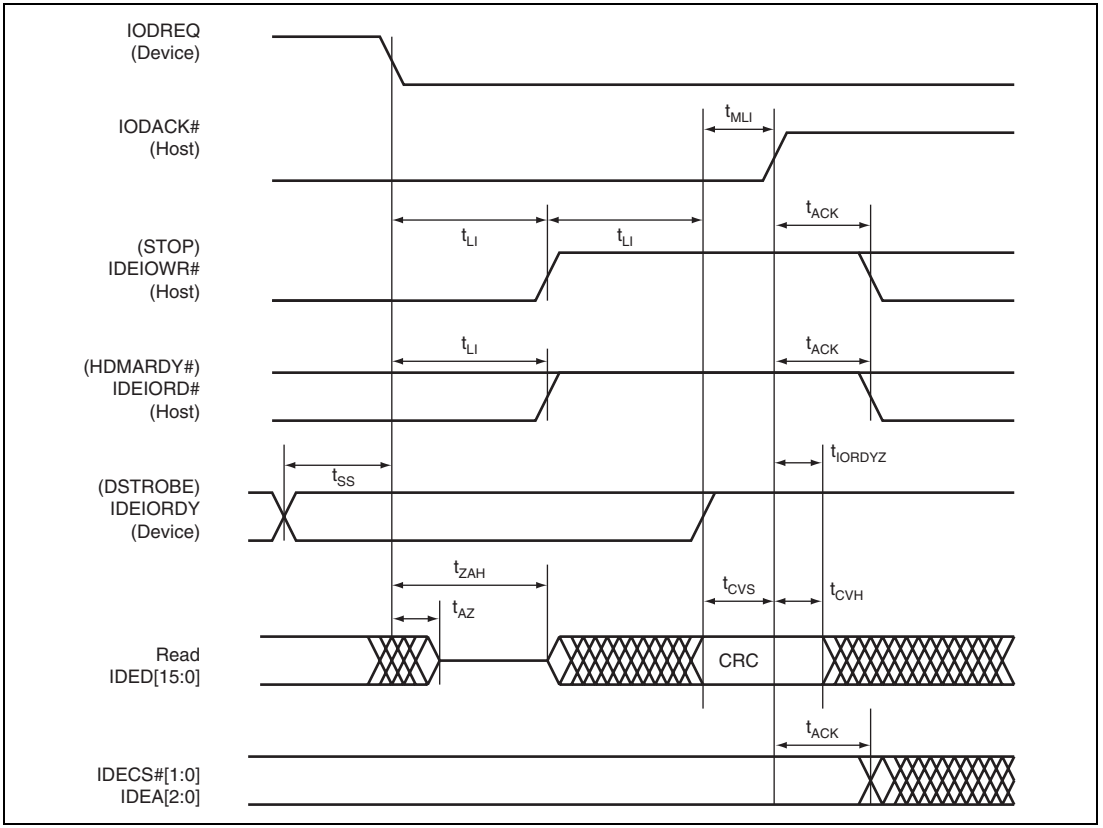


Figure 33.68 Device Terminating Ultra DMA Data-In Burst

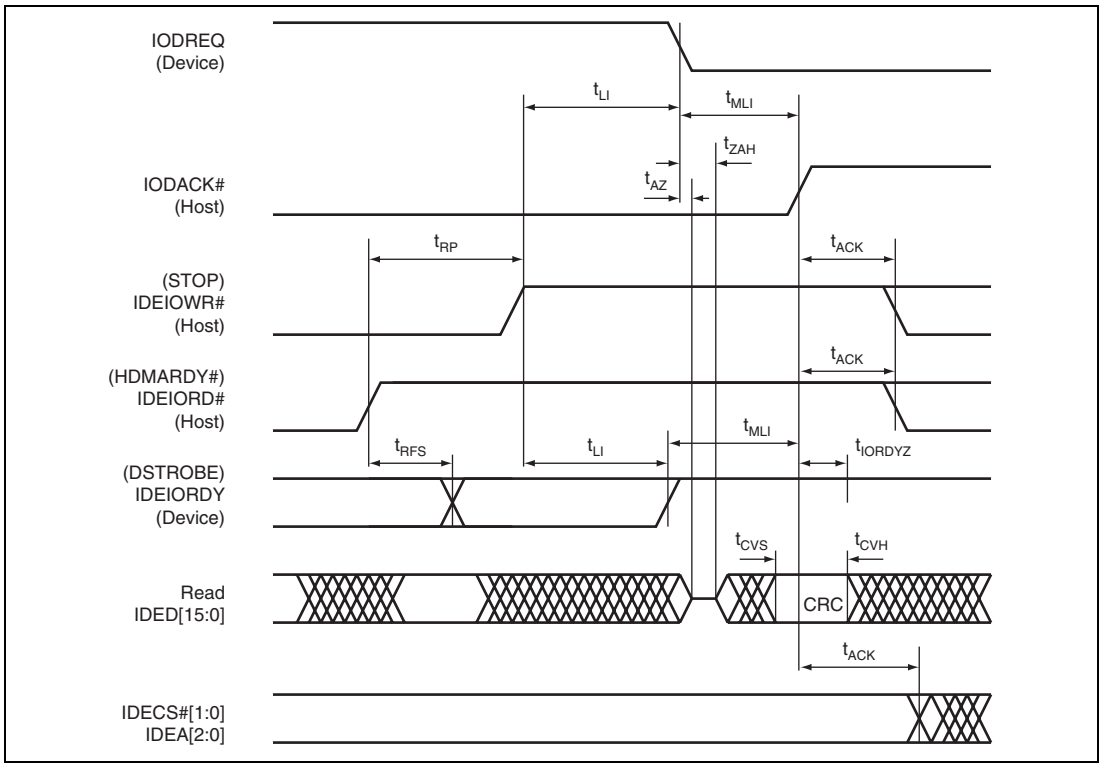


Figure 33.69 Host Terminating Ultra DMA Data-In Burst

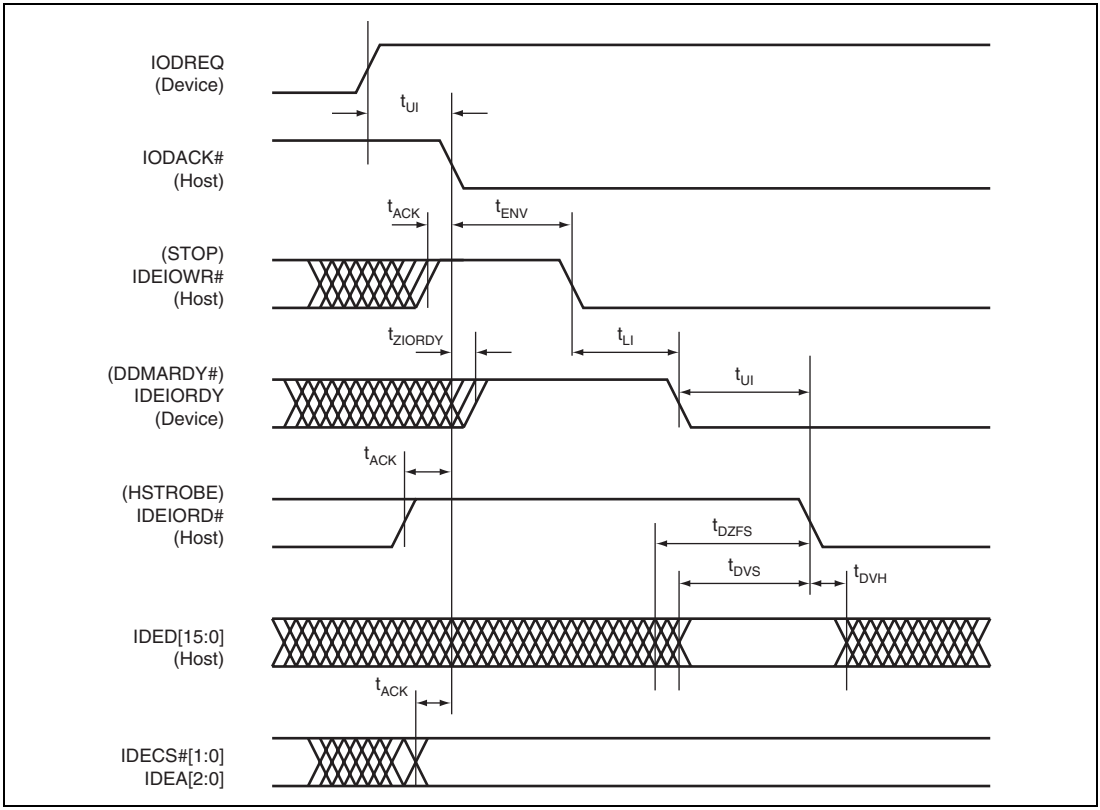


Figure 33.70 Initiating Ultra DMA Data-Out Burst

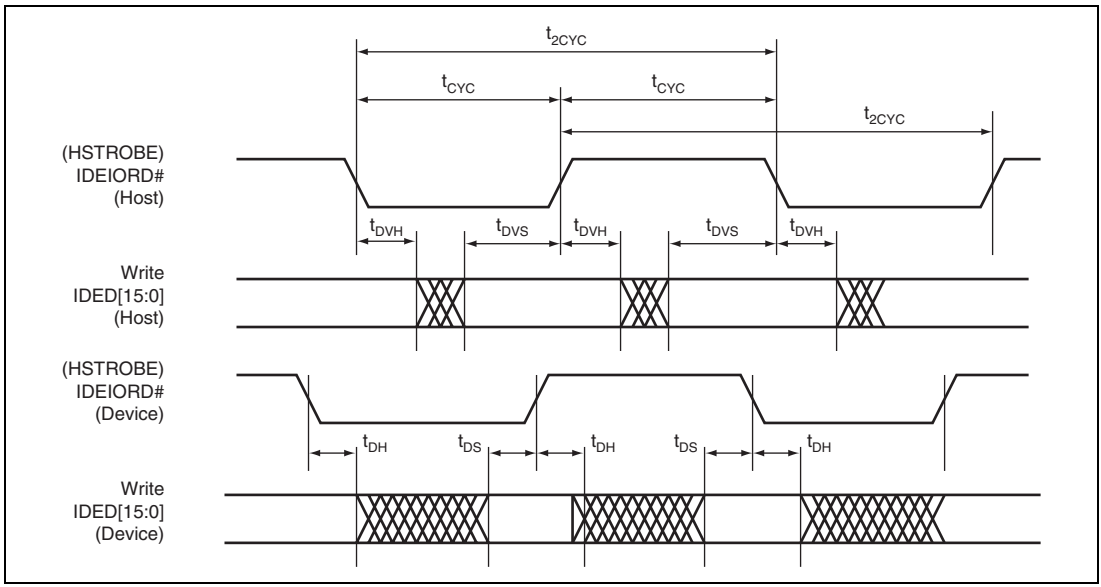


Figure 33.71 Sustained Ultra DMA Data-Out Burst

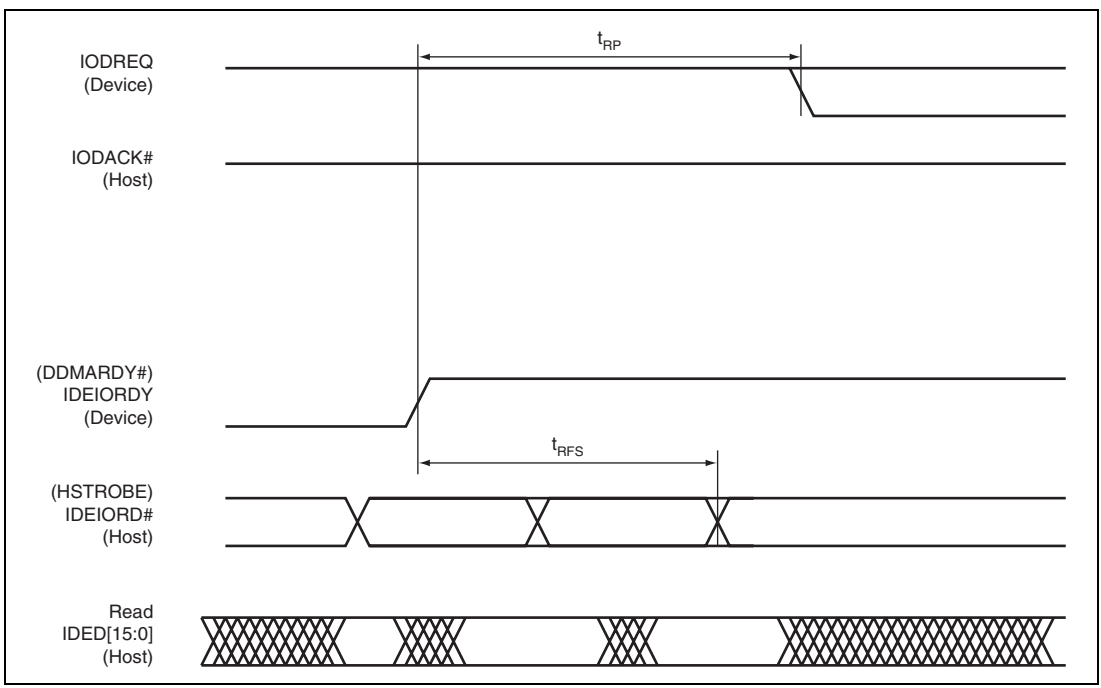


Figure 33.72 Device Pausing Ultra DMA Data-Out Burst

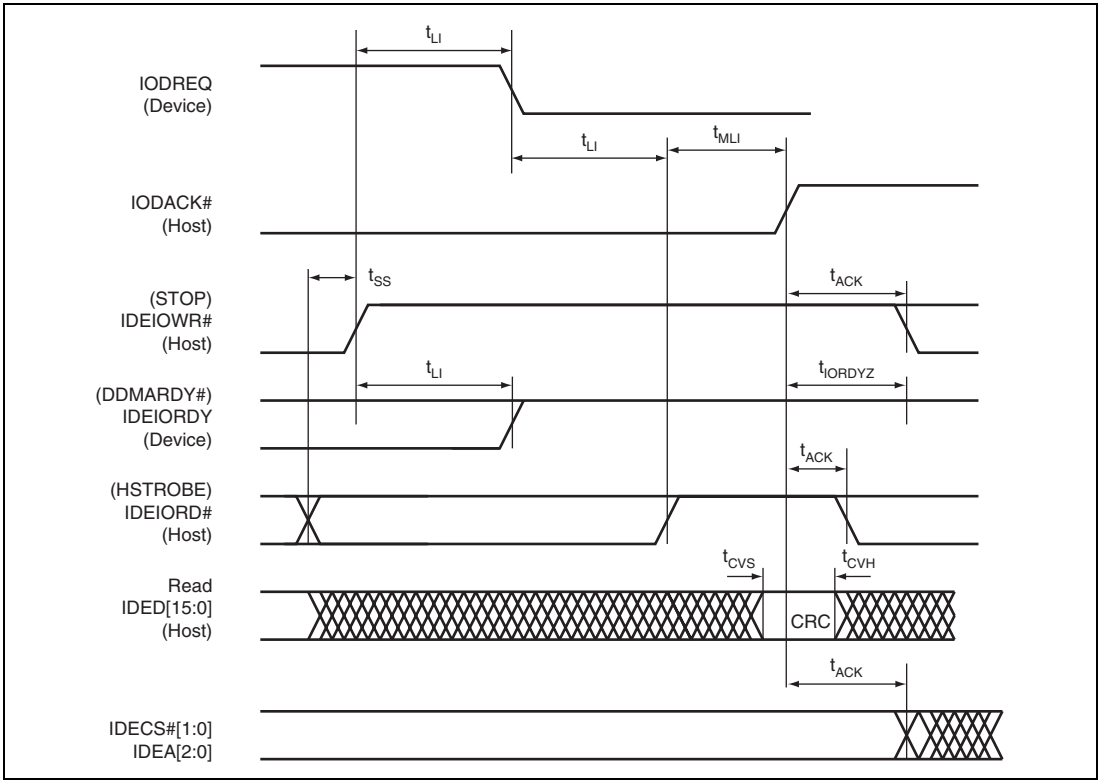


Figure 33.73 Host Terminating Ultra DMA Data-Out Burst

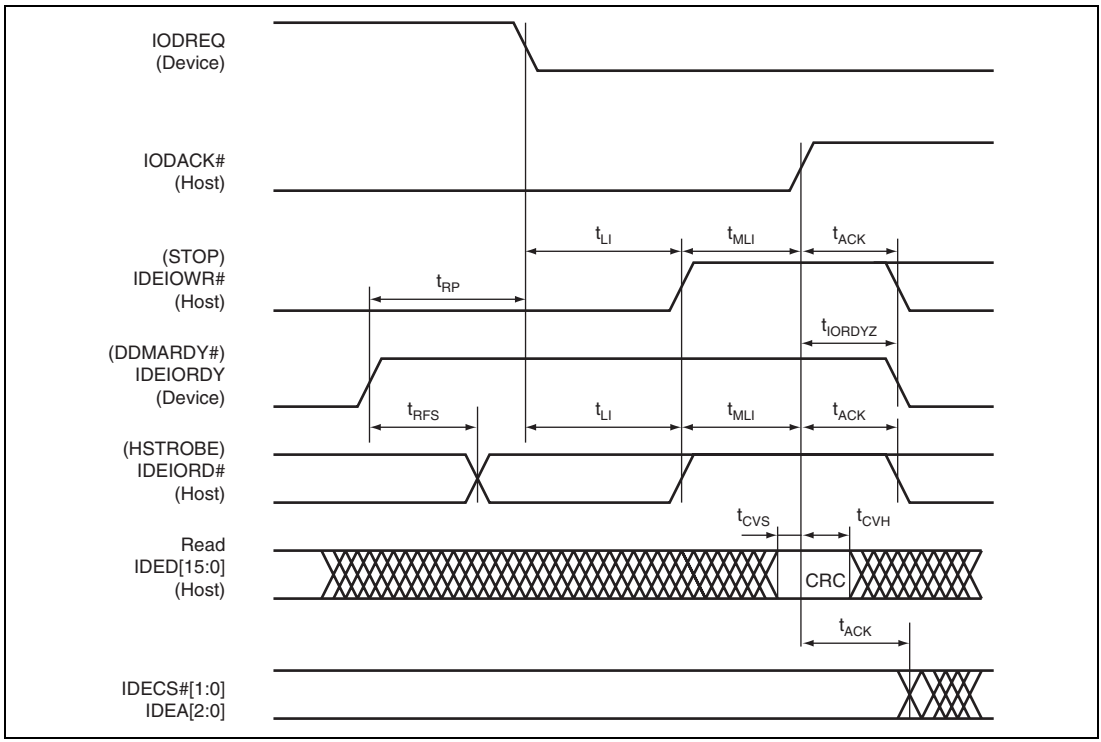


Figure 33.74 Device Terminating Ultra DMA Data-Out Burst

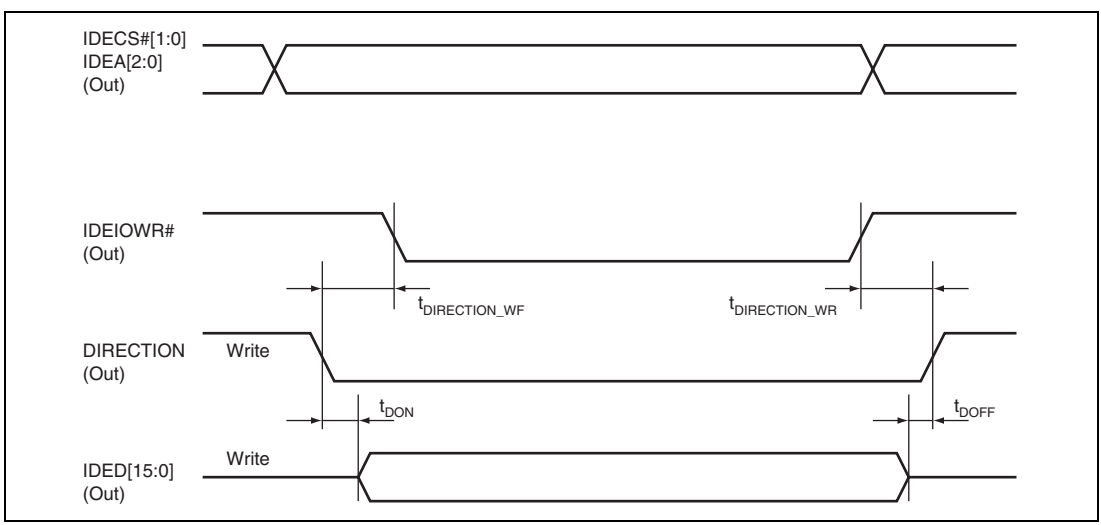


Figure 33.75 PIO Data Transfer to Device (DIRECTION)

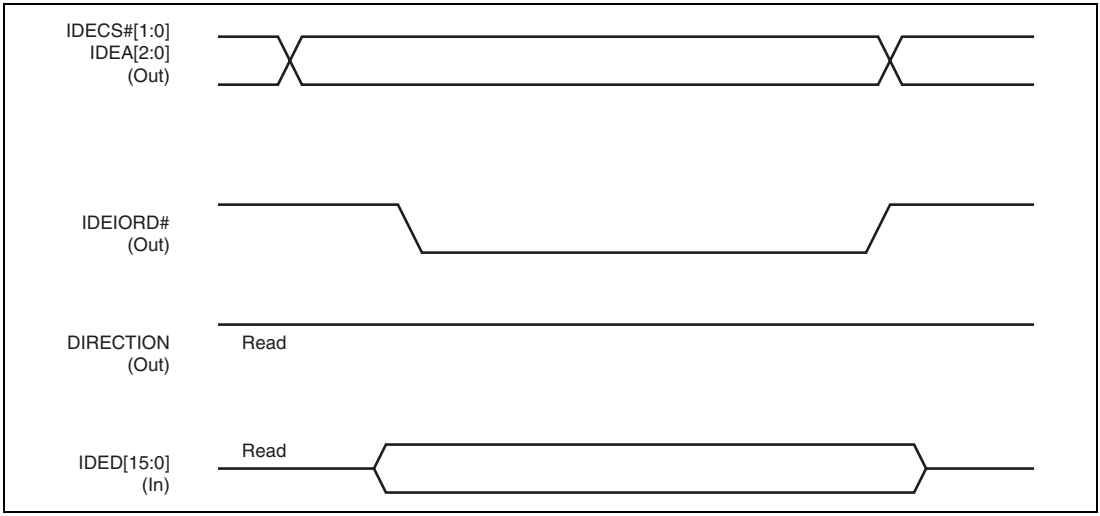


Figure 33.76 PIO Data Transfer from Device (DIRECTION)

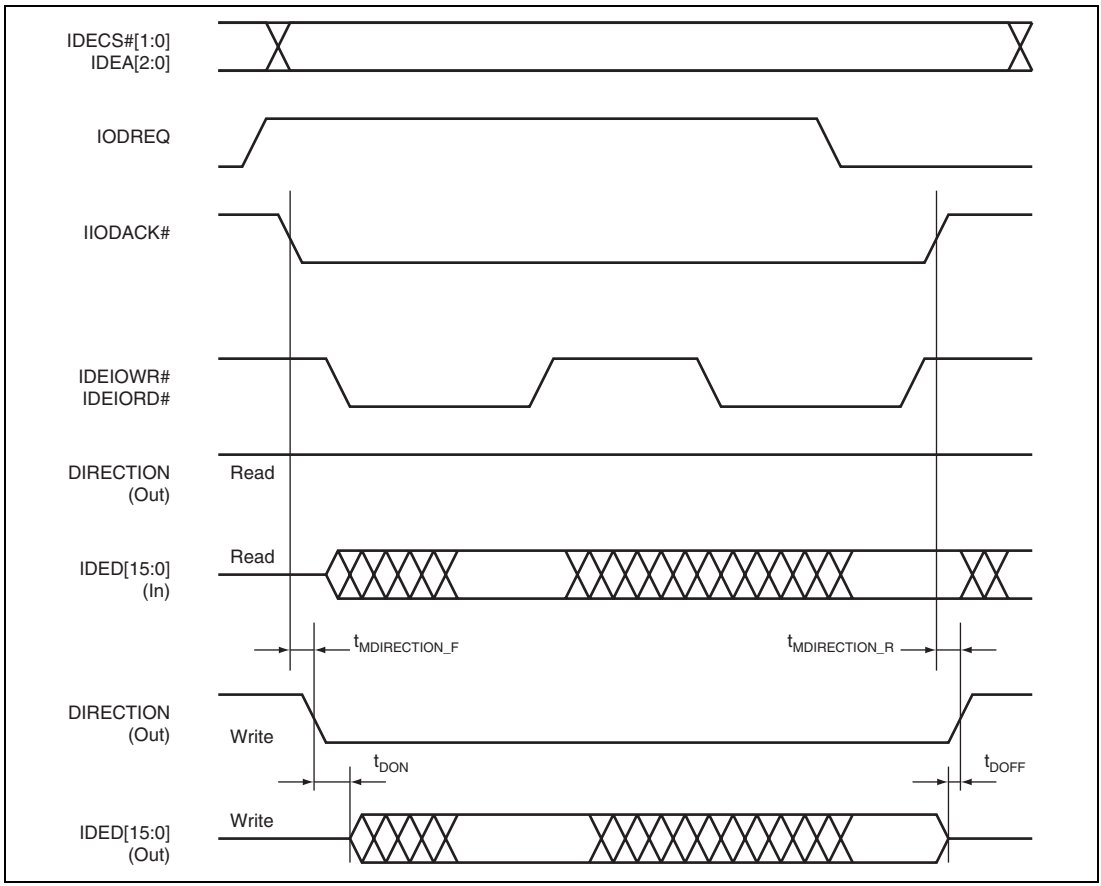


Figure 33.77 Multiword DMA Data Transfer (DIRECTION)

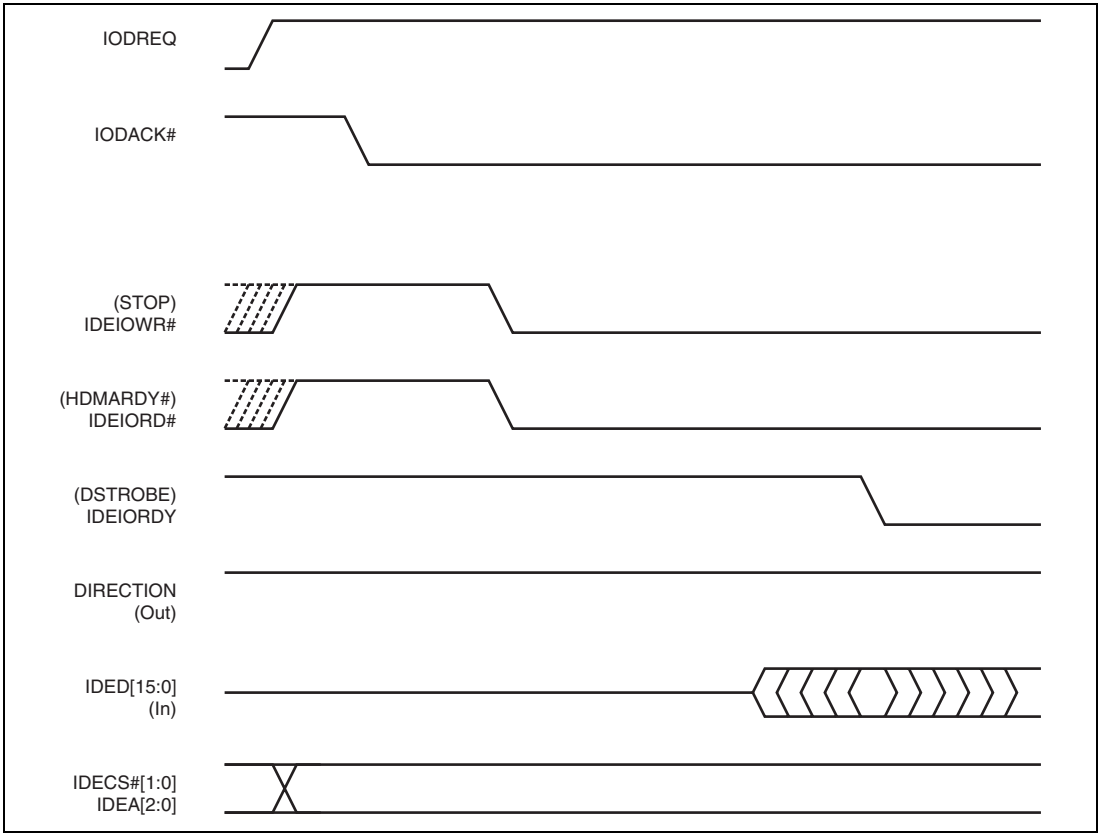


Figure 33.78 Initiating Ultra DMA Data-In Burst (DIRECTION)

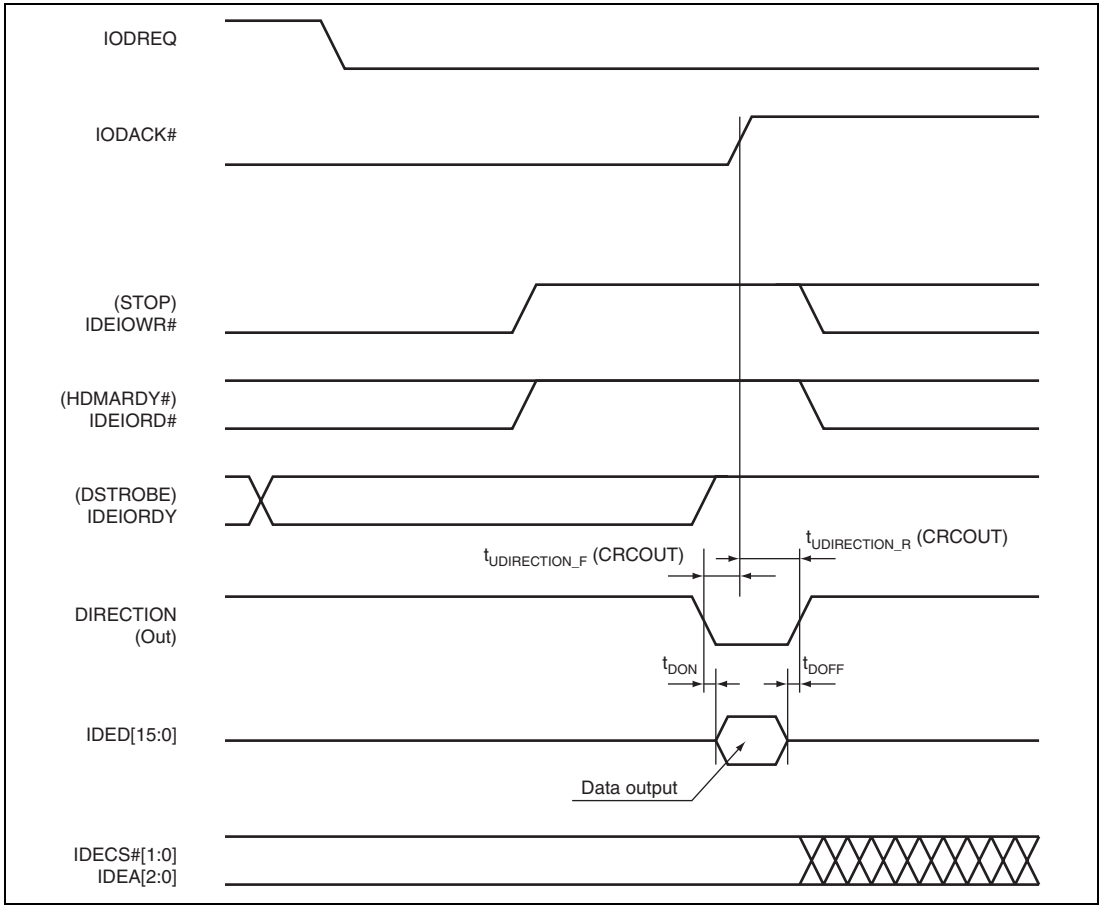


Figure 33.79 Device Terminating Ultra DMA Data-In Burst (DIRECTION)

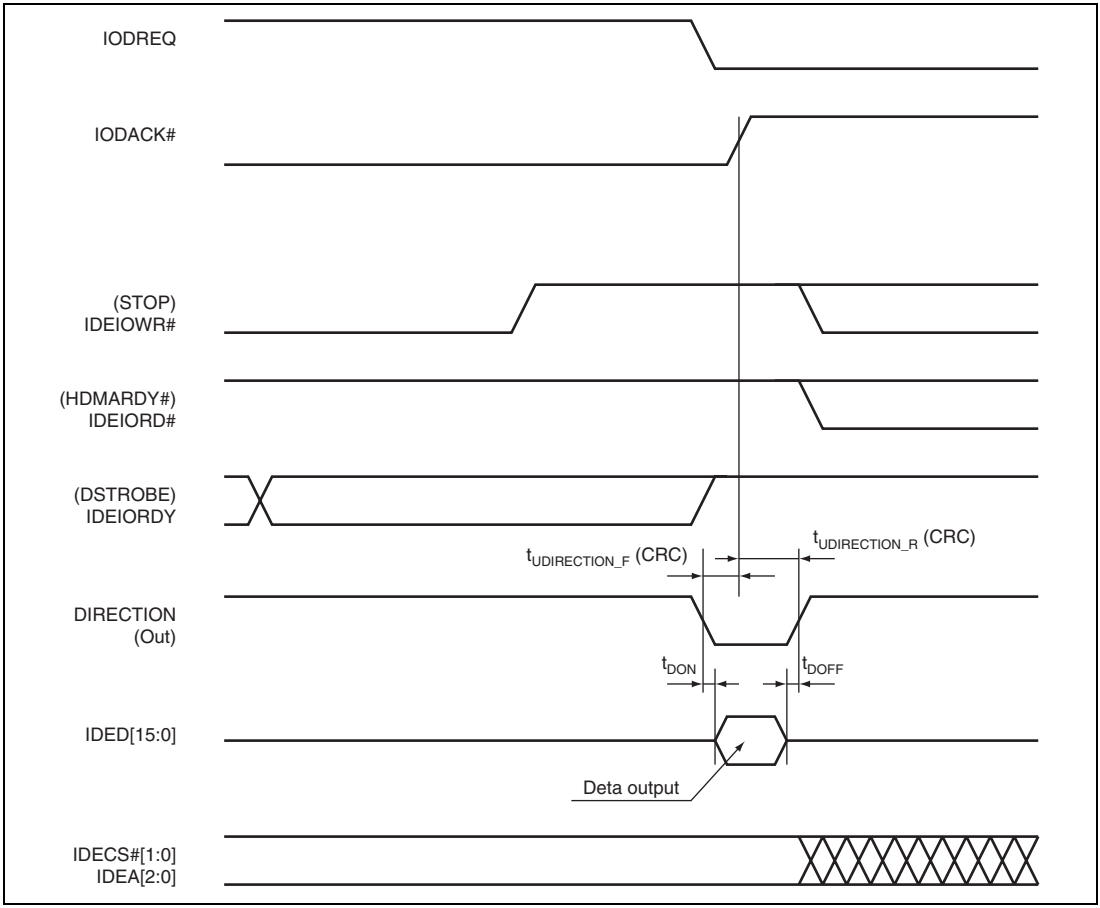


Figure 33.80 Host Terminating Ultra DMA Data-In Burst (DIRECTION)

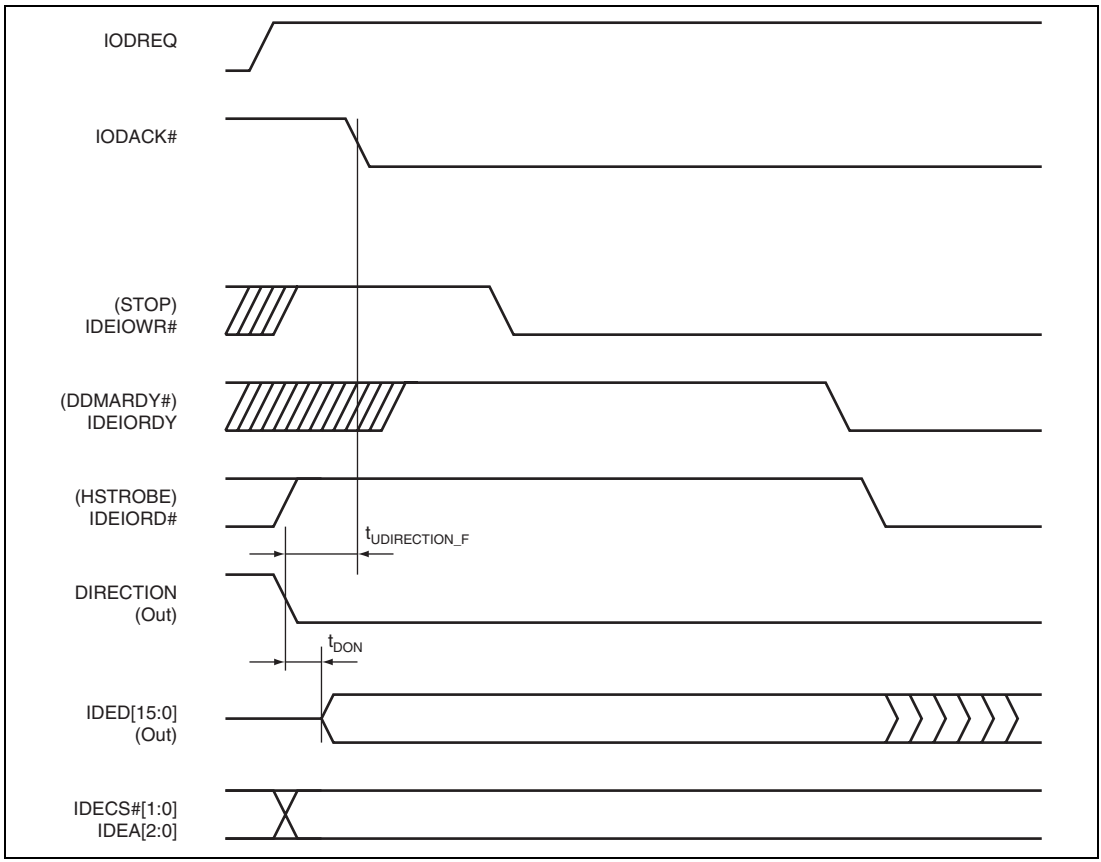


Figure 33.81 Initiating Ultra DMA Data-Out Burst (DIRECTION)

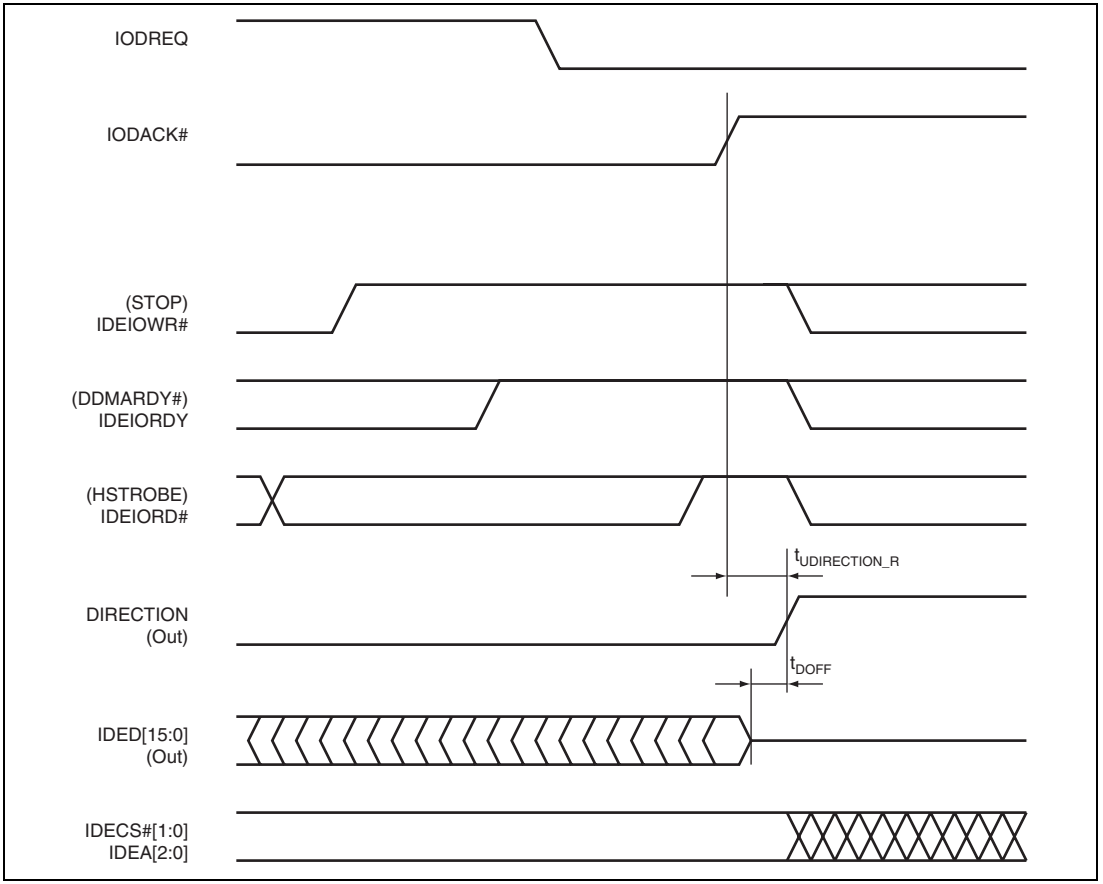


Figure 33.82 Host Terminating Ultra DMA Data-Out Burst (DIRECTION)

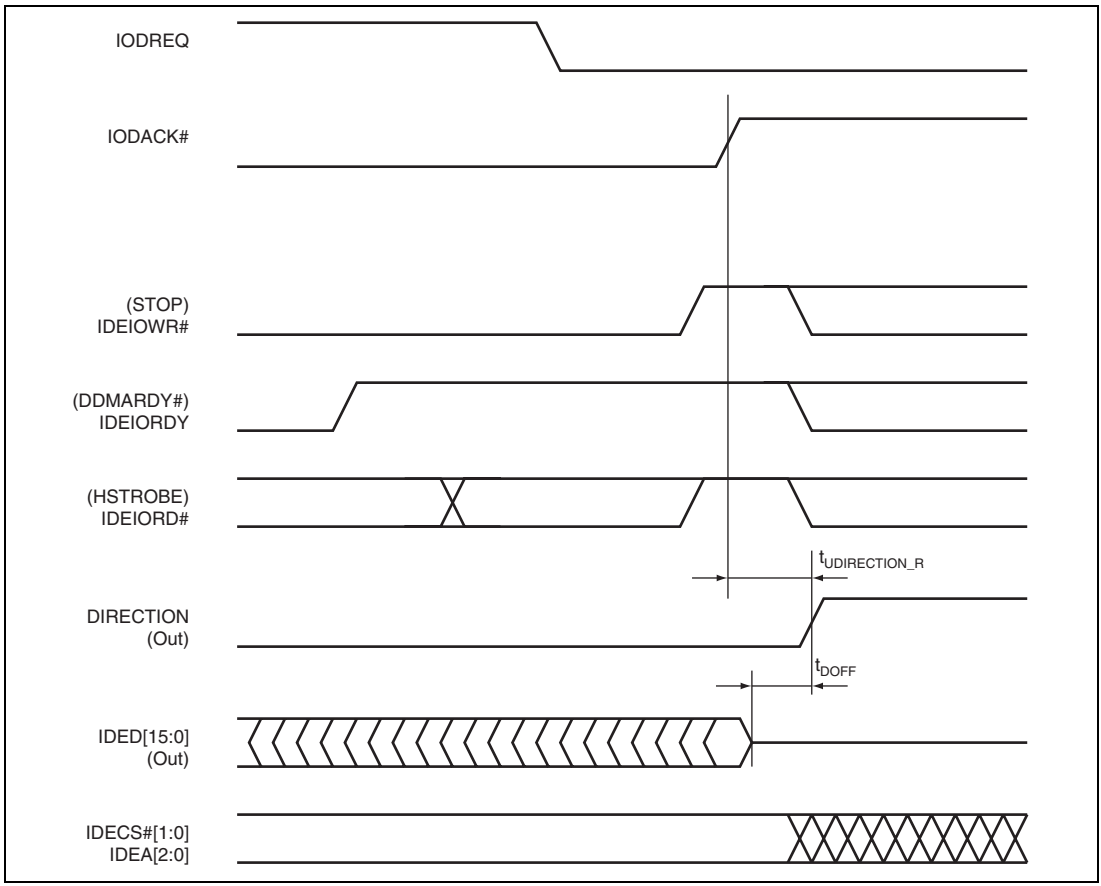


Figure 33.83 Device Terminating Ultra DMA Data-Out Burst (DIRECTION)

33.4.17 2DG Timing

Table 33.30 2DG Video Input Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = −20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
VICLK clock input cycle time	t_{VCKcyc}	34	40	ns	Figure 33.84
Input data setup time	t_{VS}	5	—	ns	
Input data hold time	t_{VH}	3	—	ns	

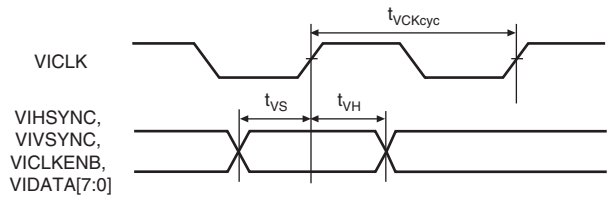


Figure 33.84 Video Input Timing

Table 33.31 2DG Display Output Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
DCLKIN clock input cycle time	t_{DCKcyc}	83	200	ns	Figure 33.85
DCLKIN clock input low-level pulse width	t_{DCKL}	34	—	ns	
DCLKIN clock input high-level pulse width	t_{DCKH}	34	—	ns	
DCLKIN clock input rise time	t_{DCKr}	—	3	ns	
DCLKIN clock input fall time	t_{DCKf}	—	3	ns	
Output data delay time	t_{DD}	0	15	ns	Figure 33.86

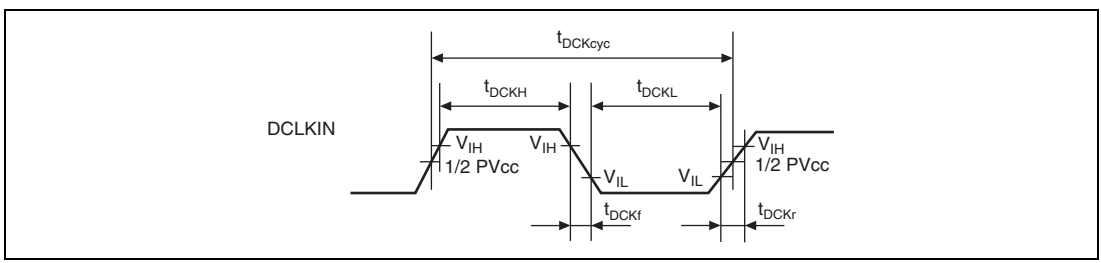


Figure 33.85 DCLKIN Clock Input Timing

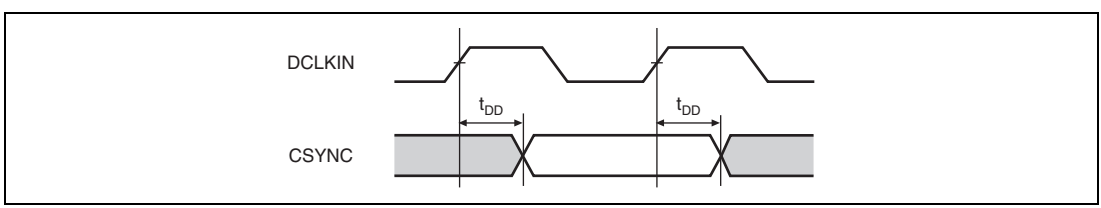


Figure 33.86 Display Output Timing

Table 33.32 VIDEO OUT D/A Converter Characteristics

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, RL = 180 Ω, R_{ext} = 5.23 kΩ, T_a = –20 to 85 °C

Item	Min.	Typ.	Max.	Unit	Figure
Resolution	6	6	6	bits	
Differential linearity error	—	±0.5	±1.0	LSB	
Integral linearity error	—	±1.5	±3.0	LSB	

33.4.18 I/O Port Timing

Table 33.33 I/O Port Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
Output data delay time	t_{PORTD}	—	100	ns	Figure 33.87
Input data setup time	t_{PORTS}	100	—		
Input data hold time	t_{PORTH}	100	—		

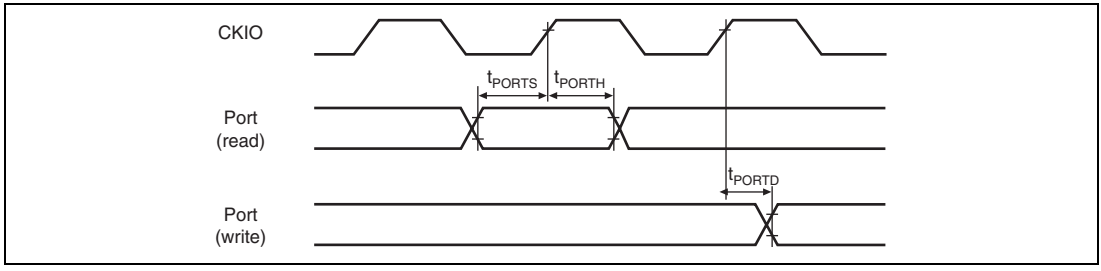


Figure 33.87 I/O Port Timing

33.4.19 H-UDI Timing

Table 33.34 H-UDI Timing

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
TCK cycle time	t_{TCKcyc}	50*	—	ns	Figure 33.88
TCK high pulse width	t_{TCKH}	0.4	0.6	t_{TCKcyc}	
TCK low pulse width	t_{TCKL}	0.4	0.6	t_{TCKcyc}	
TDI setup time	t_{TDIS}	10	—	ns	Figure 33.89
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}	—	16	ns	
Capture register setup time	t_{CAPTS}	10	—	ns	Figure 33.90
Capture register hold time	t_{CAPTH}	10	—	ns	
Update register delay time	$t_{UPDATED}$	—	20	ns	

Note: * Should be greater than the peripheral clock (Pϕ) cycle time.

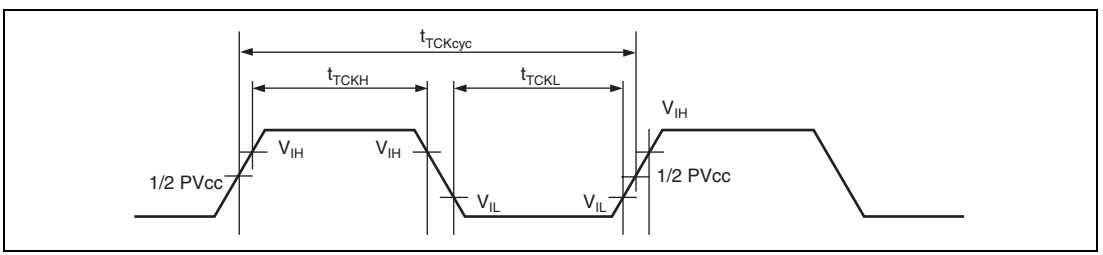


Figure 33.88 TCK Input Timing

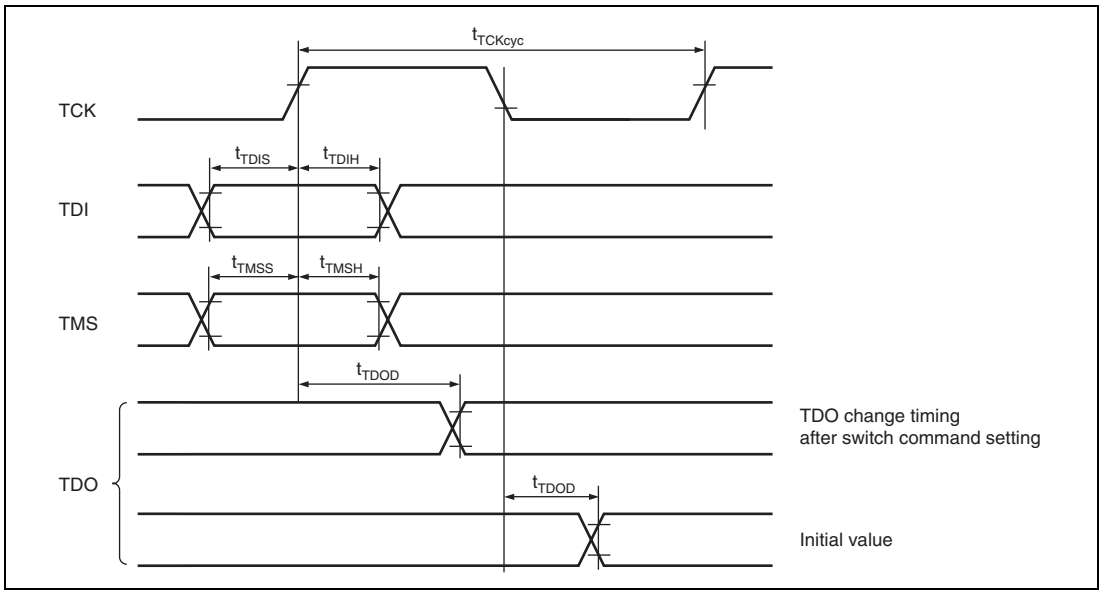


Figure 33.89 H-UDI Data Transfer Timing

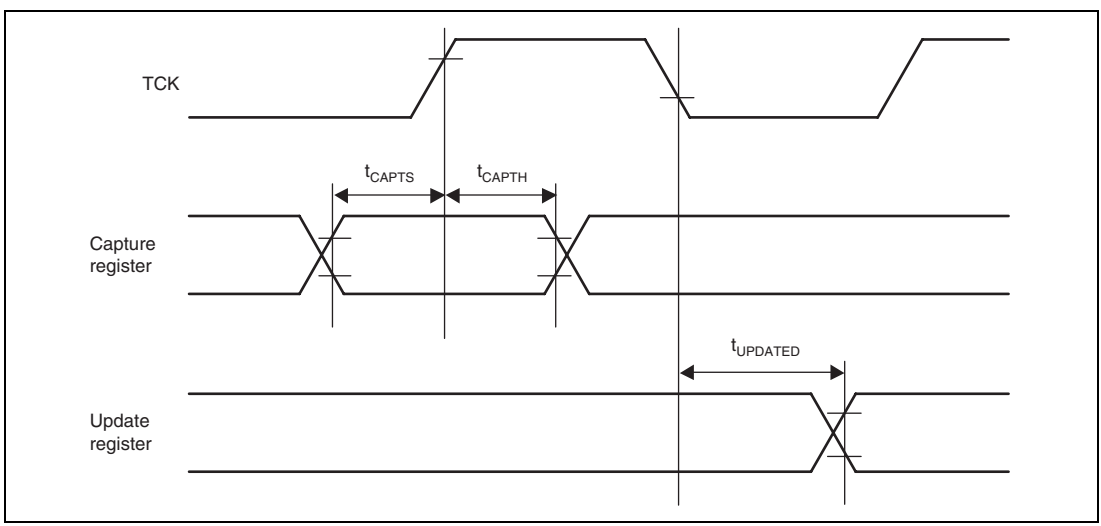
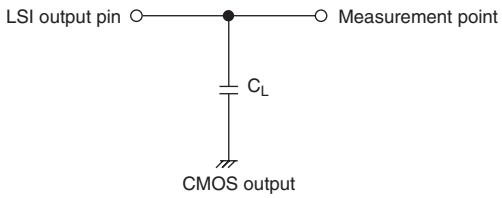


Figure 33.90 Boundary Scan Input/Output Timing

33.4.20 AC Characteristics Measurement Conditions

- I/O signal reference level: $PV_{CC}/2$ (PV_{CC} = 3.0 to 3.6 V, V_{CC} = 1.1 to 1.3 V)
- Input pulse level: V_{SS} to 3.0 V (where $\overline{RES}$, $\overline{MRES}$, NMI, MD0, MD_CLK1, MD_CLK0, $\overline{ASEMD}$, $\overline{TRST}$, and Schmitt trigger input pins are within V_{SS} to PV_{CC}.)
- Input rise and fall times: 1 ns



Note: C_L is the total value that includes the capacitance of measurement tools. The capacitance of each pin is set as 30 pF.

Figure 33.91 Output Load Circuit

33.5 A/D Converter Characteristics

Table 33.35 A/D Converter Characteristics

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = –20 to 85 °C

Item	Min.	Typ.	Max.	Unit
Resolution	10	10	10	bits
Conversion time	3.9	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.0*	LSB
Offset error	—	—	±2.0*	LSB
Full-scale error	—	—	±2.0*	LSB
Quantization error	—	—	±0.5*	LSB
Absolute accuracy	—	—	±4.0	LSB

Note: * Reference values

33.6 D/A Converter Characteristics

Table 33.36 lists the D/A converter characteristics.

Table 33.36 D/A Converter Characteristics

Conditions: V_{cc} = $PLLV_{cc}$ = 1.1 to 1.3 V, $USBDV_{cc}$ = 1.1 to 1.3 V, $USBAV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, $2DGAPV_{cc}0$ = 3.0 to 3.6 V, $2DGAPV_{cc}1$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = $USBAV_{ss}$ = AV_{ss} = $USBAPV_{ss}$ = $2DGAPV_{ss}0$ = $2DGAPV_{ss}1$ = 0 V, T_a = −20 to 85 °C

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	8	8	8	bits	
Conversion time	10	—	—	µs	Load capacitance 20 pF
Absolute accuracy	—	±2.0	±3.0	LSB	Load resistance 2 MΩ
	—	—	±2.5	LSB	Load resistance 4 MΩ

33.7 Usage Notes

Place a multilayer ceramic capacitor as a bypass capacitor per a pair of power supply pins. Place the bypass capacitor as close as possible to the power supply pins in this LSI. The capacity values of the capacitor are 0.1 μF to 0.33 μF (recommended values). For the capacitor related to crystal oscillation, see section 5.8, Notes on Board Design.

Table 33.37 shows the combinations of external capacitor.

Table 33.37 Combinations of External Capacitor

Power Supply		**Ground**	
Pin No.	**Pin Name**	**Pin No.**	**Pin Name**
B20, C4, C5, C11, C19, D5, D11, D18, E17, E18, K3, K4, L17, L18, T3, T4, U3, U10, U16, V2, V10, V16, V17, W1	V_{cc}	A1, B2, C3, D4, J9, J10, J11, J12, K9, K10, K11, K12, L9, L10, L11, L12, M9, M10, M11, M12, M19, N17, N18, N19, P17, P18, P19, U17, U20, V18, W19, Y20	V_{ss}
A20, B19, C10, C17, C18, D3, D10, D17, E3, E4, K17, K18, L3, L4, M17, M18, T17, T18, U4, U11, U18, V3, V4, V11, W2, Y1	PV_{cc}		
T20	$USBDV_{cc}$		
Y3	$PLLV_{cc}$	Y2	$PLLV_{ss}$
Y16	AV_{cc}	Y15	AV_{ss}
W16	AV_{ref}		
U19	$USBAV_{cc}$	V19	$USBAV_{ss}$
R18	$USBAPV_{cc}$	T19	$USBAPV_{ss}$
U15	$2DGAPV_{cc}0$	U14	$2DGAPV_{ss}0$
V15	$2DGAPV_{cc}1$	V14	$2DGAPV_{ss}1$

Appendix

A. Pin States

Table A.1 shows the state of pin function in each operating mode. For input/output pin function, the input buffer state is shown in the table above and the output buffer state is in the table below.

Table A.1 Pin States

Pin Function				Pin State						
						Pin State Retained*[2]			Power-Down State	
							Other than Power-On Reset			
Type	Pin Name			Other than Power-On Reset	Power-On Reset*[1]	Power-On Reset*[3]	CS0KEE PE=1*[12]	CS0KEE PE=0*[12]	Deep Standby Mode*[4]	Software Standby Mode
Clock	EXTAL*[5]	Clock operation mode	0, 1	I	I	I			Z	Z
	XTAL*[5]			O	O	O			L	L
	CKIO	Clock operation mode	0, 1, 3	O/Z*[6]	O	O+/Z+*[6]		O/Z*[6]	O+/Z+*[6]	O/Z*[6]
			2	I	I	I			Z	I
System control	$\overline{\text{RES}}$			I	I	I			I	I
	$\overline{\text{MRES}}$			I	—	—			I/Z*[9]	I
	$\overline{\text{WDTOVF}}$			O	—	H+			H+	O+
Operation mode control	MD			I	I	I			I	I
	MD_CLK1, MD_CLK0			I	I	I			I	I
	$\overline{\text{ASEMD}}$			I	I	I			I	I
Interrupt	NMI			I	I	I			I	I
	IRQ7 to IRQ0 (PA7 to PA0, PD2 to PD0, PG3 to PG0)			I	—	—			Z	I
	IRQ7 to IRQ0 (PC3 to PC0, PJ3 to PJ0)			I	—	—			I/Z*[9]	I
	PINT7 to PINT0 (PA15 to PA8)			I	—	—			Z	Z

Pin Function				Pin State						
						Pin State Retained*2			Power-Down State	
							Other than Power-On Reset			
Type	Pin Name			Other than Power-On Reset	Power-On Reset*1	Power-On Reset*3	CS0KEE PE=1*12	CS0KEE PE=0*12	Deep Standby Mode*4	Software Standby Mode
UBC	$\overline{\text{UBCTRG}}$			O	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
Address bus	A25 to A21, A0			O	—	O+/Z+*8			O+/Z+*8	O+/Z+*8
	A20 to A2			O	O	O+/Z+*8		O	O+/Z+*8	O+/Z+*8
	A1	Area 0 data bus width	16	O	O	O+/Z+*8		O	O+/Z+*8	O+/Z+*8
			32		—	O+/Z+*8				
Bus control	D31 to D16	Area 0 data bus width	16	I/Z	—	—			Z	Z
				O/Z		Z				
			32	I/Z	Z	Z	I/Z			
				O/Z			Z	O/Z		
	D15 to D0			I/Z	Z	Z	I/Z		Z	Z
				O/Z			Z	O/Z		
	$\overline{\text{CS0}}$			O	H	H+/Z+*8		O	H+/Z+*8	H+/Z+*8
	$\overline{\text{CS5}}$ to $\overline{\text{CS1}}$			O	—	H+/Z+*8			H+/Z+*8	H+/Z+*8
	$\overline{\text{RD}}$			O	H	H+/Z+*8		O	H+/Z+*8	H+/Z+*8
	RD_$\overline{\text{WR}}$/$\overline{\text{WE}}$			O	—	H+/Z+*8			H+/Z+*8	H+/Z+*8
	$\overline{\text{WAIT}}$			I	—	—			Z	Z
	$\overline{\text{WE3}}$/$\overline{\text{BC3}}$/DQM3, $\overline{\text{WE2}}$/$\overline{\text{BC2}}$/DQM2, $\overline{\text{WE1}}$/$\overline{\text{BC1}}$/DQM1, $\overline{\text{WE0}}$/$\overline{\text{BC0}}$/DQM0			O	—	H+/Z+*8			H+/Z+*8	H+/Z+*8
	$\overline{\text{SDCS1}}$, $\overline{\text{SDCS0}}$			O	—	O+/Z+*8			O+/Z+*8	O+/Z+*8
	$\overline{\text{RAS}}$, $\overline{\text{CAS}}$			O	—	O+/Z+*8			O+/Z+*8	O+/Z+*8
	$\overline{\text{SDWE}}$			O	—	H+/Z+*8			H+/Z+*8	H+/Z+*8
	CKE			O	—	O+/Z+*8			O+/Z+*8	O+/Z+*8

Pin Function		Pin State						
				Pin State Retained*2			Power-Down State	
					Other than Power-On Reset			
Type	Pin Name	Other than Power-On Reset	Power-On Reset*1	Power-On Reset*3	CS0KEE PE=1*12	CS0KEE PE=0*12	Deep Standby Mode*4	Software Standby Mode
DMAC	DREQ3 to DREQ0	I	—	—			Z	Z
	$\overline{\text{DACK3}}$ to $\overline{\text{DACK0}}$	O	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
	$\overline{\text{DACT3}}$ to $\overline{\text{DACT0}}$	O	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
	$\overline{\text{TEND3}}$ to $\overline{\text{TEND0}}$	O	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
MTU2	TCLKA, TCLKB, TCLKC, TCLKD	I	—	—			Z	Z
	TIOC0A, TIOC0B, TIOC0C, TIOC0D	I	—	—			Z	Z
		O		O+/Z+*7			O+/Z+*7	O+/Z+*7
	TIOC1A, TIOC1B	I	—	—			Z	Z
		O		O+/Z+*7			O+/Z+*7	O+/Z+*7
	TIOC2A, TIOC2B	I	—	—			Z	Z
		O		O+/Z+*7			O+/Z+*7	O+/Z+*7
	TIOC3A, TIOC3B, TIOC3C, TIOC3D	I	—	—			Z	Z
		O		O+/Z+*7			O+/Z+*7	O+/Z+*7
	TIOC4A, TIOC4B, TIOC4C, TIOC4D	I	—	—			Z	Z
		O		O+/Z+*7			O+/Z+*7	O+/Z+*7
RTC	RTC_X1*5	I/Z*10	I	I	I/Z*10		I	I/Z*10
	RTC_X2*5	O/H*10	O	O	O/H*10		O	O/H*10
SCIF	TxD5 to TxD0	O/Z	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
	RxD5 to RxD0	I	—	—			Z	Z
	SCK5, SCK2, SCK1, SCK0	I	—	—			Z	Z
		O/Z		O+/Z+*7			O+/Z+*7	O+/Z+*7
	$\overline{\text{RTS0}}$	I	—	—			Z	Z
		O/Z		O+/Z+*7			O+/Z+*7	O+/Z+*7
	$\overline{\text{CTS0}}$	I	—	—			Z	Z
		O/Z		O+/Z+*7			O+/Z+*7	O+/Z+*7

Pin Function		Pin State						
				Pin State Retained*2			Power-Down State	
					Other than Power-On Reset			
Type	Pin Name	Other than Power-On Reset	Power-On Reset*1	Power-On Reset*3	CS0KEE PE=1*12	CS0KEE PE=0*12	Deep Standby Mode*4	Software Standby Mode
SSU	SSO1, SSO0	I	—	—			Z	Z
		O/Z		Z				
	SSI1, SSI0	I	—	—			Z	Z
		O/Z		Z				
	SSCK1, SSCK0	I	—	—			Z	Z
		O/Z		Z				
	$\overline{\text{SCS1}}$, $\overline{\text{SCS0}}$	I	—	—			Z	Z
		O/Z		Z				
IIC3	SCL3 to SCL0	I	—	—			Z	Z
		O/Z						
	SDA3 to SDA0	I	—	—			Z	Z
		O/Z						
SSIF	SSIDATA5 to SSIDATA0	I	—	—			Z	Z
		O/Z		O+/Z+*7			O+/Z+*7	O+/Z+*7
	SSISCK5 to SSISCK0	I	—	—			Z	Z
		O/Z		O+/Z+*7			O+/Z+*7	O+/Z+*7
	SSIWS5 to SSIWS0	I	—	—			Z	Z
		O/Z		O+/Z+*7			O+/Z+*7	O+/Z+*7
	AUDIO_CLK	I	—	—			Z	Z
	AUDIO_X1*5	I/Z*11	I	I/Z*11			Z	Z
	AUDIO_X2*5	O/L*11	O	O/L*11			L	L
RCAN-TL1	CTx1, CTx0	O	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
	CRx1, CRx0	I	—	—			Z	Z
ADC	AN7 to AN0	I	—	—			Z	Z
	$\overline{\text{ADTRG}}$	I	—	—			Z	Z
DAC	DA1, DA0	O/Z	—	—			Z	O+/Z+

Pin Function		Pin State						
				Pin State Retained*2			Power-Down State	
					Other than Power-On Reset			
Type	Pin Name	Other than Power-On Reset	Power-On Reset*1	Power-On Reset*3	CS0KEE PE=1*12	CS0KEEP E=0*12	Deep Standby Mode*4	Software Standby Mode
FLCTL	FOE	O	—	O+/Z+*7			O/Z*7	O/Z*7
	FSC	O	—	O+/Z+*7			O/Z*7	O/Z*7
	$\overline{\text{FCE}}$	O	—	O+/Z+*7			O/Z*7	O/Z*7
	FCDE	O	—	O+/Z+*7			O/Z*7	O/Z*7
	FRB	I	—	—			Z	Z
	$\overline{\text{FWE}}$	O	—	O+/Z+*7			O/Z*7	O/Z*7
	NAF7 to NAF0	I	—	—			Z	Z
		O/Z	—	O+/Z+*7			O/Z*7	O/Z*7
USB	DP1, DP0, DM1, DM0	I/Z	Z	Z	I/Z		Z	I/Z
		O/Z	Z	Z	O/Z		Z	O+/Z+
	VBUS	I	I	I			I	I
	REFIN	I	I	I			I	I
	USB_X1*5	I	I	I			Z	Z
	USB_X2*5	O	O	O			L	L
ATAPI	IDED15 to IDED0	I	—	—			Z	Z
		O/Z	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
	IDEA2 to IDEA0	O/Z	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
	IODACK#	O/Z	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
	IODREQ	O/Z	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
	IDECS#[1:0]	O/Z	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
	IDEIOWR#	O/Z	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
	IDEIORD#	O/Z	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
	IDEIORDY	I	—	—			Z	Z
	IDEINT	I	—	—			Z	Z
	IDERST#	O/Z	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
	DIRECTION	O	—	O+/Z+*7			O+/Z+*7	O+/Z+*7

Pin Function				Pin State						
						Pin State Retained*2			Power-Down State	
							Other than Power-On Reset			
Type	Pin Name			Other than Power-On Reset	Power-On Reset*1	Power-On Reset*3	CS0KEEPE=1*12	CS0KEEPE=0*12	Deep Standby Mode*4	Software Standby Mode
2DG	R, G, B			O	O	O			Z	O+
	REXT			I	I	I			I	I
	CBU			O	O	O			O	O
	CSYNC			O	—	O+/Z+*7			O+/Z+*7	O+/Z+*7
	DCLKIN			I	—	—			Z	Z
	VIHSYNC			I	—	—			Z	Z
	VIVSYNC			I	—	—			Z	Z
	VIDATA7 to VIDATA0			I	—	—			Z	Z
	VICLKENB			I	—	—			Z	Z
I/O port	PA15 to PA0	Area 0 data bus width	16	I	Z	Z	I		Z	Z
				O		O+/Z+*7			O+/Z+*7	O+/Z+*7
			32	I	—	—			Z	Z
				O		O+/Z+*7		—	O+/Z+*7	O+/Z+*7
	PB18 to PB2, PB0			I	Z	Z	I		Z	Z
				O		O+/Z+*7			O+/Z+*7	O+/Z+*7
	PB1	Area 0 data bus width	16	I	—	—			Z	Z
				O		O/Z*7		—	O+/Z+*7	O+/Z+*7
			32	I	Z	Z	I		Z	Z
				O		O+/Z+*7			O+/Z+*7	O+/Z+*7
	PC10 to PC0			I	Z	Z	I		Z	Z
				O		O+/Z+*7			O+/Z+*7	O+/Z+*7
	PD2 to PD0			I	Z	Z	I		Z	Z
				O		O+/Z+*7			O+/Z+*7	O+/Z+*7
	PE13, PE11, PE9, PE7 to PE0			I	Z	Z	I		Z	Z
				O		O+/Z+*7			O+/Z+*7	O+/Z+*7

Pin Function		Pin State						
				Pin State Retained*2			Power-Down State	
					Other than Power-On Reset			
Type	Pin Name	Other than Power-On Reset	Power-On Reset*1	Power-On Reset*3	CS0KEEPE=1*12	CS0KEEPE=0*12	Deep Standby Mode*4	Software Standby Mode
I/O port	PE12, PE10, PE8	I	Z	Z	I		Z	Z
	PF4 to PF0	I	Z	Z	I		Z	Z
		O		O+/Z+*7			O+/Z+*7	O+/Z+*7
	PG7 to PG0	I	Z	Z	I		Z	Z
	PJ12 to PJ0	I	Z	Z	I		Z	Z
		O		O+/Z+*7			O+/Z+*7	O+/Z+*7
	PH15 to PH0	I	Z	Z	I		Z	Z
		O		O+/Z+*7			O+/Z+*7	O+/Z+*7
	PK1, PK0	I	Z	Z	I		Z	Z
		O		O+/Z+*7			O+/Z+*7	O+/Z+*7
H-UDI	$\overline{\text{TRST}}$	I	I	I			Z	I
	TCK	I	I	I			Z	I
	TDI	I	I	I			Z	I
	TDO	O/Z*13	O/Z*13	O/Z*13			O+/Z+*13	O+/Z+*13
	TMS	I	I	I			Z	I
Emulator *14	$\overline{\text{AUDSYNC}}$	—	—	—			—	—
	AUDCK	—	—	—			—	—
	AUDATA3 to AUDATA0	—	—	—			—	—
	$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$	Z	Z	Z			Z	Z

[Legend]

I: Input

O: Output

H: High-level output

L: Low-level output

Z: Input pins retain their state, and output pins become high-impedance,.

+: Output state is retained.

Notes:
1. Indicates the power-on reset by low-level input to the $\overline{\text{RES}}$ pin. The pin states after a power-on reset by the H-UDI reset assert command or WDT overflow are the same as the initial pin states at normal operation (see section 27, Pin Function Controller (PFC)).
2. Indicates the pin states that the IOKEEP bit in the deep standby cancel source flag register (DSFR) is cleared if the chip has recovered from deep standby mode by the input on any of pins NMI, $\overline{\text{MRES}}$, and IRQ7 to IRQ0 (see section 30, Power-Down Modes).
3. This LSI shifts to the power-on reset state for a certain period after the recovery from the deep standby mode (see section 30, Power-Down Modes).
4. The week keeper and pull-up circuits included in the I/O pins are turned off.
5. When pins for the connection with a crystal resonator are not used, the input pins (EXTAL, RTC_X1, AUDIO_X1, and USB_X1) must be fixed (pulled up, pulled down, connected to power supply, or connected to ground) and the output pins (XTAL, RTC_X2, AUDIO_X2, and USB_X2) must be open.
6. Depends on the setting of the CKOEN bit in the frequency control register (FRQCR) of the CPG (see section 5, Clock Pulse Generator (CPG)).
7. Depends on the setting of the HIZ bit in the high impedance control register (HIZCR) (see section 30, Power-Down Modes).
8. Depends on the setting of the HIZBSC bit in the high impedance control register (HIZCR) (see section 30, Power-Down Modes).
9. Depends on the setting of the corresponding bit in the deep standby cancel source select register (DSSSR) (see section 30, Power-Down Modes).
10. Depends on the setting of the RTCEN bit in the RTC control register (RCR2) of the RTC (see section 15, Realtime Clock (RTC)).
11. Depends on the AXTALE bit in the standby control register (STBCR1) (see section 30, Power-Down Modes).
12. Depend on the CS0KEEPE bit in the deep standby control register (DSCTR) (see section 30, Power-Down Modes).
13. Z when the TAP controller of the H-UDI is neither the Shift-DR nor Shift-IR state.
14. These are the pin states in product chip mode ($\overline{\text{ASEMD}}$=H). See the Emulation Manual (provisional title) for the pin states in ASE mode ($\overline{\text{ASEMD}}$=L).

B. Treatment of Unused Pins

Table B.1 Handling of Pins that are not in Use (Except for H-UDI and Emulator Interface Pins)

Pin	Handling
NMI	Fix this pin at a high level (pull up or connect to the power-supply level).
DP, DM, and VBUS	Connect these pins to PVss.
REFRIN	Connect this pin, via a 5.6 kΩ ±20% resistor, to USBAPVcc.
Dedicated USB power pins (USBAPVcc, USBAPVss, USBAVcc, USBAVss, USBDVcc)	Connect the power-supply pins to the power-supply level and ground pins to the ground level.
AVref	Connect this pin to AVcc.
Dedicated A/D and D/A power pins (AVcc, AVss)	Connect the power-supply pin to the power-supply level and ground pin to the ground level.
R, G, B, REXT, and CBU	Open-circuit
Dedicated 2DG power pins (2DGAPVcc0, 2DGAPVcc1, 2DGAPVss0, 2DGAPVss1)	Connect the power-supply pins to the power-supply level and ground pins to the ground level.
Pins with weak keeper or pull-up	Open-circuit
Dedicated input pins other than those listed above	Fix the level on the pins (pull them up or down, or connect them to the power-supply or ground level).
Input/output pins other than those listed above	Make the input-pin settings and then fix the level (pull them up or down); alternatively, make the output-pin settings and leave the pins open-circuit.
Dedicated output pins other than those listed above	Open-circuit

Note: It is recommended that the values of pull-up or pull-down resistors are in the range from 4.7 kΩ to 100 kΩ.

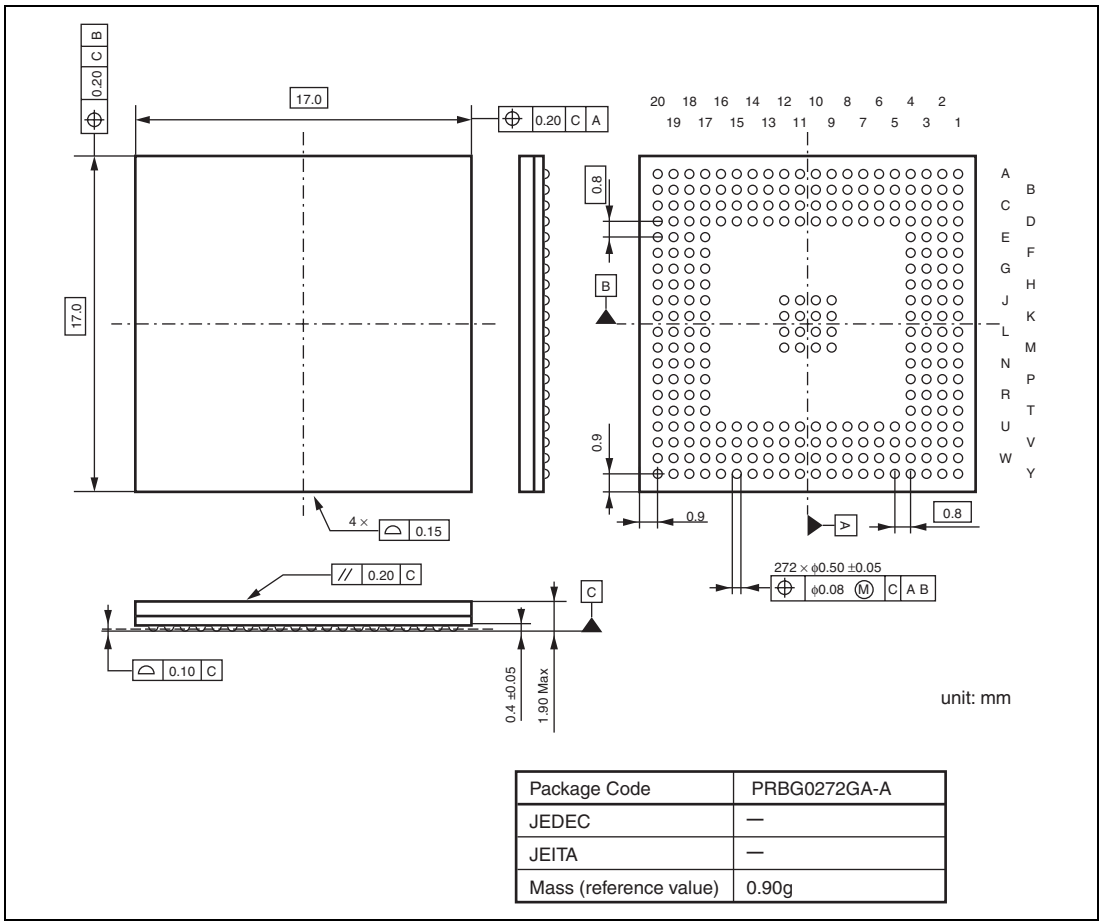
Table B.2 Handling of Pins that are not in Use (When H-UDI is Not Used in Product Chip Mode)

Pin	Handling
$\overline{\text{ASEMD}}$	Fix this pin at a high level (pull up or connect to the powersupply level).
$\overline{\text{TRST}}$	Pull-down with a 1 kΩ resistor. Or, fix this pin at a low level when power-up or release the deep-standby mode by $\overline{\text{RES}}$ pin, and open-circuit other than above.
TCK, TMS, TDI, TDO, $\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$	Open-circuit

Notes: 1. When using the H-UDI, handle these pins as described in the manual for the emulator.
2. It is recommended that the values of pull-up or pull-down resistors are in the range from 4.7 kΩ to 100 kΩ.

C. Package Dimensions

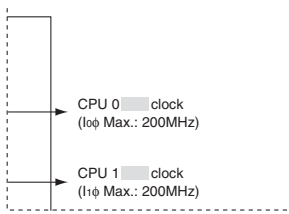
The package dimension that is shown in the Renesas Semiconductor Package Data Book has Priority.



Package Code	PRBG0272GA-A
JEDEC	—
JEITA	—
Mass (reference value)	0.90g

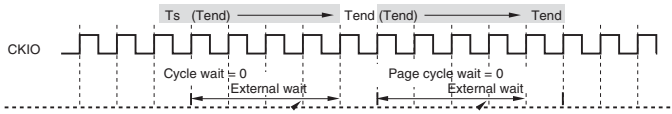
Figure C.1 Package Dimensions

Main Revisions for This Edition

Item	Page	Revision (See Manual for Details)
Preface	iv	Company name changed This LSI is an RISC (Reduced Instruction Set Computer) microcomputer which includes a Renesas -original RISC CPU as its core, and the peripheral functions required …
1.1 SH7205 Features	1	Company name changed This LSI is a single-chip RISC (reduced instruction set computer) microcontroller that includes two Renesas -original RISC CPUs as its cores, and the peripheral functions required to configure a system.
Table 1.1 SH7205 Features	2	Table amended **Items** / **Specification** CPU: • Renesas original SuperH architecture • Compatible with SH-1, SH-2, and SH-2E at object code level
Section 5 Clock Pulse Generator (CPG)	119	Description amended This LSI has a clock pulse generator (CPG) that generates CPU0 clock ($I_0\phi$), CPU1 clock ($I_1\phi$), a peripheral clock (Pϕ), and a bus clock (Bϕ).
5.1 Features		Description amended • Three clocks generated independently CPU0 clock ($I_0\phi$) and CPU1 clock ($I_1\phi$) for the CPU and cache, a peripheral clock (P ϕ) for peripheral modules, … • Frequency change function CPU0 clock, CPU1 clock, and peripheral clock frequencies can be changed independently using the PLL (phase locked loop) circuits and divider circuits within the CPG.
Figure 5.1 Block Diagram of Clock Pulse Generator	120	Figure amended 

Item	Page	Revision (See Manual for Details)
5.1 Features (4) Divider 2	121	Description amended Divider 2 generates a clock signal whose operating frequency can be used for the CPU0 clock, CPU1 clock, peripheral clock, and bus clock. The division ratio of the CPU0 clock, CPU1 clock, and peripheral clock are set by the frequency control register.
(7) Frequency Control Register 0 (FRQCR0)		Description amended …, the frequency multiplication rate of the PLL circuit, and the frequency division ratio of the CPU0 clock ($I_0\phi$) and the peripheral clock ($P\phi$).
(8) Frequency Control Register 1 (FRQCR1)	122	Description amended The frequency control register 1 (FRQCR1) has control bits assigned for the following function: frequency division ratio of the CPU1 clock ($I_1\phi$).
5.3 Clock Operating Modes Table 5.3 Relationship between Clock Operating Mode and Frequency Range	126 to 128	Table header amended Ratio of Internal Clock Frequencies (I_0:I_1:B:P)*2 / Selectable Frequency Range (MHz): Input Clock*3 / CPU0 Clock ($I_0\phi$) / CPU1 Clock ($I_1\phi$) / Bus Clock (Bϕ = CKIO Pin) / Peripheral Clock (Pϕ)
5.4.1 Frequency Control Registers 0 and 1 (FRQCR0 and FRQCR1)	129	Description amended The register also specifies the frequency multiplication rate of the PLL circuit and the frequency division ratio for the CPU0 clock ($I_0\phi$) and peripheral clock ($P\phi$).
(1) FRQCR0	131	Table amended Bit / Bit Name / Initial Value / R/W / Description 5, 4 / IFC[1:0] / 10/01* / R/W / Division Ratio of CPU0 Clock Frequency ($I_0\phi$) Specify the division ratio for the CPU0 clock, which is used in division of the output frequency of the PLL circuit.
(2) FRQCR1	133	Description amended FRQCR1 is a 16-bit readable/writable register used to specify the frequency division ratio of the CPU1 clock ($I_1\phi$).
		Table amended Bit / Bit Name / Initial Value / R/W / Description 5, 4 / IFC[1:0] / 10/01* / R/W / Division Ratio of CPU0 Clock Frequency ($I_0\phi$) Specify the division ratio for the CPU0 clock, which is used in division of the output frequency of the PLL circuit.

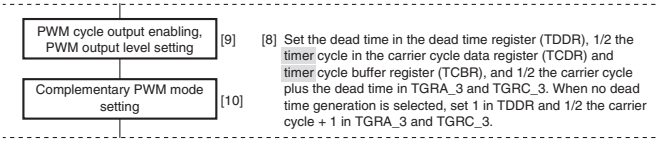
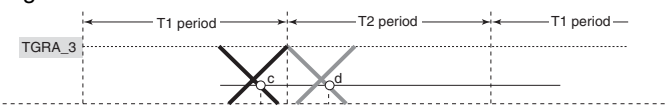
Item	Page	Revision (See Manual for Details)
5.5 Changing the Frequency	134	Description amended The frequencies of the CPU0 clock ($I_0\phi$), CPU1 clock ($I_1\phi$), and peripheral clock (P ϕ) can be changed either by changing the multiplication rate of the PLL circuit or by changing …
5.5.1 Changing the Multiplication Rate		Description amended 4. This LSI pauses temporarily and WDT0 starts to increment. A clock is supplied to the WDT only, and the other internal clocks all stop. The clock will continue to be output at the CKIO pin. …
5.5.2 Changing the Division Ratio	136	Description amended … When changing the division ratio of only the CPU1 clock, it is not necessary to place CPU1 in the sleep state. • When the CPU0 clock is changed from CPU0 … • When the CPU1 clock is changed from CPU1 …
5.5.3 Notes on Changing the Multiplication Rate and Division Ratio	137	Description amended 1. When the division ratio for the CPU1 clock is changed, if IFC[1:0] of FRQCR1 are changed while …
5.8.1 Note on Using a PLL Oscillation Circuit	141	Description added To prevent such malfunction, the analog power supply pin Vcc and digital power supply pin PVcc should not supply the same resources on the board if at all possible. Ensure that PLLVcc has the same electric potential as Vcc.
6.2.2 Types of Reset Table 6.6 Reset States	149	Table and title replaced
6.10.4 Interrupt Control via Modification of Interrupt Mask Bits	166	Newly added
10.3.1 Address Map Table 10.2 Address Map	278	Table note amended Notes: 2. … These devices are accessed directly from individual CPUs without using the internal bus.
10.3.2 Data Bus Width and Related Pin Settings for Each Area	279	Section replaced

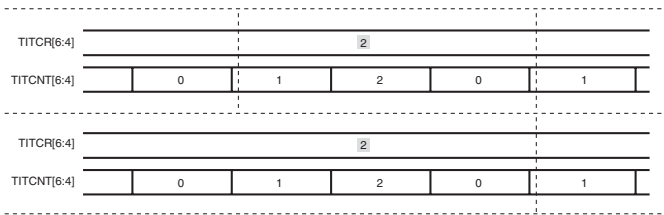
Item	Page	Revision (See Manual for Details)
10.4.8 SDRAM Refresh Control Register 1 (SDRFCNT1) (a) Auto-Refresh Request Interval and DRFC Set Value	297	Equation amended DRFC = (Auto-refresh request interval / Bus clock cycle) – 1
10.5.1 Accessing CS Space (3) External Wait Function (b) Page Access Operation Figure 10.7 External Wait Timing Example (Page Read Access to 16-Bit Channel)	321	Figure amended 
12.3.16 Timer Output Master Enable Register (TOER)	517	Description added Set TOER of CH3 and CH4 prior to setting TIOR of CH3 and CH4. Set TOER when count operation of TCNT channels 3 and 4 is halted.
12.3.17 Timer Output Control Register 1 (TOCR1)	519	Table and note amended **Bit** / **Bit Name** / **Initial value** / **R/W** / **Description** 3 / TOCL / 0 / R/(W)*3 / TOC Register Write Protection*2 This bit selects the enable/disable of write access to the ... 1 / OLSN / 0 / R/W / Output Level Select N*2*4 This bit selects the reverse phase output level in reset- ... Notes: 3. After power-on reset, 1 can be written only once. After 1 has been written, 0 cannot be written. 4. If there is no dead time, the reverse phase output is the inversion of the forward phase. Set OLSP and OLSN to the same value.
12.3.18 Timer Output Control Register 2 (TOCR2)	523	Table note amened Note: * Setting the TOCS bit in TOCR1 to 1 makes this bit setting valid. If there is no dead time, the reverse phase output is the inversion of the forward phase. Set OLSiP and OLSiN to the same value (i = 1, 2, or 3).

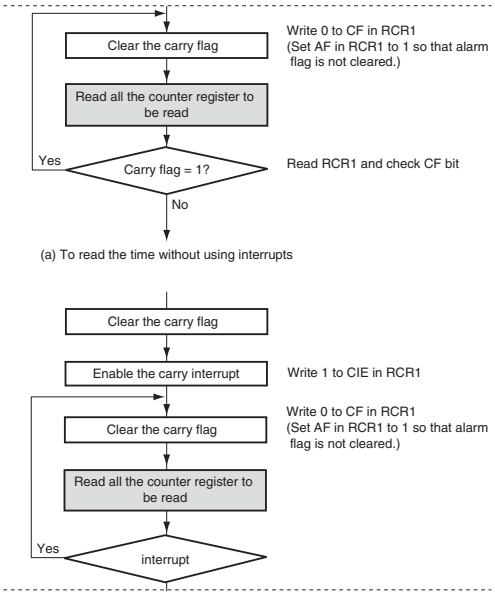
Item	Page	Revision (See Manual for Details)
12.3.23 Timer Cycle Data Register (TCDR)	529	Description amended TCDR is a 16-bit register used only in complementary PWM mode. Set half the PWM carrier sync value (a value of two times TDDR + 3 or greater) as the TCDR register value.
12.3.29 Timer Waveform Control Register (TWCR)	537	Table amended

Bit	Bit Name	Initial Value	R/W	Description
0	WRE	0	R/(W)	Initial Output Suppression Enable Selects the waveform output when synchronous counter clearing occurs in complementary PWM mode. The initial output is suppressed only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial value specified in TOCR is output regardless of the WRE bit setting. The initial value is also output when synchronous clearing occurs in the Tb interval at the trough immediately after TCNT_3 and TCNT_4 start operation. For the Tb interval at the trough in complementary PWM mode, see figure 12.40. 0: Outputs the initial value specified in TOCR 1: Suppresses initial output [Setting condition] • When 1 is written to WRE after reading WRE = 0

12.4.4 Cascaded Operation	550	Description amended For simultaneous input capture of TCNT_1 and TCNT_2 during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). The edge detection that is the condition for input capture uses a signal representing the logical OR of the original input pin and the added input pins. For details, see (4) Cascaded Operation Example (c).
(1) Example of Cascaded Operation Setting Procedure Figure 12.20 Cascaded Operation Setting Procedure	551	Figure amended [1] Set bits TPSC2 to TPSC0 in the channel 1 TCR to B'111 to select TCNT_2 overflow/ underflow counting.
(4) Cascaded Operation Example (c) Figure 12.23 Cascaded Operation Example (c)	553	Figure replaced
12.4.5 PWM Modes	555	Description amended • PWM mode 2 … Upon counter clearing by a cycle register compare match, the output value of each pin is the initial value set in TIOR.

Item	Page	Revision (See Manual for Details)
12.4.8 Complementary PWM Mode	570	Description deleted Table 12.52 shows the PWM output pins used. Table 12.53 shows the settings of the registers used.
(1) Example of Complementary PWM Mode Setting Procedure Figure 12.38 Example of Complementary PWM Mode Setting Procedure	573	Figure amended 
(2) Outline of Complementary PWM Mode Operation (g) PWM Cycle Setting	581	Description amended With dead time: TGRA_3 set value = TCDR set value + TDDR set value TCDR set value > two times TDDR + 2 Without dead time: TGRA_3 set value = TCDR set value + 1 TCDR set value > 4
(j) Complementary PWM Mode PWM Output Generation Method	586	Description amended 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 compare register. While TCNTS is counting, compare register and temporary …
Figure 12.46 Example of Complementary PWM Mode Waveform Output (1)	587	Figure amended 
(k) Complementary PWM Mode 0% and 100% Duty Output	591	Description amended 100% duty output is performed when the compare 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 compare register value is set to the same value as TGRA_3.

Item	Page	Revision (See Manual for Details)
12.4.8 Complementary PWM Mode (n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode	593	Description added … immediately after the counters start operation, initial value output is not suppressed. When using the initial output suppression function, make sure to set compare registers TGRB_3, TGRA_4, and TGRB_4 to a value twice or more the setting of dead time data register TDDR. If synchronous clearing occurs with the compare registers set to a value less than twice the setting of TDDR, the PWM output dead time may be too short (or nonexistent) or illegal active-level PWM negative-phase output may occur during the initial output suppression interval. For details, see section 12.7.23, Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode.
(3) Interrupt Skipping in Complementary PWM Mode (c) Buffer Transfer Control Linked with Interrupt Skipping Figure 12.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)	607	Figure amended 
12.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode	645, 646	Newly added
12.8.2 Reset Start Operation	647	Description amended The MTU2 output pins (TIOC*) are initialized low by a power-on reset and in deep standby mode.
14.5.7 Internal Reset in Watchdog Timer Mode	705	Newly added
15.4.2 Setting Time Figure 15.2 Setting Time	731	Figure replaced

Item	Page	Revision (See Manual for Details)
15.4.3 Reading Time Figure 15.3 Reading Time	732	Figure amended 
15.5.4 Usage Notes when Writing to and Reading the Register	735	Description amended • When reading a counter register such as the second counter after having written to the register, follow the procedure in section 15.4.2, Setting Time. In this case, it is necessary to write to all the counters, from second to year, in succession. Do not read the counter registers during the write processing shown as (2) in figure 15.2.
16.3.6 Serial Control Register (SCSCR)	750	Table amended **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 3 / REIE / 0 / R/W / Receive Error Interrupt Enable ... 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.

Item	Page	Revision (See Manual for Details)

16.3.8 Bit Rate Register (SCBRR)

Table 16.4 Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0) (2)

763

Table amended

	Pφ (MHz)											
	12.288			14.7456			16			20		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	3	64	0.70	3	70	0.03	3	88	–0.25
150	2	159	0.00	2	191	0.00	2	207	0.16	3	64	0.16
300	2	79	0.00	2	95	0.00	2	103	0.16	2	129	0.16
600	1	159	0.00	1	191	0.00	1	207	0.16	2	64	0.16
1200	1	79	0.00	1	95	0.00	1	103	0.16	1	129	0.16
2400	0	159	0.00	0	191	0.00	0	207	0.16	1	64	0.16
4800	0	79	0.00	0	95	0.00	0	103	0.16	0	129	0.16
9600	0	39	0.00	0	47	0.00	0	51	0.16	0	64	0.16
19200	0	19	0.00	0	23	0.00	0	25	0.16	0	32	–1.36
31250	0	11	2.40	0	14	1.70	0	15	0.00	0	19	0.00
38400	0	9	0.00	0	11	0.00	0	12	0.16	0	15	1.73

Table 16.5 Bit Rates and SCBRR Settings (Clock Synchronous Mode)

766

Table and legend amended

	Pφ (MHz)									
	8		16		28.7		30		33	
Bit Rate (bit/s)	n	N	n	N	n	N	n	N	n	N
500 k	0	3	0	7	—	—	0	14	—	—
1 M			0	3	—	—	—	—	—	—
2 M					—	—	—	—	—	—

[Legend]

Blank: No setting possible, or the electrical characteristics of the SH7205 cannot be satisfied regardless of the device being communicated with.

—: Setting possible, but error occurs

Table 16.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

768

Table amended

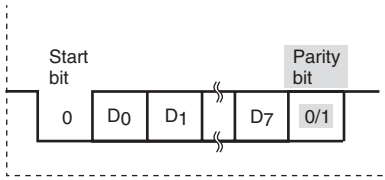
	Settings				
Pφ (MHz)	BGDM	ABCS	n	N	Maximum Bit Rate (bits/s)
20	0	0	0	0	625000
		1	0	0	1250000
	1	0	0	0	1250000
		1	0	0	2500000

Table 16.7 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

769

Table amended

		Settings	
Pφ (MHz)	External Input Clock (MHz)	ABCS	Maximum Bit Rate (bits/s)
16	4.0000	0	250000
		1	500000
20	5.0000	0	312500
		1	625000

Item	Page	Revision (See Manual for Details)
16.4.2 Operation in Asynchronous Mode (3) Transmitting and Receiving Data Figure 16.10 Example of Operation Using Modem Control (RTS)	792	Figure amended 
18.3.1 I²C Bus Control Register 1 (ICCR1)	851	Table amended Bit: 7 — Bit Name: ICE — Initial Value: 0 — R/W: R/W — Description: I²C Bus Interface 3 Enable 0: SCL and SDA output is disabled. (Input to SCL and SDA is enabled) 1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.)
18.7.5 Note on I²C-bus Interface Master Receive Mode 18.7.6 Note on IICRST and BBSY bits 18.7.7 Note on Issuance of Stop Conditions in Master Transmit Mode while ACKE = 1	888	Newly added
Section 19 Serial Sound Interface with FIFO (SSIF)	889	Description amended The serial sound interface with FIFO (SSIF) is a module designed to send or receive audio data interface with various devices offering I²S bus compatibility.
19.4.1 Bus Format	913	Description amended The bus format can be selected from one of the four major modes shown in table 19.3.

Item	Page	Revision (See Manual for Details)
19.4.2 Non-Compressed Modes	914	Description amended It supports the I²S compatible format as well as many more variants on these modes.
(5) Operating Setting Related to Word Length		Description amended There are many configurations the SSIF module supports, but some of the combinations are shown below for the I²S compatible format, MSB-first and left-aligned format, and MSB-first and right-aligned format.
	915	Description amended Figures 19.3 and 19.4 demonstrate the supported I²S compatible format both without and with padding. • I²S Compatible Format
Figure 19.3 I²S Compatible Format (without Padding) Figure 19.4 I²S Compatible Format (with Padding)		Figure title amended
	916	Description amended Figure 19.5 shows the MSB-first and left-aligned format and figure 19.6 shows the MSB-first and right-aligned format. … • MSB-first and Left-Aligned Format … • MSB-first and Right-Aligned Format
Figure 19.5 MSB-first and Left-Aligned Format (Transmitted and received in the order of serial data and padding bits)		Figure title amended

Item	Page	Revision (See Manual for Details)
19.4.2 Non-Compressed Modes (5) Operating Setting Related to Word Length Figure 19.6 MSB-first and Right-Aligned Format (Transmitted and received in the order of padding bits and serial data)	916	Figure title amended
(6) Multi-channel Formats	917	Description amended Some devices extend the definition of the specification by I^2S bus and allow more than 2 channels to be transferred within two system words.
19.4.4 Transmit Operation (1) Transmission Using DMA Controller Figure 19.20 Transmission Using DMA Controller	926	Figure replaced
(2) Transmission Using Interrupt Data Flow Control Figure 19.21 Transmission Using Interrupt Data Flow Control	927	
19.4.5 Receive Operation (1) Reception Using DMA Controller Figure 19.22 Reception Using DMA Controller	929	
(2) Reception Using Interrupt Data Flow Control Figure 19.23 Reception Using Interrupt Data Flow Control	930	

Item	Page	Revision (See Manual for Details)
20.2 Architecture	937	Description amended • Timer … Contains registers such as TCNTR, TTCR0, CMAX_TEW, RFTROFF, TSR, CCR, CYCTR, RFMK, TCMR0, TCMR1, TCMR2 and TTTSEL.
20.3.3 RCAN-TL1 Control Registers (2) General Status Register (GSR)	964	Table amended

Bit 1: GSR1	Description
0	[Reset condition] When (TEC < 96 and REC < 96) or Bus Off (Initial value)
1	[Setting condition] When 96 ≤ TEC < 256 or 96 ≤ REC < 256

Bit 0: GSR0	Description
0	[Reset condition] Recovery from bus off state or after a HW or SW reset (Initial value)
1	[Setting condition] When TEC ≥ 256 (bus off state)

(3) Bit Configuration Register (BCR0, BCR1)	965	Description amended Where: BRP (Baud Rate Pre-scaler) is the value stored in BCR0 incremented by 1 and fclk is the used peripheral clock frequency.
	967	Table amended

Bit 7: BRP[7]	Bit 6: BRP[6]	Bit 5: BRP[5]	Bit 4: BRP[4]	Bit 3: BRP[3]	Bit 2: BRP[2]	Bit 1: BRP[1]	Bit 0: BRP[0]	Description
0	0	0	0	0	0	0	0	2 × peripheral clock (Initial value)
0	0	0	0	0	0	0	1	4 × peripheral clock
0	0	0	0	0	0	1	0	6 × peripheral clock
:	:	:	:	:	:	:	:	2 × (register value + 1)
:	:	:	:	:	:	:	:	× peripheral clock
1	1	1	1	1	1	1	1	512 × peripheral clock

20.3.5 Timer Registers (1) Time Trigger Control Register0 (TTCR0)	994	Description and table amended Bit 14 — TimeStamp value: Specifies if the Timestamp for transmission and reception in Mailboxes 15 to 0 must contain the Cycle Time (CYCTR) or the concatenation of CCR[5:0] + CYCTR[15:6]. This feature is very useful for time triggered transmission to monitor Rx_Trigger.

Bit14: TTCR0 14	Description
0	CYCTR[15:0] is used for the TimeStamp in Mailboxes 15 to 0 (initial value)
1	CCR[5:0] + CYCTR[15:6] is used for the TimeStamp in Mailboxes 15 to 0

Item	Page	Revision (See Manual for Details)
20.5 Interrupt Sources Table 20.2 RCAN-TL1-n*1 Interrupt Sources	1031	Table and note amended **Interrupt** / **Description** / **Interrupt Flag** / **DMAC Activation** RM0n*1*2, RM1n*1*2 / Data frame reception / IRR1*3 / Possible*4 RM0n*1*2, RM1n*1*2 / Remote frame reception / IRR2*3 / Possible*4 SLEn*1 / Message transmission/transmission disabled (slot empty) / IRR8 / Not possible Notes: 4. The DMAC is activated only by an RM0n interrupt.
21.1 Features	1037	Description deleted • Minimum conversion time: 3.9 μs per channel
21.3.2 A/D Control/Status Register (ADCSR)	1042, 1044, 1045	Table and note amended **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 15 / ADF / 0 / R/(W)*1 / A/D End Flag Status flag indicating the end of A/D conversion. ... 7, 6 / CKS[1:0] / 01 / R/W / Clock Select These bits select the A/D conversion time*2. Set the A/D conversion time while A/D conversion is halted (ADST = 0). 00: Conversion time = 138 t_{pcyc}*3 (maximum) 01: Conversion time = 274 t_{pcyc}*3 (maximum) 10: Conversion time = 546 t_{pcyc}*3 (maximum) 11: Setting prohibited Notes: 1. The flag can only be cleared by writing 0 to it after reading it as 1. However, in the following cases as well the flag is cleared by writing 0 to it: … 2. Set the A/D conversion time to minimum or more values to meet the absolute accuracy of the A/D conversion characteristics. 3. tpcyc indicates the peripheral clock (Pϕ) cycle.
21.4.5 Input Sampling and A/D Conversion Time Table 21.4 A/D Conversion Time (Single Mode)	1055	Table note amended Note: Values in the table are the numbers of t_{pcyc}. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.
Table 21.5 A/D Conversion Time (Multi Mode and Scan Mode)	1056	Table header and note amended **CKS1** / **CKS0** / **Conversion Time (t_{pcyc})** Note: Values in the table are the numbers of t_{pcyc}. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.

Item	Page	Revision (See Manual for Details)
21.7.7 A/D Conversion in Deep Standby Mode	—	Section deleted
22.5.3 D/A Conversion in Deep Standby Mode	—	Section deleted
24.1 Features (5) Pipe Configuration	1126	Description amended PIPE0: Control transfer, continuous transfer mode, 256-byte fixed-size single buffer
24.3.1 System Configuration Control Register 0 (SYSCFG0)	1136, 1137	Table amended **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 4 / DPRPU / 0 / R/W / PORT0 D+ Line Resistor Control ... 1: Pulling up the line is enabled. Note: Set this bit to 0 when the USB is disconnected. Include the following processing when this bit is changed from 1 to 0. (1) Set the DPRPU bit to 0. (2) Wait for at least 1 s. (3) Set the DCFM bit to 1. (4) Wait for at least 200 ns. (5) Set the DCFM bit to 0. 0 / USBE / 0 / R/W / ... When the host controller function is selected, this bit should be set to 1 after setting DRPD to 1, eliminating LNST bit chattering, and checking that the USB bus has been settled.
24.3.5 Device State Control Register 0 (DVSTCTR0)	1145	Table amended **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 4 / UACT / 0 / R/W / PORT0 USB Bus Enable ... Note: Writing 1 to this bit should be done at the end of the USB bus reset process (writing 0 to USBRST) or at the end of the resume process from the suspended state (writing 0 to RESUME).
24.3.6 Device State Control Register 1 (DVSTCTR1)	1150	Table amended **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 4 / UACT / 0 / R/W / PORT1 USB Bus Enable ... Note: Writing 1 to this bit should be done at the end of the USB bus reset process (writing 0 to USBRST) or at the end of the resume process from the suspended state (writing 0 to RESUME).

Item	Page	Revision (See Manual for Details)
24.3.7 Test Mode Register (TESTMODE)	1153	Table amended Bit: 3 to 0 — Bit Name: UTST[3:0] — Initial Value: 0000 — R/W: R/W Description: • When the host controller function is selected ... Procedure for setting the UTST bits: 1. Power-on reset the system. 2. Start the clock supply (set SCKE to 1). 3. Set DCFM and DRPD to 1 (setting HSE to 1 is not required).
24.3.11 FIFO Port Control Registers (CFIFOCTR, D0FIFOCTR, D1FIFOCTR)	1166, 1167	Table amended Bit: 15 — Bit Name: BVAL — Initial Value: 0 — R/W: R/W*1 Description: Buffer Memory Valid Flag ... When the data of the maximum packet size has been written for the pipe in non-continuous transfer mode, this module sets this bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission. When the specified pipe is in the transmitting direction, if 1 is written to BVAL and BCLR bits simultaneously, this module clears the data that was written before, enabling transmission of a zero-length packet. 0: Invalid 1: Writing ended Bit: 14 — Bit Name: BCLR — Initial Value: 0 — R/W: R/W*2 Description: CPU Buffer Clear ... When double buffer mode is set for the FIFO buffer assigned to the specified pipe, this module clears only one plane of the FIFO buffer even when both planes are read-enabled.
24.3.18 SOF Output Configuration Register (SOFCFG)	1181	Table amended Bit: 6 — Bit Name: BRDYM — Initial Value: 0 — R/W: R/W Description: PIPEBRDY Interrupt Status Clear Timing ... 1: Automatic clearing by data reading from the FIFO buffer or data writing to the FIFO buffer Note: This bit should be set when making initial settings to the module (before communication starts). Do not change the setting of this bit after communication starts.
24.3.22 BRDY Interrupt Status Register (BRDYSTS) (2) When BRDYM = 0 and BFRE = 1 are Set (b) Packets for the Value Set in Bits TRNCNT have been Received When the Transaction Counter (Bits TRNCNT) is Used	1194	Description amended When either of the above conditions is satisfied and reading of that data has been completed, this module determines that all the data for one round of transfer have been read out. When a zero-length packet is received while the FIFO buffer is empty, the USB 2.0 host/function module determines that all the data for a single transfer has been read at the point at which the FRDY bit is set to 1 and the DTLN bit cleared to 0 in the FIFO port control register. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding FIFOCTR register.

Item	Page	Revision (See Manual for Details)
24.3.32 DCP Configuration Register (DCPCFG)	1212	Bit table amended

10	9	8	7	6	5
—	—	CNTMD	SHT NAK	—	—
-	-	0	0	-	-
R	R	R/W	R/W	R	R

1212, 1213 — Table and note amended

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.
8	CNTMD	0	R/W	Continuous Transfer Mode* Specifies whether or not the DCP operates in continuous transfer mode. Based on the setting value of this bit, the USB module determines completion of transmit and receive to and from the FIFO buffer allocated to the DCP as described in table 24.12. 0: Non-continuous transfer mode 1: Continuous transfer mode
7	SHTNAK	0	R/W	Pipe Disable at Transfer-End* Specifies whether or not to change the PID bits to NAK at transfer-end when the DCP is operating in the receive direction. When this bit is set to 1, the USB module changes the PID bits to NAK when it determines that a transfer to the DCP has finished. The USB module determines that transfer-end has occurred when a short packet of data (or a zero-length packet) is received normally. 0: Continue pipe operation at transfer-end 1: Disable pipe operation at transfer-end Note: Clear this bit to 0 when the DCP is operating in the transmit direction.
6, 5	—	Undefined	R	Reserved Undefined values are read from these bits. The write value should always be 0.

Notes: * When changing the setting of the DCP's PID bits after they have been changed from BUF to NAK, confirm that both CSSTS and PBUSY are cleared to 0 before changing the PID bits. However, it is not necessary to check the PBUSY bit if the PID bits were changed to NAK by the USB module.

Change the setting of the CNTMD bit when CSSTS = 0, PID = NAK, and the CURPIPE bits are unset. When changing the setting of the CNTMD bit after USB communication using the DCP, write 1 to BCLR and clear the FIFO buffer allocated to the DCP, in addition to ensuring that the above three registers are as described.

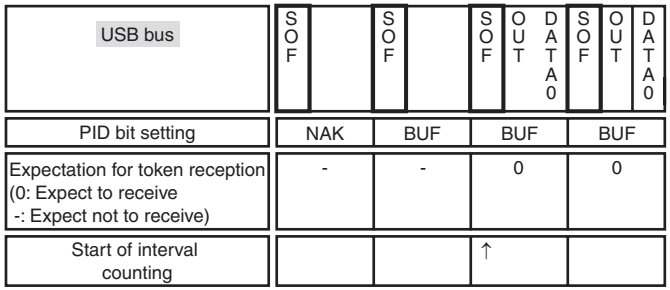
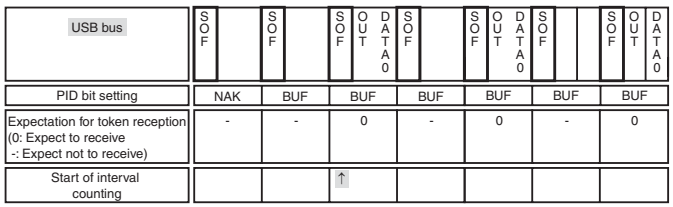
Change the setting of the SHTNAK bit when CSSTS = 0 and PID = NAK.

Item	Page	Revision (See Manual for Details)
24.3.32 DCP Configuration Register (DCPCFG) Table 24.12 Setting Value of CNTMD Bit and Method of Determining Completion of Transmit and Receive to and from FIFO Buffer	1214	Newly added
24.3.34 DCP Control Register (DCPCTR)	1221	Table amended

Bit	Bit Name	Initial Value	R/W	Description
5	PBUSY	0	R	Pipe Busy This bit indicates whether the relevant pipe is used or not for the transaction. 0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction. The USB 2.0 host/function module switches the PBUSY bit from 0 to 1 when a USB transaction using the pertinent pipe starts. It clears the PBUSY bit from 1 to 0 when one transaction completes successfully. Reading this bit after PID has been set to NAK allows checking that modification of the pipe settings is possible. For details, see 25.4.3 (1), Pipe Control Register Modifying Procedures.

Item	Page	Revision (See Manual for Details)
24.3.36 Pipe Configuration Register (PIPECFG) Table 24.13 CNTMD Bit Setting and Method of Determining Completion of Transmission/Reception for the FIFO Buffer	1230	Table amended

CNTMD Bit Setting	Method of Determining Readiness for Reading and Transmission
0	Condition for the FIFO buffer being ready for reading when set for receiving direction (DIR = 0): • This controller receives one packet.
	Conditions for the FIFO buffer being ready for transmission when set for transmitting direction (DIR = 1): When either of the following conditions is met (1) Data equal to the maximum packet size is written to the FIFO buffer. (2) A short packet of data (or a zero-byte packet) is written to the FIFO buffer and 1 is written to BVAL.
1	Conditions for the FIFO buffer being ready for reading when set for receiving direction (DIR = 0): When one of (1) to (4) below is satisfied (1) The number of received data bytes in the FIFO buffer assigned to the selected pipe has become equal to the allocated number of bytes ([BUFSIZE + 1) × 64). (2) This controller receives a short packet other than a zero-length packet. (3) This controller receives a zero-length packet when data is already stored in the FIFO buffer assigned to the selected pipe. (4) This controller has received packets of the number of times set in the transaction counter for the selected pipe.
	Conditions for the FIFO buffer being ready for transmission when set for transmitting direction (DIR = 1): When one of (1) to (3) below is satisfied (1) The number of data bytes written to the FIFO buffer has become equal to the size of one plane of the FIFO buffer assigned to the selected pipe. (2) Data of the size smaller than the size of one plane of the FIFO buffer assigned to the selected pipe (including 0 bytes) is written to the FIFO buffer and 1 is written to BVAL. (3) During DMA transfer operation with the DMA transfer-end sampling enable bit (TENDE) set to 1, the number of bytes of data written is less than one plane of the FIFO buffer allocated to selected pipe (or zero bytes are written), and a DMA transfer-end signal is received while the last write is in progress.

Item	Page	Revision (See Manual for Details)
24.3.39 Pipe Timing Control Register (PIPEPERI) (2) When the Function Controller Function is Selected Figure 24.3 (Micro) Frames and Expectation for Token Reception when IITV = 0	1239	Figure amended 
Figure 24.4 (Micro) Frames and Expectation for Token Reception when IITV = 1		
	1240	Description amended (a) Power-on reset (b) When ACLRM = 1 is set (c) When USB bus reset is detected

Item	Page	Revision (See Manual for Details)
24.3.40 PIPEn Control Registers (PIPEnCTR) (n = 1 to 5)	1240	Bit table amended

Bit:	15	14	13	12	11	10	9	8
	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR
Initial value:	0	0	0	0	—	0	0	0
R/W:	R	R	R*1/W*2	R	R	R/W	R/W	R*1/W*2

	1242, 1244 to 1246	Table amended

Bit	Bit Name	Initial Value	R/W	Description
13	CSCLR	0	R*1/W*2	C-Split Status Clear Bit Set this bit to 1 to clear the CSSTS bit of the ...
9	ACLRM	0	R/W	Auto Buffer Clear Mode*3 Enables or disables automatic buffer clear mode for the pertinent pipe. To clear the information in the FIFO buffer assigned to the pertinent pipe completely, write 1 and then 0 to this bit successively. Table 24.14 (1) lists the information cleared in this module by successive writing of 0 and 1 to this bit, and table 24.14 (2) lists the cases in which clearing of the information is necessary.
5	PBUSY	0	R	Pipe Busy Indicates whether the pertinent pipe is currently used for the transaction. This bit changes from 0 to 1 upon start of the USB transaction for the pertinent pipe, and changes from 1 to 0 upon normal completion of one transaction. Reading this bit after setting PID to NAK allows confirming that modification of the pipe settings has become possible. 0: The pertinent pipe is not currently used for the transaction. 1: The pertinent pipe is currently used for the transaction.
1, 0	PID[1:0]	00	R/W	This module modifies the setting of these bits as ... Note: After modifying the setting of these bits from BUF to NAK during USB communication using the pertinent pipe, check that PBUSY is 0 to

Table 24.15 (1) Information Cleared by this Module by Setting ACLRM = 1	1248	Table replaced
Table 24.15 (2) Cases That Require Setting ACLRM to 1		

Item | **Page** | **Revision (See Manual for Details)**

24.3.40 PIPEn Control Registers (PIPEnCTR) (n = 1 to 5)

Table 24.17 Operation of This Module depending on PID Setting (when Function Controller Function is Selected)

1249

Table amended

PID Bits	Transfer Type (TYPE Bits)	Transfer Direction (DIR Bit)	Operation of This Module
00 (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host.
	Isochronous	Receiving direction (DIR = 0)	Returns nothing in response to a token from the USB host.
		Transmitting direction (DIR = 1)	Returns a zero-length packet in response to a token from the USB host.

24.3.41 PIPEn Control Registers (PIPEnCTR) (n = 6 to 9)

1251

Bit table amended

Bit:	15	14	13	12	11	10	9	8
	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR
Initial value:	0	-	0	0	-	-	0	0
R/W:	R	R	R*1/ W*2	R	R	R	R/W	R*1/ W*2

1252 to 1255

Table amended

Bit	Bit Name	Initial Value	R/W	Description
13	CSCLR	0	R*1/W*2	C-Split Status Clear Bit Set this bit to 1 to clear the CSSTS bit of the ...
12	CSSTS	0	R	CSSTS Status Bit ... This bit indicates 1 upon start of the C-Split and indicates 0 upon detection of C-Split completion. If the USB device is detached when processing of a C-SPLIT transaction is in progress, the CSSTS bit may remain set to 1. In such a case (DTCH = 1 detected), use the CSCLR bit to clear the CSSTS bit to 0.
9	ACLRM	0	R/W	Auto Buffer Clear Mode*3 ... To clear the information in the FIFO buffer assigned to the pertinent pipe completely, write 1 and then 0 to this bit successively. Table 24.17 (1) lists the information cleared in this module by successive writing of 0 and 1 to this bit, and table 24.17 (2) lists the cases in which clearing of the information is necessary.
5	PBUSY	0	R	Pipe Busy Indicates whether the pertinent pipe is currently used for the transaction. This bit changes from 0 to 1 upon start of the USB transaction for the pertinent pipe, and changes from 1 to 0 upon normal completion of one transaction. Reading this bit after setting PID to NAK allows confirming that modification of the pipe settings has become possible. 0: The pertinent pipe is not currently used for the transaction. 1: The pertinent pipe is currently used for the transaction.
1, 0	PID[1:0]	00	R/W	This module modifies the setting of these bits as ... Note: After modifying the setting of these bits from BUF to NAK during USB communication using the pertinent pipe, check that PBUSY is 0 to ...

Item	Page	Revision (See Manual for Details)
24.3.41 PIPEn Control Registers (PIPEnCTR) (n = 6 to 9) Table 24.18 (1) Information Cleared by this Module by Setting ACLRM = 1 Table 24.18 (2) Cases That Require Setting ACLRM to 1	1256	Table replaced
24.3.43 Transaction Counter Registers (PIPEnTRN) (n = 1 to 5)	1259	Table amended Bit / Bit Name / Initial Value / R/W / Description 15 to 0 / TRNCNT[15:0] / H'0000 / R/W / Transaction Counter* ... 3. The TRCLR bit has been set to 1 when the following conditions are satisfied.
24.4.1 System Control (3) USB Data Bus Resistor Control	1266	Description added … When the function controller function is selected, set the DPRPU bit in the SYSCFG register to 1 after the connection to the USB host is recognized to pull up the D+ signal. When disconnection from the USB host is recognized, manipulate the DPRPU and DCFM bits as follows: 1. Clear the DPRPU bit to 0. 2. Wait for at least 1 µs. 3. Set the DCFM bit to 1. 4. Wait for at least 200 ns. 5. Clear the DCFM bit to 0.

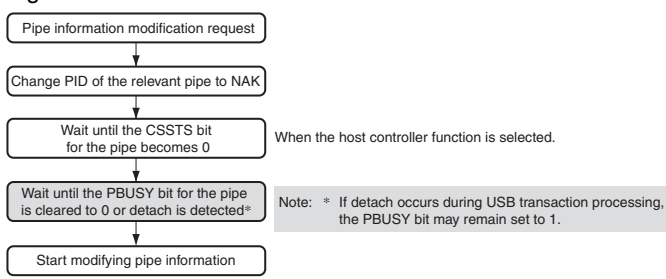
Item	Page	Revision (See Manual for Details)
24.4.1 System Control (4) Register Access Wait Control	1266	Description amended The following restrictions apply to numbers of cycles for access to registers of this module below SYSSTS0. The cycle time for consecutive access to registers of this module must be at least 4 USB clock (48 MHz) cycles (83.33 ns). … Calculation: (2 cycles (access cycle for registers of this module) + 1 cycle (interval between consecutive access operations) + BWAIT) × 1/66 MHz × 83.33 ns BWAIT = 3 … Calculation: (2 cycles (access cycle for registers of this module) + 2 cycles (access cycle for internal memory) + BWAIT) × 1/66 MHz × 83.33 ns BWAIT = 2
24.4.2 Interrupt Functions (1) Interrupt Control Overview Table 24.21 Interrupt Generation Conditions	1269	Table amended

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
NRDY	Buffer not ready interrupt	... When the function controller function is selected: • When a token is received while the PID bits are set to BUF and transmission is not enabled for the buffer memory • When a CRC error or a bit stuffing error occurred during data reception in isochronous transfer • When an interval error occurred during data reception in isochronous transfer	Host, function	NRDYSTS. PIPENRDY
BRDY	Buffer ready interrupt	When the buffer is ready (reading or writing is enabled)	Host, function	BRDYSTS. PIPENRDY
BCHG	Bus change interrupt	When a change of USB bus state is detected	Host	

(3) Control Transfer Stage Transition Interrupt (Function Controller Function)	1274	Description amended (This module retains the setup stage end, and after the interrupt status has been cleared by software, a CTRT interrupt is generated.)

Item	Page	Revision (See Manual for Details)
24.4.3 Pipe Control Table 24.22 Pipe Setting Items	1276, 1277	Table amended

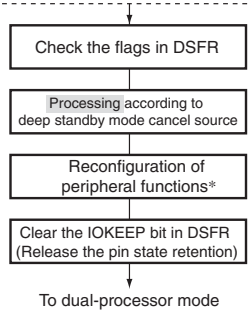
Register Name	Bit Name	Setting Contents	Remarks
DCPCFG PIPECFG	CNTMD	Selects continuous transfer or non-continuous transfer	DCP: Can be set PIPE1 and PIPE2: Can be set (only when bulk transfer has been selected). PIPE3 to PIPE5: Can be set With continuous transmission and reception, the buffer size should be set to an integer multiple of the payload.
	SHTNAK	Selects disabled state for pipe when transfer ends	DCP: Can be set PIPE1 and PIPE2: Can be set (only when bulk transfer has been selected) PIPE3 to PIPE5: Can be set
PIPEPERI	IFIS	Buffer flush	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE9: Cannot be set
	IITV	Interval counter	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE5: Cannot be set PIPE6 and PIPE9: Can be set (only when host controller function has been selected)

(1) Pipe Control Register Modifying Procedures Pipe Control Register Modifying Procedures	1279	Description amended • Bits in PIPEnTRE and PIPEnTRN • Bits in DEVADDn Note: In addition to the above, observe the setting procedures described in the register descriptions regarding the settings of the CSCLR bit and DEVADDn register.
Figure 24.8 Procedure for Changing Pipe Information from a USB Communication Enabled State (PID = BUF)	1279	Figure amended  • Bits in DCPCFG and DCPMAXP • Bits in PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI • The ACLRM bit in PIPEnCTR

<table>
<tr><th>Item</th><th>Page</th><th>Revision (See Manual for Details)</th></tr>
<tr><td>24.4.3 Pipe Control
(2) Maximum Packet Size Setting</td><td>1280</td><td>Newly added</td></tr>
<tr><td>(3) Response PID</td><td>1280 to 1282</td><td></td></tr>
<tr><td>(4) Data PID Sequence Bit</td><td>1282</td><td></td></tr>
<tr><td>24.4.4 FIFO Buffer
(1) FIFO Buffer Allocation</td><td>1283</td><td>Description amended
… When continuous transfer mode has been selected using the CNTMD bit in PIPE CFG, the BUFSIZE bits should be set so that the buffer memory size should be an integral multiple of the maximum packet size. When double buffer mode has been selected using the DBLB bit in PIPE CFG, two planes of the memory area specified using the BUFSIZE bits in PIPEBUF can be assigned to a single pipe.</td></tr>
<tr><td>(3) FIFO Port Functions
(a) FIFO Port Selection</td><td>1286</td><td>Description amended
The buffer memory access direction is in accord with the ISEL bit setting for the DCP, and in accord with the DIR bit in PIPE CFG for other pipes.</td></tr>
<tr><td>Table 24.25 FIFO Port Access Categorized by Pipe</td><td></td><td>Table amended
<table>
<tr><th>Pipe</th><th>Access Method</th><th>Port that can be Used</th></tr>
<tr><td>DCP</td><td>CPU access</td><td>CFIFO port register</td></tr>
<tr><td>PIPE1 to PIPE7</td><td>CPU access</td><td>CFIFO port register</td></tr>
<tr><td></td><td>DMA access</td><td>D0FIFO/D1FIFO port register</td></tr>
</table></td></tr>
<tr><td>24.4.5 Control Transfers (DCP)
(1) Control Transfers when the Host Controller Function is Selected
(b) Data Stage</td><td>1293</td><td>Description amended
Transaction is done by setting the data PID = DATA1 and the PID bit = BUF using the SQSET bit in DCPCTR. Completion of data transfer is detected using the BRDY or BEMP interrupts.</td></tr>
<tr><td>(2) Control Transfers when the Function Controller Function is Selected
(b) Data Stage</td><td>1295</td><td>Description amended
… Before accessing the DCP buffer memory, the access direction should be specified using the ISEL bit in CFIFOSEL.
A transaction is executed by setting the PID bits in the DCPCTR register to BUF. The BRDY interrupt or the BEMP interrupt can be used to detect the end of data transfer. Use the BRDY interrupt to detect the end of control write transfers and the BEMP interrupt to detect the end of control read transfers.</td></tr>
</table>

Item	Page	Revision (See Manual for Details)
24.4.5 Control Transfers (DCP) (2) Control Transfers when the Function Controller Function is Selected (d) Control Transfer Auto Response Function	1295	Description amended 1. bmRequestType ≠ H'00 2. wIndex ≠ H'00 3. wLength ≠ H'00 4. wValue > H'7F 5. DVSQ = 011 (Configured)
24.4.6 Bulk Transfers (PIPE1 to PIPE5) (1) PING Packet Control when the Host Controller Function is Selected	1296	Newly added
(2) NYET Handshake Control when the Function Controller Function is Selected	1296, 1297	
24.4.8 Isochronous Transfers (PIPE1 and PIPE2) (1) Error Detection with Isochronous Transfers	1299 to 1301	Newly added
(2) DATA-PID	1301	
(3) Interval Counter (b) Interval Counter Initialization when the Function Controller Function is Selected	1302	Description amended • Buffer memory initialization using the ACLRM bit The IITV bits are not initialized but the count value is. • USB bus reset
(4) Setup of Data to be Transmitted using Isochronous Transfer when the Function Controller Function is Selected	1304	Description amended Figure 24.14 shows an example of transmission using the isochronous transfer transmission data setup function with this module, when IITV = 0 (every frame) has been set.
Figure 24.14 Example of Data Setup Function Operation		Figure replaced

Item	Page	Revision (See Manual for Details)
24.4.8 Isochronous Transfers (PIPE1 and PIPE2) (5) Isochronous Transfer Transmission Buffer Flush when the Function Controller Function is Selected Figure 24.15 Example of Buffer Flush Function Operation	1305	Figure replaced
Figure 24.16 Example of an Interval Error Being Generated when IITV = 1	1306	
24.4.9 SOF Interpolation Function	1307	Description amended • Refreshing of the frame number and the micro-frame number • SOFR interrupt and μSOF lock • Isochronous transfer interval count
29.2.3 Data Retention	1549	Newly added
30.2.2 Standby Control Register 2 (STBCR2)	1556	Description amended STBCR2 is an 8-bit readable/writable register that controls the operation of individual modules .
30.2.3 Standby Control Register 3 (STBCR3)	1557	Description amended STBCR3 is an 8-bit readable/writable register that controls the operation of modules .
30.2.4 Standby Control Register 4 (STBCR4)	1559	Description amended STBCR4 is an 8-bit readable/writable register that controls the operation of modules .
30.2.5 Standby Control Register 5 (STBCR5)	1560	Description amended STBCR5 is an 8-bit readable/writable register that controls the operation of modules .
30.2.6 Standby Control Register 6 (STBCR6)	1561	Description amended STBCR6 is an 8-bit readable/writable register that controls the operation of modules .

Item	Page	Revision (See Manual for Details)
30.2.7 Standby Control Register 7 (STBCR7)	1563	Description amended STBCR7 is an 8-bit readable/writable register that controls the operation of modules .
30.3.7 Deep Standby Mode (2) Canceling Deep Standby Mode Figure 30.4 Flowchart of Canceling Deep Standby Mode	1589	Figure amended 
(4) Note on Transition to Deep Standby Mode	1592	Newly added
32.2 Register Bits	1745	Table amended

Module Name	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
USB	USBLENG	WLENGTH[15:8]							
		WLENGTH[7:0]							
	DCPCFG	—	—	—	—	—	—	—	CNTMD
		SHTNAK	—	—	DIR	—	—	—	—

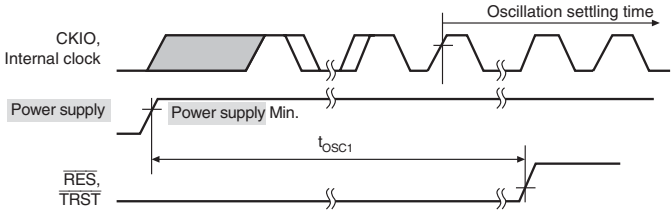
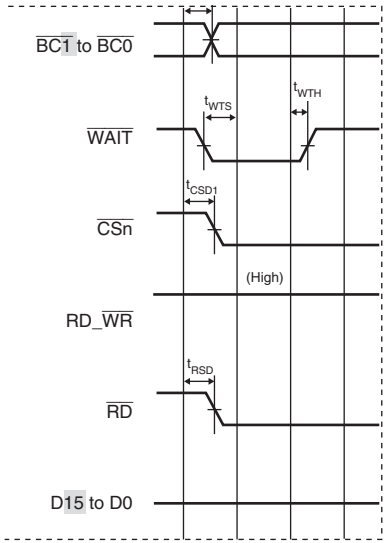
32.3 Register States in Each Operating Mode	1760, 1762	Table and note amended

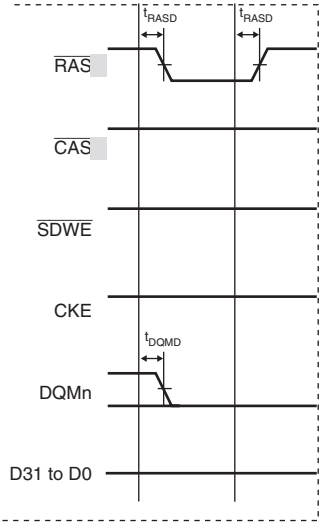
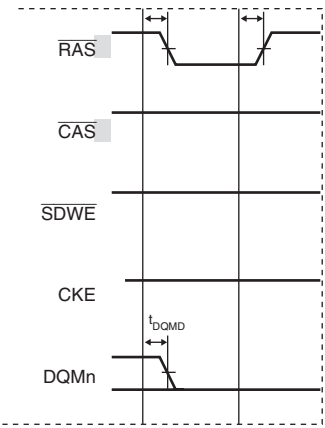
Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Deep Standby	Software Standby	Module Standby	Sleep
WDT	WTCSR0	Initialized	Initialized	Initialized	Retained	—	Retained
	WTCNT0	Initialized	Initialized	Initialized	Retained	—	Retained
	WRCSR0	Initialized*1	Retained	Initialized	Retained	—	Retained
	WTCSR1	Initialized	Initialized	Initialized	Retained	—	Retained
	WTCNT1	Initialized	Initialized	Initialized	Retained	—	Retained
	WRCSR1	Initialized*1	Retained	Initialized	Retained	—	Retained

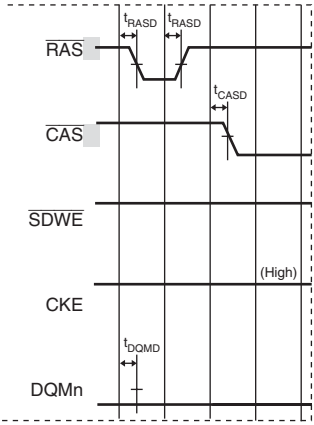
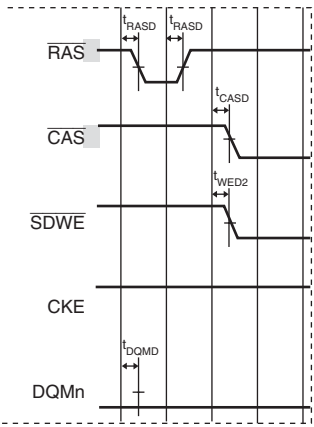
Notes: 5. Bits BC2 to BC0 are initialized.

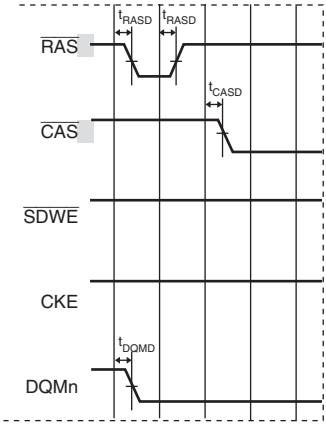
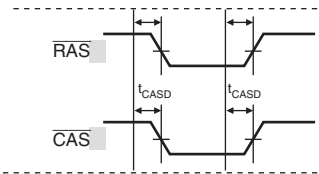
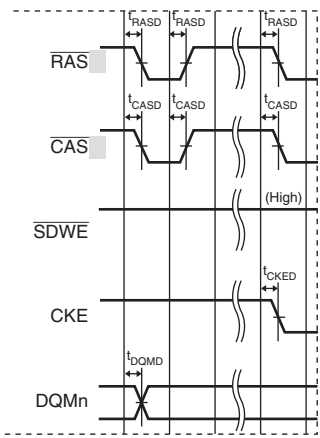
33.4 AC Characteristics Table 33.5 Operating Frequency	1775	Table and title amended

Item		Symbol	Min.	Max.	Unit	Remarks
Operating frequency	CPU0 clock($I_0\phi$)	f	40.00	200.00	MHz	
	CPU1 clock($I_1\phi$)		40.00	200.00	MHz	
	Bus clock(Bϕ)		40.00	66.66	MHz	
	Peripheral clock (Pϕ)		10.00	33.33	MHz	

Item	Page	Revision (See Manual for Details)
33.4.1 Clock Timing Figure 33.5 Power-On Oscillation Settling Time	1778	Figure amended 
33.4.3 Bus Timing Figure 33.15 External Address Space: Timing with External Wait (Page Read Access to 16-Bit Width Channel, External Read Data Continuous Assert Mode, Read Cycle Wait = 3, Page Read Cycle Wait = 3, Other Wait Settings = 0, External Wait Cycles = 2)	1790	Figure amended 

Item	Page	Revision (See Manual for Details)
33.4.3 Bus Timing Figure 33.16 SDRAM Space: Single-Read Bus Timing (DCL = 2 (2 Cycles), DRCD = 1 (2 Cycles), DPCG = 1 (2 Cycles))	1791	Figure amended 
Figure 33.17 SDRAM Space: Single-Write Bus Timing (DCL = 2 (2 Cycles), DRCD = 1 (2 Cycles), DPCG = 1 (2 Cycles))	1792	Figure amended 

Item	Page	Revision (See Manual for Details)
33.4.3 Bus Timing Figure 33.18 SDRAM Space: Multiple-Read Bus Timing (4 Data Accesses, DCL = 2 (2 Cycles), DRCD = 1 (2 Cycles), DPCG = 1 (2 Cycles))	1793	Figure amended 
Figure 33.19 SDRAM Space: Multiple-Write Bus Timing (4 Data Accesses, DCL = 2 (2 Cycles), DRCD = 1 (2 Cycles), DPCG = 1 (2 Cycles))	1794	Figure amended 

Item	Page	Revision (See Manual for Details)
33.4.3 Bus Timing Figure 33.20 SDRAM Space: Bus Timing of Multiple-Read Across Rows (8 Data Accesses, DCL = 2 (2 Cycles), DRCD = 1 (2 Cycles), DPCG = 1 (2 Cycles))	1795	Figure amended 
Figure 33.21 SDRAM Space: Mode Register Set Bus Timing	1796	Figure amended 
Figure 33.22 SDRAM Space: Self-Refresh Bus Timing	1797	Figure amended 

Item	Page	Revision (See Manual for Details)
33.4.10 IIC3 Timing Table 33.15 (1) IIC3 Timing I^2C Bus Format	1807	Table and title amended **Item** / **Symbol** / **Min.** / **Max.** / **Unit** / **Figure** SCL input cycle time / t_{SCL} / 12 t_{icyc}*1 + 600 / — / ns / Figure 33.35 (1) SCL input high pulse width / t_{SCLH} / 3 t_{icyc}*1 + 300 / — / ns
Figure 33.35 (1) IIC3 Input/Output Timing	1808	Figure title amended
Table 33.15 (2) IIC3 Timing Clock Synchronized Serial Format		Table added
Figure 33.35 (2) Clock Input/Output Timing	1809	Figure added
Figure 33.35 (3) Transmission and Reception Timing		

Index

D

E

F

N

O

P

Q

R

S

T

U

V

W

SH7205 Group User's Manual: Hardware

Publication Date: 1st Edition, March, 2008
Rev.3.00, March 27, 2014

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

SH7205 Group
User's Manual: Hardware

Renesas Electronics Corporation

R01UH0473EJ0300
(Previous Number: REJ09B0372-0200)

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Renesas Electronics:

R5S72050W200BG